Autor: Kamil Król

Numer indeksu: 244949

Sprawozdanie

## **Wprowadzenie**

Do realizacji zadań wybrałem język Python3.7.

## **Zadanie 1**

* 1. **Opis programu**

Stworzony przeze mnie program umożliwia zakodowanie wiadomości do ramki i jej odkodowanie. Ramka rozpoczyna się i kończy ustaloną sekwencją bitów: 01111110. Aby uniknąć w dalszej części ramki sekwencji zamykającej/otwierającej stosuje się tzw. rozpychanie bitów. Zapobiega ono błędom polegającym na zinterpretowaniu części wiadomości jako sekwencji zamykającej ramkę. Po sekwencji rozpoczynającej ramkę znajduje się rozepchana wiadomość, a następnie rozepchane pole kontrolne crc. Funkcję obliczającą wartości crc wziąłem z biblioteki zlib. Na końcu ramki znajduje się sekwencja zamykająca ramkę.

* 1. **Prosty test programu**

Wybrałem krótką wiadomość, która zawiera w sobie sekwencję zamykającą/otwierająca i ciąg jedynek długości 10.

Wiadomość:

1110011001111110011111111110

Ręcznie zakodowana wiadomość (bez pola crc):

011111101110011001111101001111101111100[32 bity crc]01111110

Ramka obliczona przez program:

0111111011100110011111010011111011111000000001011001011101100100010110101111110

Program zadziałał zgodnie z oczekiwaniami.

Odkodowana wartość:

1110011001111110011111111110

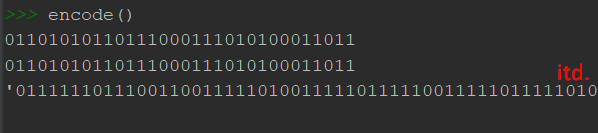
Program poprawnie odkodował wiadomość.

* 1. **Test ze zmianą zaszyfrowanej ramki**

Poniżej kodowana wiadomość.



Kodowanie wiadomości.



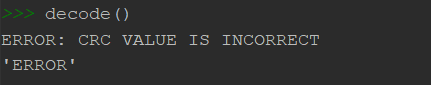
Ramka powstała na skutek powyższego kodowania



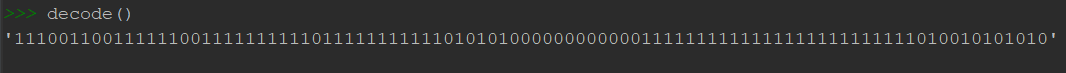
Wprowadzamy zmianę w ramce.



Próba odkodowania.



Odkodowanie poprawnej ramki.



* 1. **J**

## **Zadanie 2**

* 1. **Opis**

Kanał odpowiada za propagowanie sygnału. Dana stacja może nadać sygnał jeżeli widzi, że kanał komunikacyjny (komórka nad nią) jest wolny. Każda stacja ma określony stopień gadatliwości. W sytuacji kiedy zajdzie kolizja dwie stacje, które brały w niej udział losują szczeliny czasowe zgodnie ze wzorem.

Gdzie R(x) to losowa liczba całkowita z zakresu [0,x], k to ilość prób, a t to minimalna długość komunikatu.

Jeżeli ilość prób przekroczy 15 to uznaję, że kanał uległ przeciążeniu.

Po wylosowaniu szczeliny stacje czekają.

* 1. **Proponowany model kanału**

Wybrałem kanał długości 15, ze stacjami na pozycjach 2 i 7.

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

A B

Fragment symulacji, podczas której doszło do kolizji.

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 B 0 0 0 0 0 0 0

0 0 0 0 0 0 B B B 0 0 0 0 0 0

0 0 A 0 0 B B B B B 0 0 0 0 0

0 A A A B B B B B B B 0 0 0 0

A A A # # B B B B B B B 0 0 0

A A # # # # B B B B B B B 0 0

A # ! # # # # B B B B B B B 0

# ! ! ! # # # # B B B B B B B

! ! ! ! ! # # ! # B B B B B B

* 1. **Wpływ ilości stacji na ilość kolizji**

Długość kanału to 30. Stacje są rozłożone równomiernie.

|  |  |
| --- | --- |
| Ilość stacji | Ilość kolizji |
| 2 | 5 |
| 4 | 15 |
| 6 | 21 |
| 8 | 29 |
| 10 | 34 |

* 1. **Wpływ długości kanału na ilość kolizji**

Umieściłem 2 stacje na końcach kanałów o różnych długościach.

|  |  |
| --- | --- |
| Długość kanału | Ilość kolizji |
| 30 | 6 |
| 100 | 3 |
| 200 | 2 |
| 400 | 1 |
| 800 | 1 |
| 1000 | 1 |

Wyniki są inne od tych jakie oczekiwałem. Wynika to z faktu, że czas każdej próby jest zawsze taki sam (3000 interwałów czasowych), w związku z tym przy dłuższych kanałach wylosowane szczeliny czasowe są dłuższe, a więc stacje dłużej pozostają bezczynne.

* 1. **Wpływ gadatliwości stacji na ilość kolizji**

Do testu przyjąłem długość kanału równą 50. Gadatliwość stacji to prawdopodobieństwo tego ze w danym interwale czasowym stacja zacznie nadawać. Stacji jest 3 odpowiednio na pozycjach 2, 15, 47.

|  |  |  |
| --- | --- | --- |
| Gadatliwość wszystkich stacji | Ilość kolizji | Ilość sukcesów |
| 0.05 | 9 | 22 |
| 0.2 | 13 | 22 |
| 0.5 | 14 | 20 |