

---

**Today's topics:**

**Some basic interconnect network concepts**

**Topology**

## **Exploiting Concurrency**

---

- **In multiple cores or multiple sockets**
  - **communication takes center stage**
- **Ubiquitous networking**
  - **LAN & WAN space = Internet (you already know this stuff)**
    - » **key is dealing with chaos**
      - **arbitrary machine platforms**
        - big Endian vs. little
        - varying OS management layer
      - **arbitrary topology**
        - must support continual change
        - current user base 1.6 billion
    - » **result – general but inefficient**
      - **price to be paid for generality**
      - **layer model of who supports what**
        - application, OS, NIC, router
        - 7 layer ISO model
          - which never is really implemented
          - but it's the basic idea
    - » **doesn't work in high performance parallel system world**
      - **where both performance and efficiency become critical**

## High Performance Systems

---

- **One or multi-socket**
  - **some cost functions change but game is similar**
    - » **note common trend**
      - **multi-socket approach continually moves on-socket**
        - perhaps with some low-level implementation changes
- **SAN – system area network**
  - **focus on performance, reliability, packaging, and efficiency**
    - » **performance**
      - **minimum packet latency for an unloaded system**
      - **average packet latency**
        - under various load factors
    - » **reliability**
      - **SAN's consider failure as rare**
        - should provide some fault tolerance
          - K's to M's of components → something is likely to fail
    - » **packaging**
      - **minimize SKU's**
    - » **efficiency**
      - **ED or ED<sup>2</sup> product combined metric considerations**

## SAN Difference

---

- **Proprietary vs. standards based?**
  - **company X makes mondo parallel gizmo**
    - » see [www.top500.org](http://www.top500.org)
    - » they also create their own interconnect system
- **Datacenters and the “Cloud” are a bit different**
  - **in-cabinet (in-rack)**
    - » **possibly proprietary**
      - **top of rack switch**
        - blade to blade efficient
        - convert to standard oriented comm between cabinets
  - **between cabinets**
    - » **often more standards oriented**
      - **hypertransport**
      - **QPI**
      - **xGigE: x = 1/10/40/100**
    - » **switches**
      - **CISCO is the market leader**
        - same switches for IP and SAN traffic

## 3 Essential Components

---

- **Topology**
  - **graph of terminals and switches**
    - » **focus today**
- **Routing Algorithm**
  - **how does a packet or message get from source to destination**
    - » **heavy impact on lots of switch micro-architecture choices**
      - buffering
      - virtual channels
      - flow control
    - » **deterministic, oblivious, adaptive**
      - focus of the next lecture
- **Switch micro-architecture**
  - **router/switch architecture**
    - » **implement the routing algorithm**
    - » **& support the traffic model**
- **Key - all 3 are tightly coupled**

## 2 Variants: Network Type

---

- **Indirect networks**
  - **2 kinds of switches → 2 SKU's**
    - » **those that connect to terminals & switches**
      - terminals
        - processors, storage, ...
        - send and receive messages/packets
      - other switches
        - that form the core
    - » **those that connect only to other switches**
      - sometimes called "core" switches
- **Direct networks**
  - **1 type of switch → 1 SKU**
    - » **each switch has some number of ports**
      - some ports connect to other switches
      - some ports connect to terminals

## 2 Variants: Switching Type

- **Circuit switching**
  - **create electrical path from source to destination**
    - » **used in old telephone networks**
    - » **super efficient**
      - **no intermediate header examination, buffering, etc.**
      - **real time performance was easy**
        - **busy vs. good to go**
    - » **low throughput**
      - **no traffic interleaving**
- **Packet switching**
  - **break transaction up into packets**
    - » **fixed or variable size**
    - » **at each hop**
      - **examine destination, select route, send if route available**
        - **note extra work per hop → hop count is an important metric**
    - » **traffic interleaved → increased resource utilization and throughput**

## Topology

- **Consider first**
  - **heavy influence on other interconnect decisions**
    - » **routing algorithm and switch architecture**
  - **BUT**
    - » **except for that influence it might be the least important**
- **Open ended game**
  - **no way to cover all the options**
    - » **e.g. describe all graphs**
  - **lots of tower of Babel effects**
    - » **topologically donut and coffee cup are the same**
      - **as are fat-tree (Lelerson) & folded-Clos (Dally)**
- **Hierarchy is possible**
  - **different topologies may occur at different levels**
- **Today**
  - **focus on some basic options**

## Bus

- **Simplest and first interconnect**
  - we've seen adv. in snooping SMP configurations



- **Requires arbitration**
  - synchronous – can pipeline xfer & master
  - asynchronous – detect collision and backoff
    - » Ethernet choice
- **Problem: long = slow**
  - scalability, signal integrity,
- **Improvements**
  - slotted bus – TDM style
  - wider to support multiple transactions

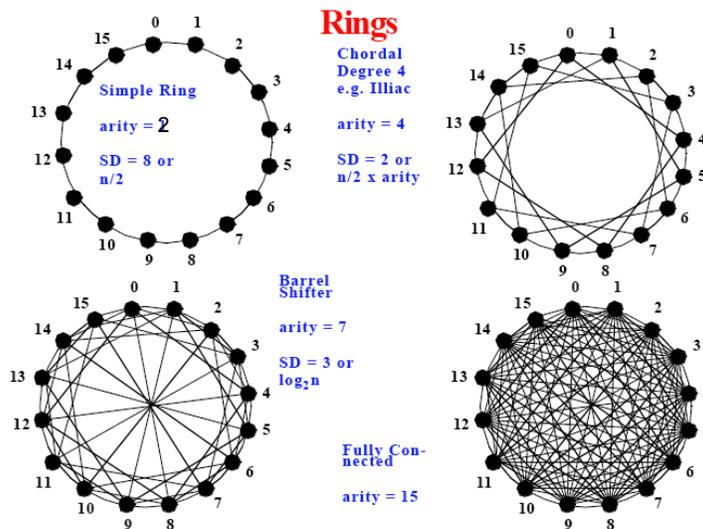
## Some Cost Issues

- **Radix of the switch**
  - number of inputs & outputs
    - » here we'll consider bi-directional links
      - # = radix (sometimes called "arity")
      - NOTE: some literature: radix := # inputs + # outputs
        - question link is 1 or 2 channels
        - 1 channel requires arbitration like the bus
        - 2 unidirectional channels/link is obvious choice
          - config. cost and cabling errors get reduced
- **Switching Diameter**
  - worst case hop count
    - » effectively a measure of what happens when locality is rare
- **ITRS constraints**
  - pin count and per pin bandwidth expected to be flat
  - choice
    - » increase radix → decrease link bandwidth → decreased hops
    - » tough choice

## Performance Issues

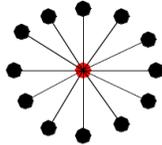
- **Bisection bandwidth**
  - **cut network in half – bandwidth between halves**
    - » **for some topologies choice of half will yield different values**
- **Path diversity**
  - **how many shortest paths are there**
  - **utility will depend on routing algorithm**
- **Per link bandwidth**
  - **pin toggle rate \* number of wires (or waveguides)**
  - **diversion**
    - » **additional factor available with RF or optical channels**
      - **# of lambda's**
      - **we'll ignore these new options for now**
        - **on the horizon sure but both have some issues**

## Simple Direct Network



## Simple Indirect/Direct Network

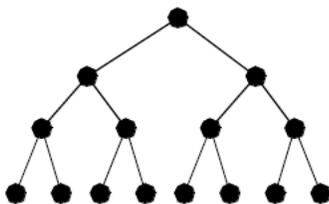
- **Star**



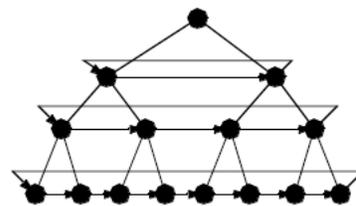
- **Weird radix**

- center node could connect to others = direct
- or be different from periphery = indirect
  - » typically periphery is the NIC
    - good for LAN's
    - horrible for SAN's
      - congestion at center node
      - over provision center node is the common out
  - » clear scaling problem

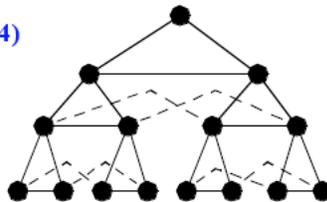
## “Skinny” Trees



Simple e.g.  
DDM1  
(leaf fanout = 4)



Level connected e.g. NonVon

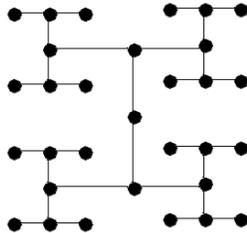


Sibling connected  
e.g. Pepe

## Space Filling Tree's

- **Note boards and chips are rectangular**
  - **even better if they are close to square**

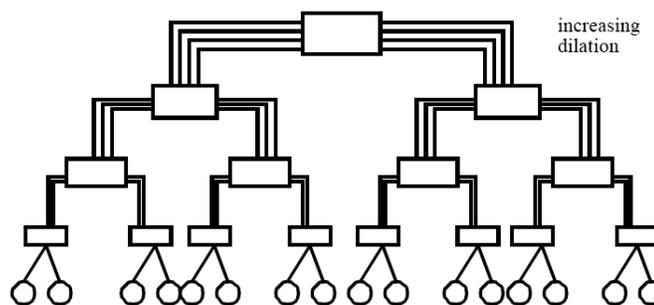
- **H-tree**



- **uniform spacing of terminal nodes**
  - » **often used for reducing skew in clock trees**
    - **or memories with multiple mats or chips**
      - **where broadcast to all is the norm**
- **regular wiring pattern**
  - » **eases floor planning**
  - » **important for on-chip – relatively useless in a warehouse**

## Fat Trees

- **This one is tapered**



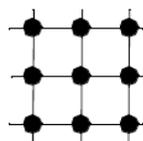
- **Questions**

- **what changes to support full bisection bandwidth?**
- **how can a single switch type be used to construct a fat tree?**

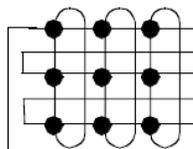
## Leiserson's Original Idea

- **Routing**
  - no LCA routing – always go to the top “core” level
  - random up choice
    - » load balancing if you don't really know what's going on
  - deterministic down choice
- **First real machine to employ this concept**
  - TMI CM-5
- **Now a common choice for supercomputers and data center interconnects**
- **How about**
  - expansion?
  - cabling complexity?

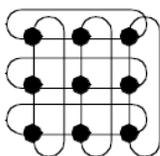
## 2D Quad Meshes



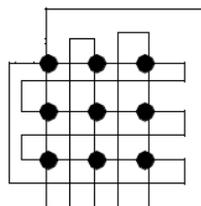
Unwrapped



Iliac Mesh



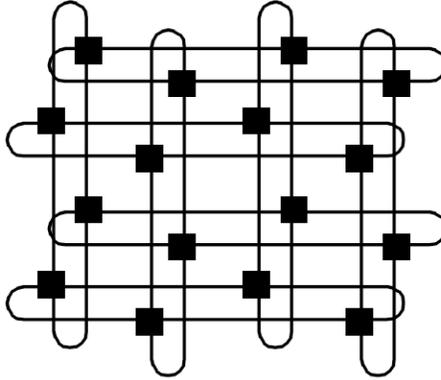
Torus



Twisted Torus

## Folded Torus

- Same mesh idea but keep wire length's the same
  - Bill Dally idea



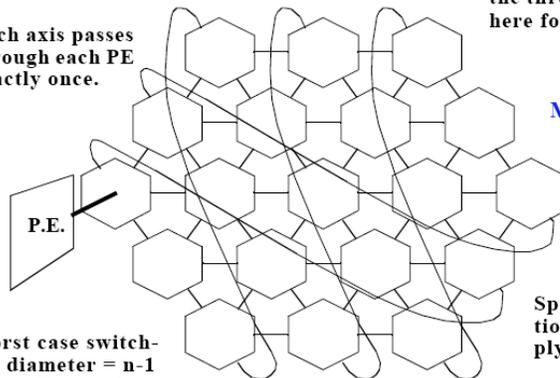
## Hex Mesh

Continuous processing surface.

Each axis passes through each PE exactly once.

Only a single axis of the three are wrapped here for clarity.

Mayfly

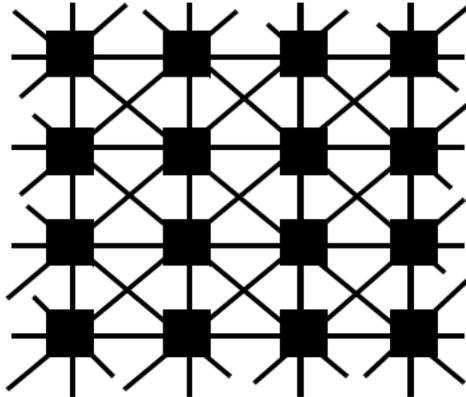


Worst case switching diameter =  $n-1$  for an E- $n$  surface.

Sparse population simple - simply short the

E-3 surface contains 19 processing ele-

## Oct/X Mesh

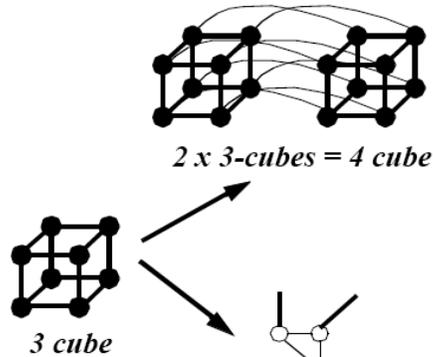
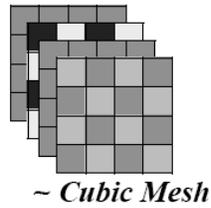


Note non-planar wiring – occurs in all meshes > hex

## Motivating 3D Interconnects

- **Harder to draw if you're a geek**
  - where's an artist when you need one?
- **Real world is 3D**
  - lots of modeling problems fall into a 3D space
  - consider Ocean
    - » divide world into cubes
      - 6 neighbor cells
    - » simulate via standard relaxation method
      - calculate inside values from boundary
      - calculate new boundary values
      - exchange boundary values with 6 neighbors
      - continue until
        - you or the machine dies
        - or you get the right/converged answer per time step
      - move to next time step
        - continue until you've had enough

## 3D Interconnects



Replace each node with  
proper size cycle  
e.g. 3-cube-connected-cycle

## n-Dimensional Networks

- **Several options**
  - **start simple – binary n-cube**
    - » no way I can draw them
    - » concept is simple
      - each node has an n-bit index
      - link to each node @ Hamming distance = 1
      - radix = n
  - **real machines**
    - » CalTech Cosmic Cube
    - » Intel IPSC
    - » nCube
  - **fallen from grace**
    - » wiring complexity and packaging prove too costly
    - » radix and link bandwidth trade-off problem

## Multistage Networks

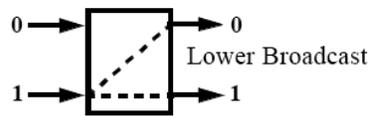
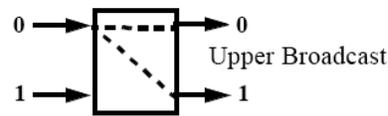
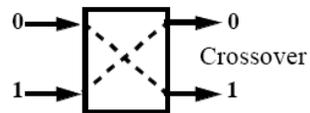
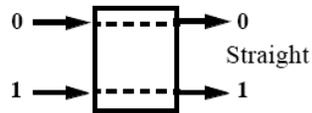
- **Basis – 2x2 Quine Switch**

- **4 states**

- » **note not all modes used in practice**

- » **consider the difference**

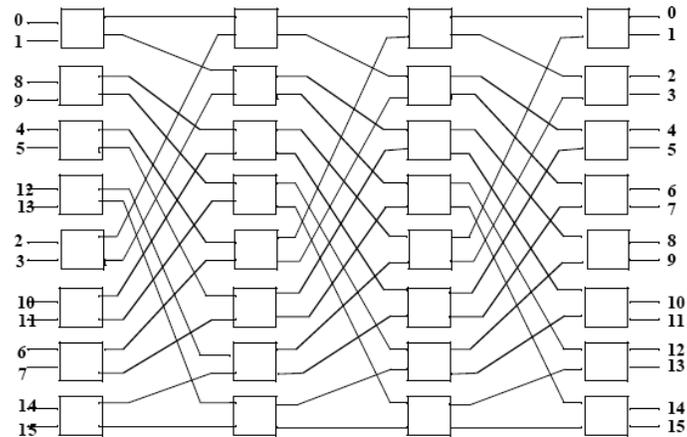
- **asynchronous vs. synchronous traffic**



## Shuffle/Omega/Banyan Networks

- **Tower of Babel syndrome**

- **routing algorithm? expandability? bisection B/W? stages?**

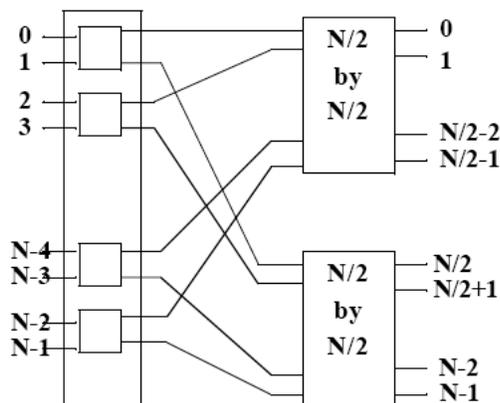


## Shuffle (cont'd)

- **Routing simple**
  - binary destination value
  - 0 → top, 1 → bottom
- **Expanding**
  - no copy and add a stage
    - » even though  $\log_2(T)$  stages required
  - unwire half of everything
    - » add some stuff and rewire
  - blocking
  - complex wiring pattern
    - » albeit regular – e.g. shuffle
- **Real? machines**
  - Uofl Cedar
  - NYU Ultra and IBM RP-3
    - » took advantage of combining options
      - broadcast & multi-cast options

## Recursive Construction: Baseline Networks

- **Modularizing wiring via recursive structure**



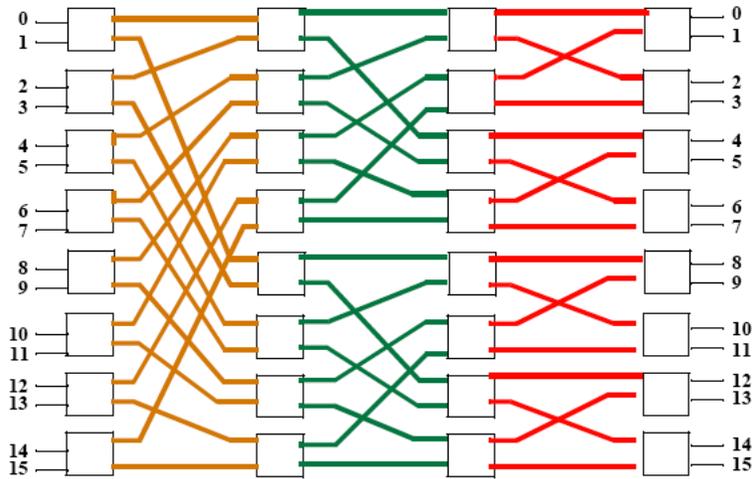
Double Configuration to  $N \times N$  by adding  $N/2$  base switches plus another  $N/2$  box - and wire them up

Routing algorithm is the same

So are the blocking and combining possibilities

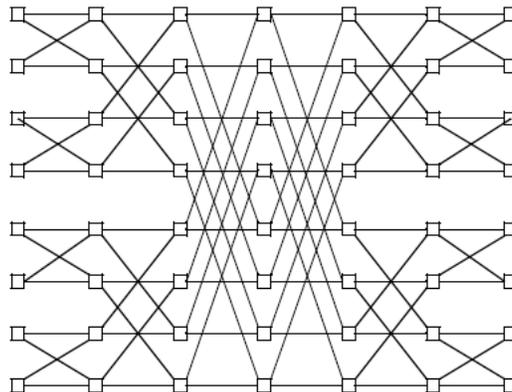
Also called:  
*Butterfly Networks*

## 16x16 Baseline Example



## Benes Networks

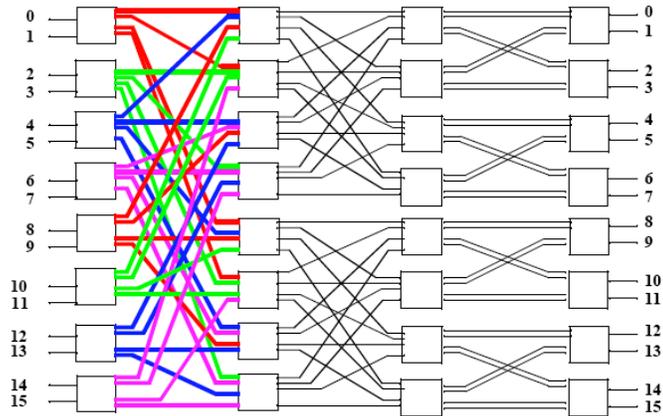
- Back to back butterfly's



- fold in the middle
  - what do you end up with?

## Dilating Paths

- **Increased cost but fault tolerant**
  - **to both failure and congestion**



## And Finally Crossbars

- **True non-blocking behavior**
  - **no destination conflict then there is a path**
  - **problem  $N^2$  switches**
- **What about scheduling**
  - **simple**
- **Reducing switch count**
  - **cross-bars of cross-bars**
    - » **recursive game again**
    - » **first done by Shannon's gang at AT&T**
      - **In particular Clos**
  - **scheduling**
    - » **easy for synchronous traffic**
    - » **harder for asych traffic**
  - **64 x 64 YARC**
    - » **array of 8x8 of 8x8's**

## Concluding Remarks

---

- **Lot's of topologies**
  - **this lecture presented some of the options**
- **But a lot of other things are important**
  - **routing algorithm**
    - » **next**
  - **switch micro-architecture and examples**
    - » **a week from now**
- **Key**
  - **complex space**
  - **increasing importance as we move to multi-**
    - » **cores or sockets**
- **Great reference text**
  - **William J. Dally and Brian Towles, Principles and Practices of Interconnection Networks Morgan Kaufmann, 2004**
- **Research literature is more than extensive**