Name_____   Student ID_____   Department/Year_____

2nd Examination

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

10:20-12:10 Wednesday
May 10, 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) 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 10 Mbps. (1pt)
   (2) Tell the resolution the viewer is watching when the measured available bandwidth is 2.3 Mbps. (1pt)
   (3) To enable the selection, the DASH client needs to know the available bandwidth. Suggest a way to measure the available bandwidth from the server to the client. (2pt)

Sample Solution:
   (1) 1440p
   (2) 480p
   (3) Whatever that works

2. (ch26, 6pt) Recall these two terms – buffer starvation and buffer overflow.
   (1) What is buffer starvation? (1pt)
   (2) How to prevent buffer starvation? (2pt)
   (3) What is buffer overflow? (1pt)
   (4) How to prevent buffer overflow? (2pt)

Sample Solution:
   In ch26, these terms mean the client-side buffer starving and overflowing.
   Describe the terms in your own words and justify for your view.

3. (ch26, 2pt) Instead of a large, mega server, a CDN is used to support a large-scale video service today. Tell which of the issues below are weaknesses of using a large, mega server. Grading policy: -1pt per wrong choice till 0pt left.
   (a) Single point of failure
   (b) Point of network congestion
   (c) Long path to distant clients
   (d) Multiple copies of the video sent over outgoing link

Sample Solution:

4. (ch26, 8pt) Between these two types of CDN – enter deep and bring home.
    (1) Which avoids network congestion better and why? (2pt)
    (2) Which allows shorter delay to distant clients and why? (2pt)
    (3) Which reduces more the chance of sending multiple copies of the video over the outgoing link and why? (2pt)
    (4) Which costs less and why? (2pt)

Sample Solution:
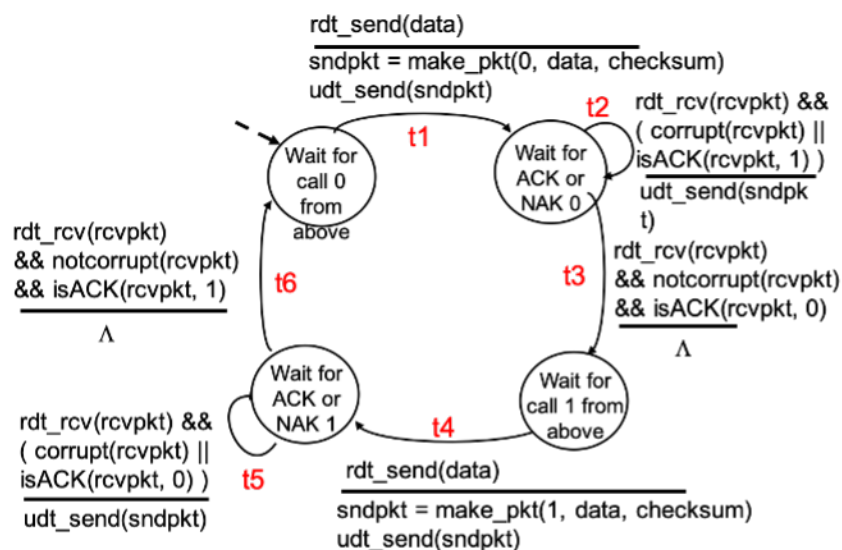   Anything that's genuinely from you and makes sense

5. (ch3, 6pt) For each of the functions below, tell whether it is implemented in (a) TCP, (b) UDP, (c) both, or (d) none.
    (1) bit error detection (1pt)
    (2) bit error recovery (1pt)
    (3) congestion control (1pt)
    (4) connection management (1pt)
    (5) logical communication between hosts (1pt)
    (6) breaking messages into segments (1pt)
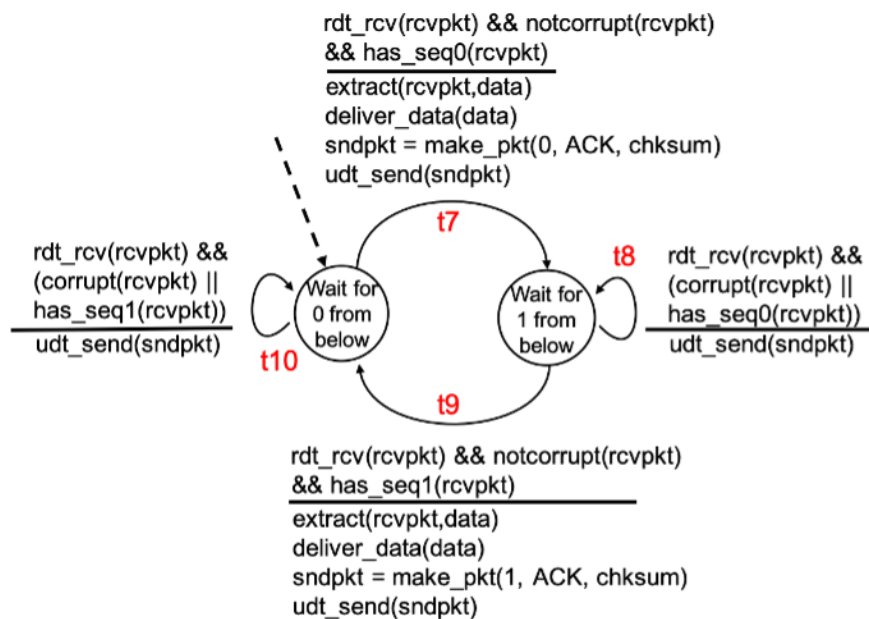
Sample Solution:
    (c, a, a, a, d, c)

6. (ch34, 6pt) Below is the FSM of rdt 2.2. Tell the sequence of transitions for the following scenarios.
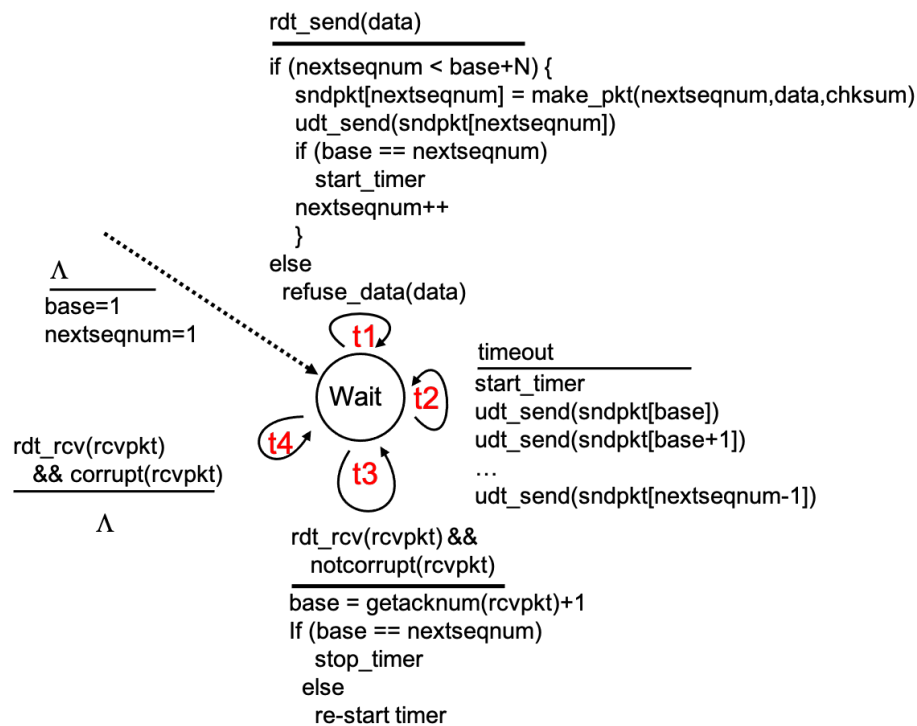   rdt 2.2 sender:

rdt 2.2 receiver:

rdt_rcv(rcvpkt) && notcorrupt(rcvpkt)
&& has_seq0(rcvpkt)
_____
extract(rcvpkt,data)
deliver_data(data)
sndpkt = make_pkt(0, ACK, chksum)
udt_send(sndpkt)

t7

t8

rdt_rcv(rcvpkt) &&
(corrupt(rcvpkt) ||
has_seq1(rcvpkt))
_____
udt_send(sndpkt)

rdt_rcv(rcvpkt) &&
(corrupt(rcvpkt) ||
has_seq0(rcvpkt))
_____
udt_send(sndpkt)

Wait for 0 from below

Wait for 1 from below

t10

t9

rdt_rcv(rcvpkt) && notcorrupt(rcvpkt)
&& has_seq1(rcvpkt)
_____
extract(rcvpkt,data)
deliver_data(data)
sndpkt = make_pkt(1, ACK, chksum)
udt_send(sndpkt)

(1) Both the sender and receiver start from the initial state. The sender sends one packet. There are no bit errors at all. (1pt)

(2) Continue from (1). The sender sends another packet. There are no bit errors at all. (1pt)

(3) Continue from (2). The sender sends one more packet. There is a bit error in the packet and no more bit errors afterwards. (1pt)

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

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

(6) Continue from (5). The sender sends one more packet. The packet is lost and no more bit errors or packet loss afterwards. (1pt)
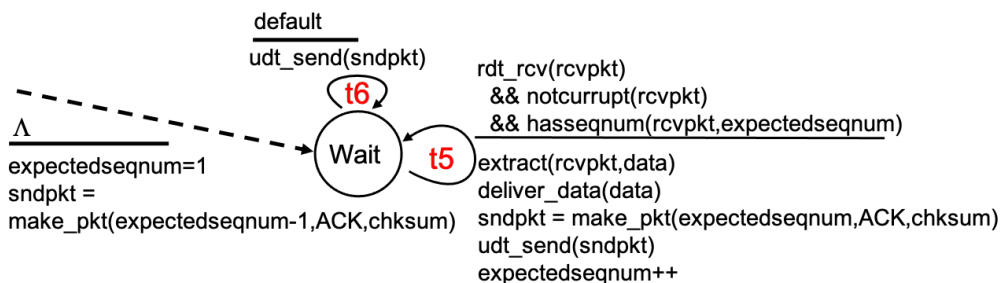
Sample Solution:

  (1) t1, t7, t3

  (2) t4, t9, t6

  (3) t1, t10, t2, t7, t3

  (4) t4, t9, t5, t10, t6

  (5) t4, t8, t5, t9, t6

  (6) t1… and the system hangs

7. (ch34, 6pt) Below is the FSM of GBN. Let N = 10. Assume that there are only packet losses and no bit errors in the underlying channel.

GBN sender

rdt_send(data)
‾‾‾‾‾‾‾‾‾‾‾‾‾‾
if (nextseqnum < base+N) {
    sndpkt[nextseqnum] = make_pkt(nextseqnum,data,chksum)
    udt_send(sndpkt[nextseqnum])
    if (base == nextseqnum)
        start_timer
    nextseqnum++
}
else
    refuse_data(data)

Λ
‾‾‾‾‾‾
base=1
nextseqnum=1

t1
t2
Wait

timeout
‾‾‾‾‾‾‾‾‾‾‾‾
start_timer
udt_send(sndpkt[base])
udt_send(sndpkt[base+1])
…
udt_send(sndpkt[nextseqnum-1])

rdt_rcv(rcvpkt)
    && corrupt(rcvpkt)
‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾
Λ

t4
t3

rdt_rcv(rcvpkt) &&
    notcorrupt(rcvpkt)
‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾
base = getacknum(rcvpkt)+1
If (base == nextseqnum)
    stop_timer
else
    re-start timer

GBN receiver

default
‾‾‾‾‾‾‾
udt_send(sndpkt)

t6
t5
Wait

Λ
‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾
expectedseqnum=1
sndpkt =
make_pkt(expectedseqnum-1,ACK,chksum)

rdt_rcv(rcvpkt)
    && notcurrupt(rcvpkt)
    && hasseqnum(rcvpkt,expectedseqnum)
‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾
extract(rcvpkt,data)
deliver_data(data)
sndpkt = make_pkt(expectedseqnum,ACK,chksum)
udt_send(sndpkt)
expectedseqnum++

(1) Suppose the sender gets the call from above to send 4 packet – pkt1, pkt2, pkt3 and pkt4. Tell the sequence of transitions triggered sending these 4 packets. (1pt)

(2) Continue from (1). Suppose only 3 packets arrive at the receiver and in this order – pkt1, pkt3 pkt4. Tell the sequence of transitions triggered receiving each of the 3 packets. (1pt)

(3) Continue from (2). Arrival of pkt1, pkt3, and pk4 will generate ack packets. Tell the sequence numbers of the ack packets generated. (1pt)

(4) Continue from (3). Arrival of the ack packets at the sender will trigger transitions at the sender side. Tell the transitions triggered by the ack packets. (1pt)

(5) Continue from (4). Tell the value of the send_base (base in the FSM) after receiving all 3 ack packets. (1pt)

(6) Continue from (4). Tell the number of times the timer is re-started after receiving all 3 ack packets. (1pt)

8. (ch35, 8pt) 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. Shortly after responding, B initiates closing of the connection by sending a FIN packet.
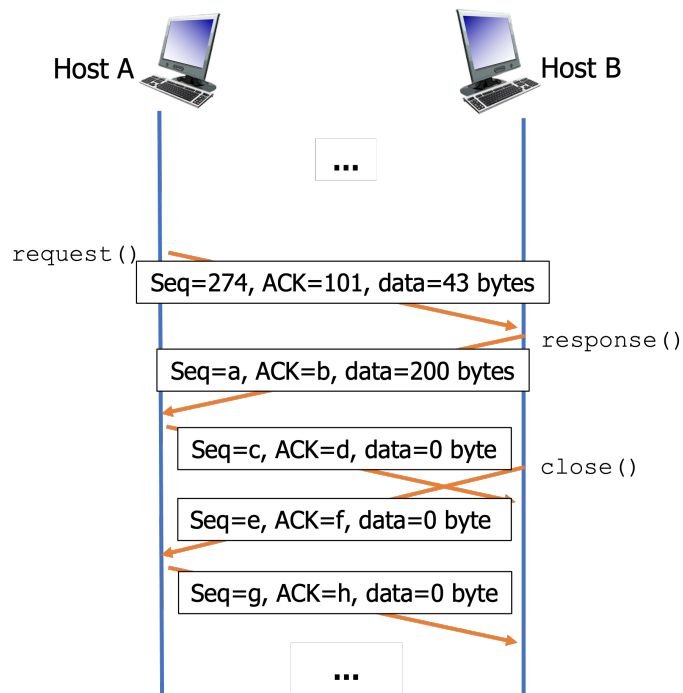
(1) Tell the value of c. (1pt)

(2) Tell the value of d. (1pt)

(3) Tell the value of g. (1pt)

(4) Tell the value of h. (1pt)

(5) Tell why the ACK sequence number of the FIN packet needs to be the FIN packet sequence number + 1. (2pt)

(6) Tell why the ACK sequence number of the SYN packet needs to be the SYN packet sequence number + 1. (2pt)

9. (ch35, 4pt) The delayed mechanism in TCP's receiver comes with advantages and disadvantages.
    (1) Tell one advantage of implementing the delayed ACK receiver. (2pt)
    (2) Tell one disadvantage of implementing the delayed ACK receiver. (2pt)

Sample Solution:
    (1) The number of ACK packets sent is reduced. I.e., The traffic volume (rcv to snd) is lower.
    (2) The time to send the next round of data packets may be longer. I.e., The time to complete the transfer is longer.

10. (ch36, 6pt) 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) Tell two advantages of taking the end-to-end approach. (2pt)
    (4) Tell two advantages of taking the network-assisted approach. (2pt)

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) Ease of deployment
        Ease of making changes (complexity at the edge)
        Lower router load
    (4) Precise estimation of available bandwidth
        Quick detection of congestion (right at the router where the congestion happens)
        Lower end-system load

11. (ch37, 6pt) TCP's congestion control mechanism defines not only how cwnd changes, but also how ssthresh changes. Consider the following possible changes to cwnd and ssthresh.
   (a) cwnd = 1MSS
    (b) cwnd = (1/2)cwnd

(c) cwnd = cwnd+1MSS

(d) cwnd = cwnd+MSS*(MSS/cwnd)

(e) cwnd = (1/2)cwnd+3MSS

(f) cwnd = (1/2)ssthresh

(g) cwnd = ssthresh

(h) ssthresh = 1MSS

(i) ssthresh = (1/2)cwnd

(j) ssthresh = cwnd+1MSS

(k) ssthresh = cwnd+MSS*(MSS/cwnd)

(l) ssthresh = (1/2)cwnd+3MSS

(m) ssthresh = (1/2)ssthresh

(n) ssthresh = cwnd

(1) Tell how ssthresh is changed when 3 duplicate ACKs are detected in a TCP connection with fast recovery. (1pt)

(2) Tell how cwnd is changed when 3 duplicate ACKs are detected in a TCP connection with fast recovery. (1pt)

(3) Tell how cwnd is changed when a duplicate ACK is received in the fast recovery state of a TCP connection. (1pt)

(4) Tell how cwnd is changed when a new ACK is received in the fast recovery state of a TCP connection. (1pt)

(5) Tell how ssthresh is changed when a timeout is detected in the fast recovery state of a TCP connection. (1pt)

(6) Tell how cwnd is changed when a timeout is detected in the fast recovery state of a TCP connection. (1pt)

Sample Solution:

i, e, c, g, i, a

12. (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 sender sends data in MSS packets if possible, and (2) if there remains a partial MSS in the cwnd, the sender generates a packet with a partial MSS size.
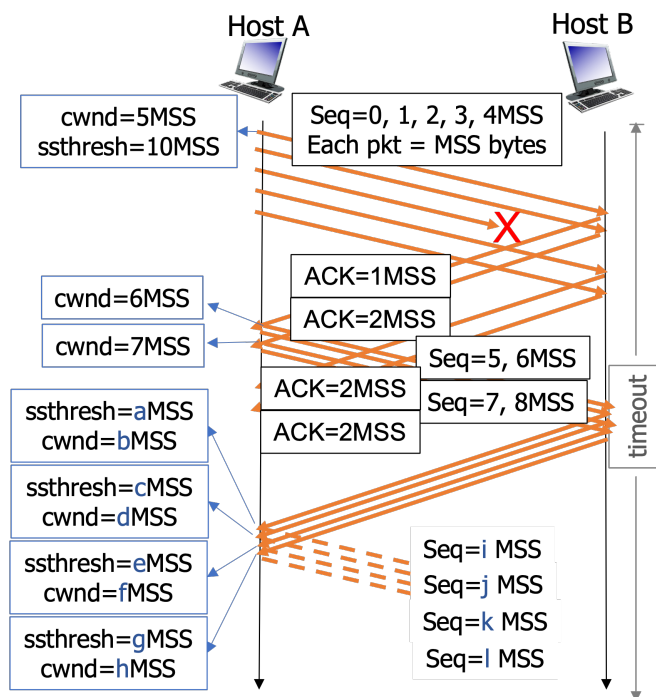
(1) Tell the value of b. (1pt)

(2) Tell the value of d. (1pt)

(3) Tell the value of f. (1pt)

(4) Tell the value of h. (1pt)

(5) Tell the value of i. (1pt)

(6) Tell the value of j. (1pt)

(7) Tell the value of k. (1pt)

(8) Tell the value of l. (1pt)

Sample Solution:

(b, d, f, h, i, j, k, l)=(6.5, 7.5, 8.5, 9.5, 2, 9, 9.5, 10.5)

13. (PA, 14pt) 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. Grading policy: pts for later problems will be given only when the former ones are completed.

(1) Develop exam2-p13-1.go such that it connects to the server running on port 12000 and then closes the connection. (1pt)

(2) Develop exam2-p13-2.go such that it connects to the server running on port 12000, sends "play\n" and then closes the connection. (1pt)

(3) Extend exam2-p13-2.go to exam2-p13-3.go such that it connects to the server running on port 12000, sends "play\n", receives a line of message from the server, prints the message on the screen, and then closes the connection. (2pt)

You should see the response from the server on port 12000. It says it is the "Price is Right!" game engine. In the game, the player guesses the price of a merchandize. The true price lies in the range 1-10000. Given a guess, the game engine responses with either "Too high", "Too low", or "Bingo". "Too high" means the guess is higher than the true price and "Too low" means the guess is lower. In either case, the game engine will continue to wait for the next guess until the guess is right. "Bingo" means the guessed price is right and the game ends at this point.

(4) Extend the code to exam2-p13-4.go such that it connects to the server running on port 12000, sends "play\n", receives a line of message from the server, prints the message on the screen. Then, prompts the user for a guess, sends the number in one line (+'\n' at the end of the number), receives a line of message from the server, prints the line of message on the screen, and then closes the connection. (3pt)

(5) To complete the game client, extend the code to exam2-p13-5.go such that it connects to the server running on port 12000, sends "play\n", receives a line of message from the server, prints the message on the screen. Then, repeatedly prompts the user for a guess, sends the number in one line (+'\n' at the end of the number), receives a line of message from the server and prints the line of message on the screen until the server response is "Bingo\n". (5pt)

(6) Tell the true price of the merchandize. (2pt)

Sample Solution:
    Whatever that works.

14. (PA, 16pt) 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. Grading policy: pts for later problems will be given only when the former ones are completed.

(1) Develop exam2-p14-1.go running on your exam port # such that it allows multiple clients such as your exam2-p13-1.go to connect and then close. (1pt)

(2) Develop exam2-p14-2.go such that it allows multiple clients such as your exam2-p13-2.go to connect, send "play\n", and then close. (2pt)

(3) Develop exam2-p14-3.go such that it allows multiple clients such as your exam2-p13-3.go to connect, send "play\n", receive a line of response from your exam2-p14-3.go saying "Welcome to The Price is Right! Let's guess the price of a 12-person tent.\n" and then close. (3pt)

(4) Develop exam2-p14-4.go such that it allows multiple clients such as your exam2-p13-4.go to connect, send "play\n", receive a short response from your exam2-p14-4.go saying "Welcome to The Price is Right! Let's guess the price of a 12-person tent.\n". Afterwards, set the true price to 9053, allow the client to send a number in 1-10000 followed by "\n", receive a short response from your exam2-p14-4.go saying the guess is "Too high\n" if the guess is higher than the true answer, "Too low\n" if the guess is lower, or "Bingo\n" if the guess is right. Then, close the connection. (4pt)

(5) Develop exam2-p14-5.go such that it allows multiple clients such as your exam2-p13-5.go to connect, send "play\n", receive a short response from your exam2-p14-5.go saying "Welcome to The Price is Right! Let's guess the price of a 12-person tent.\n". Afterwards, set the true price to 9053, allow repetition of – the client to send a number in 1-10000 followed by "\n", receive a short response from your exam2-p14-5.go saying the guess is "Too high\n" if the guess is higher than the true answer, "Too low\n" if the guess is lower, or "Bingo\n" if the guess is right. The repetition stops and the connection closes when the guess is right. (6pt)

Sample Solution:

Whatever that works.