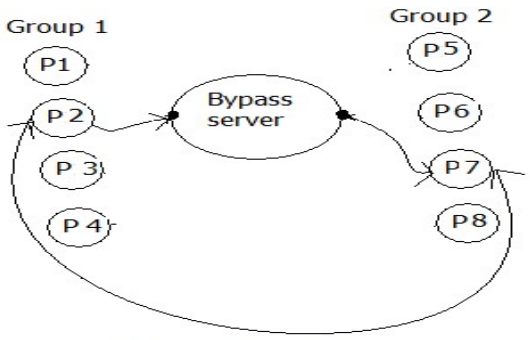
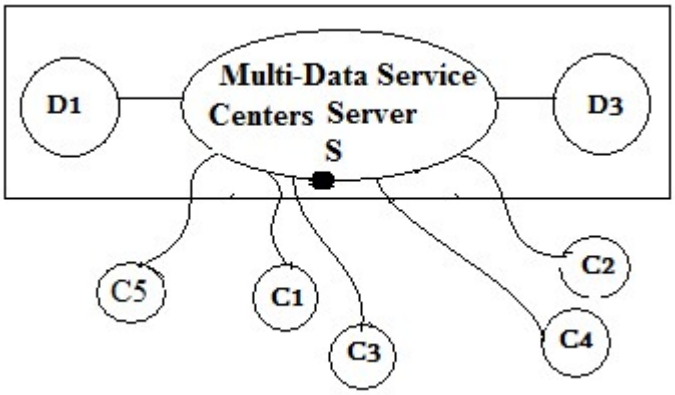
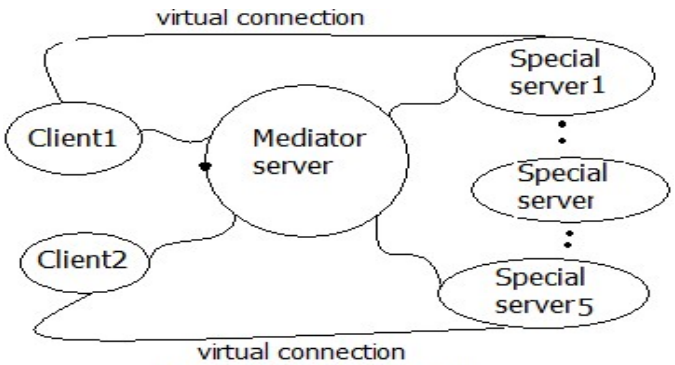
1.	**Chat server with poll or select()**
2.	**inetd with select() and threads S1 with process, S2 with thread, S3 connection less**
3.	**Bypass Server:** Let processes P1, P2, P3 and P4 are in Group1 and processes P5, P6, P7 and P8 are in Group2. The Group1 processes are willing to communicate with Group2 processes, but they cannot contact on their own. That's why, a process in Group1 sends its intention of contacting a particular process in Group2 to Bypass server. Then the Bypass server conveys (passes) that intention to the corresponding process in Group2, so that both the processes can further proceed and will be communicating in a connection-oriented way. Assume that Bypass server knows the address details of all the Group 2 processes and all processes are in different computer systems. The Bypass server will be listening to a single port for the requests from the Group1 processes. For example, if process P2 likes to communicate with process P7, it informs the same to the Bypass server. The Bypass sever conveys this to process P7. Now a connection-oriented communication will be established by processes P2 and P7 and they will be communicating directly as shown in the figure below. Implement Bypass Server, a process in Gropu1 and a process in Group2 .  Figure: Bypass Sever

4.	**Multi–Data Service centers Server:** A Multi–Data Service centers Server S provides four Data service servers with executable files as D1.exe, D2.exe, D3.exe and D4.exe. All the Clients first connect to the well-known point of Multi–Data Service centers Server S. The Clients also inform server S, about the data service number they would like to use as numbers 1, 2, 3, and 4. Depending on the data service number request from Client, server S arranges a separate corresponding Data service sever D_i (if it is not existing), and the Client receives the data of standard output of data server D_i as shown in figure below. If such D_i is already existing, then sever S will not create a new D_i , but it arranges that the Client receives the data of standard output of data server D_i it has requested. Suppose at the moment three Clients C1, C3, C5 requested for D1 service and C2, C4 requested for D3 service, one D1 process and one D3 process along with main sever process S will be existing in the scenario as shown in figure below. In other words, only one data service server process will exist, regardless of the number of client requests for it. Implement Multi–Data Service centers Server S, D1, D2, D3, D4 data service servers and Client. 
5.	**Mediator:** Assume that Clients are not allowed to communicate directly with Special servers. The Clients request the Mediator for such facility by providing the Special server number. The Mediator connects to the corresponding Special server on behalf of the client and mediates the communication as Client←→Mediator ←→Special server. The Mediator handles all connections of the Special server with a single separate thread i.e. five threads for five Special servers. Assume that Mediator knows the address details of all Special server processes and all processes are in different computer systems (see figure). Implement **Mediator, Special Server** and a **Client process** using BSD sockets  Figure: Mediator Server