Towards Performance Prediction of Compositional Models in Industrial GALS Designs

Nicolas Coste
Inria Grenoble – Rhône-Alpes
STMicroelectronics Grenoble

Holger Hermanns
Universität des Saarlandes
Inria Grenoble – Rhône-Alpes

Etienne Lantreibecq
STMicroelectronics Grenoble

Wendelin Serwe
Inria Grenoble – Rhône-Alpes

Talk outline

- Introduction
- Modelling Flow
- Performance Flow
- Case-Study: the *xSTream* Architecture
- Conclusion

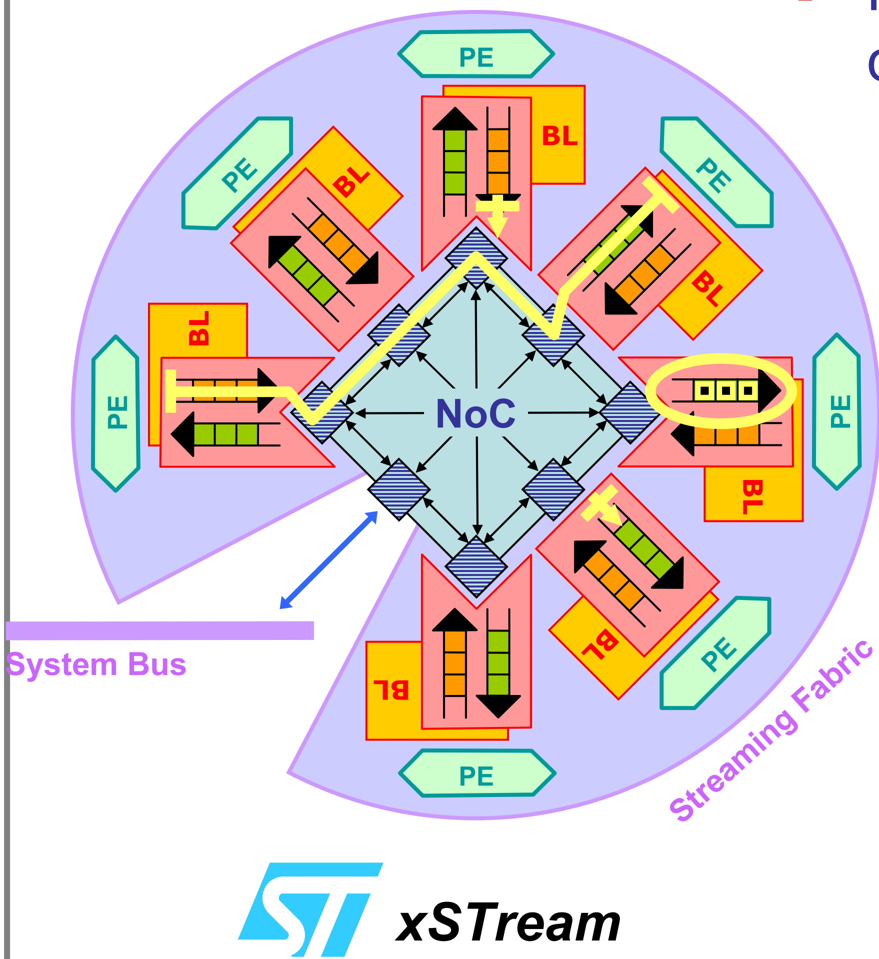
Introduction

- Systems-on-Chip (SoCs) are targeted:
  - Complexity increasing in time (parallelism)
  - Validity required: functional and performance correctness

- Performance measures are required before prototypes and precise description of the architecture are available

- Currently used methods for performance evaluation are rough (based on simulations).

Introduction



- Multiprocessor dataflow architecture designed at STMicroelectronics
  - Target: high performance embedded multimedia streaming applications
- Expected Performance measures:
  - Latency
  - Throughput
  - Resource utilization

Talk outline

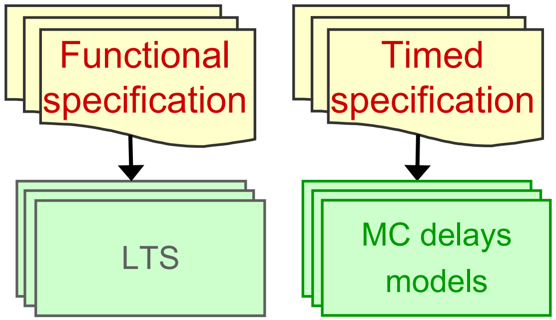
- Introduction
- Modelling Flow
- Performance Flow
- Case-Study: the *xSTream* Architecture
- Conclusion

Modelling Flow

- LOTOS models available for functional verification
  - Reuse those functional models

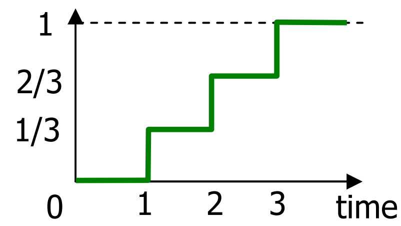
    "1 single model for functional verification and performance evaluation"
  - Keep the compositional approach (and use of non-determinism for abstraction)
- From a functional to a timed model:
  - Enrichment of LTS models by time information
  - State space explosion prevention
    (classical problem using parallel composition)
  - Preservation of performance properties w.r.t. compositions

Modelling Flow

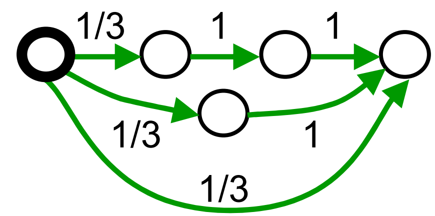


- Time model: Markov Chain (MC)
  Probabilistic steps are time steps!

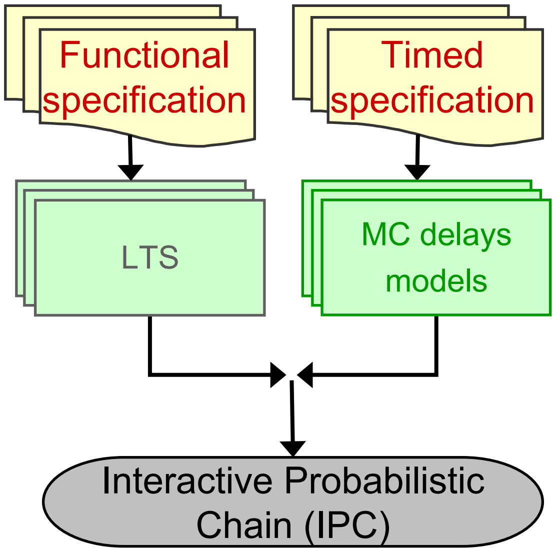
Cumulative Distribution Function (CDF) of a delay



Markov chain model

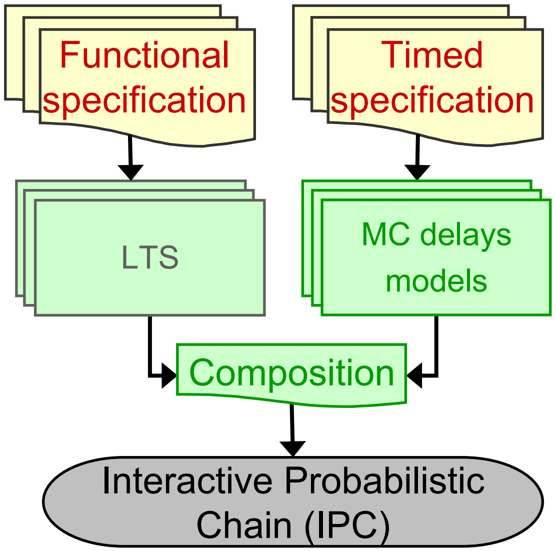


Modelling Flow



- **Time model:** Markov Chain (MC)
  Probabilistic steps are time steps!
- **Model:** Interactive Probabilistic Chain (IPC)
  An IPC $D = \langle S, \mathcal{A}, \longrightarrow, \Longrightarrow, \hat{s} \rangle$ is a quintuple where:
  - S is a set of states
  - $\mathcal{A}$ is a set of actions (including τ)
  - $\longrightarrow \subset S \times \mathcal{A} \times S$ is a set of interactive transitions
  - $\Longrightarrow \subset S \times [0, 1] \times S \rightarrow \mathbb{N}$ is a multi-set of probabilistic transitions
  - $\hat{s} \in S$ is the initial state

Modelling Flow



- Time model: Markov Chain (MC)
  Probabilistic steps are time steps!
- Model: Interactive Probabilistic Chain (IPC)
- Semantic Rules

Interactive Transitions

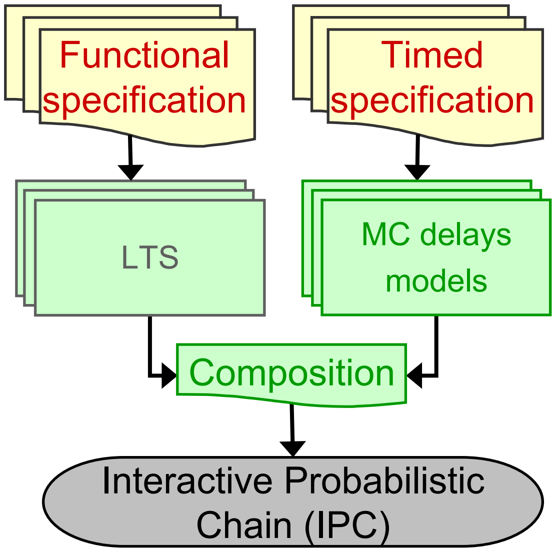
$$\frac{B_1 \xrightarrow{a} B_1' \quad a \notin A}{B_1 |[A]| B_2 \xrightarrow{a} B_1' |[A]| B_2} \qquad \frac{B_1 \xrightarrow{a} B_1'}{B_1 [\,] B_2 \xrightarrow{a} B_1'}$$

$$\frac{B_2 \xrightarrow{a} B_2' \quad a \notin A}{B_1 |[A]| B_2 \xrightarrow{a} B_1 |[A]| B_2'} \qquad \frac{B_2 \xrightarrow{a} B_2'}{B_1 [\,] B_2 \xrightarrow{a} B_2'}$$

$$\frac{\widetilde{B} = B \quad B \xrightarrow{a} B'}{\widetilde{B} \xrightarrow{a} B'} \qquad \frac{}{a; B \xrightarrow{a} B}$$

$$\frac{B_1 \xrightarrow{a} B_1' \quad B_2 \xrightarrow{a} B_2' \quad a \in A}{B_1 |[A]| B_2 \xrightarrow{a} B_1' |[A]| B_2'}$$

Modelling Flow



- Time model: Markov Chain (MC)
  Probabilistic steps are time steps!
- Model: Interactive Probabilistic Chain (IPC)
- Semantic Rules

Interactive Transitions

$$\frac{B_1 \xrightarrow{a} B_1' \qquad a \notin A}{B_1 |[\, A \,]| B_2 \xrightarrow{a} B_1' |[\, A \,]| B_2} \qquad \frac{B_1 \xrightarrow{a} B_1'}{B_1 [\,] B_2 \xrightarrow{a} B_1'}$$

$$\frac{B_2 \xrightarrow{a} B_2' \qquad a \notin A}{B_1 |[\, A \,]| B_2 \xrightarrow{a} B_1 |[\, A \,]| B_2'} \qquad \frac{B_2 \xrightarrow{a} B_2'}{B_1 [\,] B_2 \xrightarrow{a} B_2'}$$

$$\frac{\widetilde{B} = B \qquad B \xrightarrow{a} B'}{\widetilde{B} \xrightarrow{a} B'} \qquad \frac{}{a;B \xrightarrow{a} B}$$

$$\frac{B_1 \xrightarrow{a} B_1' \qquad B_2 \xrightarrow{a} B_2' \qquad a \in A}{B_1 |[\, A \,]| B_2 \xrightarrow{a} B_1' |[\, A \,]| B_2'}$$

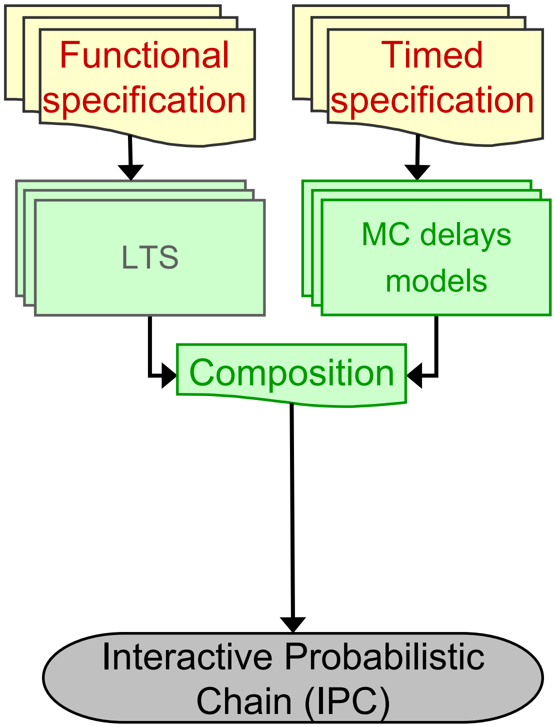
Probabilistic Transitions

$$\frac{}{\delta \xRightarrow{1} \delta} \qquad \frac{}{\sum_i p_i :: B_i \xRightarrow{p_i} B_i}$$

$$\frac{B_1 \xRightarrow{p_1} B_1' \qquad B_2 \xRightarrow{p_2} B_2'}{B_1 [\,] B_2 \xRightarrow{p_1\, p_2} B_1' [\,] B_2'} \qquad \frac{\widetilde{B} = B \qquad B \xRightarrow{p} B'}{\widetilde{B} \xRightarrow{p} B'}$$

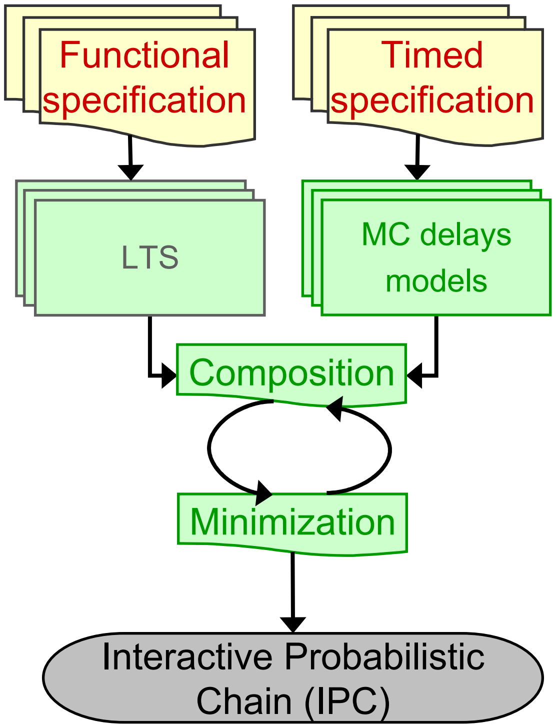
$$\frac{B_1 \xRightarrow{p_1} B_1' \qquad B_2 \xRightarrow{p_2} B_2'}{B_1 |[\, A \,]| B_2 \xRightarrow{p_1\, p_2} B_1' |[\, A \,]| B_2'} \qquad \frac{}{a;B \xRightarrow{1} a;B}$$

Modelling Flow



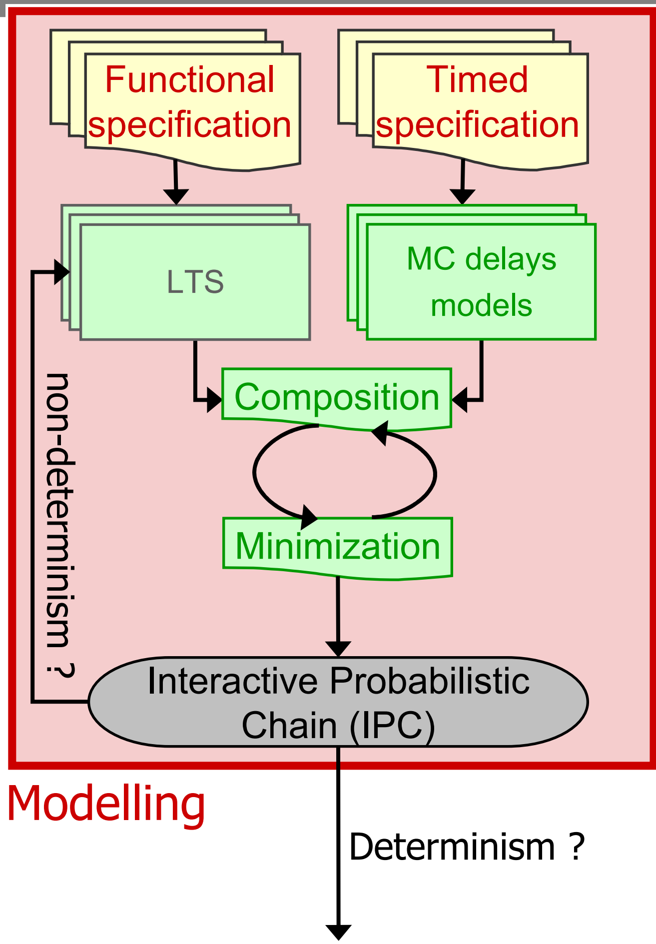
- Time model: Markov Chain (MC)
  Probabilistic steps are time steps!
- Model: Interactive Probabilistic Chain (IPC)
- Semantic Rules
- Compositional approach
  - Fight against state space explosion
  - Definition of a branching probabilistic bisimulation (b.p.b.)
  - The b.p.b. is a congruence w.r.t. the parallel operator

Modelling Flow



- **Time model:** Markov Chain (MC)
  Probabilistic steps are time steps!
- **Model:** Interactive Probabilistic Chain (IPC)
- Semantic Rules
- Compositional approach
  - Fight against state space explosion
  - Definition of a branching probabilistic bisimulation (b.p.b.)
  - The b.p.b. is a congruence w.r.t. the parallel operator
  - Iteratively: minimize w.r.t. the b.p.b. and compose

Modelling Flow

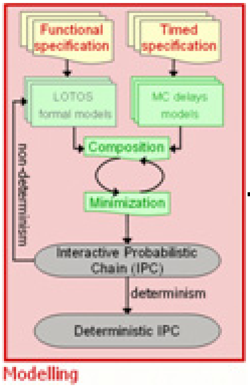


- Time model: Markov Chain (MC)
  Probabilistic steps are time steps!
- Model: Interactive Probabilistic Chain (IPC)
- Semantic Rules
- Compositional approach
- Management of non-determinism

Talk outline

- Introduction
- Modelling Flow
- Performance Flow
- Case-Study: the *xSTream* Architecture
- Conclusion

Performance Flow



non-zeno

Deterministic IPC (dIPC)

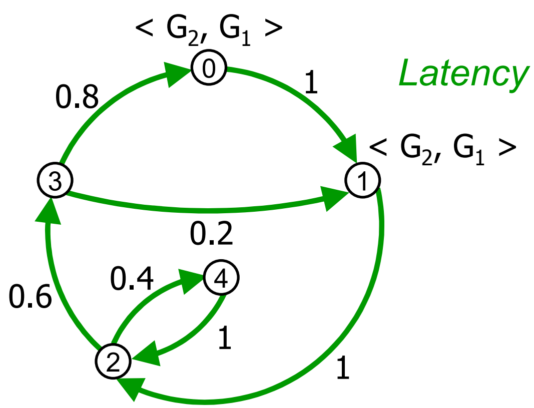
Minimization guided by targeted latency

Deterministic IPC
reduced state space

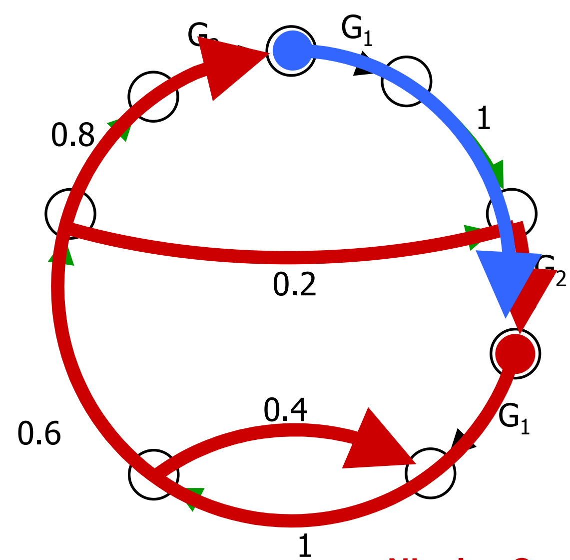
Associated Markov chain

- Targeted result: latency distribution

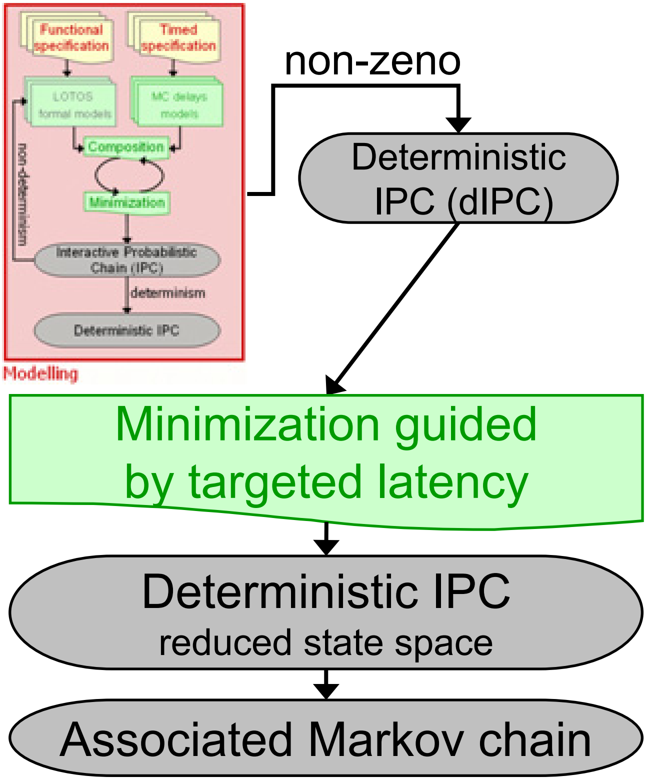
latency: in a dIPC = time between two interactive transitions



Latency between G_1 and G_2 ?



Performance Flow



- Targeted result: latency distribution

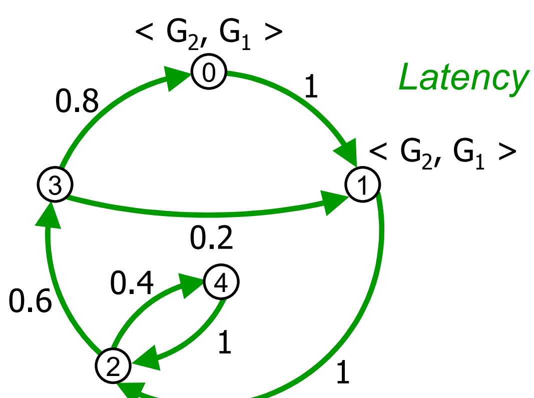
  latency: in a dIPC = time between two interactive transitions

- Study of the associated MC

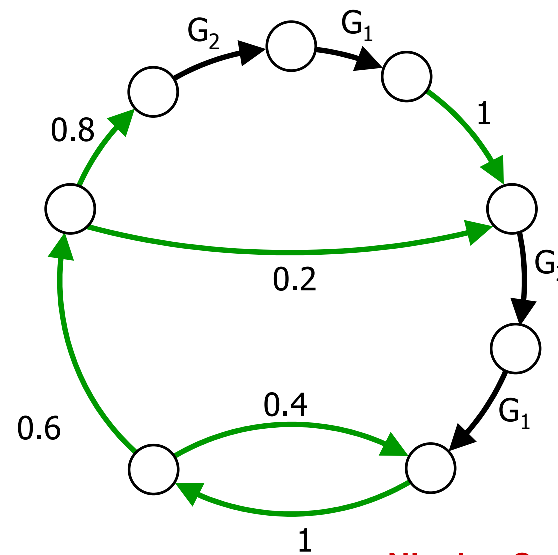
  latency: in the MC = time between two sets of states α and ω

$\alpha = \{0,1\}$

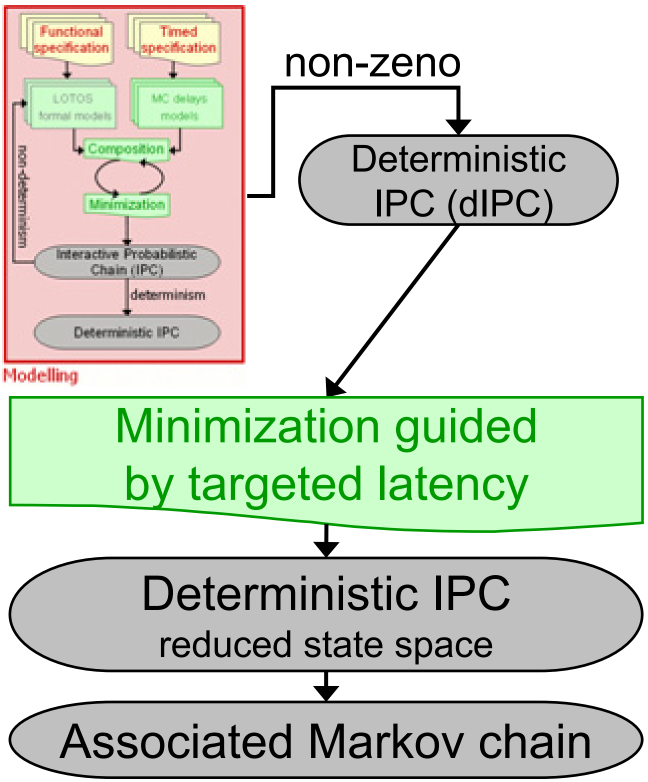
$\omega = \{0,1\}$



Latency between G_1 and G_2 ?

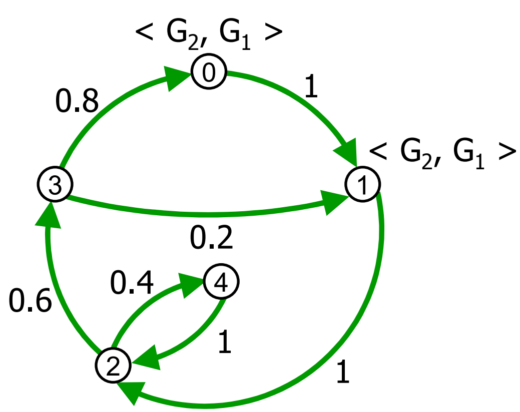


Performance Flow



$\alpha = \{0,1\}$

$\omega = \{0,1\}$



- Targeted result: latency distribution

  latency: in a dIPC = time between two interactive transitions

- Study of the associated MC

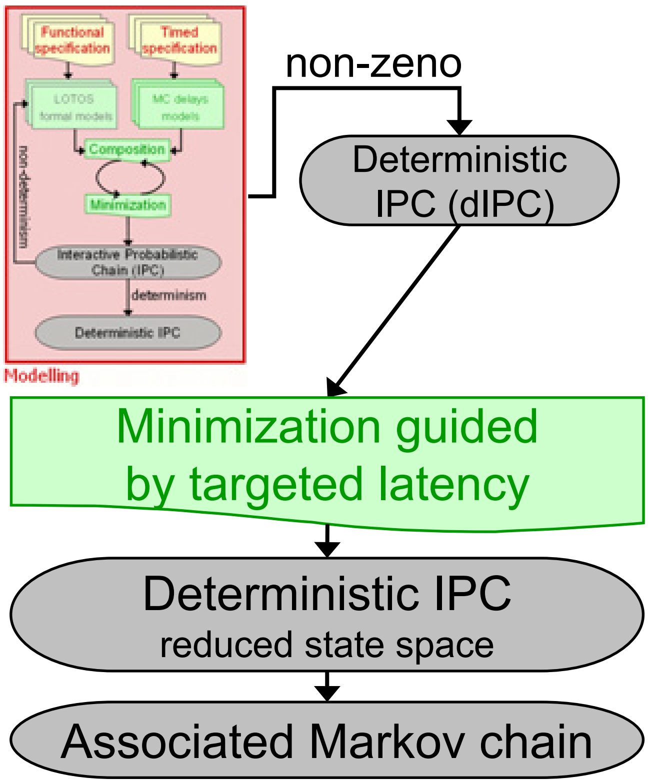
  latency: in the MC = time between two sets of states α and ω

- Property:

  2 branching equivalent dIPCs ⇔ 2 strongly equivalent MCs

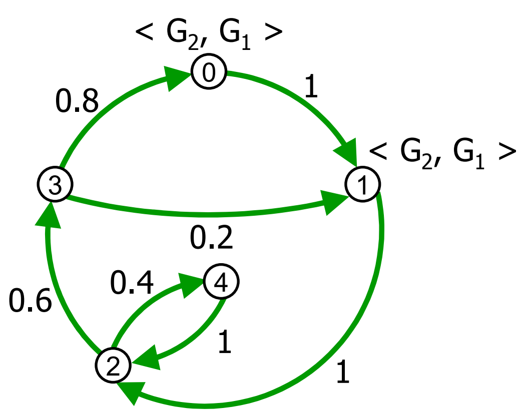
  - Markovian properties preserved along minimizations
  - Extracted performance results preserved

Performance Flow



$\alpha = \{0,1\}$

$\omega = \{0,1\}$



- Targeted result: latency distribution

  latency: in a dIPC = time between two interactive transitions

- Study of the associated MC

  latency: in the MC = time between two sets of states α and ω

  latency → random variable at t_0: $L_{t_0}(\alpha,\omega)$

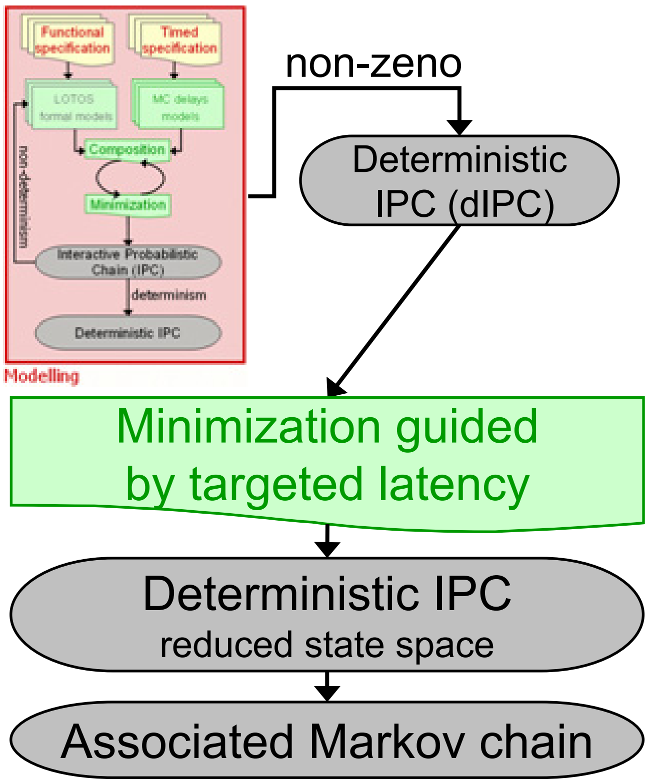
$$L_{t_0}(\alpha,\omega) = \min\{t \mid t > 0 \ \wedge \ X_{t_0+t} \in \omega\}$$

if $X_{t_0} \in \alpha$ and 0 otherwise.

Long-run average latency (Cesàro limit):

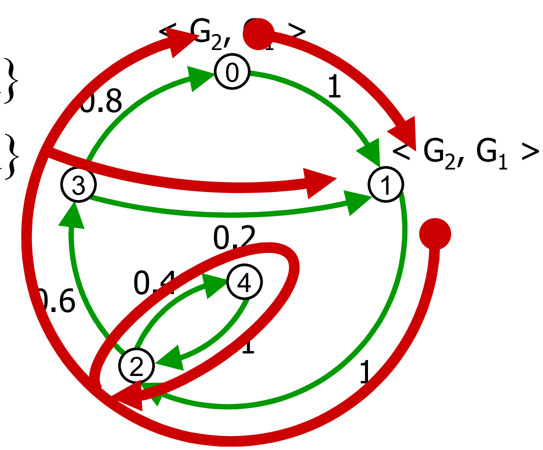
$$L(\alpha,\omega) = \lim_{t\to\infty} \frac{1}{t} \sum_{t_0=0}^{t} L_{t_0}(\alpha,\omega)$$

Performance Flow



$\alpha = \{0,1\}$

$\omega = \{0,1\}$



- **Targeted result: latency distribution**

  latency: in a dIPC = time between two interactive transitions

- **Study of the associated MC**

  latency: in the MC = time between two sets of states α and ω

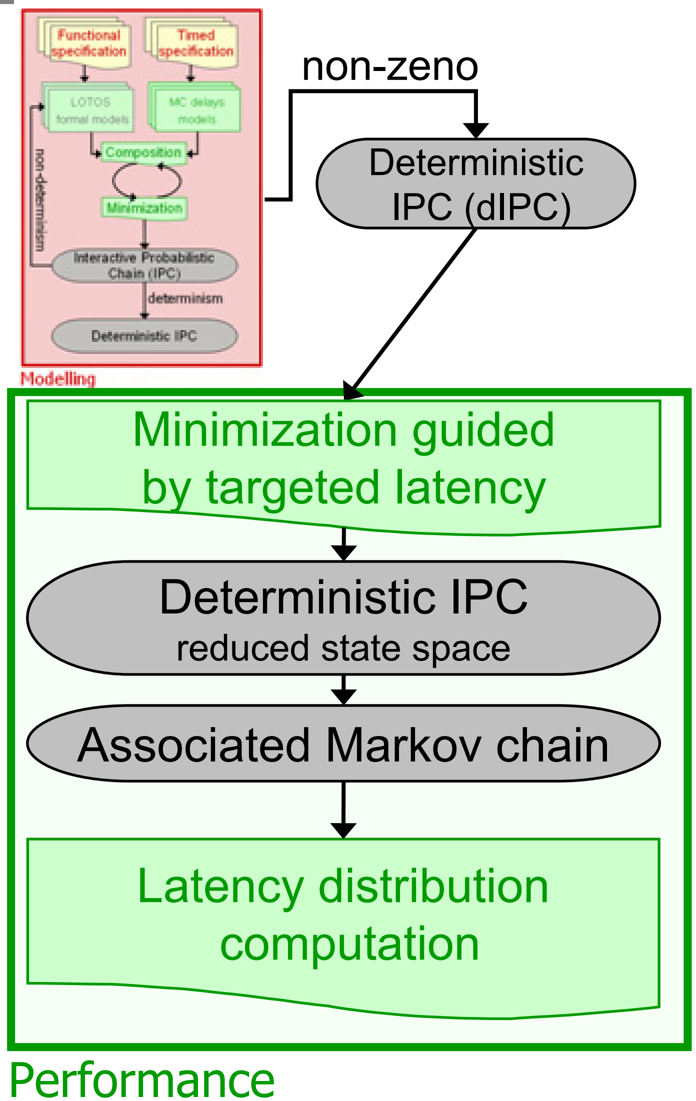
  Long-run average latency: $L(\alpha,\omega)$

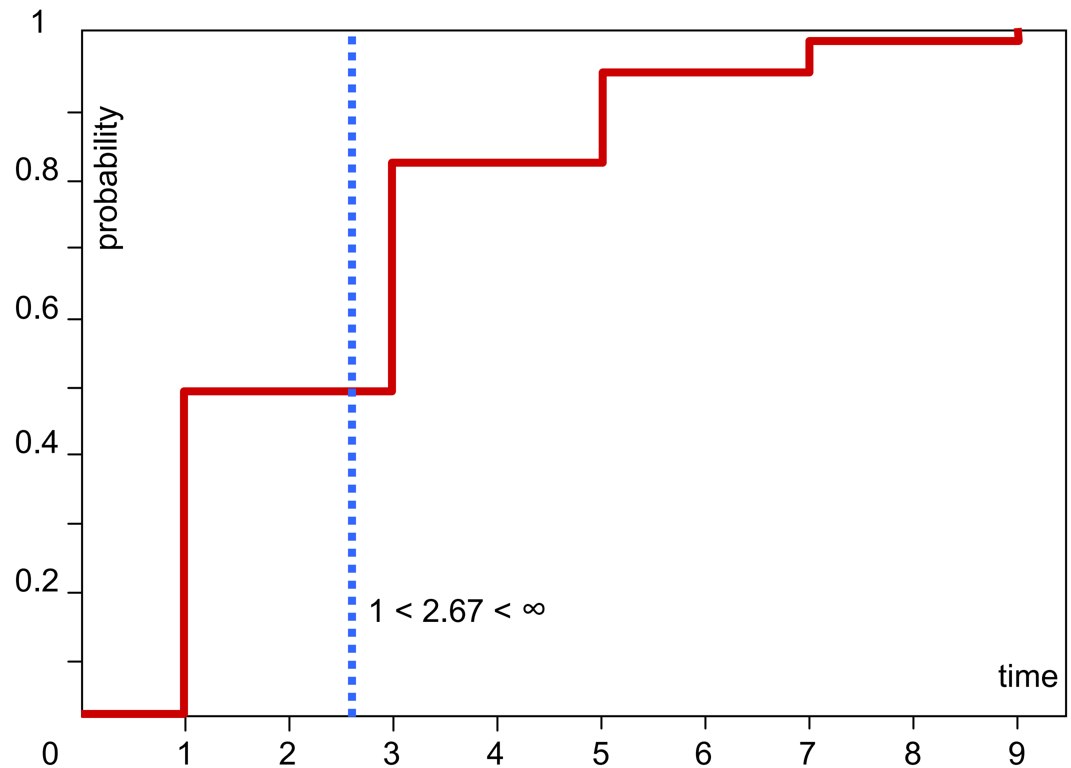
$$\Pr\big(L(\alpha,\omega)=t\big) = \sum_{c_b\in\alpha} \frac{\pi(c_b)}{\sum_{c\in\alpha}\pi(c)} \underbrace{\Pr\big(L(\{c_b\},\omega)=t\big)}_{\text{Distribution of the latency starting in a particular state of } \alpha}$$

Chance of being in c_b on long run

Distribution of the latency starting in a particular state of α

Performance Flow

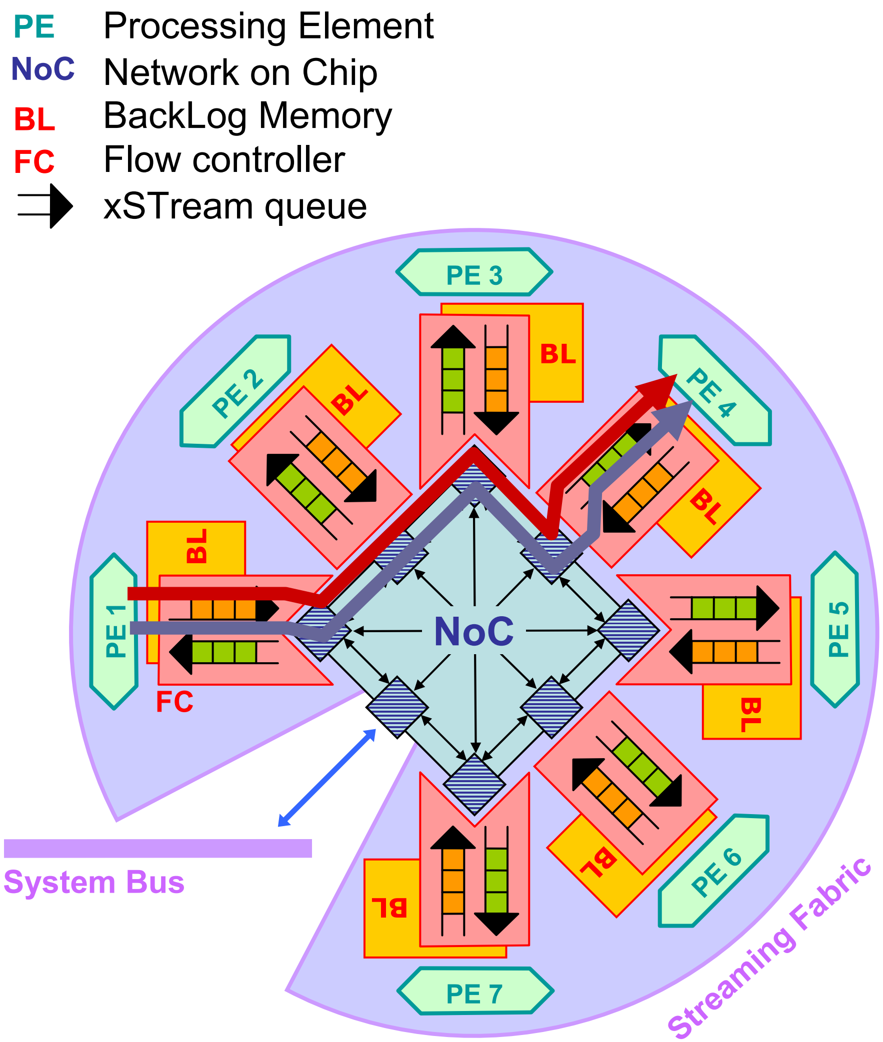




Talk outline

- Introduction
- Modelling Flow
- Performance Flow
- Case-Study: the *xSTream* Architecture
- Conclusion

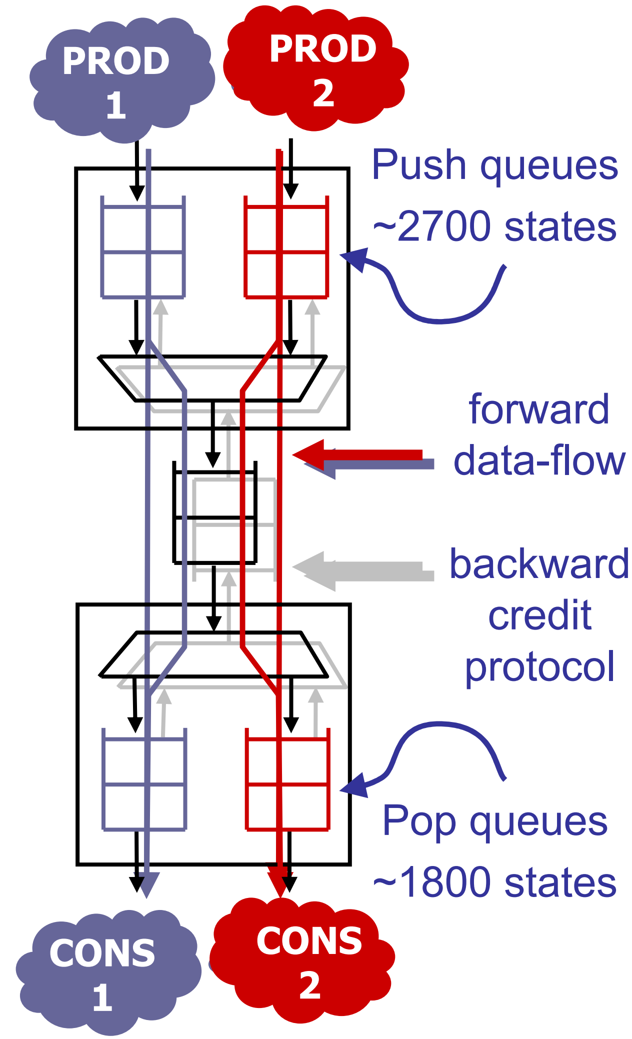
The xSTream Case-Study



- Two flows from PE 1 to another PE, say PE 4.
- Time to remove an element from a Pop queue ?
  - Available elements
    - → Pop immediate
  - No element
    - → Pop delayed until next element arrives
- A Pop latency close to its minimal value means the communication architecture absorbs production/consumption bursts

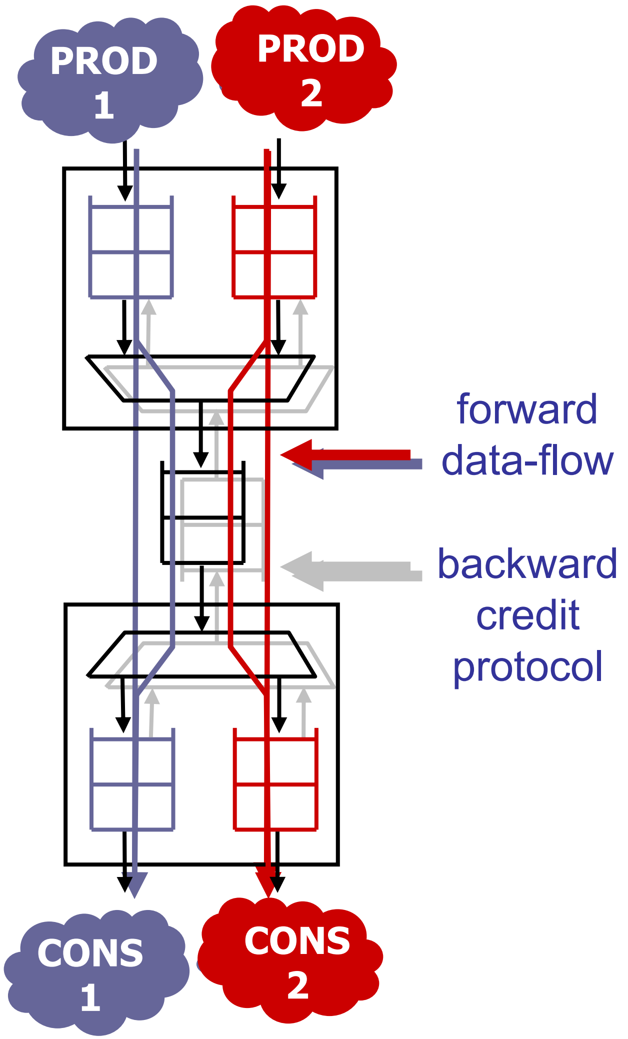
The xSTream Case-Study : functional model

- 2 applications (producer/consumer pairs) sharing the NoC and flow controllers
- Producer inserts elements in a Push queue
- Consumer gets elements from a Pop queue
- A multiplexer is used to access the NoC. Indeed, the 2 used Push queues are in the same flow controller
- A demultiplexer is used to get data from NoC and send them to the right Pop queue
- The NoC is abstracted by a buffer : the 2 data-flows are sharing the same virtual channel on the NoC
- Functional verification showed that the credit protocol is mandatory (possible deadlocks)



The xSTream Case-Study : timed model

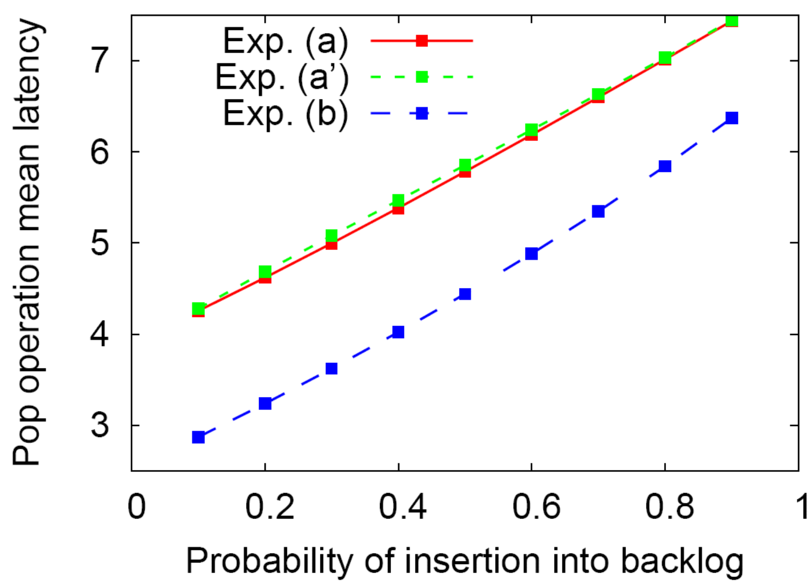
- Insertion of 14 delays
- Backlog mechanism abstracted by probabilistic delays
- Study of 3 different configurations of the credit protocol: exp. (a), (a') and (b)
  - credit protocol used in its worst configuration prod./cons. rates greater in exp. (a') than in exp. (a)
  - better configuration of credit protocol for exp. (b) than for experiments (a) and (a')

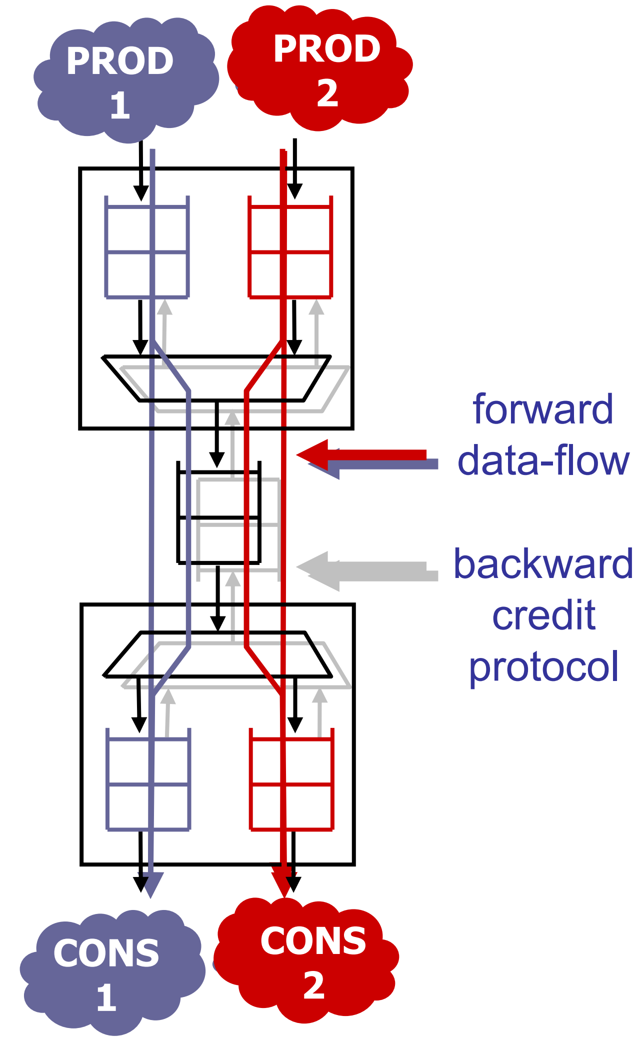


The xSTream Case-Study : timed model

	Exp.	(a)	(a’)	(b)
IPC size	States	9.2M	20.3M	19.7M
	trans.	39.7M	82.8M	92.0M
Associated MC size	states	207k	380k	235k
	trans.	822k	1487k	1186k

- Study of 3 different configurations of the credit protocol: exp. (a), (a’) and (b)
  - credit protocol used in its worst configuration prod./cons. rates greater in exp. (a’) than in exp. (a)
  - better configuration of credit protocol for exp. (b) than for experiments (a) and (a’)





Talk outline

- Introduction
- Modelling Flow
- Performance Flow
- Case-Study: the *xSTream* Architecture
- Conclusion

Conclusion

- Methodology for functional verification and performance evaluation on the same model (IPC)

- Definition of latency and how to compute its distribution for an IPC
  - Translation of deterministic IPC in Markov Chain
  - Computation of latency distribution in a Markov chain
  - We did not give solutions for non-deterministic systems

- Results computed are distributions
  - Minimum, maximum and average values are thus easily available
  - Possibility to have probabilistic results (ex: Pr [latency < k])

Conclusion

Thank you for your attention !

Questions ?