A Comprehensive and Accurate Latency Model for Network-on-Chip Performance Analysis

Zhiliang Qian[1], Da-Cheng Juan[2], Paul Bogdan[3], Chi-Ying Tsui[1], Diana Marculescu[2] and Radu Marculescu[2]

[1]The Hong Kong University of Science and Technology, Hong Kong
[2]Carnegie Mellon University, Pittsburgh, U.S.A
[3]University of Southern California, Los Angeles, U.S.A

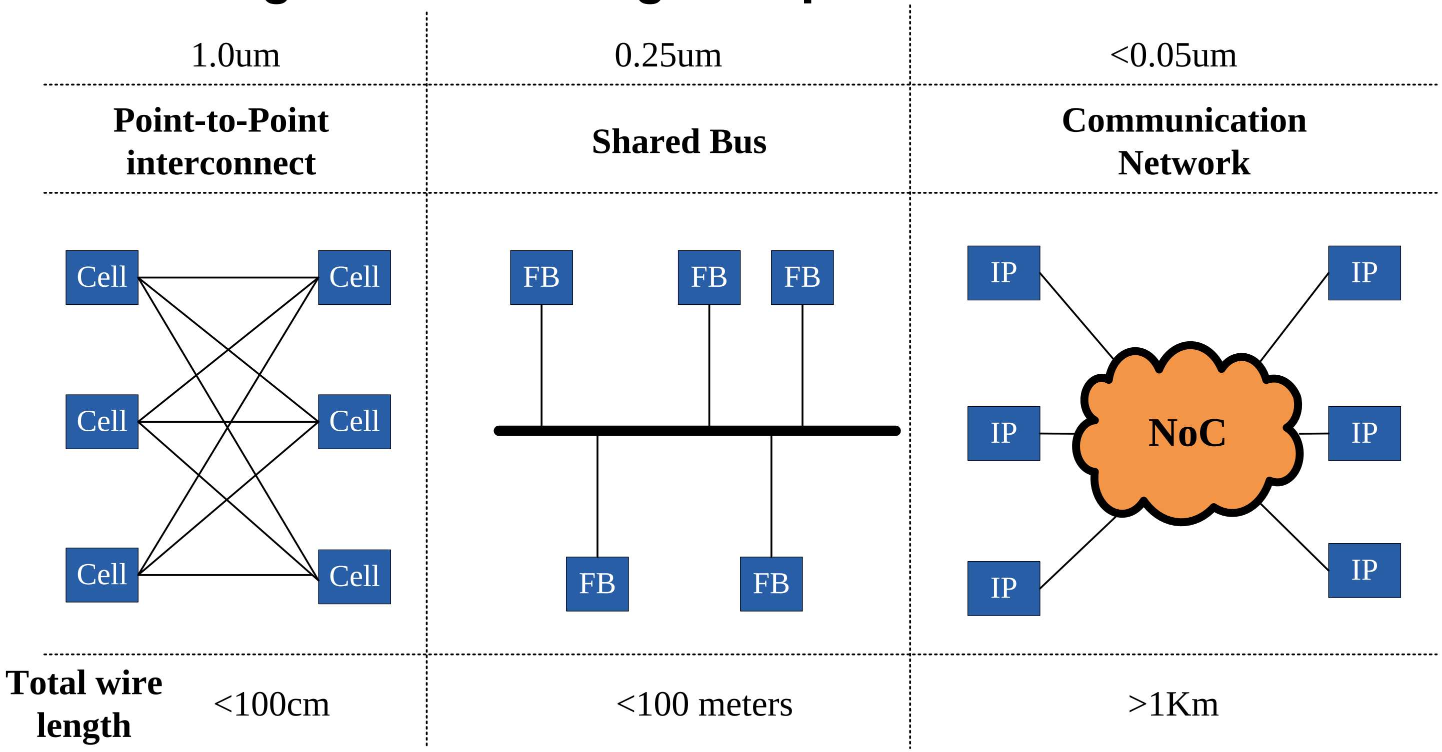
IEEE/ACM ASP-DAC'14, 22 Jan., 2014, Singapore

Outline

- Introduction
- NoC Modeling for Performance Analysis
  - NoC end-to-end delay calculation
  - Link dependency analysis
  - GE-type traffic modeling
  - Wormhole router based NoC latency model
- Experimental results
  - Simulation setup
  - Evaluation under synthetic traffic patterns
  - Evaluation under realistic benchmarks
- Conclusion

Network-on-Chips (NoCs)

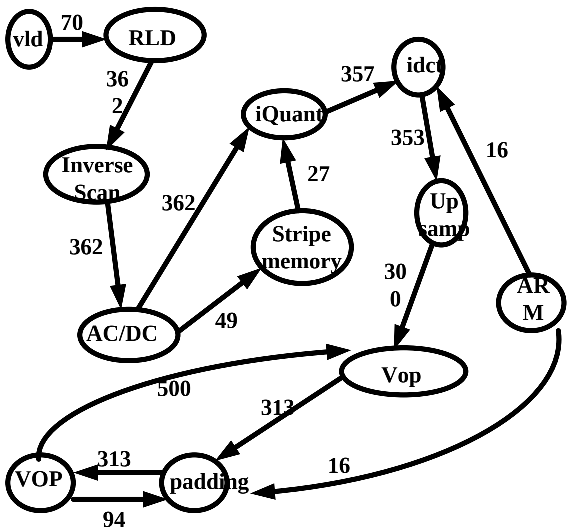
- With technology scaling down, more and more components can be integrated on a single chip.

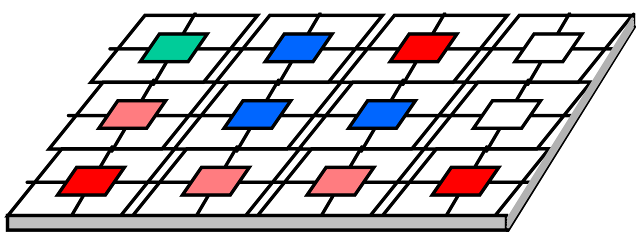


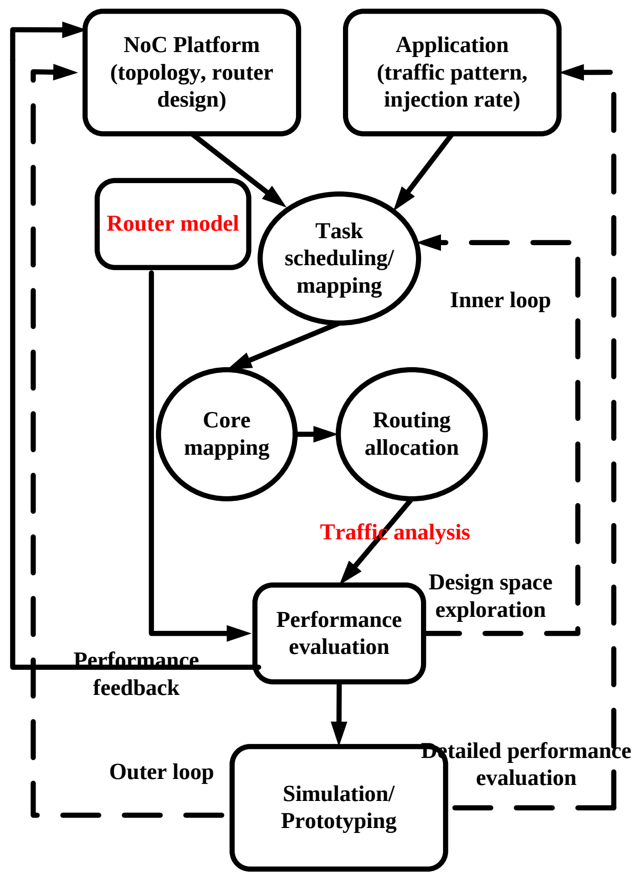
  - ➢ An efficient way to manage the communication of on-chip resources plays the key role in future system design.

NoC design space exploration

- **A large design space needs to be explored for an optimal design**
  - Task mapping, allocation, buffer sizing, routing algorithm etc.
  - Accurate and fast performance evaluation is required during the exploration -> analytical performance evaluation model

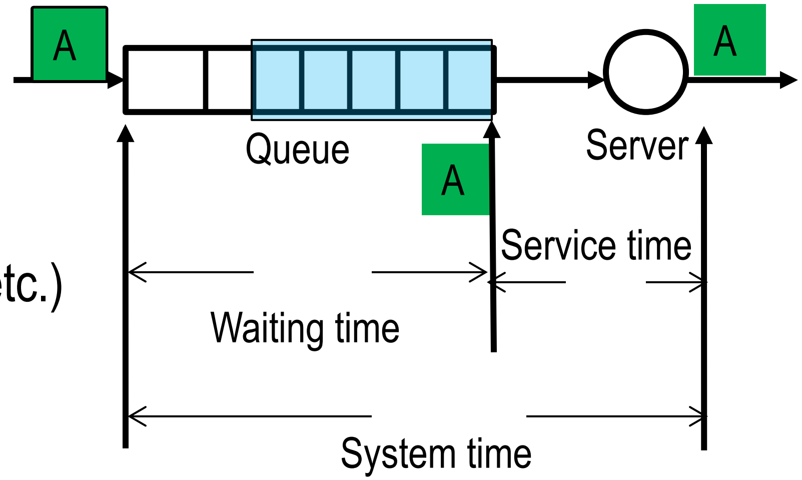


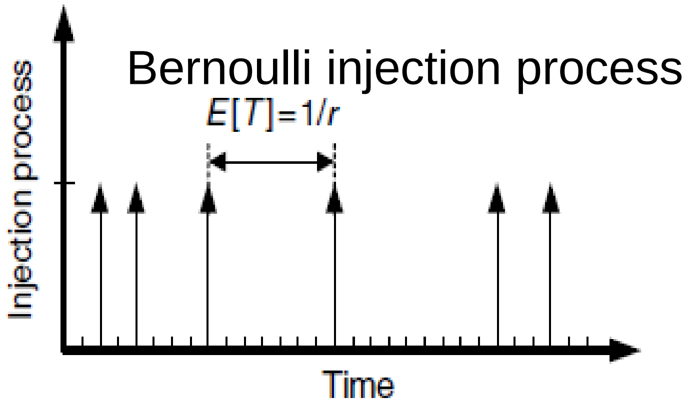


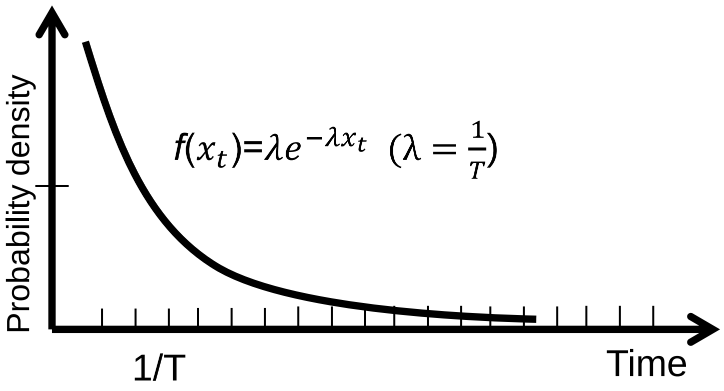


Introduction- queuing-theory-based analytical model

- **Queuing-theory-based delay estimation**
  - Customer (packet) arrival process
  - System (server) service process
  - Number of servers
  - Service discipline (FCFS, Round-robin etc.)
  - System time and waiting time







Queuing-theory-based NoC latency model

- **Previous arts and motivation of this work**

NoC latency model	Previous NoC analytical models				This work
	[VLSI 2007]	[TCAD'12, ICCAD'09]	[TVLSI'13]	[NoCs'11]	
Traffic model for the application					
Queue	M/M/1	M/G/1/K	G/G/1	M/M/m/K	G/G/1/K
Arrival	Poisson	Poisson	General	Poisson	General
Service	Markov	General	General	Markov	General
NoC architecture modeled					
Buffer	Small	K packets	B flits	Small	B flits
PB ratio[1]	m ($\gg 1$)	< 1	arbitrary	m ($\gg 1$)	arbitrary
Arbitration	Round robin	Round robin	Fixed priority	Round robin	Round robin

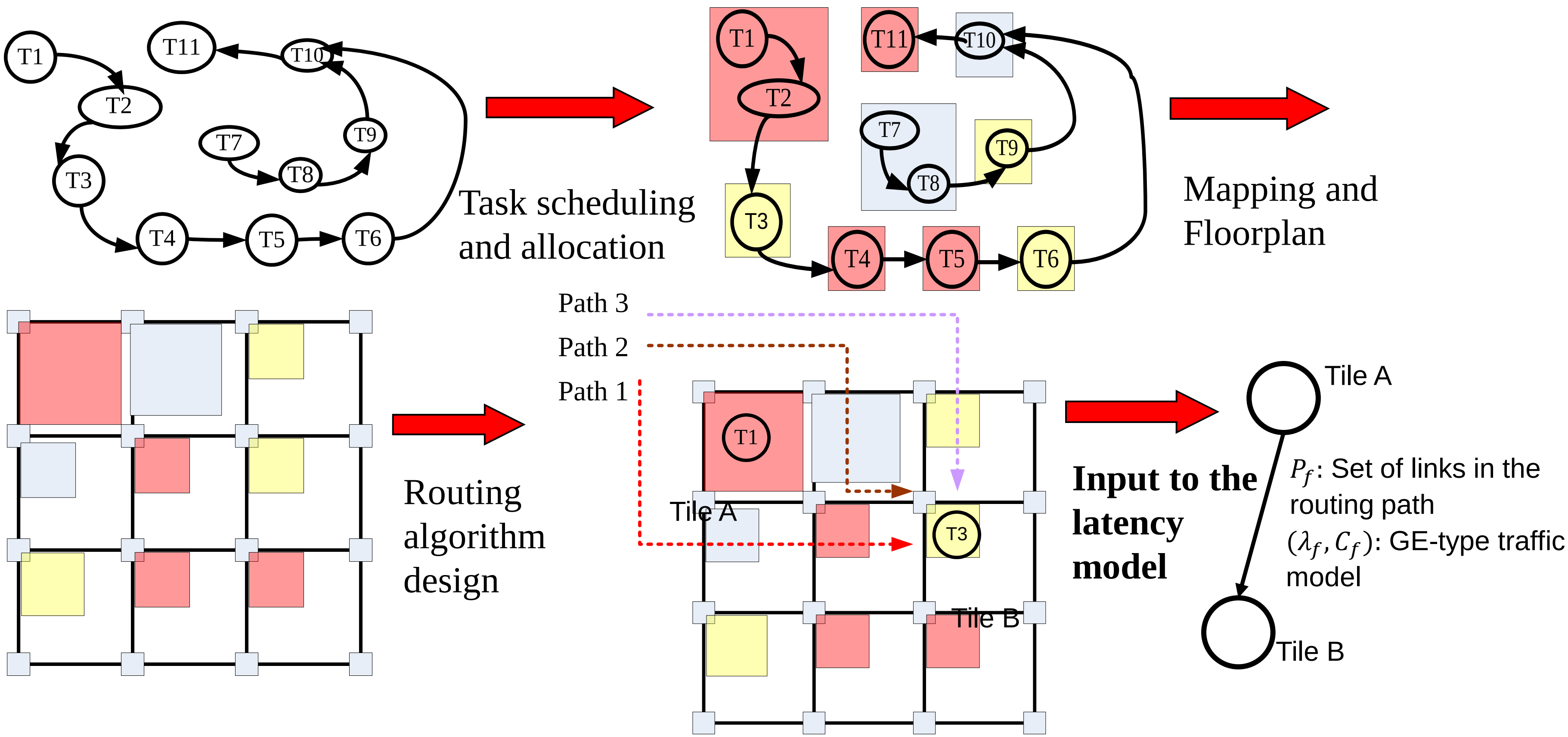
[1] PB ratio is defined as the ratio of average packet size (m flits) to the buffer depth (B flits)

Outline

- Introduction
- NoC Modeling for Performance Analysis
  - NoC end-to-end delay calculation
  - Link dependency analysis
  - GE-type traffic modeling
  - Wormhole router based NoC latency model
- Experimental results
  - Simulation setup
  - Evaluation under synthetic traffic patterns
  - Evaluation under realistic benchmarks
- Conclusion

Input to the NoC latency model

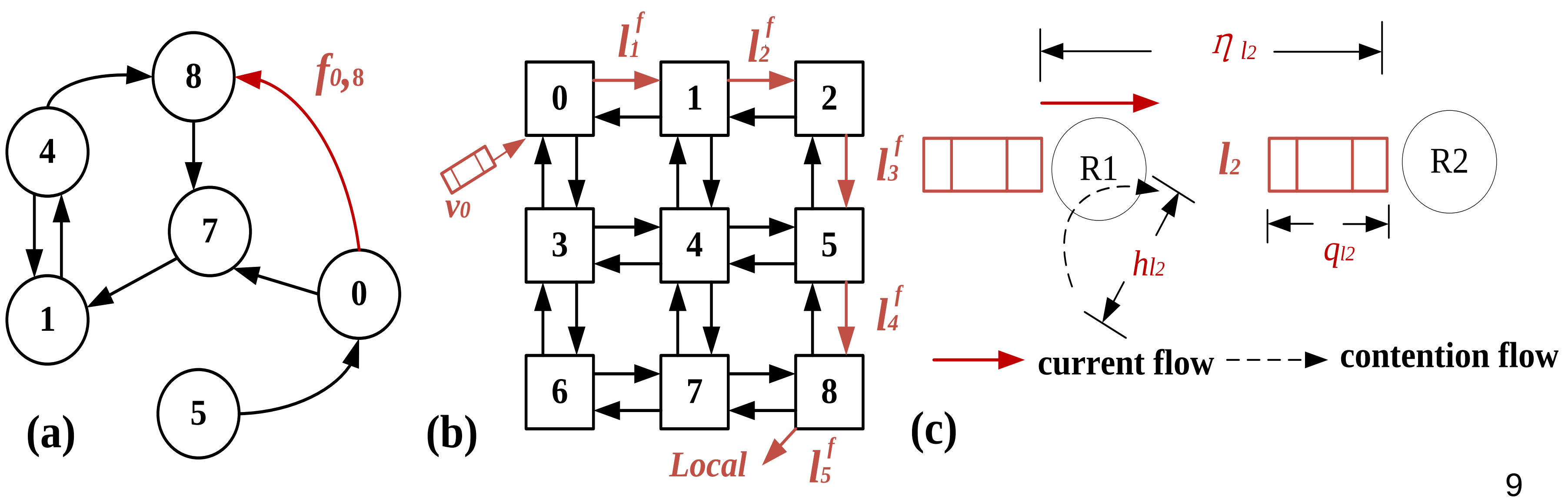
- The application has been scheduled and mapped onto the NoC.
- A deterministic routing algorithm is used to avoid deadlock.



Performance metrics: average latency $L = (\sum_{f \in F} \lambda_f \times L_f) / \sum_{f \in F} \lambda_f$

NoC end-to-end delay calculation

- **The end-to-end flow latency $L_{s,d}$ of a specific flow $f_{s,d}$ consists of three parts:** $L_{s,d} = v_s + \eta_{s,d} + h_{s,d}$
  - The queuing time at the source v_s
  - The packet transfer time in the path $\eta_{s,d} = (m+1) + \sum_{i=1}^{d_f} \eta_{l^f{}_i}$
  - The path acquisition time $h_{s,d} = \sum_{i=1}^{d_f} h_{l^f{}_i}$

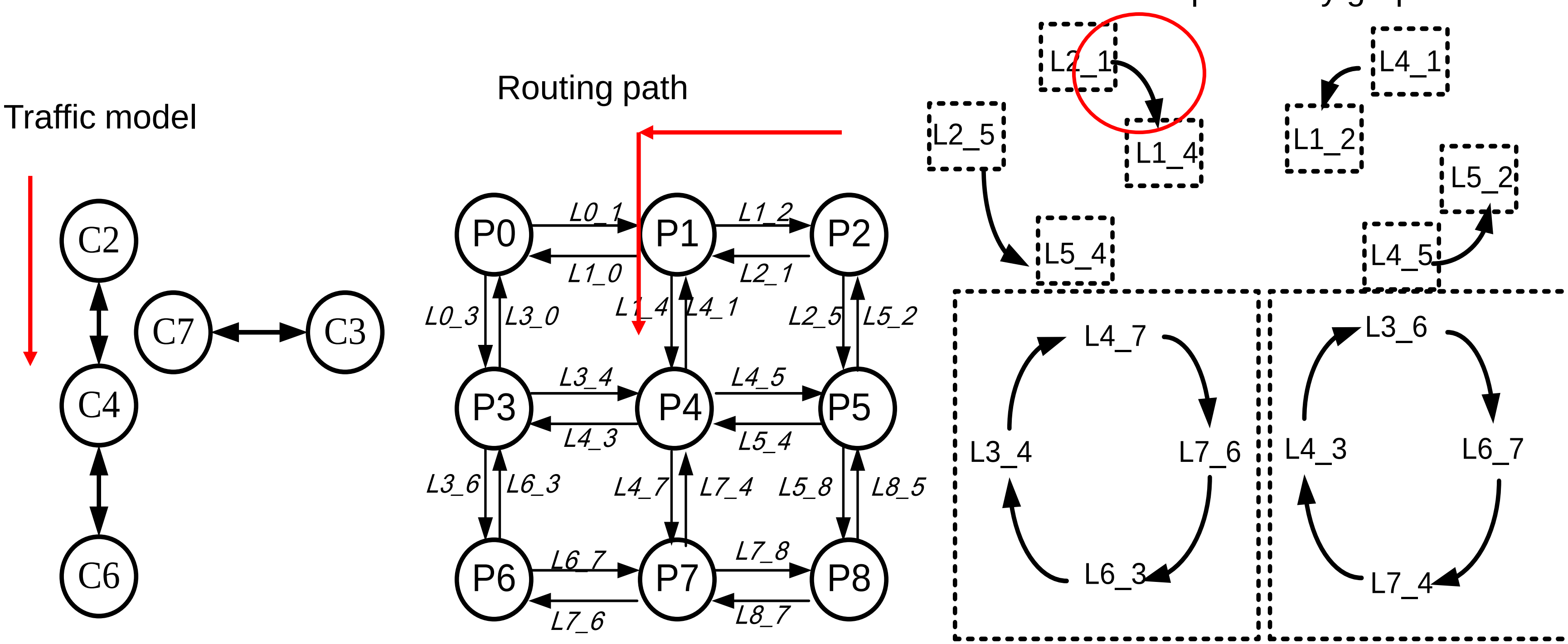


Outline

- Introduction
- NoC Modeling for Performance Analysis
  - NoC end-to-end delay calculation
  - Link dependency analysis
  - GE-type traffic modeling
  - Wormhole router based NoC latency model
- Experimental results
  - Simulation setup
  - Evaluation under synthetic traffic patterns
  - Evaluation under realistic benchmarks
- Conclusion

Link dependency graph

- **Building the channel (link) dependency graph**



Core Communication graph

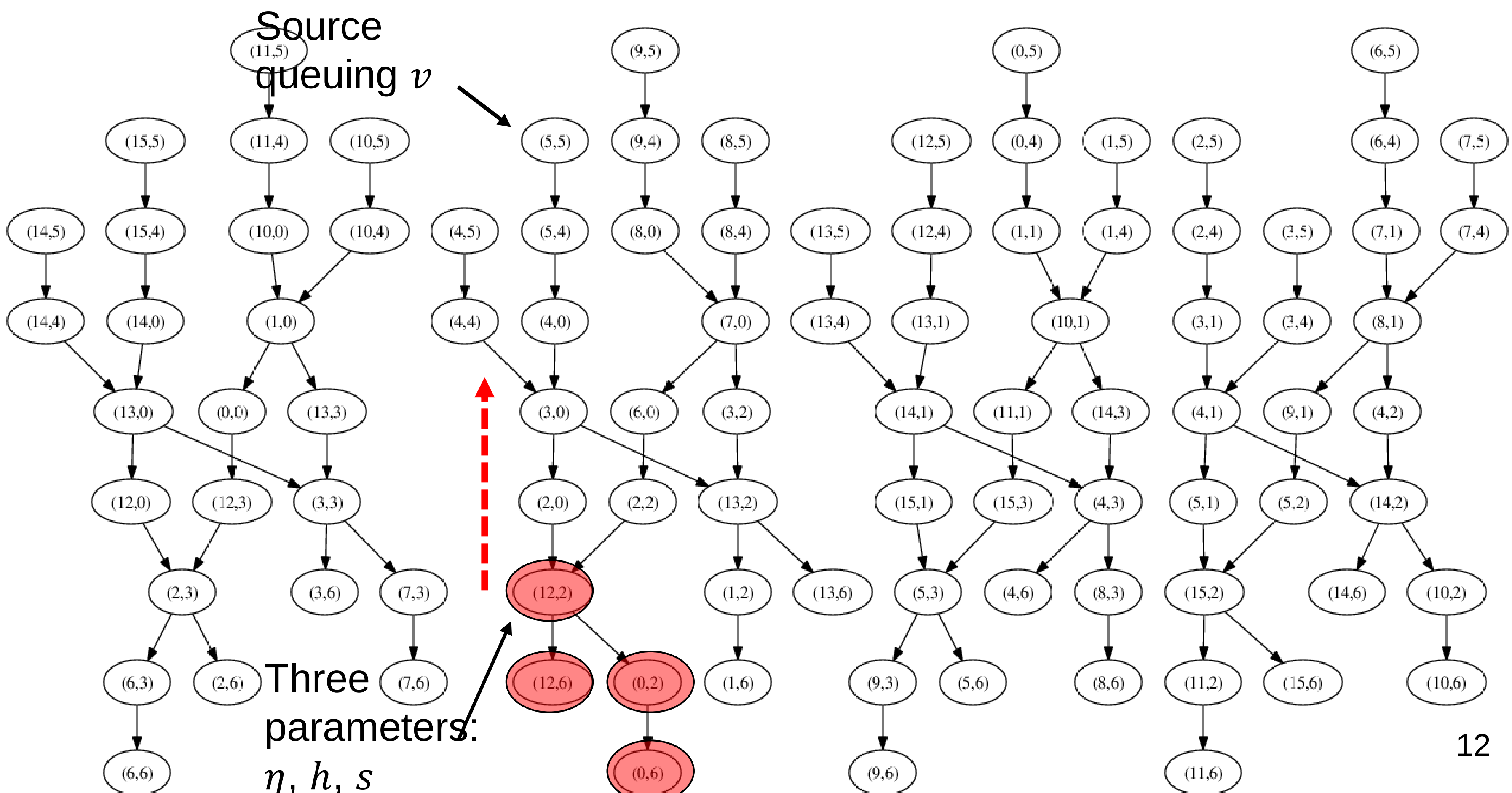
NOC mesh

Channel Dependency Graph (CDG) of the application communication

Link dependency analysis

- **Topological sort algorithm is applied on the obtained CDG to find out the proper order to analyze the queuing delays.**

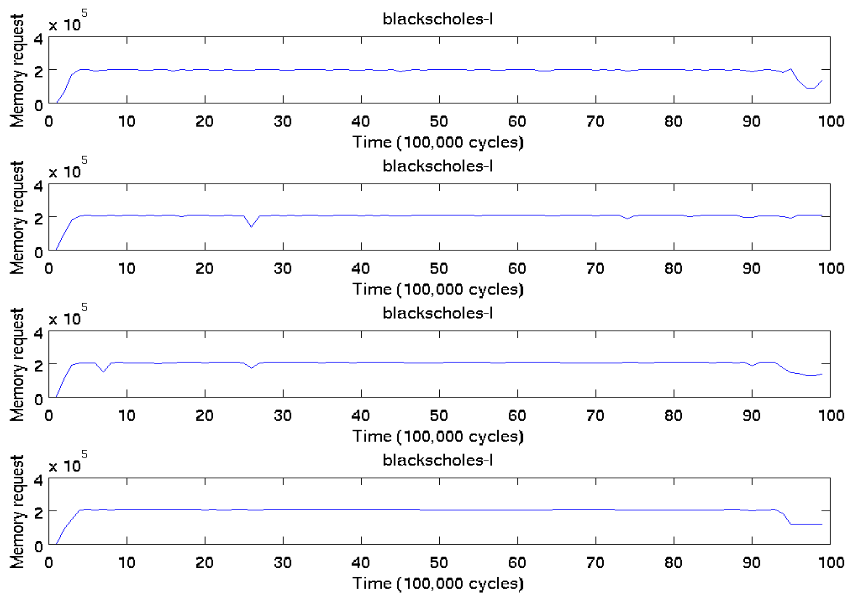
A sample channel dependency graph

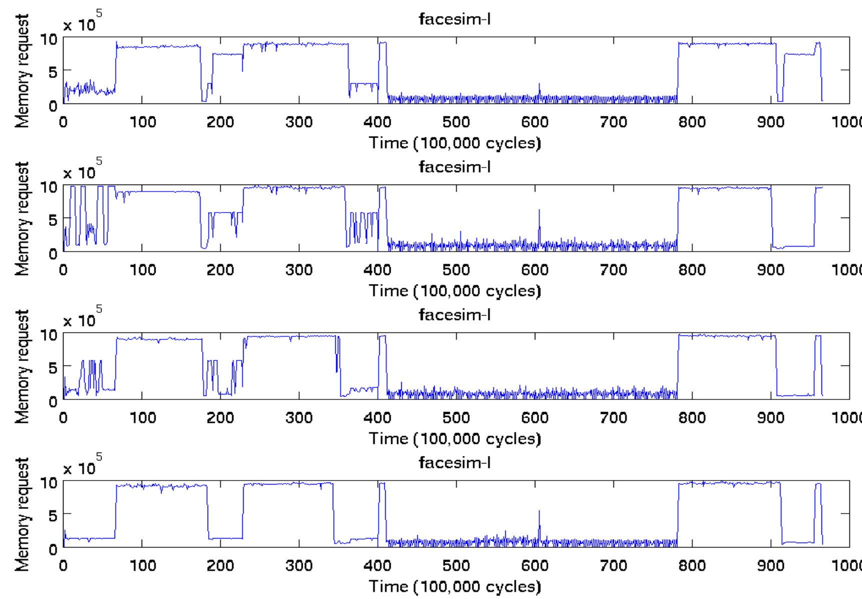


Outline

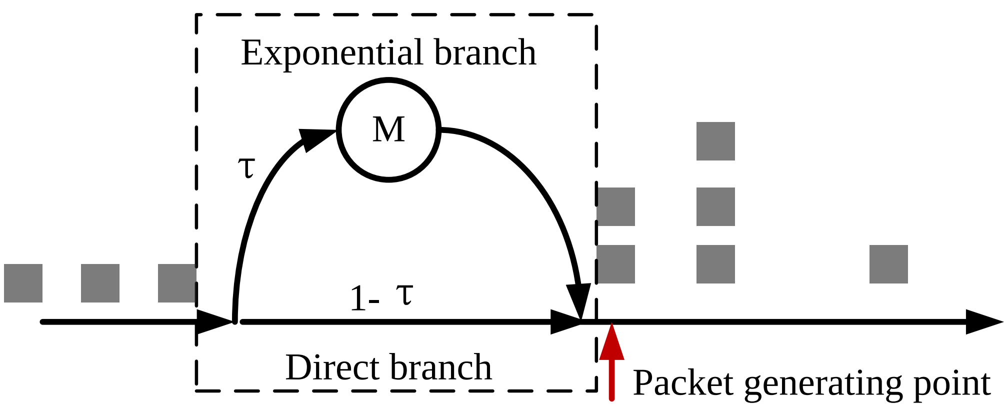
- Introduction
- **NoC Modeling for Performance Analysis**
  - NoC end-to-end delay calculation
  - Link dependency analysis
  - **GE-type traffic modeling**
  - Wormhole router based NoC latency model
- Experimental results
  - Simulation setup
  - Evaluation under synthetic traffic patterns
  - Evaluation under realistic benchmarks
- Conclusion

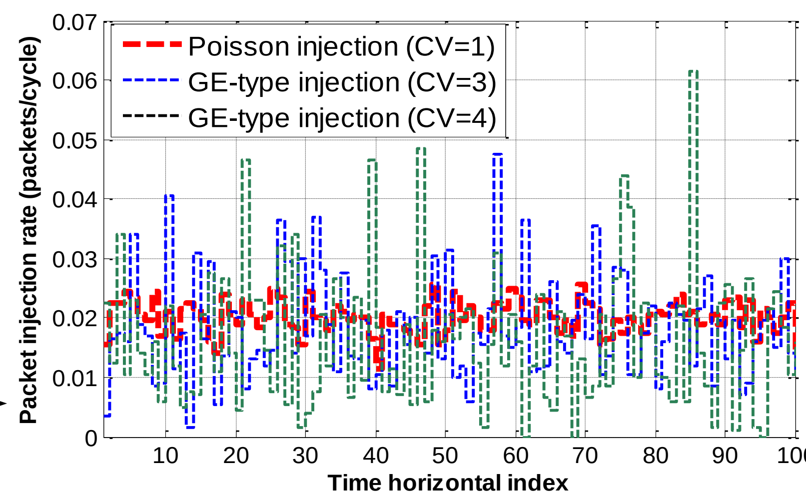
Modeling the bursty traffic input





- Generalized exponential (GE) distribution





GE-type traffic modeling

- **The GE-type cumulative distribution function (cdf) of inter-arrival time is:**

$$F(t) = P(X \le t) = 1 - \tau e^{-\tau\lambda t}, \qquad t \ge 0$$

Where the parameter $\tau = \frac{2}{1+C^2}$ and C^2 is the square coefficient of variation

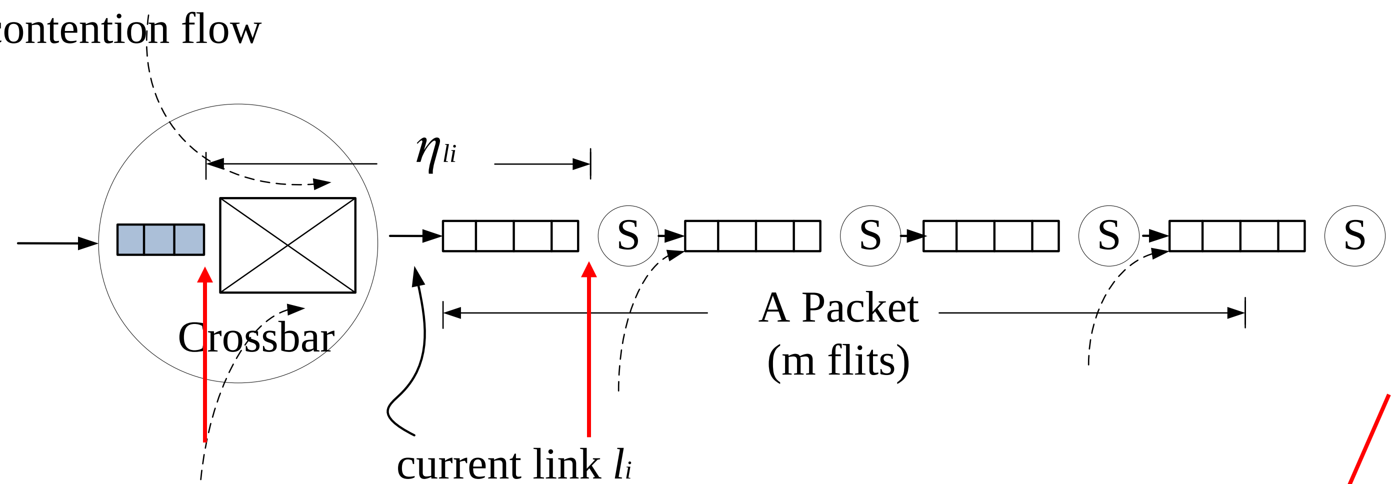
- **In this work, we use the GE distribution to model the traffic input of each flow, which is characterized by two parameters:**
  - λ : the average packet arrival rate (packets/cycle)
  - C: the coefficient of variation of this traffic flow, i.e., $C = \frac{\sigma}{\lambda}$, where σ is the standard derivation of the packet inter-arrival times.
- **Accordingly, the GE/G/1/K queuing model is used to analyze the channel waiting time by considering the traffic burstness.**

Outline

- Introduction
- NoC Modeling for Performance Analysis
  - NoC end-to-end delay calculation
  - Link dependency analysis
  - GE-type traffic modeling
  - Wormhole router based NoC latency model
- Experimental results
  - Simulation setup
  - Evaluation under synthetic traffic patterns
  - Evaluation under realistic benchmarks
- Conclusion

Flit transfer time η calculation

- **The flit transfer time η of link l_{ab} is defined as the time taken for the header flit after being granted link access to reach the buffer front in link l_{ab}**



Mean flit rate arriving at the buffer is: $\lambda_{l_{ab}} = m \times \lambda^{packet}{}_{l_{ab}} = m \times \sum_{f \in F_{l_{ab}}} \lambda_f$

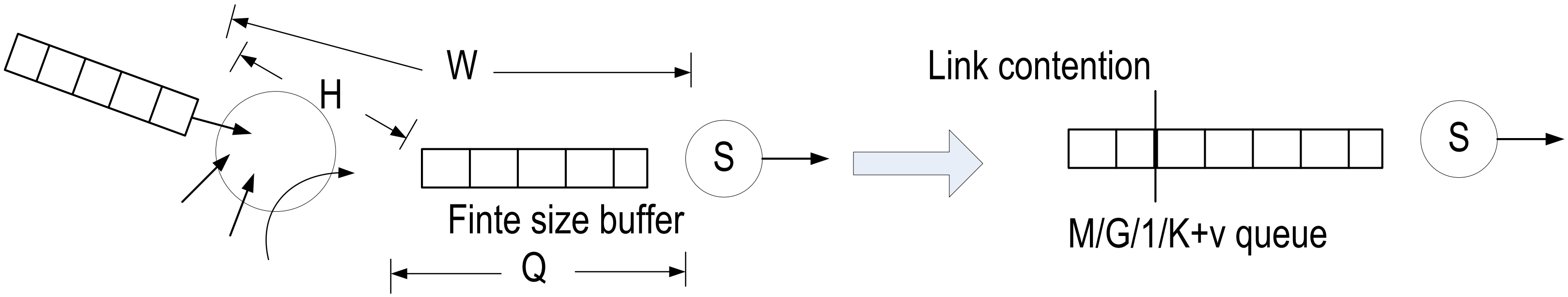
Mean time to serve a flit in this queuing system is the weighted average of the service time from all flows passing through l_{ab}: $s_{flit}^{l_{ab}} = \frac{\sum_{\forall f \in F_{l_{ab}}} \left[\lambda_f \times \left(\frac{n_{l_{i+1}}^f}{m} + 1 \right) \right]}{\sum_{\forall f \in F_{l_{ab}}} \lambda_f}$

Mean service time for flow f

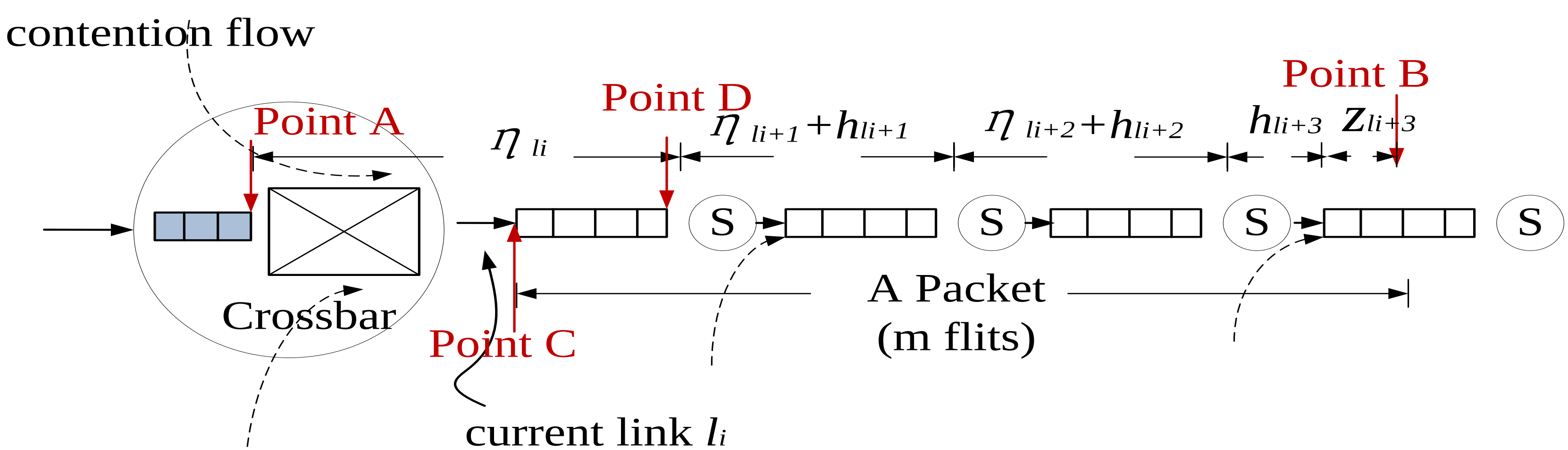
Illustration of path acquisition time

- **Service time in wormhole NoC to obtain *h*:**

Waiting time W includes: 1) the link contention time H and 2) the time for the header flit to reach the buffer head Q



Service time S is bounded by the time where the header reaches the node that the accumulated buffer spaces between can hold the whole worm packet.



Path acquisition time h calculation

- **Number of effective subsequent links of link l with respect to the path p:**

$$\Lambda^p(l) = \begin{cases} \left\lfloor \frac{m}{B} \right\rfloor & if\ r(p,l) > \left\lfloor \frac{m}{B} \right\rfloor \\ r(p,l) & elsewise \end{cases}$$

where $r(p,l)$ is the function returns the number of remaining hops from link l towards the destination of path p.

- **The service time of link l with respect to the path p is [1]:**

$$s_{l_i^f} = \begin{cases} \left[m\left(m + x_{l_i}^f\right) + 2x_{l_i}^f m \right] / (m + 2x_{l_i}^f) & if\ x_{l_i}^f < m \\ \left[m\left(m + x_{l_i}^f\right) + 2\left(x_{l_i}^f\right)^2 \right] / (m + 2x_{l_i}^f) & otherwise \end{cases}$$

- **The channel service time of link l :**

$$\bar{s}_{l_{ab}} = \sum\nolimits_{\forall f \in F_{l_{ab}}} (\lambda_f \times s_{l_i^f}) / \sum\nolimits_{\forall f \in F_{l_{ab}}} \lambda_f$$

$$C_{s_{l_{ab}}}^2 = \frac{\overline{s_{l_{ab}}^2}}{\left(\bar{s}_{l_{ab}}\right)^2} - 1 = \left(\frac{\sum_{\forall f \in F_{l_{ab}}} \lambda_f \times s_{l_i^f}^2}{\sum_{\forall f \in F_{l_{ab}}} \lambda_f}\right) / \left(\bar{s}_{l_{ab}}\right)^2 - 1$$

[1] P.-C. Hu, L. Kleinrock, An Analytical model for wormhole routing with finite size input Buffers, 15th International Telegraphic Congress, 1998

GE/G/1/K queue based h calculation

- **Diffusion approximation for the steady state distribution probability P_n of the M/G/1/K queue with arrival rate λ and service rate μ :**

$$P_n = \begin{cases} c \times p'_n \quad (0 \leq n \leq K) \\ 1 - \frac{1-c\left(1-\frac{\lambda}{\mu}\right)}{\frac{\lambda}{\mu}} (n = K+1) \end{cases}$$

Where the normalization constant $c = (1 - \frac{\lambda}{\mu}(1 - \sum_{j=0}^{K} P_n))^{-1}$ and p'_n is the steady state probability of M/G/1/∞ queue [2]

- **Applying Little's formula to obtain the waiting time :** $h_l{}' = (\sum_{i=1}^{K+1} i \times P_i) / \lambda$
- **Taking the arrival traffic burstiness in GE/G/1/K model by refining the results of M/G/1/K queue:**

$$h_{l_{ab}} = \frac{({C_s}_{l_{ab}}^2 + {C_a}_{l_{ab}}^2)}{(1 + {C_s}_{l_{ab}}^2)} h'_{l_{ab}}$$

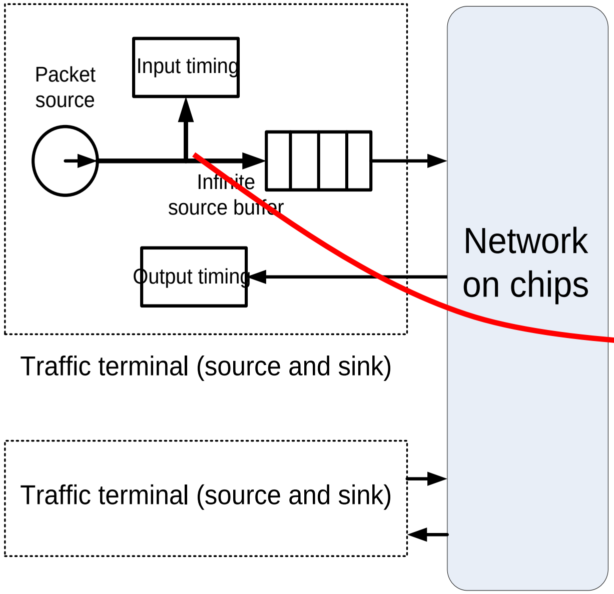
[2] M.C. Lai, et.al. An accurate and efficient performance analysis approach based on queuing model for Network on Chip. In *Proceedings of* ICCAD,2009

Source queuing time v_s

- **The source queue is modeled as a GE/G/1/∞ system:**

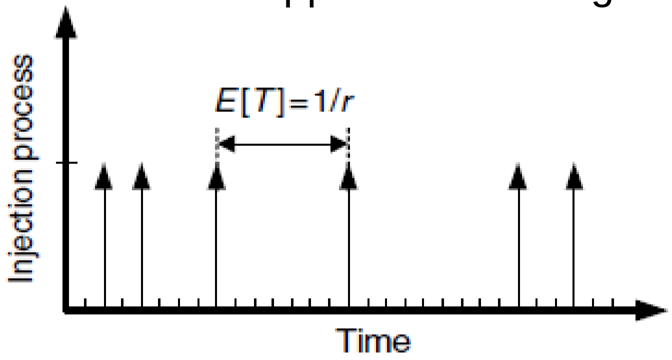
$$v_s = \frac{\overline{s}_{l_s}}{2}\left(1 + \frac{C_a^2 + \lambda_a \times \frac{\left(\overline{s}_{l_s} - m\right)^2}{\overline{s}_{l_s}}}{1 - \lambda_a \times \overline{s}_{l_s}}\right) - \overline{s}_{l_s}$$

where the arrival process is characterized by (λ_a, C_a^2) in the GE type traffic model and the service time at source is represented as $\overline{s}_{l_s}$.



Open loop source queuing measurement setup

Application traffic generation trace



Proposed NoC latency analysis flow

- **Link dependency analysis to obtain the link order G**
- **For each link l_{ab} in G:**
  - Calculate the flit transfer time η
  - Calculate the link service time s
  - Compute the path acquisition time h
- **Calculate the source queuing time v**
- **Form the latency for each flow in application**

foreach $l_{ab} \in G$
    $(\lambda_{l_{ab}}, C_{a_{l_{ab}}}^2)$ = *traffic_model* ($F_{l_{ab}}$)
    $\eta_{l_{ab}}$ = *calculate_transfer_time*($\lambda_{l_{ab}}, s_{flit}^{l_{ab}}, m, B$)
      foreach $f \in F_{l_{ab}}$ and $l_i^f = l_{ab}$
        $s_{l_i}^f$ = *calculate_link_service_time* ( )
      end
    $(\bar{s}_{l_{ab}}, C_{s_{l_{ab}}}^2)$ = *service_time* ( )
  if $a \neq b$ // the links between the routers
    $h_{l_{ab}}$ = *GE_G_1_K_queue* ($\lambda_{l_{ab}}, C_{a_{l_{ab}}}^2, \bar{s}_{l_{ab}}, C_{l_{ab}}^2, k$)
  else   // the link is the source link
    v_a = *GE_G_1 _queue* ($\lambda_{l_{ab}}, C_{a_{l_{ab}}}^2, \bar{s}_{l_{ab}}, C_{s_{l_{ab}}}^2$)
  endif
endfor
foreach $f \in F$
    $L_{s,d}$=*calculate_flow_latency*( )
end

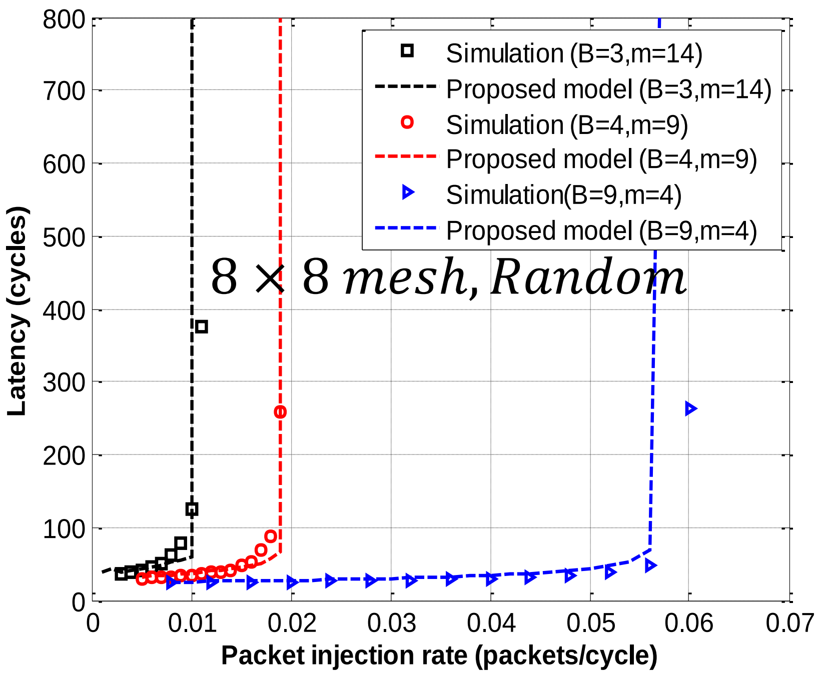
Outline

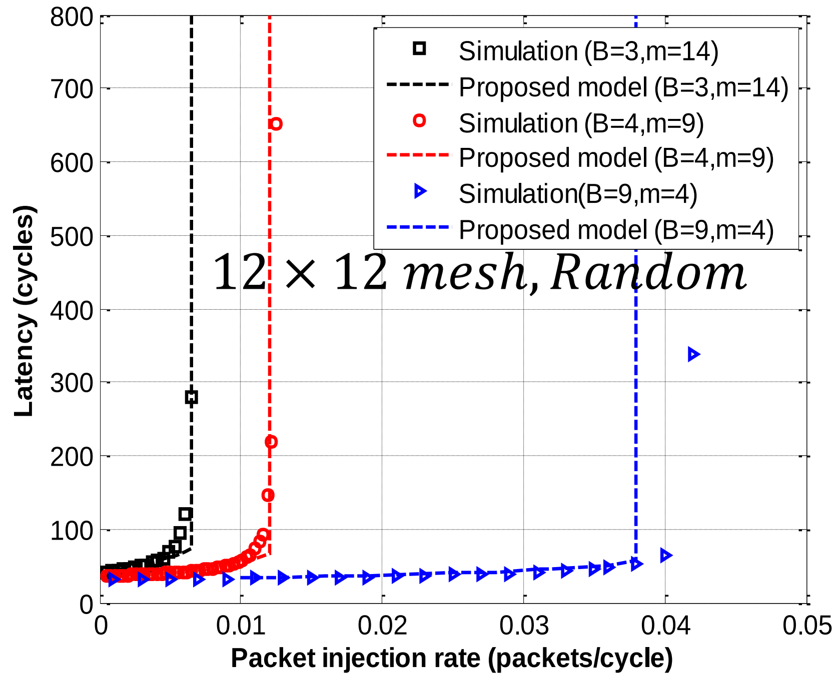
- Introduction
- NoC Modeling for Performance Analysis
  - NoC end-to-end delay calculation
  - Link dependency analysis
  - GE-type traffic modeling
  - Wormhole router based NoC latency model
- **Experimental results**
  - **Simulation setup**
  - Evaluation under synthetic traffic patterns
  - Evaluation under realistic benchmarks
- Conclusion

Simulation setup

- **The proposed analytical latency model is implemented in MATLAB and its accuracy is compared with Booksim simulator.**
- **Each router takes two cycles to route a flit and the link traversal stage takes an additional one cycle.**
- **Different buffer depth (B *flits*) and packet length (m *flits*) combinations are evaluated.**
- **Both synthetic and real applications are adopted:**
  - Random and shuffle traffic on 8×8 and 12×12 meshes
  - MMS (Multimedia system)
  - DVOPD (Video object plane decoder)
  - MPEG4 (MPEG decoder)
  - SPECweb99 applications

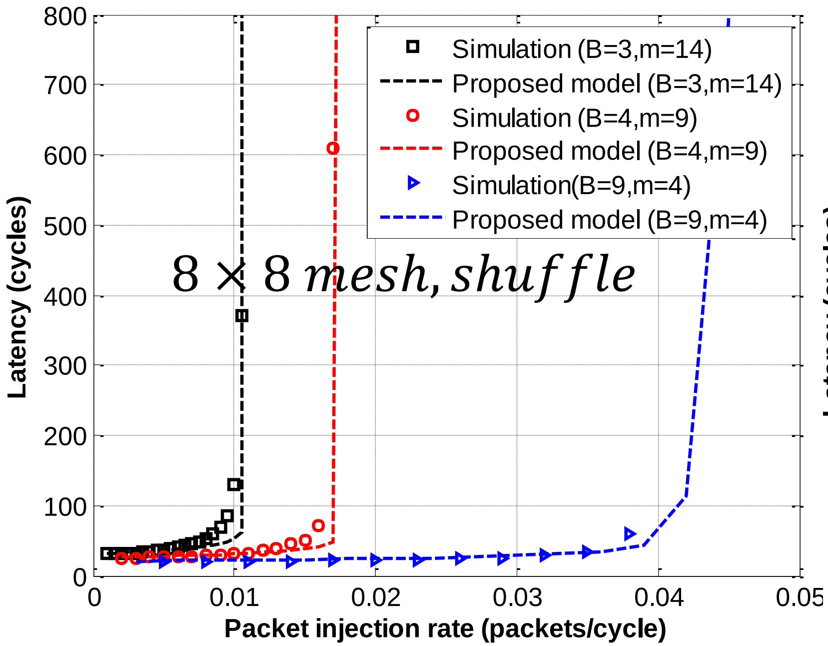
Evaluation under random traffic patterns

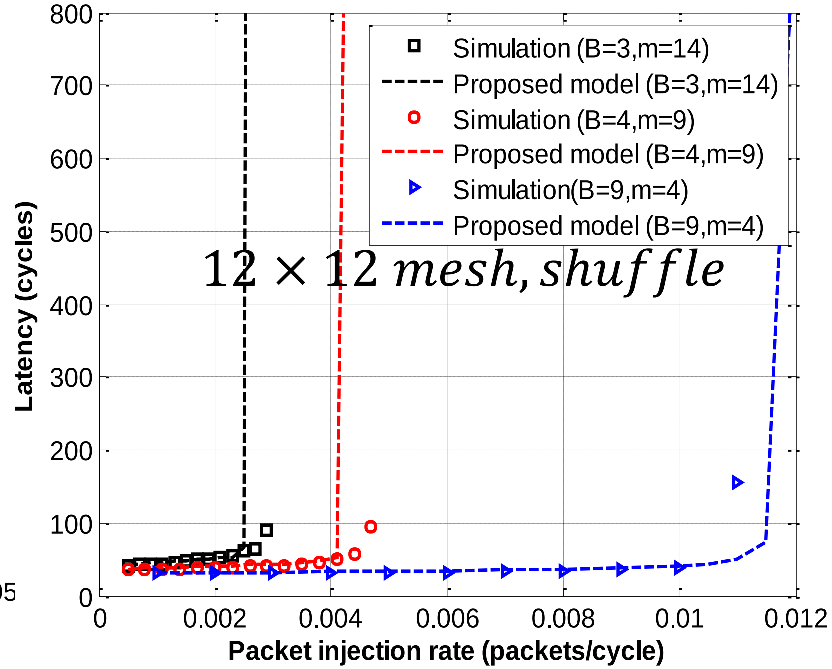




- The proposed latency model works for a variety of buffer depth and packet size combinations.
- For random traffic, about 5.2%-9.9% errors are introduced in predicting the network saturation point.

Evaluation under shuffle traffic patterns

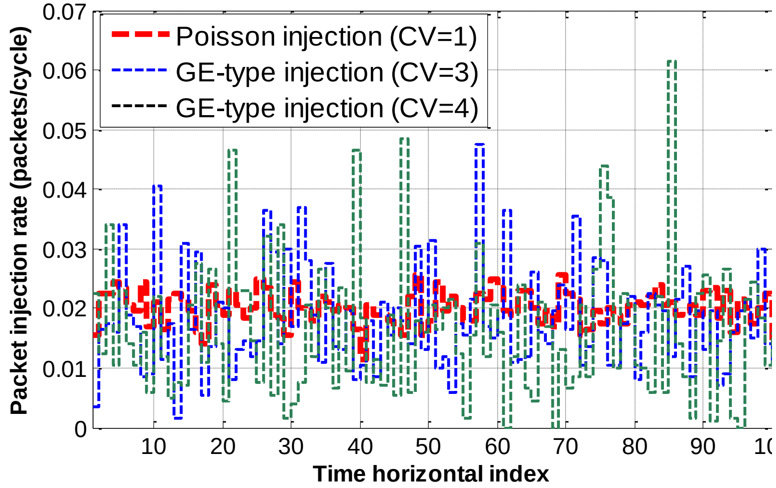


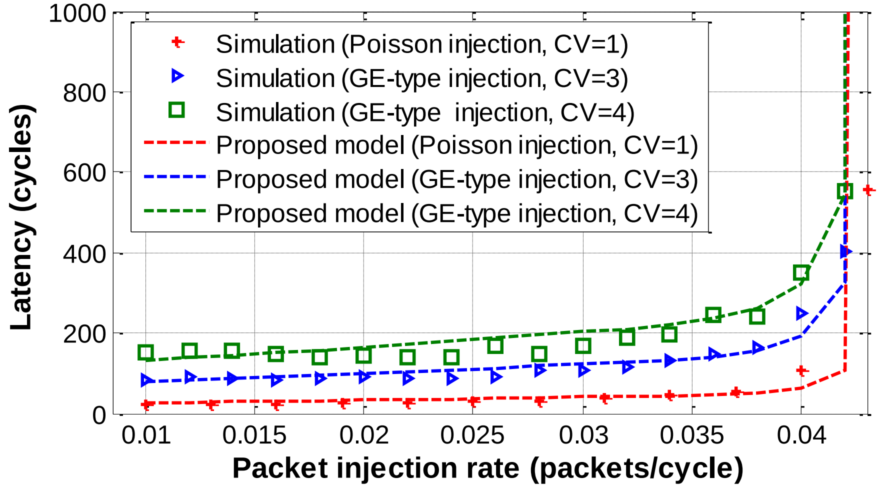


- For the traffic patterns such as shuffle, a little larger error (10.8%-13%) is introduced due to the uneven traffic arrival rates across the channels.
- Overall, the analytical model achieves 70X speedup over the simulations for both traffic patterns.

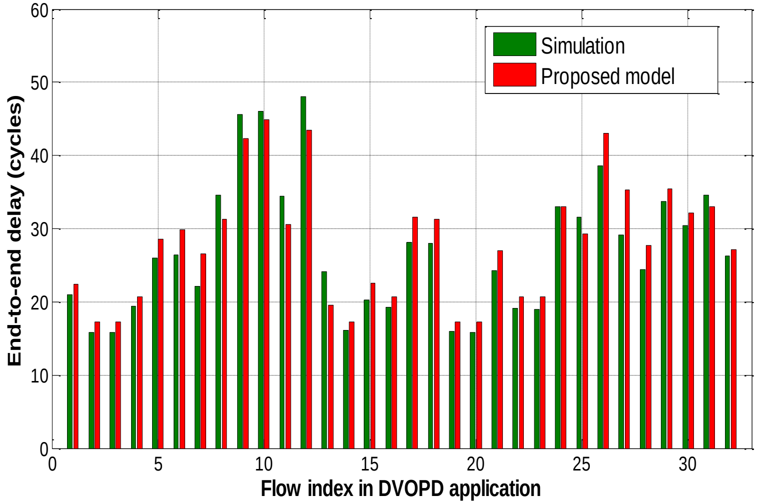
Evaluation under burst and real traffic

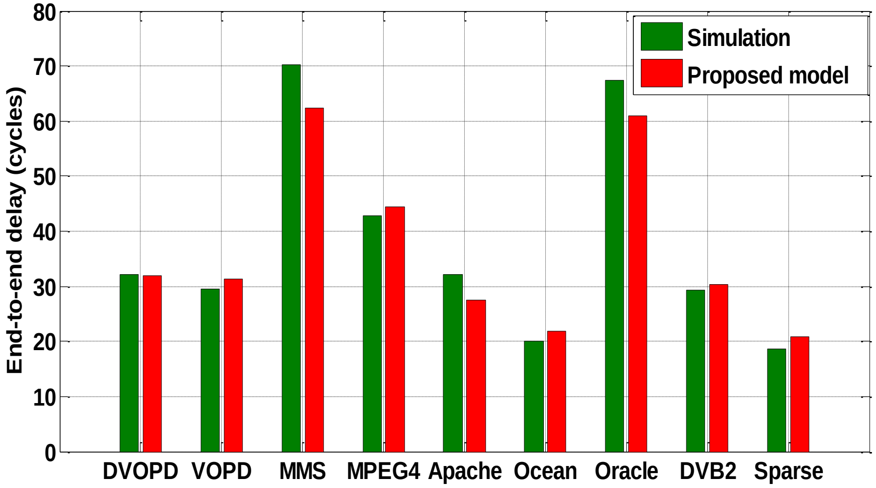
Comparison of Poisson and GE-type traffic injection:





Evaluation under real application traces:





Outline

- Introduction
- NoC Modeling for Performance Analysis
  - NoC end-to-end delay calculation
  - Link dependency analysis
  - GE-type traffic modeling
  - Wormhole router based NoC latency model
- Experimental results
  - Simulation setup
  - Evaluation under synthetic traffic patterns
  - Evaluation under realistic benchmarks
- **Conclusion**

Conclusion

- In this work, we propose a new NoC latency model which generalizes the previous work by modeling:
  - The arrival traffic burstiness
  - The general service time distribution
  - The finite buffer depth and arbitrary packet length combinations
- A link dependency analysis technique is proposed to determine the order of applying queuing analysis
- The accuracy of the model is demonstrated using both the synthetic traffic and real applications.
- A 70X speedup over simulation is achieved with less than 13% error in the proposed analytical model, which benefit the NoC synthesis process.

- **Thank you!!**

- **Q&A**