21.06.2020 r.

Projekt NIDSC
Kanał transmisyjny z użyciem ARQ

Maciej Borowski 248891
Miron Oskroba 236705

prowadzący:
prof. Henryk Maciejewski

1. Cel i założenia projektu

1.1. Wstęp teoretyczny

Metoda ARQ to technika, która po wykryciu błędu w przesyle danych automatycznie inicjuje retransmisję. Nadajnik dołącza do każdego pakietu różnego rodzaju bity kontrolne na podstawie których weryfikowana jest poprawność przesyłu w odbiorniku i wysyłana jest informacja zwrotna. Dzięki tej koncepcji pakiety są dostarczane do miejsca docelowego dokładnie raz, bez duplikatów, w tej samej kolejności w której zostały wysłane.

1.2. Kody detekcyjne użyte w projekcie

1.2.1. Bit parzystości

Najbardziej podstawowa metoda sprawdzania poprawności przesyłu. Do każdego pakietu dodawany jest bit określający czy liczba bitów o wartości 1 jest parzysta, czy nie. Następnie, w odbiorniku, sprawdzane jest czy liczba “jedynek” w pakiecie faktycznie zgadza się z bitem parzystości i na tej podstawie wysyłana jest informacja zwrotna.

1.2.2. Cyclic redundancy check (CRC)

Metoda CRC bazuje na algorytmie obliczania sum kontrolnych wykorzystujący tzw. wielomiany CRC, czyli po prostu ciąg bitów. Algorytm oblicza resztę z dzielenia pakietu danych przez określony wielomian i dodaje ją na koniec.

1.3. Model BSC kanału transmisyjnego

W tym modelu prawdopodobieństwo przekłamania pojedynczego bitu jest z góry określane. Pakiet, złożony z bitowych wartości, wysyłany przez nadajnik przechodzi przez kanał transmisyjny i każdy bit jest przełamywany lub nie, zgodnie z prawdopodobieństwem przekłamania.

1.4. Użyte środowisko

Badania były robione w języku python, w wersji 3.8.2 z użyciem programu PyCharm 2019.3.3. Wszystkie stanowiska na których opracowany był kod symulatora miały podobne parametry.

2. Opis narzędzia

2.1 Użyte biblioteki zewnętrzne

W projekcie skorzystano z dodatkowych bibliotek zewnętrznych:

-komm
-crc16
-numpy
-random

2.2. Opis kodu

Celem lepszej organizacji kodu, program został napisany w wersji obiektowej, dzieląc kod na klasy:

- **ARQ-Symulator**: Klasa będąca punktem wejścia programu - zawiera kod do sterowania pozostałymi klasami. Zawiera menu programu, w którym można regulować poszczególne parametry takie jak: tryb kodowania, długość sygnału, długość pakietu czy prawdopodobieństwo zakłócenia). Można symulację uruchomić bądź skorzystać z opcji zautomatyzowanego testu, który wypisuje średnią wyników do pliku .txt.
- **Generator**: Generuje pakiet początkowy.. Zaimplementowane zostały dwa tryby kodowania: “CRC” oraz “PAR”, czyli odpowiednio kod CRC oraz kod bitu parzystości.
- **Sender**: Gdy wiadomy jest już tryb kodowania należy przekazać go do klasy Sender, aby zakodowała pakiet w odpowiednim trybie.
- **Transmitter**: Następnie mając określoną zmienną *cross_prob*, następuje symulacja losowych zakłóceń, a przesyłany pakiet jest ‘zmieniany’
- **Receiver**: Przetwarza otrzymany pakiet
- **Analyzer**: W zależności od wybranego trybu kodowania, analizowany jest otrzymany pakiet oraz wyznaczane są wartości: Redundant bits, Bit Error Rate, którymi posłużono się do stworzenia wykresów ostatecznych.

3. Organizacja eksperymentu symulacyjnego

3.1. Plan eksperymentu

Celem wyznaczenia parametrów niezbędnych do stworzenia wykresu zależności Bit Error Rate od ilości bitów nadmiarowych postanowiono napisać moduł testowy - czyli opcję w menu programu głównego, która dla zadanej liczby powtórzeń wyznaczy średnią wartość BER dla aktualnych parametrów:

-długość sygnału
-długość pakietu
-tryb kodowania
-prawdopodobieństwo zakłócenia

oraz wyznaczy ilość bitów nadmiarowych dla wyznaczonej kombinacji parametrów. Ze względu na implementację kodu CRC przy pomocy biblioteki zewnętrznej crc16, parametr ‘długość pakietu’ dla tego kodu wynosi maksymalnie 16 bitów.

3.2.1 Założenia pomiarowe dla bitu parzystości

Pomiary będą wykonywane dla długości sygnału 61440 bitów, dla długości pakietów: 160,80,40,20,16,10,8,4,2 oraz dla dwóch wartości prawdopodobieństw zakłóceń równych 10^-5 oraz 5*10^-6.

3.2.2 Założenia pomiarowe dla kodu CRC

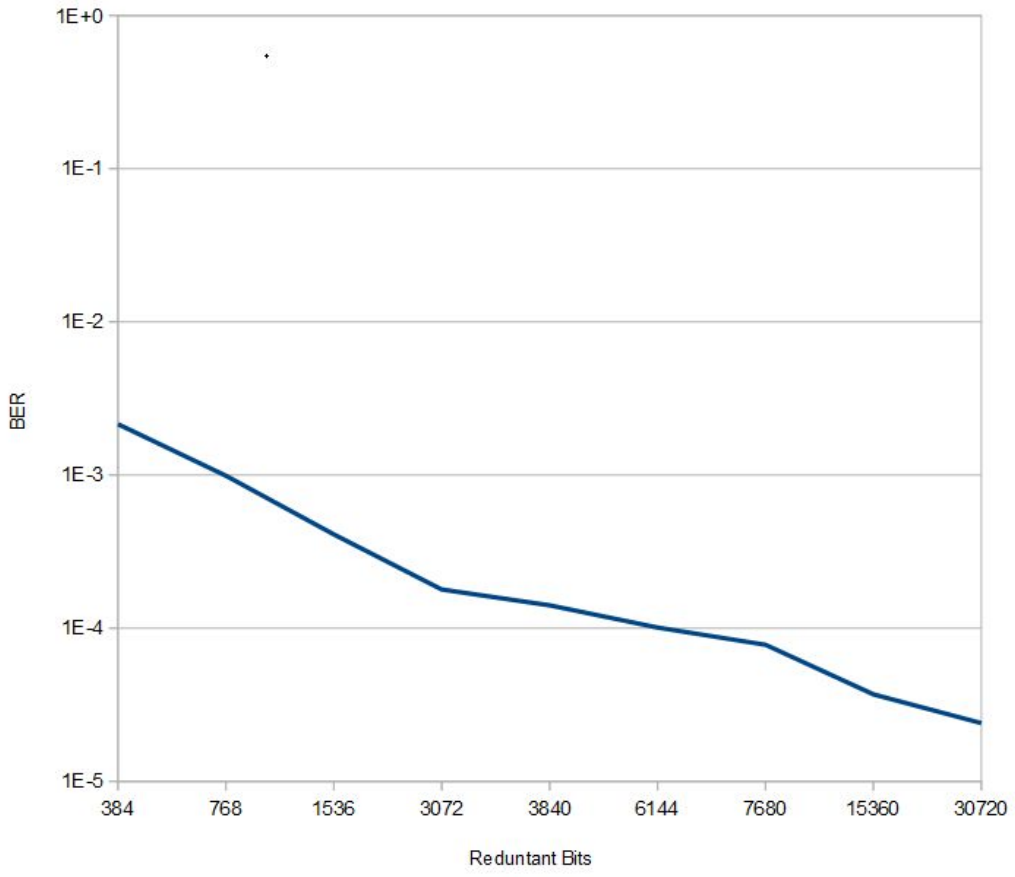
Pomiary będą wykonywane dla długości sygnału 61440 bitów, dla długości pakietów: 16,12,8,6,4,3,2 oraz dla dwóch wartości prawdopodobieństw zakłóceń równych 10^-5 oraz 5*10^-6.

3.3. Przebieg eksperymentu

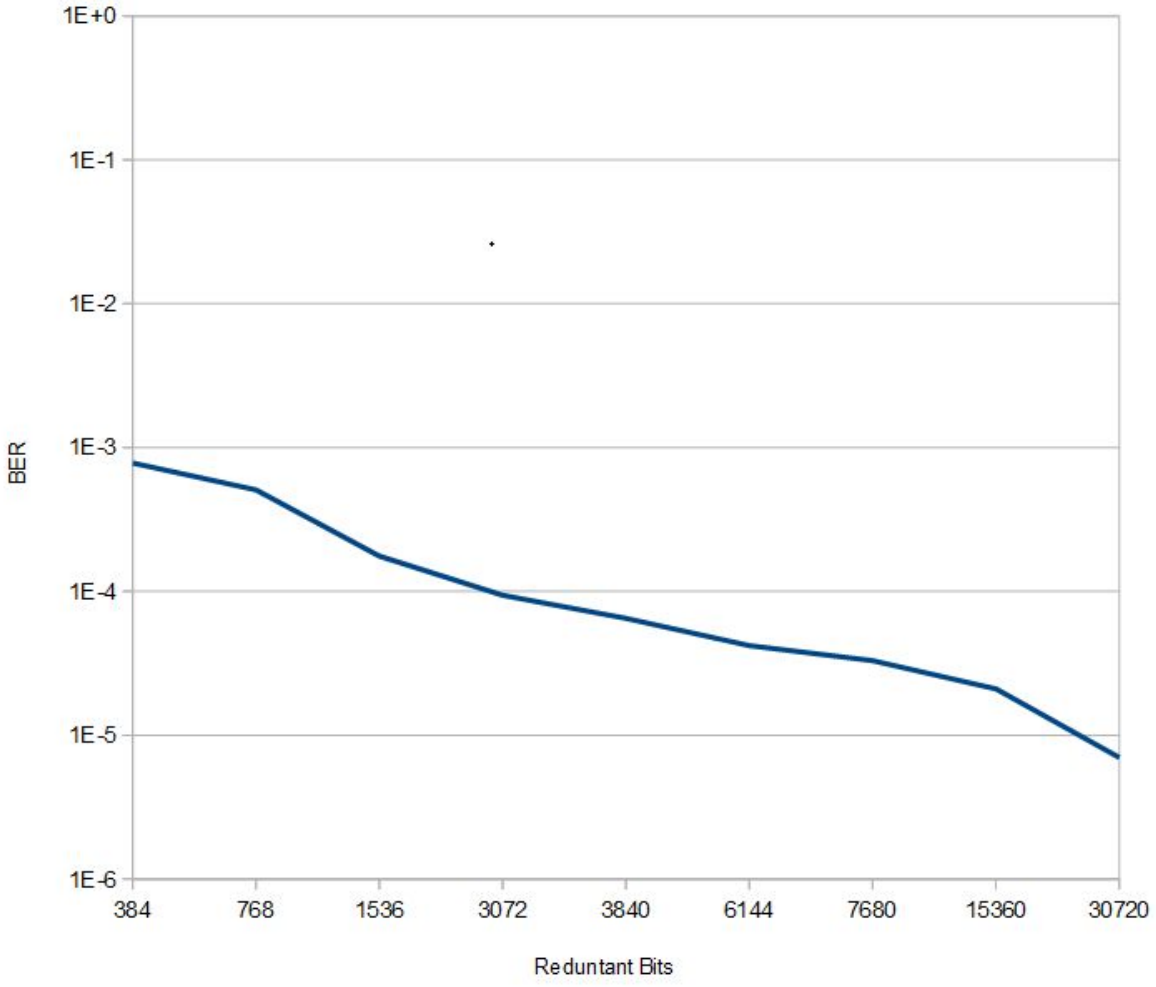
Dla każdego pakietu generowanego w generatorze w nadajniku dodawany był kod nadmiarowy. Kod z nadajnika przesyłany był do odbiornika przez moduł transmisyjny symulujący zachowanie BSC, trafiał do modułu analizującego. W odbiorniku następowała weryfikacja poprawności odebranych pakietów oraz dawany był sygnał do ewentualnej retransmisji. Kolejnym istotnym elementem było obliczenie współczynników BER oraz nadmiarowości w Analyzer'ze.

4. Wyniki

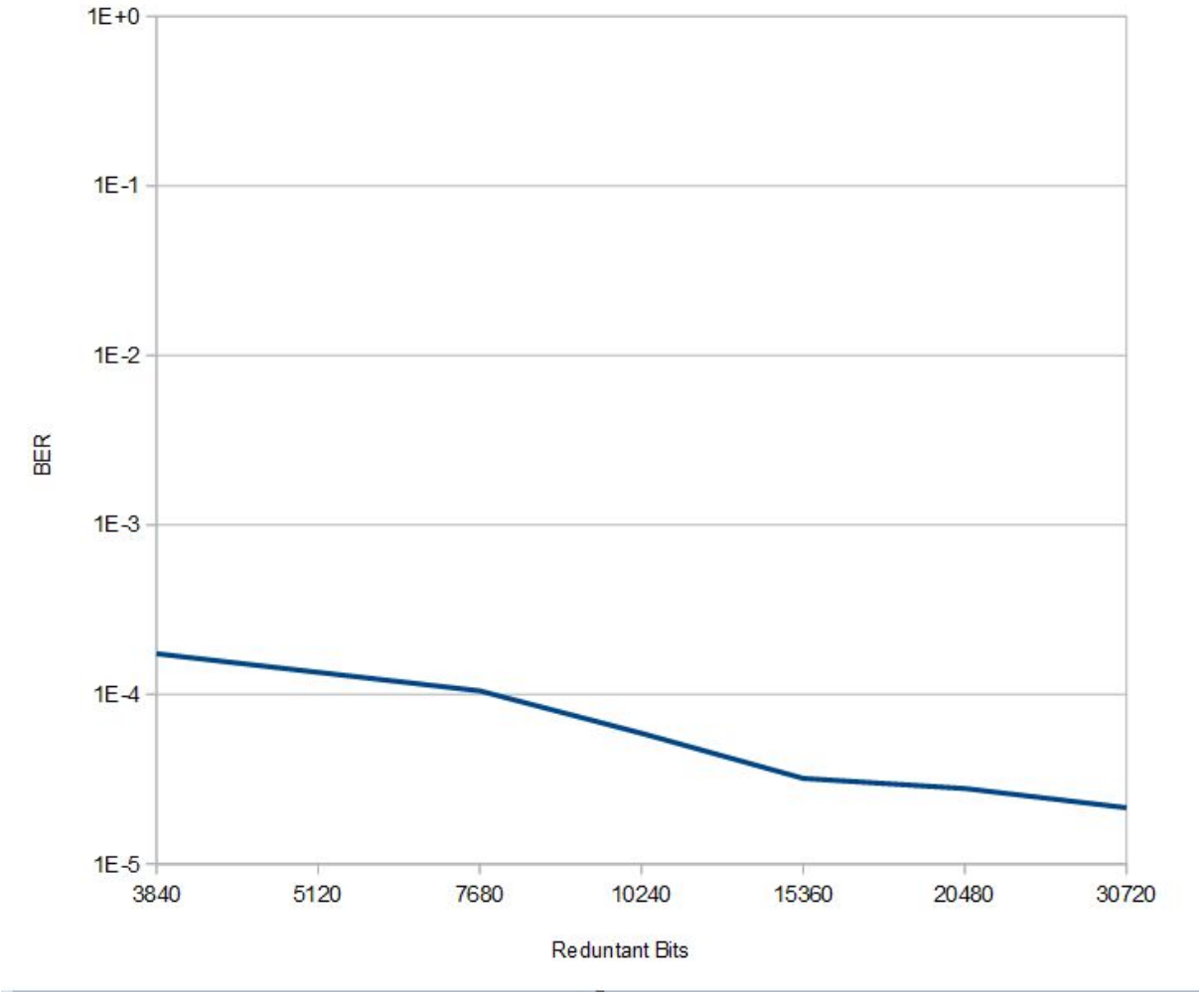
4.1. Wykresy



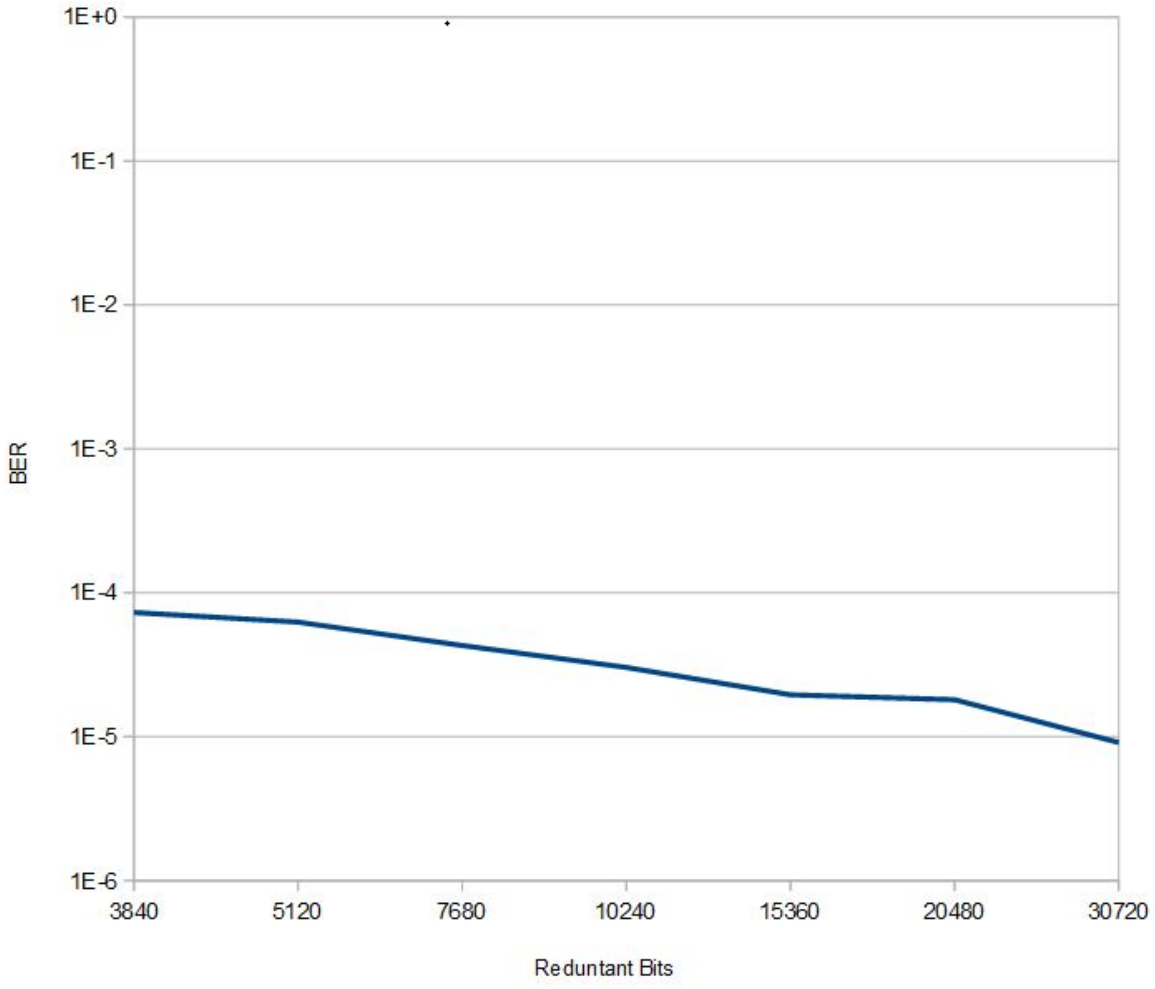
Rys. 1 Wykres zależności BER od ilości bitów nadmiarowych dla prawdopodobieństwa przekłamania 1E-5 oraz zastosowanego kodowania bitu parzystości



Rys. 2. Wykres zależności BER od ilości bitów nadmiarowych dla prawdopodobieństwa przekłamania 5E-6 oraz zastosowanego kodowania bitu parzystości



Rys. 3. Wykres zależności BER od ilości bitów nadmiarowych dla prawdopodobieństwa przekłamania 1E-5 oraz zastosowanego kodowania CRC



Rys. 4. Wykres zależności BER od ilości bitów nadmiarowych dla prawdopodobieństwa przekłamania 5E-6 oraz zastosowanego kodowania CRC

4.2. Analiza wyników

- Wraz z maleniem długości pakietu BER maleje,
- Dla mniejszych prawdopodobieństw zakłócenia BER jest mniejsze,
- Tryb kodowania CRC daje podobne wyniki BER jak kodowanie kodem bitu parzystości.
- Wartości BER w granicach tolerancji (10^{-4} , 10^{-5}) osiągane są w przypadku CRC dla długości pakietów 2, 3 lub 4 (wartości nadmiarowości kolejno: 30720, 20480, 15360)
- Wartości BER w granicach tolerancji (10^{-4} , 10^{-5}) osiągane są w przypadku kodowania bitem parzystości dla długości pakietów 2, 4 lub 6 (wartości nadmiarowości kolejno: 30720, 15360, 7680)

5. Wnioski

- Moduł analizujący ze względu na logikę oraz estetykę kodu mógłby znaleźć się wewnątrz modułu odbiornika,
- Wykresy wykazują potrzebę retransmisji dla danego trybu kodowania,
- Im dłuższy pakiet, tym większa szansa na wystąpienie błędu, lecz dla dłuższych pakietów dany kod detekcyjny, może nie rejestrować błędu,
- Retransmisja nie została zaimplementowana jak zakładano, bo głównym celem było sprawdzenie skuteczności zastosowanych kodów nadmiarowych oraz poprawności przesyłania danych,
- Zgodnie z założeniami wraz ze wzrostem nadmiarowości BER maleje dla obu trybów kodowania,
- Projekt został zrealizowany, lecz nie wszystkie założenia zrealizowano.