



# In-network Cache Coherence [Micro 2006]

Noel Easley<sup>a</sup>, Li-Shiuan Peh<sup>a</sup>, Li Shang<sup>b</sup>

Departments of Electrical Engineering<sup>a</sup>  
Princeton University

Departments of Electrical & Computer Engineering<sup>b</sup>  
Queen's University



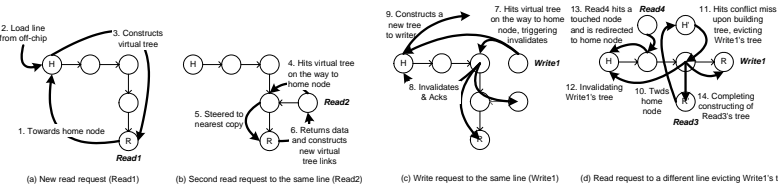
## Abstract

Efficient, scalable cache coherence is of great importance to high-performance CMPs. However, the widely-used directory-based protocols face well-known problems in both delay and scalability. To address this issue, we proposed in-network cache coherence which demonstrates excellent, scalable performance. Different from existing work which treats the interconnect solely as a communication media, our solution tailors the protocol design and coherence management within the on-chip network, which opens up in-transit optimization opportunities. Our protocol demonstrates good, scalable performance, with 27.2% and 41.2% decreases in read and write latency on average for a 4-by-4 network, and 39.5% and 42.8% improvements for reads and writes respectively for an 8-by-8 network for a range of SPLASH-2 benchmarks. We see this work as the first step of leveraging the network's inherent scalability to realize highly-scalable CMP architectures.

## Introduction

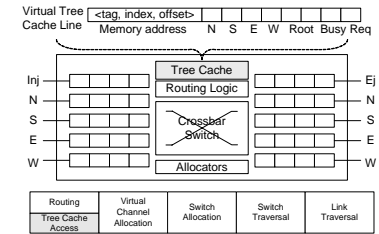
With the trend towards increasing number of processor cores in future chip architectures, scalable directory-based protocols for maintaining cache coherence will be needed. However, directory-based protocols face well-known problems in delay and scalability. Most current protocol optimizations targeting these problems maintain a firm abstraction of the interconnection network fabric as a communication medium — protocol optimizations consist of end-to-end messages between requestor, directory and sharer nodes; while network optimizations separately target lowering communication latency for coherence messages. In this work, we propose an implementation of cache coherence protocols within the network, embedding directories within each router node that manage and steer requests towards nearby data copies, enabling in-transit optimization of memory access delay. Simulation results across a range of SPLASH-2 benchmarks demonstrate significant performance improvement and good system scalability, with up to 44.5% and 56% savings in average memory access latency for 16 and 64-node systems, respectively, when compared against the baseline directory cache coherence protocol. Detailed microarchitecture and implementation characterization affirms the low area and delay impact of in-network coherence.

## In-network cache coherence protocols

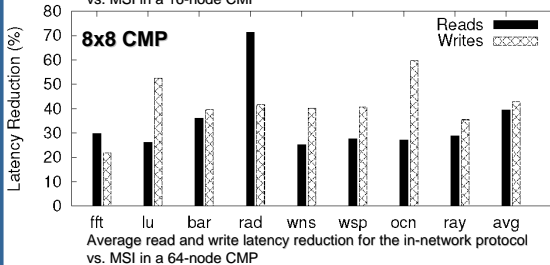
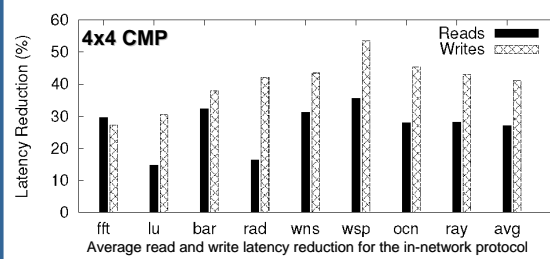


The central thesis of in-network cache coherence is the moving of coherence directories from the home directory nodes into the network fabric. Virtual trees are maintained within the network in place of coherence directories to keep track of sharers, one for each cache line. The virtual tree consists of one root node (R), all nodes that are currently sharing this line, as well as the intermediate nodes between the root and the sharers thus maintaining the connectivity of the tree. The nodes of the tree are connected by virtual links with each link between two nodes always pointing towards the root node. These virtual trees are stored in virtual tree caches at each router within the network. As reads and writes are routed towards the home node, if they encounter a virtual tree in-transit, the virtual tree takes over as the routing function and steers read requests and write invalidates appropriately towards the sharers instead.

## Router microarchitecture



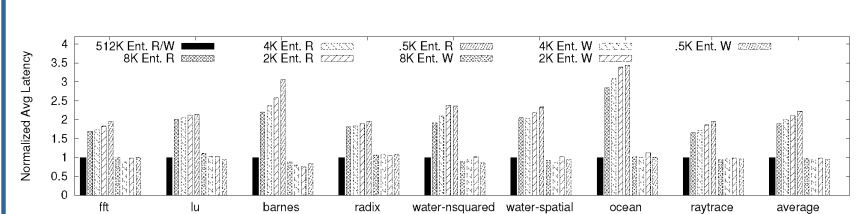
Each on-chip interconnect router is equipped with a virtual tree cache. Each entry of the virtual tree cache forms one-hop connection of the virtual network. Tree cache access is the first router pipeline stage. It determines the destination of the shared data.



## Performance and scalability

In-network protocol vs. standard directory-based MESI directory protocol

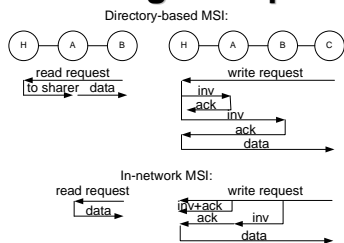
- 4x4 Mesh: Avg. 35.5% read latency reduction  
41.2% write latency reduction
- 8x8 Mesh: Avg. 35.0% read latency reduction  
48.0% write latency reduction



## Area efficiency

**High area efficiency:** In general, increasing the tree cache size results in steadily reduction of average read and write latencies. However, a 4K-entry tree cache design can effectively reduce the read and write access latencies with small area overhead.  
**Scalable area efficiency:** For a 16-node system, our in-network introduces 56% more storage overhead compared to the directory based protocol. When the system scales to 64 nodes, our in-network protocol improves the storage efficiency by 58% compared to the directory based protocol.

## Motivating example



## Conclusion

We propose the embedding of cache coherence protocols within the network, separately from the data they manage, so as to leverage the inherent performance and storage scalability of on-chip networks. While there has been abundant prior work on network optimizations for cache coherence protocols, to date, most prior protocols have maintained a strict abstraction of the network as a communication fabric. Here we detailed how the directories of classic directory-based protocols can be moved into the network, maintained in the form of virtual trees that steer read and write requests in-transit, towards nearby copies. Our evaluations on a range of SPLASH-2 benchmarks demonstrate up to 44.5% savings in average memory latency on a 16-processor system. Furthermore, the performance scalability is demonstrated by an average memory access savings of up to 56% savings on a 64-processor system. Ultimately, we envision the embedding of more distributed coordination functions within the on-chip network, leveraging the network's inherent scalability to realize high-performance highly-concurrent chips of the future.