HOMEWORK 1

Name: Raj Shah

NetID & RUID: - ras637 & 221003377

CS352: Internet Technology

Section: 04

Instructor: Desheng Zhang

Problem 1: Consider the following circuit-switch network in Fig 1, and four switches A, B, C and D, going in the clockwise direction, and 4 circuits on each link. (i) In this network, calculate the maximum number of connections that can be in progress simultaneously at any one time? (ii) If all connections are between switches A and C, what is the maximum number of connections that can be in progress simultaneously? (iii) If we want to have 4 connections between A and C, and another 4 connections between B and D. Can we route these calls through the 4 links to accommodate all 8 connections?

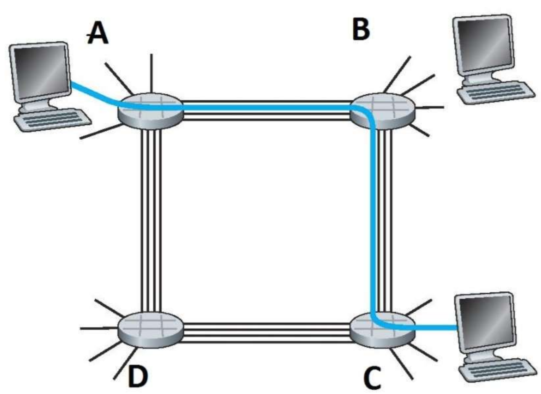

Fig 1

Solution:

(i) In this network, calculate the maximum number of connections that can be in progress simultaneously at any one time?

a. Since there are 4 links (A-B, B-C, C-D, D-A) and each link can handle 4 circuits, the total number of circuits available in the network is

$$4 \text{ Links} \times 4 \text{ circuits per link} = 16 \text{ circuits}$$

b. Each connection occupies a circuit on each link along its path.

c. The maximum number of simultaneous connections occurs when the total circuits (16) are fully utilized.

Thus, 16 circuits can be active at the same time.

(ii) If all connections are between switches A and C, what is the maximum number of connections that can be in progress simultaneously?

a. To establish a connection from A to C, there are two possible paths:
   i. $A \rightarrow B \rightarrow C$ (2 links)
   ii. $A \rightarrow D \rightarrow C$ (2 links)
b. Each connection from A to C will consume 2 circuits (one for each link it traverses).
c. Since each link has 4 circuits, the bottleneck occurs at the links used in the path.

Each of the two paths (A-B-C and A-D-C) can each support 4 connections (since A-B and A-D each have 4 circuits available).

Thus, the maximum number of A-C connections is:

$4(A \rightarrow B \rightarrow C) + 4(A \rightarrow D \rightarrow C) = 8$ connections

(iii) If we want to have 4 connections between A and C, and another 4 connections between B and D. Can we route these calls through the 4 links to accommodate all 8 connections?

a. We need to establish:
   i. 4 connections between A and C
   ii. 4 connections between B and D
b. Possible paths:
   i. A to C can use A-B-C (4 circuits) or A-D-C (4 circuits).
   ii. B to D can use B-C-D (4 circuits) or B-A-D (4 circuits).
c. Since each link has 4 circuits, we can distribute the connections as follows:
   i. Route 4 A-C connections via A-B-C (2 links, 4 circuits used).
   ii. Route 4 B-D connections via B-C-D (2 links, 4 circuits used).

Since these connections do not exceed the 4 circuits per link limit, all 8 connections can be accommodated.

Final Answers:

(i) Maximum simultaneous connections in the network: 16
(ii) Maximum simultaneous A-C connections: 8
(iii) Can we route 4 A-C and 4 B-D connections? Yes

Problem 2: Consider the throughput example corresponding to Fig.2. If we have are M client-server pairs instead of 10. Denote Rs, Rc, and R for the rates of the server links, client links, and network link. If all other links have enough capacity and that there is no other traffic in the network besides the traffic generated by the M client-server pairs. Please derive a general expression for throughput in terms of Rs, Rc , R , and M .

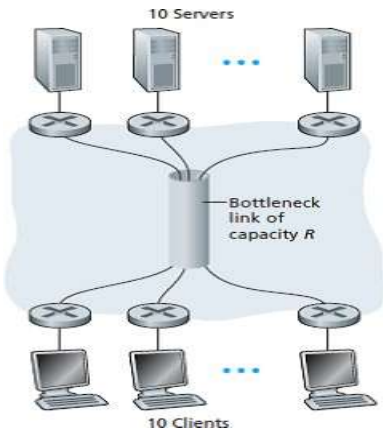

Fig 2

Solution:

Understanding the Network Constraints

- R_s = Rate of server links (each server's outgoing rate).
- R_s = Rate of client links (each client's receiving rate).
- R = Rate of the bottleneck network link (shared across all client-server pairs).
- M = Number of client-server pairs.

Thus, the per-connection throughput is limited by:

$$T = min(\mathrm{Rs}, \mathrm{Rc}, \frac{R}{M})$$

Total Network Throughput

The total throughput of the system (sum of all M connections) is:

$$Ttotal = min(\mathrm{Rs}, \mathrm{Rc}, \frac{R}{M})$$

Final Expression for Throughput

Thus, the total network throughput is:

$$Ttotal = min(\mathrm{Rs}, \mathrm{Rc}, \frac{R}{M})$$

Server capacity (R_s).

Client capacity (R_c).

Network capacity (R) is shared among M pairs.

Problem 3: Assume an HTTP client that wants to obtain a web document at a given URL. The IP address of the HTTP server is unknown at first. The web document at the URL has one embedded GIF image that resides at the same server as the original document. What transport and application layer protocols besides HTTP are needed in this scenario?

Solution:

In this scenario, the HTTP client needs to retrieve a web document from an unknown IP address and then fetch an embedded GIF image from the same server. Several transport and application layer protocols are required to accomplish this task.

1. **Domain Name System (DNS) - (Application Layer)**
   a. Since the IP address of the HTTP server is unknown at first, the client must resolve the domain name (e.g., www.example.com) to an IP address.
   b. This is done using the DNS protocol, which operates at the application layer.
   c. The client sends a DNS query to a DNS server to get the corresponding IP address.
2. **Transmission Control Protocol (TCP) - (Transport Layer)**
   a. HTTP relies on TCP for reliable, connection-oriented communication.
   b. The client initiates a TCP three-way handshake with the web server on port 80 (HTTP) or port 443 (HTTPS) before sending an HTTP request.
   c. TCP ensures that the HTTP requests and responses are delivered in order and without errors.
3. **HyperText Transfer Protocol (HTTP) - (Application Layer)**
   a. The client sends an HTTP GET request to retrieve the web document.
   b. The server responds with the requested document.
   c. The client then parses the document and finds the embedded GIF image.
4. **Additional HTTP Request for Embedded GIF (Application Layer)**
   a. Since the GIF image resides on the same server, the client makes another HTTP
   b. GET request for the image.
   c. The image is then retrieved and rendered by the browser.

Summary of Required Protocols:

Protocol	Layer	Purpose
DNS	Application	Resolves domain name to an IP address
TCP	Transport	Establishes a reliable connection for HTTP

HTTP	Application	Fetches the web document and the embedded GIF

Thus, besides HTTP, the client also needs DNS (for name resolution) and TCP (for reliable data transfer).

Problem 4: Suppose you click on a link to obtain a Web page in your Web browser. The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary to obtain this IP address. Assume that DNS servers are visited before your host receives the IP address from DNS; the successive visits incur an RTT of RTT1,...,RTTn. Further assume that the Web page associated with the link contains exactly one object, consisting of a small amount of HTML text. Let RTT0 denote the RTT between the local host and the server containing this object. Assuming zero transmission time of the object, how much time it will take from when the client clicks on the link until the client receives the object?

Solution:

1. **DNS Lookup**
    - **The client does not have the IP address cached, so it must query n successive DNS servers.**
    - **Each DNS lookup takes an RTT (Round-Trip Time), denoted as RTT_1, RTT_2,..., RTT_n.**
    - **The total DNS resolution time is:**

$$\mathbf{RTT_{DNS} = RTT_1, RTT_2, \ldots, RTT_n.}$$

2. **Establishing a TCP Connection**
    - **Once the IP address is resolved, the client initiates an HTTP request, but first, a TCP connection must be established with the server.**
    - **The TCP connection setup (via the three-way handshake) requires 1 RTT between the client and server, denoted as RTT_0.**
3. **Sending the HTTP Request and Receiving the Object**
    - **After the TCP connection is established:**
        1. **The client sends an HTTP request (typically a GET request).**
        2. **The server processes the request and sends back the HTML object.**
        3. **This requires one additional RTT(RTT_0).**

Total Time Calculation

Adding up all the required steps:

$$T_{Total} = (RTT_1, RTT_2, \ldots, RTT_n) + RTT_0 + RTT_0$$

$$T_{Total} = (RTT_1, RTT_{2,\ldots}, RTT_n) + 2\ RTT_0$$

Final Answer

$$T_{Total} = \sum_{i=1}^{n} \text{RTTi} + 2\ \text{RTTo}$$

This formula gives the total time from clicking the link to receiving the web object, assuming zero transmission time.

Problem 5: Referring to the above problem, suppose this HTML file references 8 very small objects on the same server. Neglecting transmission times, how much time elapses with (i) Non-persistent HTTP with no parallel TCP connections? (ii) Non-persistent HTTP with the browser configured for 5 parallel connections? (iii) Persistent HTTP?

Solution:

1. **Non-Persistent HTTP (No Parallel TCP Connections)**
   a. **Each object requires a new TCP connection (2 RTTs per object).**
   b. **Each connection requires 2 RTTs (1 for handshake, 1 for request/response).**
   c. **Downloading 8 objects sequentially:**

$$8 \text{ x } 2\ RTT_0 = 16\ RTT_0$$

Total Time

$$\sum_{i=1}^{n} \text{RTTi} + 18\ \text{RTTo}$$

2. **Non-Persistent HTTP (5 Parallel Connections)**
   a. **The browser initiates 5 downloads simultaneously (taking 2 RTTs).**
   b. **The remaining 3 objects download afterward (another 2 RTTs).**
   c. **Total Time**

$$\sum_{i=1}^{n} \text{RTTi} + 6\ \text{RTTo}$$

3. **Persistent HTTP**
   a. **A single TCP connection is maintained.**
   b. **Each object retrieval takes 1 RTT instead of 2.**
   c. **Total for 8 objects:** 8 RTTo

Total Time:

Without Pipeline

$$\sum_{i=1}^{n} \mathrm{RTTi} + 10\ \mathrm{RTTo}$$

With Pipeline

$$\sum_{i=1}^{n} \mathrm{RTTi} + 3\ \mathrm{RTTo}$$