



Winning with Pinning in NoC

Ahmed Abousamra, Rami Melhem,

Alex Jones

University of Pittsburgh

Outline



- Problem Description
- Prior Work
 - Reducing hop count
 - Reducing latency through router
- Our Suggested Solution
- Simulation Results
- Conclusion

Problem Description



- We consider multi-threaded applications running on CMPs
- Threads share data, thus cache coherence traffic travel on the NoC
- Performance is affected by the communication latency on the NoC
- Our work aims to improve this latency

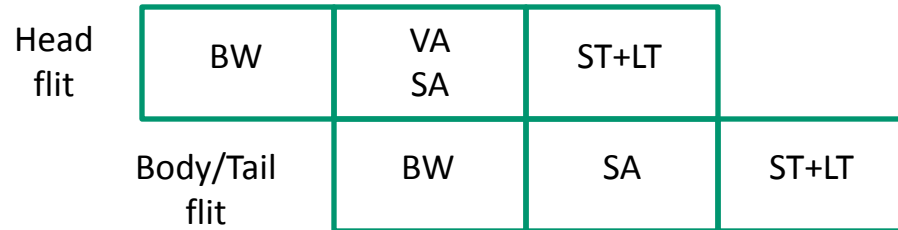


Prior Work - Reduce communication latency by reducing global hop count

- Examples:
 - Flattened butterfly topology: high radix routers to enrich connectivity (*MICRO 2007*)
 - Concentrated mesh: share each router among multiple nodes (*ICS 2006*)
 - Hierarchical NoC such as Hybrid Ring/Mesh interconnect (*NOCS 2007*)
 - A low-radix low-diameter 3D interconnect for 3D stacked chips connects every pair of communication points in at most 3 hops (*HPCA 2009*)

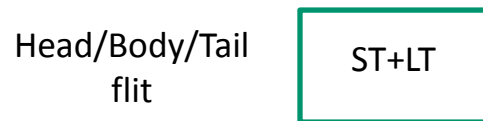


Prior Work – Reduce communication latency through router pipelining



Router Pipeline for Packet Switched Traffic

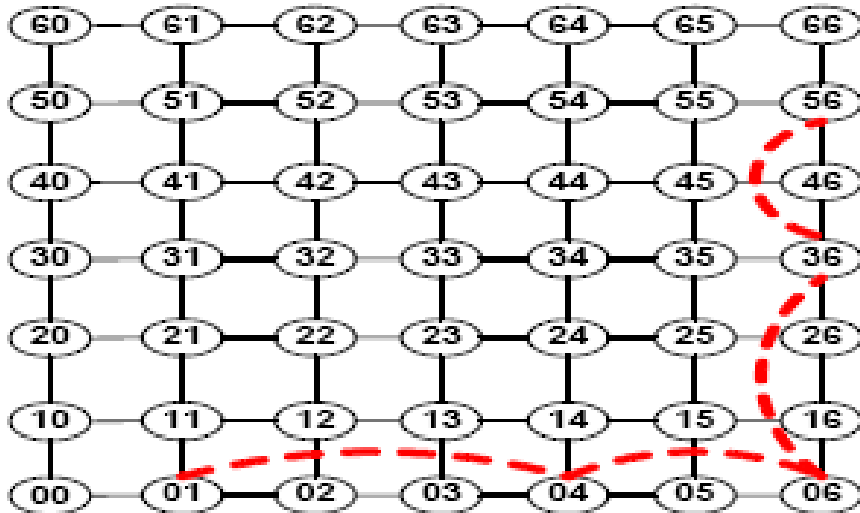
Allowing some of the traffic to bypass the router pipeline



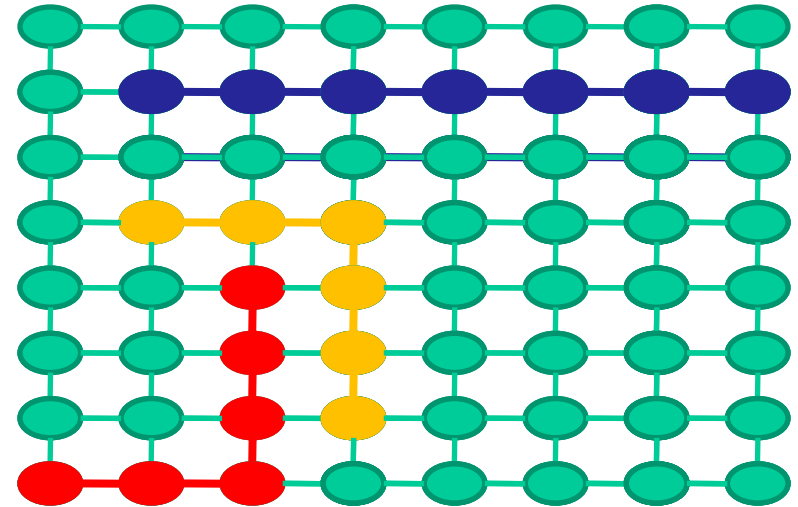
Router Pipeline for Circuit Switched Traffic



Prior Work – Reduce communication latency through router pipeline bypassing



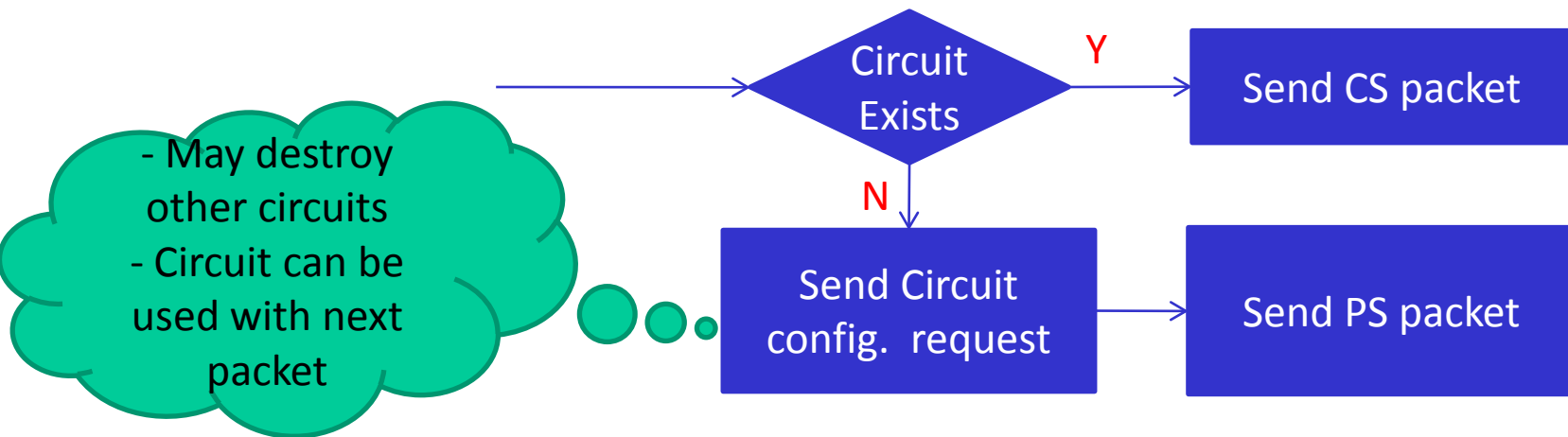
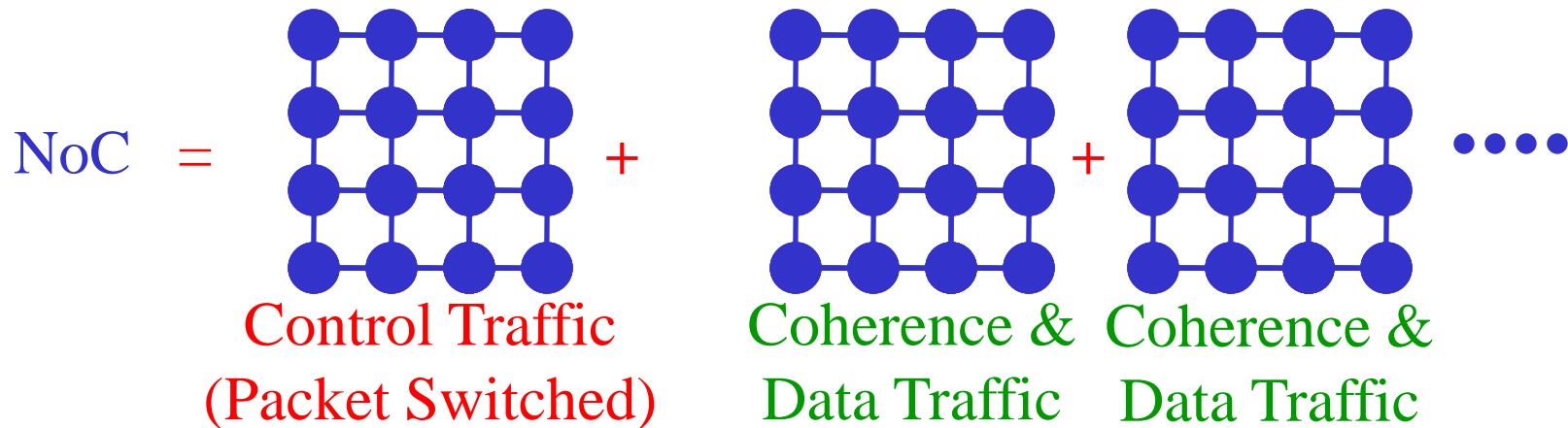
EVC: Express Virtual Channels



HCT: Hybrid circuit switching (PS & CS traffic)

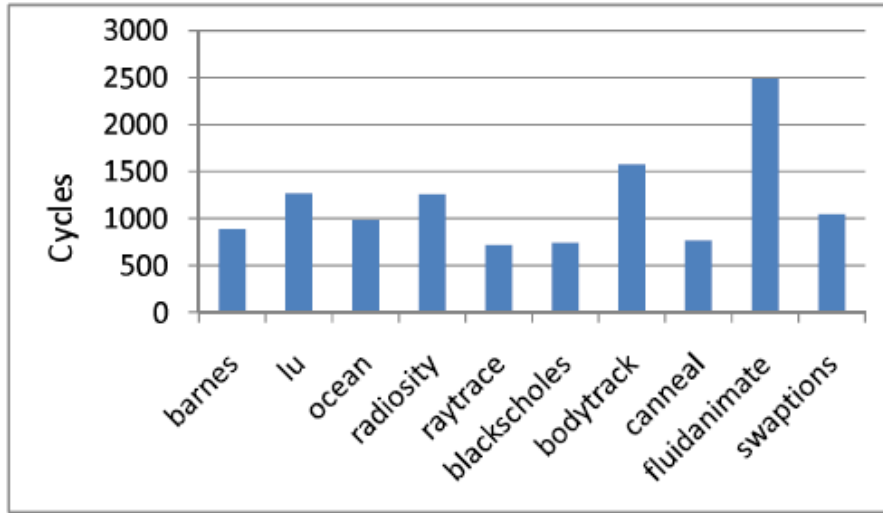


Prior Work – Hybrid Circuit Switched NoC

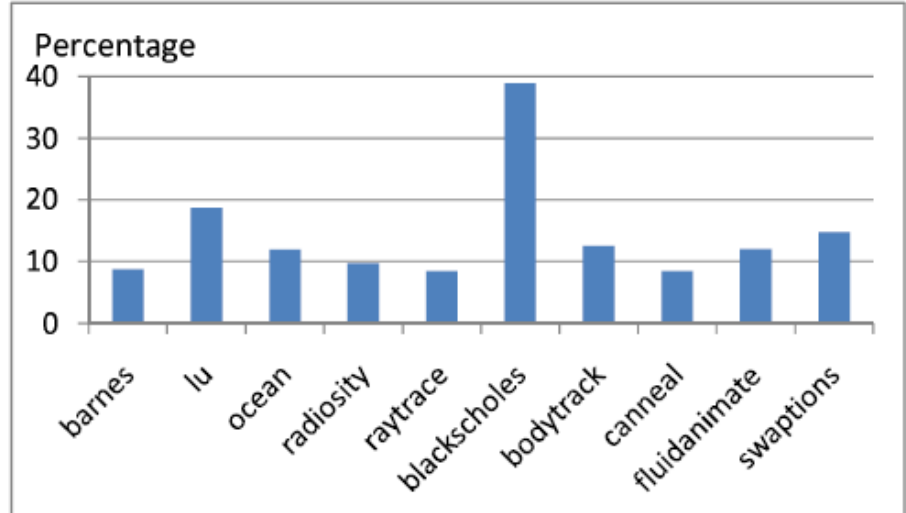




Motivation – What happens when circuits are created on demand



Average number of cycles between sending two consecutive packets from same source to same destination



% of flits traveling on circuits from source to destination

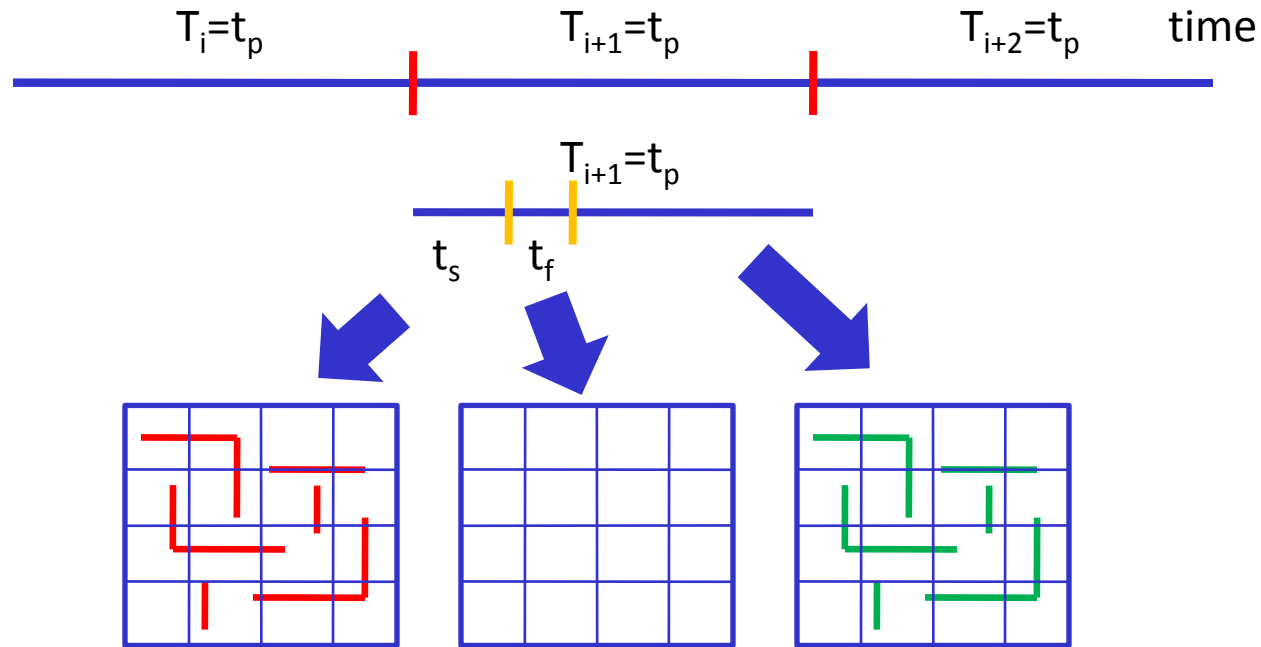


Our Approach - Circuit Configuration and Pinning

- We want to increase circuit re-use
- To take advantage of temporal communication locality, our suggested solution is:
 - Configure the most important circuits
 - Pinning: keep the circuit configuration stable
 - Periodically repeat the above steps to cope with changes in communication



Our Approach - Circuit Configuration and Pinning



- Decide new circuits & configuration
- Not Active Yet

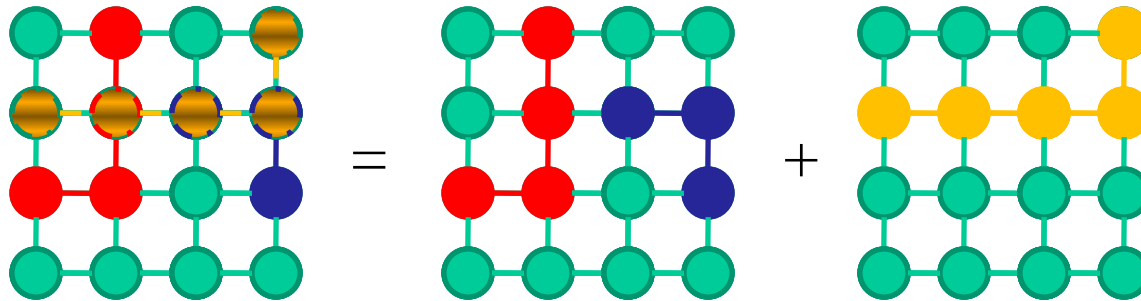
Flush in-flight CS packets

New configuration becomes active



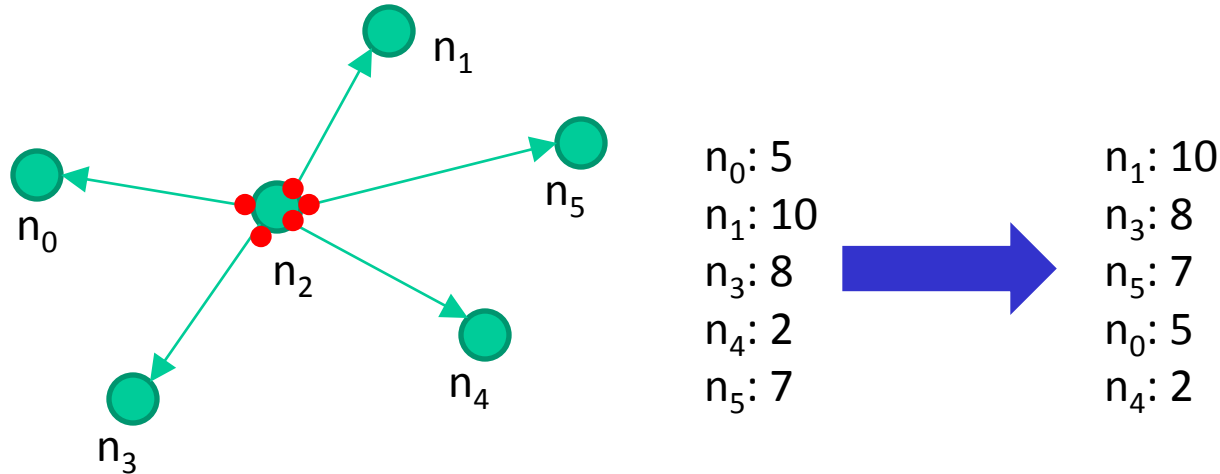
How many circuits can be established from each source?

- Conflicts can arise due to limited resources.
- Can resolve conflicts by using multiple interconnection planes





How to determine the most important circuits?



- Keep count of the packets sent from each source to each destination.
- The greater the number of packets sent, the more important the circuit is.



How to configure Circuits?

- Through a centralized controller

Round 1

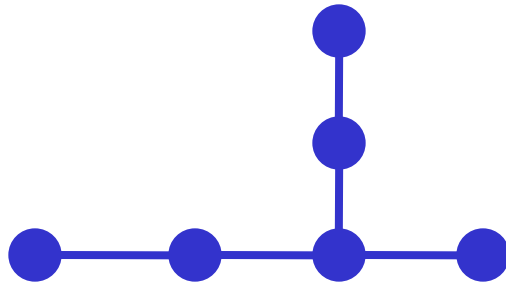
Round 2

n_0	n_1	n_2	n_3	n_4	n_5
n_5	n_0	n_1	n_1	n_0	n_2
<u>n_3</u>	<u>n_4</u>	n_3	<u>n_5</u>	n_5	<u>n_3</u>
<u>n_1</u>	n_5	<u>n_5</u>	n_4	n_2	n_0
n_2	n_3	n_0	n_2	<u>n_1</u>	n_1
n_4	n_2	n_4	n_0	n_3	n_4

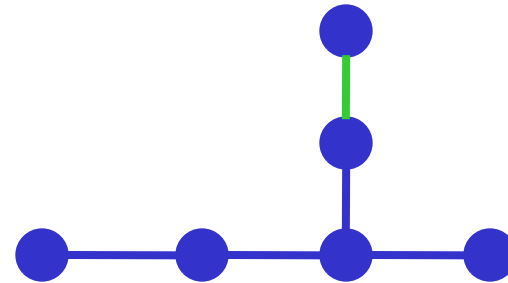
How to configure Circuits?



- Or in a distributed fashion through a two phase reserve-confirm algorithm for each round.



Reservation Succeeds

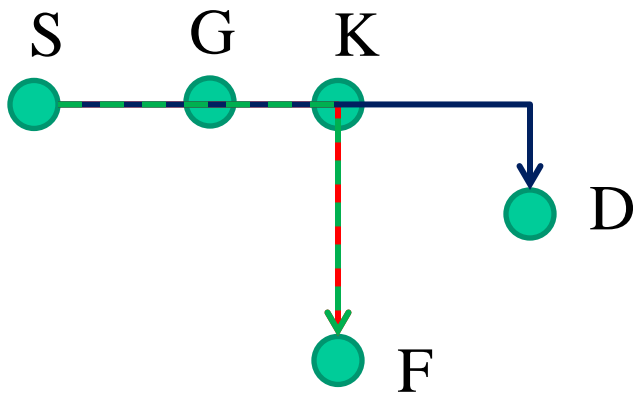


Reservation Fails



Partial Circuit Routing

- With stable circuit configuration, we can use partial circuit routing.
- Use a unary counter to indicate the number of links on the partial circuit.



Unary counter example:

100 = CS 10 = CS 1 = PS

Simulation Environment



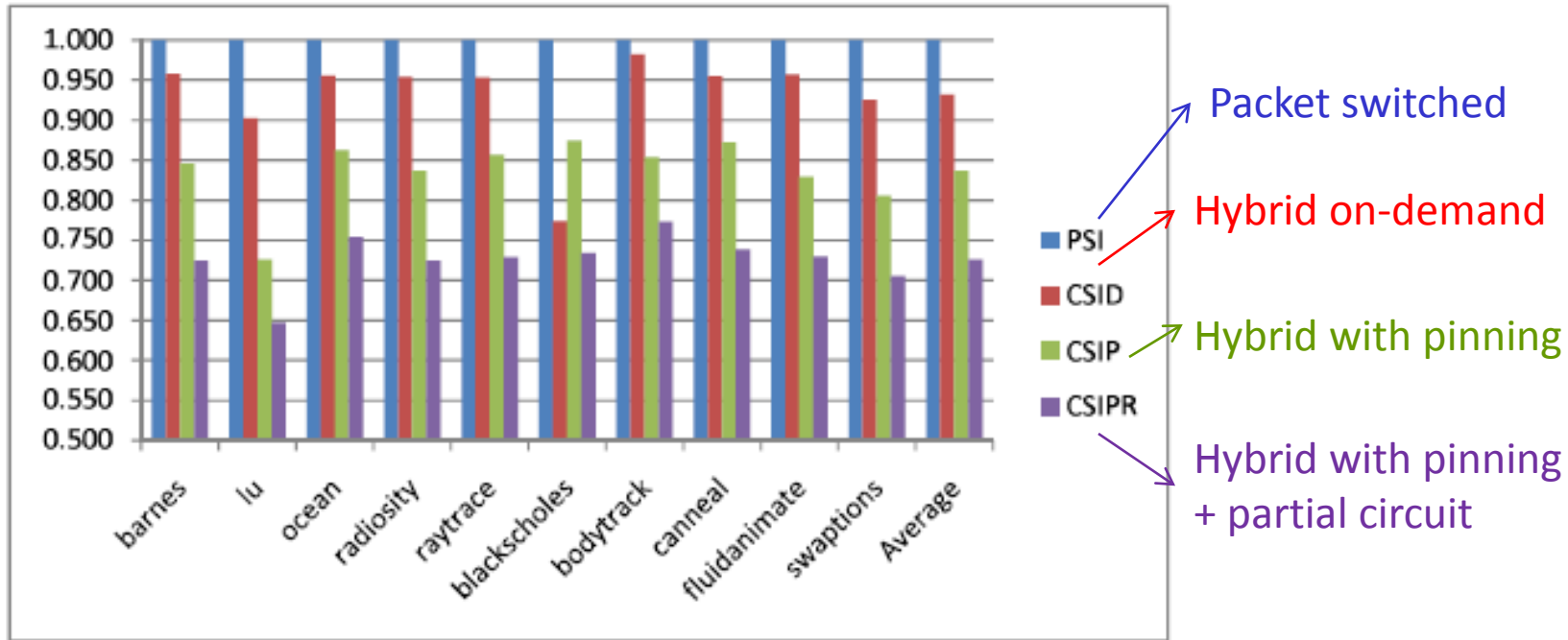
- We use Simics to simulate 16 in-order 2-issue Sparc processors
- Two Cache Organizations
 - Distributed shared L2 (SNUCA)
 - Private L2
- Two Bank sizes
 - 1 MB
 - 256 KB
- Splash-2 and Parsec 1.0 Benchmarks

Simulation Environment



- Four types of NoCs:
 - *Packet Switched Interconnect (PSI): 1 plane, 64 byte link width*
 - *Hybrid Circuit Switched Interconnect with On-Demand Circuit Configuration (CSID): 4 planes, 16 byte link width*
 - *Hybrid Circuit Switched Interconnect with Pinned Circuit Configuration (CSIP): 4 planes, 16 byte link width*
 - *Hybrid Circuit Switched Interconnect with Pinned Circuit Configuration and Partial Circuit Routing (CSIPR): 4 planes, 16 byte link width*

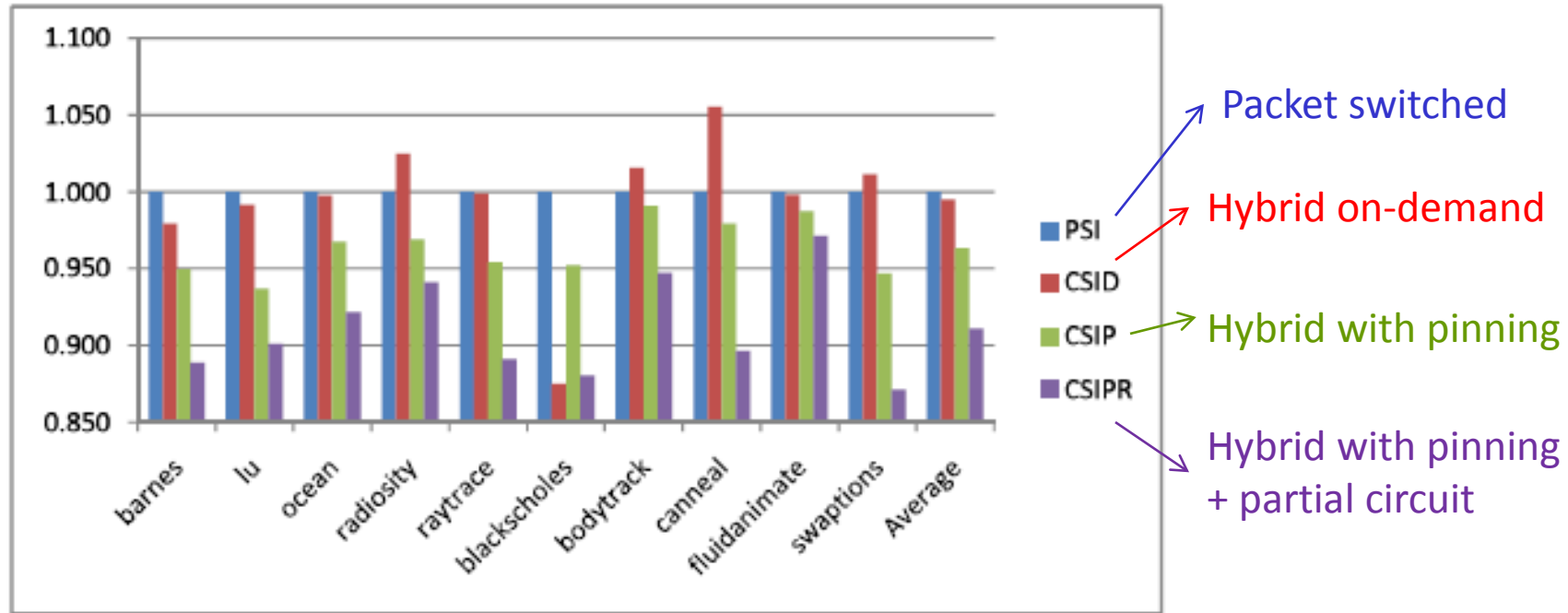
Simulation Results



SNUCA L2 (1MB) - Normalized Average Flit Latency



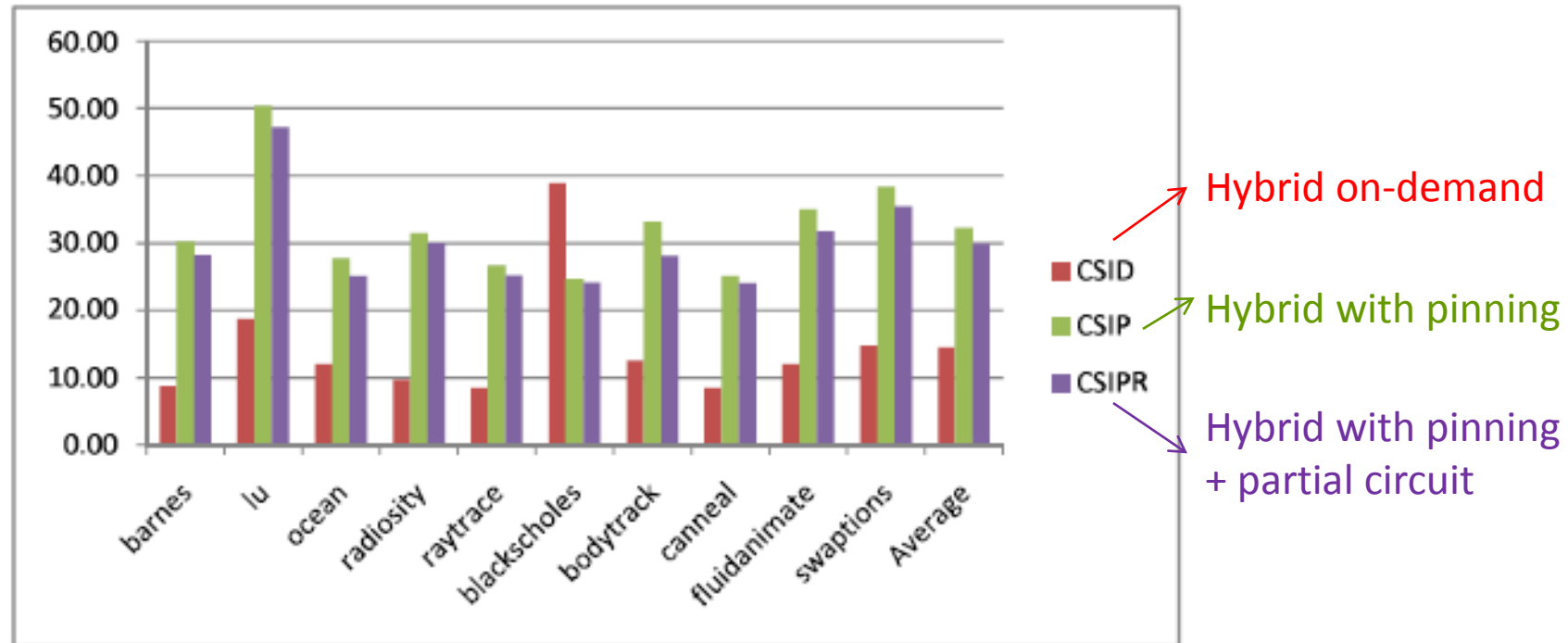
Simulation Results



SNUCA L2 (1MB) - Normalized Execution Time



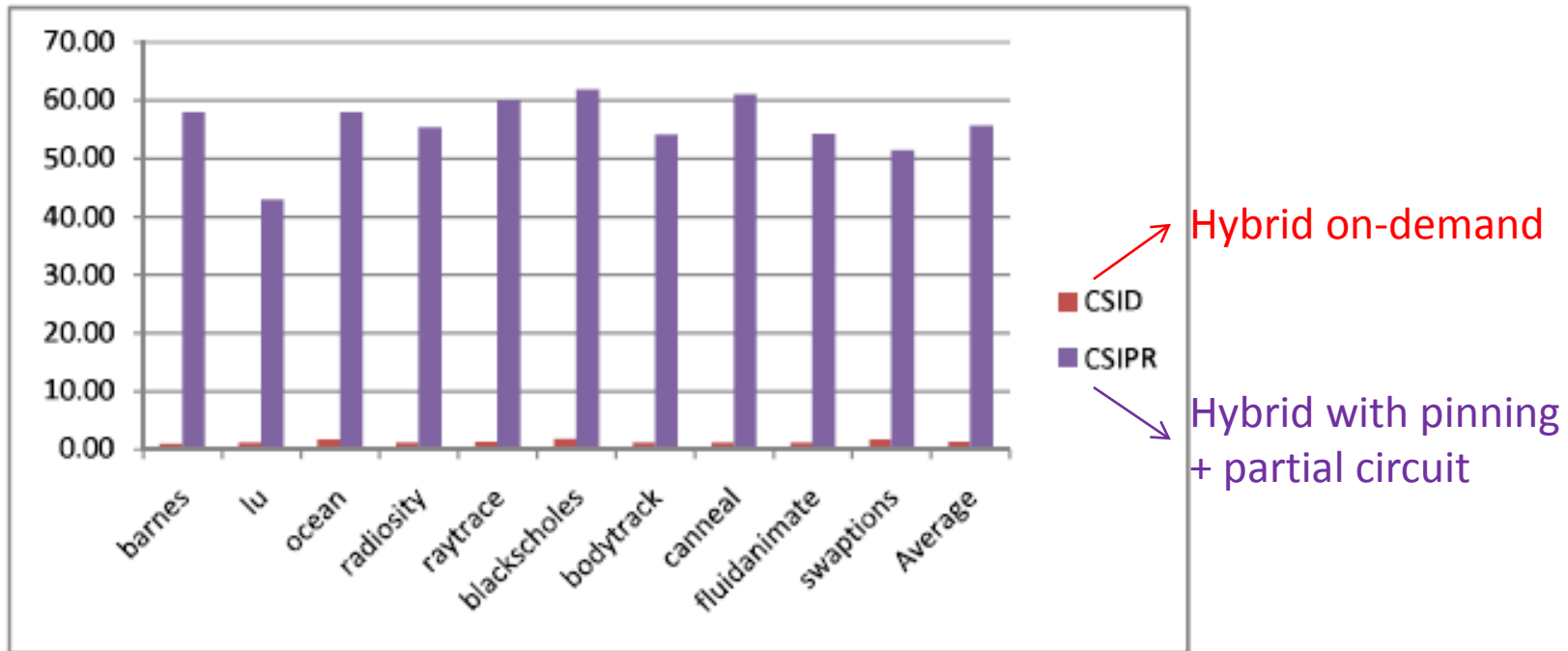
Simulation Results



SNUCA L2 (1MB) - % Flits using complete circuits



Simulation Results



SNUCA L2 (1MB) - % Flits using partial circuits



Conclusion

- Pinning attempts to exploit communication locality to improve communication latency
- Adapt to changes through periodic reconfiguration.
- Pinning alone improves flit latency over on-demand circuits by 10%,
- Partial circuit routing adds another 10% for a total of 20% improvement



Thank You!

Questions?