EP2120 Internetworking/Internetteknik
IK2218 Internets Protokoll och Principer

Homework Assignment 3

Solutions due 23:00, September 30, 2016
Review due 23:00, October 4, 2016

Problems

1. Socket Interface (15 p)

The pseudo-code sample below (with most details omitted) describes an application that uses the socket interface (API) for communication.

```
s = socket(…);
bind(s, …);
listen(s, …);
while true {
    t = accept(s, …);
    recv(t, …);
    ProcessRequest(…);
    send(t, …);
    close(t);
}
```

a) Is the sample code for a client or a server? Does it use TCP or UDP? Explain your answer. (5 p)
b) The textbook gives two examples of communication using the socket interface: 1) connection-oriented, concurrent communication and 2) connectionless, iterative communication. Characterize the communication in the sample code using this terminology. (5 p)
c) In practice, this kind of communication is not frequently used. What is the main limitation? (5 p)

Solution

a) The code is for a TCP server. It does not initiate the communication itself; instead, it waits to be contacted. Hence, it is a server. Only a TCP server uses `listen()` and `accept()`.
b) Connection-oriented, iterative communication.
c) The server can only deal with one client request at a time. (How severe that limitation would be depends on the complexity of the request processing, represented by the `ProcessRequest()` function.)

2. Web (35 p)

Suppose that you click on a link on a web page, which causes the following HTTP request to be sent.

```
GET /swordfish HTTP/1.1
Host: www.feathers.org
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/
*;q=0.8
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_5)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/29.0.1547.76
Safari/537.36
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8
```

a) Which web document does the browser request? Answer by giving the URL. (3 p)

b) Which version of HTTP is the browser using? What kind of connection is requested, persistent or non-persistent? (Note that the textbook does not cover header fields related to persistence, so you may want to consult other resources, on the Internet for example.) (4 p)

The server gives the following response:

```
HTTP/1.1 302 Found
Date: Fri, 23 Sep 2016 12:34:56 GMT
Server: Apache/2.2.3 (Red Hat)
Location: https://www.feathers.org/swordfish
Content-Length: 286
Connection: close
Content-Type: text/html; charset=iso-8859-1

more data…
```

c) What kind of connection does the server accept, persistent or non-persistent? (4 p)

d) The server seems to give a positive response ("302 Found"), but this is not the normal response for a successful request. Answer the following three questions: Why does the server give this response? What is the browser supposed to do? Why do you think the server has been configured to respond in this way? (5 p)

e) Assume that you instead receive a "200 OK" response with an HTML object that contains three other objects (it could be a web page with three images, for instance). Suppose that the client you are using is strictly sequential, so that it fetches one object at a time. How long time does it take from that you click on the link until the entire document has been received? Explain your solution.

The round-trip time between client and server is *RTT*. All objects are very small and the connection is fast, so transmission time is negligible, and so is processing time on the server. Consider the following two cases:

1. Non-persistent HTTP is used.
2. Persistent HTTP is used. (10 p)

f) Suppose instead that the client is configured to use as much parallelism as possible. How long time would it then take for the two cases?

1. Non-persistent HTTP is used.
2. Persistent HTTP is used. (Hint: *HTTP pipelining*) (10 p)

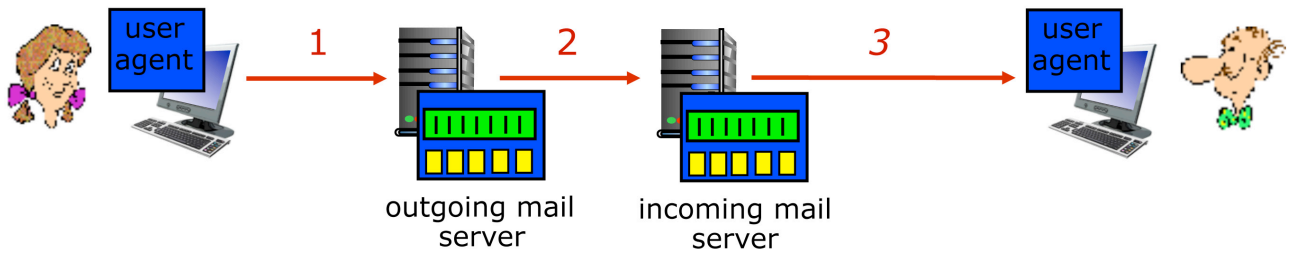
Solution

a) http:/www.feathers.org/swordfish
b) HTTP version 1.1. The browser is requesting a persistent connection.
c) The server accepts a non-persistent connection.
d) 1) This response is a normal response when a page is moved. 2) The client is redirected to the new location, as specified by the "Location" header, so the client should send another request for the new object. 3) In this case, the client is redirected from a non-encrypted connection to an encrypted connection, i.e., from a URL with "http:" to a URL with "https:", so the server has been configured to enforce encrypted communication.
e) The transaction involves fetching the main HTML object, and then the three objects it contains.
   1. 8×*RTT*. First one TCP SYN/ACK and one HTTP request/response for the main HTML object. Then, for each contained object, one TCP SYN/ACK and one HTTP request/response.
   2. 5×*RTT* . One TCP SYN/ACK and then one HTTP request/reply for each object.
f)
   1. 4×*RTT*. First one TCP SYN/ACK plus one HTTP request/response for the main HTML object. Then the contained objects are fetched in parallel, each with a TCP SYN/ACK and a HTTP request/response.
   2. 3×*RTT*. First one TCP SYN/ACK plus one HTTP request/response for the main HTML object. Then the client uses HTTP pipelining and sends requests for the contained objects back-to-back, without waiting for responses in between. The responses are returned back-to-back.

Grading suggestions: The main thing here is to demonstrate an understanding of how TCP SYN/ACK and HTTP request/response are combined. In which order are they performed, and what can be done in parallel? The time it takes to close connections is not relevant, since normally a client would not block waiting for TCP connection termination. However, no points should be deducted for taking connection termination into consideration. Incomplete or missing explanations render deduction in points, though.

3. Mail (25 p)

Consider the scenario when Alice (left) sends an email to Bob (right). There are two intermediate systems: *outgoing mail server* and *incoming mail server*.



Consider the following questions:

a) What is the purpose of the outgoing mail server? Discuss what would happen if it were removed, so that Alice's user agent directly connects to the incoming mail server. What would be the limitations (if any)? (10 p)

b) What is the purpose of the incoming mail server? Discuss what would happen if it were removed, so that the outgoing mail server connects directly to Bob's user agent. What would be the limitations (if any)? (10 p)

c) For each of the connections 1–3, explain what application-layer protocol(s) are used for the mail transfer. (5 p)

Solution

a) If the outgoing mail server is removed, Alice's UA needs to contact the incoming mail server directly in order to send the mail. The incoming mail server may not be able to receive the mail – it could be busy or otherwise unavailable – and then Alice's UA cannot send the mail, and has to wait until the incoming mail server becomes available. Hence, the outgoing mail server offloads Alice's UA by queuing messages waiting to be sent.
(An outgoing mail server is also important for controlling the sending of mail from an organization. It can control what clients are allowed to send mail as a way to prevent abuse, spam for instance. *This is not a required part of the solutions, though, so there is no reduction in points for not mentioning this.*)

b) The incoming mail server stores Bob's mail, and Bob can access the mail whenever he wants. If the server is removed, Bob has to be online in order to receive mail. (It also means that Bob will store all mail on his computer, so he cannot access his mail from another computer. (*No reduction in points for not mentioning this.*)

1: SMTP. 2: SMTP. 3: POP or IMAP (HTTP is also a valid answer, if the UA is web-based).

4. DNS (25 p)

You use the "dig" lookup tool to get the IP address of KTH's web server "www.kth.se". You want to try different name servers, so you specify the DNS server as an argument to dig (the server's IP address prepended with '@'). You run the following three commands:

```
dig @192.36.135.107 www.kth.se
dig @192.36.148.17 www.kth.se
dig @130.237.72.250 www.kth.se
```

a) Explain the results: Describe the responses from the three DNS servers. What do the responses say? We distinguish between four kinds of name servers: root, TLD, authoritative, and local. You can deduce just from studying the responses what kind of DNS server it is (the flags are useful here, among other things). For each of the three cases, describe the kind of name server that is responding to your query. Explain what the response contains. Also, for each name server, explain whether or not the answer contains the IP address you are looking for. *You only need to discuss the responses at a general level – you should not discuss or describe the details of the messages, the different fields, their contents, etc. However, you probably need to study the responses carefully in order to be able to explain them.* (15 p)

b) Organizations often have multiple name servers for their domains. What is the purpose of having multiple name servers? How are they kept synchronized? (5 p)

c) It is sometimes useful to have more than one name in DNS for the same computer. Is it possible for a computer to have two names under different top-level domains? For example, one name in the "com." TLD and one in "se.". If so, give an example when this might be useful. (5 p)

Solution

DNS

a) The first response (192.36.135.107) is from a TLD server for the "se" top-level domain. The response contains the (authoritative) name servers for "kth.se". The second response is from a root server, and gives the TLD servers for the "se" top-level domain. The last answer is from an authoritative server for "kth.se", and gives the IP address of the web server.
b) Having multiple name servers is for redundancy. When an organization has multiple name servers, there is one primary server and several secondary servers. The zone file is updated on the primary server, and then the updates are transferred automatically to the secondaries through "zone transfers".
c) It is perfectly legal to have names in different top-level domains. A web server for the "feathers" company could have the names "www.feathers.com" and "www.feathers.se". When it gets a request to the name in the "se" domain, it responds with a page in Swedish, otherwise in English. (Remember that an HTTP request contains the host name in the "Host:" header field, so the server can examine the header to see what host name was used.)