



# Network-on-Chip Performance Analysis and Optimization for Deep Learning Applications

**Sumit K. Mandal**  
Preliminary Examination  
10 June 2021

**e lab**  **WISCONSIN**  
UNIVERSITY OF WISCONSIN-MADISON

1

## Agenda

- **Motivation**
- **Preliminary Work-1:** Performance Analysis of NoCs
  - Priority-Aware NoCs
  - NoCs with Bursty Traffic
  - NoCs with Deflection Routing
- **Preliminary Work-2:** Communication-Aware Hardware Accelerator for DNNs
- **Proposed Work**
  - Multi-Objective Optimization to Design Optimized NoC
  - Accelerator for Graph Convolutional Network (GCN)
- **Conclusions**

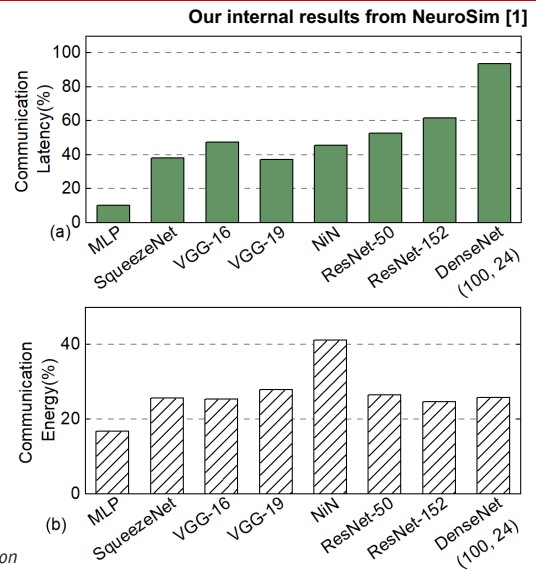
2

2

## Motivation: Communication in Deep Learning

- **Deep Neural Networks (DNNs) are growing deeper and denser day-by-day**
- **Hardware accelerators for DNNs reduce the execution time with respect to general purpose processors**
- **DNNs exhibit high communication volume between layers**
  - Contributes up to 90% of the inference latency
  - Contributes up to 40% of the energy
- **Need for an efficient on-chip communication techniques for DNN accelerators**

[1] Chen, Pai-Yu, Xiaochen Peng, and Shimeng Yu. "NeuroSim: A circuit-level macro model for benchmarking neuro-inspired architectures in online learning." *IEEE Trans. on CAD of Integrated Circuits and Systems* 37.12 (2018): 3067-3080.

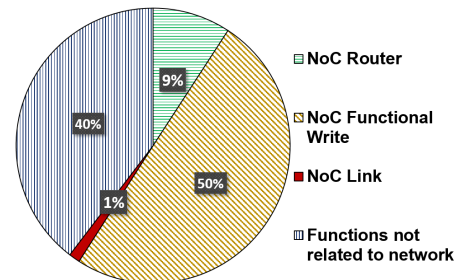
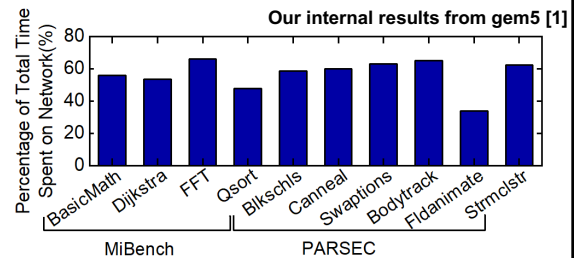


3

3

## Motivation: NoC Performance Analysis

- **System modelling challenges**
  - Both SW and HW are growing in complexity
  - Complex applications require long runs for meaningful pre-silicon performance, power, and thermal analysis
- **Research need**
  - Communication fabric: Central shared resource
  - Fast and accurate system level modeling
  - Automated generation of high-level performance and power models for SoCs



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

4

4

## Agenda

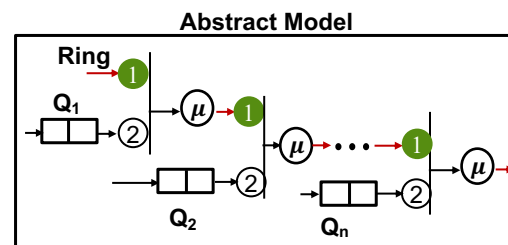
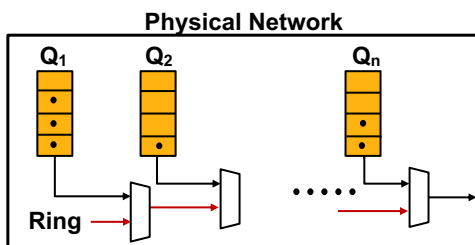
- **Motivation**
- **Preliminary Work-1: Performance Analysis of NoCs**
  - Priority-Aware NoCs
  - NoCs with Bursty Traffic
  - NoCs with Deflection Routing
- **Preliminary Work-2: Communication-Aware Hardware Accelerator for DNNs**
- **Proposed Work**
  - Multi-Objective Optimization to Design Optimized NoC
  - Accelerator for Graph Convolutional Network (GCN)
- **Conclusions**

5

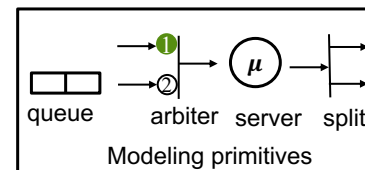
5

## 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**
- Inputs with filled color denote higher priority

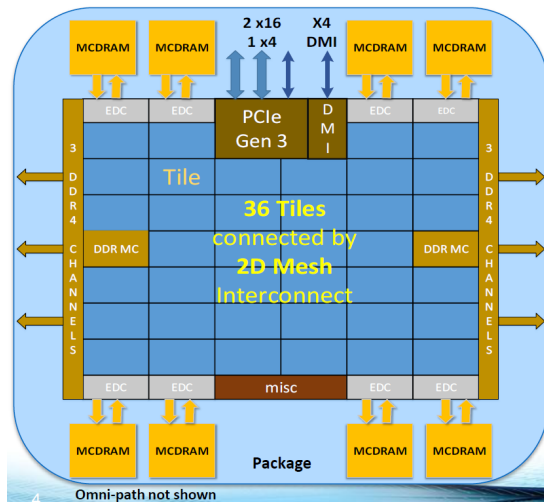


6

6

## Example Fabric: Xeon Phi (KNL) Processor

Also used in client and Xeon server (e.g., Skylake, Icelake)



| TILE | 2 VPU | CHA    | 2 VPU |
|------|-------|--------|-------|
| Core | Core  | 1MB L2 | Core  |

VPU: Vector Processing Unit

**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+**

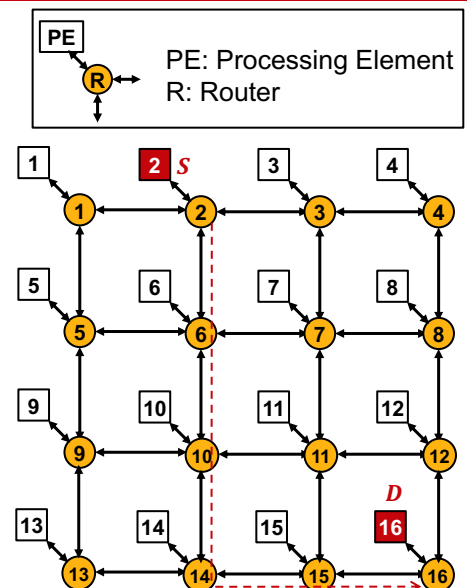
Source Intel. All products, computer systems, dates and figures specified are preliminary based on current expectations, and are subject to change without notice. KNL data are preliminary based on current expectations and are subject to change without notice. Binary Compatible with Intel Xeon processors using Haswell architecture and Intel Xeon Phi. Standard numbers are based on STREAM-like memory access patterns using 100MB of memory. Results have been estimated based on internal Intel analysis and are not intended for comparison with other processors. Any difference in system configuration or implementation may affect performance.

7

7

## Performance Analysis of Comm. Fabric

- 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 ( $L_h$ )
  - $L_v$  and  $L_h$  depend on the source-destination pair and fabric topology
- $$L_{SD} = W_Q^S + L_v + W_Q^J + L_h$$
- $W_Q^S$  and  $W_Q^J$  depend on injection rates at different queues and need detailed analysis



8

8

## Prior Work on NoC Performance Analysis

| NoC     | Priority-based | Multiple classes | Scalable | Bursty Traffic |
|---------|----------------|------------------|----------|----------------|
| [1,2,5] | ✗              | ✓                | ✓        | ✓              |
| [3,4]   | ✓              | ✗                | ✓        | ✓              |

| NoC       | Priority-based | Multiple classes | Scalable | Bursty Traffic |
|-----------|----------------|------------------|----------|----------------|
| This work | ✓              | ✓                | ✓        | ✓              |

| Off-chip Network | Priority-based | Multiple classes | Scalable | Bursty Traffic |
|------------------|----------------|------------------|----------|----------------|
| [6,7]            | ✓              | ✗                | ✓        | ✗              |
| [8]              | ✓              | ✓                | ✗        | ✗              |

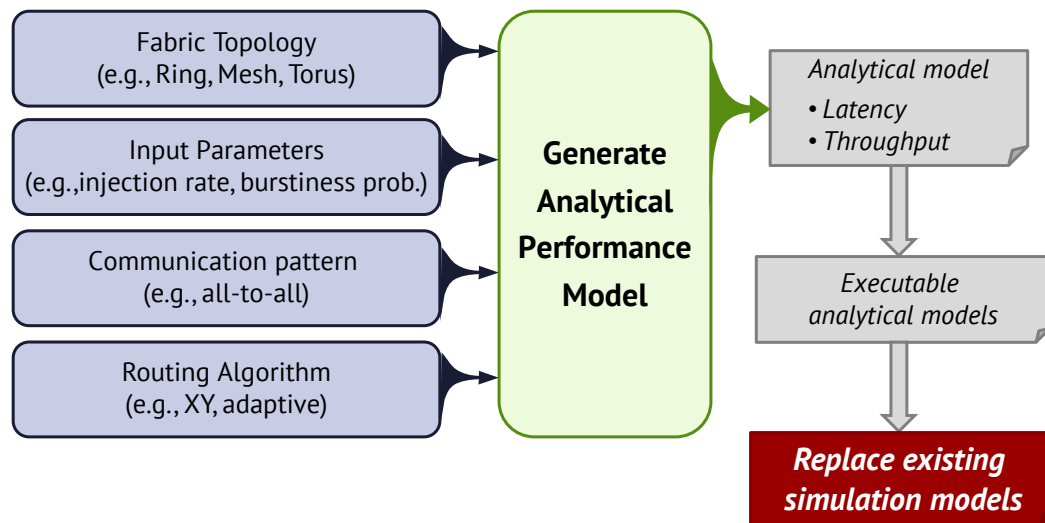
*In addition, deflection routing is not considered by prior work  
We extended our work to account for deflection routing*

- [1] Ogras et al. "An analytical approach for network-on-chip performance analysis." *IEEE Trans. on Computer-Aided Design of IC and Systems* (2010)
- [2] Bogdan et al. "Non-stationary traffic analysis and its implications on multicore platform design." *IEEE Trans. on Computer-Aided Design of IC and Systems* (2011)
- [3] Kiasari et al. "An analytical latency model for networks-on-chip." *IEEE Trans. on Very Large-Scale Integration Systems* 21.1 (2013)
- [4] Kashif et al. "Bounding buffer space requirements for real-time priority-aware networks." *Design Automation Conference (ASP-DAC)*, 2014
- [5] Qian et al. "A support vector regression (SVR)-based latency model for network-on-chip (NoC) architectures." *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* 35.3 (2016)
- [6] Bertsekas et al. *Data networks*. Vol. 2. New Jersey: Prentice-Hall International, 1992.
- [7] Walraevens, Joris. *Discrete-time queueing models with priorities*. Diss. Ghent University, 2004.
- [8] G. Bolch et al. *Queueing Networks and Markov Chains*. Wiley, 2006

9

9

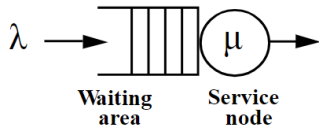
## Overview of the Automated Flow



10

10

## Background: Queuing Systems

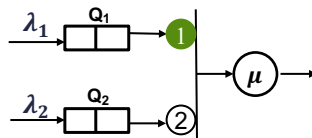


$\lambda$  = arrival rate,  $\mu$  = service rate

Server utilization ( $\rho$ ) =  $\frac{\lambda}{\mu}$

- Kendall's notation for queuing discipline: A/B/m
- Arrival and departure may have different distribution (e.g. Poisson (M), Deterministic (D), General (G)).

Priority Rule: (1>2)



$$W_1 = \frac{R_1}{1 - \rho_1}$$

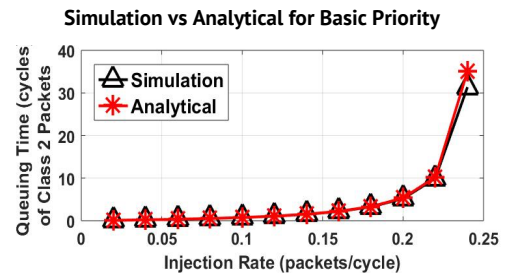
$$W_2 = \frac{R_2 + \rho_1 W_1}{1 - \rho_1 - \rho_2}$$

$$\rho_i = \frac{\lambda_i}{\mu_i}$$

W: average waiting time

T: service time

R: average residual time



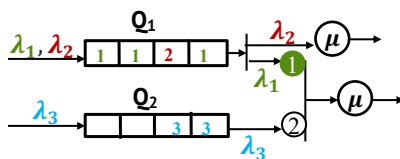
Geo/D/1 with  $T = 2$ , 1% average error

11

11

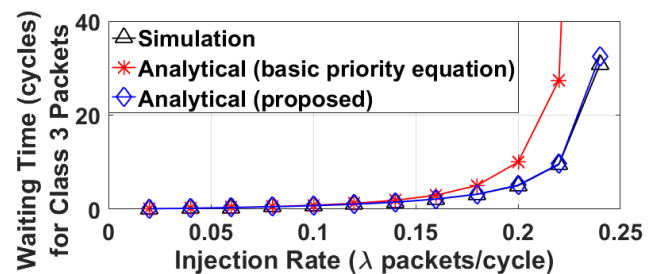
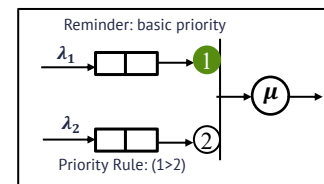
## Limitations of Basic Priority Based Models (1)

Split on **high** priority



Applying basic priority equation for **class 3** tokens results in **pessimistic solution**.

Reason: Not all packets of **class 1** will block packets of **class 3**. Packets of **class 3** can occupy the server if a **class 2** packets is being served.

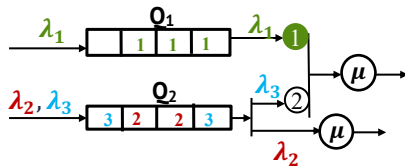


13

13

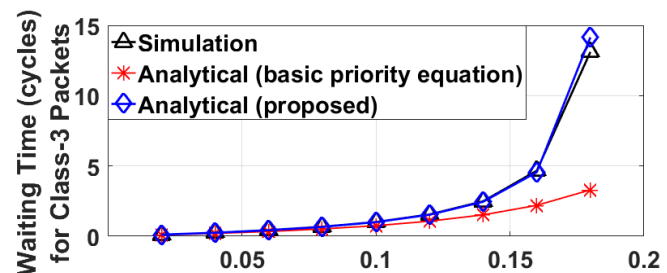
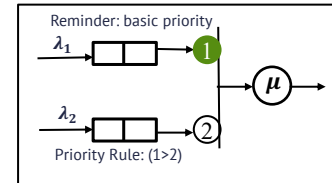
## Limitations of Basic Priority Based Models (2)

Split on **low** priority



Applying basic priority equation for **class 3** tokens results in **optimistic solution**.

Reason: Packets of **class 2** will have effect of **class 1** indirectly as **class 2** packets have to wait due to **class 3** packets



14

14

## Proposed Network Transformations

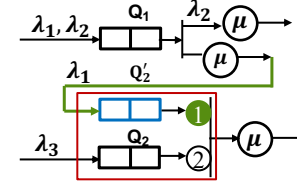
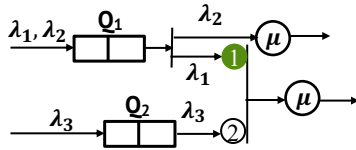
- **Extend decomposition method [1] to handle priority arbitration based multi-class networks in industry**
- **We identified two transformations**
  - ST: structural transformation
  - RT: service rate transformation
- **Complex priority-based networks are decomposed iteratively to systems of equivalent queues using ST/RT**
- **Obtain a closed form analytical expression for the equivalent system**

[1] G. Bolch et al. *Queueing Networks and Markov Chains*. Wiley. 2006

15

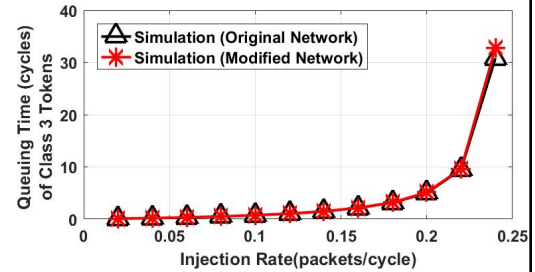
15

## Structural Transformation (ST)



- Packets of class 3 do not have to wait for all packets present in  $Q_1$ 
  - Packets of class 3 and 2 can be served at a same time
- Packets of class 3 will wait only when the server is busy serving class 1 packets
  - Need to decompose class 1 and class 2 packets
- Proposed transformation separates class 1 packets and puts them in a virtual queue ( $Q'_2$ )
- Equivalence is demonstrated by the result shown on the right

Comparison of Original and Modified Network



16

16

## Analytical Model after ST

- ST enables us to derive close-form analytical equations
- Expression of residual time of class 1 in  $Q'_2$

$$R_1^{Q'_2} = \frac{1}{2} \rho_3 (T - 1) + \frac{\frac{1}{2} \rho_3 T (C_{D_1}^2 + 1 - \mu)}{2} - \frac{\rho_1 \mu}{2}$$

- Waiting time of class 1 in  $Q'_2$

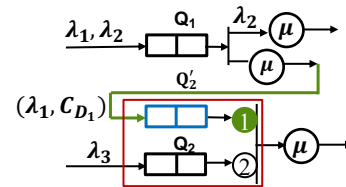
$$W_1^{Q'_2} = \frac{R_1^{Q'_2}}{1 - \rho_1}$$

- Residual time of class 3

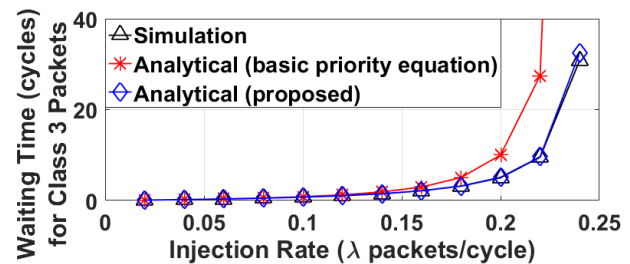
$$R_3 = R_1^{Q'_2} + \rho_1$$

- Finally, waiting time of class 3

$$W_3 = \frac{R_3 + \rho_1 W_1^{Q'_2}}{1 - \rho_1 - \rho_3}$$



Comparison of Simulation and Analytical Models



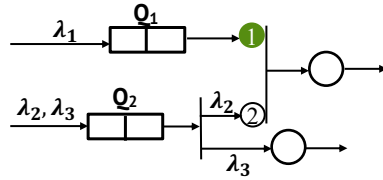
Geo/D/1 with  $T = 2$ , 3% average error

where  $C_{D_1}^2$ : obtained from decomposition technique

17

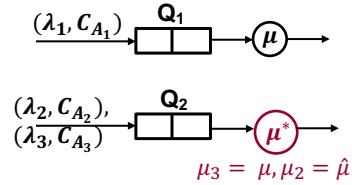
17

## Service Rate Transformation (RT)



### Observations

- Packets in  $Q_1$  effectively increase the service time of class 2 packet
  - Need to modify the service time of class 2 packet
- Challenging to model since not all packets in  $Q_2$  will wait for packets in  $Q_1$
- **Insight:** Modified service time of class 2 is independent of incoming distribution of class 3



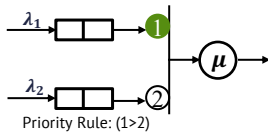
### Proposed Approach

- Decompose priority arbitration by modifying the service time of class 2
- Approximate first order moments of modified service rate ( $\hat{\mu}$ )
- $\hat{\mu} = \frac{1}{\bar{T}} = \frac{1}{T + \Delta T}$ ,  $\Delta T$  is computed through busy-period concept

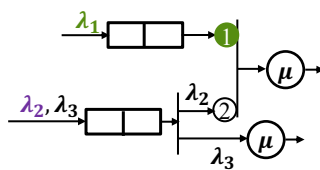
18

18

## Service Rate Transformation (RT): 2<sup>nd</sup> Moment



(a)



(b)

### Simple Priority

$$W_2 = \frac{R_2 + \rho_1 W_1}{1 - \rho_1 - \rho_2}$$

### Service rate transformation

$$W_2 = \frac{\hat{R}_2}{1 - \hat{\rho}_2} + \Delta T, \text{ where } \hat{\rho}_2 = \lambda_2(T + \Delta T)$$

Decoupled from the busy period. It is common for both traffic classes 2 and 3 in Figure (b), while the  $\Delta T$  term applies only to traffic class 2

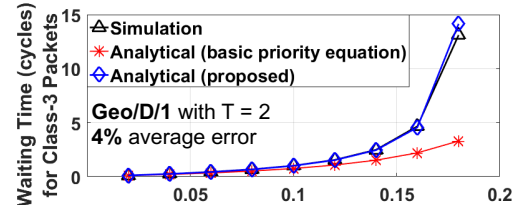
### Re-arranging the terms to get $\hat{R}_2$ :

$$\hat{R}_2 = (W_2 - \Delta T)(1 - \hat{\rho}_2)$$

$$W_2 = \frac{\hat{R}_2 + R_3}{1 - \rho_3 - \hat{\rho}_2} + \Delta T$$

$$W_3 = \frac{\hat{R}_2 + R_3}{1 - \rho_3 - \hat{\rho}_2}$$

### Comparison of Simulation and Analytical Models



20

20

# Model Generation Flow

**Input:** Injection rates for all traffic classes ( $\lambda$ ), Network topology, Routing algorithm

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

For each Queue and traffic class:

## 1. Do structural transformation

Get all classes having higher priority and calculate  $C_A$

Get reference waiting time expression ( $W_{ref}$ ) using  $C_A$

## 2. Do service rate transformation

Calculate modified service time ( $\hat{T}$ ) using  $W_{ref}$

Calculate effective residual time ( $\hat{R}$ ) using  $\hat{T}$

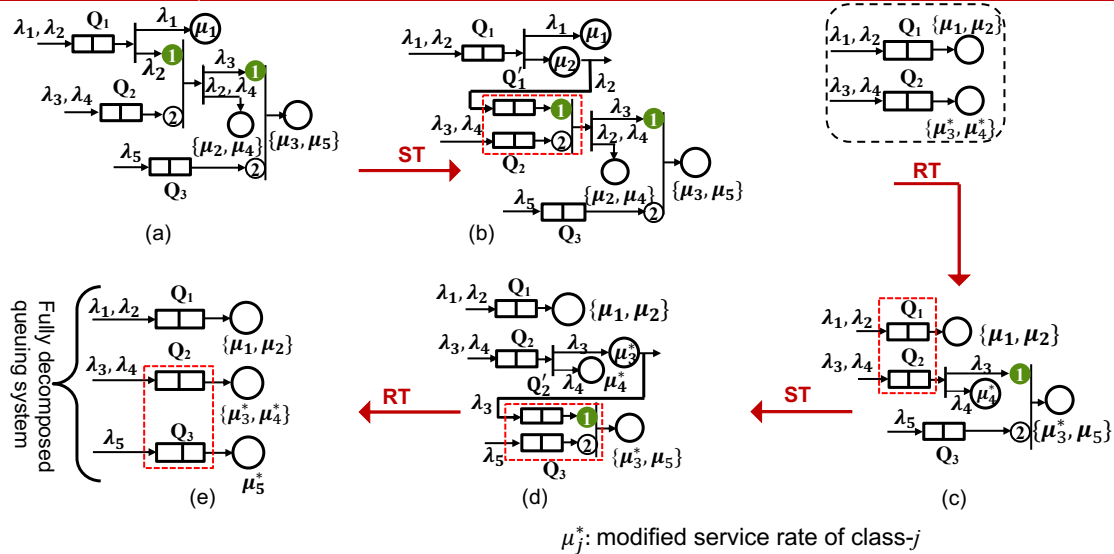
Calculate waiting time ( $W$ ) using  $\lambda$ ,  $\hat{T}$  and  $\hat{R}$

\*A graphical illustration of a representative example given in the next slide

21

21

# A Representative Example



ST  $\rightarrow$  Structural Transformation RT  $\rightarrow$  Service Rate Transformation

22

22

## 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  $\lambda_{max}$
  - $\lambda_{max}$  is the injection rate at which the maximum server utilization is 1



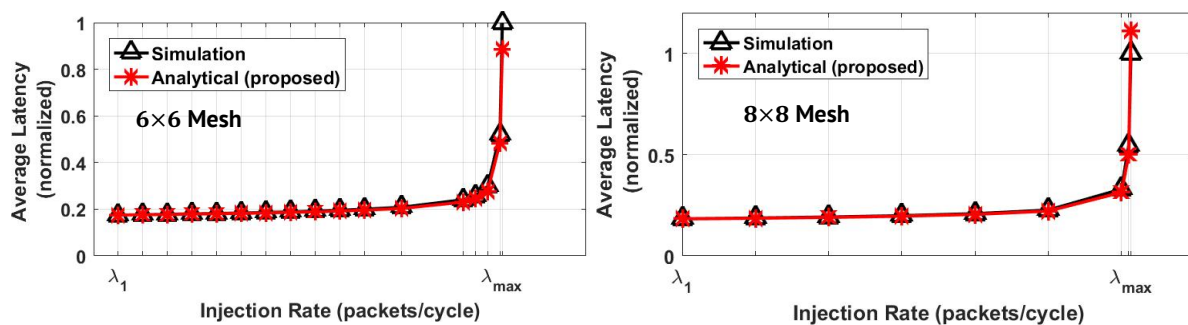
\*Mandal, Sumit K., et al. "Analytical Performance Models for NoCs with Multiple Priority Traffic Classes." *ACM Transactions on Embedded Computing Systems (TECS)* 18.5s (2019): 1-21.

23

23

## Evaluation on xPLORE: Mesh Topology (Geo Traffic)

- Traffic pattern is all to all (i.e. each node is sending tokens to all nodes) with YX routing
- Injection rate for each source destination pair is equal



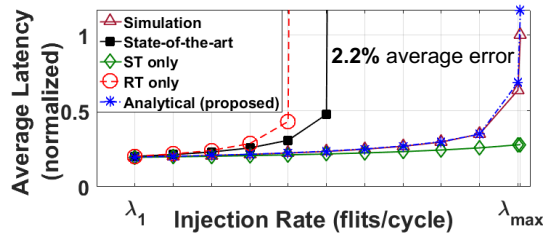
- We observe **less than 4% error** between simulation and analysis for mesh
- Analytical models are **2-3 orders of magnitude faster** than simulation models

24

24

## Verification with Intel® Xeon® Scalable Server (Geo Traffic)

- **One variation of scalable Intel® Xeon®**
  - 26 tiles with Core + LLC, 2 memory controllers on a 6x6 mesh
- **Synthetic injection**
  - All cores send packets to all caches
- **Validated latencies for cache-coherency flow**



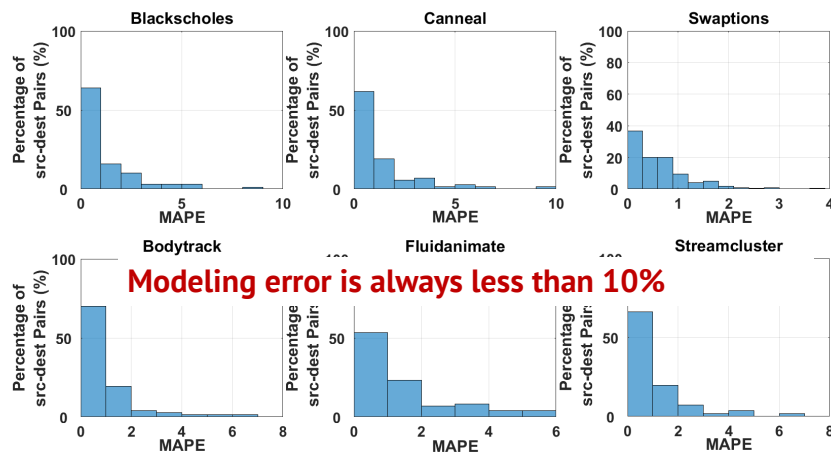
| Model Accuracy (%) |                 |              |
|--------------------|-----------------|--------------|
| LLC Hit Rate (%)   | Address Network | Data Network |
| 100                | 98.8            | 93.9         |
| 50                 | 97.7            | 98.1         |
| 0                  | 97.7            | 98.0         |

25

25

## Evaluation with Real Applications

- **Collected real app traces from gem5 in Full System mode**
  - Applications (16-threaded) from PARSEC suite
  - Average statistics over 1M cycles



26

26

## Agenda

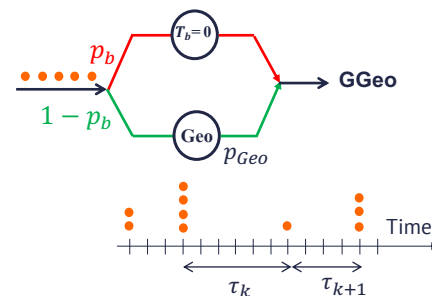
- **Motivation**
- **Preliminary Work-1: Performance Analysis of NoCs**
  - Priority-Aware NoCs
  - NoCs with Bursty Traffic
  - NoCs with Deflection Routing
- **Preliminary Work-2: Communication-Aware Hardware Accelerator for DNNs**
- **Proposed Work**
  - Multi-Objective Optimization to Design Optimized NoC
  - Accelerator for Graph Convolutional Network (GCN)
- **Conclusions**

27

27

## Analytical Modeling with Bursty Traffic

- **Industrial NoCs execute bursty traffic**
  - Analytical models assuming input traffic as geometric can not capture burstiness
- **Generalized Geometric distribution (GGeo) has been studied in academia to address bursty traffic**
- **GGeo can be fully described by the first two moments**
- **Probability of burstiness ( $p_b$ ) is approximated from input traffic**

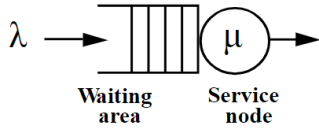


- Selection of branches is Bernoulli trial
- $p_b$ : controls the level of burstiness
- $p_{Geo}$ : determines the inter-arrival time between the bulk of arrivals

28

28

## Background: Queuing Systems with GGeo Input

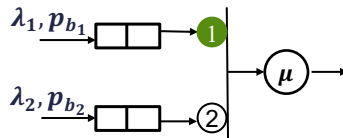


$\lambda$  = arrival rate,  $\mu$  = service rate

Server utilization ( $\rho$ ) =  $\frac{\lambda}{\mu}$

- Kendall's notation for queuing discipline: A/B/m
- Arrival and departure may have different distribution (e.g. Poisson (M), Deterministic (D), General (G)).

Priority Rule: (1>2)



$$W_1 = \frac{R_1 + T \frac{p_{b1}}{1-p_{b1}}}{1-\rho_1}$$

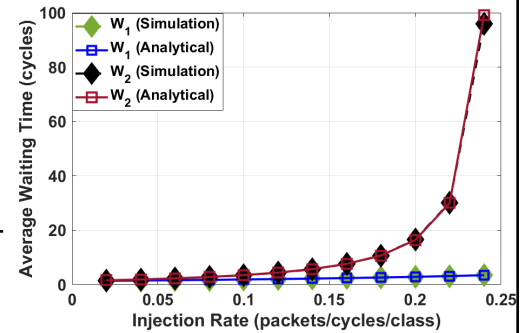
$$W_2 = \frac{R_2 + T \frac{p_{b2}}{1-p_{b2}} + \rho_1 W_1}{1-\rho_1-\rho_2}$$

$$\rho_i = \frac{\lambda_i}{\mu_i}$$

W: average waiting time

T: service time

R: average residual time



GGeo/D/1 with  $p_{b1} = p_{b2} = 0.4, T = 2$ ,  
1% average error

29

29

## Analytical Model after Structural Transformation

- ST enables us to derive close-form analytical equations
- Expression of residual time of class 1 in  $Q'_2$

$$R_1^{Q'_2} = \frac{1}{2} \rho_3 (T - 1) + \frac{\frac{1}{2} \rho_3 T (C_{D1}^2 + 1 - \mu)}{2} - \frac{\rho_1 \mu}{2}$$

- Waiting time of class 1 in  $Q'_2$

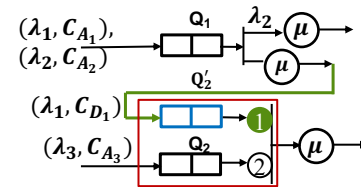
$$W_1^{Q'_2} = \frac{R_1^{Q'_2} + T \frac{p_{b1}}{1-p_{b1}}}{1-\rho_1}$$

- Residual time of class 3

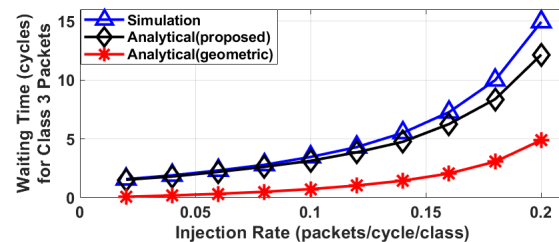
$$R_3 = R_1^{Q'_2} + \rho_1$$

- Finally, waiting time of class 3

$$W_3 = \frac{R_3 + \rho_1 W_1^{Q'_2} + T \frac{p_{b3}}{1-p_{b3}}}{1-\rho_1-\rho_3}$$



Comparison of Simulation and Analytical Models



GGeo/D/1 with  $p_b = 0.4, T = 2$ , 6% average error

31

where  $C_{D1}^2$ : obtained from decomposition technique

31

## Computing $\mu_2^*, C_{s2}^*$

- We first compute the probability that there is no packet of class-2 in  $Q_2$  ( $p_2(0)$ ) through entropy maximization method [1]

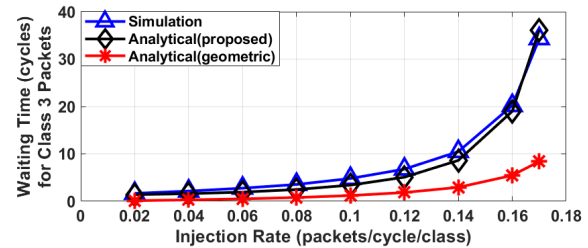
$$p_2(0) = 1 - \rho_2 - \rho_2 \frac{n_{21}}{\rho_1 + n_{21}}$$

- $\mu_2^*$  is computed using  $p_2(0)$

$$T_2^* = \frac{1 - p_2(0)}{\lambda_2}, \mu_2^* = 1/T_2^*$$

- $C_{s2}^*$  is computed through the methodology described in [2]

$$(C_{s2}^*)^2 = \frac{(1 - \rho_2^*)(2n_2 - \rho_2^*) - \rho_2^* C_{A2}^*}{(\rho_2^*)^2}$$



GGeo/D/1 with  $p_b = 0.4$ ,  $T = 2$ , 7% average error

[1] Kouvatsos, Demetres, and Irfan Awan. "Entropy maximisation and open queueing networks with priorities and blocking." *Performance Evaluation* 51.2-4 (2003): 191-227.

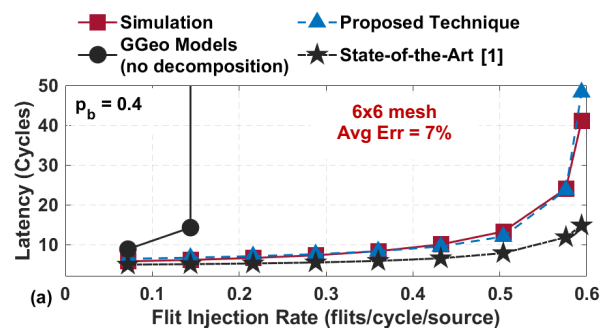
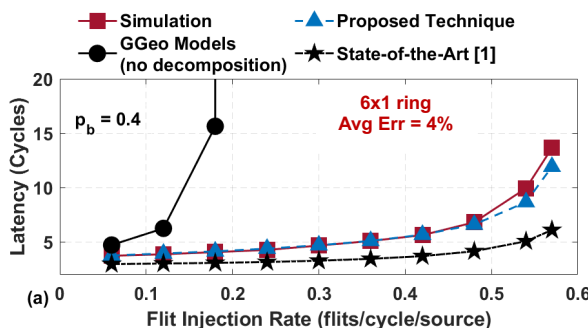
[2] Kouvatsos, Demetres D., and Nasreddine Tabet-Aouel. "GGeo-type approximations for general discrete-time queueing systems." *Proceedings of the IFIP TC6 Task Group/WG6. 4 International Workshop on Performance of Communication Systems: Modelling and Performance Evaluation of ATM Technology*. 1993.

33

33

## Evaluations with Synthetic Bursty Traffic\*

- Achieved <10% modeling error for rings & mesh with bursty traffic ( $p_b = 0.4$ )



- GGeo models without decomposition technique overestimates the latency
- State-of-the-art models which ignore bursty traffic underestimates the latency

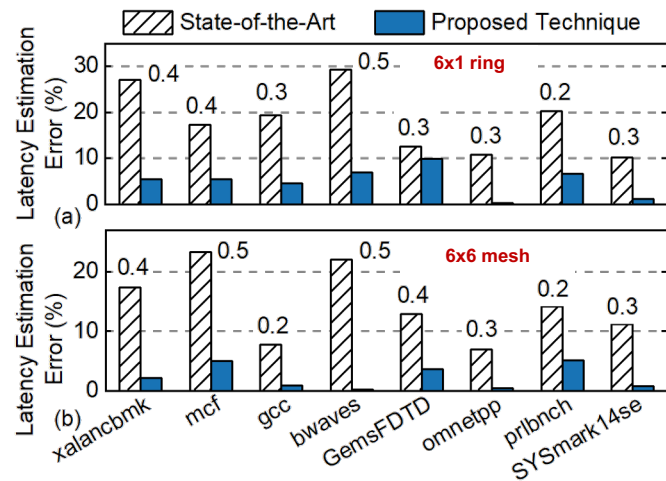
- \*Mandal, Sumit K., et al. "Analytical Performance Modeling of NoCs under Priority Arbitration and Bursty Traffic." *IEEE Embedded Systems Letters* (2020).
- [1] Mandal, Sumit K., et al. "Analytical Performance Models for NoCs with Multiple Priority Traffic Classes," *ACM Transactions on Embedded Computing Systems (TECS)* 18.5s (2019): 1-21.

34

34

## Evaluations with Real Applications with Burstiness

- Evaluated the proposed model with real applications with different probability of burstiness (shown on the top of each bar)
- Achieve **4%** modeling error on average
- State-of-the-art model does not consider burstiness
  - Errors are as high as **30%**



35

35

## Agenda

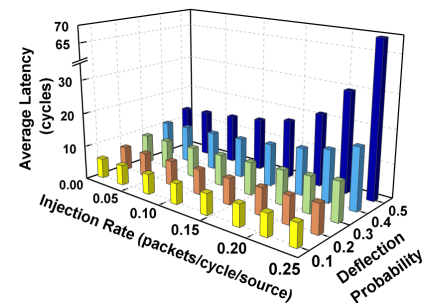
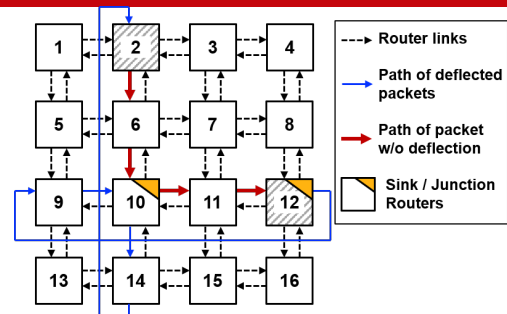
- Motivation
- Preliminary Work-1: Performance Analysis of NoCs
  - Priority-Aware NoCs
  - NoCs with Bursty Traffic
  - NoCs with Deflection Routing
- Preliminary Work-2: Communication-Aware Hardware Accelerator for DNNs
- Proposed Work
  - Multi-Objective Optimization to Design Optimized NoC
  - Accelerator for Graph Convolutional Network (GCN)
- Conclusions

36

36

## Background: Deflection Routing

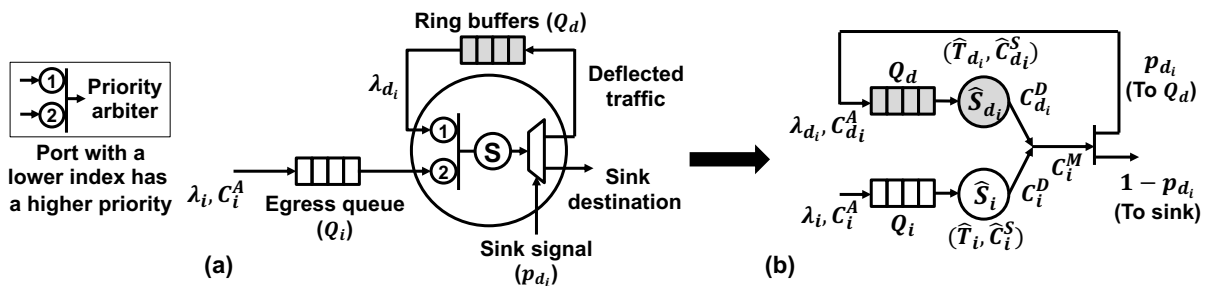
- **Packets are deflected if ingress at the junction (Node 10) or the destination (Node 12) is full**
  - The deflected packets circulate within the same row or column
  - Increase congestion towards the source
- **NoC analytical models need to take deflection into account**
  - Average latency varies in the range of 6-70 cycles for an injection rate of 0.25 when deflection probability varies from 0.1 to 0.5



37

37

## Analytical Modeling for a System with Single Class



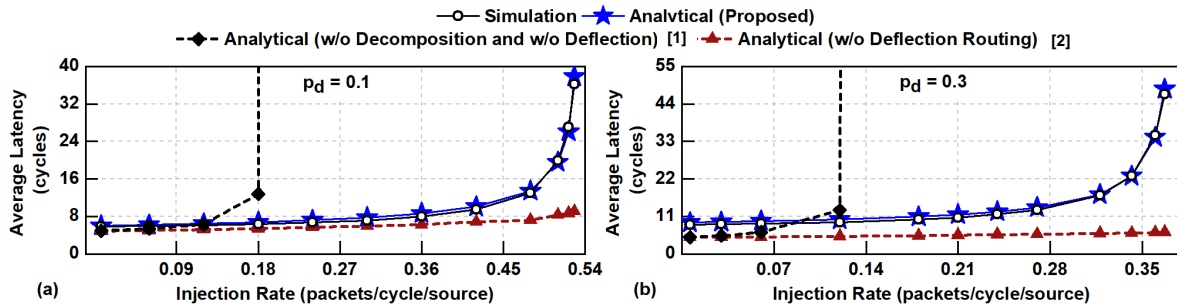
- The behavior of deflected packets is modeled by a buffer ( $Q_d$ )
- Deflected packets and the packet in the egress queue ( $Q_i$ ) form a priority-aware queuing system
- Rate of deflected packets ( $\lambda_{d_i}$ ) and coefficient of variation of deflected packets ( $C_{d_i}^A$ ) are computed

38

38

## Evaluation on 6x6 Mesh with Geometric Input\*

- Achieved <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

\*Mandal, Sumit K., et al. "Performance analysis of priority-aware NoCs with deflection routing under traffic congestion." *Proceedings of the 39th International Conference on Computer-Aided Design*. 2020.

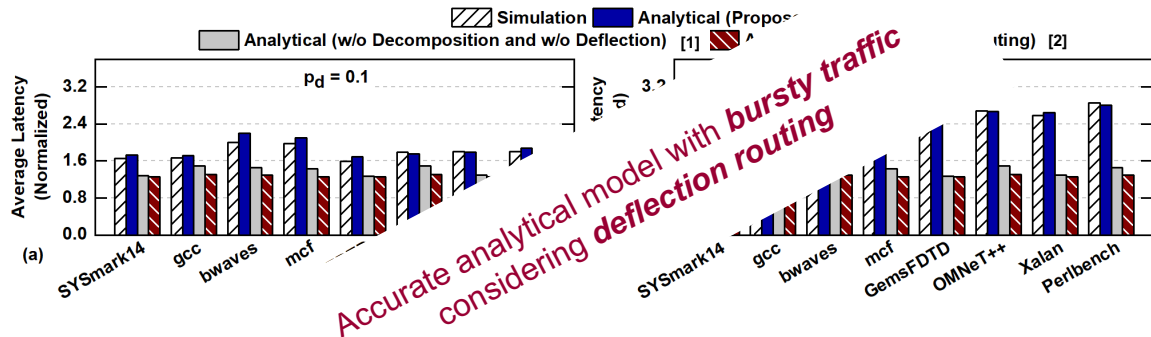
[1] Kiasari, Abbas Eslami, Zhonghai Lu, and Axel Jantsch. "An analytical latency model for networks-on-chip." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 21.1 (2012): 113-123.

[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.

41

## Evaluation on Real Application (Bursty) with 6x6 Mesh

- Achieved <5% modeling error on average for  $p_d = 0.1$  and  $p_d = 0.3$



[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.

[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.

42

42

## Agenda

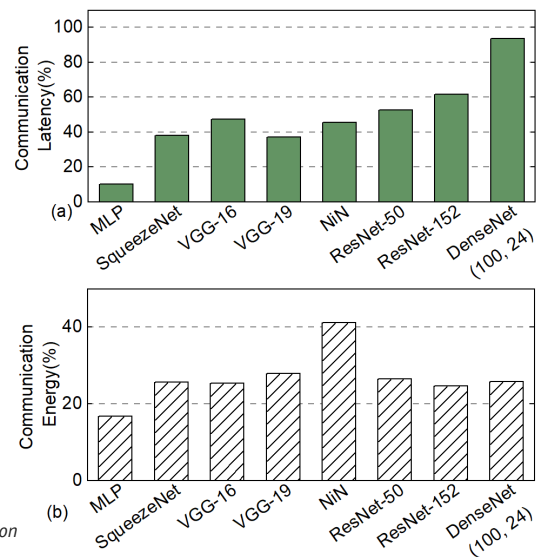
- **Motivation**
- **Preliminary Work-1: Performance Analysis of NoCs**
  - Priority-Aware NoCs
  - NoCs with Bursty Traffic
  - NoCs with Deflection Routing
- **Preliminary Work-2: Communication-Aware Hardware Accelerator for DNNs**
- **Proposed Work**
  - Multi-Objective Optimization to Design Optimized NoC
  - Accelerator for Graph Convolutional Network (GCN)
- **Conclusions**

43

43

## Bottleneck of the DNN Accelerator: Communication

- **Communication volume increases with deeper and denser DNNs**
- **Target hardware: In-memory Computing based**
  - Exhibits less latency from off-chip memory accesses
- **Up to 90% of the inference latency comes from communication**
- **Up to 40% of the energy is consumed by communication**
- **Need for an efficient on-chip communication strategy for DNN accelerators**



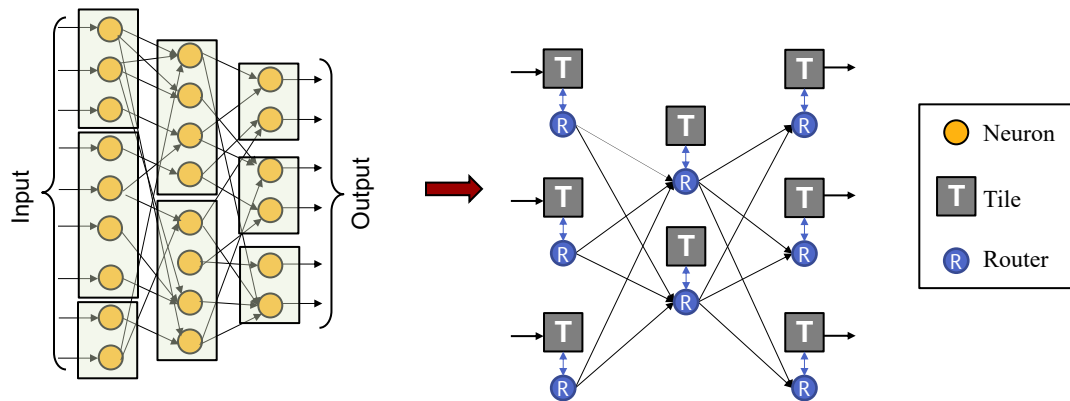
[1] Chen, Pai-Yu, Xiaochen Peng, and Shimeng Yu. "NeuroSim: A circuit-level macro model for benchmarking neuro-inspired architectures in online learning." *IEEE Trans. on CAD of Integrated Circuits and Systems* 37.12 (2018): 3067-3080.

44

44

## Background of In-Memory Computing (IMC)

- A set of nodes are mapped to network-on-chip (NoC) routers
  - Need to balance the computation and minimize the communication overhead

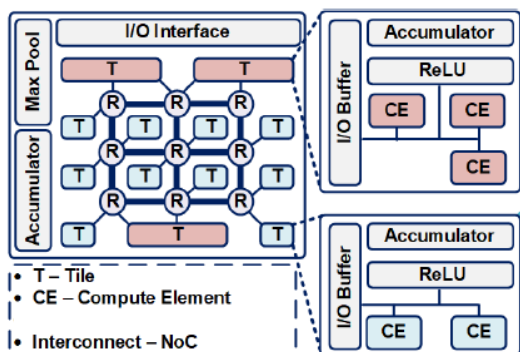


45

45

## Operations in IMC-based Hardware

- Hierarchical structure: NoC → Tiles → Computing Elements (CE)



### Step-by-step Operation

1. Compute: Tiles with *Layer 1* nodes compute their output
2. Communicate: Send their output to the tiles with *Layer 2* nodes
3. Compute: Tiles with *Layer 2* nodes compute their output

....

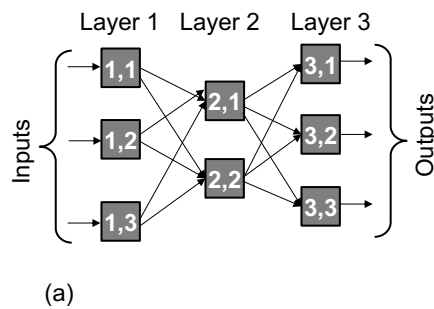
**Communication between multiple tiles**  
 - Crucial in overall performance

46

46

## Achieving Minimum Possible Communication Latency: Example

- **Target:** to achieve minimum possible communication latency for a given DNN on IMC-based hardware platform
- **Example of *minimum possible communication latency***



(b)

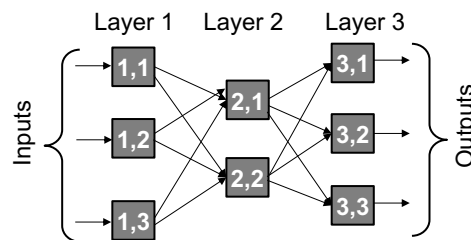
| Layer 1 – Layer 2   |       | Layer 2 – Layer 3   |       |
|---------------------|-------|---------------------|-------|
| Link                | Cycle | Link                | Cycle |
| $L_{1,1} - L_{2,1}$ | 1     | $L_{2,1} - L_{3,1}$ | 4     |
| $L_{1,2} - L_{2,2}$ | 1     | $L_{2,2} - L_{3,2}$ | 4     |
| $L_{1,1} - L_{2,2}$ | 2     | $L_{2,1} - L_{3,2}$ | 5     |
| $L_{1,3} - L_{2,1}$ | 2     | $L_{2,2} - L_{3,1}$ | 5     |
| $L_{1,2} - L_{2,1}$ | 3     | $L_{2,2} - L_{3,2}$ | 6     |
| $L_{1,3} - L_{2,2}$ | 3     | $L_{2,1} - L_{3,3}$ | 6     |

47

47

## Key Insights

- A single router can not send/receive more than one packet in a cycle. However, multiple routers can send/receive packets in parallel.
- Since a router can send/receive only one packet in a cycle, the congestion is minimum i.e., no more than a transaction is scheduled through a particular link.

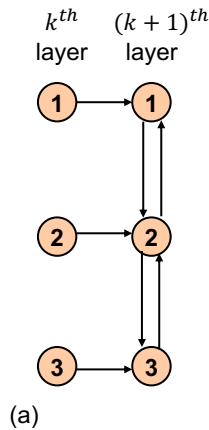


48

48

## Our Solution: Custom Interconnect (A Simple Case)

- Two consecutive layers consists of equal number of routers
  - Three routers in this case
- Each router can send/receive only one packet at one cycle
- In one round of communication each routers in  $k^{th}$  layer send a packet to each router in  $(k+1)^{th}$  layer
- One transaction is finished in 3 cycles
  - Minimum possible latency



(b)

| Link                        | Cycle | Transaction             |
|-----------------------------|-------|-------------------------|
| $L_{k,1} - L_{(k+1),1}$     | 1     | $R_{k,1} - R_{(k+1),1}$ |
| $L_{k,2} - L_{(k+1),2}$     | 1     | $R_{k,2} - R_{(k+1),2}$ |
| $L_{k,3} - L_{(k+1),3}$     | 1     | $R_{k,3} - R_{(k+1),3}$ |
| $L_{(k+1),1} - L_{(k+1),2}$ | 2     | $R_{k,1} - R_{(k+1),2}$ |
| $L_{(k+1),2} - L_{(k+1),3}$ | 2     | $R_{k,2} - R_{(k+1),3}$ |
| $L_{(k+1),3} - L_{(k+1),2}$ | 2     | $R_{k,3} - R_{(k+1),2}$ |
| $L_{(k+1),2} - L_{(k+1),1}$ | 2     | $R_{k,2} - R_{(k+1),1}$ |
| $L_{(k+1),2} - L_{(k+1),3}$ | 3     | $R_{k,1} - R_{(k+1),3}$ |
| $L_{(k+1),2} - L_{(k+1),1}$ | 3     | $R_{k,3} - R_{(k+1),1}$ |

49

49

## Proof by Induction

### New Transactions:

**Horizontal Link** ( $L_{1,N} - L_{2,N}$ ) will be active on cycle 1 which does not overlap with any of the transaction of the configuration with  $(N-1)$  routers

**Upward Vertical Link:** New transactions will carry the packet from the router at the position  $(1,N)$ . The vertical link ( $L_{2,N} - L_{2,(N-1)}$ ) will be activated on cycle 1. All other new transactions will happen 1 cycle after the last transaction for the configuration with  $(N-1)$  routers.

**Downward Vertical Link:** New transactions will happen only in  $(L_{2,(N-1)} - L_{2,N})$

### Schedules in general:

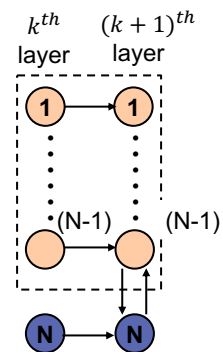
**Horizontal Link:**  $L_{k,n} - L_{k+1,n}$  in cycle 1 carries transaction  $T_{k,n} - T_{k+1,n}$

**Upward Vertical Link:**  $L_{k+1,n} - L_{k+1,n-1}$  in  $m^{th}$  cycle carries the transaction

$T_{k,n+m-2} - T_{k+1,n-1}$ , if the transaction exists

**Downward Vertical Link:**  $L_{k+1,n} - L_{k+1,n+1}$  in  $m^{th}$  cycle carries the transaction

$T_{k,n-m+2} - T_{k+1,n+1}$ , if the transaction exists



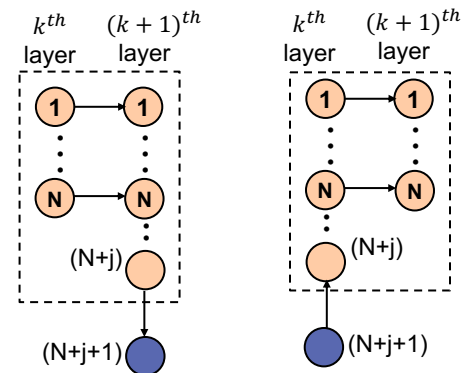
50

50

## Other Cases

- We also construct the schedules when the number of the routers in two consecutive layers are not equal
- Detailed proof is shown in [1]
- We obtain minimum possible communication latency for a give DNN

[1] Mandal, Sumit K., Gokul Krishnan, Chaitali Chakrabarti, Jae-Sun Seo, Yu Cao, and Umit Y. Ogras. "A Latency-Optimized Reconfigurable NoC for In-Memory Acceleration of DNNs." *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 10, no. 3 (2020): 362-375.



51

51

## Experimental Setup

- Performance of the computation fabric and communication fabric is estimated separately
  - Computation fabric: NeuroSim
  - Communication fabric: BookSim
- A wide range of DNNs is chosen
  - Number of parameters vary from 0.43M-45.6M
- Baseline design
  - Mesh NoC
  - Each tile is connected to a dedicated router

52

52

## Experimental Results\*

- **Layer-wise improvements shown for VGG-19**

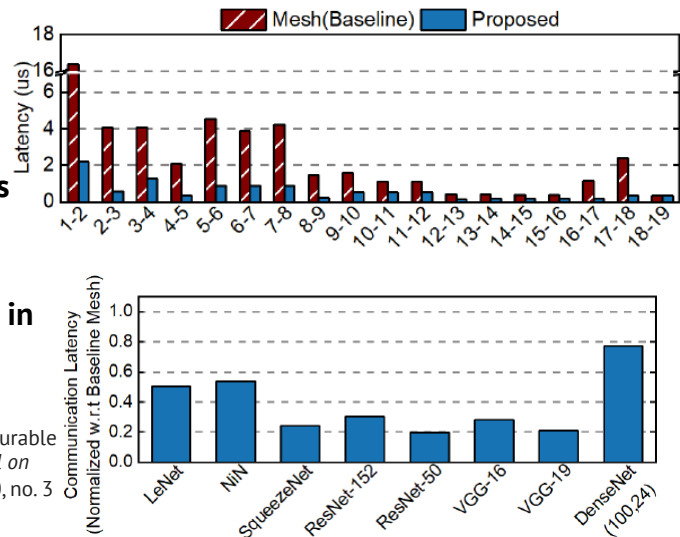
- Up to 8× improvement in communication latency

- **Improvement in communication latency is shown for different DNNs**

- Normalized with respect to the baseline communication latency

- **Results in up to 25% improvement in inference latency**

\*Mandal, Sumit K. et al. "A Latency-Optimized Reconfigurable NoC for In-Memory Acceleration of DNNs." *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 10, no. 3 (2020): 362-375.



53

53

## Agenda

- **Motivation**
- **Preliminary Work-1: Performance Analysis of NoCs**
  - Priority-Aware NoCs
  - NoCs with Bursty Traffic
  - NoCs with Deflection Routing
- **Preliminary Work-2: Communication-Aware Hardware Accelerator for DNNs**
- **Proposed Work**
  - Multi-Objective Optimization to Design Optimized NoC
  - Accelerator for Graph Convolutional Network (GCN)
- **Conclusions**

54

54

## Proposed Work-1: Optimizing NoC with Analytical Models

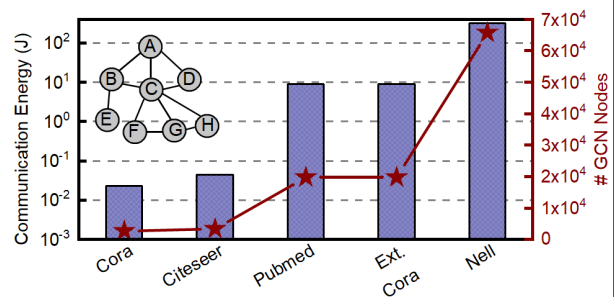
- Majority of the NoC used for DNN accelerators incorporate round robin (RR) arbitration
- However, RR arbitration does not provide latency fairness
- Weighted Round Robin (WRR) arbitration
  - Example: Packets originate from a source which is far from the destination is provided higher weights during arbitration
- Challenge: Assigning weights to different flows in the NoC
  - Distribute weights to the flows to minimize overall latency
- Proposal: Construct analytical model of NoCs with WRR
  - Use the analytical model to perform multi-objective optimization

55

55

## Proposed Work-2: Communication-Aware Accelerator for GCN

- Graph Convolutional Network (GCN) exhibits very high volume of communication
- In-Memory Computing (IMC) worsens the communication latency since all memory elements are on-chip
- Challenge: Straightforward implementation is unrealistic
  - One router for each node of the GCN
  - Up to 200 J of communication energy
- Proposal: Construct an efficient NoC which balances computation as well as communication



56

56

## Conclusion

- **Communication is a critical component for recently emerged complex applications**
- **Cycle-accurate NoC simulation is a bottleneck in full system performance evaluation**
- **Constructed *fast and accurate* analytical model for priority-aware NoCs which can handle multiple classes, bursty traffic and deflection routing**
- **Constructed *latency-optimized NoC* for DNN accelerator**
- **Next steps:**
  - Construct analytical models for NoCs with WRR arbitration
  - Communication-aware accelerator for GCNs

57

57

## List of Publications

### NoC Performance Analysis

1. **Mandal, S. K.**, Ayoub, R., Kishinevsky, M., & Ogras, U. Y. (2019). Analytical performance models for NoCs with multiple priority traffic classes. *ACM Transactions on Embedded Computing Systems (TECS)*, 18(5s), 1-21.
2. **Mandal, S. K.**, Ayoub, R., Kishinevsky, M., Islam, M. M., & Ogras, U. Y. (2020). Analytical Performance Modeling of NoCs under Priority Arbitration and Bursty Traffic. *IEEE Embedded Systems Letters*.
3. **Mandal, S. K.**, Krishnakumar, A., Ayoub, R., Kishinevsky, M., & Ogras, U. Y. (2020, November). Performance analysis of priority-aware NoCs with deflection routing under traffic congestion. In *Proceedings of the 39th International Conference on Computer-Aided Design* (pp. 1-9).
4. Ayoub, R., Kishinevsky, M., **Mandal, S. K.**, & Ogras, U. Y. (2020, November). Analytical modeling of NoCs for fast simulation and design exploration. In *Proceedings of the Workshop on System-Level Interconnect: Problems and Pathfinding Workshop* (pp. 1-1).

### Communication-Aware Hardware Accelerator for DNNs

5. **Mandal, S. K.**, Krishnan, G., Chakrabarti, C., Seo, J. S., Cao, Y., & Ogras, U. Y. (2020). A Latency-Optimized Reconfigurable NoC for In-Memory Acceleration of DNNs. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS)*, 10(3), 362-375.
6. Krishnan, G., **Mandal, S. K.**, Chakrabarti, C., Seo, J. S., Ogras, U. Y., & Cao, Y. (2020). Interconnect-aware area and energy optimization for in-memory acceleration of DNNs. *IEEE Design & Test*, 37(6), 79-87.
7. Krishnan, G., **Mandal, S. K.**, Chakrabarti, C., Seo, J. S., Ogras, U. Y., & Cao, Y. Impact of On-Chip Interconnect on In-Memory Acceleration of Deep Neural Networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, To Appear.

58

58

## Other Work: Power Management for Heterogeneous SoCs

8. **Mandal, S. K.**, Bhat, G., Patil, C. A., Doppa, J. R., Pande, P. P., & Ogras, U. Y. (2019). Dynamic resource management of heterogeneous mobile platforms via imitation learning. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 27(12), 2842-2854.
9. **Mandal, S. K.**, Bhat, G., Doppa, J. R., Pande, P. P., & Ogras, U. Y. (2020). An energy-aware online learning framework for resource management in heterogeneous platforms. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 25(3), 1-26.
10. **Mandal, S. K.**, Ogras, U. Y., Doppa, J. R., Ayoub, R. Z., Kishinevsky, M., & Pande, P. P. (2020, July). Online adaptive learning for runtime resource management of heterogeneous SoCs. In *2020 57th ACM/IEEE Design Automation Conference (DAC)* (pp. 1-6). IEEE.
11. Gupta, U., **Mandal, S. K.**, Mao, M., Chakrabarti, C., & Ogras, U. Y. (2019). A deep Q-learning approach for dynamic management of heterogeneous processors. *IEEE Computer Architecture Letters*, 18(1), 14-17.
12. Bhat, G., **Mandal, S. K.**, Gupta, U., & Ogras, U. Y. (2018, November). Online learning for adaptive optimization of heterogeneous SoCs. In *Proceedings of the International Conference on Computer-Aided Design* (pp. 1-6).
13. Pasricha, S., Ayoub, R., Kishinevsky, M., **Mandal, S. K.**, & Ogras, U. Y. (2020). A survey on energy management for mobile and IoT devices. *IEEE Design & Test*, 37(5), 7-24.

59

59

## Other Collaborative Work

14. **Task scheduling with imitation learning (IL)**: Krishnakumar, A., Arda, S. E., Goksoy, A. A., **Mandal, S. K.**, Ogras, U. Y., Sartor, A. L., & Marculescu, R. (2020). Runtime Task Scheduling Using Imitation Learning for Heterogeneous Many-Core Systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(11), 4064-4077.
15. **Flexible Hybrid Electronics (FHE)**: Bhat, G., Gao, H., **Mandal, S. K.**, Ogras, U. Y., & Ozev, S. (2020). Determining mechanical stress testing parameters for the designs with low computational overhead. *IEEE Design & Test*, 37(4), 35-41.
16. **3D Ultrasounds Imaging**: Zhou, J., **Mandal, S. K.**, West, B. L., Wei, S., Ogras, U. Y., Kripfgans, O. D., ... & Chakrabarti, C. (2020). Front-End Architecture Design for Low-Complexity 3-D Ultrasound Imaging Based on Synthetic Aperture Sequential Beamforming. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*.

### Book Chapter:

17. **Mandal S.K.**, Krishnakumar A., Ogras U. Y. (2021). Energy-Efficient Networks-on-Chip Architectures: Design and Run-Time Optimization: Network-on-Chip Security and Privacy pp55-75, Springer
18. Mandal, S. K., "Network-on-Chip (NoC) Performance Analysis and Optimization for Deep Learning Applications," Technical Report, June, 2021, Available [online] [https://sumitkmandal.ece.wisc.edu/data/prelims\\_report\\_sumit\\_v3.pdf](https://sumitkmandal.ece.wisc.edu/data/prelims_report_sumit_v3.pdf)

60

60