**Technologie sieciowe**

**Lista 1**

Maurycy Sosnowski (261705)

**Zadania:**

Napisz sprawozdanie zawierające: **opis programów, wywołania dla poniższych zagadnień z analizą wyników, wnioski dotyczące przydatności tych programów.**

1. Sprawdź za pomocą pingu ile jest węzłów na trasie do (i od) wybranego, odległego geograficznie, serwera.  
   Uwaga: trasy tam i z powrotem mogą być różne.
2. Zbadaj jaki wpływ ma na to wielkość pakietu. Zbadaj jak wielkość pakietu wpływa na obserwowane czasy propagacji. Wyniki sprawdź dla różnych adresów DNS. Jeśli potrzeba, sporządź wykres.
3. Zbadaj jaki wpływ na powyższe ma konieczność fragmentacji pakietów. Jaki największy niefragmentowany pakiet uda się przesłać. Dlaczego i od czego to zależy?
4. Przeanalizuj te same zagadnienia dla krótkich tras (do serwerów bliskich geograficznie).
5. Określ "średnicę" internetu (najdłuższą ścieżkę, którą uda się wyszukać).
6. Czy potrafisz wyszukać trasy przebiegające przez sieci wirtualne (zdalne platformy "cloud computing"). Ile węzłów mają ścieżki w tym przypadku?
7. Porównaj otrzymane wyniki przy pomocy Traceroute.
8. Do czego służy WireShark? Jak można tą funkcjonalność odnieść do Ping lub Traceroute.

**Opisy programów:**

Obraz zawierający tekst

Opis wygenerowany automatycznie**Ping** -narzędzie umożliwiające sprawdzenie czy istnieje połączenie między naszym urządzeniem a wybranym serwerem. Generuje ono pakiet/wiadomość o określonych parametrach i wysyła na wybrany serwer, a gdy otrzyma odpowiedź zwraca dane o połączeniu (np. średni czas podróży wiadomości, informacje o zagubionych pakietach).

*(Przykładowe wywołanie ping dla google.com)*

Przykładowe parametry/flagi (Windows):

-*f* -zapobiega fragmentacji

-*l [liczba]* -określa rozmiar pakietu

*-n [liczba]*  -określa ilość wysyłanych żądań

-*w [czas] -*określa czas oczekiwania na odpowiedź

*-i [liczba]*-określa TTL (time to leave)

**Traceroute** - narzędzie umożliwiające śledzenie/badanie trasy przesyłanych pakietów do wybranego serwera. Podaje informacje węzłach (hopach) jakie musi pokonać pakiet podczas drogi na serwer. W systemie Windows odpowiednikiem jest komenda **tracert**.

Przykładowe parametry/flagi (Windows):

*-h [liczba]* -określa maksymalną liczbę hopów

*-w [czas]* -określa czas oczekiwania na każdą odpowiedź

Obraz zawierający tekst

Opis wygenerowany automatycznie

*(Przykładowe wywołanie tracert dla google.com)*

Obraz zawierający tekst

Opis wygenerowany automatycznie**Wireshark** – dawniej Ethereal, bezpłatne oprogramowanie opensource do analizowania pakietów oraz monitorowania ruchu sieciowego. Umożliwia też np. śledzenie, przechwytywanie, nagrywanie i dekodowanie pakietów. Dzięki dodatkom potrafi też rozpoznawać i dekodować protokoły komunikacyjne.

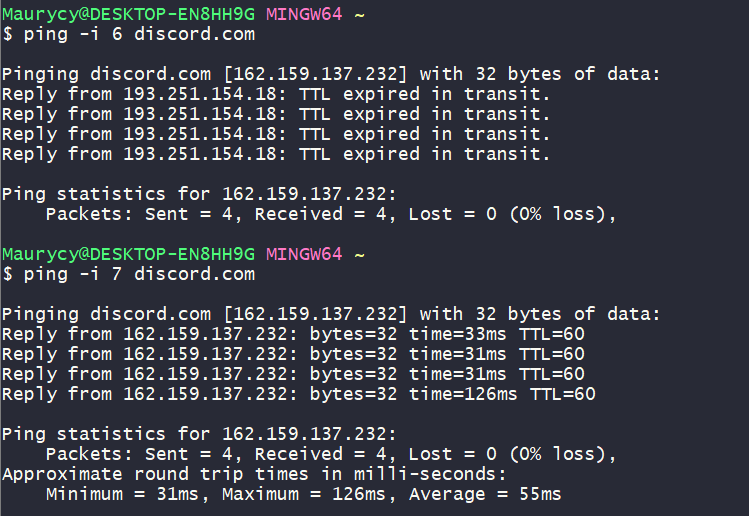
*(Przykładowe uruchomienie Wireshark dla mojego Wi-Fi)*

**Rozwiązania zadań:**

**Zadanie 1 (sprawdzanie ilości węzłów na trasie):**

Odległość **do** wybranego serwera przy użyciu pingu można sprawdzić stopniowo zwiększając TTL (przy użyciu flagi -i) do momentu nawiązania połączenia. Podane przez nas TTL jest wartością która określa maksymalną liczbę hopów jaką może wykonać pakiet (jest to równoważne maksymalnej dozwolonej ilości węzłów przez które może przejść) zatem znajdując minimalne TTL dla którego otrzymamy odpowiedź od serwera możemy określić ilość węzłów na trasie.

Odległość **od** wybranego serwera możemy odczytać wywołując ping i analizując podane zwrotne TTL. Standardowo na początku ma ono wartość równą 32, 64, 128 lub 255 i zmniejsza się o jeden po każdym przebytym w drodze powrotnej węźle.



*Przykładowo analizując flagę ‘ -i ’ z powyższego obrazka możemy wywnioskować że na trasie do podanego serwera mamy 7 węzłów, natomiast obserwując zwrotne TTL możemy zauważyć że wracając pakiet pokonał 64 – 60 = 4 węzły.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Serwer** | **lokalizacja** | **Trasa do serwera** | **Trasa od serwera** |
| yahoo.co.jp | Japonia | 14 węzłów | 15 węzłów |

*Powyżej sprawdzenie ilości węzłów na trasie do odległego geograficznie serwera.*

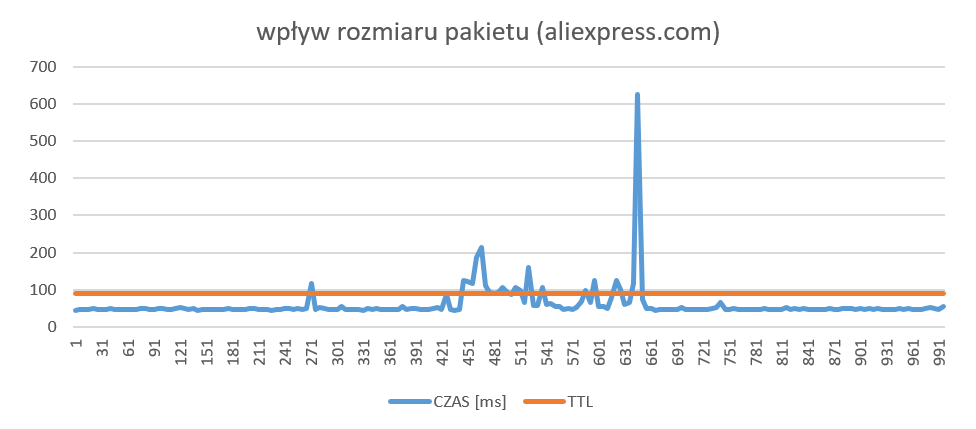
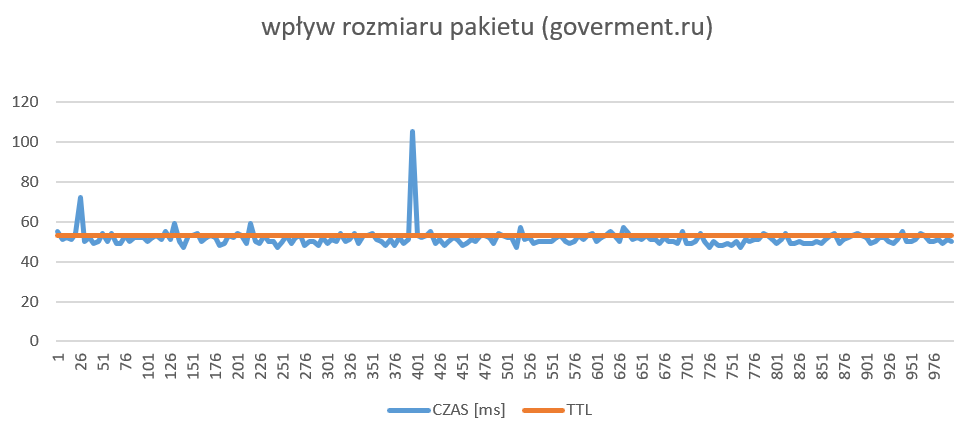
**Zadanie 2 (sprawdzanie znaczenia wielkości pakietu):**

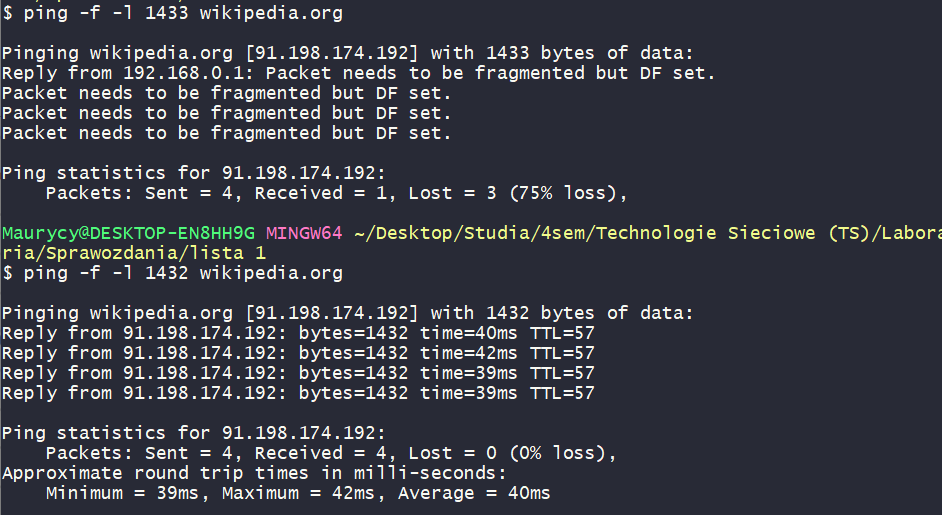
Obraz zawierający tekst

Opis wygenerowany automatycznie*Przy pomocy poniższego skryptu odczytuje wpływ rozmiaru pakietów (nie fragmentowanych) z przedziału od 1 do 1000 bajtów, na czas propagacji i ilość skoków (dokładniej TTL który jest bezpośrednio związany z ilością skoków). Otrzymane wartości zapisuję do plików .csv na podstawie których generuję poniższe wykresy.*

Obraz zawierający tekst

Opis wygenerowany automatycznie

Na podstawie powyższych wykresów można zauważyć, że wielkość pakietu nie ma wpływu na ilość skoków. Jeżeli chodzi o czas propagacji to można zaobserwować spore skoki wartości dla niektórych pakietów, jednak najprawdopodobniej powstały one na skutek innych czynników gdyż po ponownym sprawdzeniu czas propagacji dla tych pakietów utrzymywał się blisko średniej. Biorąc to pod uwagę można uznać, że rozmiar pakietu nie ma również dużego wpływu na czas propagacji.

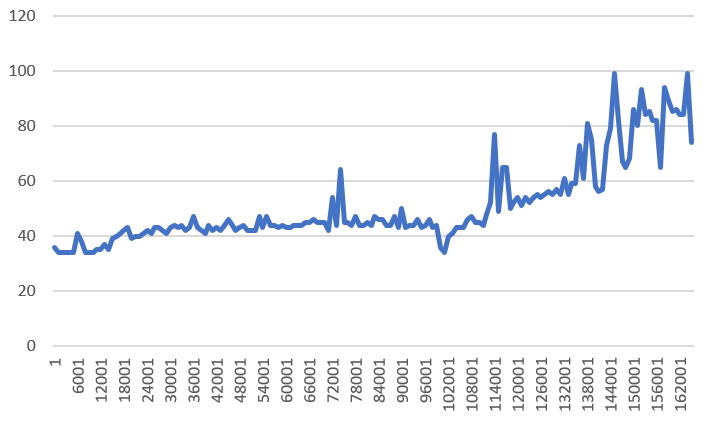
**Zadanie 3 (fragmentacja pakietów):**

Jak widać na obrazku obok, największy niefragmentowany pakiet jaki udało mi się wysłać miał wielkość 1432b. Wartość ta jest zależna od MTU (Maximum Transmission Unit).

Obraz zawierający tekst

Opis wygenerowany automatycznieAnalizując wykres obok, z wynikami testów na wpływ wielkości pakietu (bez flagi zapobiegającej fragmentacji), możemy zauważyć, że czas propagacji praktycznie nie uległ zmianie (jedyną różnicą jest brak nagłych skoków dla niektórych pakietów). Ilość skoków również nie uległa zmianie.

*Powyżej wykres wpływu wielkości pakietu na czas propagacji bez flagi ‘ -f ’ dla aliexpress.com (Serie1 to czas[ms], Serie2 to TTL)*



Korzystając z fragmentacji możemy znacznie zwiększyć rozmiar pakietu dzięki czemu można wykonać testy z poprzedniego zadania na zdecydowanie większą skalę. Dzięki szerszej perspektywie możemy zaobserwować, że czas propagacji istotnie powoli wzrasta wraz ze zwiększaniem pakietu (*powyżej testy wpływu wielkości pakietu dla wikipedia.org*).

**Zadanie 4 (Bliskie geograficzne serwery):**

Uruchamiając ping i traceroute dla serwerów w różnych odległościach geograficznych możemy zaobserwować, że zazwyczaj dla ilość potrzebnych skoków wzrasta przy większej odległości geograficznej, co istotnie jest dość intuicyjne. (przykład poniżej)

|  |  |  |  |
| --- | --- | --- | --- |
| **Serwer** | **lokalizacja** | **Trasa do serwera** | **Trasa od serwera** |
| yahoo.co.jp | Japonia | 14 węzłów | 15 węzłów |
| edukacja.pwr.wroc.pl | Polska | 12 węzłów | 11 węzłów |

**Zadanie 5 (Średnica internetu):**

Jest to największa liczba skoków/trasa pakietu. U mnie taką wartością była trasa z serwera aliexpress.com, która wyniosła 128 – 90 = 38 skoków. Obraz zawierający tekst

Opis wygenerowany automatycznie

**Zadanie 6 (Sieci wirtualne):**

Sieci wirtualne możemy rozpoznać stosując ping kilkukrotnie z odstępem czasowym i obserwując odpowiedzi z różnych adresów IP lub różnice w wartości TTL. Modyfikując wartość TTL sieci takie są znacznie trudniejsze do śledzenia i ciężko określić długość ścieżki.

**Zadanie 7 (Wyniki z traceroute):**

Wyniki z zadania 2 i 4 (trasa do serwera), otrzymane używając komendy ping i metodologii opisanej w zadaniu 2, są zgodne z wynikami otrzymanymi przy użyciu traceroute.

**Zadanie 8 (Wireshark):**

Opis Wiresharka znajduje się w zadaniu 1. Podobnie jak ping czy traceroute narzędzie to służy do monitorowania i analizy sieci, jednak jest zdecydowanie bardziej rozbudowane. Jeżeli chodzi o różnice to Wireshark, w przeciwieństwie do pingu i tracerouta, wykonuje analizę na bazie przechwyconych pakietów (sam pakietów nie wysyła).

**Podsumowanie:**

Powyższe programy są przydatne do szeroko pojętej analizy sieci. Pomagają one zrozumieć strukturę sieci i zasady jej działania. Dzięki umiejętnemu ich wykorzystaniu możemy np. sprawdzać czy istnieje połączenie między danymi serwerami, jaka jest prędkość łącza, jak wygląda trasa przesyłanych pakietów, co wpływa na czas ich propagacji. Informacje takie mogą być bardzo przydatne podczas wstępnej analizy np. gdy chcemy zdiagnozować co jest przyczyną problemów z połączeniem.