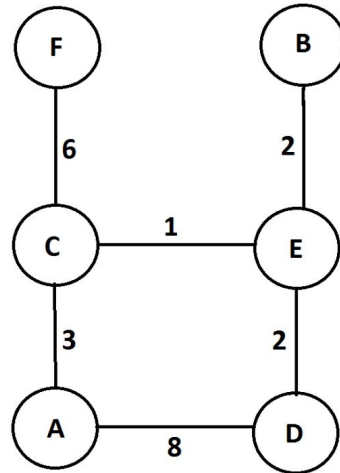


## 1. Distance-vector routing

For the network shown below, provide a global distance-vector table, indicating the distance from each node to all other nodes for each sub-questions.



a. This is how the table looks initially, when each node only knows the distance to its immediate neighbors

Node	Distance to reach each node					
	A	B	C	D	E	F
A	0	Inf	3	8	Inf	Inf
B	inf	0	inf	inf	2	inf
C	3	inf	0	inf	1	6
D	8	inf	inf	0	2	inf
E	inf	2	1	2	0	inf
F	inf	inf	6	inf	inf	0

**b. After the nodes report the information from (a) to its neighbors**

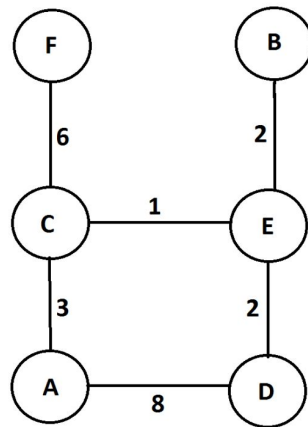
Node	Distance to reach each node					
	A	B	C	D	E	F
A	0	Inf	3	8	4	9
B	inf	0	3	4	2	inf
C	3	3	0	3	1	6
D	8	4	3	0	2	inf
E	4	2	1	2	0	7
F	9	inf	6	inf	7	0

**c. After the nodes once again report the information from (b) to its neighbors**

Node	Distance to reach each node					
	A	B	C	D	E	F
A	0	6	3	6	4	9
B	6	0	3	4	2	9
C	3	3	0	3	1	6
D	6	4	3	0	2	9
E	4	2	1	2	0	7
F	9	9	6	9	7	0

## 2. Link-state routing

How the link-state algorithm would build the routing table for node D. To start you off, the first entry of the table for D has been provided below; use this to get to the next step, and the next, and so on, to get the final routing table for D.

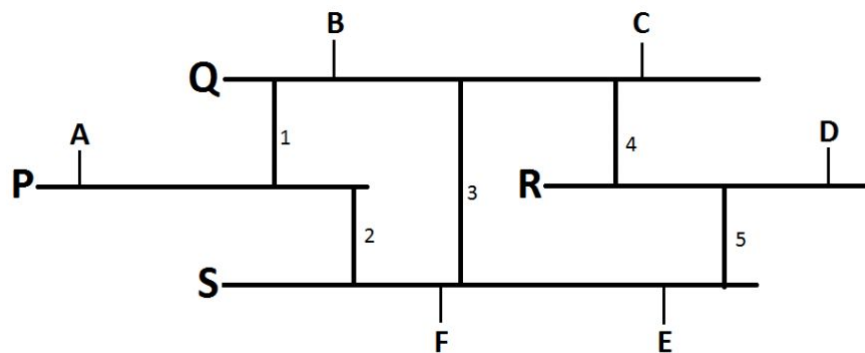


Routing Table for D

Confirmed	Pending
(D, 0, -)	
(D, 0, -)	(A,8,A), (E,2,E)
(D,0,-) (E,2,E)	(A,8,A) (B,4,E) (C,3,E)
(D,0,-) (E,2,E) (C,3,E)	(A,6,E) (B,4,E) (F,9,E)
(D,0,-) (E,2,E) (C,3,E) (B,4,E)	(A,6,E) (F,9,E)
(D,0,-) (E,2,E) (C,3,E) (B,4,E) (A,6,E)	(F,9,E)
(D,0,-) (E,2,E) (C,3,E) (B,4,E) (A,6,E) (F,9,E)	

## Border Gateway Protocol

Consider the network shown below, wherein horizontal lines represent transit providers and numbered vertical lines represent inter-provider links. Assume that networks implement the standard BGP route-selection policy, e.g., preferring shorter AS paths to longer ones. Please provide your answer in format:



<ISP 1, ISP 2, ISP 3...> across links #, #, ...

a. Suppose that each network adopts a policy that, when choosing among egress links to the next AS, outbound traffic is carried as far possible within the current AS (so-called “cold potato routing”). What route and links would packets traverse if:

1. A communicates with B

<P, Q> across link 1

2. E communicates with C

<S, Q> across link 3

3. D communicates with A

<R, Q, P> across links 4 and 1

b. Now, we switch from the “cold potato routing” to “hot potato routing”! It is a policy wherein outbound traffic is routed to the closest link to the next AS in order to minimize the intra-AS cost

1. B communicates with D

<Q, R> across link 4

2. A communicates with F

<P, S> across link 2

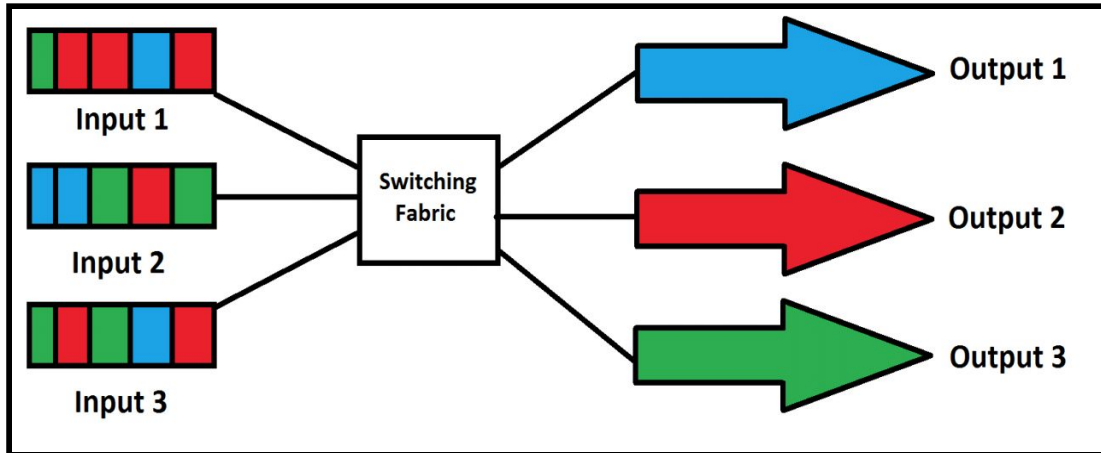
3. D communicates with A

<R, S, P> across link 5 and 2

## Buffering and scheduling

Suppose we have a router with three input ports and three output ports.

- Assume all of the output ports operate at the same speed.
- Assume that the blue-colored packets go out on Output 1, red-colored packets go out on Output 2, and green-colored packets go out on Output 3.



a. Assume that the switching fabric works fast enough such that it is able to take a packet from each input queue and transfer it to the output in one clock cycle. Also assume the scheduling time to be trivial.

Assume a simple scheduling algorithm:

**Step 1:** If input queue 1 is not empty, send packet at the head.

**Step 2:** If input queue 2 is not empty, send packet at the head, unless there's already packet from other input queue occupied the output queue at the same clock cycle

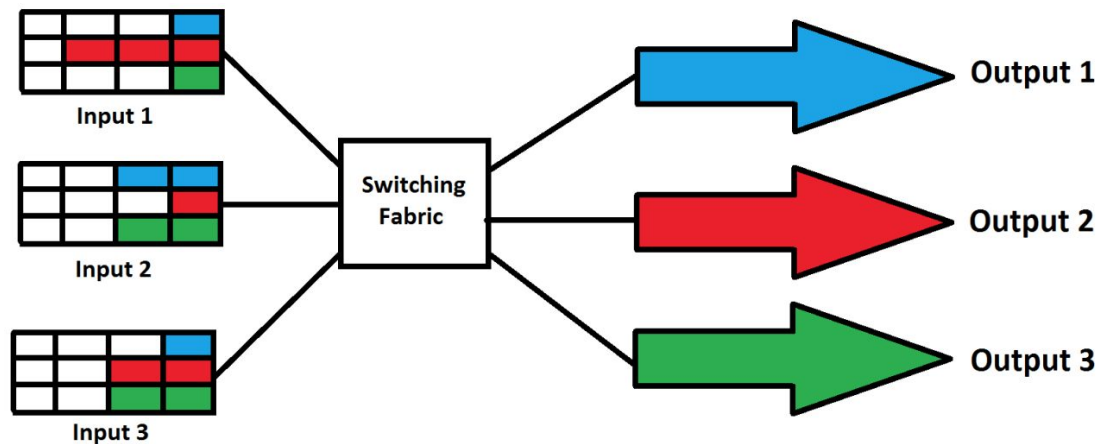
**Step 3:** If input queue 3 is not empty, send packet at the head, unless there's already packet from other input queue occupied the output queue at the same clock cycle

We assume the right most packet in each input queue is the "head" of the queue.

Identify which packet(s) will be transferred at each clock cycle and fill in the table below.

Clock Cycle	1	2	3	4	5	6	7	8	9	10
Input 1 sends	Red	Blue	Red	Red	Green	X	X	X	X	X
Input 2 sends	Green	Red	Green	Blue	Blue	X	X	X	X	X
Input 3 sends	X	X	X	X	Red	Blue	Green	Red	Green	X

(b) Now assume that the input queues make use of a virtual output queue. Also, assume the same scheduling algorithm as mentioned before, except that each input queue has a virtual output queue that will be scheduled in the preference order of blue, red, and green.



(b).1 What problem that is faced in part (a) of the problem does this solve? A one line answer would do.

**This solution solves the head-of-line blocking problem.**

(b).2 Identify which packet(s) will be transferred at each clock cycle and fill in the table below.

Clock Cycle	1	2	3	4	5	6	7	8	9	10
Input 1 sends	Blue	Red	Red	Red	Green	X				
Input 2 sends	Red	Blue	Blue	Green	X	Green				
Input 3 sends	Green	Green	X	Blue	Red	Red				