* Computer Network and Security (CNS) - Assignment Number - 4

Name :- Kaustubh Shrikant Kabra.
Class :- Third Year Engineering.
Div :- A Roll Number :- 38
Batch :- T-2
Department :- Computer Department
College :- AISSMS's IOIT.

★ Explain error control and flow control in TCP with example.

→ Error control -

1. Error control is meant for the transmission of error free data from sender to receiver.
2. To detect error in data, the approaches are : Checksum, Cyclic Redundancy Check and Parity Checking.
3. To correct error in data, the approaches are : Hamming code, Reed-Solomon code, Low-Density Parity check codes.
4. It is used to detect and correct the error occurred in the code.
5. Example :- Stop and wait ARQ and Sliding Window ARQ.

Flow control -

1. Flow control is meant only for the transmission of data from sender to receiver.
2. For Flow control there are two approaches : Feedback-based Flow control and Rate-based Flow control.

3. It prevents the loss of data and avoid over running of receive buffers.
4. Examples :- Stop and wait Protocol and Sliding Window Protocol.

★ Define socket ? Explain with suitable example the connection between client and server with the help of socket ?

→ Socket -

Sockets allow communication between two different processes on the same or different machines. To be more precise, its a way to talk to other computers using standard Unix file descriptors. In Unix, every I/O action is done by writing or reading a file descriptor.

To a programmer, a socket looks and behaves much like a low-level file descriptor. This is because commands such as read() and write() work with sockets in the same way they do with files and pipes.

The entire process can be broken down into following steps :-

TCP Server -

1. using create(), create TCP socket.
2. using bind(), Bind the socket to server address.
3. using listen(), put the server socket in a passive mode, where it waits for the client to approach the server to make a connection.
4. using accept(), At this point, connection is established between client and server, and they are ready to transfer data.
5. Go back to step 3.

TCP client -

1. Create TCP socket.
2. Connect newly created client socket to server.

Compilation -

Server side:

```
gcc server.c -o server
./server
```

Client side:

```
gcc client.c -o client
./client
```

Output -

Server side:

```
Socket successfully created..
Socket successfully binded..
Server listening..
server accept the client...
From client: hi
        To client: hello
From client: exit
        To client: exit
Server Exit...
```

Client side:

```
Socket successfully created..
connected to the server..
Enter the string : hi
```

From Server : hello
Enter the string : exit
From Server : exit
Client Exit...