HTTP PERFORMANCE

- **Objective:** In this problem, we consider the performance of HTTP, comparing non-persistent HTTP with persistent HTTP.

- Suppose the page your browser wants to download is **500 Kbits** long, and contains **5 embedded images** (with file names img01.jpg, img02.jpg, … img05.jpg), each of which is also **100 Kbits** in length.

- The page and the 5 images are all stored on the same server which has a **250 ms RTT** from your browser. We will abstract the network path between your browser and the web server as an**100 Mbps** link.

HTTP PERFORMANCE

- **Objective:** In this problem, we consider the performance of HTTP, comparing non-persistent HTTP with persistent HTTP.

- Suppose the page your browser wants to download is **500 Kbits** long, and contains **5 embedded images** (with file names img01.jpg, img02.jpg, … img05.jpg), each of which is also **100 Kbits** in length.

- The page and the 5 images are all stored on the same server which has a **250 ms RTT** from your browser. We will abstract the network path between your browser and the web server as an**100 Mbps** link.

HTTP PERFORMANCE

- **Objective:** In this problem, we consider the performance of HTTP, comparing non-persistent HTTP with persistent HTTP.

- Suppose the page your browser wants to download is **500 Kbits** long, and contains **5 embedded images** (with file names img01.jpg, img02.jpg, … img05.jpg), each of which is also **100 Kbits** in length.

- The page and the 5 images are all stored on the same server which has a **250 ms RTT** from your browser. We will abstract the network path between your browser and the web server as an**100 Mbps** link.

HTTP PERFORMANCE

- You can assume that the time it takes to transmit a **GET** message into the path is **zero**, but you should account for the time it takes to transmit the base file and the embedded objects into the "link." This means that the server-to-client “link” has both a **125 ms** one-way propagation delay, as well as a transmission delay associated with it. The time needed to setup up TCP connections is **1 RTT**.

NON PERSISTENT HTTP

1- Assuming non-persistent HTTP (and assuming no parallel connections are open between the browser and server). How long is the ***response time*** - the time from the when the user requests the URL to the point in time when the page and its embedded objects are displayed? Make sure you describe the various components that contribute to this delay.

2- Again assume non-persistent HTTP, but now assume that the browser can open as many parallel TCP connections to the server as it wants. What is the response time in this case?

SOLUTION

1- The transmission time of the 500 Kbits page

= L/R = $500\times10^{3} / 100\times10^{6} = 5\times10^{-3}$ =5ms

- The transmission time of each 100 Kbits image

= $100\times10^{3} / 100\times10^{6} = 10^{-3}$ = 1ms

- The delays associated with this scenario are:

= 2*RTT *6 + transmission time for file & images

= 2*250*6 + 5ms + (5*1)ms = 3.01 s

SOLUTION

2- The delays associated with this scenario are:
- 2 RTT for first object +
- 2 RTT for all images (parallel) +
- transmission time for file +
- transmission time for images

= 500 ms + 500 ms + 5 ms + 5 ms = 1010 ms = 1.01 s

PERSISTENT HTTP

3- Now assume persistent HTTP (i.e., HTTP1.1). What is the response time, assuming no parallel connections?

4- Now suppose persistent HTTP with parallel connections is used. What is the response time?

SOLUTION

3- 2RTT for base file + RTT*5 (for the 5 images) + transmission time for file + transmission time for images
= 500 + 250*5 + 5 +5 = 1760 ms = 1.76 s

4- 2RTT for the base file + RTT for all images (parallel)+ transmission time for file + transmission time for images
= 500 + 250 + 5 + 5 = 76 ms = 0.76 s