EP2120 Internetworking/Internetteknik
IK2218 Internets Protokoll och Principer

Homework Assignment 3

Solutions due 17:00, October 4, 2013
Review due 17:00, October 8, 2013

Problems

1. TCP (30 p)

Consider a recently established TCP connection between processes P_A and P_B on hosts A and B, respectively. The three-way handshake has been done, but no data has been sent yet. TCP on Host B announced a receiver window size of 5000 bytes to TCP on Host A, and Process P_B can read the received data from TCP as soon as they arrive. Process P_A has 15000 bytes to send via TCP. The path MTU between the two hosts is known to be 1040 bytes. The propagation time is 200ms, and the link speed is 8Mbps. It takes *1ms* for TCP to generate a segment (with or without data) and this can be done in parallel with sending a previously generated segment.

The receiver uses delayed acknowledgements with a delay of *200ms* (or at most two full segments). The size of a segment having a TCP header only is negligible in terms of transmission time. IP options are not used. Process P_A sends the first segment at time t_0 with sequence number ISN+1. CWND is originally set to 1 MSS and the slow start threshold is 65535 bytes. Assume that the granularity G of the heartbeat timer is 0.5 seconds.

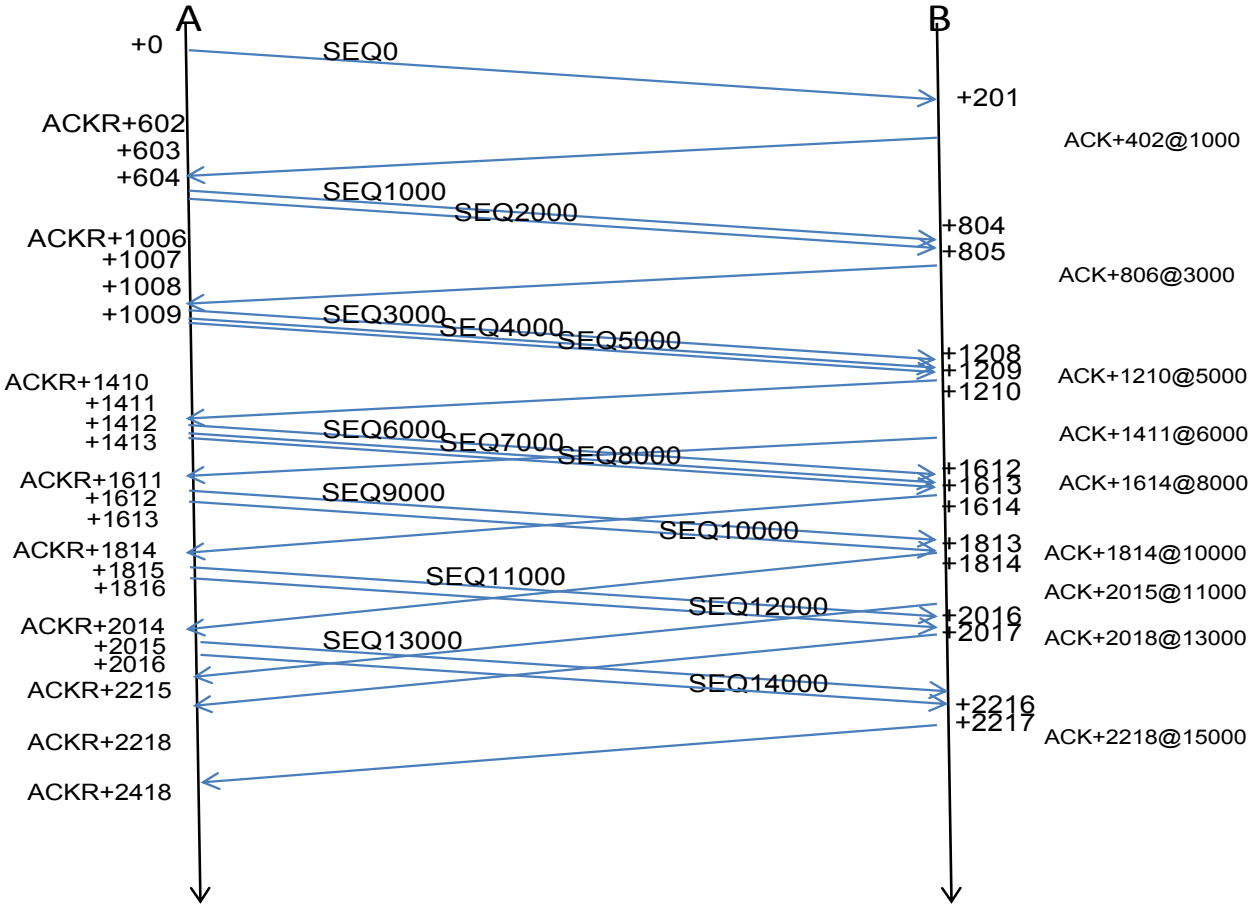
a) What is the MSS used by TCP? (3 p)
b) What is the bandwidth-delay product of the communication channel? Is the advertised receiver window of B big enough? If not, how big should it be for A to be able to fully utilize the channel? (7 p)
c) Provide the sequence of segments sent by TCP from host A. For each segment sent from host A provide the time it is sent, and the sequence number of the first byte it contains. For the first *four* segments sent from host A also provides the SRTT, RTTVAR and the RTO values of the sender TCP at the time the segment is sent. Assume that outgoing segments are handled before incoming segments in case more than one event happens at the same time! (15 p)
d) At what time does A receive the acknowledgement for the last segment? (5 p)

Hint 1: Try to first draw the sequence of segment exchanges to get the order of the segments right.
Hint 2: Consult RFC 2988 for details on how to calculate the SRTT, RTTVAR and the RTO. The description provided in the course book (3ed and 4ed) is not correct.

Solution

a) MSS=1040-20 (TCP header)-20(IP header)
b) The bandwidth delay product is RTTxBW=0.4*8*10^6bits=3.2*10^6 bits=400KB. Thus, the received window is not big enough, it should be at least 400KB.
c) After connection establishment RTO=max(2.5+G,3)=3s, this is at t_0. SRTT=RTTVAR=0.

The following figure shows the segments sent. Sending time is relative to t0, sequence number is relative to ISN+1! (ACK+time@x) is the sending time of an acknowledgement with ESN x, (ACKR+time) is the reception time of an acknowledgement.



The SRTT, RTTVAR and RTO values for the first four segments sent by A are:

1) 0, 0, 3 (initial values as given in the RFC)

ACK received at t0+602. Measured RTT=602ms. SRTT=0.602, RTTVAR=0.602/2=0.301. RTO=SRTT+max(G, 4*RTTVAR)=0.602+1.2=1.806s.

2) 0.602, 0.301, 1.806
3) 0.602, 0.301, 1.806

ACK received at t0+1006. Measured RTT equals 402ms.
RTTVAR=0.75*0.301+0.25*|0.602-0.402|=0.2758,
SRTT=7/8*0.602+1/8*0.402=0.577.
RTO=SRTT+max(G, 4*RTTVAR)=0.577+1.0312 =1.6802s

4) 0.577, 0.2758, 1.6802

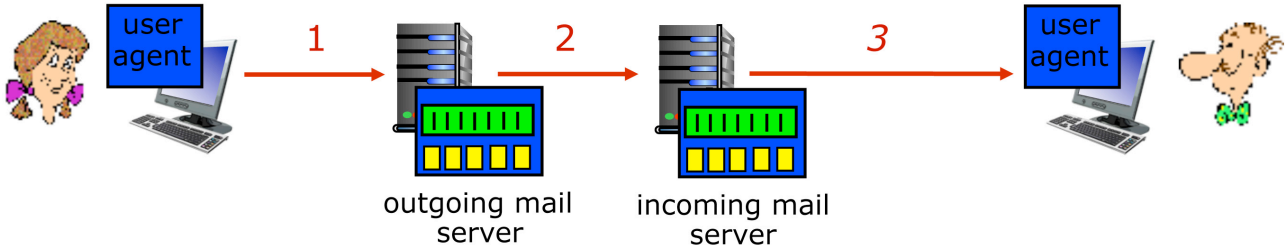
Observe that even though the RTT is constant, due to delayed acknowledgements the RTTVAR is not 0.

d) At t0+2418ms.

Grading suggestions: Important things to consider are whether flow control and congestion control are correctly accounted for (slow start, and max 5 segments outstanding). Delayed ACKs are also important, there should be 1 ACK for every two segments or a delay of 200ms. The 1ms to prepare segments was added to spread out the reception of the segments slightly, missing it is not a major issue, but the solution should be consistent. Missing the transmission time (1ms) is a mistake.

2. 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.*)

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

3. Web (25 p)

Suppose that you click on a link on a web page, which causes the following HTTP request to be sent. ("`\r\n`" denotes carriage return and line feed.)

```
GET /social HTTP/1.1\r\n
Host: www.kth.se\r\n
```

```
Connection: keep-alive\r\n
Cache-Control: max-age=0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/
*;q=0.8\r\n
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\r\n
Accept-Encoding: gzip,deflate,sdch\r\n
Accept-Language: en-US,en;q=0.8\r\n
\r\n
```

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\r\n
Date: Thu, 26 Sep 2013 20:54:23 GMT\r\n
Server: Apache/2.2.3 (Red Hat)\r\n
Location: https://www.kth.se/social/\r\n
Content-Length: 286\r\n
Connection: close\r\n
Content-Type: text/html; charset=iso-8859-1\r\n
\r\n
more data…
```

c) What kind of connection does the server accept, persistent or non-persistent? (3 p)
d) The server seems to give a positive response ("302 Found"), but this is not the normal response for a successful request. Why does the server give this response, and what is the browser supposed to do? (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 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. (5 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*) (5 p)

Solution

a) http:/www.kth.se/social
b) HTTP version 1.1. The browser is requesting a persistent connection.
c) The server accepts a non-persistent connection.
d) This response is a normal response when a page is moved. The client is redirected to the new location, as specified by the "Location" header. (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:").

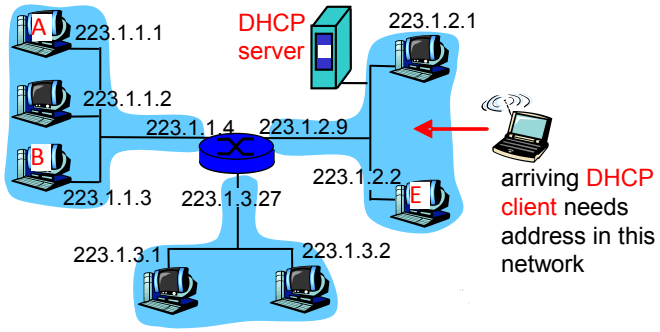
e) The transaction involves fetching the main HTML object, and then the three objects it contains.
   1. 8×*RTT*. First one TCP SYN/ACK and one HTTP request/response for the main HTML object. Then, for each contained object, one TCP SYN/ACK and one HTTP request/response.
   2. 5×*RTT* . One TCP SYN/ACK 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.

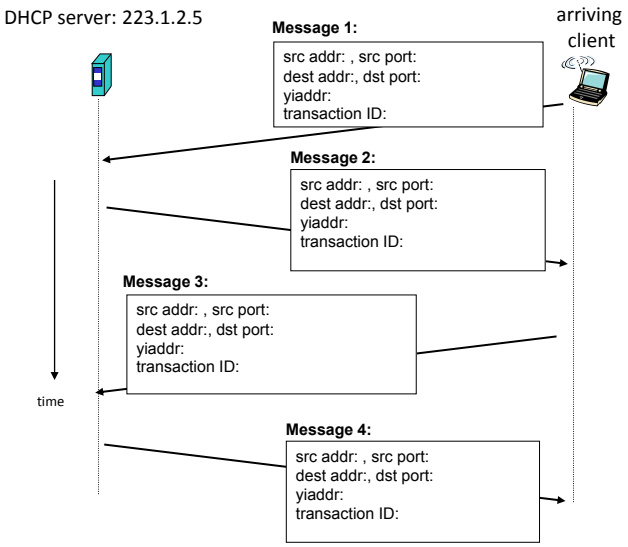
Grading suggestions: The main thing here is to demonstrate an understanding of how TCP SYN/ACK and HTTP request/response are combined. In which order are they performed, and what can be done in parallel? The time it takes to close connections is not relevant, since normally a client would not block waiting 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.

4. DHCP (20 p)

Consider the following scenario, where a DHCP client arrives and requests an IP address from the DCHP server.



In the simplest case, four DHCP messages will be exchanged according to the figure below. Name these four DHCP messages (message type) and fill in the missing fields in each message. You can assume that the subnet to which the DHCP client arrives is a /24 network and that all addresses below 223.1.2.10 are occupied. Based on that, you can let the DHCP server hand out a suitable IP address. You also have to select reasonable transaction IDs.



1

Solution

Valid IP addresses for yipaddr from server are 223.1.2.10-223.1.2.254. Transaction ID should be the same value for DHCP DISCOVER and DHCP OFFER, and a new value to be used in DHCP REQUEST and DHCP ACK. Some implementations of DHCP actually use the same transaction ID for all four messages, so such a solution to this problem should also be considered OK.