Networks Ex02
Itamar Chuvali
200048734

Problem 1

A	B	C	D	E
F	T	F	F	F

Problem 2a

RTT_{dns} + 14RTT_{web}
(Two RTTweb for the page and two for each image)

Problem 2b

RTT_{dns} + 6RTT_{web}
(Two RTTweb for the page, two for the 1st 4 images and two for the last two images)

Problem 2c

RTT_{dns} + 8RTT_{web}
(One RTTweb to establish connection, one for the page and one for each image)

Problem 2d

RTT_{dns} + 3RTT_{web}
(One RTTweb to establish connection, one RTTweb for the page and one pipelined request for the first 4 images and another for the 2 other images in parallel)

Problem 2e

2RTT_{dns} + 4RTT_{web}
(One RTTweb to establish connection, one RTTweb for the page. We then open 2 DNS requests and 4 img requests from original server in parallel. We then establish connection to other servers and get images (2 RTTweb))

Problem 3a

"After an AUTH command has been successfully completed, no more AUTH commands may be issued in the same session. After a successful AUTH command completes, a server MUST reject any AUTH commands with an -ERR reply.
The AUTH command may only be given during the AUTHORIZATION state."

Problem 3b
The section of the document RFC 1939 that defines TOP is "7. Optional POP3 Commands".

In this section we see that "TOP msg n" may only be given in the TRANSACTION state.

Arguments:
msg – a message-number

n – a non-negative number of lines

The POP3 server sends the headers of the message, a blank line, and the first **n** lines of the body of the message with message number: **msg**.

The result for TOP 1 10000 will result in the header of msg 1, a blank line, and the entire message body as it is smaller than 10000.
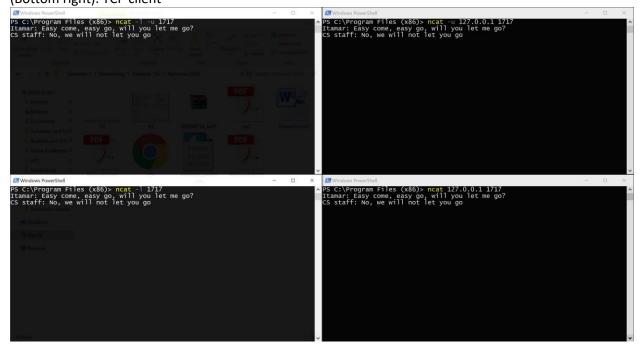
Problem 3c

MTA = Mail Transfer Agents

An MTA is software that transfers electronic mail messages from one computer to another, using client–server application architecture.

58.88.21.177 sent the original msg.

Problem 4a

(Top left): UDP server
(Top right): UDP client
(Bottom left): TCP server
(Bottom right): TCP client

Problem 4b

It still works, UDP is an unreliable connection that doesn't implement any flow control system.

Problem 4c

It doesn't work. TCP requires a 'handshake' which depends on a response from the server.

Problem 4d

It won't work as the listening port on which the connection can be established is not used.

Problem 4e

The UDP client creates a connection but when you attempt to send data, it'll close as there is no server listening to its protocol. The TCP connection remains waiting for a TCP connection (it does not register the UDP client).

Problem 5

```
server [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (07 Dec 2017, 5:06:41 PM)
Exercise 2. November 2017. Itamar Chuvali
GET http://search.lores.eu/indexo.htm HTTP/1.1
Host: search.lores.eu
Proxy-Connection: keep-alive
Cache-Control: max-age=0|
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en,en-GB;q=0.9
If-Modified-Since: Wed, 03 Jun 2015 16:40:40 GMT
```

"If-Modified-Since: Wed, 03 Jun 2015 16:40:40 GMT"

The line above appears in the header. This implies that the cached site will be overridden should the page have been modified any time after that date.

Problem 6a

Not possible.

As the algorithm has only 2 potential inputs, for an invalid agreement to occur, all proper processes must all start off with 0 and eventually agree on 1, or vice versa.

WLOG (between 1 & 0), were this to be the case, as there are 13 proper processes, we will receive s (13) or more 1's and therefore will agree on 1 every iteration.

Problem 6c

Not possible.

Similarly to 6a:

As the algorithm has only 2 potential inputs, for an invalid agreement to occur, all proper processes must all start off with 0 and eventually agree on 1, or vice versa.

WLOG (between 1 & 0), were this to be the case, as there are 13 proper processes, we will receive 16 or more 1's and therefore will agree on 1 every iteration, and therefore also the t iteration.