Lecture 21: Coherence and Interconnection Networks

Papers:

- Flexible Snooping: Adaptive Filtering and Forwarding in Embedded Ring Multiprocessors, UIUC, ISCA-06
- Coherence-Ordering for Ring-Based Chip Multiprocessors, Wisconsin, MICRO-06 (very brief)
- In-Network Cache Coherence, Princeton, MICRO-06

Motivation

- CMPs are the standard
- cheaper to build medium size machines
  - 32 to 128 cores
- shared memory, cache coherent
  - easier to program, easier to manage
- cache coherence is a necessary evil

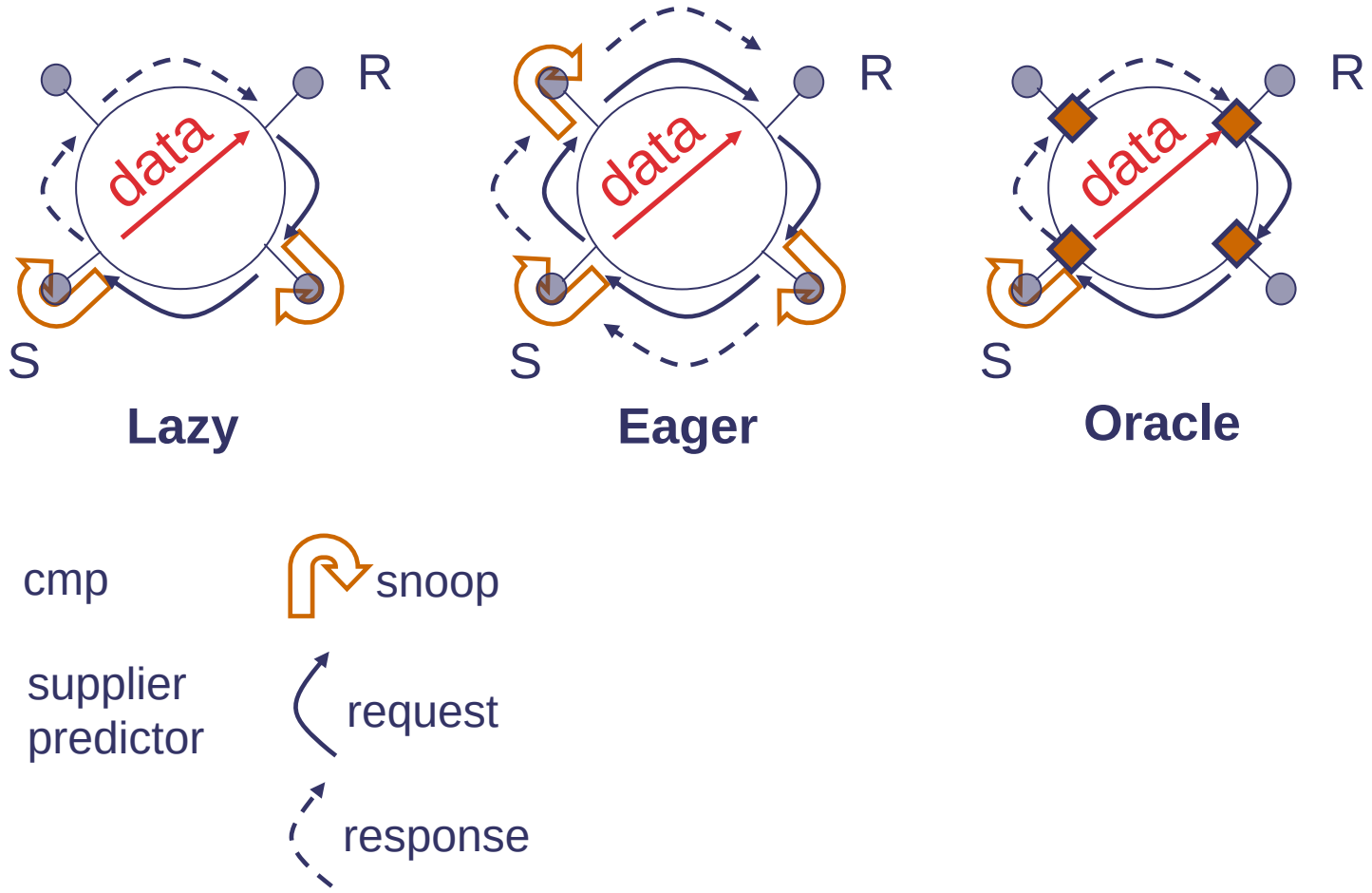
Cache coherence solutions

strategy	ordered network?	pros	cons
snoopy broadcast bus	yes	simple	difficult to scale
directory based protocol	no	scalable	indirection, extra hardware
snoopy embedded ring	no	simple	long latencies
In-Network Directory	no	scalable	Complexity

Ring Interconnect based snoop protocols

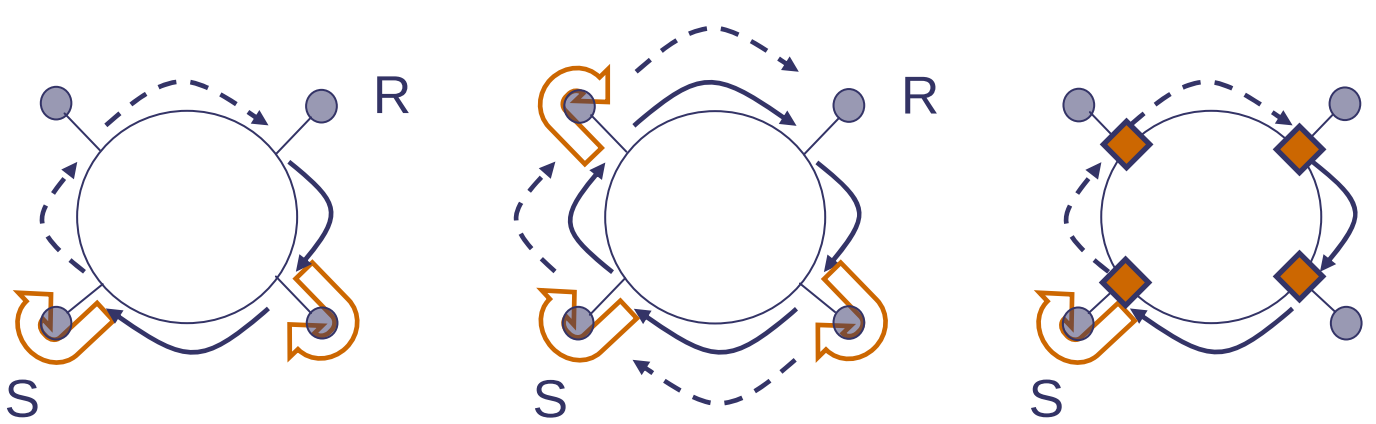
- Investigated by Barroso *et al.* in early nineties
- Why?
  - Short, fast point-to-point link
  - Fewer (data) ports
  - Less complex than packet-switched
  - Simple, distributed arbitration
  - Exploitable ordering for coherence

Ring in action



Courtesy: UIUC-ISCA06

Ring in action



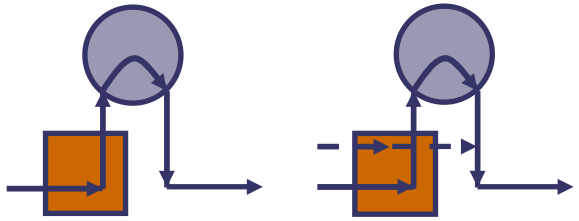
	Lazy	Eager	Oracle
latency	↑	↓	↓
snoops	↓	↑	↓↓
messages	↓	↑	↓

- **goal:** adaptive schemes that approximate Oracle's behavior

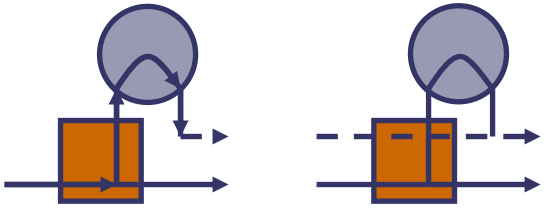
Courtesy: UIUC-ISCA06

Primitive snooping actions

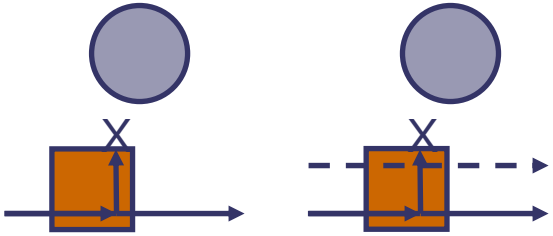
- snoop and then forward
  - \+ fewer messages



- forward and then snoop
  - \+ shorter latency



- forward only
  - \+ fewer snoops
  - \+ shorter latency
  - – false negative predictions **not** allowed



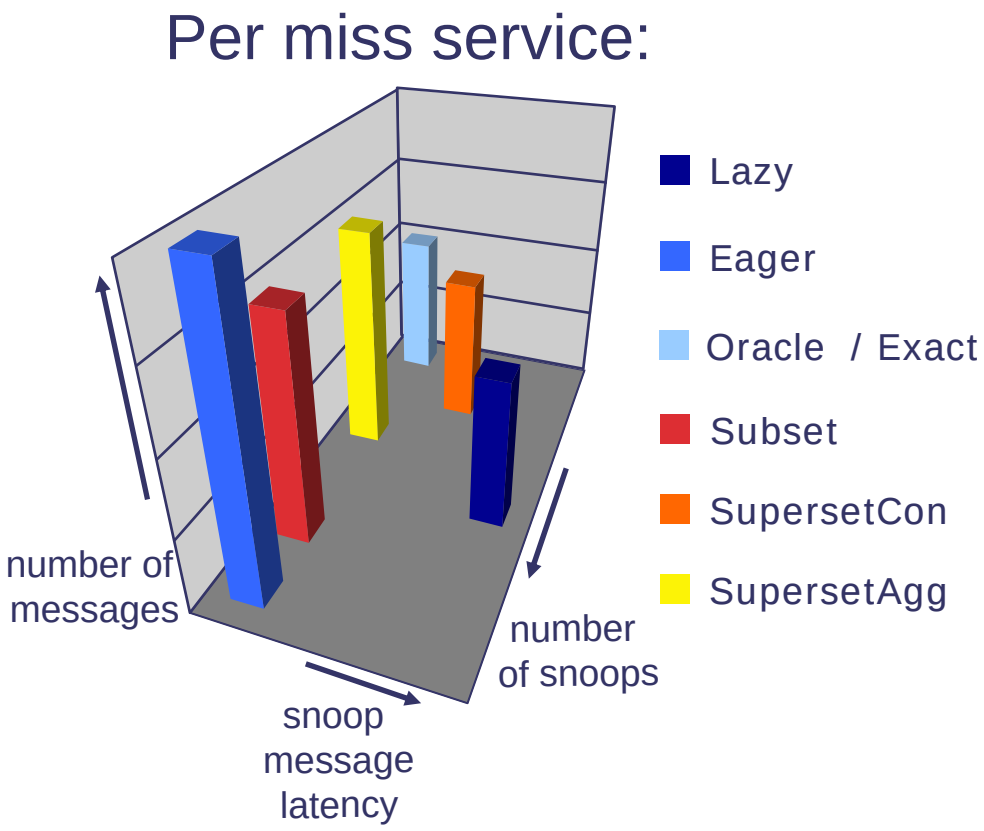
Courtesy: UIUC-ISCA06

Predictor implementation

- Subset
  - associative table:
    subset of addresses that can be supplied by node
- Superset
  - bloom filter: superset of addresses that can be supplied by node
  - associative table (exclude cache):
    addresses that recently suffered false positives
- Exact
  - associative table: all addresses that can be supplied by node
  - downgrading: if address has to be evicted from predictor table, corresponding line in node has to be downgraded

Algorithms

algorithm		negative	positive
Subset		forward then snoop	snoop
Superset	**Con**	forward	snoop then forward
	Agg		forward then snoop
Exact		forward	snoop

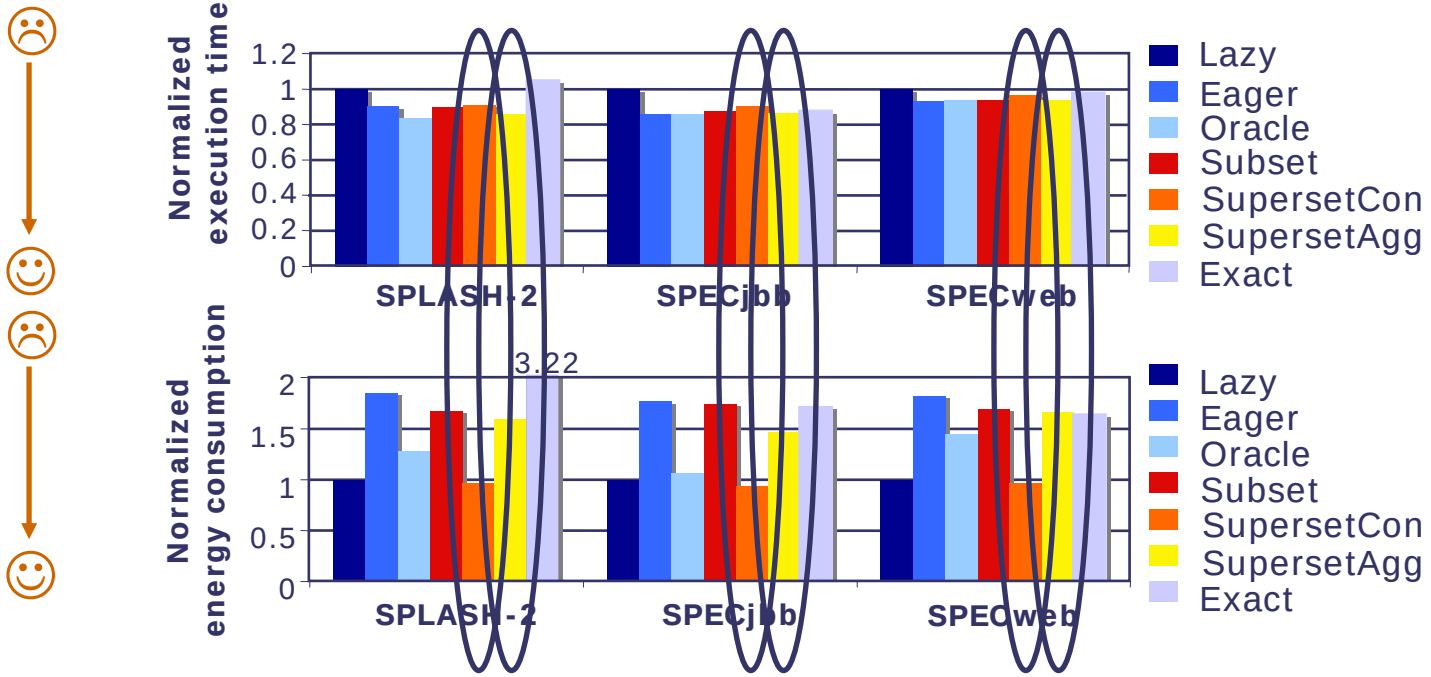


Courtesy: UIUC-ISCA06

Experiments

- 8 CMPs, 4 ooo cores each = 32 cores
  - private L2 caches
- on-chip bus interconnect
- off-chip 2D torus interconnect with embedded unidirectional ring
- per node predictors: latency of 3 processor cycles
- sesc simulator (sesc.sourceforge.net)
- SPLASH-2, SPECjbb, SPECweb

Most cost-effective algorithms



- high performance: Superset Aggressive
  - faster than Eager at lower energy consumption
- energy conscious: Superset Conservative
  - slightly slower than Eager at much lower energy consumption

Courtesy: UIUC-ISCA06

Issues

- Implementation of Ring Interconnects
  - Already available in commercial processors
    - IBM Power series & Cell
    - Next generation Intel processors
  - Limited Scalability (medium range systems)
- Power Dissipation
  - Overhead due to design of routers (trivial though)
- Conflicts & Races
  - Ring is not totally ordered
    - may lead to retries (unbounded?)

Ordering Points – avoiding retries

- Employ a node as an ordering point statically
  - Creates total ordering of requests!
  - Performance hit due to centralization

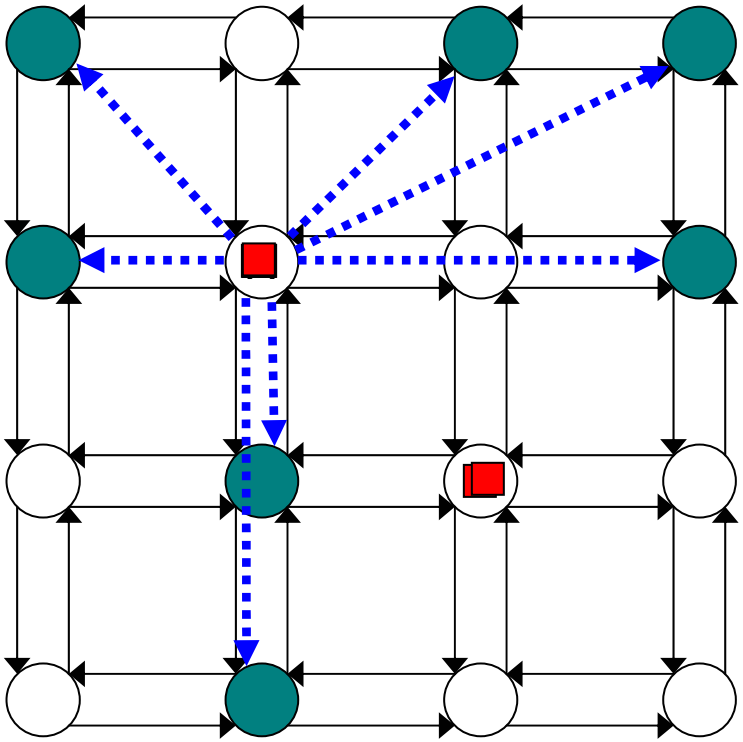
- Ring-Order (Wisconsin, MICRO 2006)
  - Combines ordering and eager forwarding
    - Ordering for stability
    - Eager forwarding for performance
- Inspired from token coherence
  - Use of a priority token
    - breaks conflicts
    - provides ordering

In-Network Cache Coherence

- Traditional Directory based Coherence
  - Decouple the protocol and the Communication medium
    - Protocol optimizations for end-to-end communication
    - Network optimizations for reduced latency
- Known problems due to technology scaling
  - Wire delays
  - Storage overhead of directories (80 core CMPs)
- Expose the interconnection medium to the protocol
  - Directory = Centralized serializing point
    - Distribute the directory with-in the network

Current: Directory-Based Protocol

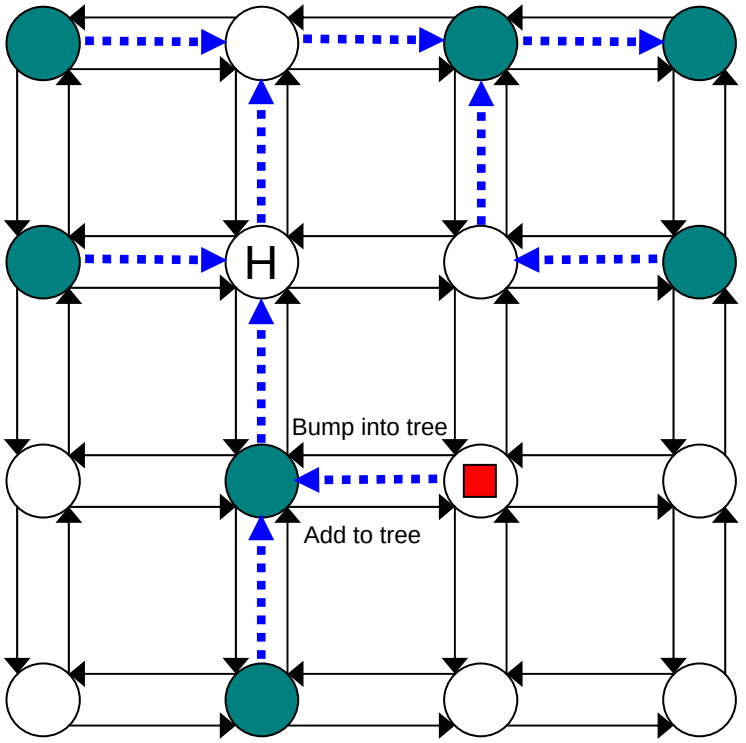
- Home node (H) keeps a list (or partial list) of sharers
- Read: Round trip overhead
  - Reads don't take advantage of locality
- Write: Up to two round trip overhead
  - Invalidations inefficient



Courtesy:Princeton-MICRO06

New Approach: Virtual Network

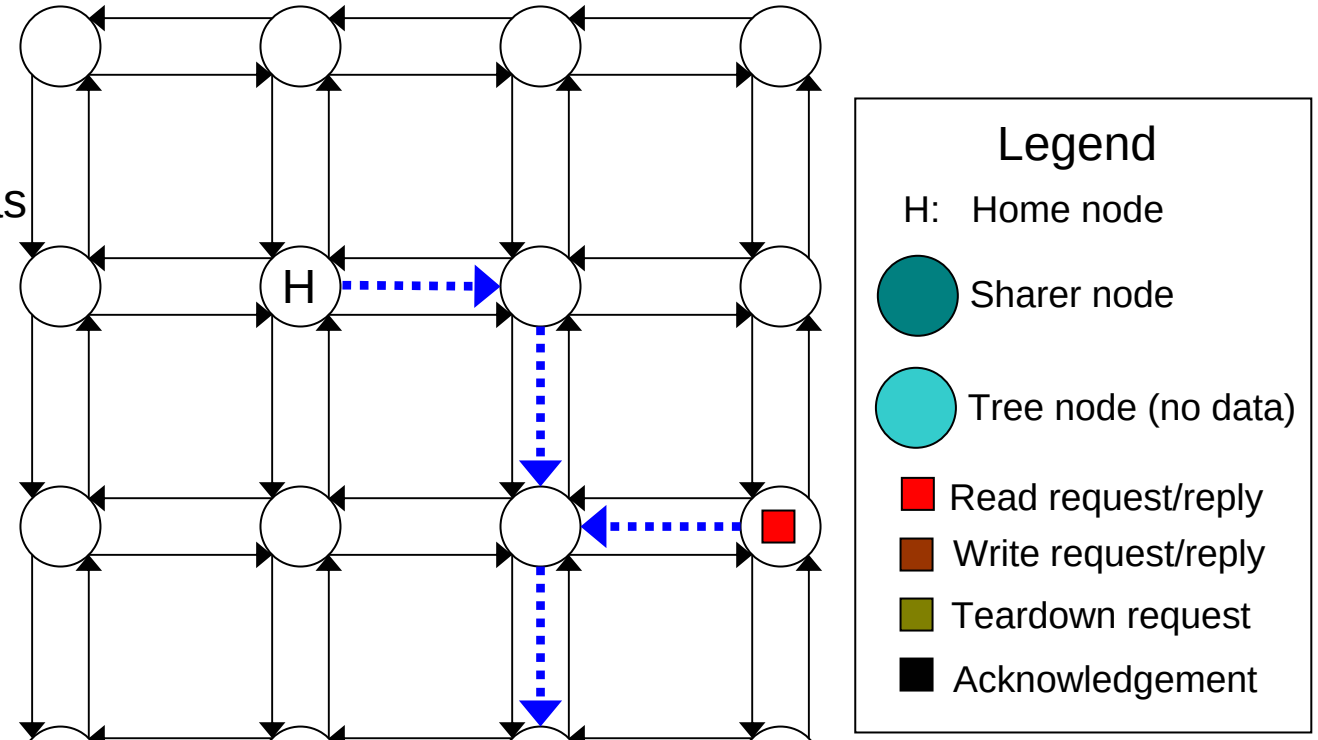
- Network is not just for getting from A to B
- It can keep track of sharers and maintain coherence
- How? With trees stored within routers
  - Tree connecting home node with sharers
  - Tree stored within routers
  - On the way to home node, when requests “bump” into trees, routers will re-route them towards sharers.



Courtesy:Princeton-MICRO06

Reads Locate Data Efficiently

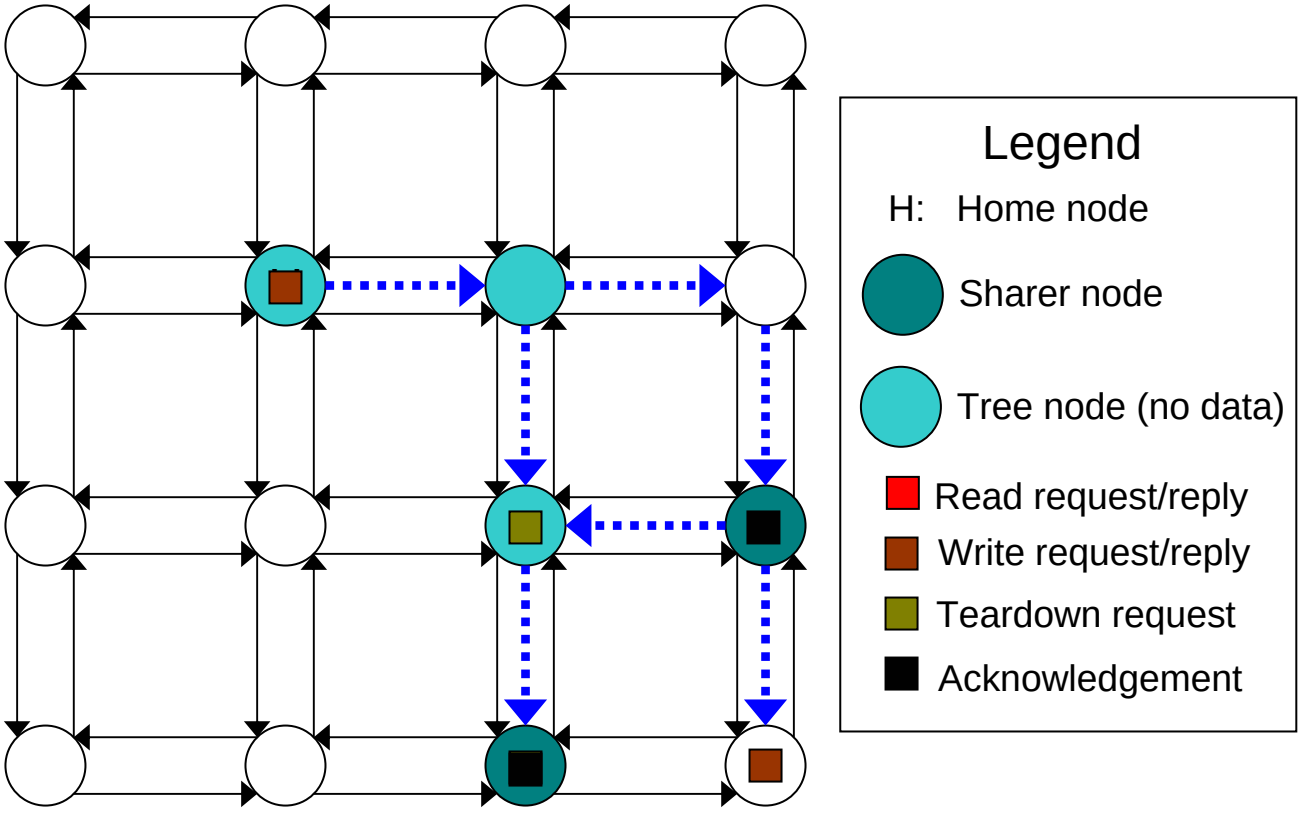
- Read Request injected into the network
- Tree constructed as read reply is sent back
- New read injected into the network
- Request is redirected by the network
- Data is obtained from sharer and returned



Courtesy:Princeton-MICRO06

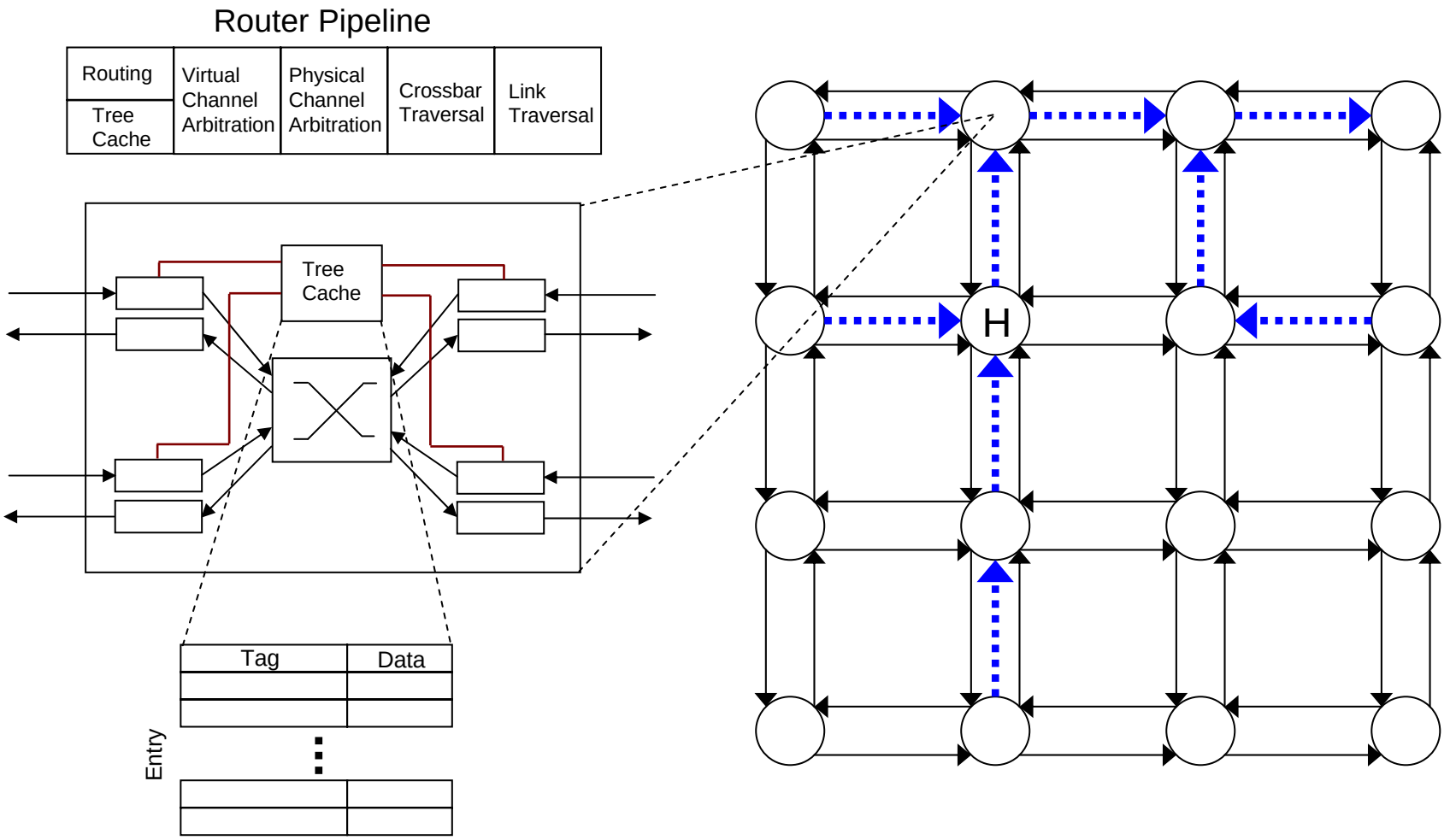
Writes Invalidate Data Efficiently

- Write Request injected into the network
- In-transit invalidations
- Acknowledgements spawned at leaf nodes
- Wait for acknowledgements
- Send out write reply



Courtesy:Princeton-MICRO06

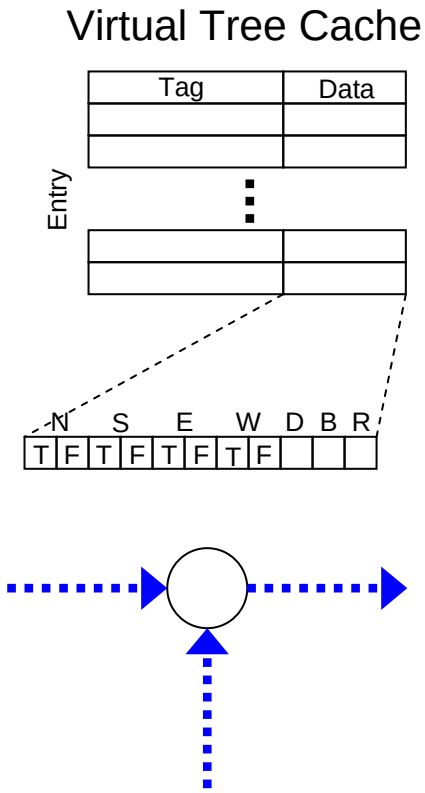
Router Micro-architecture



Courtesy:Princeton-MICRO06

Router Micro-architecture: Virtual Tree Cache

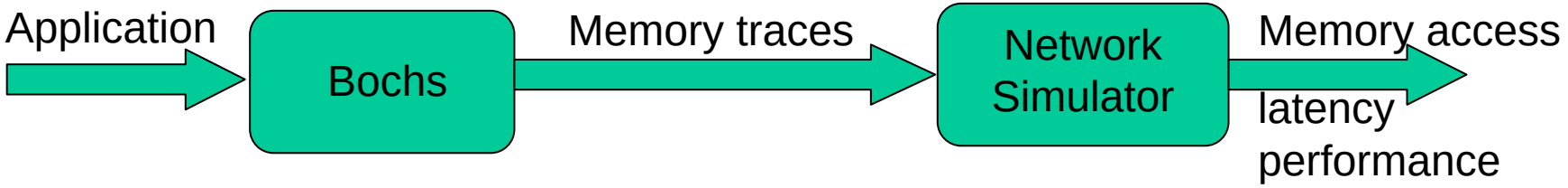
- Each cache line in the virtual tree cache contains
  - 2 bits (To, From root node) for each direction (NSEW)
  - 1 bit if the local node contains valid data
  - 1 busy bit
  - 1 Outstanding request bit



Courtesy:Princeton-MICRO06

Methodology

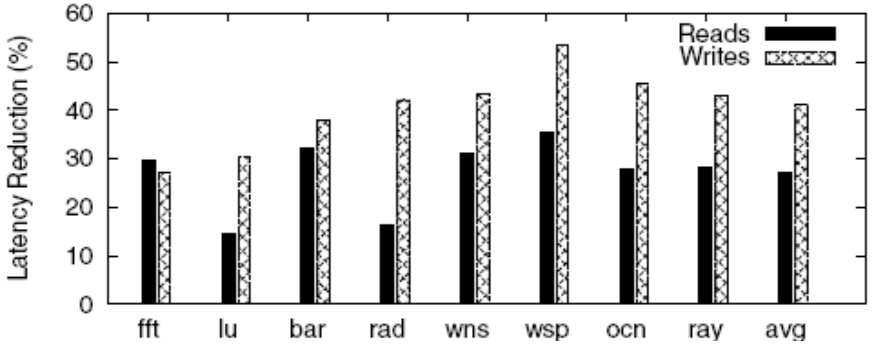
- Benchmarks: SPLASH-2
- Multithreaded simulator: Bochs running 16 or 64-way
- Network and virtual tree simulator
  - Trace-driven
    - Full-system simulation becomes prohibitive with many cores
    - Motivation: explore scalability
  - Cycle-accurate
  - Each message is modeled
  - Models network contention (arbitration)
  - Calculates average request completion times



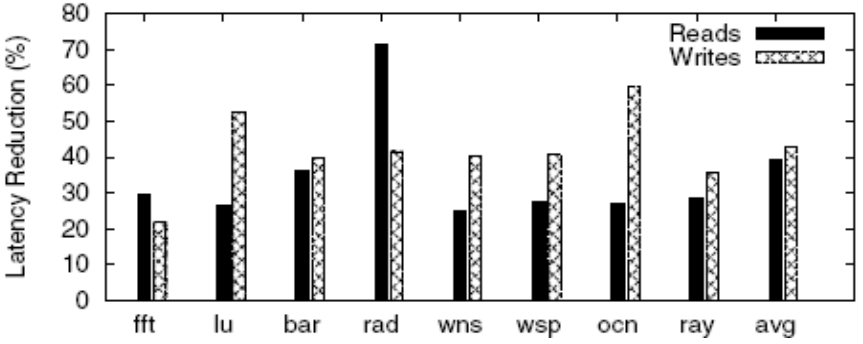
Performance Scalability

- Compare in-network virtual tree protocol to standard directory protocol
- Good improvement
  - 4x4 Mesh: Avg. 35.5% read latency reduction, 41.2% write latency reduction
- Scalable
  - 8x8 Mesh: Avg. 35% read and 48% write latency reduction

4x4 Mesh

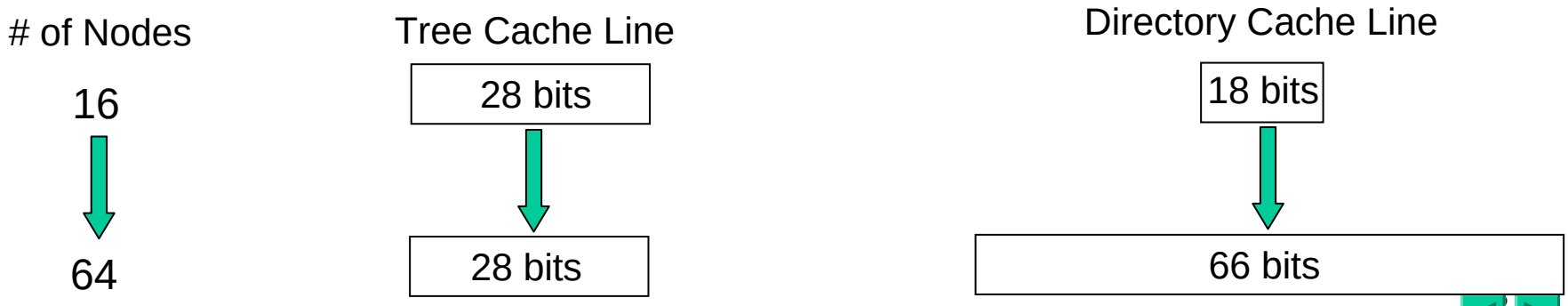


8x8 Mesh



Storage Scalability

- Directory protocol: O(# Nodes)
- Virtual tree protocol: O(# Ports) ~O(1)
- Storage overhead
  - 4x4 mesh: 56% more storage bits
  - 8x8 mesh: 58% *fewer* storage bits



Courtesy:Princeton-MICRO06

Issues

- Power Dissipation
  - Major Bottleneck
    - complex functionality in the routers
- Complexity
  - Complexity of routers, arbiters, etc.
  - Lack of verification tools
- CAD tool support
  - More research needed for ease of adoption

Summary

- More and more cores on a CMP
  - Natural trend towards distribution of state
  - More interaction between interconnects and protocols
- Ring interconnects => medium size systems
  - Simple design, low cost
  - Limited Scalability
- In-Network coherence => large scale systems
  - Scalability
  - Severe implementation and power constraints on chip

Thank you & Questions