raport SIK

Michał Zmyślony

April 2024

1 Plan

Badanie efektywności programów bedzie podzielone na pare sekcji:
1) testowane lokalnie
2) testowanie na dwóch maszynach wirtualnych w tej samej sieci
3) testowanie na dwóch maszynach wirtualnych w tej samej sieci z właczonym opóźnieniem 50ms
4) testowanie na dwóch maszynach wirtualnych w tej samej sieci z właczonym gubieniem pakietów 25%

w każdym z warunków testowych bedzie uruchamiany ten sam skrypt ,który 5 razy bedzie powtarzać procedure. Procedura dla każdego z protokołów (udp, udpr, tcp) sprawdza szybkość i poprawność przesyłanego pliku (polecenie diff). Testowane jest 5 rozmiarów plików: 1B, 26KB, 2,8MB, 14MB oraz 110MB. Dodatkowo każda cała operacja jest powtarzana dla różnych długości pakietów:
- 64B, mały prawdopodobnie za mały rozmiar
- 1500B, Maximum Transmission Unit
- 10000B
Stałe do testowania sie nie zmieniaja i sa ustawione nastepujaco:
MAX RETRANSMITS - 10
MAX WAIT - 10
Wyniki programów, zmierzone czasy sa zapisywane w tabelkach i przedstawione poniżej.

2 Testowane lokalnie na jednym komputerze

W nawiasie podana jest liczba udanych, pełnym przesłań pliku na 5 prób.
Można zauważyć pare tendencji: czas przesyłania rośnie liniowo wzgledem rozmiaru pliku, TCP i UDPR sa całkiem niezawodne, natomiast udp dla plików /¿2.8MB jest wyjatkowo mocno zawodny. Dodatkowo tcp jest znaczaco szybszy niż udpr dostarczajac tak samo dobre wyniki.

Table 1: Czas transferu danych testowany lokalnie

	1B	28KB	2.8MB	14MB	110MB
TCP (64B)	0.007 (5)	0.051 (5)	0.348 (5)	1.388 (5)	11.533 (5)
UDP (64B)	0.007 (5)	0.014 (5)	0.246 (0)	1.035 (0)	8.559 (0)
UDPR (64B)	0.008 (5)	0.039 (5)	2.092 (5)	11.352 (5)	82.174 (5)
TCP (1500B)	0.007 (5)	0.047 (5)	0.073 (5)	0.211 (5)	1.271 (5)
UDP (1500B)	0.007 (5)	0.008 (5)	0.059 (0)	0.143 (0)	0.870 (0)
UDPR (1500B)	0.007 (5)	0.012 (5)	0.154 (5)	0.573 (5)	5.102 (5)
TCP (10KB)	0.007 (5)	0.048 (5)	0.093 (5)	0.134 (0)	0.830 (5)
UDP (10KB)	0.007 (5)	0.009 (5)	0.040 (5)	0.399 (0)	0.545 (0)
UDPR (10KB)	0.007 (5)	0.008 (5)	0.085 (5)	0.105 (5)	1.326 (5)
TCP (64KB)	0.008 (5)	0.012 (5)	0.058 (5)	0.139 (5)	0.696 (5)
UDP (64KB)	0.007 (5)	0.008 (5)	0.045 (0)	0.102 (0)	0.537 (0)
UDPR (64KB)	0.008 (5)	0.010 (5)	0.065 (5)	0.173 (5)	0.819 (5)

3 Dwie maszyny wirtualne

Table 2: Czas transferu danych testowany na dwóch maszynach wirtualnych

	1B	28KB	2.8MB	14MB	110MB
TCP (64B)	0.006 (5)	0.066 (5)	2.822 (5)	13.559 (5)	110.470 (5)
UDP (64B)	0.006 (5)	0.069 (0)	1.305 (0)	3.512 (0)	24.943 (0)
UDPR (64B)	0.020 (5)	0.125 (5)	9.351 (5)	40.556 (5)	302.449 (5)
TCP (1500B)	0.022 (5)	0.034	2.764 (5)	11.342 (5)	88.342 (5)
UDP (1500B)	0.006 (5)	0.018 (0)	0.268 (0)	5.709 (0)	40.106 (0)
UDPR (1500B)	0.007 (5)	0.237 (5)	9.896 (5)	37.552	231.949 (5)
TCP (10KB)	0.017 (5)	0.079 (5)	0.104 (5)	0.361 (5)	3.107 (5)
UDP (10KB)	0.009 (5)	0.028 (5)	0.166 (0)	0.267 (0)	1.832 (0)
UDPR (10KB)	0.009 (5)	0.026 (5)	0.422 (5)	0.769 (5)	4.270 (5)

w tym przypadku trendy ogólne sa podobne jak wyżej, natomiast bezwzgledne wartości sa parokrotnie wyższe od swoich odpowiedników wyżej.

4 Opóźnienie 1000ms

w przypadku wprowadzenia opóźnienia protokół udp przestaje być wiarygodny, natomiast przy używaniu udpr nie udało mi sie zmierzyć rozsadnego czasu wykonywania. Jedyny wiarygodny i zaskakujaco sensowny czasowo jest tcp.

5 Strata pakietów 25%

w tym przypadku ogólne trendy sa podobne natomiast czesto udpr oraz udp po prostu sie psuja. Nastepuje timeout lub odebranie złego pakietu i klient sie

Table 3: Czas transferu danych testowany na dwóch maszynach wirtualnych z 50ms opóźnienia

	1B	28KB	2.8MB	14MB	110MB
TCP (64B)	0.253 (5)	0.347 (5)	2.039 (5)	12.644 (5)	73.779 (5)
UDP (64B)	0.204 (5)	error	error	error	error
UDPR (64B)	0.126 (5)	31.768 (5)	ponad rozsadny czas	ponad rozsadny czas	ponad rozsadny czas
TCP (1500B)	0.411 (5)	0.325 (5)	2.054 (5)	12.379 (5)	82.222 (5)
UDP (1500B)	0.211 (5)	0.275/error	error	error	error
UDPR (1500B)	0.189 (5)	55.059	ponad rozsadny czas	ponad rozsadny czas	ponad rozsadny czas
TCP (10KB)	0.188 (5)	0.308 (5)	2.892 (5)	12.202 (5)	66.889 (5)
UDP (10KB)	0.099 (5)	error	error	error	error
UDPR (10KB)	0.229 (5)	25.541	ponad rozsadny czas	ponad rozsadny czas	ponad rozsadny czas

Table 4: Czas transferu danych testowany na dwóch maszynach wirtualnych z 25% strata pakietów

	1B	28KB	2.8MB	14MB	110MB
TCP (64B)	0.19 (5)	2.957 (5)	7.095 (5)	22.341 (5)	168.798 (5)
UDP (64B)	0.005 (0)	0.035 (0)	0.811 (0)	10.282 (0)	31.910 (0)
UDPR (64B)	0.021 / timeout	112.525 / timeout	timeout	timeout	timeout
TCP (1500B)	0.111 (5)	0.277 (5)	7.261 (5)	28.389 (5)	144.340
UDP (1500B)	0.005 (5)	error/timeout	error/timeout	error/timeout	error/timeout
UDPR (1500B)	0.021 / timeout	timeout	timeout	timeout	timeout
TCP (10KB)	0.022 (5)	0.099 (5)	0.351 (5)	3.664	24.728
UDP (10KB)	0.019 (5)	0.020 (4)	timeout	timeout	timeout
UDPR (10KB)	0.022 (5)	0.026 (5)	timeout/error	timeout/error	timeout/error

wyłacza.

6 Podsumowanie

W prawdziwym życiu nie bedziemy mieć warunków idealnych takich jak w przypadku 1 i 2. Dlatego, wydaje sie, że jeżeli zależy nam na dostaniu każdego pakietu to niestety musimy zrezygnować z udp. Natomiast w przypadku udpr bedziemy musieli długo oczekiwać. Obiektywnie najlepszym protokołem do takiego zadania jest tcp. Pomimo teoretycznie dłuższego działania (3 way handshake) to w praktyce tcp wydaje sie najlepszym rozwiazaniem.