A stochastic hybrid process algebra

Vashti Galpin, Jane Hillston (LFCS, School of Informatics, University of Edinburgh)
Luca Bortolussi (Department of Mathematics and Computer Science, University of Trieste)

Vashti.Galpin@ed.ac.uk,Jane.Hillston@ed.ac.uk,luca@dmi.units.it

HYPE is a hybrid process algebra [1].

- It models discrete and continuous behaviour.
- It allows for very compositional modelling.
- Events are discrete.
- Event conditions are urgent or non-urgent.
  - An urgent condition is a boolean formula and the event must occur when the condition becomes true
  - A non-urgent condition is $\bot$ and the event happens at some unspecified point in the future.
- A structured operational semantics defines a labelled transition system.
- The transition system of a model is interpreted as a hybrid automaton.

Stochastic HYPE includes stochastic behaviour as well.

- Events are discrete or stochastic.
- All discrete events are urgent.
- Stochastic events are associated with an exponential distribution.
- Transition systems are interpreted as Transition Driven Stochastic Hybrid Automata [2], a subset of Piecewise Deterministic Markov Processes [3].

A *stochastic HYPE model* is a tuple $(ConSys, \mathcal{V}, \mathcal{X}, IN, IT, \mathcal{E}, \mathcal{A}, ec, iv, EC, ID)$ where

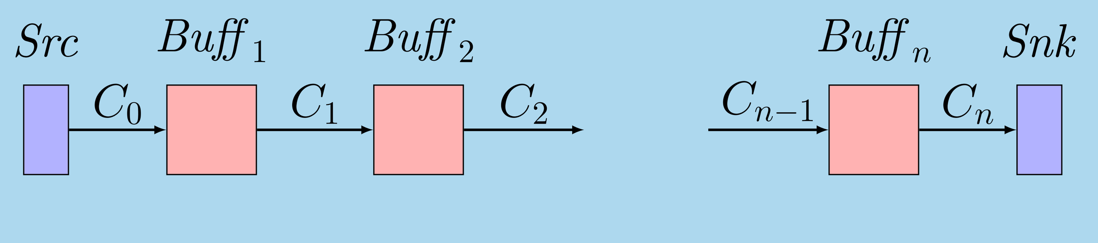
- $ConSys$, controlled system as illustrated below
- $\mathcal{V}$, variables; $\mathcal{X}$ formal variables (both finite sets)
- IN, influence names; IT, influence type names; ID, influence type definitions, $[\![I(\overrightarrow{X})]\!] = f(\overrightarrow{X})$
- $\mathcal{E}$, events; $\underline{a}$, discrete; $\overline{a}$ stochastic
- $\mathcal{A}$, activities, $\alpha(\overrightarrow{X}) = (\iota, r, I(\overrightarrow{X})) \in (IN \times \mathbb{R} \times IT)$
- $EC \subseteq ActivationConditions \times Resets$, event conditions; $ec(\underline{a}) = (\phi, \psi)$; $ec(\overline{a}) = (r, \psi)$
- $ec : \mathcal{E} \to EC$, maps events to event conditions
- $iv : IN \to \mathcal{V}$, maps influence names to variables

Delay tolerant networks (DTNs)

- have intermittent connectivity between nodes,
- packets (bundles) cannot always be forwarded, and
- nodes require additional buffer space for storage.

In our model of a DTN

- we consider a simple model as illustrated,
- incoming packets are dropped when a buffer is full,
- we can experiment with buffer size requirements.



A simple delay-tolerant network

A buffer component

- is the basic network element,
- has an input and out subcomponent and
- has an associated variable B.

The input subcomponent is defined by

$$\begin{aligned} I_{C,B} \stackrel{def}{=} {} & \overline{\mathrm{on}}_C{:}(i_B, s_C, c).I_{C,B} + \overline{\mathrm{off}}_C{:}(i_B, 0, c).I_{C,B} + \\ & \underline{\mathrm{full}}_B{:}(i_B, 0, c).I_{C,B} + \underline{\mathrm{init}}_B{:}(i_B, s_C, c).I_{C,B} + \\ & \underline{\mathrm{nf\text{-}on}}_B{:}(i_B, s_C, c).I_{C,B} + \underline{\mathrm{nf\text{-}off}}{:}(i_B, 0, c).I_{C,B} \end{aligned}$$

- Events $\overline{\mathrm{on}}_C$ and $\overline{\mathrm{off}}_C$ occur stochastically and describe the status of input connection C, giving conditions $ec(\overline{\mathrm{on}}_C) = (r\_on_C, true)$ and $ec(\overline{\mathrm{on}}_C) = (r\_off_C, true)$.
- The event $\underline{\mathrm{full}}_B$ has condition $ec(\underline{\mathrm{full}}_C) = (B = \max_B, true)$ which describes when the buffer is full by a check on the value of the variable B.
- The event $\underline{\mathrm{init}}$ occurs once, immediately at the start.
- Events $\underline{\mathrm{nf\text{-}on}}_B$ and $\underline{\mathrm{nf\text{-}off}}_B$ capture when the buffer stops being full and have the same condition $(B < \max_B, true)$.
- All resets are *true* meaning that no variable changes value on an event.
- Events are followed by activities; (i_B, s_C, c) or $(i_B, 0, c)$
  - i_B: activity/influence name.
  - s_C: influence of arrival of packets on the variable B
  - c: influence is to be treated in a constant fashion.

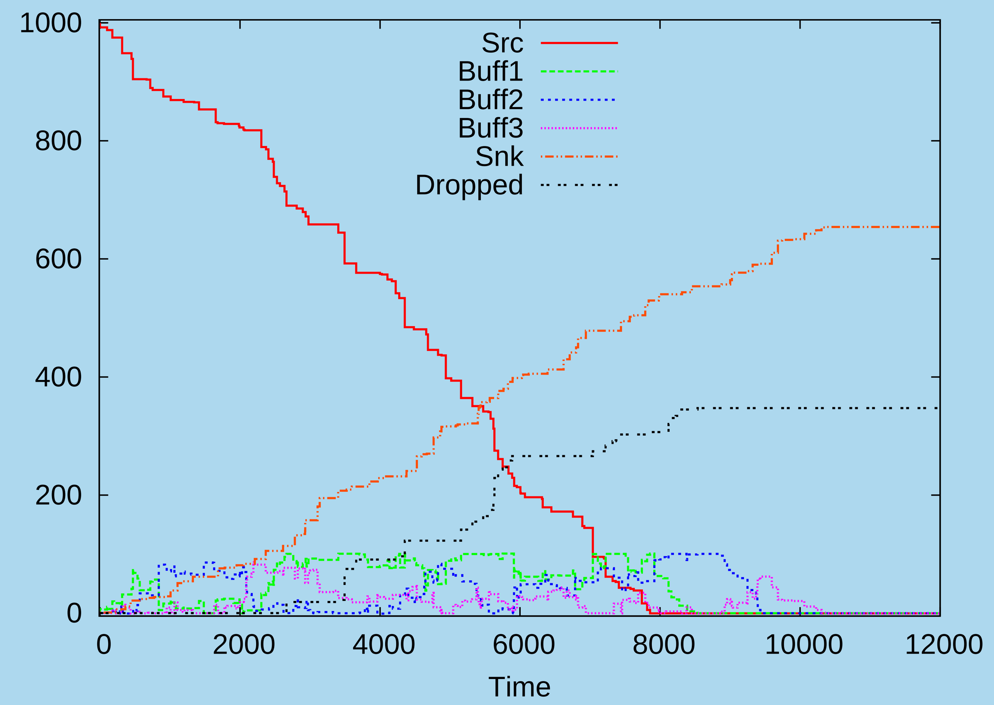
The output subcomponent is similarly defined by

$$\begin{aligned} O_{B,C} \stackrel{def}{=} {} & \overline{\mathrm{on}}_C{:}(o_B, -t_C, c).O_{B,C} + \overline{\mathrm{off}}_C{:}(o_B, 0, c).O_{B,C} + \\ & \underline{\mathrm{empty}}_B{:}(o_B, 0, c).O_{B,C} + \underline{\mathrm{init}}_B{:}(o_B, -t_C, c).O_{B,C} + \\ & \underline{\mathrm{ne\text{-}on}}_B{:}(o_B, -t_C, c).O_{B,C} + \underline{\mathrm{ne\text{-}off}}{:}(o_B, 0, c).O_{B,C} \end{aligned}$$

- C is the outgoing connection.
- The influence o_B describes the effect of the output of packets on the variable B.

These can be composed to give the buffer component

$$Buff_{C,B,C'} \stackrel{def}{=} I_{C,B} \underset{\underline{\mathrm{init}}}{\bowtie} O_{B,C'}$$



Network with buffer size 100

The controlled system

- Events in $Buff_{C,B,C'}$ are not constrained.
- Controller components schedule events.
- Controller components have events but no activities.

We define a input controller (output is similar).

$$\begin{aligned} ConU0_{C,B} & \stackrel{def}{=} \overline{\mathrm{on}}_C.ConU1_{C,B} \\ ConU1_{C,B} & \stackrel{def}{=} \overline{\mathrm{off}}_C.ConU0_{C,B} + \underline{\mathrm{full}}_B.ConU2_{C,B} \\ ConU2_{C,B} & \stackrel{def}{=} \overline{\mathrm{off}}_C.ConU3_{C,B} + \underline{\mathrm{nf\text{-}on}}_B.ConU1_{C,B} \\ ConU3_{C,B} & \stackrel{def}{=} \overline{\mathrm{on}}_C.\underline{\mathrm{full}}_B.ConU2_{C,B} + \underline{\mathrm{nf\text{-}off}}_B.ConU0_{C,B} \end{aligned}$$

The buffer controller is defined as

$$Con_{C,B,C'} \stackrel{def}{=} ConU1_{C,B} \underset{\emptyset}{\bowtie} ConT1_{B,C'}$$

The controlled system (where M contains all events) is

$$System \stackrel{def}{=} Buff_{C,B,C'} \underset{M}{\bowtie} \underline{\mathrm{init}}.Con_{C,B,C'}$$

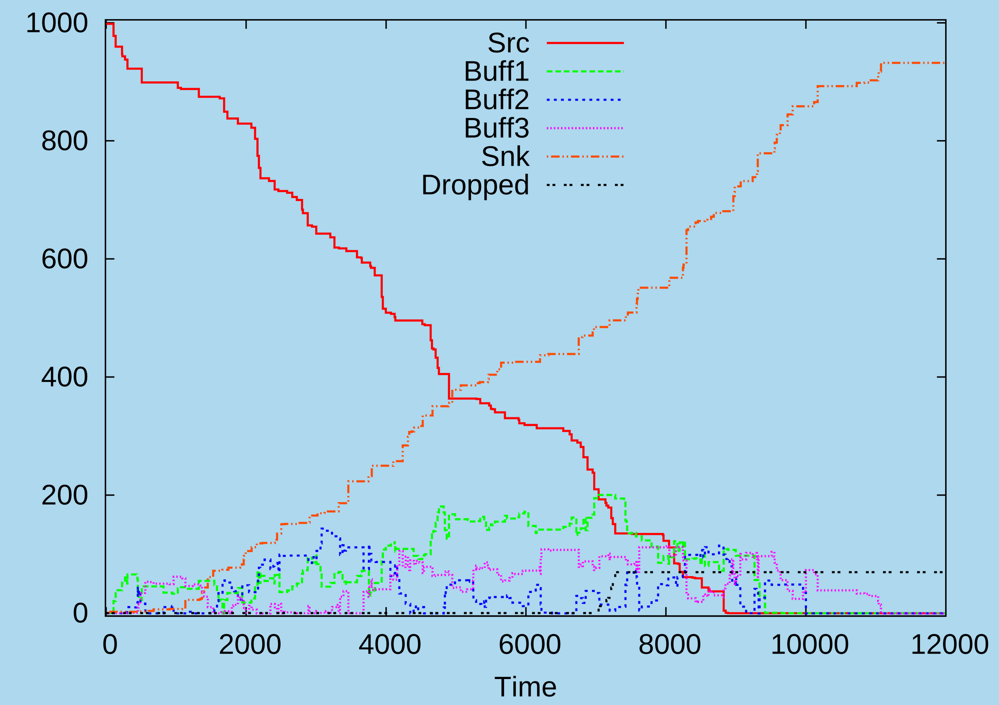
Semantics

- Structured operational semantics generates a labelled transition system which forms the basis of the Transition Driven Stochastic Hybrid Automaton.
- Ordinary differential equations (ODEs) are obtained from each state σ of the transition system.
- $iv(i_B) = iv(o_B) = B$.

$$dB/dt = \sum\{ s \times [\![I(\overrightarrow{X})]\!] \mid iv(\iota) = B, \sigma(\iota) = (s, I(\overrightarrow{X}))\}$$

In the state where

- both connections are available and the buffer is not full then, $dB/dt = s_C - t_{C'}$;
- only C is available and the buffer is not full then $dB/dt = s_C$; and
- only C is available and the buffer is full $dB/dt = 0$.



Network with buffer size 200

A simple network model

- Packets start at a Src component.
- Packets end at a Snk component if not dropped.
- There are n buffers as illustrated.

The network model is

$$\begin{aligned} & (Src \underset{L_0}{\bowtie} (Buff_{C_0,B_1,C_1} \underset{L_1}{\bowtie} \ldots (Buff_{C_{n-1},B_n,C_n} \underset{L_n}{\bowtie} Snk)\ldots) \\ & \underset{M}{\bowtie} \underline{\mathrm{init}}.(Con_{C_0,B_1,C_1} \underset{L'_1}{\bowtie} \ldots \underset{L'_{n-1}}{\bowtie} Con_{C_{n-1},B_n,C_n}) \end{aligned}$$

where M is as before, $L'_i = \{\overline{\mathrm{on}}_{C_i}, \overline{\mathrm{off}}_{C_i}\}$ and $L_i = L'_i \cup \{\underline{\mathrm{init}}\}$.

- Each graph shows a single simulation.
- At the start there are 1000 packets at Src.
- The network has three buffers.
- The two simulations consider different buffer sizes.
- More packets are lost in the first simulation where buffer size is smaller
- This model requires further investigation and experimentation.
- More complex network topologies will be modelled.

References

[1] V. Galpin, L. Bortolussi, and J. Hillston. HYPE: a process algebra for compositional flows and emergent behaviour. In *Proceedings of CONCUR 2009*.

[2] L. Bortolussi and A. Policriti. Hybrid semantics of stochastic programs with dynamic reconfiguration. In *Proceedings of CompMod 2009*.

[3] M.H.A. Davis. *Markov Models and Optimization*. 1993.

[4] K. Fall and S. Farrell. DTN: An architectural retrospective. *IEEE Journal on Selected Areas in Communications*, 26, 2008.