

# Towards an ideal interconnection fabric for many-core chips

Li-Shiuan Peh  
Assistant Professor  
Department of Electrical Engineering  
Princeton University

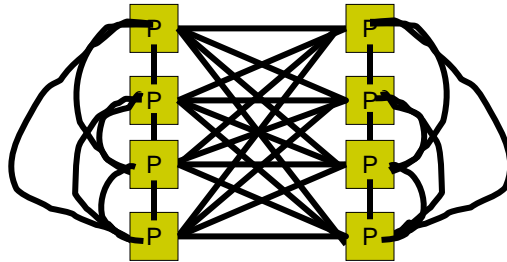


## The interconnection problem in many-core chips

- 1 -> Multi-core -> Many-core
  - How do they communicate?
- Holy grail: Speed, bandwidth, energy of dedicated wires



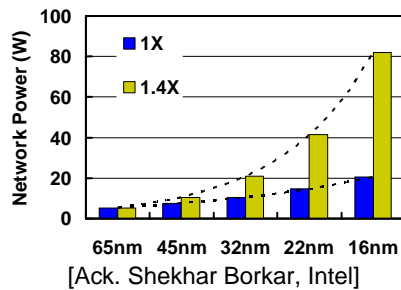
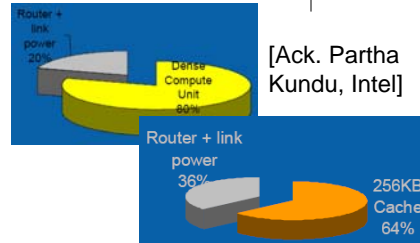
- Why this is unrealizable:



# Scalable solution: On-chip networks, BUT..



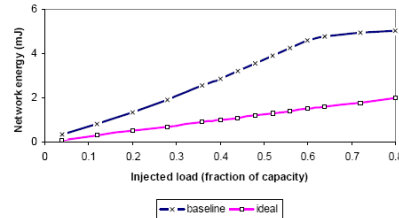
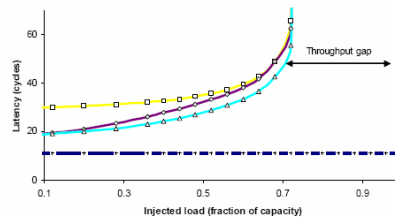
- Solution: *Multiplex communication on wires*
  - BUT: Cost – complex routers
  - Need to go through a router pipeline every hop
    - Wire energy-delay + Router energy-delay
    - Throughput degradation due to additional resource contention (buffers, switch etc.)
- State-of-the-art network today:
  - Power way over budget
  - Will be worse in the future
  - “Router power prohibitive” Shekhar Borkar, Intel



# Existing gap between state-of-the-art and ideal interconnect



- State-of-the-art:
  - Virtual channels [Dally, ISCA'90]
  - LA: Lookahead routing [SGI]
  - BY: Bypassing [Alpha]
  - SP: Speculation [Peh&Dally, HPCA'01]
  - Power-driven router microarchitectures [Wang,Peh,Malik, MICRO'03]

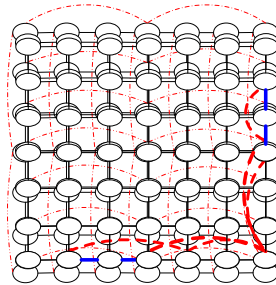
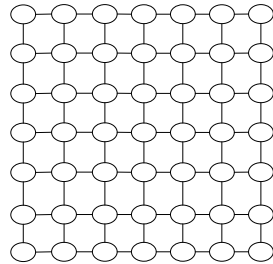


Latency gap: 2X; Throughput gap: 70%;  
Energy gap: 2.5X of ideal

# Express virtual channels (EVCs)



- Connect distant nodes by “*virtual dedicated links*”
- Skip router pipelines at intermediate nodes



Path from node 01 to 56

Dynamic EVCs:

- All nodes along an EVC are classified as:
  - EVCs of varying lengths (of every dimension)
    - Bypass (e.g. 01, 02 etc. for x-dimension)

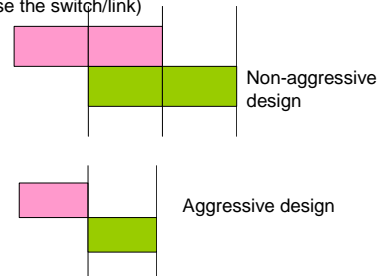
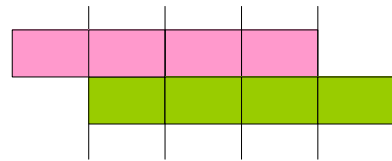
[Kumar, Peh, Kundu, Jha, ISCA'07]

# Reducing router overhead

When bypassing a node *virtually* using an EVC –



- no VC allocation (keep traveling on the same EVC it already holds)
- no switch allocation (EVC flits given higher priority to use the switch/link)
- no buffering
- **Latency impact:**



- **Energy impact:**

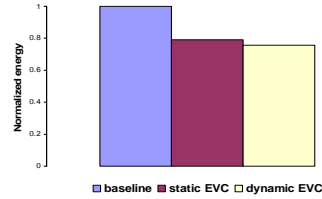
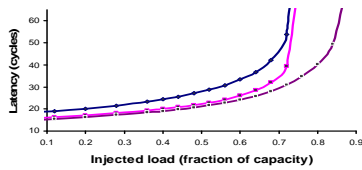
$$E_{router} = E_{buffer\_write} + E_{buffer\_read} + E_{vc\_arb} + E_{sw\_arb} + E_{xbar}$$

- **Throughput impact:**
  - No need to allocate resources at every node
  - Lower contention → higher throughput

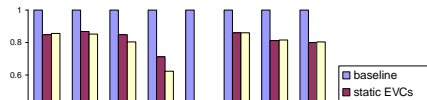
[Kumar, Peh, Kundu, Jha, ISCA'07]

60  
60  
50  
30  
40  
30  
20  
10  
10  
00  
00

## Evaluation results



- Uniform random traffic (7x7 mesh with 2-hop EVCs)
  - 44% latency reduction before saturation
  - 24.5% energy reduction
- Excellent scalability with network size (52% latency reduction and 38% energy reduction before saturation for a 10x10 mesh with 3-hop EVCs)



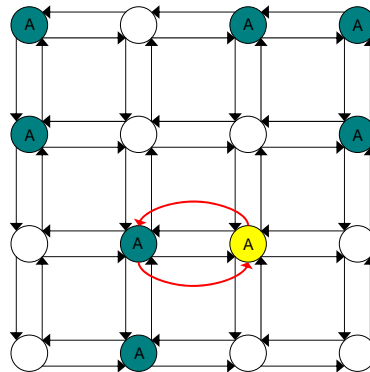
- 20 - 84% latency reduction for SPLASH-2 benchmarks

Approaching ideal interconnect:  
Latency gap: 1.2X; Throughput gap: 88%; Energy gap: 1.5X;

[Kumar, Per, Kumar, Jna, ISCA 07]

## Ideal interconnection fabric: Beyond dedicated wires

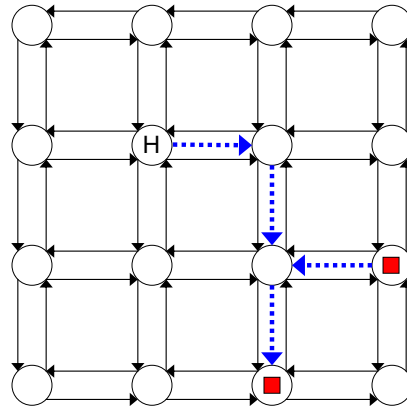
- Ideal interconnection fabric: Minimal communications
  - Go straight to nearest data
  - Invalidate all copies simultaneously
- **Network-driven computing:** A network should not just handle communications, it should shape it.
- Embed chip-wide coordination functions within the network



# In-network cache coherence: 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



**Legend**

- H: Home node
- Sharer node
- Tree node (no data)
- Read request/reply
- Write request/reply
- Teardown request
- Acknowledgement

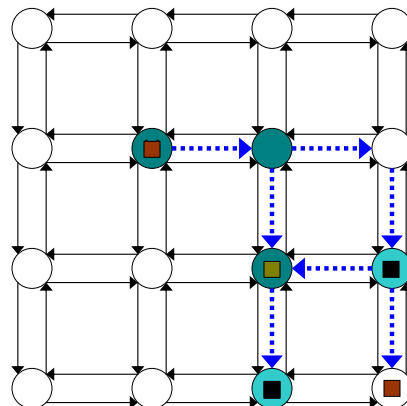


[Eisley, Peh, Shang, MICRO'06]

# 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



**Legend**

- H: Home node
- Sharer node
- Tree node (no data)
- Read request/reply
- Write request/reply
- Teardown request
- Acknowledgement

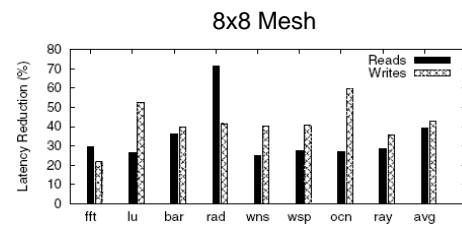
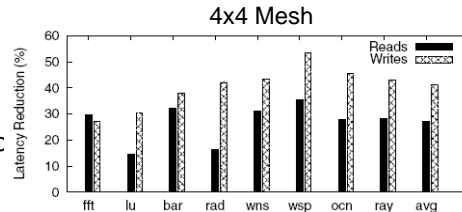


[Eisley, Peh, Shang, MICRO'06]

# 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

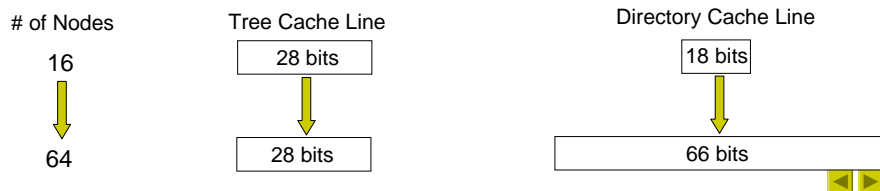


[Eisley, Peh, Shang, MICRO'06]

# Storage Scalability



- Directory protocol:  $O(\# \text{ Nodes})$
- Virtual tree protocol:  $O(\# \text{ Ports}) \sim O(1)$
- Storage overhead
  - 4x4 mesh: 56% more storage bits
  - 8x8 mesh: 58% fewer storage bits



[Eisley, Peh, Shang, MICRO'06]

## Conclusions and next steps..



- Interconnection fabric is major stumbling block in the realization of future many-core chips
  - Needs low-power, high-bandwidth, fast interconnection fabric that approaches dedicated wires
- Current efforts:
  - Low-power on-chip network design
  - Network-driven computing
    - In-network data management: caching, prefetching, replication
    - In-network code launching, migration, scheduling