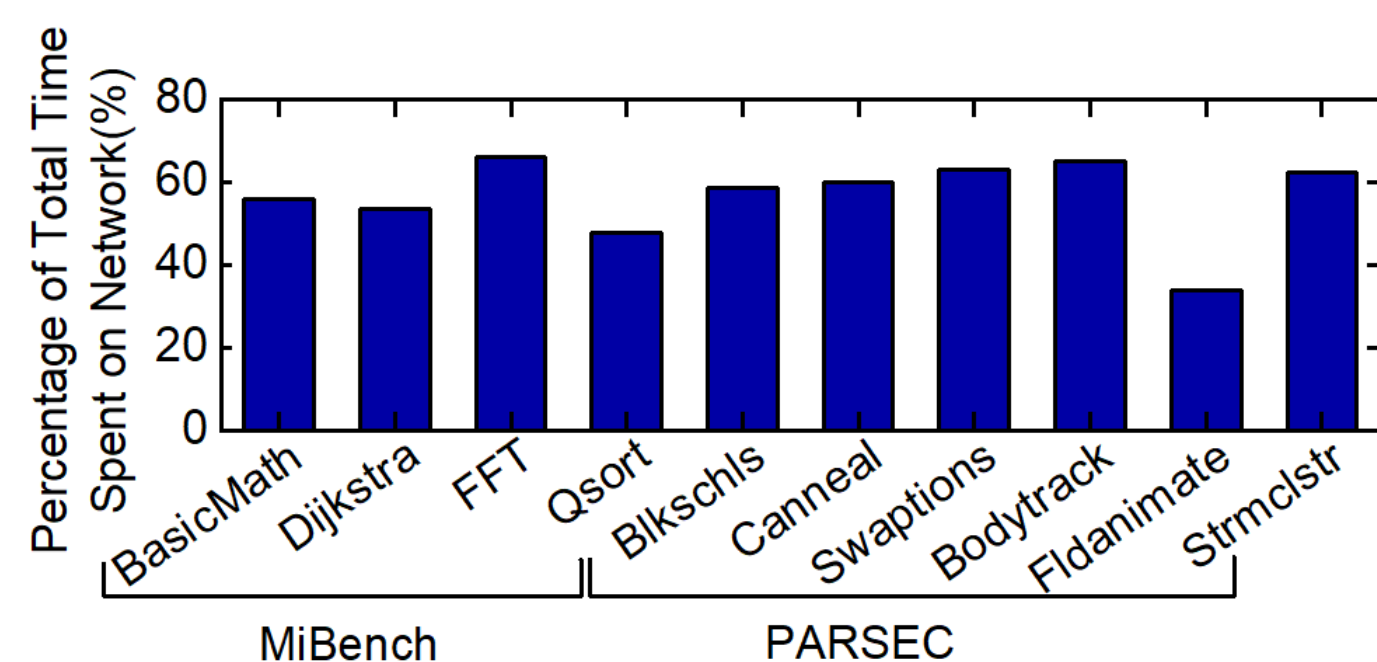


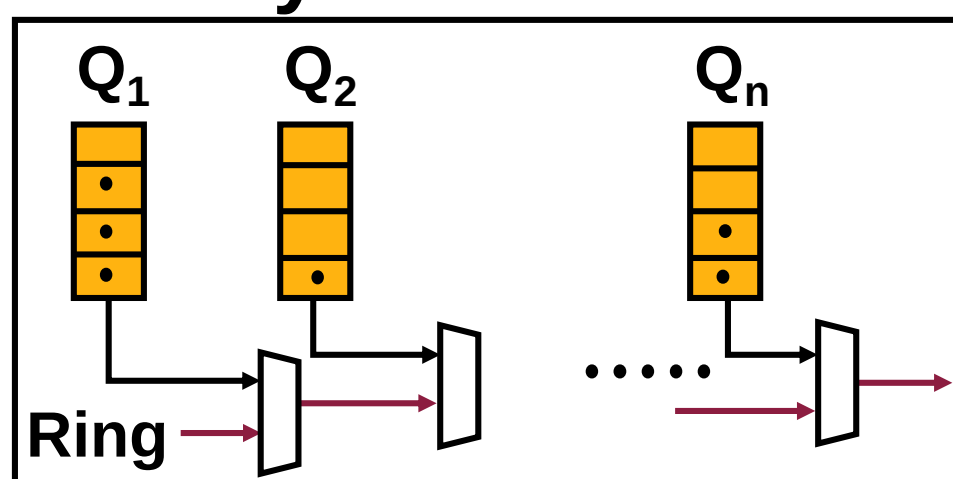
Motivation

- Emerging applications require long (~min) simulations for meaningful power-performance analysis
- Significant amount of sim. time is spent on NoC
- Fast and accurate system level modeling is required

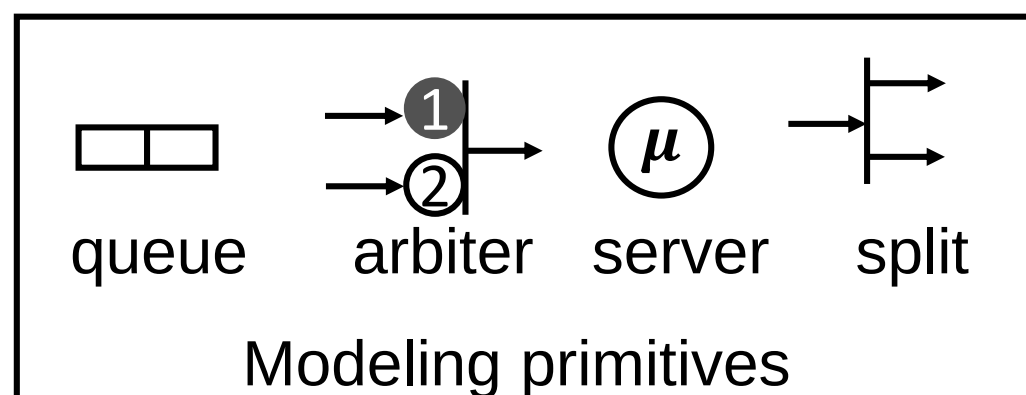


Priority Aware NoC

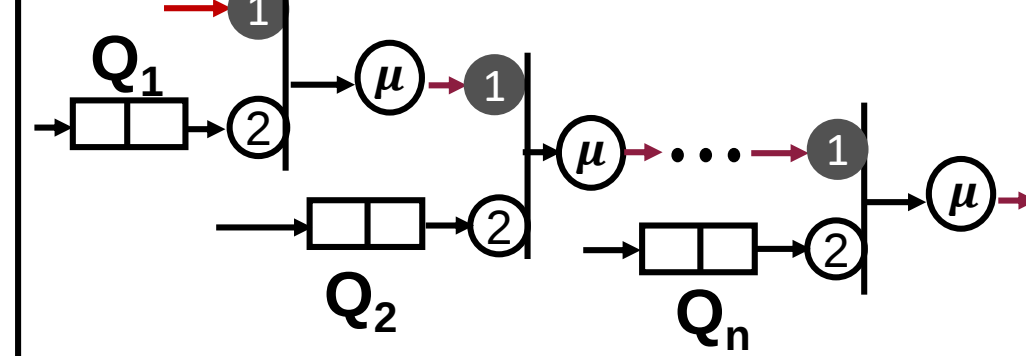
Physical Network



Notation

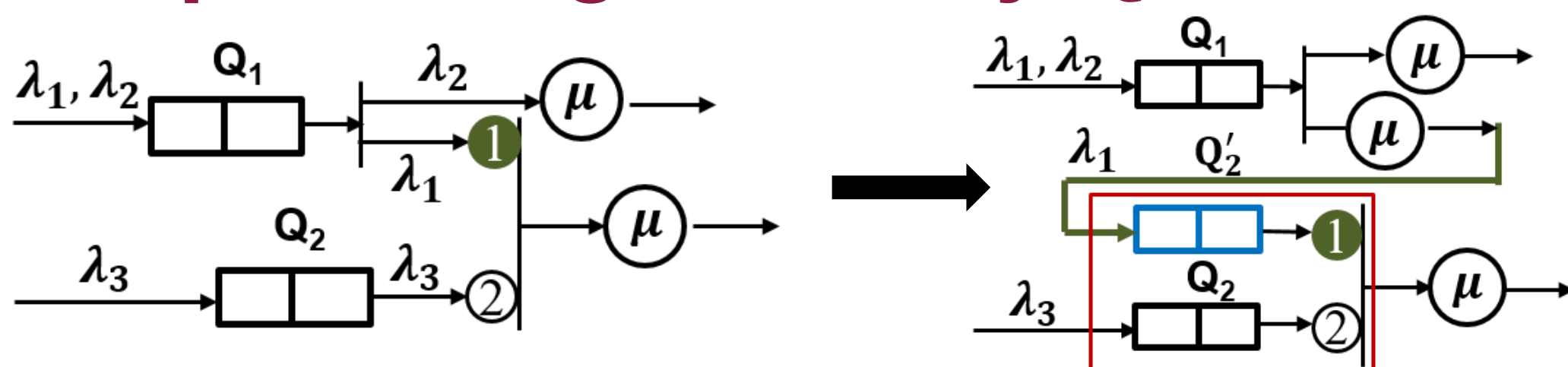


Abstract Model



- Mux in routers modeled using priority arbiters and servers
- Inputs with filled color have higher priority

Split at High Priority Queue



- Waiting time of class 1

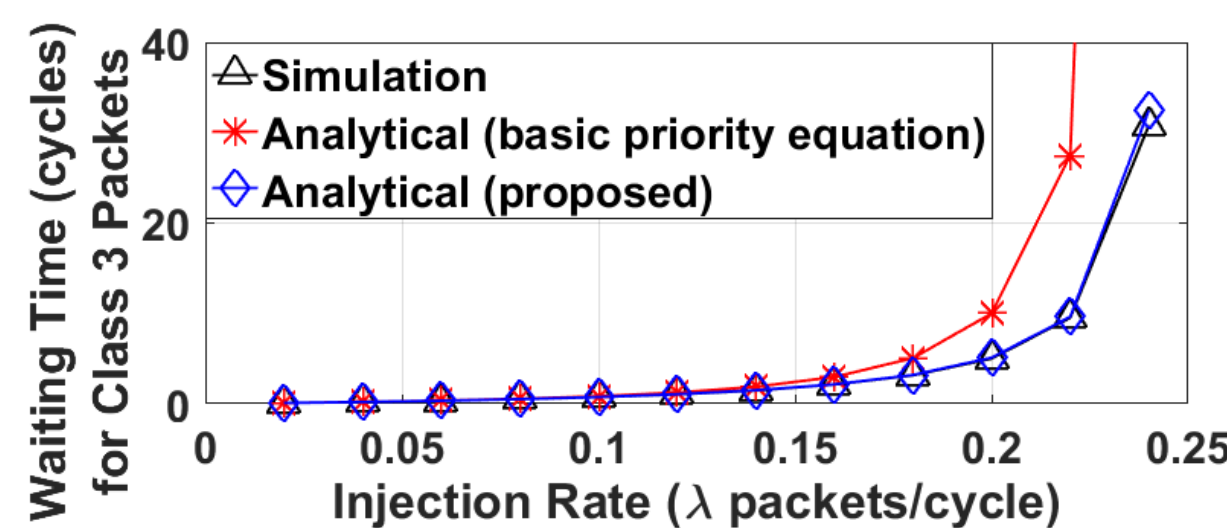
$$W_1^{Q'_2} = \frac{R_1^{Q'_2}}{1 - \rho_1}$$

- Residual time of class 3

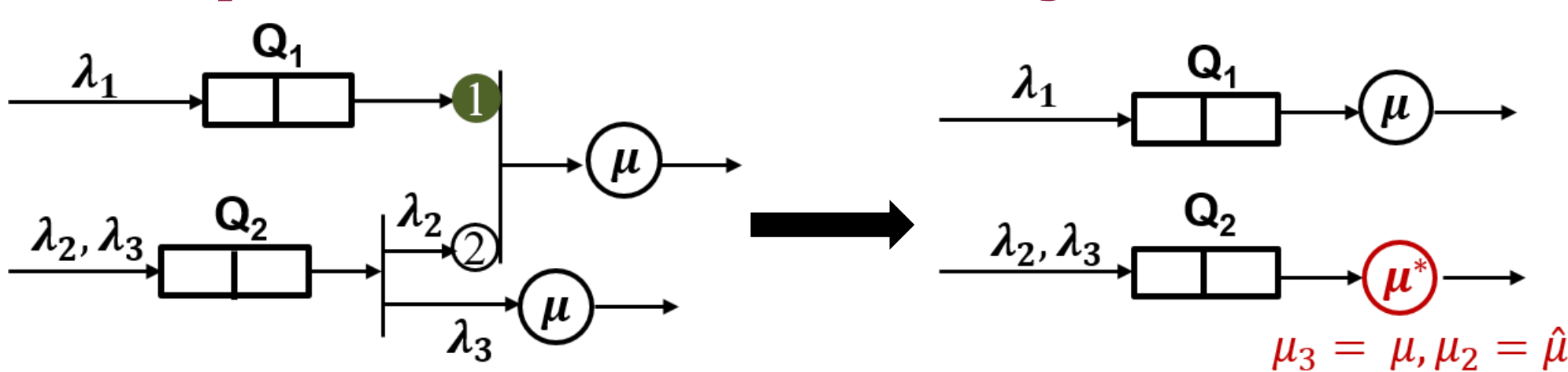
$$R_3 = R_1^{Q'_2} + \rho_1$$

- Waiting time of class 3

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



Split at Low Priority Queue

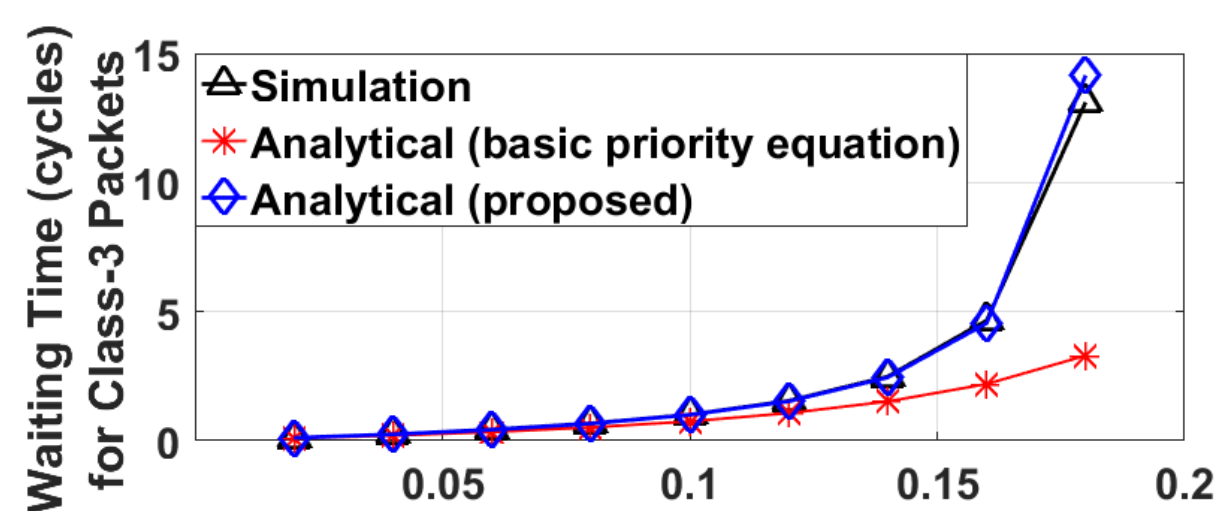


- Service time of class 2 is modified (ΔT)
- Residual time of class 2

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

- Waiting time of class 2

$$W_2 = \frac{\hat{R}_2 + R_3}{1 - \rho_2 - \rho_3} + \Delta T$$



Model Generation Flow

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

1. Structural Transformation

Get all classes having higher priority and calculate co-efficient of variation of inter-arrival time (C_A)

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

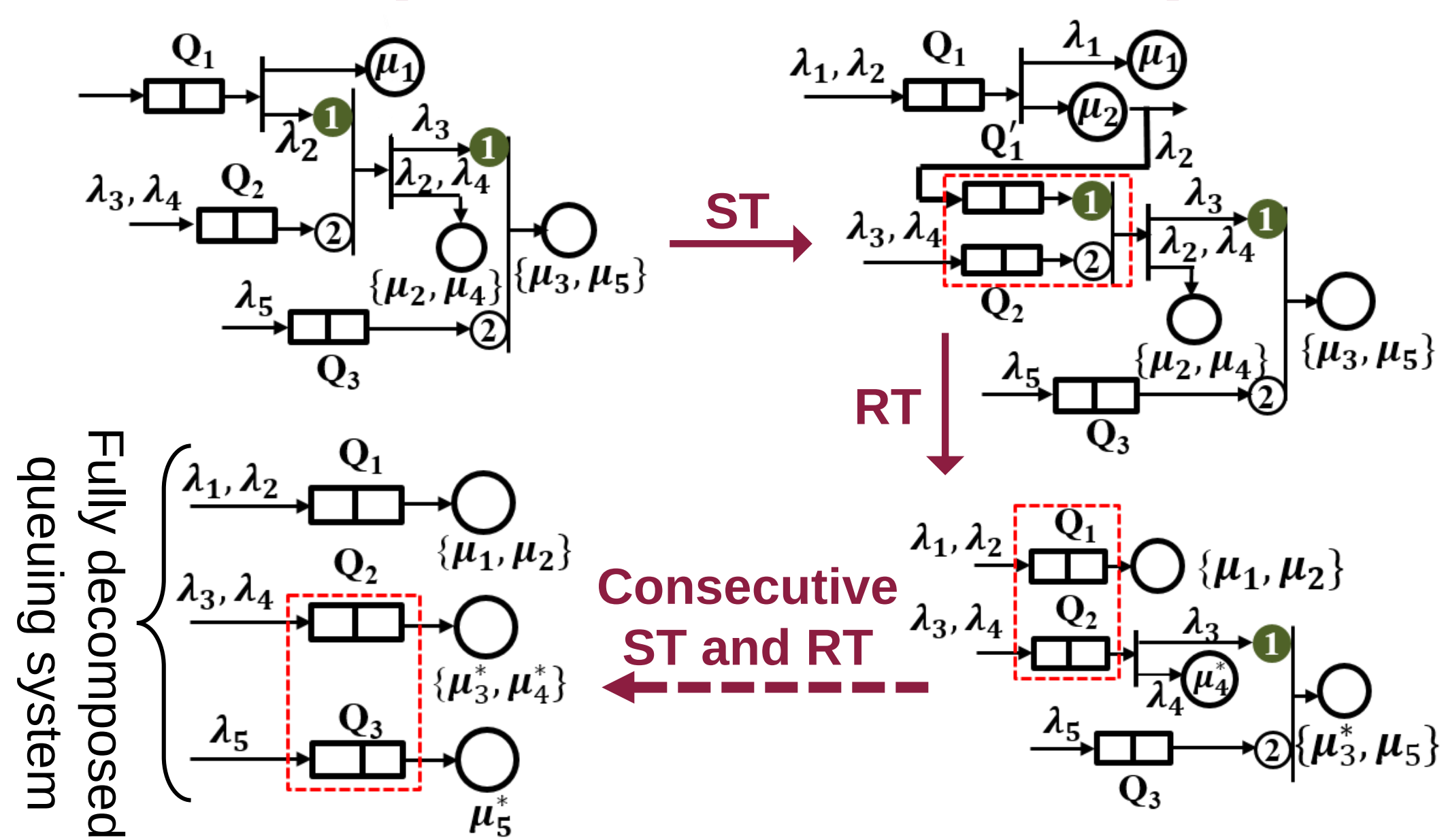
2. Service Rate Transformation

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 $\hat{R}$

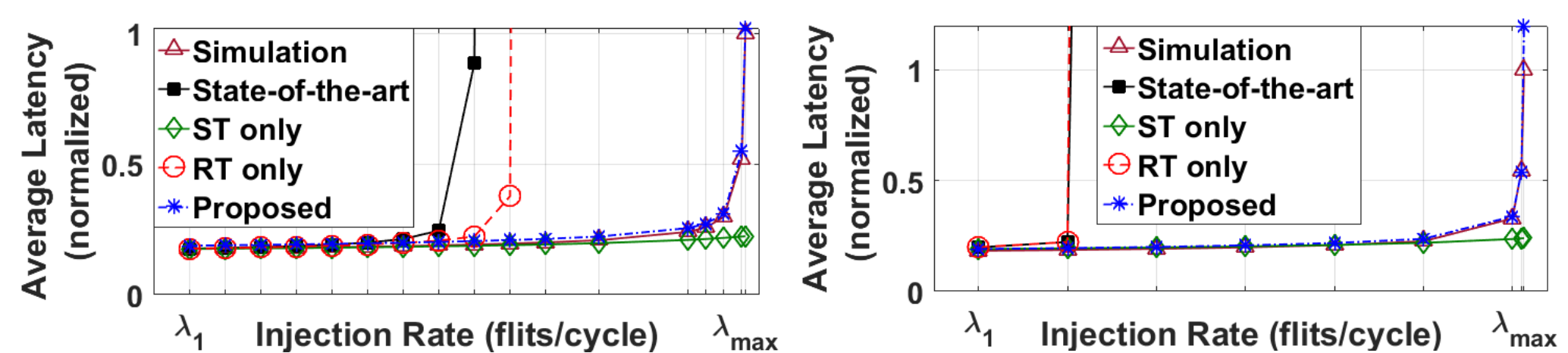
A Representative Example



ST → Structural Transformation, RT → Service Rate Transformation

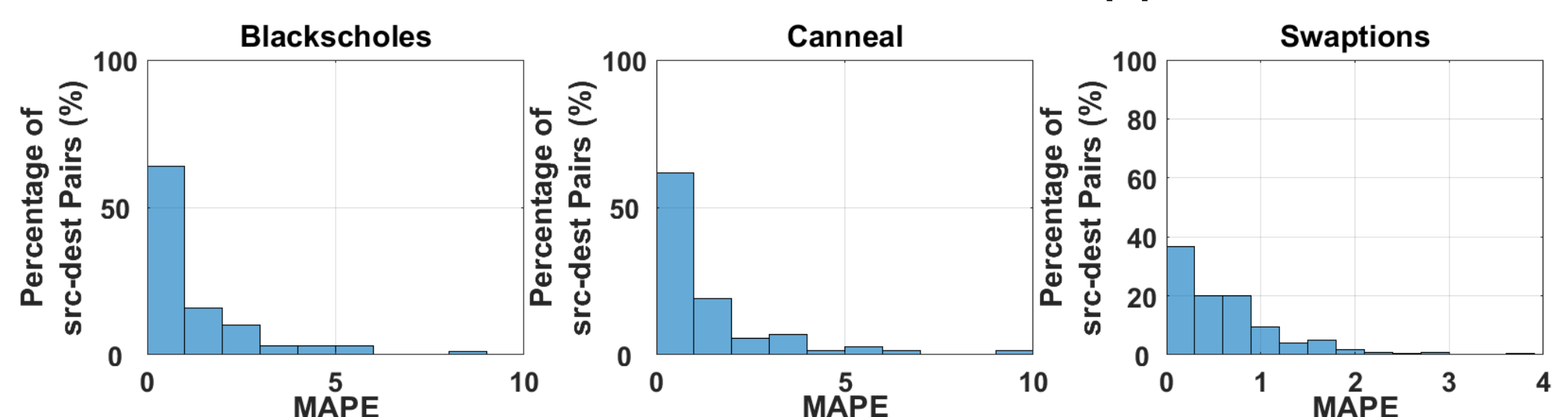
Evaluation with Synthetic Traffic

- Achieve less than 4% error compared to simulation
- Proposed models are 2-3 orders of magnitude faster than simulation models



Evaluation with Real Applications

- Collected real app traces from gem5-Full System
- Achieve less than 10% error for all applications



Conclusion

- Analytical models for priority-aware NoC are obtained
 - Transforming a given network iteratively
- Achieve >90% accuracy for synthetic and real apps