Assignment 4: App Layer 2

Problem 1. Non-Persistent vs. Persistent HTTP 1.1

(2pts) This is extension of the question in the previous assignment. Here, you MUST consider propagation delays (=RTTs). The idea of the previous assignment was to give you understanding that regardless whether you are sending file in parallel or sequential, you CANNOT exceed the available bandwidth: if you have 5Mbps, you can only send max 5Mbps (you can send at slower speed, but not faster). If you are sending something in parallel, then those transmissions will simply share that 5Mbps (each will be using less). Therefore, the savings with HTTP 1.0 vs HTTP 1.0/multiple connections vs HTTP 1.1/persistent can only come from the propagation delays---the connection setup, the number and sequence of request being sent. The reason you can save here is because the requests are very small and will (practically) never use the whole available bandwidth.

Now, consider a very simple webpage shown below. Calculate the minimum time needed to retrieve the page and referenced images using HTTP 1.0 with parallel connections (when you setup as many TCP connections as you need) and HTTP/1.1 with persistent connection.

Assume RTT between client and server to be 1 second, bandwidth 10Mbps (=1280 Kilobytes/s), HTML page size 128 kilobytes, each image size exactly 320 Kilobytes) (you can ignore processing delays, but compared to assignment 3, you need to consider the propagation delay). For each, show time in seconds and briefly **HOW** you calculated the value.

For each calculation, remember to include RTT for TCP connection establishment, time to send HTTP request, and time to receive the requested info. For parallel case, bandwidth cannot magically appear. In other words, transmission delay is directly related to channel bandwidth, even in parallel case, it is required for each individual file.

```
<!DOCTYPE html>
<html>
 <body>
  <p>
   <img src="IMG_4083.jpg" width="400px" alt="Hot pool"/>
   <img src="IMG_4113.jpg" width="400px" alt="Geiser" />
   <img src="IMG_4114.jpg" width="400px" alt="More Geisers" />
  </p>
      <a href="https://en.wikipedia.org/wiki/Foobar">Foobar</a>
      … more HTML to make it 64 kilobytes of text
 </body>
</html>
```

a) Non-persistent parallel HTTP 1.0 (new TCP connection for each request, BUT **can create parallel** TCP connections)

b) Persistent HTTP 1.1 **with pipelining** (can re-use TCP connection for requests, single TCP connection at a time)

1RRT + (3 * 320Kbps) + 128Kbps

Problem 2. SMTP

(2pts)

Let's assume you are running your own SMTP server and the server received an email with the following headers in it:

```
Return-Path: <bounces+23855172@em2930.service.tiktok.com>
Delivered-To: aa@cs.fiu.edu
Received: from mail.cs.fiu.edu ([127.0.0.1])
          by mail.cs.fiu.edu with LMTP
          id B0LwK3tP4GPacxQAj2tSdQ
          (envelope-from <bounces+23855172-49a2 @em2930.service.tiktok.com>)
          for <aa@cs.fiu.edu>; Sun, 05 Feb 2023 19:53:15 -0500
...
Date: Mon, 06 Feb 2023 00:53:14 +0000 (UTC)
From: TikTok <notification@service.tiktok.com>
Message-ID: <a87ae172-aed9-4cdd-9a22-0353f4e42aec.notification@service.tiktok.com>
Subject: Leon The Cat Dad: I think you know my #cat
Reply-To: edm.support@tiktok.com

...
```

a) Can your SMTP server reliably determine that this email is indeed from TikTok and not some kind of spam?

No

b) Assuming this email was received from the TCP connection from IP 149.72.157.70, will such email pass an SPF check? For this task, you need to manually run SPF check: pick the right domain name and determine if that domain designates 149.72.157.70 as email server.

No the IP address does not resolve to 149.72.157.70

c) If SPF check succeeds, does it guarantee the email is not a spam? Briefly explain.

Problem 3. Video Streaming and CDN

(2pts)

a) Video Streaming

Which TCP-based protocol Youtube is using to "stream" videos? What aspect of "streaming" the client can control the streaming?

UDP

The client can control the content requested from the CDN servers

b) HTTP Proxy vs CDN

What are the differences between "traditional" HTTP proxies and CDN services? Are they completely different concepts, have some similarities, the same? Briefly explain why.

They are similar concepts. content delivery networks use the same HTTP request and response pattern to deliver data similar to HTTP proxies which behave as intermediates for client to server connections.

Problem 4. DNS

(3pts)

From the lecture, book, and other resources, you have learned the basics of the DNS protocol. The objective of this task is to master these basics and learn the available tools. To answer these questions, you would need to use "dig" command-line tool or its web equivalent (https://www.digwebinterface.com).

a) How does a caching resolver (your computer) knows where to send "recursive" DNS queries (= IP of the caching resolver)?

the caching resolver queries the root zone servers for lists of domains that match the domain . if none of the delegated DNS servers return a domain that matches the desired domain name, then the caching resolver queries the next level of delegated DNS servers.

b) How does a stub resolver knows where to send queries when its cache is completely empty? (= IP of a root zone server)?

there is a pre-configured file that has the IP address of a set of root servers.

c) Using dig command or web interface, obtain A record for "www.microsoft.com" from two different regions (one from USA and one outside USA). In command line, you need to use @ip-of-caching-resolver and in web interface, you need to select a different caching resolver

USA : @68.47.229.151 : 20.112.52.29

Turkey : @46.20.159.27 : 20.53.203.50

d) Results of DNS queries are designed to be cached for a period of time by the resolvers. Where this time comes from? Show example (e.g., using the previous part)

Caching servers only store the most recent domains requested and domains which are spatially adjacent to those domains. the database/list is updated continuously and domains low in priority fall off the database/list.