CNT 3004C-01 Intro to Computer Networks **Brandon Thompson**
Prof. Ding Due Date: 2/24/20

Web 1

Obtain the HTTP/1.1 specification (RFC 2616). Answer the following questions:

PROBLEM #1:

Explain the mechanism used for signaling between the client and server to indicate that a persistent connection is being closed. Can the client, the server, or both signal the close of a connection?

SOLUTION:

- Mechanism used for signaling persistent connection to close is a connection-token of ”close” in the connection header sent in the request that request becomes the last one for the connection (section 8.1.2.1 of RFC 2616).
- Both the client and server can signal the closing of a connection by using the Connection header field (section 8.1.2 of RFC 2616).

PROBLEM #2:

Can a client open three or more simultaneous connections with a given server?

SOLUTION:
Clients should limit the number of simultaneous connections that they maintain to a given server. A single-user client should not maintain more than 2 connections with any server or proxy. A proxy should use up to $2 * N$ where N is the number of simultaneous active users (section 8.1.4 RFC 2616).

Assuming that the client in question is a single-user client, no, they would only be able to open 2 connections to a server.

PROBLEM #3:

Either a server or a client may close a transport connection between them if either one detects a connection has been idle for some time. Is it possible that one side starts closing a connection while the other side is transmitting data via this connection? Explain.

SOLUTION:
A client, server or proxy may close the transport connection at any time. For For example, a client might have started to send a new request at the same time that the server decides to close the ”idle” connection. From server’s point of view, the connection is being closed because it was idle, but from the client’s point of view there is a request in progress (section 8.1.4 RFC 2616).

Yes, a connection could be closed on one end but not on another. RFC also states that ”clients, servers, and proxies must be able to recover from asynchronous close events.”