Assignment1

1

1. One-way Propagation Delay: The propagation delay d is calculated by the formula:

$$d = \frac{Distance}{Speed\ of\ Light}$$

2. Transmission Time: The time t it takes to transmit the data at a given rate is given by:

$$t = \frac{Size\ of\ Data}{Transmission\ Rate}$$

3. Total Time for data to be received on Earth:

$$Total\ Time = Propagation\ Delay + Transmission\ Time$$

(a) One-way Propagation Delay

- Distance between Earth and Mars at closest approach = 62.07×10^9 meters (62.07 Gm)
- Speed of light = 3×10^8 meters/second

$$Propagation\ Delay = \frac{62.07 \times 10^9}{3 \times 10^8}$$

$$Propagation\ Delay \approx 206.9\ seconds$$

(b) Time of Picture Receipt on Earth

- Image size = 4Mb = 4×10^6 bits
- Transmission rate = 256kbps = 256×10^3 bits/second

$$Transmission\ Time = \frac{4 \times 10^6}{256 \times 10^3}$$

$$Transmission\ Time = 15.625\ seconds$$

Total Time for the picture to be received on Earth would be the sum of the propagation delay and the transmission time:

$$Total\ Time = 206.9 + 15.625$$

$$Total\ Time = 222.525\ seconds$$

So, at the time of closest approach, the image from the rover would be received on Earth in approximately 222.525 seconds after being sent.

2

1. Queuing Delay: Queuing delay is the time a packet waits in the queue before it can be transmitted.
2. Transmission Time: t is given by:

$$t = \frac{\text{Size of Data}}{\text{Transmission Rate}}$$

3. Maximum Queuing Delay: This is the delay experienced by the last packet in the queue.
4. Minimum Queuing Delay: This is the delay experienced by the first packet in the queue. If the outgoing link is free, this is typically zero.
5. Average Queuing Delay: This is the average of the queuing delays experienced by all the packets.

Given

- Size of each packet = 3200 bytes = 3200×8 bits = 25600 bits
- Number of packets = 10
- Transmission rate = 20 Mbps = 20×10^6 bits/second

(a) Maximum Queuing Delay

To find the maximum queuing delay, we must find the time it takes to transmit all packets before the last one.

Transmission time for one packet:

$$t = \frac{25600}{20 \times 10^6} = \frac{25600}{2 \times 10^7} = \frac{128}{10^6} = 0.128 \text{ seconds} = 128 \text{ ms}$$

Maximum Queuing Delay for the last packet:

$$\text{Max Queuing Delay} = (10 - 1) \times 128 \text{ ms} = 9 \times 128 \text{ ms} = 1152 \text{ ms}$$

(b) Minimum Queuing Delay

The minimum queuing delay would be for the first packet. If the outgoing link is free (which is given), the minimum queuing delay is zero.

$$\text{Min Queuing Delay} = 0 \text{ ms}$$

c Average Queuing Delay

The average queuing delay is the average of the delays experienced by all the packets. It's the sum of the queuing delays divided by the number of packets.

$$\text{Average Queuing Delay} = \frac{0 + 128 + 256 + \ldots + 1152}{10}$$

$$\text{Average Queuing Delay} = \frac{(9 \times 10/2) \times 128}{10} = \frac{9 \times 128}{2} = \frac{1152}{2} = 576 \text{ ms}$$

(d) Queue Limitations

If the queue can only hold 6 packets, then the maximum queuing delay would be for the last packet in the limited queue.

$$\text{Max Queuing Delay with Limited Queue} = (6 - 1) \times 128 \text{ ms} = 5 \times 128 \text{ ms} = 640 \text{ ms}$$

For the other 4 packets that cannot be held in the queue, they would typically be dropped or sent to a different queue, depending on the router's packet dropping or buffering policy.

3

(a) Throughput with Original Rates

The throughput will be determined by the link with the lowest rate (the bottleneck). The rates given are:

- $R1 = 500$ kbps
- $R2 = 2$ Mbps
- $R3 = 1$ Mbps

The bottleneck is $R1 = 500$ kbps. So the throughput for the file transfer is 500 kbps.

(b) Time to Transfer File with Original Rates

The file size is 4 million bytes. First, let's convert this size into bits:

$$\text{File Size} = 4 \times 10^6 \text{ bytes} \times 8 \text{ bits/byte} = 32 \times 10^6 \text{ bits} = 32 \text{ Mbits}$$

The time t to transfer this file at 500 kbps is:

$$t = \frac{32 \text{ Mbits}}{500 \text{ kbps}} = \frac{32 \times 10^6 \text{ bits}}{500 \times 10^3 \text{ bits/sec}} = \frac{32 \times 10^6}{500 \times 10^3} \text{ sec} = 64 \text{ sec}$$

c Throughput with Reduced $R2$

Now $R2$ is reduced to 100 kbps, which becomes the new bottleneck. So the throughput for the file transfer is now 100 kbps. The time t to transfer this file at 100 kbps is:

$$t = \frac{32 \text{ Mbits}}{100 \text{ kbps}} = \frac{32 \times 10^6 \text{ bits}}{100 \times 10^3 \text{ bits/sec}} = \frac{32 \times 10^6}{100 \times 10^3} \text{ sec} = 320 \text{ sec}$$

So with $R2$ reduced to 100 kbps:

1. The throughput becomes 100 kbps.
2. The time to transfer the 4 million byte file becomes 320 seconds.

4

Given:

- Base HTML file size: 5 KBytes = $5 \times 8 \times 1024$ bits = $40,960$ bits
- Each referenced object: 200 KBytes = $200 \times 8 \times 1024$ bits =$1,638,400$ bits
- Each control message: 200 bits
- Transmission rate: 10 Mbps = 10×10^6 bits/sec
- One-way propagation delay: 50 ms = 0.05 sec
- Number of referenced objects: 8

Definitions and Formulas:

1. Round-trip time (RTT) = $2 \times \text{Propagation delay}$
2. Transmission time = $\frac{\text{File Size}}{\text{Transmission Rate}}$

3. Total time for each method will depend on various factors such as connection establishment, transmission times, and RTT.

Common Calculations:

1. RTT = 2×0.05 sec = 0.1sec
2. Transmission time for the base HTML file = $\frac{40{,}960}{10 \times 10^6}$sec = 0.004096 sec
3. Transmission time for one referenced object = $\frac{1{,}638{,}400}{10 \times 10^6}$ sec = 0.16384 sec

(a) Using basic non-persistent HTTP with no parallel connections

For each object (including the base HTML file), we need to establish a new connection, send a request, and then receive the object. This process involves:

- 1 RTT for the TCP handshake
- 1 RTT for the HTTP request and response

For the base HTML file:

$$1 \text{ RTT (handshake)} + 1 \text{ RTT (HTTP)} + 0.004096 \text{ sec (transmission)} = 0.1 + 0.1 + 0.004096 = 0.204096 \text{ sec}$$

For each referenced object (8 in total):

$$1 \text{ RTT (handshake)} + 1 \text{ RTT (HTTP)} + 0.16384 \text{ sec (transmission)} = 0.1 + 0.1 + 0.16384 = 0.36384 \text{ sec}$$

Total time:

$$0.204096 \text{ sec (base)} + 8 \times 0.36384 \text{ sec (each object)} = 0.204096 + 2.91072 = 3.114816 \text{ sec}$$

(b) Using non-persistent HTTP with parallel connections

The bandwidth is shared, so each of the 8 connections will receive $1/8$ of the 10 Mbps bandwidth, or 1.25 Mbps.

New transmission time for one referenced object = $\frac{1{,}638{,}400}{1.25 \times 10^6}$sec = 1.31072 sec

Total time:

$$0.204096 \text{ sec (base)} + 0.1 \text{ sec (handshake for first object)} + 0.1 \text{ sec (HTTP request for first object)} + 1.31072 \text{ sec (transmission for one object)} = 0.20$$

c Using persistent HTTP (non-pipelined, no parallel connections)

In persistent HTTP, one connection is established and kept open.

Total time:

$$0.204096 \text{ sec (base)} + 8 \times (1 \text{ RTT (HTTP) } + 0.16384 \text{ sec (transmission)}) = 0.204096 + 8 \times (0.1 + 0.16384) = 0.204096 + 8 \times 0.26384 = 0.204096$$

In summary:

- For basic non-persistent HTTP with no parallel connections, it will take 3.114816 sec
- For non-persistent HTTP with parallel connections, it will take 1.714816 sec
- For persistent HTTP (non-pipelined, no parallel connections), it will take 2.314816 sec

5

When you enter a URL in your web browser and the IP address associated with that URL is not cached on your local host, a DNS (Domain Name System) lookup will be initiated to resolve the URL into an IP address. Assuming that no caching has occurred, the following DNS servers will typically be involved:

1. **Local DNS Server (Recursive Resolver)**: Your browser will first query the local DNS server. If it doesn't have the information, it will start a series of queries to find the correct IP address. This is usually the DNS server of your Internet Service Provider (ISP), or it could be a public DNS server like Google's 8.8.8.8.
2. **Root DNS Server**: The local DNS server queries the root DNS server if it doesn't have the IP address in its cache. The root server returns a reference to a TLD (Top-Level Domain) DNS server.
3. **TLD DNS Server**: The local DNS server then queries the TLD DNS server for the domain. For example, for `www.example.com`, this would be the `.com` TLD DNS server. The TLD server returns a reference to an Authoritative DNS server that has the actual IP address.
4. **Authoritative DNS Server**: Finally, the local DNS server queries the authoritative DNS server for the actual IP address of the domain. Once it gets the IP address, it returns it to your browser.

Minimum Number of DNS Queries

1. One query to the local DNS server, which might involve:
   1. One query to the Root DNS server.
   2. One query to the TLD DNS server.
   3. One query to the Authoritative DNS server for the domain.

So, the minimum number of DNS queries would be four if we consider the chain of queries that the local DNS server has to make on behalf of your initial query.

Types of Servers

- Local DNS Server (Recursive Resolver)
- Root DNS Server
- TLD (Top-Level Domain) DNS Server
- Authoritative DNS Server

In summary, a minimum of four DNS queries will be issued, and they will be sent to these types of servers. Note that this is a simplified explanation and actual DNS resolution may involve more complex scenarios, but this should give you a good basic understanding.

6

(a) HTTP Version

HTTP1.1

(b) Client ip address

10.6.15.132

c Server ip address

128.59.105.24

(d) Parallel Connections

My browser do open multiple parallel connections, as many TCP handshakes happening around the same time.

(e) Total Number Of Connections

(f) Persistent Connections

Persistent connections in HTTP/1.1 are indicated by the header `Connection: keep-alive`. Look for this in the `GET` request and the server's response. However, there're no such header, so not persistent connections.