

Niezawodność i diagnostyka układów cyfrowych	
Temat projekt: Transmisja w systemie ARQ (Automatic Repeat Request).	
Marek Tutka	276314
Mateusz Głuchowski	272885
Data oddania:	17.06.2024
Ocena:	

Spis treści

1.	Krótki opis projektu	2
2.	Wstęp teoretyczny	2
3.	Wyniki eksperymentów	3
4.	Wnioski	7
5.	Podsumowanie	8

1. Krótki opis projektu

Celem projektu jest zaimplementowanie systemu transmisji ARQ (eng. Automatic Repeat Request) z wykorzystaniem przynajmniej dwóch różnych kodów detekcyjnych takich jak bit parzystości, kody Hamminga, CRC8 oraz CRC16. System zostanie zbudowany w architekturze klient-serwer, gdzie do komunikacji pomiędzy klientem a serwerem zostanie użyty protokół UDP.

Aby ocenić wpływ parametrów sieciowych na wydajność transmisji, takich jak opóźnienie (eng. latency), utrata pakietów (eng. drops), wielkość okna (eng. window size) przeprowadzone zostaną symulacje z użyciem emulatora sieciowego GNS3 lub CML. Jeśli użycie emulatora okaże się niemożliwe, zaimplementowany zostanie algorytm symulujący straty pakietów

W ramach projektu porównane zostaną dwa algorytmy ARQ: Stop-and-Wait oraz Selective Repeat. Analiza wyników będzie obejmować różne scenariusze sieciowe oraz parametry transmisji, co pozwoli na wyciągnięcie wniosków dotyczących optymalnych warunków pracy dla każdego z algorytmów. Wyniki eksperymentów zostaną przedstawione w formie wykresów i tabel, a ich interpretacja posłuży do sformułowania końcowych wniosków dotyczących efektywności i niezawodności systemu ARQ w różnych warunkach.

2. Wstęp teoretyczny

We wstępie przedstawimy definicje kluczowych pojęć dotyczących projektu.

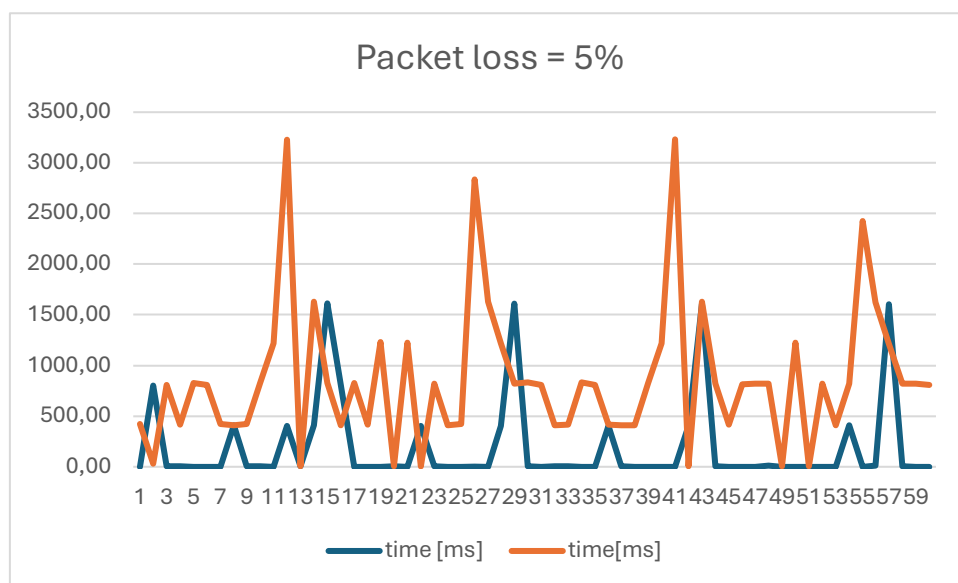
- **SYSTEM ARQ** – technika używana w komunikacji danych, szczególnie w sieciach komputerowych oraz komunikacji bezprzewodowej. Jest to mechanizm, który pozwala na automatyczne wykrywanie i korektę błędów w transmisji danych poprzez ponowne przesłanie pakietów, które zostały utracone lub uszkodzone w drodze do odbiorcy. Powszechnie używany w sieciach telekomunikacyjnych, gdzie niezawodność transmisji danych jest kluczowa.
- **STOP-AND-WAIT** - algorytm, w którym nadawca wysyła pojedynczy pakiet danych i czeka na potwierdzenie (eng. ACK – acknowledgment) od odbiorcy. Odbiorca otrzymuje pakiet i sprawdza jego poprawność, następnie przesyła do nadawcy, informując go o prawidłowym odebraniu. Jeżeli nadawca nie otrzyma ACK w konkretnym czasie, zakłada, że pakiet został uszkodzony lub utracony i wysyła go ponownie.

- **SELECTIVE REPEAT** – algorytm, w którym nadawca może wysyłać wiele pakietów danych zanim otrzyma potwierdzenie od odbiorcy. Odbiorca otrzymuje pakiety, sprawdza ich poprawność i przesyła potwierdzenie (ACK) dla każdego pakietu osobno. Jeśli nadawca nie otrzyma potwierdzenia odnośnie konkretnego pakietu – wysyła go ponownie. Odbiorca może zachować poprawnie odebrane pakiety w buforze do momentu, gdy wszystkie pakiety poprzedzające brakujący zostaną otrzymane. Wtedy może przestać ACK dla wszystkich poprawnie odebranych pakietów, włącznie z brakującym.

3. Wyniki eksperymentów

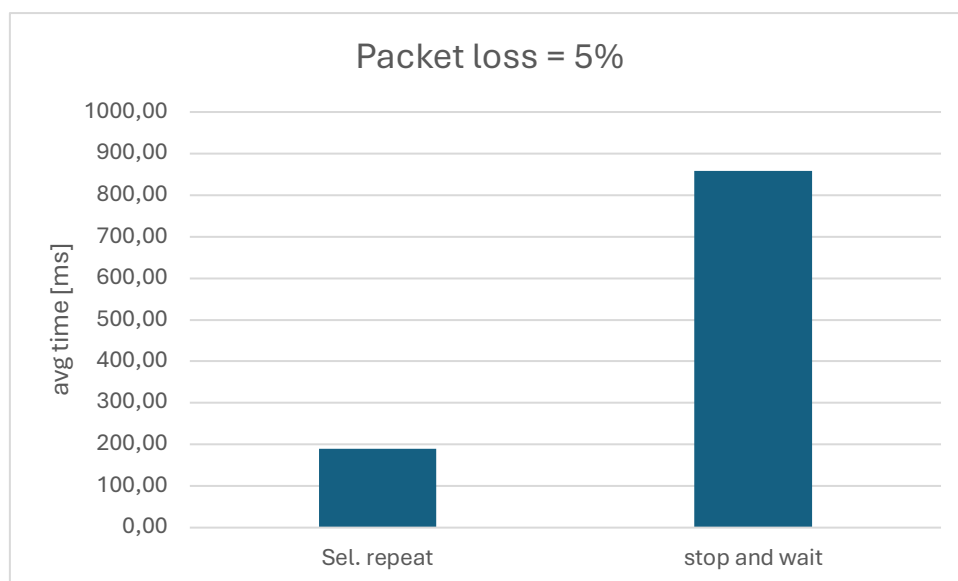
Poniżej zostały przedstawione wyniki eksperymentu na podstawie opisanych wcześniej założeń.

Poniższe wykresy przedstawiają porównanie na podstawie dobranych procentowych spadków pakietów.

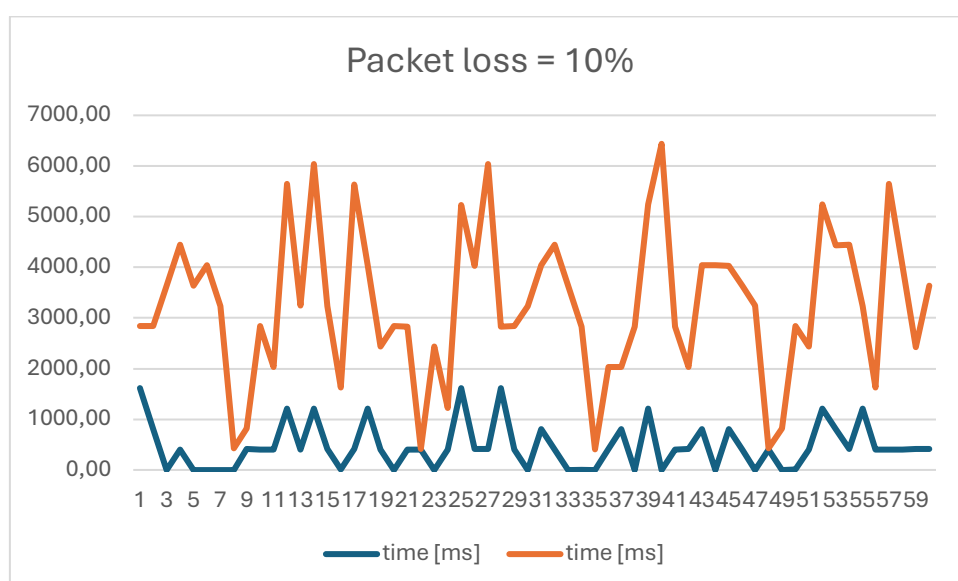


Rys. 1 - Utrata pakietów dla spadku 5% dla algorytmu Selective Repeat (kolor pomarańczowy) oraz algorytmu Stop-and-Wait (kolor niebieski).

Na rys. 1 widać różnicę w liczbie utraconych pakietów między dwoma algorytmami przy 5% utracie pakietów. Jak widać, Selective Repeat radzi sobie lepiej, tracąc mniej pakietów niż Stop-and-Wait.

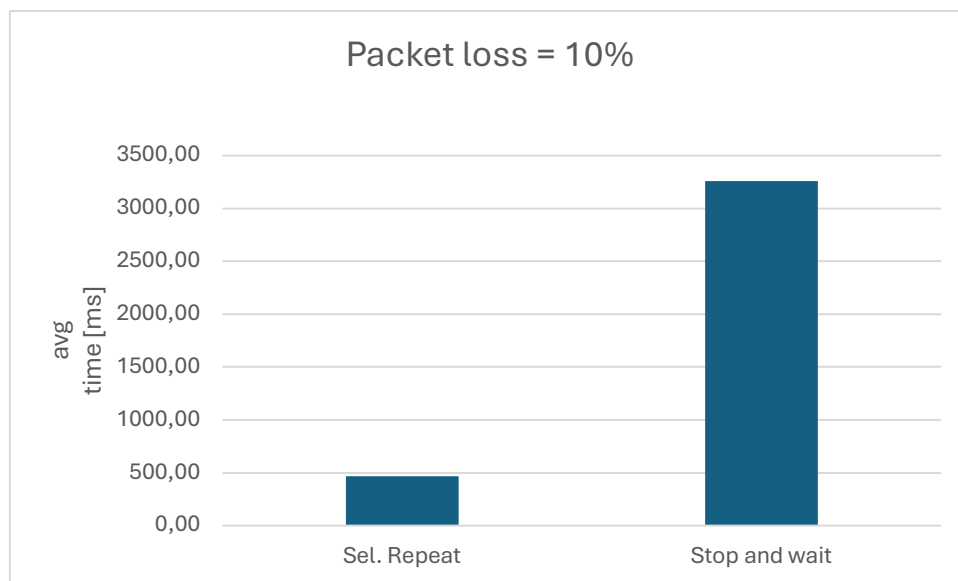


Rys. 2 - uśrednione wyniki pomiarów dla rys. 1.

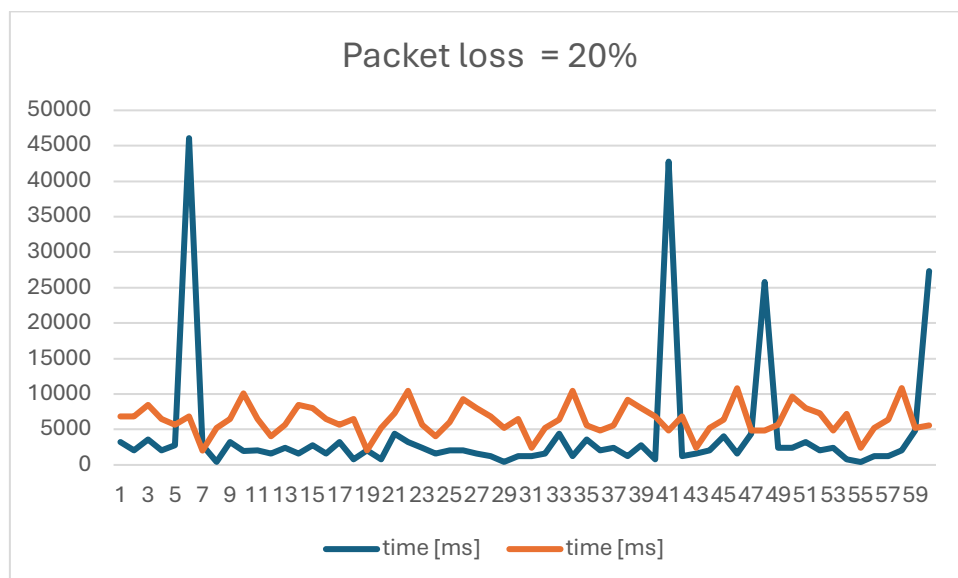


Rys. 3 - utrata pakietów dla spadku % dla algorytmu Selective Repeat (kolor pomarańczowy) oraz algorytmu Stop-and-Wait (kolor niebieski).

Na rys. 3 zilustrowany jest wpływ zwiększenia utraty pakietów dla spadku 10% na wydajność obu algorytmów. Selective Repeat nadal wykazuje lepsze wyniki w porównaniu do Stop-and-Wait.

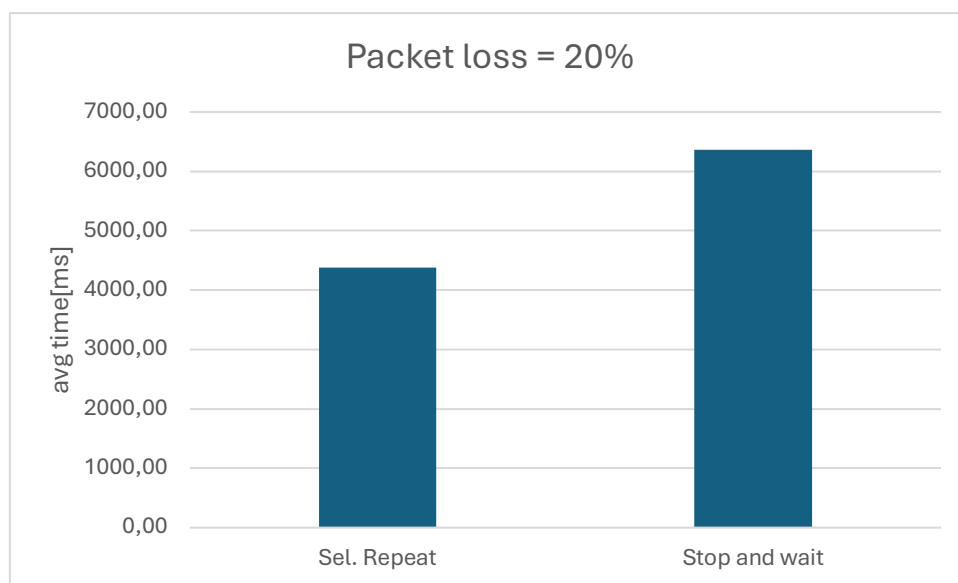


Rys. 4 - uśrednione wyniki pomiarów dla rys. 3.



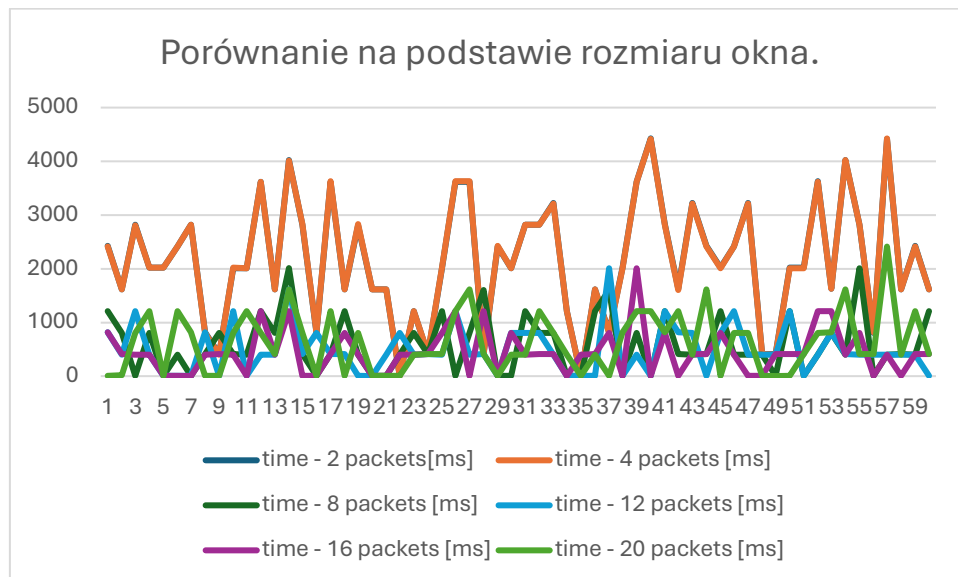
Rys. 5 - przedstawia utratę pakietów dla spadku 20%, dla algorytmu selective repeat (kolor pomarańczowy) oraz algorytmu stop and wait (kolor niebieski).

Na rys. 5 wykazuje przy 20% utracie pakietów, różnice w wydajności między Selective Repeat a Stop-and-Wait są jeszcze bardziej widoczne. Selective Repeat jest bardziej efektywny, tracąc mniej pakietów.



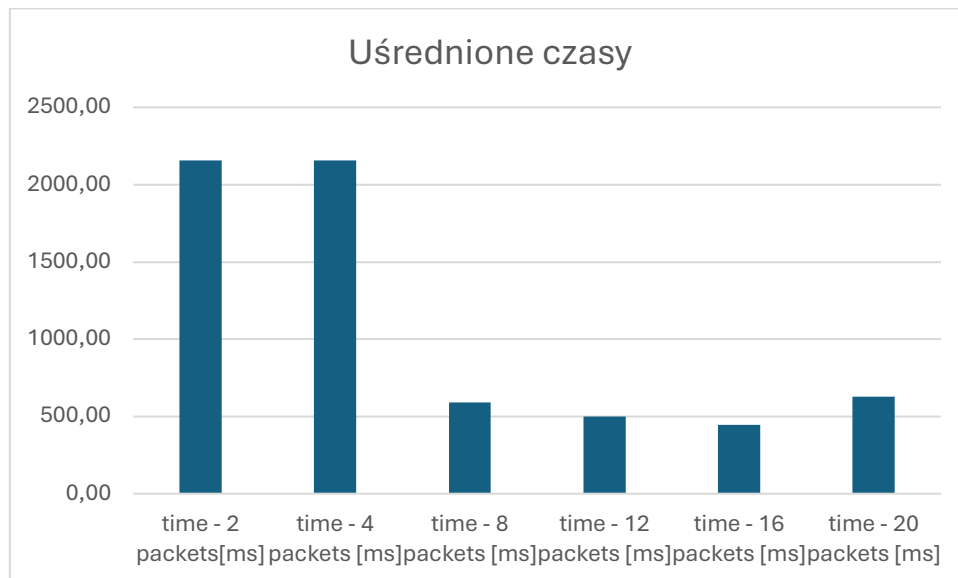
Rys. 6 - uśrednione wyniki pomiarów dla rys. 5.

Poniższy wykresy przedstawia porównanie na podstawie rozmiarów okna.



Rys. 7 - porównanie czasu transmisji dla różnych rozmiarów okna (2, 4, 8, 12, 16 pakietów) w algorytmach Stop-and-Wait oraz Selective Repeat.

Rys. 7 pokazuje, jak rozmiar okna wpływa na czas transmisji. Dla małych okien (2 pakiety) czas transmisji jest dłuższy, podczas gdy dla średnich okien (8-12 pakietów) uzyskano najlepszy balans między czasem transmisji a stabilnością. Duże okna (16 pakietów) nie przyniosły już znaczących korzyści.



Rys. 8 - uśrednione wyniki pomiarów dla rys. 7.

4. Wnioski

W ramach projektu dotyczącego transmisji w systemie ARQ (Automatic Repeat Request) zaimplementowano system transmisji ARQ, testując różne algorytmy detekcji błędów oraz wpływ parametrów sieciowych na wydajność transmisji. Przeprowadzono liczne symulacje, których wyniki zostały przedstawione w tabelach i wykresach. Analiza wyników prowadzi do kilku istotnych wniosków:

1. Porównanie Algorytmów: Stop-and-Wait vs. Selective Repeat

- **Stop-and-Wait:** Czas transmisji dla algorytmu Stop-and-Wait jest znacząco dłuższy w porównaniu do Selective Repeat przy różnych wartościach utraty pakietów (5%, 10%, 20%). Przy wysokiej utracie pakietów (20%), czas transmisji dla Stop-and-Wait był ponad 3 razy dłuższy niż dla Selective Repeat.
- **Selective Repeat:** Algorytm Selective Repeat wykazuje wyraźnie lepszą wydajność pod względem czasu transmisji, zwłaszcza przy większych wartościach utraty pakietów. Pomimo wyższego obciążenia pamięci i złożoności implementacji, jest bardziej efektywny w scenariuszach z wysoką utratą pakietów.

2. Wpływ Utraty Pakietów

- ☐ Zwiększenie procentowej utraty pakietów prowadzi do znaczącego wzrostu czasu transmisji w obu algorytmach. Jednak wzrost ten jest mniej drastyczny w przypadku Selective Repeat.
- ☐ W przypadku Stop-and-Wait, przy 20% utracie pakietów, czas transmisji jest znacznie wyższy, co może wskazywać na jego nieoptymalność w środowiskach o wysokiej awaryjności.

3. Wpływ Rozmiaru Okna na Wydajność

- ☐ Małe okna (2 pakiety): Transmisja jest stabilna, ale czas przesyłu jest dłuższy.
- ☐ Średnie okna (4, 8, 12 pakietów): Optymalny balans między czasem transmisji a stabilnością. Najlepsze wyniki osiągnięto dla okna o rozmiarze 8 i 12 pakietów.
- ☐ Duże okna (16 pakietów): Nie przyniosły one już znaczących korzyści w porównaniu do średnich okien, ale czas transmisji nadal był korzystny.

4. Symulacja Warunków Sieciowych

- ☐ Testowanie różnych parametrów sieciowych takich jak opóźnienie (latency), jitter i utrata pakietów przy użyciu emulatora GNS3 lub CML wykazało, że realne warunki sieciowe mogą znacząco wpływać na wydajność transmisji.
- ☐ Implementacja algorytmu symulującego straty pakietów była skuteczna w ocenie wydajności systemu ARQ w różnych scenariuszach.

5. Zastosowanie Protokołu UDP

- ☐ Wykorzystanie protokołu UDP w architekturze client-server było skuteczne, aczkolwiek wymagało dodatkowych mechanizmów detekcji i korekcji błędów ze względu na brak wbudowanych funkcji niezawodnościowych.

5. Podsumowanie

Przeprowadzone badania wykazały, że wybór odpowiedniego algorytmu ARQ i dostosowanie parametrów sieciowych ma kluczowe znaczenie dla efektywności transmisji danych. Selective Repeat okazuje się bardziej wydajny w środowiskach o wysokiej utracie pakietów, podczas gdy Stop-and-Wait jest bardziej adekwatny w stabilniejszych warunkach sieciowych. Dalsze badania mogą skupić się na optymalizacji rozmiaru okna transmisyjnego oraz dalszej symulacji różnych warunków sieciowych w celu bardziej precyzyjnego dostosowania systemu ARQ do specyficznych scenariuszy aplikacyjnych.