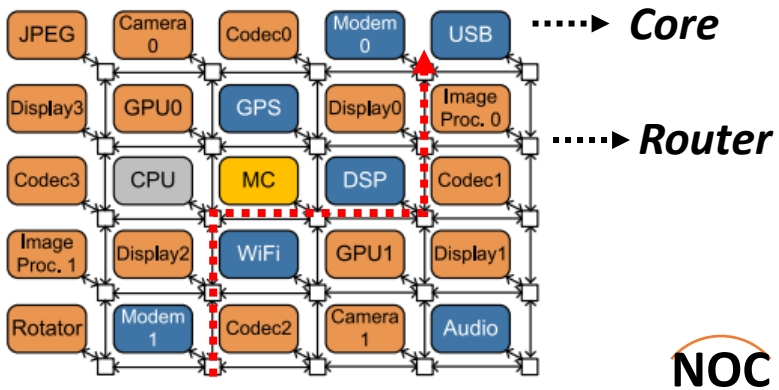
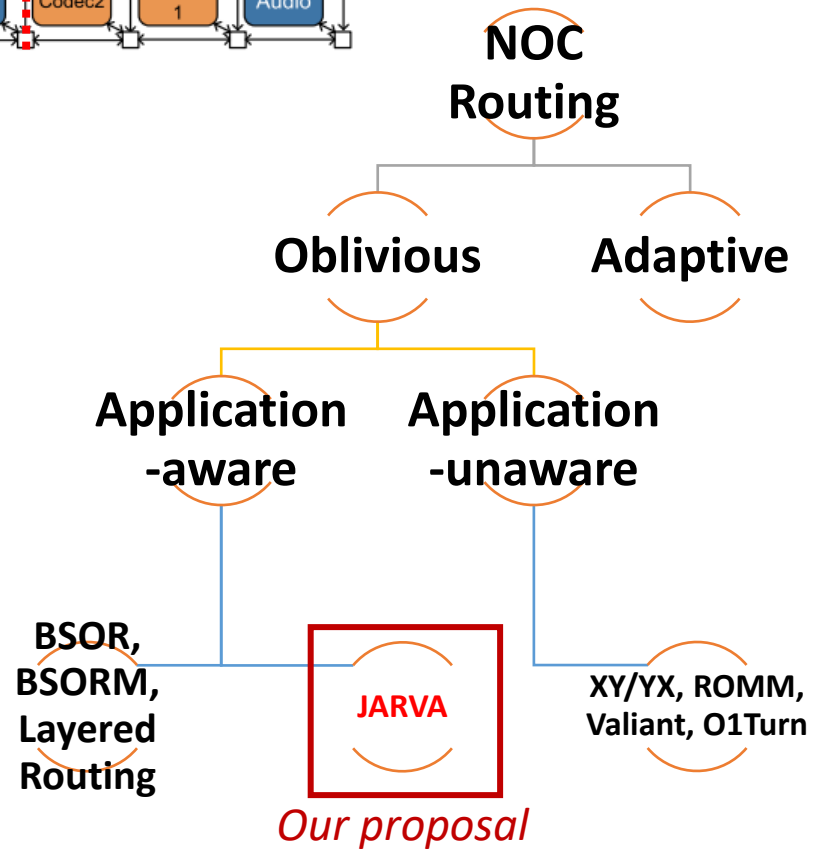
JARVA: Joint Application-Aware Oblivious Routing and Static Virtual Channel Allocation

Uday Mallappa, Chung-Kuan Cheng and Bill Lin

University of California, San Diego

Routing in Network-on-Chip (NOC)





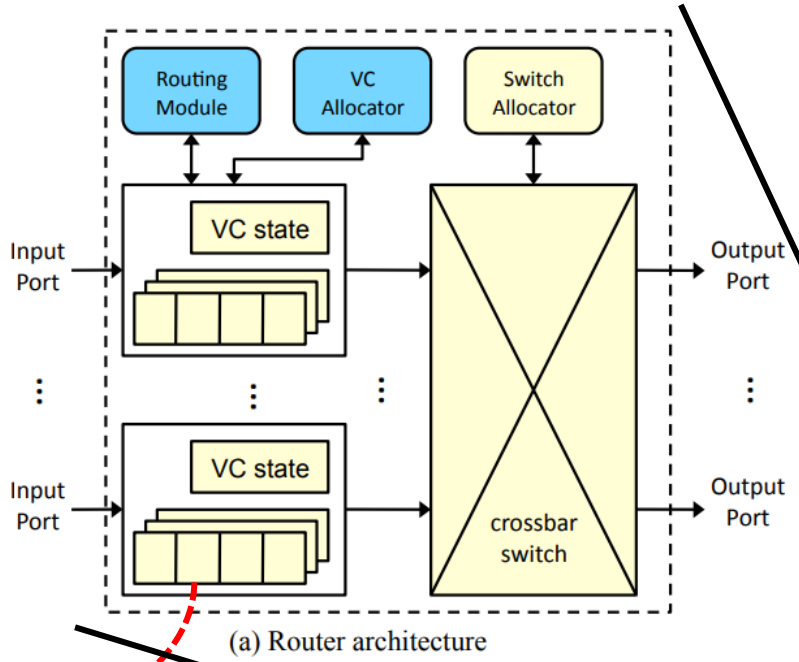
- Routing algorithm: Systematic derivation of a path between any source and destination pair in the network
- Oblivious: Path is completely determined by the source and the destination address
- Adaptive: Considers the network state when making routing decisions dynamically
- Application-aware: Statically determine deadlock-free routes considering an application's communication characteristics

Terminology

- A M x N 2D Mesh network: Represented by a directed graph G = (V, E), where each node u ∈ V corresponds to a processing core and an associated router

- Packet: A unit of flow that needs to be transferred from a source to destination node
- Flits: A packet is broken into flits – flits do not contain additional headers
- Demand: Number of flow units (or packets) that need to be transferred from a source to destination node
- Channel load: Sum of packets that physically utilize a specific channel

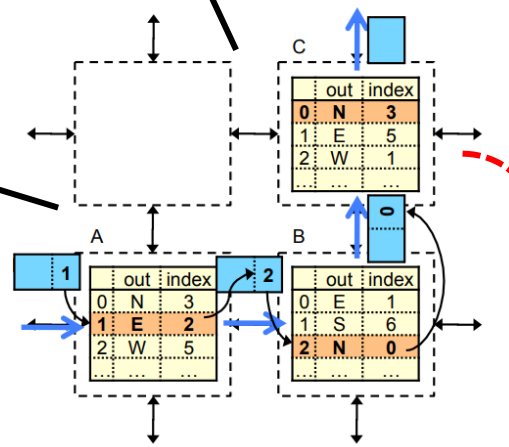
Router Architecture



(a) Router architecture

- A typical router pipeline:
  - **RC** (Routing Computation)
  - **VC** (Virtual Channel Allocation)
  - **SA** (Switch Allocation)
  - **ST** (Switch Traversal)

Virtual Channel (VC): *Multiple input buffers for each physical channel so that flits can flow as if there are multiple "virtual" channels*

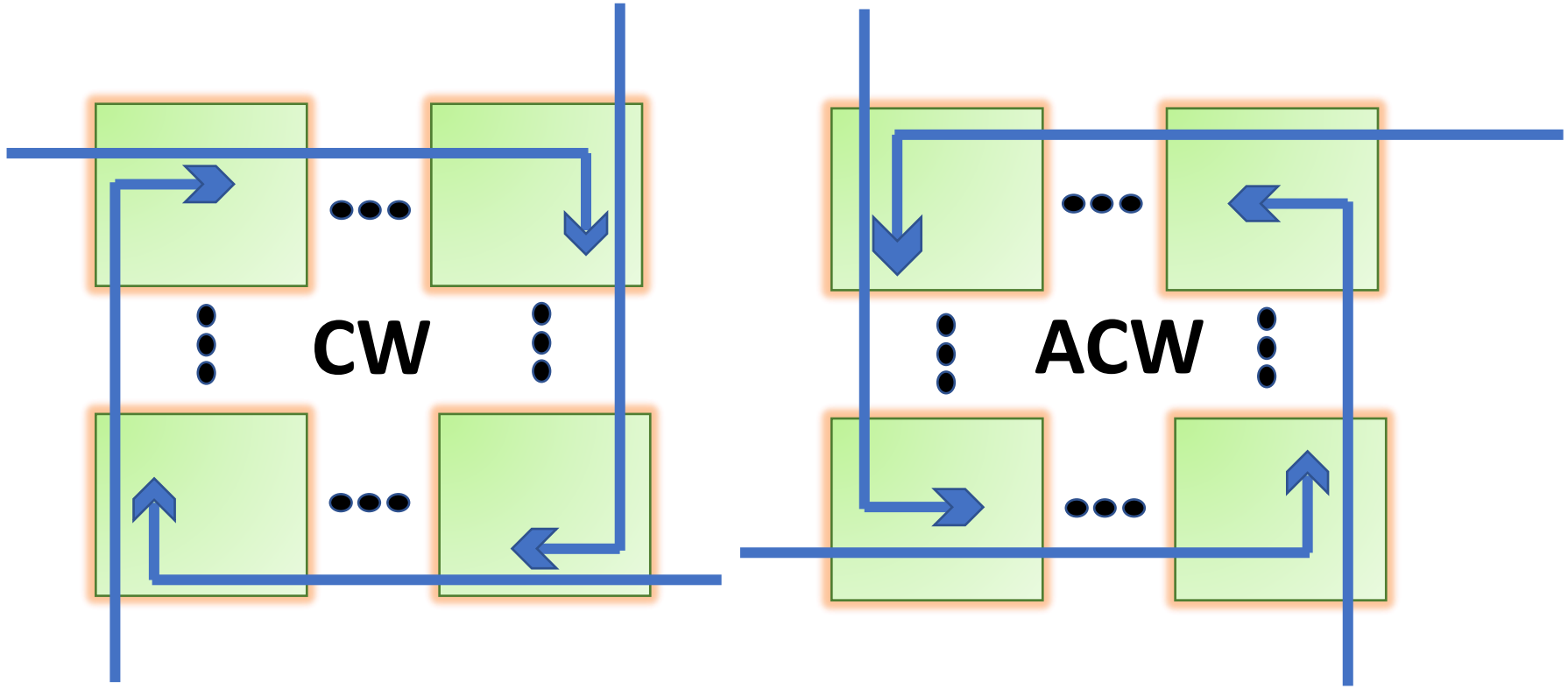


We assume Table-based routing to implement JARVA

M. A. Kinsy et al., "Optimal and heuristic application-aware oblivious routing," IEEE Transactions on Computers, 2013.

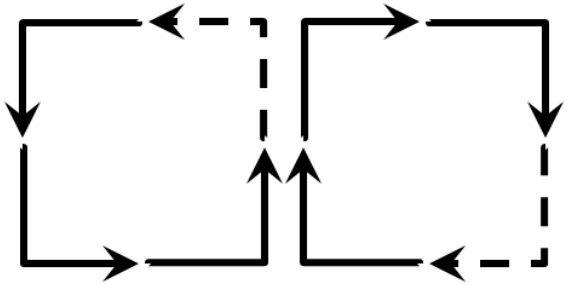
Routing Deadlocks

Deadlock: A cyclic buffer-dependency of routes such that the forward progress of the routing packets becomes impossible

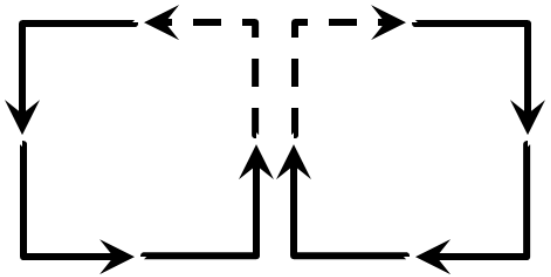


- ❑ **Deadlocks are common both in off-chip and on-chip communication networks**
- ❑ **Correct-by-construct routing algorithms can avoid deadlock-fostered system breakdowns**

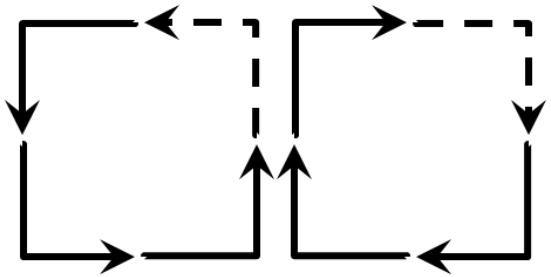
Turn Models



West-First



North-Last



Negative-First

- C. J. Glass et al. proposed a systematic way of generating deadlock-free routes
- The idea is to avoid cyclic dependencies by restricting certain flow turns
- A total of 12 turn models (3 basic restrictions x 4 rotations per restriction)
- If a set of routes conform to one of the 12 turn models, deadlock freedom is assured with any number of virtual channels

Problem statement

- Given
  - A set of K flows, indexed by $i = 1, 2, ..., K$
  - Each flow i is defined by (s_i, t_i, d_i), where s_i and t_i are the source and destination, respectively, and d_i is the demand.
  - We assume $s_i \neq t_i$ and multiple flows may have the same source and destination.
- A route for a flow i is a path from s_i to t_i.

- Objective: Find a deadlock-free routing and VC assignment solution such that maximum channel load of the network is optimal

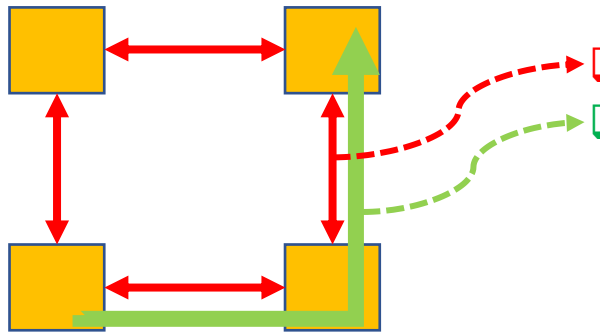
Prior solutions

- Problem statement: Deadlock-free oblivious routing with known communication characteristics
- Strict order (XY/YX, ROMM, Valiant, O1Turn): Application independent and sub-optimal
  - Dally and Seitz., "Deadlock-free message routing in multiprocessor interconnection networks"
  - T. Nesson and J. S. Lennart, "Romm routing on mesh and torus networks"
  - L. G. Valiant and G. J. Brebner, "Universal schemes for parallel communication"
  - D. Seo et al., "Near-optimal worst-case throughput routing for two-dimensional mesh networks"
- Heuristic (BSOR and BSORM) and exact methods (MILP): Restriction on flow turns
  - M. A. Kinsy et al., "Optimal and heuristic application-aware oblivious routing"
- Layered routing: Sequential routing and VC assignment
  - O. Lysne et al., "Layered routing in irregular networks"

- **JARVA: Joint Routing and VC allocation**
  - **Simultaneously solved for all the flows**
  - **No restriction on flow turns**

Channel Dependency Graph

A channel dependency graph (CDG) is a directed graph D = (C, A), where the vertex set C corresponds to a set of channels in a network and the arc set A consists of pairs of channels $(c_i, c_j); c_i, c_j \in C$ such that there is a direct dependency from c_i to c_j if there is routing of a flow that transitions from channel c_i to c_j.



- Each channel corresponds to a vertex in CDG
- Adjacent channel-pair in a routing path correspond to an edge in CDG

Portion of a 2D Mesh

Layered Channel Dependency Graph

A layered channel dependency graph (L-CDG) for a 2D mesh G = (V, E) with H virtual channels per link is a channel dependency graph D = (C, A), with the following properties:

- Each link e ∈ E in the 2D mesh G = (V, E) has exactly H virtual channels, indexed h = 1, 2, . . . , H;

$$c_e^h \text{ is a virtual channel on link } e \text{ with VC index } h$$

- The vertex set C in the L-CDG corresponds to |E| × H channels in the network
- There is a direct dependency in the arc set A $\text{from } c_i^g \text{ to } c_j^h$ iff there is a routing of a flow that transitions along two consecutive links from channel c_i^g to c_j^h

JARVA's proposition

W.R.T L-CDG

Condition 1: If a route on two consecutive links correspond to a restricted turn, the VC index assigned to the second link must be strictly increasing.

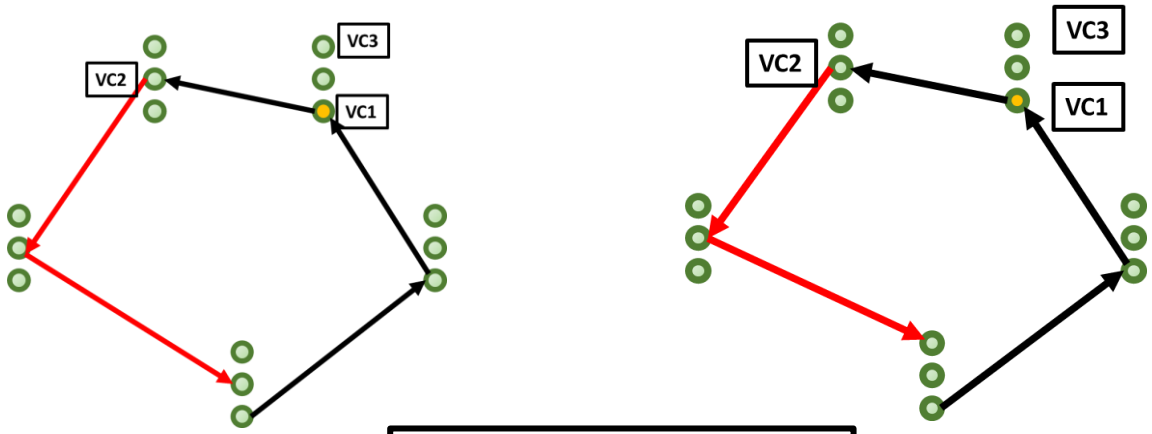
Condition 2: If a route on two consecutive links does not correspond to a restricted turn, the VC index assigned to the second link can remain the same or be strictly increasing.

Big Idea to avoid deadlock: Always increase the VC index at the restricted turn and never decrease the VC index otherwise !

If a joint application-aware oblivious routing and static VC assignment for a set of flows satisfies the above conditions for a 2D mesh with H virtual channels per link, then the corresponding solution is deadlock free

Intuitive Proof for JARVA's deadlock freedom

Deadlocks are avoided on a VC layer by requiring the VC assignment to go up to a higher indexed VC layer on restricted turns, thereby making it impossible to have cyclic dependencies on a layer



L-CDG representation

Deadlocks also cannot occur between layers since the VC assignments for consecutive links can never go back down to a lower layer, thereby making it impossible to have cyclic dependencies between layers

Boolean Formulation

Term	Definition
$\alpha_i(u,v)$	A Boolean (0/1) variable indicating if flow i is assigned to link (u,v)
$\beta_i^c(u,v)$	A Boolean (0/1) variable indicating if flow i is assigned to link (u,v) using VC with index c
$load(u,v)$	Total load on link (u,v)
MCL	The maximum channel load

- Our goal is to jointly perform application-aware oblivious routing and static virtual channel allocation as a global optimization problem
- SMT (E.g., Z3 Solver) is well-suited to formulate our joint optimization problem as Boolean constraints and objectives, using Boolean variables, integer variables, and real variables.

Boolean Formulation: Constraints

Flow conservation (EO, AMO, OR constraints)

$$\forall i, \quad \underset{(s_i,v)\in E}{\mathbf{EO}} \left\{\alpha_i(s_i,v)\right\} \forall i, \quad \underset{(w,t_i)\in E}{\mathbf{EO}} \left\{\alpha_i(w,t_i)\right\}$$

Source and Destination

$$\forall i,\ \forall u \neq s_i, t_i, \quad \underset{(v,u)\in E}{\mathbf{AMO}} \left\{\alpha_i(v,u)\right\} \ \& \ \underset{(u,w)\in E}{\mathbf{AMO}} \left\{\alpha_i(u,w)\right\}$$

Non-source and Non-destination

$$\forall u \neq s_i, t_i, \quad \bigvee_{(v,u)\in E} \alpha_i(v,u) = \bigvee_{(u,w)\in E} \alpha_i(u,w)$$

For source and destination nodes, each flow should be assigned to exactly one outgoing link emanating from source node and exactly one incoming link going into destination node. For non-source and non-destination nodes, flow in the incoming link should match the flow in the outgoing link

Boolean Formulation: Constraints

VC Assignment (EO, OR constraints)

$$\forall (u,v) \in E, \quad \alpha_i(u,v) \Rightarrow \underset{\forall c}{\mathbf{EO}} \left\{ \beta_i^c(u,v) \right\}$$

$$\forall (u,v) \in E, \quad \neg\alpha_i(u,v) \Rightarrow \neg \bigvee_{\forall c} \left\{ \beta_i^c(u,v) \right\}$$

If a flow is assigned to a link, then it must also be statically assigned to exactly one of its VCs. In other words, a flow cannot use two VCs of the same link

Boolean Formulation: Constraints

Deadlock avoidance (AMO, AND, OR)

$$\forall i,\ \forall \{(v,u),(u,w)\} \in T,$$
$$\bigwedge_{\forall (c_1,c_2),\ c_2 \le c_1} \mathbf{AMO}\left\{\beta_i^{c_1}(v,u), \beta_i^{c_2}(u,w)\right\}$$

Increasing for restricted turns

$$\forall i, \forall u \neq s_i, t_i$$
$$\bigwedge_{\forall (c_1,c_2),\ c_2 < c_1} \mathbf{AMO}\left\{ \bigvee_{(v,u)\in E} \beta_i^{c_1}(v,u), \bigvee_{(u,w)\in E} \beta_i^{c_2}(u,w)\right\}$$

Non-decreasing otherwise

VC transition is always non-decreasing, meaning that the VCs assigned on two consecutive links for a flow can remain the same or change to a higher-indexed VC

VC assigned to the flow transitions from a lower-indexed VC to a higher-indexed VC (increasing) at the restricted turn

Boolean Formulation: Objective

If-then-else constraint

$$\forall (u,v) \in E, \quad load(u,v) = \sum_{i=1}^{K} \mathbf{ITE}\Big(\alpha_i(u,v), d_i, 0\Big)$$

The load on each link is the sum-total of the demands of flows assigned to it

$$\forall (u,v) \in E, \quad MCL \geq load(u,v)$$

$$\min MCL$$

The goal is to minimize the “maximum channel load” of the network

Intuitively, the injection rate is inversely proportional to MCL with respect to the link capacity and minimizing the MCL can optimize network throughput and network latency

JARVA's formulation complexity

Complexity as a function of the nodes in the network |V |, the number of edges in the network |E|, the number of flows |K| and the number of virtual channels |C| per edge

Constraint Definition	# Constraints	# Literals per constraint
Flow conservation	$2 * \|V\| * \|K\|$	8
Static VC assignment	$2 * \|E\| * \|K\|$	$\|C\|$
Deadlock-freedom	$\|V\| * \|K\|$	$\|C\|^2$
Load constraint	$\|E\|$	$\|K\|$
Load bound	$\|E\|$	1

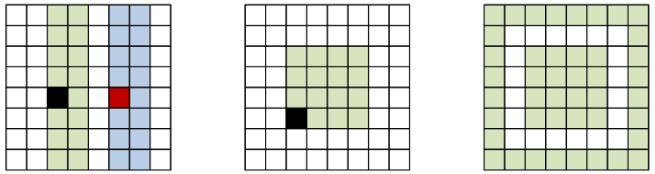
Design of experiments

Routing Algorithms
XY/YX
ROMM
Valiant
O1Turn
BSOR/BSORM
JARVA

Synthetic Benchmarks

Transpose, Shuffle and Bit-complement

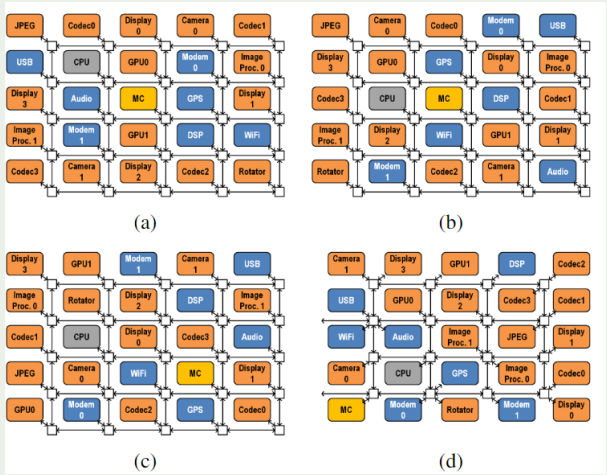
Hotspot patterns



(a) Hotspot1. (b) Hotspot2. (c) P-Transpose.

Real Camcorder Benchmarks

Heterogeneous cores arranged in 2D network



Packet size:

- Synthetic: 1-6 flits per packet
- Camcorder: Communication characteristics are derived from [11]

[11] Y. Song et al., "Improving memory efficiency in heterogeneous mpsocs through row-buffer locality-aware forwarding," ACM TACO, 2020

Results: Maximum Channel Load (MCL)

Synthetic Benchmarks

Traffic	XY	YX	ROMM	VAL	O1TURN	BSOR	BSORM	JARVA	Δ
Transpose	17	16	14	17	13	8	7	6*	14.28%
Bitcomp	10	10	19	16	14	11	12	8*	20.00%
Shuffle	16	15	17	27	16	11	10	7*	30.00%
Hotspot1	29	30	26	53	23	17	20	15*	11.76%
Hotspot2	36	43	30	35	27	15	21	13*	13.33%
P-Transpose	19	17	16	16	13	7	8	6*	14.28%

Camcorder Benchmarks

Traffic	XY	YX	ROMM	VAL	O1TURN	BSOR	BSORM	JARVA	Δ
Camcorder(a)	407	404	357	486	322	277	263	207*	21.29%
Camcorder(b)	241	294	233	476	264	241	259	207*	11.16%
Camcorder(c)	608	555	483	493	400	225	287	207*	8.00%
Camcorder(d)	553	681	500	614	424	258	447	219*	14.70%

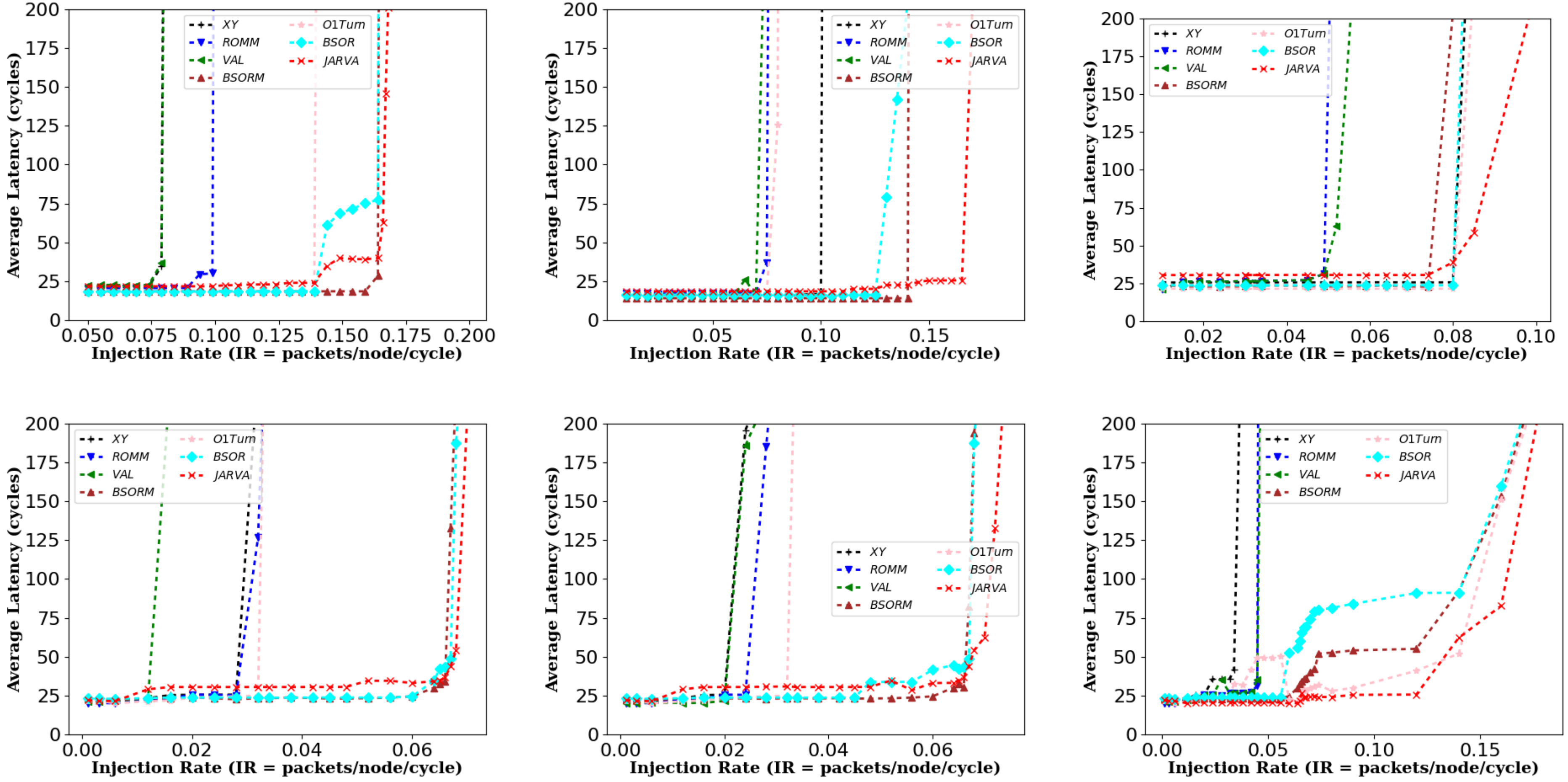
Hypothesis: **The routing algorithm with lower maximum channel load can withstand higher injection rates before reaching network saturation**

Flit-level simulations: Setup

Simulator	**NOXIM, HORNET**
VCs	4
Packet size	1-6 flow units for synthetic, 33MB/s – 3.9 GB/s for real benchmarks
Mesh	8x8, 5x5 2D Mesh
Warmup	10000 cycles
Simulation	500000 cycles

Flit-level simulations: Results

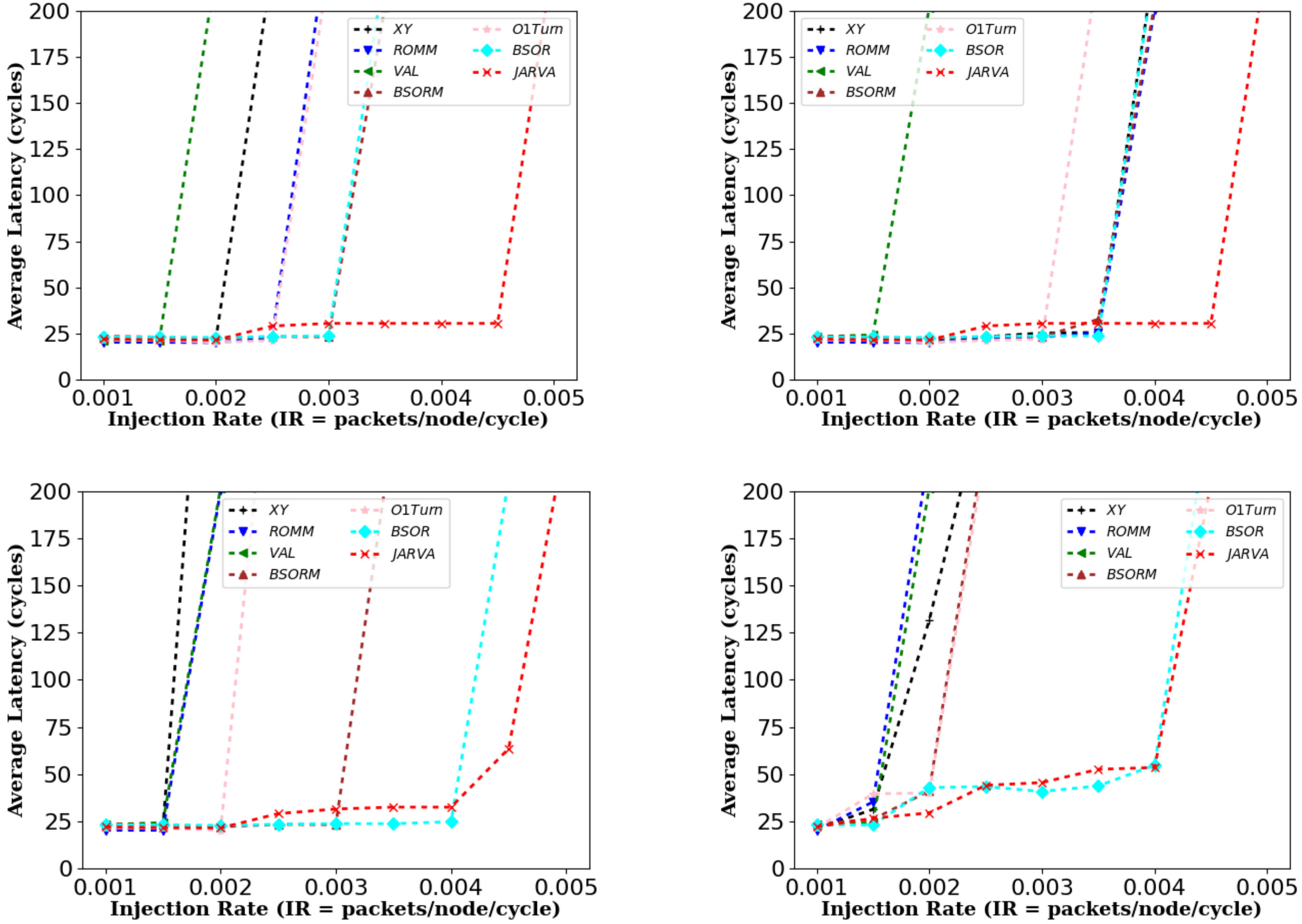
Synthetic Benchmarks



Observation 1: JARVA derived routing solutions for synthetic benchmarks can **withstand higher injection** rates

Flit-level simulations: Results

Camcorder Benchmarks



Observation 2: JARVA derived routing solutions for real camcorder benchmarks can **withstand higher injection** rates

Conclusions

- We propose a joint application-aware oblivious routing and static virtual channel allocation (JARVA) framework for better deadlock-free performance
- We avoid unnecessary routing restrictions on the flows as long as there is a corresponding deadlock-free VC assignment
- We use SMT framework [Z3] to capture complex conditional constraints that are needed to model our joint optimization problem
- Our framework can be extended to other interesting objective functions and topologies
- JARVA can achieve up to 21.29% improvement for the “camcorder application” and up to 30% improvement for synthetic benchmarks