Technologie Sieciowe - lista pierwsza

Bartosz Rajczyk

14 marca 2019

1 Wstęp

Sprawozdanie to porusza kwestie podstawowej architektury sieciowej i jej badania przy pomocy polecania `ping`, `tracert` oraz programu Wireshark. Analizuję w nim odległość w węzłach do serwerów w różnych miejscach świata, szukam przekroju światowej sieci, sprawdzam maksymalne rozmiary niefragmentowalnych pakietów oraz zmianę ich ścieżek zależnie od wielkości.

1.1 Opis narzędzi

1.1.1 ping

`ping` służy do wysyłania sygnału echo do serwera docelowego. Generuje on pakiet o okreslonych parametrach i wysyła pod zadany adres, a kiedy (i jeżeli) otrzyma odpowiedź, zwraca jej parametry tekstowo w konsoli. Przykładowe wywołanie:

```
Pinging google.com [216.58.215.78] with 32 bytes of data:
Reply from 216.58.215.78: bytes=32 time=136ms TTL=52
Reply from 216.58.215.78: bytes=32 time=106ms TTL=52
Reply from 216.58.215.78: bytes=32 time=97ms  TTL=52
Reply from 216.58.215.78: bytes=32 time=211ms TTL=52

Ping statistics for 216.58.215.78:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 97ms, Maximum = 211ms, Average = 137ms
```

Otrzymujemy więc dane o adresie, czasie w którym dane przechodziły oraz parametrze TTL pakietów zwrotnych. TTL określa maksymalną ilość węzłów, przez którą może przejść jeszcze pakiet. Podstawowymi wartościami są tu zwykle 64, 128 i 255. Po każdym przekazaniu pakietu, węzeł przekazujący dekrementuje ją, a kiedy osiągnie 0 - nie jest przekazywany dalej. Ustawiając ją w naszych zapytaniach przy pomocy parametru $-$i $<$ttl$>$ i wywołując polecenie ping aż do otrzymania odpowiedzi, możemy zbadać, jak daleką drogę potrzebuje przejść pakiet wysyłany przez nas, zanim dotrze do serwera docelowego. Możemy także

sprecyzować rozmiar wysyłanych danych, aby sprawdzić, jak wpływa to na ilość skoków czy fragmentację.

1.2 tracert

Traceroute lub tracert pokazuje ścieżkę, jaką przebywają pakiety wysyłane od nas do docelowego serwera. W tym celu program wysyła pakiety z inkrementowanymi TTL, dzięki czemu kolejne routery na ścieżce pakietów odrzucają je - w ten sposób dostajemy informacje o tym, jaki router odrzucił pakiet i wiemy, że normalnie musiałby przejść dalej. Przykładowe wywołanie:

```
traceroute to wcss.pl (156.17.193.248), 30 hops max, 60 byte packets
 1  * * *
 2  primary.cat.forpsi.net (81.2.193.3)  4.437 ms  4.546 ms
4.661 ms
 3  sk.r242-six.forpsi.net (81.2.192.242)  8.341 ms  8.766 ms
8.889 ms
 4  Sanet-gw.six.sk (192.108.148.10)  8.320 ms  8.355 ms
8.381 ms
 5  ZU-Zilina.sanet2.sk (194.160.8.11)  11.163 ms  11.189 ms
11.209 ms
 6  poznan-gw1.10Gb.rtr.pionier.gov.pl (194.160.126.82)  22.103 ms
22.053 ms  22.015 ms
 7  z-poznan-gw3.wroclaw.10Gb.rtr.pionier.gov.pl (212.191.224.106)
26.460 ms  26.442 ms  26.429 ms
 8  fw1-vsys3-primary.wcss.wroc.pl (156.17.252.136)  27.161 ms
27.103 ms  27.133 ms
 9  www.wcss.pl (156.17.193.248)  27.125 ms  27.115 ms  27.158 ms
```

1.3 WireShark

WireShark to program służący do monitorowania ruchu sieciowego na naszym urządzeniu. Wyświetla on w kolejności chronologicznej pakiety wysyłane i odbierane przez któreś z używanych w naszym systemie urządzeń sieciowych, analizuje je, wyświetla podstawowe informacje (czas, źródło, cel, protokół, długość itd.) oraz pozwala przeczytać treść pakietów. Przy jego pomocy możemy monitorować cały ruch sieciowy generowany przez system, a przy pomocy zestawu flag do filtrowania możemy określić, które wydarzenia w sieci nas konkretnie interesują oraz podejrzeć ich treść.

2 Testy praktyczne

2.1 Odległości serwerów

Używając metodologii opisanej w sekcji ping, sprawdziłem kilka różnych serwerów otrzymując następujące rezultaty.

serwer	**lokalizacja**	**od nas**	**od nich**	**lat.**
wcss.pl	Wrocław	7	6	5 ms
pg.edu.pl	Gdańsk	9	7	15 ms
bartor.net	Czechy	12	10	32 ms
fu-berlin.de	Niemcy	14	12	36 ms
harvard.edu	Stany Zjednoczone	10	9	24 ms
kyoto-u.ac.jp	Japonia	13	13	188 ms
sydney.edu.au	Australia	24	28	500 ms
victoria.ac.nz	Nowa Zelandia	25	26	450 ms
en.sjtu.edu.cn	Chiny	30	26	480 ms

Tablica 1: sprawdzone wartości TTL

2.2 Wielkość pakietów

Sprawdziłem teraz wpływ wielkości pakietu na ich ścieżkę przy różnych serwerach.

serwer	**rozmiar**	**odpowiedź**	**TTL**	**czas**
wcss.pl	32	32	246	23 ms
wcss.pl	256	256	246	27 ms
wcss.pl	1432	1432	246	20 ms
wcss.pl	1433	timeout	timeout	timeout
harvard.edu	32	32	55	42 ms
harvard.edu	256	256	55	41 ms
harvard.edu	1432	1432	55	40 ms
harvard.edu	1433	timeout	timeout	timeout
google.com	32	32	55	19 ms
google.com	256	64	55	20 ms

Tablica 2: wielkości pakietów

Wartości brzegowe w tabeli (1432 bajty) zostały osiągnięte dla polecenia uruchamianego z konsoli Powershell na Windowsie 10 u mnie w domu; na serwerze z Ubuntu 18.04 w innej sieci wartością tą było 1472 w każdym przypadku.

2.3 Fragmentacja

Test został przeprowadzony na moim serwerze pod adresem bartor.net, gdyż tylko on pozwalał mi na pingowanie się z pofragmentowanymi pakietami. Fragmentację pakietów podglądałem programem WireShark, a zapytania wysyłałem przy pomocy `ping`.

rozmiar	**n fragmentów**	**TTL**	**czas**
32	1	55	41
1500	2	55/55	52
3000	3	55/55/55	55
4500	4	55/55/55/55	48
14792	10	55/../55	55
14793	timeout	timeout	timeout

Tablica 3: fragmentacja pakietów

3 Wnioski

3.1 Odległości

Nie jest wielkim zaskoczeniem spostrzeżenie, że bliższe geograficznie serwery łatwiej jest (przy użyciu mniejszej ilości skoków) osiągnąć połączeniem internetowym. Jeden test wymyka się jednak z tego schematu; jest to serwer uniwersytetu Harvarda w Stanach Zjednoczonych. Jego odległość w skokach była mniejsza niż testowanych Niemiec czy chociażby Czech, które to kraje bezpośrednio graniczą z Polską. Może być to świadectwem istnienia względnie niedaleko nas dalekiego, transatlantycznego węzła komunikacyjnego, który łączy nas bezpośrednio z dalekim zachodem; natomiast połączenia z bliższymi krajami są realizowane bardziej pośrednio, przy użyciu istniejącej między nimi infrastruktury. Przekrój sieci udało mi się wyznaczyć na okolice 24 - 26 skoków; jest to największa ilość węzłów na drodze pakietów wysyłanych do Nowej Zelandii, a więc po przeciwnej stronie globu. Ciekawy rezultat (chociaż nie tak ciekawy, jak miałem nadzieję) dały Chiny, które mimo tego że są bliżej nas, kazały pakietowi podróżować jeszcze dalej (w ich stronę), natomiast odsyłały pakiety z nadal dużą (bo równą tej z Nowej Zelandii) liczbie skoków, lecz już wyraźnie mniejszą, niż musieliśmy wykonać my do nich. Może być to związane z istnieniem tzw. Projektu Złota Tarcza, gigantycznego firewalla filtrującego ruch sieciowy wchodzący i wychodzący z Chin.

3.2 Wielkości a czasy

W testach przeprowadzonych przeze mnie nie udało mi się bezpośrednio powiązać rozmiaru pakietu z czasem jego propagacji do serwera docelowego i odpowiedzi, nie zmieniała się także jego trasa (TTL we wszystkich testach z różnym rozmiarem wynosił zawsze tyle samo). Na ich podstawie wyciągnąłbym wniosek, że parametr ten rzeczywiście nie ma wpływu na realny przebieg pakietu.

3.3 Fragmentacja

Program WireShark okazał się wyjątkowo pomocny w analizowaniu sposobu wysyłania pofragmentowanych danych oraz treści samych pakietów. Po pierwsze,

powyżej 1472 bajtów rozmiaru w poleceniu `ping` (1514 według programu WireShark) pakiety były rozbijane na mniejsze częsci. Zgodnie z logami programu, najpier wysyłany był pakiet protokołu IPv4 nazywany Fragmented IP Protocol zawierający identyfikator oraz offset, a następnie właściwy pakiet protokołu ICMP (Internet Control Message Protocol) z informacją o pingu. Kolejne pakiety w serii posiadają ten sam identyfikator oraz wyższe offsety, a ich główną częścią są dane (nota bene danymi w echo są kolejne znaki ASCII, zapętlające się). Odpowiedź wygląda tak samo.

WireShark wskazuje nawet pakiet, w którym wiadomość została złożona- np. `[Reassembled in #80986]`. Warte wspomnienia jest to, że każdy "fragment"posiada identyczny TTL w moich wszystkich testach, co sugerowałoby również, że praktycznie cała wiadomość (pomimo bycia rozbitą na różne pakiety) porusza się tą samą trasą.

3.4 Dodatkowe obserwacje

Używając polecenia `curl` z PowerShella możemy (używając protokołu HTTP czy HTTPS) "poprosićśerwer o udostępnienie nam jakiegoś zasobu. To pozwala nam także zaobserwować, jak dużo pakietów wysyłanych jest pomiędzy serwerem a komputerem, kiedy np. ładujemy przeciętną stronę internetową. Załadowanie strony głównej (bez plików wyświetlanych w przeglądarce - sam HTML, pliki ładowane w przeglądarce osobnymi zapytaniami) na moim serwerze wymagało przesłania stąd i z powrotem łącznie około 20 pakietów. Przy ładowaniu strony głównej onet.pl było to prawie 1000 pakietów (ze względu na długość ładowanego zasobu).

4 Podsumowanie

Przeprowadzone testy, wyciągnięte wnioski i użyte narzędzia dały mi ogólny wgląd w sposób działania sieci, zarządzania w niej pakietami oraz połączeniami.