Homework 1 Suggested Solutions

Question 1

This information can be found online and textbook, but to summarise:

- Business deals between ISPs to provide faster routes at lower prices
- ISPs subscribes to IXP to forward consumer data between each other

Question 2

Part 1

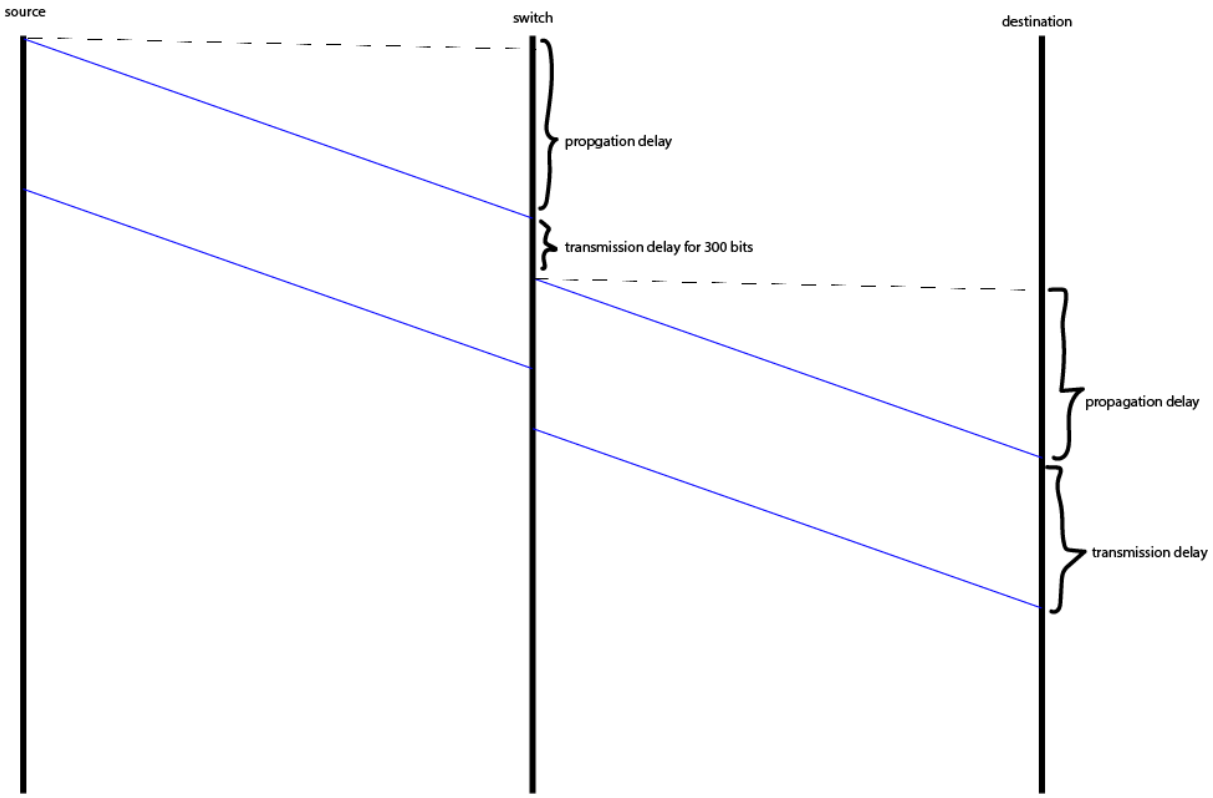
Since this is a store-and-forward system,
Transmission delay of each link: 12e3 / 100e6 = 120 microseconds
Total delay of each link: $t_{trans} + t_{prop} = 132\ \mu s$
End-to-end delay: $264\ \mu s$

Part 2

Same as above but multiplied by 5
End-to-end delay: $660\ \mu s$

Part 3

Draw the space time diagram:



Total delay: $12\mu s + \frac{300}{100e6} + 12\mu s + 120\mu s = 147\mu s$

Question 3

Part 1

Remember to include the protocol (HTTP)

`http://gaia.cs.umass.edu/cs453/index.html`

Part 2

HTTP/1.1, as seen in the request line.

Part 3

The browser is requesting a persistent connection (`Connection: keep-alive`).

Pros (for non-persistent): it saves server resources (keeping open sockets) and is simpler to implement (along with other reasons)

Cons (for non-persistent): significant overhead from opening and closing TCP connection when sending many requests (along with other reasons)

Part 4

You can't tell. IP address information is not available in the application layer.

Part 5

A desktop web browser (Mozilla Firefox). It helps the server make decisions on what kind of document to return to the client depending on the browser type/version, for instance, redirecting to a mobile version of the web site.

Note: in practice however, because the agent field is easily spoofed, most web servers delegate compatibility checks to Javascript in the HTML document. The most common use case is for tracking and analytics.

Part 6

Since the HTTP request line is always encoded in ASCII, the number of characters is equal to the number of bytes. 30 + 2 (for the CRLF) = 32.

Question 4

Part 1

Yes because of the 200 status code in the status line. The date can be found in the `Date` header (07 Mar 2008)

Part 2

The date in the `Last-Modified` header. (10 Dec 2005)

Part 3

The number of bytes in the `Content-Length` header (3874)

Part 4

The body starts after the double CRLF. Since the request body is in ASCII (default if `Content-Encoding` is not specified), the first 5 bytes is **the ASCII representation of** `<!doc`

Server agreed to persistent connection.

Part 5

The last modified and etag header can be useful to caches and web browsers to tell them whether their version of the document is outdated.

Question 5

Part 1

During the freezing period, the application freezes (no playing) until the minimum buffer size (Q) has been reached.

$$t_{freeze} = Q/x$$

When the contents of the buffer are more than Q, playout begins. The application plays until the buffer is completely empty. Important assumption here is that the depletion rate is more than the fill rate (as given in the question).

$$t_{playout} = Q/(r - x)$$

Part 2

This is like filling a water bottle with a hole at the bottom, but the leaking rate is smaller than the fill rate (as given by the question), so it will eventually become full. However the leak (playback) only occurs when the min buffer size is reached.

$$t_{full} = \frac{Q}{x} + \frac{B - Q}{x - r}$$

Question 6

Part 1

Equation from Textbook Chapter 2 Section 5 Figure 2.1

$$D_{cs} \geq \max\left\{\frac{NF}{u_s}, \frac{F}{d_{\min}}\right\}.$$

Since all the peers have the same download rate, $d_{min} = d_i$

$$\frac{NF}{u_s} = \frac{100 * 6e9}{30e6} = 20000s$$

$$\frac{F}{d_{min}} = \frac{F}{d_i} = \frac{6e9}{2e6} = 3000s$$

The minimum time to distribute is thus 200000s.

Part 2

Equation from Textbook Chapter 2 Section 5 Figure 2.2

$$D_{\text{P2P}} \geq \max\left\{\frac{F}{u_s}, \frac{F}{d_{\min}}, \frac{NF}{u_s + \sum_{i=1}^{N} u_i}\right\}$$

All the peers have the same upload and download rate (given).

$$\frac{F}{u_s} = \frac{6e9}{30e6} = 200s$$

$$\frac{F}{d_{min}} = \frac{F}{d_i} = \frac{6e9}{2e6} = 3000s$$

$$\frac{NF}{u_s + \sum_{i=1}^{N} u_i} = \frac{100 * 6e9}{30e6 + 100e6} \approx 4615$$

The minimum time to distribute is thus 4615s.

Question 7

Note: there are many possible answers for these, the ones below are just one suggestion

Part 1

They all use TCP by default as the underlying transport layer. All of these applications require a reliable data transfer method to ensure delivery and integrity of messages.

Part 2

HTTP (in the simplest implementation) is not connection oriented (despite using TCP) and stateless by design because it operates on a request-response model for fetching web documents. SMTP on the other hand is connection oriented because a mail client can establish a single connection to send multiple requests to the server.

Part 3

MQTT offers service levels that include “at most once” delivery, which means the messages might not necessarily reach all subscribers. This is because MQTT is used in embedded IoT applications where network conditions may not be stable. On the other hand, SMTP requires complete transmission of messages to ensure document integrity.

Part 4

HTTP is half-duplex by design because it operates on a request-response model meant to fetch static web documents. Websocket on the other hand offers an asynchronous full duplex communication channel between the client and the server for real time applications such as chat applications.