Homework 4

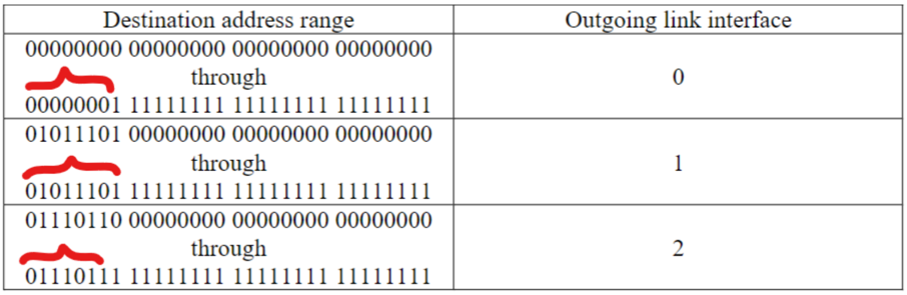
0. Academic integrity

I have read and understood the course academic integrity policy.

Chapter 4

1. **Longest prefix matching** – Consider a datagram network using 32-bit host addresses.

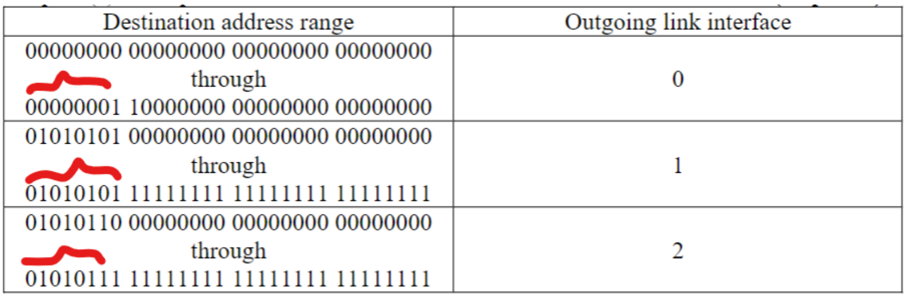
a) Suppose that a router has three interfaces, numbered 0, 1, 2, and that packets are to be forwarded to these link interfaces as follows. Any address not within the ranges in the table below should not be forwarded to an outgoing link interface. Create a forwarding table using longest prefix matching.



Destination address range	Outgoing link interface
00000000 00000000 00000000 00000000 through 00000001 11111111 11111111 11111111	0
01011101 00000000 00000000 00000000 through 01011101 11111111 11111111 11111111	1
01110110 00000000 00000000 00000000 through 01110111 11111111 11111111 11111111	2

Destination address range	Outgoing link interface
0000000	0
01011101	1
0111011	2

b) Repeat (a) when packets are to be forwarded to the link interfaces as follows.



Destination address range	Outgoing link interface
00000000 00000000 00000000 00000000 through 00000001 10000000 00000000 00000000	0
01010101 00000000 00000000 00000000 through 01010101 11111111 11111111 11111111	1
01010110 00000000 00000000 00000000 through 01010111 11111111 11111111 11111111	2

Destination address range	Outgoing link interface
0000000	0
01010101	1
0101011	2

2. **Subnets** – An organization has been assigned the prefix 222.3.4.0/24 and wants to form subnets for 4 departments A, B, C, and D, with hosts as follows:

A	95 hosts
B	47 hosts
C	28 hosts
D	17 hosts

There are 187 hosts in all. Give a possible arrangement of subnet IP addresses to make this possible. Use the notation a.b.c.d/x.

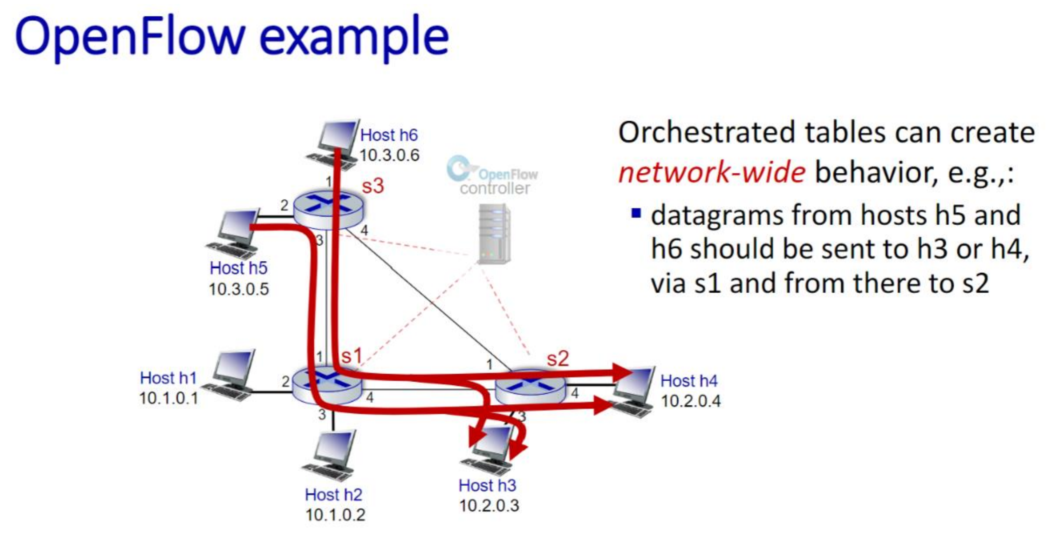
First from the original prefix 222.3.4.0 we get IP of main host, wich is a C class IP.
A's interval:
First, subnet A needs 95 hosts and $2^6 = 64 \leq 95 \leq 2^7 = 128$, we need to borrow 7 bits of the main host. At the same time, we should also consider to remove the case where bits filled with 0 and bits filled with 1, because they are subnet network address and broadcast address respectively. As a consequence, the scope of A's IP is 222.3.4.1 to 222.3.4.126 (222.3.4.00000001 – 222.3.4.01111110).
The notation a.b.c.d/x is 222.3.4.0/25.

B's interval:
Second subnet B needs 47 hosts and $2^5 = 32 \leq 47 \leq 2^6 = 64$, we need to borrow 6 bits of the main host. As with the analysis of subnet A, this subnetting problem also requires the removal of the first and last. As a consequence, the scope of B's IP is 222.3.4.129 to 222.3.4.190 (222.3.4.10000001 – 222.3.4.10111110).
The notation a.b.c.d/x is 222.3.4.128/26.

C's interval:
Third subnet C needs 28 hosts and $2^4 = 16 \leq 28 \leq 2^5 = 32$, we need to borrow 5 bits of the main host. As with the analysis of subnet A, this subnetting problem also requires the removal of the first and last. As a consequence, the scope of C's IP is 222.3.4.193 to 222.3.4.222 (222.3.4.11000001 – 222.3.4. 11011110).
The notation a.b.c.d/x is 222.3.4.192/27.

D's interval:
Finally subnet D needs 17 hosts and $2^4 = 16 \leq 17 \leq 2^5 = 32$, we need to borrow 5 bits of the main host. As with the analysis of subnet A, this subnetting problem also requires the removal of the first and last. As a consequence, the scope of D's IP is 222.3.4.225 to 222.3.4.254 (222.3.4.11100001 – 222.3.4. 11111110).
The notation a.b.c.d/x is 222.3.4.224/27

3. **OpenFlow** – Consider the SDN OpenFlow network shown in Figure 4.30 (slide 84). Suppose that the desired forwarding behavior for datagrams arriving at s2 is as follows:



OpenFlow example

Orchestrated tables can create *network-wide* behavior, e.g.,:

- datagrams from hosts h5 and h6 should be sent to h3 or h4, via s1 and from there to s2

• Any datagrams arriving on input port 1 from hosts h5 or h6 that are destined to hosts h1 or h2 should be forwarded over output port 2;

s2 Flow Table	
Match	Action
Ingress Port = 1, IP Src = 10.3.0.5, IP Dst = 10.1.0.1	Forward(2)
Ingress Port = 1, IP Src = 10.3.0.5, IP Dst = 10.1.0.2	Forward(2)
Ingress Port = 1, IP Src = 10.3.0.6, IP Dst = 10.1.0.1	Forward(2)
Ingress Port = 1, IP Src = 10.3.0.6, IP Dst = 10.1.0.2	Forward(2)

• Any datagrams arriving on input port 2 from hosts h1 or h2 that are destined to hosts h5 or h6 should be forwarded over output port 1;

s2 Flow Table	
Match	Action
Ingress Port = 2, IP Src = 10.1.0.1, IP Dst = 10.3.0.5	Forward(1)
Ingress Port = 2, IP Src = 10.1.0.1, IP Dst = 10.3.0.6	Forward(1)
Ingress Port = 2, IP Src = 10.1.0.2, IP Dst = 10.3.0.5	Forward(1)
Ingress Port = 2, IP Src = 10.1.0.2, IP Dst = 10.3.0.6	Forward(1)

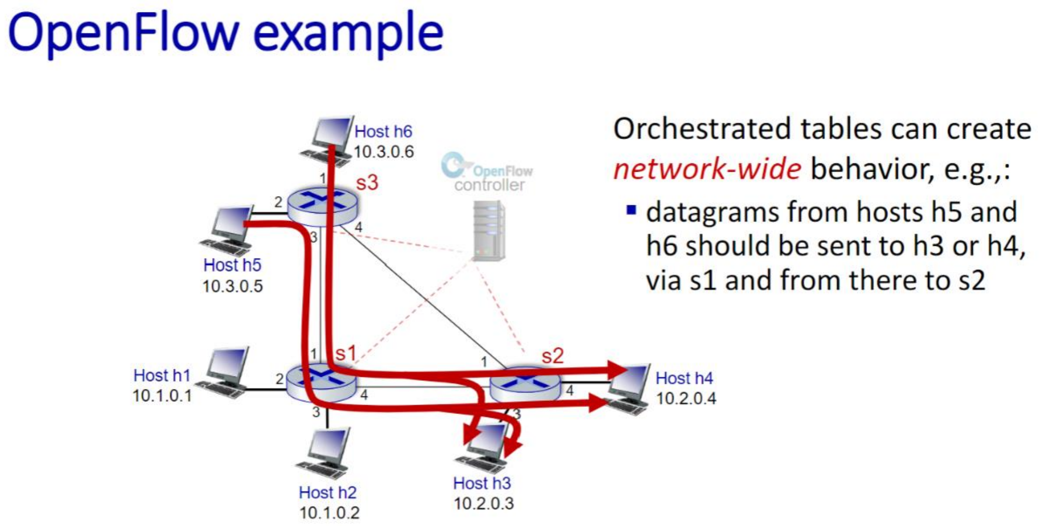
• Any arriving datagrams on input ports 1or 2 and destined to hosts h3 or h4 should be delivered to the host specified;

s2 Flow Table	
Match	Action
Ingress Port = 1, IP Dst = 10.2.0.3	Forward(3)
Ingress Port = 2, IP Dst = 10.2.0.3	Forward(3)
Ingress Port = 1, IP Dst = 10.2.0.4	Forward(4)
Ingress Port = 2, IP Dst = 10.2.0.4	Forward(4)

- Hosts h3 and h4 should be able to send datagrams to each other.

s2 Flow Table	
Match	Action
Ingress Port = 3, IP Src = 10.2.0.3, IP Dst = 10.2.0.4	Forward(4)
Ingress Port = 4, IP Src = 10.2.0.4, IP Dst = 10.2.0.3	Forward(3)

4. **OpenFlow** – Consider again the SDN OpenFlow network shown in Figure 4.30 (slide 84). Suppose we want switch s2 to function as a firewall. Specify the flow table in s2 that implements the following firewall behaviors (specify a different flow table for each of the four firewalling behaviors below) for delivery of datagrams destined to h3 and h4. You do not need to specify the forwarding behavior in s2 that forwards traffic to other routers.



OpenFlow example

Orchestrated tables can create *network-wide* behavior, e.g.,:

- datagrams from hosts h5 and h6 should be sent to h3 or h4, via s1 and from there to s2

- Only traffic arriving from hosts h1 and h6 should be delivered to hosts h3 or h4 (i.e., that arriving traffic from hosts h2 and h5 is blocked).

s2 Flow Table	
Match	Action
IP Src = 10.1.0.1, IP Dst = 10.2.0.3	Forward(3)
IP Src = 10.3.0.6, IP Dst = 10.2.0.3	Forward(3)
IP Src = 10.1.0.1, IP Dst = 10.2.0.4	Forward(4)
IP Src = 10.3.0.6, IP Dst = 10.2.0.4	Forward(4)

• Only TCP traffic is allowed to be delivered to hosts h3 or h4 (i.e., that UDP traffic is blocked).

s2 Flow Table	
Match	Action
Protocol is TCP, IP Dst = 10.2.0.3	Forward(3)
Protocol is TCP, IP Dst = 10.2.0.4	Forward(4)

• Only traffic destined to h3 is to be delivered (i.e., all traffic to h4 is blocked).

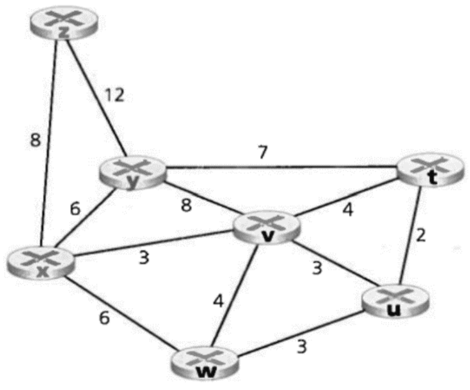
s2 Flow Table	
Match	Action
IP Dst = 10.2.0.3	Forward(3)

• Only UDP traffic from h1 and destined to h3 is to be delivered. All other traffic is blocked.

s2 Flow Table	
Match	Action
Protocol is UDP, IP Src = 10.1.0.1, IP Dst = 10.2.0.3	Forward(3)

Chapter 5

5. **Dijkstra's algorithm** – Consider the following network. With the indicated link costs, use Dijkstra's shortest-path algorithm to compute the shortest path from x to all network nodes. Show how the algorithm works by computing a table similar to Table 5.1.

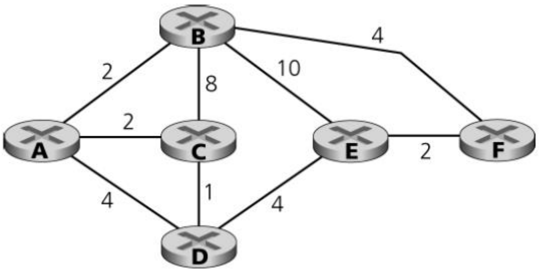


Step	N’	D/p(z)	D/p(u)	D/p(w)	D/p(v)	D/p(t)	D/p(y)
0	x	8, x	∞	6, x	3, x	∞	6, x
1	xv	8, x	6, v	6, x		7, v	6, x
2	xvu	8, x		6, x		7, v	6, x
3	xvuw	8, x				7, v	6, x
4	xvuwy	8, x				7,v	
5	xvuwyt	8, x					
6	xvuwytz						

Step 1: D(v) = 3 is the smallest, throw v into N’.
Step 2: D(u) = 6 is the smallest, throw u into N’.
Step 3: D(w) = 6 is the smallest, throw w into N’.
Step 4: D(y) = 6 is the smallest, throw y into N’.
Step 5: D(t) = 7 is the smallest, throw t into N’.
Step 6: D(z) = 8 is the smallest, throw z into N’.

6. **Distance vector algorithm** – Consider the network below:

a) What are A, B, C, D, E, and F’s distance vectors? Note: you do not have to run the distance vector algorithm; you should be able to compute the distance vectors by inspection. Recall that a node’s distance vector is the vector of **the least cost paths** from itself to each of the other nodes in the network.



A’s distance vectors:
(A,A): 0,
(A,B): 2 route is A – B,
(A,C): 2, route is A – C,
(A,D): 3, route is A – C – D,
(A,E): 7, route is A – C – D – E,
(A,F): 6, route is A – B – F.

B’s distance vectors:
(B,A): 2, route is B – A,
(B,B): 0,
(B,C): 4, route is B – A – C,
(B,D): 5, route is B – A – C – D,

(B,E): 6, route is B – F – E,
(B,F): 4, route is B – F.

C's distance vectors:
(C,A): 2, route is C – A,
(C,B): 4, route is C – A – B,
(C,C): 0,
(C,D): 1, route is C – D,
(C,E): 5, route is C – D – E,
(C,F): 7, route is C – D – E – F.

D's distance vectors:
(D,A): 3, route is D – C – A,
(D,B): 5, route is D – C – A – B,
(D,C): 1, route is D – C,
(D,D): 0,
(D,E): 4, route is D – E,
(D,F): 6, route is D – E – F.

E's distance vectors:
(E,A): 7, route is E – D – C – A,
(E,B): 6, route is E – F – B,
(E,C): 5, route is E – D – C,
(E,D): 4, route is E – D,
(E,E):0,
(E,F): 2, route is E – F.

F's distance vectors:
(F,A): 6, route is F – B – A,
(F,B): 4, route is F – B,
(F,C): 7, route is F – E – D – C,
(F,D): 6, route is F – E – D,
(F,E): 2, route is F – E,
(F,F): 0.

b) Now consider node C. From which other nodes does C receive distance vectors?

A, B, D are linked to C.

c) Consider node C again. Through which neighbor will C route its packets destined to E? Explain how you arrived at your answer, given the distance vectors that C has received from its neighbors.

D. Because there are only two considered paths (C, D, E) and (C, B, E) and obviously 1 + 4 = 5 < 8 + 10 = 18.

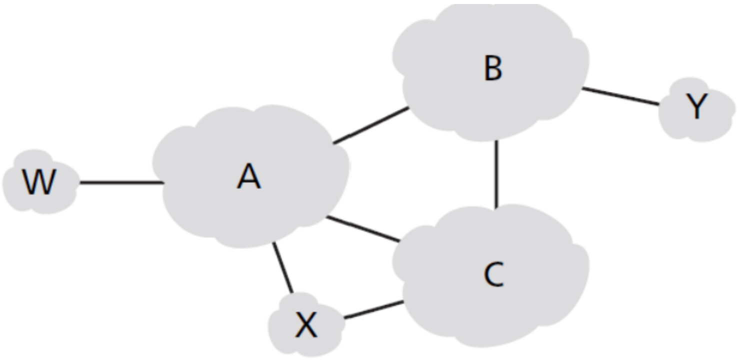
d) Consider node E. From which other nodes does E receive distance vectors?

B, D, F are linked to E.

e) Consider node E again. Through which neighbor will E route its packets destined to B? Explain how you arrived at your answer, given the distance vectors that E has received from its neighbors.

F. Because there are three considered paths (E, B), (E, D, C, A, B) and (E, F, B) and obviously 2 + 4 = 6 is the minimum.

7. **BGP** – Consider the network below in which network W is a customer of ISP A, network Y is a customer of ISP B, and network X is a customer of both ISPs A and C.



a) What BGP routes will A advertise to X?

W-A-X, Y-A-X, since A must provide every BGP information to customers.

b) What routes will X advertise to A?

No routes, since X is a customer of A.

c) What routes will A advertise to C?

W-A-C, X-C, since it need to prevent B's free ride, there's no Y.

For each answer, proide a one-sentence explanation.

8. **Bonus problem** – You are interested in detecting the number of hosts behind a NAT. You observe that the IP layer stamps an identification number sequentially on each IP packet. The identification number of the first IP packet generated by a host is a random number, and the identification numbers of the subsequent IP packets are sequentially assigned. Assume all IP packets generated by hosts behind

the NAT are sent to the outside world.

a) Based on this observation, and assuming you can sniff all packets sent by the NAT to the outside, can you outline a simple technique that detects the number of unique hosts behind a NAT? Justify your answer.

According to the question we know that the identification number of each IP packet is sequential. Thus we could filter out each successive sequence number by sorting and counting, and then by selecting and dividing the ID number of IP packets sent from a known NAT then we can get the number of hidden hosts.

b) If the identification numbers are not sequentially assigned but randomly assigned, would your technique work? Justify your answer.

No. The random assigned identification number will make its order regularity not exist. All numbers are mixed up, thus we can't identify which is which.