Name_______________ Student ID____________________ Department/Year___________

2nd Examination

Introduction to Computer Networks (Online)
Class#: EE 4020, Class-ID: 901E31110
Fall 2023

10:20-12:10 Thursday
November 23, 2023

Cautions

1. There are in total 100 points to earn. You have 90 minutes to answer the questions. Skim through all questions and start from the questions you are more confident with.
2. Use only English to answer the questions. Misspelling and grammar errors will be tolerated, but you want to make sure with these errors your answers will still make sense.

1. (ch26, 4pt) Which of the following is true about DASH? (4pt) Grading Policy: -1pt per wrong choice.
   (a) Intelligence at servers
   (b) Server divides a video file into chunks.
   (c) Server encodes each chunk at different rates.
   (d) Client measures client-to-server bandwidth periodically.
   (e) Client chooses coding rates of the chunks depending on available bandwidth.
   (f) Client pauses chunk requests when the buffer is (close to) full.
   (g) Client requests chunks from a nearby server.

Sample Solution:

(b)(c)(e)(f)(g)

2. (ch26, 2pt) A viewer is watching a video on Netflix. Assume Netflix streams videos via DASH and the videos are H.264-encoded in a number of resolutions as follows.

Resolution	Bitrate
2160p (4K)	16 Mbps
1440p (2K)	9.6 Mbps
1080p	5 Mbps
720p	2.5 Mbps
480p	1.2 Mbps
360p	800 Kbps

   (1) Tell the resolution the viewer is watching when the measured available bandwidth is 5 Mbps. (1pt)
   (2) Tell the resolution the viewer is watching when the measured available bandwidth is 1 Mbps. (1pt)

Sample Solution:

(1) 1080p
(2) 360p

3. (ch26, 4pt) Compare two types of CDN – enter deep and bring home.
   (1) Give one advantage and one disadvantage of using the enter deep CDN. (1pt)
   (2) Give one advantage and one disadvantage of using the bring home CDN. (1pt)
   (3) Which one would you prefer starting a company (0pt) and why (2pt)?

Sample Solution:

(1) Advantage: lower delay, lower chance of congestion, lower network-wide bandwidth consumption.
    Disadvantage: high cost

(2) Advantage: lower cost.
    Disadvantage: higher delay, higher chance of congest, higher overall bandwidth consumption.
(3) Anything that's genuinely from you and makes sense

4. (ch3, 6pt) For each of the functions below, tell whether it is implemented in (a) TCP, (b) UDP, (c) both, or (d) none.
   (1) data transfer reliability (1pt)
   (2) data confidentiality (1pt)
   (3) flow control (1pt)
   (4) connection management (1pt)
   (5) multiplexing (1pt)
   (6) connectionless demultiplexing (1pt)

Sample Solution:

(a, d, a, a, c, b)

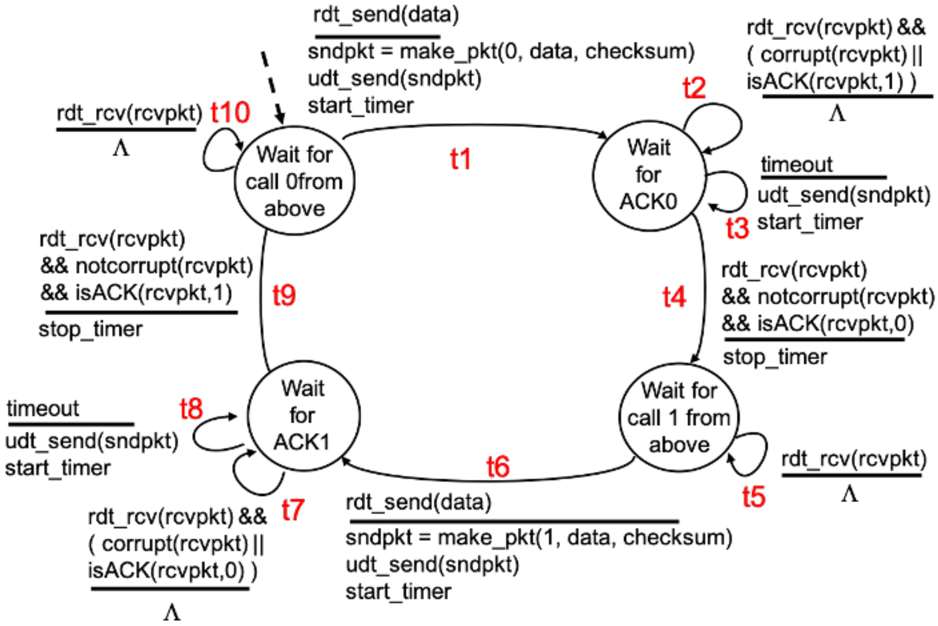
5. (ch33, 6pt) UDP is a best-effort transport layer service. Protocols such as DNS, SNMP, and DHCP use UDP to transmit data.
   (1) Recall what DNS is for and how it works. (0pt) Tell why DNS transmits via UDP. (2pt)
   (2) Look up what SNMP is for and how it works. (0pt) Tell why SNMP transmits via UDP. (2pt)
   (3) Look up what Skype is for and how it works. (0pt) Tell why Skype transmits via UDP. (2pt)

Sample Solution:

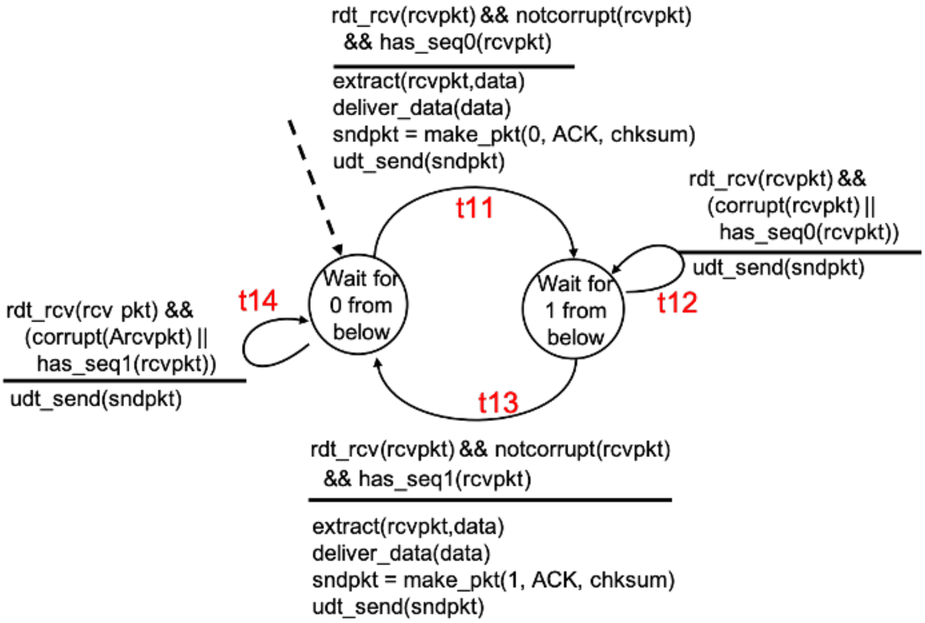
Whatever makes sense.

6. (ch34, 6pt) Below is the FSM of rdt 3.0. Tell the sequence of transitions for the following scenarios.

   <u>rdt 3.0 sender</u>



rdt 3.0 receiver



(1) Both the sender and receiver start from the initial state. The sender sends one packet. There is a bit error in the packet and no more bit errors or packet losses afterwards. (1pt)

(2) Continue from (1). The sender sends one more packet. The packet is lost but no more bit errors or packet losses afterwards. (1pt)

(3) Continue from (2). The sender sends one more packet. There is a bit error in the “ACK 0” packet coming back and no more bit errors or packet losses afterwards. (1pt)

(4) Continue from (3). The sender sends one more packet. The "ACK 1" packet coming back is lost and no more bit errors or packet losses afterwards. (1pt)
(5) In rdt 3.0 sender FSM, t2 does not do anything when the ACK packet received is corrupted or a duplicate. Do you like this? (0pt) Why or why not? (2pt)

Sample Solution:

(1) t1, t14, t2, t3, t11, t4
(2) t6, t8, t13, t9
(3) t1, t11, t2, t3, t12, t4
(4) t6, t13, t8, t14, t9
(5) take your pick and justify your own answers

7. (ch34, 6pt) Consider the FSM of SR. Let N = 10. Assume a pair of sender and receiver just establishes a connection and the packet sequence number starts from 1.

SR sender

t1: data from above:
- if next available seq # in window, send pkt

t2: timeout(n):
- resend pkt n, restart timer

t3: ACK(n) in [sendbase,sendbase+N-1]:
- mark pkt n as received
- if n is the smallest unACKed pkt, advance window base to next unACKed seq #

SR receiver

t4: pkt n in [rcvbase, rcvbase+N-1]
- send ACK(n)
- out-of-order: buffer
- in-order: deliver (also deliver buffered, in-order pkts), advance window to next not-yet-received pkt

t5: pkt n in [rcvbase-N,rcvbase-1]
- ACK(n)

t6: otherwise:
- ignore

(1) Suppose the sender gets the call from above to send 4 packet – DATA 1, DATA 2, DATA 3, and DATA 4. DATA # means it is a data packet carrying sequence number #. Tell the sequence of transitions triggered sending these 4 data packets. (1pt)
(2) Continue from (1). Suppose the 4 packets arrive at the receiver in this order – DATA 1, DATA 3, DATA 4, DATA 2. Tell the sequence of transitions triggered receiving these 4 data packets. (1pt)
(3) Continue from (2). Arrival of DATA 1, DATA 3, DATA 4, and DATA 2 will generate ack packets. Tell the sequence numbers of these ack packets. (1pt)
(4) Continue from (3). How many of the 4 data packets are buffered temporarily? (1pt)
(5) Continue from (4). Arrival of the ack packets at the sender will trigger a sequence of transitions. Tell the sequence of transitions triggered receiving the 4 ack packets. (1pt)
(6) Continue from (5). Tell the value of send_base (sendbase in the FSM) upon receiving each of the 4 ack packets. (1pt)

Sample Solution:

(1) t1 triggered 4 times

(2) t4 triggered 4 times

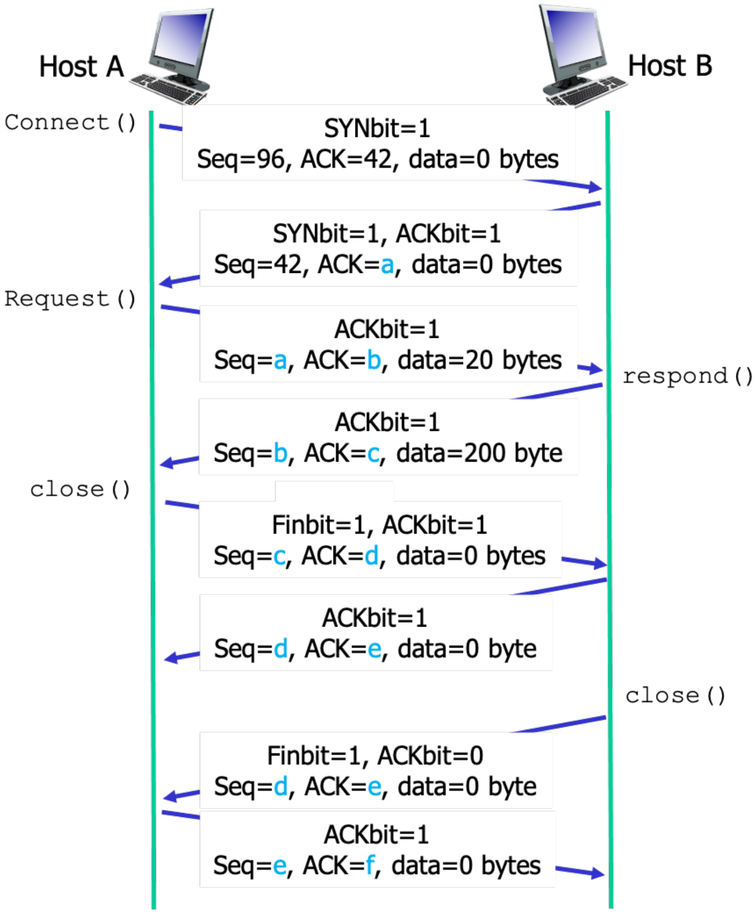
(3) 1, 3, 4, 2

(4) 2 (pkt3 and pkt4 are temporarily buffered until the receiver receives pkt2)

(5) t3 triggered 4 times

(6) 2 2 2 5

8. (ch35, 6pt) Depicted below is a part of packet exchange in a TCP connection between Host A and B. Host A sends a request. Host B sends back a response. After receiving the response, A initiates closing of the connection by sending a FIN packet. Not long, B closes the connection by sending a FIN packet as well.



(1) Tell the value of a. (1pt)
(2) Tell the value of b. (1pt)
(3) Tell the value of c. (1pt)
(4) Tell the value of d. (1pt)
(5) Tell the value of e. (1pt)
(6) Tell the value of f. (1pt)

Sample Solution:

(a, b, c, d, e, f) = (97, 43, 117, 243, 118, 244)

9. (ch35, 4pt) Recall the algorithm estimating the retransmission timeout – TimeoutInterval.
   (1) Why does the algorithm omit calculating the RTT of a data and ACK packet pair, i.e., getting a sampleRTT, for a data packet that has been retransmitted? (1pt)
   (2) Why does the algorithm calculate a smoothed average, i.e., estimatedRTT, instead of using the most recent sampleRTT? (1pt)
   (3) What does DevRTT estimate? (1pt)
   (4) The algorithm sets the retransmission TimeoutInterval to estimatedRTT+4*DevRTT. Why 4 times DevRTT, not 3 times or 5 times? (1pt)

Sample Solution:

(1) Not sure which data packet the ACK is acknowledging for.
(2) sampleRTT is the instantaneous RTT and it varies significantly from time to time. A smoothed average gives a longer-term, stable estimate.
(3) Variation of instantaneous RTTs to the estimatedRTT
(4) Empirically determined, heuristic, or something more elaborated -- if RTTs over a long time converges to a normal distribution, estimatedRTT (the avg) + 4*DevRTT (the variation) covers a very high % of RTTs (68–95–99.7 rule), therefore reducing the chance of premature timeout to a super low %.

10. (ch35, 3pt) TCP's rdt mechanism is an extension of GBN's. Tell which of the following is in TCP's rdt mechanism but not in GBN's. Grading Policy: -1pt per wrong choice.
    (a) One timer for a flight of packets transmitted
    (b) A specific mechanism to estimate the timeout interval
    (c) Retransmitting a flight of packets after a timeout event
    (d) Buffering out-of-order packets at the receiver
    (e) Sending delayed ACKs
    (f) Counting data by bytes

Sample Solution:

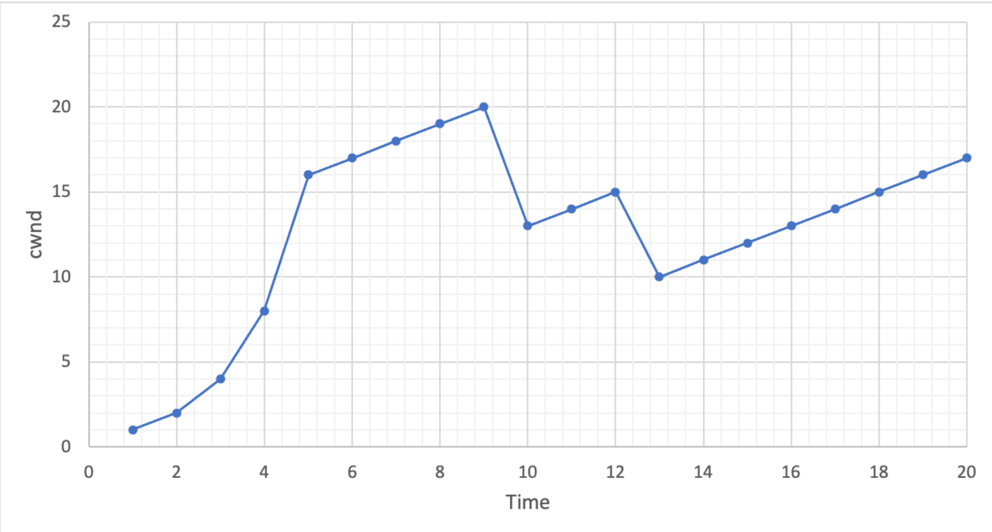
(b)(d)(e)(f)

11. (ch36, 3pt) There are two general approaches to congestion control – end to end and network assisted.
    (1) Describe how the end-to-end approach works. (1pt)
    (2) Describe how the network-assisted approach works. (1pt)
    (3) Which approach is taken by TCP (1pt)?

Sample Solution:

(1) Having the end hosts to detect packet loss or delay and inform the traffic source to set the sending rate accordingly
(2) Having routers to track the free buffer space and inform the traffic source to set the sending rate accordingly
(3) End-to-end approach

12. (ch37, 8pt) Drawn in the plot below is the progress of cwnd over time in a TCP connection.

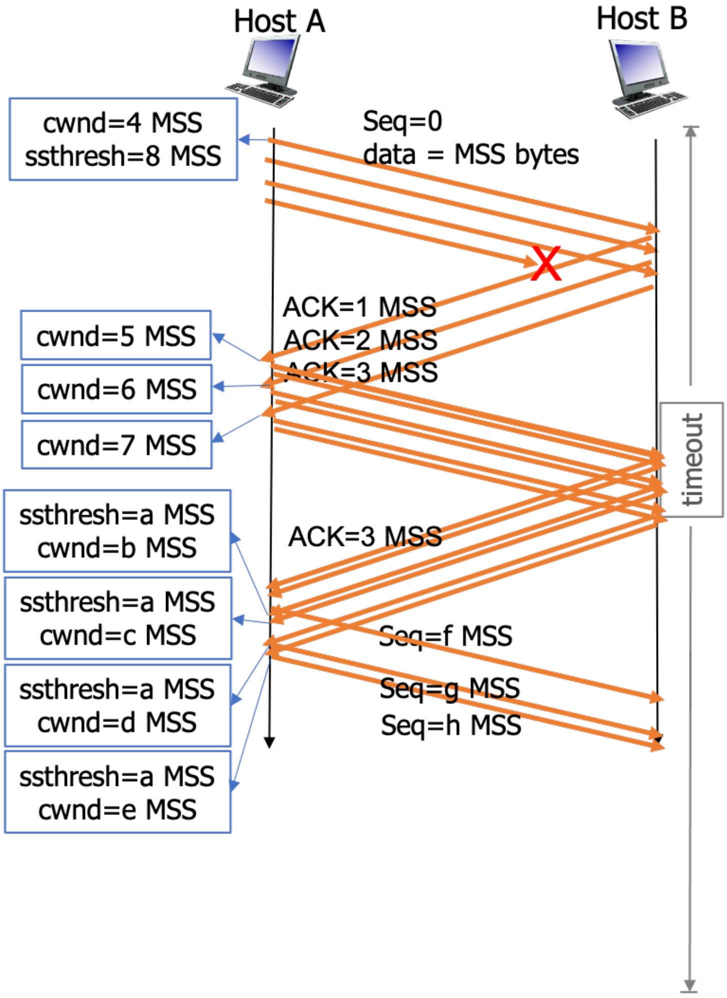


(1) Is this a TCP connection with fast recovery or not? (1pt)
(2) What is the state of the TCP connection during Time = 1-5? (1pt)
(3) What is the state of the TCP connection during Time = 5-9? (1pt)
(4) What is the event that triggers the cwnd reduction at Time = 10? (1pt)
(5) What is the ssthresh at Time = 10? (1pt)
(6) What is the state of the TCP connection during Time = 10-12? (1pt)
(7) What is the event that triggers the cwnd reduction at Time = 13? (1pt)
(8) What is the state of the TCP connection during Time = 13-20? (1pt)

Sample Solution:

(1) TCP with fast recovery.
(2) Slow start state
(3) Congestion avoidance state
(4) The sender has accumulated 3 duplicate ACKs
(5) 10
(6) Fast recovery state
(7) The sender receives a new ACK
(8) Congestion avoidance

13. (ch37, 8pt) The figure below illustrates a part of the data and ACK packet exchange in a TCP connection with fast recovery. Assume that (1) there are always data from above, (2) the packets are always MSS bytes large. If there remains a partial MSS in the cwnd, the sender does not generate a smaller packet to send. That is, if the cwnd = 2.5 MSS, the sender sends only 2 packets or allows 2 packets in flight.



(1) Tell the value of a. (1pt)
(2) Tell the value of b. (1pt)
(3) Tell the value of c. (1pt)
(4) Tell the value of d. (1pt)
(5) Tell the value of e. (1pt)
(6) Tell the value of f. (1pt)
(7) Tell the value of g. (1pt)
(8) Tell the value of h. (1pt)

Sample Solution:

(a, b, c, d, e, f, g, h)=(3.5, 6.5, 7.5, 8.5, 9.5, 3, 10, 11)

14. (ch37, 4pt) Recall the explicit congestion notification (ECN) option of TCP. The way it works is simple. The TCP receiver sets the ECE bit on in a packet to inform the sender there's a congestion in the network. The TCP sender sets the CWR bit on in another packet to inform the receiver the congestion window size has been reduced.

    A subtle detail that is neglected in the lecture is this – how can the TCP receiver at the transport layer see the ToS field in the network layer header when the network layer header will be removed before the packet can be passed up to the transport layer? To make this possible, a TCP receiver will need to be able to read the network layer header in addition to the transport layer header. Do you like the design? If yes, why? If not, why not? (4pt)

Sample Solution:

Take your pick and justify.

15. (PA, 15pt) Please go on the PA workstation and work under the exam2 directory for this problem set. **Create the exam2 directory** if you have not yet done so. **Move the unix-test.txt in the home directory to the exam2 directory** if you have not yet done so. Grading policy: pts for later problems will be given only when the formers are completed.

    Through this problem set, you will be implementing a file upload client that works without the need of sending the file size to the server. The server on port 12000 is a file upload server that responds a line – "200 OK for UPLOAD\n" – when receiving a command "UPLOAD\n". The server begins to receive content from your client until a line that contains only "EOF\n" arrives. When the "EOF\n" line is received, the server generates a file, called "whatever.txt" out of content received after "UPLOAD\n" and responds a line – "<file size> bytes received\n".

    (1) Develop exam2-p15-1.go such that it **connects to the server running on port 12000** and then closes the connection. (2pt)
    (2) Extend the code to exam2-p15-2.go such that it connects to the server running on port 12000, **sends "UPLOAD\n"** and then closes the connection. (2pt)
    (3) Extend the code to exam2-p15-3.go such that it connects to the server running on port 12000, sends "UPLOAD\n", **receives a line of message from the server, prints the message on the client screen**, and then closes the connection. (2pt)
    (4) Extend the code to exam2-p15-4.go such that it connects to the server running on port 12000, sends "UPLOAD\n", receives a line of message from the server, prints the message on the client screen. Then, **prompts the user for a filename, sends the file content to the server,** and then closes the connection. **Upload unix-test.txt for testing.** (3pt)
    (5) Extend the code to exam2-p15-5.go such that it connects to the server running on port 12000, sends "UPLOAD\n", receives a line of message from the server, prints the message on the client screen. Then, prompts the user for a filename, sends the file content to the server, **sends "\nEOF\n"**, and then closes the connection. Upload unix-test.txt for testing. (3pt)
    (6) Extend the code to exam2-p15-6.go such that it connects to the server running on port 12000, sends "UPLOAD\n", receives a line of message from the server, prints the message on the client screen. Then, prompts the user for a filename, sends the file content to the server, sends "\nEOF\n", **receives a line of message from the server, prints the message on the client screen** and then closes the connection. Upload unix-test.txt for testing. (3pt)

Sample Solution:

Whatever that works.

16. (PA, 15pt) Please go on the PA workstation and work under the exam2 directory for this problem set. Create the exam2 directory if you have not yet done so. Move the unix-test.txt in the home directory to the exam2 directory if you have not yet done so. Grading policy: pts for later problems will be given only when the formers are completed.

Through this problem set, you will be implementing a file upload server that works with the client in the previous problem set. **Please use your exam port # for the server to avoid conflicts with other students' servers.** Your server should respond a line – "200 OK for UPLOAD\n" – when receiving a command "UPLOAD\n". The server will continue to receive content from the client until a line that contains only "EOF\n" arrives. When the "EOF\n" line is received, your server generates a file, called "whatever.txt" out of all content received after "UPLOAD\n" and responds a line – "<file size> bytes received\n".

(1) Develop exam2-p16-1.go running on your exam port # such that it allows multiple clients such as your exam2-p15-1.go to **connect and then close**. (3pt)
(2) Develop exam2-p16-2.go such that it allows multiple clients such as your exam2-p15-3.go to connect. Your exam2-p16-2.go should **respond with "200 OK for UPLOAD\n" upon receiving "UPLOAD\n" from the client**. (3pt)
(3) Develop exam2-p16-3.go such that it allows multiple clients such as your exam2-p15-5.go to connect. Your exam2-p16-3.go should respond with "200 OK for UPLOAD\n" upon receiving "UPLOAD\n" from the client. Your exam2-p16-3.go will **continue to receive from the client until a line that is exactly –"EOF\n". When the "EOF\n" line is received, print the total number of bytes received on screen**. Upload unix-test.txt for testing. (4pt)
(4) Extend your exam2-p16-3.go to exam2-p16-4.go such that it **stores the file content received in a file called "whatever.txt"**. Upload unix-test.txt for testing. (3pt)
(5) Extend your exam2-p16-4.go to exam2-p16-5.go such that it allows multiple clients such as your exam2-p15-6.go to connect. Your exam2-p16-5.go **responds a line – "<file size> bytes received\n" after "whatever.txt" is generated**. Upload unix-test.txt for testing. (2pt)

Sample Solution:

Whatever that works.