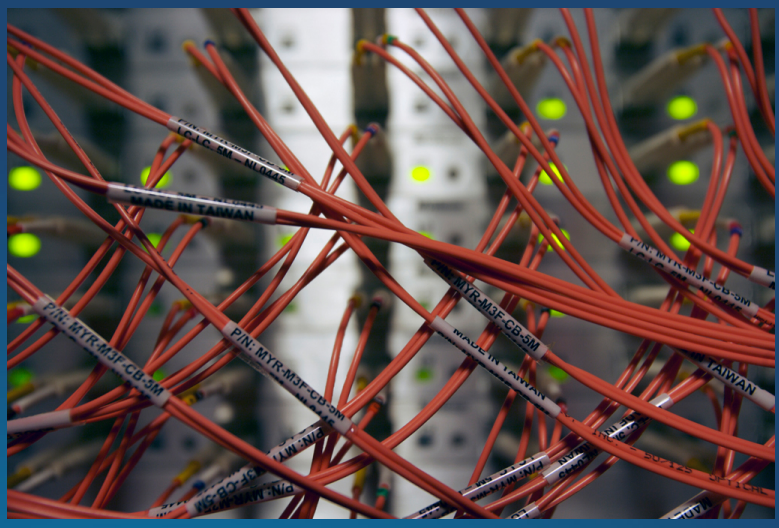
Interconnection Networks

Frédéric Desprez
INRIA



Some References

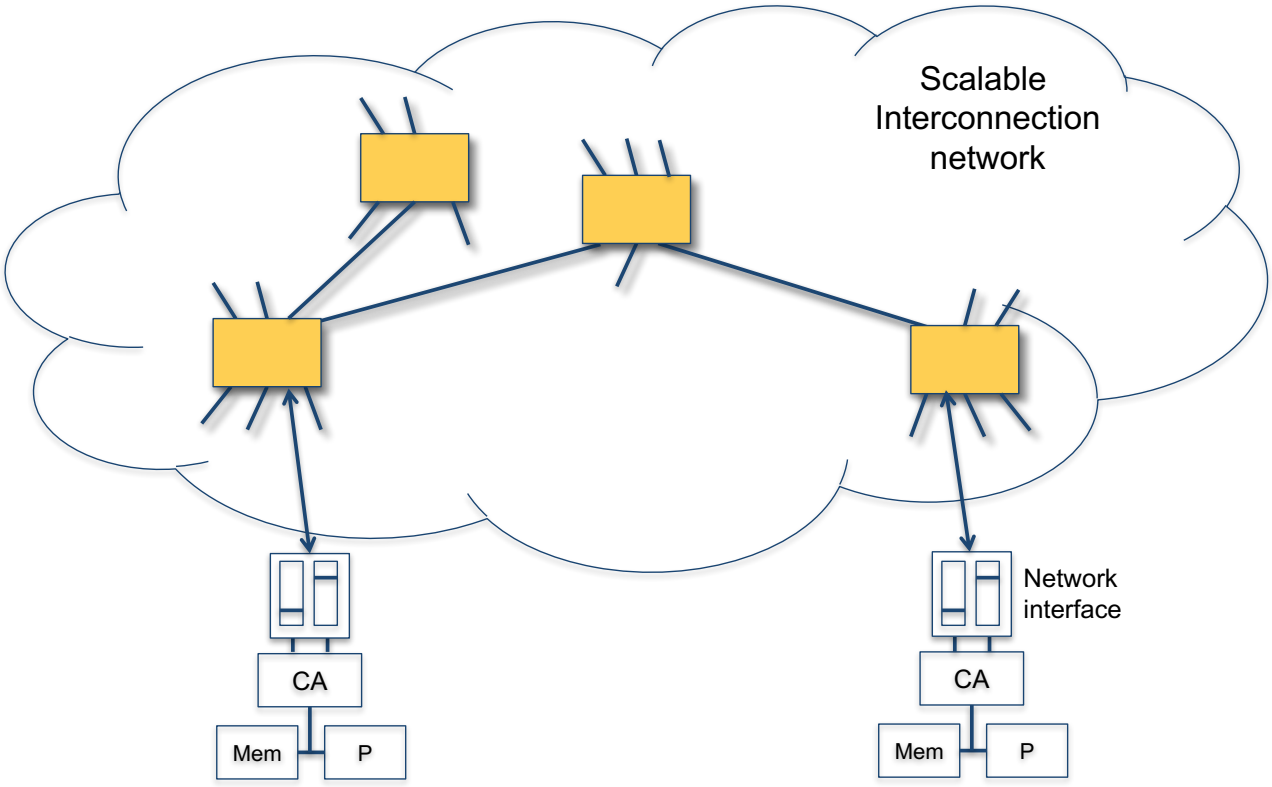
• **Parallel Programming – For Multicore and Cluster System,** T. Rauber, G. Rünger
• Lecture "**Calcul hautes performance – architectures et modèles de programmation**", Françoise Roch, Observatoire des Sciences de l'Univers de Grenoble Mesocentre CIMENT
• **4 visions about HPC - A chat**, X. Vigouroux, Bull
• **Parallel Computer Architecture – A Hardware/Software Approach,** D.E. Culler and J.P. Singh
• **Parallel Computer Architecture and Programming (CMU 15-418/618),** Todd Mowry and Brian Railing
• **Interconnection Network Architectures for High-Performance Computing**, Cyriel Minkenberg, IBM

https://www.systems.ethz.ch/sites/default/files/file/Spring2013_Courses/AdvCompNetw_Spring2013/13-hpc.pdf

Introduction

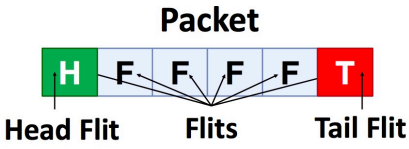
- Communications = overhead !!
- How should computation units be connected ?
  - For shared memory platforms, connecting memories with processors
  - For distributed memory platforms, need of a scalable high-performance network
    - Thousands of nodes exchanging data
- Relation between the topology of the network and the performance of global communication patterns
- Mathematical characteristics of networks + network models (latency, bandwidth, network protocols)

Introduction, Contd



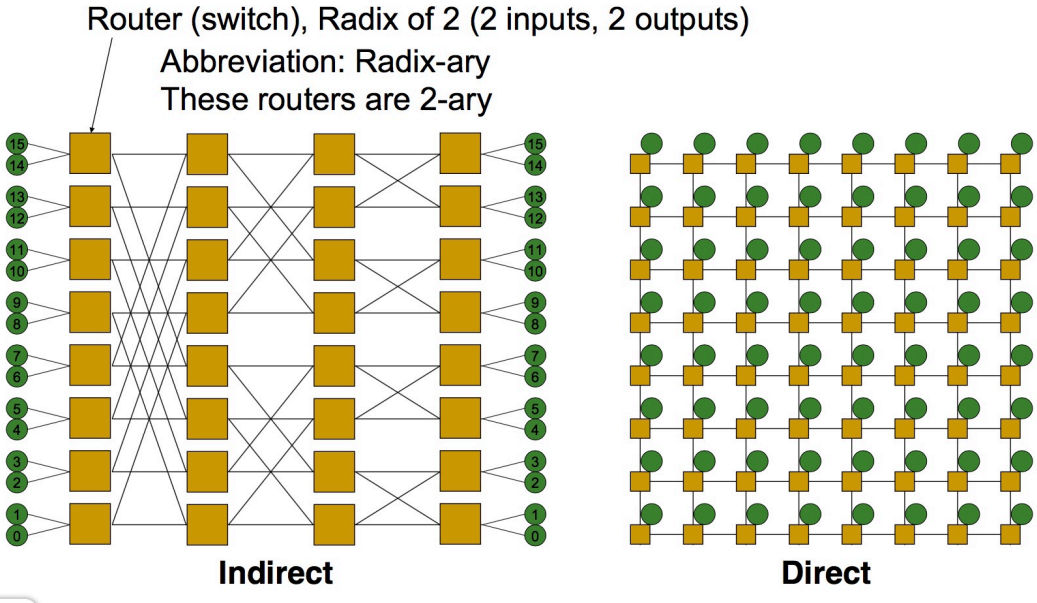
Terminology

- **Network interface**
  - Connects endpoints (e.g. cores) to network
  - Decouples computation/communication
- **Links**
  - Bundle of wires that carries a signal
- **Switch/router**
  - Connects fixed number of input channels to fixed number of output channels
- **Channel**
  - A single logical connection between routers/switches
- **Node**
  - A network endpoint connected to a router/switch
- **Message**
  - Unit of transfer for network clients (e.g. cores,memory)
- **Packet**
  - Unit of transfer for network
- **Flit**
  - Flow control digit
  - Unit of flow control within network



Terminology, Contd.

- **Direct or indirect networks**
  - Endpoints sit “inside” (direct) or “outside” (indirect) the network
    - E.g. mesh is direct; every node is both endpoint and switch



Formalism

- **Graph** G=(V,E)
  - V: switches and nodes
  - E: communication links
- ***Route***: (v_0, ..., v_k) path of length k between node 0 and node k, where (v_i,v_{i+1}) ∈ E
- ***Routing distance***
- ***Diameter****:* maximum length between two nodes
- ***Average distance:*** average number of hops across all valid routes
- ***Degree****:* number of input (output) channels of a node
- ***Bisection width:*** Minimum number of parallel connections that must be removed to have two equal parts

What Characterizes a Network?

Latency

- Time taken by a message to go from one node to another
  - A memory load that misses the cache has a latency of 200 cycles
  - A packet takes 20 ms to be sent from my computer to Google

Bandwidth (available bandwidth)

- The rate at which operations are performed
- *b = wf*
  - Where w is the width (in bytes) and f is the send frequency: f = 1 / t (in Hz)

Throughput (delivered bandwidth)

- How much bandwidth offered can be truly used
  - Memory can provide data to the processor at 25 GB/sec
  - A communication link can send 10 million messages per second

What Characterizes a Network? Contd.

Topology

- Physical network interconnection structure
- Specifies way switches are wired
- Affects routing, reliability, throughput, latency, building ease

Routing Algorithm

- How does a message get from source to destination
- Restricts all paths that messages can follow
- Many algorithms with different properties (static or adaptive)

Switching strategy

- How a message crosses a path
- Circuit switching vs. Packet switching

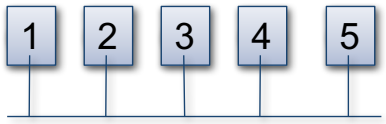
Flow control mechanism

- When a message (or piece of message) crosses a path, what happens when there is traffic? What do we store within the network?

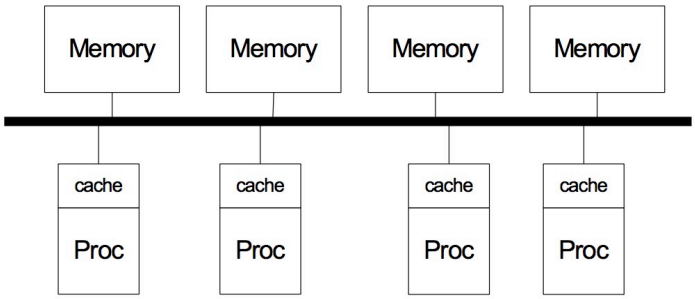
Goals

- Latency must be as small as possible
- High throughput
- As many concurrent transfers as possible
  - The bisection width gives the potential number of parallel connections
- Lowest possible cost/energy consumption

Bus (e.g. Ethernet)

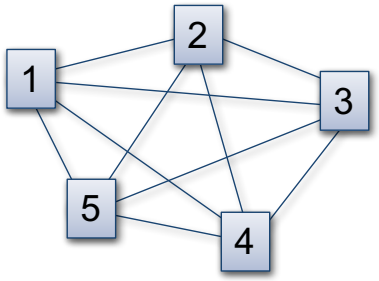


- Degree = 1
- Diameter = 1
- No routing
- Bisection width = 1
  - CSMA/CD protocol
  - Limited bus length



- Dynamic network
- Simplest one
- Lower cost

Fully Connected Network

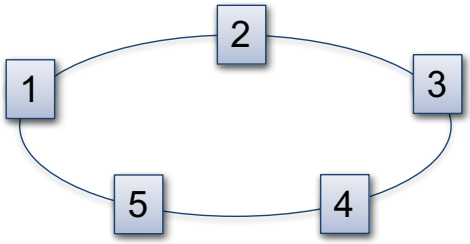


- Degree = n-1
  - too costly for large networks
- Diameter = 1
- Bisection width = $\lfloor n/2 \rfloor \lceil n/2 \rceil$

  When the network is cut in two parts, each node has a connection to n / 2 other nodes. There are n / 2 nodes like that.

- Static network
- Connection between every pair of nodes

Ring



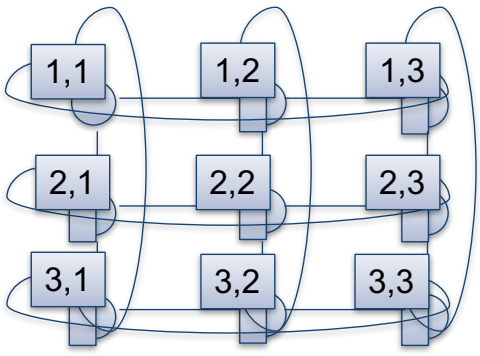
- Degree = 2
- Diameter = $\lfloor n/2 \rfloor$
  - slow for big networks
- Bisection width = 2

Static network

A node i is connected to nodes i+1 and i-1 modulo n.

Examples: FDDI, SCI, FiberChannel Arbitrated Loop, KSR1, IBM Cell

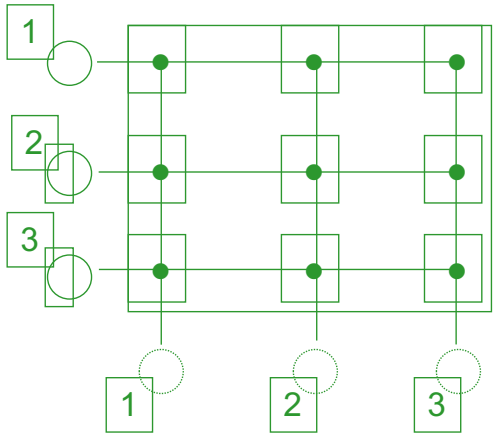
d-Dimensional Torus



- For d dimensions
- Degree = d
- Diameter = d ($\sqrt[d]{n} - 1$)
- Bisection width = ($\sqrt[d]{n}$) d–1

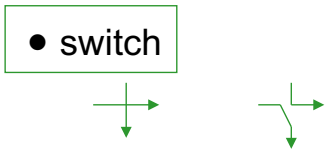
Static network

Crossbar



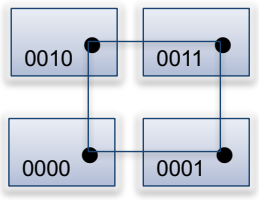
- Fast and costly (n^2 switches)
- Processor x memory

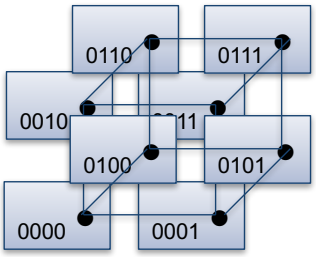
- Degree = 1
- Diameter = 2
- Bisection width = n/2
  - Ex: 4x4, 8x8, 16x16



Dynamic network

Hypercube



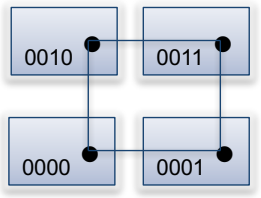


- **Hamming distance** =
  - Number of bits that differ in the representation of two numbers
  - Two nodes are connected if their Hamming distance is 1
  - Routing from x to y reduces the Hamming distance

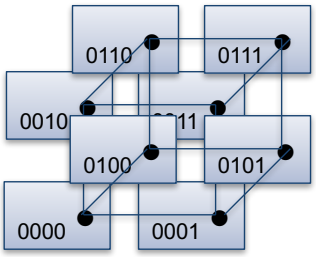
Static network

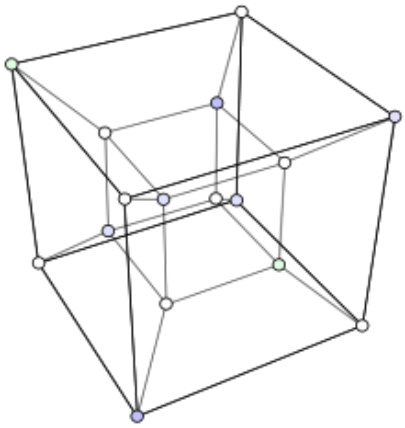
Hypercube, Contd

k dimensions, n= 2^k nodes



- Degree = k
- Diameter = k
- Bisection width = n/2
  - Two (k-1)-hypercubes are connected through n/2 links to produce a k-hypercube

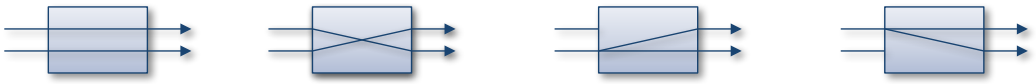




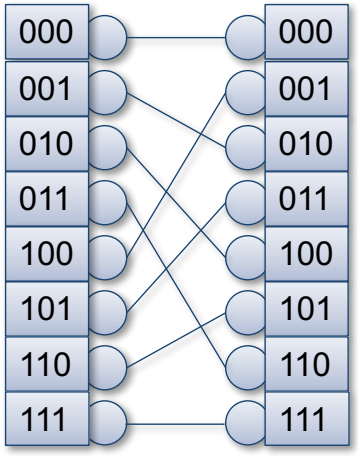
Intel iPSC/860, SGI Origin 2000

Omega Network

Basic block: 2x2 Shuffle



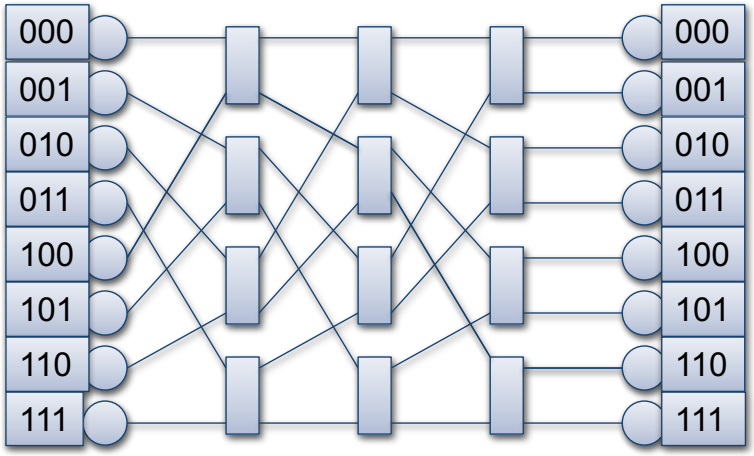
Perfect Shuffle



Omega Network, Contd.

$Log_2 n$ levels of 2x2 shuffle blocks

Dynamic network

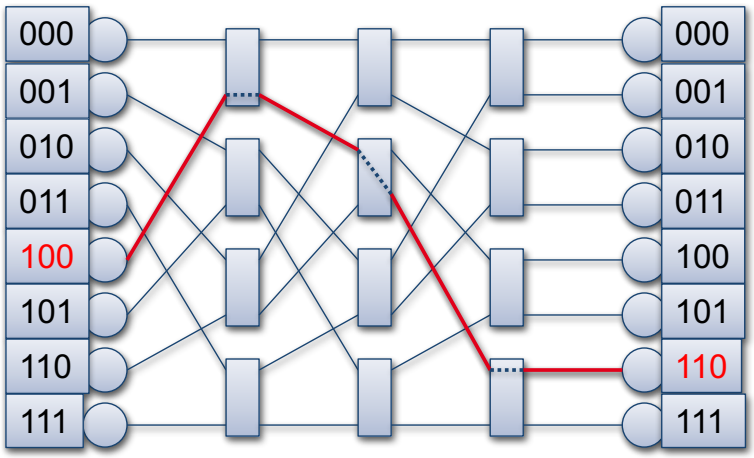


Level i looks for bit i
If 1 then go up
If 0 then go down

Omega Network, Contd.

$Log_2 n$ levels of 2x2 shuffle blocks

Dynamic network



Level i looks for bit i
If 1 then go down
If 0 then go up

Example 100 sends to 110

Omega Network, Contd.

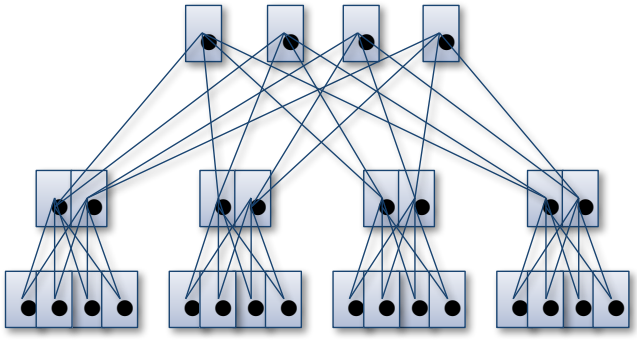
- n nodes
- (n/2) $\log_2 n$ blocks
- Degree = 2 for the nodes, 4 for the blocks
- Diameter = $\log_2 n$
- Bisection width = n/2
  - For a random permutation, n / 2 messages are supposed to cross the network in parallel
  - Extreme cases
    - If all the nodes want to go to 0, a single message in parallel
    - If each node sends a message, n parallel messages

Fat Tree /Clos Network

- Nodes = tree leaves
- The tree has a diameter of $2\log_2 n$
- A simple tree has a bisection width = 1
  - bottleneck

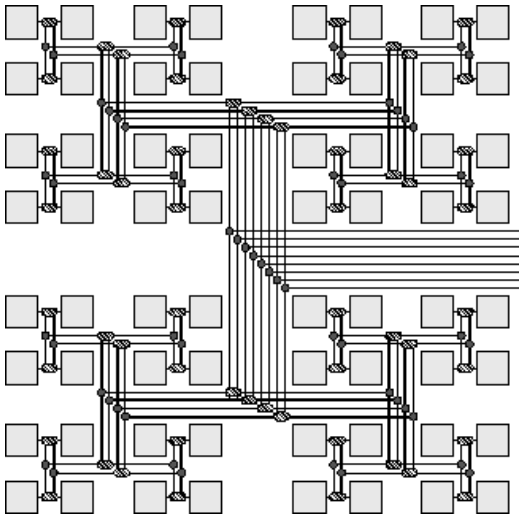
Fat Tree

- Links at level i have twice the capacity that those at level i-1
- At level i of the switches with 2^i inputs and 2^i outputs
- Also known as the Clos network



Fat Tree /Clos Network, Contd.

- **Routing**
  - Direct path to the lowest common parent
  - When there is an alternative one chooses at random
  - Fault-tolerant to nodes faults
- **Diameter:** $2\log_2 n$,
- **Bisection width:** n



CM-5

Summary

Network G with n nodes	Degree $g(G)$	Diameter $\delta(G)$	Edge connectivity	Bisection bandwidth
Complete Graph	$n-1$	1	$n-1$	$(\frac{n}{2})^2$
Linear Array	2	$n-1$	1	1
Ring	2	$\lfloor \frac{n}{2} \rfloor$	2	2
d-dimensional mesh $n = r^d$	$2d$	$d(\sqrt[d]{n}-1)$	d	$n^{\frac{d-1}{d}}$
d-dimensional torus $n = r^d$	$2d$	$d\lfloor \frac{\sqrt[d]{n}}{2} \rfloor$	$2d$	$2n^{\frac{d-1}{d}}$
k-dimensional hypercube $(n = 2^k)$	$\log n$	$\log n$	$\log n$	$\frac{n}{2}$
k-dimensional CCC network $(n = k2^k$ for $k \geq 3)$	3	$2k-1+\lfloor \frac{k}{2} \rfloor$	3	$\frac{n}{2k}$
Complete binary tree $(n = 2^k - 1)$	3	$2\log \frac{n+1}{2}$	1	1
k-ary d-Cube $(n = k^d)$	$2d$	$d\lfloor \frac{k}{2} \rfloor$	$2d$	$2k^{d-1}$

Switching

How a message crosses the network from one node to another

Circuit switching

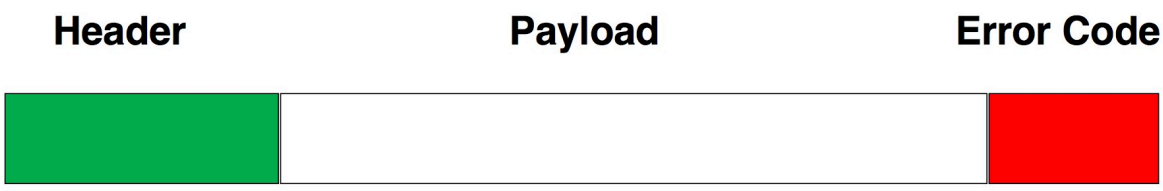
- A path is established from the source to the destination
- (no one else can use those links)
- All packets will take this path
- Phone Analogy (wired)
  + Faster and higher bandwidth
  - setting up and bringing down links slow

Packet switching

- A message is split into a sequence of packets that can be sent on different paths
- Better use of network resources
  + No setup, bring down time
  - Potentially slower (must dynamically switch)

Packet Switching, Packet Format

- **Header**
  - routing and control information
- **Payload**
  - carries data (non HW specific information)
  - can be further divided (framing, protocol stacks...)
- **Error Code**
  - generally at tail of packet so it can be generated on the way out



Packet Switching, routing

Two basic approaches to routing packets, based on what a switch does when a packet arrives

1) Store-and-forward

2) Cut-through
   - Virtual cut-through
   - Wormhole

Packet Switching: Store-and-Forward

- A packet is stored entirely before being forwarded

- **Drawbacks**
  - Need of a lot of memory to store incoming packets

- **Advantage**
  - Switching is done step by step.
  - Little danger of blocking

Packet Switching: Cut Through

A packet can arrive partially in a switch and leave its tail on the other nodes

- It can be on more than two switches

The re-send decision shall be taken immediately

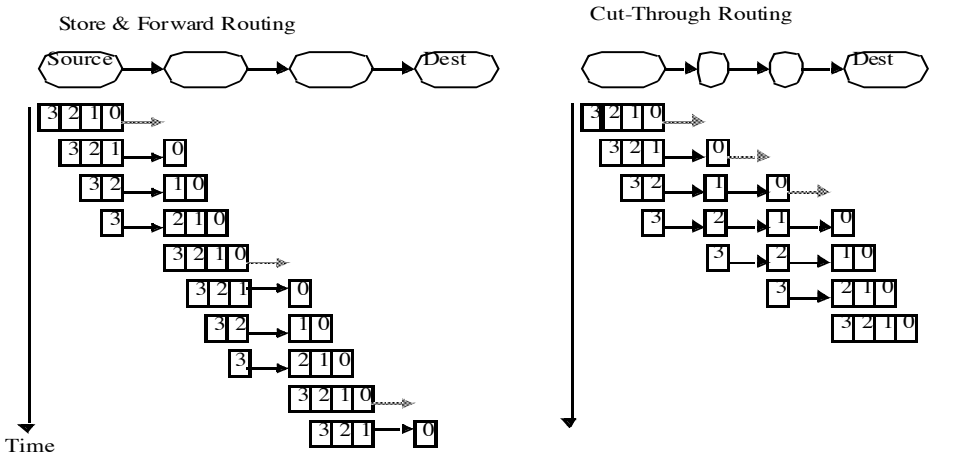
What happens if the head blocks?

- ***Cut-through***: collect the rest of the message where is the head

  Tends towards the store-and-forward model in case of strong contention

- ***Wormhole***: If the head blocks, the whole message hangs

Store & Forward vs Cut-Through



$h(n/b + \Delta)$ vs $n/b + h\,\Delta$

h: number of hops

n: message's size

b: bandwidth

Δ: routing delay per hop

Routing Algorithms

How do I know where a packet should go?

- Topology does not determine routing

Routing algorithms

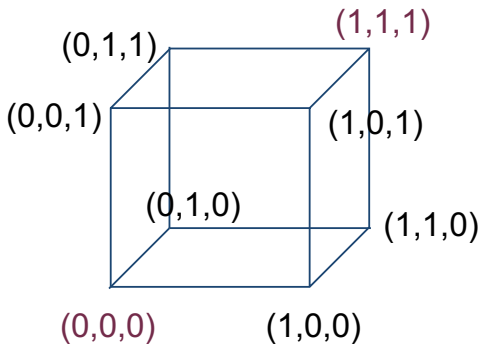
1) Arithmetic
2) Based on source
3) With a table (*table lookup*)
4) Adaptive: The route is determined by the state of the network (taking into account the contention)

Arithmetic Routing

On a regular topology, use simple arithmetic to determine the path

E.g., XY routing in a 3D torus

- The packet header contains a signed offset to the destination (by dimension)
- For each jump, switch +/- to reduce the offset in the dimension
- When x == 0 and y = 0, then we have reached the destination



Source-Based and Table-Based Routing

Source-based routing

- The source specifies the output port for each switch on the route
- Simple switches
- No control state
- Header removed whenever switch is crossed
- Used by Myrinet
- Can not become adaptive

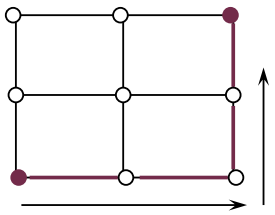
Table-based routing (*Table Lookup*)

- Small header: Contains a field that is an index in a table for the output port
- Large tables that must be kept up to date

Deterministic or Adaptive Routing

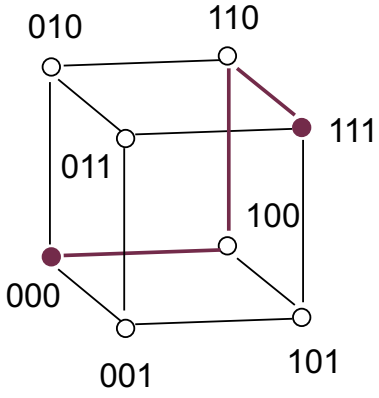
Deterministic

- Follows a pre-defined path
  - K-ary d-cube: dimensional routing
    (x1, y1) → (x2, y2)
    First Dx = x2 - x1,
    Then Dy = y2 - y1,
  - Tree: common ancestor
  - Simple algorithms can become blocking



Adaptive

- Route determined by contentions on the output port
  - Essential for fault tolerance
  - At least multi-paths
  - Can improve network utilization



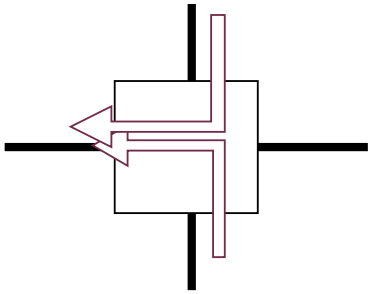
Contention

Two packets trying to use the same link at the same time

- Limited bufferization
- Loss?

Most networks of parallel machines hang

- Traffic can be returned to the source



Communication Performance: Latency

$Time(n)_{s-d}$ = overhead + routing delay + channel occupation + contention delay

- Overhead: time required to initiate sending and receiving a message
- Occupation = $(n + n_e) / b$
  - n: data size
  - n_e: packet envelop's size
- Routing delay
- Contention

Communications Performance: Bandwidth

What affects the local bandwidth?

- Packet density $b \times n/(n + n_e)$
- Routing delay $b \times n / (n + n_e + w\Delta)$

  Δ: number of cycles to wait for a routing decision

  w: channel width
- Contention

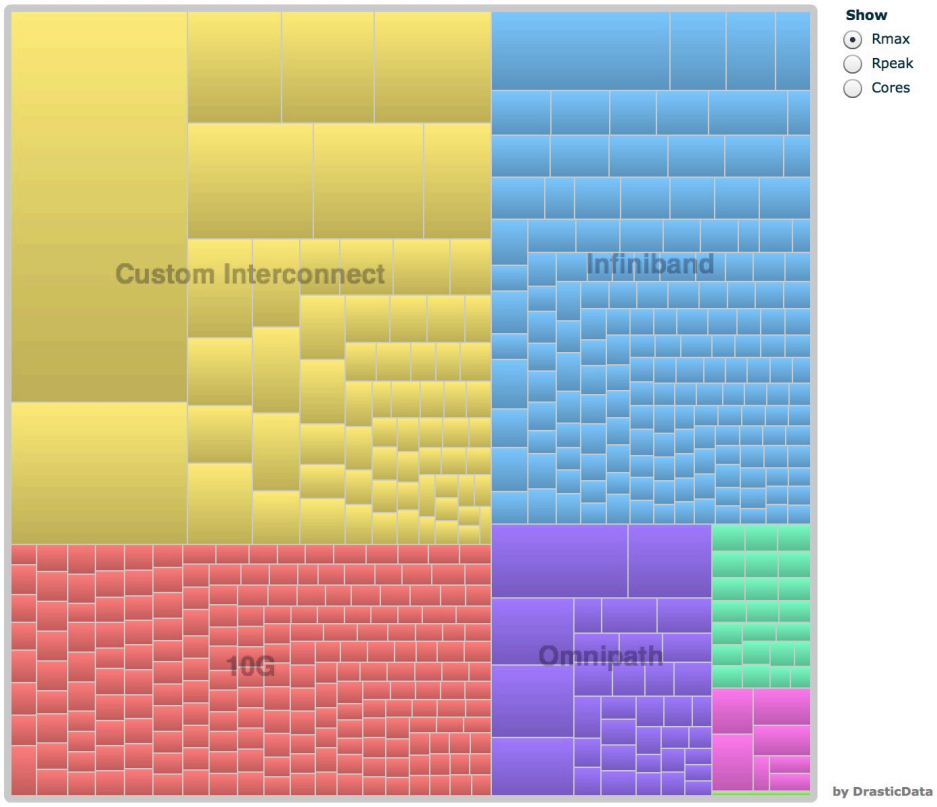
Aggregate bandwidth

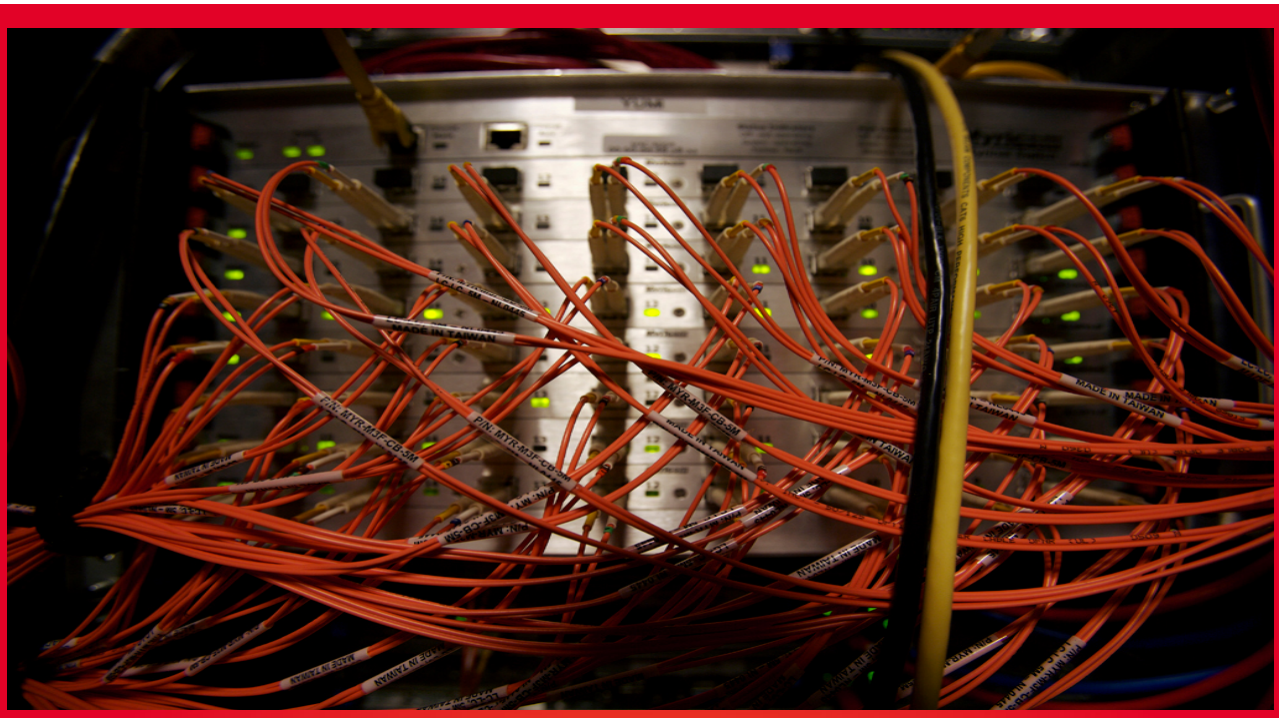
- Bisection width
  - Sum of the bandwidths of the smallest set of links that partition the network
  - Poor if non-uniform distribution of communications
- Total bandwidth of all channels

Ethernet, Infiniband, Omnipath

	Ethernet	InfiniBand	Omnipath
Commonly used in what kinds of network	Local area network(LAN) or wide area network(WAN)	Interprocess communication (IPC) network	Interprocess communication (IPC) network
Transmission medium	Copper/optical	Copper/optical	Copper/optical
Bandwidth	1Gb/10Gb	2.5Gb~120Gb	100Gb
Latency	High	Low	Low
Cost	Low	High	High

Top500





INVENTEURS DU MONDE NUMÉRIQUE

Interconnection networks

Name	Latency	Bandwidth	Topology	
Gigabit	100-150 us	1 Gb/s	Star or Fat Tree	Low cost for small systems
Infiniband 4x	3.5-7 us	10-20 Gb/s	Fat Tree	- Mostly used right now
Myrinet	3.5-7 us	2-8 Gb/s	Clos	- Mature, de facto standard - Switches at 256+256 ports - cost ~$500/card + port
NUMAlink4	1-2 us	8-16 Gb/s	Fat Tree	- SGI proprietary - Special uproc for I/Os - shmem
Quadrics	1-2 us	9 Gb/s	Fat Tree	- Costly
SCI/Dolphin	1-2 us	4 Gb/s	2D/3D Tore	- Costs more than Myrinet