MPCS 54001, Winter 2021

Homework 2 - Solutions

Due: January 28 at 5:30 pm

Submit your assignment to Gradescope. This work must be entirely your own. If you need help, post questions to Ed Discussion and/or visit the staff during office hours. As a reminder, if you make a public post on Ed Discussion, please don't give away the answer!

---

1. Consider the transport layer used by HTTP-based protocols.

    a. Why does HTTP typically run on top of TCP rather than UDP? (5 points)
    HTTP requires that all application data be received in the correct order and without gaps. TCP provides this service whereas UDP does not.

    b. Which transport protocol does DASH run over and why? (10 points)
    Although multimedia streaming applications historically ran over UDP, Dynamic Adaptive Streaming over HTTP (DASH) runs over TCP because it is HTTP-based.

    c. What motivated the creation of the QUIC protocol? Name one reason why QUIC is implemented at the application layer, rather than the transport layer. (10 points)
    QUIC offers lower connection establishment latency and improved congestion control compared to the combination of HTTP2 + TLS + TCP.

    It was implemented at the application layer due to ease of deployment as an application library used by browsers and web servers. Updating the transport layer would require implementation in the operating system kernel, which would be much more difficult to roll out.

2. Describe how the following features improve HTTP performance: HTTP persistent connections, HTTP pipelining, and parallel HTTP connections. Why are HTTP persistent connections necessary for HTTP pipelining? (25 points)

TCP was not optimized for very short-lived connections (e.g., an HTTP message with no more than 10 segments could potentially have 7 segments of connection overhead).

HTTP persistent connections allow established TCP connections to stay open for multiple request-response message exchanges, rather than only a single message exchange. This reduces connection overhead because the cost of a single connection is amortized across multiple message exchanges.

HTTP pipelining over persistent connections allows the client to send multiple requests to the server without waiting for each individual response before sending the next request. The elimination of the additional round-trip waiting times reduces latency.

Parallel HTTP connections allow the client to open multiple simultaneous connections to the server to download multiple objects in parallel. Although the client might experience better individual performance, this approach increases network congestion and server load.

3. To answer the following questions, you will probably want to read the documentation for the `dig` tool. In your answers, be sure to specify actual commands that can be run from the Linux command line.

    a. What `dig` command would trace the delegation path for `cs.illinois.edu`? (5 points)
    dig +trace cs.illinois.edu

    b. What `dig` command would *directly* query one of the root name servers for `cs.illinois.edu` in *recursive mode*? Was a recursive query actually performed? Why or why not? (10 points)
    dig +recurse @{a-m}.root-servers.net cs.illinois.edu

    No, a warning message indicated that the recursive query was not performed even though recursive mode was requested. Root name servers typically do not

perform recursive queries. If they did, then they would be more susceptible to overload (and DoS attacks).

c. What `dig` command would only return the authoritative name servers for `cs.illinois.edu`? (5 points)
dig cs.illinois.edu NS

d. What `dig` command would *directly* query an authoritative name server for the IP address of `cs.illinois.edu`? (5 points)
dig @dns{1-3}.illinois.edu cs.illinois.edu A
(Explicitly specifying record type A above is optional).

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

a. 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? (5 points)
Persistent connections are discussed in section 8 of RFC 2616. Sections 8.1.2 and 8.1.2.1 of the RFC indicate that either the client or the server can indicate to the other that it is going to close the persistent connection. It does so by including the connection-token "close" in the Connection-header field of the http request/reply.

b. What encryption services are provided by HTTP? (5 points)
HTTP/1.1 does not provide any encryption services.

c. Can a client open three or more simultaneous connections with a given server? (5 points)
(From RFC 2616) "Clients that use persistent connections 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." However, as discussed in class, many browsers ignore this guidance in practice.

d. Either a server or a client may close a transport connection between them if either one detects the 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 connect? Explain. (10 points)

   Yes. (From RFC 2616) "A client might have started to send a new request at the same time that the server has decided to close the "idle" connection. From the server's point of view, the connection is being closed while it was idle, but from the client's point of view, a request is in progress."