

Zadanie 1

- **Cel zadania**
 - Stworzenie programu ramkującego
 - Implementacja zasady rozpychania bitów
 - Weryfikacja poprawności ramki metodą CRC
- **Zasada działania programu dla trybu encode** (decode analogicznie tylko w odwrotnej kolejności):
 - Program czytuje z otrzymanego pliku zadaną wiadomość
 - Gdy wiadomość jest za długa (parametr m), dzieli ją na pomniejsze wiadomości
 - Oblicza długość wiadomości i na tej podstawie ustala długość CRC
 - $\text{dane} < 64$ – crc8
 - $64 \leq \text{dane} < 16384$ – crc16
 - $16384 \leq \text{dane}$ – crc32
 - Długość obliczonego crc jest dopełniana do maksymalnej możliwej otrzymanej długości, czyli:
 - crc8 – ma długość 8
 - crc16 – ma długość 16
 - crc32 – ma długość 32
 - Do danych doklejane jest wyliczone crc
 - Wykorzystywane jest rozpychanie bitów na danych + crc
 - Na początek i koniec ramki wstawiany jest znak początku i końca
- **Budowa ramki:**
 - $01111110 + \text{rozpychanieBitowe}(\text{dane} + \text{crc}(\text{dane})) + 01111110$
- **Sposób uruchomienia programu** (działa dla wersji Pythona: 3.8.2):
 - -t oznacza typ wykonywanej operacji (encode | decode | test), domyślnie encode
 - test – sprawdza czy dla zadanych danych program dobrze koduje i dekoduje
 - decode – dekodowanie wiadomości i zapisywanie jej do wybranego pliku
 - encode – kodowanie wiadomości i zapisywanie jej do wybranego pliku
 - -i plik z danymi, domyślnie input.txt
 - -o plik w którym ma być wynik programu (test nie obsługuje tego parametru), domyślnie output.txt
 - -m maksymalna długość ramki (decode nie obsługuje tego parametru), domyślnie 62

Zadanie 2

- **Cel zadania**
 - Stworzenie działającej symulacji sieci z wykorzystaniem CSMA/CD
 - Wyświetlanie medium transmisyjnego
 - Symulowanie propagowania sygnału za pomocą propagacji wartości w tabeli
- **Zasada działania programu:**
 - Program emuluje medium transmisyjne z wykorzystaniem listy stringów.
 - Routery dodawane są do medium transmisyjnego

- Każdy router posiada dostęp jedynie do ośrodka transmisyjnego i na podstawie swojego wycinka „kabla” ustala czy może coś wysłać
- Długość każdej wiadomości wynosi minimum jakie może w sieci występować, (obliczane jest w metodzie `Wire.generateMessageSize()`)
- Routery upewniają się czy mogą transmitować (kabel musi być w miejscu podłączenia pusty) i po odliczeniu IFG następuje transmisja wiadomości
- Ponieważ wiadomość jest na tyle długa, by osiągnąć najdalej położonego routera przed końcem transmisji, zgodnie z zasadą CD kolizja może nastąpić jedynie w przypadku gdy któryś z routerów nadaje i w tym momencie otrzyma wiadomość
- Wtedy zauważona zostaje kolizja i wysyłany zostaje sygnał zagłuszający (Jam) o długości co najmniej 2x odległość od najdalszych routerów
- Otrzymawszy taki sygnał routery czekają na jego ustanie i losuje czas czekania zgodnie z zasadą, że z każdym niepowodzeniem zwiększa się $x = \begin{cases} 2^i, & i < 11 \\ 2^{10}, & x \geq 11 \end{cases}$ po 16 niepowodzeniu uznaje się, że nie ma połączenia
- Losuję liczbę od 0 do x i mnożę ją z $2 * t$, gdzie t to odległość pomiędzy routerami
- **Sposób uruchamiania programu** (działa dla wersji Javy: 13.0.1+9 oraz Mavena: 3.6.2)
 - Do sprawozdania załączony został jar z programem (znajduje się w folderze `zad2\ethernet\target`), by go uruchomić należy wpisać w cmd: `java -jar ethernet-1.0.jar`
 - Załączony został również projekt. Celem kompilacji projektu należy wpisać: `mvn package`
- **Argumenty przyjmowane przez program:**
 - `-h` wyświetla informację o możliwych danych do wprowadzenia
 - `-c int x` , ustawia długość przewodu na x (domyślnie 30)
 - `-r int x` , ustawia ilość routerów na x (domyślnie 2)
 - `-l int x` , ustawia ilość iteracji, które wykona program (domyślnie 100000)
 - `-ifg int x` , ustawia odległość między ramkami na x (domyślnie 2)
 - `-p double x`, ustawia prawdopodobieństwo wysłania pakietu przez router w pojedynczej iteracji
- **Wnioski:**
 - Ilość routerów ma duży wpływ na poprawność działania sieci (im mniej tym lepiej)
 - Gdy jeden z routerów wysyła nieustannie pakiety, reszta nie może nic wysłać - następuje zagarnięcie łącza
 - Im dłuższy przewód, tym dłuższe muszą być wysyłane pakiety
 - CSMA/CD ze względu na swoją prostotę i mały wymóg (jeden przewód) dobrze nadaje się do sieci lokalnej
 - Niestety, ponieważ wiadomości muszą być długości średnicy sieci, sieć nie powinna być bardzo rozbudowana
 - Nie nadaje się do sieci bardzo mocno obciążonej