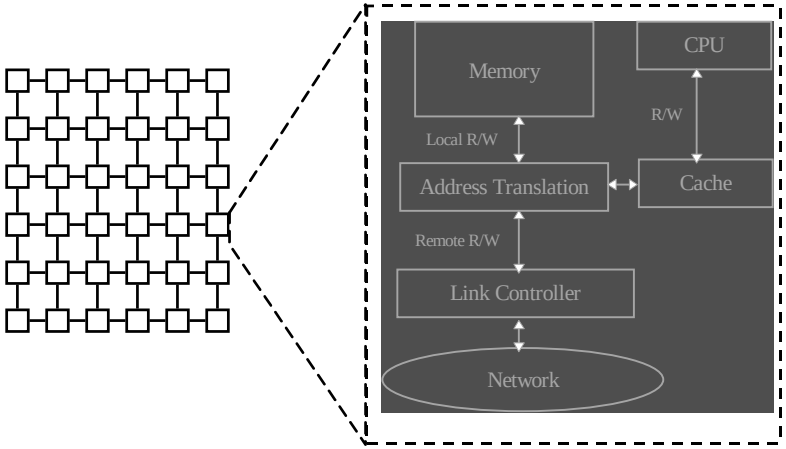
Example



Lecture 9

Distributed systems, network topologies and routing

Characteristics

- Non-uniform delays to access memory.
- Complex cache protocols.
- Private and global memory models.

Distributed systems

- Powerful processing elements are combined with memory modules to form a node.
- Constructed as a graph with nodes as the vertices.

Diameter

- The maximum the of the minimum path lengths between any two nodes in the topology.
- A path is the number of links that are traversed by a message to reach a destination node from a source node.
- A minimum path is the shortest possible path between a source and destination.

Topologies

- Topologies are based on known graphs with desirable properties like
  - high connectivity
  - low diameter
  - fixed order

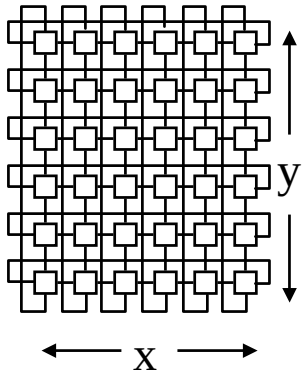
Node Order

- The order of a graph is given by the maximum number of arcs incident at a vertex.
- In the computer network context the order of a node refers to the number of connections a node in the network has to other nodes.

Connectivity

- The number of nodes connected to other nodes.
- In a fully connected topology every node has a direct connection to every other node.
- Obviously this is a problem for large numbers of nodes.
  - wire layout
  - chip pin limitations

Torus

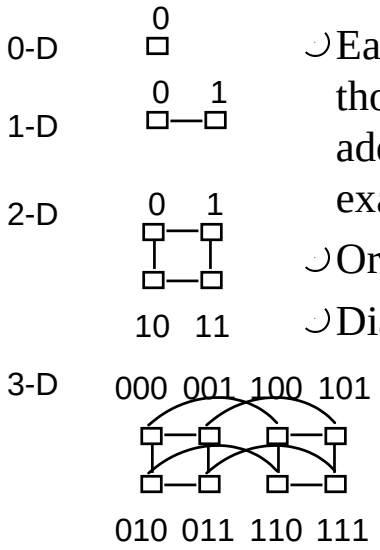


- Order = 4
- Diameter =
- Which nodes are the furthest apart in a torus?

What are topologies?

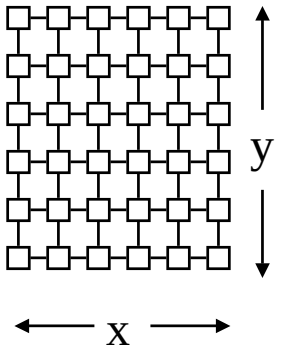
- Simple graphs like
  - mesh, torus
  - hypercube
  - ring, chordal ring.
- As well as hybrid designs like
  - Hierarchical Cubic Network (HCN)
  - Extended hypercube
  - Cube Connected Cycles

Hypercube



- Each node is connected to those others whose binary addresses differ from its by exactly one bit.
- Order = ?
- Diameter= ?

Mesh



- Order = 4
- Diameter =

What resources are needed

- Buffer space on each node in which to store the message. This depends on the message forwarding strategy.
  - Store and Forward
  - Wormhole routing/Virtual Cut-Through
- Link bandwidth. Can a link be committed to transfering a message packet in this cycle?

Routing

- The process of deciding how best to deliver a message.
  - Which path to take?
  - What resources are needed?
  - How to avoid deadlock?

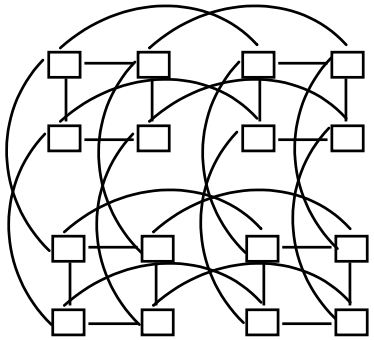
Avoiding deadlock

- Allocate shared resources in a fixed global order.
- Back off and retry later. This can lead to livelock.
- Livelock is the recurrent occurence of resource allocation sequences that lead to backoff in all the particiants.

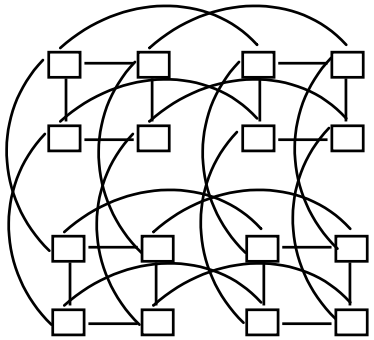
Which path to take?

- Traverse a link that reduces the distance to the destination.
- Some routing schemes allow a message to take a path which increases the distance to the destination if the links required for an optimal path are in high demand.

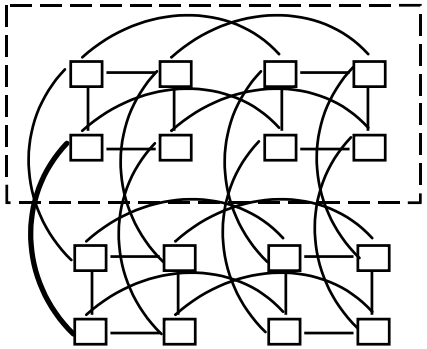
A routing example



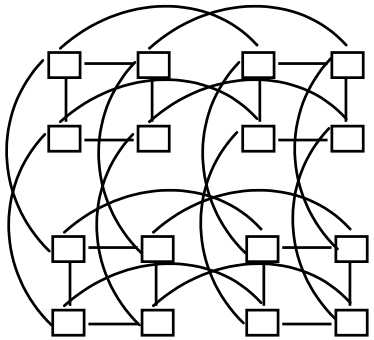
A routing example



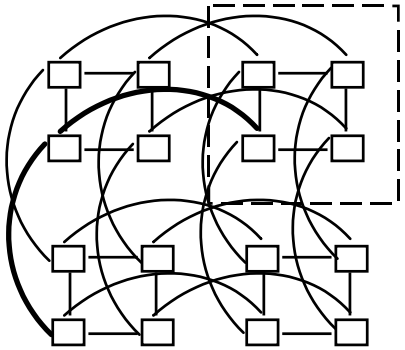
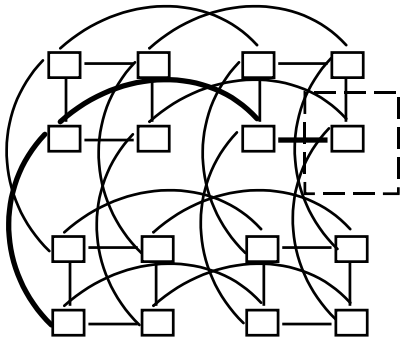
A routing example



A routing example



A routing example



A routing example