Birla Institute of Technology & Science, Pilani
Pilani Campus

II SEMESTER 2021-2022
Assignment-2

Course No.:	**IS F462**	**Course Title:**	**Network Programming**
Deadline:	**As on Canvas**	**Maximum Marks:**	**60M (15%)**

Note:

- Maximum of two students per group. Upload code in Canvas.
- Name your file idno1_idno2_assignment1.tar .
- Any three problems out of four (in both assignments) to be submitted

P1. A file contains N (>1000) URLs of webpages. A programmer wants to find out the longest path common to all the given webpages. A path consists of intermediate router interface addresses through which the packet travels. Using I/O multiplexing, conceive and implement a program which finds out path for each of the URLs and the longest path common to them.
Deliverables:

- pathfinder.c
- PDF file explaining design decisions and documentation

[30M]

P2. Consider the following paragraph given in section 16.5 of the textbook.

> We provide this example using simultaneous connects because it is a nice example using nonblocking I/O and one whose performance impact can be measured. It is also a feature used by a popular Web application, the Netscape browser. There are pitfalls in this technique if there is any congestion in the network. Chapter 21 of TCPv1 describes TCP's slow-start and congestion avoidance algorithms in detail. When multiple connections are established from a client to a server, there is no communication between the connections at the TCP layer. That is, if one connection encounters a packet loss, the other connections to the same server are not notified, and it is highly probable that the other connections will soon encounter packet loss unless they slow down. These additional connections are sending more packets into an already congested network. This technique also increases the load at any given time on the server.

Consider the underlined lines. It tells about the missing coordination among the TCP connections or clients accessing the same web server. Design a solution that can enable this coordination considering the fact that TCP doesn't tell the application immediately about the packet loss. Generally TCP tries for a few times before it concludes about packet loss. You design should be able to detect packet losses as soon as the host comes to know about it and notify the client who are accessing that server. Your program should take care of accepting requests from clients, detecting packet losses, and notifying the clients. Implement the solution designed.

Deliverables:

- tcpclient.c, error_detector.c
- PDF file explaining design decisions and documentation

[30M]

--&--