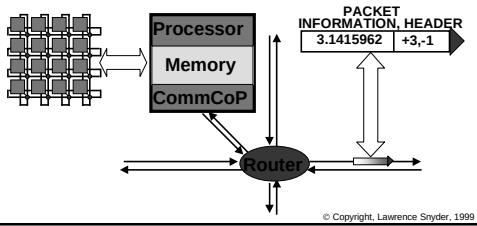
Interprocessor Communication

There are two main differences between parallel computers & sequential computers: Multiple processors and the hardware to connect them together. That hardware is the most crucial part of the design

Basics Of Network Routing

Routers can be integrated with the processors or they can be collected into a separate network component -- logically the same

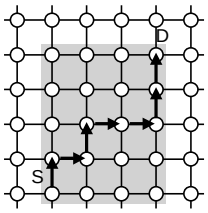


Goals Of Network Routing

Must have --
- High throughput
- Low latency

Must be --
- Deadlock-free
- Livelock-free
- Starvation-free

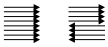
Should be insensitive to --
- Congestion
- Bursts
- Faults



A hard design is essential, there are no algorithmic advantages

Physical Connection

The wires connecting two switches can be unidirectional with information flow alternating directions, or bidirectional with half the wires permanently assigned in each direction
- For sustained information flow in both directions, the bandwidth and latency are the same
- With one packet in the network, the latency is the same (first flit arrives at the same time), but the bandwidth is doubled



A "flit" is a flow control unit

A "phit" is a physical transmission unit

Destination Addressing

- In a regular topology the switches can compute the path to the destination knowing only the destination address
  - Fitting the destination address into the first phit allows the node to begin routing immediately
- For irregular networks it is common to use "source" routing, i.e. the route is computed before injection into the network and is prefixed to the information
  - Each link address is removed as it's used

Transport Approaches -- Circuit Switching

Circuit switching
- A static path is set up between source and destination nodes
- Once established, information is then transmitted in pipelined fashion along the path
- The path is "torn down" after when the transmission is over

- Good for large quantities of data
- Set up/Tear down are overhead

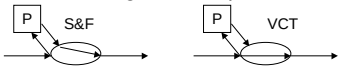
Circuit switching is inherited from telephony switching

Transport Approaches -- Packet Switching

In packet switching, the transmission is divided up into units (packets) with routing information prefixed onto each

- Each packet treated independently, preventing any transmission from monopolizing resources
- Biased to favor short transmissions
- Allows for adaptivity
- Header overhead; pipelining is less effective

- Original formulation used “store and forward”
- Virtual Cut Through has eclipsed S&F



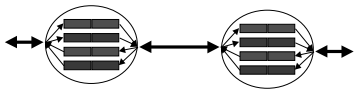
Xport Approaches -- Wormhole Switching

- Wormhole routers send entire message in a single packet; “dynamically circuit switched”
  - Eliminates overhead of set-up/tear-down
  - Fully exploits pipelining, minimizes header bits
  - Still monopolizes resources, penalizing short messages
  - Message delivered in order
- WH is the most popular transport method for interconnection networks -- simpler
- Compromise schemes
  - Large, e.g. page, variable length packets
  - Allow small messages to “play through”

Virtual Channels

A single physical network can transport data for logically separate networks

- Keep separate buffers for each net
- Virtual channels are often used to safeguard against deadlock within a single network design



Router Design

- Router design is an intensively studied topic
- Inventing a routing algoirthm is the easy part ... demonstrating that it is a low latency, high throughput, deadlock free, livelock free, starvation free, reliable, etc. is tougher
- Generally ...
  - Low latency is the most significant property
  - Throughput -- delivered bits -- is next
  - The only interesting case is “performance under load,” so the challenge is handling contention

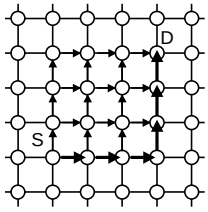
Topologies

- Many regular network topologies have been considered ... there is no best topology
- A common family of useful topologies are the k-ary d-cubes, which have k nodes in each of d dimensions
  - 2-ary d-cube is the d-dimensional binary hypercube
  - n-ary 2-cube is an nxn mesh or torus
- The routing algorithms considered will apply at least to the k-ary d-cube family

Oblivous Routing

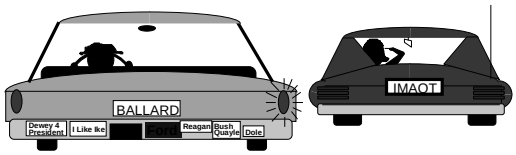
Oblivious Routers -- Use a single path between any [source,destination] pair

- Dimension order
- Simple logic, very fast
- Virtual cut through
- State-of-the-art for MIMD machines



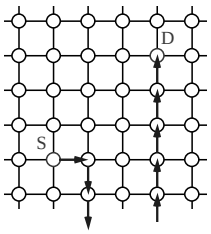
Oblivious Routers

Many drivers take a single path to a destination, oblivious to congestion and opportunities to avoid it



Randomized Oblivious Routers

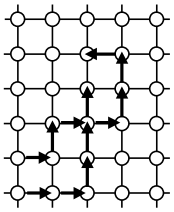
- Randomized routers attempt to neutralize network contention by randomizing the paths
  - Select a random intermediate node
  - Route obliviously to intermediate, then on to destination
  - Introduces a 2x overhead



Adaptive Routing

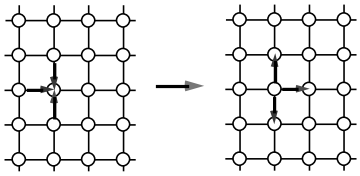
Adaptive Routers -- Take alternate paths to avoid congestion

- Two types:
  - Minimal Adaptive: Limit alternatives to shortest paths →
    Must always go forward
  - Nonminimal Adaptive: Any alternative path possible →
    Backup is allowed



Deflection Routers

- Hot potato routing tries to keep things moving
  - An adaptive synchronous approach
  - Incoming packets are matched to outgoing channels
  - Losers are assigned arbitrarily
  - All packets leave on next step

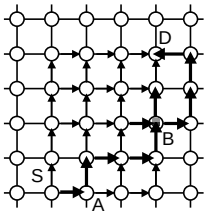


Chaos Router

Chaos router prefers any minimal path from source to destination, but will take ANY path

- Take random shortest path whenever possible (A) e.g. light traffic
- Wait briefly for moderate congestion to clear
- In heavy congestion, when no space remains for local waiting, deroute (B) a random packet

A derouting packet takes a path that moves it further from its destination



Chaos Router Properties

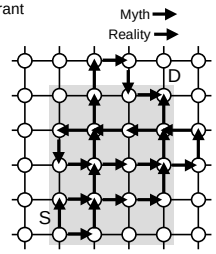
Packets take randomized *minimal* paths except in cases of extremely high congestion

Chaos routers are inherently fault tolerant

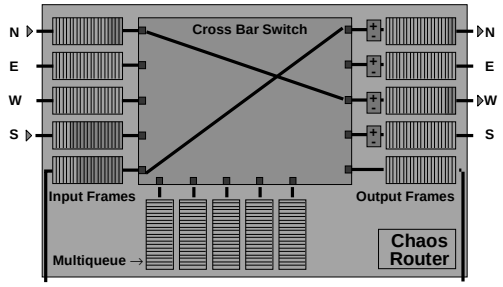
Adaptivity reduces latency and increases throughput by selecting packet paths incrementally based on local congestion

. . . packets take a productive path if it's available

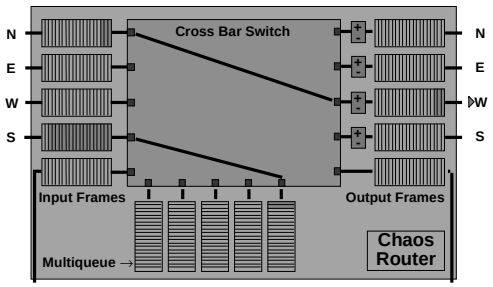
The packets of a message can be delivered out of order, and so must be reassembled at destination



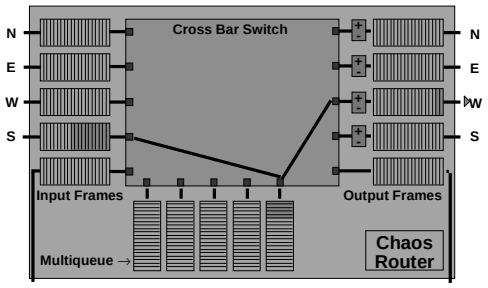
Chaos Router Operation



Moving Into Multiqueue



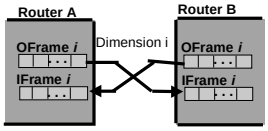
Cutting Through Multiqueue



Deadlock

Deadlock is a condition where packets are permanently blocked

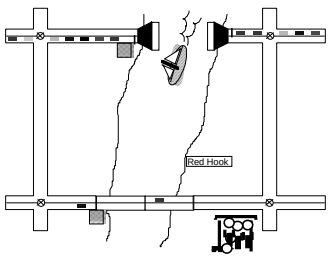
- Deadlock is avoided in the Chaos router by the packet exchange protocol -- a channel wanting to send must be willing to receive a packet



Invariant: One of the four buffers is always available

Livelock

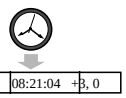
The Ballard and Fremont Bridges



Solving Livelock By Priorities

Livelock is the condition where packets continually circulate, but are not delivered to their destinations ... standard solution

- Timestamp each packet
- When packets compete for channel, pick oldest
- Eventually, packets are delivered or become oldest



3.1415962	08:21:04	+3, 0

3.1415962	08:21:04	+3,0
2.3761150	08:20:04	+5,0
42.321156	08:20:24	+1,0

Must prioritize

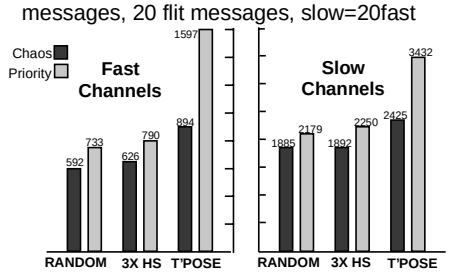
Solving Livelock By Randomizing

Livelock prevention hampers high performance, but it is very rare ... "stir things up" and gamble

- By randomly selecting the message for derouting, the Chaos router is probabilistically livelock free
- *Probabilistic livelock freedom*-- the probability a message remains in network for *t* seconds goes to 0 as *t* increases; probabilistic ≈ determinstic, in practice

Chaos vs Priorities

Simulation: 256 node hypercube, 150,000 messages, 20 flit messages, slow=20fast



An Implementation

Design by Kevin Bolding --

- Degree 4, suitable for mesh, torus, ...
- 20 phit packets, 16-bit phits, 5 frame multiqueue
- Linear feedback shift register pseudo randomizer
- Bi-directional channels alternating at packet boundaries, separated-injection delivery channels
- Node latency, 4 ticks at 15ns clock
- Technology: 1.2µ CMOS, scalable design rules
- Comparable to the Elko Router, an oblivious router designed at Caltech in the same technology

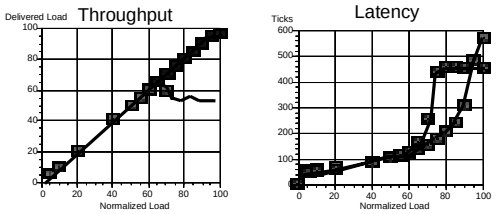
Performance Assessment

Evaluation by Melanie Fulgham

- Chaos and Elko networks simulated at flit level
- "Batched means" method for computing 95% confidence intervals
- Expected throughput -- proportion of the network bisection bandwidth utilized
- Expected latency -- a packet's injection-to-delivery time, exclusive of source queueing
- Learmonth-Lewis prime-modulus, multiplicative congruential pseudo-random number generator

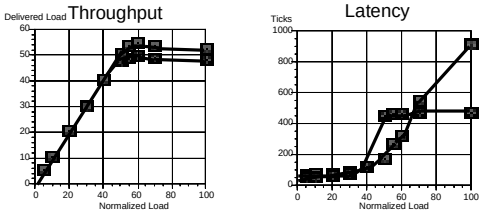
Random: all destinations equally likely, including self
Permutations: transpose, bit-reversal, complement, perfect shuffle
Hot spots: 10 positions 4x more likely to be a destination

Throughput and Latency



16x16 2-D Torus, Random Traffic

Throughput and Latency

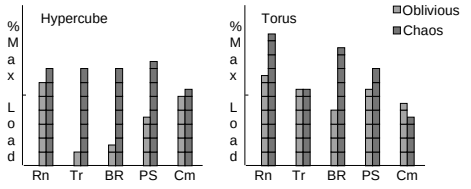


16x16 2-D Torus, Transpose Traffic

Saturation

Oblivious & Chaotic routers on representative nonuniform loads -- 256 node topologies, continuous injection

Saturation point normalized to bisection bandwidth

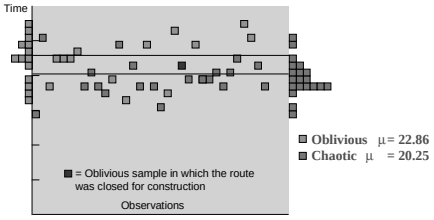


Experimental Commuting

Methodology

Adopt fixed shortest path oblivious routes between home & UW

When the clock parity was odd, I used an oblivious algorithm; otherwise, I used a Chaotic algorithm



Input/Output Driven Router Design

What initiates a routing decision?

Packet arrival -- input driven

Availability of output channel -- output driven

Chaos Router was the first to use an output driven protocol

When a packet arrives, find a productive output channel.

When an output channel becomes free, find a packet that can use it. Randomize if more than one.

Many routing algorithms can be implemented using either input or output driven protocols, but **output driven is better**

Benefits of Output Driven

Comparisons on 256-node torus, mesh networks for different routers

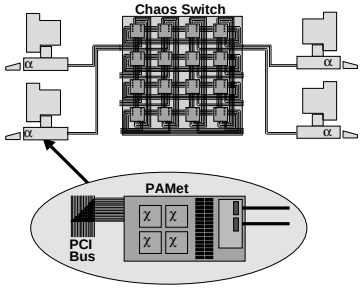
Determine saturation level -- the point at which the network can no longer keep up with arriving traffic using 5% granularities

Advantage of output driven over input driven saturation levels (5%)

Router		Rn	BR	Cm	PS	Tr	HS1	HS2
Torus	Oblivious							
	*-Channels							
	Min-Triplex							
Mesh	Oblivious(nvc)							
	Oblivious							
	*-Channels							
	Min-Triplex							

30% 25% 20% 15% 10% 5% 0% -5% -10%

Applying ICN Technology To LANs,SANs



Conclusions

Chaos router is a randomizing, nonminimal adaptive packet router:

Deterministically deadlock free, probabilistically livelock free

Simulation studies indicate excellent performance

Chip design demonstrates practicality

Chaos is a friend of mine. **-- Bob Dylan**

More Reading

- W. Dally & C. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Transactions on Computers* C-36:547-553, 1987
- P. Kermani, L. Kleinrock, "Virtual cut-through: A new . . . technique," *Computer Networks* 3:267-286, 1979
- S. Konstantinidou, *Deterministic & Chaotic Adaptive Routing in Multicomputers*, PhD Dissertation, University of Washington, 1991
- K.W. Bolding, *Chaotic Routing -- Design and Implementation*, PhD Dissertation, University of Washington, 1993
- J. Ngai & C. Seitz, "A framework for adaptive routing in multicomputer networks," ACM Symposium on Parallell Algorithms and Architectures, pp. 1-9, 1989
- S. Konstantinidou, L. Snyder, "Chaos Router . . .," ACM Symposium on Parallell Algorithms and Architectures , pp. 21-30, 1990

More Reading

- C. Seitz & W. Su, "A family of routing ... chips based on Mosaic," Symp Int. Sys., Springer Verlag, pp. 320-337, 1993
- K. Bolding, M. Fulgham & L Snyder "A case for Chaos adaptive routing," *IEEE Transactions on Computers* 46(12):1281-1291, 1997
- M. Fulgham & L. Snyder, "A Comparison of Input and Output Dirven Routers," Lecture Notes In Computer Science 1123, Springer-Verlag pp. 195-204, 1996
- Melanie L. Fulgham, *Multicomputer Routing Techniques*, PhD Dissertation, University of Washington, 1997
- B. Smith, "Architecture & applications of HEP multiprocessor computer system," Proc. SPIE, pp. 241-248, 1981
- L. Valiant, G. Brebner, "Universal schemes for parallel communication," Proc. 13th ACM Symposium On Theory of Computation, pp.263-277, 1981