Wojciech Kałuziński
Technologie sieciowe LISTA3

ZAD 1

„*Napisz program ramkujący zgodnie z zasadą "rozpychania bitów" (podaną na wykładzie), oraz weryfikujący poprawność ramki metodą CRC . Program ma odczytywać pewien źródłowy plik tekstowy 'Z' zawierający dowolny ciąg złożony ze znaków '0' i '1' (symulujący strumień bitów) i zapisywać ramkami odpowiednio sformatowany ciąg do innego pliku tekstowego 'W'. Program powinien obliczać i wstawiać do ramki pola kontrolne CRC - formatowane za pomocą ciągów złożonych ze znaków '0' i '1'. Napisz program, realizujący procedure odwrotną, tzn. który odczytuje plik wynikowy 'W' i dla poprawnych danych CRC przepisuje jego zawartość tak, aby otrzymać kopię oryginalnego pliku źródłowego 'Z'.*”

Opis programu:

Program został napisany w języku Python i składa się z 5 plików:

- settings.py – główne parametry programu
- my_encod.py – tworzenie randomowej wiadomości do zakodowania, uruchamianie funkcji kodującej, zapis zakodowanej wiadomości do pliku
- my_decod.py – odczyt zakodowanej wiadomości z pliku, uruchamianie funkcji dekodującej, zapis zdekodowanej wiadomości do pliku
- main_func – funkcja stosująca metodę CRC z biblioteki zlib, funkcje odpowiedzialne za „rozpychanie” wiadomości, funkcja dodająca znaczniki końca i początku wiadomosci oraz funkcje odwrotne
- testing – testowanie funkcji zawartych w pliku main_func

(randomowa wiadomość testowa jest zapisywana w pliku *dane_wejsciowe.txt*, wiadomość zakodowana w *zakodowane.txt, a* zdekodowana w *zdekodowane.txt*

Opis CRC:

System sum kontrolnych wykorzystywany do wykrywania przypadkowych błędów pojawiających się podczas przesyłania i magazynowania danych binarnych.
n-bitowy cykliczny kod nadmiarowy (n-bitowy CRC) definiuje się jako resztę z dzielenia ciągu danych przez (n+1)-bitowy dzielnik CRC, zwany również wielomianem CRC.
Nadawca (lub np. proces zapisujący dane na dysk) oblicza CRC dla wiadomości, którą chce wysłać i dodaje je do wysyłanych danych. Odbiorca (lub np. proces odczytujący dane z dysku) również oblicza sumę kontrolną danych, które otrzymał i porównuje ją z wartością CRC wysłaną przez nadawcę. Jeśli wyniki się różnią, doszło do przekłamania informacji.

„Rozpychanie”:

Wiadomość, która jest przygotowywana do wysłania otrzymuje znaczniki początku i końca, które wyglądają następująco: „01111110”, dlatego aby program dekodujący, odczytując wiadomość np.: „...0011111100101...” nie pomylił przytoczonego fragmentu z końcem wiadomości, program kodujący po każdych 5 jedynkach, będących częścią komunikatu, wstawia - „rozpycha” dodatkowe ‘0’. Należy również uwzględnić operację odwrotną w programie dekodującym.

ZAD2

„*Napisz program (grupę programów) do symulowania ethernetowej metody dostępu do medium transmisyjnego (CSMA/CD). Wspólne łącze realizowane jest za pomocą tablicy: propagacja sygnału symulowana jest za pomocą propagacji wartości do sąsiednich komórek. Zrealizuj ćwiczenie tak, aby symulacje można było w łatwy sposób testować i aby otrzymane wyniki były łatwe w interpretacji.*”

Opis programu:
Program został napisany w języku Python i składa się z 2 plików:

- main_func.py – klasa Router, reprezentująca parametry i akcje urządzenia, które wysyła i odbiera sygnał, funkcja main, która przeprowadza symulację przesyłu danych
- Ethernet.py – klasa Ethernet, która reprezentuje parametry i akcje medium, w którym odbywa się symulacja – przesył danych

(po uruchomieniu programu należy podać *rozmiar wysyłanego pakietu* oraz *rozmiar Ethernetu* – odległość na jaką chcemy przesłać wiadomość, po przerwaniu działania programu zostaną zaprezentowane statystyki)

Po uruchomieniu programu i wprowadzeniu danych, tworzone są obiekty: lewy i prawy router oraz ethernet.
Routery zaczynają nadawać swoją wiadomość według odpowiedniego **algorytmu.** Gdy program wykryje kolizję, przerywa nadawanie przez routery za pomocą wyjątku.

Procedura CSMA/CD:
Jest to protokół służący do zapewnienia bezbłędnej dwukierunkowej transmisji danych między różnymi komputerami (kartami sieciowymi), przez pewne wspólne medium (sieć)
• Nasłuchiwanie łącza
• Dane wysyłane gdy nikt inny nie wysyła
• Węzeł który nic nie wysyła nasłuchuje
• Wykryta kolizja – wstrzymanie wysyłania danych, poinformowanie o kolizji (wyższy poziom prądu)
• Wstrzymanie wysyłania na losową długość czasu – zwiększaną w przypadku kolejnych kolizji

Algorytm obliczania czasu oczekiwania w przypadku kolizji:
Gdy nastąpiła kolizja:

- wstrzymaj nadawanie
- wyślij informację o kolizji
- zwiększ licznik kolizji
- poczekaj czas wylosowany z przedziału liczb naturalnych od 0 do (2^licznik kolizji)-1, gdy licznik kolizji jest większy od 10, przestań zwiększać przedział liczb naturalnych, gdy licznik kolizji jest większy od 16 zgłoś błąd połączenia
- ponów procedurę nadawania
- gdy uda się wysłać wiadomość to zresetuj licznik kolizji

Wyniki symulacji:

1.
rozmiar wysyłanego pakietu: 5
rozmiar Ethernetu: 10
wykonano: 20 prób
dostarczono: 14 wiadomości
zanotowano: 6 kolizji
sprawność: 70 %

2.
rozmiar wysyłanego pakietu: 5
rozmiar Ethernetu: 20
wykonano: 20 prób
dostarczono: 13 wiadomości
zanotowano: 7 kolizji
sprawność: 65 %

3.
rozmiar wysyłanego pakietu: 5
rozmiar Ethernetu: 30
wykonano: 20 prób
dostarczono: 11 wiadomości
zanotowano: 9 kolizji
sprawność: 55 %

4.
rozmiar wysyłanego pakietu: 5
rozmiar Ethernetu: 50
wykonano: 20 prób
dostarczono: 4 wiadomości
zanotowano: 16 kolizji
sprawność: 20 %

5.
rozmiar wysyłanego pakietu: 5
rozmiar Ethernetu: 80
wykonano: 20 prób
dostarczono: 2 wiadomości
zanotowano: 18 kolizji
sprawność: 10 %

6.
rozmiar wysyłanego pakietu: 10
rozmiar Ethernetu: 50
wykonano: 20 prób
dostarczono: 3 wiadomości
zanotowano: 17 kolizji
sprawność: 15 %

7.
rozmiar wysyłanego pakietu: 20
rozmiar Ethernetu: 50
wykonano: 20 prób
dostarczono: 1 wiadomości
zanotowano: 19 kolizji
sprawność: 5 %

8.
rozmiar wysyłanego pakietu: 40
rozmiar Ethernetu: 50
wykonano: 20 prób
dostarczono: 1 wiadomości
zanotowano: 19 kolizji
sprawność: 5 %

Wnioski:
Największy wpływ na skuteczność przesyłania wiadomości ma długość medium, w jakim się ona porusza. Długość nadawanego komunikatu nie ma za to większego wpływu na sprawność programu. Oczywistym natomiast jest to, że szansa na to, że po kolizji, uda nam się dostarczyć wiadomość wzrasta.