**Simulating the Fast Ethernet**

The project simulates a Fast Ethernet network using processes to represent stations (SPs) and a communication switch process (CSP). The simulation was created using socket programming and concurrent event handling. The main components of the system include the Station Process (SP) and Communication Switch Process (CSP). Each station process represents a station in the Ethernet network. It sends frames to other SPs through the CSP where each station has a simulation input file that guides its behavior. A SP sends a request frame to the CSP, asking permission to send a data frame. Once the CSP grants permission, the SP sends a data frame to its destination. When the CSP rejects the request frame, the SP will retry sending it up to 3 times.

The CSP manages data flow between SPs by forwarding data frames. It controls access to the communication medium by either allowing or rejecting request frames. It also holds the request frames from SPs waiting for permission to send data frames and data frames that are ready to be forwarded to the destination SP. The SPs and CSP communicate via TCP sockets. Each SP communicates with the CSP, and the CSP forwards data to the appropriate SPs.

**Key Functionalities of the Station Process include**:

* Reading frames from its simulation file.
* Sending a request frame to the CSP asking for permission to send a data frame.
* If the CSP rejects the request, it retries up to 3 times.
* Once permission is granted, it sends the data frame.
* Waiting for incoming data frames from other SPs.

**Key Functionalities of the Communication Switch Process (CSP) include**:

* Receiving request frames from SPs and replies based on the availability of space in the data frame queue.
* If the queue is full, it either queues the request or rejects it.
* Forwarding data frames to the destination SP after receiving them from a source SP.

Both SP and CSP need to handle multiple events simultaneously (reading, sending, and receiving frames). This can be implemented using select () or threading to handle multiple socket connections concurrently. The system operates based on simulation input files. Each SP has an associated file that specifies which frame to send and when to wait for incoming frames.

**Structure and Flow:**

1. **SP behavior**:
2. Reads a frame from the input file.
3. Sends a request to the CSP.
4. Waits for a response (positive, negative, or retry).
5. If positive, sends the data frame to the CSP for forwarding.
6. Logs each action (sending, receiving frames, retries).
7. **CSP behavior**:
   1. Receives request frames from SPs.
   2. Replies based on the current state of the queues (data frame and request frame).
   3. Forwards data frames to the correct destination SP.
   4. Logs all interactions with SPs such as requesting reception, forwarding, and replies.