CSAIL

Bandwidth-Adaptive Network

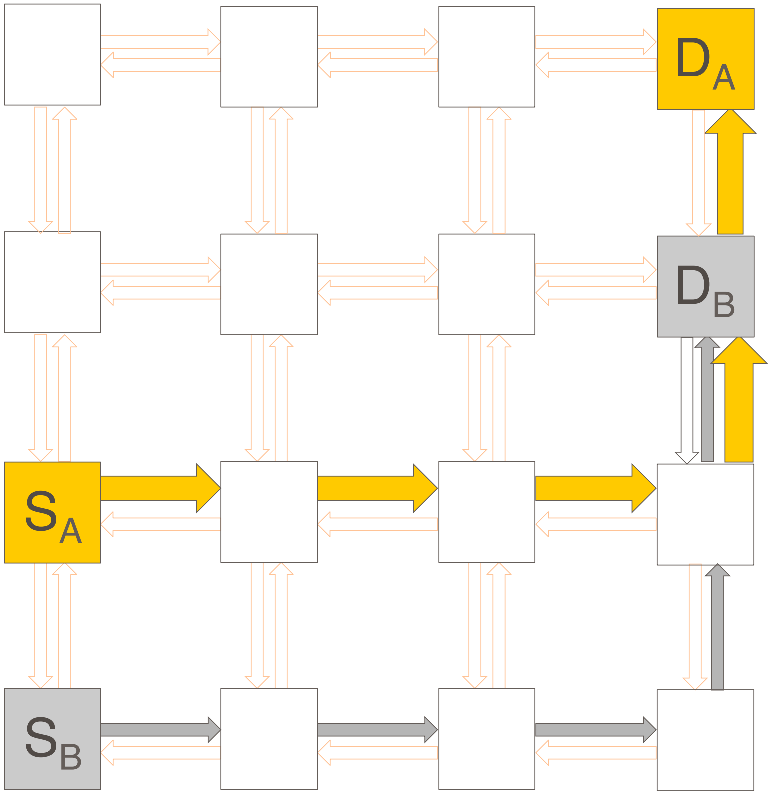
Myong Hyon “Brandon” Cho, Mieszko Lis,
Keun Sup Shim, Michel Kinsy, Tina Wen,
Srini Devadas

Motivation

Oblivious routing vs. Adaptive routing

Oblivious Routing

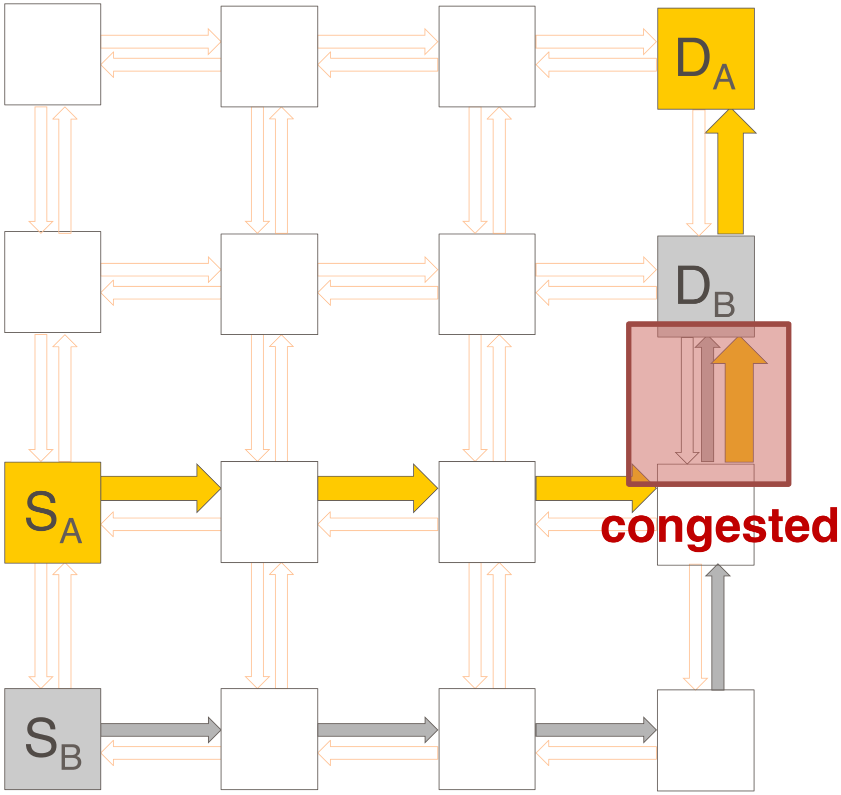
- **Determined by the source and destination locations only**



(+) Simple and fast router designs

Oblivious Routing

- **Determined by the source and destination locations only**

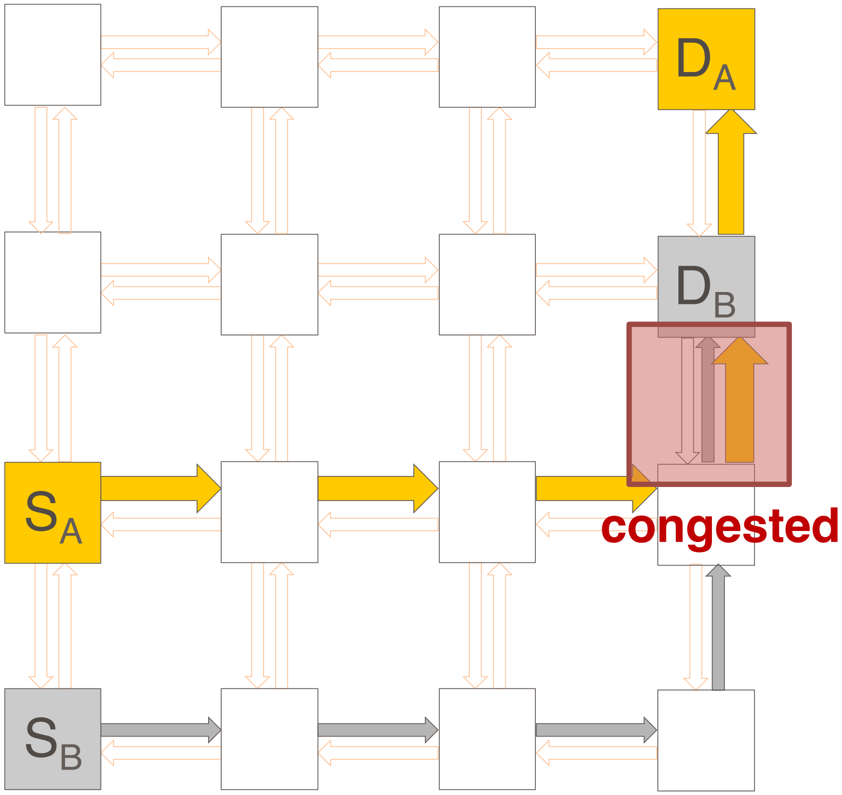


(+) Simple and fast router designs

(-) Inefficient

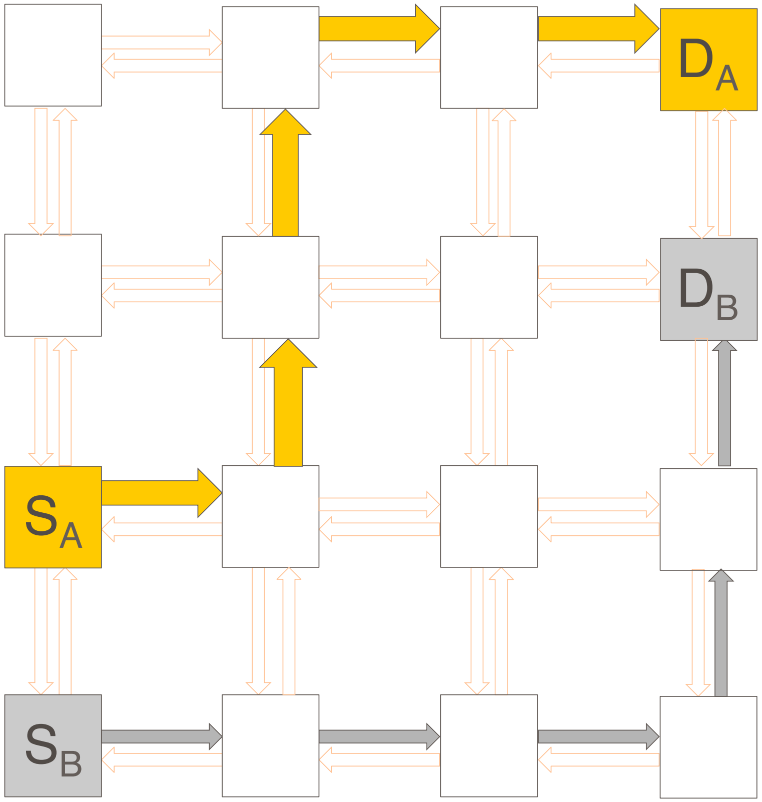
Adaptive Routing

- **Routes are dynamically adjusted to the network status**



Adaptive Routing

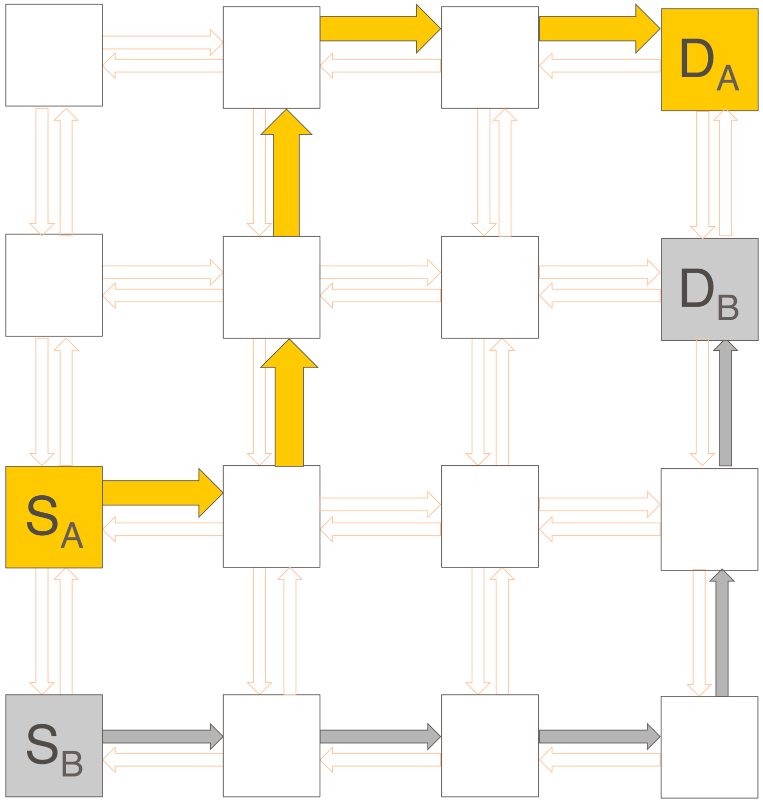
- **Routes are dynamically adjusted to the network status**



(+) Potentially better throughput and latency

Adaptive Routing

- **Routes are dynamically adjusted to the network status**



(+) Potentially better throughput and latency

(-) Complex hardware & logic

Goal

Simple routing steps

Adaptivity using local intelligence

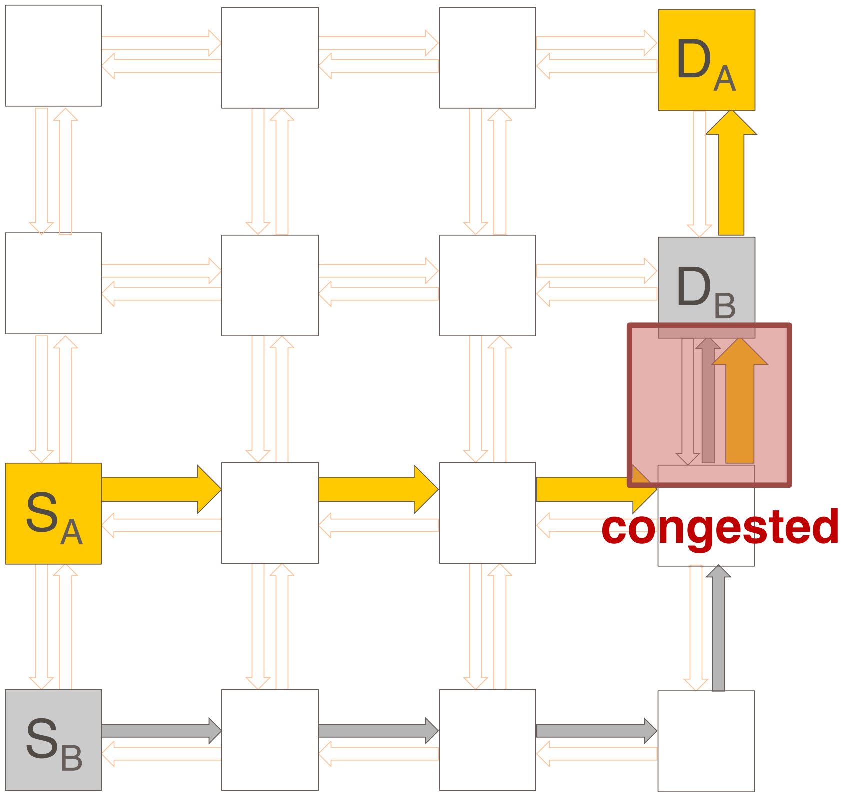
As simple as oblivious routing, as efficient as adaptive routing

Low hardware cost

Functional correctness

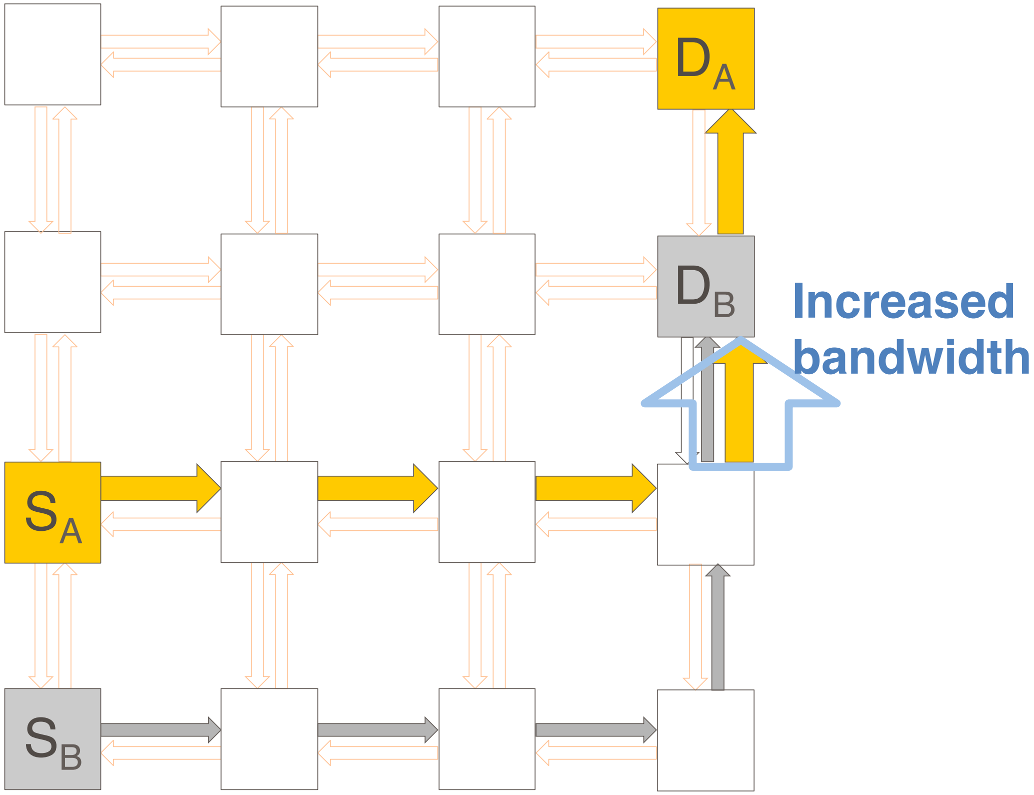
Where can we do better?

Where can we do better?



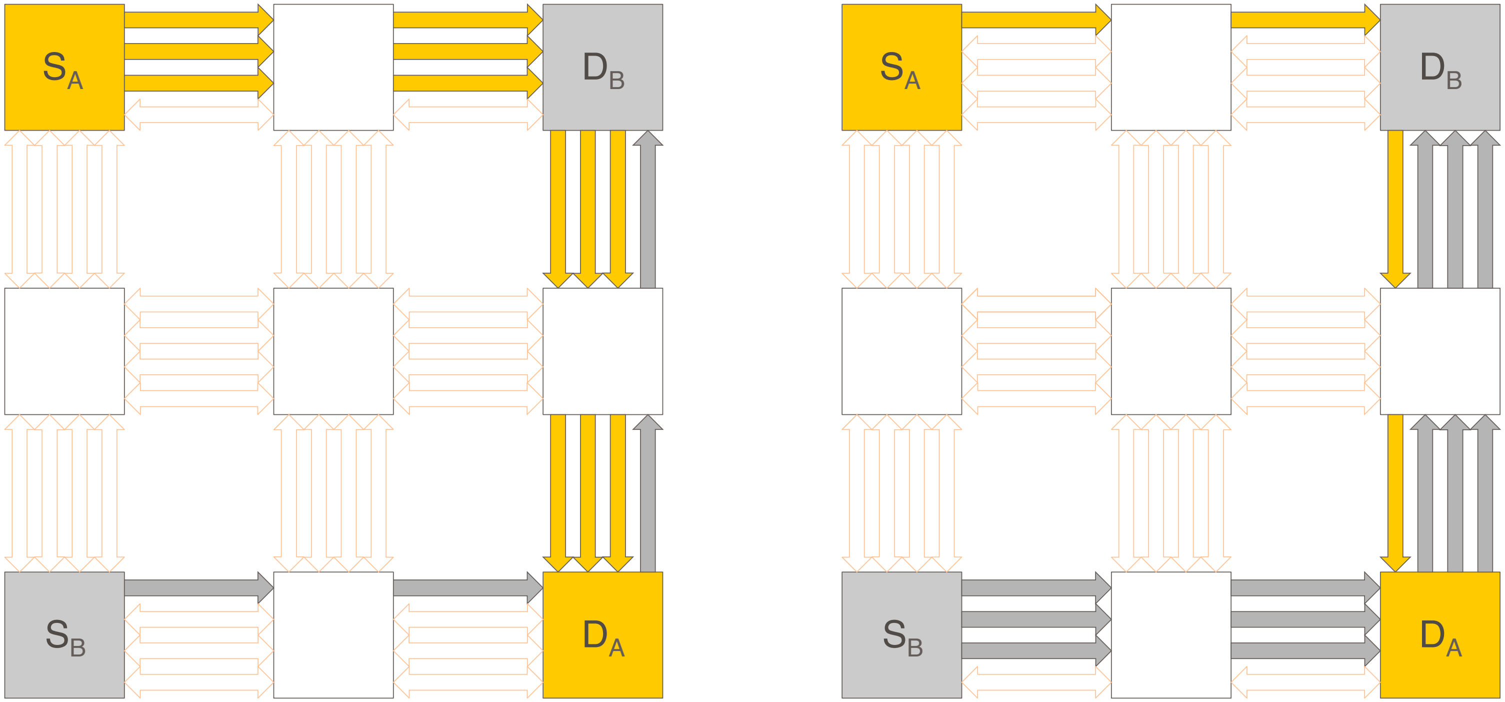
Where can we do better?

Adaptive **Network**, not routing



- A set of bidirectional links connects network nodes.
  - The bandwidth of the link in one direction can be increased at the expense of the other direction.

Adaptive **Network**, not routing



(a) Flow A is dominant

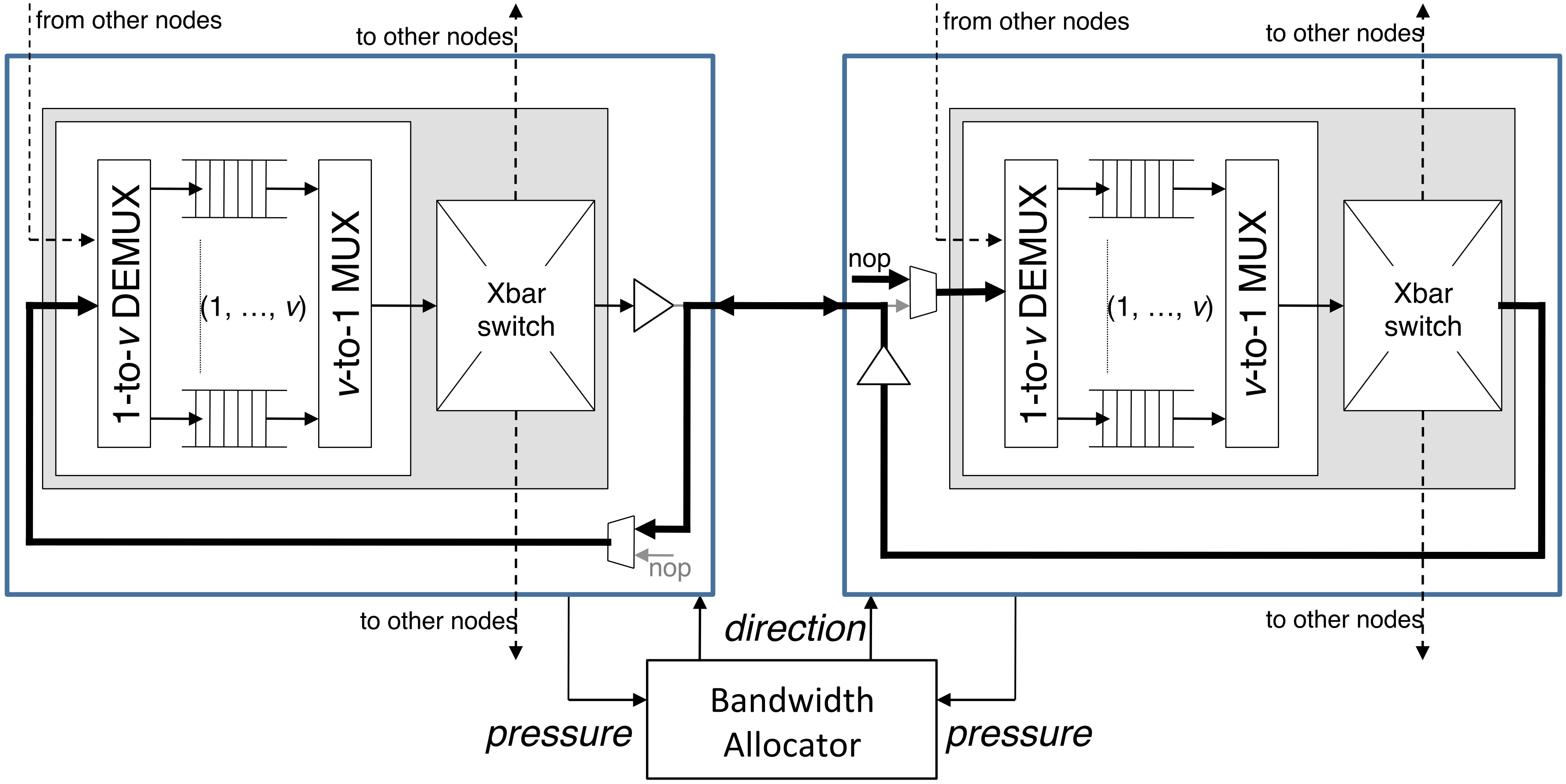
(b) Flow B is dominant

- Routes do not change – Local arbitration logic

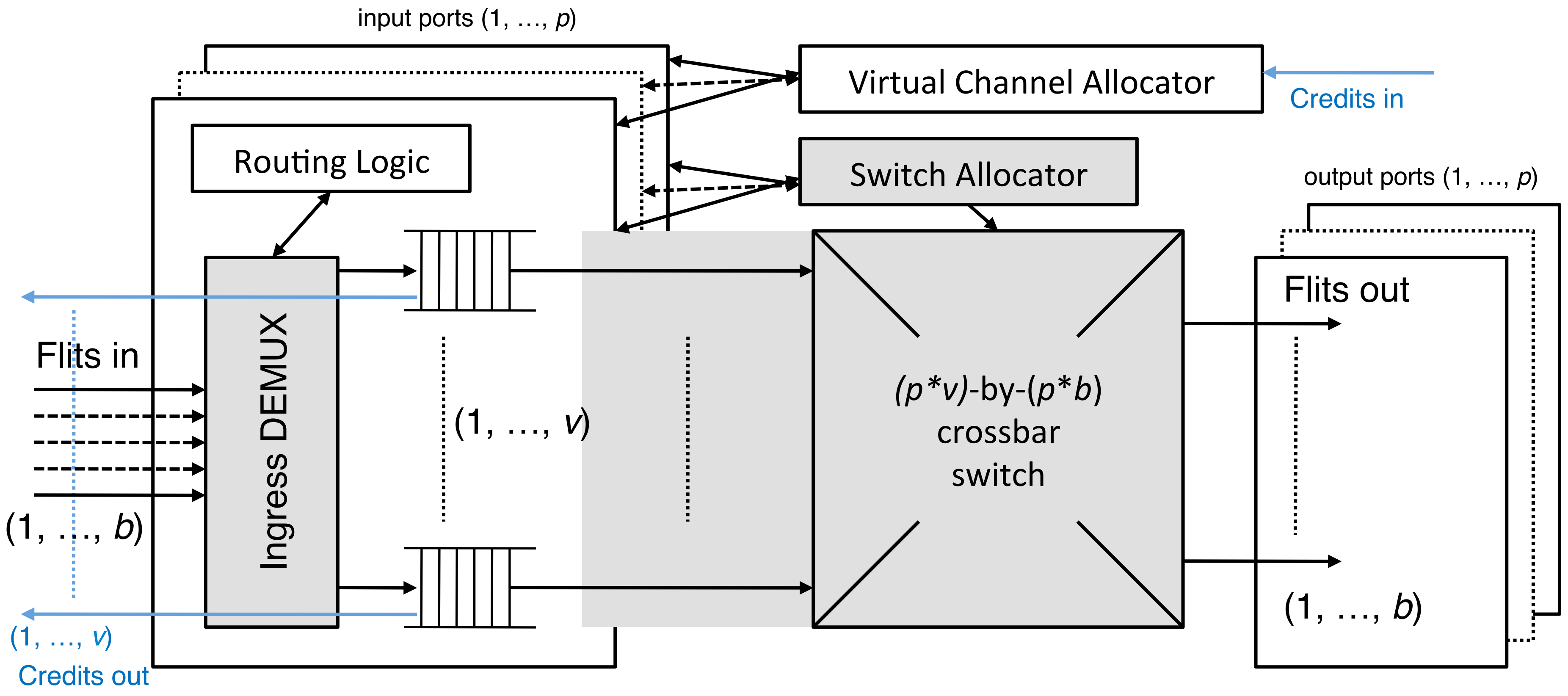
BAN

- Hardware
- Bandwidth allocation
- Deadlock avoidance

BAN connection



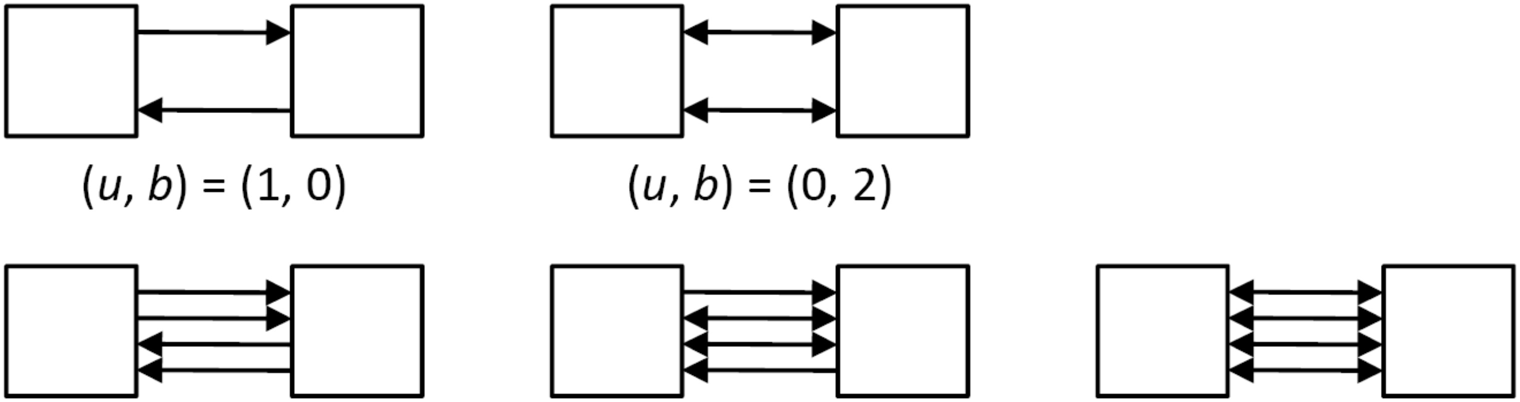
BAN router



- Shaded area is modified from a conventional router.
- The dominant cost is in the crossbar switch.

Hardware cost and Link configuration

- Router architecture can be generalized by:
  - u : the number of unidirectional links
  - b: the number of bidirectional links



$(u, b) = (1, 0)$

$(u, b) = (0, 2)$

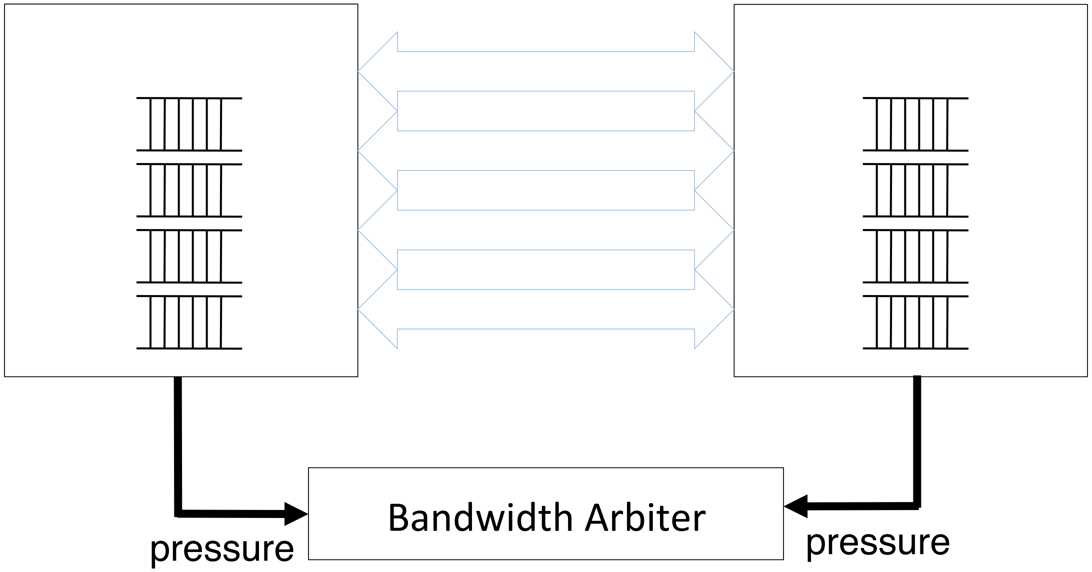
$(u, b) = (2, 0)$

$(u, b) = (1, 2)$

$(u, b) = (0, 4)$

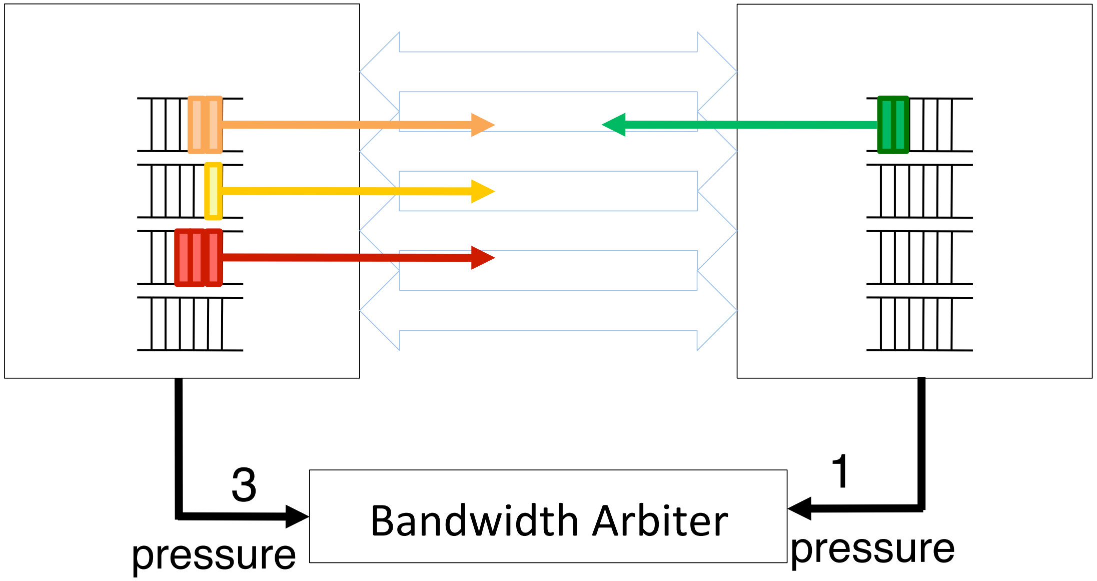
Bandwidth Allocation

- **Local Intelligence**



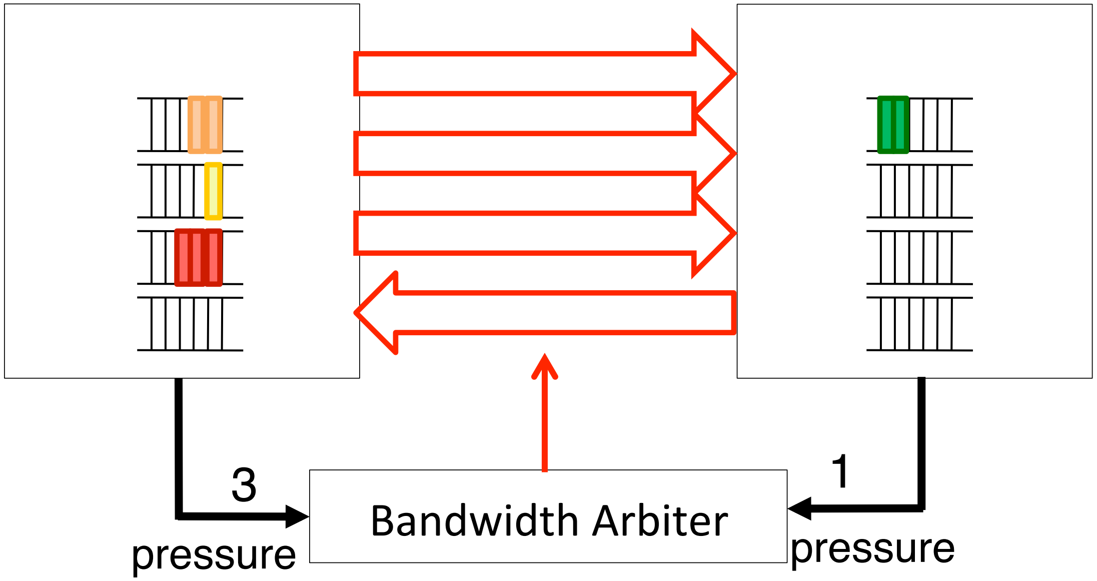
Bandwidth Allocation

- **Local Intelligence**



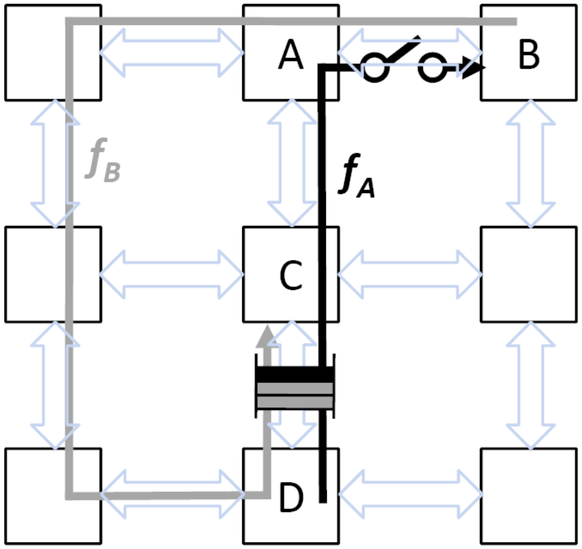
Bandwidth Allocation

- **Local Intelligence**



Bandwidth Allocation

- **A deadlock scenario**



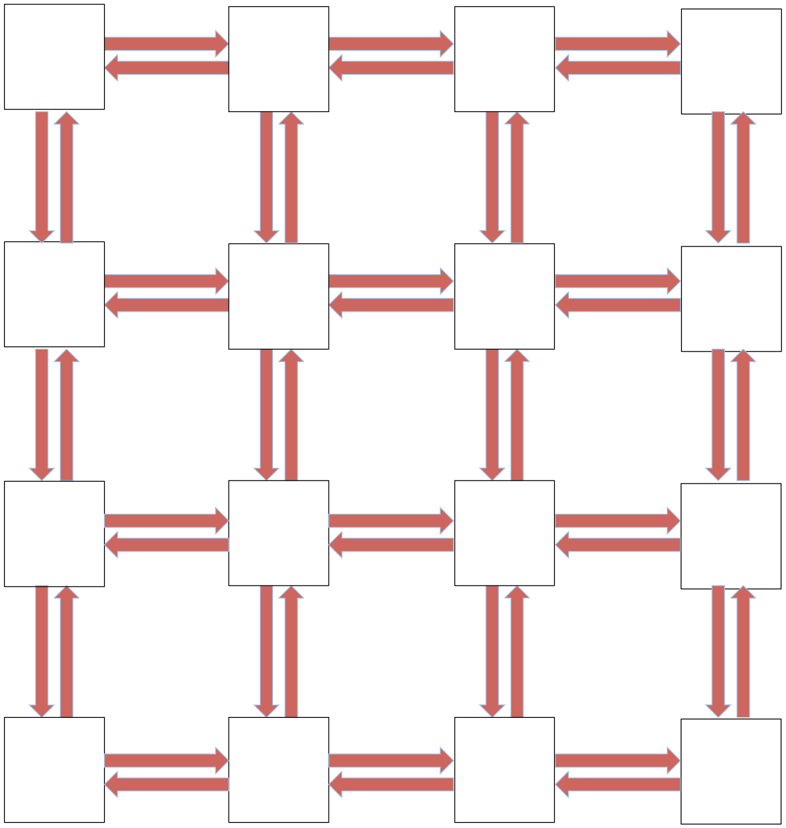
Never entirely block one direction if there is any flit which can move in that direction.

CSAIL

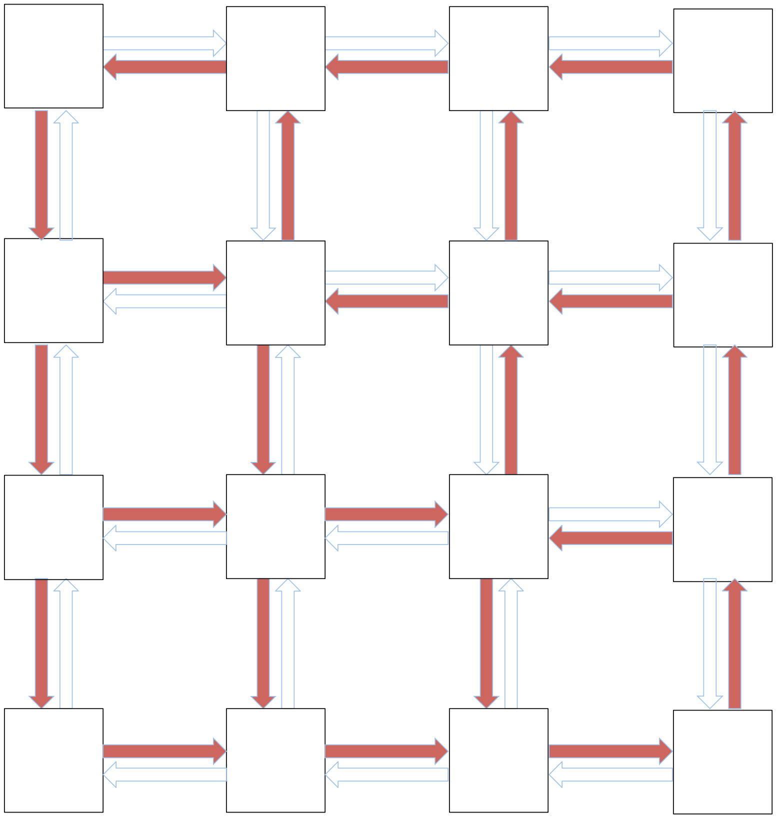
Performance Evaluation

Synthetic workloads

Symmetry vs. Anti-symmetry

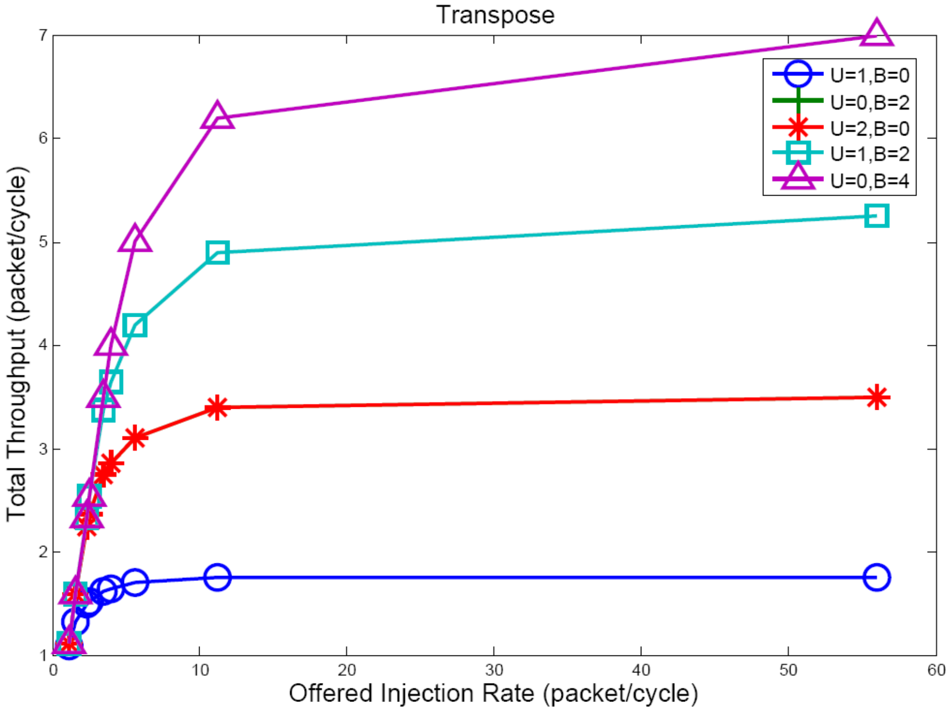


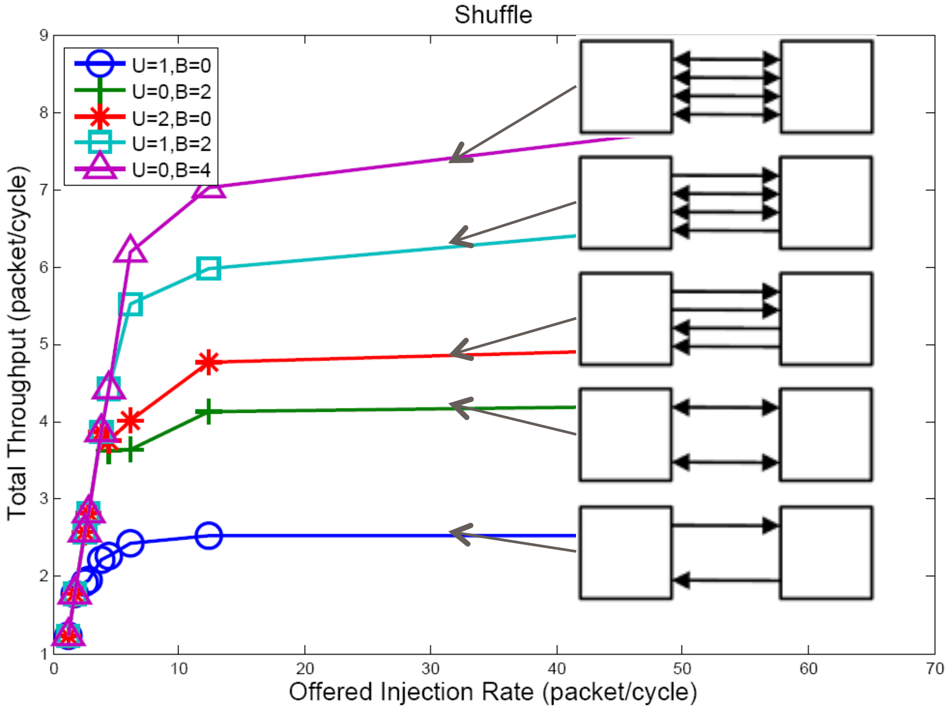
Bit-compliment



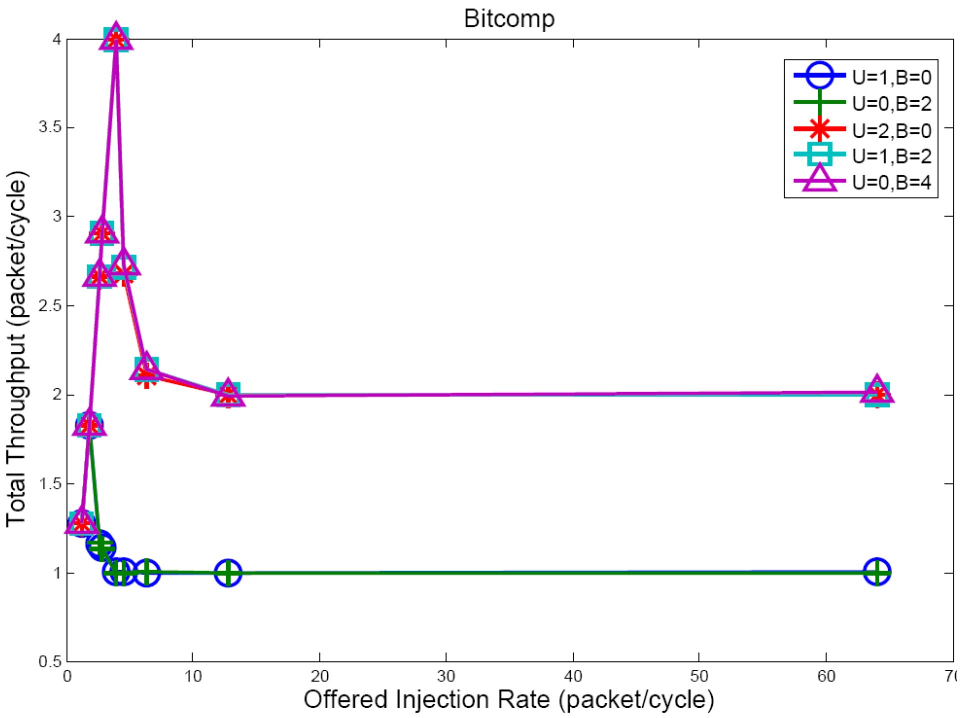
Transpose

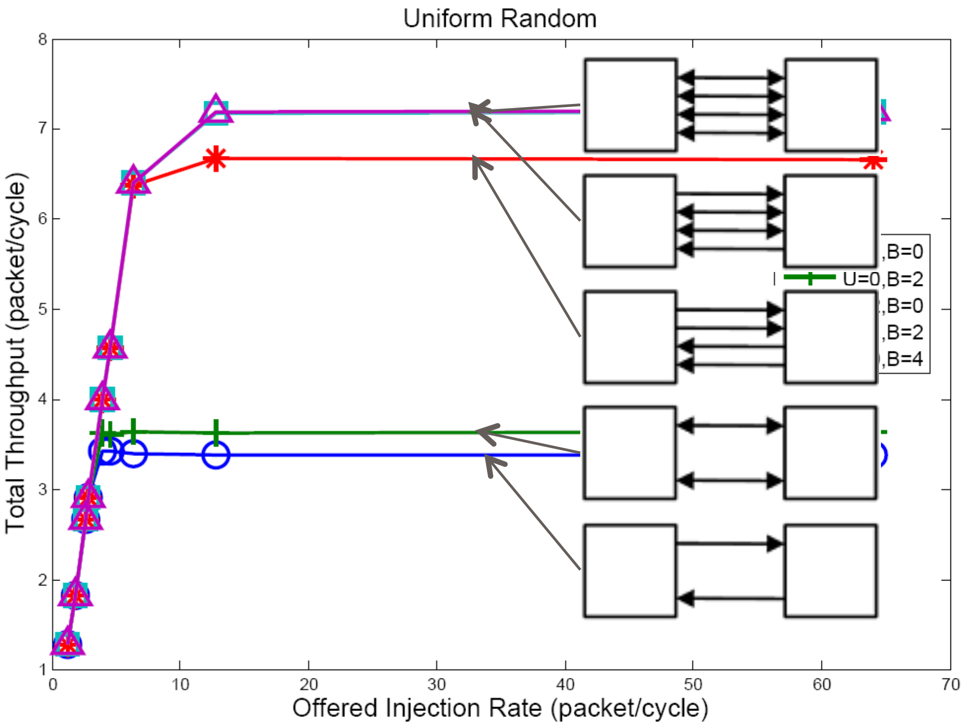
Anti-symmetry synthetic traffic



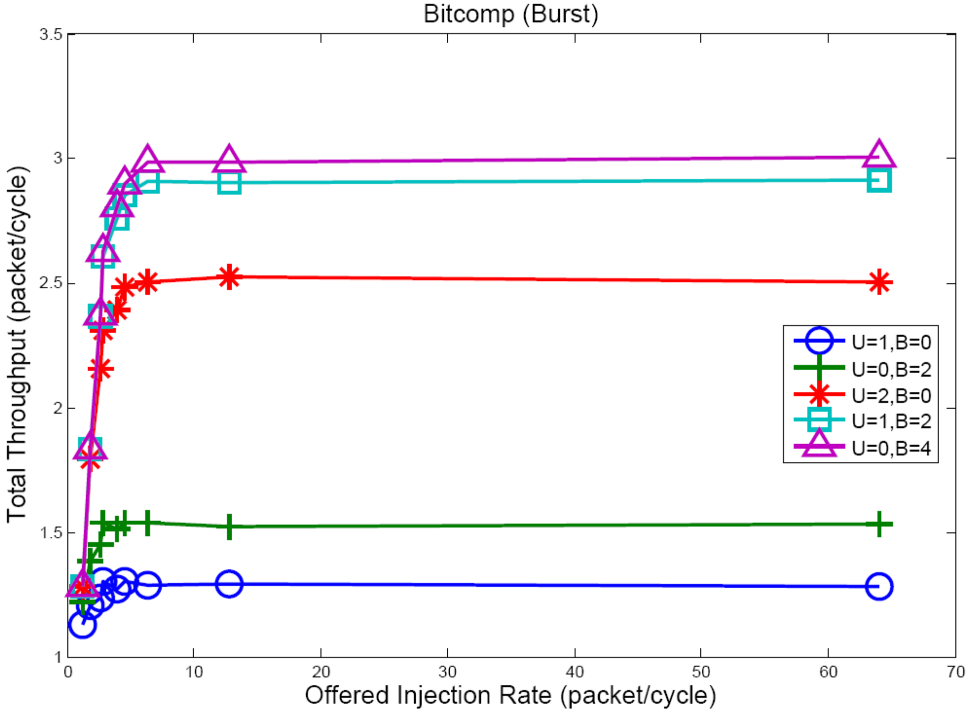


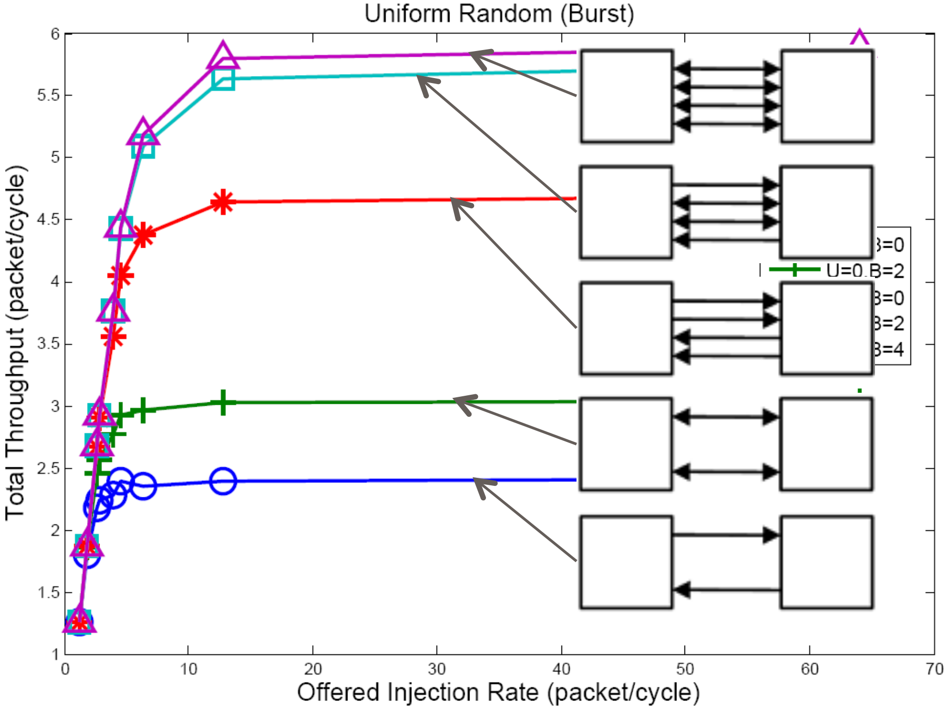
Symmetry synthetic traffic





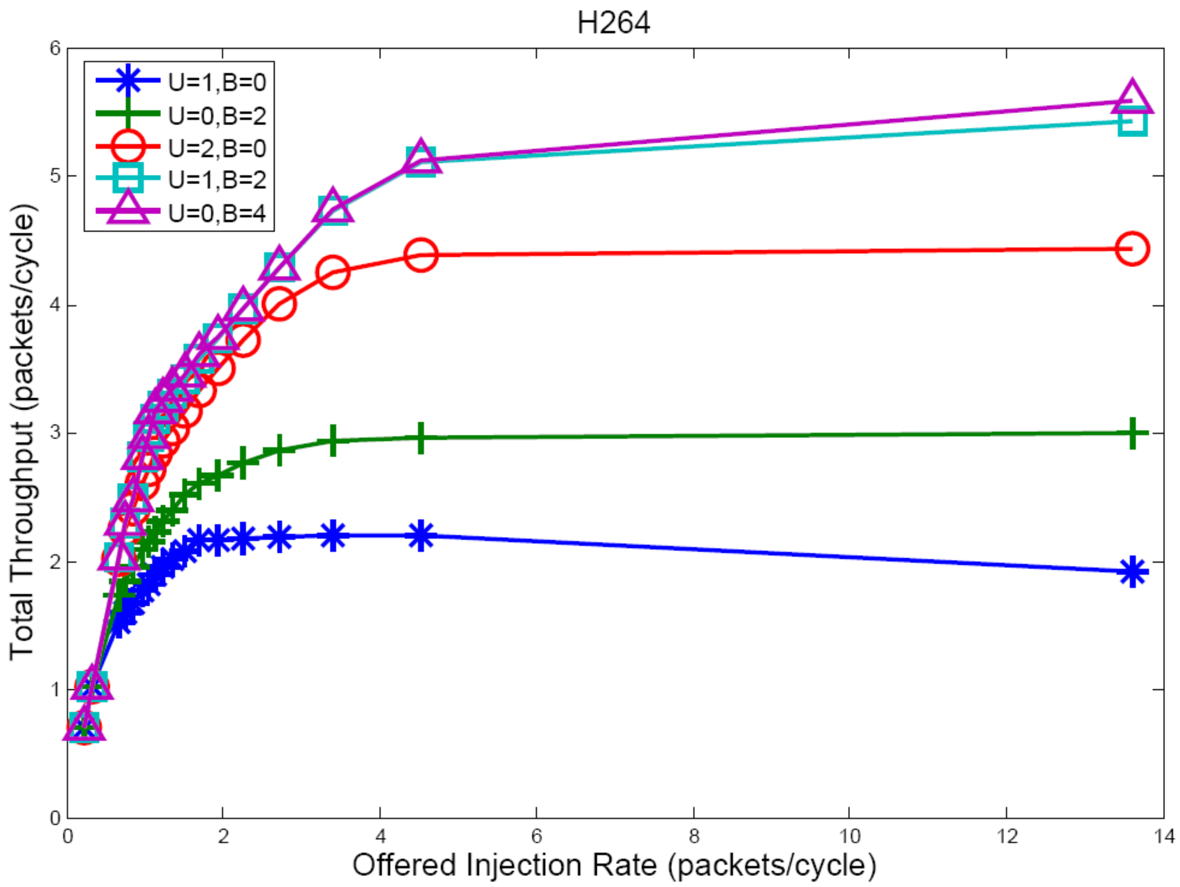
Symmetry synthetic bursty traffic





Traffic Pattern	Non-bursty	Bursty
Bit-compliment	0%	20%
Uniform Random	8%	26%

How about real application traffic…?



- *The traffic patterns in many real applications are not symmetric as data is processed by a sequence of modules.*

The saga of performance evaluation

- H.264 traffic was profiled from an ASIC decoder.

- An important question to ask : what would be the bandwidth demand on many-core general-purpose CPUs?

- DARSIM has evolved to run applications on general-purpose CPU with cycle-level network simulation.

Thank you!