CS3611 Computer Networks (Spring 2023) Homework 2

Xiangyuan Xue (521030910387)

1. SMS (Short Message Service) is a service that allows users to send short text messages to other users. The service allows users to send and receive messages of up to 160 characters (when entirely alpha-numeric) to and from GSM mobiles. The SMS protocol is defined in the 3GPP TS 23.040 standard based on the client-server model. The client is the mobile phone and the server is the SMS center. The Mobile Application Part (MAP) of the SS7 protocol included support for the transport of Short Messages through the Core Network from its inception.

   iMessage is an instant messaging service developed by Apple Inc, which includes sending text messages, images, videos and documents, getting delivery and read statuses, and end-to-end encryption so only the sender and recipient can read the messages. The iMessage protocol is based on the Apple Push Notification service (APNs), which is a proprietary, binary protocol.

   WhatsApp (WhatsApp Messenger) is an internationally available freeware, cross-platform, centralized instant messaging (IM) and voice-over-IP (VoIP) service owned by US tech conglomerate Meta. It allows users to send text and voice messages, make voice and video calls, and share images, documents, user locations, and other content. WhatsApp uses a customized version of the open standard Extensible Messaging and Presence Protocol (XMPP).

   iMessage and WhatsApp are different from SMS in that they are based on TCP/IP protocol, while SMS is based on SS7 protocol provided by the mobile network operator, so iMessage and WhatsApp can work via Wi-Fi but SMS cannot. Moreover, iMessage and WhatsApp support multimedia messages, while SMS only supports text messages. iMessage and WhatsApp work on different platforms, and WhatsApp provides even more features than iMessage.

2. On application layer, DNS protocol is needed besides HTTP protocol.

   On transport layer, TCP protocol is needed for HTTP, and UDP protocol is needed for DNS.

3. (a) The server was able to find the document successfully, which is indicated by the status code “200” and the reason phrase “OK”. The reply was provided at 12:40:46 (Greenwich Mean Time) on Tuesday, March 7th, 2008.

(b) The document was last modified at 18:28:47 (Greenwich Mean Time) on Saturday, December 10th, 2005.

(c) There are 3848 bytes in the document being returned.

(d) The first 5 bytes of the document being returned are "<!doc". The server agreed to a persistant connection, which is indicated by the header "Keep-Alive".

4. Fetching the IP address of the target server costs

$$T_{\text{DNS}} = \sum_{i=1}^{m} \text{RTT}_i$$

Since it takes RTT_0 to set up the TCP connection, and another RTT_0 request and response the object, fetching the HTML text costs

$$T_{\text{HTTP}} = 2\text{RTT}_0$$

Therefore, the time elapsed from the click to the receiption is

$$T_{\text{total}} = T_{\text{DNS}} + T_{\text{HTTP}} = 2\text{RTT}_0 + \sum_{i=1}^{m} \text{RTT}_i$$

5. (a) Since each reference costs 2RTT_0, the time elapsed is

$$T_1 = 2\text{RTT}_0 + \sum_{i=1}^{m} \text{RTT}_i + 7 \times 2\text{RTT}_0 = 16\text{RTT}_0 + \sum_{i=1}^{m} \text{RTT}_i$$

(b) Since it takes 2 rounds of requests and responses with 5 parallel connections, the time elapsed is

$$T_2 = 2\text{RTT}_0 + \sum_{i=1}^{m} \text{RTT}_i + 2 \times 2\text{RTT}_0 = 6\text{RTT}_0 + \sum_{i=1}^{m} \text{RTT}_i$$

(c) Consider persistant HTTP without pipelining, the time elapsed is

$$T_3 = 2\text{RTT}_0 + \sum_{i=1}^{m} \text{RTT}_i + 7 \times \text{RTT}_0 = 9\text{RTT}_0 + \sum_{i=1}^{m} \text{RTT}_i$$

Consider persistant HTTP with pipelining, the time elapsed is

$$T_4 = 2\text{RTT}_0 + \sum_{i=1}^{m} \text{RTT}_i + \text{RTT}_0 = 3\text{RTT}_0 + \sum_{i=1}^{m} \text{RTT}_i$$

6. (a) The average time to send an object over the access link is

$$\Delta = \frac{750000\text{bits}}{15\text{Mbps}} = 0.05\text{sec}$$

Since the arrival rate is $\beta = 18/\text{sec}$, the average access delay is

$$\frac{\Delta}{1 - \Delta\beta} = \frac{0.05\text{sec}}{1 - 0.05\text{sec} \cdot 18/\text{sec}} = 0.5\text{sec}$$

Therefore, the total average response time is

$$T_1 = 2\text{sec} + 0.5\text{sec} = 2.5\text{sec}$$

(b) Since the traffic intensity is reduced by 60%, the average access delay is

$$\frac{\Delta}{1-\Delta\beta}=\frac{0.05\text{sec}}{1-0.4\times 0.05\text{sec}\cdot 7.2/\text{sec}}\approx 0.078\text{sec}$$

If the cache hits, it almost takes 0sec to fetch the response. If the cache misses, the response time is

$$2\text{sec}+0.078\text{sec}=2.078\text{sec}$$

Therefore, the total average response time is

$$T_2=0.6\times 0\text{sec}+0.4\times 2.078\text{sec}\approx 0.83\text{sec}$$

7. For client-server distribution, the minimum distribution time can be specified as

$$T_{\text{CS}}=\max\left\{\frac{F}{d},\frac{NF}{u_s}\right\}$$

The table is prepared as follow

Table 1: Minimum distribution time for CS distribution

T_{CS}	$u=300$Kbps	$u=700$Kbps	$u=2$Mbps
$N=10$	7500sec	7500sec	7500sec
$N=100$	50000sec	50000sec	50000sec
$N=1000$	500000sec	500000sec	500000sec

For peer-to-peer distribution, the minimum distribution time can be specified as

$$T_{\text{P2P}}=\max\left\{\frac{F}{d},\frac{F}{u_s},\frac{NF}{u_s+\sum\limits_{i=1}^{N}u_i}\right\}$$

The table is prepared as follow

Table 2: Minimum distribution time for P2P distribution

T_{P2P}	$u=300$Kbps	$u=700$Kbps	$u=2$Mbps
$N=10$	7500sec	7500sec	7500sec
$N=100$	25000sec	15000sec	7500sec
$N=1000$	45455sec	20548sec	7500sec