[1]Georgia Tech

[2]Univ of Toronto

[3]Texas A&M

[4]Univ of Wisconsin

SEEC: Stochastic Escape Express Channel

Mayank Parasar[1], Natalie Enright Jerger[2], Paul V. Gratz[3],
Joshua San Miguel[4], Tushar Krishna[1]

https://github.com/noc-deadlock/seec

Presented By:

Tushar Krishna
Associate Professor
School of ECE
Georgia Institute of Technology
Email: tushar@ece.gatech.edu

November 16, 2021

Interconnection Networks are at the heart of HPC

News

Chinese 260-core processors ShenWei SW26010 enabled supercomputer Sunway TaihuLight be the most productive in the world

June 21, 2016 331 0

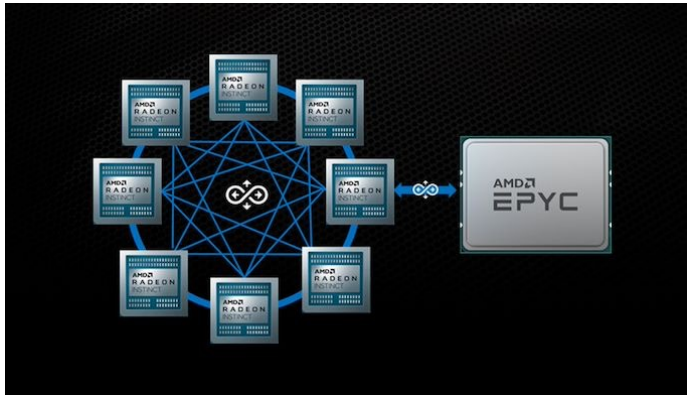
Technology

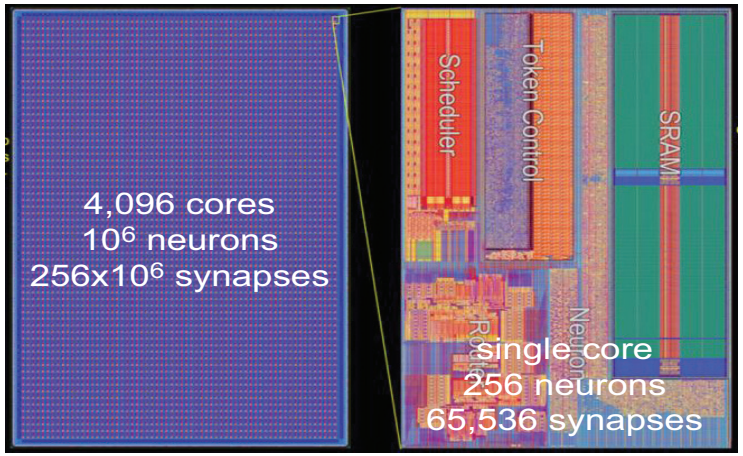
IBM reveals 'brain-like' chip with 4,096 cores

AMD's Third-Gen Infinity Architecture Enables Coherent CPU-GPU Communication

By Francisco Pires 8 days ago

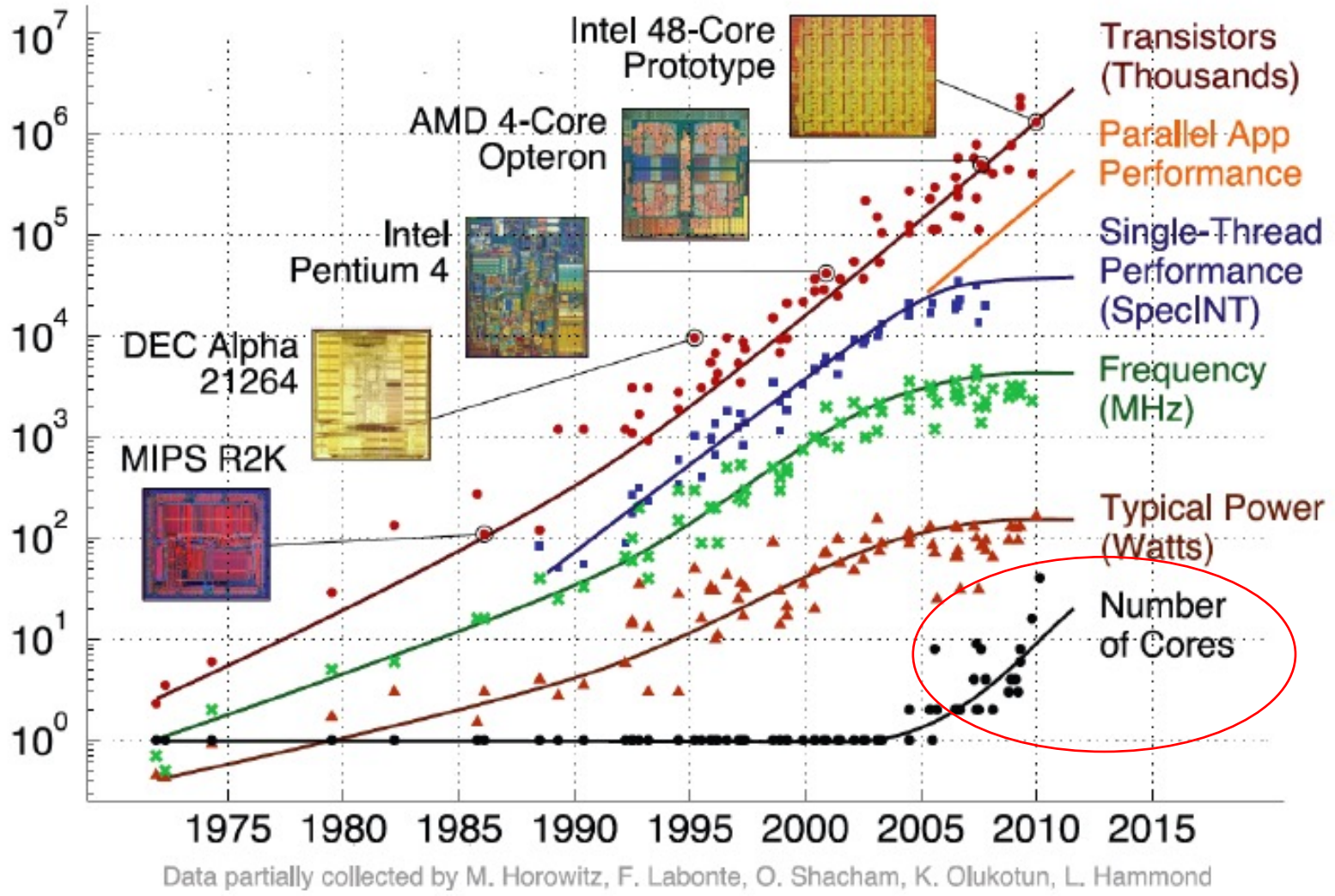
The connecting tissue for Multi-Chip-Modules and Epyc HPC





This work focuses on interconnection networks-on-chip (aka NoCs)

Why do we need Networks-on-Chip?



Power Wall

↓

Many Core

↓

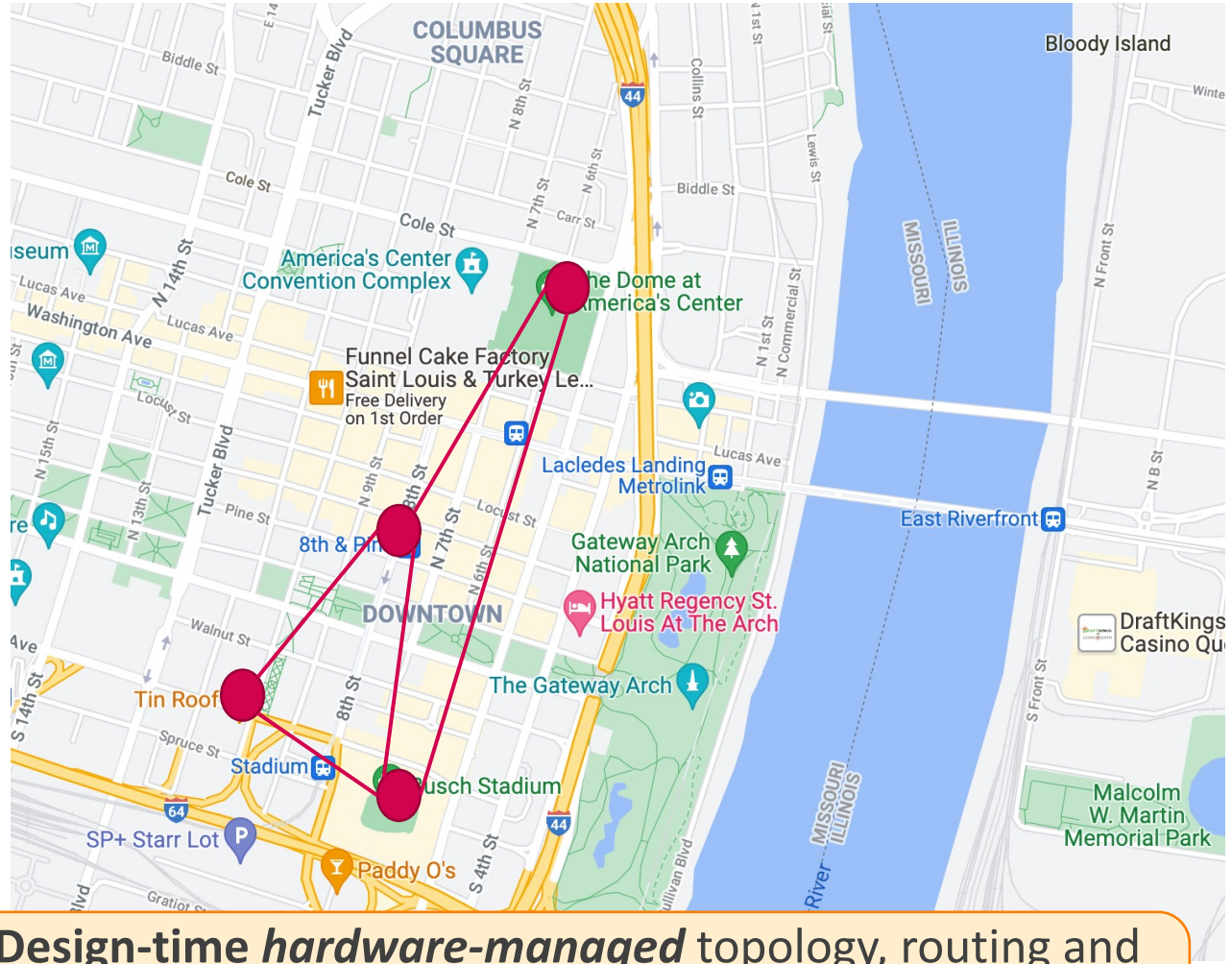
Communication

↓

Network-on-Chip

Network-on-Chip 101

- **Topology**
  - How to connect the nodes
  - *~Road Network*
- **Routing**
  - Which path should a message take
  - *~Series of road segments from source to destination*
- **Flow Control**
  - When does the message have to stop/proceed
  - *~Traffic signals at end of each road segment*
- **Router Microarchitecture**
  - How to build the routers
  - *~Design of traffic intersection (number of lanes, algorithm for turning red/green)*

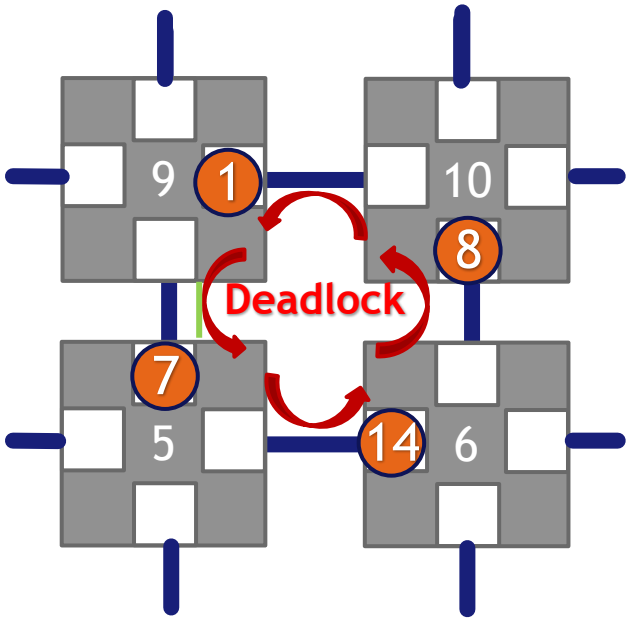


Design-time *hardware-managed* topology, routing and flow-control decisions makes networks-on-chip different from larger scale HPC / datacenter networks

Challenges with NoC Design

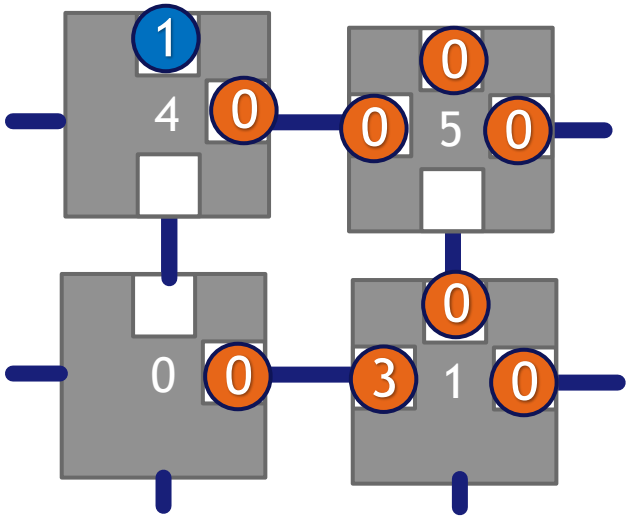
- **Correctness → Guaranteed packet delivery**
  - No Routing Deadlocks
  - No Protocol Deadlocks

→ **Challenge:** Cyclic dependence



- **Performance → Deliver as many packets as fast as possible**
  - Reduce Latency
  - Increase Throughput

→ **Challenge:** Congestion



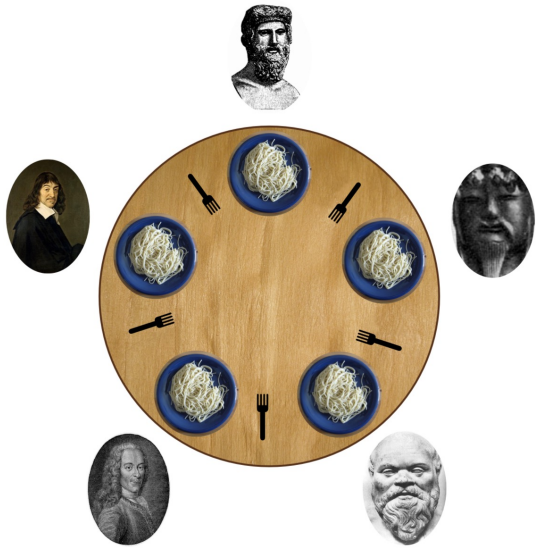
This work addresses both challenges with a unified solution

Outline

- **Background**
  - Networks-on-Chip
  - Deadlock-Freedom Techniques
- **SEEC**
  - Overview
  - Walkthrough
  - Features
  - Optimizations
- **Evaluations**
- **Conclusions**

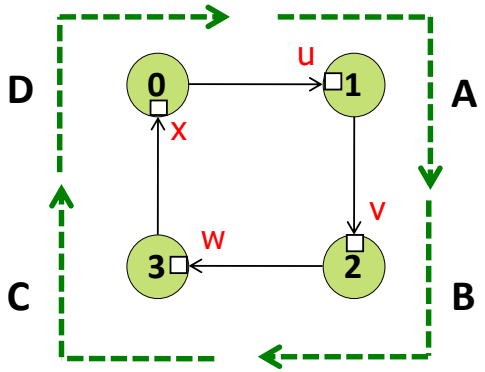
What are deadlocks?

A condition in which a set of **agents** wait indefinitely trying to acquire a set of **resources**



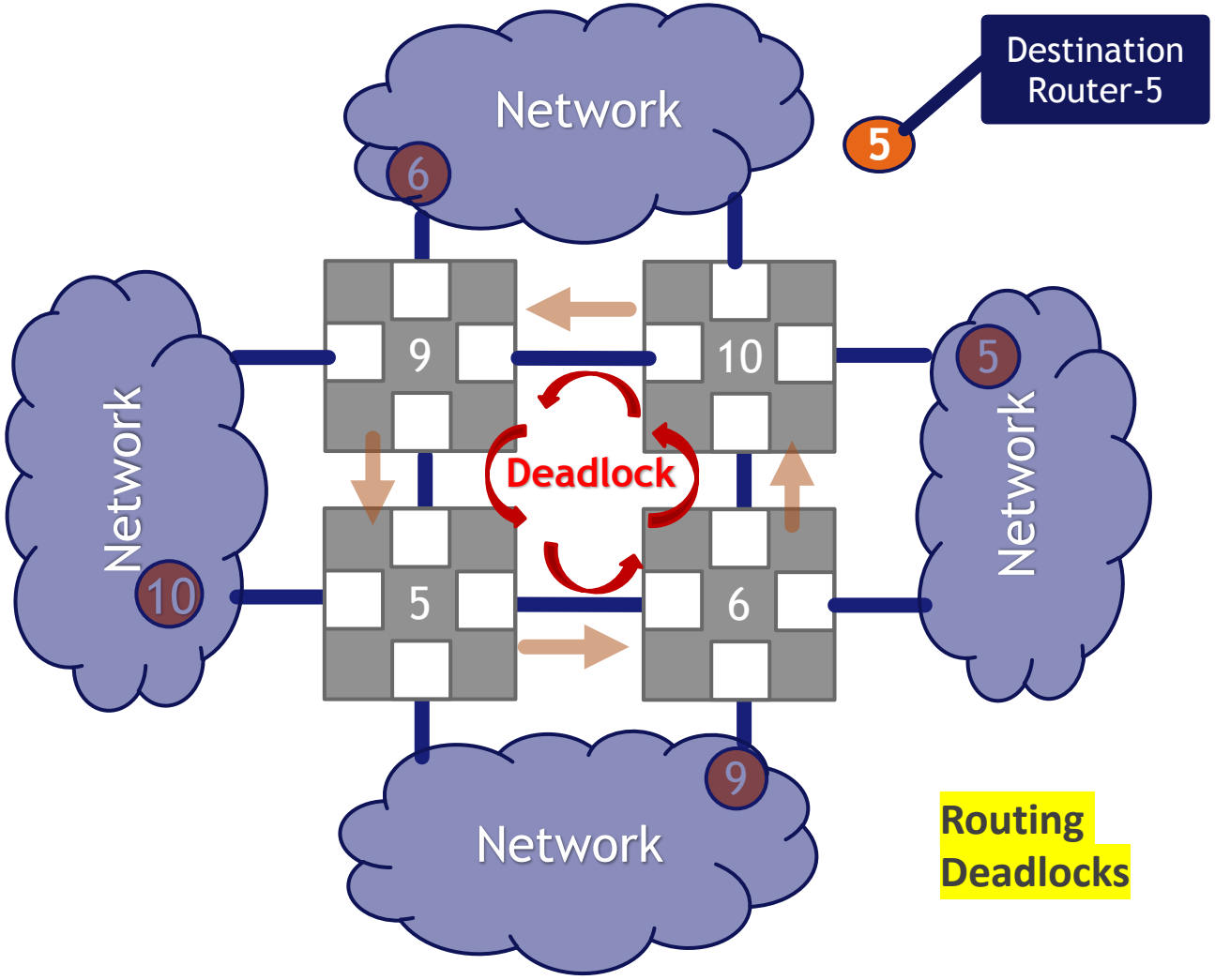
Agents:
Philosophers

Resources:
Forks

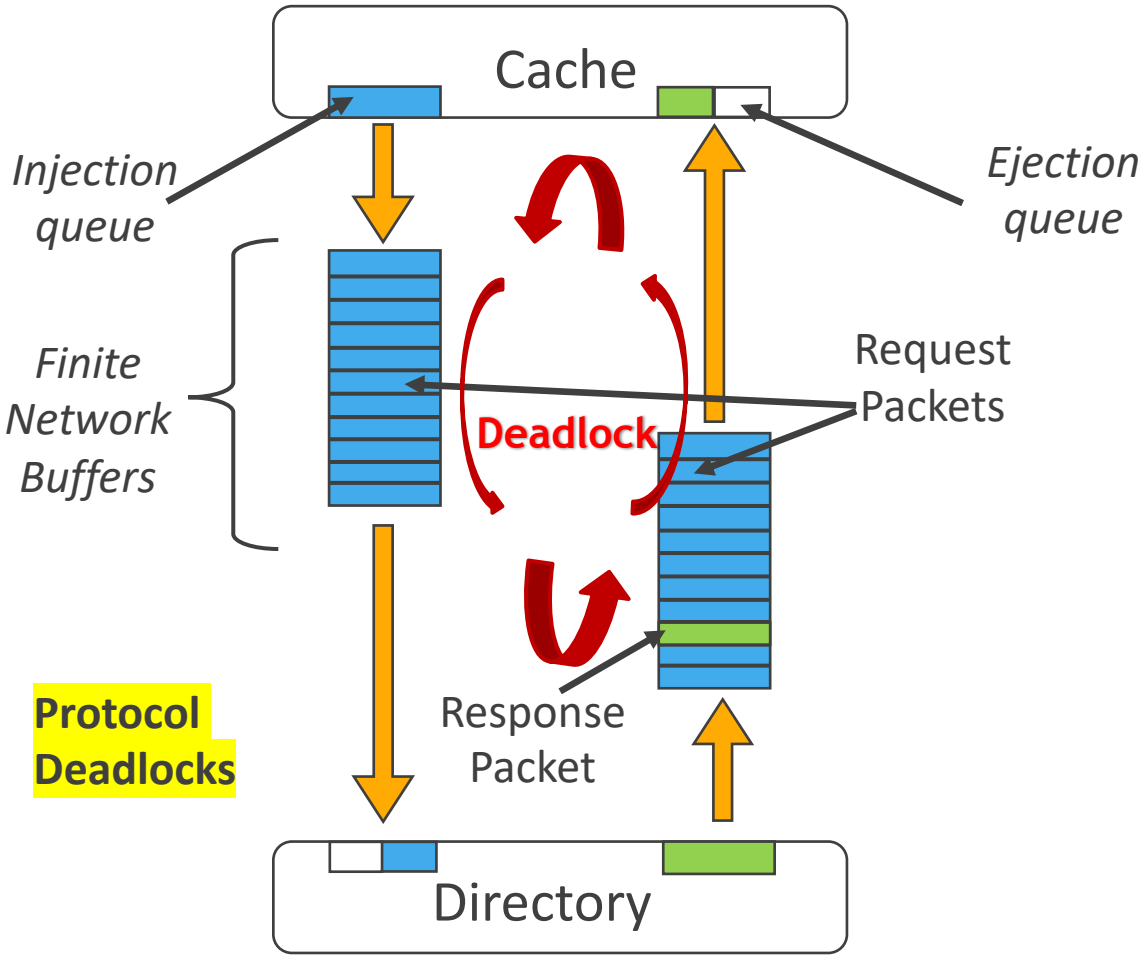


- Packet A holds buffer u (in 1) and wants buffer v (in 2)
- Packet B holds buffer v (in 2) and wants buffer w (in 3)
- Packet C holds buffer w (in 3) and wants buffer x (in 0)
- Packet D holds buffer x (in 0) and wants buffer u (in 1)

Two kinds of deadlocks



Routing Deadlocks



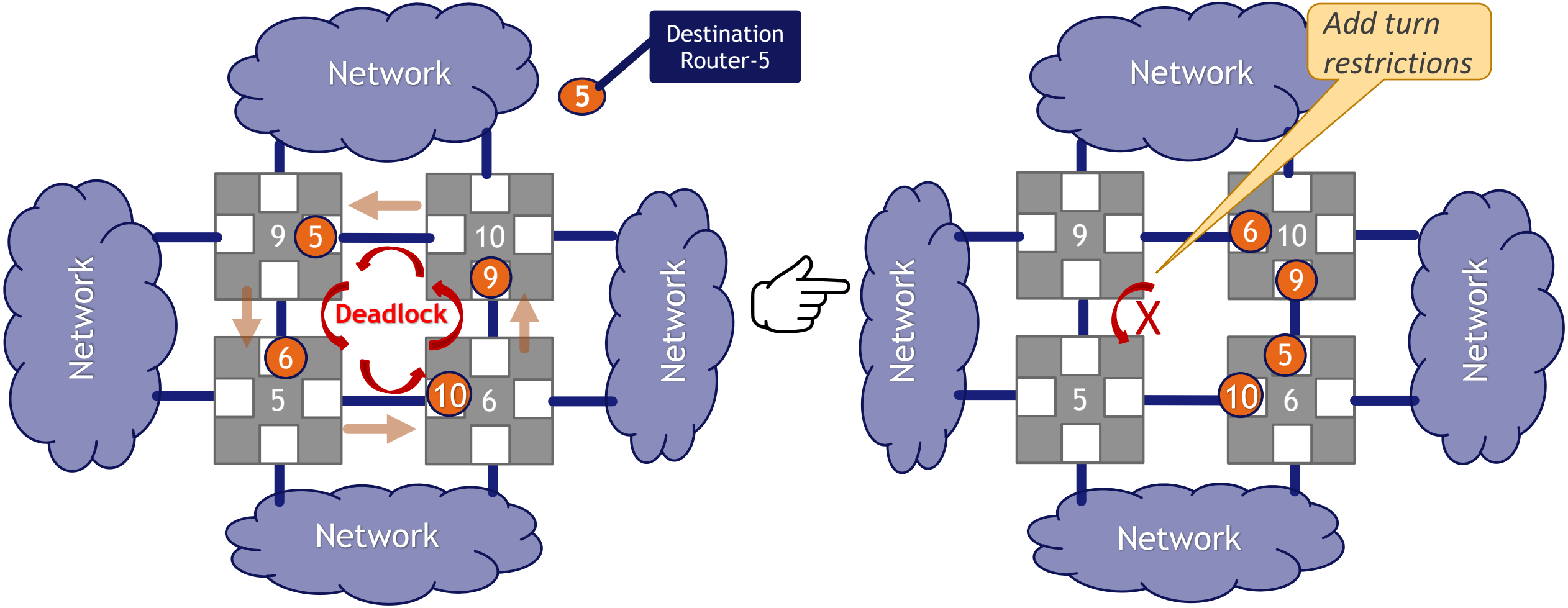
Protocol Deadlocks

Buffers occupied by the request packets renders forward movement of response packet impossible

Current solutions for deadlock-freedom

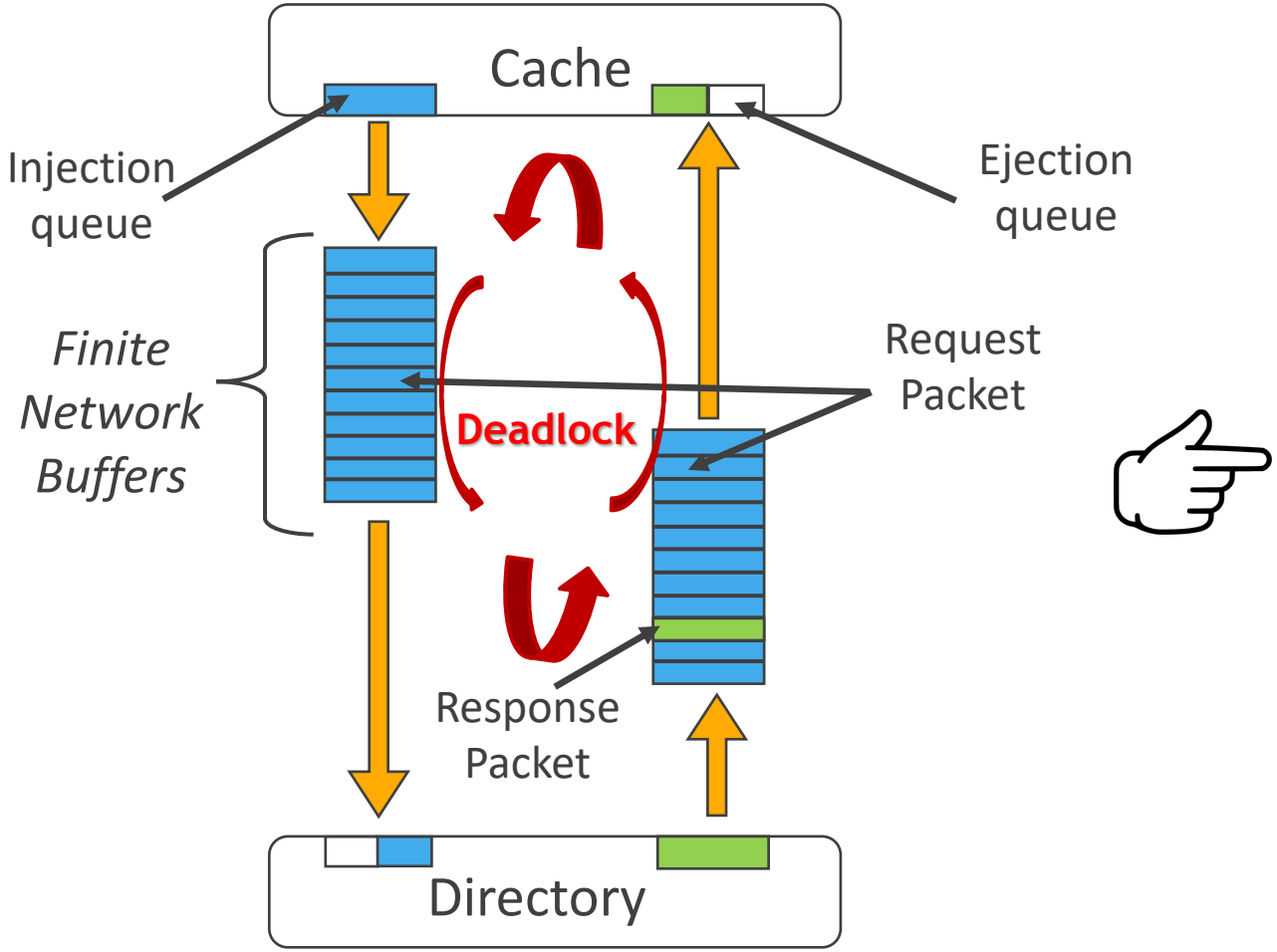
- **Proactive Techniques**
  - Avoid deadlock to begin with → Add turn restrictions or provide additional virtual lanes/channels
- **Reactive Techniques**
  - Allow deadlocks to form in the network
  - Detect deadlocks and recover from them → Add circuitry for monitoring and identifying cyclic dependencies at runtime

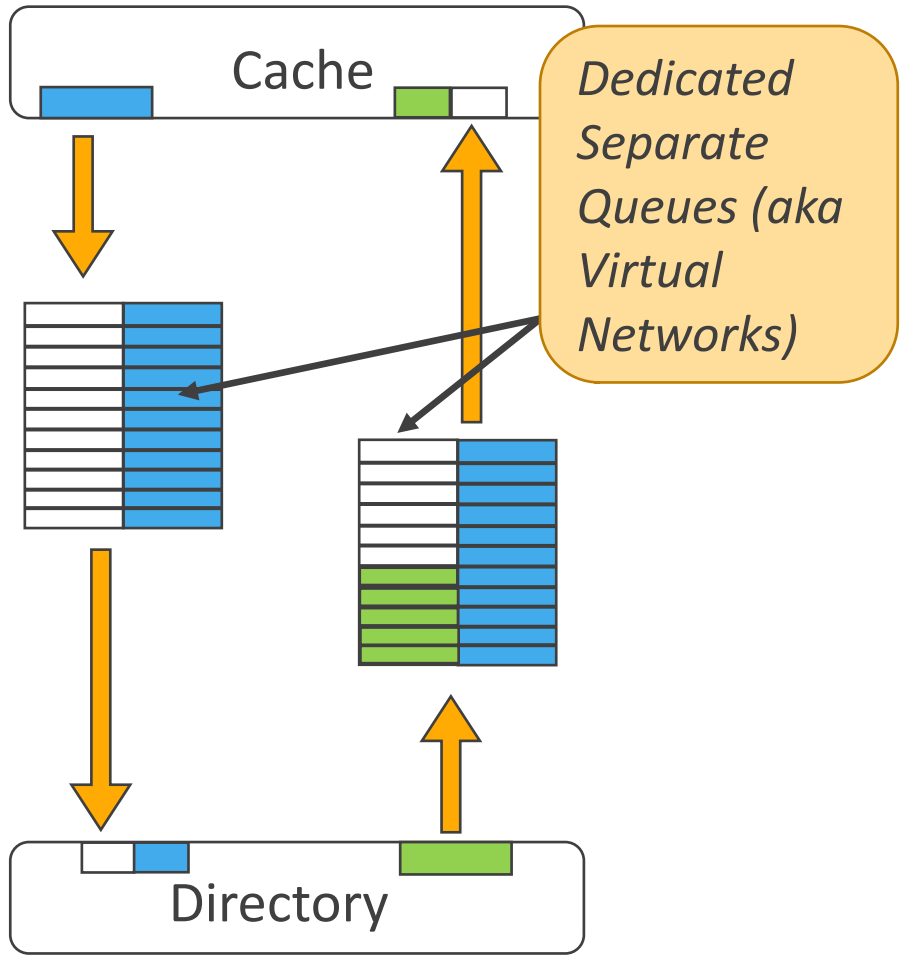
Example of proactive routing deadlock-freedom



Routing Deadlocks

Example of proactive protocol deadlock-freedom





Protocol Deadlocks

Deadlock-Freedom Techniques

Technique	Type of Solution	High Throughput	Low Area Power	Low Hardware Complexity	Routing Deadlock-free	Protocol Deadlock-free
Turn Restriction [1]	Proactive	✗	✓	✓	✓	✗
Escape VC [2]	Proactive	✓	✗	✓	✓	✗
Virtual Network [3]	Proactive	✓	✗	✓	✗	✓
Deadlock Recovery [4]	Reactive	✓	✓	✗	✓	✓

Can we do better?

[1]: Deadlock-free message routing in multiprocessor interconnection networks [IEEE Trans. Computer, 1987]

[2]: A new theory of deadlock-free adaptive routing in wormhole networks. [IEEE Trans. On Parallel and Dist. Sys, 1993]

[3]: The Alpha 21364 network architecture [MICRO 2002]

[4]: DISHA [IPDPS'95] Static Bubble [HPCA'17] SPIN [ISCA'18]

Takeaway:

- Proactive techniques add performance and area overheads.
- Reactive techniques require expensive monitoring circuitry

“Subactive” Techniques

- **Proactive Techniques**
  - Avoid deadlock to begin with
- **Reactive Techniques**
  - Allow deadlock to form in the network
  - Detect deadlocks and recover from them
- **Subactive Techniques** 
  - Allow deadlocks to form in the network
  - Obliviously flush network to remove any deadlock

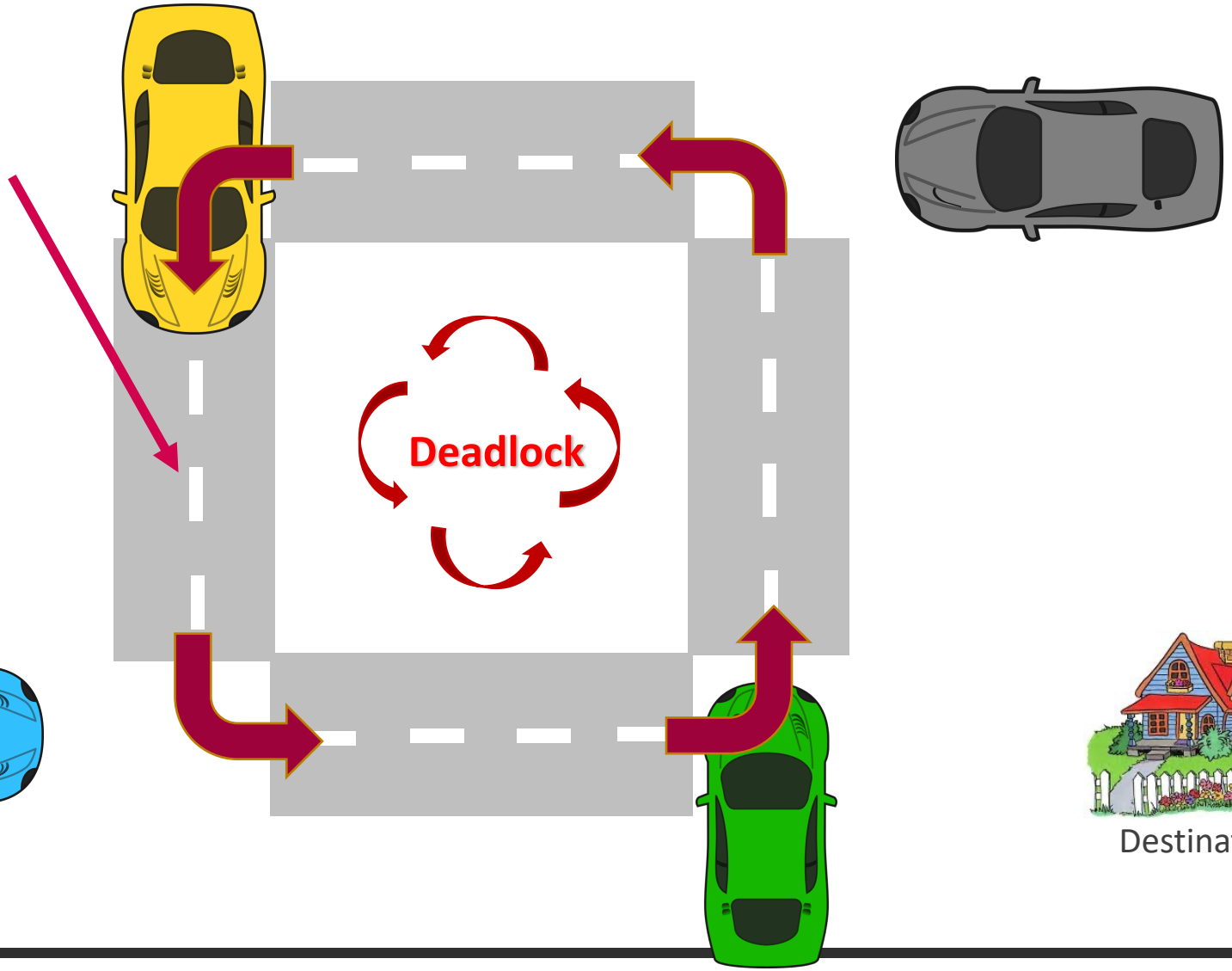
Intuition behind *Subactive*

- **Deadlocks are rare!**
  - Requires certain ensemble of packets with a particular buffer dependency for the deadlock to occur
- **A mechanism to ensure periodic draining of packets *sufficient* to guarantee deadlock-freedom**
  - No need for proactive performance restrictions / additional reserved queues
  - No need for detection and reactive recovery
- **This work (SEEC) is one of many possible subactive solutions**
  - *See our other work: SWAP (MICRO 2019), DRAIN (HPCA 2020), PitStop (HPCA 2021)*

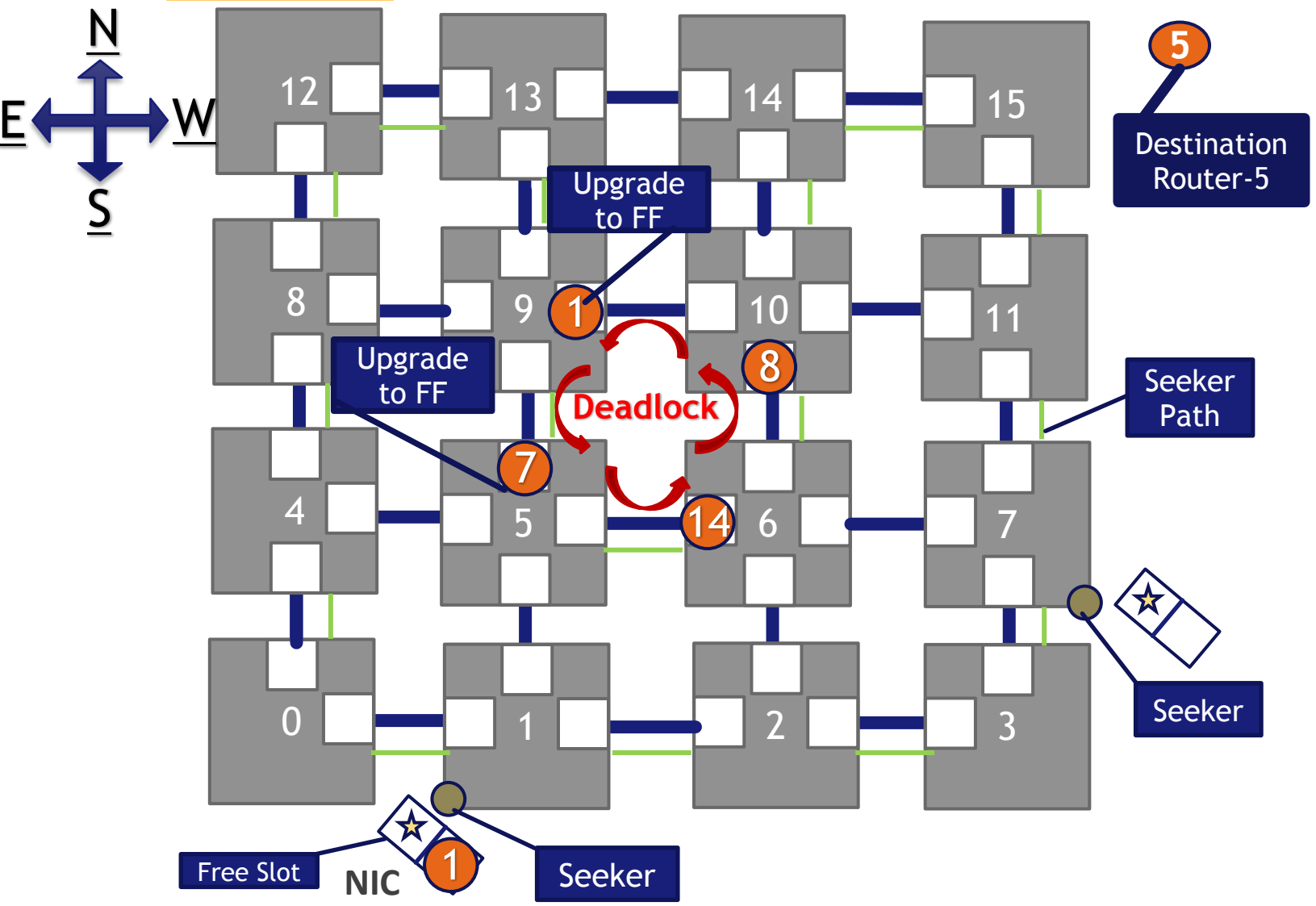
Outline

- **Background**
  - Networks-on-Chip
  - Deadlock-Freedom Techniques
- **SEEC**
  - Overview
  - Walkthrough
  - Features
  - Optimizations
- **Evaluations**
- **Conclusions**

SEEC: Deadlock Analogy



SEEC: Walk-through for Deadlock-freedom



1. Reserve a free ejection buffer at the NIC
2. Send a ***seeker*** over a sideband ***seeker path*** through all routers
3. If a packet intended for this NIC is found, upgrade it for ***Free Flow (FF)***
4. FF packet given priority over regular packets to traverse all the way to the N
5. Repeat for all NICs in round-robin manner

Outline

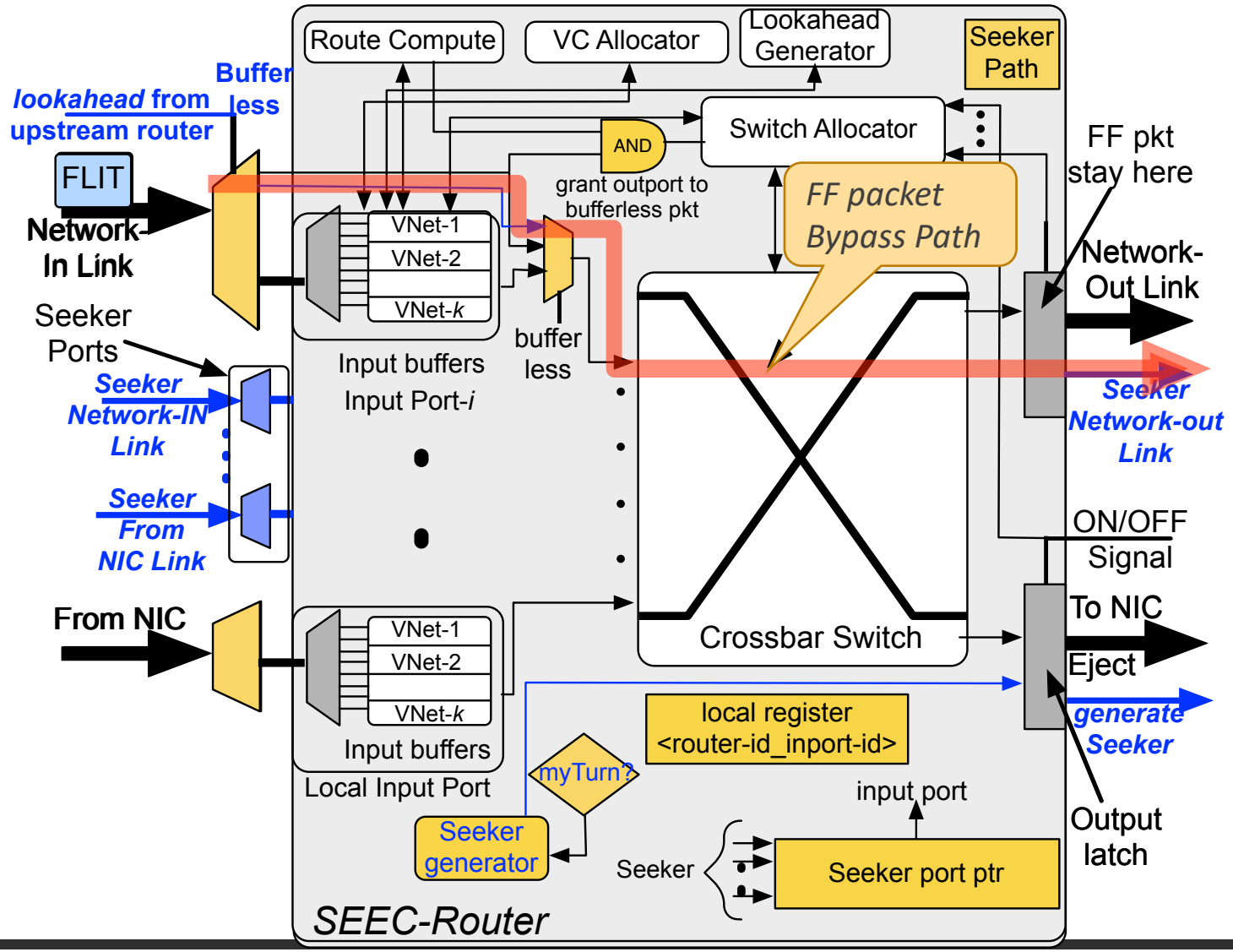
- **Background**
  - Networks-on-Chip
  - Deadlock-Freedom Techniques
- **SEEC**
  - Overview
  - Walkthrough
  - Features
  - Optimizations
- **Evaluations**
- **Conclusions**

Features of SEEC

- **Novel Flow Control: Free Flow (FF)**
  - Packets upgraded to FF via seeker
  - FF packet gets prioritized over the buffered packets
    - FF packet sends a *lookahead* signal on cycle in advance
    - *lookahead* signal sets the appropriate mux at downstream router
    - FF packet traverse the network from link to link till the NIC
  - FF packets follow ***minimal (shortest) express path*** with no buffering
  - SEEC proposes co-existence of FF packets and buffered packets

- **Seeker Path**
  - Any pre-defined path that visits all routers
  - Separate sideband network
    - could use regular links by prioritizing seeker over regular packets
  - Possible to have multiple parallel seekers → later in talk

Router Microarchitecture



Deadlock Freedom Proof

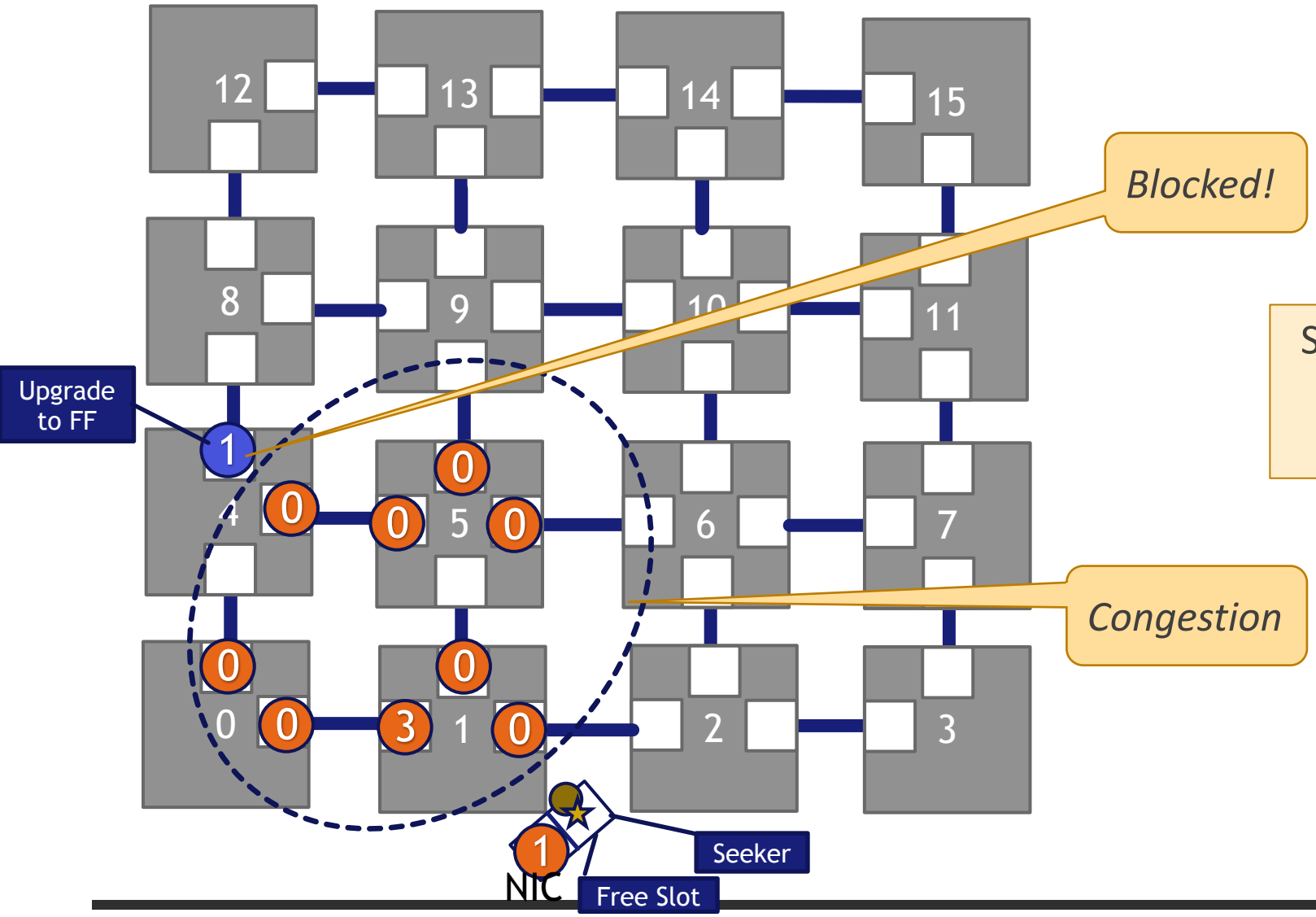
- The ejection port at NIC has separate Virtual Networks (one per protocol message class)
- The response packet can cause a protocol deadlock, if stuck in network
- SEEC is ***protocol deadlock free*** as NIC will eject any blocked response packet by sending seeker
- SEEC is ***routing deadlock free*** as all the input ports get chance to send packets via FF flow control
- SEEC is ***livelock free*** as there is no misrouting

Mathematical upper bound for deadlock persistence

- For any irregular network topology:
  - Assume diameter is *D*
  - Assume it takes on an average ‘k’ cycles at the ejection port
  - Number of nodes in the network is ‘N’
  - Let the average network router-radix of the router is ‘r’

- The maximum time it takes for a deadlock to persist
  - Deadlock involves at least 4 routers
  - [{r*(N-4)*2*D*k} + (D+k)] cycles

SEEC: Additional Benefits → Performance



SEEC helps packets to bypass congestion in the network, improving performance

Outline

- **Background**
  - Networks-on-Chip
  - Deadlock-Freedom Techniques
- **SEEC**
  - Overview
  - Walkthrough
  - Features
  - Optimizations
- **Evaluations**
- **Conclusions**

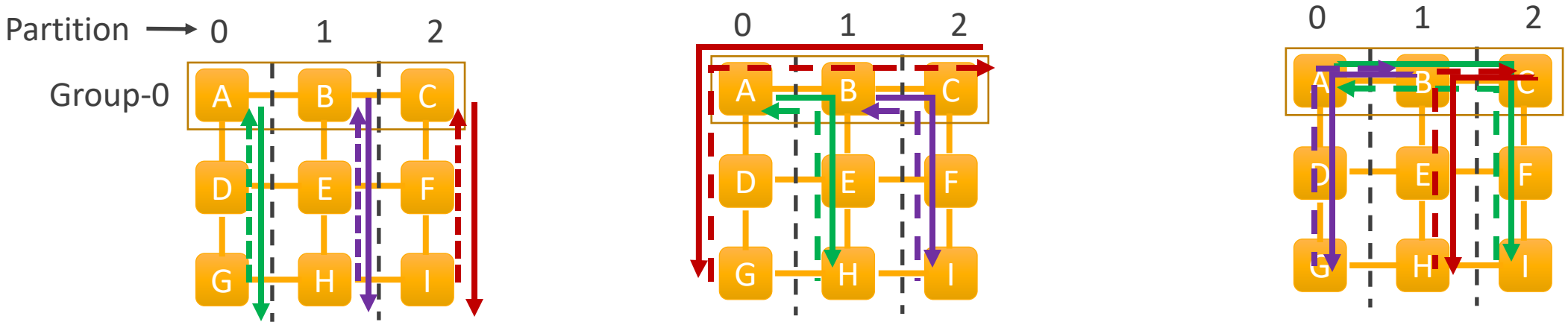
Multi-SEEC (mSEEC)

- **Motivation**
  - Path Diversity in many topologies: Multiple ways of reaching a node
- **Features of mSEEC**
  - Creates multiple FF packets that can traverse simultaneously
- **Requirement**
  - Multiple FF packets traverse minimally without overlapping paths
- **Implementation**
  - mSEEC divides the topology into independent regions

mSEEC: Walk Through

Seeker Packet

Phase 0

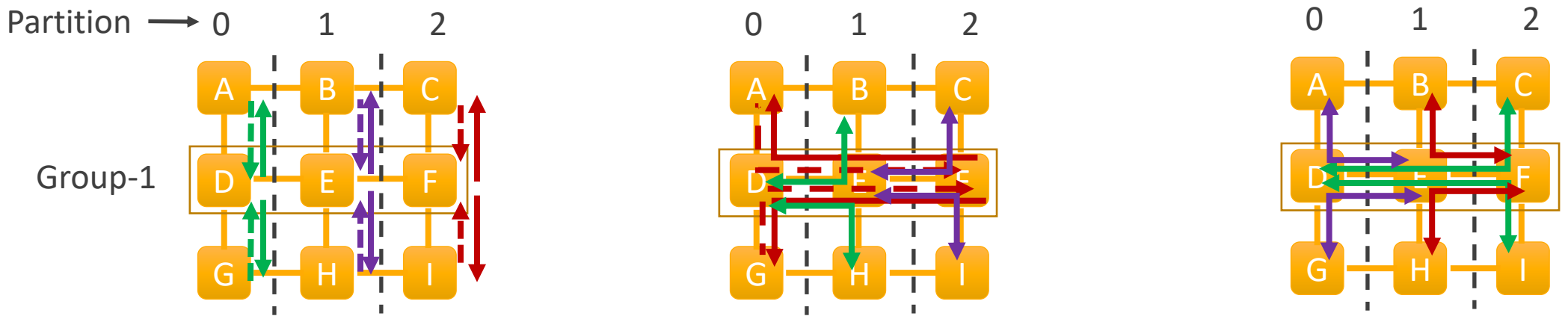


- When Phase-0 finishes:
  - each node in group-0 has searched entire topology
- Phase-1 starts

mSEEC: Walk Through

Seeker Packet

Phase 1



- When Phase-1 finishes:
  - each node in group-1 has searched entire topology
- Phase-2 starts and so on…

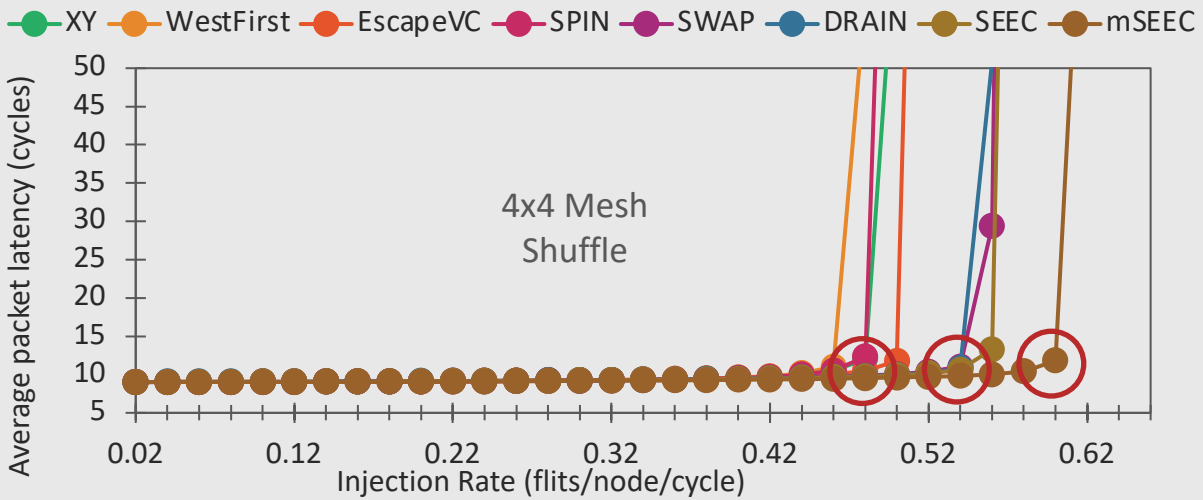
Outline

- **Background**
  - Networks-on-Chip
  - Deadlock-Freedom Techniques
- **SEEC**
  - Overview
  - Walkthrough
  - Features
  - Optimizations
- **Evaluations**
- Conclusions

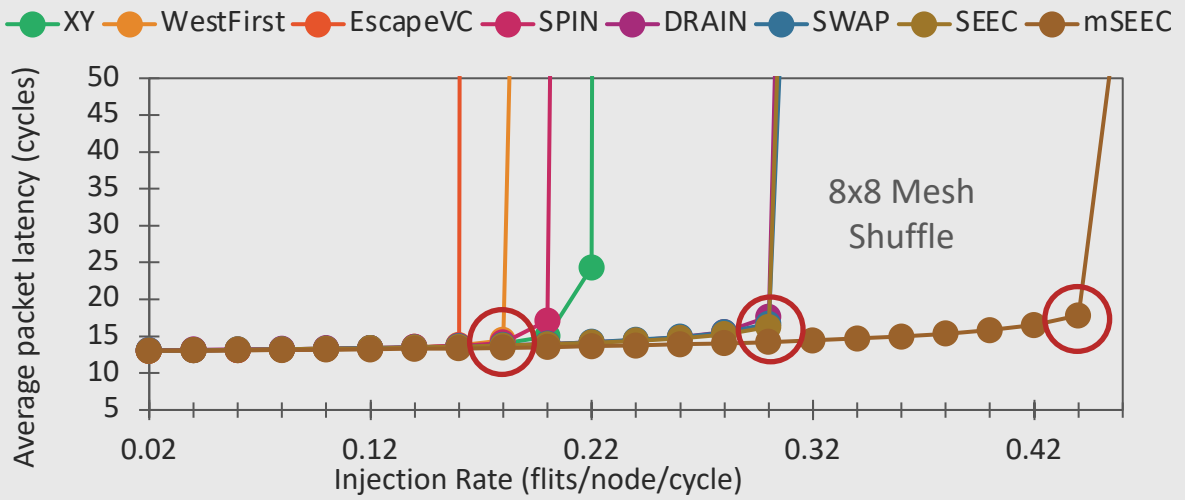
Evaluations Methodology

- gem5 for full system and synthetic simulations
- Garnet2.0 for network simulation
- SEEC/mSEEC uses minimal adaptive routing algorithm with full path diversity
- Virtual Cut Through Buffer Management
- Evaluated on 4x4 (16 nodes), 8x8 (64 nodes), and 16x16 (256 nodes) Mesh topology

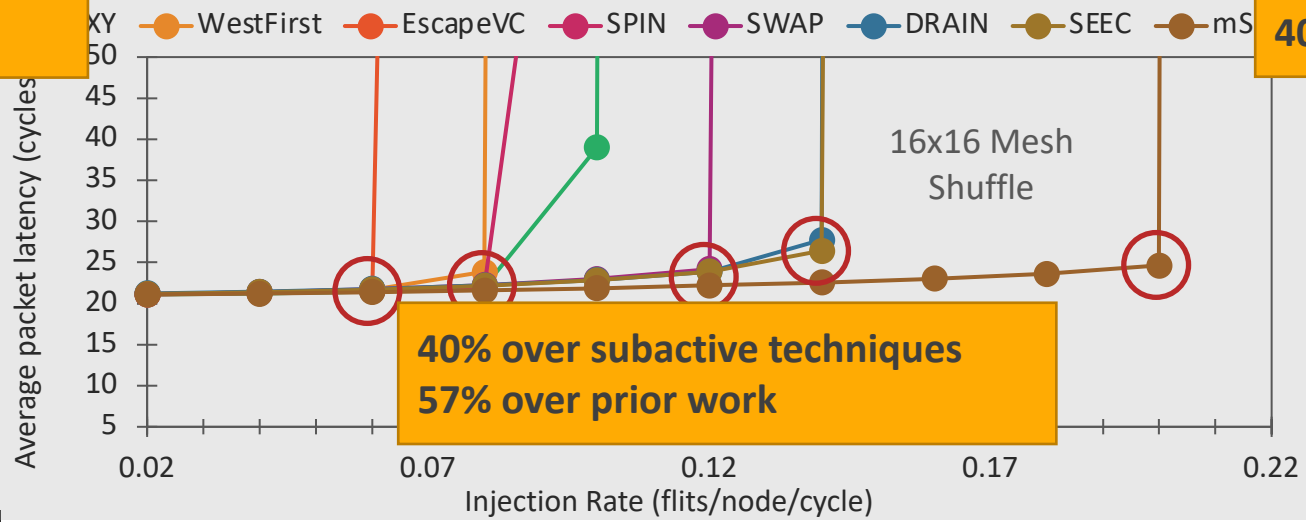
Results: Performance Sweep



11% over subactive techniques
25% over prior work

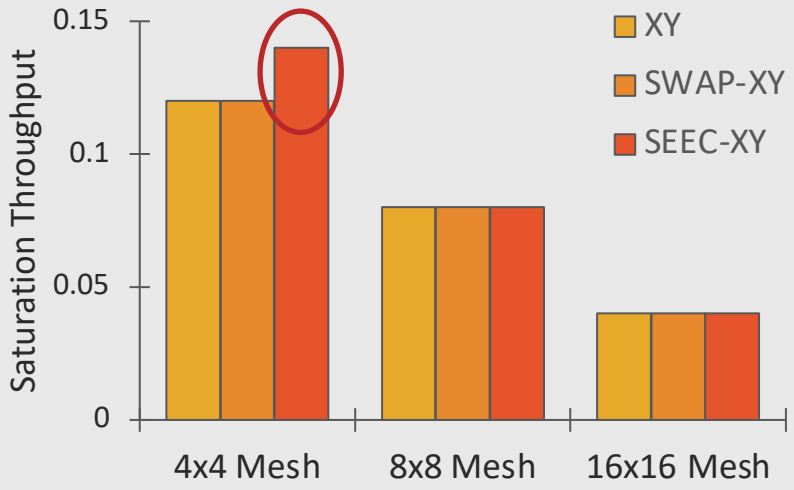


30% over subactive techniques
40% over prior work

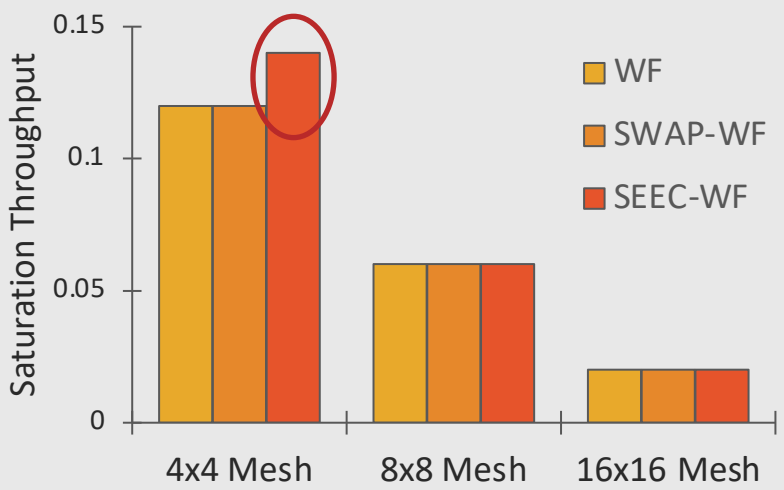


40% over subactive techniques
57% over prior work

Results: Throughput Enhancement



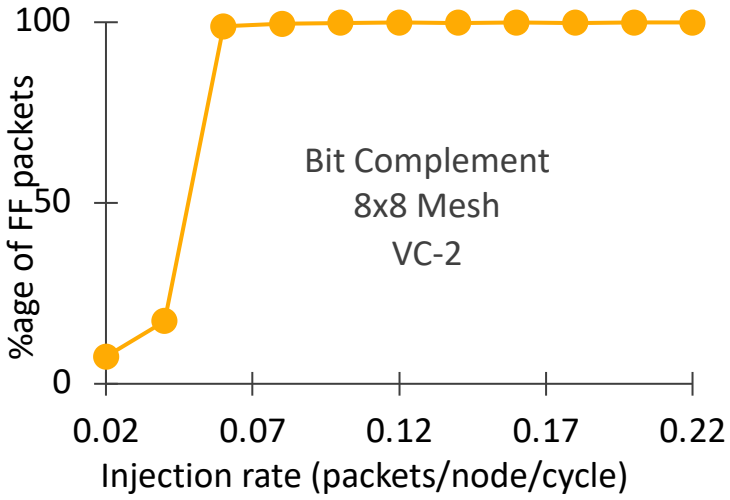
Uniform Random
XY Routing

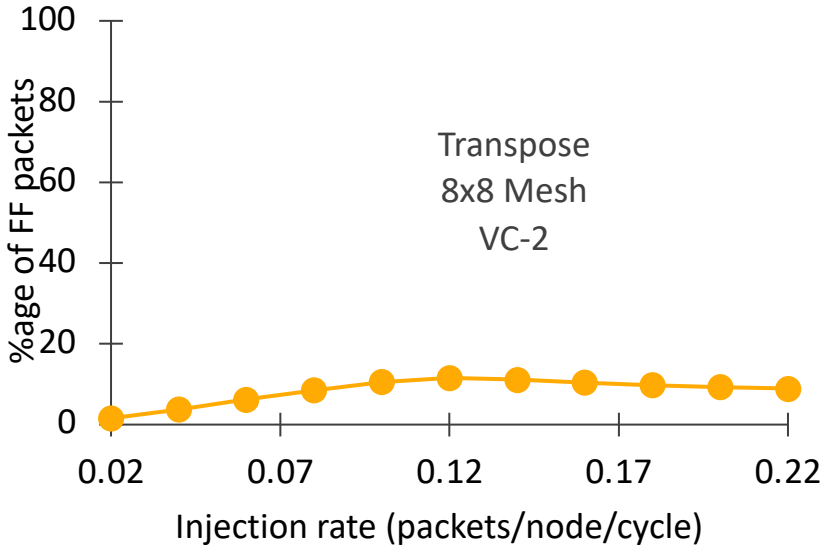


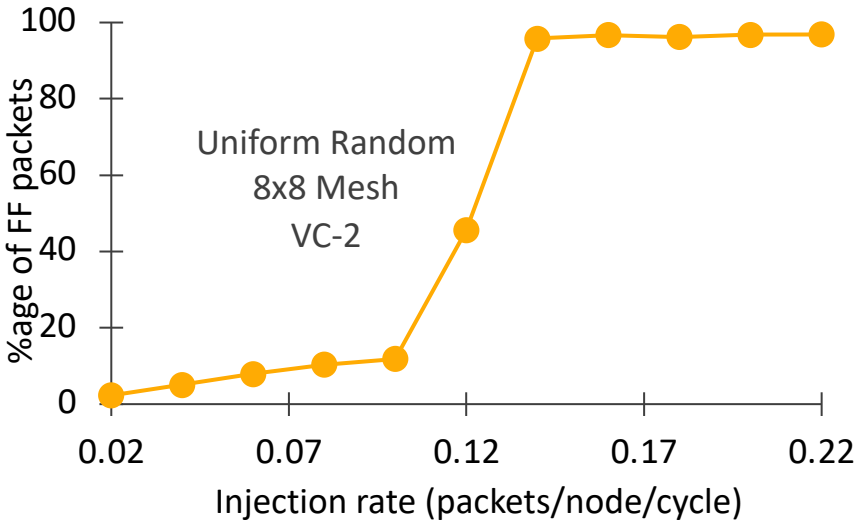
Uniform Random
West First Routing

16% throughput improvement in 4x4 Mesh

Results: SEEC-%age of FF packets

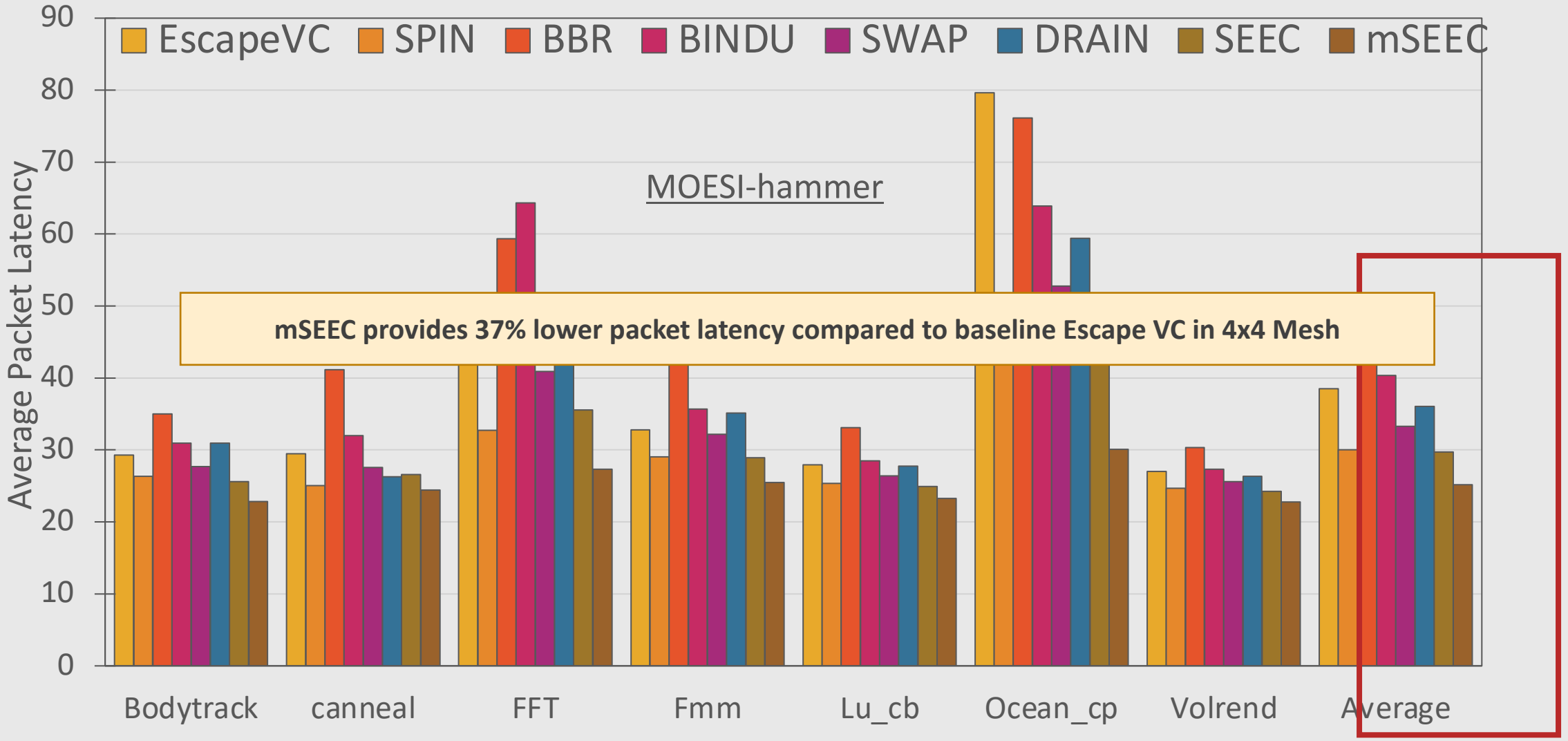




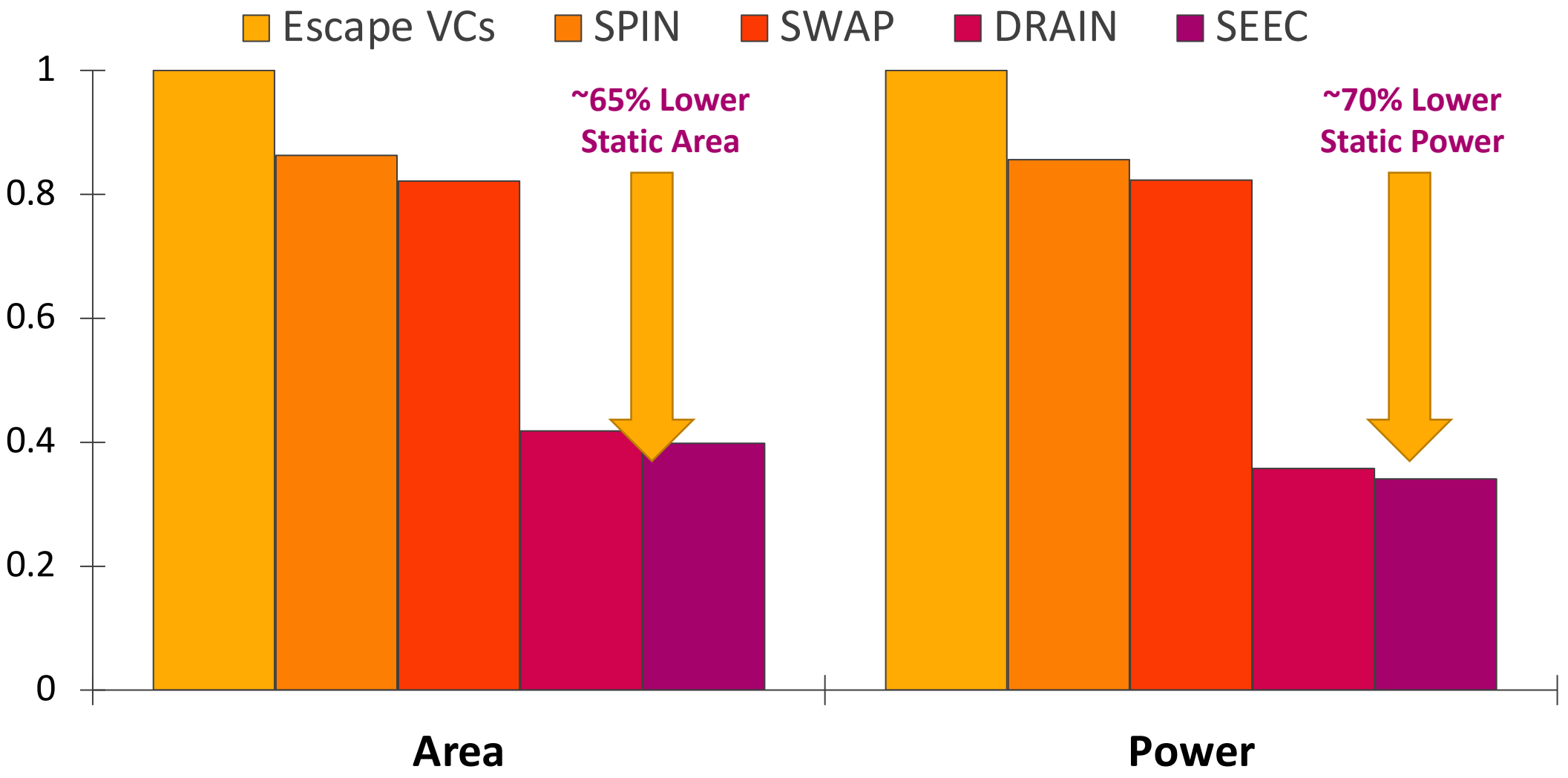


Post saturation heavy reliance on FF for packet delivery

Results: Full System Simulation



Normalized Area and Static Power



Outline

- **Background**
  - Networks-on-Chip
  - Deadlock-Freedom Techniques
- **SEEC**
  - Overview
  - Walkthrough
  - Features
  - Optimizations
- **Evaluations**
- **Conclusions**

Conclusion

- Deadlock is a correctness problem which every network must solve
- Current solutions either require complex deadlock detection or provision extra set of buffers or restrict turns
- SEEC provides a novel way of resolving deadlock by periodically moving packets for few hops along a ring constructed over the network.
  - No extra buffer needed
  - Free from complicated deadlock detection circuitry
  - Provides full path-diversity therefore high performant

Revisiting Qualitative Comparison

Codebase available here:
https://github.com/noc-deadlock/seec

Teechnique	Type of Solution	High Throughput	Low Area Power	Low Hardware Complexity	Routing Deadlock-free	Protocol Deadlock-free
Turn Restriction	Proactive	✗	✓	✓	✓	✗
Escape VC	Proactive	✓	✗	✓	✓	✗
Virtual Network	Proactive	✓	✗	✓	✗	✓
Deadlock Recovery	Reactive	✓	✓	✗	✓	✓

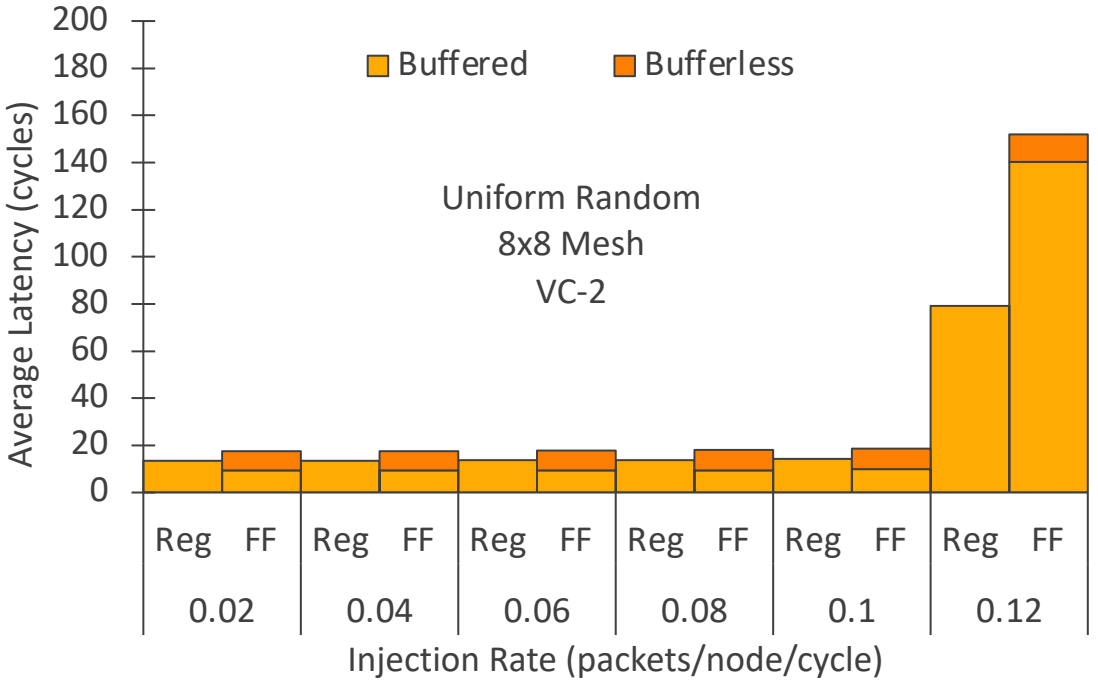
Takeaways

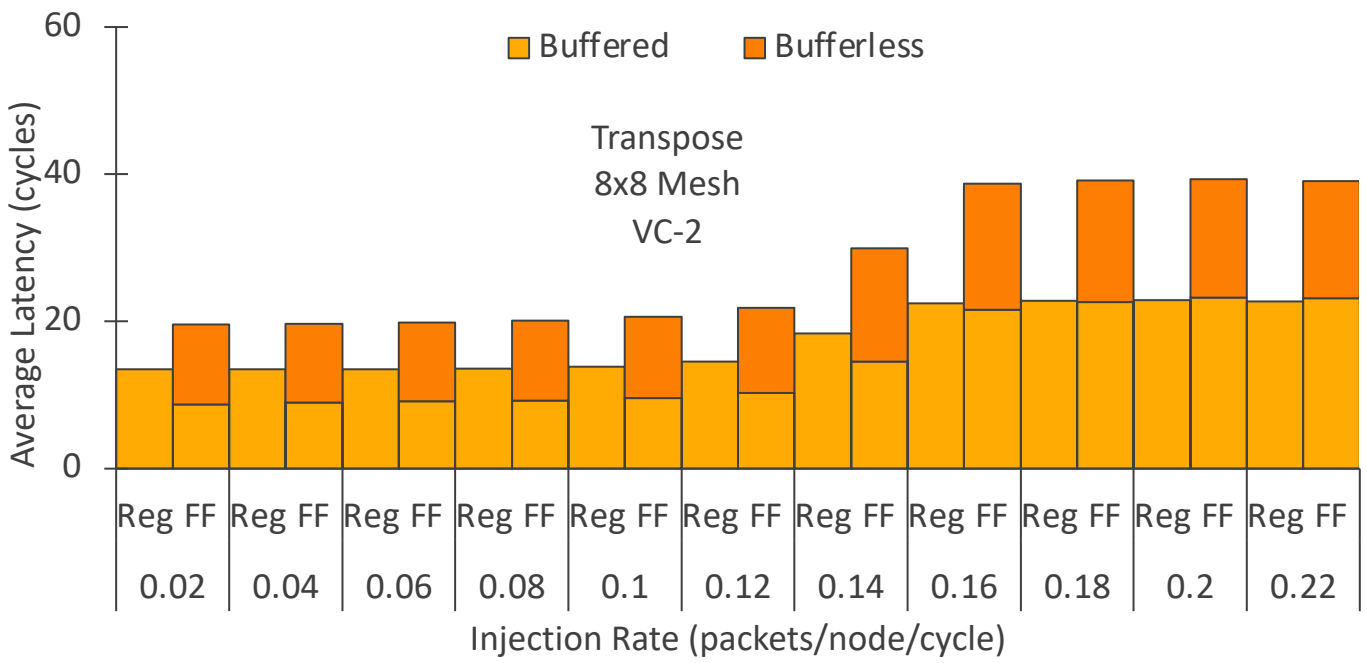
- SEEC obliviously provides deadlock freedom
- SEEC is completely decentralized
- SEEC has low hardware overhead over baseline router

Thanks!

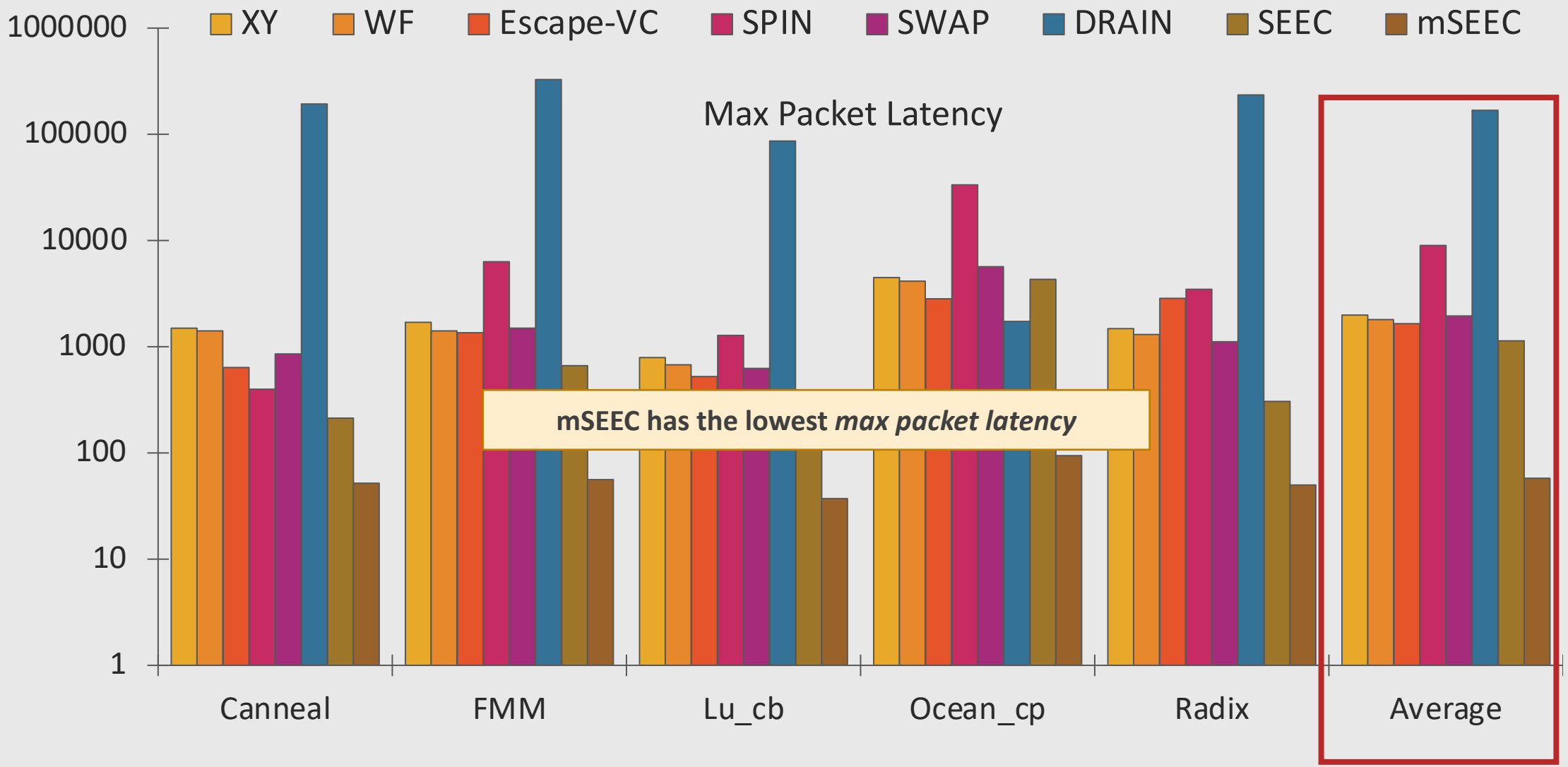
Backup

Results: SEEC-Network Latency breakdown

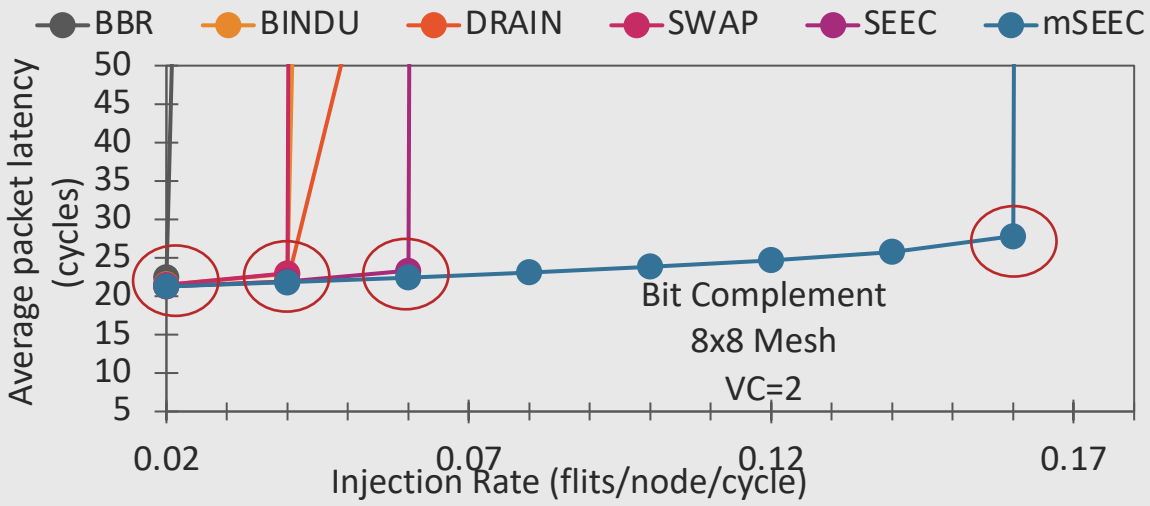


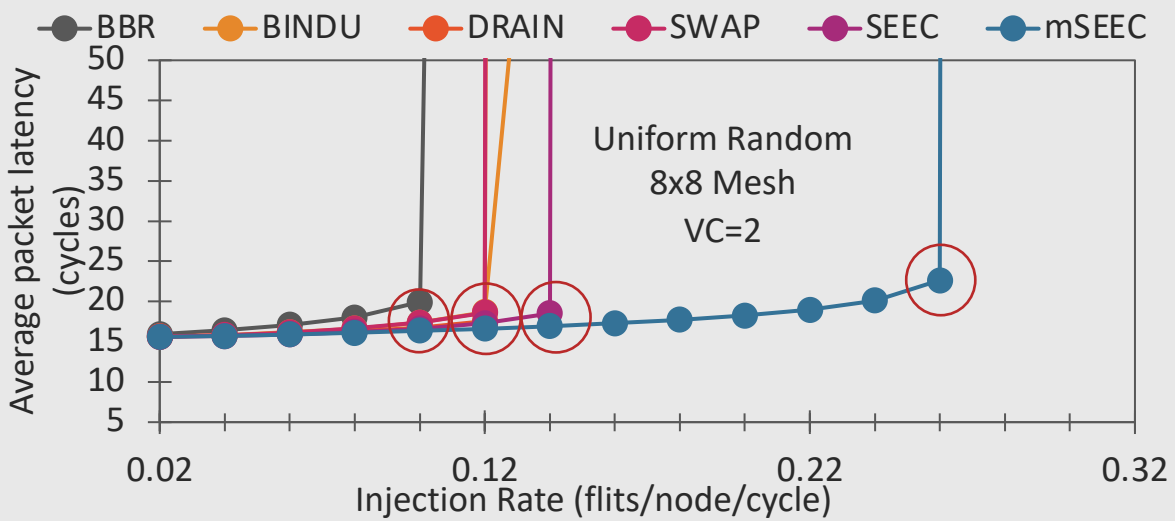


Result: Max Packet Latency



Results: Subactive Techniques





We observe following performance order (subactive techniques only) :

BBR < BINDU ≈ DRAIN ≈ SWAP < SEEC < mSEEC