Carpool: A Bufferless On-Chip Network

Supporting Adaptive Multicast and Hotspot Alleviation

Xiyue Xiang

Wentao Shi, Saugata Ghose, Peng Lu,

Onur Mutlu, Nian-Feng Tzeng

UNIVERSITY OF LOUISIANA Lafayette

LSU LOUISIANA STATE UNIVERSITY

Carnegie Mellon

ETH zürich

Executive Summary

- Problem: bufferless NoCs lack of efficient support for multicast (MC) and hotspot (HS) traffic
- Our Goal: reduce the contention caused by MC and HS traffic in a bufferless NoC with low cost
- Observation: MC flits increase serialization latency and HS flits waste network bandwidth
- Key Idea: fork MC flits adaptively when NoC is not congested and merge HS flits opportunistically
  - Carpool is the first bufferless NoC providing support for multicast and hotspot traffic
- Results
  - 43% lower latency and 8% lower power than conventional bufferless NoC
  - 26% lower latency, 50% lower power, and 64% less area than the buffered NoC with MC/HS support

1. Network-on-Chips Basics

2. Key Observations

3. Our Approach

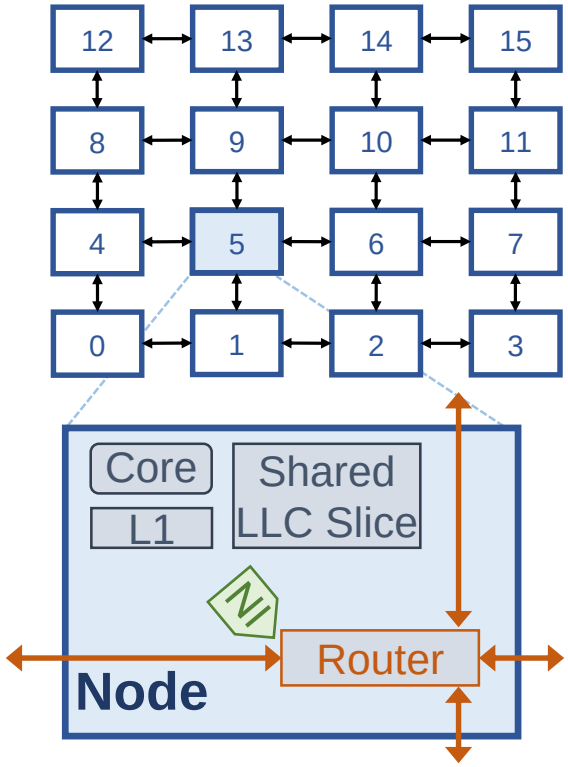
4. Hardware Implementation

5. Evaluation

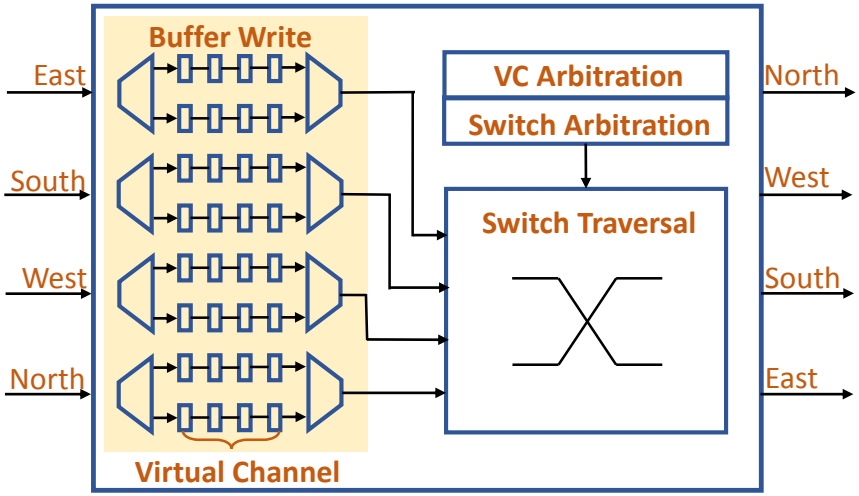
NoCs in the Silicons

Network-on-Chips

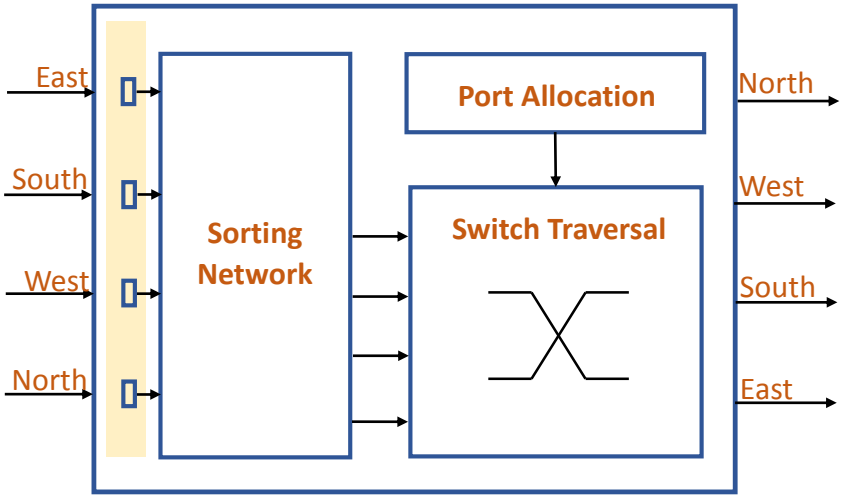
- NoCs carry the communication among nodes on the same die
- Router is the pivots of NoCs, moving traffic from node to node



Buffered vs. Bufferless Router



Buffered router



Bufferless router

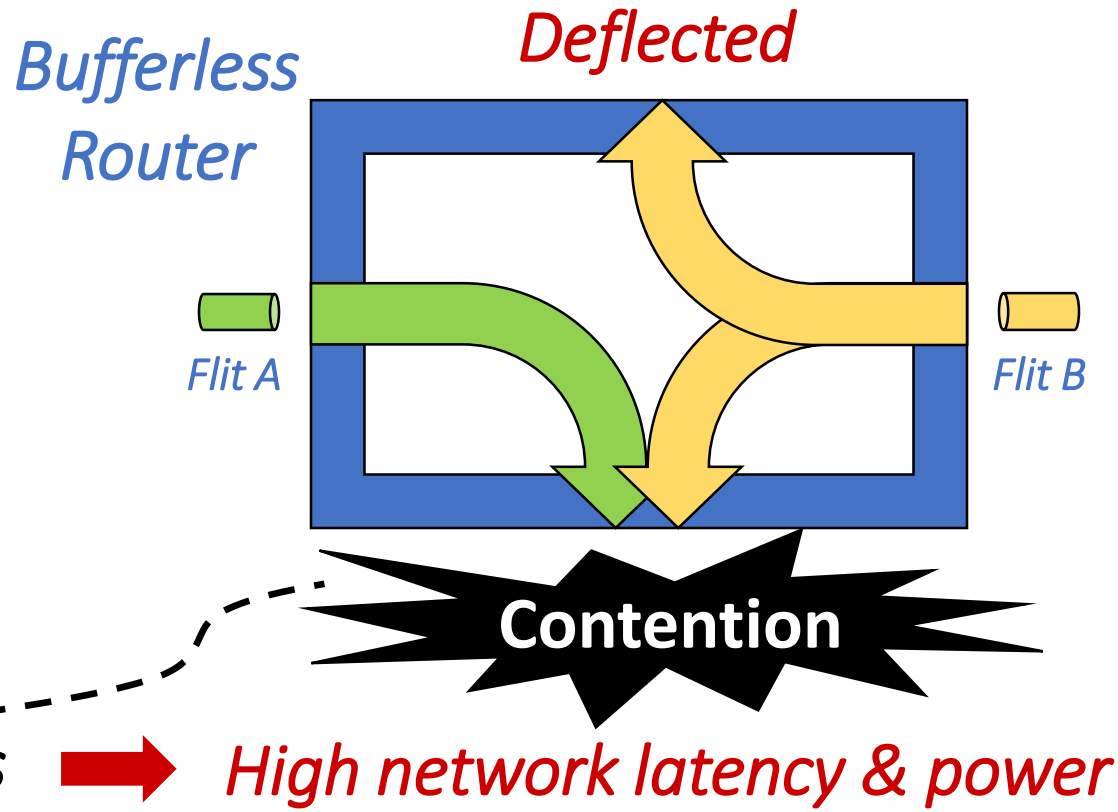
	Buffered NoCs	Bufferless NoCs
Performance	✓	✗
Power	✗	✓
Area	✗	✓

Bufferless NoCs is a compelling design option for future multicore processor due to its simplicity and power-efficiency

Bufferless NoCs Basics

Simple Operations

1. Flit Ranking
2. Port Allocation
3. Forward/Deflect



Bufferless NoCs rely on deflection to resolve flits contention, but avoid deflection as much as possible

Current bufferless NoCs lack of efficient support for multicast and hotspot traffic

1. Network-on-Chips Basics

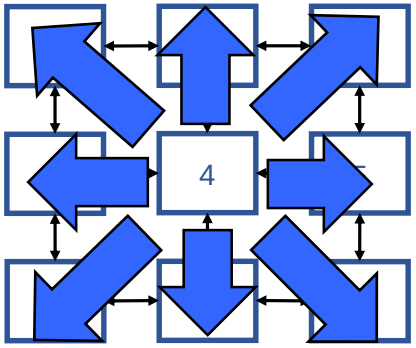
2. Key Observations

3. Our Approach

4. Hardware Implementation

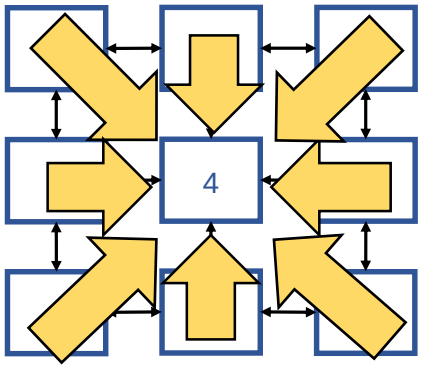
5. Evaluation

Multicast and Hotspot in NoCs



Multicast

- Originate from one node destined to multiple nodes
- Occur: invalidation
- Issue: long serialization latency

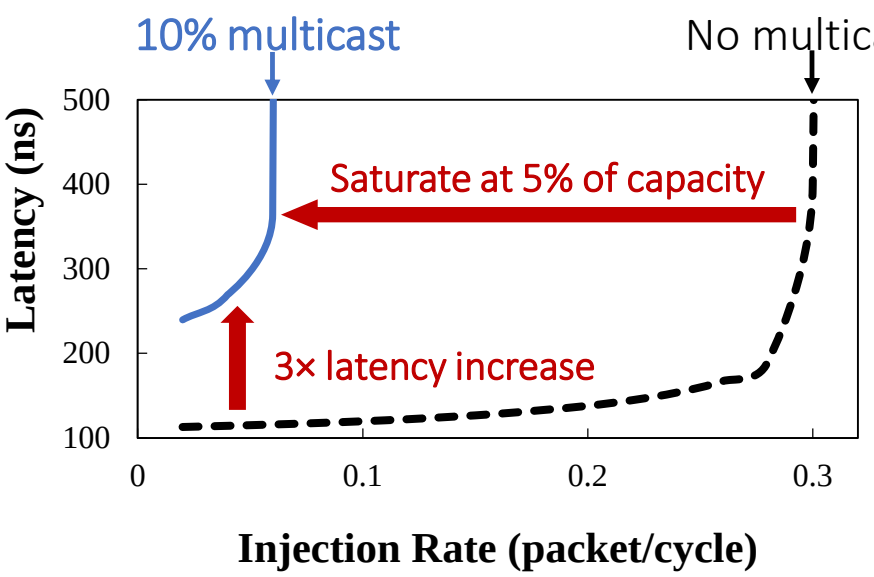


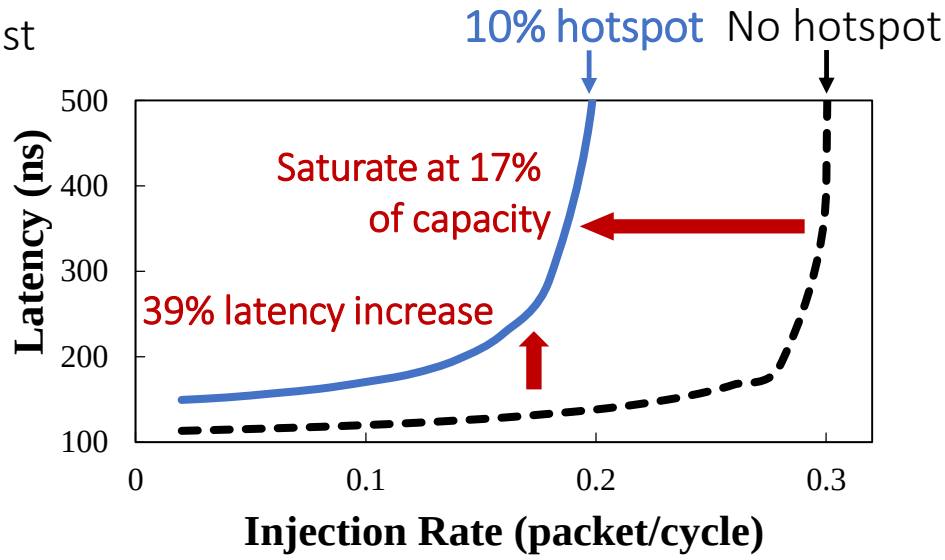
Hotspot

- Originate from multiple nodes with the same destination and payload
- Occur: acknowledgement, shared lock variables access
- Issue: waste network bandwidth

Providing support for MC and HS is very important for a bufferless NoC to deliver high performance

Impact of MC and HS Traffic In Bufferless NoCs





Multicast and Hotspot Traffic Impact:
Increase network latency & saturate NoCs prematurely

***Our Goal:** reduce the contention caused by multicast and hotspot traffic in bufferless NoCs with low complexity*

1. Network-on-Chips Basics

2. Key Observations

3. Our Approach

4. Hardware Implementation

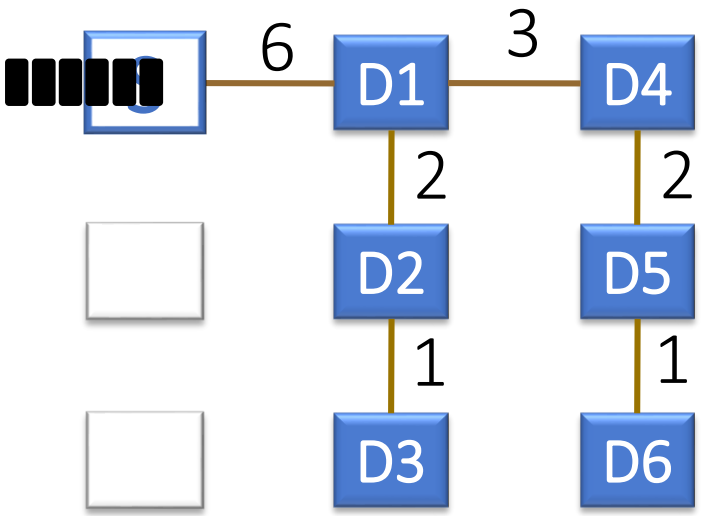
5. Evaluation

Our Approach – Carpool

- *Multicast Flit Forking*
  - Source NI injects single multicast request
  - Adaptively fork multicast flits when NoCs is not congested
  - Congestion measure: starvation rate

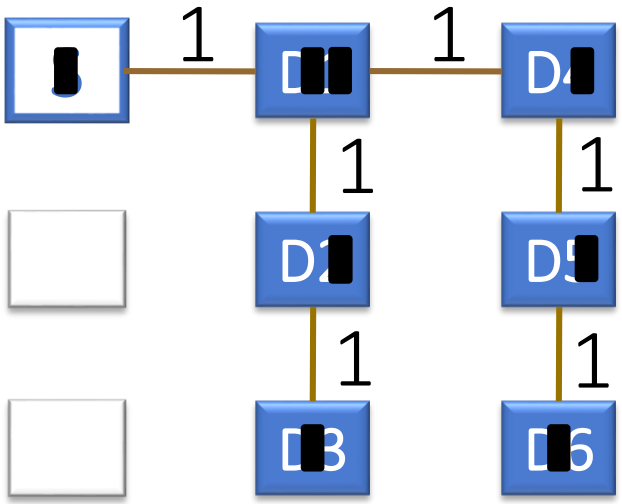
- *Hotspot Flit Merging*
  - *Tagged at the originating nodes*
  - *Intermediate routers detect hotspot flits*
  - *Some **hotspot flits** are quietly* dropped

Multicast Flit Forking



Naïve bufferless

15 transfers, 10 cycles

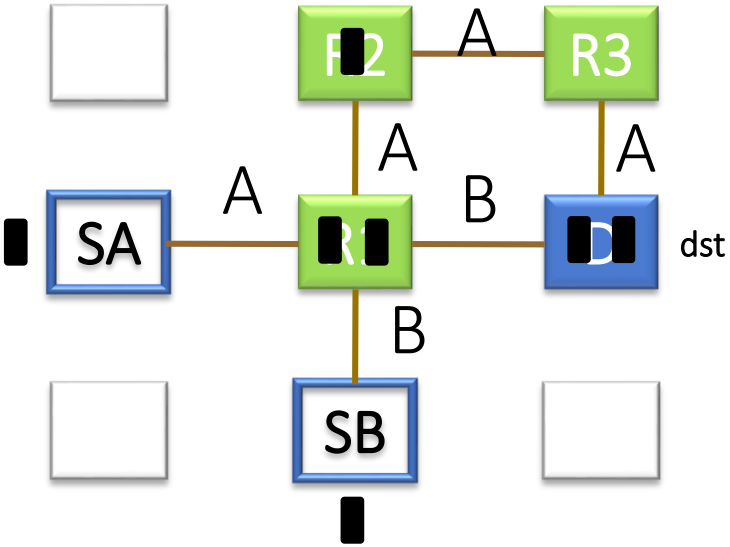


Carpool

6 transfers, 5 cycles

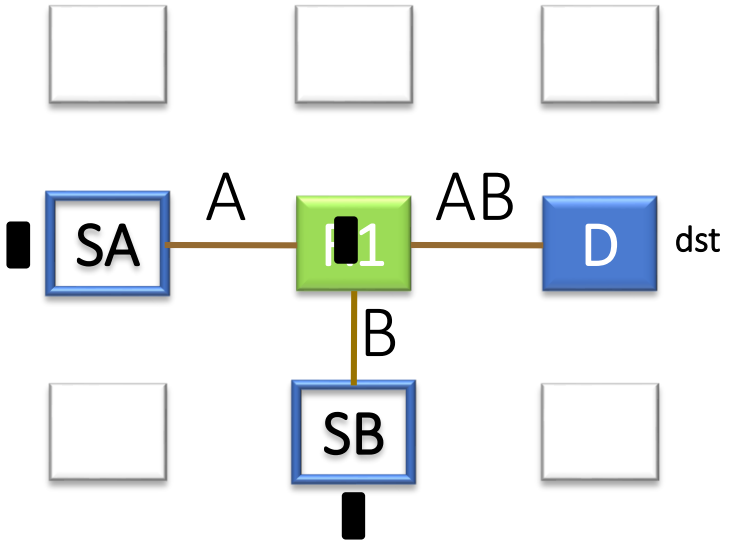
Forking multicast flits reduces the serialization latency at the source network interface

Hotspot Flit Merging



Naïve bufferless

6 transfers, 5 cycles



Carpool

3 transfers, 2 cycles

Merging hotspot flits reduces network load and improves network bandwidth utilization

Design Features

- Deadlock-free

$$incoming - removed + replicas \leq outPorts$$

- Efficient & scalable encoding
  - Two-level hierarchical representation

    {clusterID, nodeList}
  - No wire overhead
    - *nodeList* shares half of payload channels (64-bit)
    - Tradeoff: sending more flits
- Low-cost hotspot flit merging (11.5% of router area)
  - Only compare flits on higher-numbered ports

1. Network-on-Chips Basics

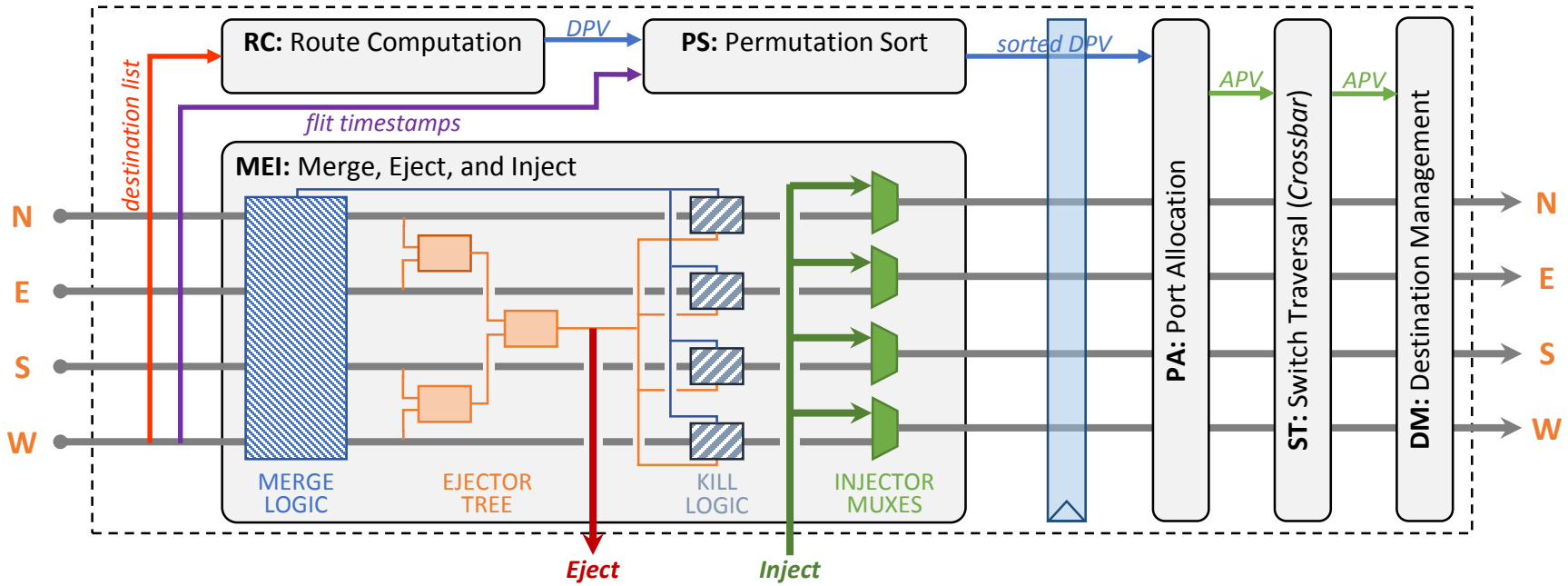
2. Key Observations

3. Our Approach

4. Hardware Implementation

5. Evaluation

Router Microarchitecture



❶ **RC**: Find out desired port vector (*DPV*)

❷ **PS**: Find flit with the highest rank

❸ **MEI**: Merge *HS* flits, eject a local-destined flit, and inject new flit

❹ **PA**: Compute allocated port vector (APV) based on sorted *DPV*

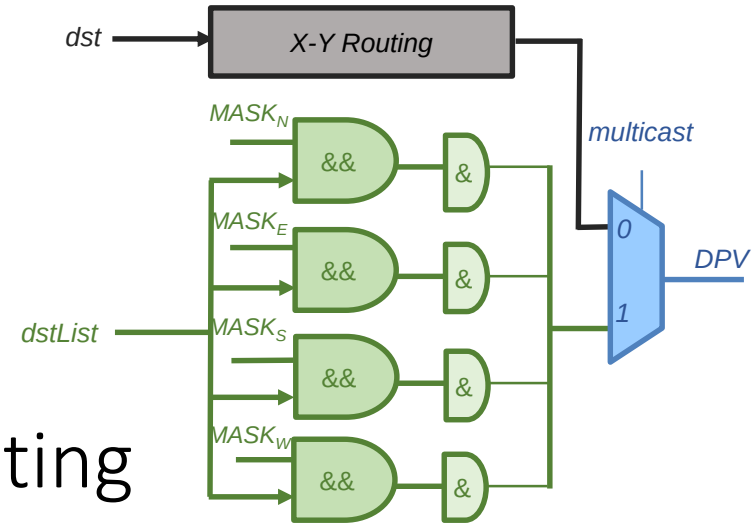
❺ **ST**: Mux the flit to output ports

❻ **DM**: Update *nodeList* of *MC* flit

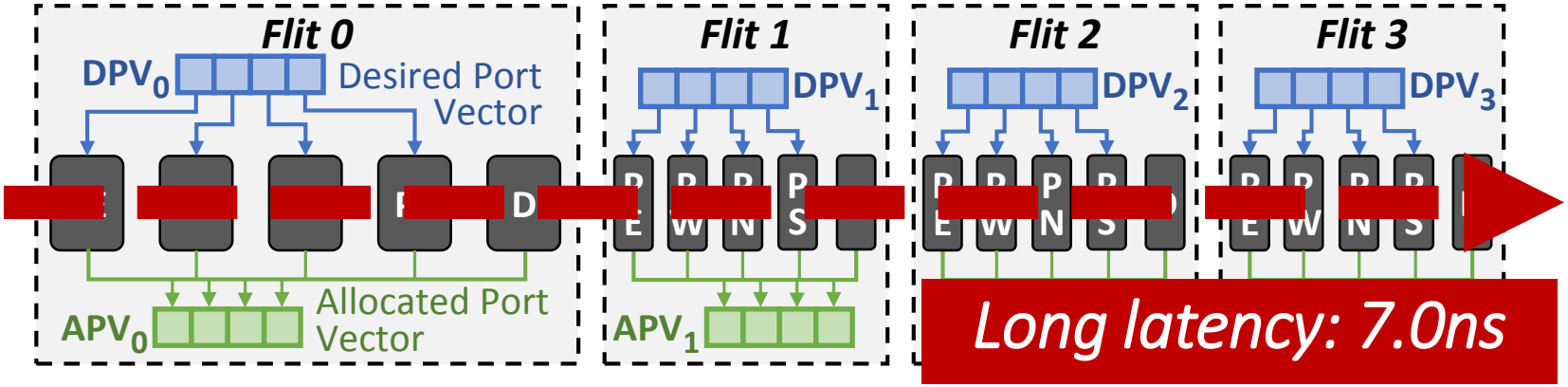
Carpool router is very simple and efficient

Route Computation

- Partition the network into *NE/SE/SW/NW* quadrant, mapping to *N/E/S/W* port
  - Each port has a bit-vector (i.e., *MASK*) to indicate nodes assigned to the mapped quadrant
- For *MC* flits, outputs are assigned based on which quadrant contains its destinations.
- For *UC/HS* flits, use X-Y routing



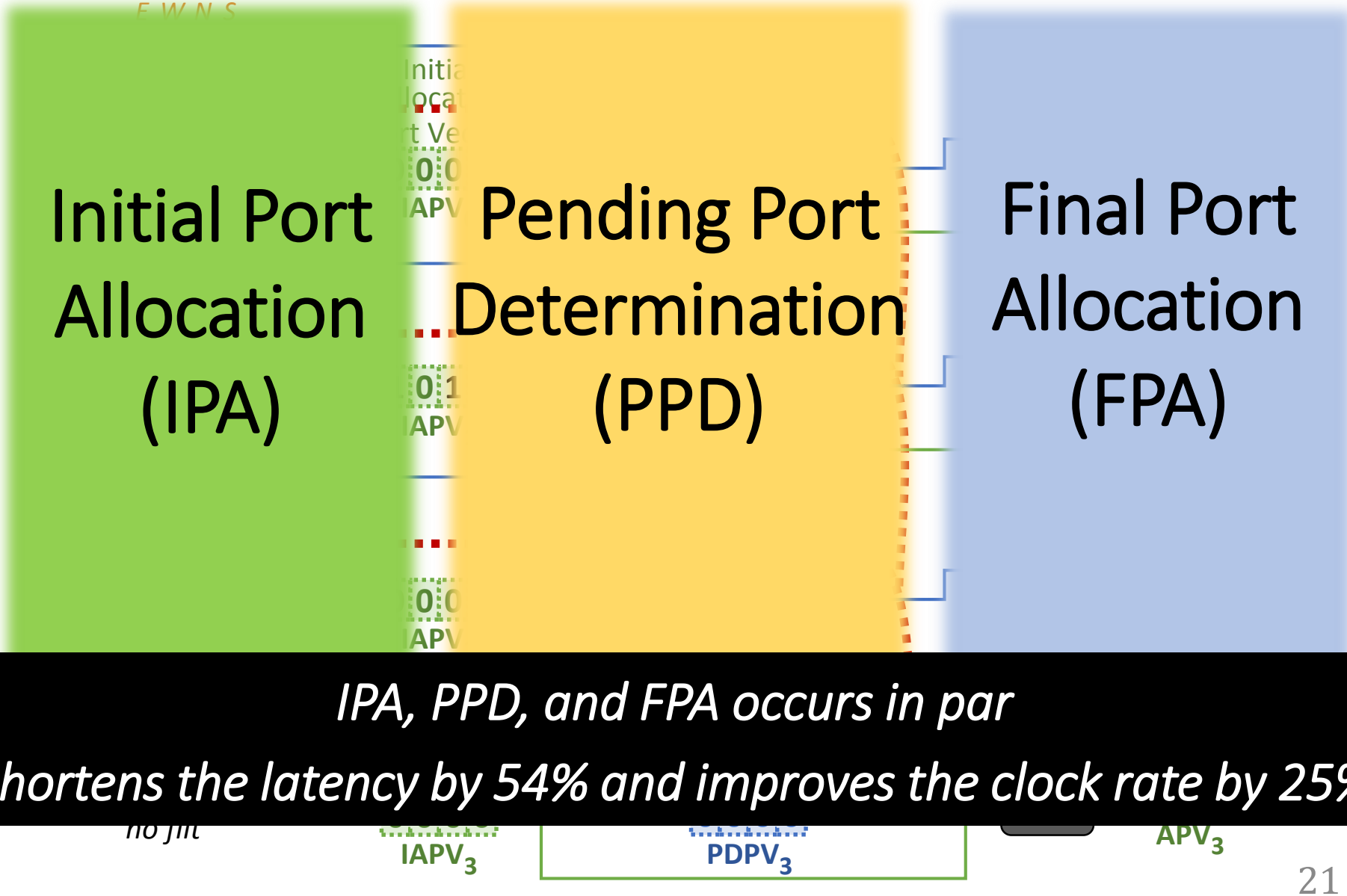
Naïve Port Allocation



- **Reasons** for *sequential* allocation
  - Enforce strict priority ➡ *Not necessary*
  - Avoid deadlock due to multicast

Sequential port allocation is over-provisioned and creates long critical path latency

Parallel Port Allocation



IPA, PPD, and FPA occurs in par

Shortens the latency by 54% and improves the clock rate by 25%

1. Network-on-Chips Basics

2. Key Observations

3. Our Approach

4. Hardware Implementation

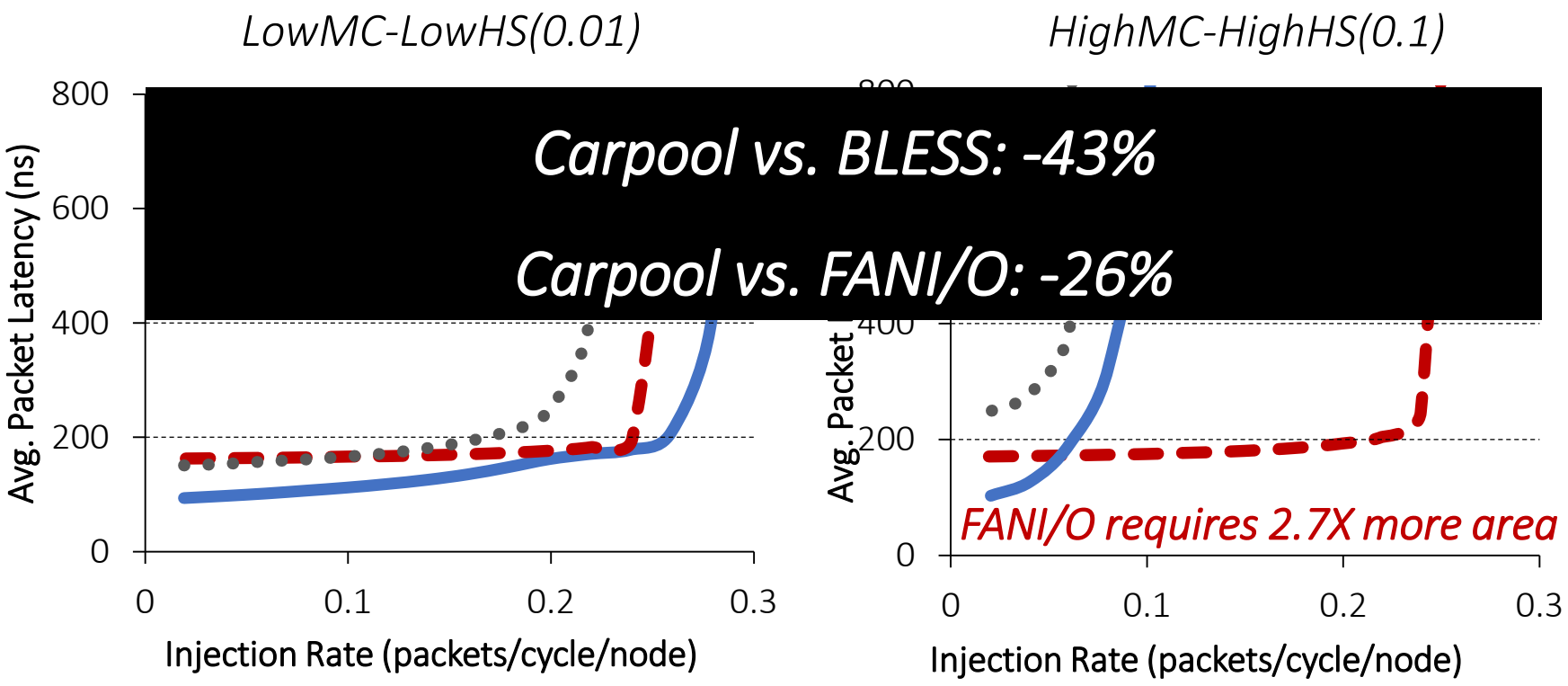
5. Evaluation

Methodology

- *Emulate the injection rate, multicast, and hotspot behavior of real system*
- *Generated packets have the probability of mc_rate/hs_rate to be a multicast/hotspot packet*
  - *probability: 0.01(Low), 0.05(Mid), 0.1(High)*
- *Area and latency are obtained through RTL synthesis based on 35nm standard cell library*

Latency

Sweep MC and HS rate for BLESS, FANI/O, Carpool

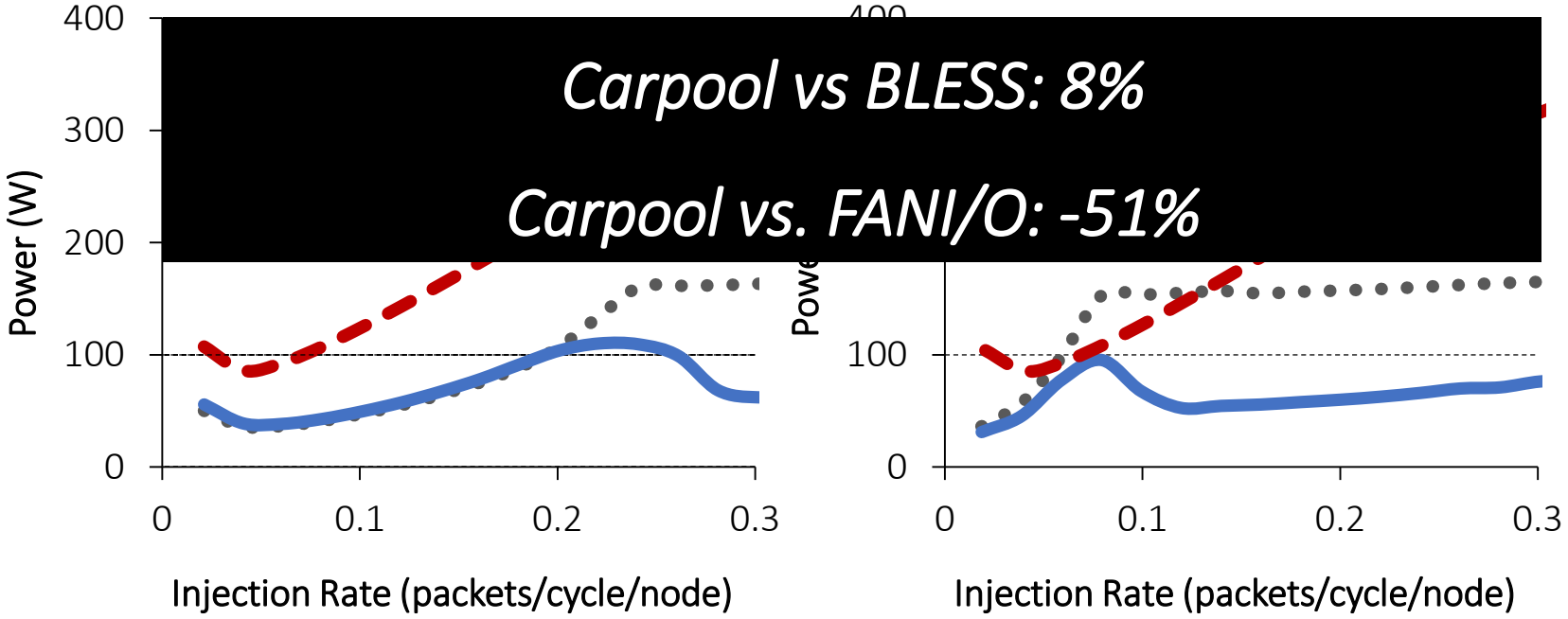


Carpool resolves network congestion caused by multicast and hotspot traffic, reducing latency

Power

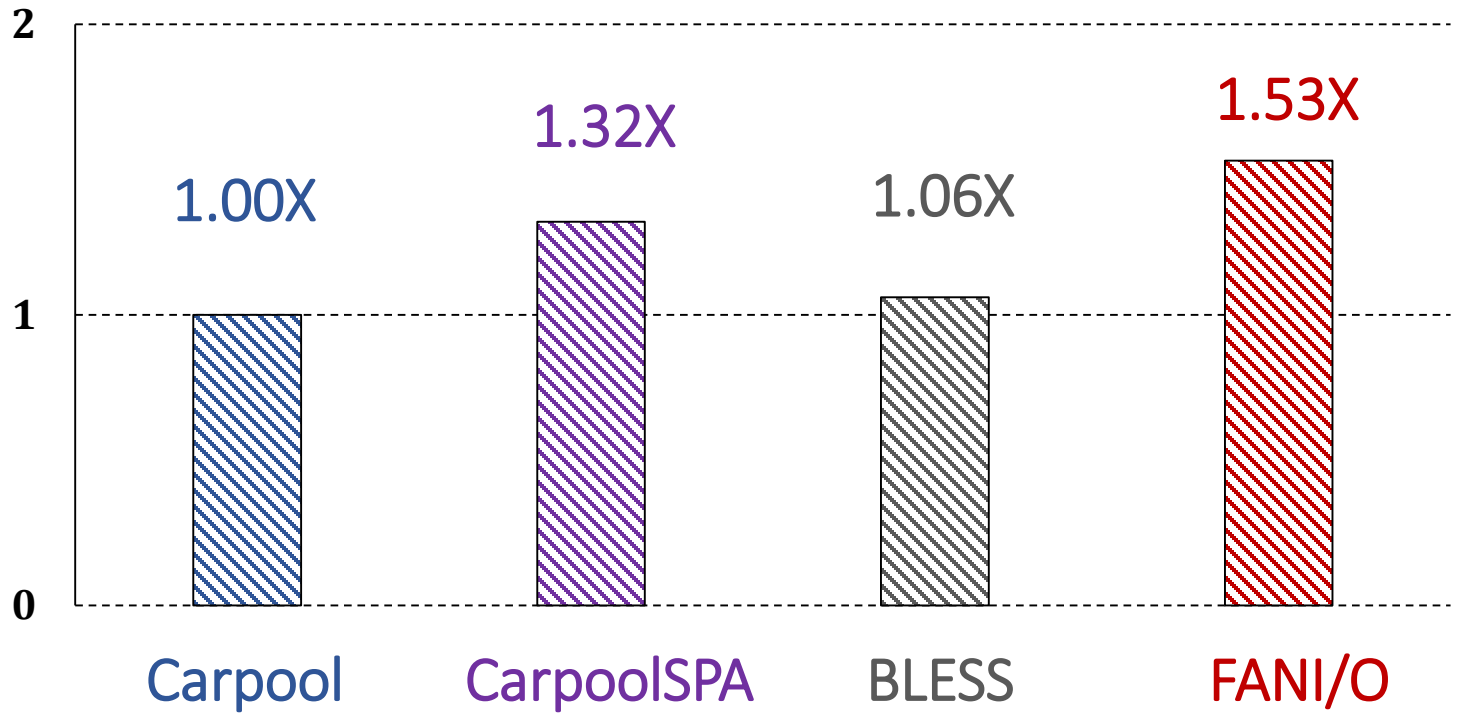
Sweep MC and HS rate for BLESS, FANI/O, Carpool

LowMC-LowHS(0.01) *HighMC-HighHS(0.1)*



Carpool reduces power due to fewer packet injection and reduced hardware complexity

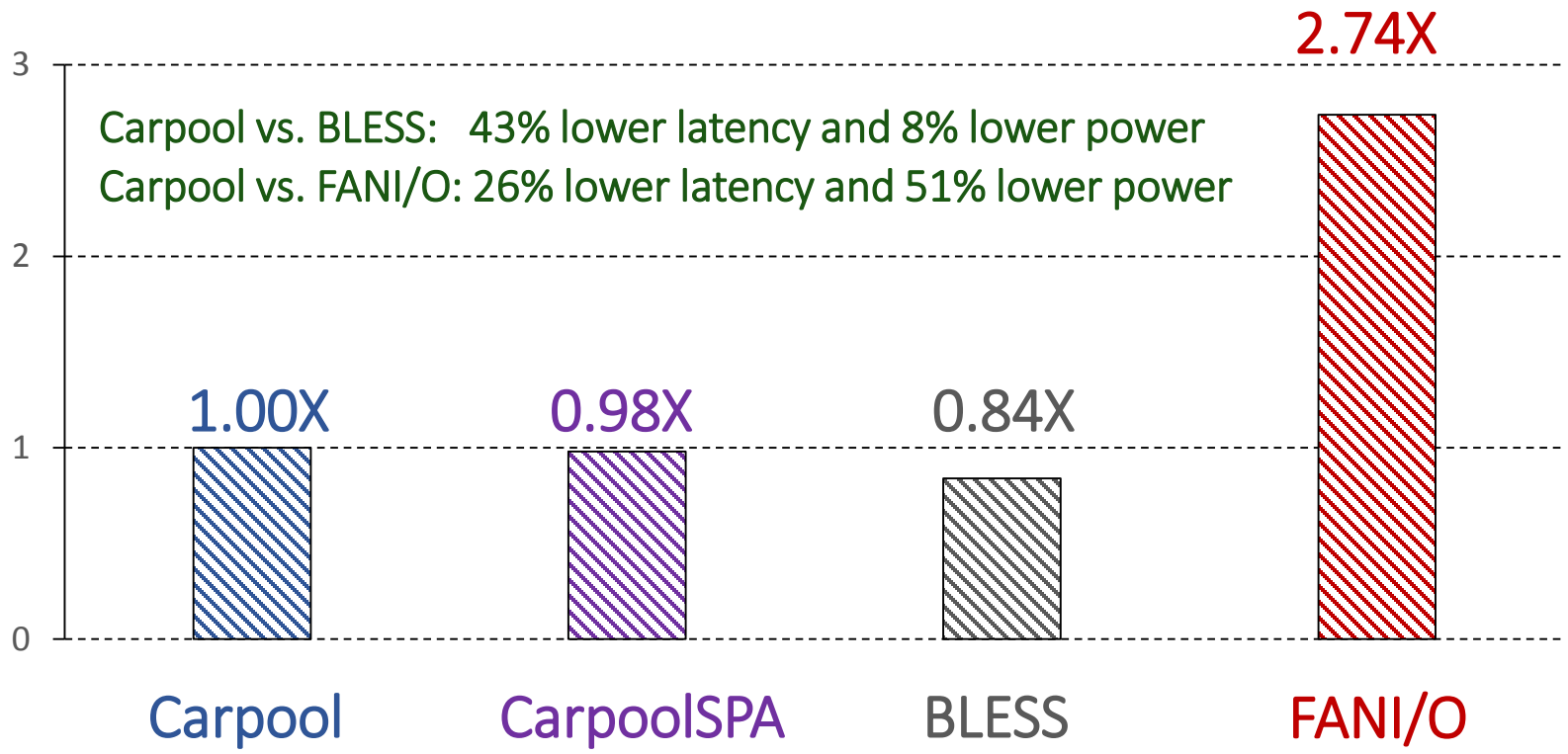
Critical Path Latency



Synthesis using Cadence Encounter based on 35*nm* standard cell library

Carpool increases the router clock rate and lifts up the network bandwidth

Area



Carpool requires much smaller area than the buffered counterpart

Summary

- Problem: bufferless NoCs lack of efficient support for multicast (MC) and hotspot (HS) traffic
- Our Goal: reduce the contention caused by MC and HS traffic in a bufferless NoC with low cost
- Observation: MC flits increase serialization latency and HS flits waste network bandwidth
- Key Idea: fork MC flits adaptively when NoC is not congested and merge HS flits opportunistically
  - Carpool is the first bufferless NoC providing support for multicast and hotspot traffic
- Results
  - 43% lower latency and 8% lower power than conventional bufferless NoC
  - 26% lower latency, 50% lower power, and 64% less area than the buffered NoC with MC/HS support

Carpool: A Bufferless On-Chip Network

Supporting Adaptive Multicast and Hotspot Alleviation

Xiyue Xiang

Wentao Shi, Saugata Ghose, Peng Lu,

Onur Mutlu, Nian-Feng Tzeng

UNIVERSITY OF LOUISIANA Lafayette

LSU LOUISIANA STATE UNIVERSITY

Carnegie Mellon

ETH zürich

Backup Slides

Related Work

- *Bufferless NoCs*
  - **Deflection-based**: [Moscibroda+ ISCA'09], [Fallin+ HPCA'11], [Fallin+ NOCS'12], [Kim+ CAL'13], [Rachata+ SBAC-PAD'14], [Kim+ NOCS'14], [Xiang+ IPDPS'16]
  - **Drop-based**: [Hayenga+ MICRO'09]
  - **Source-throttling**: [Chang+ SBAC-PAD'12], [Nychis+ SIGCOMM'12], [Daya+ DAC'16]
- *Buffered NoCs with Multicast Support*
  - **Path-based**: [Goossens+ IEEE D&T'05], [Lu+ ISVLSI'06]
  - **Tree-based**: [Jin+ HPCA'07], [Jerger+ ISCA'08], [Samman+ DATE'08], [Rodrigo+ MICRO'08], [Wang+ NOCS'09], [Krishna+ MICRO'11]
  - **Hybrid:** [Abad+ HPCA'09]

Flit Format

Unicast

Size	2	6	6	8	3	3	6	128
Field	*pkt type*	*reqID*	*mshrID*	*timestamp*	*size*	*seq#*	*dst*	*payload*

Multicast

Size	2	6	6	8	4	4	4	64	64
Field	*pkt type*	*reqID*	*mshrID*	*timestamp*	*size*	*seq#*	*cluster ID*	*nodeList*	*payload*

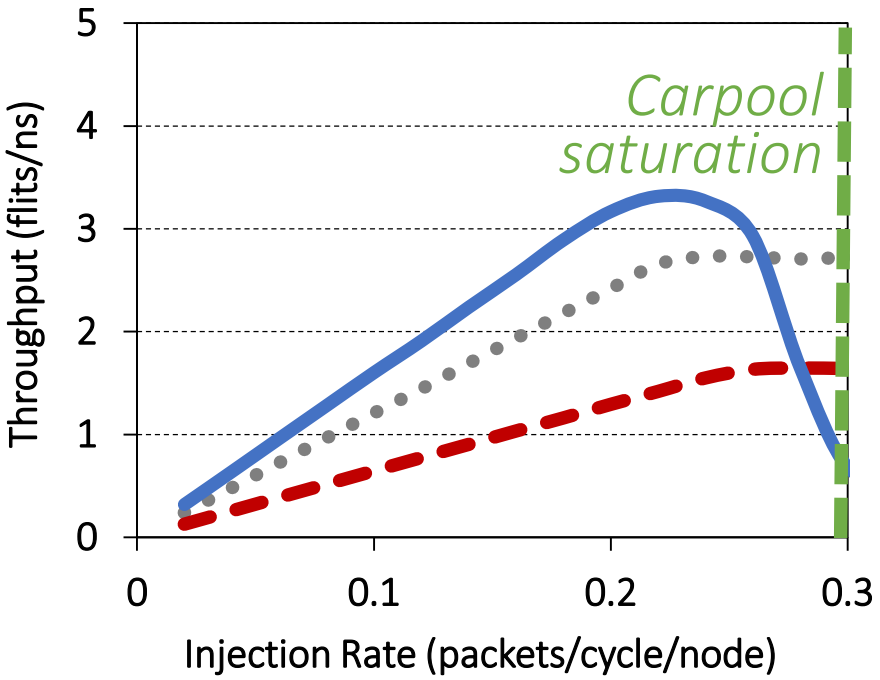
Hotspot

Size	2	6	6	8	4	4	4	64	64
Field	*pkt type*	*dst*	*mshrID*	*timestamp*	*size*	*seq#*	*cluster ID*	*nodeList*	*payload*

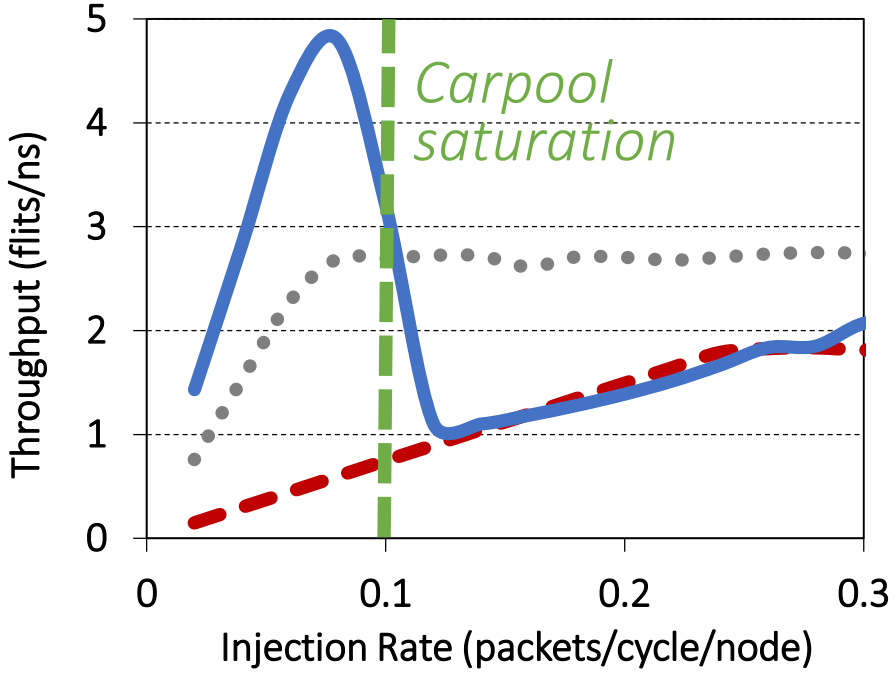
Throughput

Sweep MC and HS rate for BLESS, FANI/O, Carpool

LowMC-LowHS(0.01)

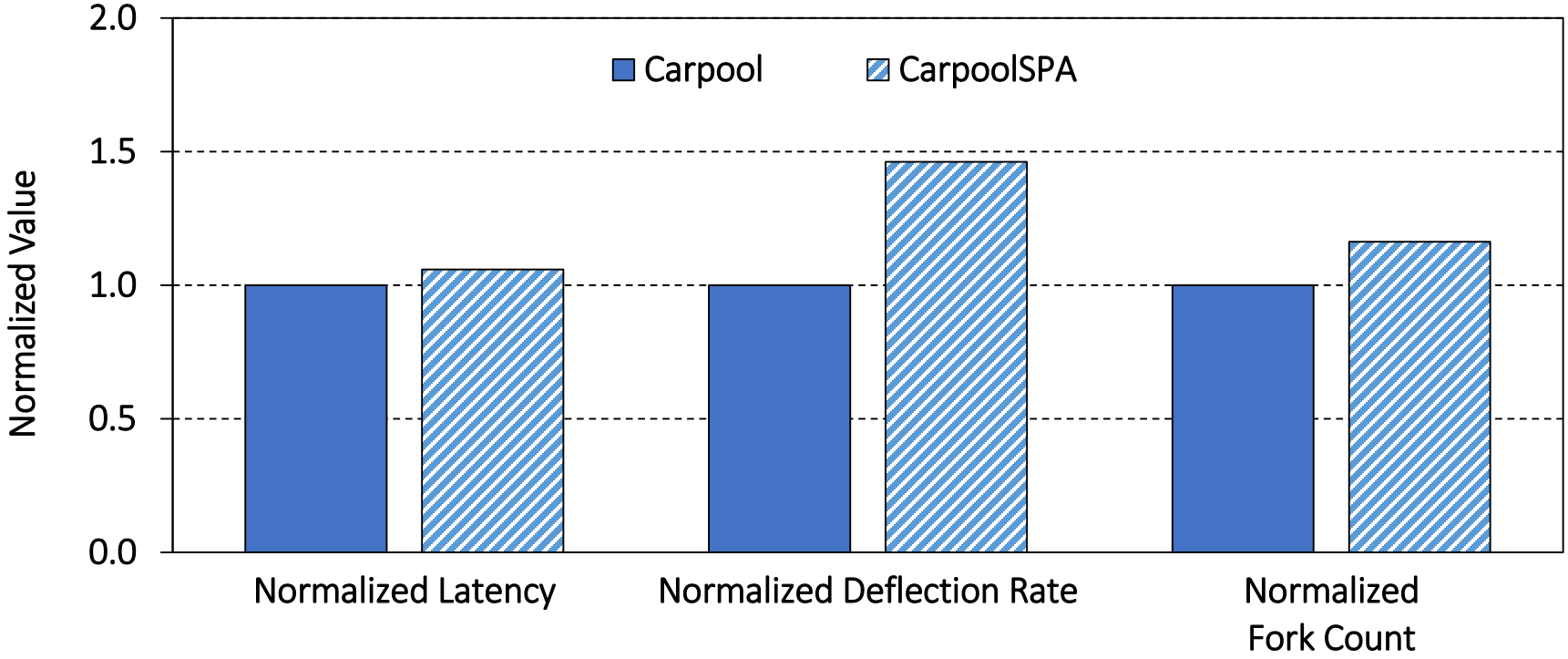


HighMC-HighHS(0.1)



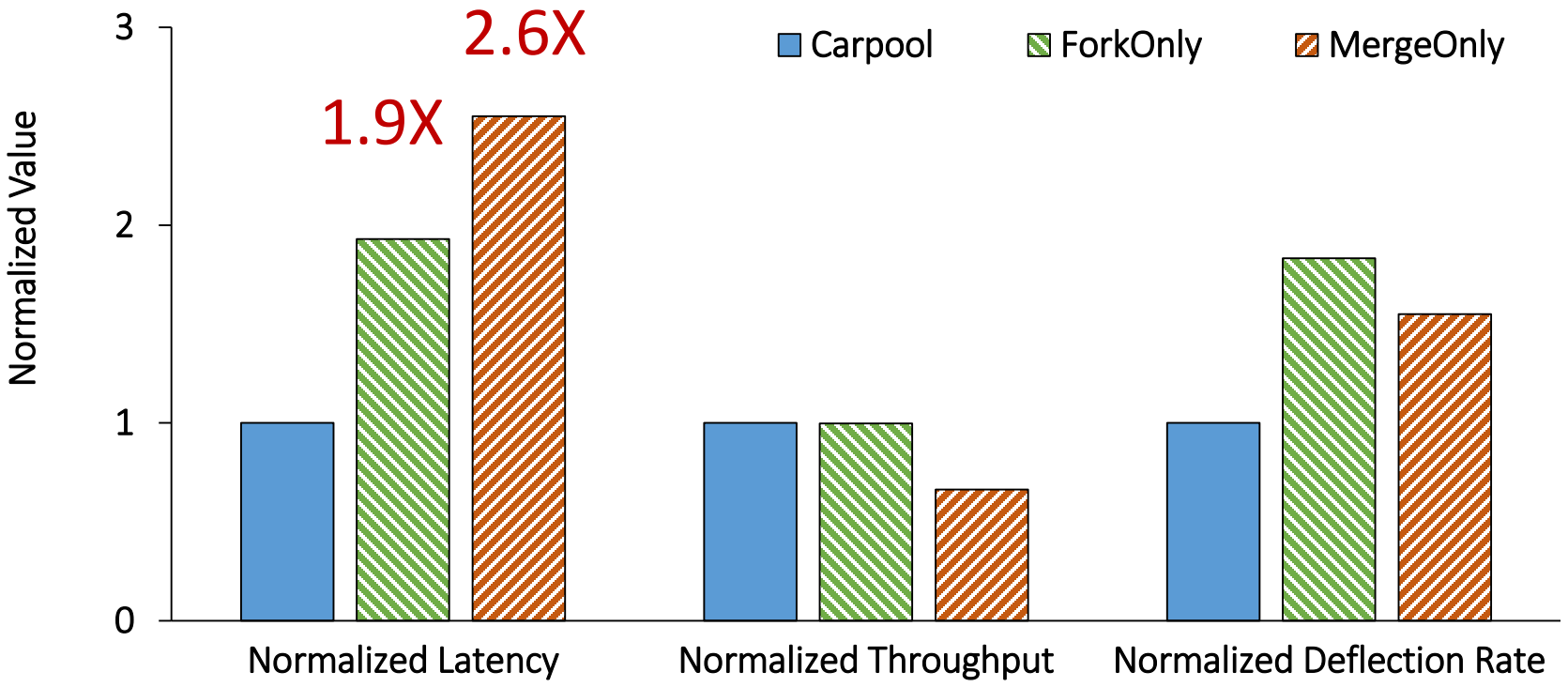
Carpool delivers much better throughput prior network saturation than both BLESS and FANI/O

Effect of Parallel Port Allocation



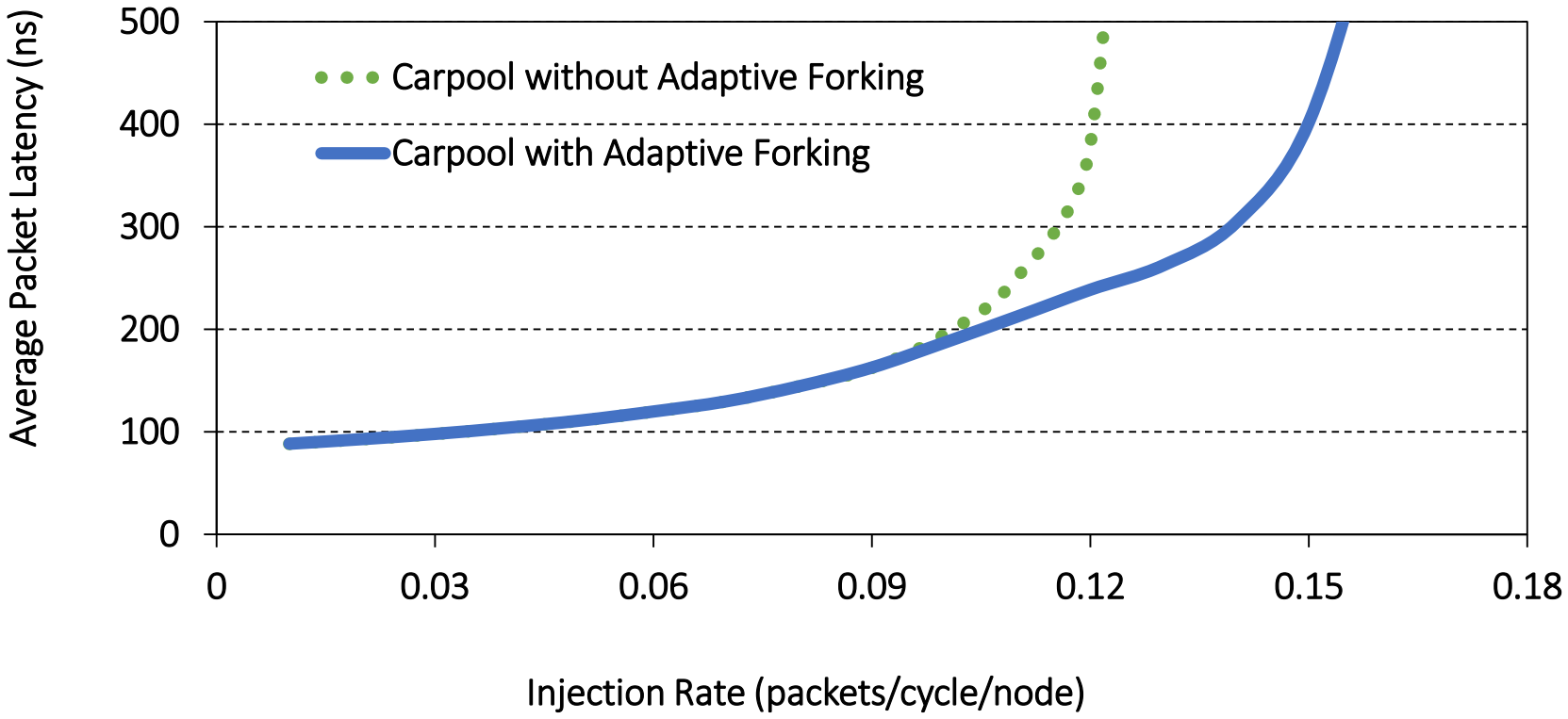
Carpool forks flits only when desired outputs are not contended, therefore reducing both deflection rate and latency

Performance Breakdown



Forking and merging in Carpool significantly reduce the deflection and improve performance

Effect of Adaptive Forking



Adaptive forking prevents NoCs being saturated prematurely by replicated multicast flits

Carpool: A Bufferless On-Chip Network

Supporting Adaptive Multicast and Hotspot Alleviation

Xiyue Xiang

Wentao Shi, Saugata Ghose, Peng Lu,

Onur Mutlu, Nian-Feng Tzeng

UNIVERSITY OF LOUISIANA Lafayette

LSU LOUISIANA STATE UNIVERSITY

Carnegie Mellon

ETH zürich