Modelling of Packet Loss in an Asynchronous Packet Switch using PEPA

Wim Vanderbauwhede

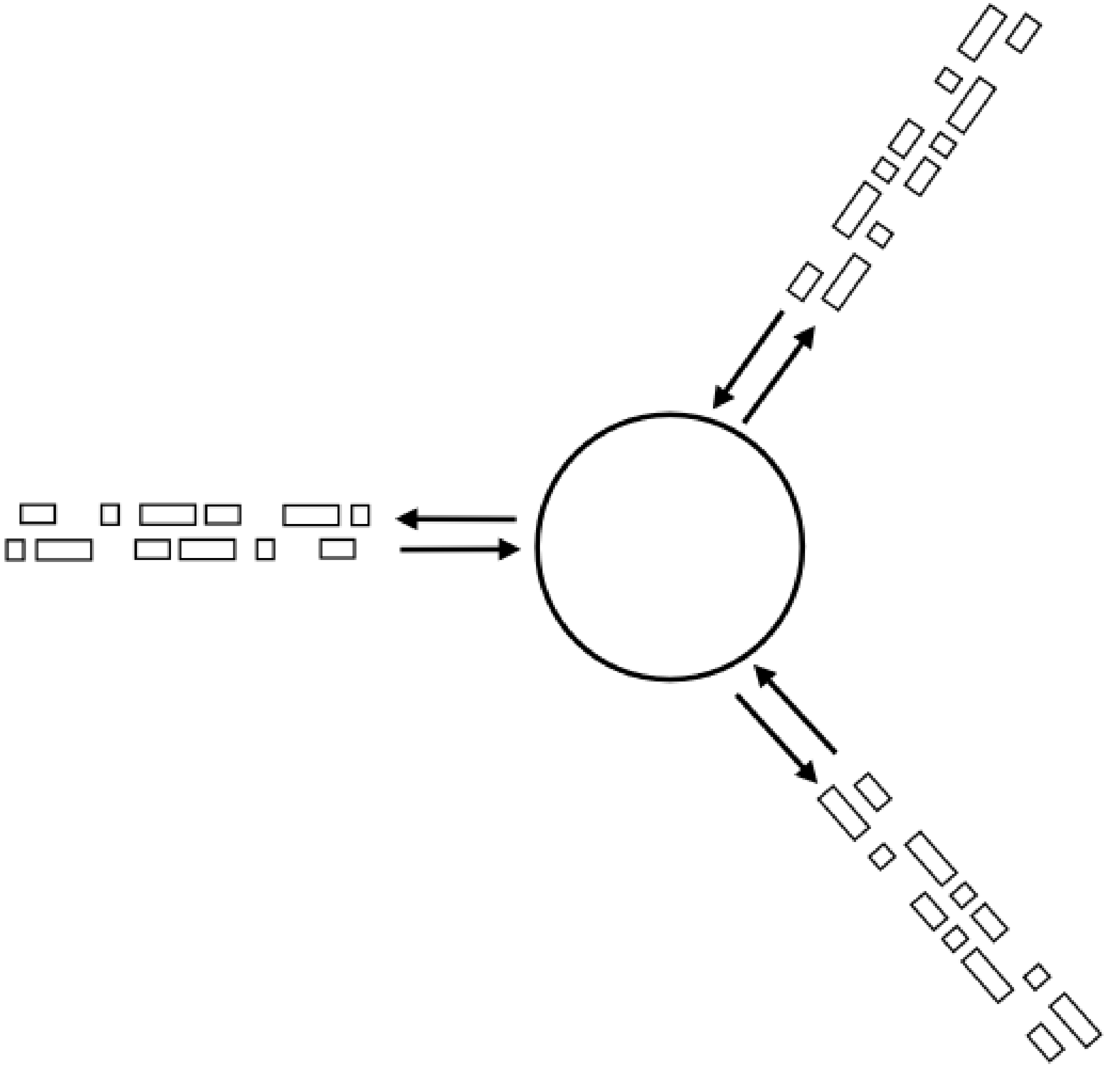
Department of Computing Science

University of Glasgow

Overview

- Asynchronous Packet Switch
- Building the PEPA Model
- Some Results
- Conclusion

Asynchronous Packet Switch (1)



Asynchronous Packet Switch (2)

Basic Functionality

- Switches data packets between ports
- Contention occurs when two or more packets want to occupy the same destination port
- Buffering is used to resolve contention

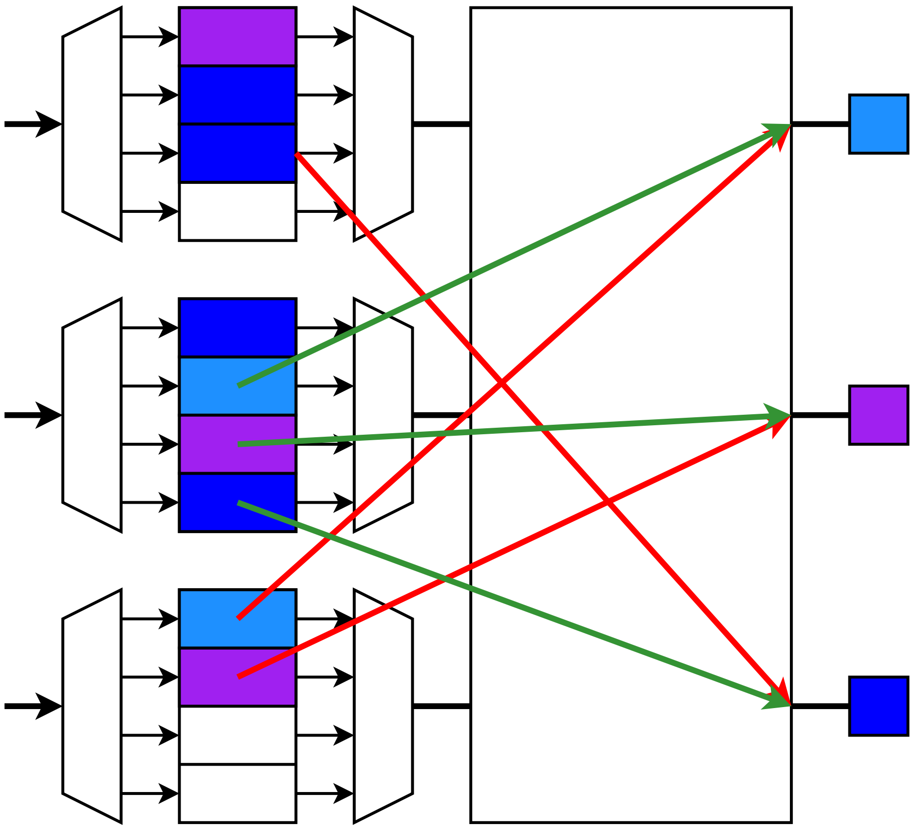
Asynchronous Packet Switch (3)

Properties

- Store-and-forward: packet is stored in buffer cell, then forwarded
- Packet must have left buffer cell completely before another packet can occupy the same cell
- Simultaneous egress from buffers is possible (transparent)
- Switch fabric is a black box

Asynchronous Packet Switch (4)

Schematic



Asynchronous Packet Switch (5)

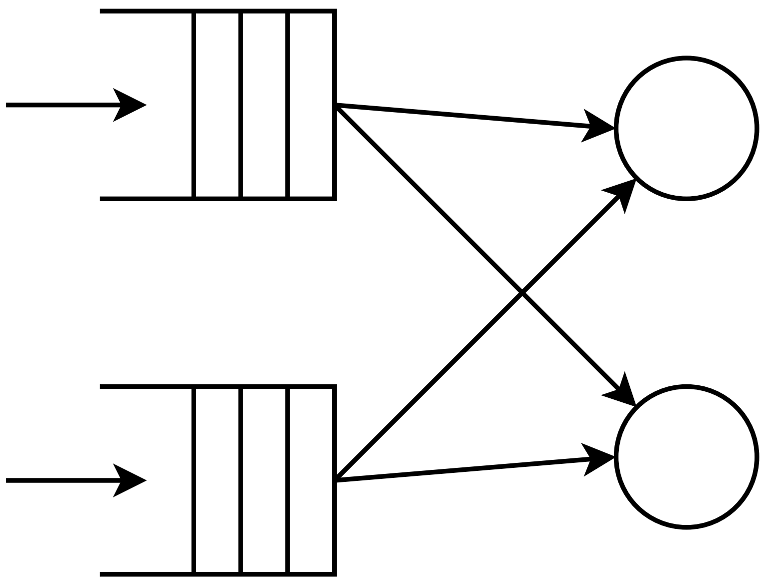
Steady-State Packet Loss

- Important performance measure for packet switch
- Depends on load and buffer depth,
- But also on traffic distribution and switch architecture.
- PEPA is useful to analyse the former, for the latter a DES is required.

Asynchronous Packet Switch (6)

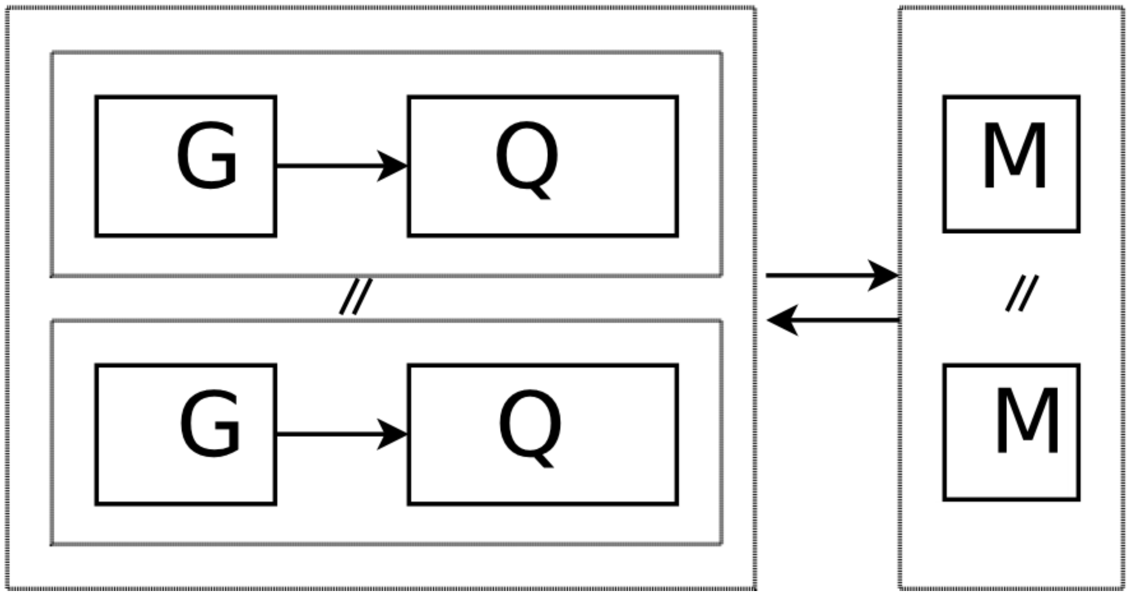
Queue Model for Packet Switch

The switch can be modelled as a system of c interacting $M/M/c/N$ queues:



The PEPA Model for the Switch

Organisation of the PEPA Model



G: traffic generator; Q: buffer queue; M: output multiplexer

//: components in parallel (empty interaction)

→: direction of active/passive interaction

Building the Model - Traffic generator

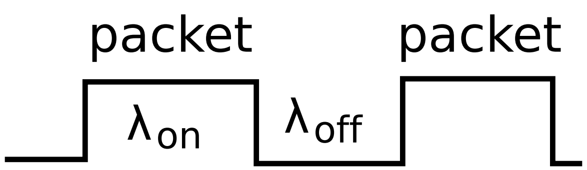
Two-state traffic generator

- Models packet traffic for packets with variable length and inter-arrival time
- We use following definitions:

  τ_{on}, τ_{off}: time period during which the source is on resp. off.

  $\lambda_{on} = 1/\tau_{on}$, $\lambda_{off} = 1/\tau_{off}$: the corresponding rates

- Model: $G = (off, \lambda_{off}).(on, \lambda_{on}).G$



Building the Model - Buffer system (1)

Buffer System Model: Definitions and Notations

- Define the **base state** Q_i as the set of states in which i out of N buffers are occupied
- Let j be the number of packets entering the buffer, k the number of packet leaving the buffer.
- We introduce following notation:

$$Q_i^{\substack{+j \\ -k}}, i \in \{0, \ldots, N\}; j \in \{0, 1\}; k \in \{0, \ldots, c\}$$

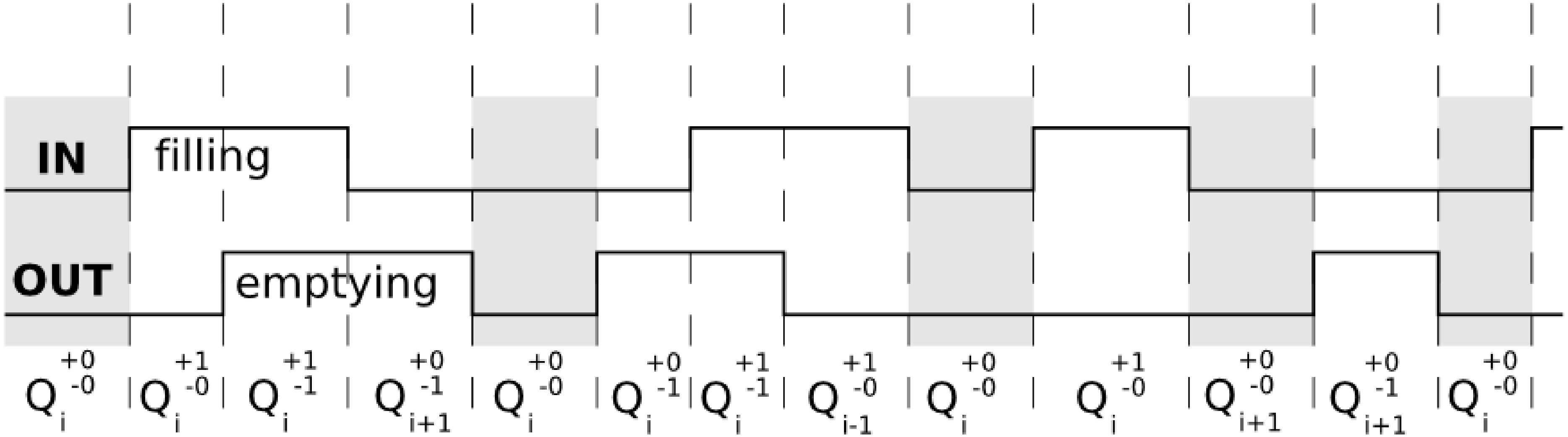
- Read as: "The queue *Q* has *i* filled buffers, *j* packets are arriving and *k* are leaving"

Building the Model - Buffer system (2)

Analysis of states and transitions

for a single M/M/c/N queue

- Example for M/M/1/N queue



Building the Model - Buffer system (3)

Actions and Rates

- State transitions are caused by the arrival of a packet or the arrival of a signal telling a packet to leave.
- Presence/absence of a packet at the ingress/egress port is modelled by the actions $\{on, off\}_{\{in,out\}}$ with corresponding rates $\lambda_{\{on,off\},\{in,out\}}$
- Buffer filling rate $\lambda_{on,in}$: models the packet length, and therefore $\lambda_{on,in} \equiv \lambda_{on,out}$.
- Signalling rate $\lambda_{off,out}$: models the delay between the the egress port becoming free and the packet starting to leave the buffer.

Building the Model - Buffer system (4)

Model for single egress ($c = 1$):

$$
\begin{aligned}
Q_i^{\overset{+0}{-0}} &= (off_{in}, \lambda_{off,in}).Q_i^{\overset{+1}{-0}} + (off_{out}, \lambda_{off,out}).Q_i^{\overset{+0}{-1}} \\
Q_i^{\overset{+1}{-0}} &= (on_{in}, \lambda_{on,in}).Q_{i+1}^{\overset{+0}{-0}} + (off_{out}, \lambda_{off,out}).Q_i^{\overset{+1}{-1}} \\
Q_i^{\overset{+0}{-1}} &= (off_{in}, \lambda_{off,in}).Q_i^{\overset{+1}{-1}} + (on_{out}, \lambda_{on,out}).Q_{i-1}^{\overset{+0}{-0}} \\
Q_i^{\overset{+1}{-1}} &= (on_{in}, \lambda_{on,in}).Q_{i+1}^{\overset{+0}{-1}} + (on_{out}, \lambda_{on,out}).Q_{i-1}^{\overset{+1}{-0}}
\end{aligned}
$$

Building the Model - Buffer system (5)

Introducing Drop States

- We are interested in the **packet loss in steady state**.
- We calculate this as **sum of the probabilities** of being in a state where the packet is being dropped.
- To do so, we must introduce “**drop states**”.
- The full state ($i = N$) leads to a **drop state** on arrival of a packet:

$$Q_N^{\substack{+0\\-0}} = (off_{in}, \lambda_{off,in}).Q_{N,d}^{\substack{+1\\-0}} + (off_{out}, \lambda_{off,out}).Q_N^{\substack{+0\\-1}}$$

$$Q_N^{\substack{+0\\-1}} = (off_{in}, \lambda_{off,in}).Q_{N,d}^{\substack{+1\\-1}} + (on_{out}, \lambda_{on,out}).Q_{N-1}^{\substack{+0\\-0}}$$

Building the Model - Buffer system (5)

- All base states ($0 < i \leq N$) gain 2 drop states: while the packet is being dropped, a packet can start/stop leaving the buffer, leading to a lower base state.

$$Q_{i,d}^{\substack{+1\\-0}} = (on_{in}, \lambda_{on,in}).Q_{i}^{\substack{+0\\-0}} + (off_{out}, \lambda_{off,out}).Q_{i,d}^{\substack{+1\\-1}}$$

$$Q_{i,d}^{\substack{+1\\-1}} = (on_{in}, \lambda_{on,in}).Q_{i}^{\substack{+0\\-1}} + (on_{out}, \lambda_{on,out}).Q_{i-1,d}^{\substack{+1\\-0}}$$

Building the Model - Buffer system (6)

Multiple Egress Model:

- Let $m = min(N, c)$ and $\lambda_{\{on,off\},out,k} = k.\lambda_{\{on,off\},out}$.
- The previous equations change to $(0 < i < N, 0 < k < m)$:

$$
\begin{aligned}
Q_{i}^{\substack{+0\\-k}} = (off_{in}, \lambda_{off,in}).Q_{i}^{\substack{+1\\-k}} + (on_{out}, \lambda_{on,out,k}).Q_{i-1}^{\substack{+0\\-(k-1)}} \\
+ (off_{out}, \lambda_{off,out,m-k}).Q_{i}^{\substack{+0\\-(k+1)}}
\end{aligned}
$$

$$
\begin{aligned}
Q_{i}^{\substack{+1\\-k}} = (off_{in}, \lambda_{off,in}).Q_{i+1}^{\substack{+0\\-k}} + (on_{out}, \lambda_{on,out,k}).Q_{i-1}^{\substack{+1\\-(k-1)}} \\
+ (off_{out}, \lambda_{off,out,m-k}).Q_{i}^{\substack{+1\\-(k+1)}}
\end{aligned}
$$

Building the Model - Buffer system (7)

Multiple Egress Drop States

- For $0 < i \leq N, 0 < k < m$

$$Q_{i,d}^{\substack{+1\\-k}} = (off_{in}, \lambda_{off,in}).Q_{i}^{\substack{+0\\-k}} + (on_{out}, \lambda_{on,out,k}).Q_{i-1,d}^{\substack{+1\\-(k-1)}} + (off_{out}, \lambda_{off,out,m-k}).Q_{i,d}^{\substack{+1\\-(k+1)}}$$

- For $0 < i \leq N, k = 0$

$$Q_{i,d}^{\substack{+1\\-0}} = (off_{in}, \lambda_{off,in}).Q_{i}^{\substack{+0\\-0}} + (off_{out}, \lambda_{off,out,m-k}).Q_{i,d}^{\substack{+1\\-1}}$$

- For $0 < i \leq N, k = m$

$$Q_{i,d}^{\substack{+1\\-m}} = (off_{in}, \lambda_{off,in}).Q_{i}^{\substack{+0\\-m}} + (on_{out}, \lambda_{on,out,k}).Q_{i-1,d}^{\substack{+1\\-(m-1)}}$$

Modelling Interacting Queues

- To combine *C* of the above queues Q_c with *c* outputs into a switch, we first introduce a multiplexer *M:*

$$M = (on_{out}, \top).(off_{out}, \lambda_{off,out}).M$$

- The final switch consists of C cooperations of G and Q_c in parallel, cooperating with *c* multiplexers in parallel

$$S_c = G \underset{on_{in},off_{in}}{\bowtie} Q_c$$

$$Switch = (S_c \parallel ... \parallel S_c) \underset{on_{out},off_{out}}{\bowtie} (M \parallel ... \parallel M)$$

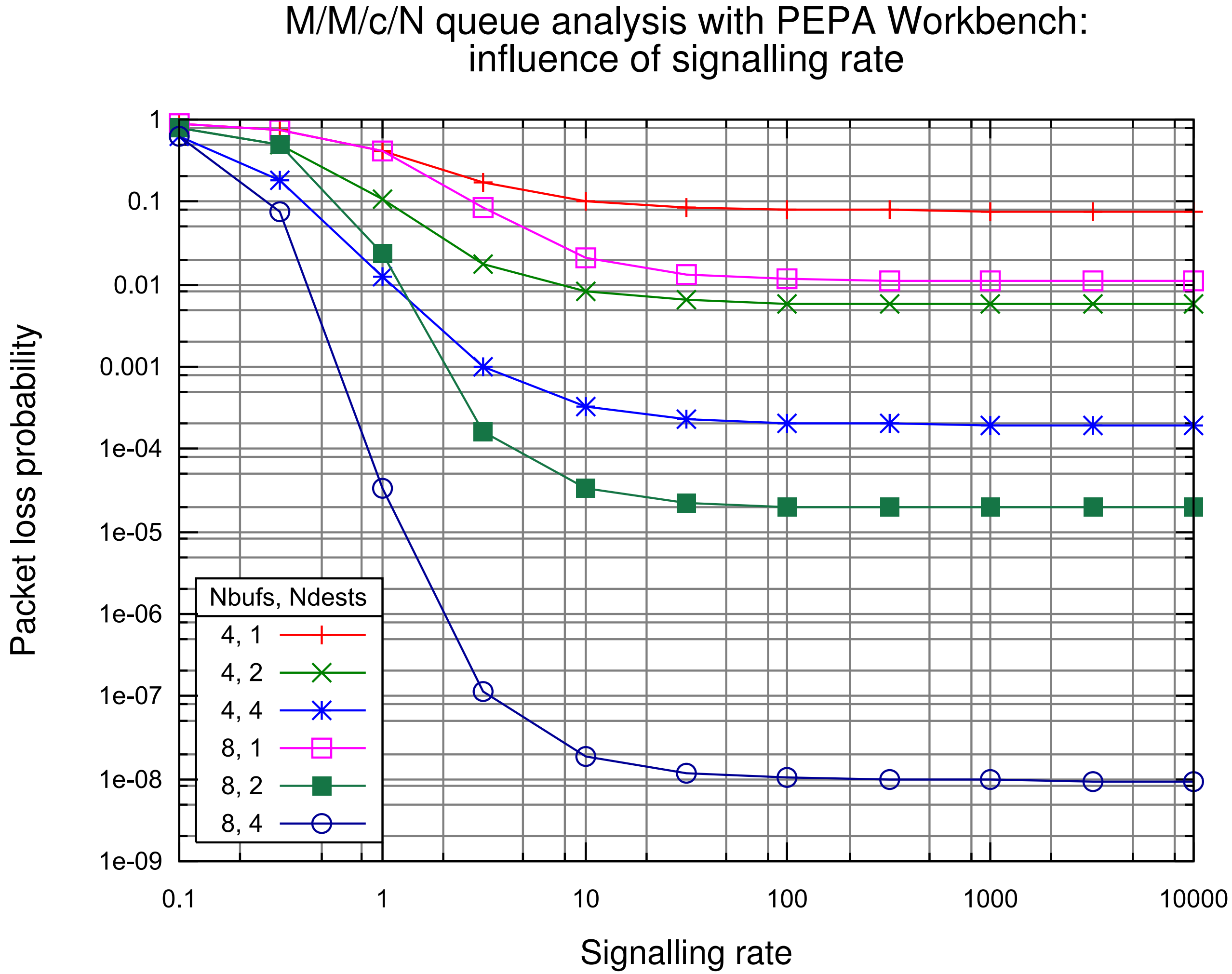
Using the PEPA Model

Toolchain for this work

- PEPA Workbench to calculate the TRM
- MatLab to calculate the steady-state solution
- A Perl script to drive the simulation:
  - generate the input file for the PEPA Workbench & run
  - generate a MatLab file & run
  - calculate the packet loss probability from the steady-state
- The Simulation::Automate Perl package to automate the DOE

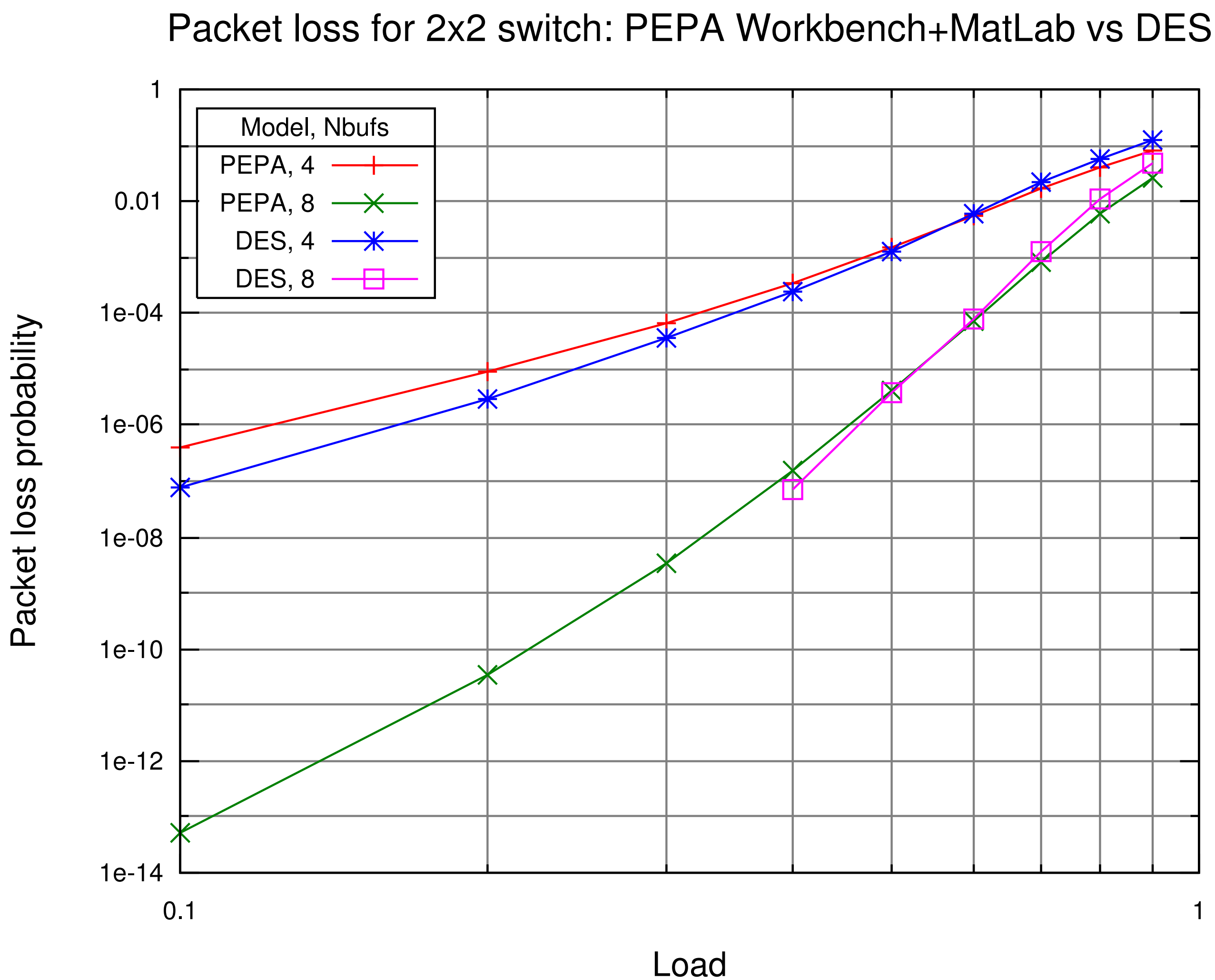
Some Results (1)

Influence of the Signalling Delay ($\frac{1}{\lambda_{off,out}}$)



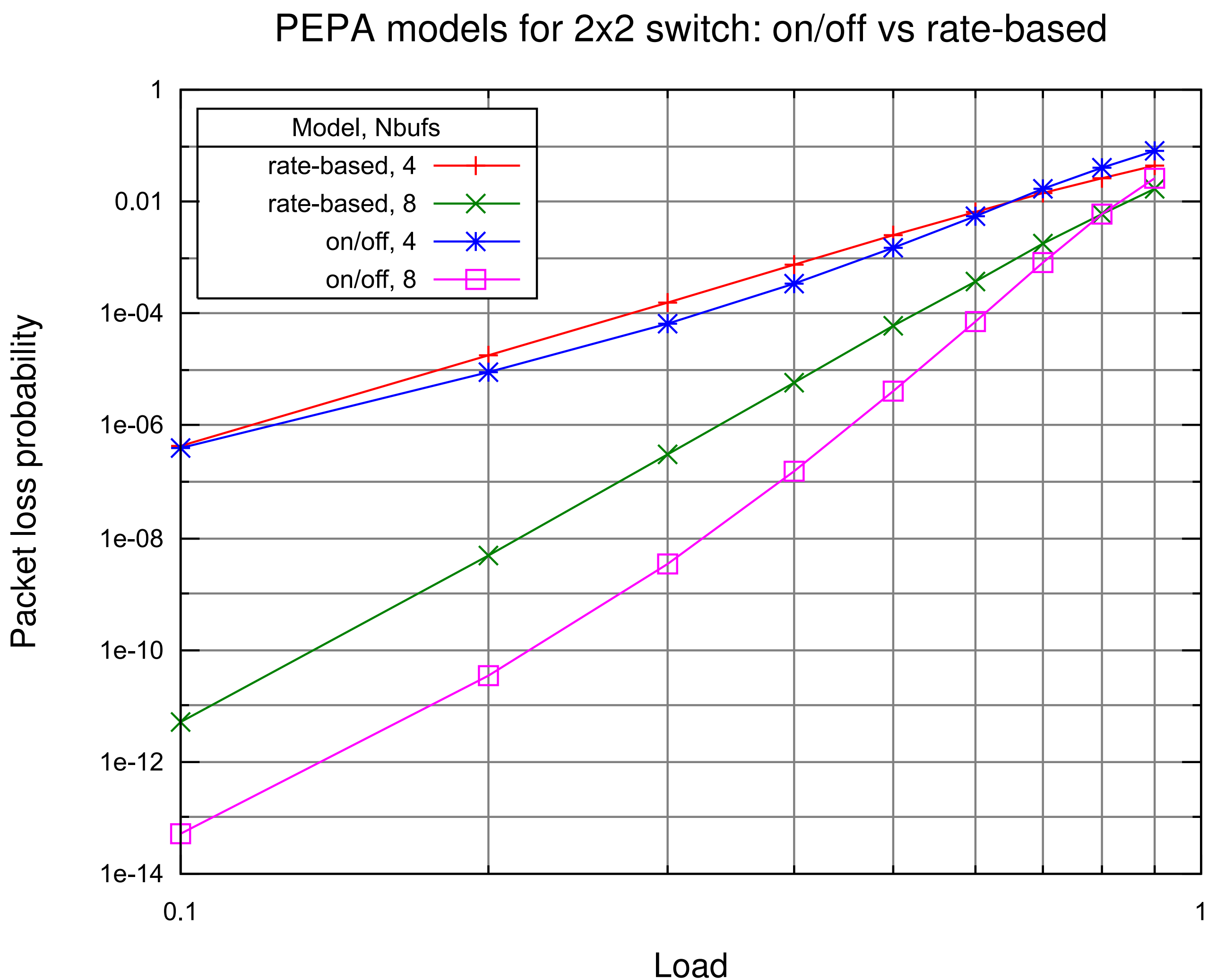
Some Results (2)

Comparison with Discrete Event Simulator



Some Results (3)

Comparison with Rate-based Model



Conclusion

- A methodology for analytical modelling of steady-state packet loss in an asynchronous packet switch
- For asynchronous buffered switches, the state space is very large
- Building a PEPA model with explicit states is non-trivial

Appendix: Rate-based Model (1)

Rate-Based PEPA Model

- Define a traffic generator generating packets at rate λ :

$$G = (in, \lambda).G$$

- And a multiplexer taking in packets at rate μ,defined as $\frac{\lambda}{\rho}$, with ρ the load:

$$M = (out, \mu).M$$

Appendix: Rate-based Model (2)

- The queue model is:

$$Q_0 = (in, \lambda).Q_1$$

$$Q_i = (in, \top).Q_{i+1} + (out, \mu).Q_{i-1} \quad , 0 < i < N$$

$$Q_N = (in, \top).Q_{Nd} + (out, \mu).Q_{N-1}$$

$$Q_{Nd} = (in, \top).Q_{Nd} + (out, \mu).Q_{N-1}$$