Parallel Architectures

Chapter 6:
Interconnection Networks

Lecturer: *Hing Wing To*

Autumn 1997

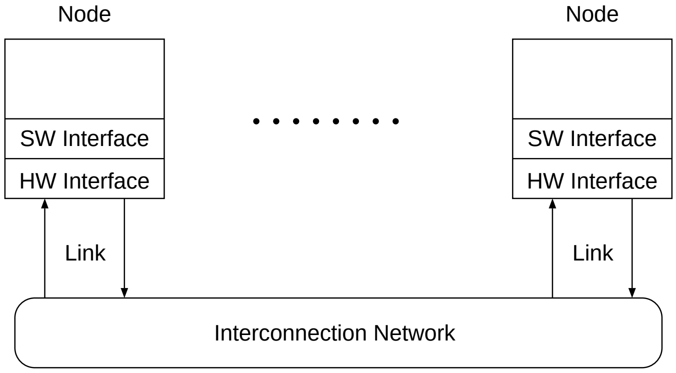
Department of Computing
Imperial College

Room 379
email: hwt@doc.ic.ac.uk

Course web site:
http://www-ala.doc.ic.ac.uk/~phjk/phjk-ParallelArchitectures.html

CHAPTER 6

6.1 Generic Interconnection Network



Key factors in the design of interconnection networks (IN) for massively parallel processors (MPP) are *bandwidth* and *latency*.

Why are these factors important for parallel architectures?

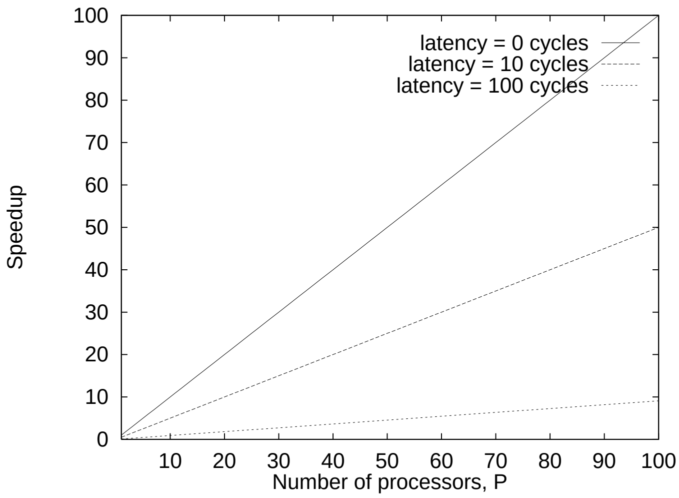
$$\text{speedup} = \frac{\text{time on one processor}}{\text{time on } P \text{ processors}}$$

Effect of IN Latency

Consider a problem which can always be partitioned into P tasks.

Assume that 10% of computation cycles will require data owned by other tasks.

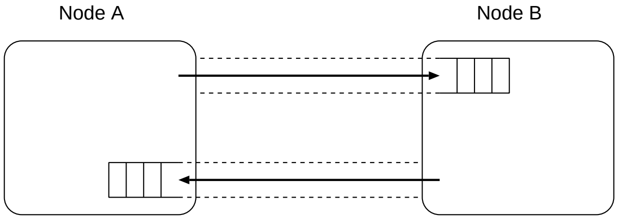
$$\text{speedup} = \frac{T_{comp}}{T_{comp}/P + T_{comp} \times 0.1/P * latency}$$



The latency of the IN can have a huge effect on parallel program performance.

6.2 A Simple Network

Consider two nodes connected by a bidirectional channel:

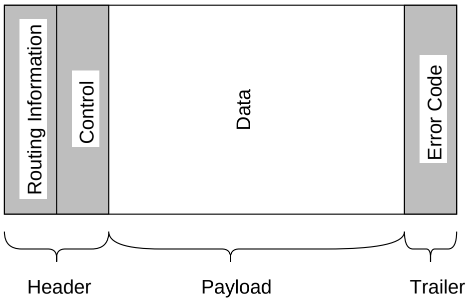


Channel Bandwidth — Maximum rate at which the channel can propagate information.

Message — Logical unit of communication.

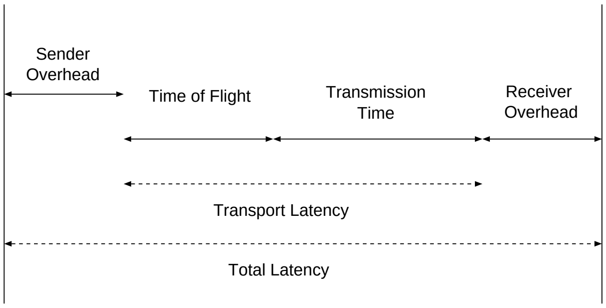
Packet — Unit of communication.

Typical packet format:



Latency of Sending a Packet

(See H&P pages 568–569)



Time taken to send a packet across IN:

$$\begin{aligned} \text{Total latency} = \ & \text{Sender O/H} + \text{Time of flight} \\ & + \frac{\text{Packet size}}{\text{Bandwidth}} \\ & + \text{Receiver O/H} \end{aligned}$$

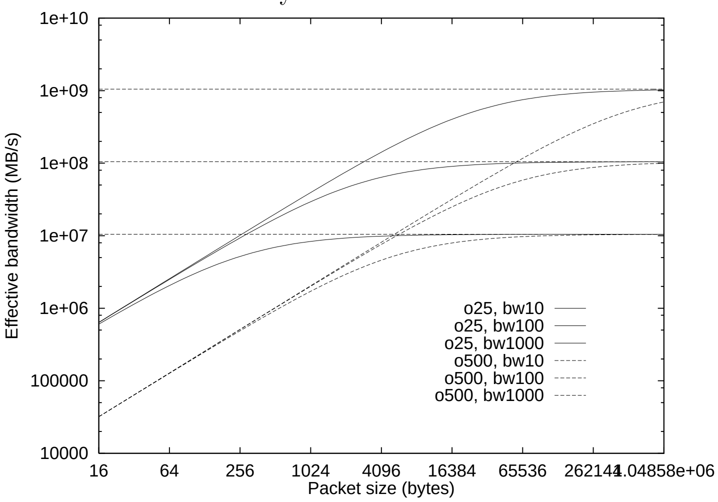
Effective bandwidth delivered by IN:

$$\text{Effective bandwidth} = \frac{\text{Packet size}}{\text{Total latency}}$$

Impact of Overheads and Message Size

Effective bandwidth for overheads of 25μs and 500μs and channel bandwidths of 10 MB/s, 100 MB/s and 1000 MB/s.

Packet size from 16 bytes to 1 MB.

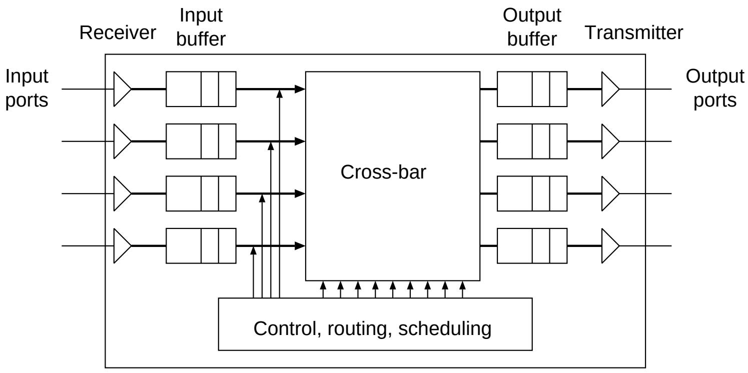


Must lower overheads as well as increase network bandwidth unless messages are very large.

6.3 Connecting Multiple Nodes

MPP IN usually composed of links and *switches*.

Basic switch organisation:



Switches connect a fixed number of input channels to a fixed number of output channels; this number is known as the *degree* of the switch.

Transmitter, link, receiver and buffer coupling form a *channel*.

A node is connected to a single switch possibly by more than one channel.

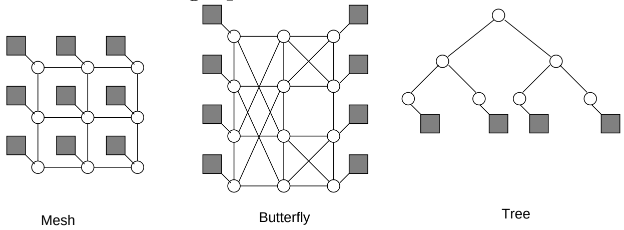
Switches *route* packets from source node to destination node.

6.4 Characteristics of IN

IN characterised by four factors:

Topology

Interconnection graph of IN.



Routing Algorithm

Method used to choose path from one node to another.

Flow Control

Method used to regulate traffic in network. Arbitrating between messages competing for same resource.

Switching Technique

Determines how data in a message traverse its route.

Supplementary Material

Provided:

A Survey of Wormhole Routing Techniques in Direct Networks, L. M. Ni and P. McKinley, February, p62–76, *IEEE Computing*, 1993.

Design and Implementation of an Interconnection Network for the AP1000, T. Horie, H. Ishihata and M. Ikesaka, *Algorithms, Software, Architecture Information Processing'92*, vol I, p555–561, Elsevier, 1992.

Good, but long:

Parallel Computer Architecture A Hardware / Software Approach, D. Culler, J. P. Singh and A. Gupta, (Pre-production draft 9/4/97), Chapter 10, `http://now.cs.berkeley.edu/~culler/book.alpha/`

Network and Processor Architecture for Message-Driven Computers, W. J. Dally, *VLSI and Parallel Computation*, Chapter 3, Morgan Kaufmann, 1990.

6.5 Units of Information

- **Message**

  Logical unit of communication.
- **Packets**

  Smallest unit of information to contain routing information.
- **Flit**

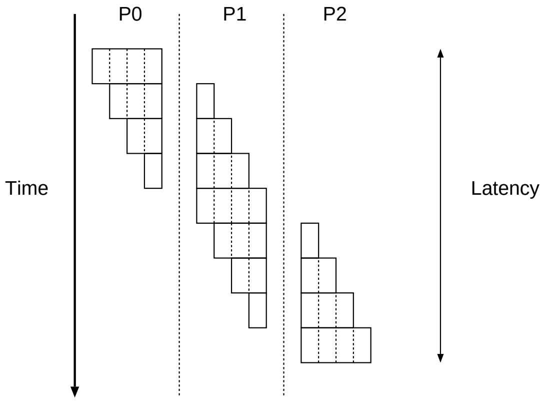
  Flow Control Unit — Smallest unit on which flow control is performed.
- **Phit**

  Physical Unit — The amount of data transferred across a link in a single cycle.

6.6 Switching Techniques

Store-and-forward Routing

Entire packet is received by a switch before it is forwarded onto next channel.



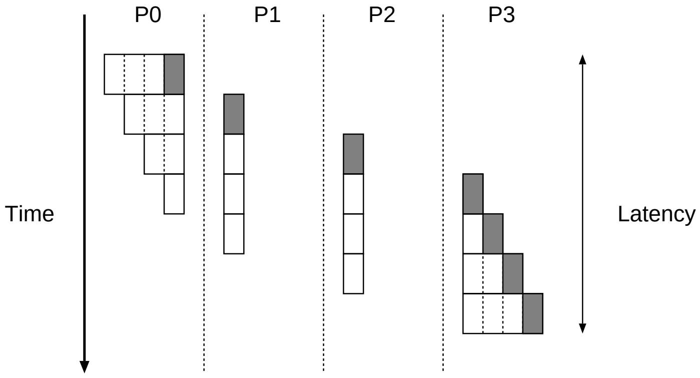
Latency of sending a packet point-to-point over an *unloaded* network is:

$$\begin{aligned}\text{Latency} = \ & \text{Sender O/H} \\ & +h\left(\text{Switch delay} + \frac{\text{Packet size}}{\text{Bandwidth}}\right) \\ & +\text{Receiver O/H}\end{aligned}$$

h is the number of hops (channels traversed).

Cut-Through Routing

A packet is forwarded by the switch as it arrives.



Packet is pipelined through the switches in units of flits.

Latency of sending a packet point-to-point over an *unloaded* network is:

$$\begin{aligned}\text{Latency} = \; & \text{Sender O/H} \\ & +h\left(\text{Switch delay} + \frac{\text{Route size}}{\text{Bandwidth}}\right) \\ & +\frac{\text{Packet size - Route size}}{\text{Bandwidth}} \\ & +\text{Receiver O/H}\end{aligned}$$

h is the number of hops (channels traversed).

Usually assume that the routing information is held in a single flit.

The model is then simplified to:

$$\begin{aligned}\text{Latency} = \; & \text{Sender O/H} \\ & +h \times \text{Switch delay} + \frac{\text{Packet size}}{\text{Bandwidth}} \\ & +\text{Receiver O/H}\end{aligned}$$

This is the model we shall use.

6.7 Topology

Topologies evaluated in terms of five parameters:

- **Bisection Width, B**

  Minimum number of channels cut which separate the network into equal halves.

  Bisection bandwidth is B $\times$ channel bandwidth.
- **Degree, d**

  Number of channels incident on a node.
- **Diameter, D**

  Longest shortest path.
- **Symmetry**

  A topology is symmetric if it is isomorphic to itself with any node labelled as the origin.
- **Wire length**

Two broad classes of topologies:

Direct Networks — Every switch is attached to a processing node.

Indirect Networks — Not every switch is attached to a processing node.

k-ary n-cube Direct Networks

Most direct network topologies have been built with k-ary n-cubes or are isomorphic to them.

$$\begin{aligned} n &= \text{dimension} \\ k &= \text{radix} \\ P &= \text{number of nodes} \\ P &= k^n \end{aligned}$$

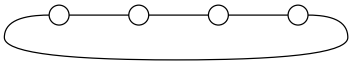
k-ary n-cube:

- k nodes in each dimension,
- each node can be labelled by an n digit number of radix (base) k,
- each node is connected to every node which has a label which differs in only one digit by one.

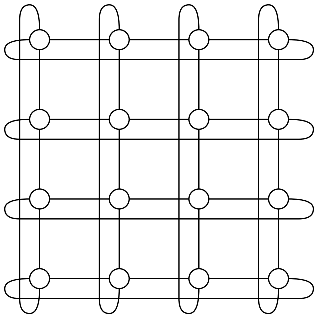
A k-ary n-cube can be constructed by connecting together k k-ary $(n-1)$-cubes together in a ring.

Some Important k-ary n-cubes

Rings — k-ary 1-cubes:



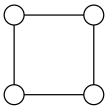
2D Tori — k-ary 2-cubes:



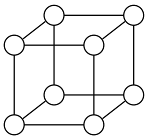
3D Tori — k-ary 3-cubes:

A 3D mesh with wrap-around channels.

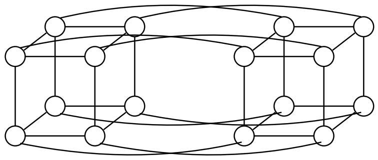
Hypercubes — 2-ary n-cubes:



2-ary 2-cube



2-ary 3-cube



2-ary 4-cube

Parameters of k-ary n-cubes

The parameters of these networks are given by the following equations:

$$\begin{aligned} B &= 2k^n/k = 2P/k \\ d &= 2n \\ D &= nk/2 \end{aligned}$$

Enables us to compare a wide variety of networks using just a few parameters.

For example assuming a fixed P we can make simple observations:

- If degree (d) of switch is limiting factor then minimise dimension n.
- If routing distance (indicated by D) is primary concern then choose high dimension n.

6.8 Cost Performance Trade-off in Unloaded Networks

Assume cost of networks is proportional to the number of wires per node.

This enables a trade-off between channel bandwidth with degree.

For P nodes:

	2D Torus	3D Torus	Hypercube
average route	$\sqrt{P}/2$	$3/4\sqrt[3]{P}$	$(\log_2 P)/2$
d	4	6	$2\log_2 P$

Assume:

$$\begin{aligned} t_s &= \text{sender overhead} \\ t_r &= \text{receiver overhead} \\ t_{sw} &= \text{switch delay} \\ bw &= \text{bandwidth of all wires incident on switch} \end{aligned}$$

Latency of sending a message of size m over an average distance using cut-through routing:

$$\text{2D Torus} = t_s + t_r + t_{sw}\frac{\sqrt{P}}{2} + \frac{m}{(bw/4)}$$

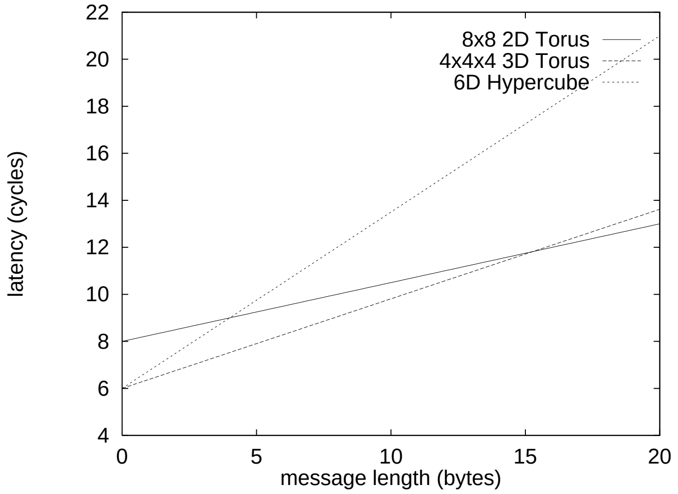
$$3\text{D Torus} = t_s + t_r + t_{sw}\frac{3\sqrt[3]{P}}{4} + \frac{m}{(bw/6)}$$

$$\text{Hypercube} = t_s + t_r + t_{sw}\frac{\log_2 P}{2} + \frac{m}{(bw/2\log_2 P)}$$

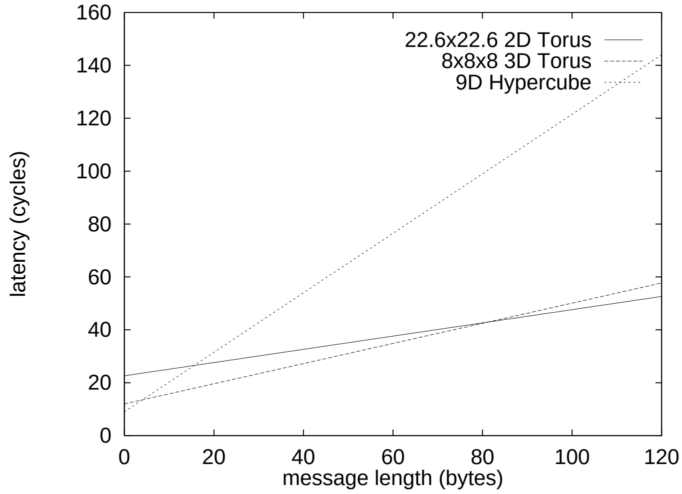
Assume a switch delay of 2 cycles, 128 wires per node each capable of transmitting one bit per cycle.

Ignore t_s and t_r as these cancel out.

For 64 nodes:



For 512 nodes:



Final choice depends on expected message length, but for long messages low dimension networks give the lowest latencies *for the given assumptions*.

6.9 Routing Algorithms

Routing algorithm determines the path used to get a packet from a source node to a destination node.

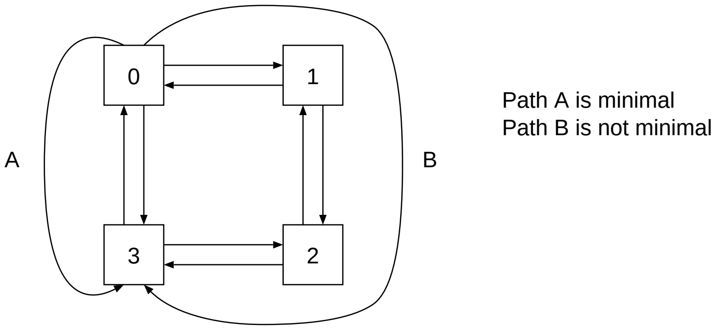
Shall study deterministic minimal routing algorithms.

Deterministic

- Path chosen depends solely on source and destination.
- Route does not depend on traffic.

Minimal

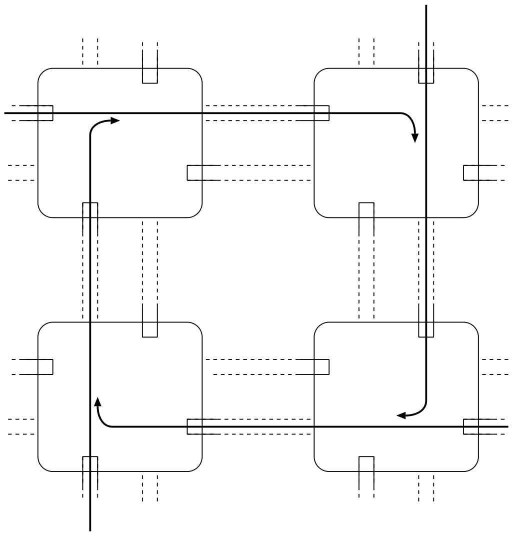
- Path chosen is shortest path.



Deadlock

One of the most important properties of a routing algorithm is to maintain *freedom from deadlock*.

One way in which deadlock can occur is if multiple messages require mutually dependent channel resources.



Dimension Order Routing

Common approach to routing for k-ary n-cubes.

Known as:

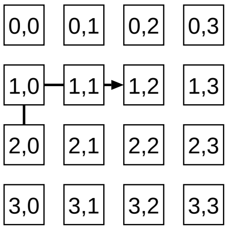
- XY-routing (or ΔX, ΔY) for 2D tori and meshes.
- e-cube routing for hypercubes

Dimensions of k-ary n-cube placed into order.

Packet first routed to correct position in higher dimension before attempting to route in next dimension.

For example in a 2D mesh, route first in X dimension, then in Y dimension.

Consider routing from (2,0) to (1,2):



Is it minimal?

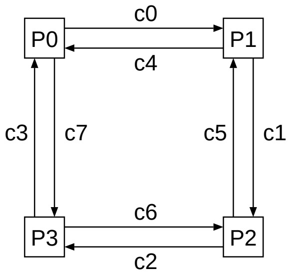
Is it deadlock free?

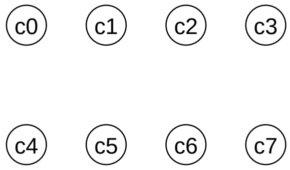
Determining Deadlock Freedom

Basic technique is to draw channel dependence graph and demonstrate that there are no cycles.

To derive channel dependence graph:

- Label all channels.
- For any two channels A and B, draw an arc if routing algorithm moves a packet from channel A to channel B.





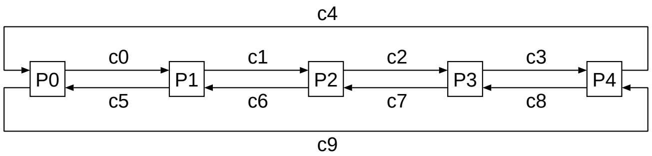
Dimension order routing for 4 node ring is deadlock free.

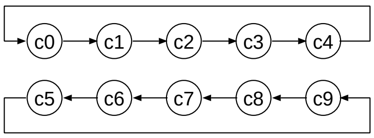
True for all k-ary n-cubes if $k < 5$ (all hypercubes).

True for all meshes.

Breaking Cycles

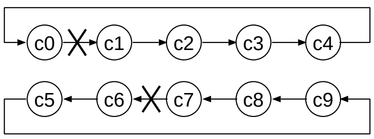
Consider 5 node ring:





Minimal routing will not be deadlock free.

Remove links to ensure deadlock free.



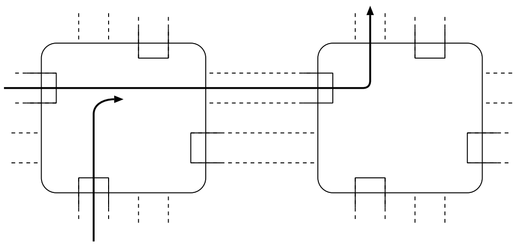
To route from P0 to P2 must use c9 c8 c7 as c0 c1 forbidden.

Routing no longer minimal!

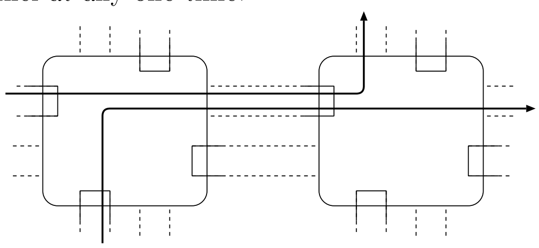
6.10 Virtual Channels and Deadlock

Virtual channels provide multiple channels over each physical channel.

With standard wormhole routing only one packet can use a given channel at any one time.

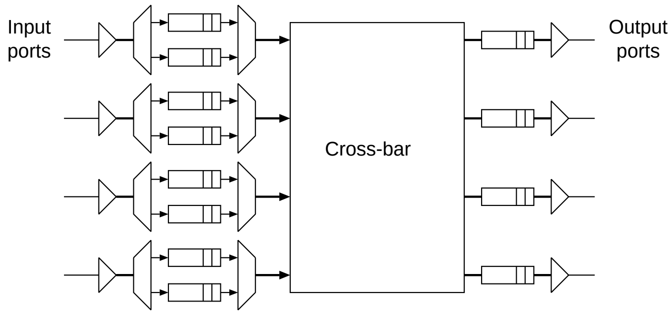


With virtual channels several packets can use a given channel at any one time.



Virtual channels are implemented by multiplexing packets across a single physical channel.

Switch organisation with virtual channels:



Since packets multiplexed across the physical channel the bandwidth of the channels is shared between the processors.

The multiplexing technique should be designed to maximize channel utilisation.

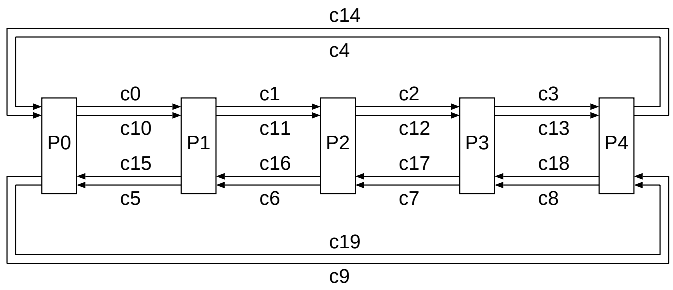
Specifically if W is channel bandwidth, and k virtual channels are active the bandwidth of the channel should be W/k, where k is less than the number of virtual channels.

Breaking Cycles with Virtual Channels

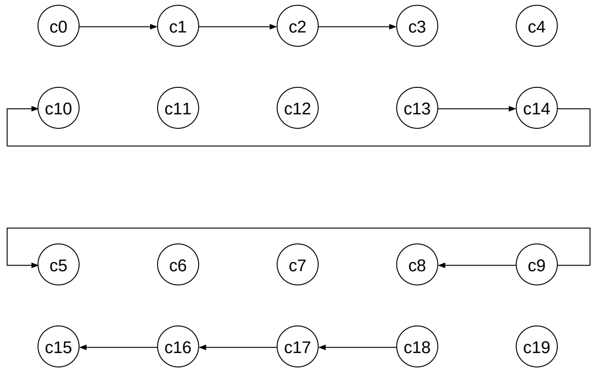
General scheme for k-ary n-cubes:

- Divide each physical channel into two virtual channels.
- Perform dimension order routing, but
  - When source node greater than destination node use high channels.
  - When source node less than destination node use low channels.

Consider 5 node ring with two virtual channels per physical channel:



Channel dependence graph:



Deadlock free as there are no cycles in graph.

6.11 Flow Control

Flow control determines the actions of two or more packets attempting to use the same resource (a channel).

With cut-through routing two options available:

Virtual cut-through

- Blocked packet spooled into buffer.
- Behaviour of blocked packet similar to store-and-forward.
- Requires large buffer space.

Worm-hole routing

- Blocked packet left strung out over route.
- Requires small amount of buffer space in each switch.
- Increased probability of contention.

Link Level Control Flow

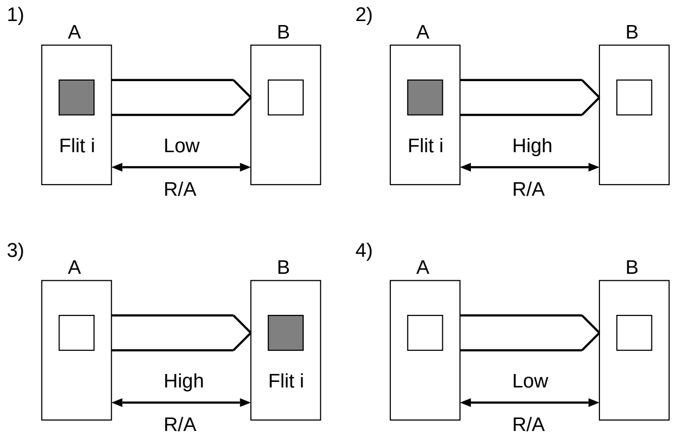
Packet headed for a full buffer is typically blocked in place.

Requires handshake between output port and input port.

A single-wire request/acknowledge (R/A) line is associated with each channel.

R/A line can only be raised by sender (requester).

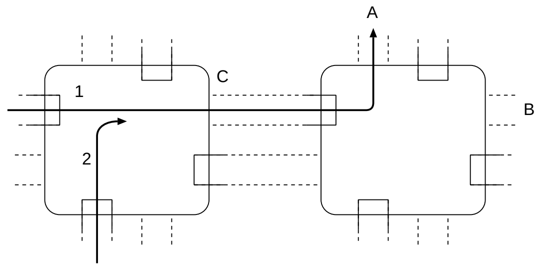
R/A line can only be lowered by receiver (acknowledging side).



Contention

When a packet has occupied a channel no other packet can use it.

If packet becomes blocked, the occupied channel becomes idle even if other packets can make productive use of it.



Packet 1 routed through channel A and channel C.

Packet 2 wishes to go through channel B.

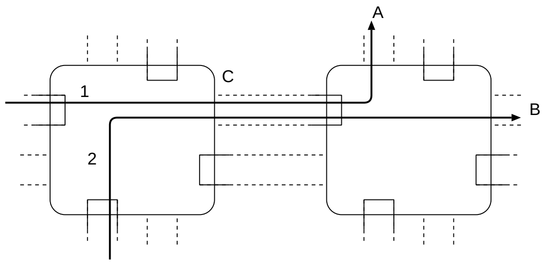
Assume packet 1 blocked. Therefore channel C idle, even though packet 2 could make productive use of it.

Furthermore packet 2 can then block further packets resulting in *tree saturation*.

Dally claims this effect limits throughput of IN to fraction (20%–50%) of network's capabilities.

6.12 Virtual Channels and Flow Control

Virtual channels enables the blocked packet to progress.



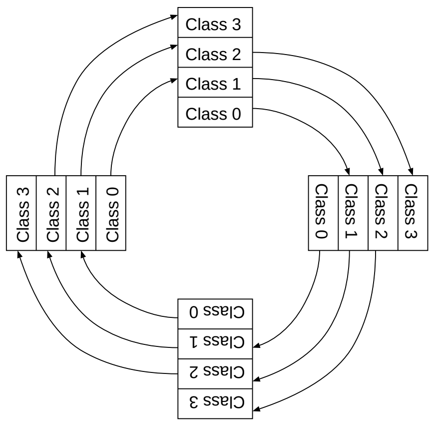
- Select physical output channel.
- Find next free virtual channel on physical channel.
- If no free virtual channel then block.
- Transmit packet along virtual channel.
- Release virtual channel.

Experiments on 2-ary butterflies with 16 flit buffers. Throughput of ≈ 25% for 1 virtual channel rising to ≈ 80% for 16 virtual channels.

Structured Buffer Pool in Wormhole Routing

Require n virtual channels, where n is maximum distance between two processors + 1.

Consider 4 node ring with *unidirectional* channels.



No cycles are formed therefore deadlock cannot occur.