Web Protocols and Practice

Assignment 2 – Solution

Exercise 2.1

Sending voice from host A to host B. Host A converts analog voice to digital 64-Kbps bits stream. Host A group the bits into 48-bytes packets. There is one link in between them with a transmission rate of 1-Mbps and propagation time of 2msec.

Consider the first bit in a packet. Before this bit can be transmitted, all of the bits in the packet must be generated. This requires:

$$\frac{48 * 8}{64 * 10^3} = 6 \text{ msec.}$$

The time required to transmit the packet is:

$$\frac{48 * 8}{1 * 10^3} = 384\ \mu\text{sec.}$$

Propagation delay = 2 msec.

The delay until decoding is:

= 6 msec + 384 μsec + 2 msec = 8.384 msec.

Exercise 2.2

Suppose all packets are L bits and the transmission rate is Rbps. Also N packets arrive to the buffer every LN/R seconds.
Then the average queueing delay of the N packets:

The first packet has no queueing delay

The second packet has a queueing delay of L/R

The third packet has a queueing delay of 2(L/R)

The nth packet has a queueing delay of (n-1) L/R

The average delay is:

$$= \sum_{n}^{N} (n-1)\, L/R = (1/N)(L/R) \sum_{n=0}^{N-1} (n)$$

$$= (1/N)\,(L/R)\,(N-1)\,N/2$$

$$= (L/R)\,(N-1)/2.$$

Exercise 2.3

Let I denote the traffic intensity, I = La/R. Suppose that the queueing delay takes the form IL/R(1-I), for I<1,

a: arrival rate of packets
R: transmission rate of the bits.

a) total delay

The transmission delay is L/R, the total delay is:

$L/R + \mathit{I}L/R(1-\mathit{I})$

$= \frac{L(1-\mathit{I}) + \mathit{I}L}{R(1-\mathit{I})}$

$= L / R(1 - \mathit{I})$

$= (L/R)/(1 - \mathit{I})$.

b) Plot the total delay as a function of L/R

Let x = L/R

Total delay = x/(1-ax)

Asymptotical behaviour by the value 1/a.

Exercise 2.4

a)

i. For these protocols to avoid data loss full reliable data transfer is required hence the use of TCP rather than UDP by HTTP, FTP, of financial transactions.
ii. TCP is used due to its connection oriented. Client and Server exchange transport-layer control information before the application-level messages begin to flow (handshaking). After the handshaking phase, the TCP connection is known to exist.

b) e-commerce with HTTP authentication, and with cookies.

i. Authentication:

1. Client sends an ordinary request message with no special header lines
2. Server responds with empty body and 401 authorization request status code (username & password).
3. Client receives the response message and prompts the user for a username and password.
4. The client resends the request message and the authorization header line.
5. After receiving the first object the client continue to send the username and password in subsequent request for objects.

ii. Cookies:

1. Client contact a web site for the first time, and the site uses cookies.
2. Server response will include a set-cookie header, which often contain an ID number set-cookie.
3. The HTTP client receives the response with the header and number, and appends a line to a special cookie file (on the client machine). The file contains the server name and the user's associated ID number.
4. In the subsequent request to the same server, the client includes a cookie header and the header line specifies the identification number for that server.

c) Persistent HTTP with pipelining and without pipelining:

1. HTTP without pipelining: The client issues a new request only when the previous response has been received.

   Total RTTs : 2 RTT + n RTT (n number of documents)

2 HTTP with pipelining:

   Total RTTs : 2RTT + 1RTT.

Note: HTTP/1.1 uses persistent with pipelining.

Exercise 2.5

a) nDNS and html file: one object

Total time: RTT1 + … + RTTn + 2RTT0

b) html files containing 3 objects:

case 1: where the base html file is considered

i. nonpersistent HTTP with no parallel TCP connections:

   Total time : RTT1 + … +RTTn + 2RTT + 3* (2RTT0).

ii. nonpersistent HTTP with parallel TCP connections:

   *Total time : RTT1 + … + RTTn + 2RTT0 + 2RTT0.

iii. persistent HTTP with pipelining:

   Total time : RTT1+ … + RTTn + 2RTT0 + RTT0.

* The argument here is that the parallel connections can be three or more.

case 2: where the base html file is not considered

i. nonpersistent HTTP with no parallel TCP connections:

Total time : RTT1 + … +RTTn + 3* (2RTT0).

ii. nonpersistent HTTP with parallel TCP connections:

Total time : RTT1 + … + RTTn + 2RTT0.

iii. persistent HTTP with pipelining:

Total time : RTT1+ … + RTTn + RTT0.