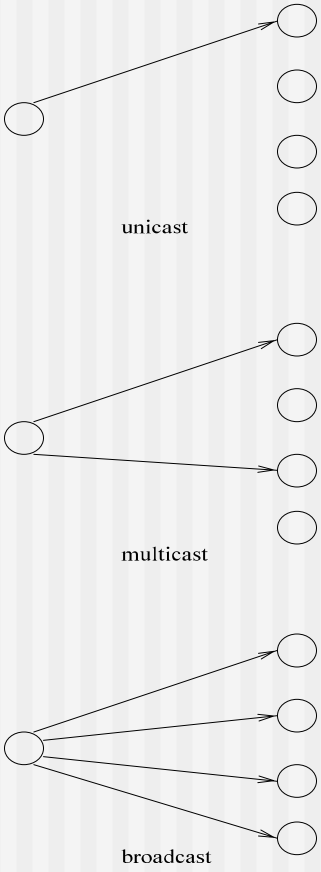
Table of Contents

Distributed Communication

One-to-one (**unicast**)

One-to-many (**multicast**)

One-to-all (**broadcast**)

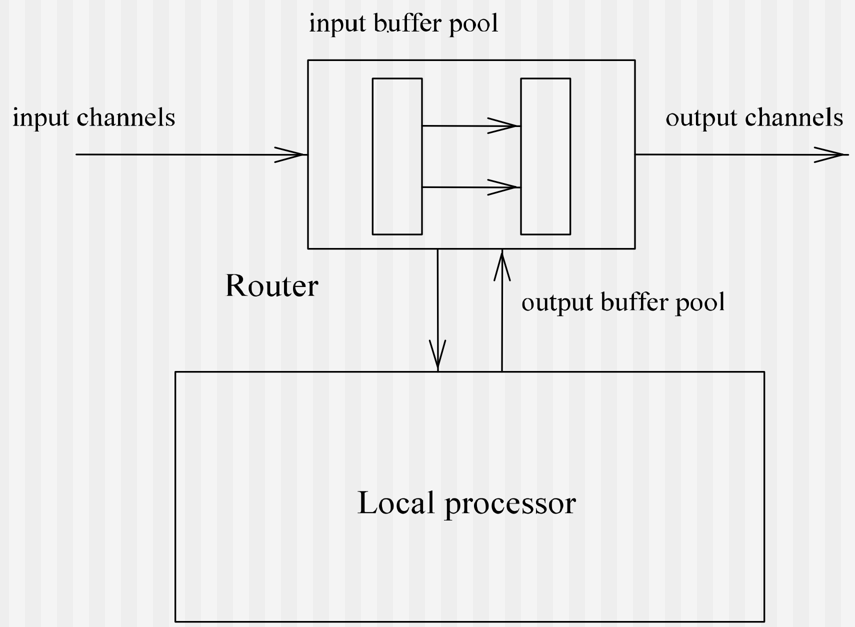


Different types of communication

Classification

- Special purpose vs. general purpose.
- Minimal vs. nonminimal.
- Deterministic vs. adaptive.
- Source routing vs. distributed routing.
- Fault-tolerant vs. non fault-tolerant.
- Redundant vs. non redundant.
- Deadlock-free vs. non deadlock-free.

Router Architecture



A general PE with a separate router.

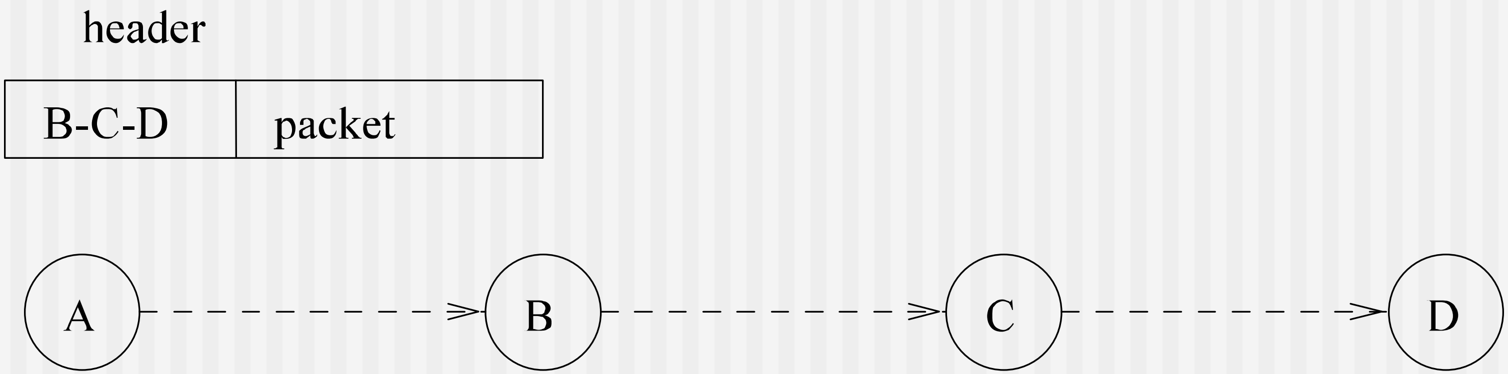
Four Factors for Communication Delay

- **Topology**. The topology of a network, typically modeled as a graph, defines how PEs are connected.
- **Routing**. Routing determines the path selected to forward a message to its destination(s).
- **Flow control**. A network consists of channels and buffers. Flow control decides the allocation of these resources as a message travels along a path.
- **Switching**. Switching is the actual mechanism that decides how a message travels from an input channel to an output channel: store-and-forward and cut-through (wormhole routing).

General-Purpose Routing

Source routing: link state (Dijkstra's algorithm)

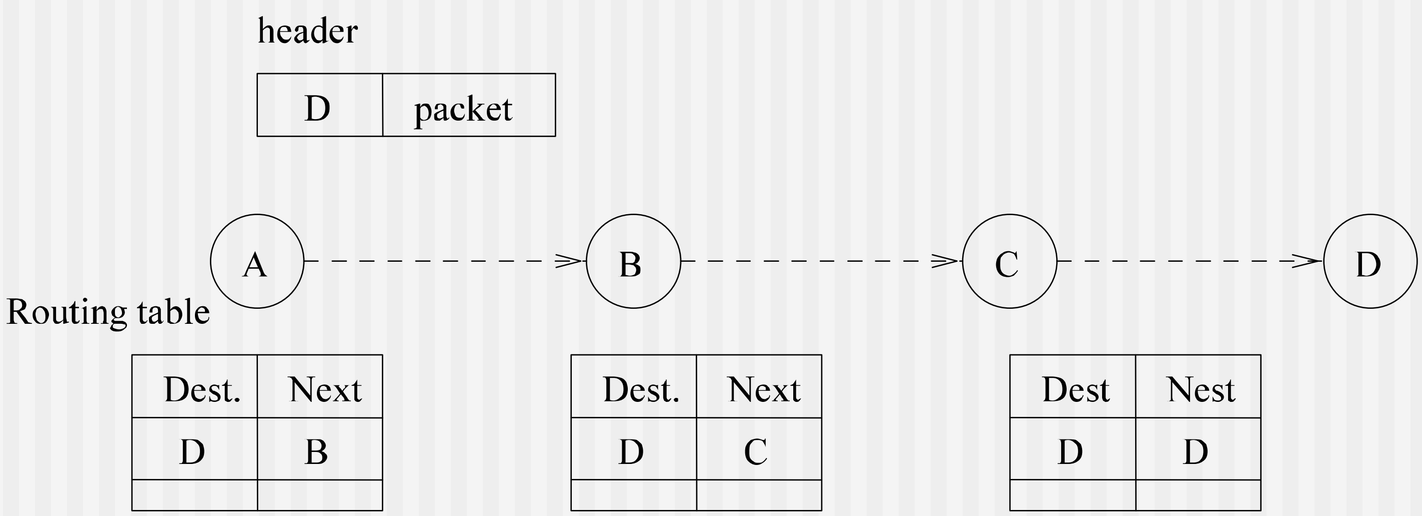
Used in Internet protocol: Open Shortest Path First (OSPF)



A sample source routing

General-Purpose Routing (Cont'd)

Distributed routing: distance vector (Bellman-Ford algorithm)
Used in Internet protocol: Routing Information Protocol (RIP) and Interior Gateway Routing Protocol (IGRP)



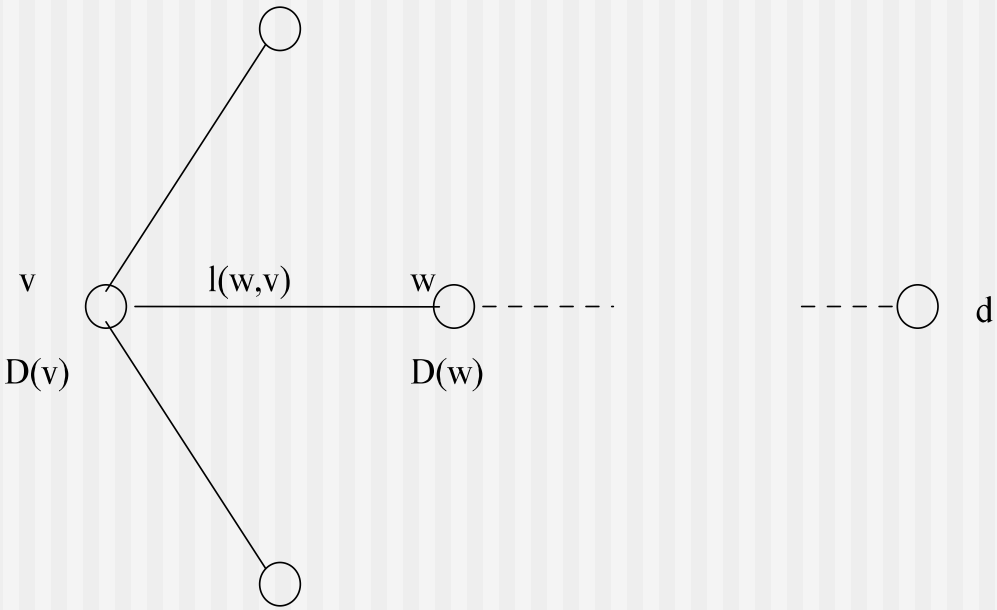
A sample distributed routing

Distributed Bellman-Ford Routing Algorithm

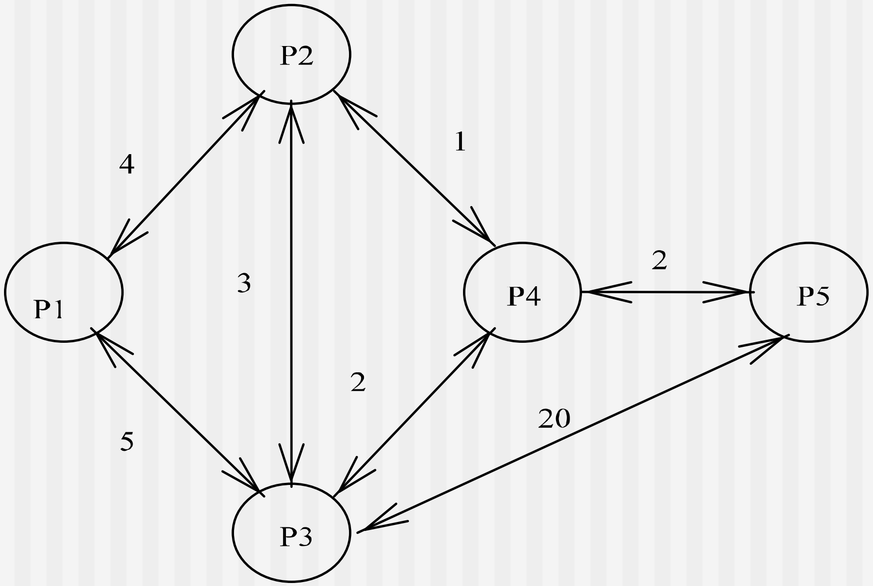
- *Initialization*. With node *d* being the destination node, set $D(d) = 0$ and label all other nodes $(., \infty)$.
- *Shortest-distance labeling of all nodes*. For each node $v \neq d$ do the following: Update $D(v)$ using the current value $D(w)$ for each neighboring node w to calculate $D(w) + l(w, v)$ and perform the following update:

$$D(v) := \min\{D(v), D(w) + l(w; v)\}$$

Distributed Bellman-Ford Algorithm (Cont'd)



Example 18



A sample network.

Example 18 (Cont'd)

Round	P1	P2	P3	P4
Initial	(., ∞)	(., ∞)	(., ∞)	(., ∞)
1	(., ∞)	(., ∞)	(5,20)	(5,2)
2	(3,25)	(4,3)	(4,4)	(5,2)
3	(2,7)	(4,3)	(4,4)	(5,2)

Bellman-Ford algorithm applied to the network with P_5 being the destination.

Looping Problem

Link (P_4; P_5) fails at the destination P_5.

Time next node	0	1	2	3	K, 4<k<15	16	17	18	19	(20, ∞)
P2	7	7	9	9	$2\lfloor n/2 \rfloor +7$	23	23	25	25	27
P3	9	9	11	11	$2\lfloor n/2 \rfloor +9$	25	25	25	25	25*

(a) Network delay table of P1

Time next node	0	1	2	3	K, 4<k<15	16	17	18	19	(20, ∞)
P1	11	11	13	13	$2\lfloor n/2 \rfloor +9$	25	27	27	29	29
P3	7	7	9	9	$2\lfloor n/2 \rfloor +7$	23	23	23	23	23
P3	3	5	5	7	$2\lfloor n/2 \rfloor +3$	19	21	21	23*	23

(b) Network delay table of P2

Looping Problem (Cont'd)

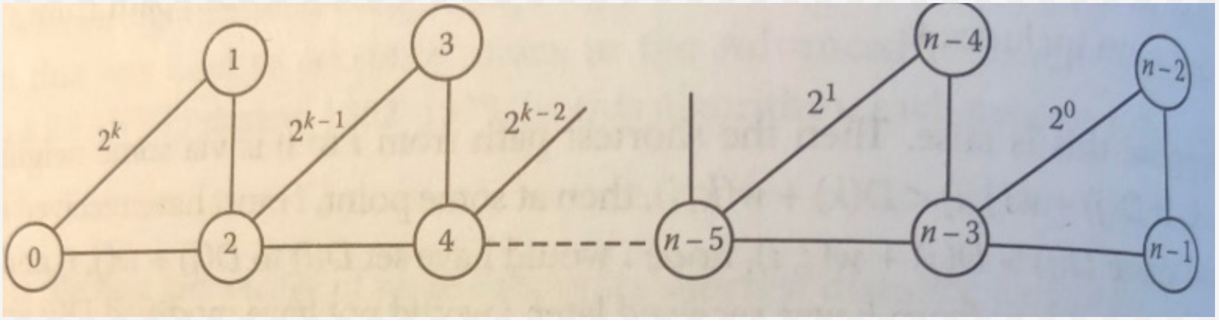
Time next node	0	1	2	3	K, $4<k<15$	16	17	18	19	$(20, \infty)$
P1	12	12	12	14	$2\lfloor n/2 \rfloor +10$	26	28	28	30	30
P2	6	6	8	8	$2\lfloor n/2 \rfloor +5$	22	22	24	24	26
P4	4	6	6	8	$2\lfloor n/2 \rfloor +4$	20	22	22	24	24
P5	20	20	20	20	20	20	20*	20	20	20

(c) Network delay table of P3

Time next node	0	1	2	3	K, $4<k<15$	16	17	18	19	$(20, \infty)$
P2	4	4	6	6	$2\lfloor n/2 \rfloor +4$	20	20	22	22	24
P3	6	6	8	8	$2\lfloor n/2 \rfloor +5$	22	22	22	22	22*
P5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞

(d) Network delay table of P4

Slow convergence in asynchronous mode



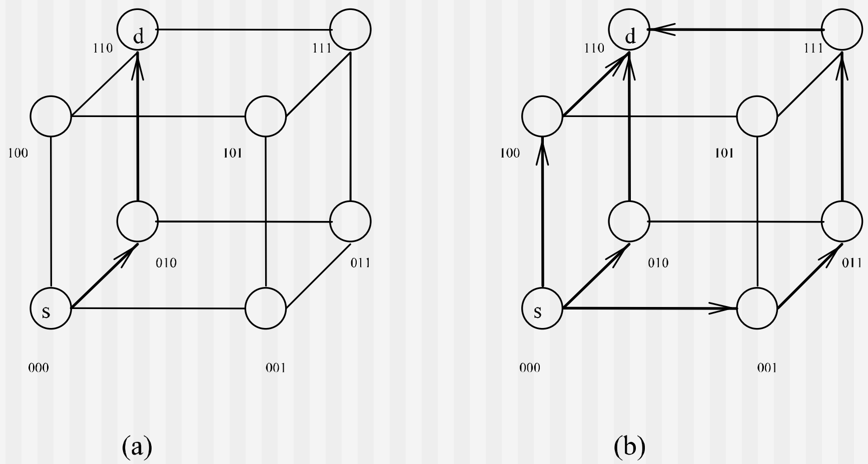
From node 0 to node n-1, unlabeled nodes have cost of 0

Paths that come in the following sequences with shorter routes

0, 1, 2, …, n-4, n-3, n-2, n-1
0, 1, 2, …, n-4, n-3, -,   n-1
0, 1, 2, …, -,   n-3 n-2, n-1
0, 1, 2, …, -,   n-3, -,   n-1
0, -, 2,   -,   n-3, -,   n-2

Special-Purpose Routing

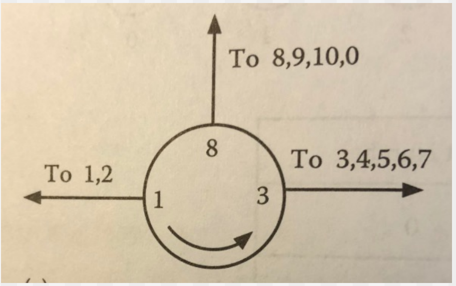
E-cube routing in n-cube: u ⊕ w as a navigation vector.

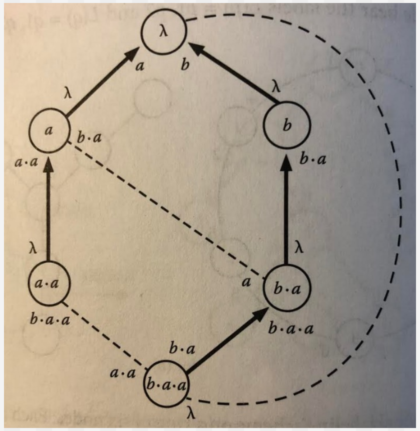


A routing in a 3-cube with source 000 and destination 110:
(a)Single path. (b) Three node-disjoint paths.

Compact Routing Table

Interval Routing: (destination, port number)



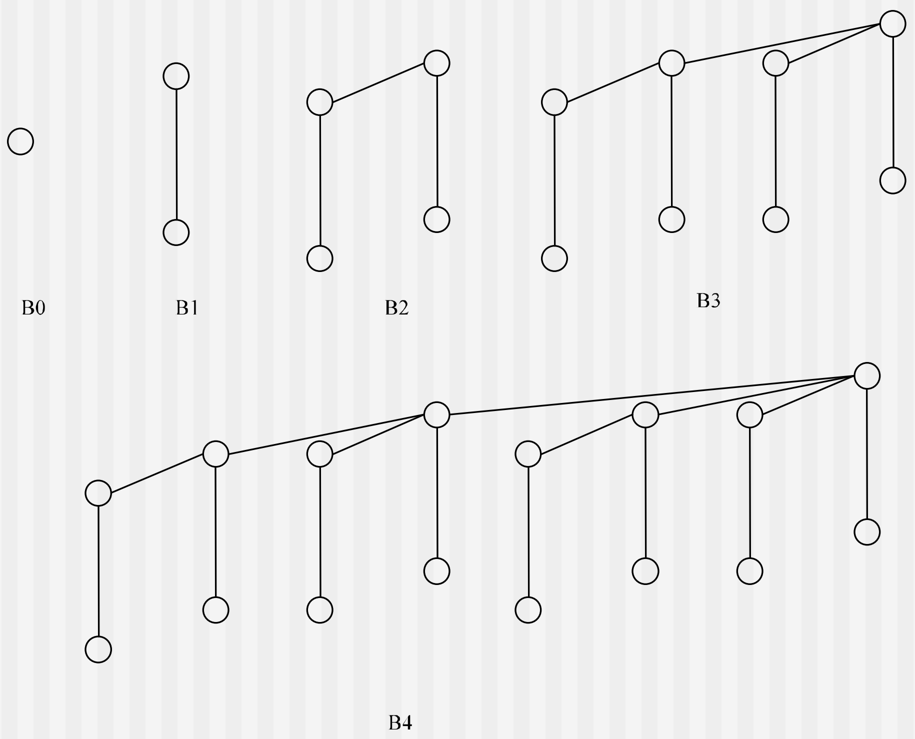


However, it does work well when a new link or node is added

Prefix Routing: forward to the port labeled with the longest prefix of destination

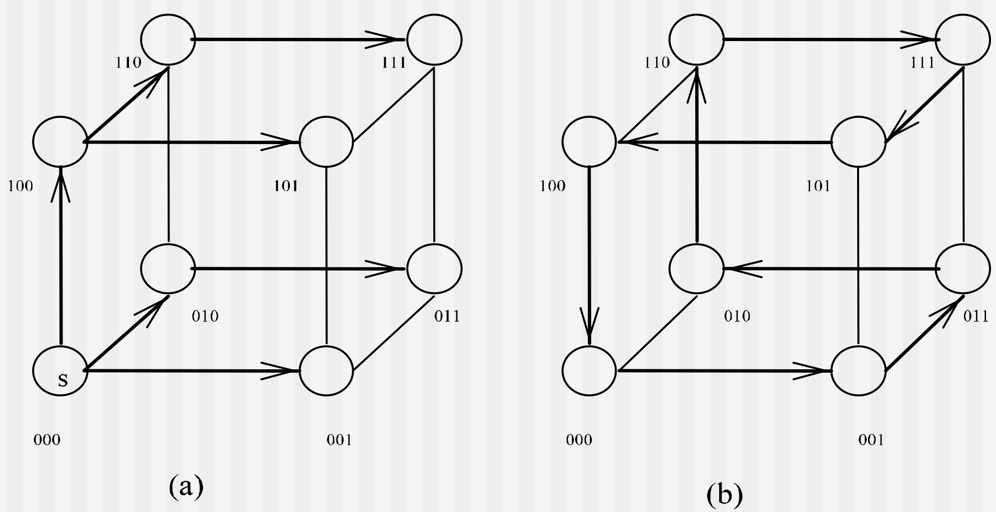
When a node has a label L, then the label of its child is L·x (λ: empty string for child to parent)

Binomial-Tree-Based Broadcasting in *N*-Cubes



The construction of binomial trees
(# of nodes at each level corresponds to a binomial number).

Hamiltonian-Cycle-Based Broadcasting in *N*-Cubes



(a) A broadcasting initiated from 000 with coordinated sequence (CS): {3, 2, 1}.

(b) A Hamiltonian cycle in a 3-cube.

Edge-disjointed Multiple Binomial Trees

- Source 000 sends m to each neighbor
- Each neighbor broadcasts m with a right rotation CS
- CS: {3, 2, 1}at 001
  CS: {1, 2, 3}at 010
  CS: {2, 1, 3}at 100

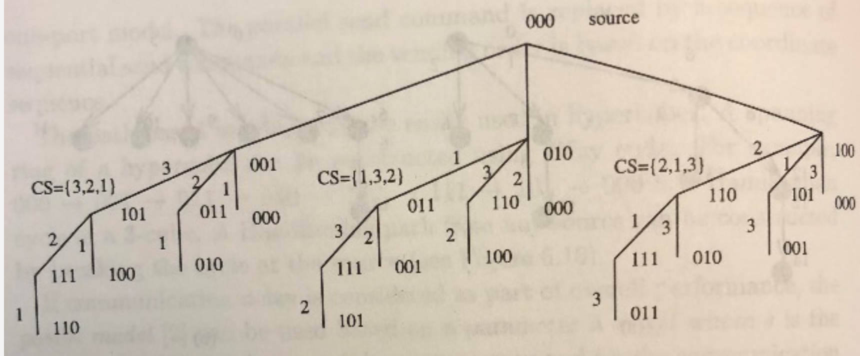


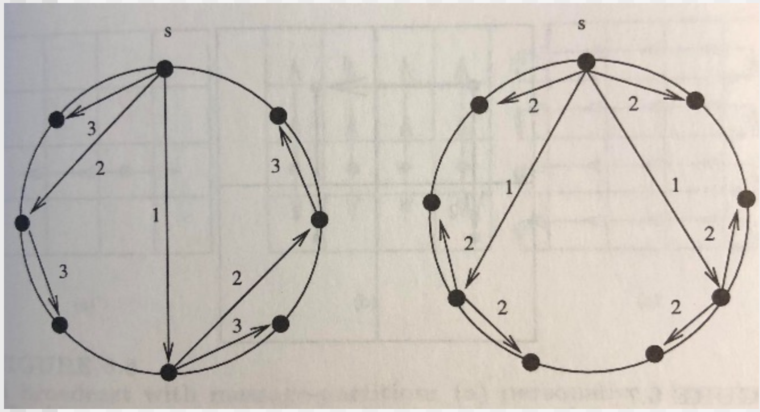
FIGURE 6.12
Edge-disjoint multiple binomial trees.

Node	Paths via Node 1	Paths via Node 2	Paths via Node 4
1	0	0-2-3	0-4-5
2	0-1-3	0	0-4-6
3	0-1	0-2	0-4-6-7
4	0-1-5	0-2-6	0
5	0-1	0-2-3-7	0-4
6	0-1-5-7	0-2	0-4
7	0-1-5	0-2-3	0-4-6

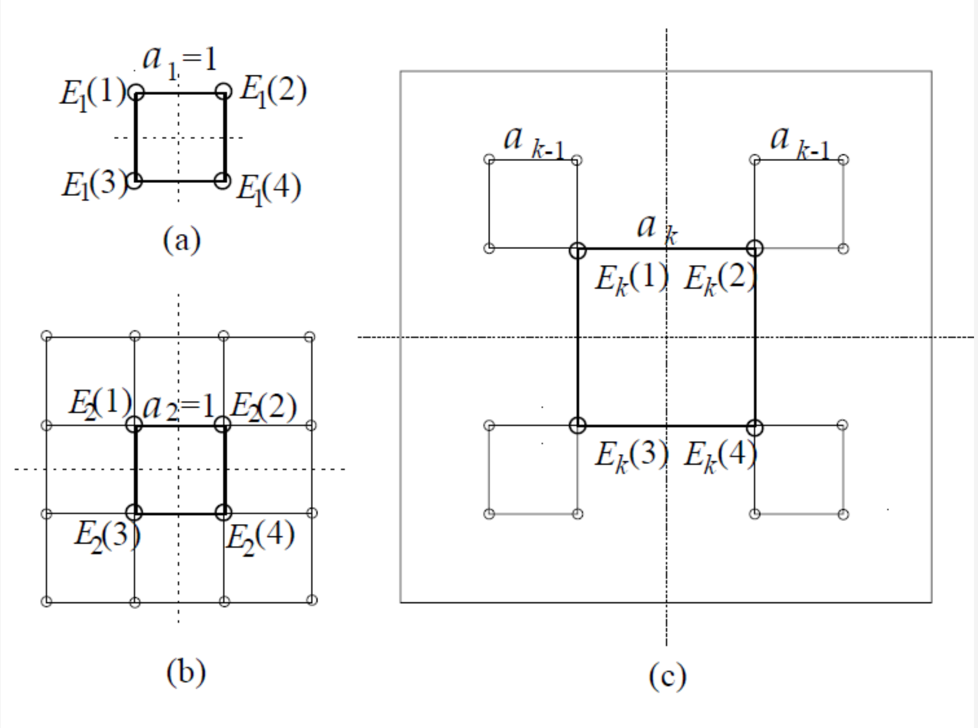
Table 6.4 Multiple paths to each node of a 3-cube.

Cut-through: recursive doubling

One-port or all-port
(without contention over links/paths)



(L) one-port and (R) all-port on ring



One-port on mesh with *minimum total distance* using ***eyes***: (a) 2x2, (b) 4x4, and (c) 2^k x 2^k meshes

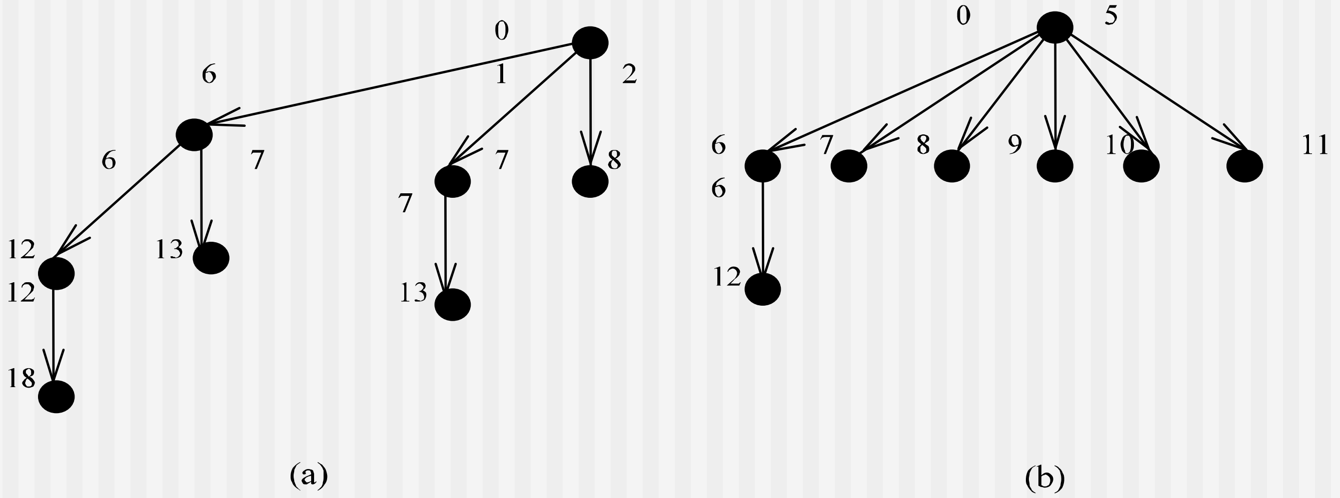
Parameterized Communication Model

Postal model:

- $\lambda = l/s$, where l is the communication latency and s is the latnecy for a node to send the next message.
- Under the **one-port model** the binomial tree is optimal when $\lambda = 1$.

$$N_\lambda(t) = N_\lambda(t-1) + N_\lambda(t-\lambda), \text{ if } t \geq \lambda; \ 1, \text{ otherwise}$$

Example 19: Broadcast Tree

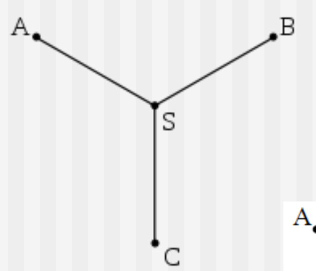


Comparison with λ = 6: (a) binomial tree and (b) optimal spanning tree.

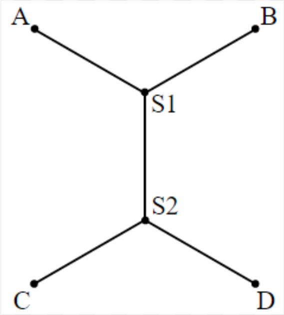
Multicasting

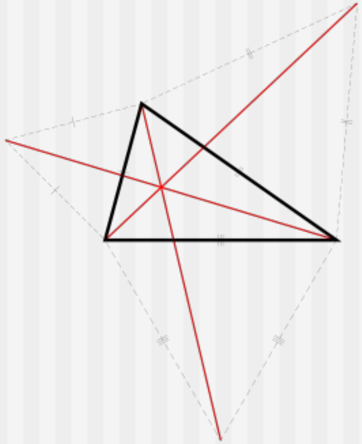
- Multicast path
- Core tree (for a graph): minimizing total length
- Shortest path tree (for a graph): minimizing path for each
- *Steiner tree* (points without a graph): a minimum tree that includes all destinations.

Three-points Steiner tree with the Fermat point S (e.g., all angles $\leq 120^o$)



In general, there N-2 Format points for given N points





Finding a minimum-weight Steiner tree is NP-hard

Focus 15: Fault-Tolerant Routing

Wu's safety level:

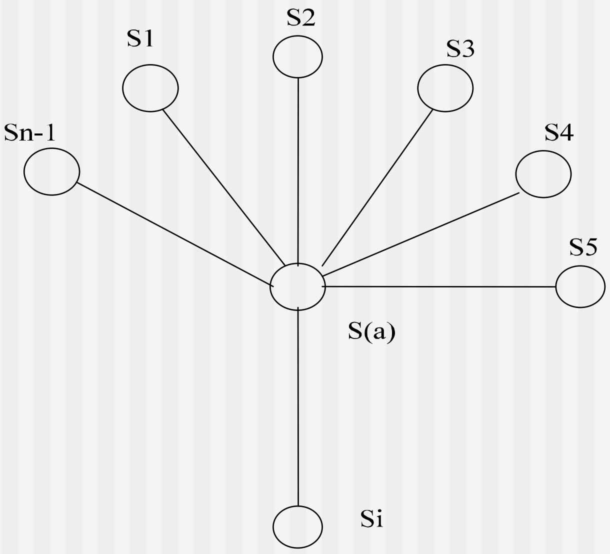
- The safety level associated with a node is an approximated measure of the number of faulty nodes in the neighborhood.
- Initially all faulty nodes have 0 as safety levels and all non-faulty nodes have n.
- Let $(S_0,S_1,S_2,\ldots,S_{n-1})$, $0 \leq S_i \leq n$, be the non-descending safety status sequence of node *a*'s neighboring nodes in an n-cube.
- Iteratively do the following: If $(S_0,S_1,S_2,\ldots,S_{n-1}) \geq (0,1,2,\ldots,n-1)$ then $S(a) = n$ else if $(S_0,S_1,S_2,\ldots S_{k-1}) \geq (0,1,2,\ldots,k-1) \wedge (S_k = k-1)$ then $S(a) = k$.

Insight: Embedding of binomial tree B_n in Q_n in terms of B_{n-1} (in a Q_{n-1}), B_{n-2}, …, B_1, and B_0 in *any orientation*.

Focus 15: Fault-Tolerant Routing (Cont'd)

Distributed algorithms: iterative exchanges (maximum n rounds) with neighbors' safety levels

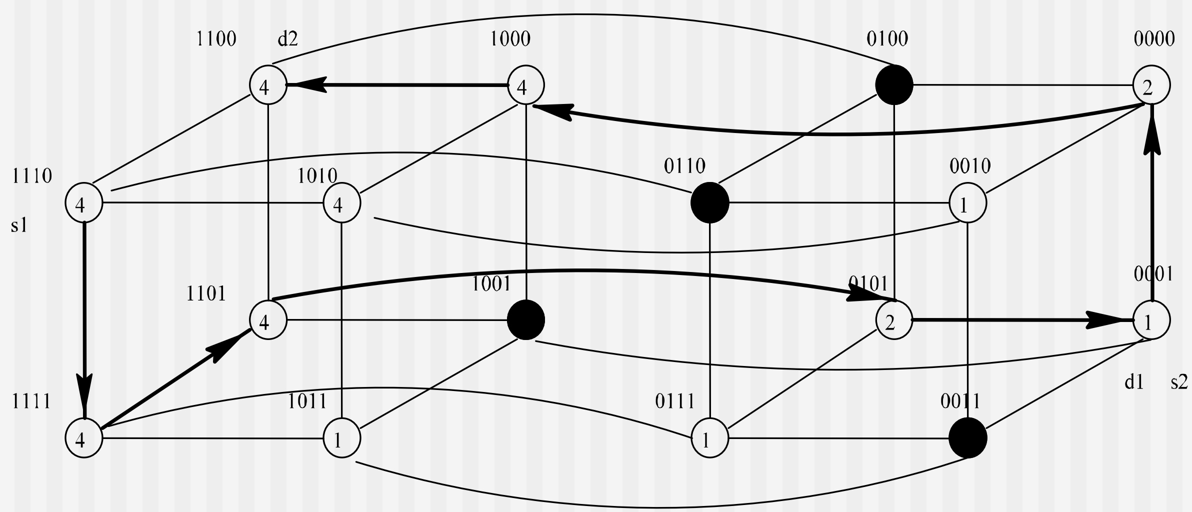
A node a is called **safe** if its level is n, i.e., S(a) =n



Fault-Tolerant Routing (Cont'd)

If the safety level of a node is k, there is at least one Hamming distance path from this node to any node within k-hop.

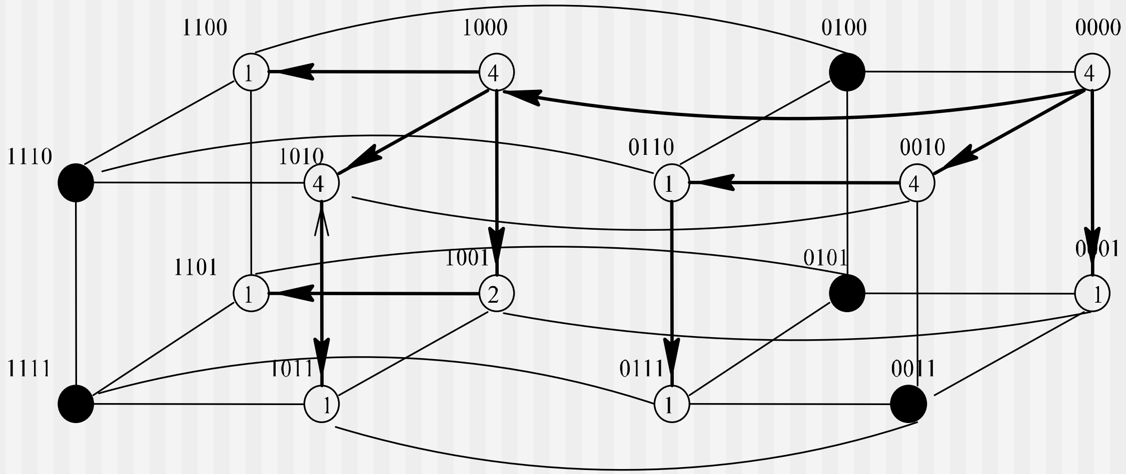
If there are at most n faults, every unsafe node has a safe neighbor.



A fault-tolerant routing using safety levels.

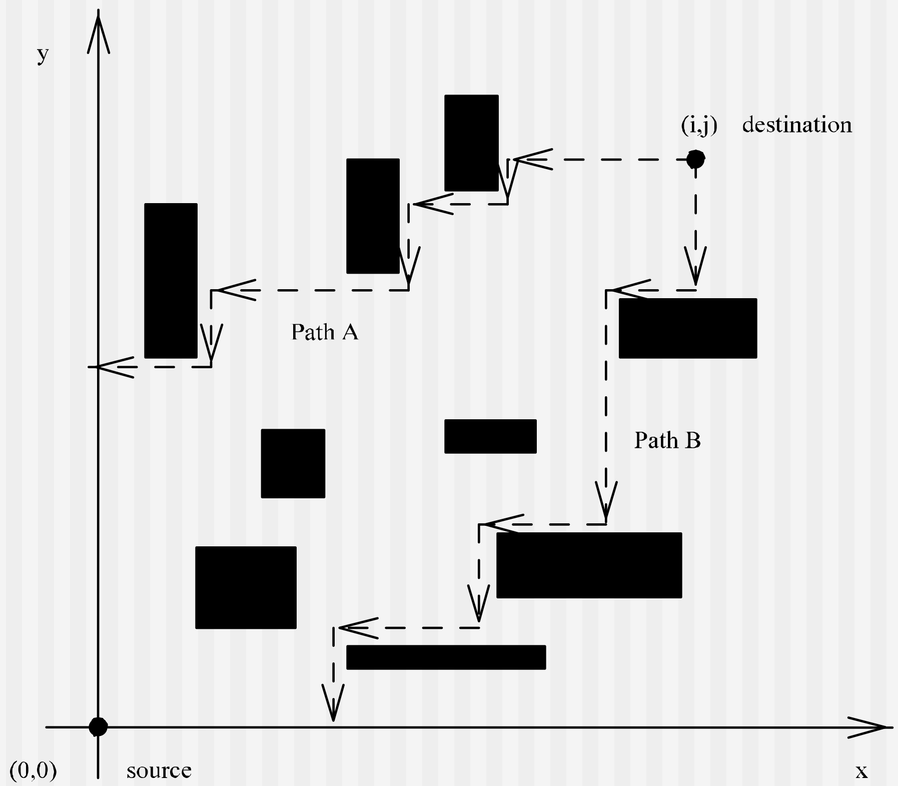
Fault-Tolerant Broadcasting

If the source node is n-safe, there exists an n-level injured spanning binomial tree in an n-cube: source can reach all non-faulty nodes through a Hamming distance path.



Broadcasting in a faulty 4-cube.

Wu's Extended Safety Level in 2-D Meshes



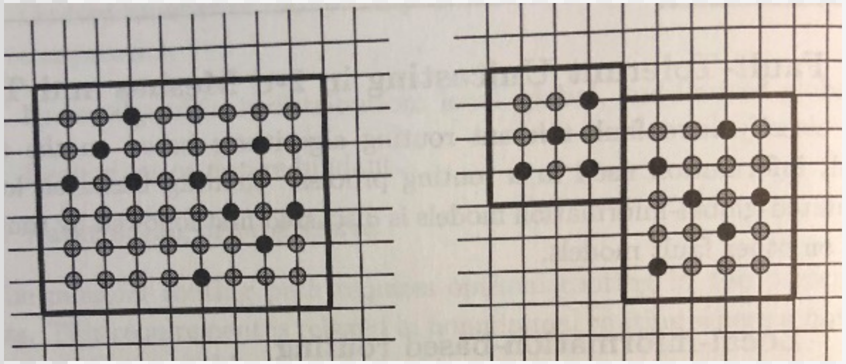
A sample region of minimal paths.

Safety Block

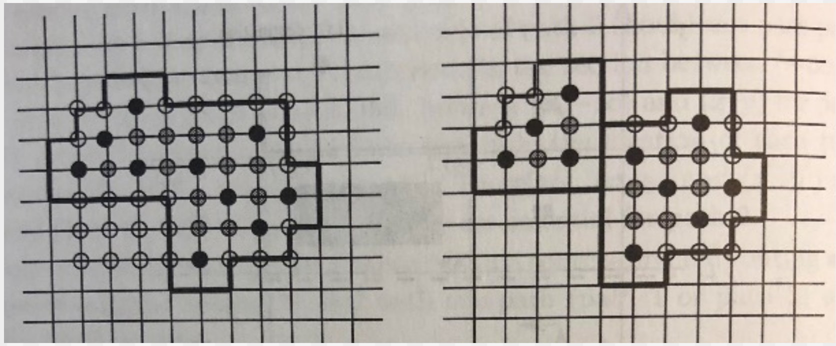
Safety block: (1) All faulty nodes are unsafe. All nonfaulty nodes are initially safe. (2) If a nonfaulty node has two or more faculty/unsafe neighbors, it is unsafe.

Extended safety block: (1). (2) …has a faulty/unsafe neighbor in both dimensions…

Wu's orthogonal convex region: All safe nodes are enabled. A unsafe node is initially disabled, but it is changed to the enabled status if it has two or more enabled neighbors.



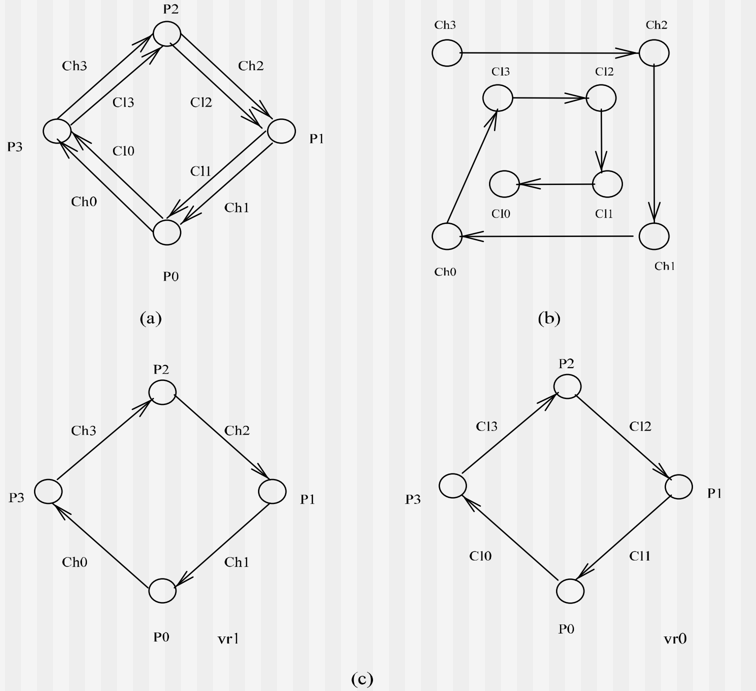
(L) Regular and (R) extended safe/unsafe



Enabled/disabled for (L) regular and (R) for extended

Deadlock-Free Routing

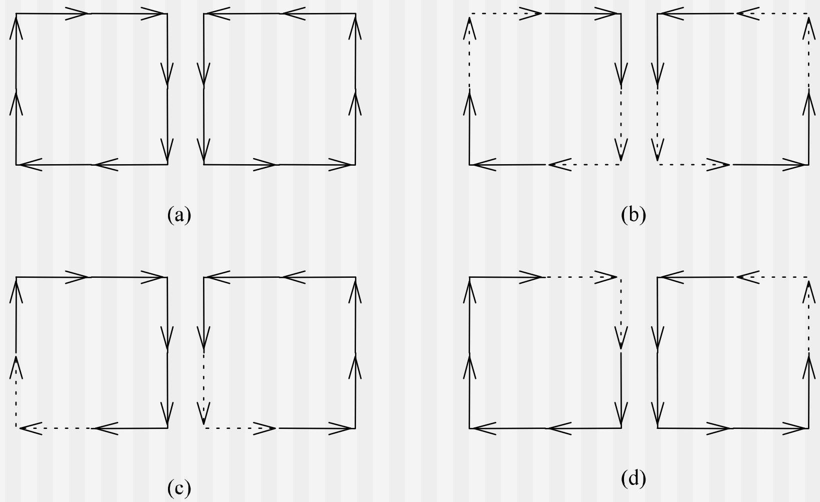
Virtual channels and **virtual networks**:



(a) A ring with two virtual channels, (b) channel dependency graph of (a), and (c) two virtual rings vr_1 and vr_0.

Focus 16: Deadlock-Free Routing Without Virtual Channels

- **XY-routing** in 2-D meshes: X dimension followed by Y dimension.
- Glass and Ni's **Turn model**: Certain turns are forbidden.



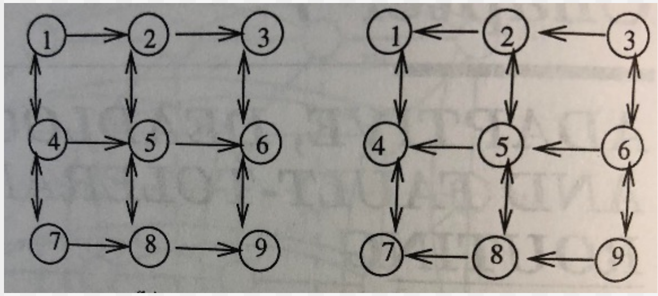
(a) Abstract cycles in 2-d meshes, (b) four turns (solid arrows) allowed in XY-routing, (c) six turns allowed in positive-first routing, and (d) six turns allowed in negative-first routing.

Planar-Adaptive Routing

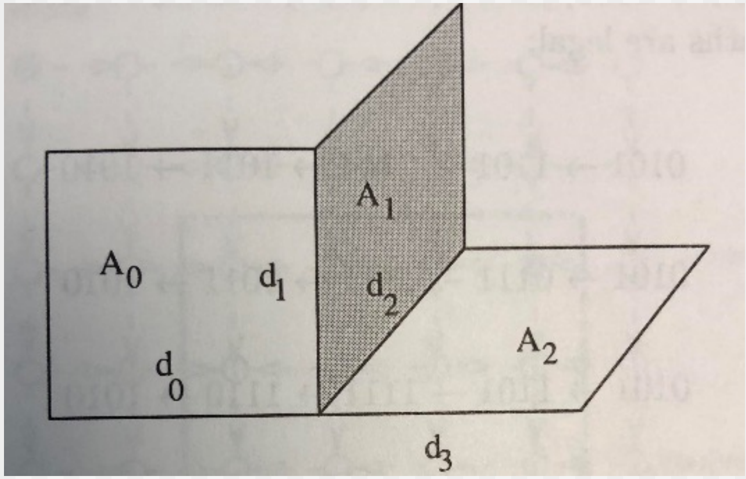
For general k-ary n-cubes, select n+1 2-D planes A_0, A_1, …, A_n. A_i spans dimension d_i and d_{i+1}.

Three virtual channels are used: one for d_i and two for d_{i+1}: $d_{i,2}$, $d_{i+1,0}$, and $d_{i+1, 1}$. (Second subscript is virtual channel number.)

Each plane has one positive and one negative subnetworks.



Positive and negative Networks in d_i and d_{i+1}



Escape channels

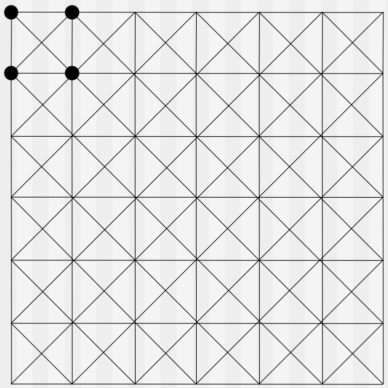
- Regular channels: non-waiting
- Escape channels: waiting
  - *Strongly connected*
  - *Strictly decreasing path*: for any pair of nodes, a decreasing (labelled) path exist.

Theorem: The minimum number of channels needed to meet the above two conditions is 2n-1, where n is the number of nodes.

L. Sheng and J. Wu, A Note on “A Tight Lower Bound on the Number of Channels Required for Deadlock-Free Wormhole Routing”, IEEE TC, Sept. 2000.

Exercise 5

1. Provide an addressing scheme for the following *extended mesh* (EM) which is a regular 2-D mesh with additional diagonal links. Provide a general shortest routing algorithm for EMs.



2. Repeat Example 18 after changing (P1, P3) to 4 and (P3, P5) to 8.

3. Suppose the postal model is used for broadcasting and $\lambda = 5$. What is the maximum number of nodes that can be reached in time unit 10. Derive the corresponding broadcast tree.

Exercise 5 (Cont'd)

4. Consider the following turn models:
   - *West-first routing*. Route a message first west, if necessary, and then adaptively south, east, and north.
   - *North-last routing*. First adaptively route a message south, east, and west; route the message north last.
   - *Negative-first routing*. First adaptively route a message along the negative X or Y axis; that is, south or west, then adaptively route the message along the positive X or Y axis.

   (a) Show all the turns allowed in each of the above three routings.

   (b) Show the corresponding routing paths using (1) positive-first, (2) west-first, (3) north-last, and (4) negative-first routing for the following unicasting: (2,1) to (5,9), (7,1) to (5,3), (6,4) to (3,1), and (1,7) to (5,2).
5. Wu and Fernandez (1992) gave the following safe and unsafe node definition: A nonfaulty node is unsafe if and only if either of the following conditions is true: (a) There are two faulty neighbors, or (b) there are at least three unsafe or faulty neighbors. Consider a 4-cube with faulty nodes 0100, 0011, 0101, 1110, and 1111. Find out the safety status (safe or unsafe) of each node.

Exercise 5 (Cont'd)

Repeat the above using Wu's safety vector. Critically compare safety node, safety level, and safety vector in terms of fault-tolerance capability and complexity. (J. Wu, Reliable communication in cube-based multipcomputers using safety vectors, IEEE TPDS, 9, (4), April 1998, 321-334.)

6. To support fault-tolerant routing in 2-D meshes, D. J. Wang (1999) proposed the following new model of faulty block: Suppose the destination is in the first quadrant of the source. Initially, label all faulty nodes as *faulty* and all non-faulty nodes as *fault-free*. If node u is fault-free, but its north neighbor and east neighbor are faulty or useless, u is labeled *useless*. If node u is fault-free, but its south neighbor and west neighbor are faulty or can't-reach, u is labeled *can't-reach*. The nodes are recursively labeled until there are no new useless or can't-reach nodes.

(a) Give an intuitive explanation of useless and can't-reach.

(b) Re-write the definition when the destination is in the second quadrant of the source.

Exercise 5 (Cont’d)

7. Chiu proposed an *odd-even turn model*, which is an extension to Glass and Ni's turn model. The odd-even turn model tries to prevent the formation of the *rightmost column segment of a cycle*. Two rules for turn are given in:
   - Rule 1: Any packet is *not* allowed to take an EN (east-north) turn at any nodes located in an even column, and it is *not* allowed to take an NW turn at any nodes located in an odd column.
   - Rule 2: Any packet is *not* allowed to take an ES turn at any nodes located in an even column, and it is *not* allowed to take a SW turn at any nodes located in an odd column.

(a) Use your own word to explain that the odd-even turn model is deadlock-free.

(b) Show *all the shortest paths* (permissible under the extended odd-even turn model) for

(a) s_1:(0, 0) and d_1:(2,2) and (b) s_2:(0,0) and d_2:(3,2)

Ge-Ming Chiu. "The odd-even turn model for adaptive routing." IEEE TPDS, 11, (7), July 2000, 729-738.