Fall Term 2006
TelCom 2310
Computer Networks
Monday 3:00 pm - 5:50 pm
Sennott Square 6110
http://www.cs.pitt.edu/_znati/tel2310.html

Homework 4 Solution

Problem 1.

1. Two types of protocols are needed:
    a. Application layer protocols: DNS and HTTP
    b. Transport layer protocols: UDP for DNS; TCP for HTTP
2. Based on the HTTP/1.1 specification:
    a. 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 including the connection-token "close" in the Connection-header field of the http request/reply.
    b. The document does not specify what encryption services are provided. HTTP communication, however, can be encrypted for confidentiality. Netscape developed the original technology called SSL (Secure Sockets Layer) for this purpose. SSL was later enhanced and standardized as TLS (Transport Layer Security) in RFC 2246. TLS is backward compatible with SSL. As the name suggests, encryption is handled on a protocol layer below HTTP. HTTP Client does '''not''' implement TLS nor SSL, it just allows to use them if available.
        1. RFC 2817 specifies the CONNECT method to establish HTTP tunnels through proxies. Such tunnels can then be switched to secure communication.
            - RFC2817: Upgrading to TLS within HTTP/1.1: **http://www.ietf.org/rfc/rfc2817.txt**
        2. RFC 2818 describes how to use HTTP over TLS connections
            - HTTP Over TLS **http://www.ietf.org/rfc/rfc2818.txt**

    Furthermore, the Apache HTTP Server module mod_ssl provides an interface to the OpenSSL library, which provides Strong Encryption using the Secure Sockets Layer and Transport Layer Security protocols.

Problem 2.

1. The total amount of time to get the IP address is:

   a. $RTT_1 + RTT_2 + \cdots + RTT_n$.

   Once the IP address is known, RTT_O elapses to set up the TCP connection and another RTT_O elapses to request and receive the small object. The total response time is:

   b. $2RTT_o + RTT_1 + RTT_2 + \cdots + RTT_n$

2. The time which elapses in each case as follows:
   a. $RTT_1 + \cdots + RTT_n + 2RTT_o + 3 \cdot 2RTT_o = 8RTT_o + RTT_1 + \cdots + RTT_n$.
   b. $RTT_1 + \cdots + RTT_n + 2RTT_o + 2RTT_o = 4RTT_o + RTT_1 + \cdots + RTT_n$.
   c. $RTT_1 + \cdots + RTT_n + 2RTT_o + RTT_o = 3RTT_o + RTT_1 + \cdots + RTT_n$

Problem 3.

a. There are N nodes in the overlay network. There are N(N-1)/2 edges.

b. In this case, each of the five Gnutella clients immediately learns that it has one less neighbor. Consider one of these five clients, called, Bob. Suppose Bob has only three neighbors after X drops out. Then Bob needs to establish a TCP connection with another peer. Bob should have a fresh list of active peers; he sequentially contacts peers on this list until one accepts his TCP connection attempt.

c. In this case, Bob does not immediately know that X has departed. Bob will only learn about X's departure when it attempts to send a message (query or ping) to X. When Bob attempts to send a message, Bob's TCP will make several unsuccessful attempts to send the message to B. Bob's TCP will then inform the Gnutella client that X is down. Bob will then try to establish a TCP connection with a new peer (see part (a)) to rebuild a fifth connection.

Problem 4.

TCP/IP over Ethernet allows data frames with a payload size up to 1460 bytes. Therefore, L = 100, 500 and 1000 bytes are within this limit. The message overhead includes:

- TCP: 20 bytes of header
- IP: 20 bytes of header
- Ethernet: total 18 bytes of header and trailer.

Therefore, we have the following:

- L = 100 bytes, 100/158 = 63% efficiency.
- L = 500 bytes, 500/558 = 90% efficiency.
- L = 1000 bytes, 1000/1058 = 95% efficiency.

Problem 5.

1. The The segment is encapsulated into a TCP segment, which is in turn encapsulated into an IP packet. Consequently, the size of the payload per segment is :

   1500 - 20 -20 = 1460 bytes

   Since the file is 1.5 Mbytes, the number of payload blocks needed to transfer the file is:

   1.5 Mbyte / 1460 byte = 1027.4, or 1028 blocks

   Therefore, the overhead is:

   ((1028 x 1500 - 1.5M)/1.5M) x 100 = 2.8%

2. Assume the stream is segmented as shown below, where the white cells represent data and the shaded cells represent the TCP header overhead.

Therefore, the block size is equal to 80 bytes and the payload size is equal to 40 bytes. Assume zero processing delay due to data arrangement and segmenting. The delay incurred by the first byte of each block = 40/8000 = 0.5 ms.