Ex2 – UDP communication between two hosts and server

מגישים:

ברוך רוזן 208920884

הראל יקיר 99999999

# General points

In this exercise we would simulate two hosts talking between them through a server.

* For this exercise we would be using the UDP protocol in version IPv4.
* We will use C in our work.
* The programing would be in windows OS, so we would use the library “Winsock”.

We would make four files:

1. hosts\_part1\_windows.cpp – A and B for the user choice.
2. hosts\_part3\_ windows.cpp – A and B for the user choice with ECC.
3. mitm\_part1\_ windows.cpp – a simple server for part 1(M).
4. mitm\_part2\_ windows.cpp – server that changes one character in every message.

# General order of operations

## A basic

1. Set a connection with M.
2. Read user input.
3. Send the user message to M including in the data B’s IP.
4. Receive message from M (data from B) and print it
5. Close connection.

## B basic

1. Set connection with M.
2. Receive message from M (data from A) and print it.
3. Read user input.
4. Send the user message to M including in the data A’s IP.
5. Close connection.

## M basic

1. Open socket and listen to A (port 67).
2. Accept data from A and print it (“dataAB”).
3. Send “dataAB” to B.
4. Receive message from B (to A).
5. Send the message from B to A.
6. Close connections

## Changes for parts 2+3

Part 2

* M would swap one char in the messages passing through him.
* A and B act normally.

Part 3

* M would swap one char in the messages passing through him.
* A and B would use the ECC method mentioned later in the document when sending and receiving data.

# Connection

In the exercise we were requested to confirm a connection between the server and host before they start to “talk”. This connection requirement is relevant for TCP and not for UDP. For the connection part we would simply print “connection with …” when the sockets are ready for communication.

# M with a “Man in the Middle” attack

For the second and third parts the server would swap one random char from the message going through it. To make it simple we would choose a random char from the message and swap it with ‘s’ (or ‘r’ if it was ‘s’).

Error correction code

For the error correction code we would use the method of code repetition.

Each letter would be repeated 3 times and so the receiving side can decide the correct data by compering 3 characters at a time. This algorithm would assume that in each set of three characters only one mistake is possible.