Name_______________ Student ID____________________ Department/Year___________

3rd Examination

Introduction to Computer Networks (Online)
Class#: EE 4020, Class-ID: 901E31110
Spring 2024

10:20-12:10 Wednesday
June 5, 2024

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, 4pt) 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) Buffering happens at the input queue.
   (f) Buffering happens at the output queue.

Sample Solution:

(a)(c)(e)(f)

2. (ch42, 3pt) Consider a traditional router containing a forwarding table as follows.

Destination Prefix	Interface
11001000 00010111 1100	(a)
11001000 00010111 10101	(b)
11001000 00010111 1010	(c)
11001000 00010111 101011	(d)
default	(e)

Now 3 packets come along with the following destination addresses. Based on the longest match principle, which interfaces will the 3 packets be forwarded to?

(1) 11001000 00010111 **11000000** 10110000 (1pt)
(2) 11001000 00010111 **11100011** 10110000 (1pt)
(3) 11001000 00010111 **10101001** 10110000 (1pt)

Sample Solution:

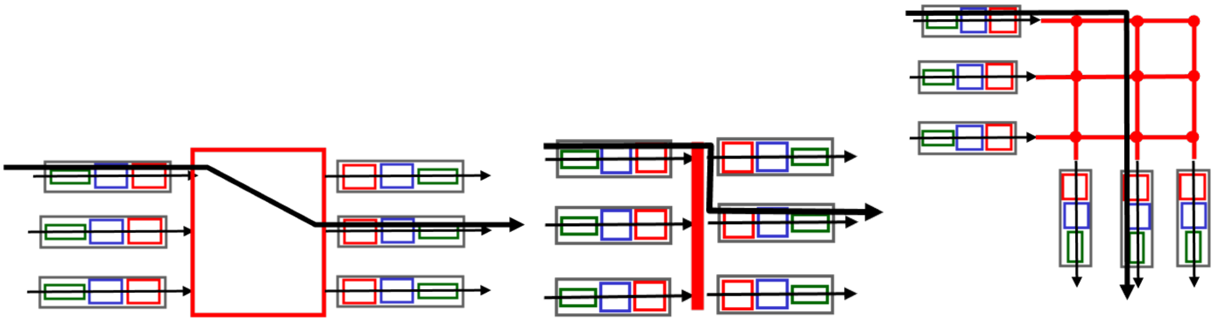
(a)(e)(b)

3. (ch42, 3pt) 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.

Sample Solution:

(a)(b)(c)(d) or (a)(b)(c)

4. (ch42, 7pt) Below are 3 forms of switching fabric in a router.



(a) (b) (c)

(1) Which one is the memory switching fabric? (1pt)
(2) Which one is the bus switching fabric? (1pt)
(3) Which one is the crossbar switching fabric? (1pt)
(4) Which one is the fastest and why? (2pt)
(5) Which one is the slowest and why? (2pt)

Sample Solution:

(1)-(3): (a)(b)(c)

(4): Crossbar. Potential of parallelism (a way to mitigate contention)

(5): Memory. Read/write to memory slow

5. (ch43, 3pt) Which of the following information does a DHCP server provide? Grading policy: -1pt per error till 0pt left.
   (a) IP address
   (b) Network mask
   (c) DNS server IP address
   (d) Lease time
   (e) First hop router IP address

Sample Solution:

(a)-(e) or (a)-(c)(e)

6. (ch43, 3pt) DHCP is a protocol for a host to automatically acquire an IP address, which is a network layer function. It is however implemented in the application layer and the DHCP messages are sent through UDP.
   (1) There is another protocol we have talked about in Chapter 2 that returns the IP address from a machine's hostname. It is implemented in the application layer and the protocol's messages are sent through UDP as well. What protocol is that? (1pt)
   (2) Why do DHCP and your answer for (1) implement network layer functions in the application layer? (1pt)

(3) Why do DHCP and your answer for (1) send the messages through UDP? (1pt)

Sample Solution:

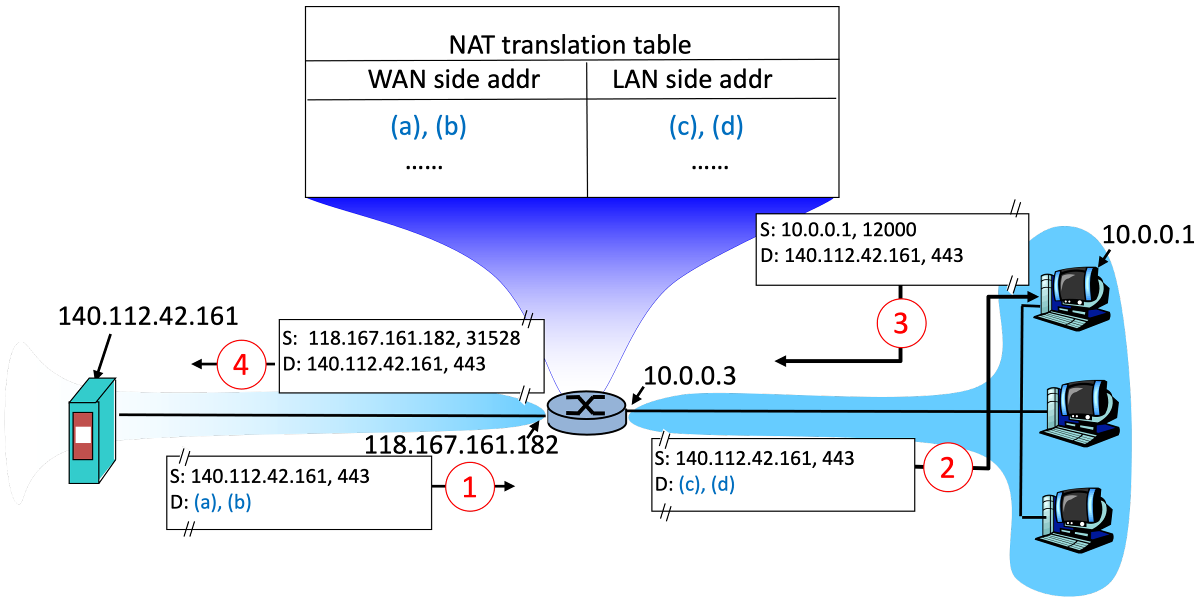
(1) DNS

(2) To put complexity at the edge (easy to make changes, less load to the routers, etc)

(3) To get the IP address asap (without the need to setup a TCP connection); In DHCP's case, no IP address to setup a TCP connection at the first place.

7. (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 118.167.161.182 to the Web server at IP address 140.112.42.161 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)

Sample Solution:

(1) 118.167.161.182

(2) 31528

(3) 10.0.0.1

(4) 12000

8. (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 = 1100 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) = (1100, x, 1, 0)
   (1100, x, 1, a)
   (1100, x, 1, b)
   (c, d, e, f)

Sample Solution:

(a, b, c, d, e, f) = (135, 270, 760, x, 0, 405)

9. (ch44, 6pt) Below is the flow table entry format.

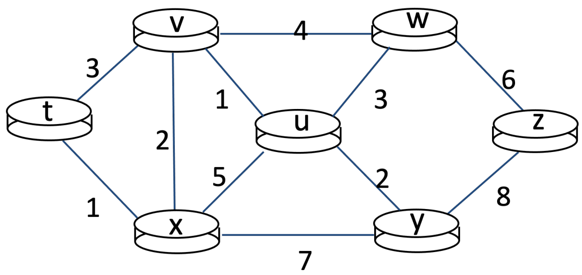
Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action	Priority

   Configure the entry(s) to achieve the functions below.
   (1) Traditional forwarding: forward packets going to subnet 192.168.1.0/24 via network interface 1. (2pt)
   (2) Firewall: For packets going to subnet 192.168.1.0/24, if they are going to the ssh server in the subnet, then forward via network interface 1, else drop. (4pt)

Sample Solution:

(1) Matching rule: IP Dst = 192.168.1.*
    Action: forward(1)
(2) Matching rule: IP Dst = 192.168.1.* and TCP dport==22
    Action: forward(1)
    Matching rule: IP Dst = 192.168.1.* and TCP dport!=22
    Action: drop

10. (ch52, 8pt) 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 initialization phase and 1pt per iteration in the looping phase.

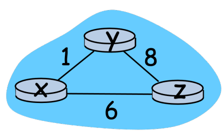


N’	D(u),p(u)	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
t						

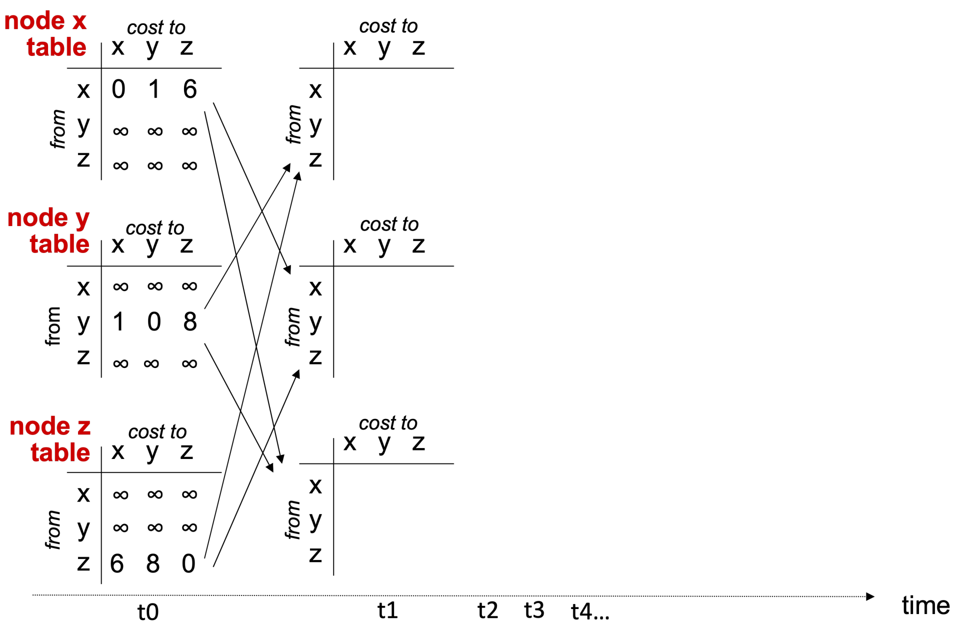
Sample Solution:

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)
t	inf	(3,t)	inf	(1,t)	inf	inf
tx	(6,x)	(3,t)	inf	--	(8,x)	inf
txv	(4,v)	--	(7,v)	--	(8,x)	inf
txvu	--	--	(7,v)	--	(6,u)	inf
txvuy	--	--	(7,v)	--	--	(14,y)
txvuyw	--	--	--	--	--	(13,w)
txvuywz	--	--	--	--	--	--

11. (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 Dz at node z at t1. Grading policy: 0pt without the full derivation. (3pt)

Sample Solution:

(1) Dx at node x at t1 = (0, 1, 6)

dx(y) = min{c(x,y)+dy(y), c(x,z)+dz(y)} = min{1+0, 6+8}=1

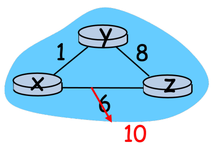
dx(z) = min{c(x,y)+dy(z), c(x,z)+dz(z)} = min{1+8, 6+0}=6

(2) Dz at node z at t1 = (6, 7, 0)

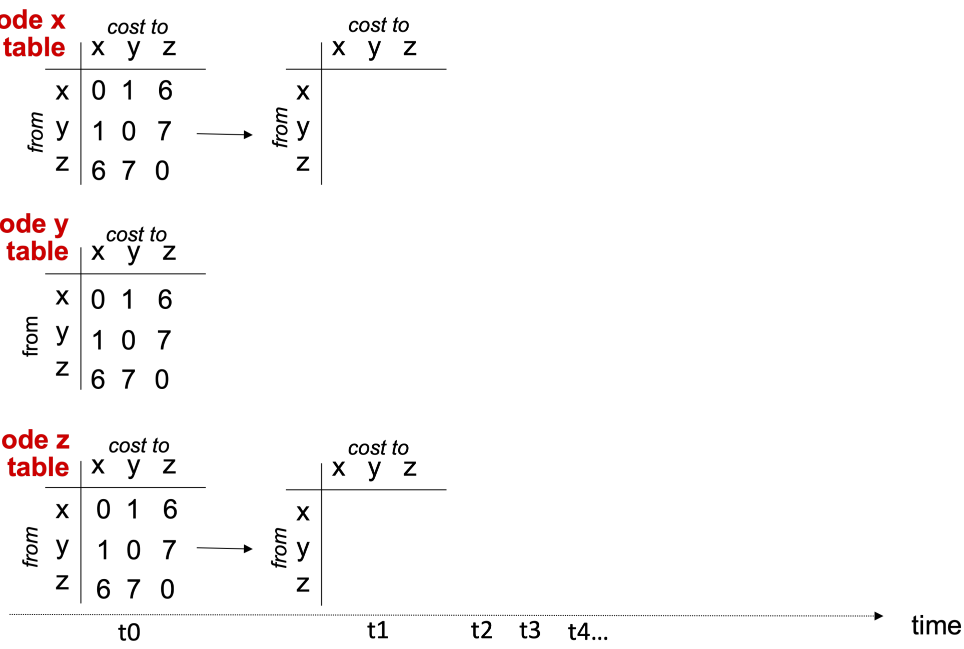
dz(x) = min{c(z,x)+dx(x), c(z,y)+dy(x)} =min{6+0, 8+1}=6

dz(y) = min{c(z,x)+dx(y), c(z,y)+dy(y)} =min{6+1, 8+0}=7

12. (ch52, 9pt) Continue from Problem Set 11. 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.



(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 Dx at node x at t3. (2pt)

Sample Solution:

(1) Dx at node x at t1 = (0, 1, 8)
(2) Dz at node z at t1 = (9, 8, 0)
(3) Dy at node y at t2 = (1, 0, 8)
(4) Dx at node x at t3 = (0, 1, 9)

13. (ch62, 8pt) Recall how CRC (cyclic redundancy check) works and assume G = 111. Grading policy: 0pt without detailed derivation.
(1) Find the R for D = 101010. Upload your answer with derivation. (2pt)
(2) Suppose <D, R> arriving at the receiver is 10101011. Would CRC find the sequence correct or not? Upload your answer with derivation. (2pt)
(3) Suppose <D, R> arriving at the receiver is 10101111. Would CRC find the sequence correct or not? Upload your answer with derivation. (2pt)
(4) Can CRC catch all 3-bit burst errors and why? (2pt)

Sample Solution:

(1) 00
(2) Yes (CRC says incorrect with a 2-bit burst error)
(3) No (CRC says correct with a 3-bit burst error)
(4) No. The theory says CRC can detect all burst errors less than 3 bits. Or by example, 10101111 contains a 3-bit burst error that is missed by CRC (according to 13-3).

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 server-test.html file in your exam account's home directory to exam3 subdirectory.

    (1) Develop exam3-p14-1.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-p14-2.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 "Hmm.. not here.". (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-p14-3.go such that it extends exam3-p14-2.go to allow HTTPS requests and responses. Use the server.key and server.cer for the secure Web server. (3pt)
    (5) Develop exam3-p14-4.go such that it extends exam3-p14-3.go and the Web server returns "Hi~" when the client requests with this URL – "https://<server IP>:<port#>/hello". (4pt)

Sample Solution:

Whatever that works.

15. (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 "Hangman Not Quite!" game client much like how you were asked to do in exam2. Your secure client will interact with the secure "Hangman Not Quite!" 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-p15-1.go. Make sure that it connects to the secure server on port 12000 and then closes the connection. (3pt)
    (2) Extend exam3-p15-1.go to exam3-p15-2.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 “Hangman Not Quite!” game engine. In this simplified Hangman game, a player is asked to figure out an English word the game engine has in mind. Given a character, the game engine responses with a string showing where the character appears in the word.

(3) Extend exam3-p15-2.go to exam3-p15-3.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 a guess, sends the character in one line (+’\n’ at the end of the character), and then closes the connection. (4pt)

(4) To complete the secure game client, extend exam3-p15-3.go to exam3-p15-4.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 a guess, sends the character in one line (+’\n’ at the end of the character), receives a line of message from the server, prints the line of message on the screen, and then closes the connection. (4pt)

Sample Solution:

Whatever that works.