

POLITECHNIKA WROCŁAWSKA

ANNA MODRZEJEWSKA

Sprawozdanie nr 3

Technologie Sieciowe

17 października 2018

1 Cel

Napisanie programu ramkującego zgodnie z zasadą rozpychania bitów, weryfikującego poprawność ramki metodą CRC i odczytującego strumień bitów na podstawie otrzymanej ramki.

Napisanie programu symulującego ethernetową metodę dostępu do medium transmisyjnego (CSMA/CD - Carrier Sense Multiple Access / with Collision Detection)

2 Realizacja

2.1 Zadanie 1.

Uwaga. Rozpychanie bitów w tym zadaniu oznacza wstawienie 0-bitu po ciągu pięciu jedynek pod rząd. W ramce 0-bit oznacza sygnał, a 1-bit oznacza brak zmiany. Zbyt długi ciąg 1-bitów grozi temu, że zegary wysyłającego i odbiorcy stracą synchronizację. Rozpychanie bitów zapewnia, że na 6 przesłanych bitów przynajmniej 1 będzie sygnałem, co umożliwi ponowną synchronizację. Ciąg sześciu jedynek pod rząd pojawia się tylko na początku ramki (tzw. flaga), co pozwala odbiorcy łatwo rozróżnić flagi od przesłanych danych.

2.1.1 Ramkowanie

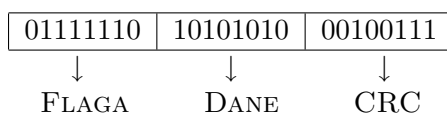
Program otrzymuje na wejściu ciąg binarny symulujący strumień bitów, następnie dzieli go po 8 bitów, dla każdego wylicza 8-bitowy CRC, rozpycha bity i zapisuje w ramkę.

Uwaga. Algorytm obliczania sum kontrolnych CRC jest opisany tu: https://pl.wikipedia.org/wiki/Cykliczny_kod_nadmiarowy

Przykładowe wywołanie programu:

Strumień bitów (Plik wejściowy)	1010101010101010101011110000010101010010101 011111111000000101011110000001111000011110111
Ramki (Plik wynikowy)	01111110101010101000100111011111101010101000100 111011111101010101100010110011111101100000101 110110011111100101010010111010011111101010101 100010110011111101111101001111101110111111000 00101011011011011111011110000100000100111111 000111100101110000111111000111101100010010111 11101101010011

Pierwsza ramka pliku wynikowego jest postaci:



2.1.2 Odczytywanie

Program realizuje procedurę odwrotną do ramkowania. Otrzymuje na wejściu plik wynikowy z ramkami z poprzedniego podpunktu i odczytuje strumień bitów wg następujących kroków:

- zapisuje do tablicy po kolei ramki, rozdzielając je flagą bitów 01111110
- dla każdej ramki usuwa 0-bit umieszczony przy rozpychaniu bitów
- jeśli w ramce była błędna wartość CRC, kończy działanie z komunikatem o błędnej ramce

- usuwa z każdej ramki 8 końcowych bitów (CRC)

Program, po otrzymaniu pliku wynikowego z poprzedniego podpunktu, zwróci kopię pliku wejściowego z poprzedniego podpunktu.

2.1.3 Weryfikowanie

Sprawdzanie, czy ramka jest poprawna, opiera się na zasadzie porównania, czy każda ramka jest taka sama, co połączenie wycinka ramki bez ośmiu końcowych znaków i wyliczonego dla tego wycinka 8-bitowego cyklicznego kodu nadmiarowego. Pseudokod:

```

1: procedure CHECKFRAME(FRAME)
2:   framelen  $\leftarrow$  length of frame
3:   sub  $\leftarrow$  frame.substring(0, framelen - 8)
4:   crc  $\leftarrow$  computeCRC8(decoded)
5:   concatenation  $\leftarrow$  sub + crc
6:   return frame.isEqualTo(concatenation)

```

2.2 Zadanie 2.

Program symuluje wysyłanie danych przy pomocy protokołu CSMA/CD. Jest to ethernetowa metoda dostępu do danych. Kiedy urządzenie posiada dane, które chce przesłać, nasłuchuje, czy linia transmisyjna jest wolna. Jeśli tak - przesyła je. Istnieje ryzyko, że dwa urządzenia zaczną wysyłać dane w tym samym momencie, lub zanim sygnał z pierwszego węzła dojdzie do drugiego. Wtedy następuje kolizja i jest wysyłany *JAM*, sygnał zagłuszania, aby urządzenia wiedziały, że dane są nieważne i przerwały transmisję. Dokładna procedura radzenia sobie z kolizją jest opisana tu: https://en.wikipedia.org/wiki/Carrier-sense_multiple_access_with_collision_detection

Program uruchamia 3 wątki wypisujące:

1. [START1]Come as you are[END1]
2. [START2]Wait a minute![END2]
3. [START3]In the end[END3]

Na początku wątek czeka 4 ms, aby zacząć pracę. Następnie pobiera aktualną długość ciągu znaków do wysłania (*oldsize*) i dodaje do niego pierwszy znak zdania, które ma wysłać. Jeśli teraz aktualna długość ciągu znaków do wyświetlenia = *oldsize* + 1, dodaje kolejny znak i znowu sprawdza. Jeśli przy którymś znaku rozmiar nie będzie się zgadzać, oznacza to, że dwa wątki naraz dodały swoje znaki, jest wyświetlany napis [JAM], licznik kolizji jest zwiększony, wątek czeka dwa razy dłużej niż poprzednio i zostaje podjęta kolejna próba przesłania całego zdania. Gdy czas oczekiwania wątku dojdzie to 512 ms, będzie podjętych 10 prób. Po 10 nieudanej próbie wątek zakończy pracę.

Przykładowy wynik działania programu:

<pre> [ST [[JAM] [JAM] [STAR [[JAM] [JAM] [START1] Come [[JAM] [JAM] [START3] In the end [↳ END3] [START1] Come as [[JAM] [JAM] [START2] Wait a minute! [END2] [START1 ↳] Come as you are [END1] </pre>
--

3 Wnioski

- Aby zmniejszyć ryzyko pomyłki przy wysyłaniu danych, używa się zasady rozpychania bitów, a aby zweryfikować poprawność ramki, wysyła się także CRC
- Ramka składa się z flagi, danych oraz CRC
- Protokół CSMA/CD był używany w ethernetowych sieciach lokalnych
- Urządzenia posługujące się tym protokołem wysyłają swoje dane wtedy, gdy w łączu nie ma innych komunikatów, a w razie kolizji wysyłają sygnał JAM zamiast danych i czekają, by ponownie przesłać dane.

```
int x = 5;  
int framelen = frame.length();  
String sub = frame.substring(0, framelen - 8);
```