

# MinBD: Minimally-Buffered Deflection Routing for Energy-Efficient Interconnect

**Chris Fallin**, Greg Nazario, Xiangyao Yu\*,  
Kevin Chang, Rachata Ausavarungnirun, Onur Mutlu

Carnegie Mellon University

\*CMU and Tsinghua University

**SAFARI** Carnegie Mellon University

# Motivation

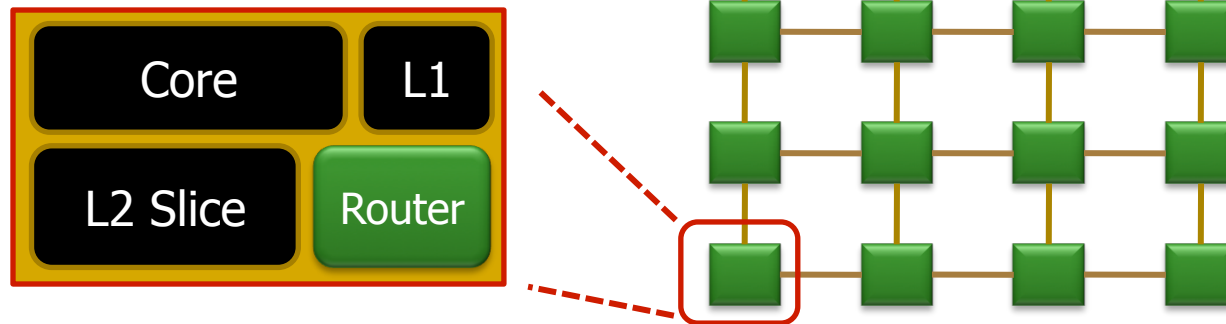
---

- In many-core chips, on-chip interconnect (NoC) consumes **significant power**

**Intel Terascale:** ~28% of chip power

**Intel SCC:** ~10%

**MIT RAW:** ~36%



- Recent work<sup>1</sup> uses **bufferless deflection routing** to reduce power and die area

---

<sup>1</sup>Moscibroda and Mutlu, "A Case for Bufferless Deflection Routing in On-Chip Networks." ISCA 2009.

# Bufferless Deflection Routing

---

- **Key idea:** Packets are never buffered in the network. When two packets contend for the same link, one is **deflected**.
  - Removing **buffers** yields significant benefits
    - Reduces **power** (CHIPPER: reduces NoC power by 55%)
    - Reduces **die area** (CHIPPER: reduces NoC area by 36%)
  - But, at **high network utilization** (load), bufferless deflection routing causes **unnecessary link & router traversals**
    - Reduces **network throughput** and application performance
    - Increases **dynamic power**
  - **Goal:** Improve **high-load performance** of low-cost deflection networks by reducing the deflection rate.
-

# Outline: This Talk

---

- **Motivation**
- **Background:** Bufferless Deflection Routing
- **MinBD:** Reducing Deflections
  - Addressing Link Contention
  - Addressing the Ejection Bottleneck
  - Improving Deflection Arbitration
- **Results**
- **Conclusions**

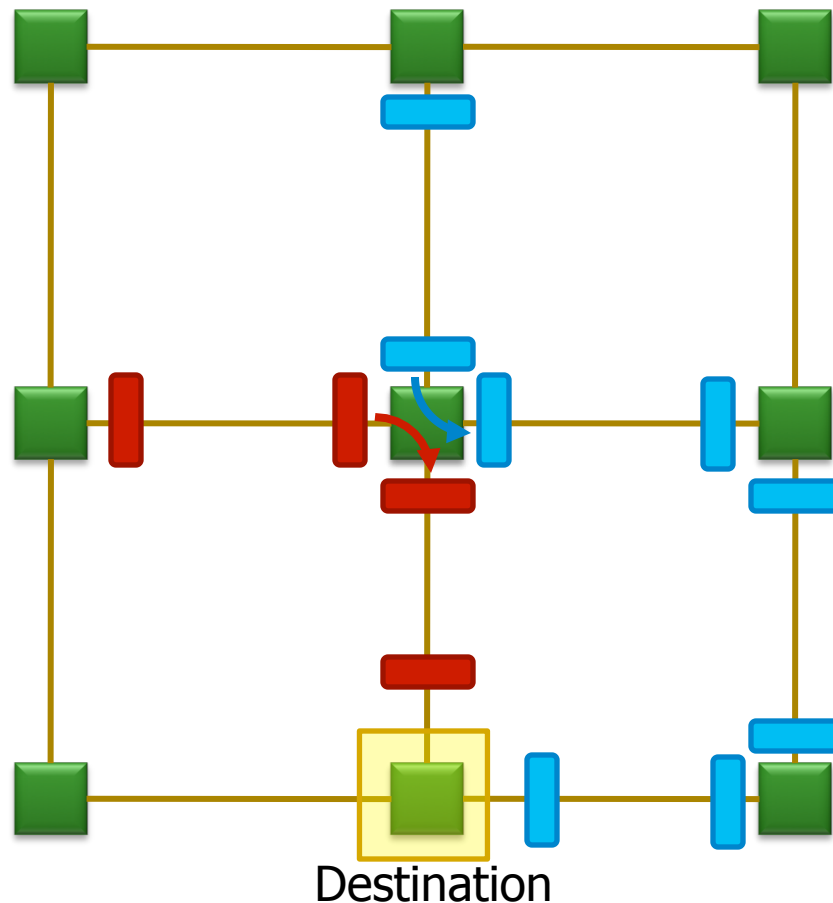
# Outline: This Talk

---

- Motivation
- **Background:** Bufferless Deflection Routing
- **MinBD:** Reducing Deflections
  - Addressing Link Contention
  - Addressing the Ejection Bottleneck
  - Improving Deflection Arbitration
- Results
- Conclusions

# Bufferless Deflection Routing

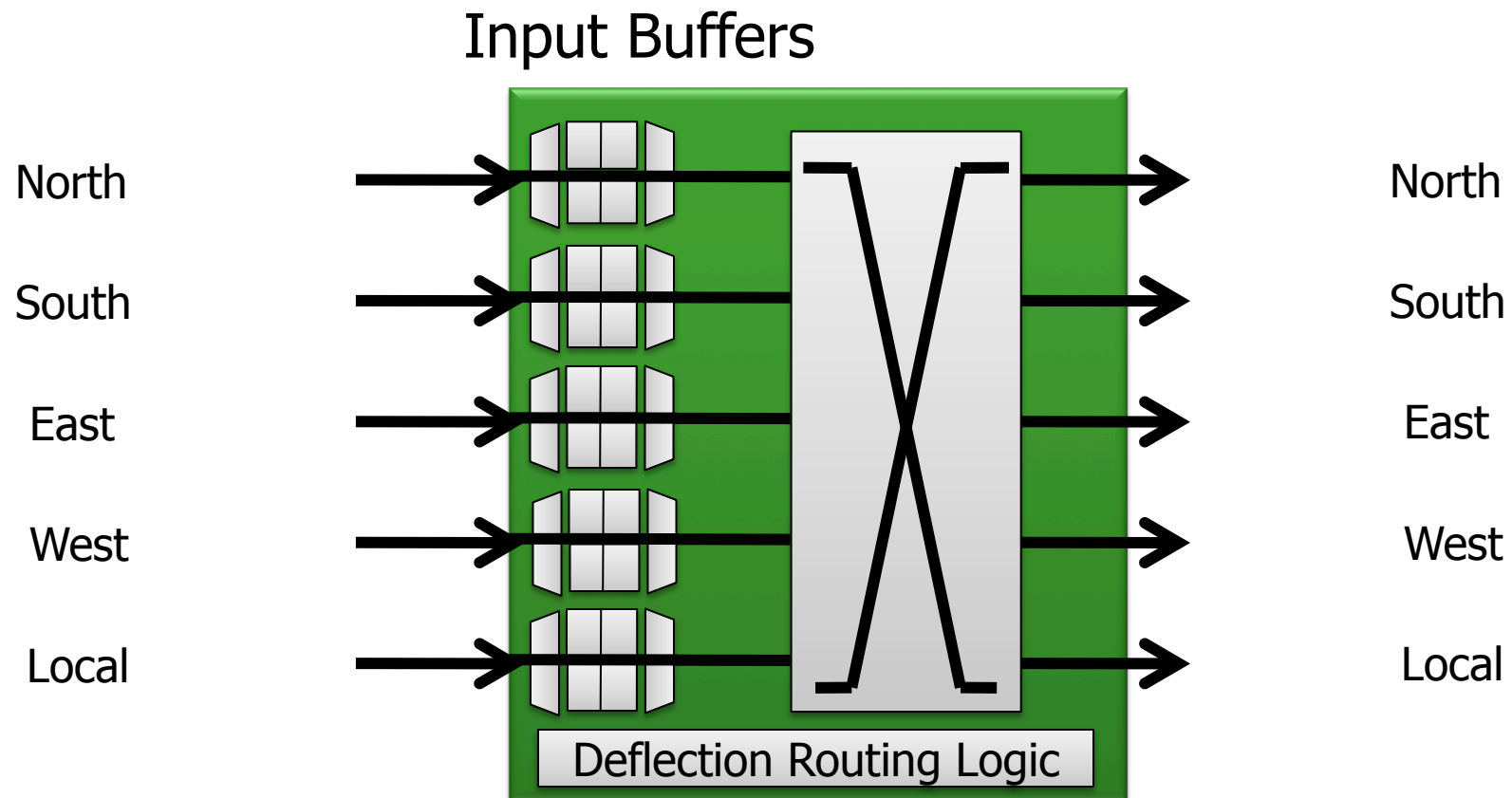
- **Key idea:** Packets are never buffered in the network. When two packets contend for the same link, one is **deflected**.<sup>1</sup>



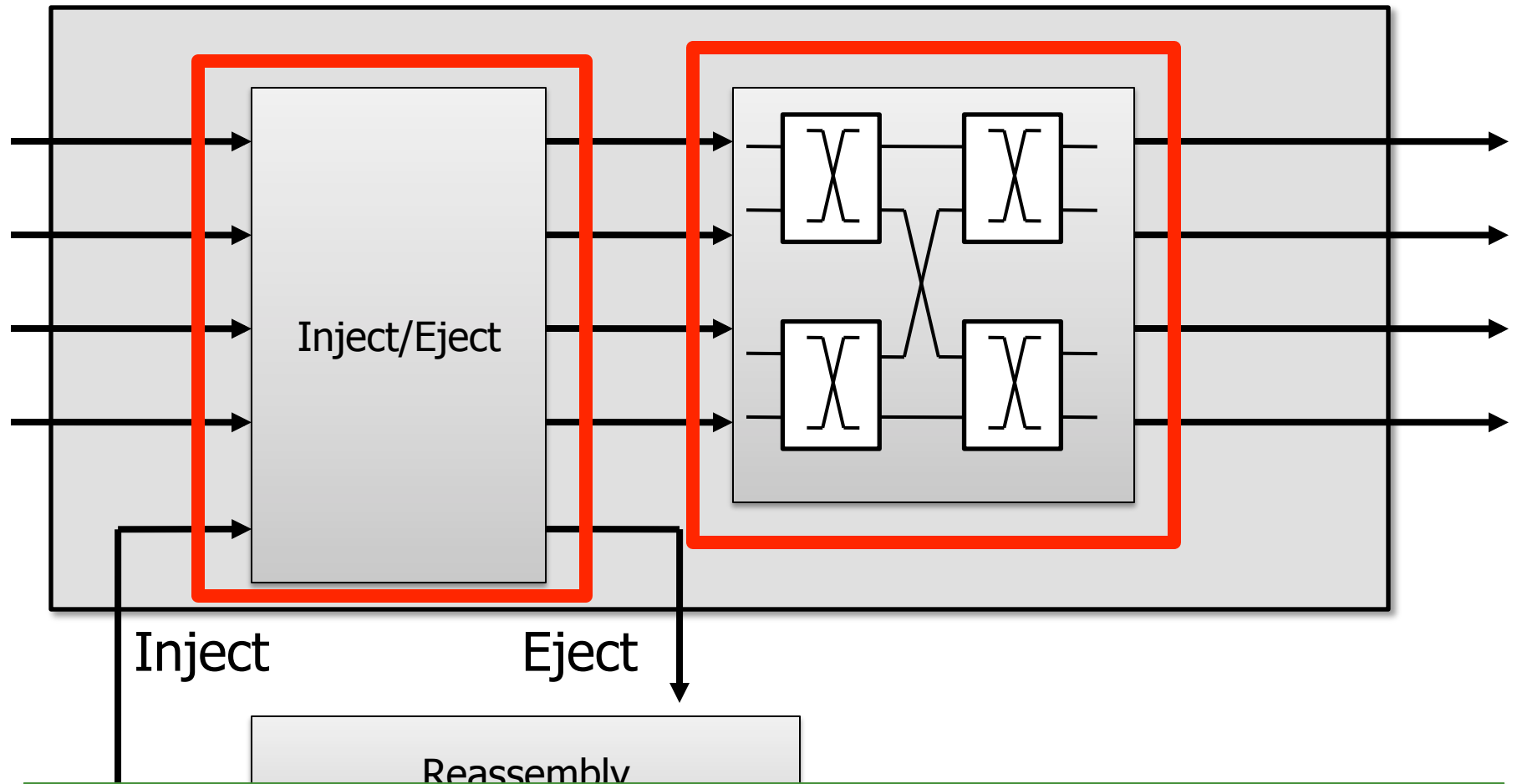
<sup>1</sup>Baran, "On Distributed Communication Networks." RAND Tech. Report., 1962 / IEEE Trans.Comm., 1964. 6

# Bufferless Deflection Routing

- Input buffers are eliminated: flits are buffered in **pipeline latches** and on **network links**



# Deflection Router Microarchitecture



## Stage 2: Deflection arbitration

Fallin et al., "CHIPPER: A Low-complexity Bufferless Deflection Router", HPCA 2011.



# Issues in Bufferless Deflection Routing

---

- **Correctness:** Deliver all packets without **livelock**
  - **CHIPPER<sup>1</sup>: Golden Packet**
  - **Globally prioritize one packet** until delivered
- **Correctness:** Reassemble packets without **deadlock**
  - **CHIPPER<sup>1</sup>: Retransmit-Once**
- **Performance:** Avoid performance degradation at **high load**
  - **MinBD**

---

<sup>1</sup> Fallin et al., "CHIPPER: A Low-complexity Bufferless Deflection Router", HPCA 2011. <sup>9</sup>

# Key Performance Issues

---

- 1. Link contention:** no buffers to hold traffic → any link contention causes a deflection  
→ use side buffers
- 2. Ejection bottleneck:** only one flit can eject per router per cycle → simultaneous arrival causes deflection  
→ eject up to 2 flits/cycle
- 3. Deflection arbitration:** practical (fast) deflection arbiters deflect unnecessarily  
→ new priority scheme (silver flit)

# Outline: This Talk

---

- **Motivation**
- **Background:** Bufferless Deflection Routing
- **MinBD:** Reducing Deflections
  - Addressing Link Contention
  - Addressing the Ejection Bottleneck
  - Improving Deflection Arbitration
- **Results**
- **Conclusions**

# Outline: This Talk

---

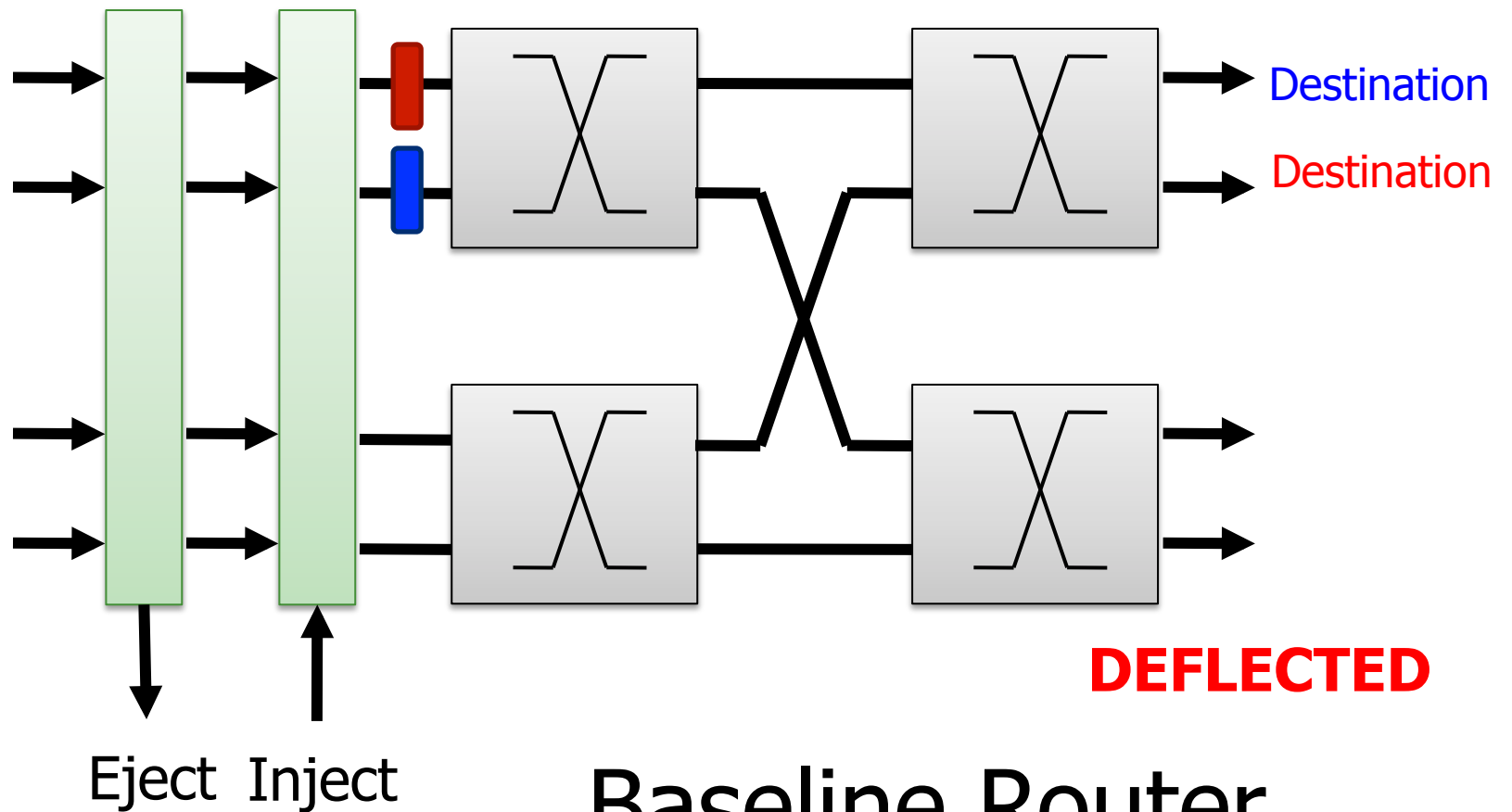
- **Motivation**
- **Background:** Bufferless Deflection Routing
- **MinBD:** Reducing Deflections
  - Addressing Link Contention
  - Addressing the Ejection Bottleneck
  - Improving Deflection Arbitration
- **Results**
- **Conclusions**

# Addressing Link Contention

---

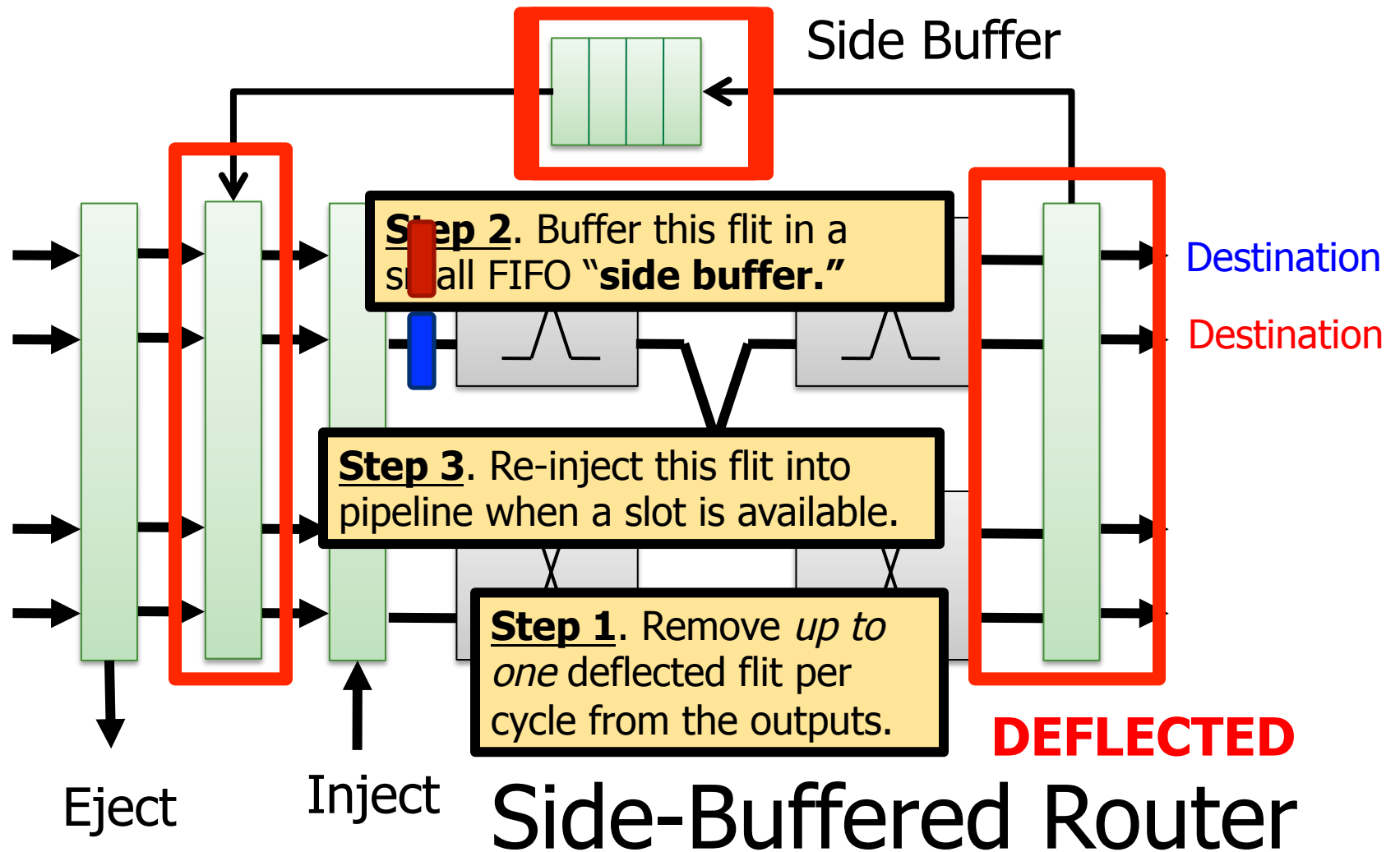
- **Problem 1:** Any link contention causes a deflection
- **Buffering** a flit can avoid deflection on contention
- But, **input buffers** are expensive:
  - All flits are buffered on every hop → **high dynamic energy**
  - Large buffers necessary → **high static energy** and **large area**
- **Key Idea 1:** add a **small buffer** to a bufferless deflection router to buffer **only** flits that **would have been deflected**

# How to Buffer Deflected Flits



<sup>1</sup> Fallin et al., "CHIPPER: A Low-complexity Bufferless Deflection Router", HPCA 2011.

# How to Buffer Deflected Flits



# Why Could A Side Buffer Work Well?

---

- Buffer some flits and deflect other flits at **per-flit level**
  - Relative to **bufferless routers**, **deflection rate reduces** (need not deflect all contending flits)
    - 4-flit buffer reduces deflection rate by **39%**
  - Relative to **buffered routers**, **buffer is more efficiently used** (need not buffer all flits)
    - similar performance with **25%** of buffer space



# Outline: This Talk

---

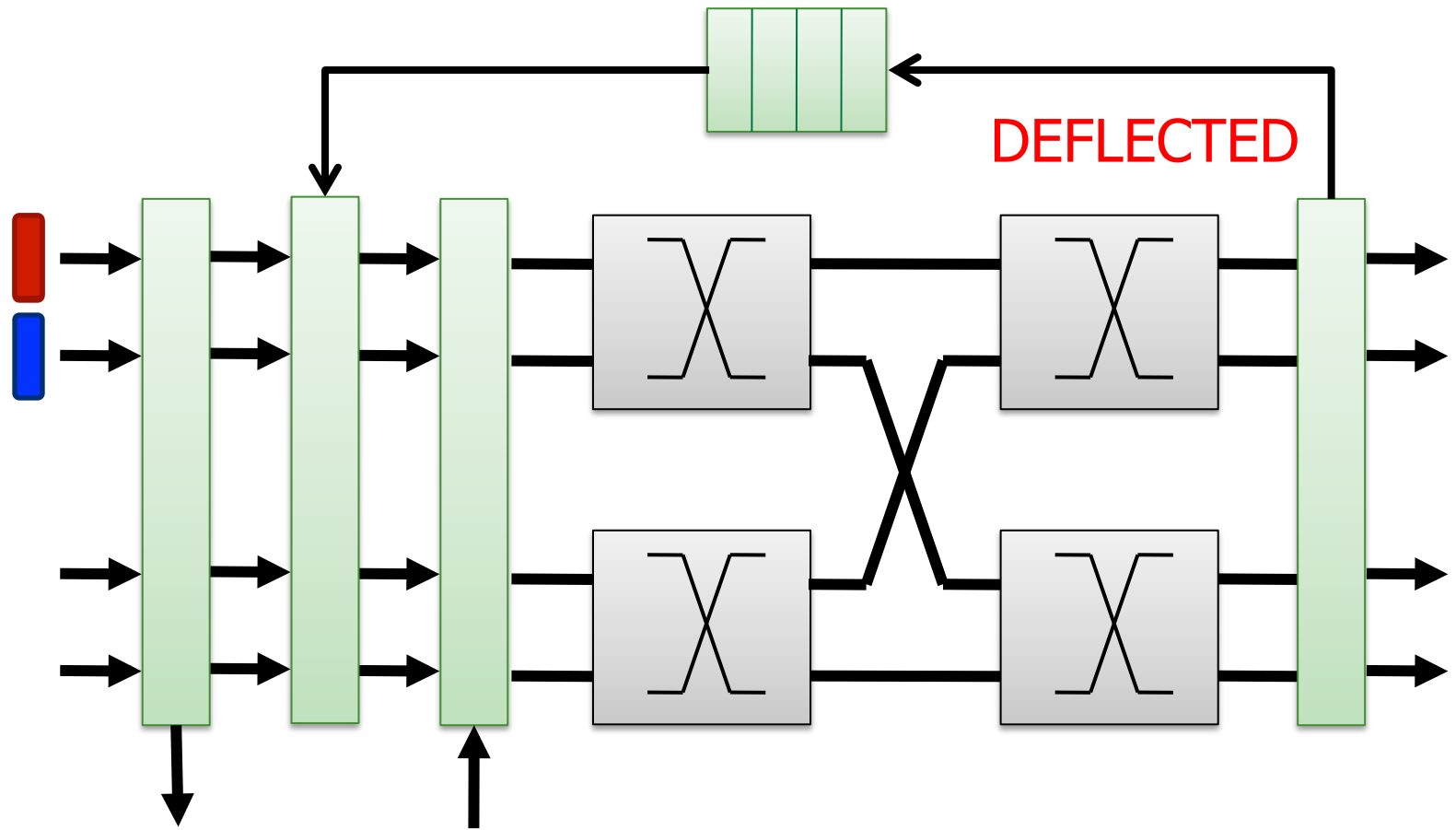
- **Motivation**
- **Background:** Bufferless Deflection Routing
- **MinBD:** Reducing Deflections
  - Addressing Link Contention
  - Addressing the Ejection Bottleneck
  - Improving Deflection Arbitration
- **Results**
- **Conclusions**

# Addressing the Ejection Bottleneck

---

- **Problem 2:** Flits deflect unnecessarily because only one flit can **eject** per router per cycle
- In 20% of all ejections,  $\geq 2$  flits could have ejected
  - all but one flit must **deflect and try again**
  - these deflected flits cause additional contention
- Ejection width of 2 flits/cycle reduces **deflection rate 21%**
- **Key idea 2:** Reduce deflections due to a single-flit ejection port by allowing **two flits** to eject per cycle

# Addressing the Ejection Bottleneck

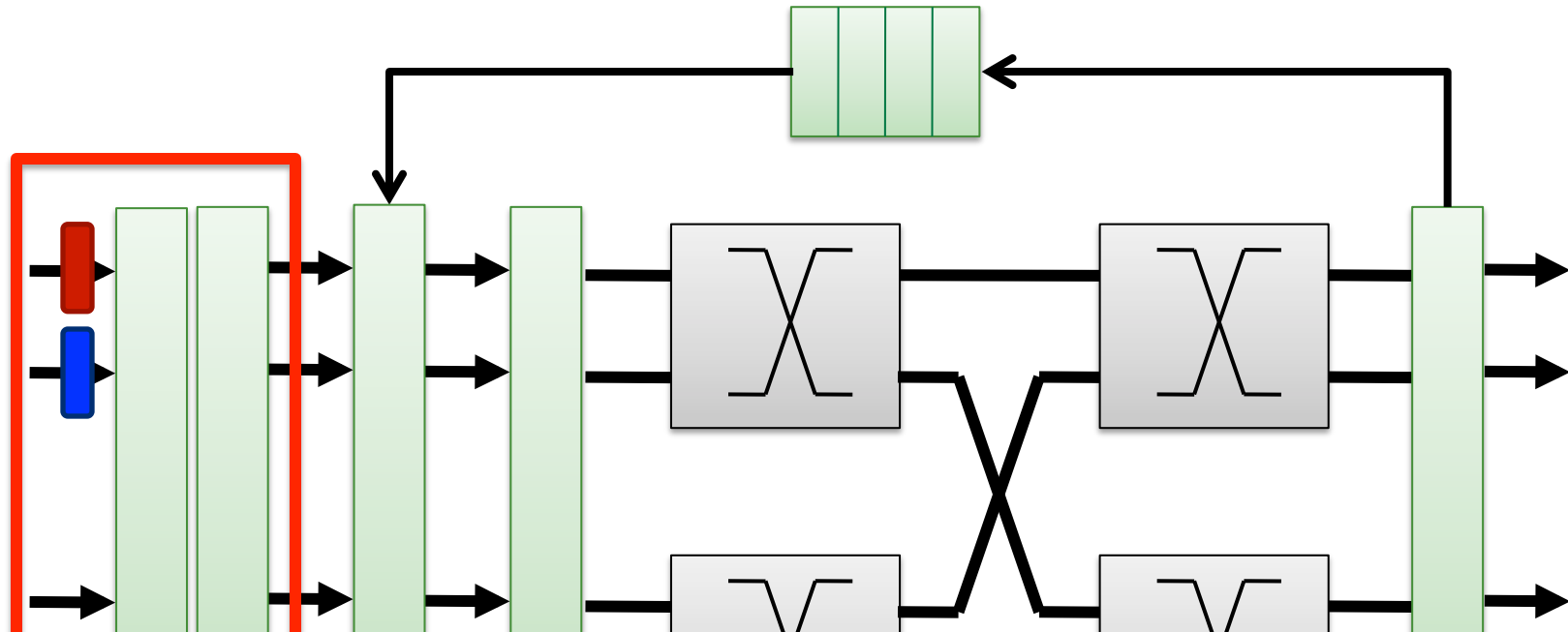


Eject

Inject

## Single-Width Ejection

# Addressing the Ejection Bottleneck



For fair comparison, **baseline routers** have dual-width ejection for perf. (not power/area)

Eject

Inject

Dual-Width Ejection

# Outline: This Talk

---

- **Motivation**
- **Background:** Bufferless Deflection Routing
- **MinBD:** Reducing Deflections
  - Addressing Link Contention
  - Addressing the Ejection Bottleneck
  - Improving Deflection Arbitration
- **Results**
- **Conclusions**

# Improving Deflection Arbitration

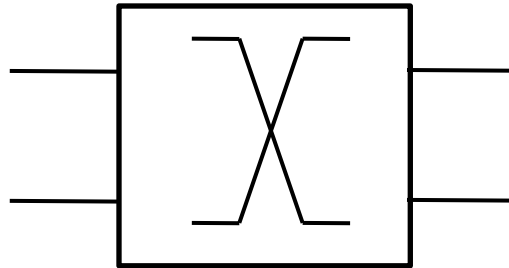
---

- **Problem 3:** Deflections occur unnecessarily because fast arbiters must use simple priority schemes
- Age-based priorities (several past works): full priority order gives fewer deflections, but requires **slow arbiters**
- State-of-the-art deflection arbitration (Golden Packet & two-stage permutation network)
  - Prioritize one packet globally (**ensure forward progress**)
  - Arbitrate other flits randomly (**fast critical path**)
- Random common case leads to **uncoordinated arbitration**

# Fast Deflection Routing Implementation

---

- Let's route in a two-input router first:



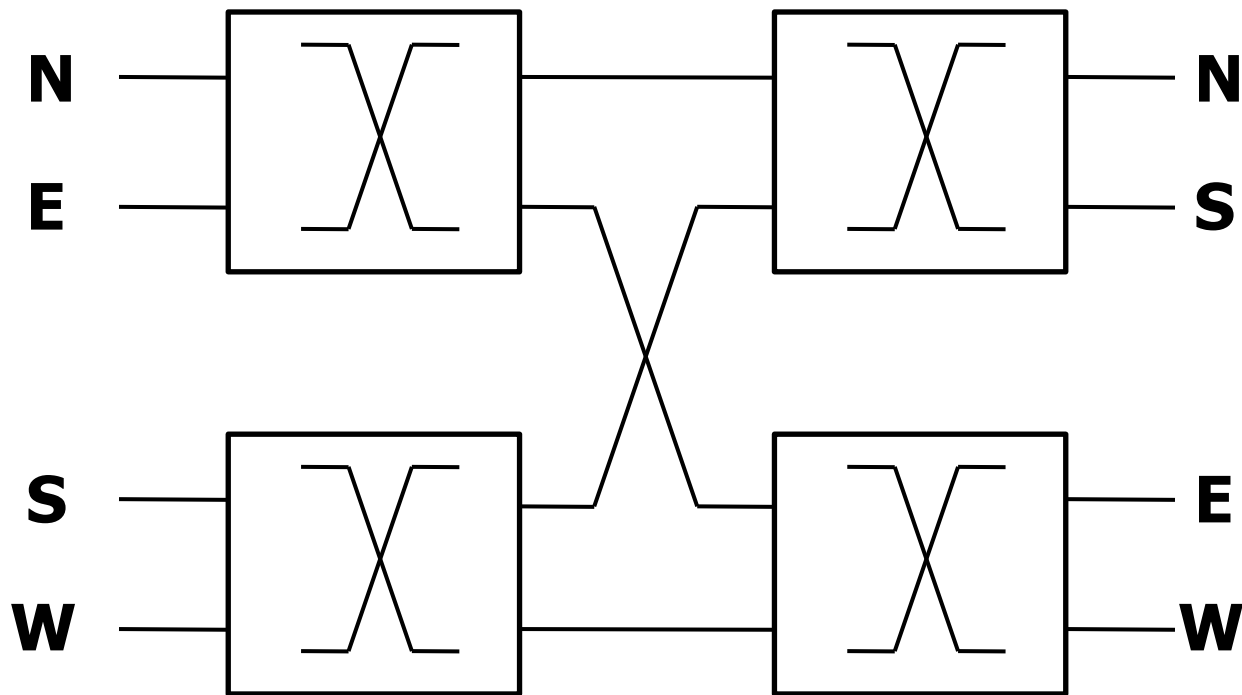
- **Step 1:** pick a "winning" flit (Golden Packet, else random)
- **Step 2:** steer the winning flit to its desired output and deflect other flit

**→ Highest-priority flit always routes to destination**

# Fast Deflection Routing with Four Inputs

---

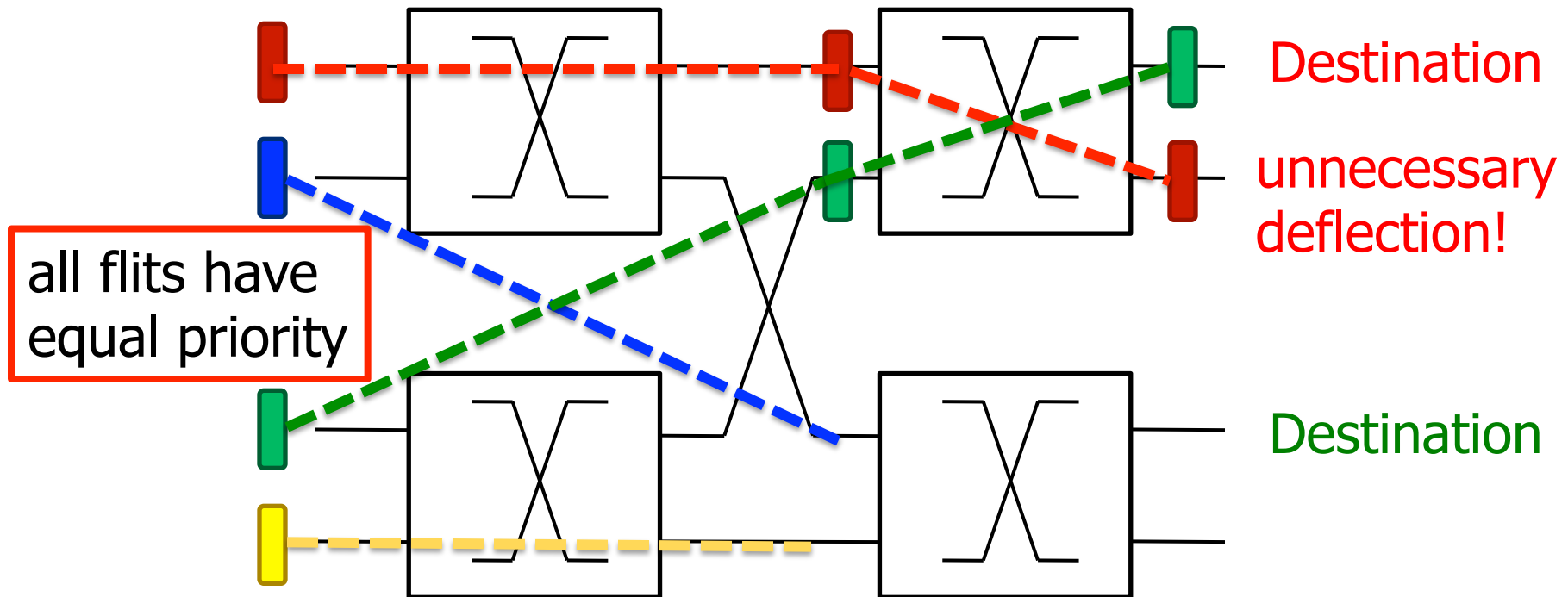
- Each block makes decisions **independently**
  - **Deflection is a distributed decision**





# Unnecessary Deflections in Fast Arbiters

- How does lack of coordination cause unnecessary deflections?
  - No flit is golden (pseudorandom arbitration)
  - Red flit wins at first stage
  - Green flit loses at first stage (must be deflected now)
  - Red flit loses at second stage; Red and Green are deflected



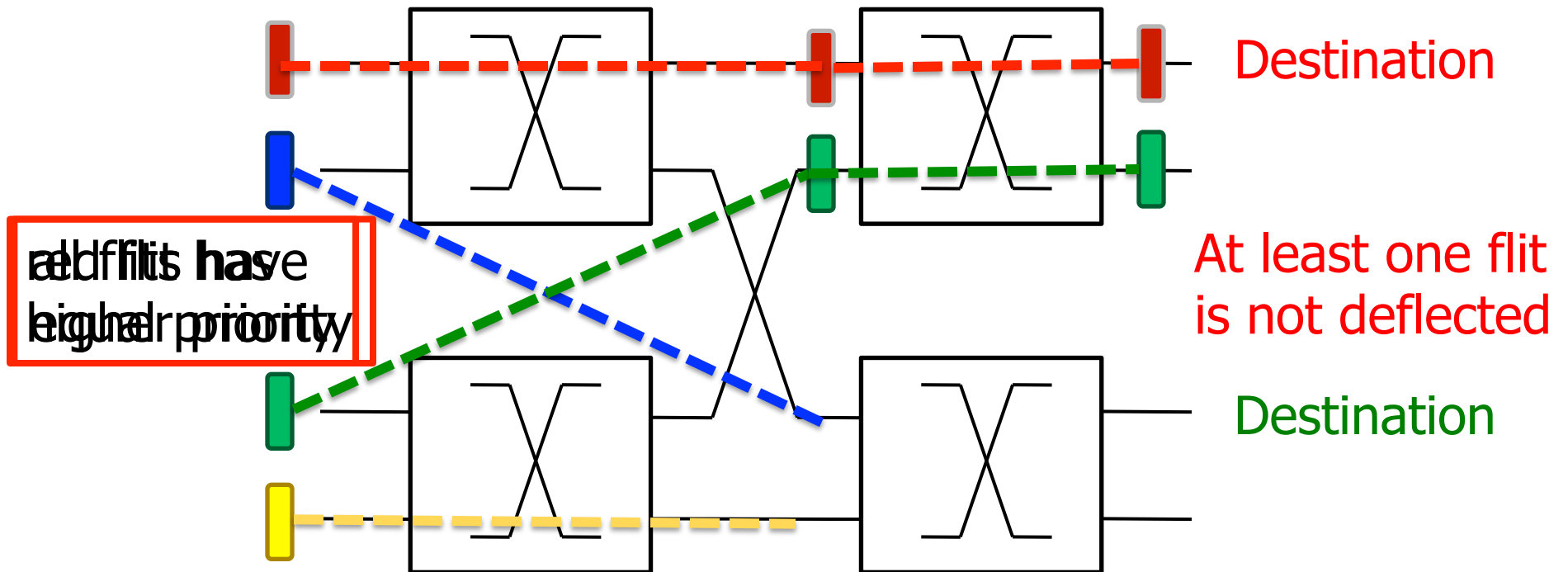
# Improving Deflection Arbitration

---

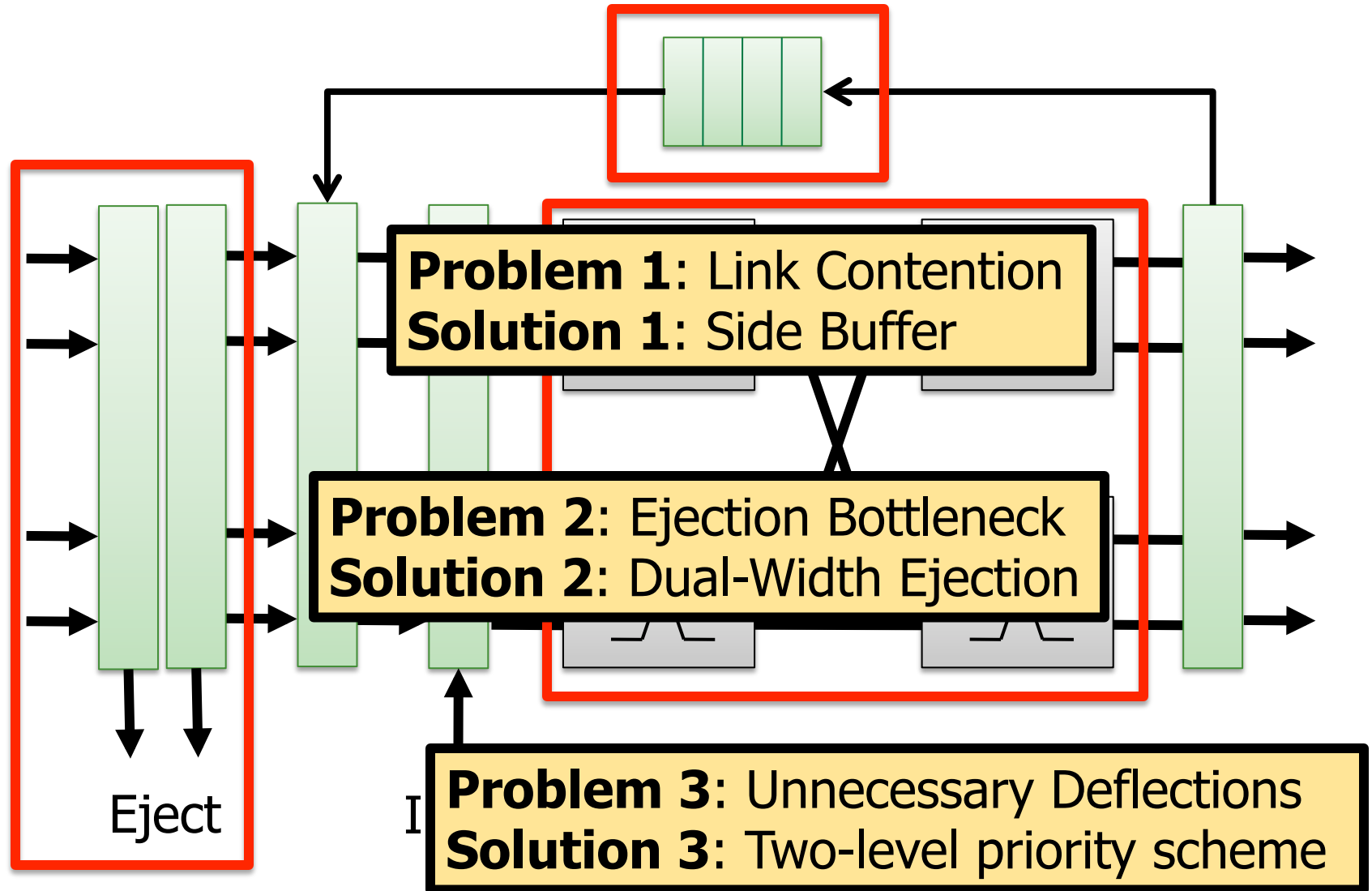
- **Key idea 3: Add a priority level** and prioritize one flit to ensure at least **one flit is not deflected in each cycle**
  
- **Highest priority:** one **Golden Packet** in network
  - Chosen in static round-robin schedule
  - **Ensures correctness**
  
- **Next-highest priority:** one **silver flit** per router per cycle
  - Chosen pseudo-randomly & local to one router
  - **Enhances performance**

# Adding A Silver Flit

- Randomly picking a silver flit ensures **one flit is not deflected**
  - No flit is golden but Red flit is silver
  - Red flit wins at first stage (silver)
  - Green flit is deflected at first stage
  - Red flit wins at second stage (silver); not deflected



# Minimally-Buffered Deflection Router



# Outline: This Talk

---

- **Motivation**
- **Background:** Bufferless Deflection Routing
- **MinBD:** Reducing Deflections
  - Addressing Link Contention
  - Addressing the Ejection Bottleneck
  - Improving Deflection Arbitration

# Outline: This Talk

---

- **Motivation**
- **Background:** Bufferless Deflection Routing
- **MinBD:** Reducing Deflections
  - Addressing Link Contention
  - Addressing the Ejection Bottleneck
  - Improving Deflection Arbitration
- **Results**
- **Conclusions**

# Methodology: Simulated System

---

- **Chip Multiprocessor Simulation**
  - **64-core** and **16-core** models
  - **Closed-loop** core/cache/NoC cycle-level model
  - Directory cache coherence protocol (SGI Origin-based)
  - 64KB L1, perfect L2 (stresses interconnect), XOR-mapping
  - Performance metric: **Weighted Speedup**  
(similar conclusions from network-level latency)
  - Workloads: multiprogrammed SPEC CPU2006
    - 75 randomly-chosen workloads
    - Binned into network-load categories by average injection rate

# Methodology: Routers and Network

---

- **Input-buffered** virtual-channel router
    - 8 VCs, 8 flits/VC [[Buffered\(8,8\)](#)]: large buffered router
    - 4 VCs, 4 flits/VC [[Buffered\(4,4\)](#)]: typical buffered router
    - 4 VCs, 1 flit/VC [[Buffered\(4,1\)](#)]: smallest deadlock-free router
    - All power-of-2 buffer sizes up to (8, 8) for perf/power sweep
  - **Bufferless deflection** router: **CHIPPER**<sup>1</sup>
  - **Bufferless-buffered hybrid** router: **AFC**<sup>2</sup>
    - Has input buffers and deflection routing logic
    - Performs coarse-grained (multi-cycle) mode switching
  - **Common parameters**
    - 2-cycle router latency, 1-cycle link latency
    - 2D-mesh topology (16-node: 4x4; 64-node: 8x8)
    - Dual ejection assumed for baseline routers (for perf. only)
- 

<sup>1</sup>Fallin et al., "CHIPPER: A Low-complexity Bufferless Deflection Router", HPCA 2011.

<sup>2</sup>Jafri et al., "Adaptive Flow Control for Robust Performance and Energy", MICRO 2010.



# Methodology: Power, Die Area, Crit. Path

---

## ■ **Hardware modeling**

- Verilog models for CHIPPER, MinBD, buffered control logic
  - Synthesized with commercial 65nm library
- ORION 2.0 for datapath: crossbar, muxes, buffers and links

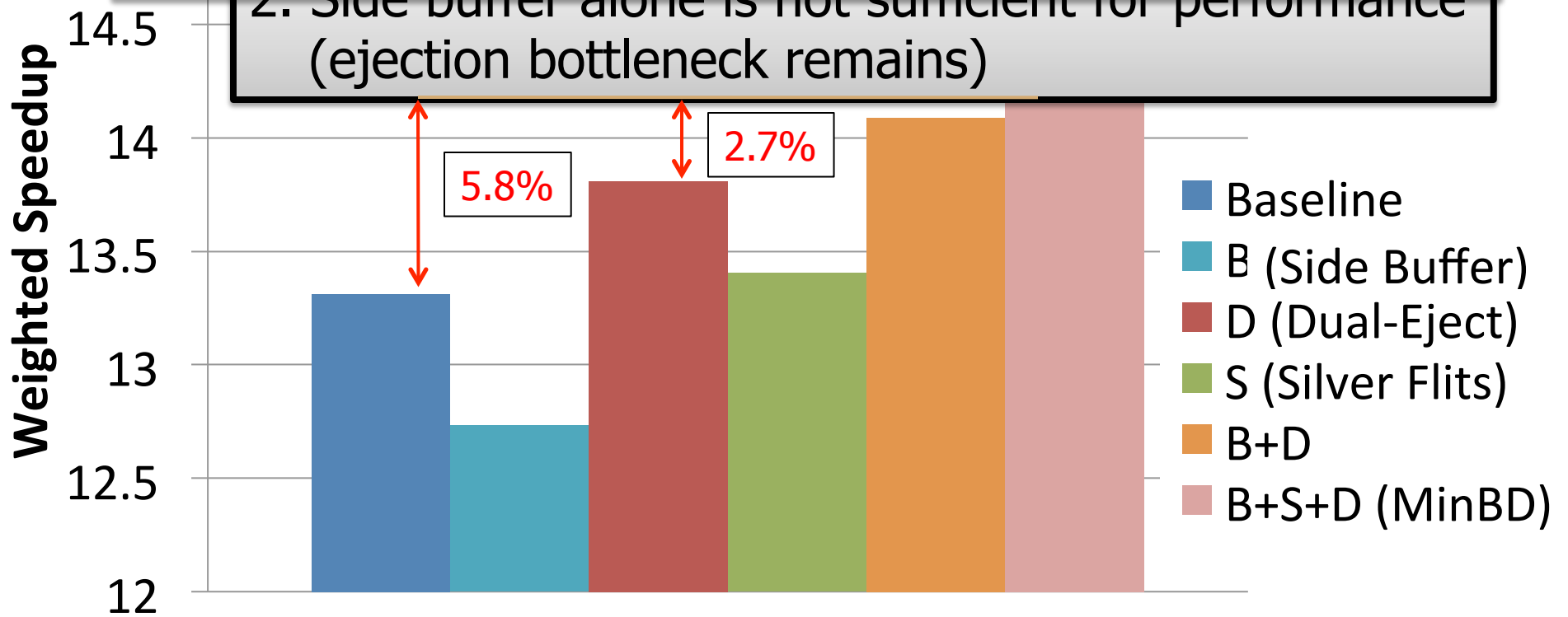
## ■ **Power**

- Static and dynamic power from hardware models
- Based on event counts in cycle-accurate simulations
- Broken down into buffer, link, other

# Reduced Deflections & Improved Perf.

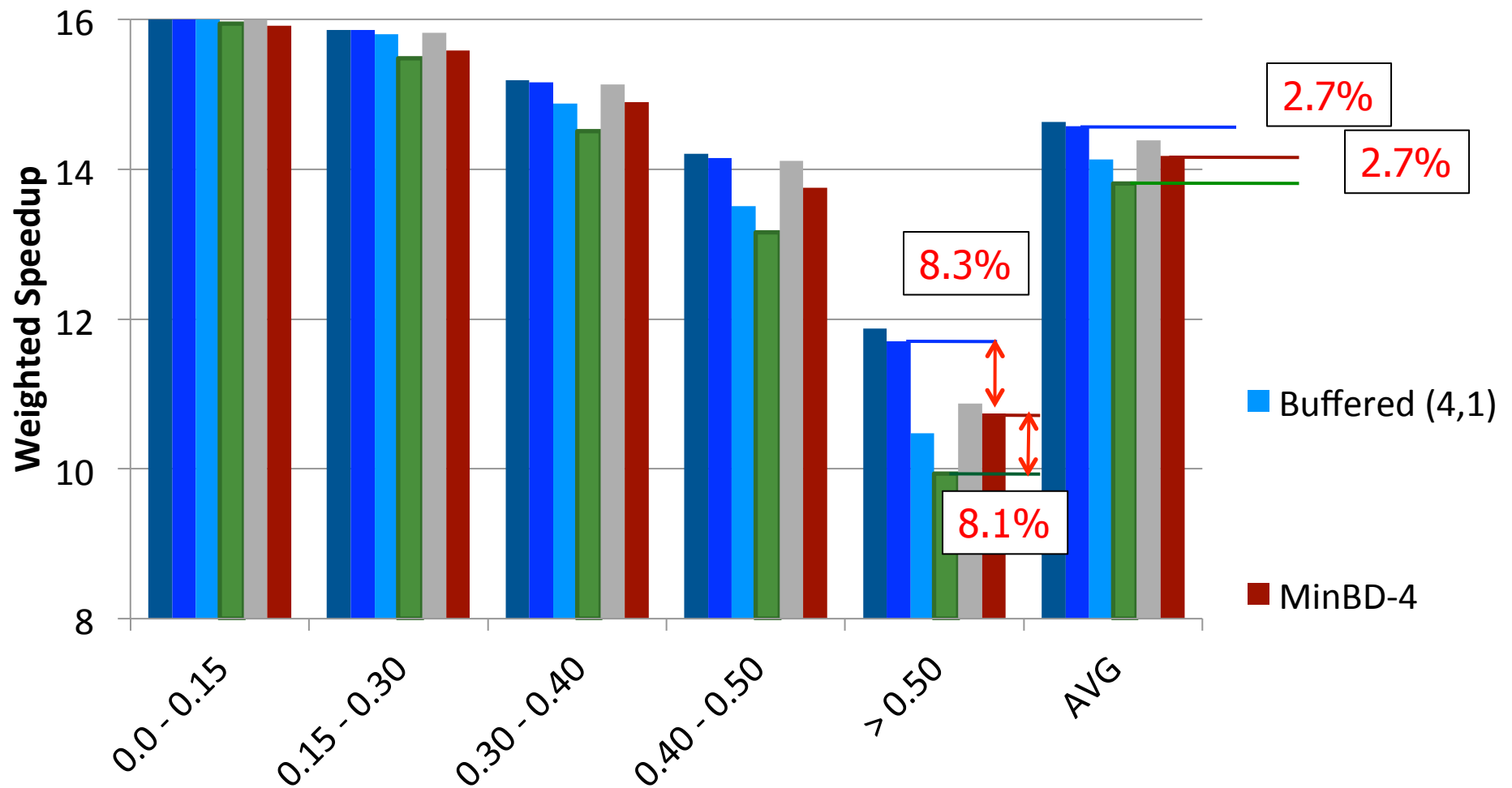
3. Overall, **5.8%** over baseline, **2.7%** over dual-eject by reducing deflections **64%** / **54%**

2. Side buffer alone is not sufficient for performance (ejection bottleneck remains)



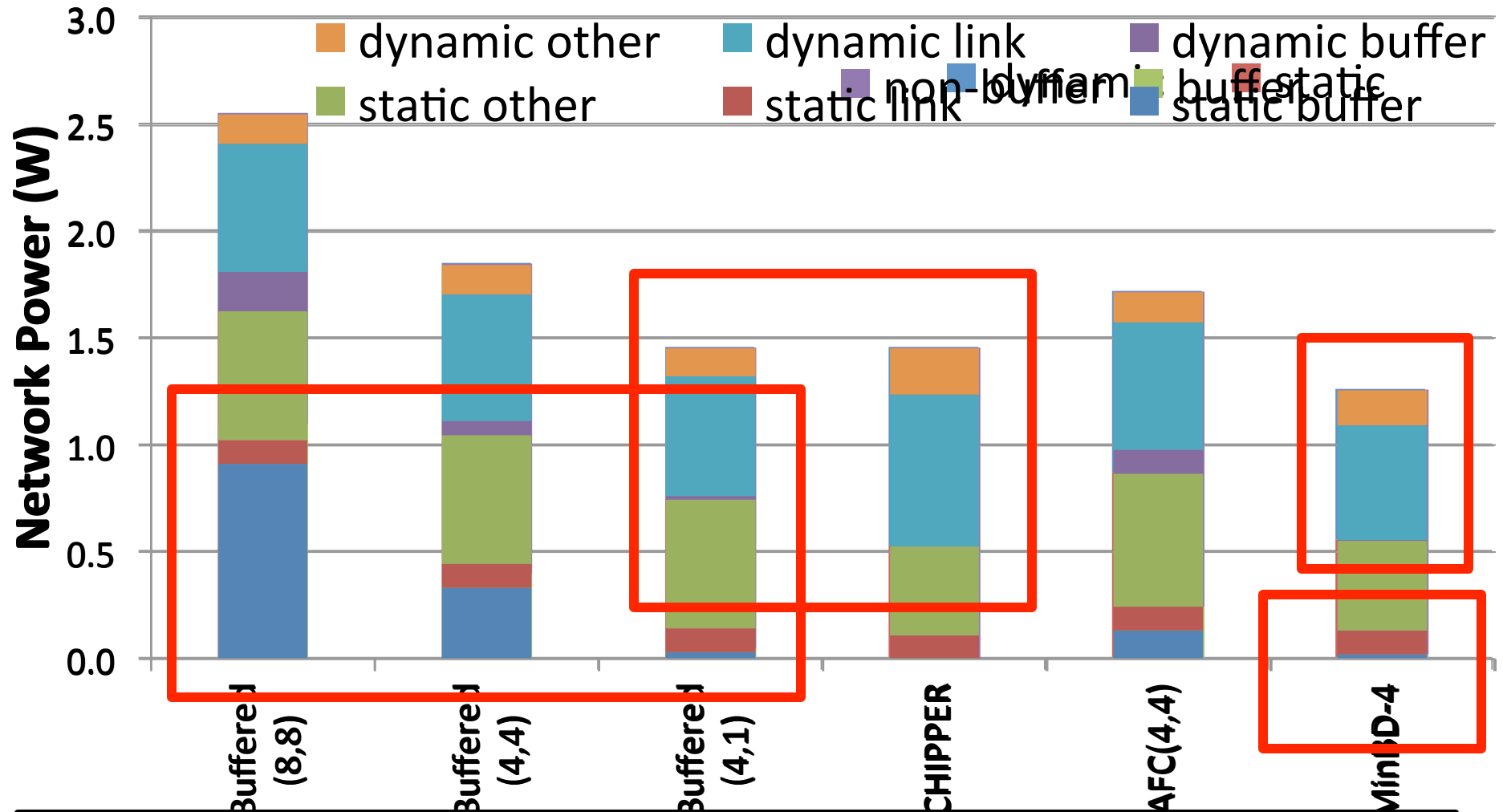
Deflection Rate	28%	17%	22%	27%	11%	10%

# Overall Performance Results



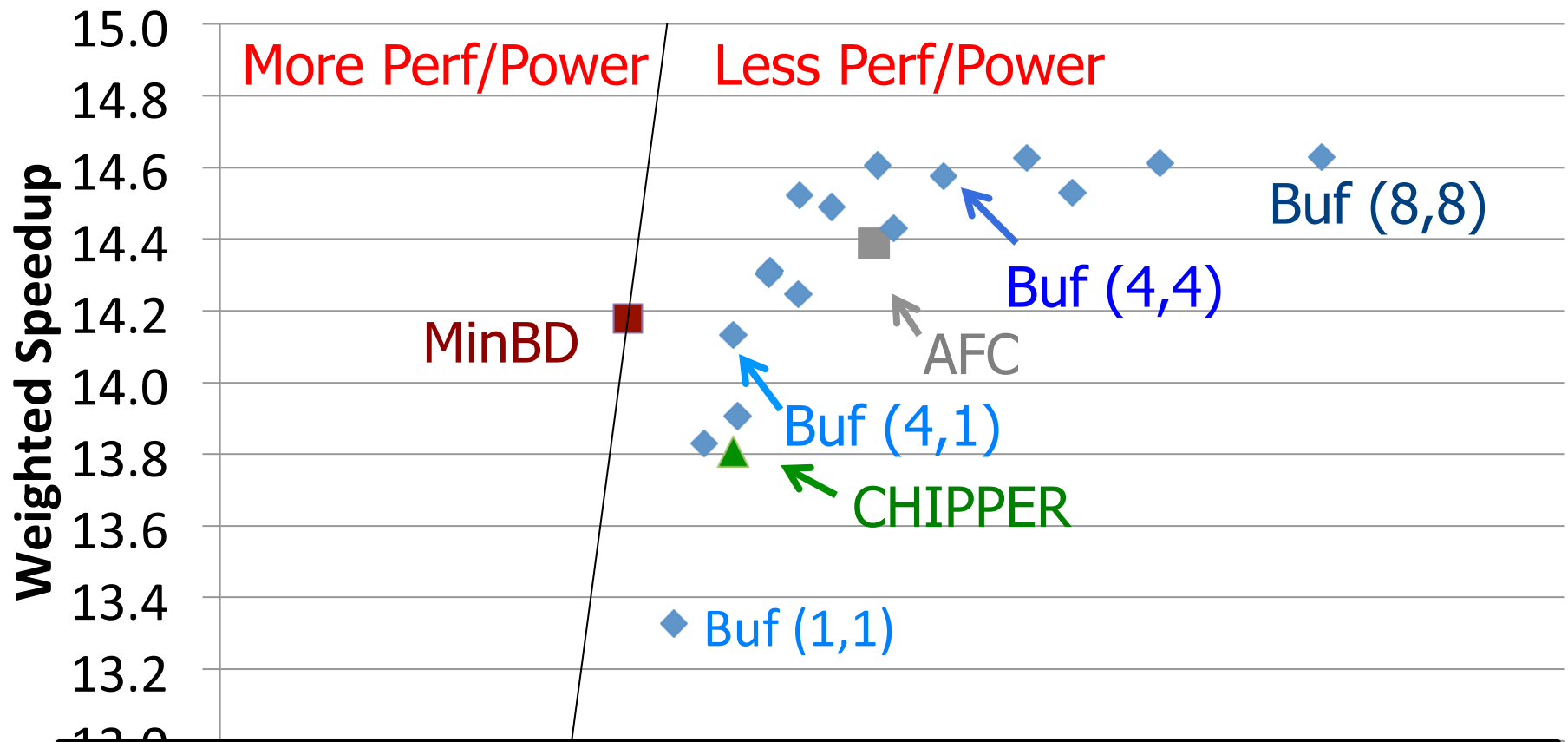
- Similar perf. to Buffered (4,1) @ 25% of buffering space
- Within **2.7%** of Buffered (4,4) (**8.3%** at high load)

# Overall Power Results



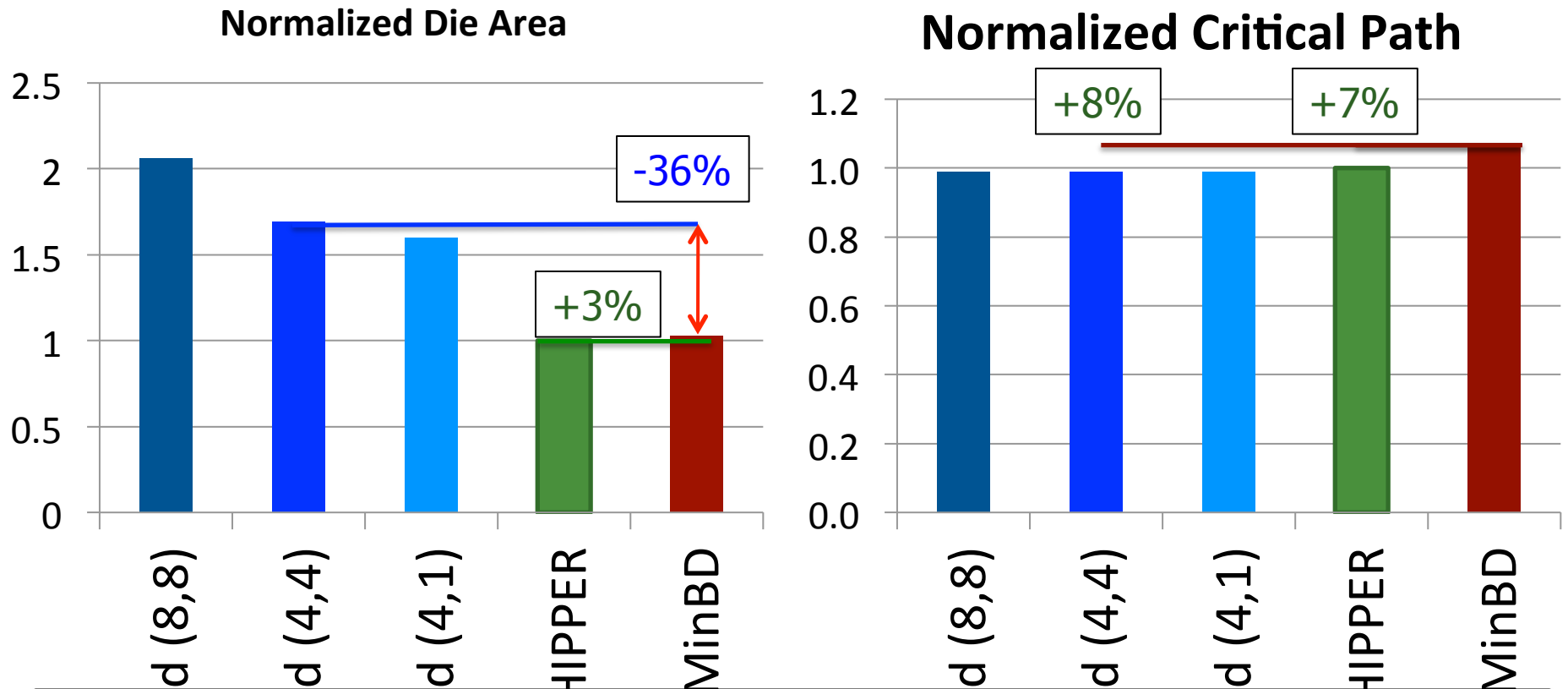
- Dynamic power increases with deflection routing
- Dynamic power reduces in MinBD relative to CHIPPER

# Performance-Power Spectrum



- Most **energy-efficient** (perf/watt) of any evaluated network router design

# Die Area and Critical Path



- Only **3%** area increase over CHIPPER (4-flit buffer)
- Increases by **7%** over CHIPPER, **8%** over Buffered (4,4)

# Conclusions

---

- Bufferless deflection routing offers **reduced power & area**
  - But, high deflection rate hurts **performance at high load**
  - **MinBD** (Minimally-Buffered Deflection Router) introduces:
    - **Side buffer** to hold **only** flits that would have been deflected
    - **Dual-width ejection** to address ejection bottleneck
    - **Two-level prioritization** to avoid unnecessary deflections
  - MinBD yields **reduced power (31%) & reduced area (36%)** relative to **buffered** routers
  - MinBD yields **improved performance (8.1% at high load)** relative to **bufferless** routers → closes half of perf. gap
  - MinBD has the **best energy efficiency** of all evaluated designs with **competitive performance**
-

---

**THANK YOU!**



# MinBD: Minimally-Buffered Deflection Routing for Energy-Efficient Interconnect

**Chris Fallin**, Greg Nazario, Xiangyao Yu\*,  
Kevin Chang, Rachata Ausavarungnirun, Onur Mutlu

Carnegie Mellon University

\*CMU and Tsinghua University

**SAFARI** Carnegie Mellon University

---

# **BACKUP SLIDES**

# Correctness: Golden Packet

---

- The Golden Packet is always prioritized long enough to be delivered (hop latency \* (max # hops + serialization delay))
  - “Epoch length”: e.g. 4x4:  $3 * (7 + 7) = 42$  cycles (pick 64 cyc)
- Golden Packet rotates statically through all packet IDs
  - E.g. 4x4: 16 senders, 16 transactions/sender → 256 choices
- Max latency is GP epoch \* # packet IDs
  - E.g.,  $64 * 256 = 16K$  cycles
- Flits in Golden Packet are arbitrated by sequence # (total order)

# Correctness: Retransmit-Once

---

- Finite reassembly buffer size may lead to buffer exhaustion
- What if a flit arrives from a new packet and no buffer is free?
- Answer 1: Refuse ejection and deflect → deadlock!
- Answer 2: Use large buffers → impractical
- **Retransmit-Once** (past work): operate opportunistically & assume available buffers
  - If no buffer space, drop packet (once) and note its ID
  - Later, reserve buffer space and retransmit (once)
- End-to-end flow control provides correct endpoint operation without in-network backpressure

# Correctness: Side Buffer

---

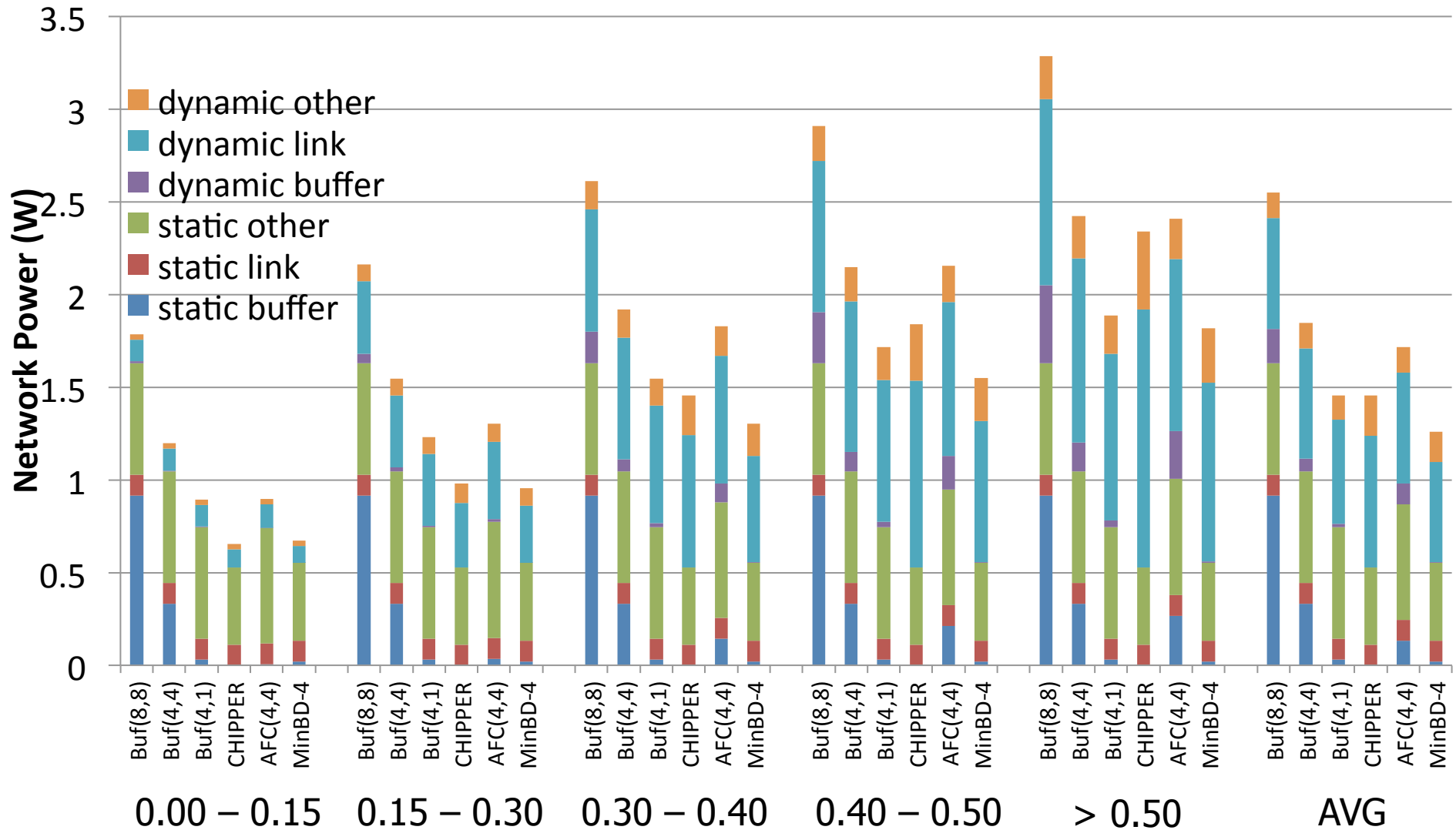
- Golden Packet ensures delivery *as long as flits keep moving*
  - What if flits get “stuck” in a side buffer?
  - Answer: **buffer redirection**
    - If buffered flit cannot re-inject after  $C_{\text{threshold}}$  cycles, then:
      1. Force one input flit per cycle into buffer (random choice)
      2. Re-inject buffered flit into resulting empty slot in network
  - If a flit is golden, it will **never enter a side buffer**
  - If a flit *becomes* golden while buffered, redirection will rescue it after  $C_{\text{threshold}} * \text{BufferSize}$  (e.g.:  $2 * 4 = 8$  cyc)
  - Extend Golden epoch to account for this
-

# Why does Side Buffer Alone Lose Perf.?

---

- Adding a side buffer reduces deflection rate
  - Raw network throughput increases
- But **ejection is still the system bottleneck**
  - Ejection rate remains nearly constant
- Side buffers are utilized → **more traffic in flight**
- Hence, **latency increases** (Little's Law):  $\sim 10\%$

# Overall Power Results



# MinBD vs. AFC

---

## ■ **AFC:**

- ❑ Combines **input buffers** and **deflection routing**
- ❑ In a given cycle, all link contention is handled by **buffers** or by **deflection** (global router mode)
- ❑ Mode-switch is heavyweight (drain input buffers) and takes multiple cycles
- ❑ Router has area footprint of buffered + bufferless, but could save power with power-gating (assumed in Jafri et al.)
- ❑ Better performance at highest loads (equal to buffered)

## ■ **MinBD:**

- ❑ Combines **deflection routing** with a **side buffer**
  - ❑ In a given cycle, some flits are buffered, some are deflected
  - ❑ Smaller router and no mode switching
  - ❑ But, loses some performance at highest load
-



# Related Work

---

- **Baran**, 1964
  - Original “hot potato” (deflection) routing
- **BLESS** (Moscibroda and Mutlu, ISCA 2009)
  - Earlier bufferless deflection router
  - Age-based arbitration → slow (did not consider critical path)
- **CHIPPER** (Fallin et al., HPCA 2011)
  - Assumed baseline for this work
- **AFC** (Jafri et al., MICRO 2010)
  - Coarse-grained bufferless-buffered hybrid
- **SCARAB** (Hayenga et al., MICRO 2009), **BPS** (Gomez+08)
  - Drop-based deflection networks
  - SCARAB: dedicated circuit-switched NACK network