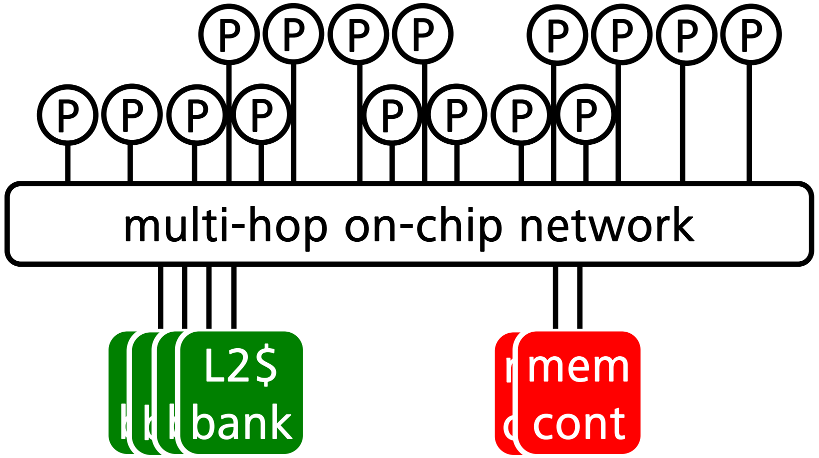
GLOBALLY-SYNCHRONIZED FRAMES FOR GUARANTEED QUALITY-OF-SERVICE IN ON-CHIP NETWORKS

Jae W. Lee (MIT)
Man Cheuk Ng (MIT)
Krste Asanovic (UC Berkeley)

June 23th 2008

ISCA-35, Beijing, China

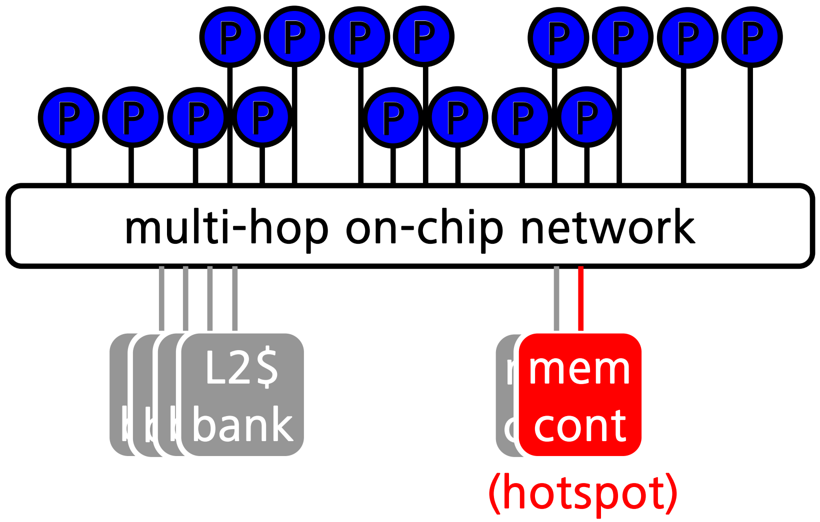
Resource sharing increases performance variation



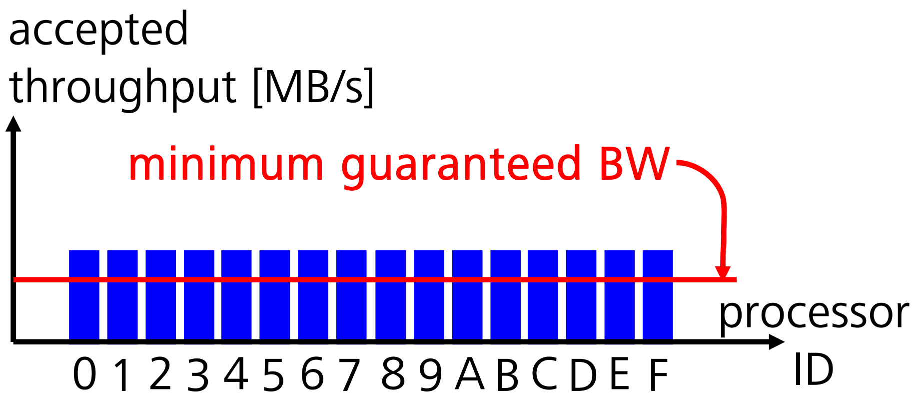
- Resource sharing
  - (**+**) reduces hardware cost
  - (**-**) increases performance variation

- This performance variation becomes larger and larger as the number of sharers (cores) increases.

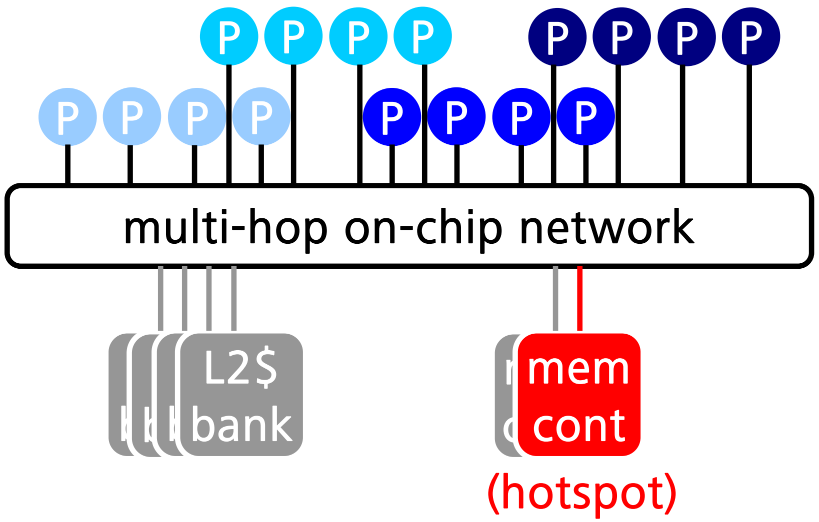
Desired quality-of-service from shared resources



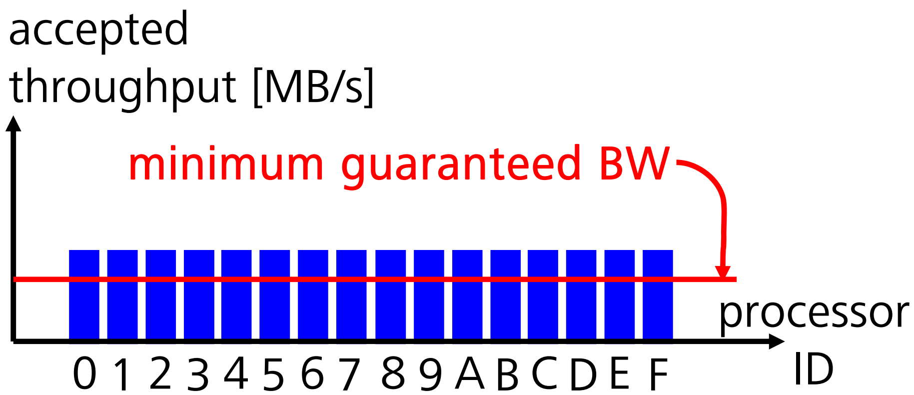
- Performance isolation (fairness)

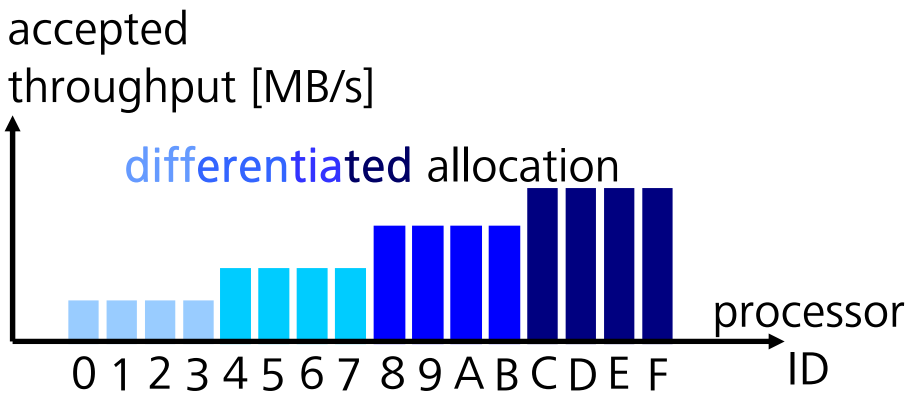


Desired quality-of-service from shared resources

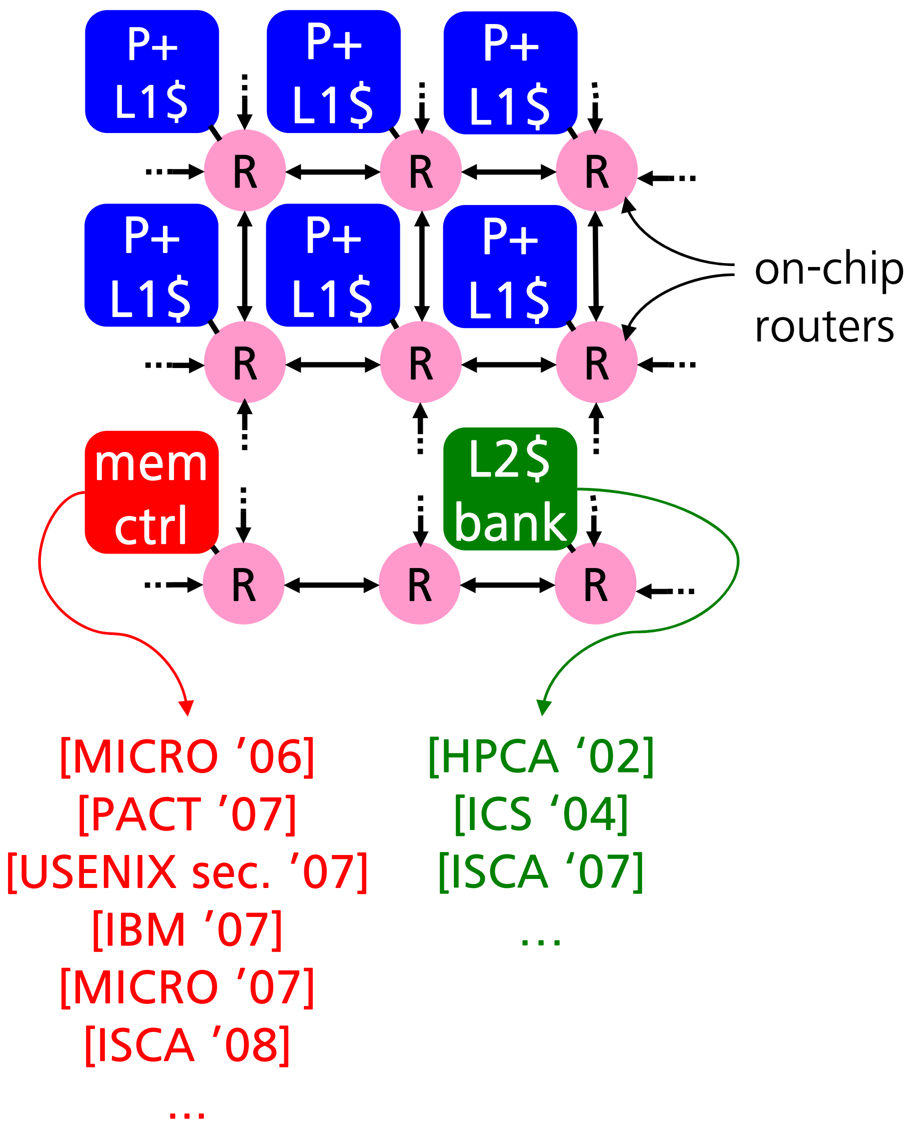


- Performance isolation (fairness)
- Differentiated services (flexibility)





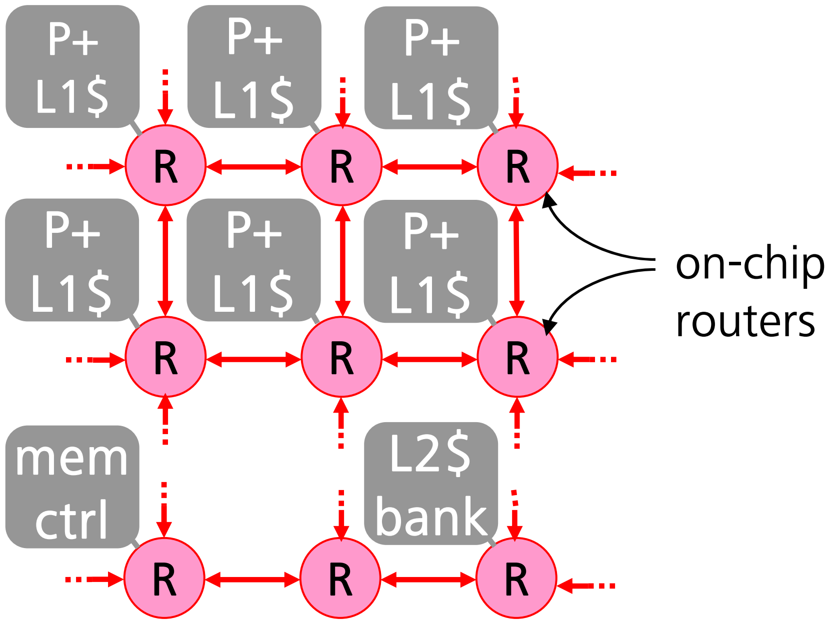
Resources w/ centralized arbitration are well investigated



- Resources with *centralized* arbitration
  - SDRAM controllers
  - L2 cache banks
- They have a single entry point for all requests.

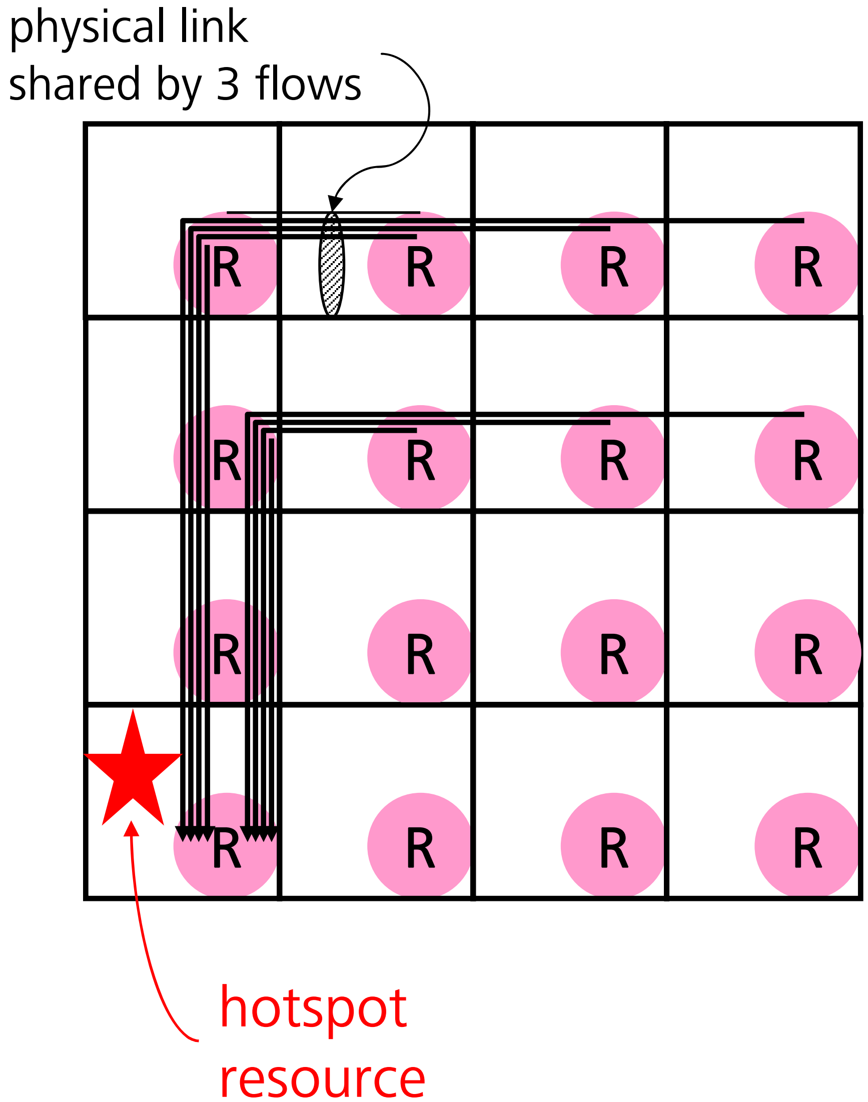
  → QoS is relatively easier and well investigated.

QoS from on-chip networks is a challenge



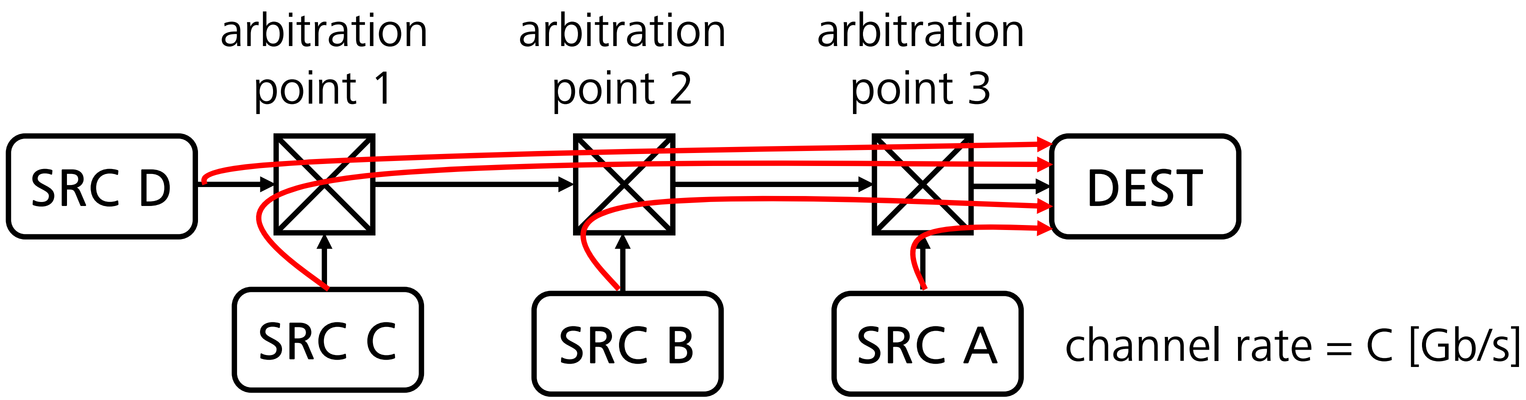
- Resources with *distributed* arbitration
  - multi-hop on-chip networks
- They have distributed arbitration points.
  → QoS is more difficult.
- Off-chip solutions cannot be directly applied because of resource constraints.

We guarantee QoS for flows



- Flow: a sequence of packets between a unique pair of end nodes (src and dest)
  - physical links shared by flows
  - multiple stages of arbitration for each packet
- We provide guaranteed QoS to each flow with:
  - minimum bandwidth guarantees
  - bounded maximum delay

Locally fair $\not\Rightarrow$ globally fair



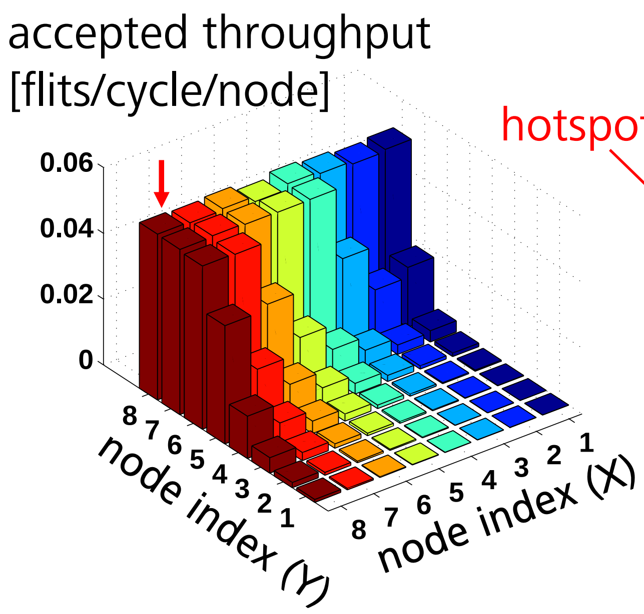
With locally fair round-robin (RR) arbitration:

- Throughput (Flow A) = (0.5) C
- Throughput (Flow B) = $(0.5)^2$ C
- Throughput (Flow C) = Throughput (Flow D) = $(0.5)^3$ C

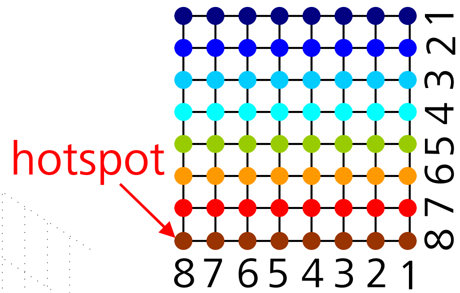
→ Throughput of a flow decreases exponentially as its distance to the destination (hotspot) increases.

Motivational simulation

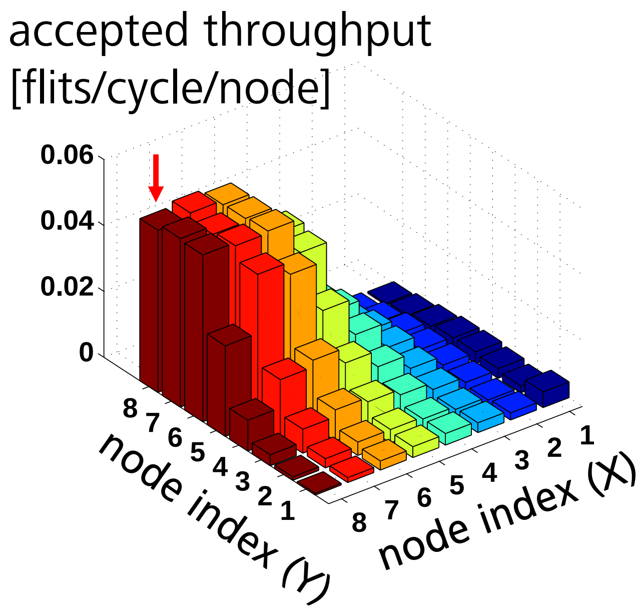
- In 8x8 mesh network with RR arbitration (hotspot at (8, 8))



w/ dimension-ordered routing



8x8 2D mesh

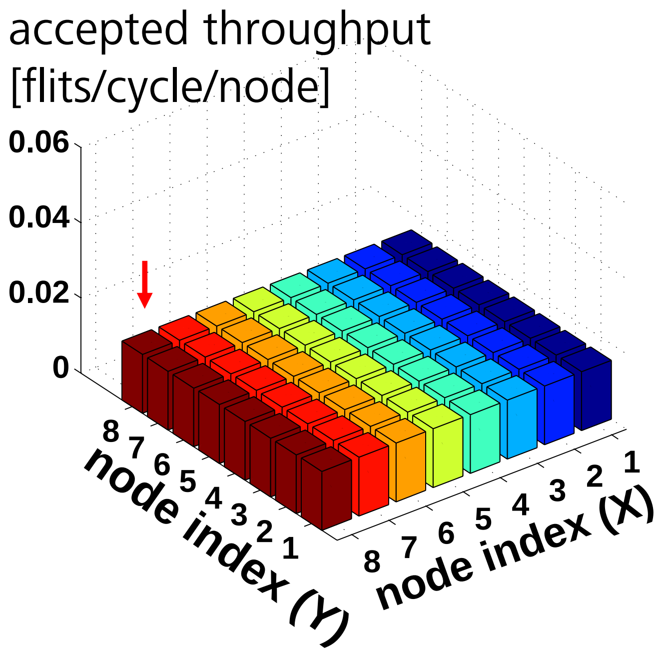


w/ minimal-adaptive routing

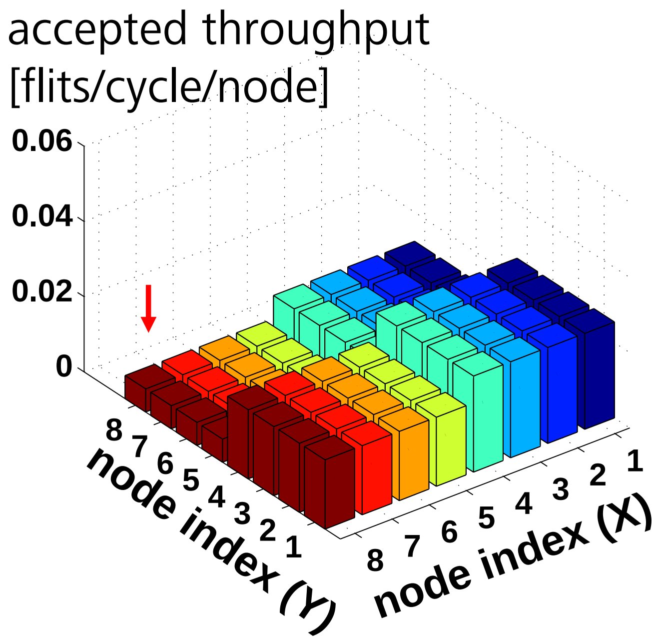
locally-fair round-robin scheduling → globally unfair bandwidth usage

Desired bandwidth allocation: an example

- Taken from simulation results with GSF:



Fair allocation



Differentiated allocation

Globally Synchronized Frames (GSF)

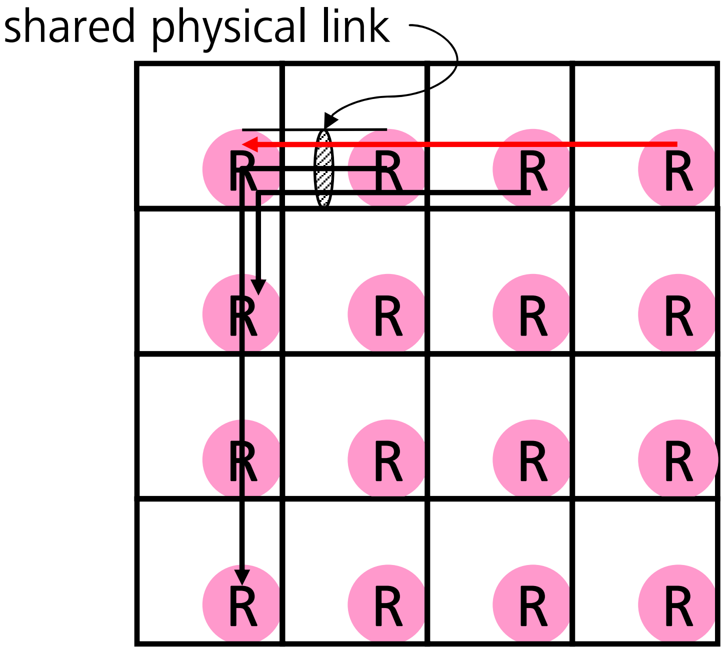
provide guaranteed QoS with minimum bandwidth guarantees and maximum delay to each flow in multi-hop on-chip networks:

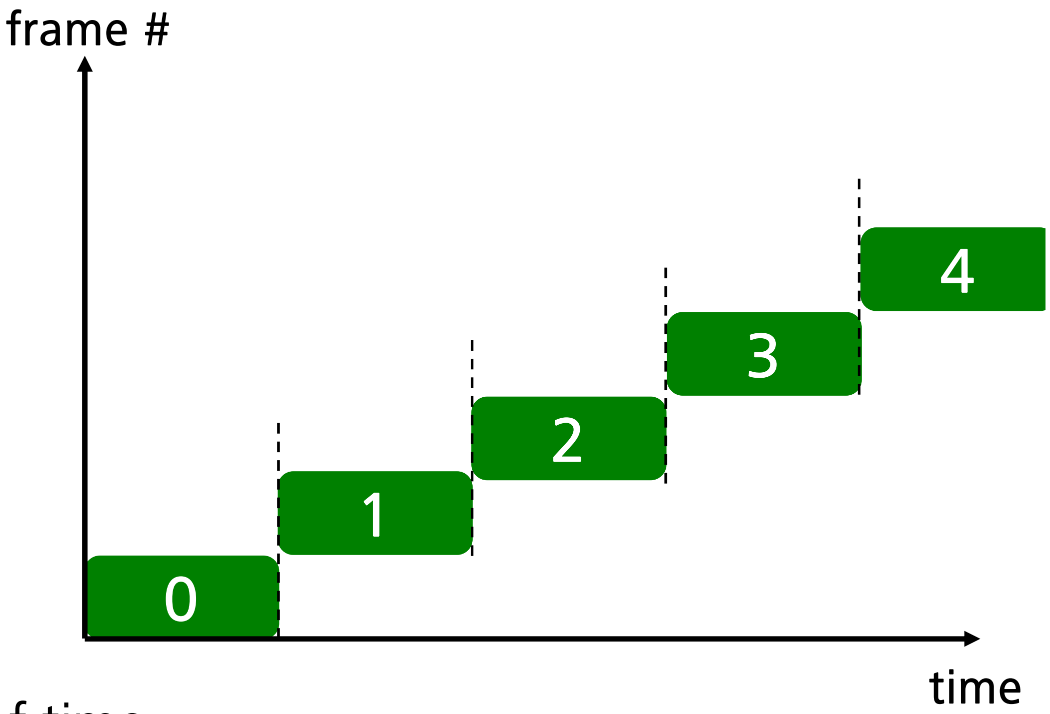
- with high network utilization comparable to best-effort virtual-channel router
- with minimal area/energy overhead by avoiding per-flow queues/structures in on-chip routers
  → scalable to # of concurrent flows

Outline of this talk

- Motivation
- Globally-Synchronized Frames: a step-by-step development of mechanism
- Implementation of GSF router
- Evaluation
- Related work
- Conclusion

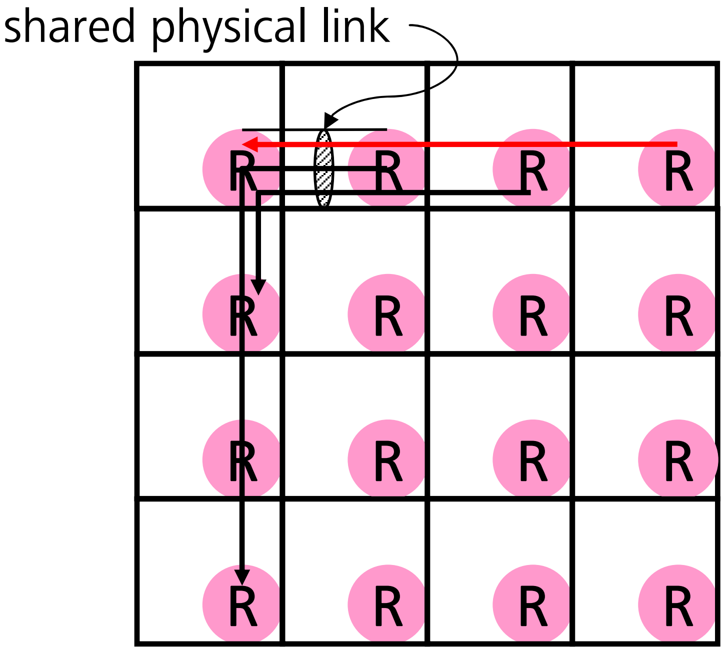
GSF takes a frame-based approach

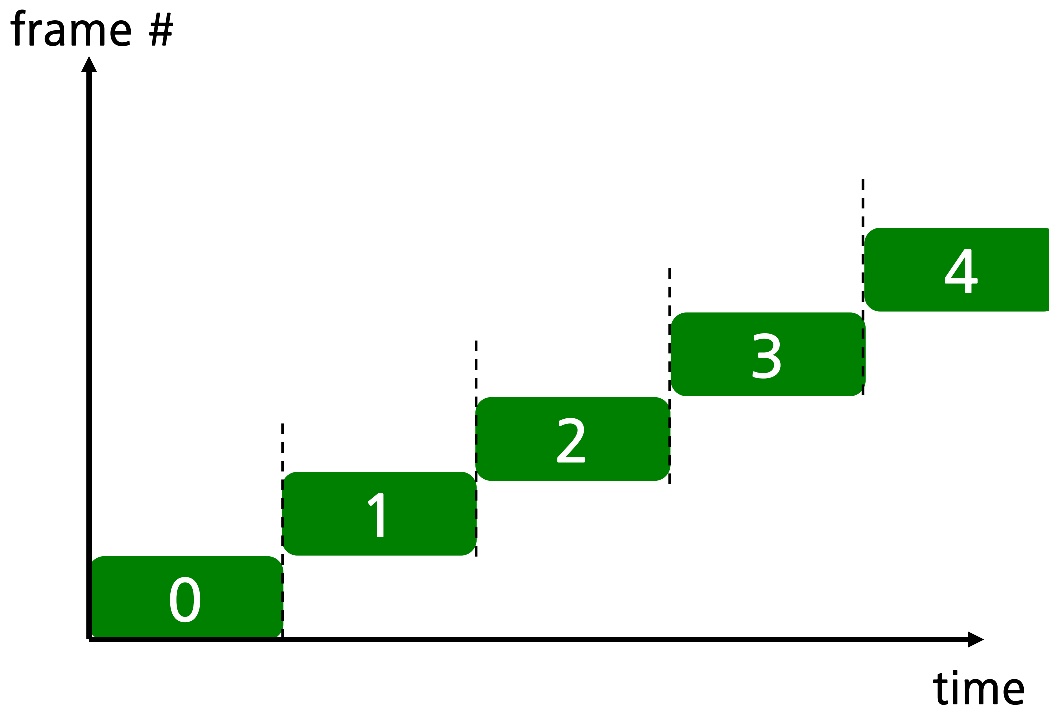




- Frame is a coarse quantization of time.
  - The network can transport a finite number of flits during this interval.
- We constrain each flow source to inject a certain number of flits per frame.
  - shorter frames → coarser BW control but lower maximum delay
  - typically 1-100s Kflits / frame (over all flows) in 8x8 mesh network

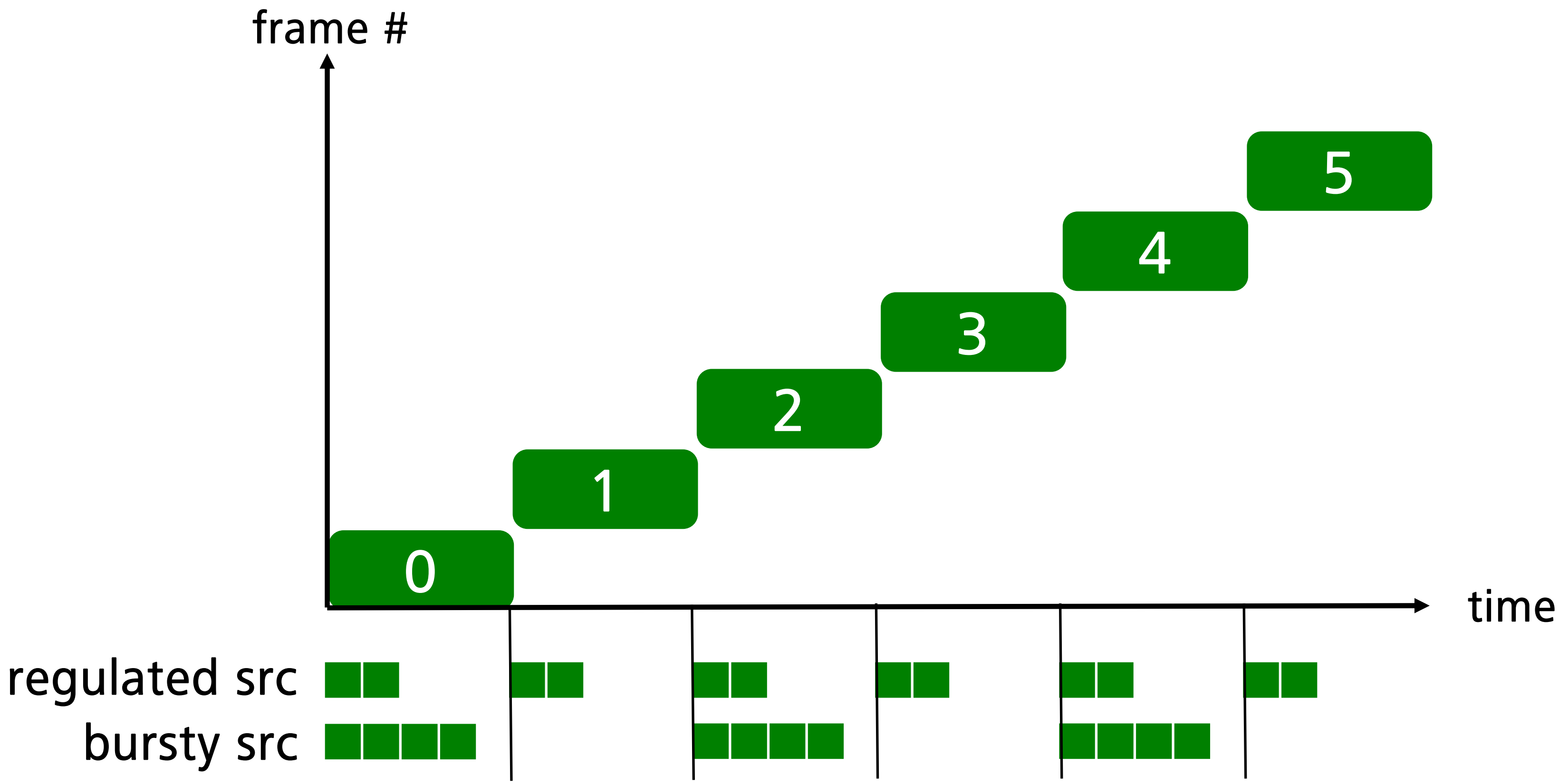
Admission control of flows





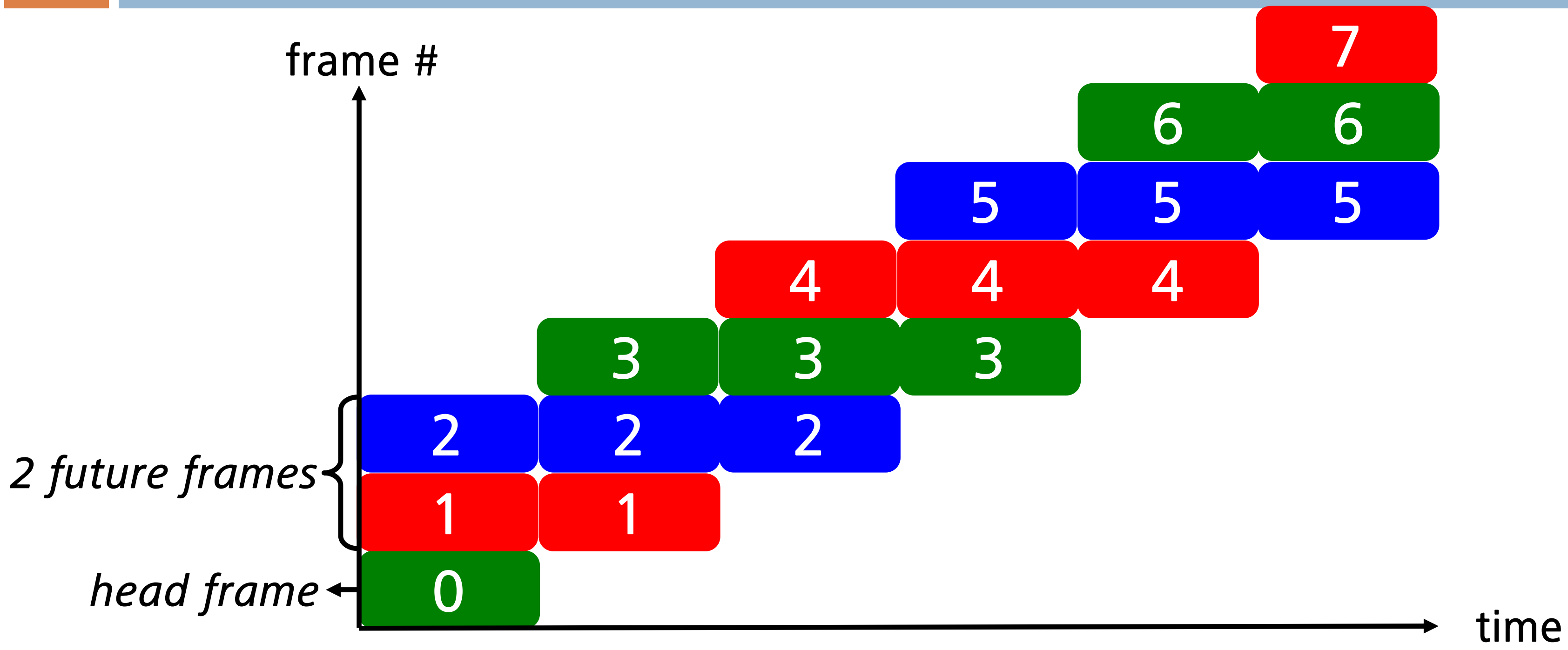
- Admission control: reject a new flow if it would make the network unable to transport all the injected flits within a frame interval

Single frame does not service bursty traffic well



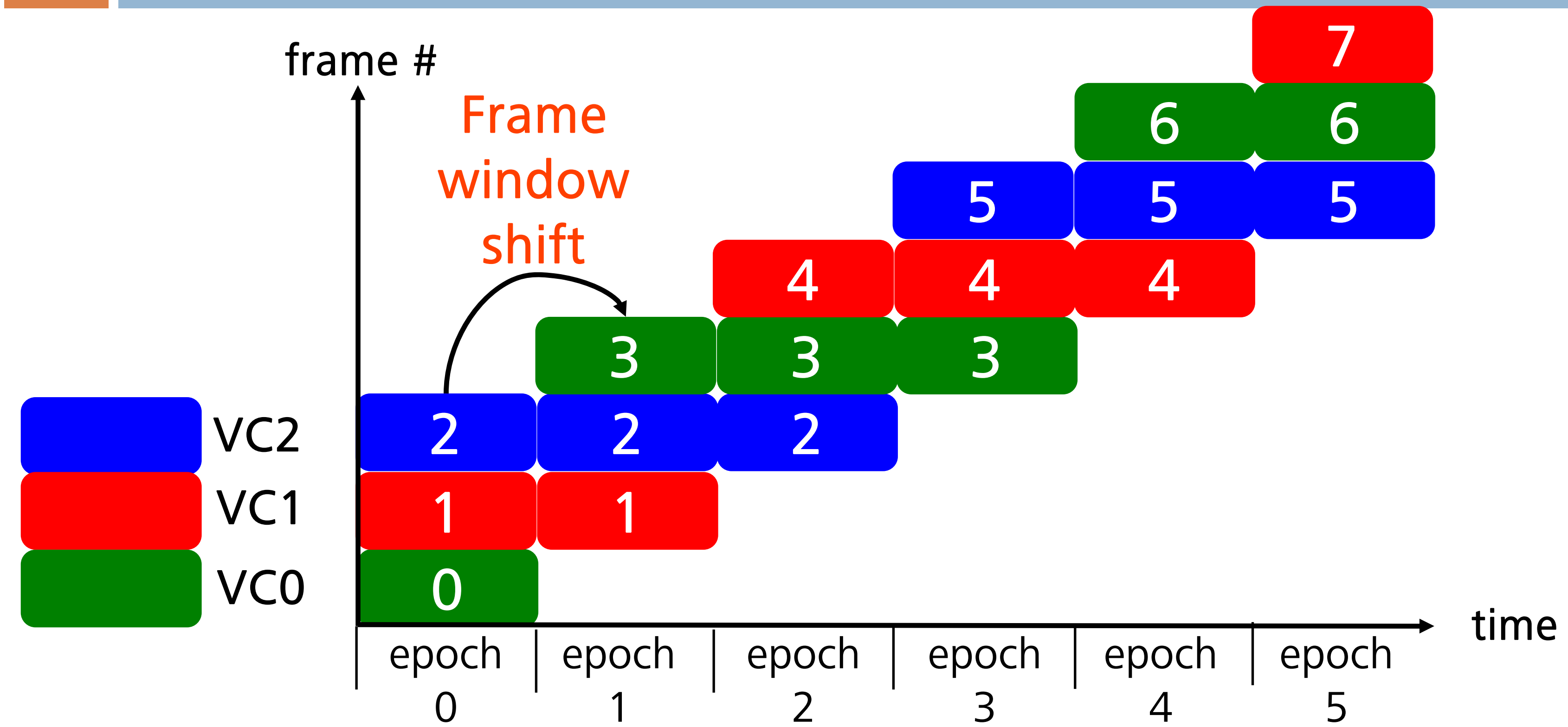
- Both traffic sources have the same long-term rate: 2 flits / frame.
- Allocating 2 flits / frame penalizes the bursty source.

Overlapping multiple frames to help bursty traffic



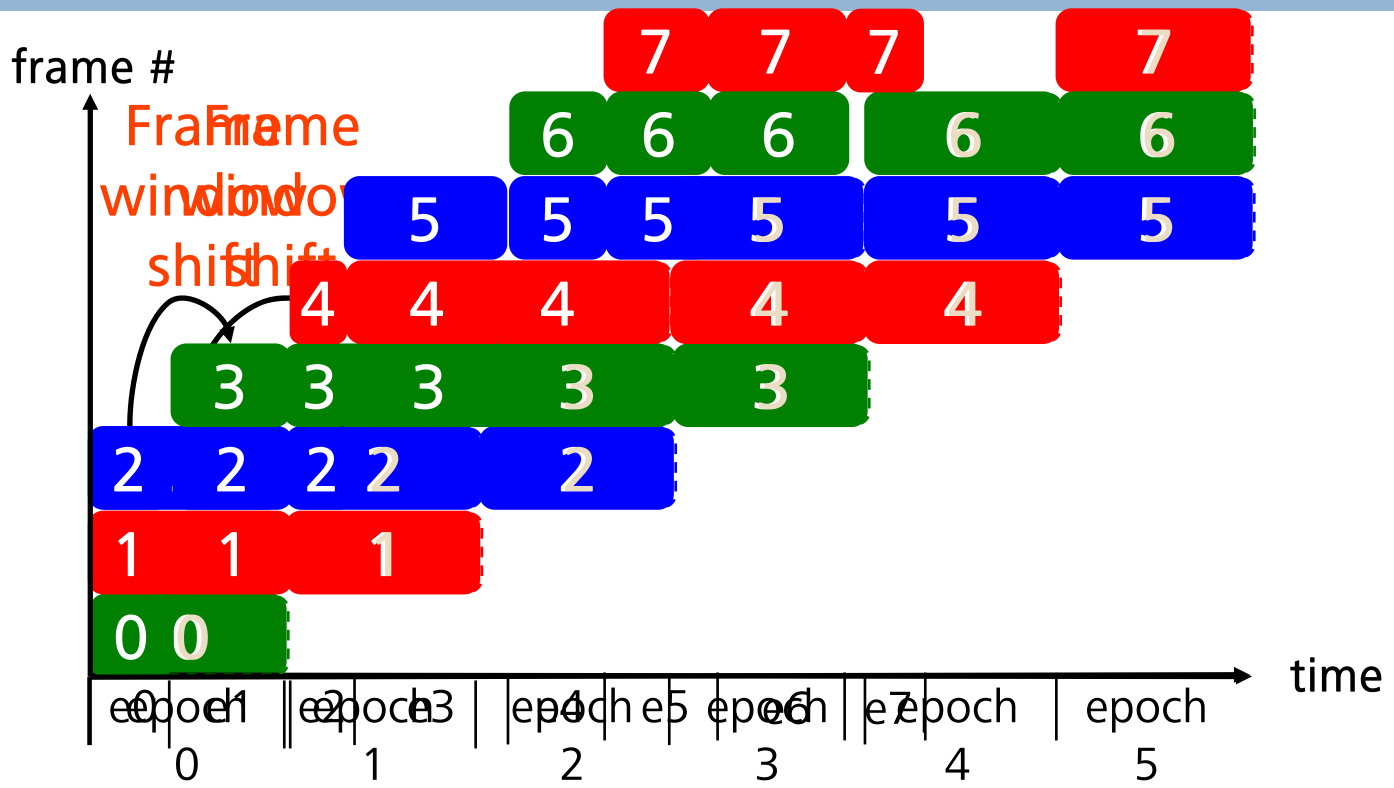
- **Overlapping multiple frames** to multiply injection slots
  - Sources can inject flits into future frames (w/ separate per-frame buffers)
  - Older frames have higher priorities for contended channels.
    - Drain time of head frame does not change.
    - Future frames can use unclaimed BW by older frames.
  - Maximum network delay < 3 * (frame interval)
- Best-effort traffic: always lowest priority (throughput ↑)

Reclamation of frame buffers



- Per-frame buffers (at each node) = virtual channels
- At every frame window shift, frame buffers (or VCs) associated with the earliest frame in the previous epoch are reclaimed for the new futuremost frame.

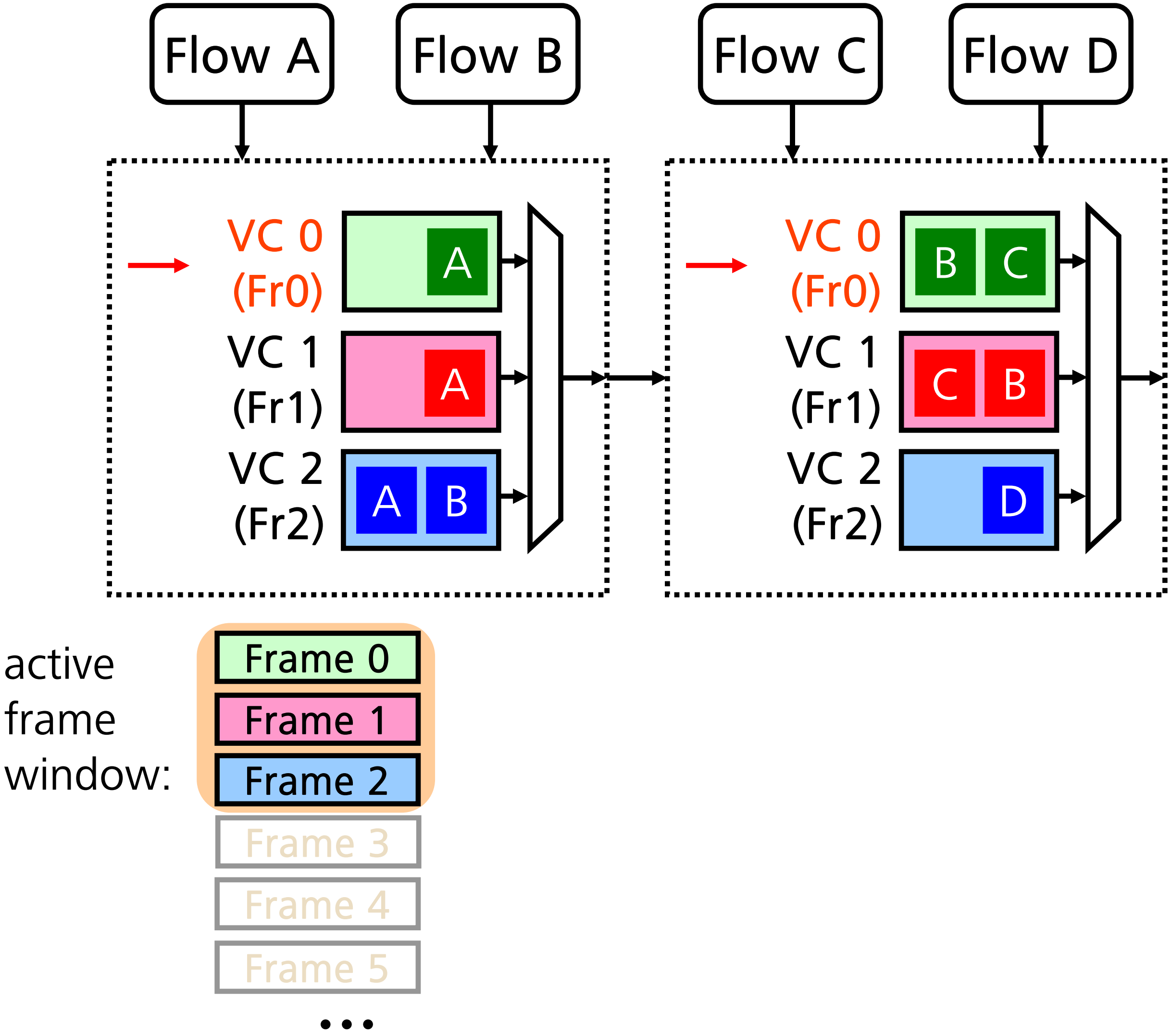
Early reclamation improves network throughput



- Observation: Head frame usually drains much earlier than frame interval → low buffer utilization
- **Terminate head frame early if empty**
  - Use a global barrier network to confirm no pending packet in router or source queue belongs to head frame.
  - Empty buffers are reclaimed much faster and overall throughput increases. (by >30% for hotspot traffic pattern)

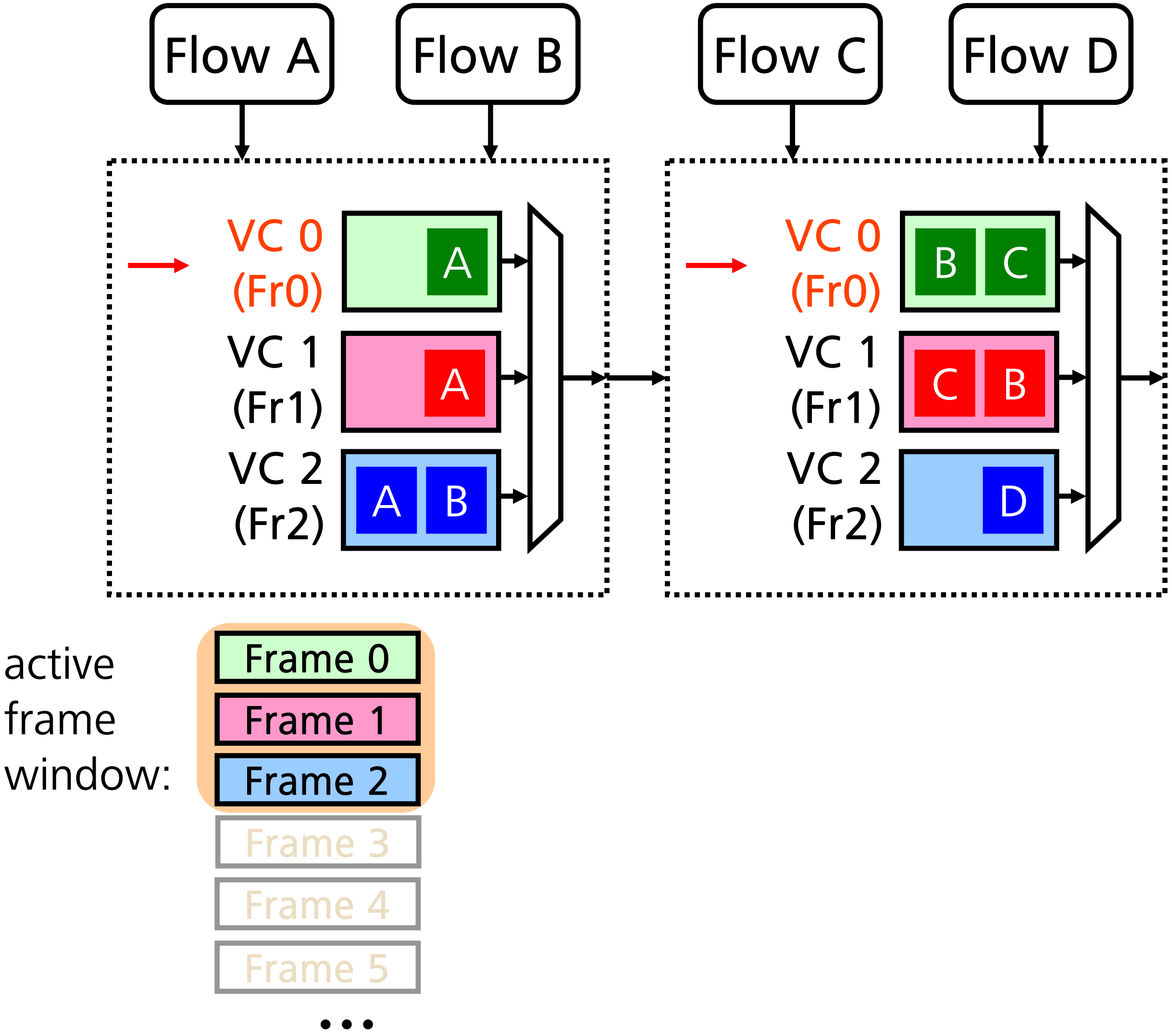
GSF in action

- GSF in action: two-router network example (3 VCs)



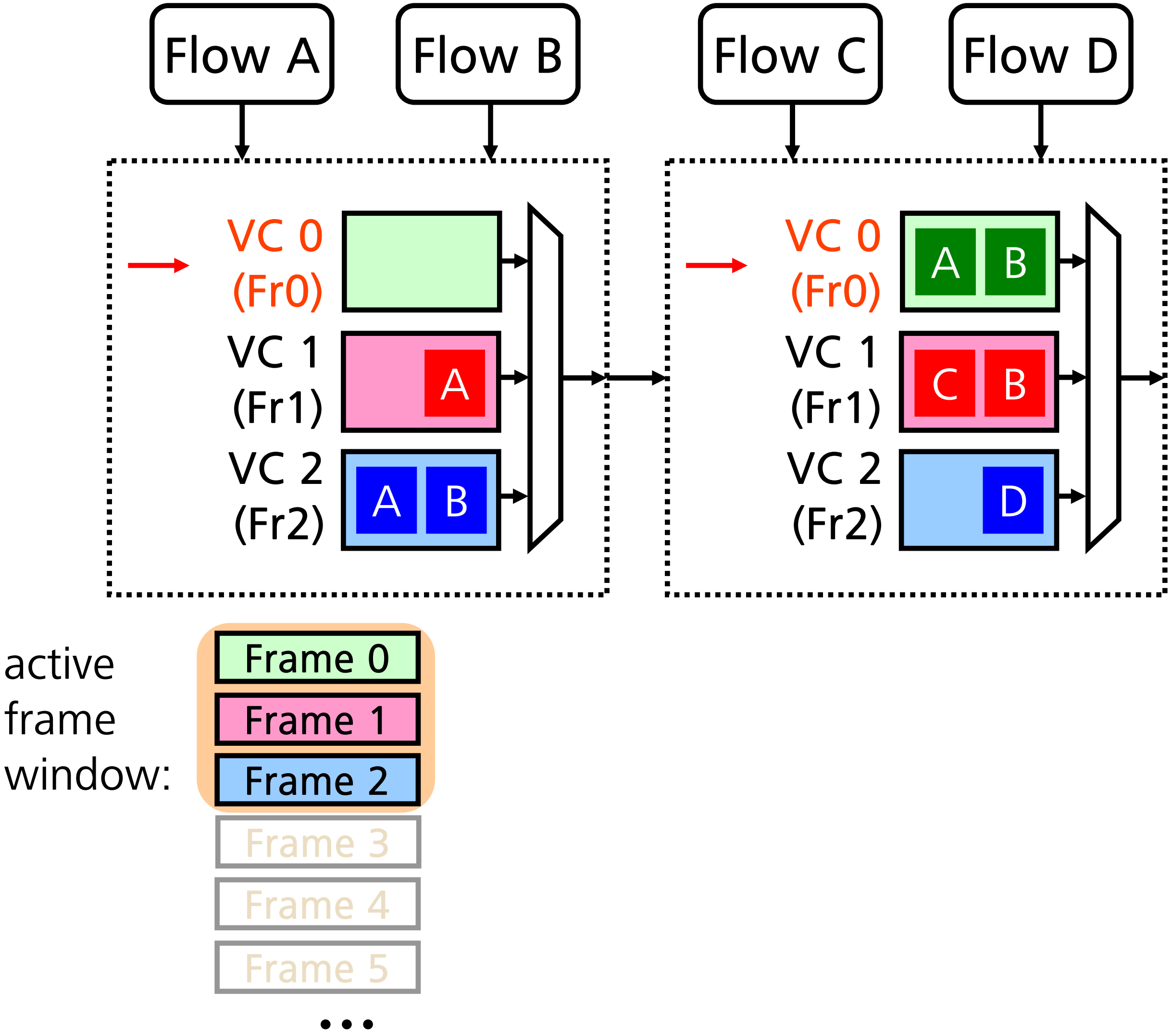
GSF in action

- GSF in action: two-router network example (3 VCs)



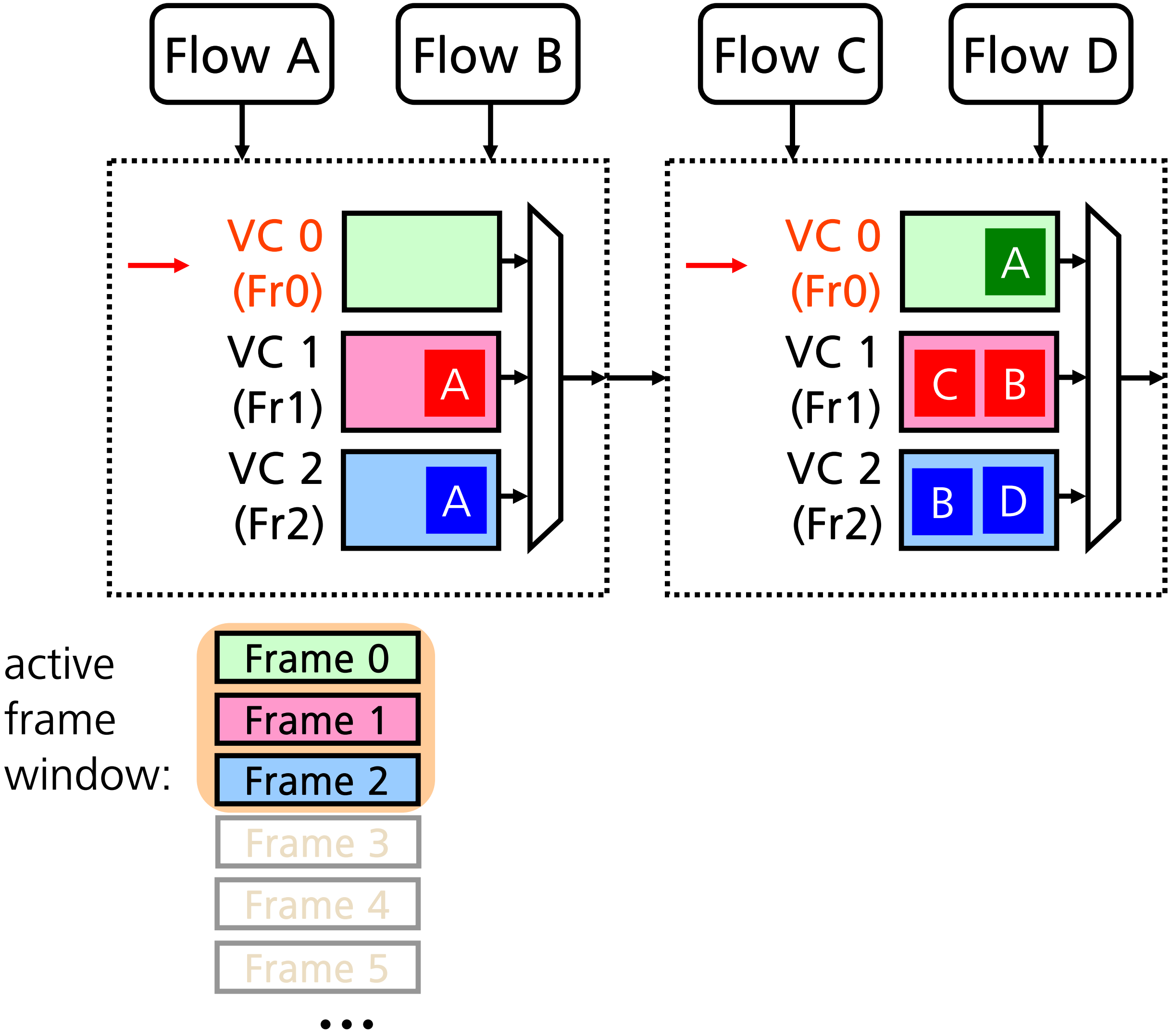
GSF in action

- GSF in action: two-router network example (3 VCs)



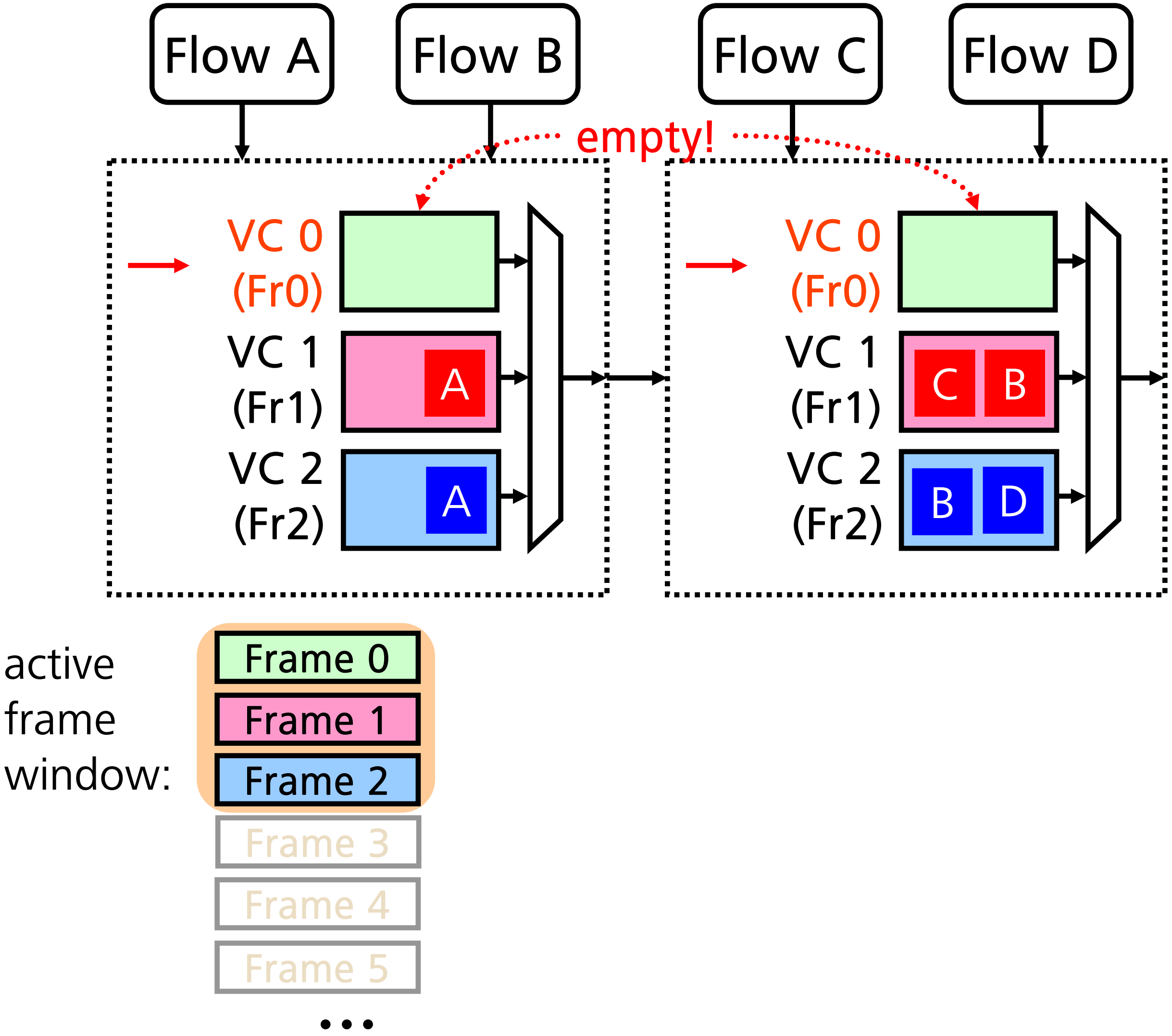
GSF in action

- GSF in action: two-router network example (3 VCs)



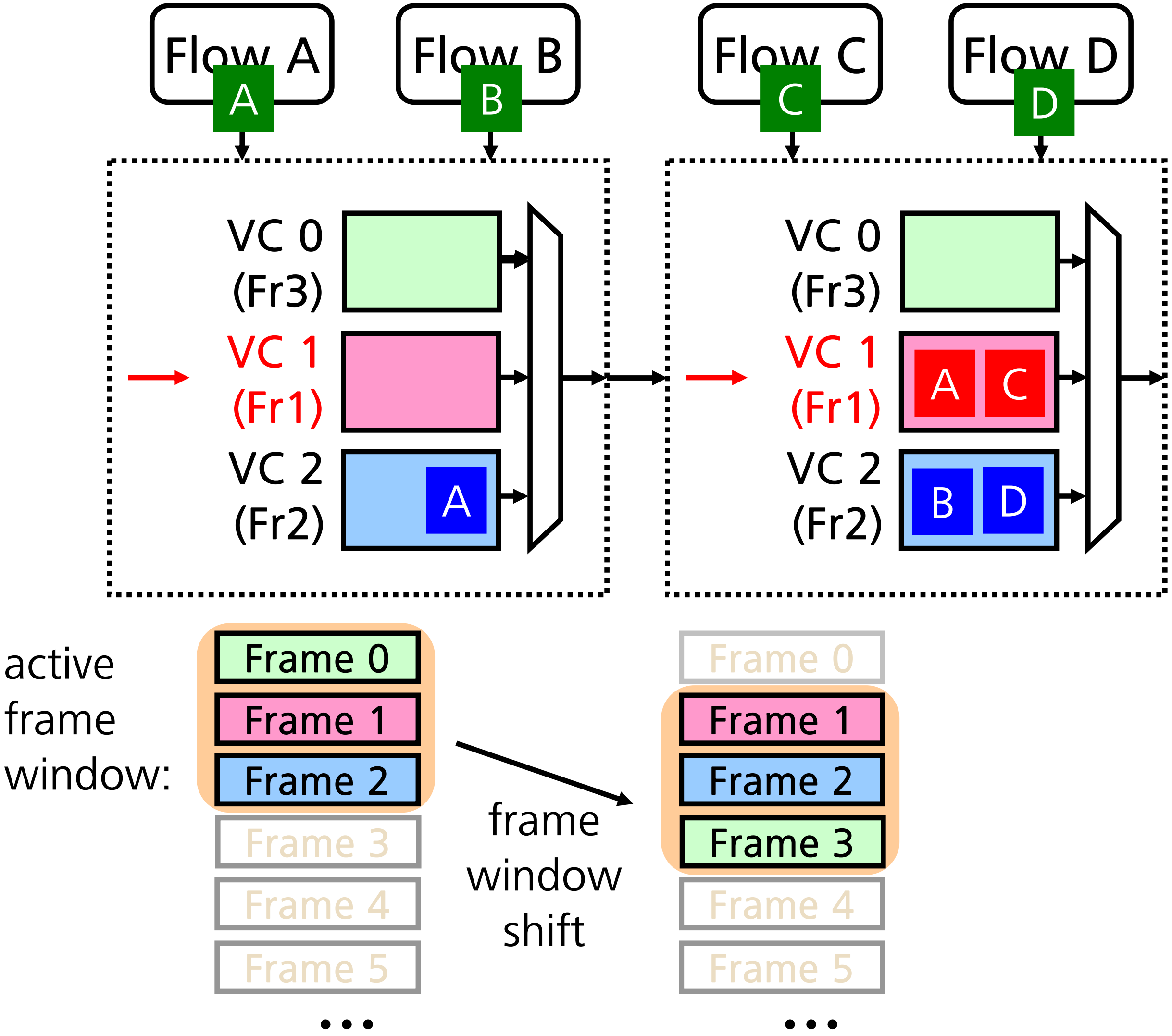
GSF in action

- GSF in action: two-router network example (3 VCs)

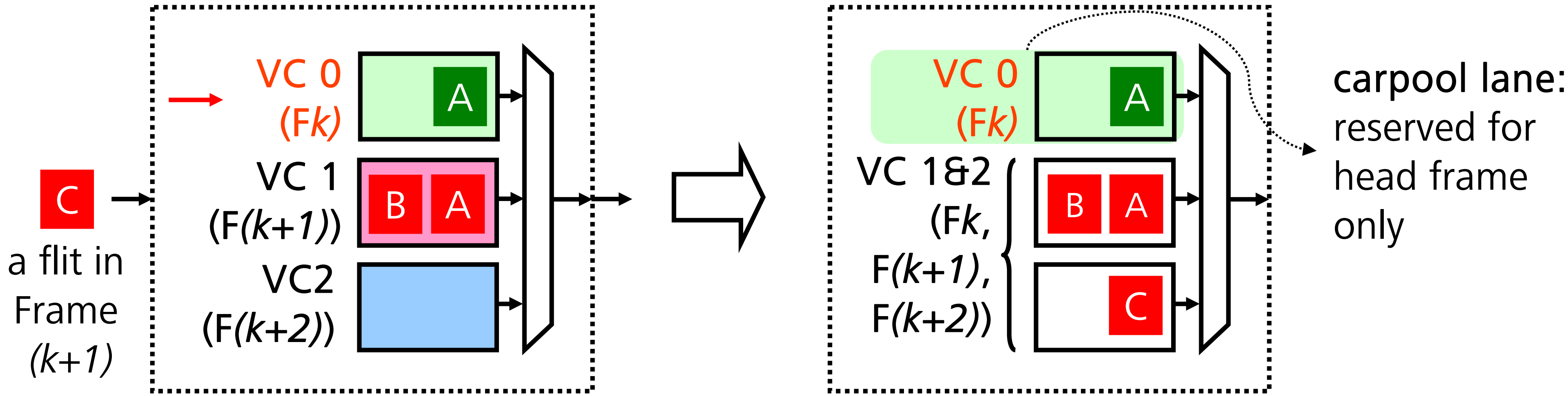


GSF in action

- GSF in action: two-router network example (3 VCs)

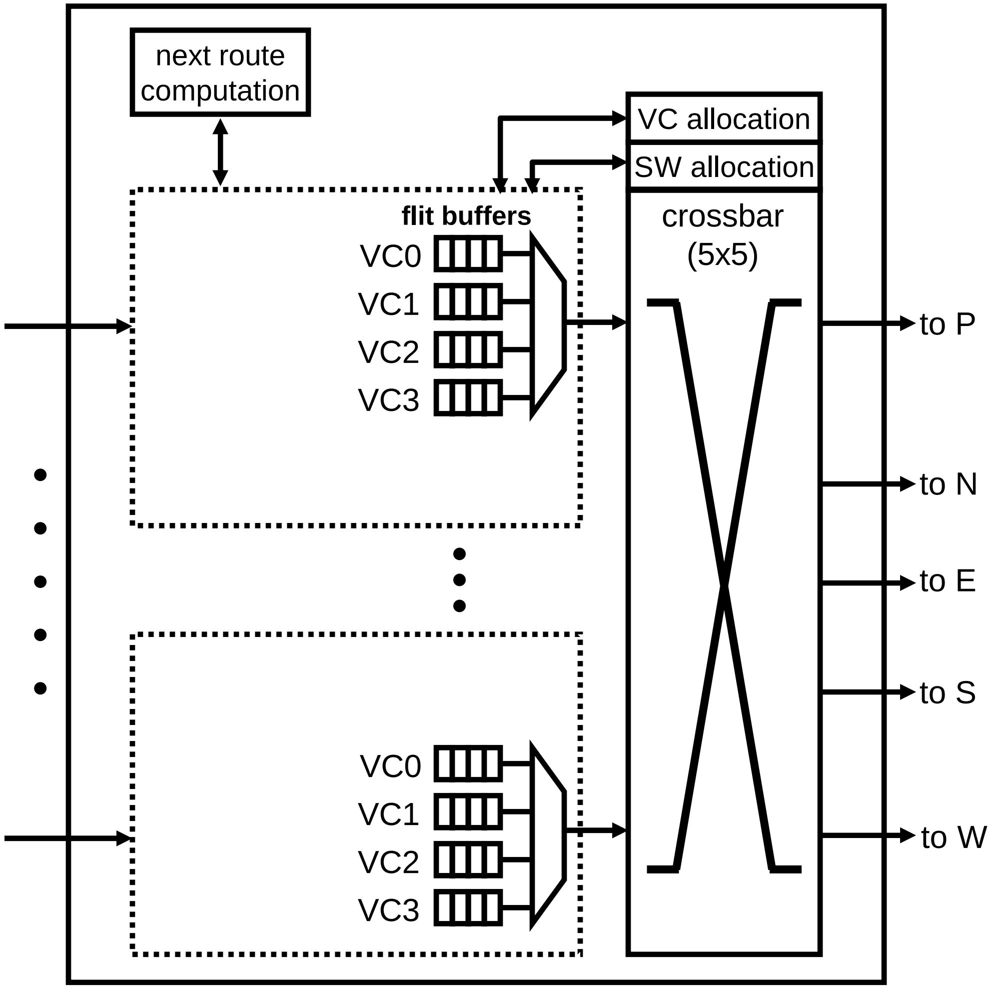


Carpool lane sharing



- Buffers are expensive in on-chip environment.
  - Cannot transport a flit even if there is an empty slot in other frame buffers.
- **Carpool lane sharing**: relaxing frame-VC mapping to improve buffer utilization
  - Reserve one frame buffer (VC0) for head frame only
    → does not increase the drain time of head frame
  - The other buffers are now colorless and can be used by any frame.
- Head-of-line (HoL) blocking prevented by not allowing two packets to occupy a VC simultaneously (OK for shallow buffers).

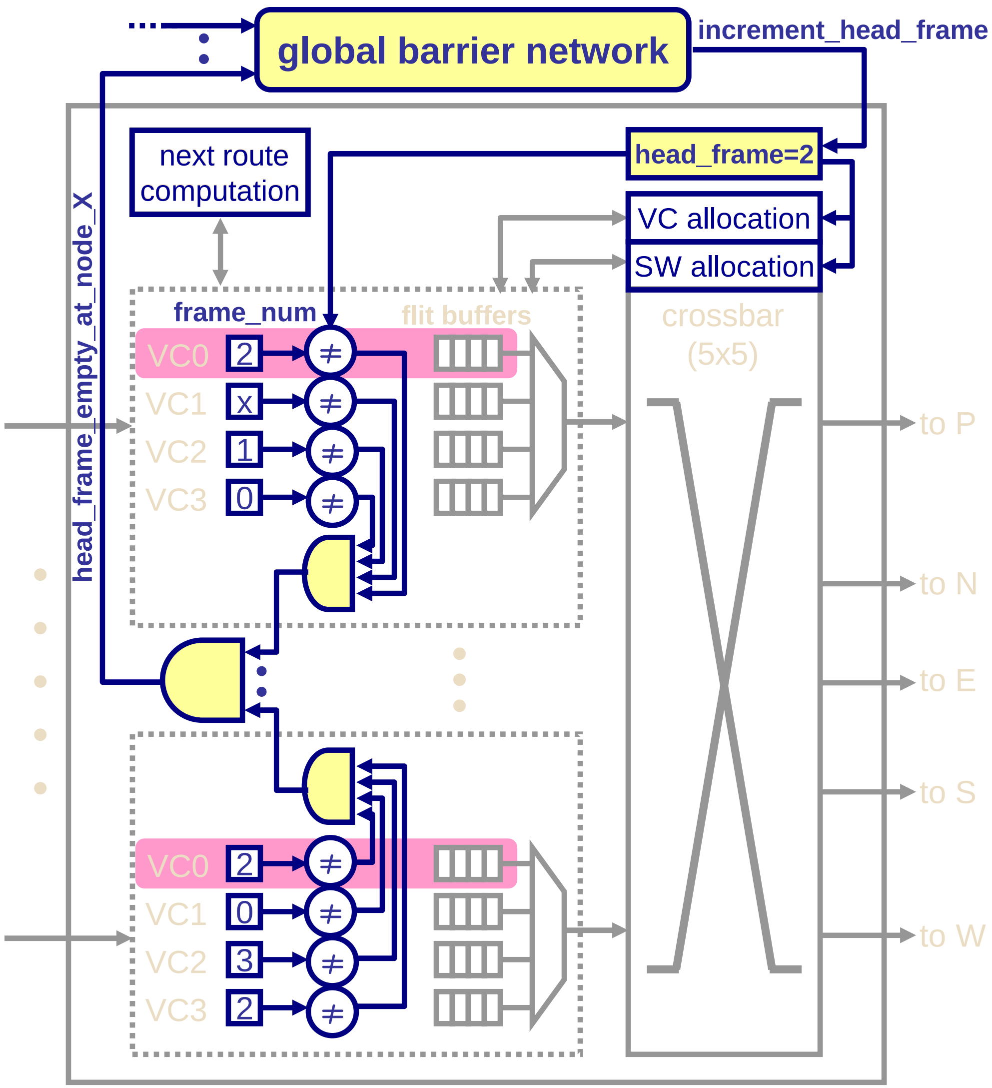
Baseline virtual channel (VC) router



Baseline router for 2D mesh networks

- Best-effort router
- Three-stage pipeline with look-ahead routing: VA/NRC-SA-ST
- Credit-based flow control
- VC, SW allocators: iSlip
  - uses round-robin arbiters (locally fair)
  - updates the priority of each arbiter only when that arbiter generates a winning grant

GSF router



- VC0: carpool lane
  - reserved for head frame only
- New registers
  - head_frame (HF) (per node)
  - frame_num (per VC)
- NRC: priority precalculation

  (frame_num-HF) (mod W)

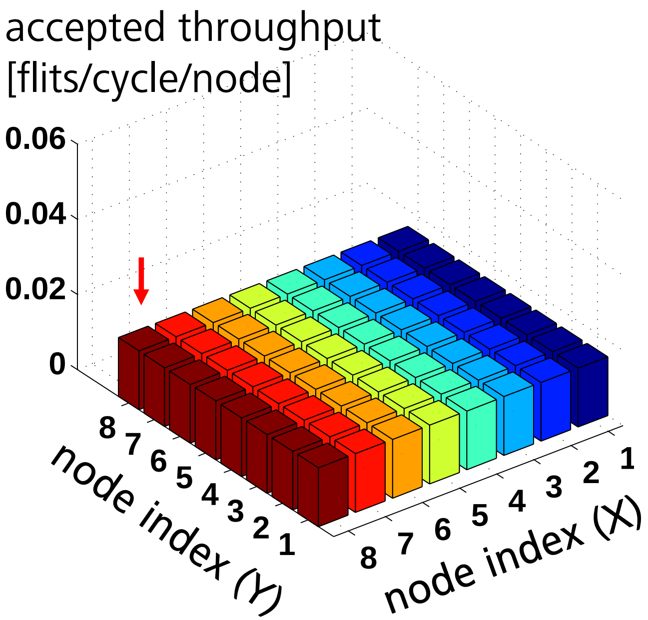
  (0 is the highest priority.)
- VC and SW allocation: priority enforcement
- Global barrier network for frame window shifting

Simulation setup

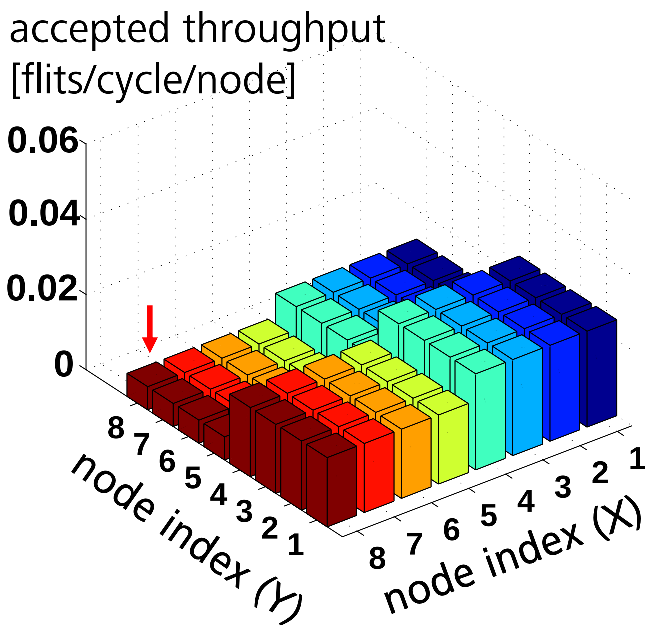
- Network simulator: *Booksim*
  - 0.5 M cycles with 50K-cycle warming up
- Network configuration
  - 8x8 2D mesh, dimension-ordered routing, 1 flit/cycle link capacity
- Four traffic patterns
  - one QoS traffic pattern: hotspot
  - three best-effort traffic patterns: uniform random, transpose, nearest neighbor
  - packet size is either 1 or 9 flits (with 50-50 chance)
- Baseline VC router
  - 3-stage pipeline (VA/NRC-SA-ST), 2-cycle credit pipeline delay
  - 6 VCs/physical link, buffer depth is 5 flits/VC
- GSF parameters
  - frame window size = 6 [frames], frame size = 1,000 [flits]
  - global barrier latency = 16 [cycles] (conservative)

Flexible guaranteed QoS provided

- All flows receive more than their minimum guaranteed bandwidth (R_i/e^{MAX}) in accessing hotspot.
  - R_i: # of flit injection slots for Flow i
  - e^{MAX}: maximum epoch interval.
- Example: 8x8 mesh network



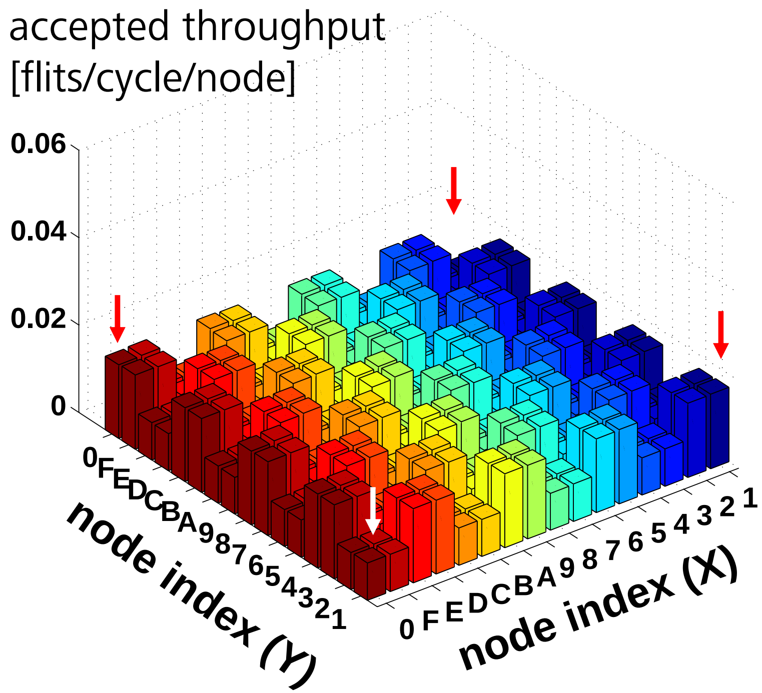
(a) fair allocation



(b) differentiated allocation

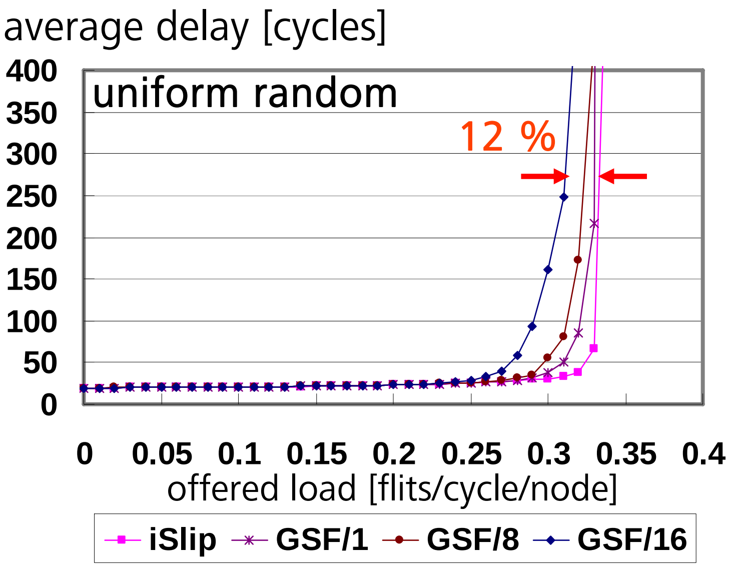
Flexible guaranteed QoS provided

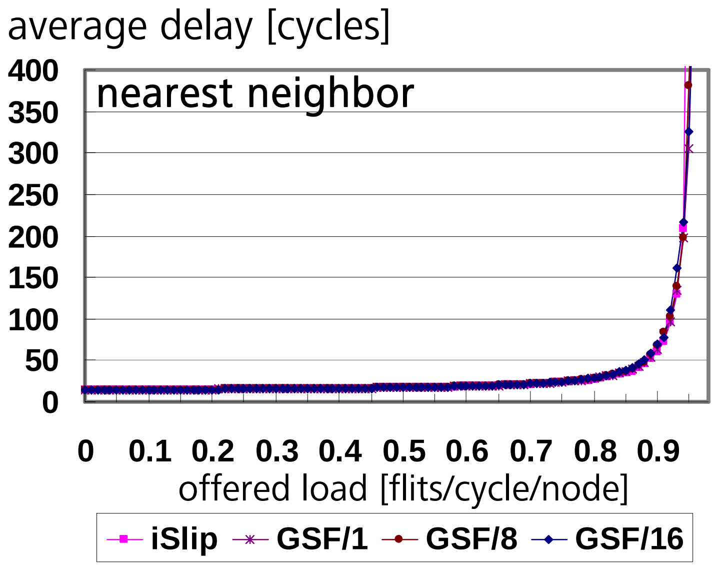
- All flows receive more than their minimum guaranteed bandwidth (R_i/e^{MAX}) in accessing hotspot.
  - R_i: # of flit injection slots for Flow I
  - e^{MAX}: maximum epoch interval.
- Example: 16x16 torus network with 4 hotspot nodes

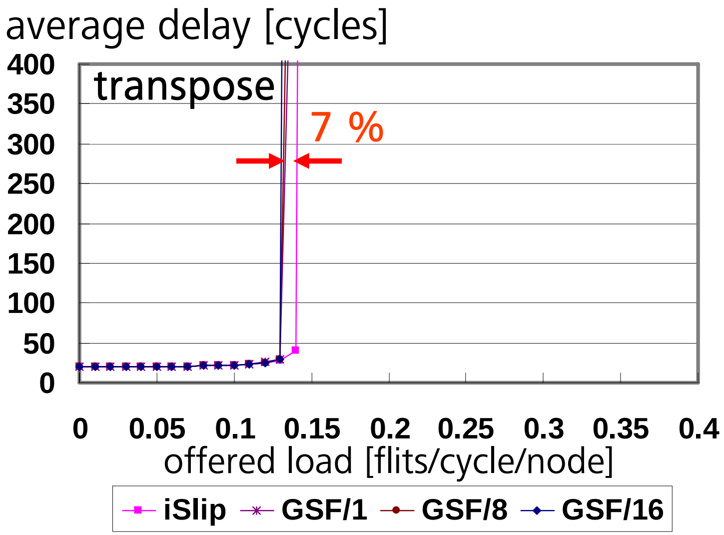


(c) differentiated allocation

Small throughput degradation for best-effort traffic







- Network behavior with non-QoS traffic
  - no latency increase in uncongested region
  - at most 12 % degradation of network saturation throughput → can be reduced with larger frame (at the cost of delay bound increase)

Related work

- QoS support in IP or multiprocessor networks
  - Fair Queueing [SIGCOMM '89],
    Virtual Clock [SIGCOMM '90]
  - Multi-rate channel switching [IEEE Comm '86]
  - Source throttling [HPCA '01]
  - Age-based arbitration [IEEE TPADS '92, SC '07]
  - Rotating Combined Queueing (RCQ) [ISCA '96]

  → *expensive, inflexible, and/or without guaranteed QoS*

- QoS on-chip networks
  - AEthereal (strict TDM; exp. channel setup) [IEEE Design & Test '05]
  - SonicsMX (per-thread queues at each node) [DATE '05]
  - MANGO clockless NoC (partitioning GS and BE VCs) [DATE '05]
  - Nostrum (routes fixed at design time) [DATE '04]

Conclusion

The GSF network is

- guaranteed QoS-capable
  - with minimum bandwidth guarantees and maximum delay
- flexible
  - fair and differentiated bandwidth allocation
  - no explicit channel setup required along the path
- robust
  - <5 % throughput degradation on average (12 % in the worst) for four traffic patterns in 8x8 mesh network
  - fairness vs overall throughput tradeoff with frame size
- simple
  - no per-flow queues/structures in on-chip routers $\rightarrow$ scalable
  - relatively small modifications to a conventional VC router