Name_____ Student ID_____ Department/Year_____

Final Examination

Introduction to Computer Networks (Online)
Class#: EE 4020, Class-ID: 901E31110
Fall 2020

10:20-12:10 Thursday
June 18, 2020

Cautions

1. There are in total 100 points to earn. You have 100 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. (Golang) Consider the following server program: server-final-1.go. Execute the server first and then use a Web client (curl, Chrome, or Firefox) to request the server with this URL: http://<IP address>:<port #>/whatever.html.

   server-final-1.go

```
package main

import "fmt"
import "bufio"
import "net"
import "net/http"

func main() {
    ln, _ := net.Listen("tcp", ":<your port#>")
    defer ln.Close()
    conn, _ := ln.Accept()
    defer conn.Close()

    reader := bufio.NewReader(conn)
    req, _ := http.ReadRequest(reader)

    fmt.Printf("Method: %s\n", req.Method)
    fmt.Printf("URI: %s\n", req.RequestURI)
}
```

   (1) Tell the output on screen of the terminal running server-final-1.go. (1%)
   (2) Tell where in the Golang source code (the file name and line #) the `Method` field of the `Request` `struct` is defined. (1%)
   (3) Tell where in the Golang source code (the file name and line #) the `RequestURI` field of the `Request` `struct` is defined. (2%)
   (4) What does `RequestURI` store? (2%)

Sample Solution:

(1) `Method: GET`
`URI: /whatever.html`

(2) File: request.go
Line 115

(3) File: request.go
Line 293

(4) The request target in the request line. I.e., the path+filename.

2. (Golang) Consider the following server programs: server-final-2.go and server-final-3.go. Both are very primitive Web servers. The intent is to send an HTTP Response such that the client program outputs `File not found`. Execute the server first and then use a Web client (curl or Chrome) to request the server with this URL: http://`<IP address>:<port #>/whatever.html`.

   server-final-2.go

```
package main

import "fmt"
import "bufio"
import "net"
import "net/http"

func main() {
    ln, _ := net.Listen("tcp", ":<your port#>")
    defer ln.Close()
    conn, _ := ln.Accept()
    defer conn.Close()

    reader := bufio.NewReader(conn)
    req, _ := http.ReadRequest(reader)
    fmt.Printf("User Agent: %s\n", req.UserAgent())

    fmt.Fprintf(conn, "HTTP/1.1 404 Not Found\n")
    fmt.Fprintf(conn, "\r\n")
    fmt.Fprintf(conn, "File not found\n")
    fmt.Fprintf(conn, "\r\n")
}
```

server-final-3.go

```
package main

import "fmt"
import "bufio"
import "net"
import "net/http"

func main() {
    ln, _ := net.Listen("tcp", ":<your port#>")
    defer ln.Close()
    conn, _ := ln.Accept()
    defer conn.Close()

    reader := bufio.NewReader(conn)
    req, _ := http.ReadRequest(reader)
    fmt.Printf("User Agent: %s\n", req.UserAgent())

    fmt.Fprintf(conn, "HTTP/1.1 404 Not Found\n")
    fmt.Fprintf(conn, "Date: ...\n")
    fmt.Fprintf(conn, "File not found\n")
    fmt.Fprintf(conn, "\r\n")
}
```

(1) Tell the output on screen of the terminal running server-final-2.go. (1%)
(2) Does server-final-2.go work as intended and why? (2%)
(3) Tell the output on screen of the terminal running server-final-3.go. (1%)
(4) Does server-final-3.go work as intended and why? (2%)

Sample Solution:

(1) `User Agent: <whatever your user agent is>`

(2) Yes. The line “File not found” succeeds and precedes “\r\n” and therefore recognized as the message body. The Web client displays the message body in the output.

(3) `User Agent: <whatever your user agent is>`

(4) No. The line “File not found” succeeds a header line without the “\r\n” line. It is therefore recognized as another header line, not the message body. Therefore, the Web client won’t show the “File not found” line in the output.

3. (Golang) Consider the following server programs: server-final-4.go and server-final-5.go. Consider also the following screen dumps running `tcpdump` in the background: tcpdump-1 and tcpdump-2. One of the dumps, not in particular order, is collected from HTTP-requesting to server-final-4.go, and the other from HTTPS-requesting to server-final-5.go.

   server-final-4.go

```
package main

import "fmt"
import “net/http”

func main() {
    hh := http.HandlerFunc(helloHandler)
    http.Handle("/hello", hh)
    fs := http.FileServer(http.Dir("."))
    http.Handle("/", http.StripPrefix("/", fs))
    http.ListenAndServe(":<your port#>", nil)
}
```

   server-final-5.go

```
package main

import "fmt"
import “net/http”

func main() {
    hh := http.HandlerFunc(helloHandler)
    http.Handle("/hello", hh)
    fs := http.FileServer(http.Dir("."))
    http.Handle("/", http.StripPrefix("/", fs))
    http.ListenAndServeTLS(":<your port#>", "server.cer",
"server.key", nil)
}
```

tcpdump-1

```
15:06:04.127512 IP localhost.57993 > localhost.11999: Flags [P.], seq
1:93, ack 1, win 6379, options [nop,nop,TS val 856596845 ecr
856596845], length 92
E.....@.@.............C...*j.............
3..m3..mGET /whatever.html HTTP/1.1
Host: 127.0.0.1:11999
User-Agent: curl/7.54.0
Accept: */*

15:06:04.127541 IP localhost.11999 > localhost.57993: Flags [.], ack
93, win 6378, options [nop,nop,TS val 856596845 ecr 856596845], length
0
E..4..@.@.............*j..C........(.....
3..m3..m
```

tcpdump-2

```
15:10:25.866593 IP localhost.58020 > localhost.11999: Flags [P.], seq
447:528, ack 1420, win 6357, options [nop,nop,TS val 856858137 ecr
856858137], length 81
E.....@.@.................X.......y.....
3...3.......L.........3..0.0o...~'....N.......i......6.../5.g....|#.z
..Q..a....e1...x/..
15:10:25.866611 IP localhost.11999 > localhost.58020: Flags [.], ack
528, win 6371, options [nop,nop,TS val 856858137 ecr 856858137],
length 0
E..4..@.@.............X...........(.....
3...3...
```

(1) Which tcpdump output is the result of HTTP-requesting to server-final-4.go? (1%)
(2) Continue from (1), Why? (2%)
(3) Which tcpdump output is the result of HTTPS-requesting server-final-5.go? (1%)
(4) Continue from (3), Why? (2%)

Sample Solution:

(1) tcpdump-1

(2) One can see a clear-text HTTP message in tcpdump-1. Therefore tcpdump-1 can't be the result of running server-final-5.go. tcpdump-1 must be from server-final-4.go.

(3) tcpdump-2

(4) The other dump must then be the result of running server-final-5.go.

4. (Golang) Consider this server program: server-final-6.go. Some would argue that `defer conn.Close()` in the `for` loop inside `func main()` results in memory leaks.

   server-final-6.go

```
package main

import "fmt"
import "bufio"
import "net"


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


func handleConnection (c net.Conn) {
    reader := bufio.NewReader(c)
    message, errr := reader.ReadString('\n')
    check(errr)
    fmt.Printf("%s", message)

    writer := bufio.NewWriter(c)
    newline := fmt.Sprintf("%d bytes received\n", len(message))
    _, errw := writer.WriteString(newline)
    check(errw)
    writer.Flush()
}
```

```
func main() {
    fmt.Println("Launching server...")
    ln, _ := net.Listen("tcp", ":<your port#>")
    defer ln.Close()

    i := 1
    for {
        conn, _ := ln.Accept()
        defer conn.Close()

        fmt.Printf("%d ", i)
        handleConnection(conn)
        i++
    }
}
```

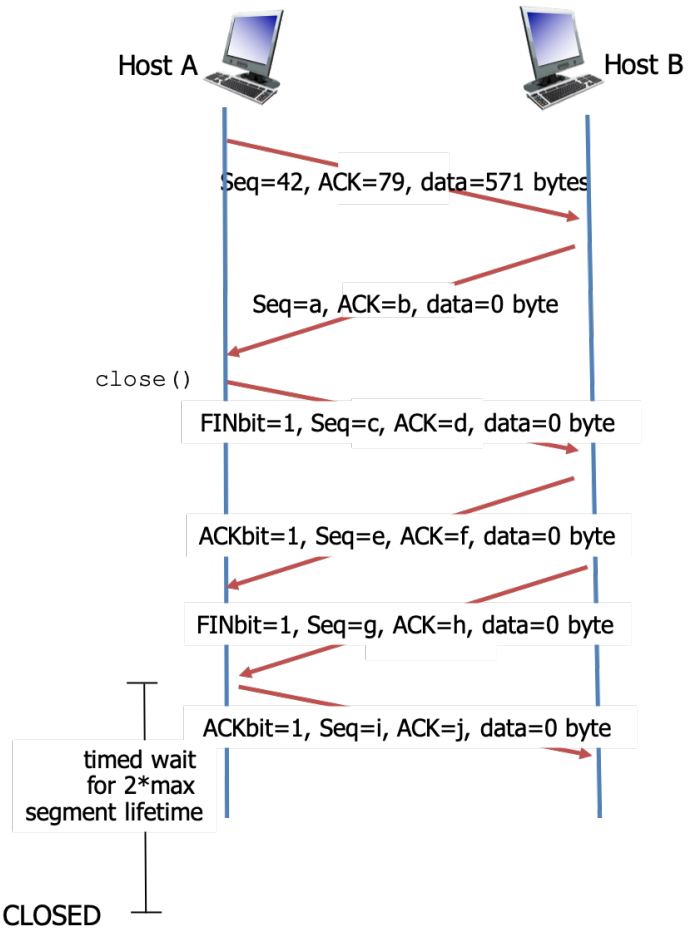
(1) If the server runs forever, is any of the `conn` created by `ln.Accept()` ever closed? (2%)
(2) What is a memory leak in general? (2%)
(3) Do you think deferring `conn.Close()` is a memory leak and why? (2%)
(4) What would you do to close the `conn` socket as soon as it's no longer needed? (2%)

Sample Solution:

(1) No. The program never terminates and therefore `conn.Close()` never get executed.
(2) Memory allocated by a program not released when it's no longer in use.
(3) Yes or no. State your reason.
(4) Several ways to re-write server-final-6.go. One example is to change the for loop as follows:

```
for {
    conn, _ := ln.Accept()
    fmt.Printf("%d ", i)
    handleConnection(conn)
    conn.Close()
    i++
}
```

5. (Transport) Depicted below is a sequence of packet exchange in a TCP connection between Host A and B. In that, Host A sends the last data packet and receives an ACK packet from Host B. Then, Host A sends a FIN packet to initiate the closing process. Host B sends a FIN packet as well when its data transmission completes.



(1) Tell the value of a, b, c, d, e, f. (6%)

(2) Towards the end, Host A needs to wait a good amount of time (2*max segment lifetime) before the connection is fully closed. Why? (2%)

Sample Solution:

(1) a=79, b=613, c=613, d=79, e=79, f=614

(2) The ACK packet back to Host B might be lost. In this case, Host B can still retransmit the FIN packet (when the timeout interval expires) until an ACK is finally received. Or to wait for the remaining (delayed, unnecessarily retransmitted) packets to go thru, so a new connection using the same port # will not accept these packets thinking they are the new ones.

6. (Transport) In TCP's reliable data transfer mechanism, the sender retransmits a data packet in case the ACK packet does not come back after the timeout interval. Setting of the timeout interval impacts the transmission efficiency. When it is too short, there'll be unnecessary retransmission. When it is too long, there'll be a long wait until the data packet can be retransmitted. The timeout interval is therefore better set to a value slightly larger than the RTT. However, TCP's algorithm (below) to estimate the timeout interval is way more complicated than just to sample the RTT.

   Step 0. Initialize α, β, EstimatedRTT, DevRTT
   Step 1. SampleRTT ← Timestamp(ACK receipt) – Timestamp(data transmission)
   Step 2. EstimatedRTT ← (1- α)*EstimatedRTT + α*SampleRTT
   Step 3. DevRTT ← (1-β)*DevRTT + β*|SampleRTT-EstimatedRTT|
   Step 4. TimeoutInterval ← EstimatedRTT + 4*DevRTT

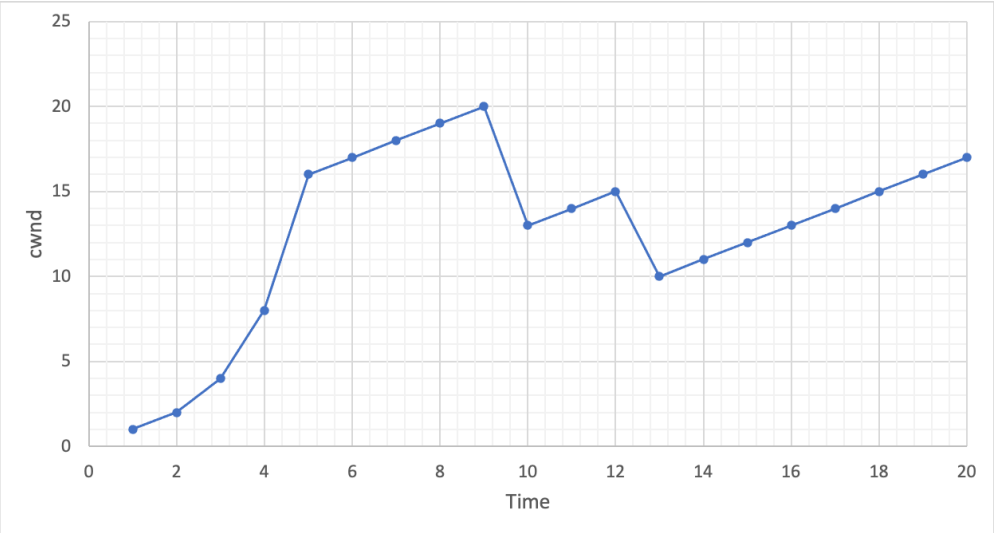
   (1) Step 1 is the part that samples the RTT of the TCP connection. Why not just use SampleRTT as the timeout interval? (1%)
   (2) Step 2 is to better estimate the RTT by taking the exponentially weighted moving average of the SampleRTT. Why not just taking a simple average of all the SampleRTT over time? (1%)
   (3) Step 3 defines a quantity, DevRTT. What is the meaning/significance of DevRTT? (2%)
   (4) Step 4 sets the TimeoutInterval finally. Why adding EstimatedRTT by 4*DevRTT, instead of 1*DevRTT or 6*DevRTT? (2%)

Sample Solution:

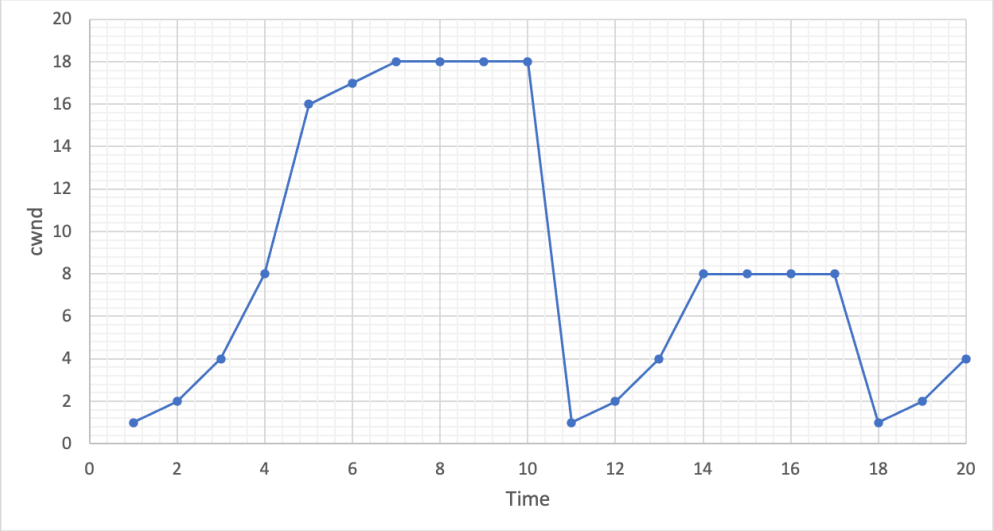
   (1) SampleRTT may fluctuate. Using just SampleRTT as the timeout interval leads to fluctuating timeout interval as well.
   (2) Looks like the designers like to take into consideration the historical SampleRTT, and in the meantime, to weigh more to recent SampleRTT. The weighted moving average captures both design considerations.
   (3) Smoothed deviation of the SampleRTT
   (4) Assuming DevRTT is a normal distribution, 4*DevRTT should be large enough to cover 99+% of SampleRTT. 1*DevRTT obviously too low. 6*DevRTT adds only a very small fraction of coverage. Much lower than 1%

7. (Transport) Drawn in the plots below are the progress of cwnd over time in a TCP connection. Each could be the result of running (1) Tahoe TCP, (2) Reno TCP without Fast Recovery, (3) Reno TCP with Fast Recovery, (4) any of the above, or (5) none of the above.

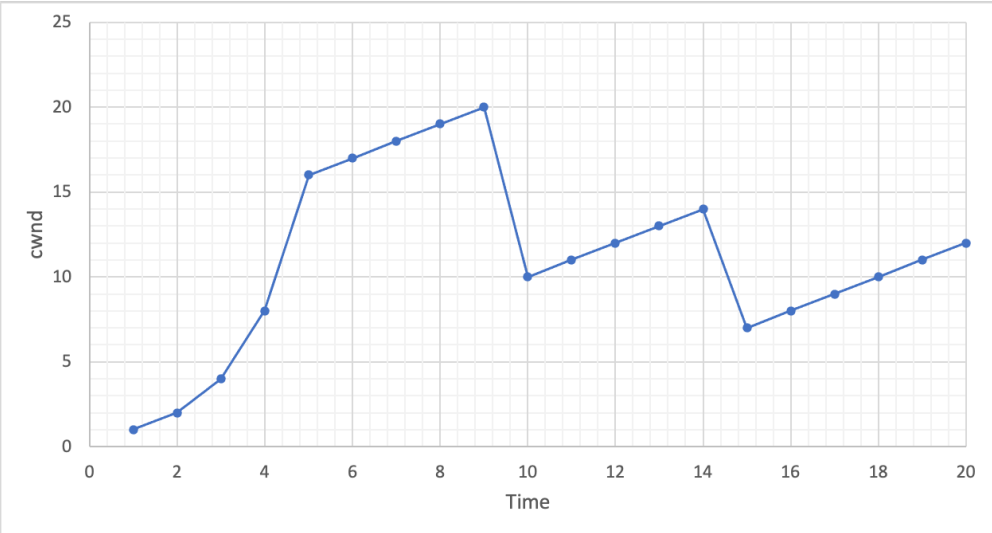
plot (a)



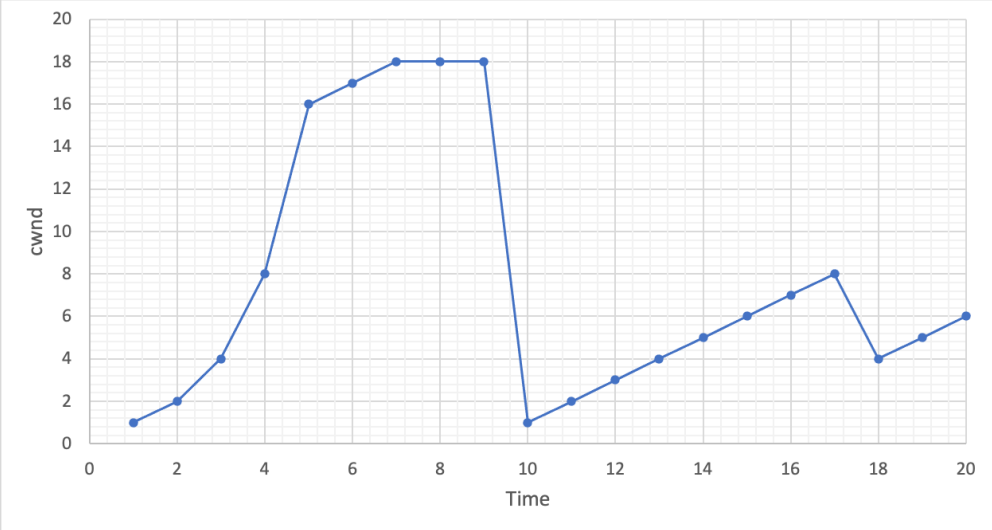
plot (b)



plot (c)



plot (d)



(1) Which option is true for plot (a)? (1%)
(2) Continue from (1). Why? (1%)
(3) Which option is true for plot (b)? (1%)
(4) Continue from (3). Why? (1%)
(5) Which option is true for plot (c)? (1%)
(6) Continue from (5). Why? (1%)
(7) Which option is true for plot (d)? (1%)
(8) Continue from (7). Why? (1%)

Sample Solution:

(1) (3) Reno with Fast Recovery

(2) cwnd dropping to new sshresh+3 and then to sshresh after a bit

(3) (4) Any of the TCP

(4) Tahoe, Reno w/ FR, Reno w/o FR behaving the same if there are only timeout losses

(5) (2) Reno w/o Fast Recovery

(6) cwnd dropping to new sshresh and then linear increasing again

(7) (5) Not any of the TCP

(8) Linear increase in slow start…

8. (Transport) Which of the following are characteristic of network-assisted congestion control. (5%)

   (1) Using packet loss and delay inferred at the data source as signals of congestion
   (2) Setting the ECN bit in the IP packet header
   (3) Adjusting the congestion window size based on the free buffer space
   (4) Adjusting the congestion window size based on the queue size
   (5) Setting the ECE bit in the TCP packet header

Sample Solution:

(2), (3), (4), (5)

9. (Data Plane) Organization YouKnowWho owns subnet 140.112.42.0/24. There are multiple departments within the organization. Consider the IP address demand and then allocate a subnet address in a.b.c.d/x form to each department.

   (1) Suppose there are 4 departments and each asks for as many IP addresses as possible. Give the 4 subnet addresses for the 4 departments. (2%)
   (2) Suppose there are 5 departments and each asks for as many IP addresses as possible. Give the 5 subnet addresses for the 5 departments. (2%)
   (3) Suppose there are 3 departments and they ask for a minimum of 40, 100, and 20 IP addresses respectively. Try to be conservative and give the 3 subnet addresses just enough to cover the requested minimums. (3%)

Sample Solution:

   (1) 140.112.42.0/26, 140.112.42.64/26, 140.1112.42.128/26, 140.112.42.192/26
   (2) 140.112.42.0/27, rest are the same except for the last octet: 32, 64, 96, 128 (160, 192, 224). (Or any 5 with a prefix length <=27, disjoint, and within 140.112.42.0/24)
   (3) 140.124.42.0/26, 140.112.42.128/25, 140.112.42.64/27 (or 26)
   Alternatively, 140.112.42.128/26, 140.112.42.0/25, 140.112.42.192/27 (or 26). Or (Or any 3 that are sufficiently large, disjoint, and within 140.112.42.0/24)

10. (Data Plane) You are hired to manage the free TPE-Metro WiFi access in station Technology Building. The WiFi APs have been set up to densely cover the platform. These APs are connected to the internal subnet 192.168.0.0/24. The subnet is connected via a NAT box to the ISP. The DHCP server, running on the NAT box, on the subnet defaults to 1 hour for all leases. The DHCP server is also refreshed every morning right before the station begins service.

Every now and then, passengers complain about not being able to connect to the free WiFi. You begin to monitor the passenger flow and observe that there're about 60 passengers waiting on the platform before their train arrives. And each train is 10 minutes apart. Let's consider the worst case, where all the passengers waiting on the platform ask to connect to TPE-Metro. Obviously, the first 60 passengers waiting for the 1st train will not complain.

(1) Until which train, you see some passengers complaining? (1%)
(2) Until which train, the 60 passengers are all satisfied again? (1%)
(3) Without changing the subnet address space, how would you change the lease time so all passengers (all day long) are satisfied? (1%)
(4) Without changing the lease time, how would you change the subnet prefix length so that all passengers (all day long) are satisfied? (1%)
(5) Now take into consideration the rush hour demand. There are typically 180 passengers waiting on the platform before their train arrives. And each train is 6 minutes apart. Without changing the subnet address space, what'll be the lease time so all passengers (all day long) are satisfied? (2%)
(6) Continue from (5). Without changing the lease time, what'll be the subnet prefix length so that all passengers (all day long) are satisfied? (2%)

Sample Solution:

(1) Train #5
(2) Train #7
(3) Lease time <= 40 mins
(4) Prefix length <= 23
(5) Lease time <= 6 mins
(6) Prefix length <= 21

11. (Data Plane) SDN routers are general machines capable of traffic discrimination based on Transport, Network, and Link layer packet headers. When an SDN router receives a packet, it looks up the flow table and find all the matching entries. Among the entries, find the one with the highest priority. Then take the action indicated in the entry. Below are the flow tables in two different SDN routers. Each consists of 2 entries.

SDN-1:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action	Priority
*	*	*	*	*	140.112.*.*	*	*	*	*	allow	high
*	*	*	*	*	*	*	*	*	*	drop	low

SDN-2:

Switch Port	MAC Src	MAC Dst	Eth Type	VLAN ID	IP Src	IP Dst	IP Proto	TCP Sport	TCP Dport	Action	Priority
*	18:65:90:df:14:cf	*	*	*	140.112.42.161	*	*	*	*	allow	high
*	*	*	*	*	140.112.42.161	*	*	*	*	drop	low

(1) Will SDN-1 allow all packets from 140.112.42.161 to go through? (1%)

(2) Suppose 140.112.0.0/16 is the subnet address of NTU. What can you do from outside of NTU to send packets through SDN-1? (2%)

(3) Will SDN-2 allow all packets from 140.112.42.161 to go through? (1%)

(4) Suppose SDN-2 is the WiFi AP of network 140.112.42.0/24 and there is already a workstation connected to the AP. The workstation's IP address and MAC address are 140.112.42.161 and 18:65:90:df:14:cf respectively. What can you do to spoof the workstation (i.e., send packet through the SDN router)? (2%)

Sample Solution:

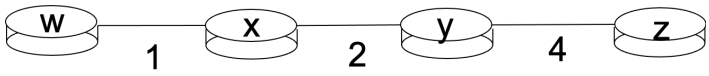
(1) Yes

(2) VPN (into NTU). IP address spoofing – come to NTU campus and try to find a guest network to join. (Anything that makes sense)

(3) No (only those with source MAC address 18:65:90:df:14:cf)

(4) MAC spoofing (by downloading drivers allowing changing of MAC address and etc. It's easier than you think: https://www.giac.org/paper/gsec/3199/mac-spoofing-an-introduction/105315). (Anything that makes sense)

12. (Control Plane) Consider a simple network of 4 routers as shown below. Complete the Link State route derivation and generate the forwarding tables for node x and y.



LS route derivation table for node x:

Travel Set	D(w),p(w)	D(y),p(y)	D(z),p(z)
x	1, x	2, x	Inf, --

Forwarding table for node x:

Destination	Output link
w	
y	
z	

LS route derivation table for node y:

Travel Set	D(w),p(w)	D(x),p(x)	D(z),p(z)
y	inf, --	2, y	4, y

Forwarding table for node y:

Destination	Output link
w	
x	
z	

(1) Complete the LS route derivation table for node x. (1%)
(2) Fill the forwarding table for node x. (1%)
(3) Complete the LS route derivation table for node y. (1%)
(4) Fill the forwarding table for node y. (1%)

Sample Solution:

(1)

Travel Set	D(w),p(w)	D(y),p(y)	D(z),p(z)
x	1, x	2, x	Inf, --
xw		2, x	Inf, --
xwy			6, y
xwyz			

(2)

Destination	Output link
w	(x, w)
y	(x, y)
z	(x, y)

(3)

Travel Set	D(w),p(w)	D(x),p(x)	D(z),p(z)
y	inf, --	2, y	4, y
yx	3, x		4, y
yxw			4, y
yxwz			

(4)

Destination	Output link
w	(y, x)
x	(y, x)
z	(y, z)

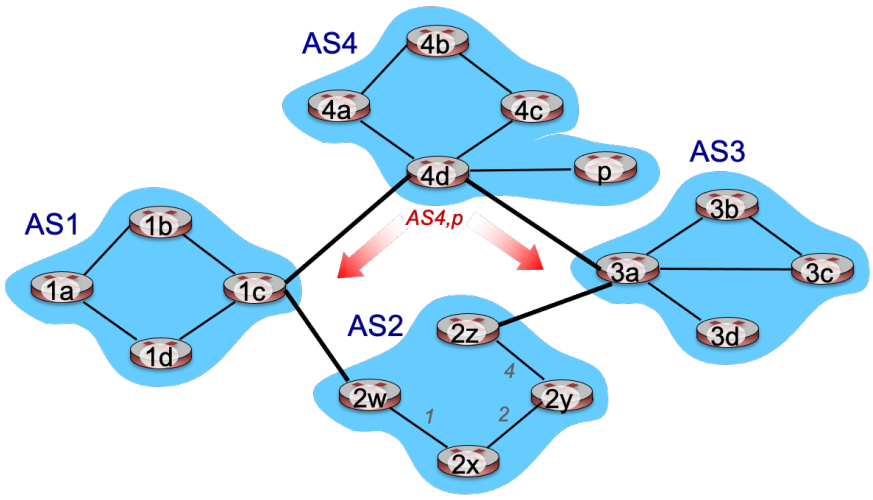
13. (Control Plane) Consider the same network (from Problem 12). Complete the Distance Vector route derivation. At t0, the DV table for each node is initialized and the initial DV is sent to the neighboring nodes. At t1, each node completes the 1st iteration of Bellman-Ford computation (based on the DV received from the neighboring nodes) and derive the new Dw, Dx, Dy, and Dz.

    (1) Tell the Dw in node w at t1. (1%)
    (2) Tell the Dx in node x at t1. (1%)
    (3) Tell the Dy in node y at t1. (1%)
    (4) Tell the Dz in node z at t1. (1%)
    (5) As a human being and seeing the entire topology, the next hop from node x to z is obviously node y. As a router and knowing only the Bellman-Ford equation and the neighbors' DVs, how does node x learn the next hop towards a destination? (2%)

Sample Solution:

    (1) Dw = (0 1 3 inf)
    (2) Dx = (1 0 2 6)
    (3) Dy = (3 2 0 4)
    (4) Dz = (inf 6 4 0)
    (5) The next hop will be the neighbor that gives the min distance in the Bellman-Ford calculation.

14. (Control Plane) 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 or export restrictions in AS3. What would 3a Advertise to 2z? (1%)

(2) Suppose there are no import or export restrictions in AS1. What would 1c Advertise to 2w? (1%)

(3) If there're no import restrictions in AS2 either, what would the entry be for destination p in 2y's forwarding table, and why? (2%)

(4) If there's an import restriction in AS2 preventing 2w from propagating the received AS path further, what would the entry be for destination p in 2x's forwarding table, and why? (2%)

Sample Solution:

(1) AS3, AS4, p

(2) AS1, AS4, p

(3)

Destination	Output link
p	(2y, 2x)

Reason – 2y receives two AS paths to p. There are no local preferences. The two paths are equal length. Therefore, 2x will hot-potato route to the gateway router that's closer.

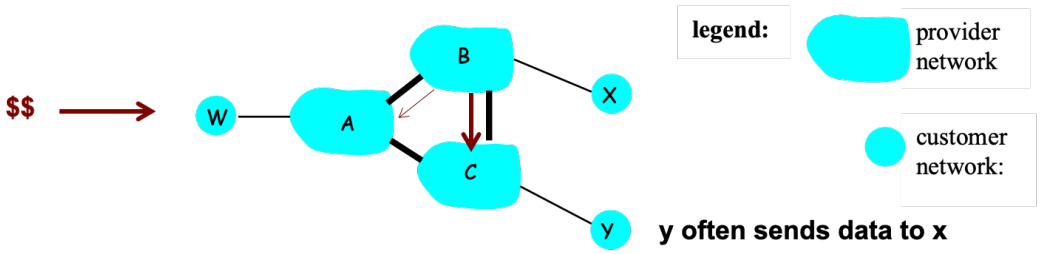
(4)

Destination	Output link
p	(2x, 2y)

Reason – 2x receives only 1 AS path from 2z. Therefore, the output link will be on the shortest path to 2z.

15. (Control Plane) Recall the following network running BGP at the inter-AS level (from Quiz #37). AS x announces its existence to B. B then advertises the AS path "Bx" to the neighboring ASes. If there's only 1 neighboring AS, B must advertise the AS path to allow x access to the rest of the Internet. Here however, B's got 2 neighboring ASes. There is a bit of flexibility and B could choose to advertise the AS path to just A, just C, or both.

    Suppose B is a newcomer to the ISP business. A and C, being in the business for a while, charge B for Internet access (just like how B charges x for Internet access). Say A charges B a lower amount of $$ per unit traffic (in and out) than C does.



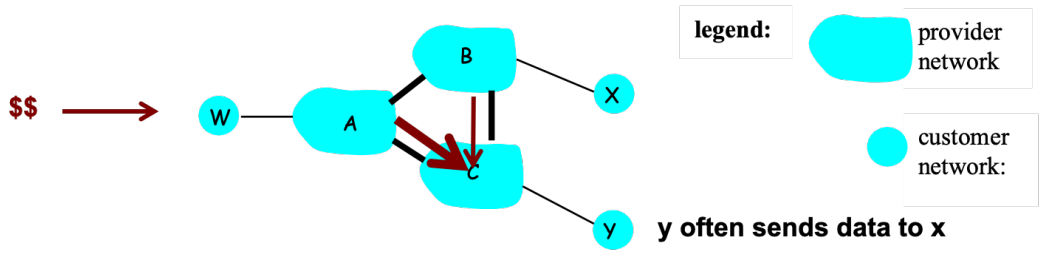
(1) Would you suggest B to advertise the AS path "Bx" to only A, only C, or both? (1%)
(2) Continue from (1). Why? (1%)
(3) If the decision is to advertise the AS path only to A, how would the network admin of B set the BGP import or export policy to enforce the rule? (1%)

Sample Solution:

(1) Only A (or your choice)
(2) To save money (as long as they can be justified)
(3) Set export policy to advertise AS paths only to A (as long as this is making sense)

16. (Control Plane) Consider the same network (from Problem 15). Let's say C receives advertisement "Bx" from B and "ABx" from A. If there's only 1 advertisement for destination x, C must prepend itself to the AS path and advertise the extended AS path further to allow y access to x. Here however, C's got 2 advertisements from the neighboring ASes. There is a bit of flexibility and C could choose to advertise just "CBx", just "CABx", or both paths to y.

    Suppose C is the most established ISP. A and B both pay C for Internet access. We just learned from C's business department that A pays more than B does per unit traffic.



(1) Would you suggest C to advertise the AS path "CBx" or "CABx", or both to y? (1%)
(2) Continue from (1). Why? (1%)
(3) If the decision is to advertise only the AS path "CBx", how would the network admin of C set the BGP import or export policy to enforce the rule? (1%)

Sample Solution:

(1) "CABx" (or your choice)
(2) To earn more (as long as they can be justified)
(3) Set import policy to accept AS paths only from B (or export policy to prepend and forward paths from B)

17. (Control Plane) ONOS and ODL are two router OSes for SDN networks.
    (1) What is the main difference between the two OSes? (1%)
    (2) Which one would you prefer if you are managing an SDN network? (1%)
    (3) Continue from (2). Why the preference? (2%)

Sample Solution:
    (1) Control apps being completely outside of the controller (ONOS, clean slate) vs. some apps being inside (ODL, legacy)
    (2) Take your pick.
    (3) Justify for yourself.