Politechnika Wrocławska

Niezawodność i diagnostyka systemów cyfrowych

Projekt – Transmisja w systemie ARQ

Autor: Paweł Maciończyk 248837

Michał Dunat 248862

Prowadzący: dr hab. inż. Henryk Maciejewski

Termin: Środa, tydzień parzysty, 07:30 – 09:00

Spis treści

Cel i założenia projektu

1. **ARQ** (Automatic Repeat Request) – to metoda kontroli błędów w transmisji danych, która wykorzystuje mechanizm potwierdzeń oraz przekroczeń limitów czasu w celu uzyskania niezawodnej i wiarygodnej transmisji danych. W momencie, gdy nadawca nie otrzyma potwierdzenia przed określonym timeout-em, pakiety danych przesyłane są ponownie, dopóki urządzenie wysyłające nie otrzyma potwierdzenia lub nie zostanie przekroczona określona liczba retransmisji. ARQ posiada 3 podstawowe protokoły zapewniające bezbłędne przesyłanie pakietów:

    - Stop and Wait – najprostszy protokół, który działa w następujący sposób: Nadawca wysyła tylko po jednym pakiecie bitów. Dopóki nie otrzyma sygnału potwierdzenia (ACK) nie wysyła kolejnego pakietu. Opisany protokół zostanie zaimplementowany w tym projekcie.
    - Go-Back-N – nadawca jednorazowo wysyła N pakietów, po ich wysłaniu oczekuje na potwierdzenie.
    - Selective Repeat - nadawca wykorzystujący ten protokół kontynuuje wysyłanie liczbę pakietów zdefiniowanych przez rozmiar okna nawet po utracie jakichkolwiek z pakietów. Odbiornik natomiast może selektywnie odrzucać tylko pojedynczy pakiet i prosić o jego retransmisję.

2. Kanały transmisyjne:

    - **BSC** (Binary Symmetric Channel) – opisuje go jeden parametr – prawdopodobieństwo odebrania niepoprawnego bitu (zwykle bardzo małe).

    - **Model Gilberta** – opisują go dwa parametry – przejścia ze stanu dobrego do złego i na odwrót. Pierwsze prawdopodobieństwo jest małe, a drugie - duże. Zły stan oznacza odbieranie niepoprawnych bitów z prawdopodobieństwem przejścia ze stanu złego do dobrego.

3. Badane kody detekcyjne:

    - **Bit parzystości** – polega on na dodaniu do paczki dodatkowego bitu, który przekazuje informację o ilości jedynek w paczce, konkretnie ich parzystości.

    - **CRC-32** – służy do wyznaczania sum kontrolnych dla dowolnych danych wejściowych. Jest to 33-bitowa liczba wyznaczająca poprawność danych ze wzorcem. Do danego pakietu danych dołączona jest wyliczona reszta z dzielenia przez wielomian CRC.

4. Środowisko programistyczne:

    - **Matlab** – środowisko wręcz stworzone do realizacji tego projektu. Wyżej wymienione kody detekcyjne są oficjalnie zaimplementowane, co ułatwi nam pracę nad projektem i zapewni poprawność otrzymanych wyników. „Communication Toolbox” to nazwa biblioteki, która implementuje wspomniany kod detekcyjny CRC-32.

5. Wyjście:

- **Efektywny BER** – BER to współczynnik ilości otrzymanych błędnych pakietów bitów do całkowitej ilości otrzymanych tychże pakietów.
  Naszym zadaniem będzie określenie efektywnego BER-u.
- **Nadmiarowość** – ilość dodatkowych bitów wysłanych w celu zabezpieczenia się przed błędami oraz wynikających z retransmisji.
  Naszym zadaniem będzie wyznaczenie nadmiarowości, która nie będzie znacząco wpływać na prędkość transmisji.

Opis symulatora

Generowanie pakietów

Generowanie pakietów odbywa się z wykorzystaniem funkcji „randi()". Można wygenerować dowolną ilość pakietów o wybranym rozmiarze.

Kody detekcyjne

Dane można zakodować przy pomocy następujących kodów detekcyjnych:

- Bit parzystości.
- CRC-32.

Bit parzystości został samodzielnie zaimplementowany (0 oznacza parzystą ilość jedynek, 1 oznacza nieparzystą ilość jedynek).

Do implementacji CRC-32 użyto biblioteki „Communication Toolbox", a dokładniej funkcji „comm.CRCGenerator()" i „comm.CRCDetector()".

Kanały transmisyjne

Dane można wysłać następującymi kanałami transmisyjnymi:

- Binary Symmetric Channel (BSC).
- Model Gilberta.

BSC został zaimplementowany z wykorzystaniem biblioteki „Communication Toolbox", a dokładniej użyto funkcji „bsc()".

Kanał Gilberta zaimplementowano samodzielnie. Kanał może znajdować się w dwóch stanach - dobrym i złym. Opisują go dwa prawdopodobieństwa - przejścia z dobrego stanu do złego (bardzo małe prawdopodobieństwo) i vice versa (bardzo duże prawdopodobieństwo). Jeżeli

kanał znajduje się w stanie dobrym to przesyłane są poprawne bity z prawdopodobieństwem „1 - pierwsze prawdopodobieństwo”, a jeżeli kanał znajduje się w stanie złym to przesyłane są niepoprawne bity z prawdopodobieństwem drugim.

Protokół ARQ

Użyto najprostszego protokołu ARQ – „Stop-and-wait”. Pakiety wysyłane są pojedynczo; jeżeli wykryto błąd następuje retransmisja aż do skutku.

Przygotowanie do uruchomienia

Aby uruchomić symulator wymagane jest posiadanie środowiska MATLAB. Należy pobrać repozytorium np. używając poniższej komendy:

```
>git clone https://github.com/Denaturatus/NiDSC-ARQ.git
```

W celu uruchomienia symulatora należy napisać w linii komend MATLAB-a:

```
>>simulation
```

Generowanie wyników

Plikiem odpowiedzialnym za przeprowadzanie eksperymentu jest plik: „simulation.m”. Parametry symulacji można modyfikować zmieniając wartości następujących zmiennych:

```
packetAmount = 20;
packetSize = 30;
errorProbability = 0.05;
goodToBadProbability = 0.02;
badToGoodProbability = 0.8;
codingType = 'PB';  % 'CRC32' or 'PB'
channel = 'GILBERT'; % 'BSC' or 'GILBERT'
loopRepetitions = 100;
```

Domyślnie wyświetlanymi wynikami są BER jaką "widzi" użytkownik i całkowita nadmiarowość transmisji:

```
userBitErrorRateAvg = errorAmountAvg / nAvg;
transmissionRedundancyAvg = (redundantBitsFromCodingAvg +
redundantBitsFromRetransmissionAvg)/ allBitsSentAvg;
disp(['BER uzytkownika: ', num2str(userBitErrorRateAvg)]);
disp(['Calkowita nadmiarowosc transmisji: ',
num2str(transmissionRedundancyAvg)]);
```

Można wyświetlić dodatkowe wyniki odkomentowując następujące linijki kodu:

```
%{
disp(['Ilosc wyslanych pakietow: ', num2str(nAvg)]);
disp(['Ilosc pakietow z bledem: ', num2str(errorAmountAvg)]);
disp(['Wszystkie przeslane bity: ', num2str(allBitsSentAvg)]);
disp(['Nadmiarowe bity wynikające z retransmisji: ',
num2str(redundantBitsFromRetransmissionAvg)]);
disp(['Nadmiarowe bity wynikające z kodowania: ',
num2str(redundantBitsFromCodingAvg)]);
%}
```

Organizacja eksperymentu symulacyjnego

Wykonane zostały następujące pomiary:

- BER blokowy jaki „widzi” użytkownik w zależności od kodu detekcyjnego i jakości kanału.
- Nadmiarowość transmisji wynikającą z nadmiaru kodowego i powtórzeń transmisji ARQ.

W przypadku wyznaczania BER-u i nadmiarowości dla różnych rozmiarów i ilości pakietów przyjęto, że kanałem transmisyjnym będzie BSC o współczynniku wystąpienia błędu równym 0,005. Wyniki uśredniano dla 100 powtórzeń w przypadku bitu parzystości, a dla CRC-32 wykonano 20 repetycji.

W przypadku mierzenia jakości kanału przyjęto stałe wartości rozmiaru i ilości pakietów: 30x20 (dla 100 powtórzeń).

Wyniki

BER blokowy:

- **Bit parzystości:**

BER jaką widzi użytkownik						
Rozmiar/Ilość pakietów	**1**	**5**	**10**	**20**	**50**	**100**
10	0,029126	0,05303	0,057493	0,052582	0,051953	0,051323
20	0,10714	0,11817	0,10953	0,11032	0,094531	0,096739
30	0,13043	0,13345	0,1327	0,13119	0,13778	0,13964
40	0,15966	0,17081	0,16805	0,15966	0,16694	0,16437
50	0,21875	0,20886	0,19936	0,21136	0,19743	0,20319
60	0,23664	0,2163	0,22601	0,22571	0,22613	0,22708
70	0,24812	0,24812	0,25484	0,25926	0,25573	0,24981
80	0,28571	0,27954	0,27378	0,27431	0,27442	0,27808
90	0,32432	0,29078	0,30652	0,29898	0,30575	0,29992
100	0,29577	0,33155	0,31129	0,31271	0,31703	0,3175

Tabela 1. BER blokowy podczas transmisji z wykorzystaniem bitu parzystości

- **CRC-32:**

BER jaką widzi użytkownik						
Rozmiar/Ilość pakietów	**1**	**5**	**10**	**20**	**50**	**100**
10	0,16667	0,18831	0,18567	0,19614	0,1862	0,17601
20	0,23077	0,20635	0,24242	0,20949	0,23664	0,2272
30	0,23077	0,25373	0,23077	0,23372	0,28673	0,26632
40	0,28571	0,30556	0,32432	0,31271	0,30265	0,31271
50	0,33333	0,35065	0,33775	0,33333	0,32065	0,33289
60	0,375	0,35897	0,38272	0,38838	0,39173	0,37656
70	0,41176	0,38272	0,39024	0,39394	0,41793	0,40306
80	0,44444	0,42529	0,40828	0,43182	0,42922	0,43477
90	0,47368	0,4898	0,4709	0,46381	0,46121	0,46627
100	0,5	0,48454	0,47644	0,45652	0,48454	0,47982

Tabela 2. BER blokowy podczas transmisji z wykorzystaniem CRC-32

Nadmiar kodowy:

- **Bit parzystości:**

Całkowita nadmiarowość transmisji						
Rozmiar/Ilość pakietów	**1**	**5**	**10**	**20**	**50**	**100**
10	0,11739	0,13912	0,14318	0,13871	0,13814	0,13757
20	0,14966	0,16016	0,15193	0,15269	0,13765	0,13975
30	0,15849	0,1614	0,16067	0,15922	0,1656	0,16739
40	0,18016	0,19104	0,18834	0,18016	0,18726	0,18475
50	0,23407	0,22437	0,21506	0,22682	0,21317	0,21881
60	0,24916	0,22915	0,23869	0,2384	0,23881	0,23975
70	0,25871	0,25871	0,26534	0,26969	0,26621	0,26038
80	0,29453	0,28843	0,28275	0,28327	0,28337	0,287
90	0,33175	0,29857	0,31414	0,30669	0,31338	0,30761
100	0,30275	0,33817	0,31811	0,31952	0,3238	0,32426

Tabela 3. Całkowita nadmiarowość transmisji z wykorzystaniem bitu parzystości

- **CRC-32:**

Całkowita nadmiarowość transmisji						
Rozmiar/Ilość pakietów	**1**	**5**	**10**	**20**	**50**	**100**
10	0,8254	0,83055	0,82992	0,83241	0,83005	0,82762
20	0,72337	0,71398	0,72786	0,71519	0,72563	0,722
30	0,64392	0,65503	0,64392	0,64535	0,671	0,66112
40	0,61706	0,62809	0,63851	0,63206	0,62647	0,63206
50	0,60569	0,61625	0,60838	0,60569	0,59796	0,60542
60	0,60326	0,59281	0,60829	0,61199	0,61417	0,60428
70	0,60611	0,58618	0,59134	0,59388	0,61034	0,60014
80	0,6121	0,59842	0,58627	0,60308	0,60123	0,60519
90	0,61993	0,63182	0,61788	0,61264	0,61073	0,61446
100	0,62879	0,61707	0,61094	0,59585	0,61707	0,6135

Tabela 4. Całkowita nadmiarowość transmisji z wykorzystaniem CRC-32

Jakość kanału:

- **Kanał BSC:**

Bit parzystości		
Prawdopodobieństwo	**BER jaką widzi użytkownik**	**Całkowita nadmiarowość transmisji**
0,0005	0,012833	0,044677
0,001	0,031477	0,06272
0,002	0,057049	0,087466
0,005	0,13978	0,16753
0,01	0,2275	0,25242
0,02	0,36102	0,38163
0,05	0,46809	0,48524

Tabela 5. BER blokowy i całkowita nadmiarowość dla kanału BSC w zależności od jego parametru podczas wykorzystania bitu parzystości

CRC32		
Prawdopodobieństwo	**BER jaką widzi użytkownik**	**Całkowita nadmiarowość transmisji**
0,0005	0,032414	0,54794
0,001	0,064984	0,5637
0,002	0,11072	0,58583
0,005	0,26335	0,65969
0,01	0,4577	0,75373
0,02	0,71538	0,87841
0,05	0,95741	0,99552

Tabela 6. BER blokowy i całkowita nadmiarowość dla kanału BSC w zależności od jego parametru podczas wykorzystania CRC-32

- **Model Gilberta:**

Bit parzystości			
goodToBad	**badToGood**	**BER jaką widzi użytkownik**	**Całkowita nadmiarowość transmisji**
0,0002	**0,8**	0,012833	0,044677
0,0005	**0,8**	0,022005	0,053553
0,001	**0,8**	0,050332	0,080967
0,002	**0,8**	0,094613	0,12382
0,005	**0,8**	0,21384	0,2392
0,01	**0,8**	0,34015	0,36144
0,02	**0,8**	0,43725	0,45541

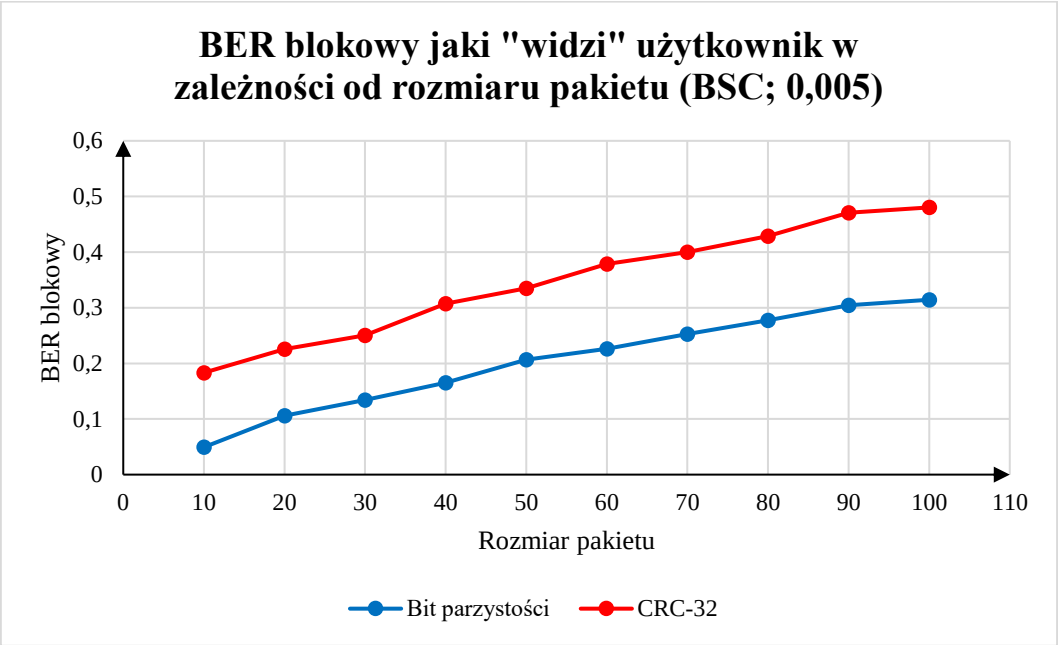
Tabela 7. BER blokowy i całkowita nadmiarowość dla kanału Gilberta w zależności od jego parametrów podczas wykorzystania bitu parzystości

CRC32			
goodToBad	**badToGood**	**BER jaką widzi użytkownik**	**Całkowita nadmiarowość transmisji**
0,0002	**0,8**	0,025341	0,54452
0,0005	**0,8**	0,04943	0,55618
0,001	**0,8**	0,11072	0,58583
0,002	**0,8**	0,1984	0,62826
0,005	**0,8**	0,43915	0,74475
0,01	**0,8**	0,68359	0,86303
0,02	**0,8**	0,90014	0,96781

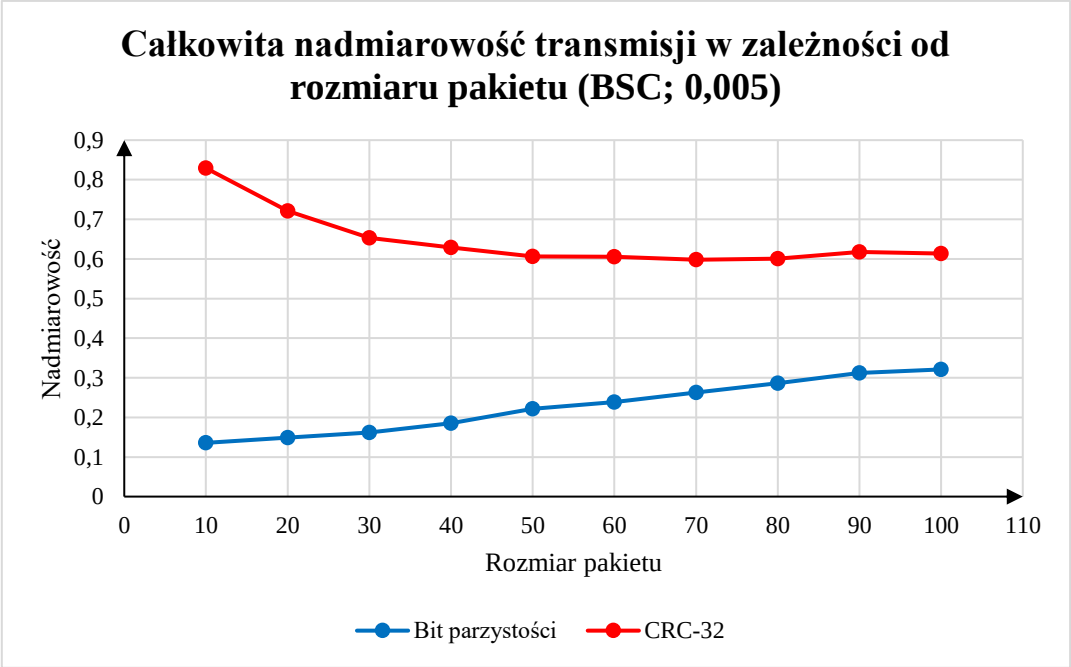
Tabela 8. BER blokowy i całkowita nadmiarowość dla kanału Gilberta w zależności od jego parametrów podczas wykorzystania CRC-32

Analiza wyników

Na podstawie wartości powyższych tabel przygotowano następujące wykresy i ich analizy:

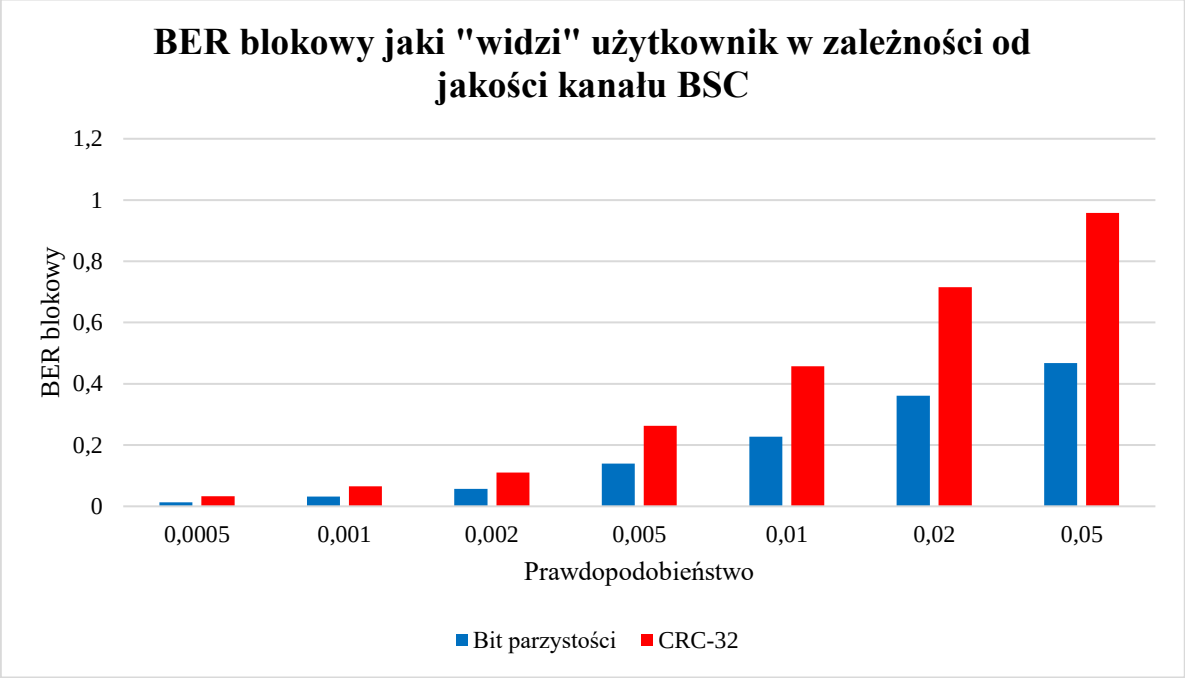


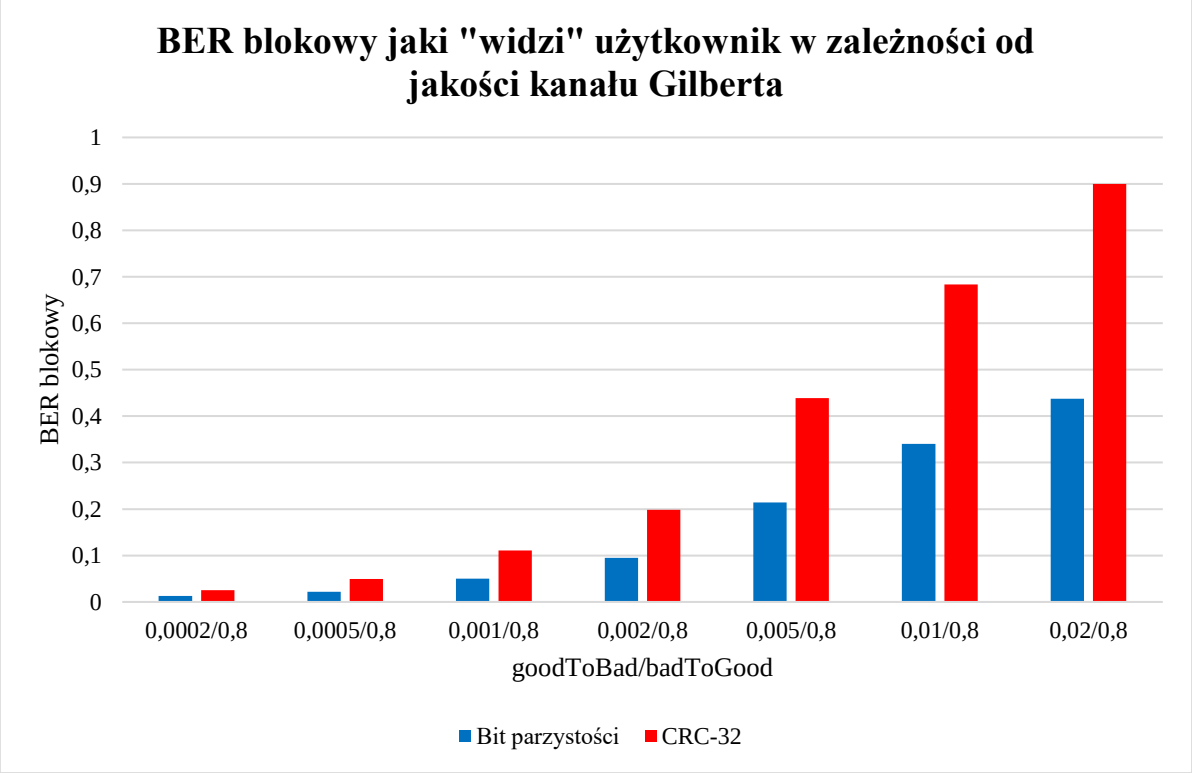
Z wykresu można odczytać, że wraz ze wzrostem rozmiaru pakietu rośnie BER widziany przez użytkownika. Ponadto dla CRC-32 BER jest większy, ponieważ suma kontrolna to dodatkowe bity, które mogą ulec przekłamaniu i tenże kod wyłapuje więcej błędów niż bit parzystości.

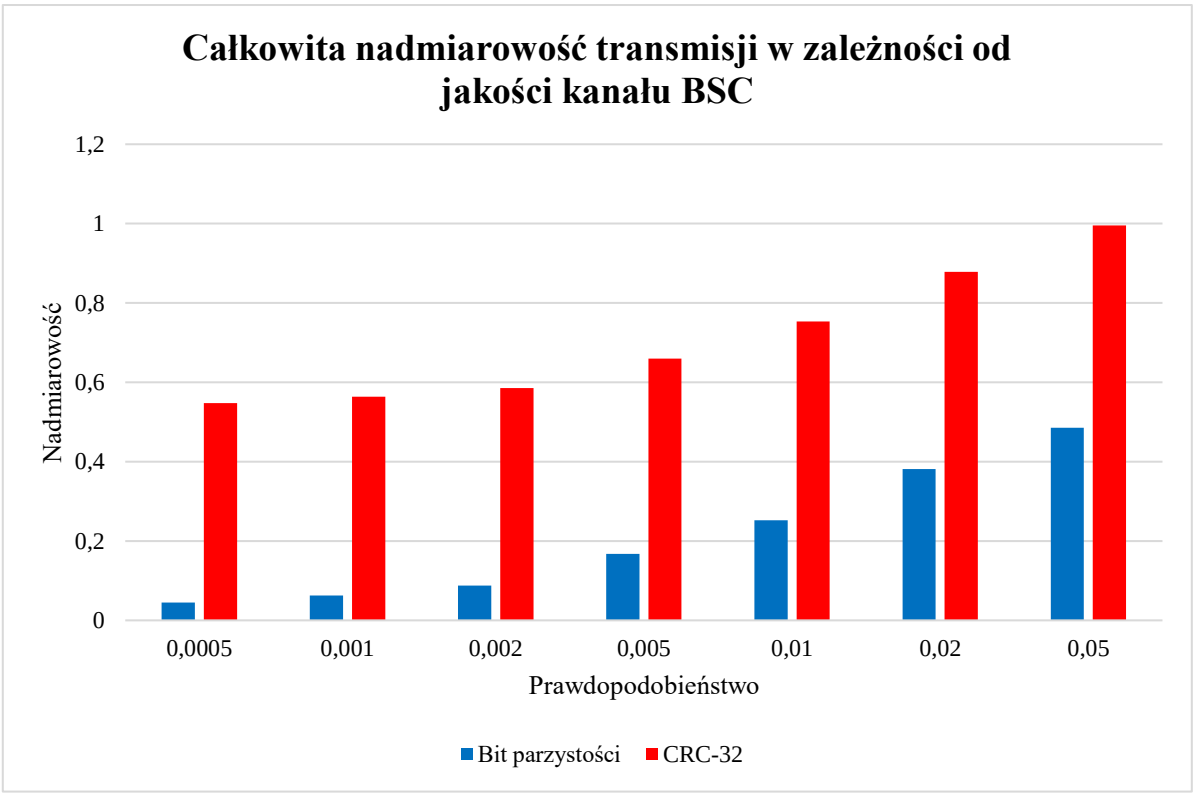


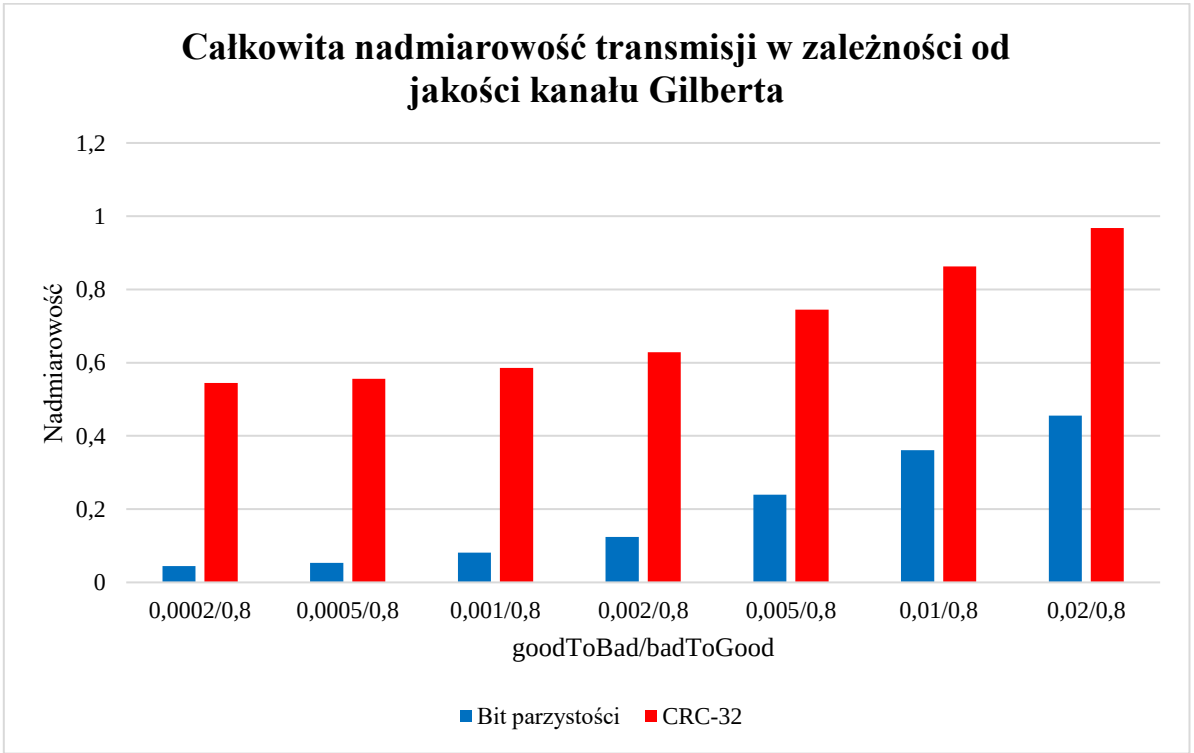
Natomiast całkowita nadmiarowość transmisji stabilizuje się od wartości 40/50 bitów w paczce. Uwzględniając dodatkowo wykres dla BER-u idealnym rozmiarem pakietu w badanym przypadku jest 40 bitów.

Zgodnie z przewidywaniami pogarszająca się jakość kanałów wpływa negatywnie na badane wartości. Mając na uwadze wcześniejszą analizę, CRC-32 wypada „gorzej” niż bit parzystości, ale w rzeczywistości lepiej kontroluje przesył danych.






Całkowita nadmiarowość transmisji w zależności od jakości kanału BSC
1,2
1
0,8
0,6
0,4
0,2
0
Nadmiarowość
0,0005
0,001
0,002
0,005
0,01
0,02
0,05
Prawdopodobieństwo
Bit parzystości
CRC-32


Całkowita nadmiarowość transmisji w zależności od jakości kanału Gilberta
1,2
1
0,8
0,6
0,4
0,2
0
Nadmiarowość
0,0002/0,8
0,0005/0,8
0,001/0,8
0,002/0,8
0,005/0,8
0,01/0,8
0,02/0,8
goodToBad/badToGood
Bit parzystości
CRC-32

Wnioski i uwagi

Projekt umożliwił poszerzenie wiedzy i umiejętności związanych z dwoma tematami: niezawodnością i diagnostyką systemów cyfrowych oraz środowiskiem programistycznym Matlab. Podkreślając wagę wyników powyższych analiz powtarzamy, że z kodem CRC-32 wiąże się wysoka nadmiarowość kodowa, ale za to daje pewność poprawnie przesłanych danych w porównaniu do bitu parzystości. Z tego powodu kodowania CRC-32 używa się np. do kontroli przesyłanych bitów nagłówka w modelu TCP/IP; dokładniej protokole IPv4, gdzie wymagany jest wysoki stopień kontroli poprawności przesyłanych danych. Dobieranie odpowiednich narzędzi jest niezwykle ważne przy realizacji tego rodzaju eksperymentów – środowisko Matlab okazało się perfekcyjne, ponieważ kod jest wyjątkowo przejrzysty i krótki. Program jednakże nie wykonuje się zbyt szybko, co w tym przypadku nie miało za dużego znaczenia.

Należy również pamiętać o tym, że warto korzystać z gotowych funkcji (np. kodowanie CRC-32), ponieważ są one odpowiednio przetestowane i przyjemne w użyciu. Z drugiej strony nie należy się bać implementacji prostszych funkcjonalności takich jak kodowanie bitem parzystości.