Technologie Sieciowe Sprawozdanie 3

Michał Fujarewicz

Zadanie 1.

Dwa programy Encoder i Decoder

Encoder wczytuje plik o ze ścieżki podanej jako argument i zapisuje zakodowany do pliku encoded.txt Dane o długości 48 do zakodowania:

Po zakodowaniu, z ustawioną maksymalną ilością danych w ramce na 32:

Na zielono zaznaczono wystąpienia w zakodowanym pliku ciągu "01111110", czyli ramki. Widzimy że program podzielił na dane na dwie części. W zakodowanym pliku poza ramkami nie występują ciągi "1" dłuższe niż 5. W trakcie kodowania do każdego takiego ciągu na końcu zostało dodane zero.

Decoder po podaniu pliku encoded.txt generuje plik decoded.txt o zawartości:

Są to nasze oryginalne dane.

W pliku encoded.txt dopisałem jedno zero wewnątrz drugiej ramki i spróbowałem go ponownie zdekodować.

Zawartość pliku decoded.txt wygląda teraz tak:

10010101010000010101111111111110

Dane z pierwszej ramki zostały zachowane, a z drugiej utracone, a program na wyjściu błędu wyświetlił o komunikat o uszkodzonych danych:

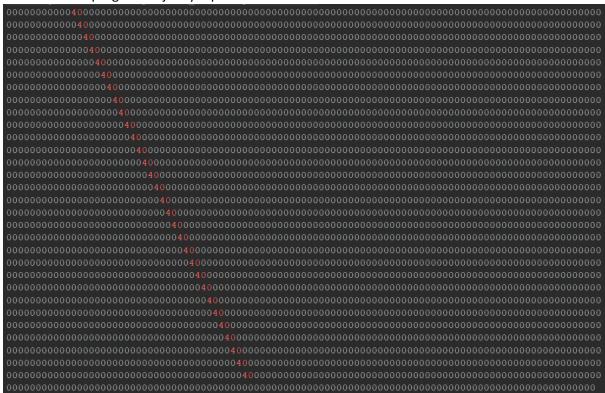
Data Corrupted

Zadanie 2

Program symulujący działanie protokołu CSMA/CD

Program symuluje kabel jako tablicę liczb całkowitych, wartość -1 oznacza brak sygnału, liczba nieujemna oznacza pakiet który musi dostać się w zadane miejsce na tablicy. W każdym kroku program przesuwa pakiet o jeden krok. Jeżeli dwa lub więcej "komputerów" zaczyna transmisję w tym samym czasie wiadomości są utracone, a komputery próbują ponownie wysłać wiadomość po opóźnieniu wylosowanym z zakresu 1 do 2^n, gdzie n to ilość prób wysłania danego pakietu

Uruchomienie programu z jednym pakietem:



Z dwoma pakietami do wysłania przez dwa różne komputery:

000000003000000009000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
$\tt 0000000000000009000000000000000000000$
0000000000000090000000000000000000000
0000000000000090000000000000000000000
$00000000000009\\ 00000000000000000000000$
0000000000009000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
$\tt 0000000000900000000000000000000000000$
$\tt 0000000099000000000000000000000000000$
$\tt 000000000000000000000000000000000000$
$\tt 0000000003000000000000000000000000000$
$\tt 0000000030000000000000000000000000000$
0000000300000000000000000000000000000
0000003000000000000000000000000000000
$\tt 0000030000000000000000000000000000000$
0000300000000000000000000000000000000
0003000000000000000000000000000000000
000000000000000000000000000000000000