Tutorial 1 | CN-CS331 | 03 Feb 2025

Deadline: 7th Feb 2025 with 2 bonus points

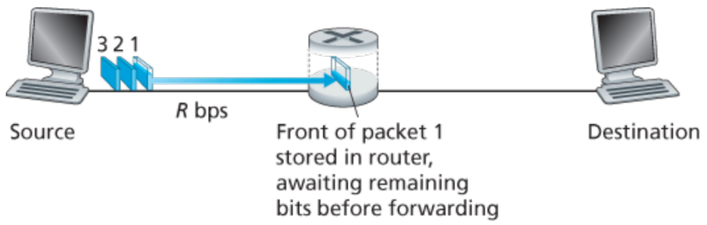
Problem 1: Consider two hosts, A and B, connected by a single link of rate R bps. Suppose the two hosts are separated by m meters, and suppose the propagation speed along the link is s meters/sec. Host A is to send a packet of size L bits to Host B.

a) Ignoring processing and queuing delays, obtain an expression for the end-to-end delay.
b) Suppose Host A begins to transmit the packet at time t = 0. At time t = d_{trans}, where is the last bit of the packet?
c) Suppose d_{prop} is greater than d_{trans}. At time t = d_{trans}, where is the first bit of the packet?
d) Suppose d_{prop} is less than d_{trans}. At time t = d_{trans}, where is the first bit of the packet?
e) Suppose s = $2.5 * 10^8 m/s$, L = 120 bits, and R = 56 kbps. Find the distance m so that d_{prop} equals d_{trans}.

Problem 2: Consider a packet of length L that begins at end system A and travels over three links to a destination end system. These three links are connected by two packet switches. Let di, si, and Ri denote the length, propagation speed, and the transmission rate of link i, for i =1,2,3. The packet switch delays each packet by d(proc). Assuming no queuing delays, in terms of di, si, Ri, (i=1,2,3) i, and L, what is the total end-to-end delay for the packet? Suppose now the packet is 1,500 bytes, the propagation speed on all three links is $2.5 * 10^8 m/s$, the transmission rates of all three links are 2 Mbps, the packet switch processing delay is 3 msec, the length of the first link is 6,000 km, the length of the second link is 3,000 km, and the length of the last link is 1,000 km. For these values, what is the end-to-end delay?

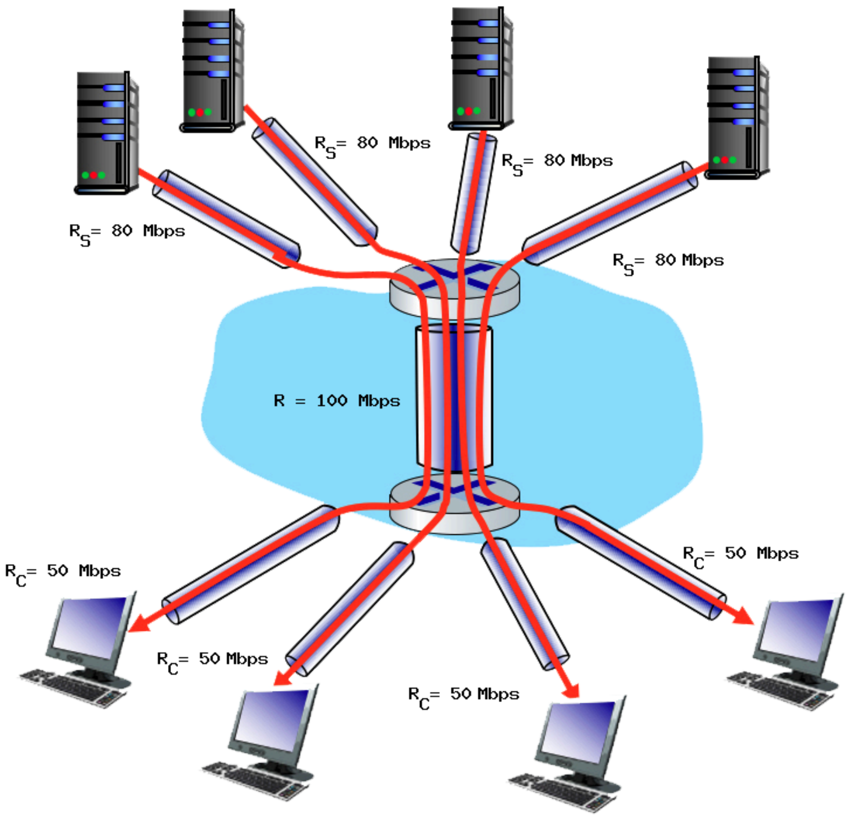
Problem 3: In the above problem, suppose R1=R2=R3=R and dproc=0. Further, suppose the packet switch does not store and forward packets but instead immediately transmits each bit it receives before waiting for the entire packet to arrive. What is the end-to-end delay?

Problem 4: Derive a formula for computing end-end packet delay of sending one packet of length 'L' over 'N' links of transmission Rate 'R'. Assume no propagation delay.
Extend the formula for sending 'P' packets back to back over 'N' links.


3 2 1
R bps
Source
Front of packet 1 stored in router, awaiting remaining bits before forwarding
Destination

Problem 5: Consider a router buffer preceding an outbound link. In this problem, you will use Little's formula, a famous formula from queuing theory. Let N denote the average number of packets in the buffer plus the packet being transmitted. Let a denote the rate of packets arriving at the link. Let d denote the average total delay (i.e., the queuing delay plus the transmission delay) experienced by a packet. Little's formula is N = a · d. Suppose that on average, the buffer contains 10 packets, and the average packet queuing delay is 10 msec. The link's transmission rate is 100 packets/sec. Using Little's formula, what is the average packet arrival rate, assuming there is no packet loss?

Problem 6: Consider the scenario shown below, with four different servers connected to four different clients over four three-hop paths. The four pairs share a common middle hop with a transmission capacity of R = 100 Mbps. The four links from the servers to the shared link have a transmission capacity of RS = 80 Mbps. Each of the four links from the shared middle link to a client has a transmission capacity of RC = 50 Mbps.



a) What is the maximum achievable end-end throughput (in Mbps) for each of the four client-to-server pairs, assuming that the middle link is fairly shared (divides its transmission rate equally)?
b) Which link is the bottleneck link? Format as Rc, Rs, or R.
c) Assuming that the servers are sending at the maximum rate possible, what are the link utilizations for the server links (RS)? Answer as a decimal.

d) Assuming that the servers are sending at the maximum rate possible, what are the link utilizations for the client links (RC)? Answer as a decimal.
e) Assuming that the servers are sending at the maximum rate possible, what is the link utilizations for the shared link (R)? Answer as a decimal.

Problem 7:
A. Suppose N packets arrive simultaneously at a link at which no packets are currently being transmitted or queued. Each packet is of length L and the link has transmission rate R. What is the average queuing delay for the N packets?
B. Now suppose that N such packets arrive at the link every LN/R seconds, What is the average queuing delay of a packet?

Problem 8: Consider sending a large file of F bits from Host A to Host B. There are three links (and two switches) between A and B, and the links are uncongested (that is, no queuing delays). Host A segments the file into segments of S bits each and adds 80 bits of header to each segment, forming packets of L= 80 + S bits. Each link has a transmission rate of R bps. Find the value of S that minimizes the delay of moving the file from Host A to Host B. Disregard propagation delay.

Problem 9: Consider the resolution of the domain name www.iitgn.ac.in by a resolver. Assume that no resource records are cached anywhere across the servers and that an iterative query mechanism is used in the resolution. The number of query-response pairs involved in completely resolving the domain name is ________________.

Problem 10: A sender's email goes through 3 SMTP servers before reaching the recipient. Each SMTP hop introduces a 30 ms processing delay, and the email travels over 4 links, each with a 20 ms propagation delay. What is the total delay?

Problem 11: A web page consists of:

1 HTML file (5 KB),
3 CSS files (2 KB each),
5 images (50 KB each).

The RTT between the client and server is 50 ms. The bandwidth is 10 Mbps. Compare the total page load time using:

a) HTTP/1.1 without persistent connections.
b) HTTP/2 with multiplexing (assume all objects can be sent in parallel over a single connection).

Problem 12: A local DNS resolver has a single-level caching system: Cache hit rate: 80%. Cache lookup time: 1 ms. If the cache misses, the resolver queries the following servers sequentially:

1. Root server: RTT = 50 ms.
2. TLD server: RTT = 70 ms.
3. Authoritative server: RTT = 100 ms.

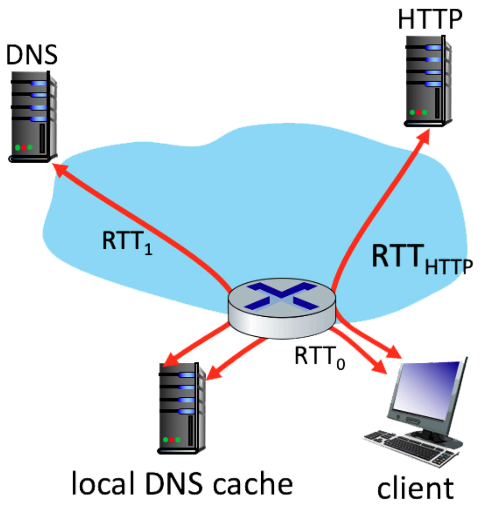
Additionally:

a) 10% of all DNS queries are invalid (e.g., non-existent domains). These queries bypass the cache and directly return an error after 10 ms.
b) The remaining 90% of queries are valid and go through the caching or external resolution process.

Calculate:

1. The average DNS resolution time .
2. How much faster (in percentage) would DNS resolution be if the cache hit rate increased to 95% , while keeping other parameters constant?

Problem 13: Suppose within your Web browser you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary to obtain the IP address. Suppose that two DNS servers are visited before your host receives the IP address from DNS. The first DNS server visited is the local DNS cache, with an RTT delay of RTT0 = 1 msec. The second DNS server contacted has an RTT of 47 msecs. Initially, let's suppose that the Web page associated with the link contains exactly one object, consisting of a small amount of HTML text. Suppose the RTT between the local host and the Web server containing the object is RTTHTTP = 72 msecs.



I. Assuming zero transmission time for the HTML object, how much time (in msec) elapses from when the client clicks on the link until the client receives the object?

II. Now suppose the HTML object references 7 very small objects on the same server. Neglecting transmission times, how much time (in msec) elapses from when the client clicks on the link until the base object and all 7 additional objects are received from web server at the client, assuming non-persistent HTTP and no parallel TCP connections?

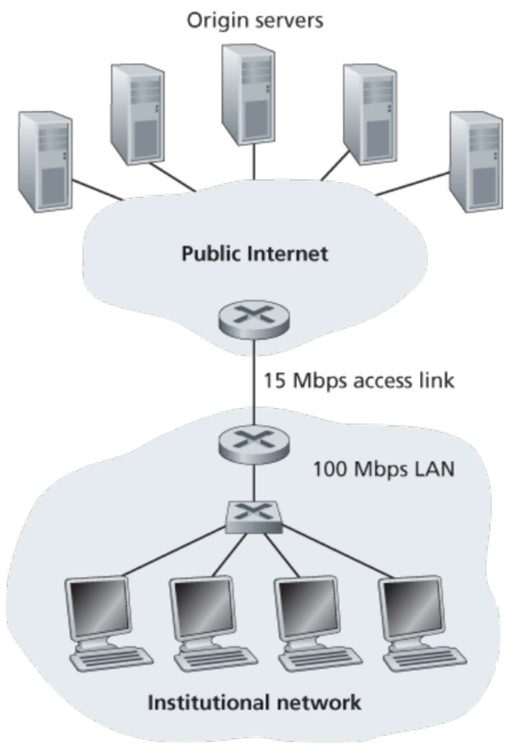
III. Suppose the HTML object references 7 very small objects on the same server, but assume that the client is configured to support a maximum of 5 parallel TCP connections, with non-persistent HTTP.

IV. Suppose the HTML object references 7 very small objects on the same server, but assume that the client is configured to support a maximum of 5 parallel TCP connections, with persistent HTTP.

V. What's the fastest method we've explored: Nonpersistent-serial, Nonpersistent-parallel, or Persistent-parallel?

Problem 14: Consider the figure given below, for which there is an institutional network connected to the Internet. Suppose that the average object size is 850,000 bits and that the average request rate from the institution's browsers to the origin servers is 16 requests per second. Also, suppose that the amount of time it takes from when the router on the Internet side of the access link forwards an HTTP request until it receives the response is three seconds on average. Model the total average response time as the sum of the average access delay (that is, the delay from the Internet router to the institution router) and the average Internet delay. For the average access delay, use $\Delta/(1-\Delta\beta)$, where Δ is the average time required to send an object over the access link and β is the arrival rate of objects to the access link.

a) Find the total average response time.
b) Now suppose a cache is installed in the institutional LAN. Suppose the miss rate is 0.4. Find the total response time.



Problem 15: Suppose within your Web browser you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary to obtain the IP address. Suppose that n DNS servers are visited before your host receives the IP address from DNS; the successive visits incur an RTT of RTT1,... .,RTTn. Further suppose that the Web page associated with the link contains exactly one object, consisting of a small amount of HTML text. Let RTT denote the RTT between the local host and the server containing the object. Assuming zero transmission time of the object, how much time elapses from when the client clicks on the link until the client receives the object?

Problems for more practice

Problem 1: Client-Server Task Processing Program Using Socket Programming

This program consists of a **client** and three different **server designs** that process tasks sent by the client.

Client Program

- The client presents a **menu** with four options:
    1. **Change the case of a string** (convert uppercase to lowercase and vice versa).
    2. **Evaluate a mathematical expression** (e.g., "2+3*5/2").
    3. **Find the reverse of a given string.**
    4. **Exit the program**.

- The client sends the selected task to the server and waits for the response.
- The menu repeats until the client chooses to exit or presses Ctrl+C.

Server Implementations

The server will be implemented in **three different ways**:

1) **Single-Process Server**
- The server can only handle **one client at a time**.
- If a client is connected, other clients must **wait** until the current client disconnects.

2) **Multi-Process Server**
- The server **creates a new process** for each connected client.
- Multiple clients can be handled **at the same time**, each running in a separate process.

3) **Multi-Threaded Server**
- The server **creates a new thread** for each connected client.
- Multiple clients can be handled **simultaneously** but with shared memory resources.

Problem 2:- Client-Server Benchmarking Task

This task involves implementing a **client-server architecture** where the client sends a string to the server, and the server responds with the **reversed string after a 3-second delay**. The server will be implemented in **three different ways**, and we will **benchmark its performance** by running multiple concurrent clients and measuring execution time.

Client Program

- The client connects to the server, **sends a string**, and waits for a response.
- The client receives the **reversed string** and then **closes the connection**.

Server Implementations

The server will be implemented in **three different ways** to handle client connections:

1) **Single-Process Server**
- Handles **only one client at a time**.
- If a client is connected, other clients **must wait** until the current client disconnects.

2) **Multi-Process Server**
- The server **creates a new process** for each client.
- Multiple clients can be **handled simultaneously**, each running in a separate process.

3) **Multi-Threaded Server**

- The server **creates a new thread** for each client.
- Multiple clients can be handled **at the same time**, with **shared resources**.

Benchmarking Task

- A Bash script (start.sh, Link:- Link) will be used to execute multiple client instances concurrently.
- The script will be run with the following command:

  bash start.sh <client_program_file> <number_of_clients>

  Example:- bash start.sh client.py 5

  This command runs client.py 5 times in parallel and measures the total execution time.

Performance Evaluation & Graph Plotting

- You must **test each server design** (Single-Process, Multi-Process, Multi-Threaded) with varying numbers of concurrent clients, ranging from **10 to 100**.
- You have to **record the execution time (latency)** for each case.
- Finally, you have to **plot a graph** with:
  - **X-axis**: Number of concurrent clients (10, 20, ..., 100).
  - **Y-axis**: Execution time (latency).