CS 498
Hot Topics in High Performance Computing

Networks and Fault Tolerance

9. Routing and Flow Control

Intro

- What did we learn in the last lecture
  - Topology metrics
    - Including minimum diameter of directed and undirected topologies (Moore bound)
  - Common network topologies
- What will we learn today
  - Some more topologies
  - Routing (schemes and metrics)
  - Flow control

Butterfly vs. Hypercube

- One can “flatten” a 2-ary k-fly by collapsing all straight connections into one node
  - Class Question: What is the degree of the resulting network?

Butterfly vs. Hypercube vs. Flat Fly

- One can “flatten” a 2-ary k-fly by collapsing all straight connections into one node
  - $\log_2(P)$
- High-radix routers become more common, one can also “flatten” higher-arity flys
  - Results in so called flat-fly networks
  - Similar to generalized hypercubes
  - Less connections (saves straight connections) → cheaper

More Topologies

- Many more topologies exist
- Most are hybrids of the discussed ones
- Research is ongoing but:
  - Topologies alone are not enough
  - Routing and Flow Control determines performance of a network!
  - Will be discussed next

What is Routing?

- Select a path from a source PE to a destination PE
  - Picking the routing function is the next logical step after picking a topology!
- Topology determines ideal performance
  - Routing determines how much of this is realized!
- A quick example: 8-PE ring
  - Only binary routing decision (right or left)

8-PE Ring

- Routing options:
  - Shortest path: pick direction with shortest distance to target
  - Uniform random: randomly pick direction
  - Weighted random: pick randomly but with probability 1-x/8 short and x/8 long (x is the min. distance)
  - Adaptive: send packet in the direction where the outgoing channel has the lowest load

8-PE Ring Tornado Pattern

- Each PE i sends to PE i+3 mod 8
- The throughput of each pattern is:
  - Shortest path: 0.33
  - Random: 0.40
  - Weighted Random: 0.53
  - Adaptive: 0.53

- PS: the same problem was given as part of a Ph.D. qualifying exam and 90% picked shortest path!

Taxonomy of Routing Functions

- Oblivious: does not utilize the state of the network for routing decisions
  - Deterministic: chooses always the same path between PEs x and y
  - Random: picks a path between x and y randomly
- Adaptive: utilizes the state of the network for routing decisions
  - Local adaptive: only considers local ports
  - Global adaptive: uses global knowledge

Minimal vs. Non-minimal Routing

- Minimal routing: all paths have minimal length
  - Same properties as minimum diameter networks
    - Lower power consumption and latency

- Non-minimal routing can lead to higher throughput for certain traffic patterns and certain topologies (not generally though)

Deterministic Routing - Butterfly

- Example: 2-ary 3-fly with 2 PEs at each leaf
  - Routing from 2->3 (top->bottom)
  - Destination address as binary: 011
    - Left, right, right
- Can be generalized to n-ary numbers for n-ary fly networks
  - Very simple routing (element in stage k picks k-th number to choose egress port)

Deterministic Routing - Cubes

- Dimension-order (e-cube) routing in k-ary n-cube networks
  - Interpret destination as radix-k number
  - Each digit is used to select dimension
  - Route in dimension order until destination reached! (each dim. may require multiple hops)
- Example: 2-d 4x4 torus routing from 1,1 to 3,2
  - Used in Cray T3D

Random Routing

- Valiant's random routing: route from PE x to y through random intermediate node z. Routes x->z and z->y are deterministic.
  - Balances traffic
  - Increases latency and channel utilization some topologies (cubes)
- Provides good worst-case performance
  - Asymptotically optimal
  - Looses locality (nearest neighbor traffic becomes global – is this practical?)

Minimal Random Routing on Flys

- Benes networks: pick random middle-level node!
  - Same minimal length, good load distribution
- Not minimal but close: fat-trees (pick random top-level node)
  - Only log(P) paths will be longer
  - Same good load distribution (turns every pattern into a random pattern, good worst-case performance)
    - Example: CM-5

Minimal Random Routing on Cubes

- K-ary n-cubes have multiple minimal routes for k>1 and d>1
  - Those routes form a minimal sub-cube (or quadrant) which can be used to distribute load
  - Pick random node within this minimal sub-cube
  - Preserves locality!
  - Does not improve worst-case performance ☹
    - See Tornado pattern
  - Example: Avici Terabit Switch Router (TSR) – Internet SONET router

Adaptive Routing

- Local vs. global adaptive routing
  - Local often has bad worst-case performance
    - Show example
  - Global state is hard to maintain in each router and hard to process fast enough (log(P))
- Minimal Adaptive Routing
  - Adaptively pick from set of minimal routes
  - Example: k-ary n-cube: IBM Blue Gene/P, Myrinet