

## ***Niezawodność i diagnostyka układów cyfrowych***

*Temat: ARQ - Automatic Repeat Request*

***Spis treści:***

- I. Opis zadania**
- II. Opis działania programu**
- III. Analiza przesyłu plików**
  - a) Protokół STOP-AND-WAIT**
  - b) Protokół GO-BACK-N**
- IV. Porównanie protokołów STOP-AND WAIT i GO-BACK-N**
- V. Podsumowanie i wnioski**
- VI. Bibliografia**

## I. Opis zadania

Zbadanie dwóch modeli kanałów dyskretnych: Binary Symetric Channel (błędy pojedyncze), model Gilberta (błędy grupowe)

Użycie minimum dwóch protokołów **STOP-AND-WAIT** i **GO-BACK-N**.

Wykrywanie błędów w oparciu o sumę kontrolną i CRC32.

Zbadanie własności kilku innych kodów np kod 2 z 5. Możliwość dodania korekcji.

## II. Opis programu

Po uruchomieniu programu wyświetlają się wszystkie dostępne opcje.

- Co chcesz zrobić?

  1. Wczytaj plik
  2. Ustaw prawdopodobieństwo błędu
  3. Przesyłaj plik za pomocą StopAndWait
  4. Przesyłaj plik za pomocą GoBackN
  5. Zapisz plik
  6. Wyświetl pomiary
  0. Wyjście

1. Menu programu.

Następnie do wyboru mamy dwa pliki do wczytania:

Lena.bmp



Rejs.bmp



```
if (choice == '1'):
    print("\nPodaj nazwę pliku: ")
    nazwa = input()
    obraz = loadImage(nazwa)
    xSize = obraz[0].size
    ySize = int(obraz.size / xSize)
    for line in obraz:
        for element in line:
            pakiety.append(Packet(int(element)))
    print("\nObraz załadowany pomyślnie!")
```

2. Wczytanie obrazu.

Kolejnym krokiem jest wprowadzenie prawdopodobieństwa wystąpienia błędu.

```
elif (choice == '2'):
    print("Podaj szansę na wystąpienie błędu (0 - 100%): ")
    damageSize = int(input())
    damageSize = damageSize / 100
```

3. Ustawienie prawdopodobieństwa błędu.

Po wyborze opcji nr 3 zaczyna się przesyłanie pliku przez protokół STOP-AND-WAIT.

```
elif (choice == '3'):
    notEnd = True
    canSend = True
    iterator = 0
    packetsSent = 0
    packetsReceived = 0
    packetsRetransmitted = 0
    damagedPacketsSent = 0
    sent = []
    kolejka.append(0)
    kolejka.append(0)
    kolejka.append(0)
    kolejka.append(0)
    kolejkaZwrotna.append(0)
    kolejkaZwrotna.append(0)
    kolejkaZwrotna.append(0)
    kolejkaZwrotna.append(0)
    print(len(kolejka))
    while notEnd:
        if kolejkaZwrotna[len(kolejkaZwrotna) - 1] == 2:
            packetsSent = packetsSent - 1
            print("Otrzymano polecenie retransmiji")
            packetsRetransmitted = packetsRetransmitted + 1
            canSend = True
        if kolejkaZwrotna[len(kolejkaZwrotna) - 1] == 1:
            print("Otrmano żądanie kolejnego pakietu")
            canSend = True
        kolejkaZwrotna.pop()
        if canSend:
            toSend = copy.deepcopy(pakiety[packetsSent])
            toSend.damageData(damageSize)
            packetsSent = packetsSent + 1
            print("Wyslano pakiet nr " + str(packetsSent))
        if toSend != 0:
            sent.insert(0, toSend)
            toSend = 0
            canSend = False
            kolejka.insert(0, 1)
```

Ciąg dalszy STOP-AND-WAIT

```
else:
    kolejka.insert(0, 0)

if kolejka[len(kolejka) - 1] == 1:
    recieved = sent[0]
    sent.pop()
    if not recieved.checkIfDamaged():
        if recieved.checkIfDamagedCRC():
            damagedPacketsSent = damagedPacketsSent + 1

    packetsReceived = packetsReceived + 1
    zmienna = 0
    for i, bit in enumerate(recieved.data):
        zmienna = zmienna + recieved.data[i] * pow(2, Packet.PACKET_SIZE - 1 - i)
    efektPrawieKoncowy.append(zmienna)
    kolejkaZwrotna.insert(0, 1)
    print("Otrzymano pakiet nr " + str(packetsReceived))
    print("Wysyłam żądanie następnego pakietu")
    elif recieved.checkIfDamaged():
        kolejkaZwrotna.insert(0, 2)
        packetsRetransmitted = packetsRetransmitted + 1
        print("Pakiet uszkodzony! Wysyłam żądanie retransmisji")
    print("Upłynęło jednostek czasu " + str(iterator))
elif kolejka[len(kolejka) - 1] == 0:
    # kolejkaZwrotna.append(0)
    kolejkaZwrotna.insert(0, 0)
    kolejka.pop()
    iterator = iterator + 1
    if packetsReceived == obraz.size:
        notEnd = False
print(packetsRetransmitted)
```

4. Protokół STOP-AND-WAIT.

Po wyborze opcji nr 4 zaczyna się przesyłanie pliku przez protokół GO-BACK-N.

```
elif (choice == '4'):
    notEnd = True
    canSend = True
    window = 5
    iterator = 0
    # packetsSent = 0
    neededPacket = 0
    packetsReceived = 0
    packetsRetransmitted = 0
    toResend = []
    sent = []
    kolejka.append(0)
    kolejka.append(0)
    kolejka.append(0)
    kolejka.append(0)
    kolejkaZwrotna.append(0)
    kolejkaZwrotna.append(0)
    kolejkaZwrotna.append(0)
    kolejkaZwrotna.append(0)
    damagedPacketsSent = 0
    Sn = 0          #sequencenumber
    Sb = 0          #sequencebase
    Sm = window-1  #sequencemax
    print(len(kolejka))
    while notEnd:
        if kolejkaZwrotna[len(kolejkaZwrotna) - 1] == 2:
            Sb = copy.deepcopy(toResend[len(toResend) - 1])
            Sn = copy.deepcopy(Sb)
            toResend.pop()
            print("Otrzymano polecenie retransmiji pakietu " + str(Sb))

        if kolejkaZwrotna[len(kolejkaZwrotna) - 1] == 1:
            if Sm < obraz.size:
                Sb = Sb + 1
                Sm = Sm + 1

        if Sn < Sm and Sn < obraz.size:
            toSend = copy.deepcopy(pakiety[Sn])
            toSend.damageData(damageSize)
            Sn = Sn + 1
            print("Wyslano pakiet nr " + str(Sn - 1))

        kolejkaZwrotna.pop()

        if toSend != 0:
            sent.insert(0, toSend)
            toSend = 0
            kolejka.insert(0, 1)
        else:
            kolejka.insert(0, 0)
```

Ciąg dalszy GO-BACK-N

```
if kolejka[len(kolejka) - 1] == 1:
    recieved = copy.deepcopy(sent[len(sent) - 1])
    sent.pop()
    packetNumber = recieved.getIndex()
    # print("PacketNumber: " + str(packetNumber))
    # print("NeededPacket: " + str(neededPacket))
    if not recieved.checkIfDamaged():
        if packetNumber == neededPacket:
            if recieved.checkIfDamagedCRC():
                damagedPacketsSent = damagedPacketsSent + 1
                neededPacket = neededPacket + 1
                packetsReceived = packetsReceived + 1
                zmienna = 0
            for i, bit in enumerate(recieved.data):
                zmienna = zmienna + recieved.data[i] * pow(2, Packet.PACKET_SIZE - 1 - i)
            efektPrawieKoncowy.append(zmienna)
            kolejkaZwrotna.insert(0, 1)
            print("Otrzymano pakiet nr " + str(packetNumber))
            print("Wysyłam żądanie pakietu " + str(neededPacket))
        else:
            kolejkaZwrotna.insert(0, 0)
    elif recieved.checkIfDamaged():
        kolejkaZwrotna.insert(0, 2)
        packetsRetransmitted = packetsRetransmitted + 1
        print("Pakiet uszkodzony! Wysyłam żądanie retransmisji pakietu " + str(neededPacket))
        toResend.insert(0, copy.deepcopy(neededPacket))
    elif kolejka[len(kolejka) - 1] == 0:
        kolejkaZwrotna.insert(0, 0)
    kolejka.pop()
    iterator = iterator + 1
    print("Ułynęło jednostek czasu " + str(iterator))
if len(efektPrawieKoncowy) == obraz.size:
    notEnd = False
```

## 5. Protokół GO-BACK-N.

Po przestaniu pliku można go zapisać.

```
elif (choice == '5'):
    efektKoncowy = np.ndarray((ySize, xSize), np.int32)
    i2 = xSize + 1
    i3 = xSize
    ii = 0
    ii2 = 0
    for i, zmienna2 in enumerate(efektPrawieKoncowy):
        if (i % i2 == i3):
            i2 = xSize
            i3 = 0
            ii = ii + 1
            ii2 = 0
            efektKoncowy[ii][ii2] = np.int32(zmienna2)
            ii2 = ii2 + 1
    saveImage(efektKoncowy, ("sent" + nazwa))
    print("Obraz zapisano pomyślnie!")
```

6. Zapis przesłanego pliku.

Do analizy niezbędne są pomiary.

```
elif (choice == '6'):
    print("\nRozmiar pliku = " + str(obraz.size))
    print("Wymiary: " + str(xSize) + "x" + str(ySize))
    print("Uzycie pasma transmisji w %: " + str((obraz.size + packetsRetransmitted)/iterator*100))
    print("Uszkodzone pakiety przeslane jako poprawne: " + str(damagedPacketsSent))
    print("Ilosc pakietow wyslana ponownie: " + str(packetsRetransmitted))
    print("Ilosc jednostek czasu ktorych wymagalo przeslanie obrazka dla czasu przesylu 4: " + str(iterator))
    print("Prawdopodobienstwo odroczenia bitu w %: " + str(damageSize * 100) + "\n\n")
```

7. Wyniki pomiarów.



### **III. Analiza przesyłu plików**

#### **a) Protokół STOP-AND-WAIT**

##### **Plik Lena.bmp**

Wyniki pomiarów:

- Pierwsze przesłanie:

Rozmiar pliku = 262144

Wymiary: 512x512

Użycie pasma transmisji w %: 13.44139572803445

Uszkodzone pakiety przesłane jako poprawne: 173

Ilość pakietów wysłana ponownie: 42700

Ilość jednostek czasu których wymagało przesłanie obrazka dla czasu przesyłu 4: 2267949

Prawdopodobieństwo odwrócenia bitu w %: 1.0

- Drugie przesłanie:

Rozmiar pliku = 262144

Wymiary: 512x512

Użycie pasma transmisji w %: 18.614468381974483

Uszkodzone pakiety przesłane jako poprawne: 19999

Ilość pakietów wysłana ponownie: 502030

Ilość jednostek czasu których wymagało przesłanie obrazka dla czasu przesyłu 4: 4105269

Prawdopodobieństwo odwrócenia bitu w %: 10.0

- Trzecie przesłanie

Rozmiar pliku = 262144

Wymiary: 512x512

Użycie pasma transmisji w %: 21.318918493856167

Uszkodzone pakiety przesłane jako poprawne: 95263

Ilość pakietów wysłana ponownie: 1256054

Ilość jednostek czasu których wymagało przesłanie obrazka dla czasu przesyłu 4: 7121365

Prawdopodobieństwo odwrócenia bitu w %: 25.0

- Czwarte przesłanie

Rozmiar pliku = 262144

Wymiary: 512x512

Użycie pasma transmisji w %: 21.699530569630493

Uszkodzone pakiety przesłane jako poprawne: 119303

Ilość pakietów wysłana ponownie: 1461364

Ilość jednostek czasu których wymagało przesłanie obrazka dla czasu przesyłu 4: 7942605

Prawdopodobieństwo odwrócenia bitu w %: 30.0

- Piąte przesłanie

Rozmiar pliku = 262144

Wymiary: 512x512

Użycie pasma transmisji w %: 22.82539004415281

Uszkodzone pakiety przesłane jako poprawne: 193207

Ilość pakietów wysłana ponownie: 2489394

Ilość jednostek czasu których wymagało przesłanie obrazka dla czasu przesyłu 4: 12054725

Prawdopodobieństwo odwrócenia bitu w %: 50.0



**Prawdopodobieństwo 1%**



**Prawdopodobieństwo 10%**



**Prawdopodobieństwo 30%**



**Prawdopodobieństwo 50%**

## **Plik Rejs.bmp**

- Pierwsze przesłanie

Rozmiar pliku = 407200

Wymiary: 800x509

Użycie pasma transmisji w %: 18.664939560891757

Uszkodzone pakiety przesłane jako poprawne: 28808

Ilość pakietów wysłana ponownie: 792528

Ilość jednostek czasu których wymagało przesłanie obrazka dla czasu przesyłu 4: 6427709

Prawdopodobieństwo odwrócenia bitu w %: 10.0

- Drugie przesłanie

Rozmiar pliku = 407200

Wymiary: 800x509

Użycie pasma transmisji w %: 20.883482006426107

Uszkodzone pakiety przesłane jako poprawne: 98955

Ilość pakietów wysłana ponownie: 1658560

Ilość jednostek czasu których wymagało przesłanie obrazka dla czasu przesyłu 4: 9891837

Prawdopodobieństwo odwrócenia bitu w %: 20.0

- Trzecie przesłanie

Rozmiar pliku = 407200

Wymiary: 800x509

Użycie pasma transmisji w %: 21.91790983037952

Uszkodzone pakiety przesłane jako poprawne: 170070

Ilość pakietów wysłana ponownie: 2488548

Ilość jednostek czasu których wymagało przesłanie obrazka dla czasu przesyłu 4: 13211789

Prawdopodobieństwo odwrócenia bitu w %: 30.0

- Czwarte przesłanie

Rozmiar pliku = 407200

Wymiary: 800x509

Użycie pasma transmisji w %: 23.526031322978007

Uszkodzone pakiety przesłane jako poprawne: 275811

Ilość pakietów wysłana ponownie: 6092112

Ilość jednostek czasu których wymagało przesłanie obrazka dla czasu przesyłu 4: 27626045

Prawdopodobieństwo odwrócenia bitu w %: 50.0



**Prawdopodobieństwo 10%**



**Prawdopodobieństwo 20%**



**Prawdopodobieństwo 50%**

## **b)      Protokół GO-BACK-N**

### **Plik Lena.bmp**

- Pierwsze przesłanie

Rozmiar pliku = 262144

Wymiary: 512x512

Użycie pasma transmisji w %: 43.27457154000529

Uszkodzone pakiety przesłane jako poprawne: 200

Ilość pakietów wysłana ponownie: 27449

Ilość jednostek czasu których wymagało przesłanie obrazka dla czasu przesyłu 4: 669199

Prawdopodobieństwo odwrócenia bitu w %: 1.0

- Drugie przesłanie

Rozmiar pliku = 262144

Wymiary: 512x512

Użycie pasma transmisji w %: 64.69568503034192

Uszkodzone pakiety przesłane jako poprawne: 20176

Ilość pakietów wysłana ponownie: 730938

Ilość jednostek czasu których wymagało przesłanie obrazka dla czasu przesyłu 4: 1535005

Prawdopodobieństwo odwrócenia bitu w %: 10.0

- Trzecie przesłanie

Rozmiar pliku = 262144

Wymiary: 512x512

Użycie pasma transmisji w %: 81.10522998361695

Uszkodzone pakiety przesłane jako poprawne: 95345

Ilość pakietów wysłana ponownie: 1664119

Ilość jednostek czasu których wymagało przesłanie obrazka dla czasu przesyłu 4: 2375017

Prawdopodobieństwo odwrócenia bitu w %: 25.0

- Czwarte przesłanie

Rozmiar pliku = 262144

Wymiary: 512x512

Użycie pasma transmisji w %: 83.34051992790408

Uszkodzone pakiety przesłane jako poprawne: 119509

Ilość pakietów wysłana ponownie: 1863911

Ilość jednostek czasu których wymagało przesłanie obrazka dla czasu przesyłu 4: 2551046

Prawdopodobieństwo odwrócenia bitu w %: 30.0

- Piąte przesłanie

Rozmiar pliku = 262144

Wymiary: 512x512

Użycie pasma transmisji w %: 88.76303858618033

Uszkodzone pakiety przesłane jako poprawne: 193405

Ilość pakietów wysłana ponownie: 2588078

Ilość jednostek czasu których wymagało przesłanie obrazka dla czasu przesyłu 4: 3211046

Prawdopodobieństwo odwrócenia bitu w %: 50.0



**Prawdopodobieństwo 1%**



**Prawdopodobieństwo 10%**



**Prawdopodobieństwo 30%**



**Prawdopodobieństwo 50%**

### **Rejs.bmp**

- Pierwsze przesłanie

Rozmiar pliku = 407200

Wymiary: 800x509

Użycie pasma transmisji w %: 64.9675428372503

Uszkodzone pakiety przesłane jako poprawne: 28958

Ilość pakietów wysłana ponownie: 1152978

Ilość jednostek czasu których wymagało przesłanie obrazka dla czasu przesyłu 4: 2401473

Prawdopodobieństwo odwrócenia bitu w %: 10.0

- Drugie przesłanie

Rozmiar pliku = 407200

Wymiary: 800x509

Użycie pasma transmisji w %: 78.52326705215076

Uszkodzone pakiety przesłane jako poprawne: 98980

Ilość pakietów wysłana ponownie: 2281238

Ilość jednostek czasu których wymagało przesłanie obrazka dla czasu przesyłu 4: 3423747

Prawdopodobieństwo odwrócenia bitu w %: 20.0

- Trzecie przesłanie

Rozmiar pliku = 407200

Wymiary: 800x509

Użycie pasma transmisji w %: 84.33297199423447

Uszkodzone pakiety przesłane jako poprawne: 170113

Ilość pakietów wysłana ponownie: 3061755

Ilość jednostek czasu których wymagało przesłanie obrazka dla czasu przesyłu 4: 4113403

Prawdopodobieństwo odwrócenia bitu w %: 30.0

- Czwarte przesłanie

Rozmiar pliku = 407200

Wymiary: 800x509

Użycie pasma transmisji w %: 91.4049311523296

Uszkodzone pakiety przesłane jako poprawne: 276142

Ilość pakietów wysłana ponownie: 5235337

Ilość jednostek czasu których wymagało przesłanie obrazka dla czasu przesyłu 4: 6173121

Prawdopodobieństwo odwrócenia bitu w %: 50.0



**Prawdopodobieństwo 10%**



**Prawdopodobieństwo 20%**



**Prawdopodobieństwo 50%**

#### IV. Porównanie protokołów STOP-AND WAIT i GO-BACK-N

Plik Lena.bmp

Prawdopodobieństwo odwrócenia bitu w %: 10

Protokół	Użycie pasma transmisji w %	Uszkodzone pakiety przesłane jako poprawne	Ilość pakietów wysłana ponownie	Ilość jednostek czasu
STOP-AND-WAIT	18.6	19999	502030	4105269
GO-BACK-N	64.7	20176	730938	1535005

Protokół GO-BACK-N:

- użył o 46.1% więcej pasma transmisji
- przesłał 177 więcej uszkodzonych pakietów jako poprawne
- ponownie wysłał o 228908 pakietów więcej
- przesłał obraz szybciej o 2570264 jednostek czasu

Plik Lena.bmp

Prawdopodobieństwo odwrócenia bitu w %: 30

Protokół	Użycie pasma transmisji w %	Uszkodzone pakiety przesłane jako poprawne	Ilość pakietów wysłana ponownie	Ilość jednostek czasu
STOP-AND-WAIT	21.7	119303	1461364	7942605
GO-BACK-N	83.3	119509	1863911	2551046

Protokół STOP-AND-WAIT:

- użył o 61,6% mniej pasma transmisji
- przesłał 206 mniej uszkodzonych pakietów jako poprawne
- ponownie wysłał o 402547 pakietów mniej
- przesłał obraz wolniej o 5391559 jednostek czasu



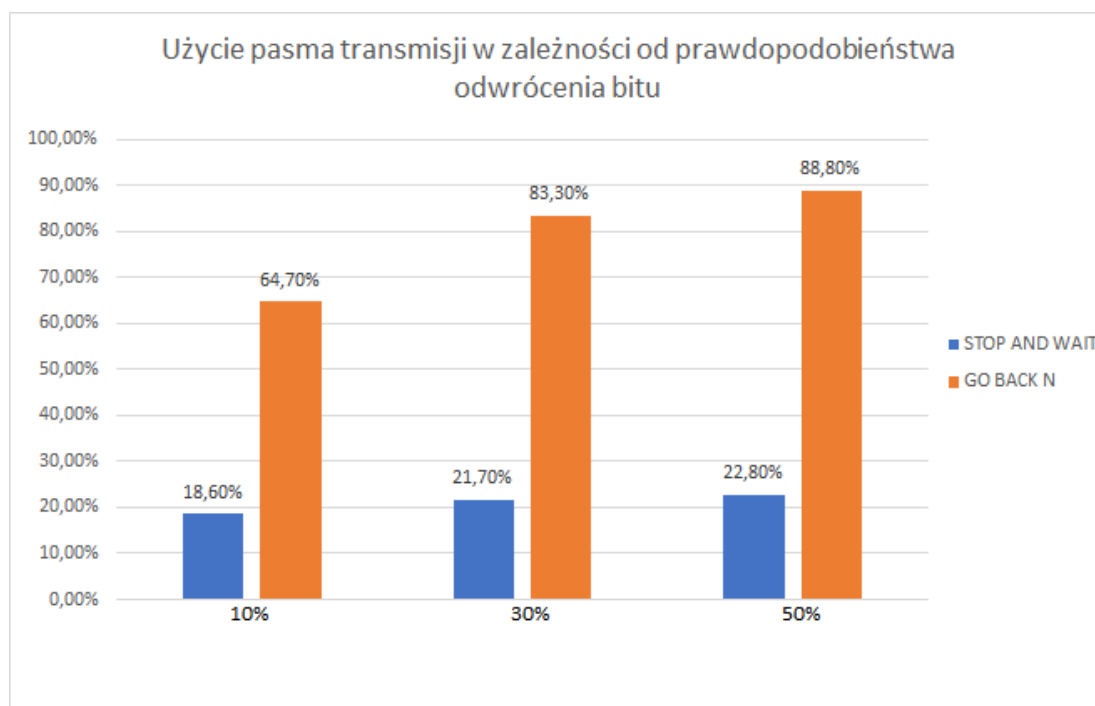
Plik Rejs.bmp

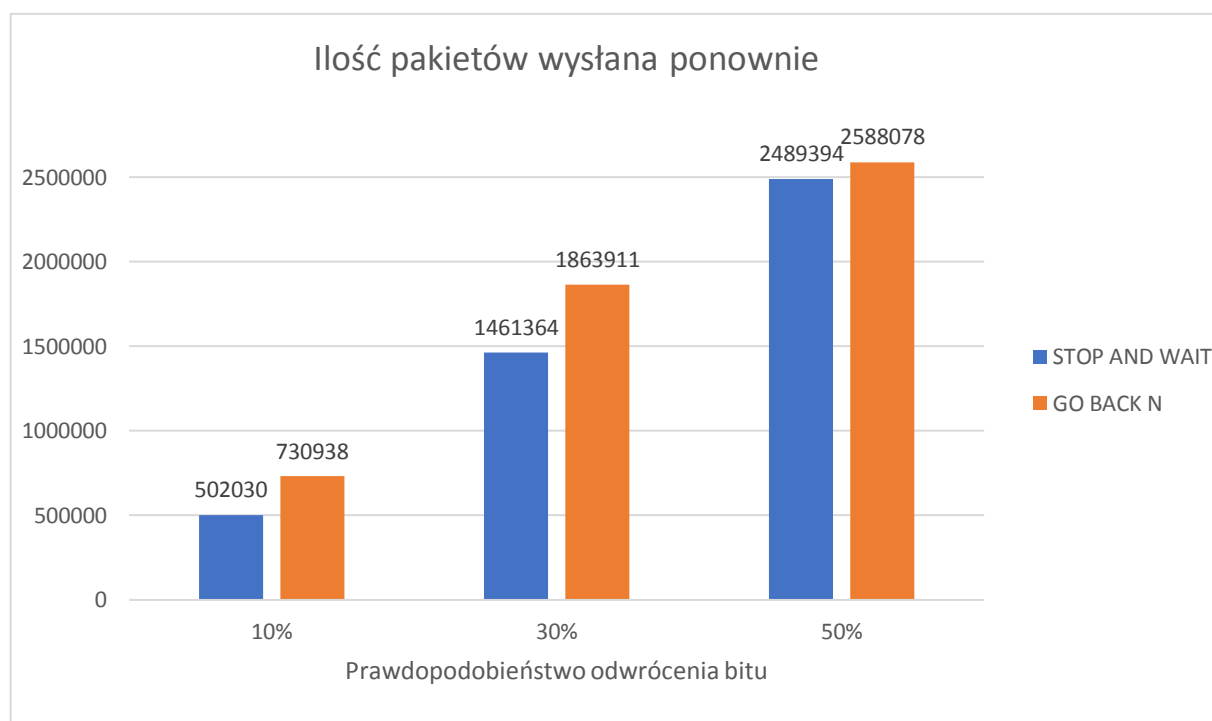
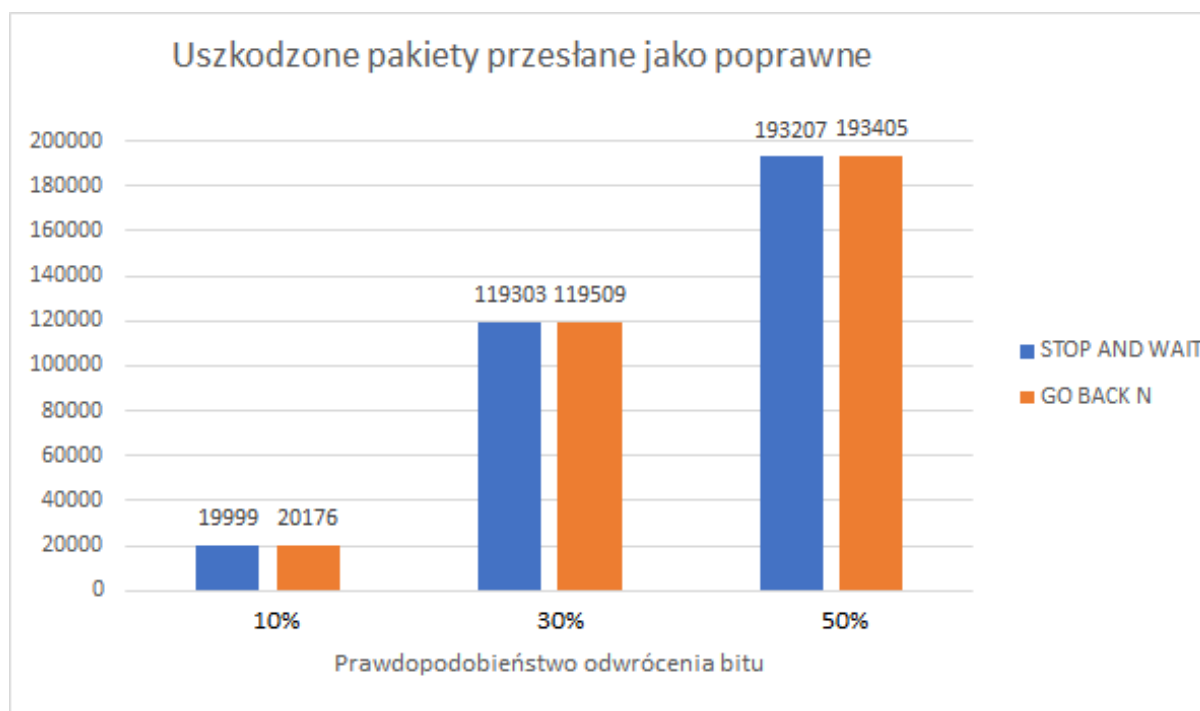
Prawdopodobieństwo odwrócenia bitu w %: 50

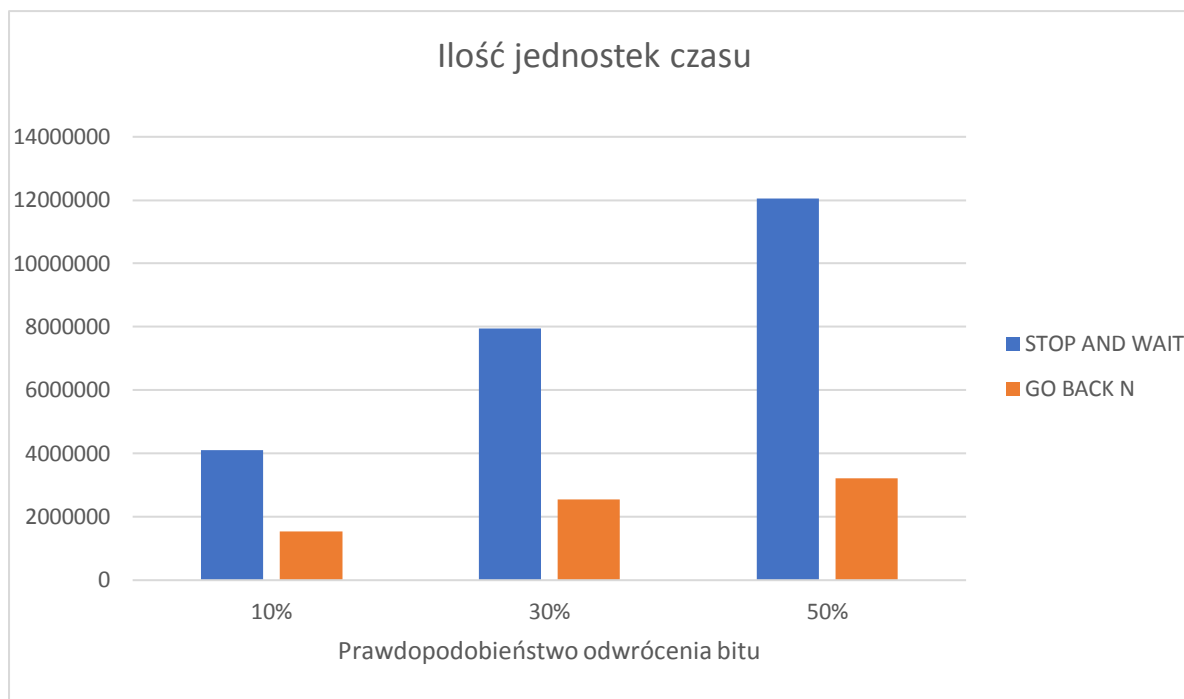
Protokół	Użycie pasma transmisji w %	Uszkodzone pakiety przesłane jako poprawne	Ilość pakietów wysłana ponownie	Ilość jednostek czasu
STOP-AND-WAIT	23.5	275811	6092112	27626045
GO-BACK-N	91.4	276142	5235337	6173121

Protokół STOP-AND-WAIT:

- użył o 67,9% mniej pasma transmisji
- przesłał 331 mniej uszkodzonych pakietów jako poprawne
- ponownie wysłał o 856775 pakietów więcej
- przesłał obraz wolniej o 21452924 jednostek czasu







## V. Wnioski

Uzyskując powyższe dane pomiarowe możemy jednoznacznie stwierdzić, że protokół GO-BACK-N jest niezaprzeczalnie bardziej wydajny niż STOP-AND-WAIT, co niestety jest okupione retransmisją części pakietów bez potrzeby. Widać też, że wraz ze wzrostem prawdopodobieństwa błędu coraz więcej uszkodzonych pakietów jest akceptowanych jako poprawne. Spowodowane jest to faktem, że uszkodzony pakiet zwraca taką samą sumę kontrolną jak pakiet nieuszkodzony. Uważamy, że realizacja projektu była ciekawym i rozwijającym przedsięwzięciem, wartym realizacji.