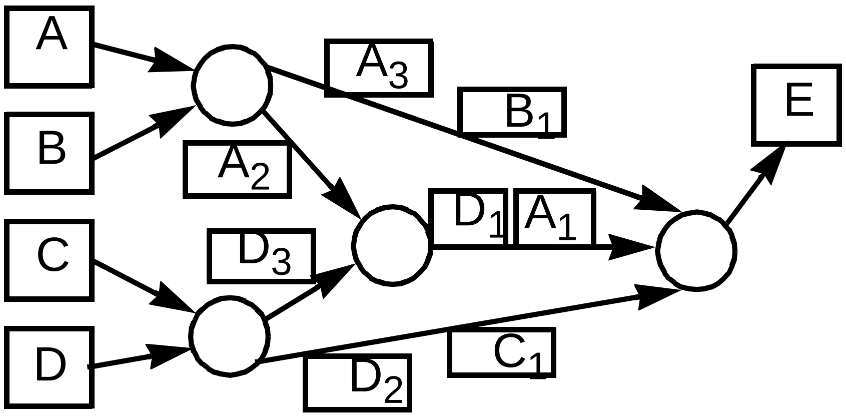
Datagram vs. Virtual Circuit

In a datagram network



M_i: ith datagram sent by Host M.

Two packets of the same user pair can travel along different routes.

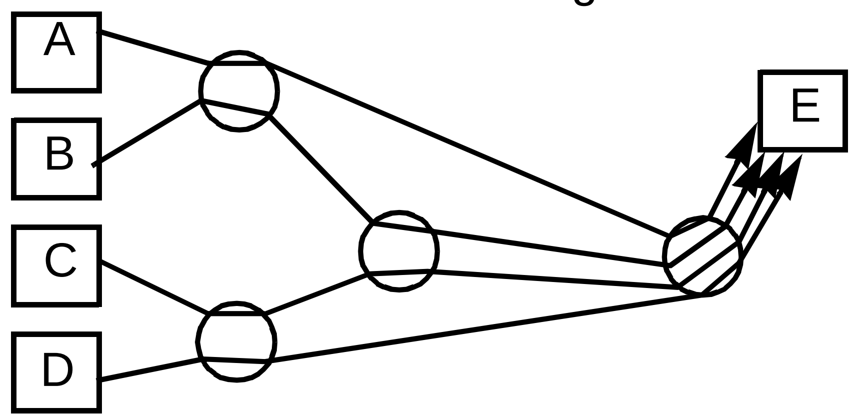
The packets can arrive out of sequence.

Packets contain full Src, Dst addresses.

Each host occupies routine table entries. Requires no connection setup.

In a virtual circuit network

Session Routing



All packets of the same virtual circuit travel along the same path.

Packet sequencing is guaranteed.

Packets contain short VC Id. (VCI).

Each VC occupies routing table entries.

Requires VC setup. First packet has large delay.

Virtual Circuit and Datagram Implementation

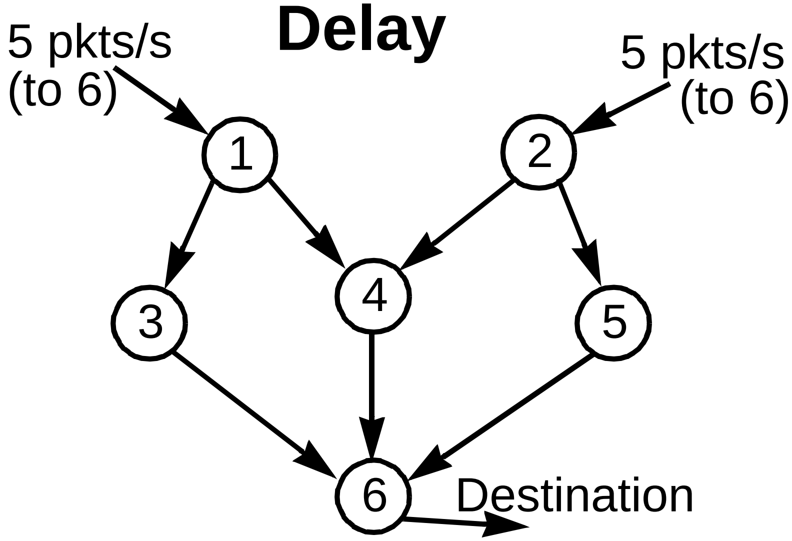
		Internal Operation	
		Datagram	Victual Circuit
External Service	Datagram	UDP over IP (packet)	IP over ATM
	Virtual Circuit	TCP over IP (message, packet)	TYMNET, SNA over ATM (Virtual and explicit route)

Routing Algorithm

- Select routes for various origin-destination (OD) pairs via shortest or optimization calculation (accommodate more OD pairs).
- Delivery of messages to the correct destination once routes are selected.
  $\rightarrow$ Use routing tables.

Performance Measures Affected By Routing

Delay

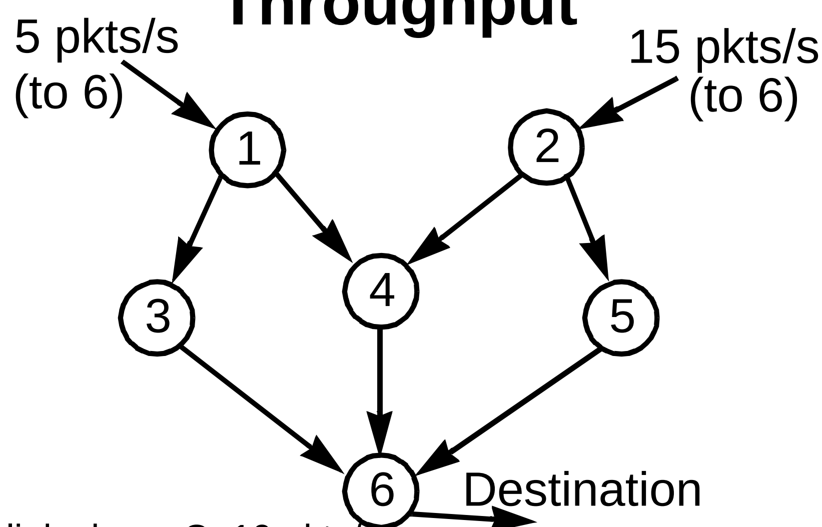


All links have C=10 pkts/s.

If all traffic is via (4,6), congestion occurs.

Via (1,3,6) and (2,5,6), the delay is small.

Throughput



All links have C=10 pkts/s.

Traffic can be accommodated by multi-path routing.

What is the max. throughput from nodes 1and 2 to 6?

How about the worst case?

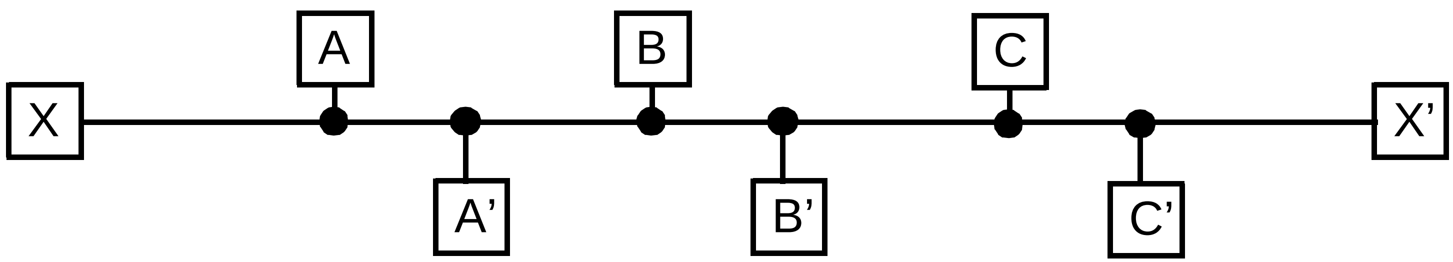
Classification of Routing Algorithms

- Centralized (all routing decisions at a single node) or Distributed (computation of routes shared by nodes)
- Static (routes are fixed for each OD pair regardless of traffic pattern) or Adaptive (responsive to traffic pattern)

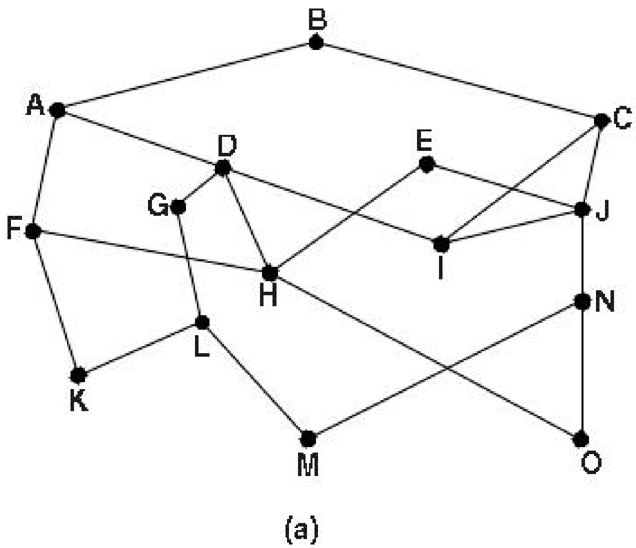
Desirable Properties of Routing Algorithms

- correctness
- simplicity
- robustness
- stability
- fairness
- optimality

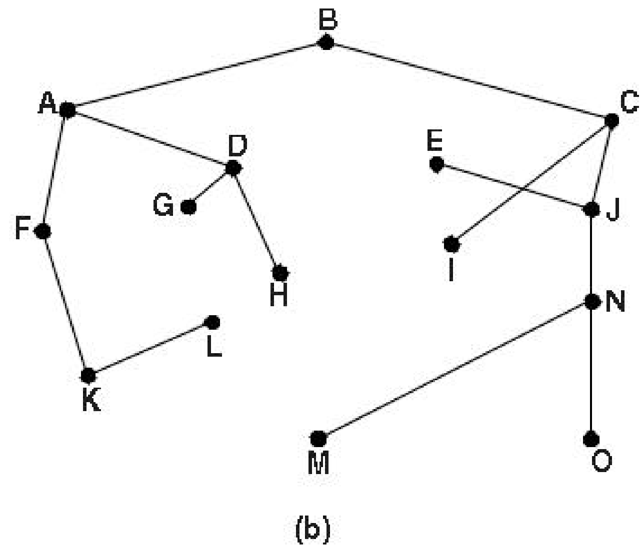
(fairness ← conflicts → optimality)



(Routing) Optimal Principle



A subnet



sink tree for router B

- If router J is on the optimal path from route I to route K, then the optimal path from J to K also falls along the same route.
- The optimal routes to a router form a sink tree.

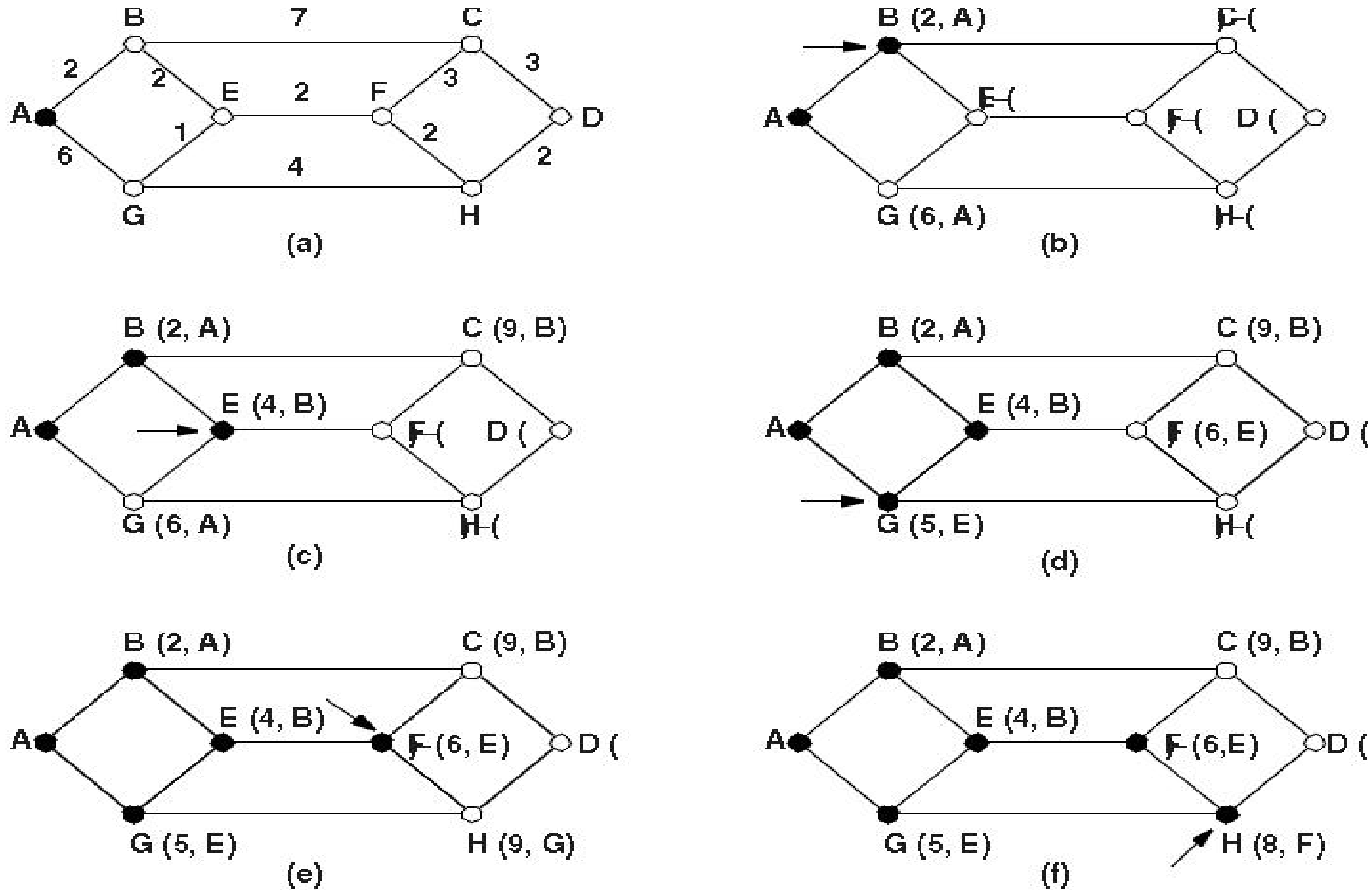
Dijkstra's Shortest Path Algorithm [DIJK59]

Find the shortest paths form a given source node to all other nodes by developing paths in order of increasing path length.

Let the set of nodes in a network be N.

1. start with a source node S in node set M. Let the nodes not in M be M'.
2. Let L with the set of links connecting M and M'.
   - Among the nodes in M' connected to M via L, find the node, n, with the lowest path cost to s. Move n to M. (Use hopcount, nodeId for tie breaker.)
   - Update the cost from s to other nodes in M' taking into consideration of the new path via n.
3. Repeat step 2 until M=N.

Computing Shortest Path



Routing Strategies

- Fixed routing
  Network Operation Center (NOC) collects information from individual nodes
  NOC carries out the least cost routing algorithms.
  NOC distributed the routing information to individual nodes.
  The above steps are carried out periodically.
- Flooding
  A node sends/relays a message along all its outgoing links.
  Rely on hop counts or time-stamps to terminate the flooding.
  Disadvantage: a lot of redundant msgs, waste bandwidth.
  Advantage: does not require NOC, reliable,
  message may arrive via a minimum hop route.

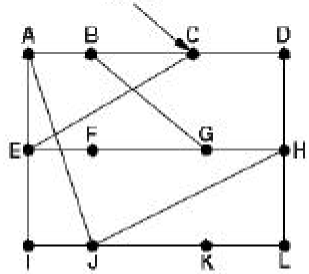
Routing in ARPANET (old version)

Use distributed, shortest path, and adaptive routing algorithm.

Periodically, each IMP

- calculates delay to other IMPs,
- exchanges delay vectors with their neighbors,
- based on the delay vectors received, compute the new routing table.

Router



(a)

To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6	New routing table for J	

Vectors received from J's four neighbors

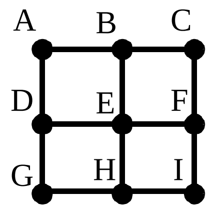
(b)

Problems: 1) low throughput, 2) susceptible to oscillations,
3) good news spread fast; bad news spread slow.

Exercise on ARPANET Routing

Consider the mesh network shown below.

Assume that the ARPAnet routing algorithm is used. Node B receives three routing vectors from



	A	E	C	New estimated delay from B	via Line
A	0	6	10		
B	21	4	5		
C	25	13	0		
D	40	7	8		
E	24	0	10		
F	29	5	9		
G	48	21	26		
H	27	14	15		
I	31	19	11		
	BA delay =50	BE delay = 3	BC delay =5		

A, E and C. With the above link delay to A, E, and C, calculate B’s new routing table and fill your result into the above vacant entries.

ARPANET Routing 1979

The old ARPANET routing algorithm has the following shortcomings:

- only consider queue length, not favor high speed lines.
- it responded slowly to congestion and delay increases.

The new ARPANET routing algorithm

- measure the delay directly using time-stamp.
- every 10 seconds, delay on each link is calculated and flood to all other nodes (not just the neighboring node).
- each node compute the new routing information based one new delay info and Dijkstra algorithm.
- it is more responsive.
- but new problems appear (Oscillations are possible and very damaging)

Count to Infinity Problem

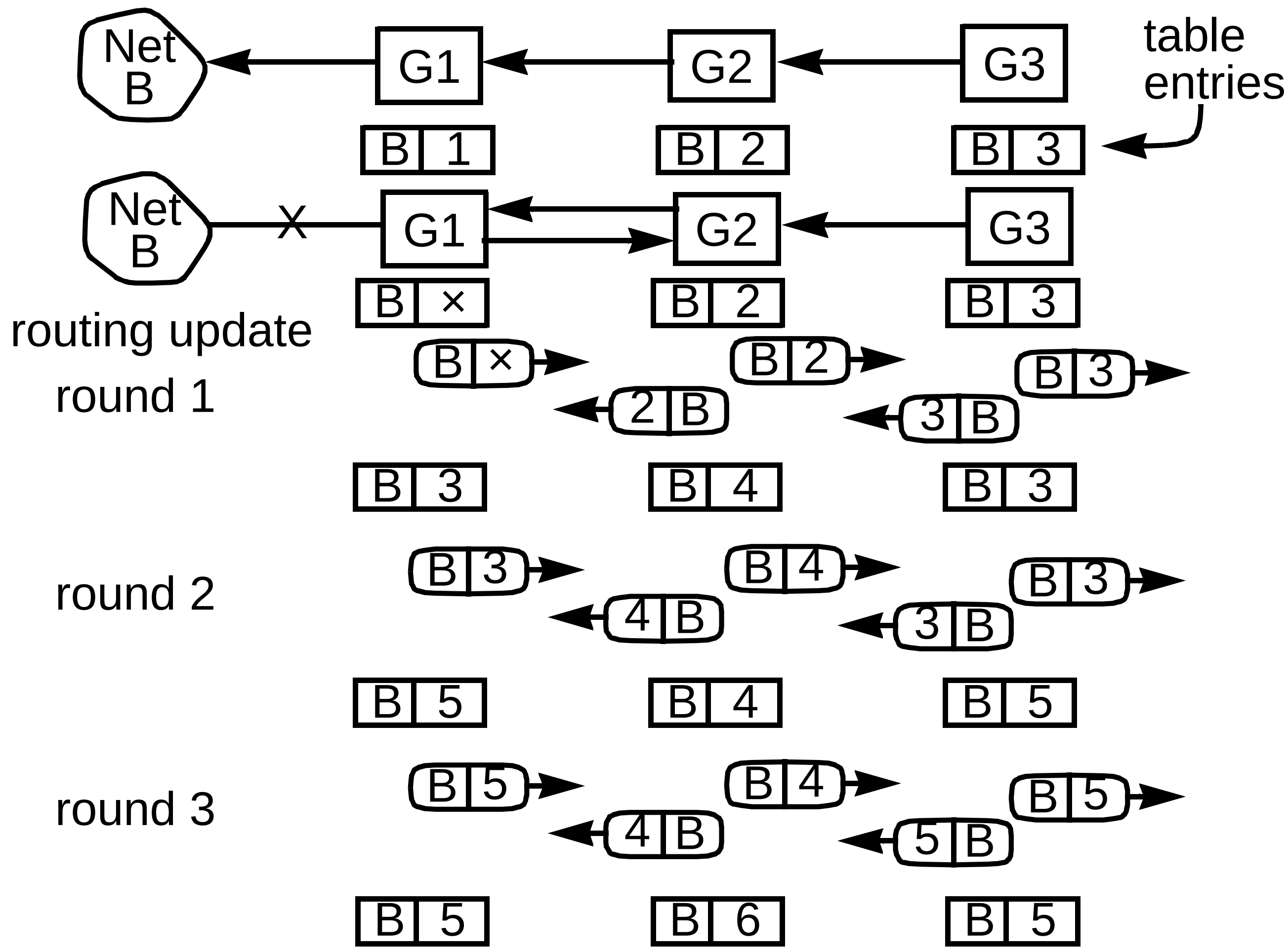
A	B	C	D	E	
					Initially
	1				After 1 exchange
	1	2			After 2 exchanges
	1	2	3		After 3 exchanges
	1	2	3	4	After 4 exchanges

(a)

A	B	C	D	E	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
		⋮			

(b)

Slow Convergence



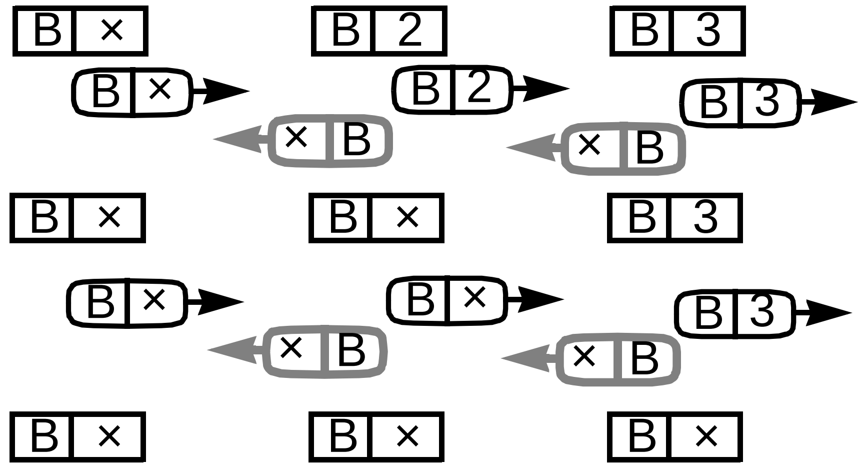
Good news travels quickly; bad news travels slowly

Split Horizon Update

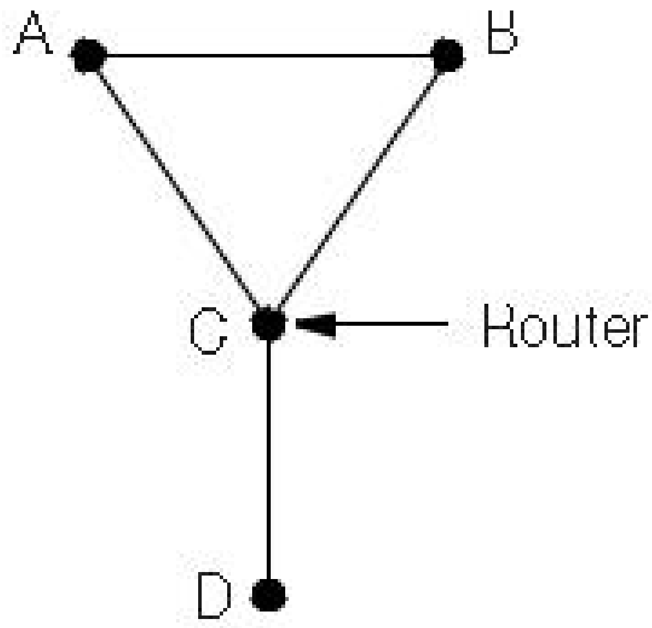
routing update

round 1

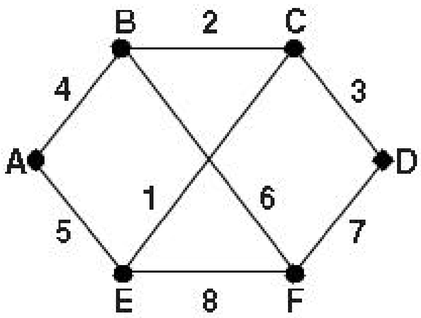
round 2



Problem can't be solved by Split Horizon



Link State Routing



(a)

Link State Packets

A		B		C		D		E		F	
Seq.		Seq.		Seq.		Seq.		Seq.		Seq.	
Age		Age		Age		Age		Age		Age	
B	4	A	4	B	2	C	3	A	5	B	6
E	5	C	2	D	3	F	7	C	1	D	7
		F	6	E	1			F	8	E	8

(b)

Oscillation in new ARPANET Routing

Oscillations are possible and very damaging.

Oscillation in ARPANET Routing

Damping the Oscillation in ARPANET Routing

Oscillations can be damped by using

1) A large bias factor (a constant that make link lengths large at zero flow).

   Side-effect: makes it less sensitive to congestion.

2) Average of several routing updates.

3) Asynchronous execution of the routing algorithm.

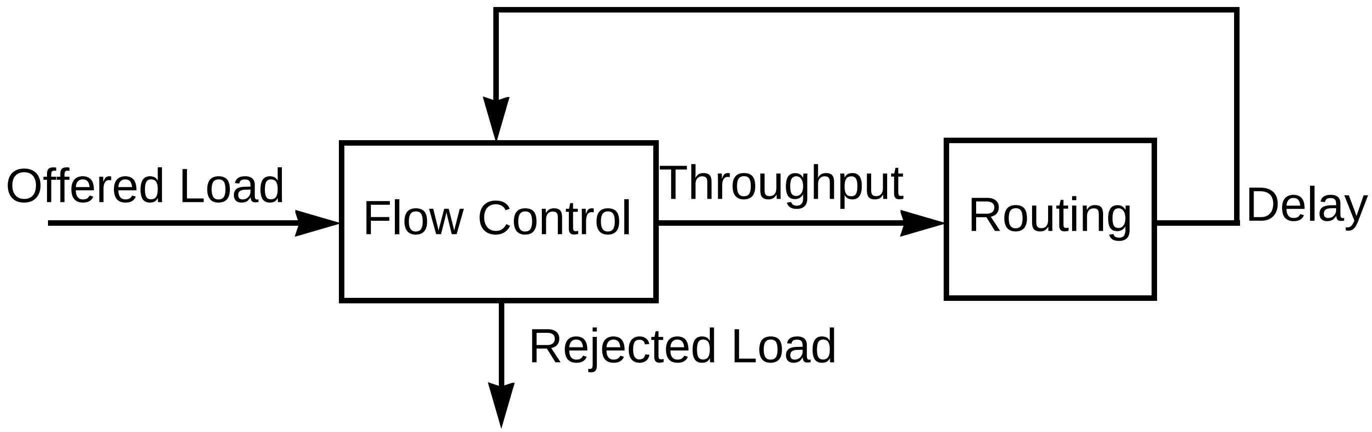
Traffic control

Deal with the control of the number of packets entering the network.

- Flow control- enable the receiver to control the msg receiving rate.
  e.g., use slide-window protocol.
- Congestion control- maintaining the number of packets within the network below the level where the network throughput starts to decrease.
  1. congested node sends choke packets to the sources.
  2. rely routing algorithms (passive)
  3. use the probing message between end points.
  4. piggyback the congestion information back to the source.
- Deadlock avoidance.
  - use structured buffer pool
  - use setup packet to reserve enough buffer and use ack msg to release buffer.

Interaction of Routing and Flow Control

As good routing keeps delay low, flow control allows more traffic into the network.



Routing determines the delay/throughput curve along which flow control operates