

Packet Switching

Raj Jain

Professor of CIS

The Ohio State University

Columbus, OH 43210

Jain@cis.ohio-state.edu

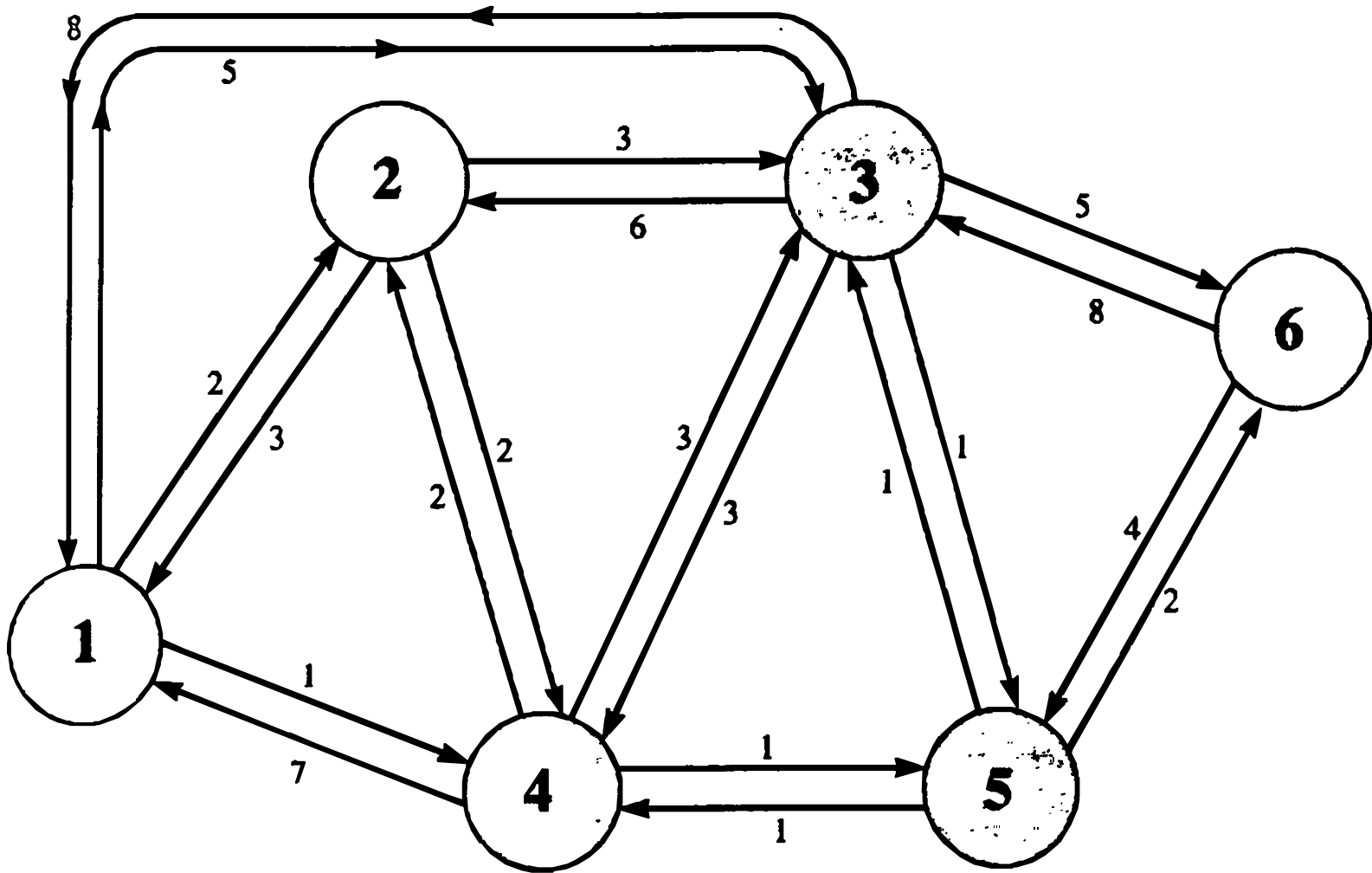
This presentation is available on-line at:

<http://www.cis.ohio-state.edu/~jain/cis677-98/>



- q Routing algorithms
- q ARPAnet routing

Routing



Rooting or Routing

- q *Rooting* is what fans do at football games, what pics do for truffles under oak trees in the Vaucluse, and what nursery workers intent on propagation do to cuttings from plants.
- q *Routing* is how one creates a beveled edge on a table top or sends a corps of infanctrymen into full scale, disorganized retreat

Ref: Piscitello and Chapin, p413

Routeing or Routing

- q Routeing: British
- q Routing: American
- q Since Oxford English Dictionary is much heavier than any other dictionary of American English, British English generally prevails in the documents produced by ISO and CCITT; wherefore, most of the international standards for routing standards use the routeing spelling.

Ref: Piscitello and Chapin, p413

Rooting or Routing

- q *Rooting* is what fans do at football games, what pics do for truffles under oak trees in the Vaucluse, and what nursery workers intent on propagation do to cuttings from plants.
- q *Routing* is how one creates a beveled edge on a table top or sends a corps of infanctrymen into full scale, disorganized retreat

Ref: Piscitello and Chapin, p413

Routeing or Routing

- q Routeing: British
- q Routing: American
- q Since Oxford English Dictionary is much heavier than any other dictionary of American English, British English generally prevails in the documents produced by ISO and CCITT; wherefore, most of the international standards for routing standards use the routeing spelling.

Ref: Piscitello and Chapin, p413

Routing Techniques Elements

- q **Performance criterion:** *Hops*, Distance, *Speed*, Delay, Cost
- q **Decision time:** *Packet*, session
- q **Decision place:** *Distributed*, centralized, Source
- q **Network information source:** None, local, *adjacent nodes*, nodes along route, all nodes
- q **Routing strategy:** Fixed, *adaptive*, random, flooding
- q **Adaptive routing update time:** Continuous, *periodic*, topology change, major load change

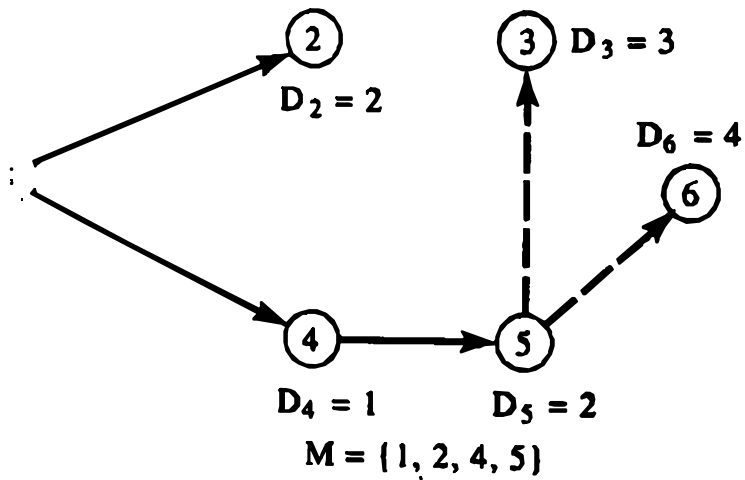
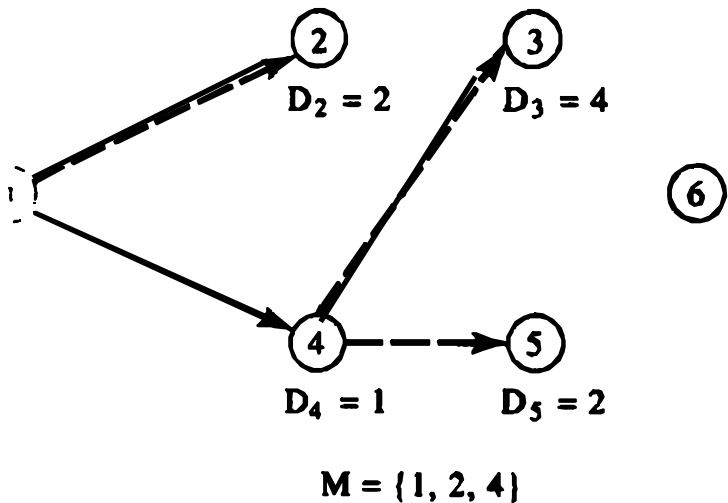
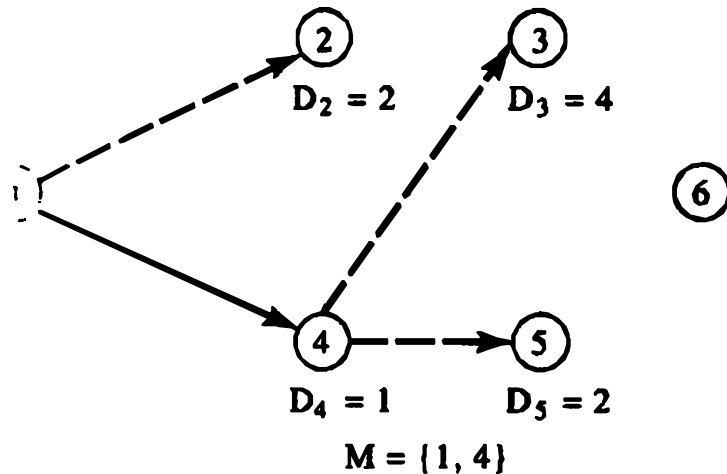
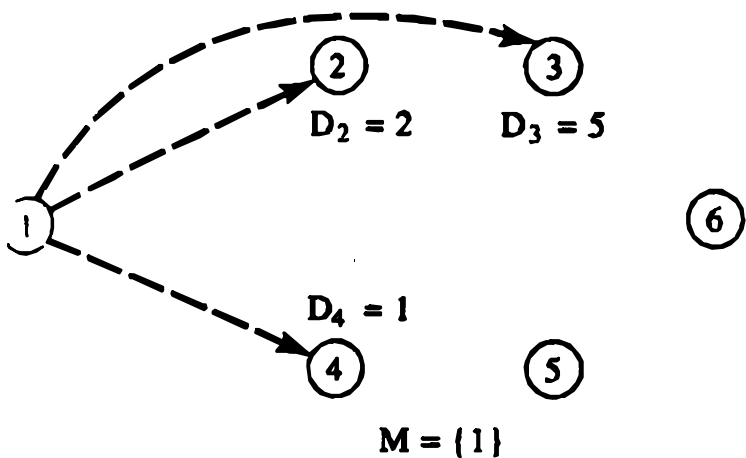
Distance Vector vs Link State

- q **Distance Vector:** Each router sends a vector of distances to its neighbors. The vector contains distances to all nodes in the network. Older method. Count to infinity problem.
- q **Link State:** Each router sends a vector of distances to all nodes. The vector contains only distances to neighbors. Newer method. Used currently in internet.

Dijkstra's Algorithm

- q Goal: Find the least cost paths from a given node to all other nodes in the network
- q Notation:
 - d_{ij} = Link cost from i to j if i and j are connected
 - D_n = Total path cost from s to n
 - M = Set of nodes so far for which the least cost path is known
- q Method:
 - q Initialize: $M = \{s\}$, $D_n = d_{sn}$
 - q Find node $w \notin M$, whose D_n is minimum
 - q Update D_n

Example

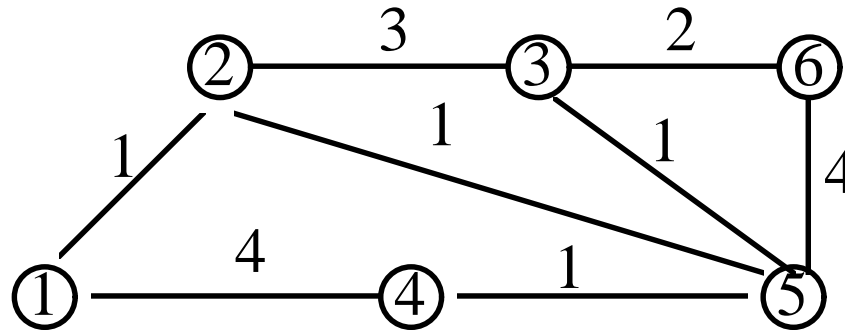


Example (Cont)

	M	D2	Path	D3	Path	D4	Path	D5	Path	D6	Path
1	{1}	2	1-2	5	1-3	1	1-4	∞	-	∞	-
2	{1,4}	2	1-2	4	1-4-3	1	1-4	2	1-4-5	∞	-
3	{1,2,4}	2	1-2	4	1-4-3	1	1-4	2	1-4-5	∞	-
4	{1,2,4,5}	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
5	{1,2,3,4,5}	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
6	{1,2,3,4,5,6}	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6

Dijkstra's Routing Algorithm

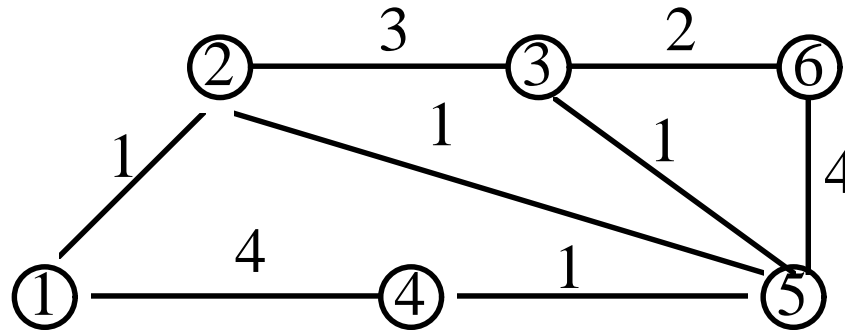
q Apply to the following network and compute paths from node 1.



	M	D2	Path	D3	Path	D4	Path	D5	Path	D6	Path
1											
2											
3											
4											
5											
6											

Dijkstra's routing algorithm

q Apply to the following network and compute paths from node 1.



	M	D2	Path	D3	Path	D4	Path	D5	Path	D6	Path
1	{1}	1	1-2	∞	-	4	1-4	∞	-	∞	-
2	{1,2}	1	1-2	4	1-2-3	4	1-4				
3	{1,2,3}	1	1-2	4	1-2-3	4	1-4	2	1-2-5	6	1-2-3-6
4	{1,2,3,5}	1	1-2	4	1-2-3	3	1-2-5-1	2	1-2-5	6	1-2-3-6
5	{1,2,3,4,5}	1	1-2	4	1-2-3	3	1-2-5-1	2	1-2-5	6	1-2-3-6
6	{1,2,3,4,5,6}	1	1-2	4	1-2-3	3	1-2-5-1	2	1-2-5	6	1-2-3-6

Bellman-Ford Algorithm

q Notation:

h = Number of hops being considered

$D_n^{(h)}$ = Cost of h -hop path from s to n

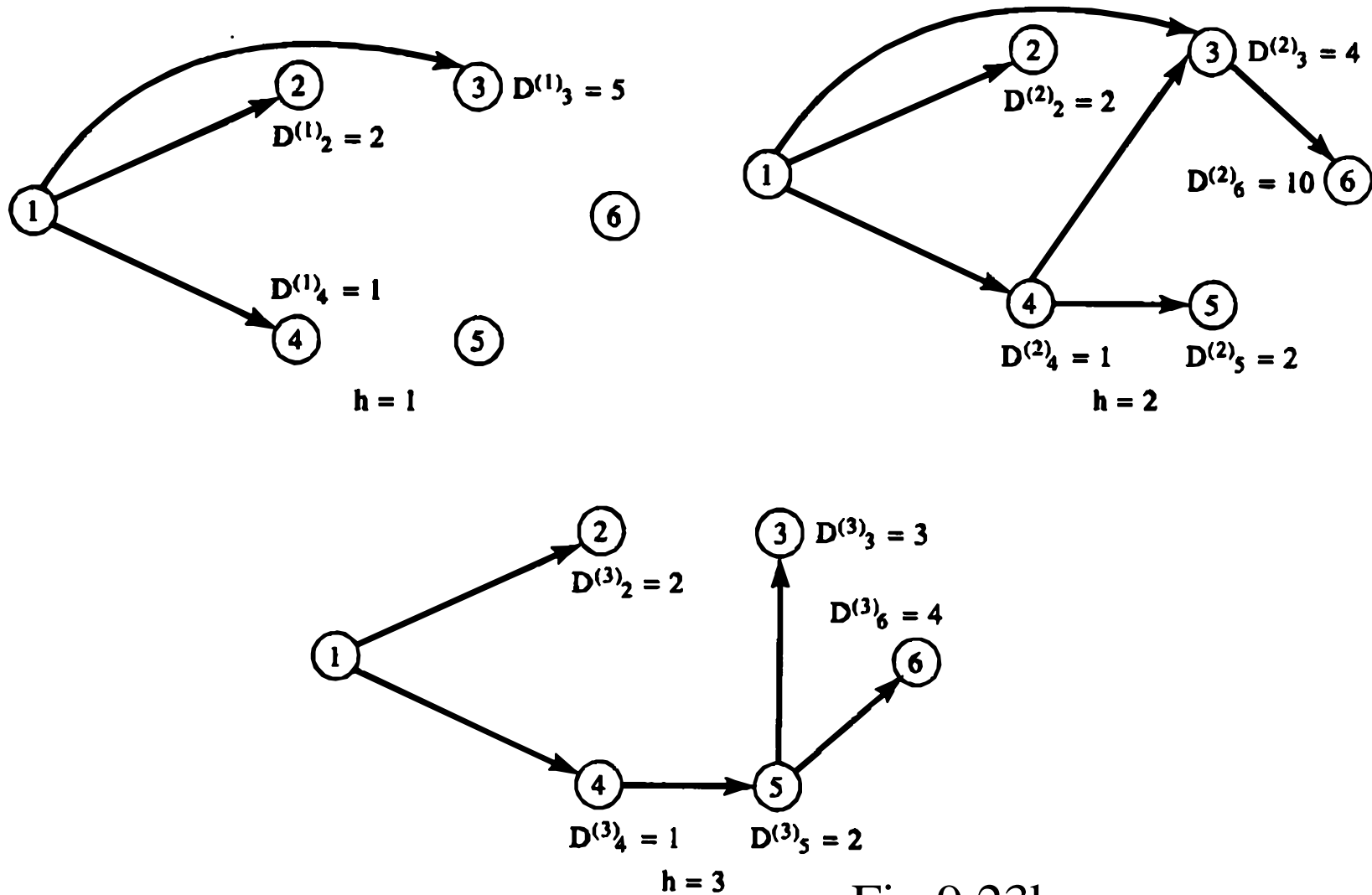
q Method: Find all nodes 1 hop away
Find all nodes 2 hops away
Find all nodes 3 hops away

q Initialize: $D_n^{(h)} = \infty$ for all $n \neq s$; $D_n^{(h)} = 0$ for all h

q Find j th node for which $h+1$ hops cost is minimum

$$D_n^{(h+1)} = \min_j [D_j^{(h)} + D_{jn}]$$

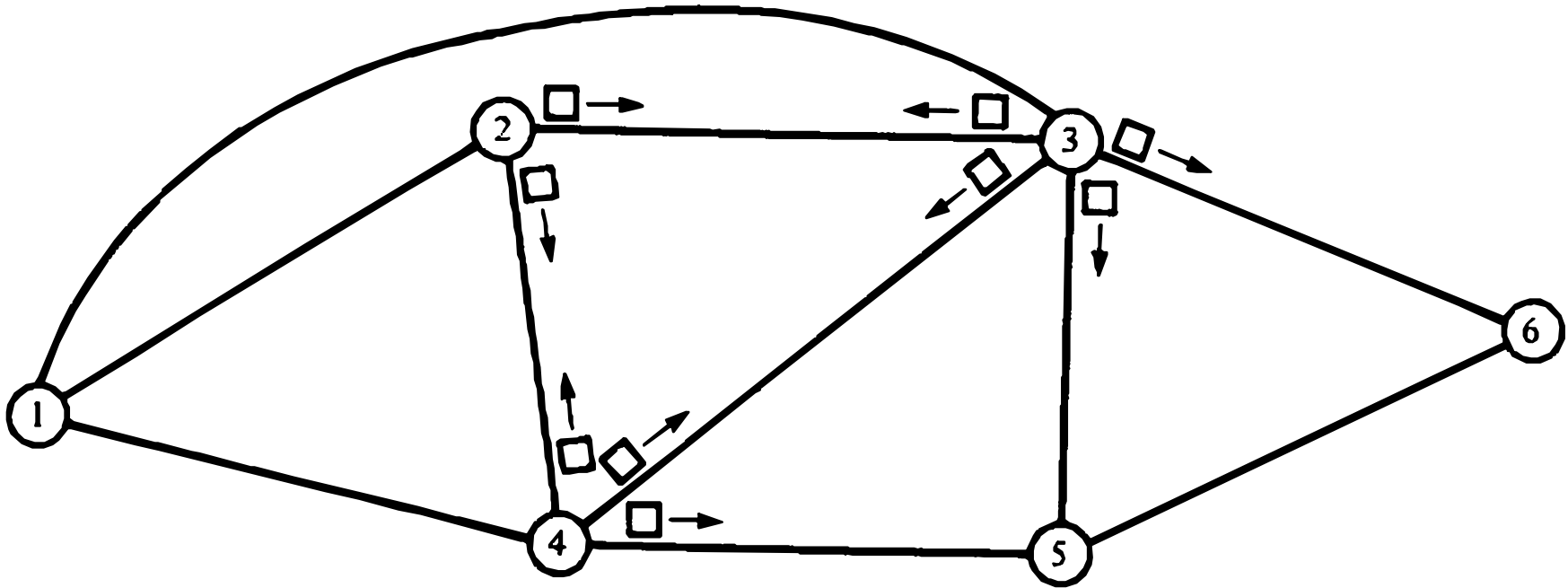
Example



Example (Cont)

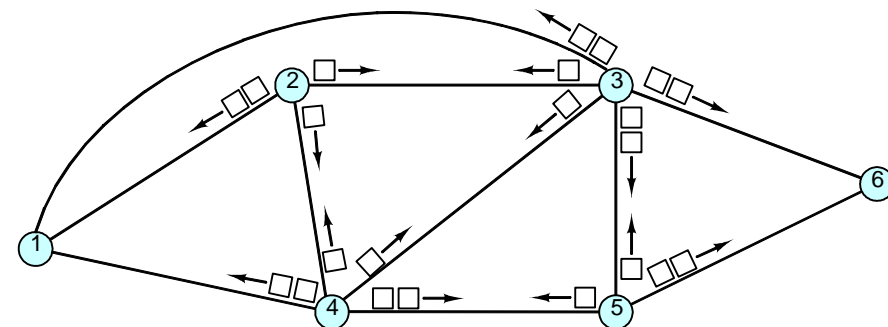
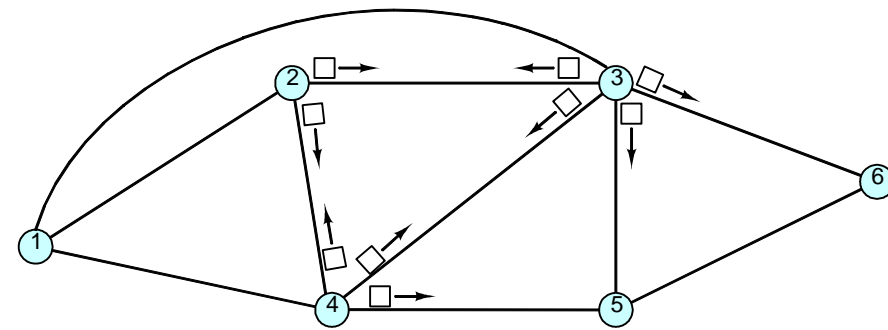
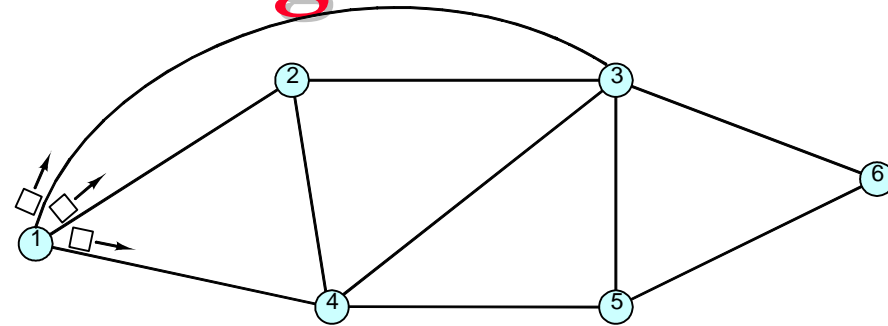
h	D(h ₂)	Path	D(h ₃)	Path	D(h ₄)	Path	D(h ₅)	Path	D(h ₆)	Path
0	∞	-	∞	-	∞	-	∞	-	∞	-
1	2	1-2	5	1-3	1	1-4	∞	-	∞	-
2	2	1-2	4	1-4-3	1	1-4	2	1-4-5	10	1-3-6
3	2	1-2	3	1-5-4-3	1	1-4	2	1-4-5	4	1-4-5-6
4	2	1-2	3	1-5-4-3	1	1-4	2	1-4-5	4	1-4-5-6

Flooding



Flooding

- q Uses all possible paths
- q Uses minimum hop path Used for source routing



ARPAnet Routing (1969-78)

- q Features: Cost=Queue length,
- q Each node sends a vector of costs (to all nodes) to neighbors. Distance vector
- q Each node computes new cost vectors based on the new info using Bellman-Ford algorithm

ARPAnet Routing Algorithm

Desti- Next
nation Delay node

1	0	\tilde{N}
2	2	2
3	5	3
4	1	4
5	6	3
6	8	3

$\underbrace{\hspace{10em}}_{D^1 \quad S^1}$

(a) Node 1's routing table before update

2	3	1
0	3	2
3	0	2
2	2	0
3	1	1
5	3	3

$\underbrace{\hspace{10em}}_{D^2 \quad D^3 \quad D^4}$

(b) Delay vectors sent to neighbor nodes

Desti- Next
nation Delay node

1	0	\tilde{N}
2	2	2
3	3	4
4	1	4
5	2	4
6	4	4

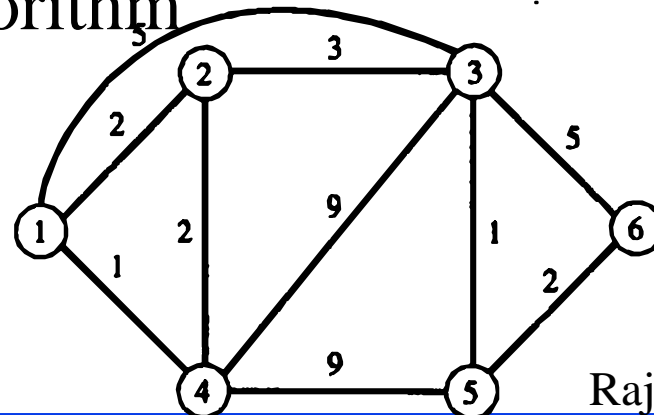
$1_{1,2} = 2$
 $1_{1,3} = 5$
 $1_{1,4} = 1$

(c) Node 1's routing table after update and link c

Fig 9.9

ARPAnet Routing (1979-86)

- q Problem with earlier algorithm: Thrashing (packets went to areas of low queue length rather than the destination), Speed not considered
- q Solution: Cost=Measured delay over 10 seconds
- q Each node floods a vector of cost to neighbors.
Link-state. Converges faster after topology changes.
- q Each node computes new cost vectors based on the new info using Dijkstra's algorithm



ARPAnet Routing (1987+)

- q Problem with 2nd Method: Correlation between delays reported and those experienced later : High in light loads, low during heavy loads
 - ⇒ Oscillations under heavy loads
 - ⇒ Unused capacity at some links, over-utilization of others, More variance in delay more frequent updates
 - More overhead

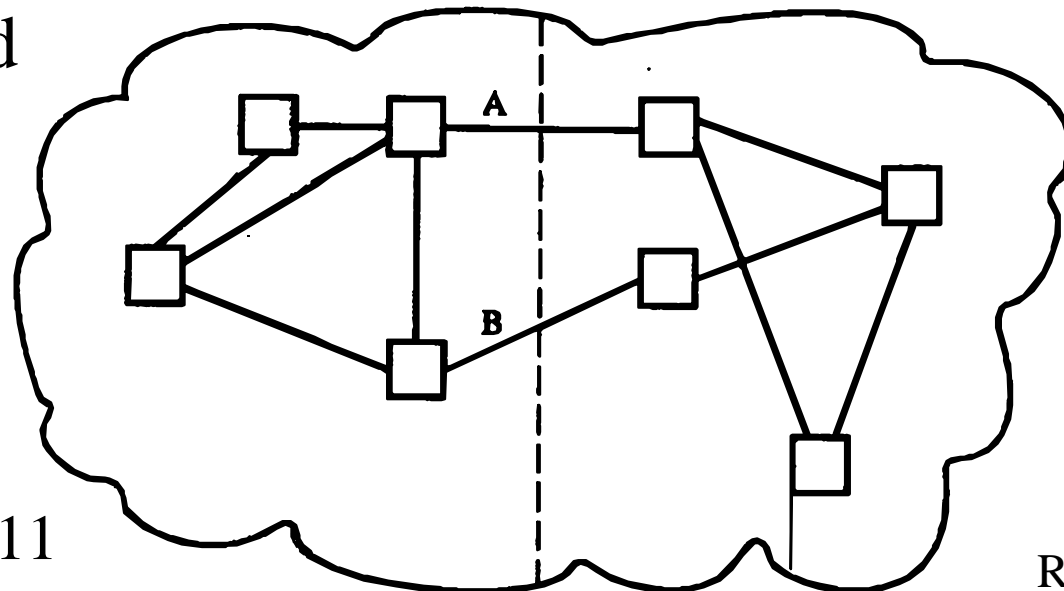


Fig 9.11

Routing Algorithm

- q Delay is averaged over 10 s
- q Link utilization = $r = 2(s-t)/(s-2t)$
where t =measured delay,
 s =service time per packet (600 bit times)
- q Exponentially weighted average utilization
 $U(n+1) = \alpha U(n) + (1-\alpha)r(n+1)$
 $= 0.5 U(n) + 0.5 r(n+1)$ with $\alpha = 0.5$
- q Link cost = $fn(U)$

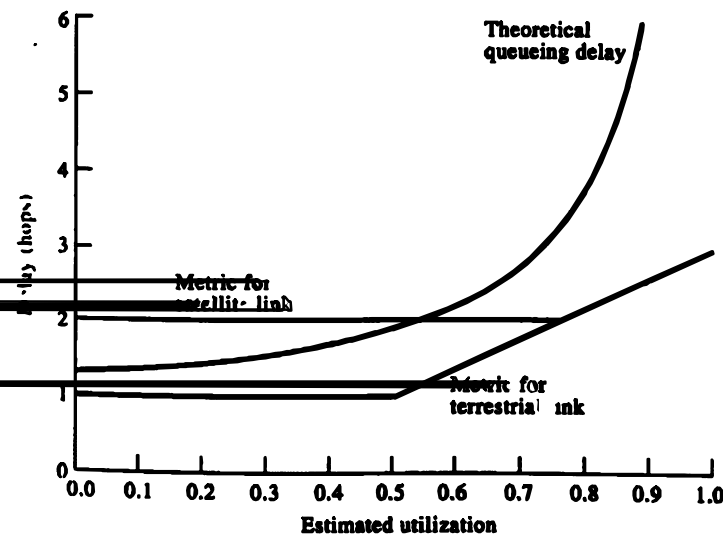
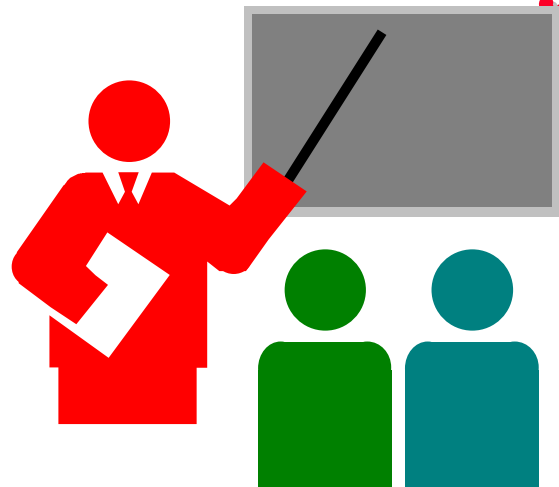


Fig 9.12

Summary



- q Connection-oriented and Connectionless
- q Routing: Least-cost, Flooding, Adaptive
- q Dijkstra's and Bellman-Ford algorithms
- q ARPAnet

Homework

- q Exercise 9.4 (in b assume a unidirectional single loop), 9.10, 9.15
- q Due: Next class