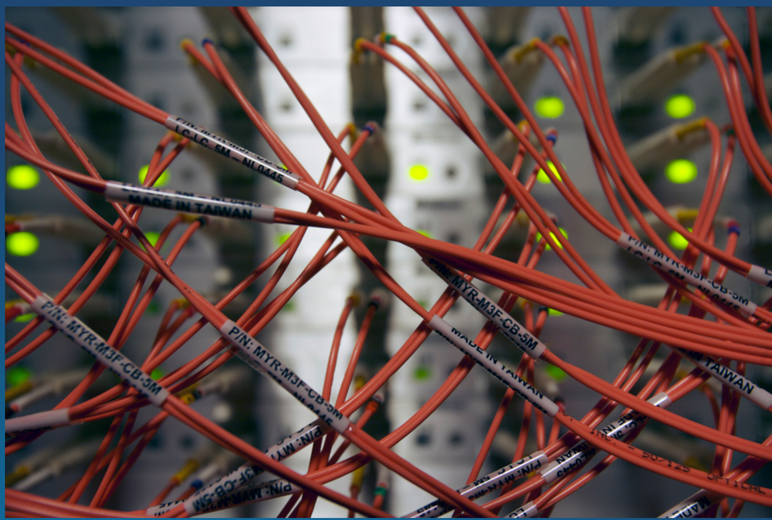
Interconnection Networks



Frédéric Desprez
INRIA

Some References

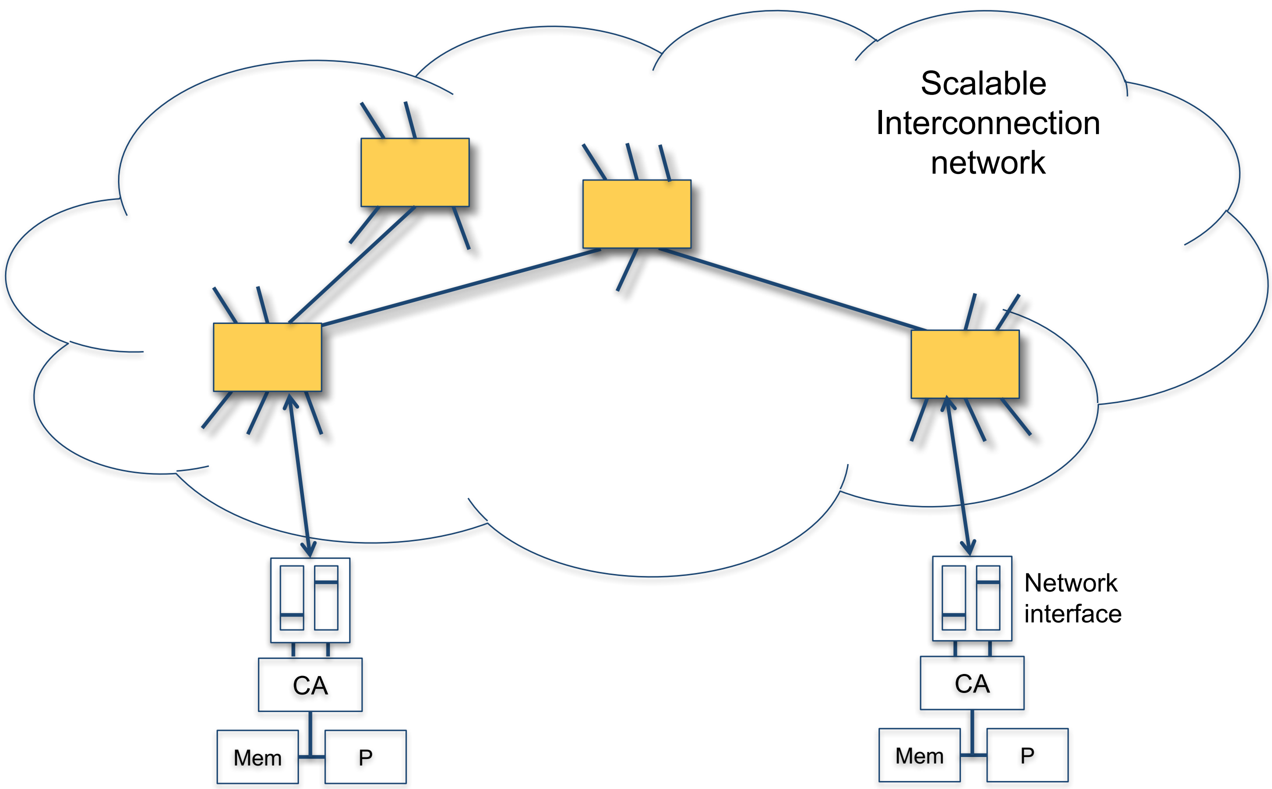
- **Parallel Programming – For Multicore and Cluster System,** T. Rauber, G. Rünger
- Lecture "**Calcul hautes performance – architectures et modèles de programmation**", Françoise Roch, Observatoire des Sciences de l'Univers de Grenoble Mesocentre CIMENT
- **4 visions about HPC - A chat**, X. Vigouroux, 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

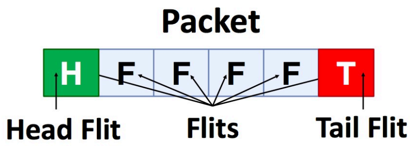
- Communications = overhead !!
- How should computation units be connected ?
  - For shared memory platforms, connecting memories with processors
  - For distributed memory platforms, need of a scalable high-performance network
    - Thousands of nodes exchanging data
- Relation between the topology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

Introduction, Contd



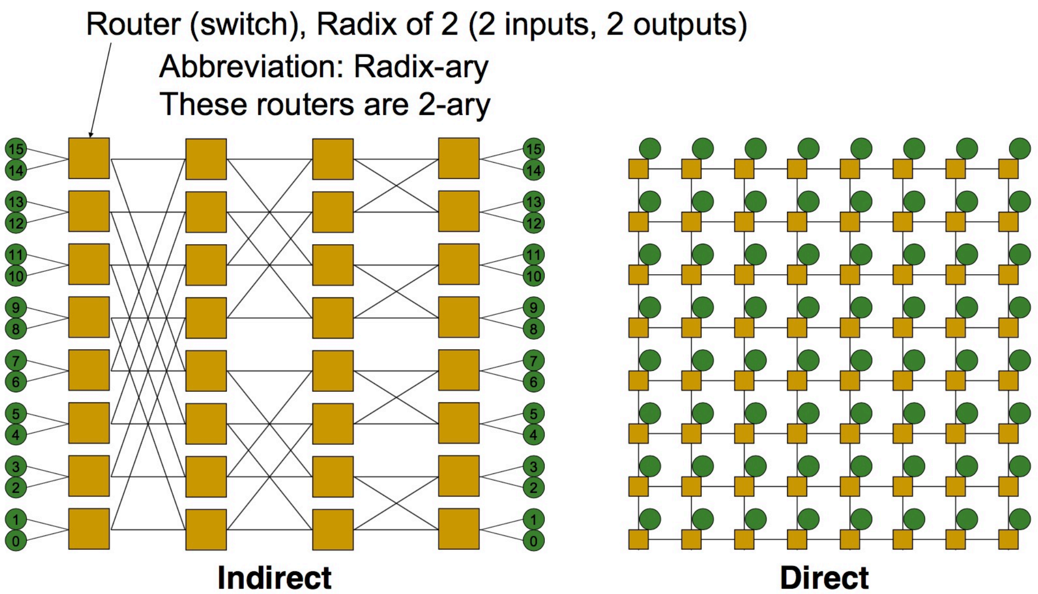
Terminology

- **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 $(v_i,v_{i+1}) \in E$
- ***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

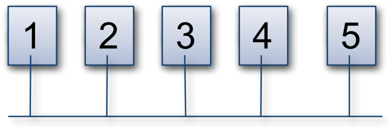
Flow control mechanism

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

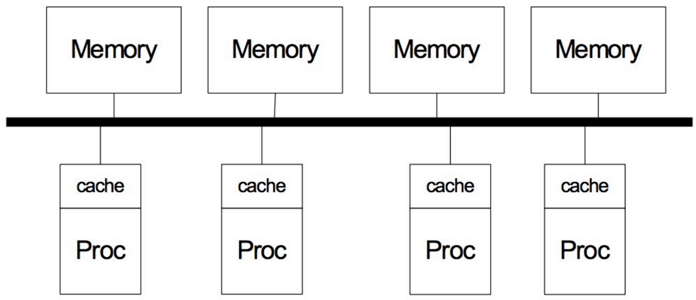
Goals

- 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)

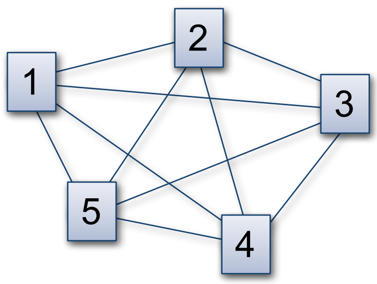


- Degree = 1
- Diameter = 1
- No routing
- Bisection width = 1
  - CSMA/CD protocol
  - Limited bus length



- Dynamic network
- Simplest one
- Lower cost

Fully Connected Network

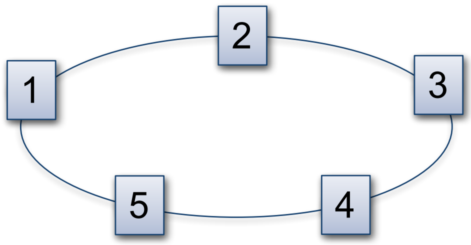


- Degree = n-1
  - too costly for large networks
- Diameter = 1
- Bisection width = $\lfloor n/2 \rfloor \lceil n/2 \rceil$

  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



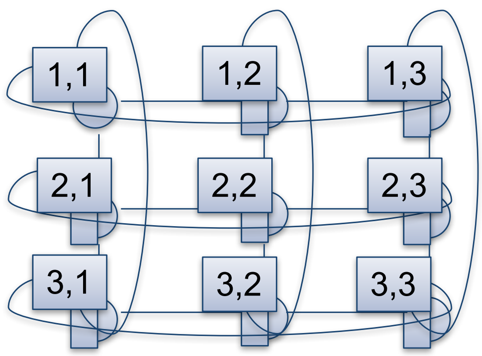
- Degree = 2
- Diameter = ⌊n/2⌋
  - slow for big networks
- Bisection width = 2

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

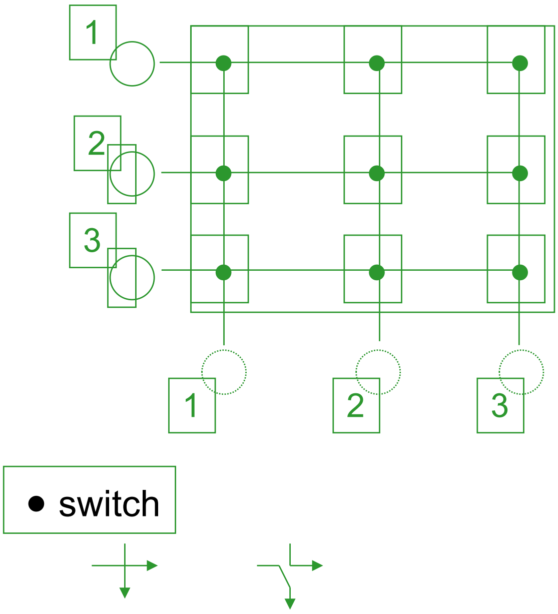
d-Dimensional Torus



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

Static network

Crossbar

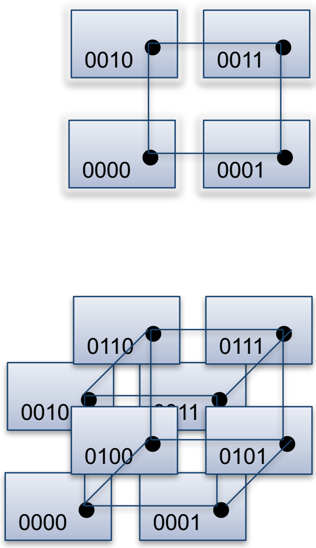


Dynamic network

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

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

Hypercube

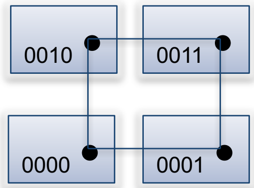


Static network

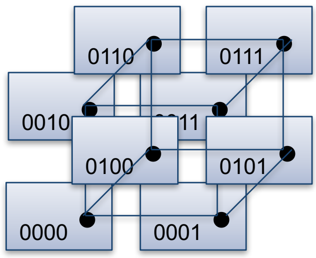
- **Hamming distance** =
  - Number of bits that differ in the representation of two numbers
  - Two nodes are connected if their Hamming distance is 1
  - Routing from x to y reduces the Hamming distance

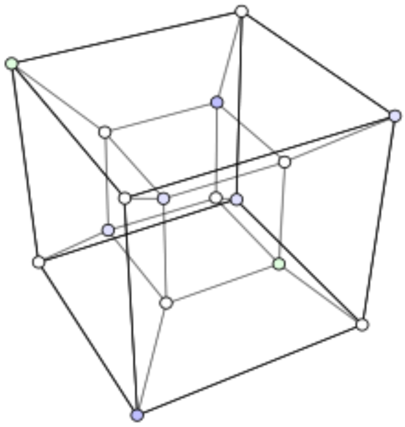
Hypercube, Contd

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



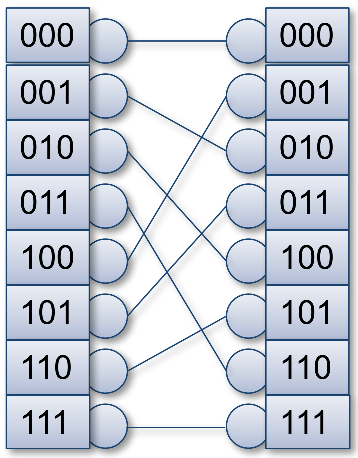


Intel iPSC/860,
SGI Origin 2000

Omega Network

Basic block: 2x2 Shuffle

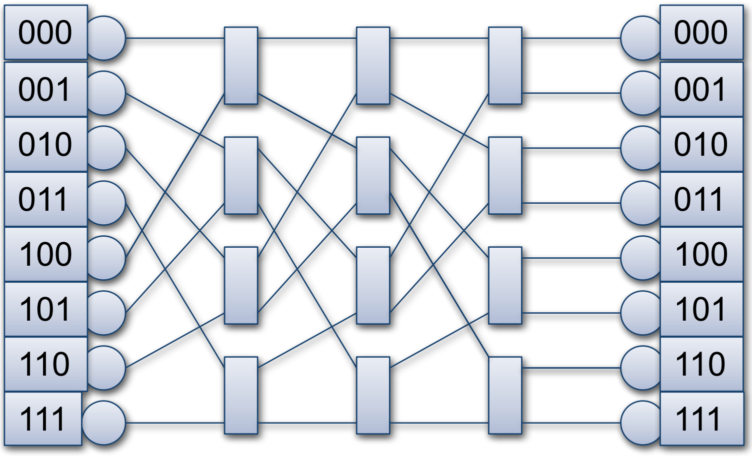
Perfect Shuffle



Omega Network, Contd.

Log$_2$n levels of 2x2 shuffle blocks

Dynamic network

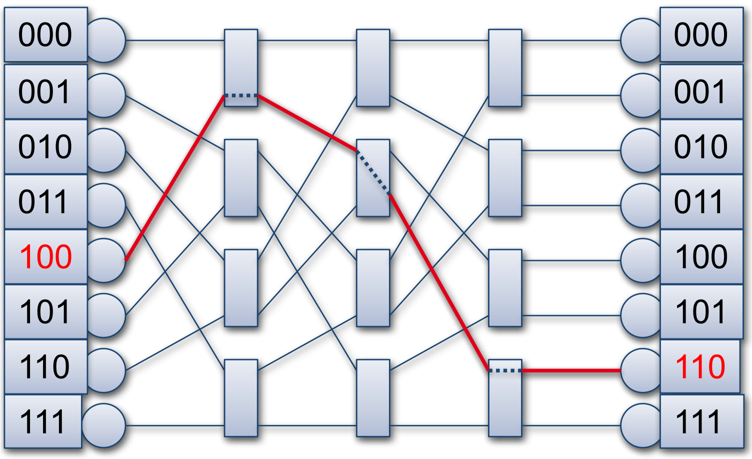


Level i looks for bit i
If 1 then go down
If 0 then go up

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.

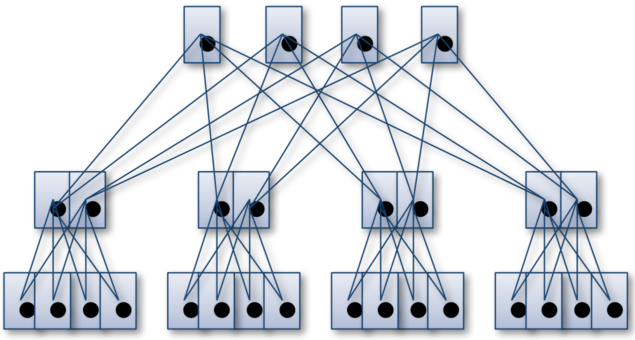
- n nodes
- $(n/2) \log_2 n$ blocks
- Degree = 2 for the nodes, 4 for the blocks
- Diameter = $\log_2 n$
- Bisection width = n/2
  - For a random permutation, n / 2 messages are supposed to cross the network in parallel
  - Extreme cases
    - If all the nodes want to go to 0, a single message in parallel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

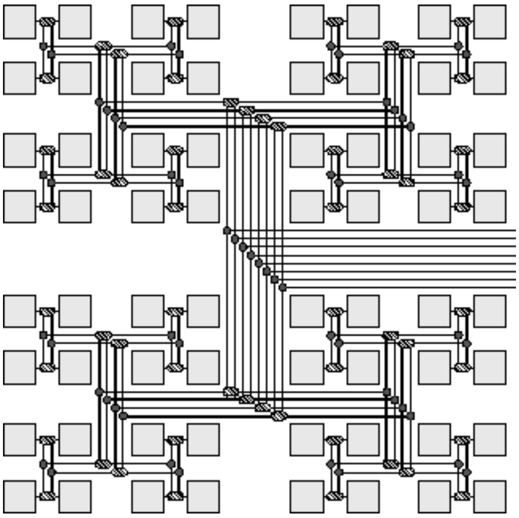
Fat Tree

- 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 $(n=k^d)$	$2d$	$d\lfloor\frac{k}{2}\rfloor$	$2d$	$2k^{d-1}$