

# Raport Sieci Komputerowe

Juliusz Pawlus

April 26, 2024

Wszystkie dane zostały zaobserwowane przy użyciu programu wireshark, który mierzył przesył danych między dwoma maszynami wirtualnymi w laboratoriach MIMUW. Czas przesyłu danych w tabelach jest podany w sekundach. Program był testowany przy *MAX\_WAIT* ustawionym na 3 sekundy, a *MAX\_RETRANSMITS* na 10. Badania była przeprowadzane na paczkach o wielkości 64000 (maksymalny rozmiar bufferu) oraz 1450 (rozmiar pakietu mieszczący się w jednym pakiecie UDP).

## 1 UDP

Dane dla pliku o wielkości 1MB (czas przesyłu danych/ilość przesłanych bajtów):

Wielkość paczki danych	1450	64000
Bez modyfikacji	Wadliwy przesył (0.023, 1'043'632)	Wadliwy przesył (0.009, 1024018)
Przepustowość (160 Mbit)	Średni przesył (0.054, 1'043'632)	Średni przesył (0.052, 1024018)
Opóźnienie (0.25)	Nigdy nie przesyła	Nigdy nie przesyła
Procent zgubionych pakietów (10%)	Nigdy nie przesyła	Nigdy nie przesyła

Wadliwy przesył - udaje się wysłać wiadomość w pełni w połowie przypadków.

Średni przesył - udaje się wysłać wiadomość praktycznie zawsze.

Zbieranie danych dla pliku o wielkości 100MB w każdym wypadku zakończyło się niepowodzeniem.

Wnioski:

- Zaimplementowany kod używa protokołu UDP w taki sposób, że bez przerw wysyła wszystko naraz bez odbierania potwierdzeń. Skutkiem takiego działania jest brak miejsca na dane w buforze. Serwer nie nadąża ze czytaniem wiadomości przez co nowo dosyłane pakiety są gubione, a klient nawet nie jest tego świadomy.
- Dla plików o rozmiarze 100MB klient UDP dostaje RJT, ponieważ wysyła paczki cały czas, a tylko niektóre z nich trafiają do serwera. Paczki, które się nie mieściły przepadają przez brak miejsca, a stąd serwer odbiera po chwili paczki o złych parametrach, co powoduje odesłanie RJT.
- Dla 1MB serwer często zdąża odebrać wystarczająco szybko, przez co czasem udaje się odebrać wiadomość, lecz zakończenie poprzez RJT wciąż jest częste.
- Mniejsza przepustowość pozwala przy małych plikach serwerowi nadążyć z odbieraniem wiadomości, ponieważ klient wolniej zapełnia bufor. Dzięki temu przesłanie często kończy się sukcesem.
- Przy opóźnieniu paczki klientów oraz RJT dochodzą później, więc w obu przypadkach klienci kończą się timeoutem.
- Przy dodatkowo gubionych pakietach nie zmienia się tak naprawdę nic, jedynie gubione są dodatkowe pakiety, co tym bardziej powoduje zakończenie się błędem.
- Rozmiar paczki nie wpływa drastycznie na pomiar.

## 2 UDPR

Dane dla pliku o wielkości 1MB (czas przesyłu danych/ilość przesłanych bajtów):

Wielkość paczki danych	1450	64000
Bez modyfikacji	(0.017, 1'024'962)	(0.023, 1'035'712)
Przepustowość (160 Mbit)	(0.655, 1'084'342)	(0.099, 1'024'962)
Opóźnienie (0.25)	(371.311, 1'086'106)	(8.378, 1'024'792)
Procent zgubionych pakietów (10%)	Nigdy nie przesyła	Nigdy nie przesyła

Dane dla pliku o wielkości 100MB (czas przesyłu danych/ilość przesłanych bajtów):

Wielkość paczki danych	1450	64000
Bez modyfikacji	(12.260, 108'414'182)	(2.371, 107'311'956)
Przepustowość (160 Mbit)	(84.141, 108'414'434)	(5.832, 102'475'374)
Opóźnienie (0.25)	BARDZO DŁUGO	(713.298, 102'672'230)
Procent zgubionych pakietów (10%)	Nigdy nie przesyła	Nigdy nie przesyła

Wnioski:

- Dla większych plików, paczki o rozmiarze 64000 działają zdecydowanie szybciej, ponieważ o wiele rzadziej potrzebne jest odbieranie accept od serwera. Protokół używający paczek o rozmiarze 1450 bardzo często musi czekać na potwierdzenie, przy czym traci dużo czasu.
- Dla mniejszych plików różnica w wielkościach paczek jest nieodczuwalna.
- Dane wynikające ze zmian przepustowości oraz opóźnienia jeszcze bardziej potwierdzają tezę, iż częste oczekiwania na accept spowalniają protokół.
- Ograniczenie przepustowości drastycznie zwiększa czas oczekiwania, ponieważ dane są wysyłane o wiele wolniej, więc zarówno serwer jak i klient bardzo długo muszą czekać na dane oraz ich akceptację. Widoczne jest to zwłaszcza przy paczkach wielkości 1450, ponieważ one muszą bardzo często czekać na akceptację, która przychodzi o wiele wolniej, niż zwykle.
- Przy opóźnieniu pakiety nie dochodzą w oczekiwanym czasie, przez co klient oraz serwer zaczynają wysyłać retransmisje, które też wysyłane są z opóźnieniem, co powoduje, że czas wysyłki drastycznie się zwiększa.
- Dla paczek o rozmiarze 1450 przy pliku 100 MB opóźnienie jest szacowane na podstawie bardzo długiego czasu wysyłania pliku 1MB.
- UDPR nigdy nie radzi sobie z przesyłem z gubieniem pakietów w trakcie. Pomimo dużej ilości retransmisji, z których część przynosiła pożądaną skutek, po czasie dane nie był w stanie dotrzeć. Dla mniejszych plików dałoby się poprawić sytuację modyfikując liczbę retransmisji oraz maksymalny czas oczekiwania.

### 3 TCP

Dane dla pliku o wielkości 1MB (czas przesyłu danych/ilość przesłanych bajtów):

Wielkość paczki danych	1450	64000
Bez modyfikacji	(0.048, 1'074'206)	(0.073, 1'023'154)
Przepustowość (160 Mbit)	(0.059, 1'066'682)	(0.062, 1'050'482)
Opóźnienie (0.25)	(4.863, 1'072'028)	(4.213, 1'051'334)
Procent zgubionych pakietów (10%)	(1.504, 1'073'950)	(0.457, 1'034'647)

Dane dla pliku o wielkości 100MB (czas przesyłu danych/ilość przesłanych bajtów):

Wielkość paczki danych	1450	64000
Bez modyfikacji	(2.289, 105'215'782)	(2.687, 108'383'618)
Przepustowość (160 Mbit)	(5.313, 106'425'218)	(5.696, 107'211'478)
Opóźnienie (0.25)	(31.410, 107'101'634)	(32.788, 108'104'469)
Procent zgubionych pakietów (10%)	Nigdy nie przesyła	Nigdy nie przesyła

Wnioski:

- Rozmiar paczki w protokole TCP nie wprowadza dużych zmian. Wynika to stąd, że w tym protokole potwierdzenia nie są wysyłane co każdy pakiet, a co pewną grupę pakietów o wyliczonej wielkości. Stąd dla pakietów o rozmiarach 64000 i 1450, wszystkie czasy wykonania wyglądają dość podobnie.
- Zmniejszona przepustowość spowalnia TCP, ale to wynika z tego, że pozwala ona na mniejszy przesył danych, więc jej wynik jest jak najbardziej spodziewany.
- Opóźnienie spowalnia wysyłanie zarówno pakietów danych, jak i ich akceptacji, przez co czas przesyłu diametralnie się zwiększa.
- TCP jako jedyne radziło sobie w jakikolwiek sposób przy takim procencie gubienia pakietów. Wynika to z o wiele bardziej optymalnego systemu retransmisji, niż ten, który ma zaimplementowane UDPR.

### 4 TCP vs UDPR

Wnioski:

- Duże zróżnicowanie w wynikach da się zaobserwować pomiędzy poszczególnymi rozmiarami pakietów.
- Dla pakietów o rozmiarze 1450 czas przesyłu był bardzo różny. Wynika to z tego, że w protokole UDPR każdy pakiet 1450 jest potwierdzany akceptacją, a w TCP grupy pakietów 1450 są potwierdzane naraz, co jest dość mocną optymalizacją.
- Dla pakietów o rozmiarze 64000 czas przesyłu był o wiele bardziej zbliżony. Wynika to stąd, że grupy pakietów akceptowanych przez tcp miały o wiele bardziej zbliżoną wielkość do rozmiaru pakietu.
- Przy zmniejszonej przepustowości różnice te są jeszcze bardziej uwidocznione, co tylko potwierdza tezę.
- Duża jest natomiast różnica czasów przy opóźnieniach dla obu rozmiarów paczek. Dla 1450 jest to dodatkowo spowodowane ponownie częstymi akceptacjami. Natomiast niezależnie od rozmiaru przy opóźnieniach zauważalna była duża ilość retransmisji w UDPR. Jest to spowodowane dużą ilością timeoutów oraz odbierania zaległych akceptacji i czekaniu na poprawne dane, co w protokole TCP jest o wiele lepiej zoptymalizowane.