Name_______________ Student ID____________________ Department/Year___________

3rd Examination

Introduction to Computer Networks (Online)
Class#: EE 4020, Class-ID: 901E31110
Spring 2025

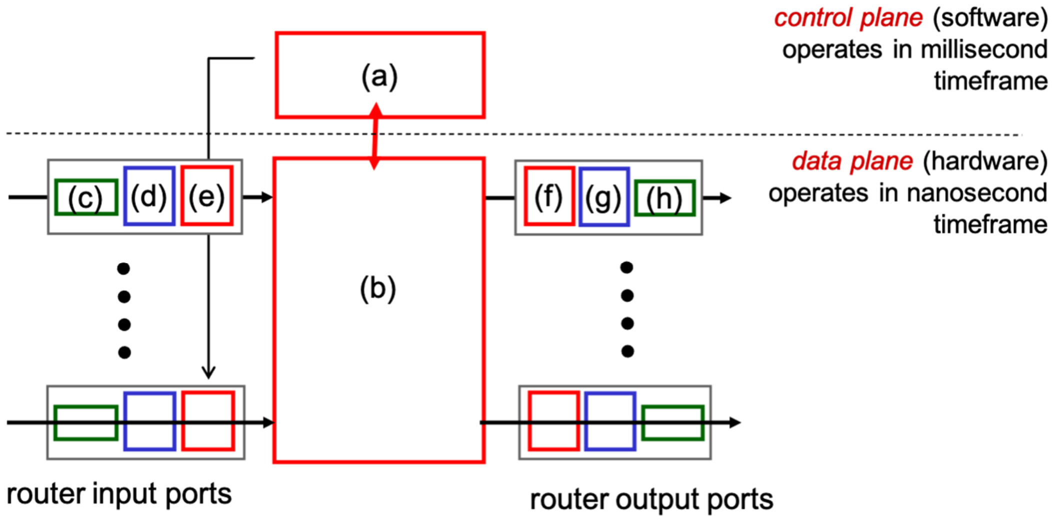
10:20-12:10 Wednesday
June 4, 2025

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.

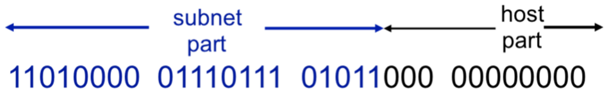
1. (ch41, 3pt) Recall the functions supported by the network layer. Tell which of the following statements are true. Grading policy: -1pt per error till 0pt left.
   (a) Forwarding is in the data plane.
   (b) Routing is in the data plane.
   (c) Destination-based forwarding is the traditional way of forwarding packets.
   (d) Generalized forwarding is the traditional way of forwarding packets.
   (e) Distributed control is the traditional way of routing.
   (f) Centralized control is the traditional way of routing.
   (g) Connection-oriented service is the service model of the Internet network layer.
   (h) Connectionless service is the service model of the Internet network layer.

2. (ch42, 5pt) Shown below is the high-level view of a generic router. See if you can tell the component implementing each of the router functions. Grading policy: -1pt per error till 0pt left.



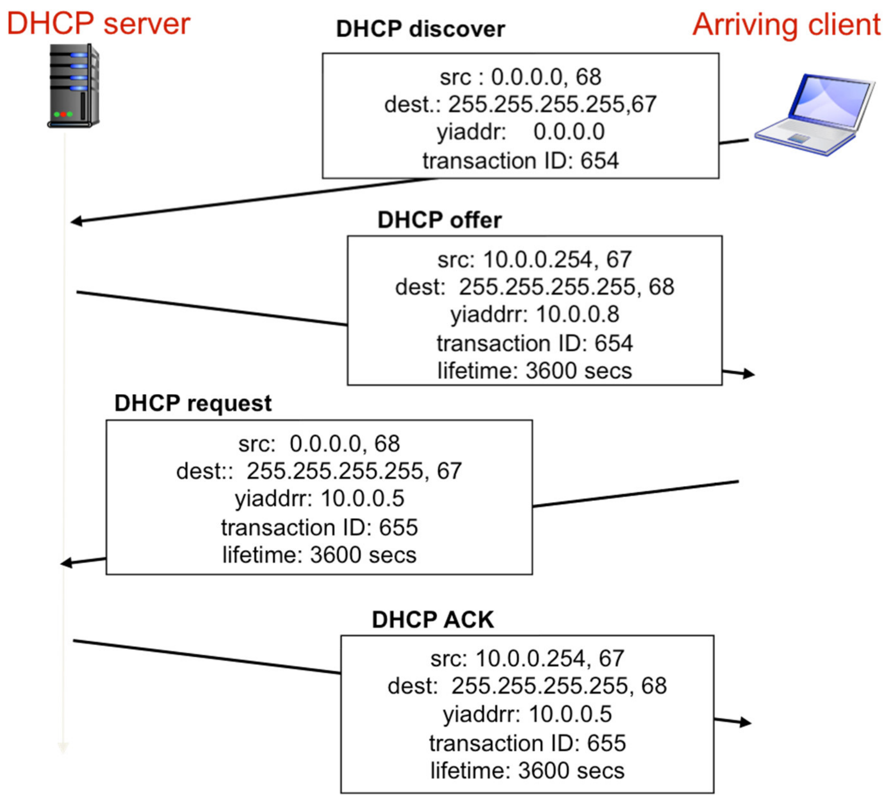
   (1) Which component(s) implements the route computation function (1pt)?
   (2) Which component(s) implements the packet switching function (1pt)?
   (3) Which component(s) implements the forwarding table lookup function (1pt)?
   (4) Which component(s) implements the dropping policy (1pt)?
   (5) Which component(s) implements the scheduling policy (1pt)?

3. (ch42, 3pt) In CIDR (Classless InterDomain Routing), the subnet part of the IP address can be in arbitrary length. Write the address of the following subnet in a.b.c.d/x format. Grading policy: -1pt per error till 0pt left.



4. (ch42, 2pt) Tell which of the followings are benefits of longest prefix matching. Grading policy: -1pt per error till 0pt left.
   (a) Saving the forwarding table space.
   (b) Speeding up the forwarding table lookup process.
   (c) Reducing the amount of the route advertisements.
   (d) Allowing division of an address block to multiple sub-blocks within an organization.

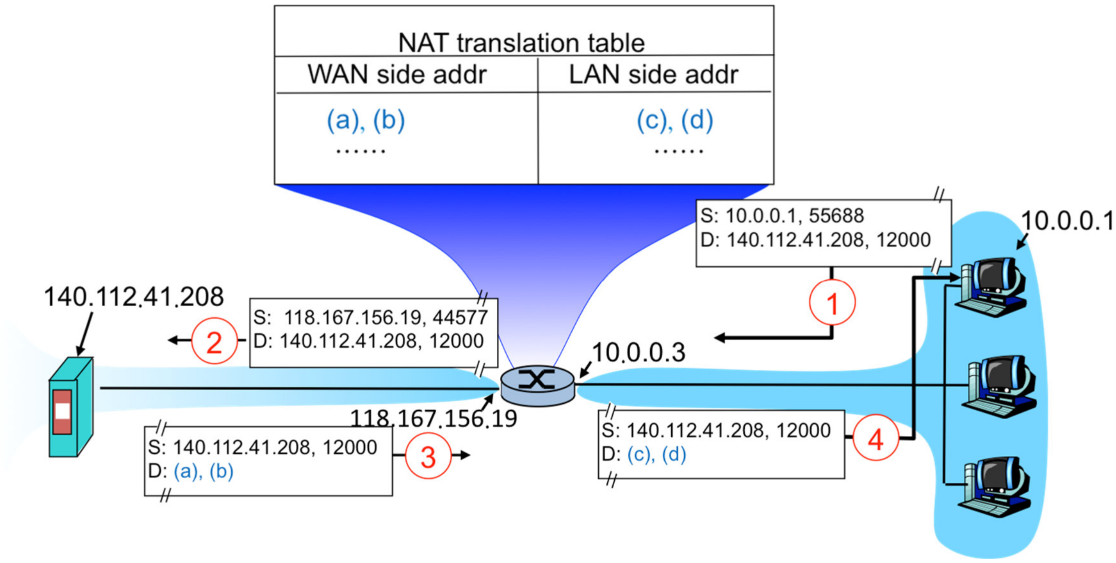
5. (ch43, 6pt) Check out the following DHCP message exchange. See if you can answer the related questions below.



   (1) Tell the IP address the DHCP server offers initially. (1pt)
   (2) Tell the IP address of the arriving client eventually. (1pt)
   (3) Which layer protocol is DHCP implemented? (1pt) If it is not implemented at the network layer, why? (1pt)
   (4) Does DHCP send messages over TCP or UDP? (1pt) And why? (1pt)

6. (ch43, 4pt) Most homes are allowed only one IP address for Internet access. To allow multiple smart devices accessing the Internet simultaneously at home, a common practice is to run NAT at the home gateway. Illustrated below is a client behind the home gateway trying to access a secure Web server on the Internet.

   A client at 10.0.0.1 sends an HTTPS request through the home gateway at IP address 118.167.156.19 to the Web server at IP address 140.112.41.208 for a web page. As the HTTPS request message goes through the home gateway, a WAN-LAN address mapping is added to the NAT translation table. Tell (a) the WAN side IP address, (b) the WAN side port number, (c) the LAN side IP address, and (d) the LAN side port number of the client.



   (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)

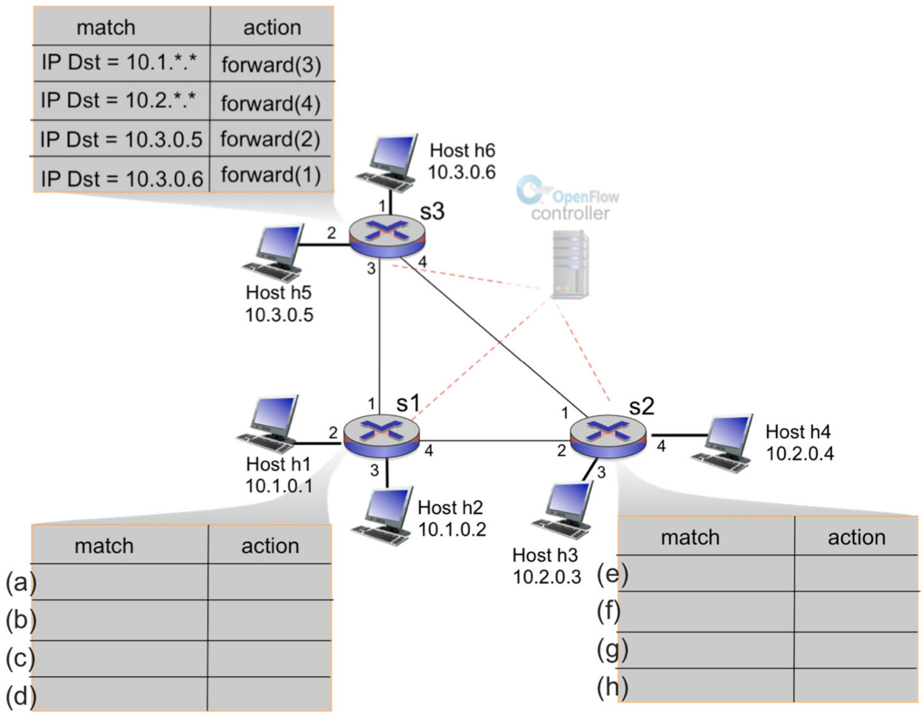
7. (ch43, 6pt) Recall how fragmentation and reassembly is done in IPv4. When a 4000-byte IPv4 packet (ID=x) runs into a link with MTU = 1500 bytes. The network layer protocol will break the original IPv4 packet into 3 fragments as follows.

   (length, ID, flag, offset) = (1500, x, 1, 0)
   (1500, x, 1, 185)
   (1040, x, 0, 370)

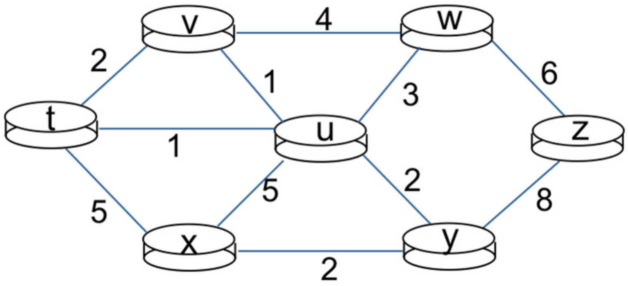
   Suppose that the 4000-byte IPv4 packet (ID=x) runs into a link with MTU = 1140 bytes instead. The network layer protocol will break the packet into 4 fragments as below. Tell the value of a, b, c, d, e, and f.

(length, ID, flag, offset) = (1140, x, 1, 0)
(1140, x, 1, a)
(1140, x, 1, b)
(c, d, e, f)

8. (ch44, 8pt) The following SDN-based network consists of 3 subnets. Each subnet consists of 2 hosts and 1 router. Let's assume the link cost between each router pair and each router-host pair is 1. The network admin would like to configure the 3 routers such that each of the 6 hosts sends traffic in the shortest distance to every other host. In the figure below, one sees how the network admin configures router s3. Follow the style and fill row (a)-(h). Grading Policy: -1pt per wrong or missing entry.

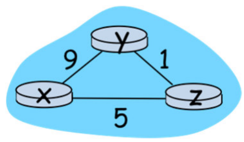


9. (ch52, 10pt) Using the Dijkstra algorithm, one can compute the shortest path from a source node to all other nodes in the network. Consider the 7-node network illustrated below. Generate the table indicating the steps deriving the N', cost D(*), and previous hop p(*) from node t to all other nodes. Note that when the path costs are equal, add nodes to the traversed set N' in alphabetic order. Fill the derivation table till Dijkstra algorithm terminates. Type 'inf' to represent infinity on the exam form. Grading policy: 2pts for the first 4 rows and 1pt for the last 2 rows (the 7th row is trivial).

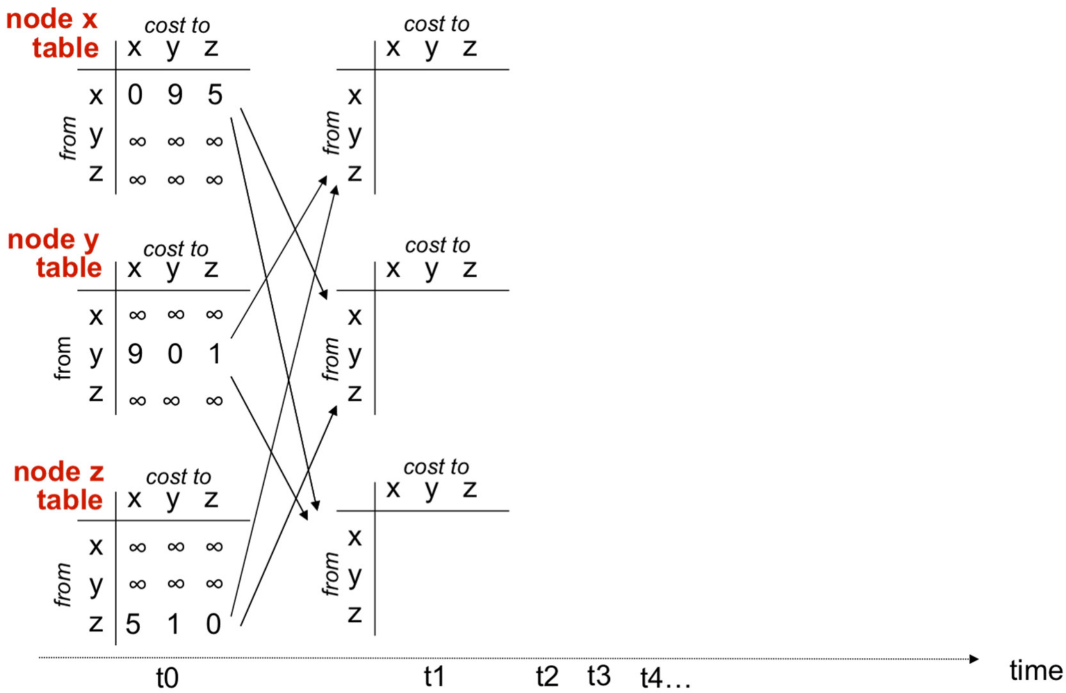


	N'	D(u),p(u)	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
Row 1	t						
Row 2							
Row 3							
Row 4							
Row 5							
Row 6							

10. (ch52, 6pt) Consider a simple network below. Develop the full DV tables at node x, y, z.

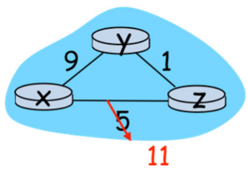


Please follow the derivation style as shown in the lecture. Assume the 3 nodes synchronously receive, compute, and send the DVs if there's any change. The initialization step at t0 is illustrated below. The first iteration happens at t1, the 2nd iteration at t2, and so on so forth.



(1) Show the Bellman-Ford computation to derive Dx at node x at t1. Grading policy: 0pt without the full derivation. (3pt)
(2) Show the Bellman-Ford computation to derive Dy at node y at t1. Grading policy: 0pt without the full derivation. (3pt)

11. (ch52, 11pt) Continue from Problem Set 10. Let's consider the following change to the network. Develop the full DV tables at node x, y, z.



Again, use the derivation style shown in the lecture. Assume the 3 nodes synchronously receive, compute, and send the DVs if there's any change. The tables before the change are those shown at t0. The first iteration at t1 is triggered by the link cost change, the 2nd iteration at t2 is triggered by the subsequent DV changes, and so on so forth. Assume also that the poisoned reverse mechanism is disabled.

node x table

t0:

from \ cost to	x	y	z
x	0	6	5
y	6	0	1
z	5	1	0

→ t1:

from \ cost to	x	y	z
x			
y			
z			

node y table

t0:

from \ cost to	x	y	z
x	0	6	5
y	6	0	1
z	5	1	0

node z table

t0:

from \ cost to	x	y	z
x	0	6	5
y	6	0	1
z	5	1	0

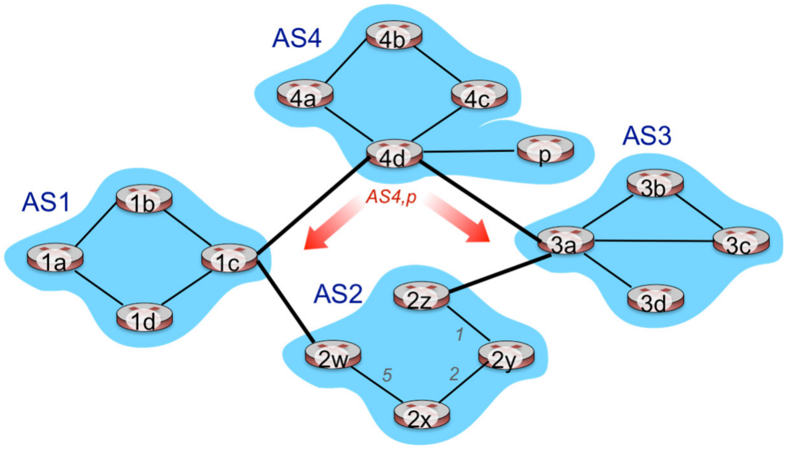
→ t1:

from \ cost to	x	y	z
x			
y			
z			

time: t0 t1 t2 t3 t4…

(1) Tell Dx at node x at t1. (3pt)
(2) Tell Dz at node z at t1. (2pt)
(3) Tell Dy at node y at t2. (2pt)
(4) Tell Dz at node z at t3. (2pt)
(5) Tell Dy at node y at t4. (1pt)
(6) Tell Dz at node z at t5. (1pt)

12. (ch53, 6pt) Consider the network below where BGP runs inter-AS and LS runs intra-AS. Destination subnet p in AS4 starts by announcing its existence. AS4's gateway router 4d then advertises the AS path "AS4, p" to 1c and 3a, the gateway routers to neighboring ASes.



(1) Suppose there are no import nor export restrictions in AS3. What would be the AS path 3a advertise to 2z? (1pt)
(2) Suppose there are no import nor export restrictions in AS1. What would be the AS path 1c advertise to 2w? (1pt)
(3) If there're no import restrictions in AS2, what would the entry be for destination p in 2x's forwarding table, and why? (2pt)
(4) If there's an import restriction in AS2 preventing 2z from propagating the received AS path further, what would the entry be for destination p in 2x's forwarding table, and why? (2pt)

13. (PA, 15pt) Please go on the PA workstation and work under the exam3 subdirectory for this problem set. Create the exam3 subdirectory if you haven't done so. Move the server-test.html file in your exam account's home directory to exam3 subdirectory.

    (1) Develop exam3-p13-1.go such that it works as a Web server – accepting HTTP requests for files and returning the files requested in the HTTP responses. Assume the files requested are always found. Use server-test.html for testing (also the test file for PA7-PA9). (3pt)
    (2) Develop exam3-p13-2.go such that it works as a Web server – accepting HTTP requests for files, returning the files requested in the HTTP responses if the files are found, else returning "404 Not Found". (4pt)
    (3) Move the server.key and server.cer file in your exam account's home directory to the exam3 subdirectory. (1pt)
    (4) Develop exam3-p13-3.go such that it extends exam3-p13-2.go to allow HTTPS requests and responses. Use the server.key and server.cer for the secure Web server. (3pt)
    (5) Develop exam3-p13-4.go such that it extends exam3-p13-3.go and the Web server returns "Hola~" when the client requests with this URL – "https://<server IP>:<port#>/hola". (4pt)

14. (PA, 15pt) Please go on the PA workstation and work under the exam3 subdirectory for this problem set. Create the exam3 subdirectory if you haven't done so. Move the client.key and client.cer file in your exam account's home directory to exam3 subdirectory.

    One can implement a pair of secure client and server to exchange encrypted messages without HTTPS. You will work towards a secure "This Number is Closer!" game client much like how you were asked to do in exam2. Your secure client will interact with the secure "This Number is Closer!" game server running on port 12000. Note that the server.key and server.cer used to configure the secure server are paired with the client.key and client.cer now in the exam3 subdirectory. Therefore, do use the client.key and client.cer to configure your client. Otherwise, your client and the server won't connect.

    A secure client example:

```
package main

import "crypto/tls"

func check(e error) {
   if e != nil {
      panic(e)
   }
}

func main() {
   cert, err := tls.LoadX509KeyPair("client.cer", "client.key")
   check(err)
   //skip checking the certificate
   config := tls.Config{Certificates: []tls.Certificate{cert}, \
   InsecureSkipVerify: true}
   conn, _ := tls.Dial("tcp", ":12000", &config)
   defer conn.Close()
}
```

    (1) Copy the secure client example above into exam3-p14-1.go. Make sure that it connects to the secure server on port 12000 and then closes the connection. (3pt)
    (2) Extend exam3-p14-1.go to exam3-p14-2.go such that it connects to the secure server

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. (4pt)

You should see the response from the server on port 12000. It says it is the secure version of the “This Number is Closer!” game engine. In this game, a player is asked to figure out the true value (in 1-100) the game engine has in mind. In each round, the player offers two numbers (in 1-100). The game engine compares the distance of the two numbers to the true value and tells which one is closer or whether they are equally close. The game engine responds with “Bingo” when both the numbers are identical to the true value.

(3) Extend exam3-p14-2.go to exam3-p14-3.go such that it connects to the secure server 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 the 1st numbers, sends the number in one line (+’\n’ at the end of the number), prompts the user for the 2nd number, sends the number in one line (+’\n’ at the end of the number), and then closes the connection. (4pt)

(4) To complete the secure game client, extend exam3-p14-3.go to exam3-p14-4.go such that it connects to the secure server 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 the 1st numbers, sends the number in one line (+’\n’ at the end of the number), prompts the user for the 2nd number, 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. (4pt)