Performance Analysis of Priority-Aware NoCs with Deflection Routing under Traffic Congestion

Sumit K. Mandal, Anish Krishnakumar,

Umit Y. Ogras

University of Wisconsin-Madison

Raid Ayoub, Miachel Kishinevsky

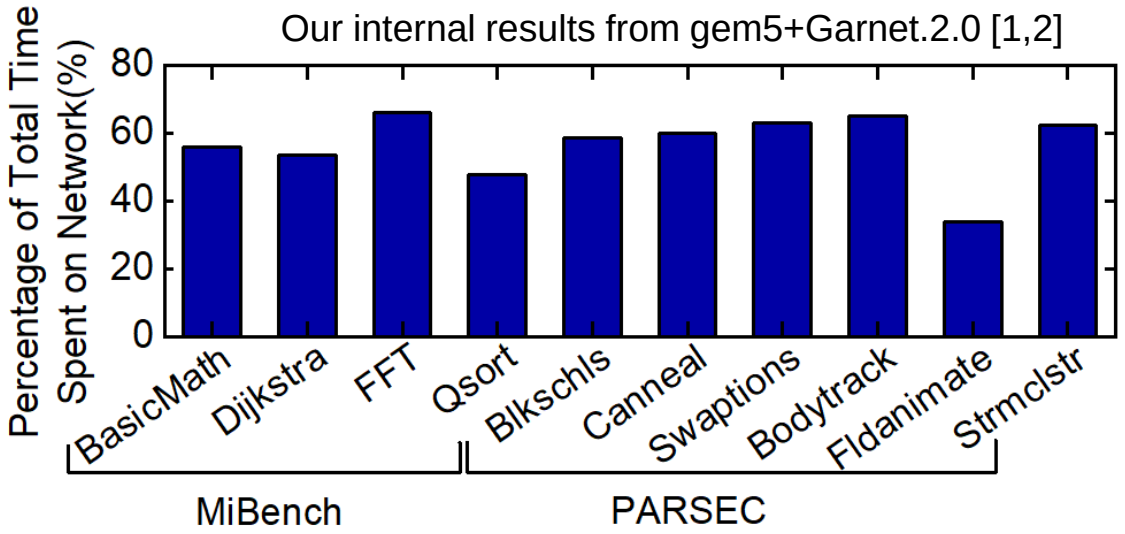
Intel Corporation

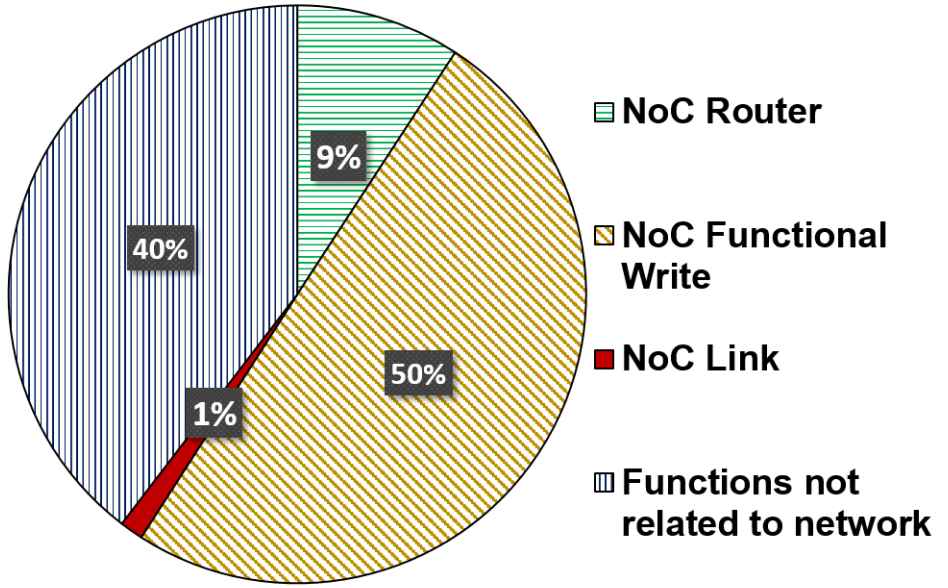
Agenda

- **Need for analytical fabric model generation**
- **Background**
  - Networks-on-chip (NoC) used in industry
  - Deflection routing
  - Prior work
- **Proposed approach**
- **Experimental results**
- **Conclusion and future work**

Need for Analytical Fabric Model Generation

- **Examples of emerging applications:**
  - Virtual reality, autonomous driving,
  - Machine learning, AI
- **System modeling challenges**
  - Both SW and HW are growing in complexity
  - Emerging applications require longer runs for meaningful pre-silicon performance, power, and thermal analysis (*minutes instead of milliseconds*)
- **Research need**
  - Communication fabric: Central shared resource
  - Fast and accurate system level modeling
  - Fast design space exploration



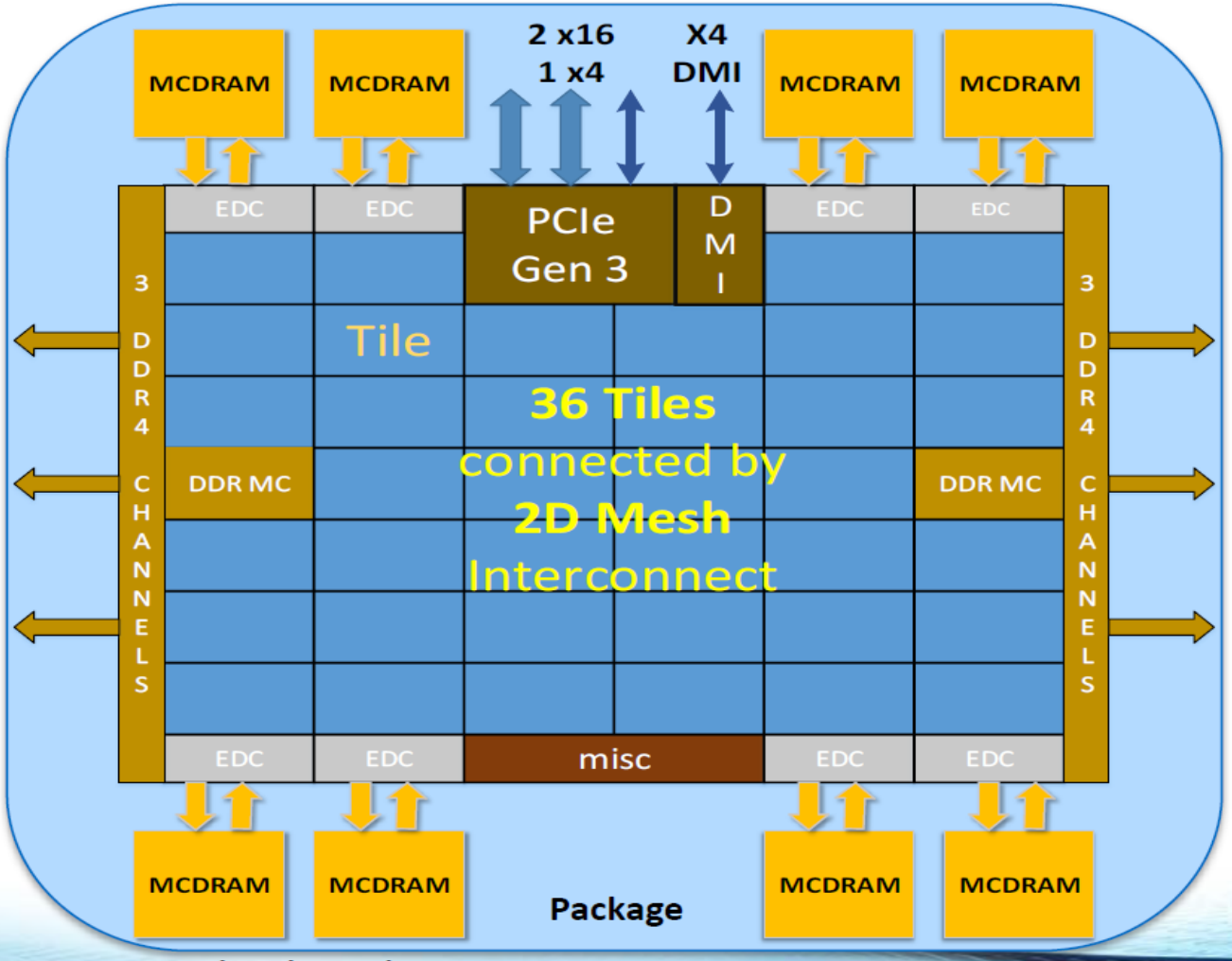


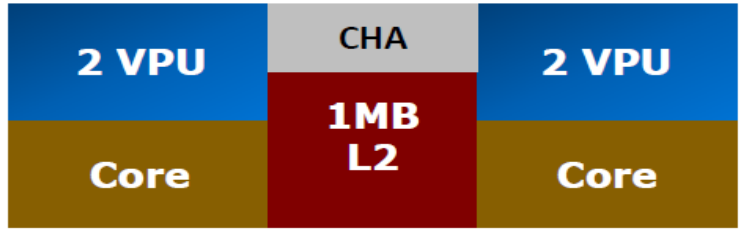
[1] Binkert, Nathan, et al. "The gem5 simulator." *ACM SIGARCH computer architecture news* 39.2 (2011): 1-7.

[2] Mandal, Sumit K., et al. "Analytical performance models for nocs with multiple priority traffic classes." *ACM TECS* 18.5s (2019): 1-21.

Example Fabric: Xeon Phi (KNL) Processor

Also used in Xeon™ servers (e.g., Skylake, Icelake) and PC clients (rings)





Chip: 36 Tiles interconnected by **2D Mesh**

Tile: 2 Cores + 2 VPU/core + 1 MB L2

Memory: MCDRAM: 16 GB on-package; High BW

DDR4: 6 channels @ 2400 up to 384GB

IO: 36 lanes PCIe Gen3. 4 lanes of DMI for chipset

Node: 1-Socket only

Fabric: Omni-Path on-package (not shown)

Vector Peak Perf: 3+TF DP and 6+TF SP Flops

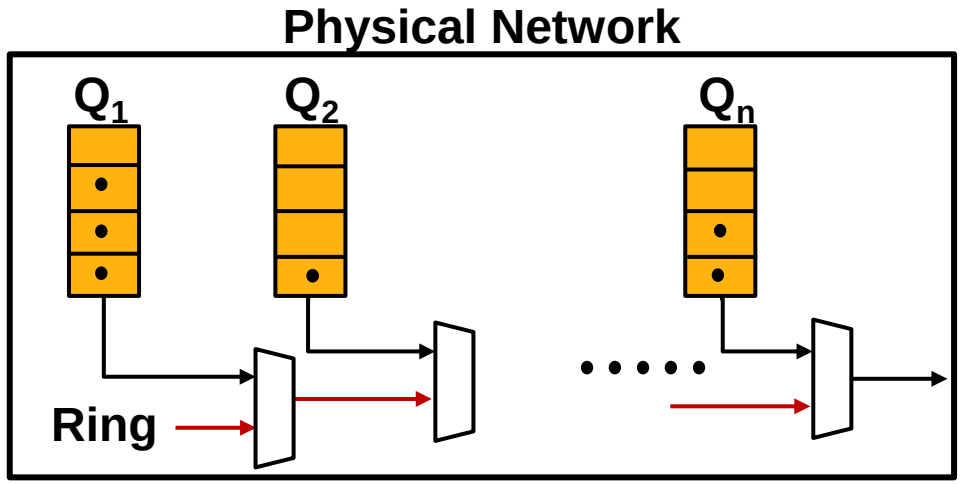
Scalar Perf: ~3x over Knights Corner

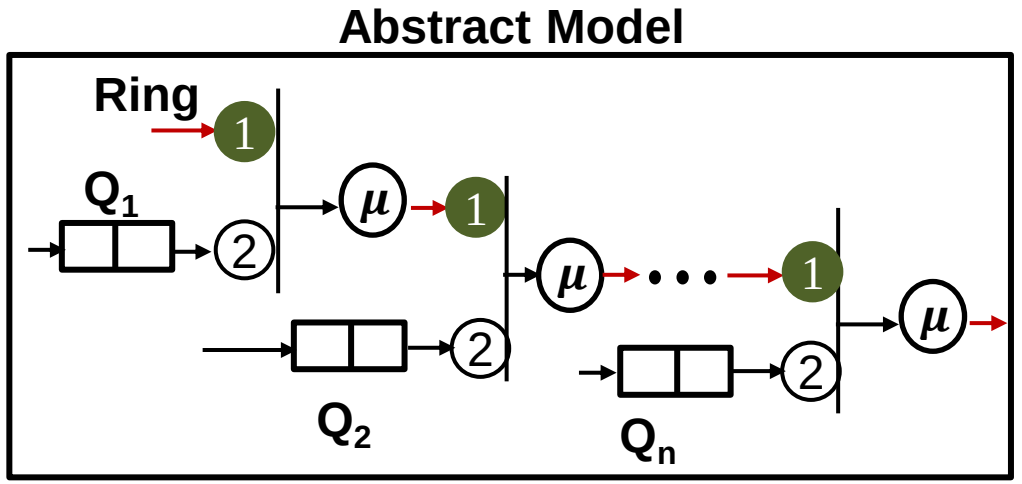
Streams Triad (GB/s): MCDRAM : 400+; DDR: 90+

Source Intel: All products, computer systems, dates and figures specified are preliminary based on current expectations, and are subject to change without notice. KNL data are preliminary based on current expectations and are subject to change without notice. 1Binary Compatible with Intel Xeon processors using Haswell Instruction Set (except TSX). 2Bandwidth numbers are based on STREAM-like memory access pattern when MCDRAM used as flat memory. Results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance.

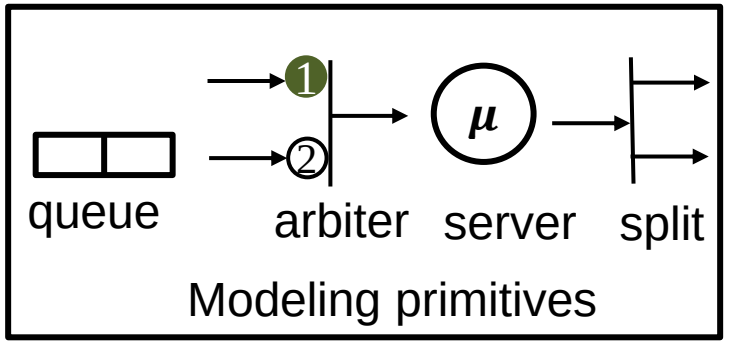
Priority Aware Networks-on-Chip Basics

- **Industrial NoCs use routers with priority arbitration to achieve predictable latency**
  - Packets in NoC have higher priority than new packets



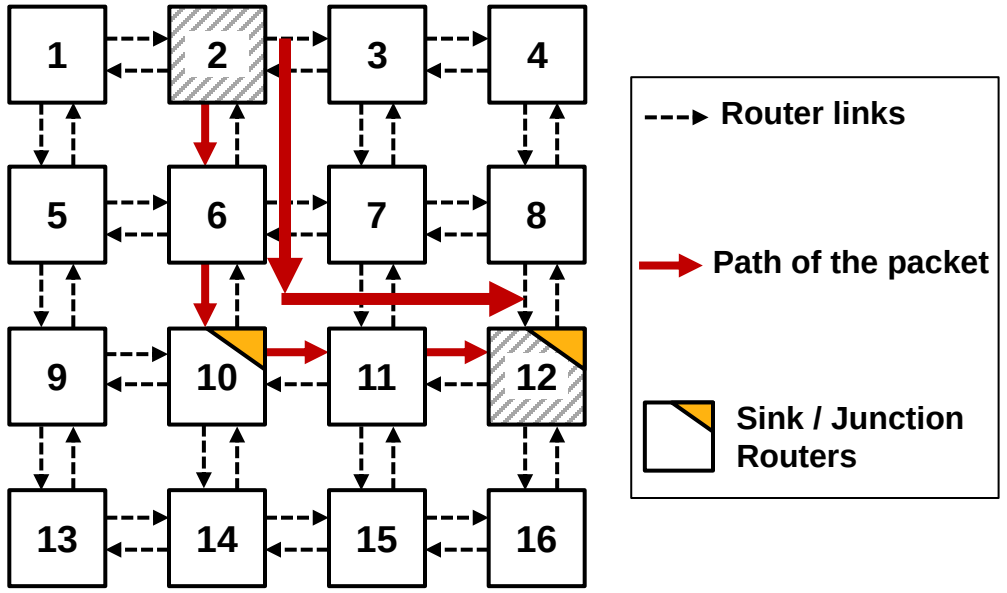


- Mux' in routers modeled as **priority arbiters** and **servers for the queues**
- Inputs with filled color denote higher priority



Performance Analysis of Communication Fabric

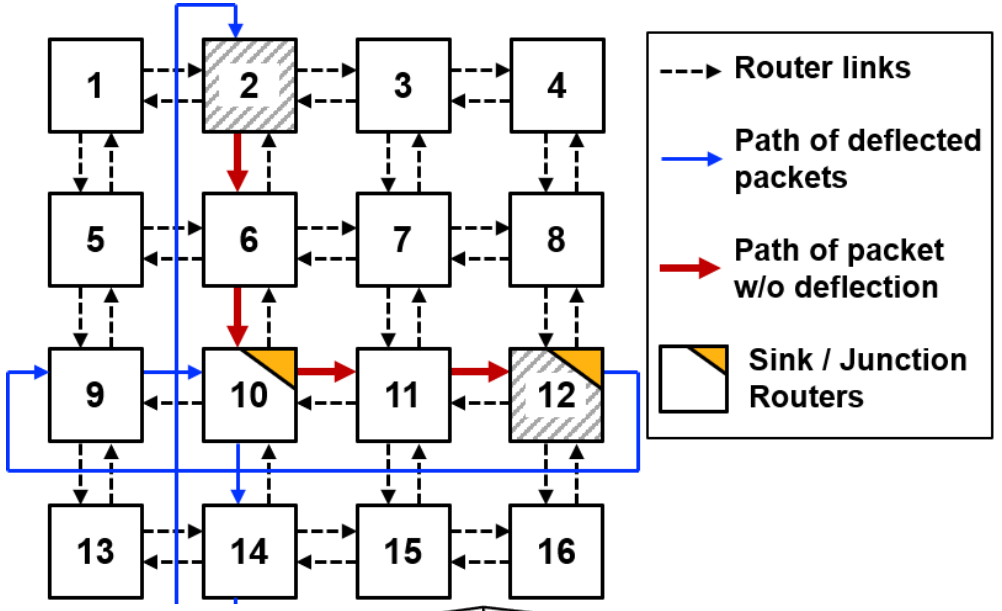
- **Example on a 4x4 mesh with YX routing**
- **Source to destination latency (L_{SD}) has four components**
  - Waiting time in source queue (W_Q^S)
  - Deterministic vertical latency (L_v)
  - Waiting time at the junction (W_Q^J)
  - Deterministic horizontal latency (L_h)
  - $\boldsymbol{L_{SD} = W_Q^S + L_v + W_Q^J + L_h}$
- **L_v and L_h depend on the source-destination pair and fabric topology**
- **W_Q^S and W_Q^J depend on injection rates at different queues and need detailed analysis**

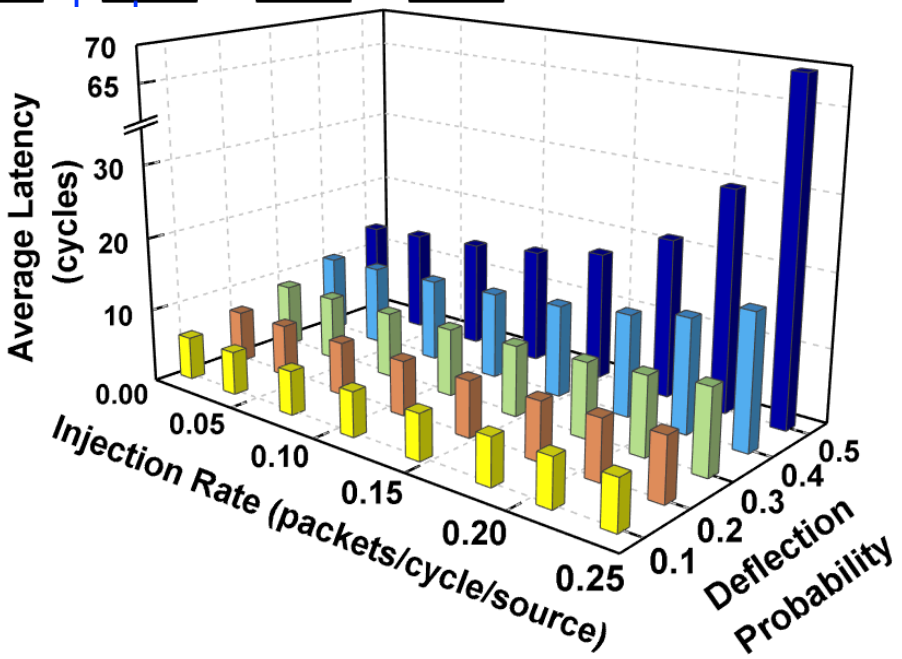


Deflection Routing

- **Packets are deflected if ingress at the junction (Node 10) or the destination (Node 12) is full**
  - The deflected packets circulate within the same row or column
  - Increase congestion towards the source

- **NoC analytical models need to take deflection into account**
  - Average latency varies in the range of 6-70 cycles for an injection rate of 0.25 when deflection probability varies from 0.1 to 0.5





Prior Work on Performance Analysis of Networks

	Priority-based	Multiple classes	Deflection Routing
[1,3,6]	✗	✓	✗
[4,5]	✓	✗	✗
[7]	✓	✓	✗
[2]	✗	✓	✓
This work	✓	✓	✓

[1] Ogras et al. "An analytical approach for network-on-chip performance analysis." *IEEE TCAD IC and Systems* (2010).

[2] Ghosh et al. "An analytical framework with bounded deflection adaptive routing for networks-on-chip." *2010 IEEE Computer Society Annual Symposium on VLSI* (2010).

[3] Bogdan et al. "Non-stationary traffic analysis and its implications on multicore platform design." *IEEE TCAD IC and Systems* (2011).

[4] Kiasari et al. "An analytical latency model for networks-on-chip." *IEEE TVLSI Systems* 21.1 (2013).

[5] Kashif et al. "Bounding buffer space requirements for real-time priority-aware networks." *ASP-DAC* (2014).

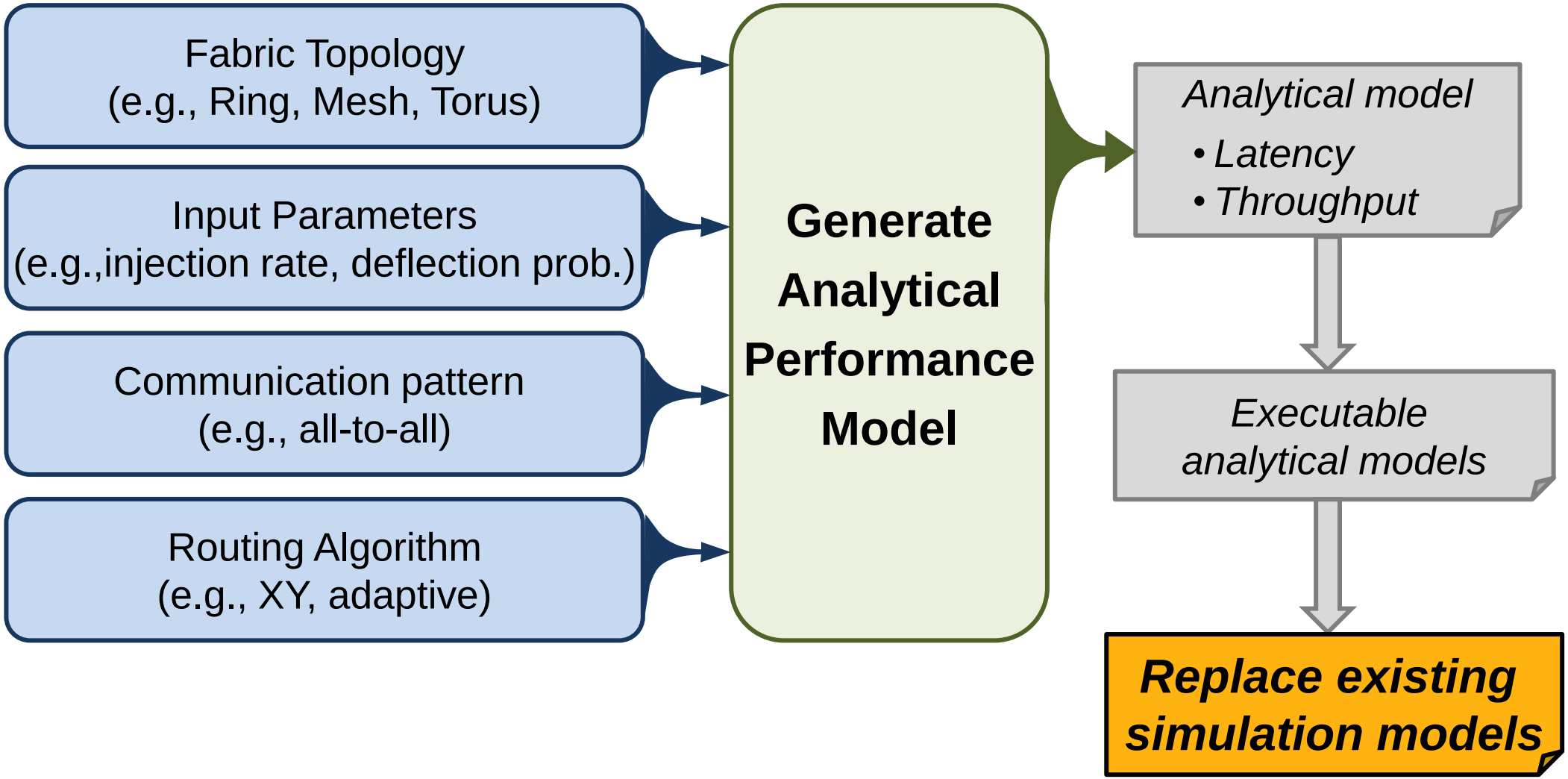
[6] Qian et al. "A support vector regression (SVR)-based latency model for network-on-chip (NoC) architectures." *IEEE TCAD IC and Systems* 35.3 (2016).

[7] Mandal, Sumit K., et al. "Analytical performance models for nocs with multiple priority traffic classes." *ACM TECS* 18.5s (2019): 1-21.

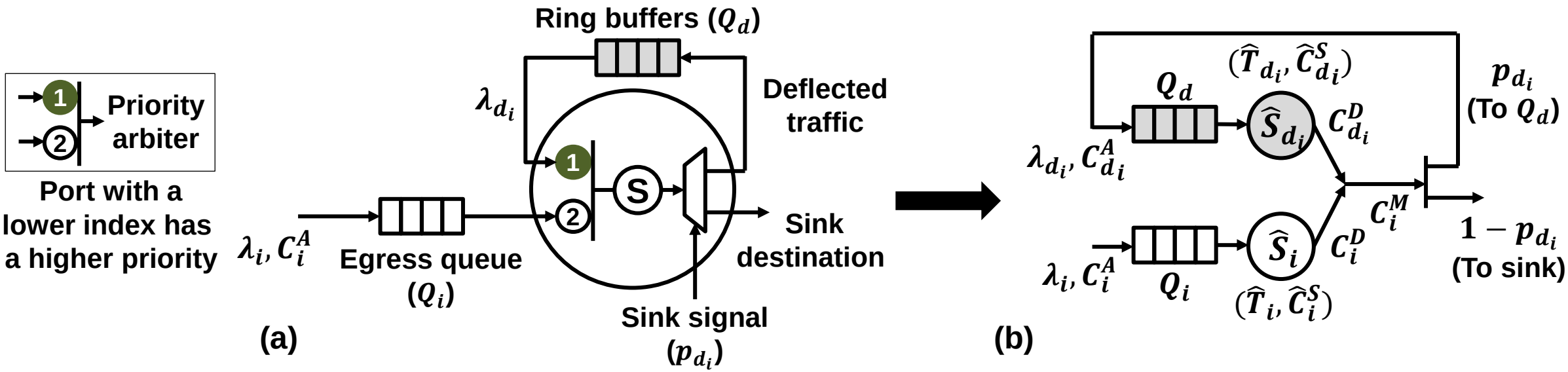
Agenda

- Need for analytical fabric model generation
- Background
  - Networks-on-chip (NoC) used in industry
  - Deflection routing
  - Prior work
- **Proposed approach**
- Experimental results
- Conclusion and future work

Overview of the Proposed Approach



Analytical Model for a System with Single Class (1)



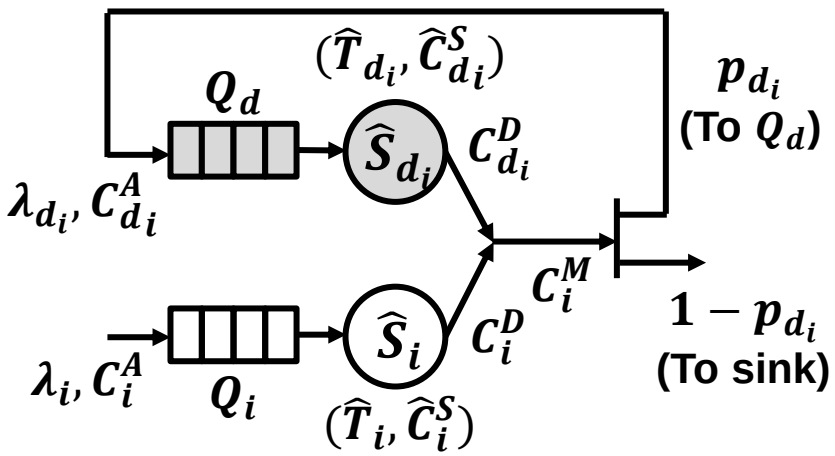
- The behavior of deflected packets is modeled by a buffer (Q_d)
- Deflected packets and the packet in the egress queue (Q_i) form a priority-aware queuing system
- Rate of deflected packets (λ_{d_i}) and coefficient of variation of deflected packets ($C_{d_i}^A$) need to be computed
  - Shown on the next slide

Analytical Model for a System with Single Class (2)

- **Computation of $\boldsymbol{\lambda_{d_i}}$**

$$N_{d_i} = 1p_{d_i}(1-p_{d_i}) + 2p_{d_i}^2(1-p_{d_i}) + \cdots = \frac{p_{d_i}}{1-p_{d_i}}$$

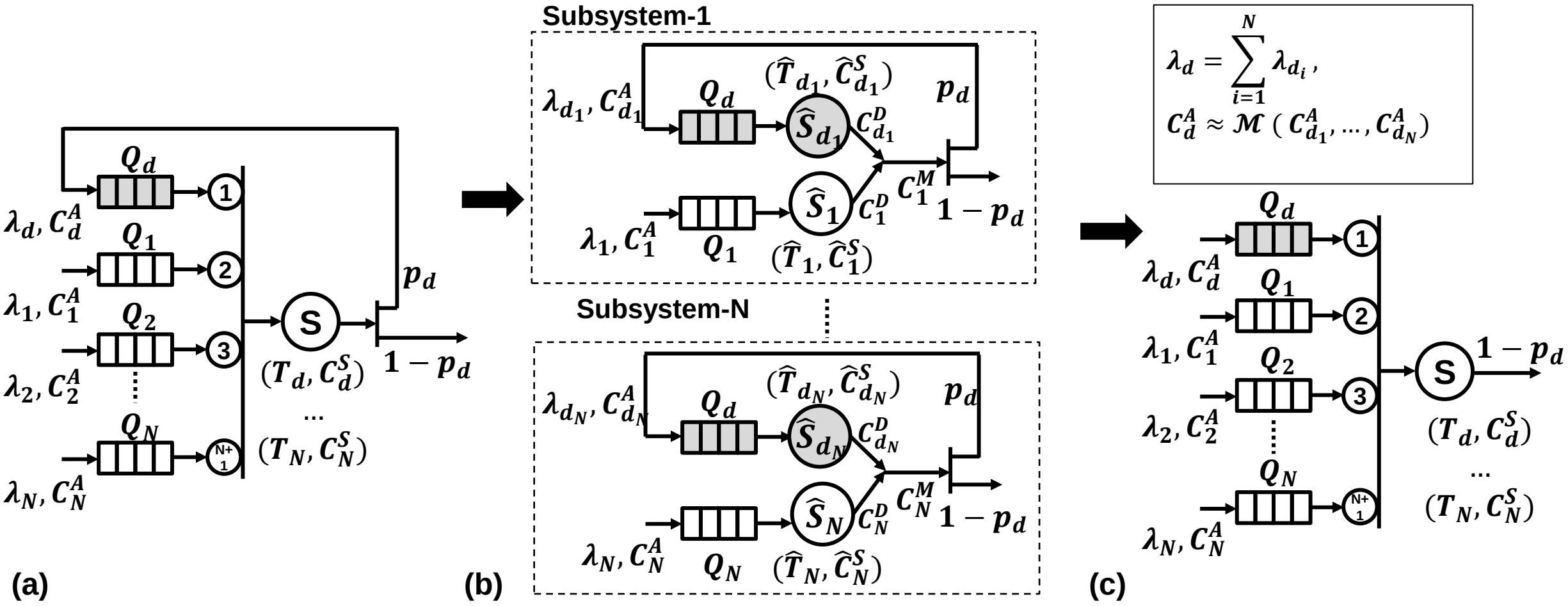
$$\lambda_{d_i} = \lambda_i N_{d_i} = \lambda_i \frac{p_{d_i}}{1-p_{d_i}}$$



- **Computation of $\boldsymbol{C^A_{d_i}}$**
  - $\hat{C}_i^S$ through entropy maximization method [1]
  - C_i^D and $C_{d_i}^D$;C_i^M by merging [2]
  - $C_{d_i}^A$ by splitting [2]
  - W_{d_i} and W_i through the analysis of priority-aware queuing system

[1] Kouvatsos, Demetres, and Irfan Awan. "Entropy maximisation and open queueing networks with priorities and blocking." *Performance Evaluation* 51.2-4 (2003): 191-227.
[2] G. Bolch et al. Queuing Networks and Markov Chains. Wiley. 2006

Analytical Model for Multiple Class: Superposition

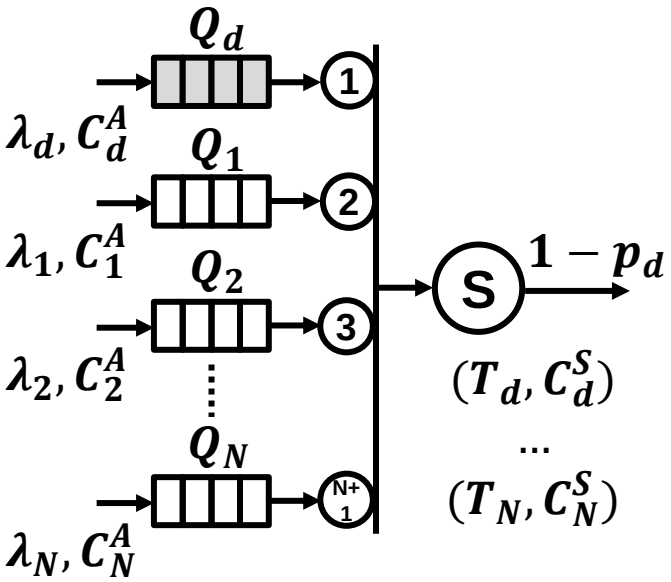


$$\lambda_d = \sum_{i=1}^{N} \lambda_{d_i},$$

$$C_d^A \approx \mathcal{M}\left(C_{d_1}^A, \ldots, C_{d_N}^A\right)$$

Analytical Model for Multiple Class: Superposition

$$\lambda_d = \sum_{i=1}^{N} \lambda_{d_i},$$
$$C_d^A \approx \mathcal{M}\,(C_{d_1}^A, \ldots, C_{d_N}^A)$$



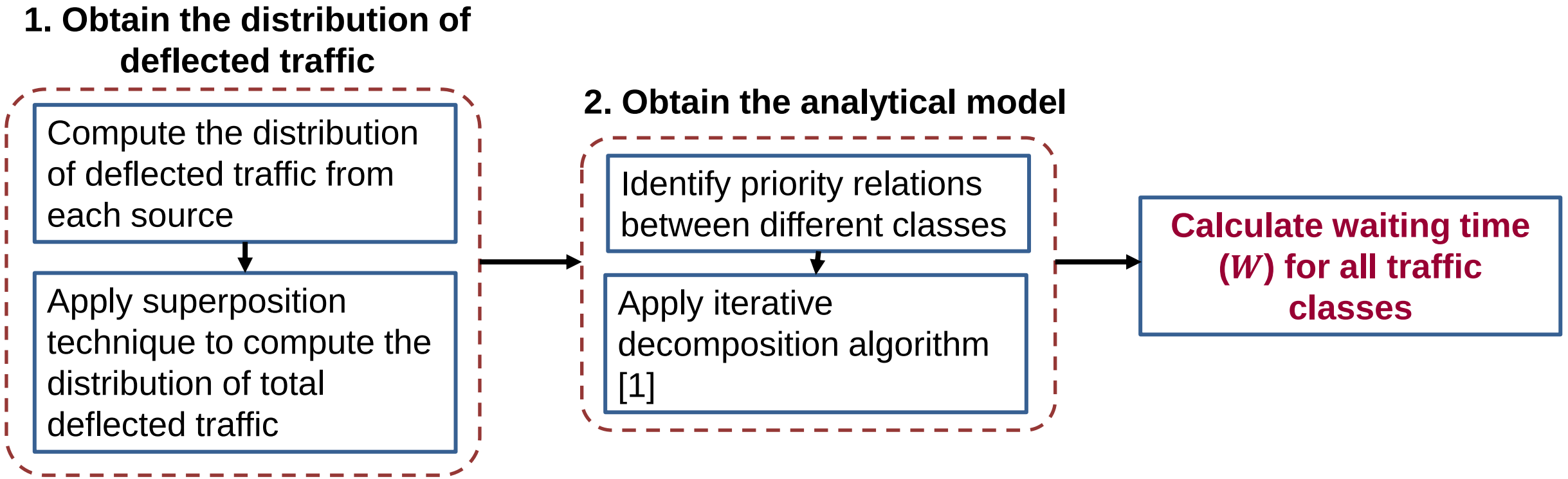
- **Analytical model for waiting time of each class i is applied**

- $$W_i = \frac{\rho_d(T_d+1)+2\rho_i W_d}{2(1-\rho_d-\sum_{n=1}^{i}\rho_n)} + \frac{\sum_{i=1}^{n-1}\rho_n(T_n+1)+2\rho_i W_n}{2(1-\rho_d-\sum_{n=1}^{i}\rho_n)} + \frac{\rho_i(T_i-1)+T_i\left(\left(C_i^A\right)^2+\lambda_i-1\right)}{2(1-\rho_d-\sum_{n=1}^{i}\rho_n)}$$

Putting Everything Together: Model Flow

Input: **Injection rates for all traffic classes (λ), Network topology, Routing algorithm, Deflection probability**

Output: **Waiting time for all traffic classes**



[1] Mandal, Sumit K., et al. "Analytical performance models for nocs with multiple priority traffic classes." *ACM TECS* 18.5s (2019): 1-21.

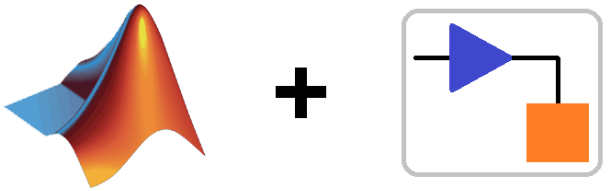
Agenda

- Need for analytical fabric model generation
- Background
  - Networks-on-chip (NoC) used in industry
  - Deflection routing
  - Prior work
- Proposed approach
- **Experimental results**
- Conclusion and future work

Experimental Setup

- **We evaluated the proposed analytical models on**
  - Ring
  - Mesh
- **Simulation parameters**
  - Simulation length: 10M cycles
  - Warm-up period: 5000 cycles
- **Traffic load**
  - Sweep from a very light load to λ_{max}
  - λ_{max} is the injection rate at which the maximum server utilization is 1
  - Vary deflection probability

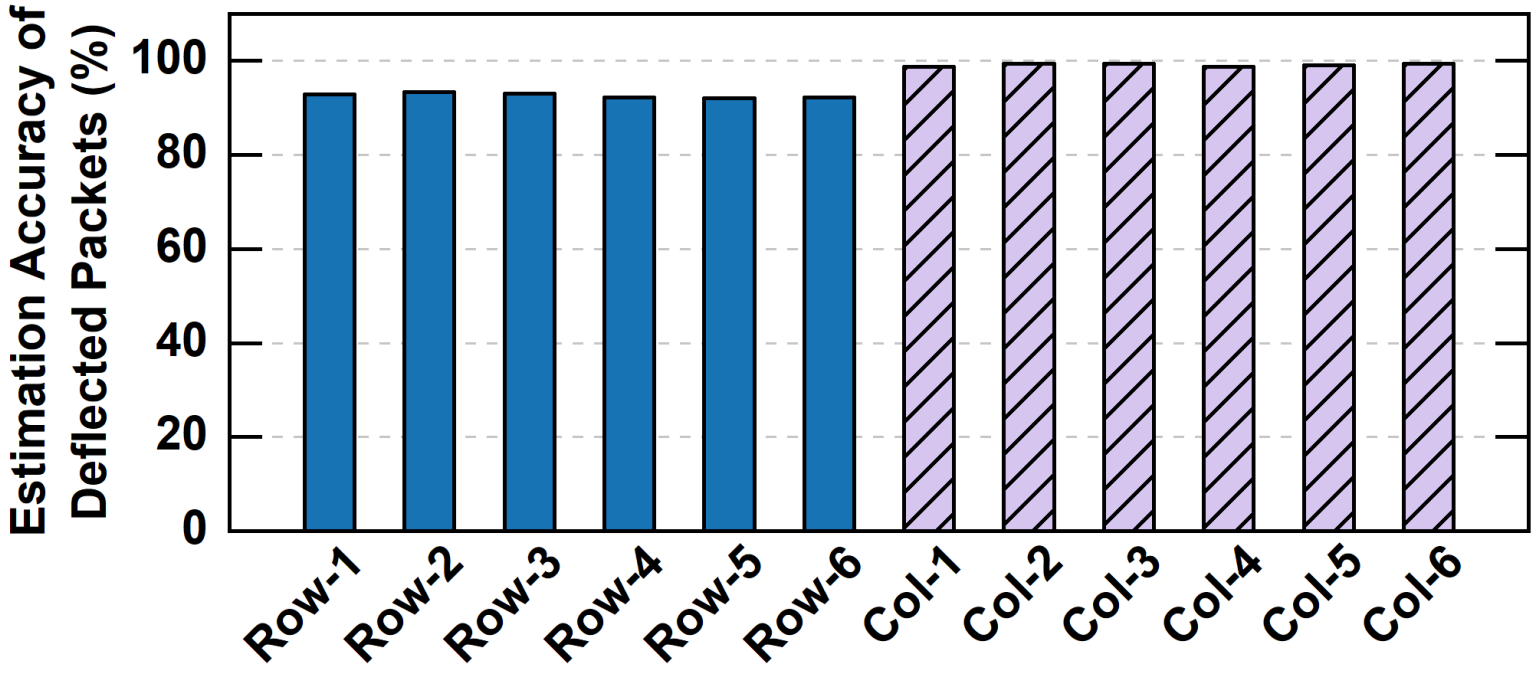
xPLORE



MATLAB
SIMULINK®

Estimation Accuracy for Deflected Traffic

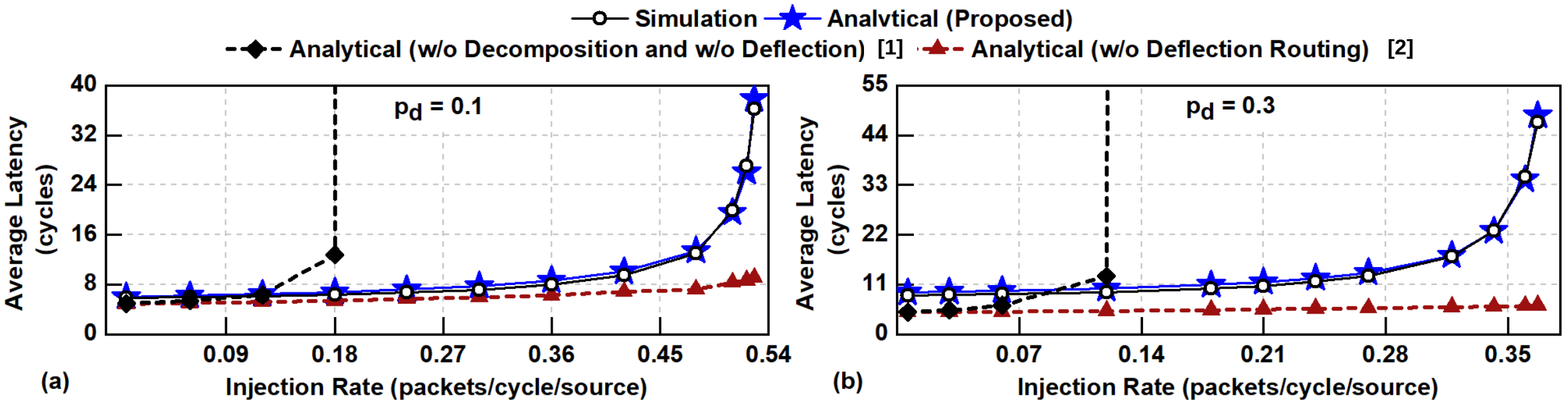
- **Estimating average number of deflected packets is a key component**
- **Evaluated the estimation accuracy for 6×6 mesh**
  - Used in Xeon Phi processor
  - Deflection probability of 0.3



Accuracy is consistently above 92%

Evaluation on 6x6 Mesh with Geometric Input

- **Achieve <8% modeling error** on average for $p_d = 0.1$ and $p_d = 0.3$



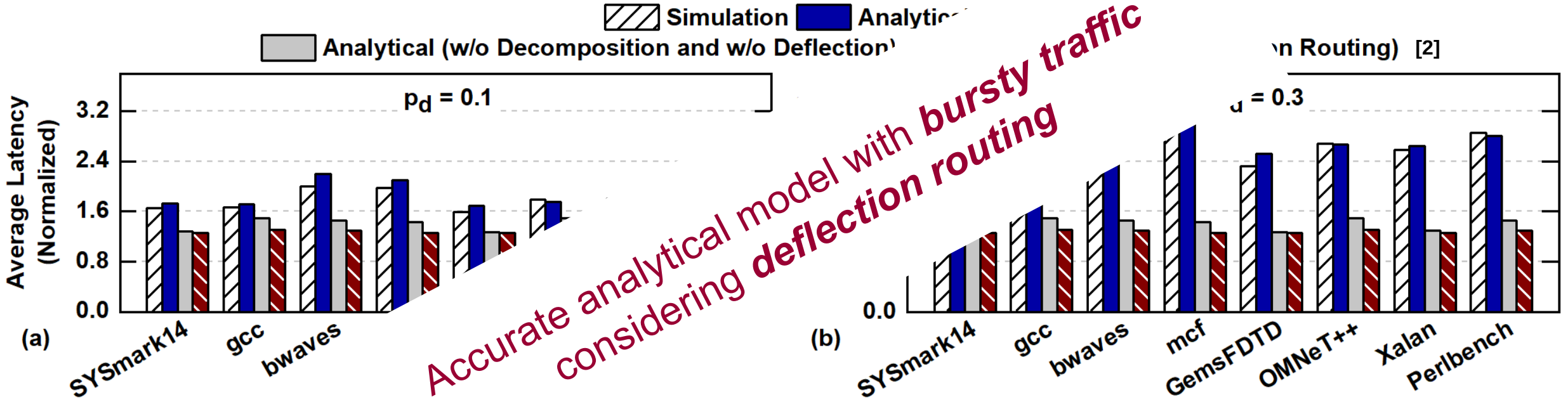
- **Models without decomposition and without deflection overestimates the latency**
- **Models which ignore deflection routing underestimates the latency**

[1] Kiasari, Abbas Eslami, Zhonghai Lu, and Axel Jantsch. "An analytical latency model for networks-on-chip." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 21.1 (2012): 113-123.

[2] Mandal, Sumit K., et al. "Analytical Performance Models for NoCs with Multiple Priority Traffic Classes." *ACM Transactions on Embedded Computing Systems (TECS)* 18.5s (2019): 1-21.

Evaluation on Real Applications (bursty) with 6x6 Mesh

- **Achieve <5% modeling error** on average for $p_d = 0.1$ $p_d = 0.3$



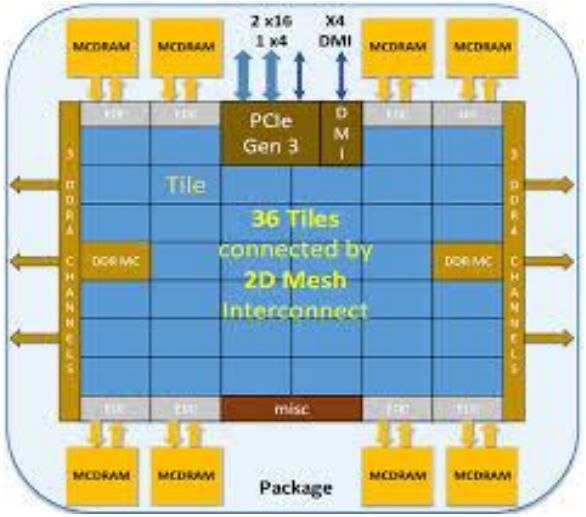
- **State-of-the-art a models are <u>unable</u> to provide accurate latency estimation**

[1] Kiasari, Abbas Eslami, Zhonghai Lu, and Axel Jantsch. "An analytical latency model for networks-on-chip." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 21.1 (2012): 113-123.

[2] Mandal, Sumit K., et al. "Analytical Performance Models for NoCs with Multiple Priority Traffic Classes." *ACM Transactions on Embedded Computing Systems (TECS)* 18.5s (2019): 1-21.

Conclusion and Future Work

- **Industrial NoCs use priority-based routers**
- **Most NoC performance analysis techniques assume fair arbitration and do not consider multiple traffic classes as well as deflection routing**
- **Presented the first technique that handles**
  - Priority, multiple traffic classes and deflection routing
- **Analytical models are significantly better than state-of-the-art techniques in the literature**
- **In future we will apply the model to put a bound on injection rate at sources (source throttling)**



THANK YOU