Centrum voor Wiskunde en Informatica

Performance Analysis of Reo Circuits

Chrétien Verhoef

Center for Mathematics and Computer Science (CWI)

CIC 2007

Research Workshop on Coinduction, Interaction and Composition

Amsterdam
2007-10-23

Overview

- Introduction / motivation
- Continuous Time Markov Chains
- Reo Coordination Language
- Quantitative Reo
- Performance Analysis of Reo Circuits
- Performance Analysis Example
- Future Research / To Do

Introduction

- Quantify performance of complex coordination systems
  - Communication networks, Web based services, etc.
- Typical questions:
  - Recognize bottlenecks?
  - Expected delay of the network?
  - Expected throughput?
  - Availability, blocking?
- Take into account:
  - Complex stochastic behaviour
  - Component dependencies of such systems
  - Quantitative behaviour environment

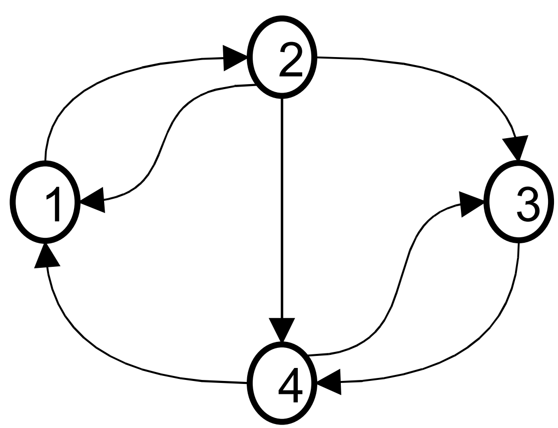
Overview

- Introduction / motivation
- **Continuous Time Markov Chains**
- Reo Coordination Language
- Quantitative Reo
- Performance Analysis of Reo Circuits
- Performance Analysis Example
- Future Research / To Do

Continuous Time Markov Chains

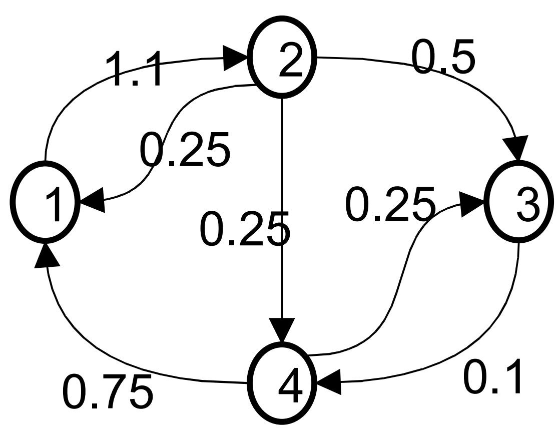
- Model coordination system as CTMC, then
- Quantify performance by using CTMC

- Markov Chain:
  - System can be in one of several states: state space
  - Transitions from one state to another



Continuous Time Markov Chains

- CTMC more concrete: Stochastic process { X(t) : t ≥ 0 }
- Process satisfies: - Markov property
  - X(t)   S   (finite state-space)
- Continuous: process defined for every time unit t ≥ 0
- Markov property: Only present state needed to determine transitions to next states → transitions with certain exponential transition rates (average # transitions per time unit)



Continuous Time Markov Chains

- Use CTMC to model complex coordination systems, and quantify performance

- Interested in long run behaviour CTMC
- Calculate steady-state probability vector:
  Probability CTMC is in a certain state at a random moment
- From steady-state probability vector →
  Calculate expected blocking probability, throughput, delay.

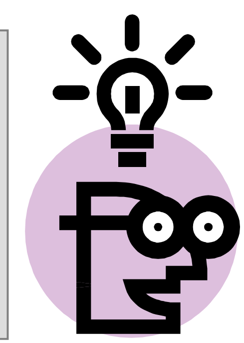
Continuous Time Markov Chains

- If we use CTMC to model system: find and define appropriate state description:
  - Finite state-space
  - Satisfies Markov property

Problems:

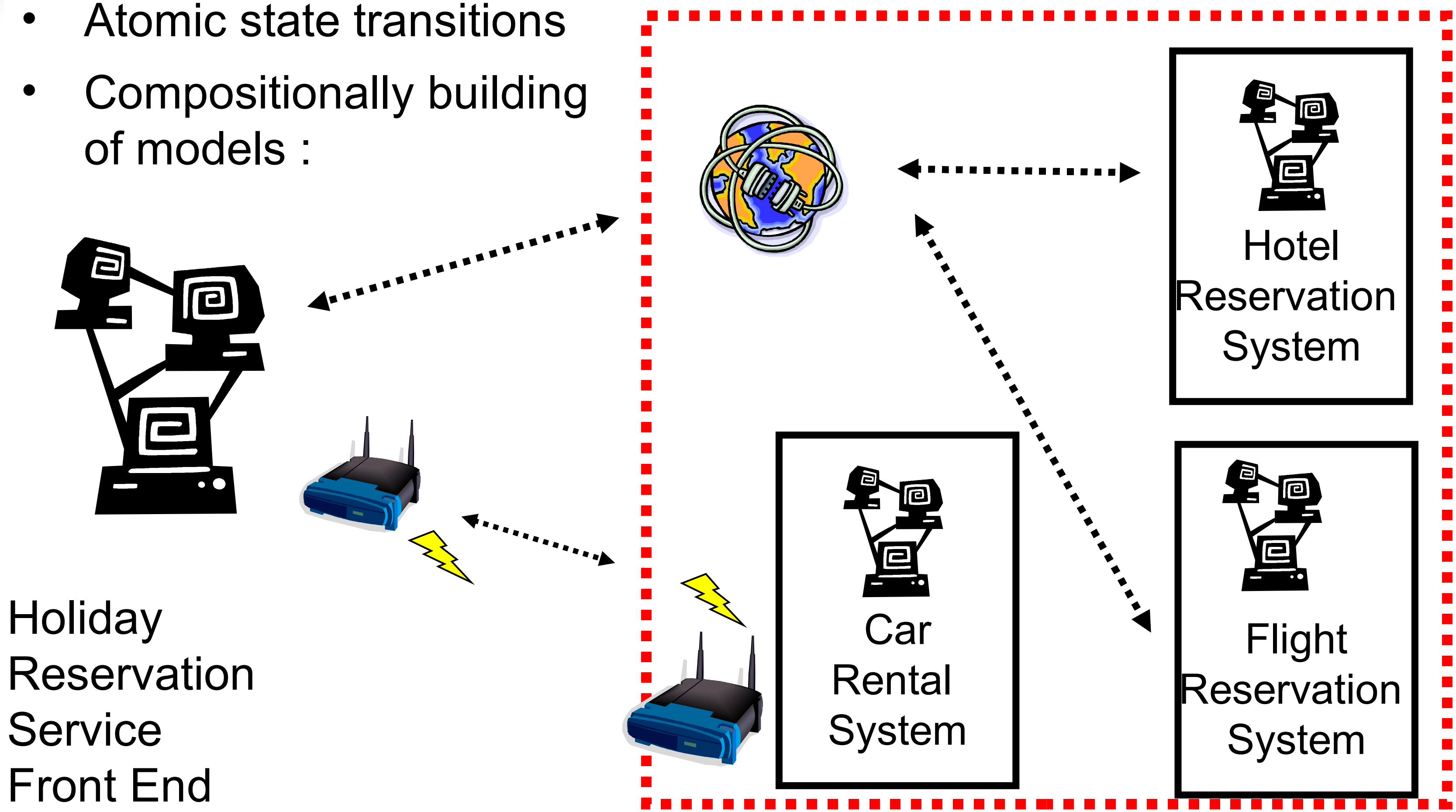
- Construction appropriate CTMC is complex
- Often leads to state-space explosion

- Idea:
  - Use Reo Coordination Language to model complex coordination systems
  - “Translate” Reo → CTMC
  - CTMC Performance Evaluation



Use the advantages of Reo:

- State-space reduction: strong synchronizing properties Reo
- Atomic state transitions
- Compositionally building of models :



Overview

- Introduction / motivation
- Continuous Time Markov Chains
- **Reo Coordination Language**
- Quantitative Reo
- Performance Analysis of Reo Circuits
- Performance Analysis Example
- Future Research / To Do

Reo Coordination Language

- Powerful means to model complex coordination systems
- Channel based coordination model, with synchronising / a-synchronising properties
- Compositional construction of models using architecturally meaningful primitives → Complex coordinators (connectors) compositionally build out of simpler ones
- Arbitrary set of simple connector types available, with well-defined behaviour
- Loose coupling of components
- Supports heterogeneous components
- Strong formal semantics

Reo Coordination Language

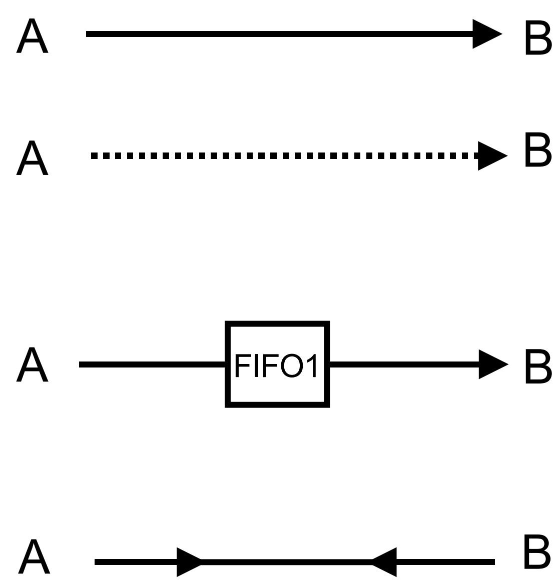
- Reo components connected through connectors (composition of simpler connectors)
- Most primitive connector is a channel (well defined behaviour):

Synchronous Channel

LossySync Channel

FIFO Channel

SyncDrain Channel



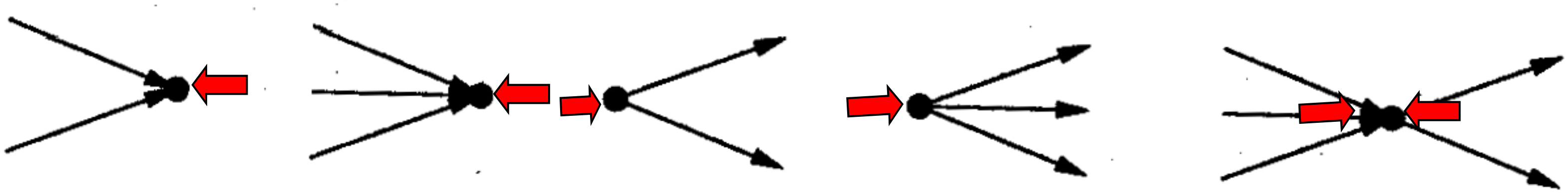
Etc.

Reo Coordination Language

Reo Nodes:

- Logical construct representing topological properties of a set of channels
- Regulates flow of data among channel ends

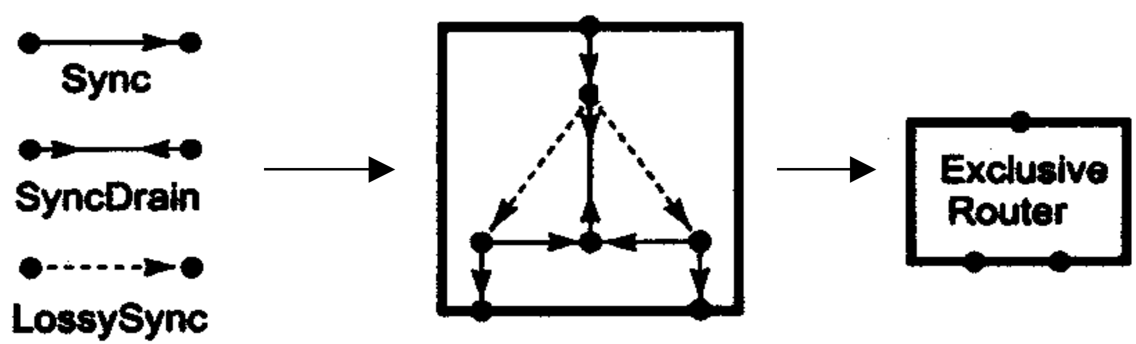
Sink Node	Source Node	Mixed Node



Arrival take actions (data dispersion))	Arrival write actions (data acceptance)	Arrival write/take actions

Reo Coordination Language

- Compositional building of models using architecturally meaningful primitives



Overview

- Introduction / motivation
- Continuous Time Markov Chains
- Reo Coordination Language
- **Quantitative Reo**
- Performance Analysis of Reo Circuits
- Performance Analysis Example
- Future Research / To Do

Quantitative Reo

- No quantitative stochastic behaviour in Reo circuits
- Ad performance properties to functional primitives:
  - Channels with delays
  - Ports with interarrival times between arriving read and write actions

Quantitative Reo: Channel Delays

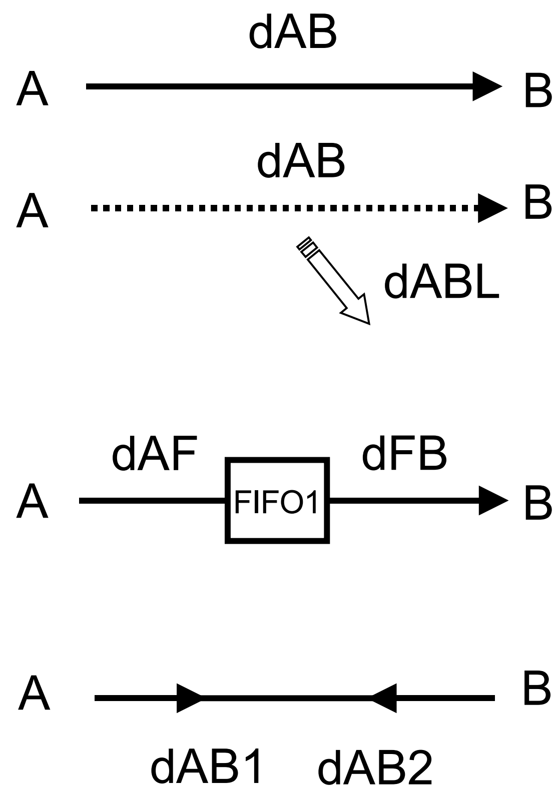
- Time it takes for a channel to synchronise with corresponding nodes

Synchronous Channel

LossySync Channel

FIFO Channel

SyncDrain Channel

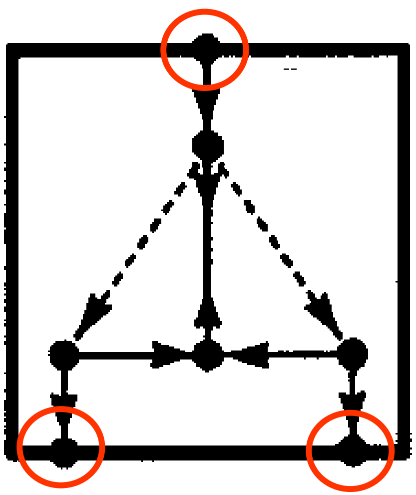


Quantitative Reo: Ports

Ports: source and sink nodes whereby circuit interacts with surrounding

Assumptions:

- Interarrival times of arriving read and write actions at ports can be specified
- Actions stay pending at ports until accepted
- At most one action can wait for acceptance at each port



- Blocking
- Denial write actions

Overview

- Introduction / motivation
- Continuous Time Markov Chains
- Reo Coordination Language
- Quantitative Reo
- **Performance Analysis of Reo Circuits**
- Performance Analysis Example
- Future Research / To Do

Performance Analysis of Reo Circuits

A. Construct quantitative Reo model

B. Use a special quantitative operational semantics for Reo language: Quantitative Intentional Automata (QIA)

C. Translate into CTMC:

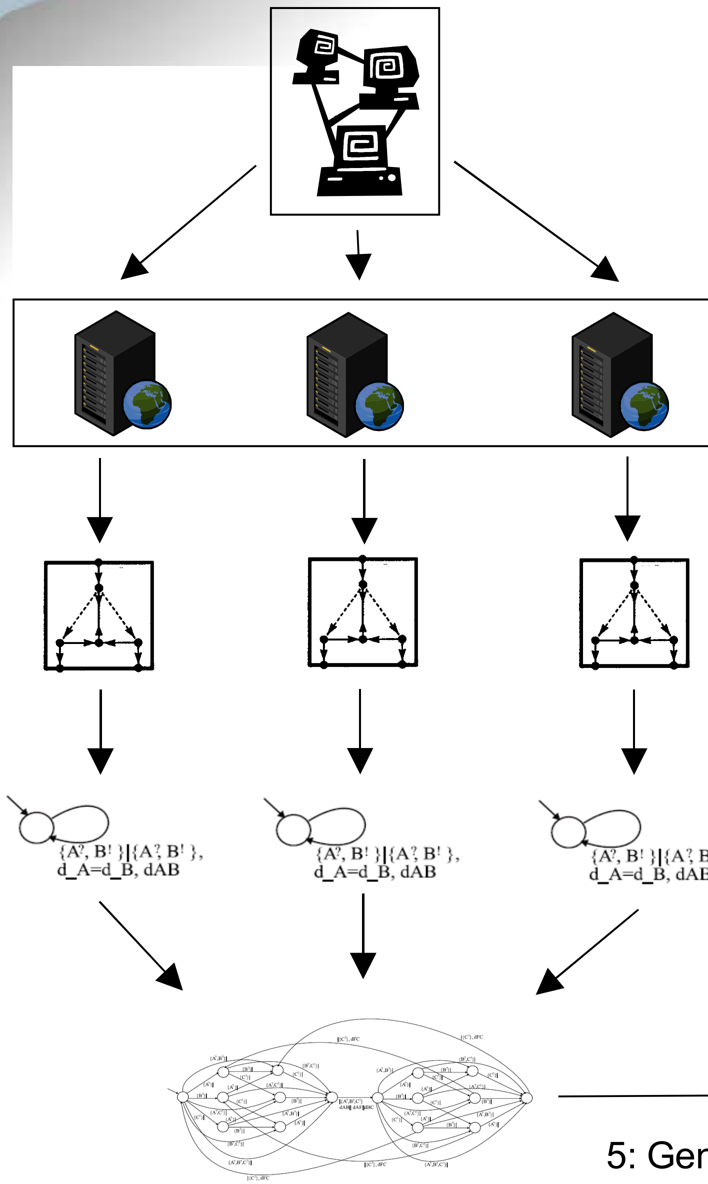
- CTMC State-space: all possible states of the Reo circuit Take into account:
  - State of ports
  - State of individual connectors (delay, buffers)
- Channel delays: independent exponentially distributed
- Interarrival times of arriving read and write actions at ports: independent exponentially distributed

Performance Analysis of Reo Circuits

D: Given steady-state distribution vector of CTMC:

- Insight in essential states, sensitivity analysis
- Approximate, e.g.:
  - Expected Blocking probabilities
  - Expected Delay: delay (sojourn time) after arrival at an arrival accepting port until arrival leaves the system
  - Expected Throughput: expected number of write requests on a certain port a circuit is able to handle per time unit

Overview: Performance Analysis of Reo Circuits



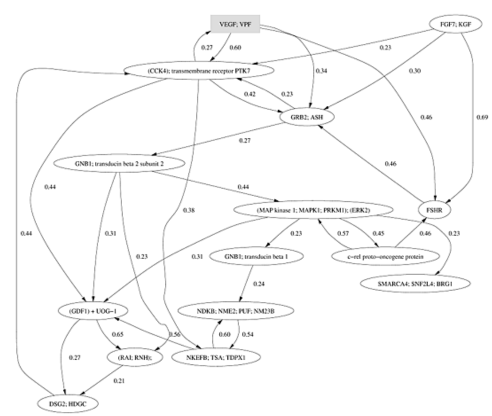
1: Decompose system

2: Reo Modeling

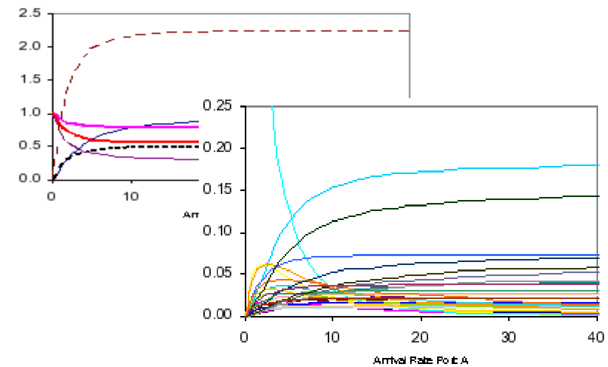
3: QIA Translation

4: Combine QIAs

5: Generate MC



6: Analyze MC

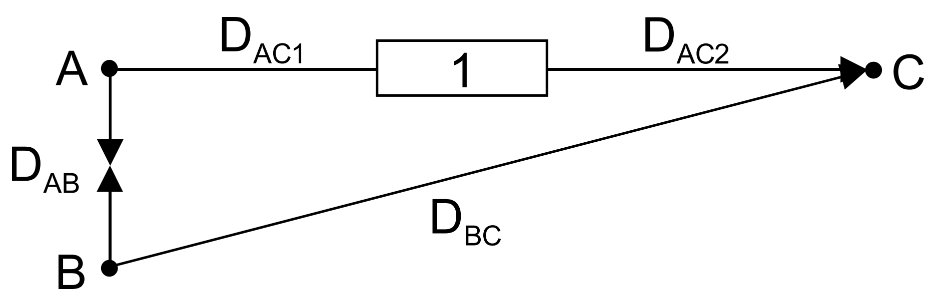


Overview

- Introduction / motivation
- Continuous Time Markov Chains
- Reo Coordination Language
- Quantitative Reo
- Performance Analysis of Reo Circuits
- **Performance Analysis Example**
- Future Research / To Do

Performance Analysis Example

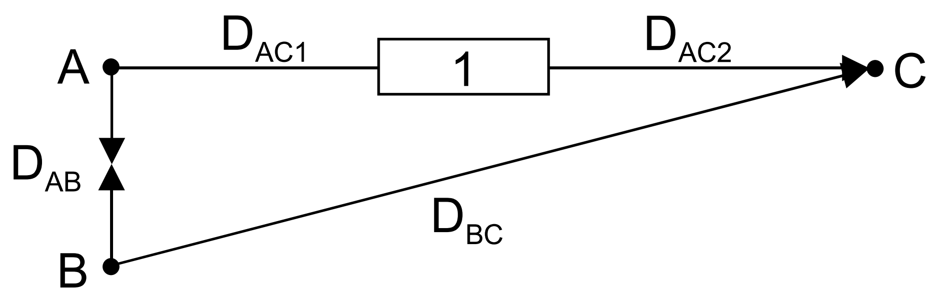
- Alternator:
  - Input ports: A, B
  - Output port: C
  - Delivers through port C, alternating what is available through A,B
- Reo Model:



- Data only accepted to flow from A into buffer:
  buffer empty, write pending at port B, read pending at port C

Performance Analysis Example

- Construct CTMC with:
  - Exponential connector delays: DAC1, DAC2, DAB, DBC
  - Exponential write action arrival rate at ports A, B
  - Exponential read action arrival rate at port C



Performance Analysis Example

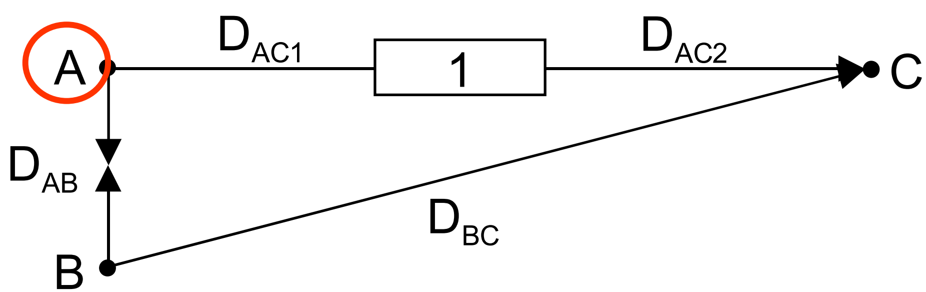
- Two examples:
  - Sensitivity analysis port A → vary mean arrival rate (=1/inter arrival time)
  - Sensitivity analysis delay BC→ vary mean delay

  - What will happen with: delays, blocking probabilities, steady-state distribution, throughput?

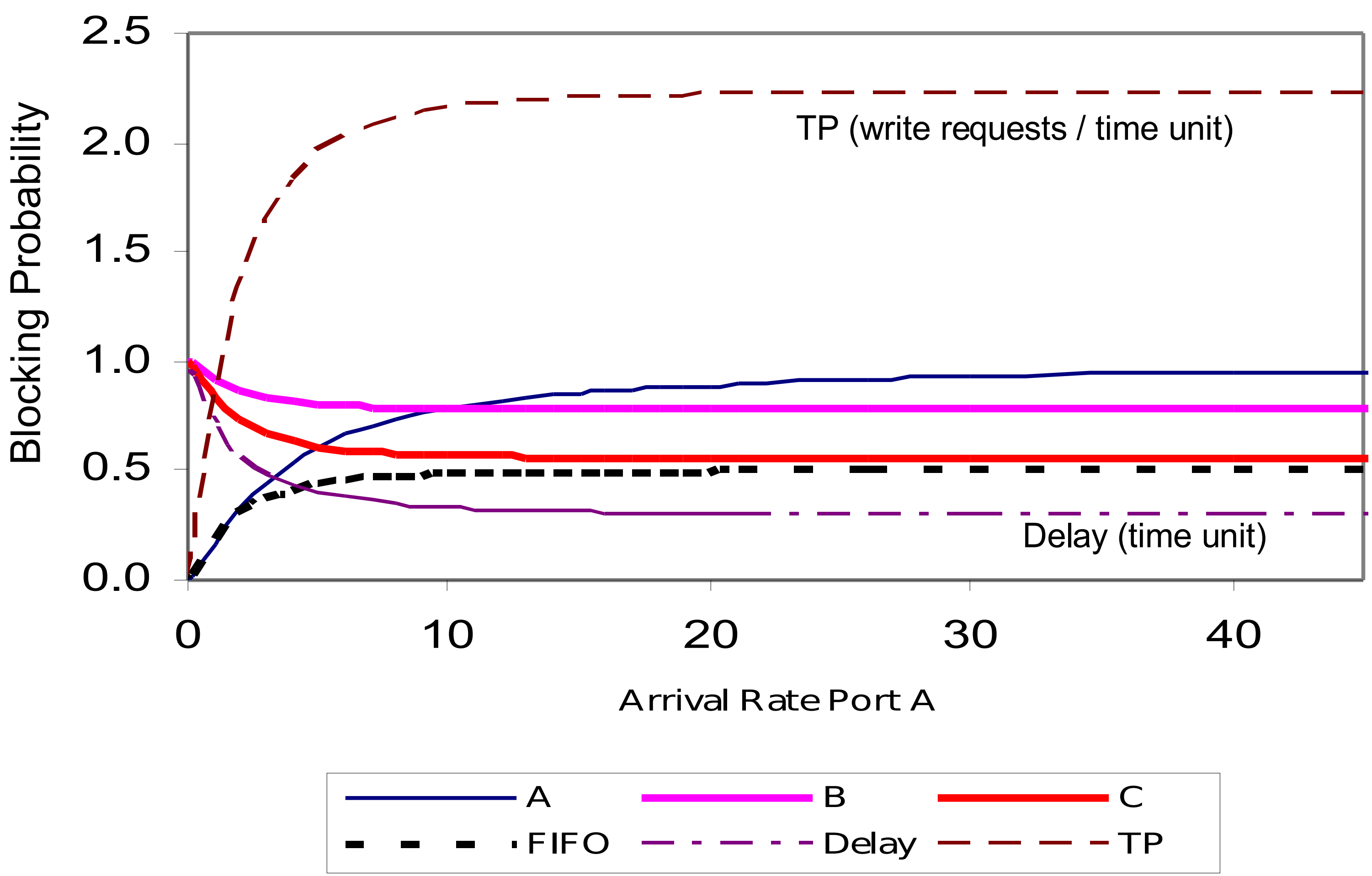
Performance Analysis Example

Sensitivity analysis port A:

vary mean arrival rate (=1/inter arrival time)

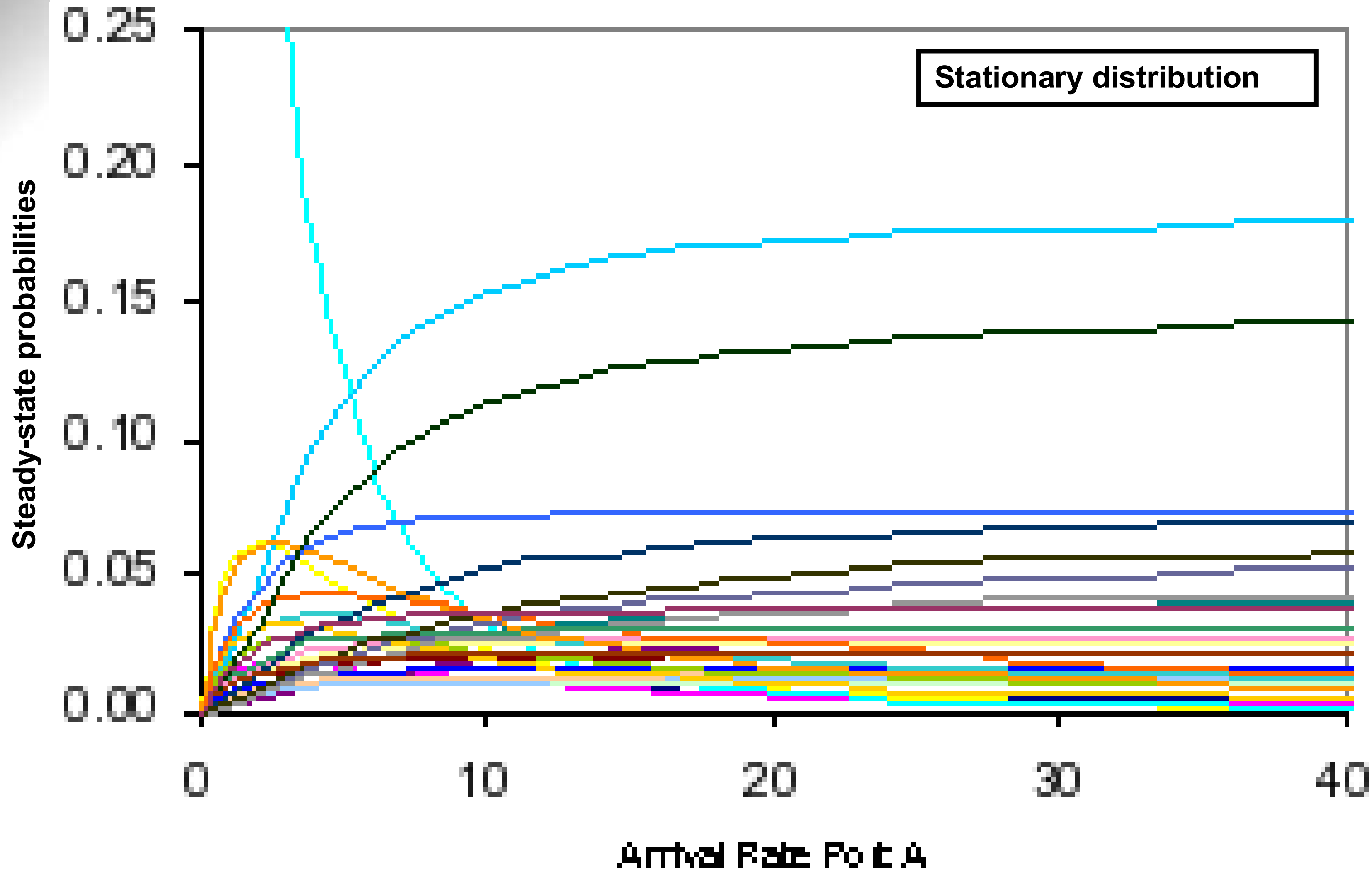


Performance Analysis Example: Port A



Centrum voor Wiskunde en Informatica

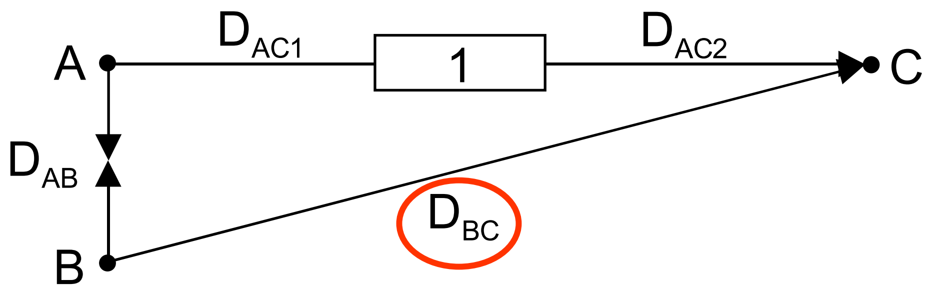
Performance Analysis Example: Port A



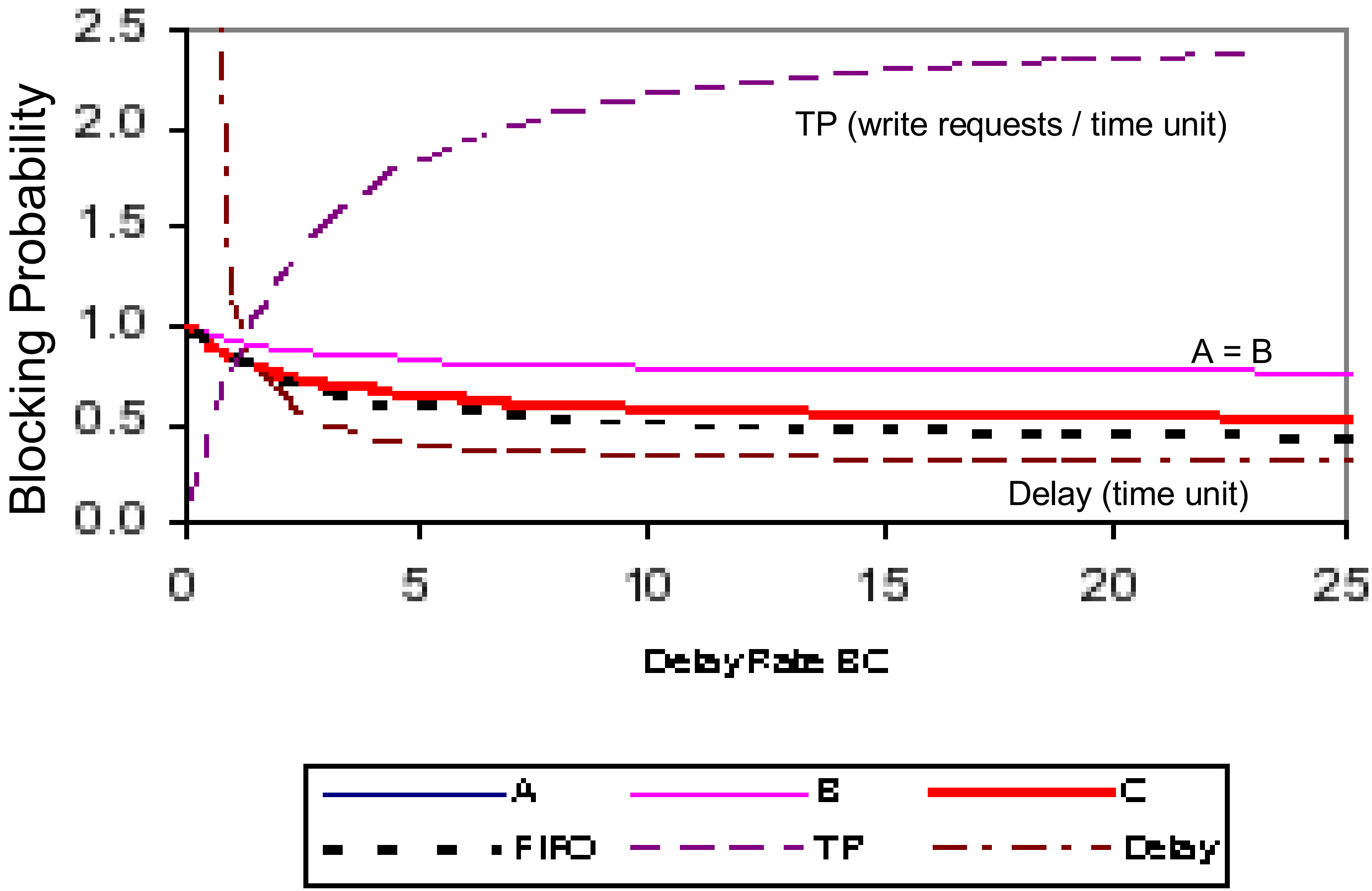
Performance Analysis Example

Sensitivity analysis delay BC:

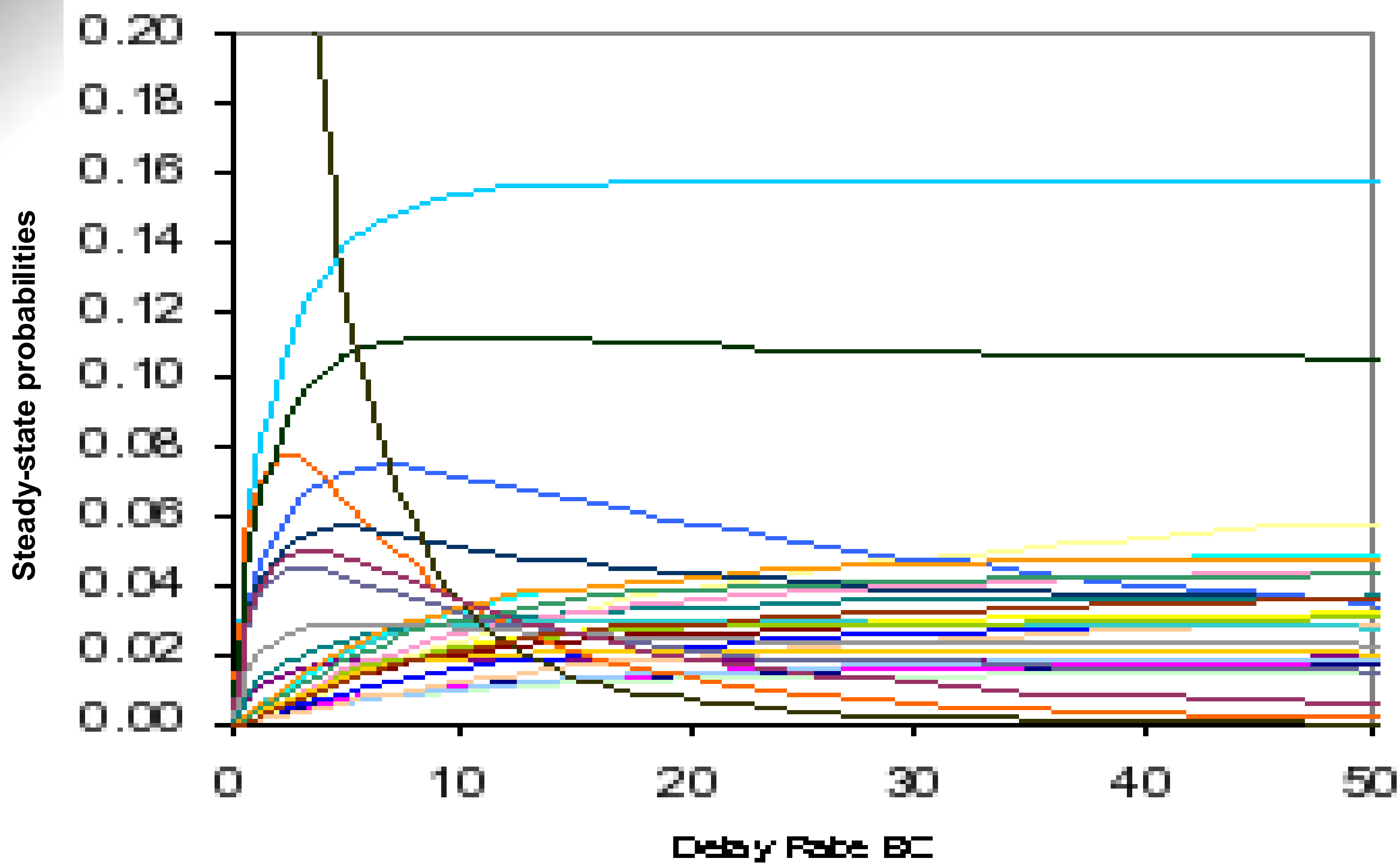
vary mean delay



Performance Analysis Example: Delay BC



Performance Analysis Example: Delay BC



Overview

- Introduction / motivation
- Continuous Time Markov Chains
- Reo Coordination Language
- Quantitative Reo
- Performance Analysis of Reo Circuits
- Performance Analysis Example
- **Future Research / To Do**

Further Research / “To Do”:

- Finish “automated” Reo-CTMC algorithm (QIA, QIA to CTMC)
  - → Current status: next presentation Young-Joo Moon
- Handle non-exponential distributed arrivals and delays?
- Introduce Reo nodes with delay? Now only delay on channels
- Calculate optimal policies for the behaviour of mixed nodes? Instead of completely randomized mixed node behaviour
- Recognize important states of Reo sub circuits and simplify CTMC by making approximations for those subsystems?

Centrum voor Wiskunde en Informatica

Performance Analysis of Reo Circuits

Chrétien Verhoef

Center for Mathematics and Computer Science (CWI)

CIC 2007

Research Workshop on Coinduction, Interaction and Composition

Amsterdam
2007-10-23