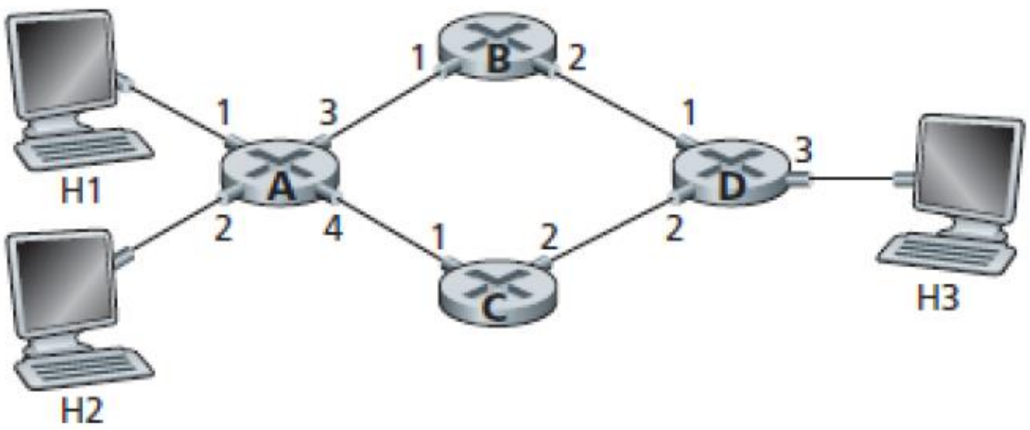
CSC358 Tutorial 8

TA: Lilin Zhang

March 14-15, 2016

Q1 Forwarding Table

- (a) Suppose the network is a **datagram network**. Show the forwarding table in router A, such that **all traffic destined to H3 is forwarded through interface 3.**

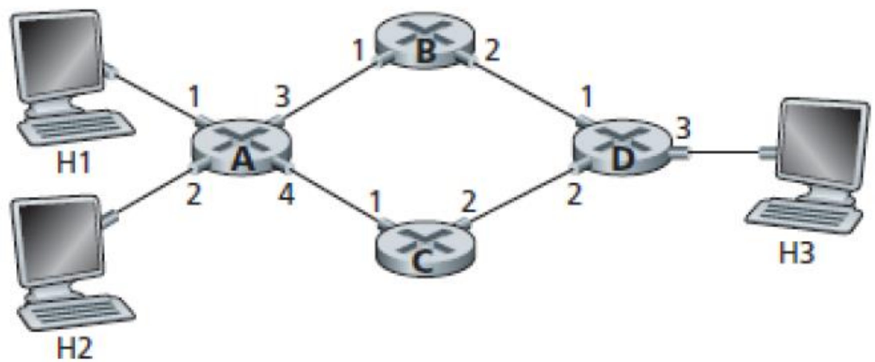


Datagram forwarding table

Destination address	Interface
H3	3

Q1 Forwarding Table

- (b) Suppose the network is a **datagram network**. Can you write a forwarding table for router A, such that all traffic **H1->H3 is forwarded through interface 3,** while all traffic **H2->H3 is forwarded through interface 4.**

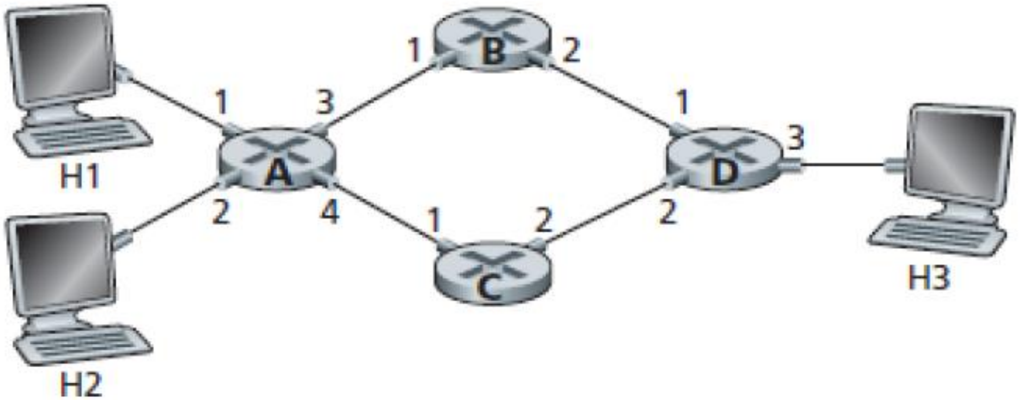


No such datagram forwarding table! (forwarding rule is based on destination address only)

Destination address	Interface
H3	3
H3	4?

Q1 Forwarding Table

- (c) Suppose the network is a **virtual circuit network**. There is one ongoing call between H1 and H3, and another ongoing call between H2 and H3. Show the forwarding table in router A, such that all traffic H1->H3 is forwarded through interface 3, while all traffic H2->H3 is forwarded through interface 4.

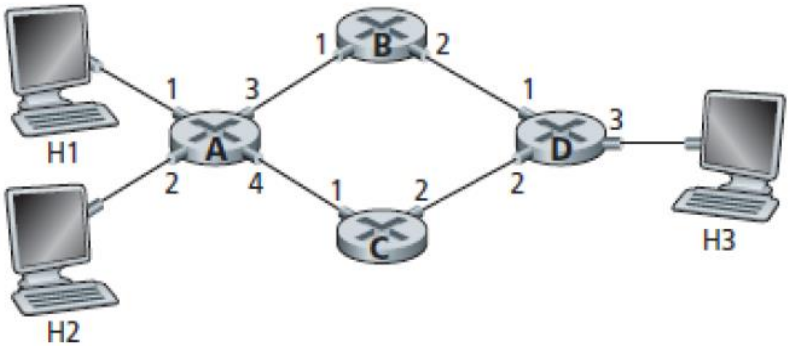


VC forwarding table
(one possibility)

Incoming interface	VC #	Outgoing interface	VC #
1	12	3	22
2	63	4	18

Q1 Forwarding Table

- (d) assume the same scenario as (c). Show the forwarding tables in router B,C,D.



B

Incoming interface	VC #	Outgoing interface	VC #
1	22	2	24

C

Incoming interface	**VC #**	**Outgoing interface**	**VC #**
1	18	2	50

A

Incoming interface	**VC #**	**Outgoing interface**	**VC #**
1	12	3	22
2	63	4	18

Router D

Incoming interface	**VC #**	**Outgoing interface**	**VC #**
1	24	3	**70**
2	50	3	**76**

Q2 Longest Prefix Match

- Consider a datagram network using 32-bit host addresses. Suppose the router has 4 links, numbered from 0 to 3. Packets are forwarded to link interfaces as follow:

Destination Address Range	Link Interface
11100000 00000000 00000000 00000000 through 11100000 00111111 11111111 11111111	0
11100000 01000000 00000000 00000000 through 11100000 01000000 11111111 11111111	1
11100000 01000001 00000000 00000000 through 11100001 01111111 11111111 11111111	2
otherwise	3

Q2 Longest Prefix Match

- (a) provide a forwarding table that has 5 entries, uses longest prefix matching, and forwards packets to the correct interfaces.

Datagram forwarding table

Destination Address Range	Link Interface
11100000 00000000 00000000 00000000 through 11100000 00111111 11111111 11111111	0
11100000 01000000 00000000 00000000 through 11100000 01000000 11111111 11111111	1
11100000 01000001 00000000 00000000 through 11100001 01111111 11111111 11111111	2
otherwise	3

Datagram forwarding table
(longest prefix match)

Prefix Match	Link Interface
11100000 00	0
11100000 01000000	1
1110000	2
11100001 1	3
otherwise	3

Q2 Longest Prefix Match

- (b) Describe how your forwarding table determines the appropriate link interface for the datagrams with destination addresses:

11001000 10010001 01010001 01010101	Matches to "otherwise" entry, link 3
11100001 01000000 11000011 00111100	Matches to "111000" entry, link 2
11100001 10000000 00010001 01110111	Matches to "1110001 1" entry, link 3

Datagram forwarding table

Prefix Match	**Link Interface**
11100000 00	0
11100000 01000000	1
1110000	2
11100001 1	3
otherwise	3

Q3 Address Range

- Consider a datagram network using 8-bit host addresses. Suppose router uses longest prefix match and has the following forwarding table:

For each of the four interfaces, write down the associated range of host addresses and the number addresses in that range.

Prefix Match	Interface
1	0
10	1
111	2
otherwise	3

Q3 Address Range

Destination Address Range	Link Interface
11000000 through (32 addresses) 11011111	0
10000000 through(64 addresses) 10111111	1
11100000 through (32 addresses) 11111111	2
00000000 through (128 addresses) 01111111	3

Prefix Match	Interface
1	0
10	1
111	2
otherwise	3

Q4 Subnet IP Address

- Consider a subnet with prefix 128.119.40.128/26. Give an example of one IP address that can be assigned to this network.
  - **10000000.01110111.0101000.10**000000/26
  - Any IP in the following range:
  - **10000000.01110111.0101000.10**000000 **10000000.01110111.0101000.10**111111
  - i.e., any IP in the range:
  - 128.119.40.128 ~ 128.119.40.191

Q4 Subnet IP Address

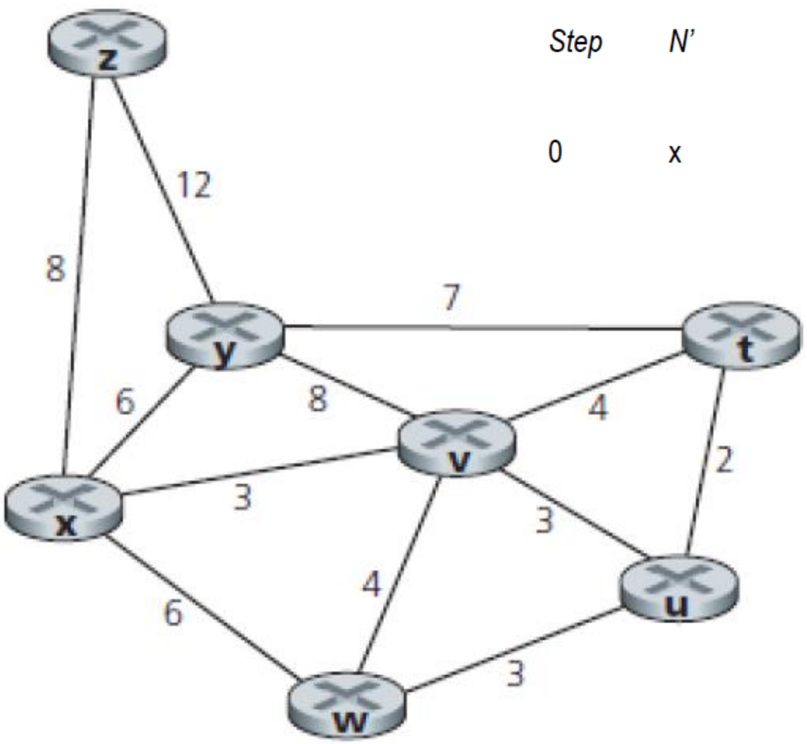
- Suppose an ISP owns a block of addresses of the form 128.119.40.64/26. Suppose it wants to create 4 subnets from this block, with each block having the same number of IP addresses. What are the prefixes for the four subnets.
  - **10000000.01110111.0101000.01**000000/26
  - 2^6 = 64 addresses in total;
  - each equal-size subnet needs to have 16 addresses (2^4)
  - **10000000.01110111.0101000.0100**0000/28 i.e., 128.119.40.64/28
  - **10000000.01110111.0101000.0101**0000/28 i.e., 128.119.40.80/28
  - **10000000.01110111.0101000.0110**0000/28 i.e., 128.119.40.96/28
  - **10000000.01110111.0101000.0111**0000/28 i.e., 128.119.40.112/28

Q5 Datagram Fragmentation

- Suppose datagrams are limited to 1500 bytes (including header) between source Host A and destination Host B. Assume a 20-byte IP header, how many datagrams would be required to send an MP3 consisting of 5 million bytes? Explain the answer.
  - Assume data is carried in TCP segments, with each TCP segment also having 20 byte header.
  - Each 1500-byte datagram can carry 1500-40 = 1460 byte of MP3 data.
  - **Ceil(5000000/1460) = 3425 number of datagrams are needed.**
  - All but the last datagram will be 1500 byte.
  - The last datagram is (5000000-3424*1460)+40=960+40=1000 byte

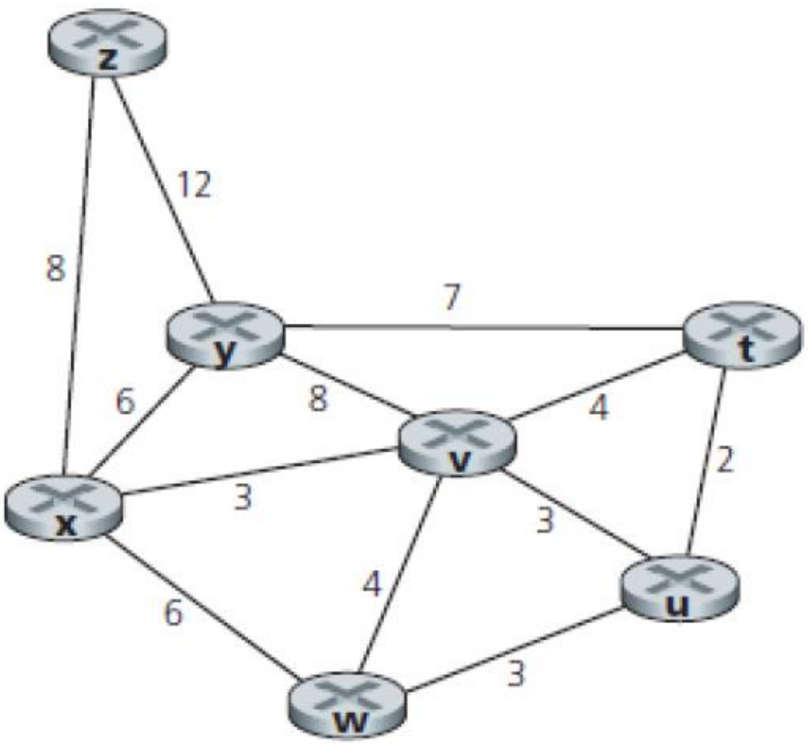
Q6 Link-State Routing Algorithm

- Consider the network. With the link cost indicated, use Dijkstra' shortest path algorithm to compute the shortest path from x to all other nodes. Show how the algorithm works.



Step	*N'*	*D(t),p(t)*	*D(u),p(u)*	*D(v),p(v)*	*D(w),p(w)*	*D(y),p(y)*	*D(z),p(z)*
0	x	∞	∞	3,x	6,x	6,x	8,x

Q6 Link-State Routing Algorithm



Step	N’	t	u	v	w	y	z
0	x	Inf	inf	**3,x**	6,x	6,x	8,x
1	xv	7,v	**6,v**	3,x	6,x	6,x	8,x
2	xvu	7,v	6,v	3,x	**6,x**	6,x	8,x
3	xvuw	7,v	6,v	3,x	6,x	**6,x**	8,x
4	xvuwy	**7,v**	6,v	3,x	6,x	6,x	8,x
5	xvuwyt	7,v	6,v	3,x	6,x	6,x	**8,x**
6	xvuwytz	7,v	6,v	3,x	6,x	6,x	8,x

Thanks!