Fall 2024, CS 3953: Computer Networks

Homework 2 Solution

Solution to problem 1

SMS (Short Message Service) is a technology that allows the sending and receiving of text messages between mobile phones over cellular networks. One SMS message can contain data of 140 bytes and it supports languages internationally. The maximum size of a message can be 160 7-bit characters, 140 8-bit characters, or 70 16-bit characters. SMS is realized through the Mobile Application Part (MAP) of the SS#7 protocol, and the Short Message protocol is defined by 3GPP TS 23.040 and 3GPP TS 23.041. In addition, MMS (Multimedia Messaging Service) extends the capability of original text messages, and support sending photos, longer text messages, and other content.

iMessage is an instant messenger service developed by Apple. iMessage supports texts, photos, audios or videos that we send to iOS devices and Macs over cellular data network or WiFi. Apple's iMessage is based on a proprietary, binary protocol APNs (Apple Push Notification Service).

WhatsApp Messenger is an instant messenger service that supports many mobile platforms such as iOS, Android, Mobile Phone, and Blackberry. WhatsApp users can send each other unlimited images, texts, audios, or videos over cellular data network or WiFi. WhatsApp uses the XMPP protocol (Extensible Messaging and Presence Protocol).

iMessage and WhatsApp are different than SMS because they use data plan to send messages and they work on TCP/IP networks, but SMS use the text messaging plan we purchase from our wireless carrier. Moreover, iMessage and WhatsApp support sending photos, videos, files, etc., while the original SMS can only send text message. Finally, iMessage and WhatsApp can work via WiFi, but SMS cannot.

Solution to problem 2

Application layer protocols: DNS and HTTP

Transport layer protocols: UDP for DNS; TCP for HTTP

Solution to problem 3

a) The document request was http://gaia.cs.umass.edu/cs453/index.html. The Host: field indicates the server's name and /cs453/index.html indicates the file name.

b) The browser is running HTTP version 1.1, as indicated just before the first <cr><lf> pair.

c) The browser is requesting a persistent connection, as indicated by the Connection: keep-alive.

d) This is a trick question. This information is not contained in an HTTP message anywhere. So there is no way to tell this from looking at the exchange of HTTP messages alone. One would need information from the IP datagrams (that carried the TCP segment that carried the HTTP GET request) to answer this question.

e) Mozilla/5.0. The browser type information is needed by the server to send different versions of the same object to different types of browsers.

Solution to problem 4

The total amount of time to get the IP address is RTT1 + RTT2 + · · · + RTTn

Once the IP address is known, RT T0 elapses to set up the TCP connection and another RTT0 elapses to request and receive the small object. The total response time is 2RTT0 + RTT1 + RTT2 + · · · + RTTn

Solution to problem 5

a) RTT1 + · · · + RTTn + 2RTT0 + 8 × 2RTT0 = 18RTT0 + RTT1 + · · · + RTTn.

b) RTT1 + · · · + RTTn + 2RTT0 + 2 × 2RTT0 = 6RTT0 + RTT1 + · · · + RTTn

c) Persistent connection with pipelining. This is the default mode of HTTP:
RTT1 + · · · + RTTn + 2RTT0 + RTT0 = 3RTT0 + RTT1 + · · · + RTTn.
Persistent connection without pipelining, without parallel connections:
RTT1 + · · · + RTTn + 2RTT0 + 8RTT0 = 10RTT0 + RTT1 + · · · + RTTn.

Solution to problem 6

1. The time to transmit an object of size L over a link or rate R is L/R. The average time is the average size of the object divided by R:
Δ = (900, 000 bits)/(20, 000, 000 bits/sec) = 0.045 sec
The traffic intensity on the link is given by
$\beta\Delta$ = (20 requests/sec)(0.045 sec/request) = 0.900.
Thus, the average access delay is (0.045 sec)/(1 - 0.900) ≈ 0.45 seconds.
The total average response time is therefore 0.45 sec + 3 sec = 3.45 sec.
2. The traffic intensity on the access link is reduced by 70% since 70% of the requests are satisfied within the institutional network. Thus the average access delay is
(0.045 sec)/[1 – (0.3)(0.900)] = 0.062 sec
The response time is approximately zero if the request is satisfied by the cache (which happens with probability 0.7);
The average response time is 0.062 sec + 3 sec = 3.062 sec for cache misses (which happens 30% of the time).
So the average response time is (0.7)(0 sec) + (0.3)(3.062 sec) = 0.919 sec
Thus the average response time is reduced from 3.062 sec to 0.919 sec.

Solution to problem 7

1. Consider a distribution scheme in which the server sends the file to each client, in parallel, at a rate of a rate of u_s/N. Note that this rate is less than each of the client's download rate, since by assumption $u_s/N \le d_{min}$. Thus each client can also receive at rate u_s/N. Since each client receives at rate u_s/N, the time for each client to receive the entire file is $F/(u_s/N) = NF/u_s$. Since all the clients receive the file in NF/u_s, the overall distribution time is also NF/u_s.
2. Consider a distribution scheme in which the server sends the file to each client, in parallel, at a rate of d_{min}. Note that the aggregate rate, Nd_{min}, is less than the server's link rate u_s, since by assumption $u_s/N \ge d_{min}$. Since each client receives at rate d_{min}, the time for each client to receive the entire file is F/d_{min}. Since all the clients receive the file in this time, the overall distribution time is also F/d_{min}.

3. From the section 2.6 of the textbook we know that
$D_{CS} \geq \max\{NF/u_s, F/d_{min}\}$ (Equation 1)
Suppose that us/N $\leq d_{min}$.
Then from Equation 1 we have $D_{CS} \geq NF/u_s$.
But from question1 we have $D_{CS} \leq NF/u_s$.
Combining these two gives:
$D_{CS} = NF/u_s$ when $u_s/N \leq d_{min}$. (Equation 2)
We can similarly show that:
$D_{CS} = F/d_{min}$ when $u_s/N \geq d_{min}$ (Equation 3).
Combining Equation 2 and Equation 3 gives the desired result.