Advances in Hybrid Systems Modeling and Simulation

Application to Network Data Flows and Particle Tracking

Prof. Rodrigo Castro
University of Buenos Aires and ICC-CONICET
Argentina

CERN – ATLAS Experiment
Trigger and Data Acquisition Team
May 7, 2018, Geneva, Switzerland

Agenda

- **Core technologies for Hybrid Systems M&S**
  - Formal treatment of Hybrid Systems
  - Interaction between Continuous and Discrete dynamics
- **Advances conCERNing relevant applications**
  - Networks of data flows
    - Discrete models for the High Level Trigger in ATLAS TDAQ
    - Fluid Flow approximations
    - Hybrid models
  - Tracking of particles motion in 3D geometries
    - Co-simulation using hybrid numerical integration
    - Speedups synthetic geometries
    - The CMS case study
- **Q&A**

Core Technologies

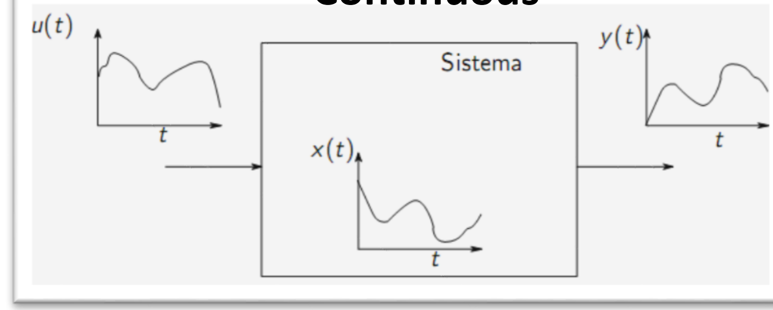
Formal treatment of Hybrid Systems

with the

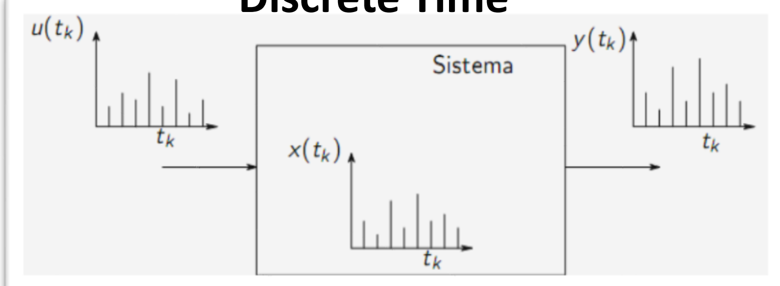
Discrete **EV**ent **S**ystems specification
(DEVS)

- Based on principles of the General Systems Theory
- **DEVS** *(Bernard Zeigler, '76, '90, 2000, 2018)* allows to:
  - **Represent exactly** any type of **discrete system**
  - **Approximate continuous systems** at any desired degree of accuracy

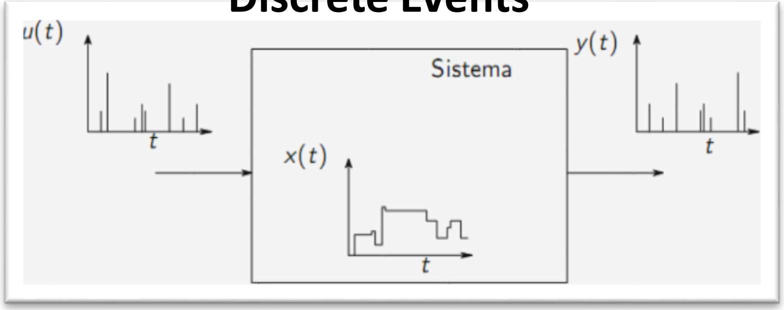
Continuous



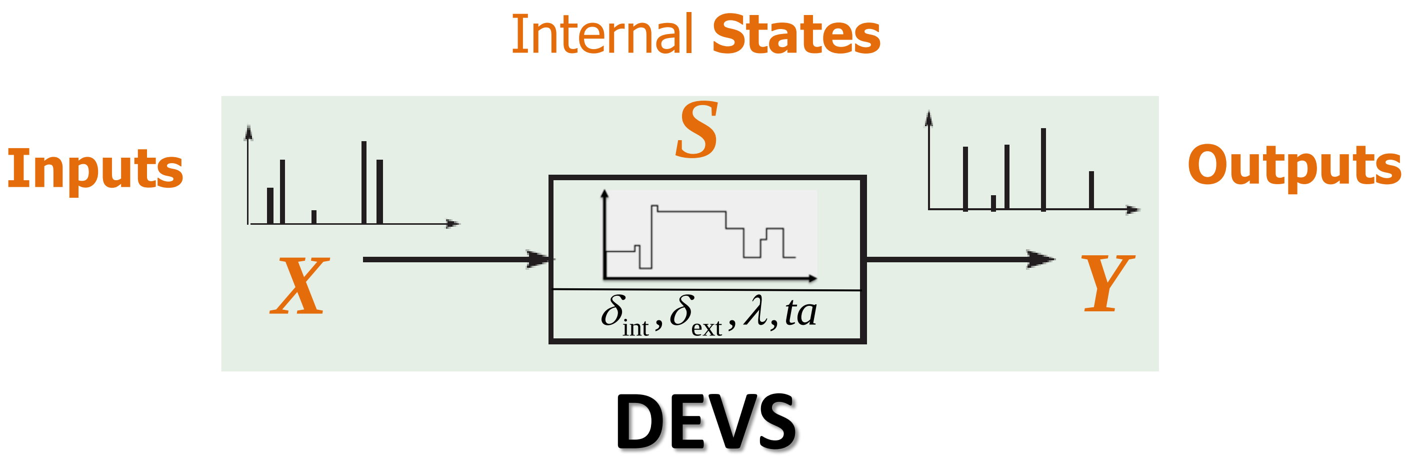
Discrete Time



Discrete Events



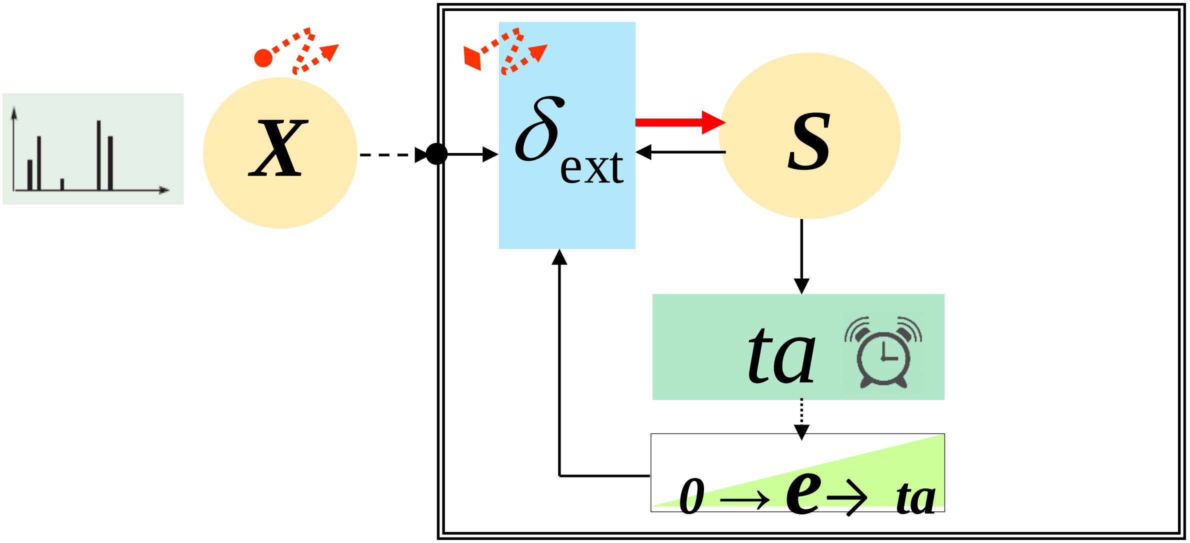
- DEVS Atomic Model: Basic unit of behavior



- A DEVS model is defined by the following mathematical tuple:

$$M_D = (\underbrace{X, Y, S}_{\text{Sets}}, \underbrace{\delta_{int}, \delta_{ext}, \lambda, ta}_{\text{Dynamical Functions}})$$

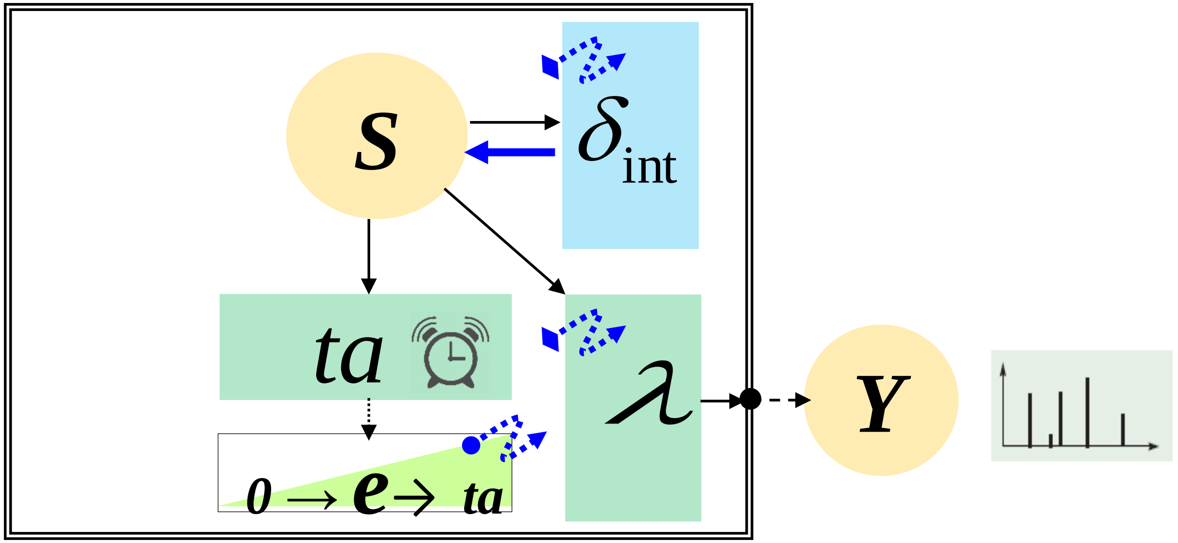
External dynamics



External transition: A message comes from other models

$$M_D = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta)$$

Sets — Dynamical Functions



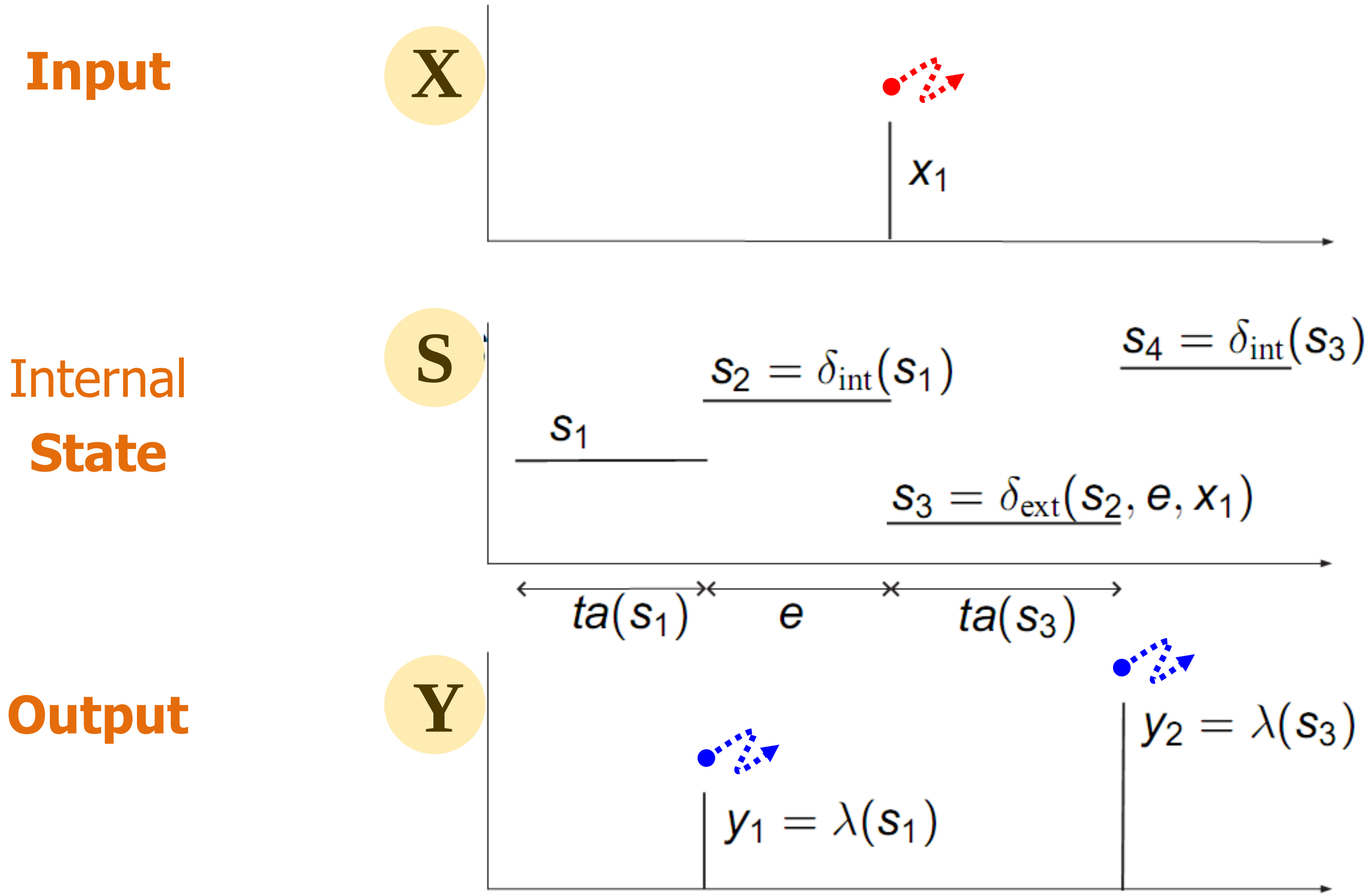
Internal Transition: Current state expired (independent from the *external system*)

$$M_D = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta)$$

Sets — Dynamical Functions

- Example

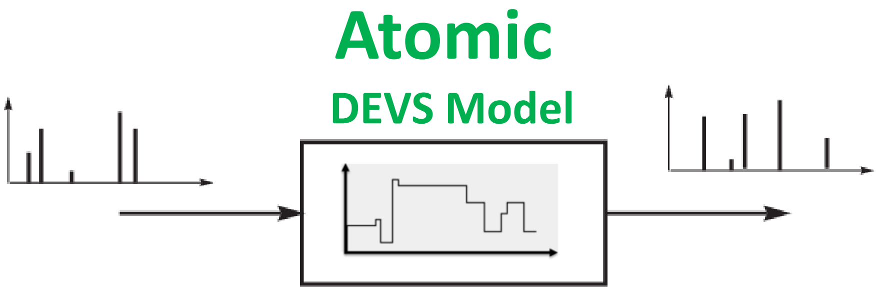
$$M_D = (X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta)$$



Properties

- A DEVS model:
  - **Processes** a **sequence of input events**
  - According to its **internal State Changes**
  - **Produces** a **sequence of output events**
  - Using a **continuous time base**

- Allows representing **any system** undergoing a **finite number of changes** within **finite time intervals** (Legitimacy property)

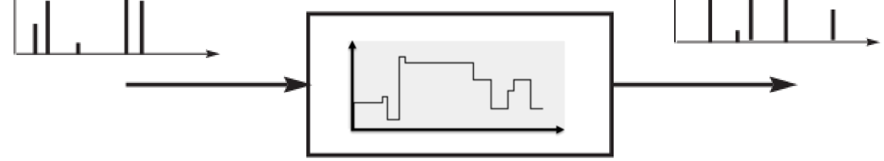


- A DEVS model:
  - **Processes** a **sequence of input events**
  - According to its **internal State Changes**
  - **Produces** a **sequence of output events**
  - Using a **continuous time base**

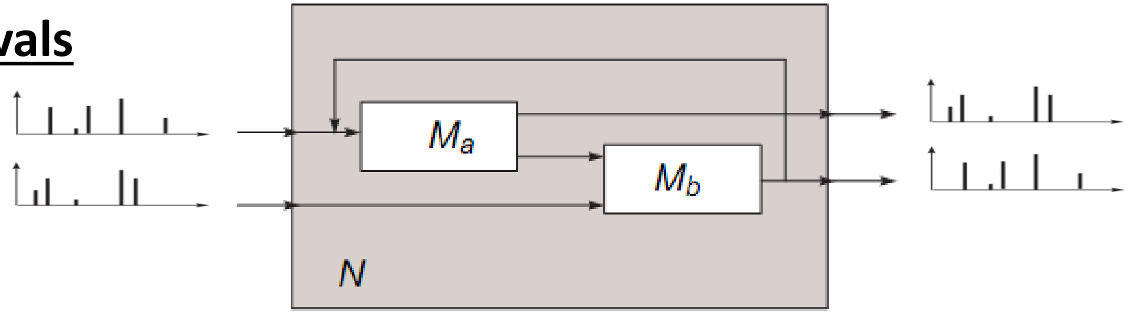
- Allows representing **any system** undergoing a **finite number of changes** within **finite time intervals** (Legitimacy property)

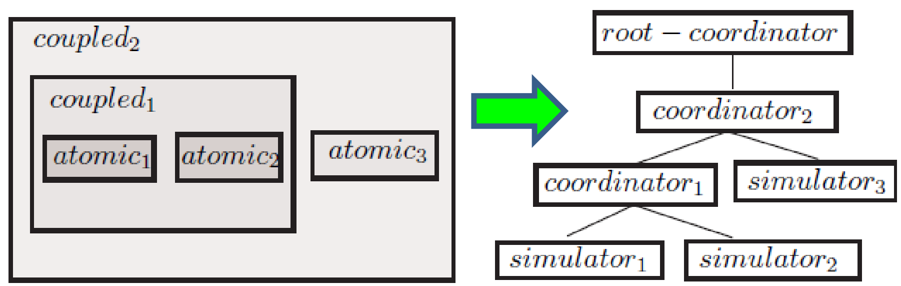
- DEVS models can be **coupled modularly and hierarchically** to build **complex systems**.
  All couplings produce new atomic models.
  (Closure under Coupling property)

Atomic
DEVS Model

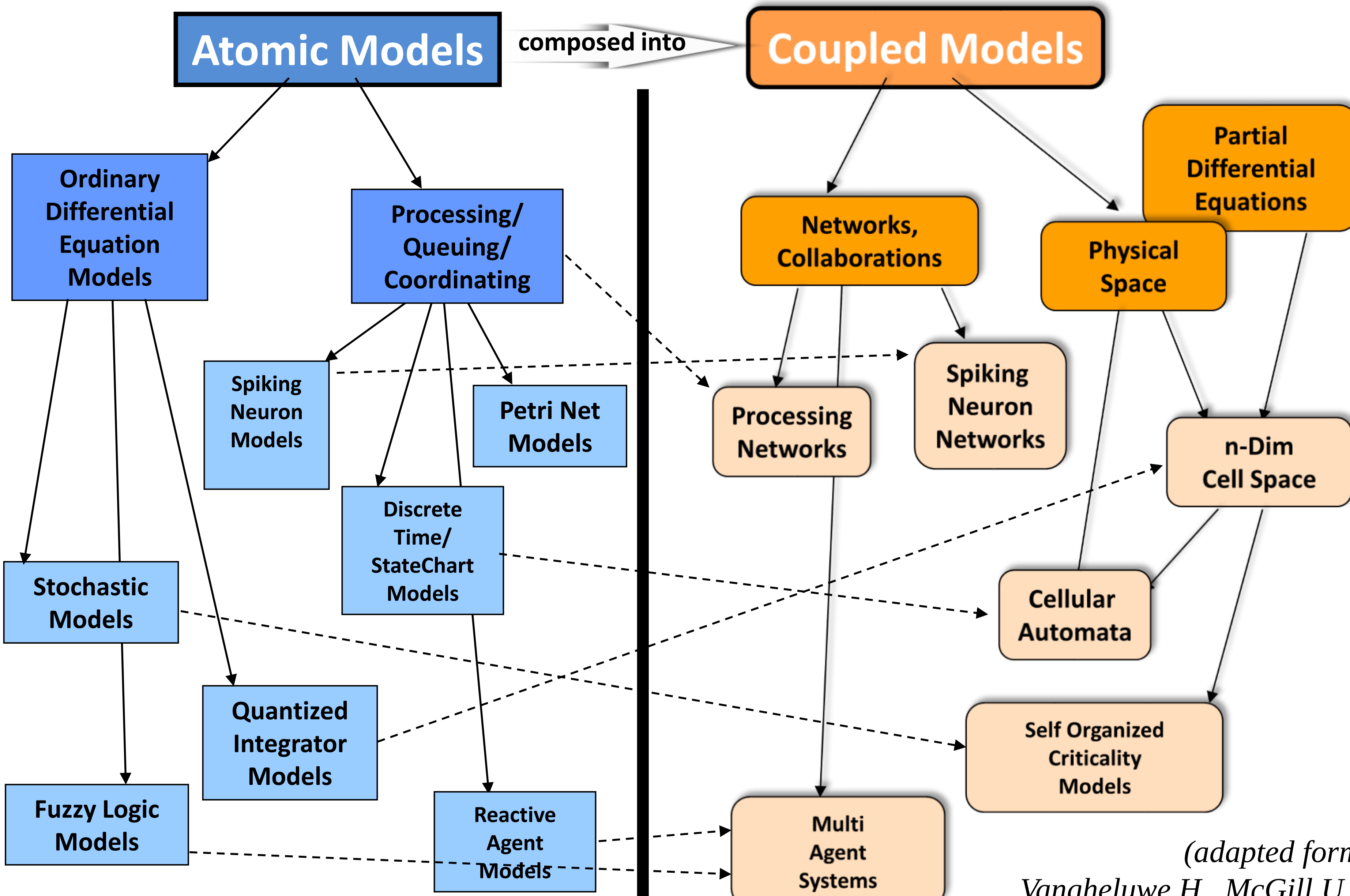


Coupled
DEVS Model





Universal DEVS Simulation Mechanism



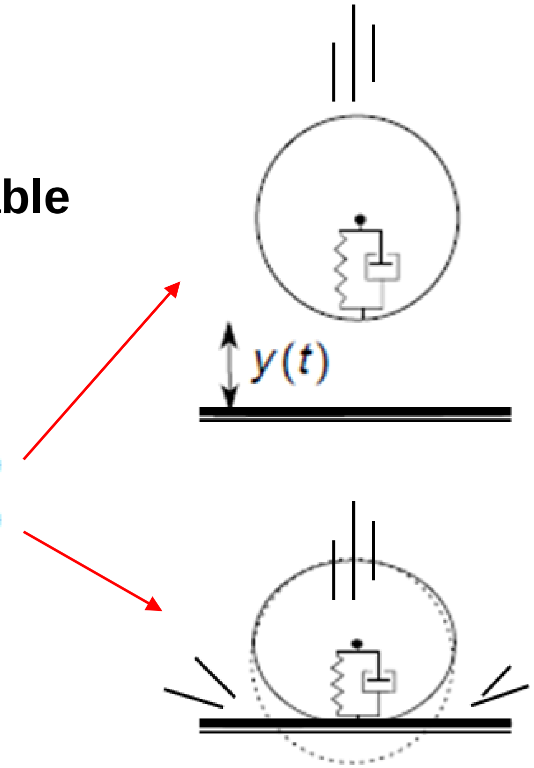
(adapted form Vangheluwe H., McGill U.)

Interaction between Continuous and Discrete dynamics with

Quantized **S**tate **S**ystems (QSS)

- Discontinuities create hybrid dynamics
  - A system experiments a **discrete discontinuity** on a **continuous State Variable**
  - Example: **Bouncing Ball**

$$\dot{y}(t) = v(t)$$
$$\dot{v}(t) = \begin{cases} -g & \textbf{if } \ y(t) > 0 \\ -g - \frac{k}{m} \cdot y(t) - \frac{b}{m} \cdot v(t) & \textbf{if } \ y(t) \leq 0 \end{cases}$$



  - Numerical methods can incur in **unacceptable** errors
  - Discontinuity events must be
    - **detected** *efficiently*
    - **handled** *properly*
  - Often computationally costly. What can be done ?

- Consider the following **second order ODE** system:

$$\begin{cases} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= -x_1(t) \end{cases}$$

- and the following approximation by **state quantization**:

$$\begin{cases} \dot{x}_1(t) &= \underline{\text{floor}}(x_2(t)) = q_2(t) \\ \dot{x}_2(t) &= -\underline{\text{floor}}(x_1(t)) = -q_1(t) \end{cases}$$

Original **state variables**

Quantized **state variables**

Equation:

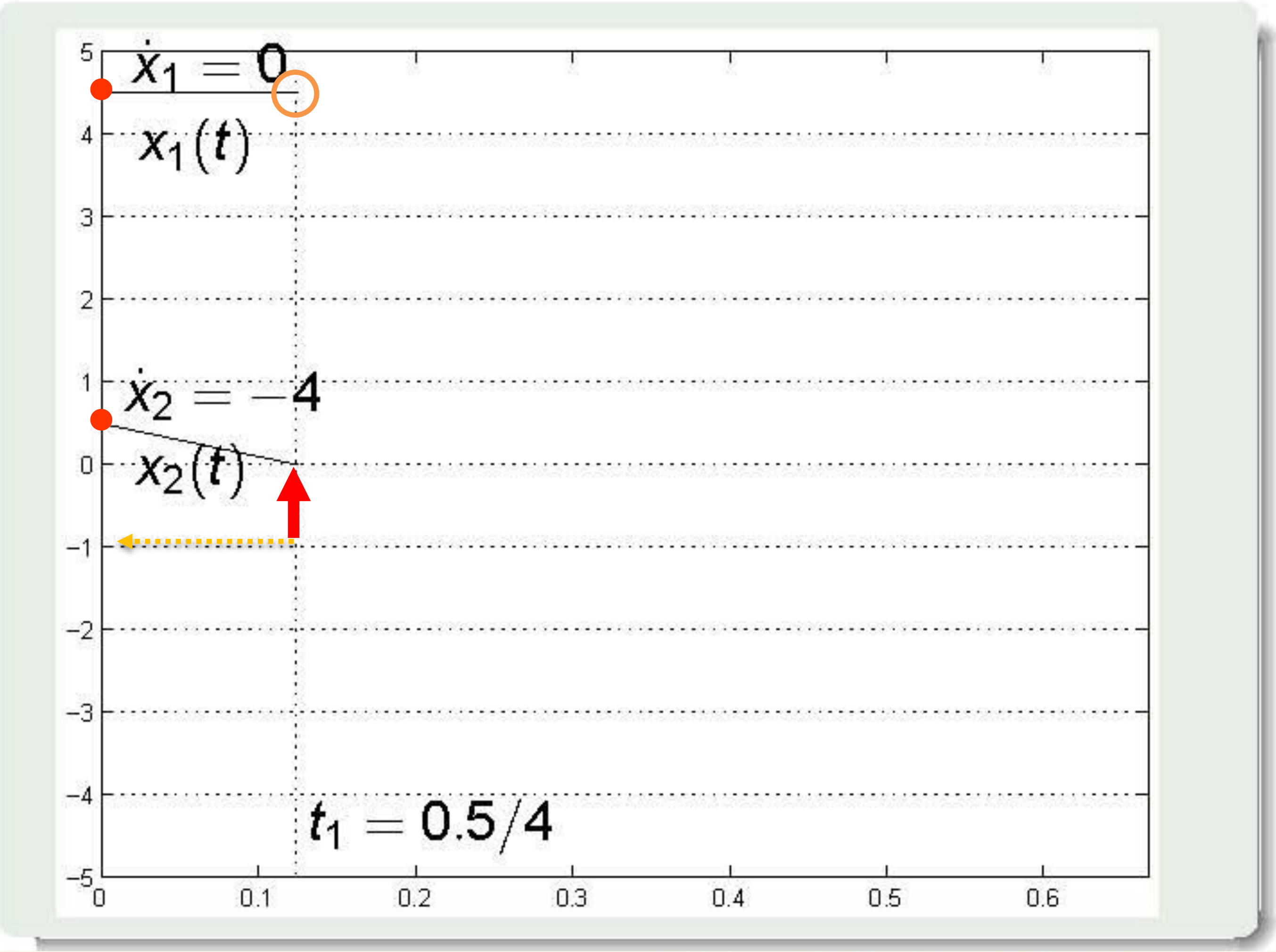
$$\begin{cases} \dot{x}_1(t) = q_2(t) \\ \dot{x}_2(t) = -q_1(t) \end{cases}$$

Can be solved.

Consider the I.C.:

$x_1(0) = 4.5,$

$x_2(0) = 0.5:$ •



↑ = ***Discrete Event:*** **Floor(·) changes.**

Equation:

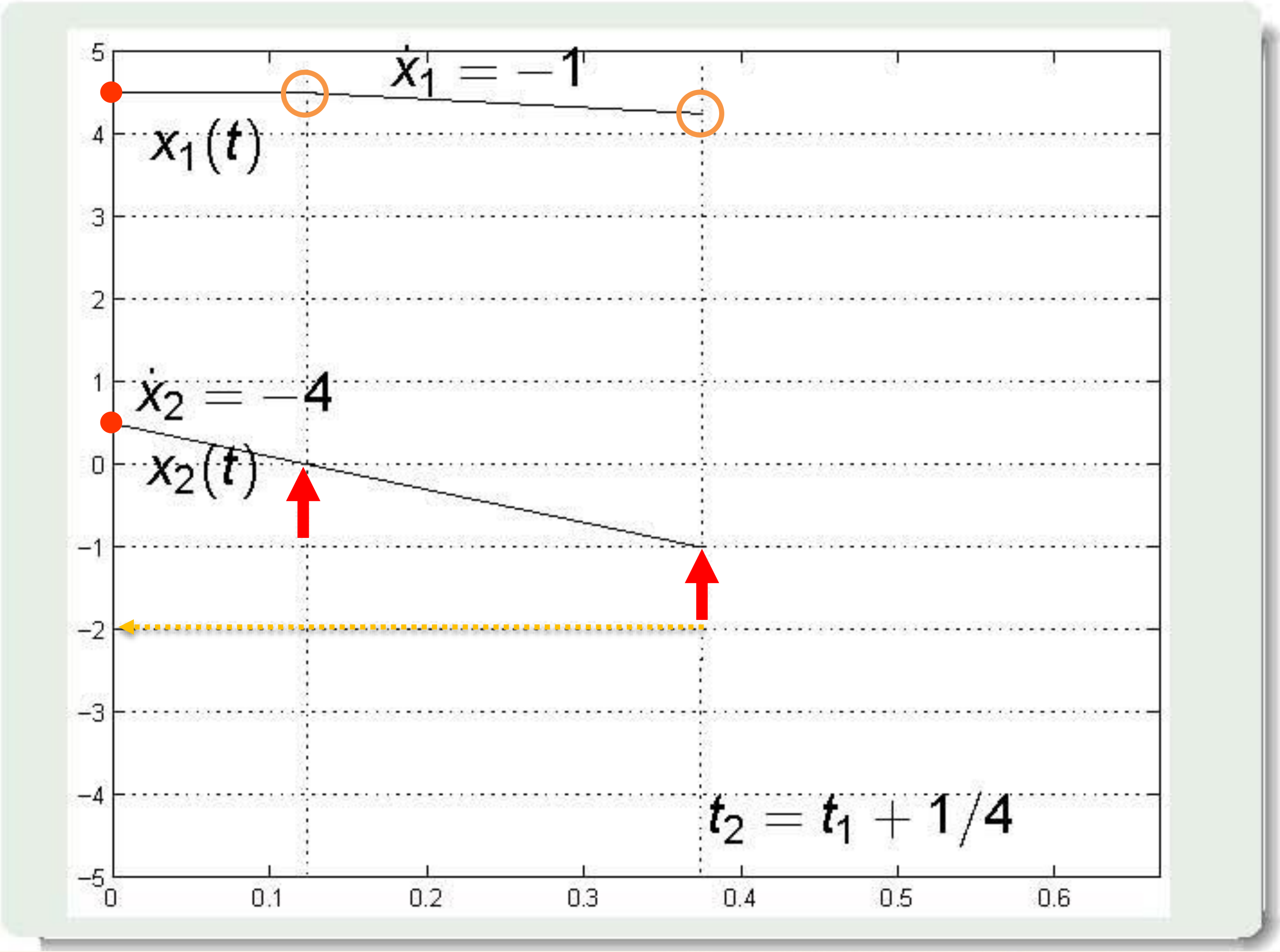
$$\begin{cases} \dot{x}_1(t) = q_2(t) \\ \dot{x}_2(t) = -q_1(t) \end{cases}$$

Can be solved.

Consider the I.C.:

$x_1(0) = 4.5,$
$x_2(0) = 0.5:$ ●

↑ = ***Discrete Event:***
Floor(·) changes.



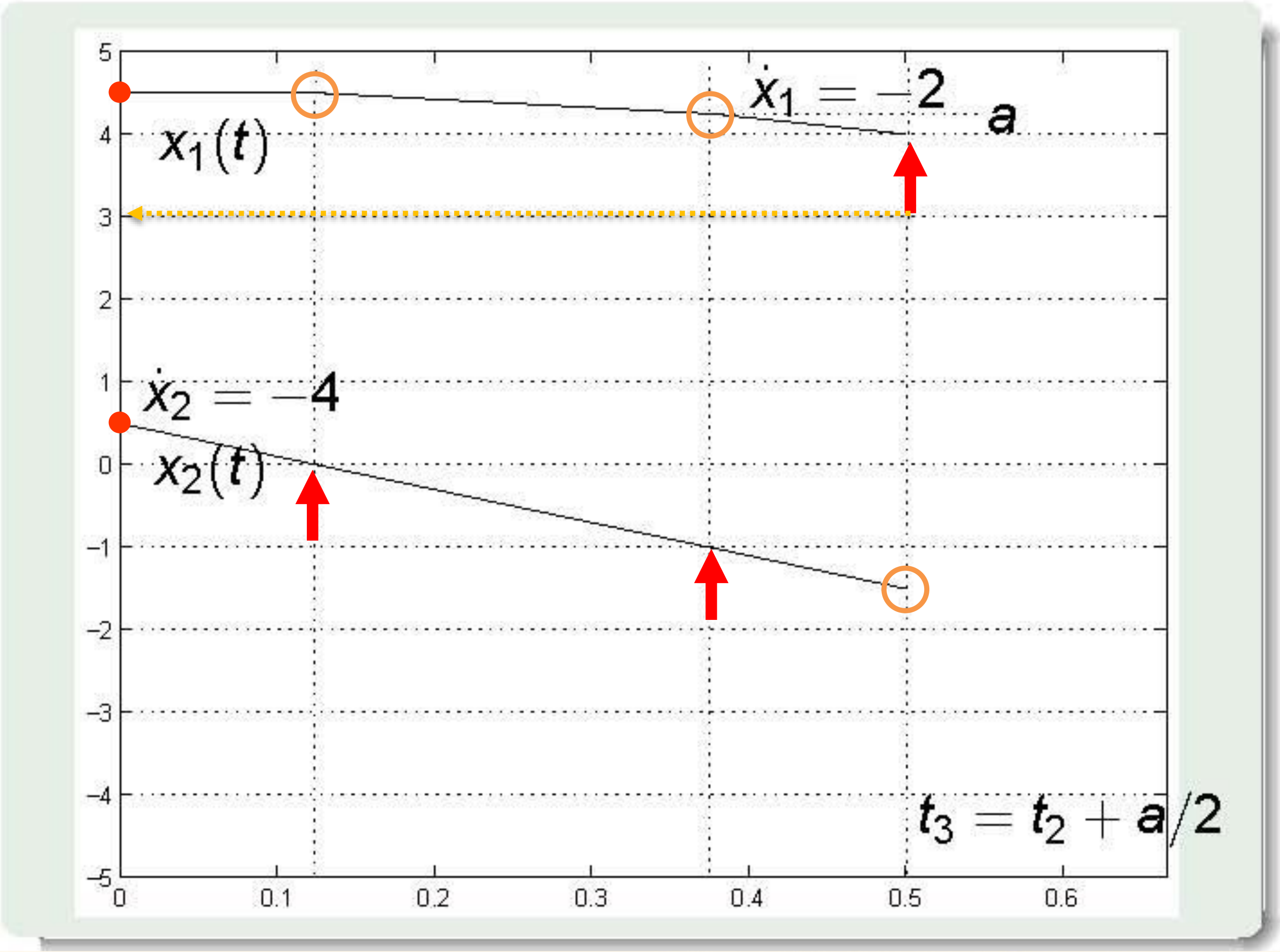
Equation:

$$\begin{cases} \dot{x}_1(t) = q_2(t) \\ \dot{x}_2(t) = -q_1(t) \end{cases}$$

Can be solved.

Consider the I.C.:
$x_1(0) = 4.5,$
$x_2(0) = 0.5:$ •

↑ = ***Discrete Event:***
Floor(·) changes.



Equation:

$$\begin{cases} \dot{x}_1(t) = q_2(t) \\ \dot{x}_2(t) = -q_1(t) \end{cases}$$

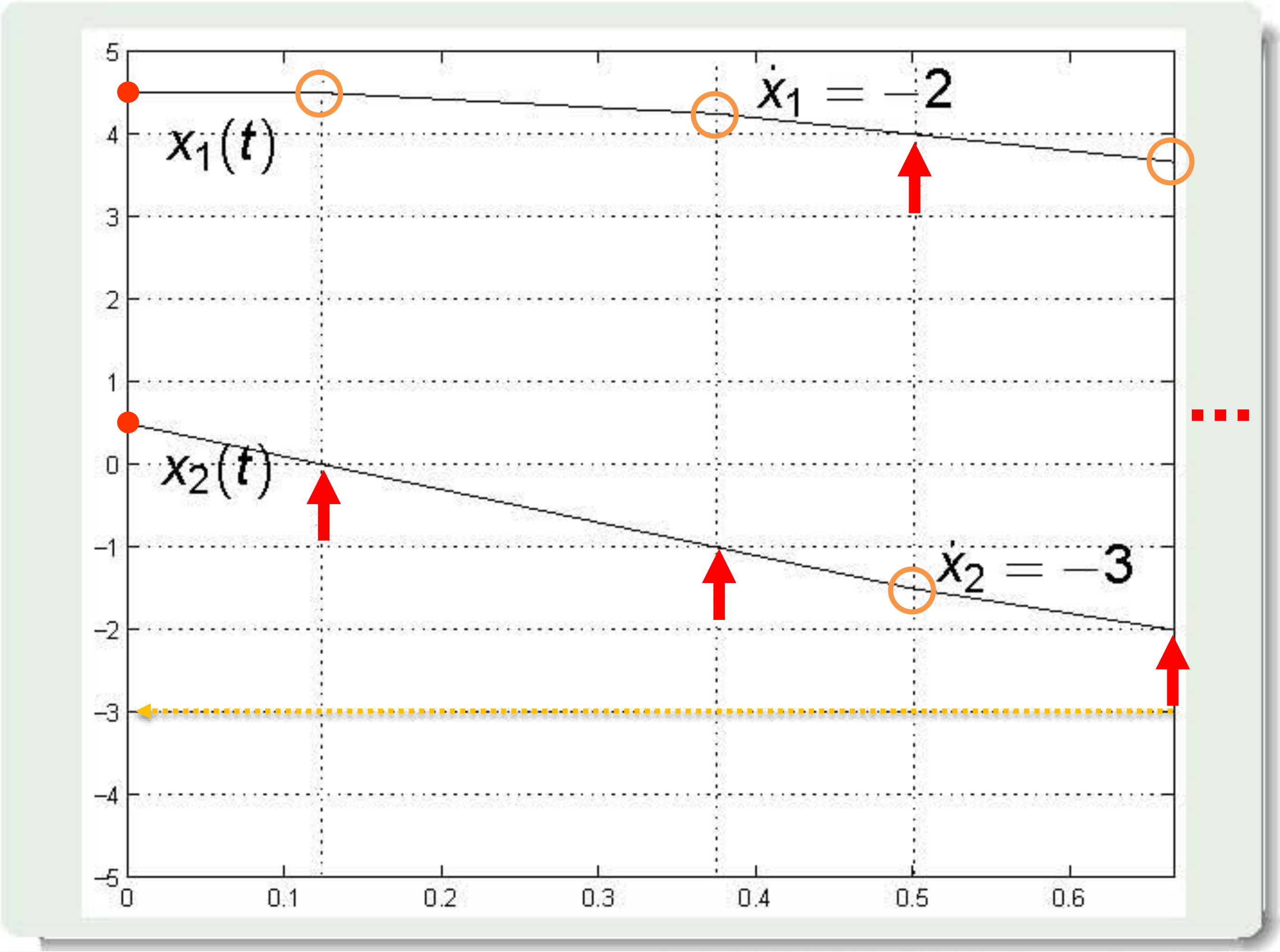
Can be solved.

Consider the I.C.:

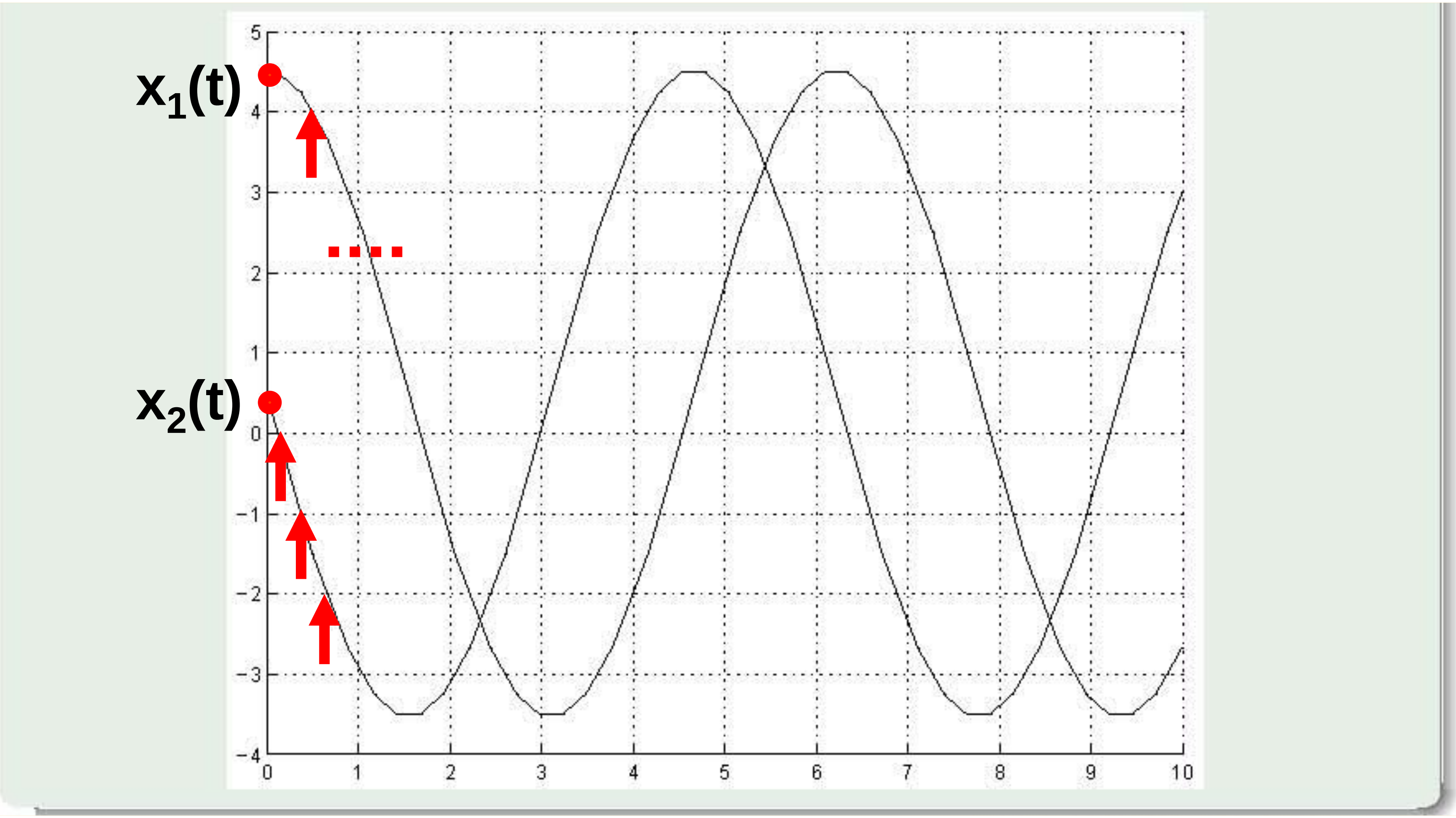
$x_1(0) = 4.5,$

$x_2(0) = 0.5$: ●

↑ = ***Discrete Event*:**
Floor(·) changes.



- Solution of the **Quantized System**:



• Differently form the Discrete Time (time slicing) integration methods presented before, the following **Quantized System**:

$$\begin{aligned} \dot{x}_1(t) &= \text{floor}(x_2(t)) = q_2(t) \\ \dot{x}_2(t) &= -\text{floor}(x_1(t)) = -q_1(t) \end{aligned}$$

• **Quantized Systems can't** be expressed as **Difference Equations** such as:

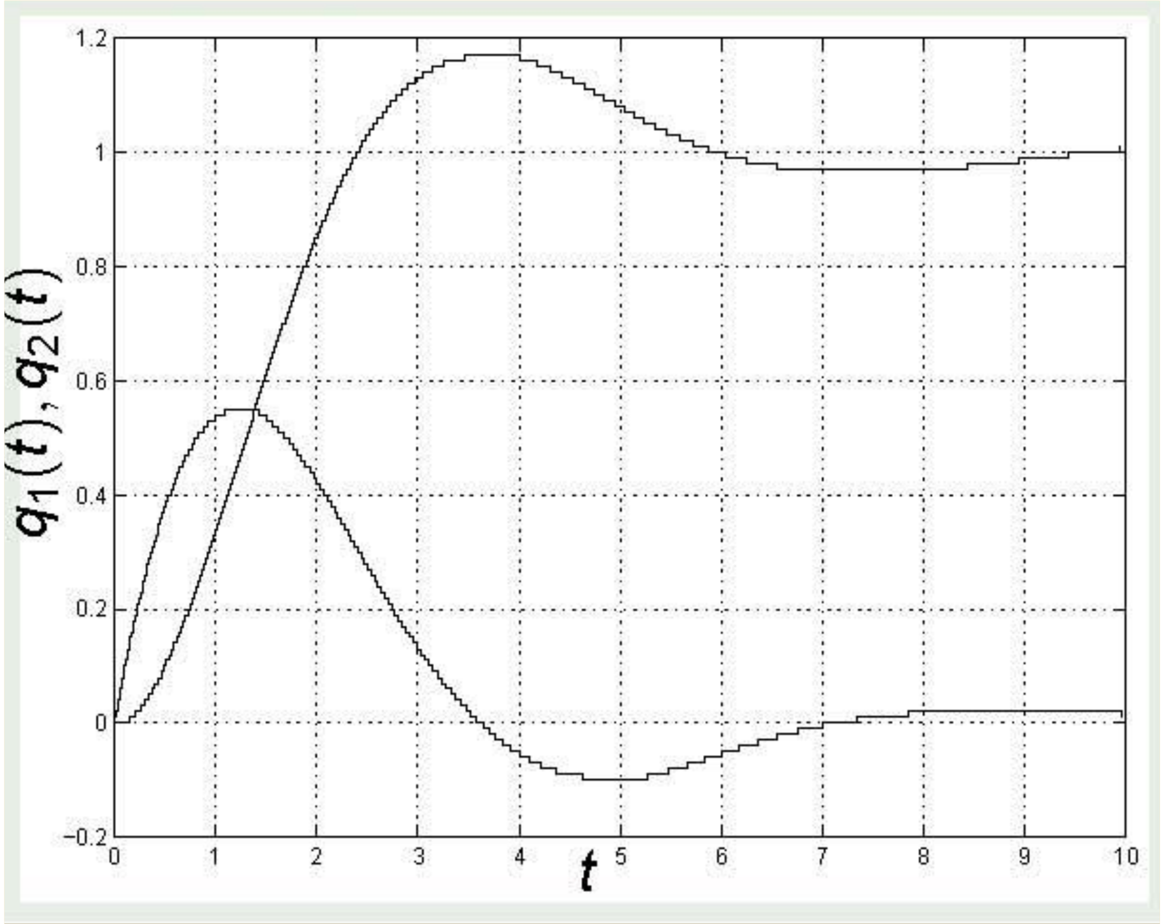
$$x(t_{k+1}) = f(x(t_k), t_k)$$

• **Quantized Systems** are instead equivalent to **DEVS models**

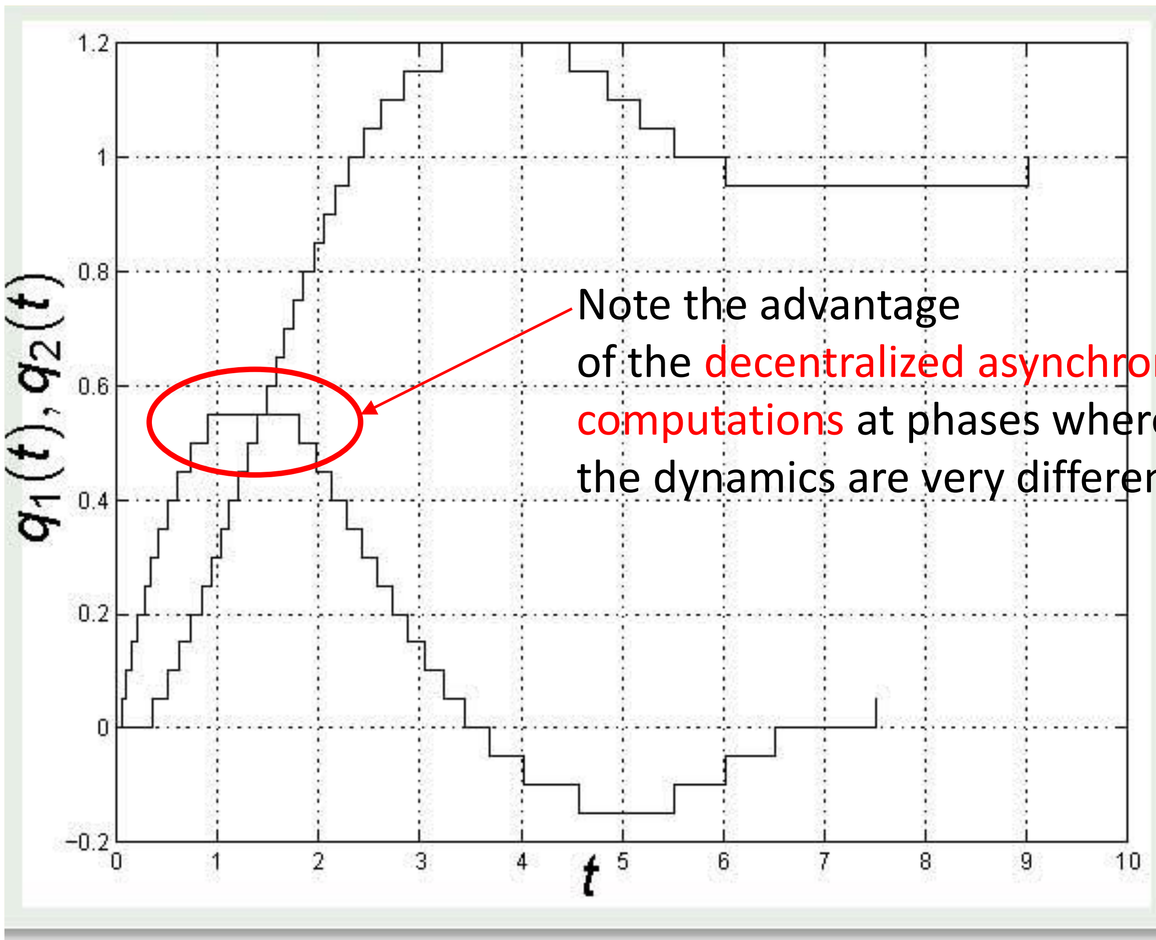
- This entails a set of unique properties from which several classes of simulation models can profit from.
- E.g. models of systems that are either heavily discontinuous, stiff, very large, or combinations of them.

- **Quantized State Systems (QSS)** *(Ernesto Kofman, 2001)*
  - Family of numerical methods based on the principle of state quantization
  - Quantize **state variables** with **discrete quanta** instead of partitioning **time** in **discrete time steps**.
  - Implemented in several **general purpose** M&S tools
    - Most advanced:
      - **PowerDEVS** (we extend it for **ATLAS TDAQ**)
      - **QSS Solver** (we extend it for **Geant4**)

- A mass-spring 2nd order system
- QSS1 solution with quantum size $\Delta Q = 0.01$

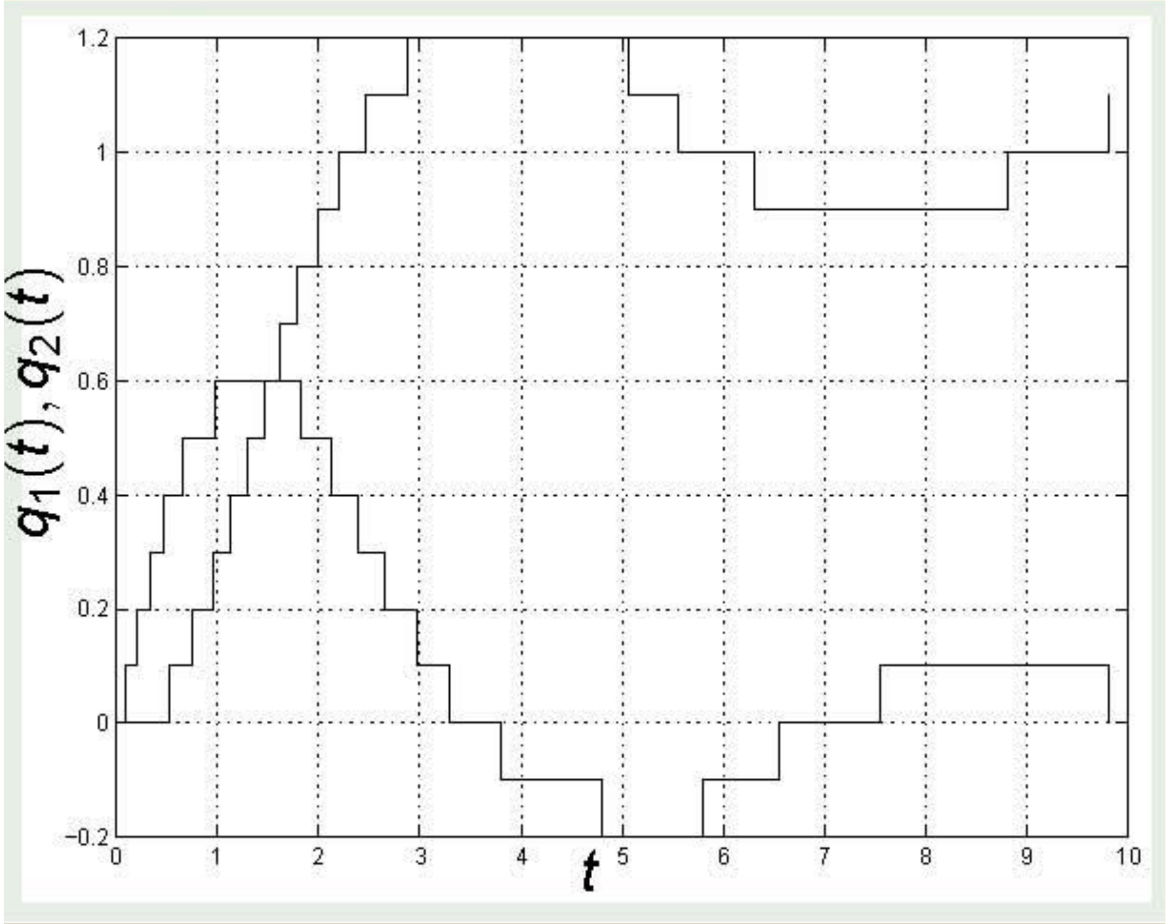


- QSS1 solution with quantum size $\Delta Q = 0.05$

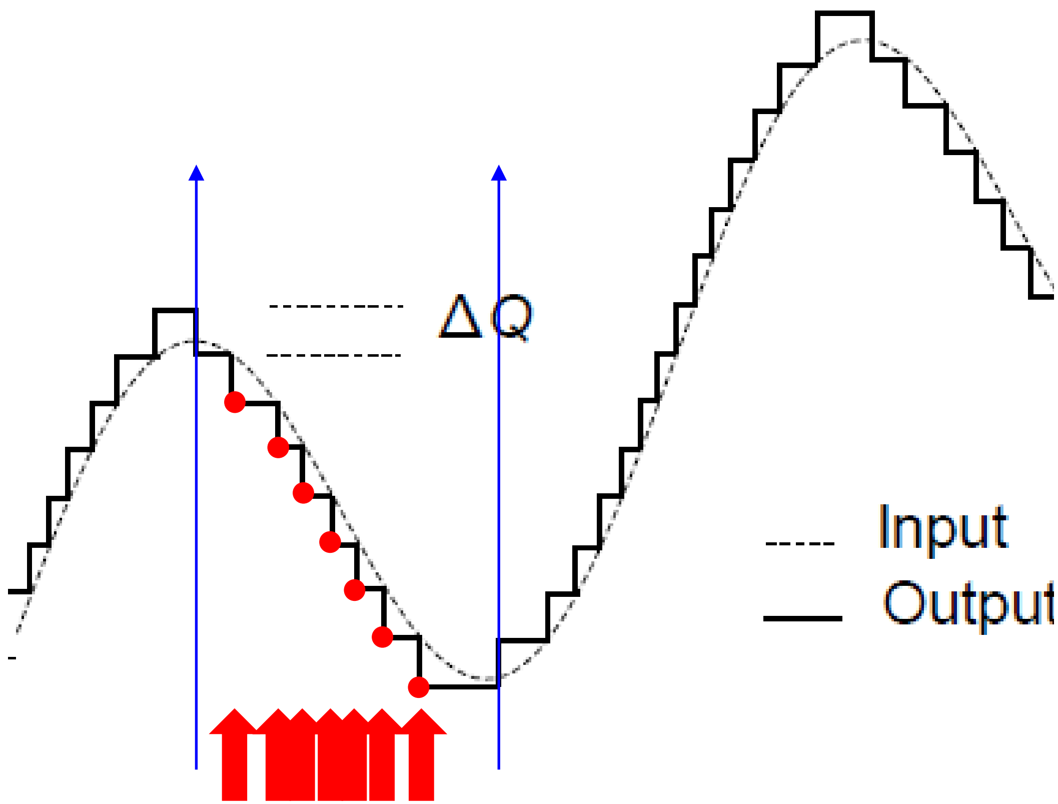


Note the advantage of the decentralized asynchronous computations at phases where the dynamics are very different

- QSS1 solution with quantum size $\Delta Q = 0.1$

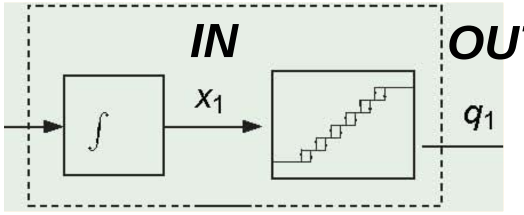


• Zero Order Quantization (e.g. with backward quantizer)

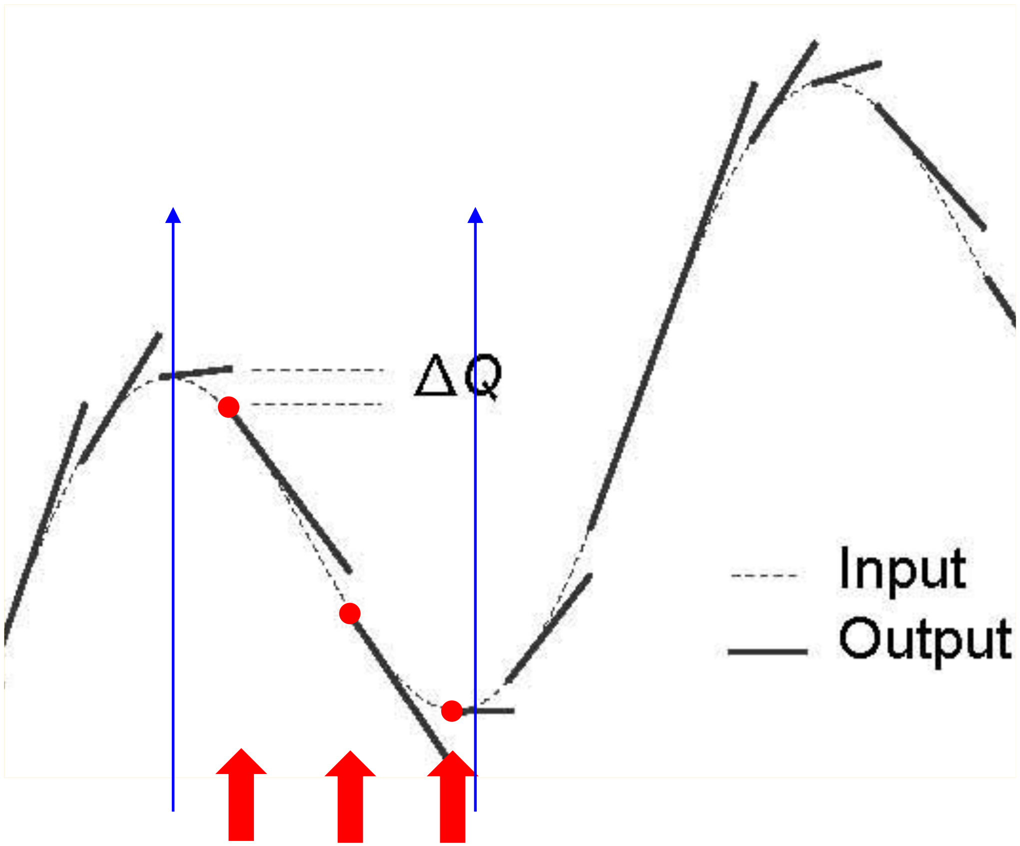


• **QSS1: First Order method.**

• **Number of steps** grows **linearly** with the **precision**.

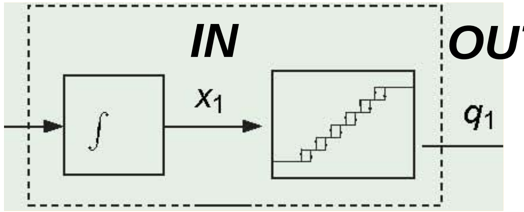


- First Order Quantization

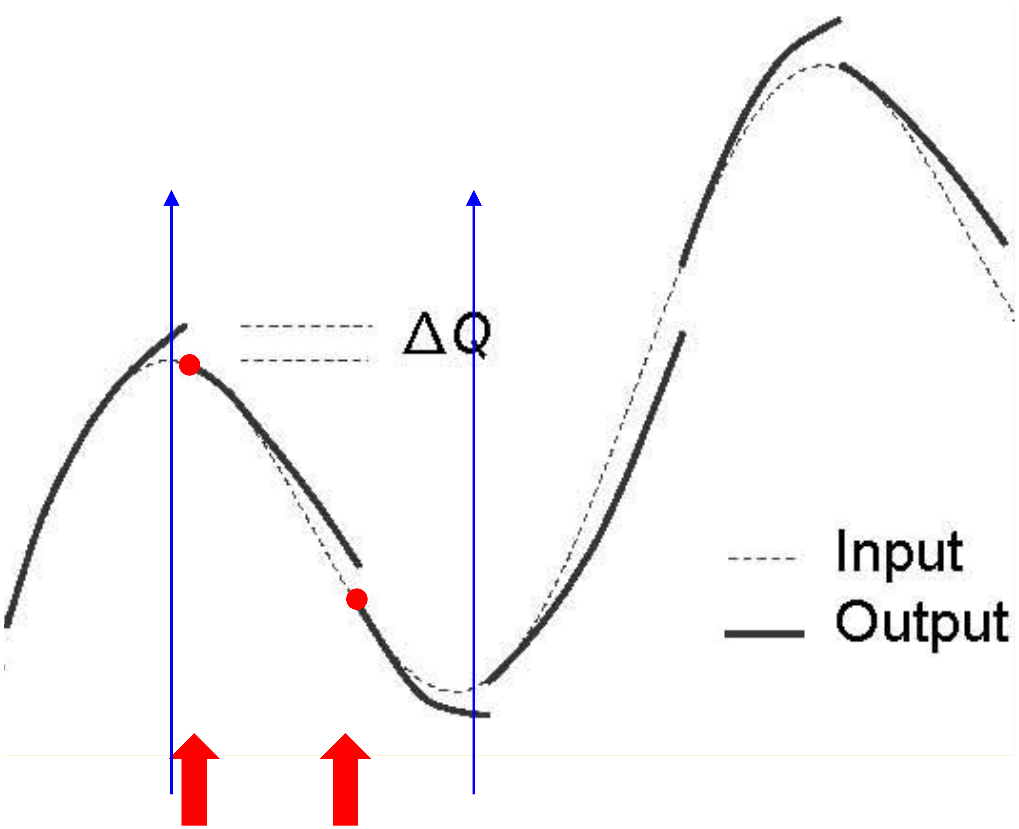


- **QSS2:** **Second Order** **method.**

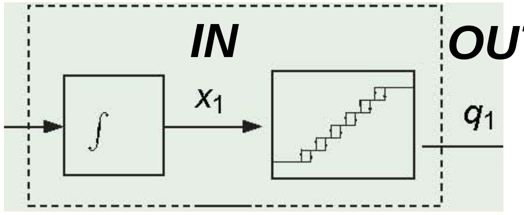
- **Number of steps** grows with the **square root** of the **precision**.



- Second Order Quantization



- **QSS3**: **Third Order method.**
- **Number of steps** grows with the **cubic root** of the **precision**.

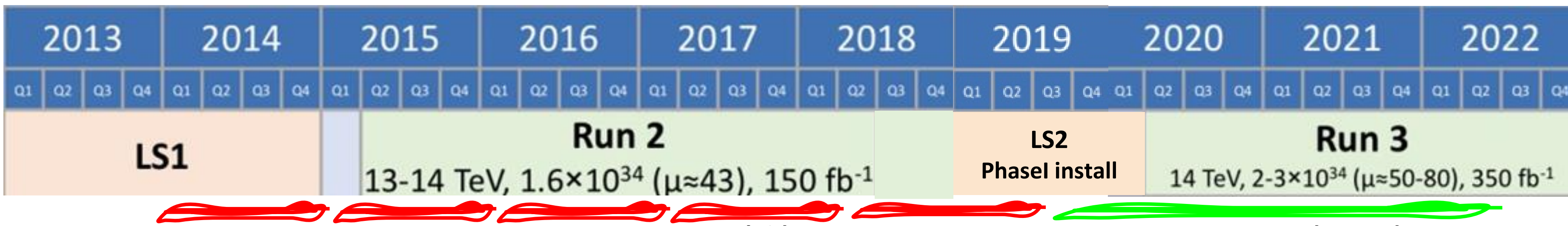


Properties

- **Naturally asynchronous**
  - Decoupled, independent computation of changes in states variables (as opposed to the time-slicing approach).
- **Dense polynomial output** on a continuous time base
  - Efficient (trivial) detection and handling of discontinuities.
- **Preserves** practical **stability**
- For linear systems the **global approximation error** can calculated analytically
- Particularly suitable for:
  - **Efficient** simulation of hybrid systems with
    - frequent discontinuities
    - sparse structure
  - **Real time** simulation

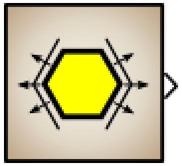
Applications

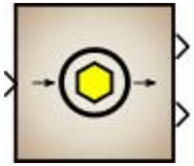
Networks of Data Flows
TDAQ ATLAS, CERN

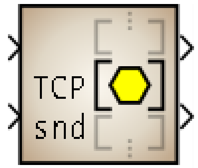


HLT packet-level model | **MASADA** | **TopoGen FELIX for PhaseI** | **Fluid models** | **HLT Hybrid models** | **More advanced models Parallel/Distributed Simulation**

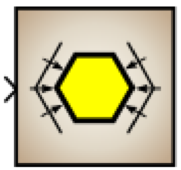
- Library of reusable models for Data Networks

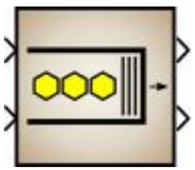

Source

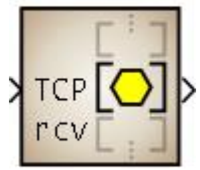

Server



TCP Sender

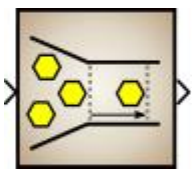

Sink

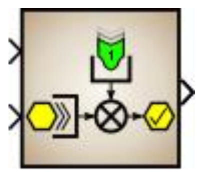

Queue

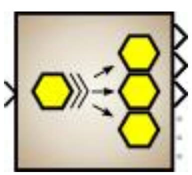


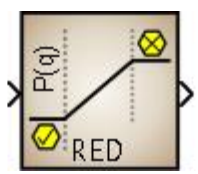
TCP Receiver


Multiplexor


Channel

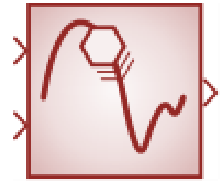

Token Bucket


Demultiplexor

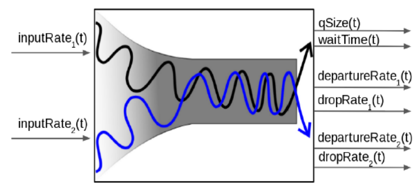


Random Early Detection

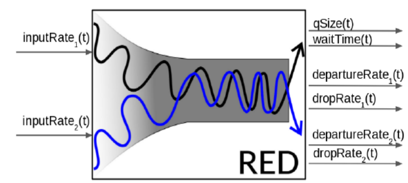
Discrete Event


Hybrid Flow Combiner

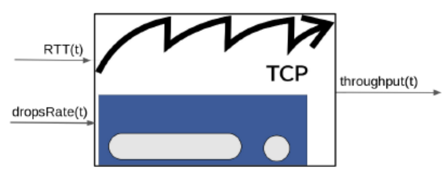
Hybrid



Fluid Queue Tail-Drop



Fluid Queue RED



Fluid Source TCP

Fluid Receiver TCP

Continuous

Data Network modeling

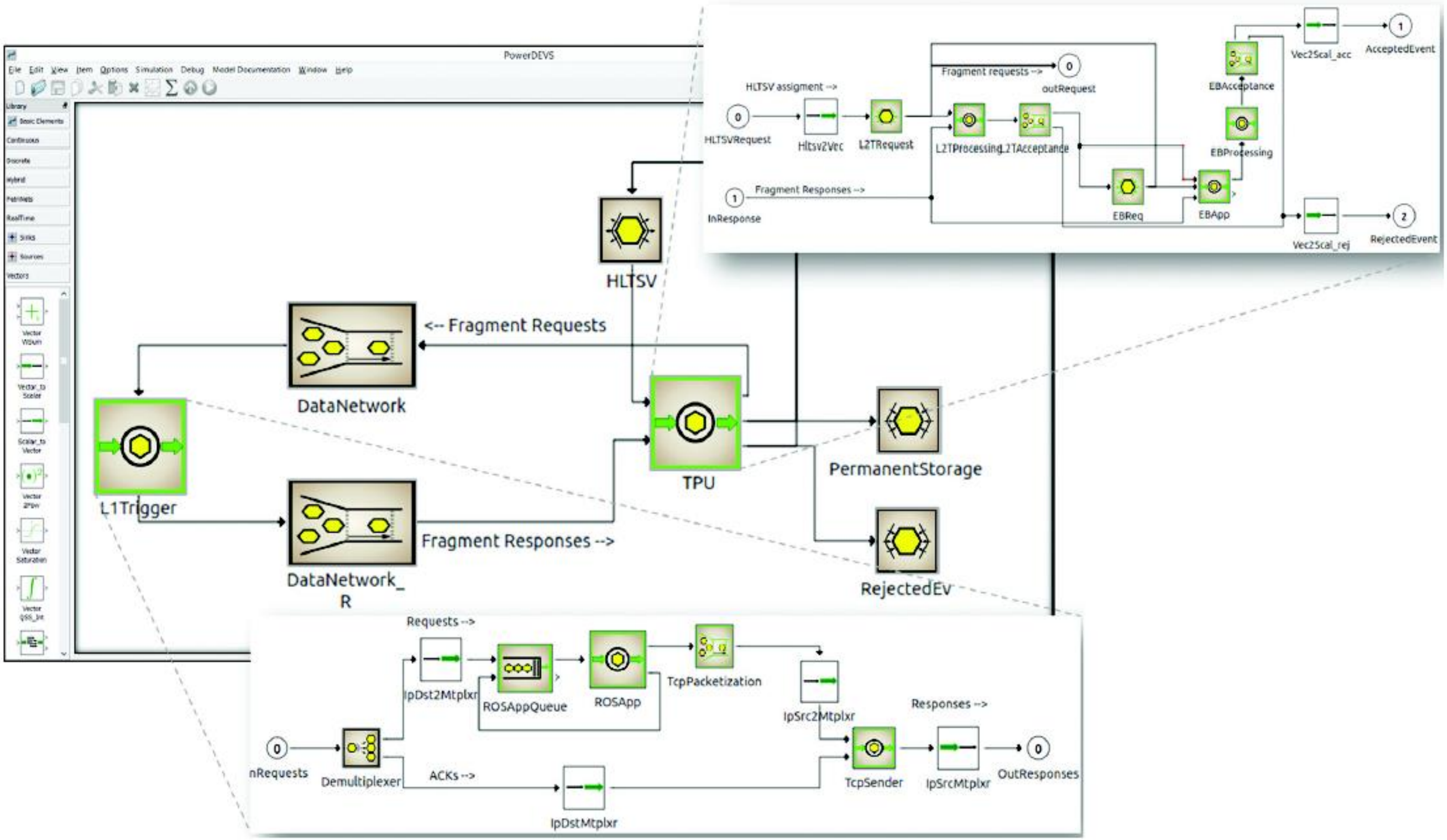


Figure 6. TDAQ simulation model implemented in PowerDEVS. Tests to validate the TCP model against the real system shifted the focus from studying averaged filtering latencies to analyzing clustered latency patterns (red curve vs. blue dots in Figure 5).

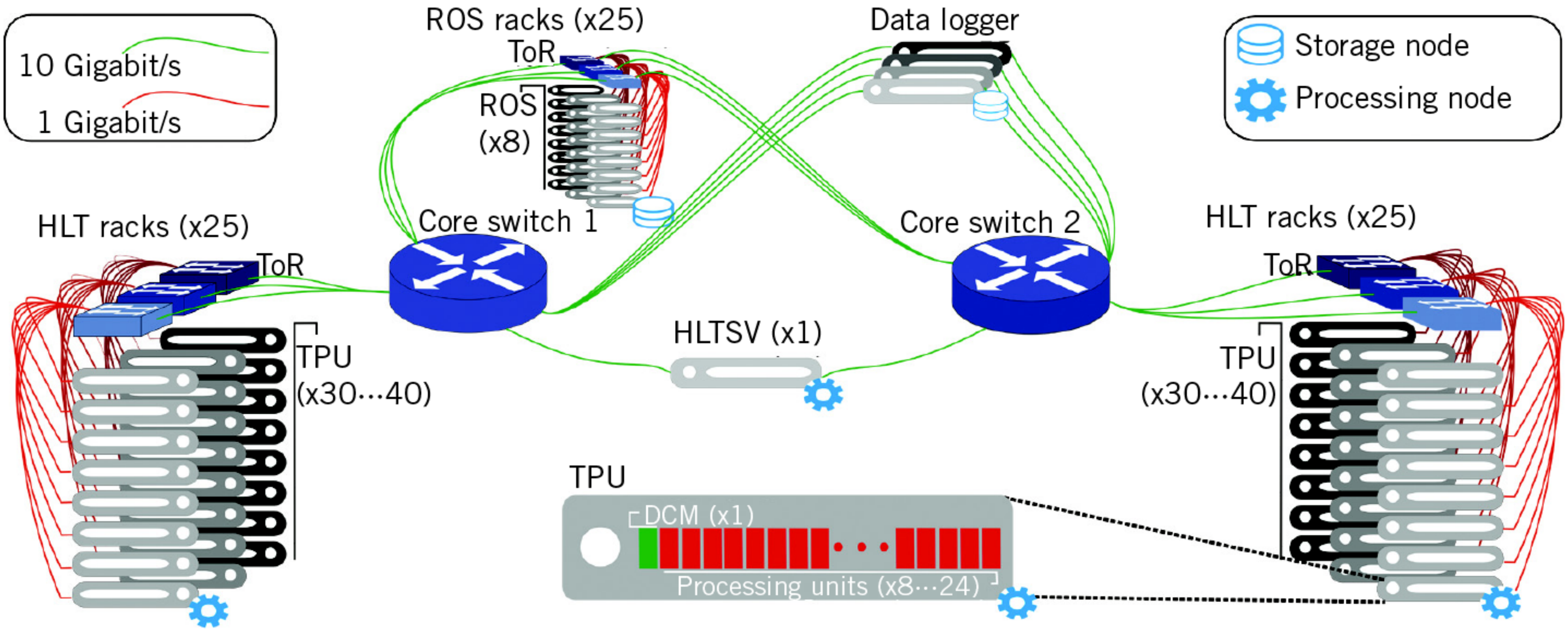


Figure 1. Topology and applications in the high-level trigger and data acquisition (TDAQ) farm. This intermediate configuration is from long shutdown 1 (LS1) in 2014.

LS1 intermediate topology

TDAQ HLT System

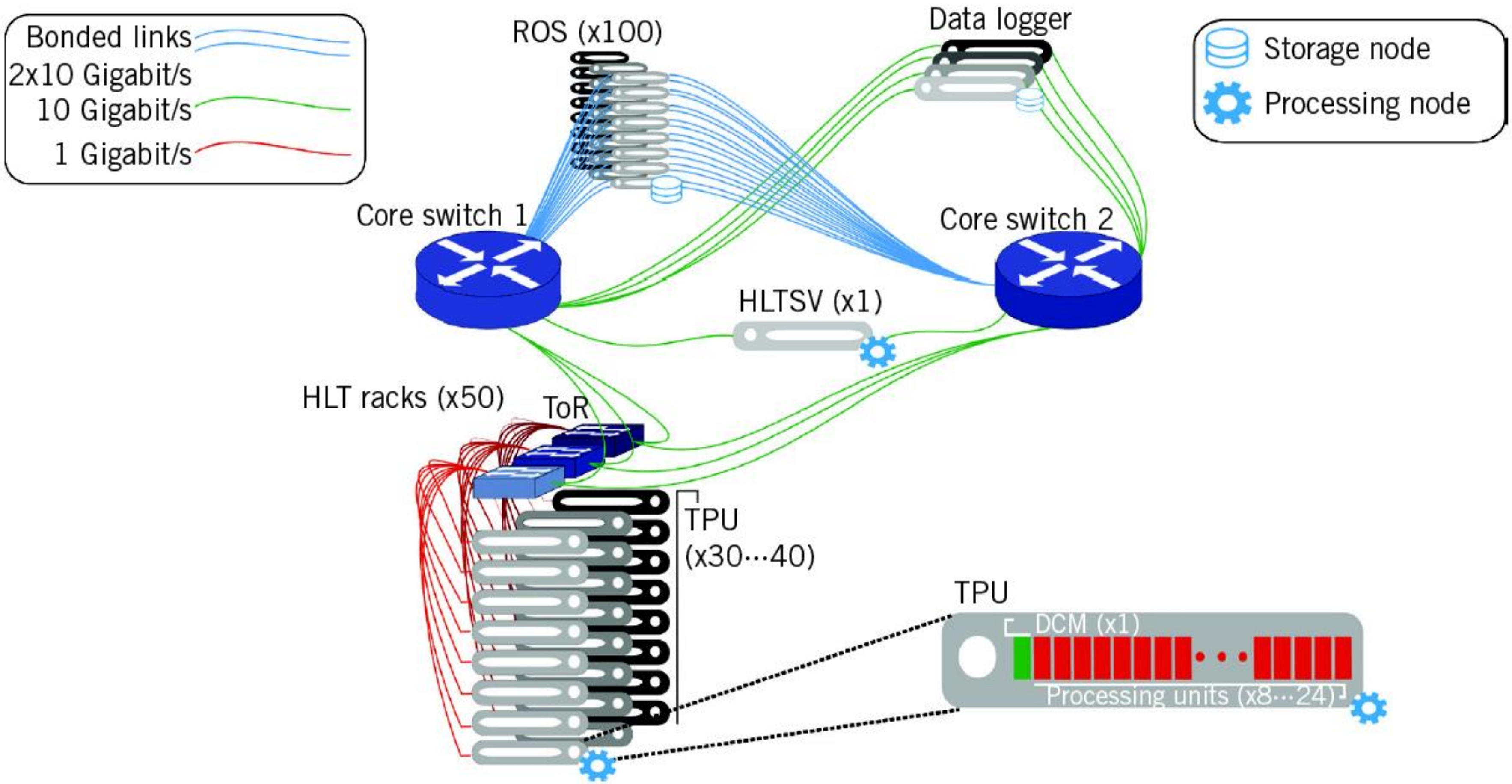


Figure 7. Topology and applications in the TDAQ HLT farm for Run2. This is an upgrade of the one in Figure 1.

Run2 target topology

TDAQ HLT Data Flow

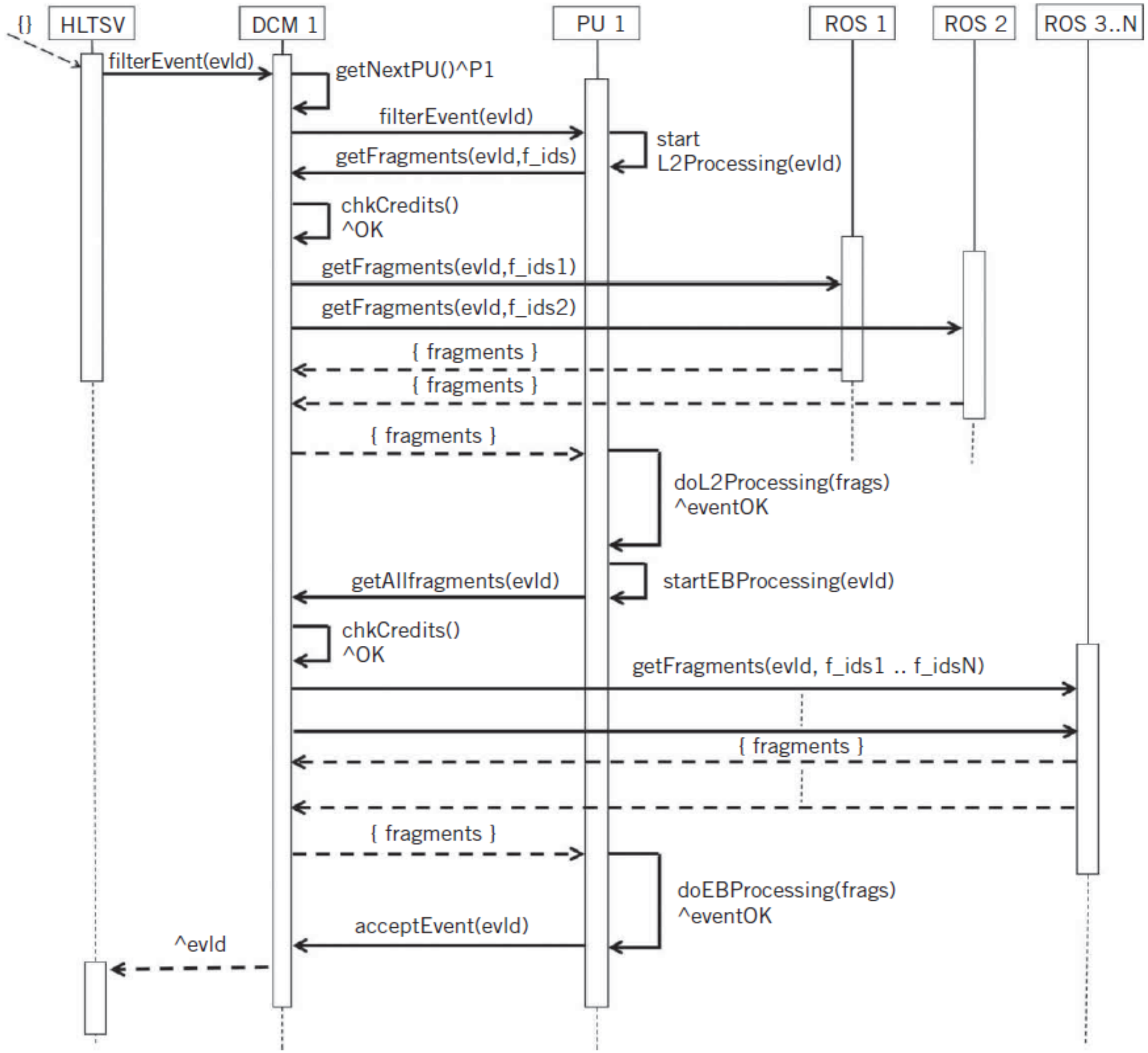


Figure 4. TDAQ application sequence diagram involved in filtering a single Event. The processing units (PUs) request information from the read-out system (ROS) in two stages: level-two (L2) filtering and Event building (EB).

Validation of simulation for the DAQ network traffic shaping strategy

Event Build Latency - Real System **Event Build Latency - Simulation**

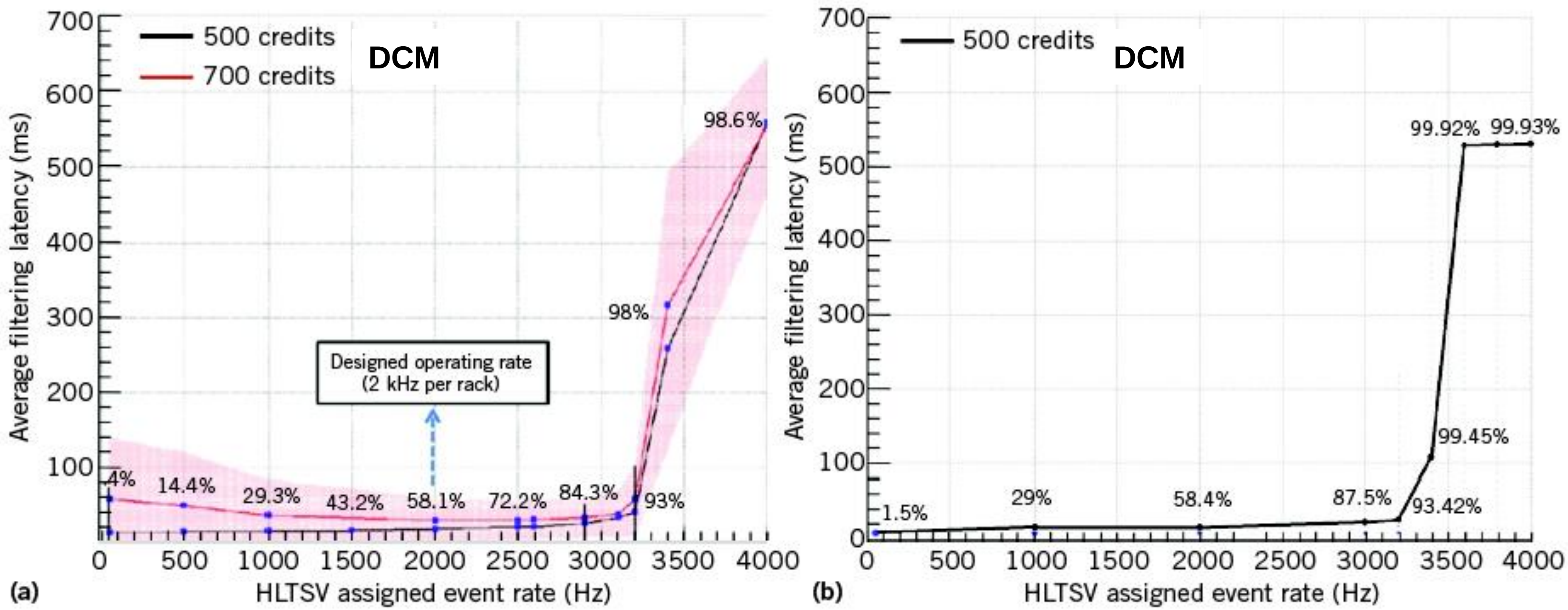
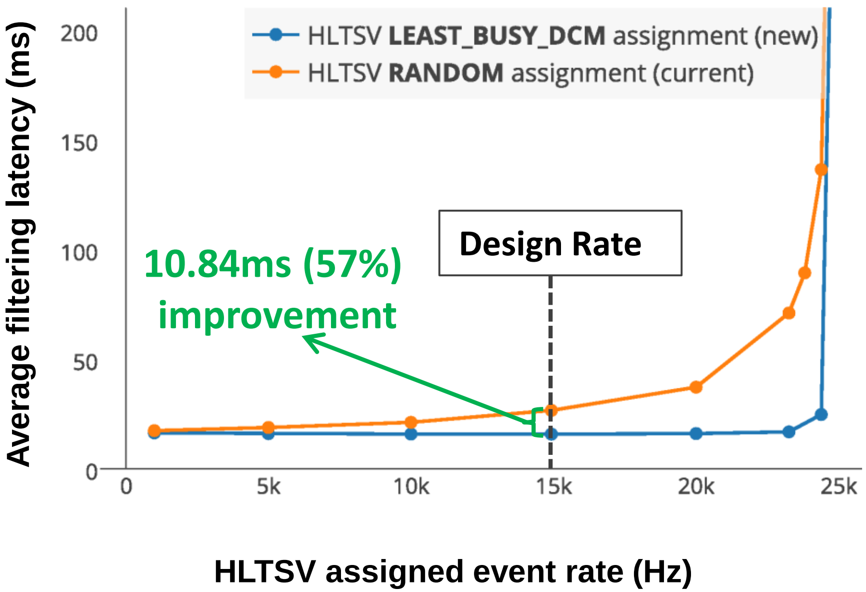
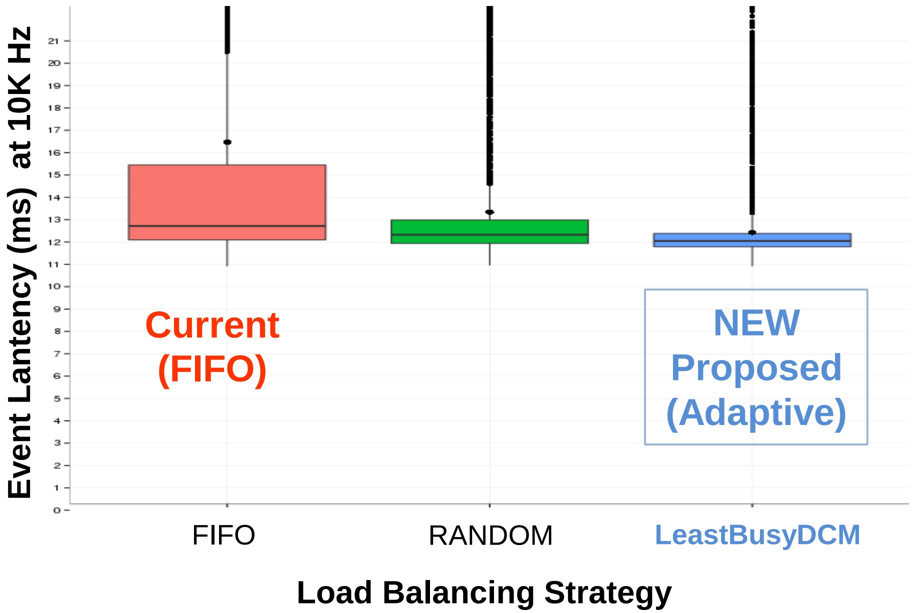


Figure 8. Average Event latency sweeping the HLTSV assignment rate (200 ROS, 1 TPU rack with 40 DCMs, 960 PUs): (a) real system measurements and (b) simulation results. Percentages represent network load, and red background shows standard deviation.

- Simulation-based search for a candidate Load Balancing to optimize DAQ Farm utilization and reduce Event Latency

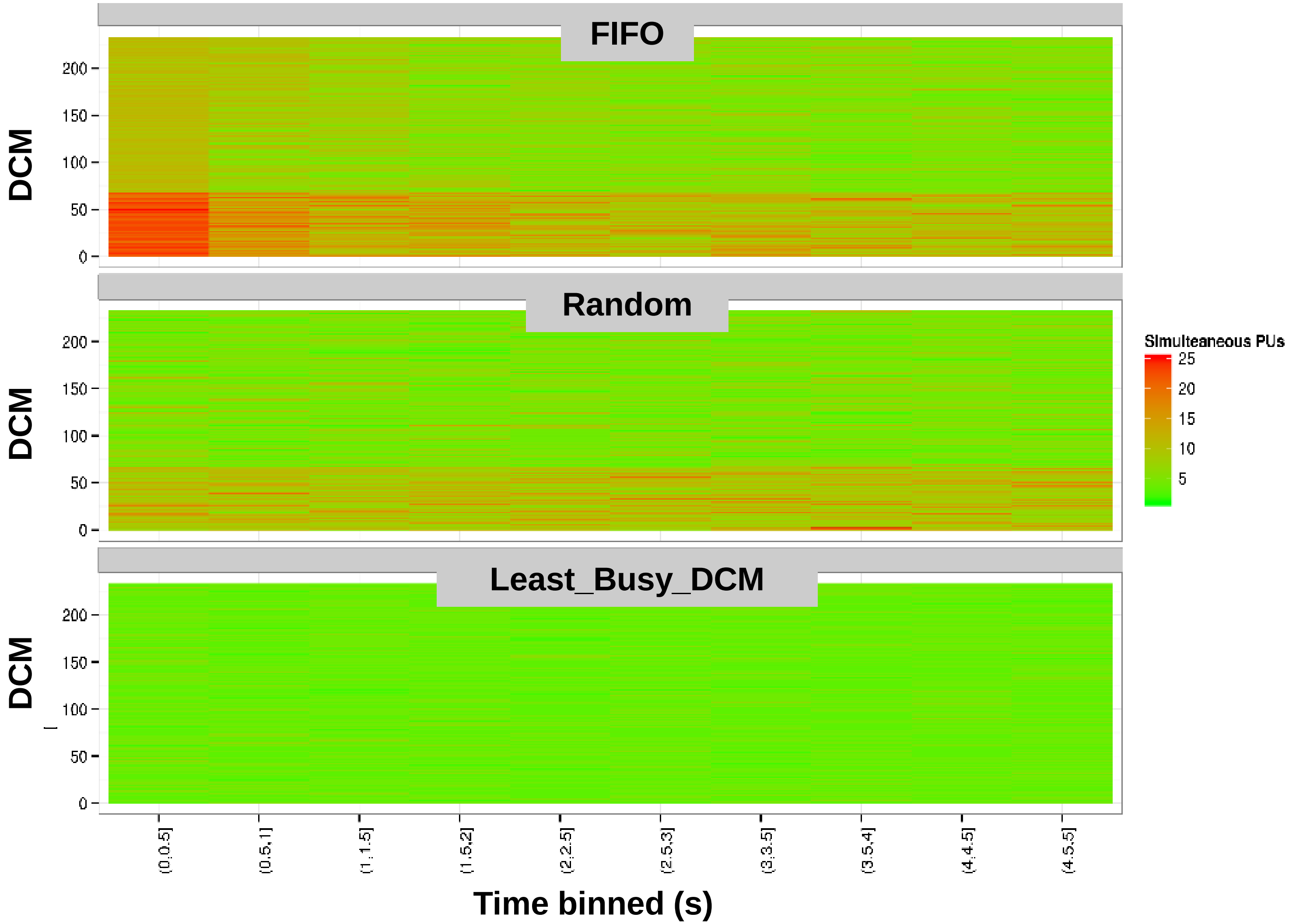
Evaluate different HLTSV load balancing strategies





- Simulation showed that an **adaptive load balancing** could improve the average Event Build Latency **as much as 50%**
- Validated on real TBED infrastructure.

Farm usage visualization – Load per TPU

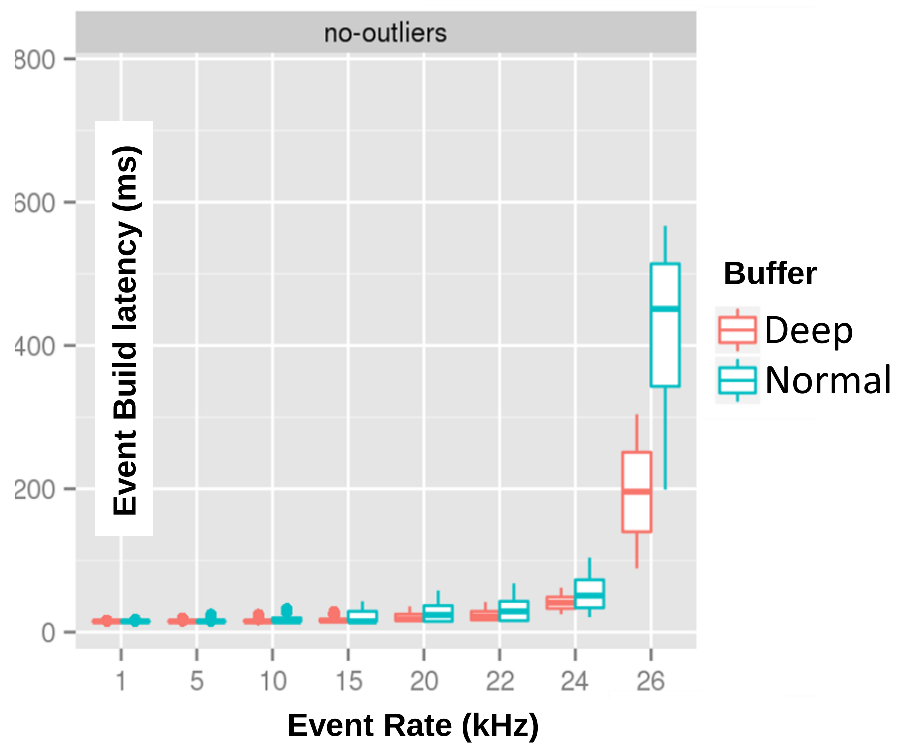


Simulation-based design

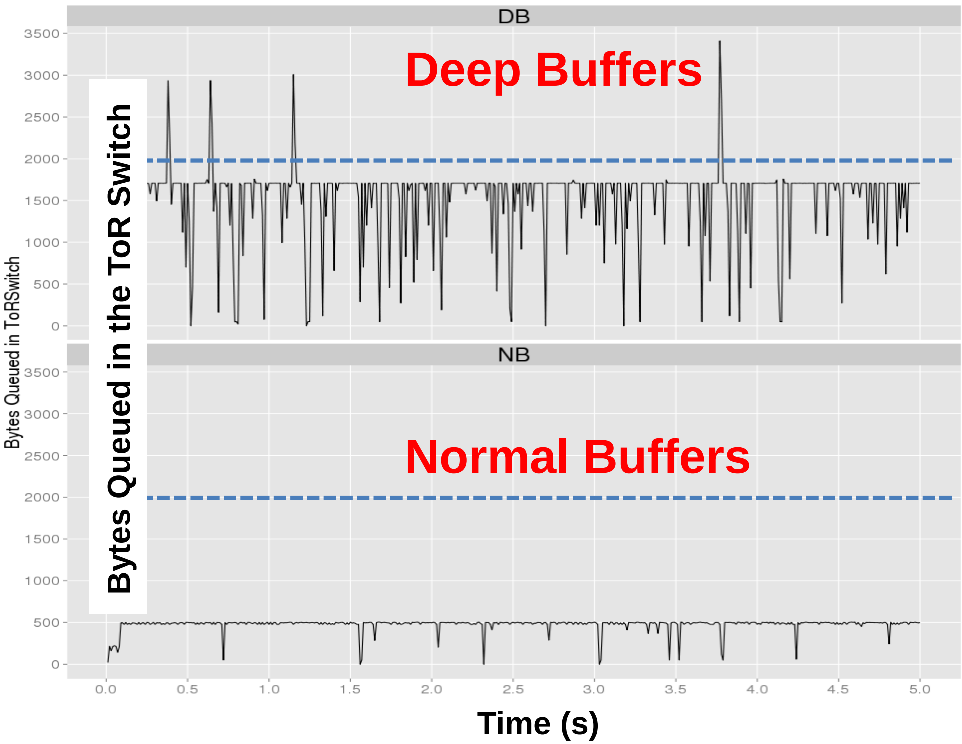
- Early evaluation of different hardware options for the ToR switches
  - Deep Buffers: 10MB shared for all ports
  - Normal buffers: 700Kb per port (39 ports per switch)

Simulation to compare different hardware options

Event Build latency at different HLTSV rates

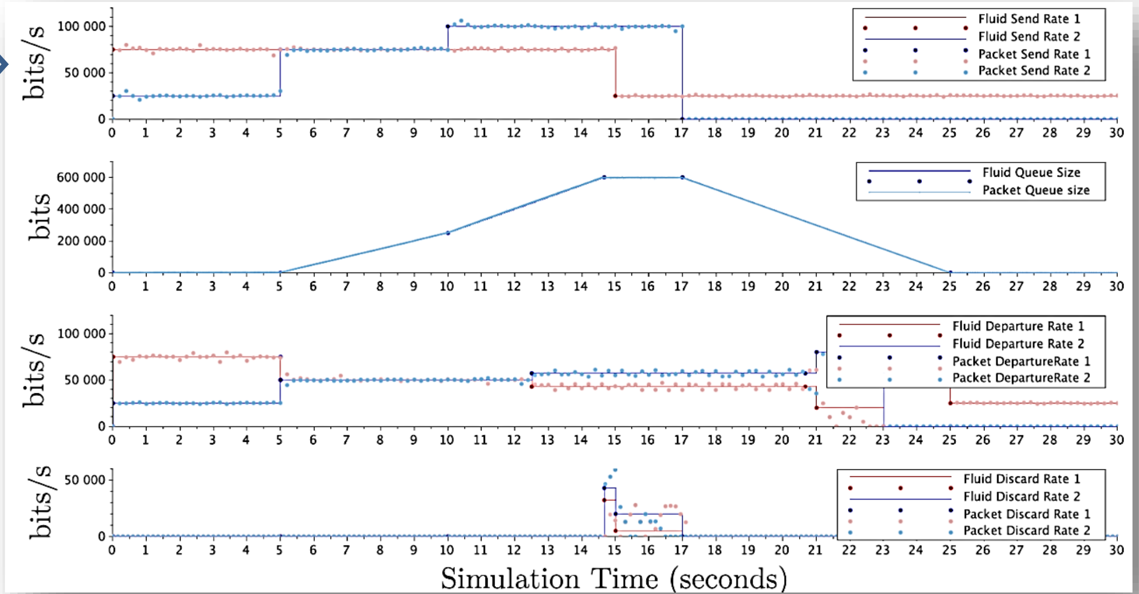


Evolution of buffered data in ToR Switches

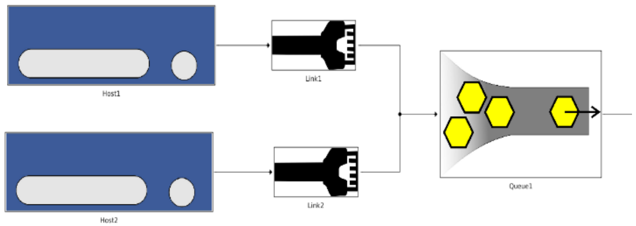


Multidomain network models

step-wise input rates
(constant packet size)

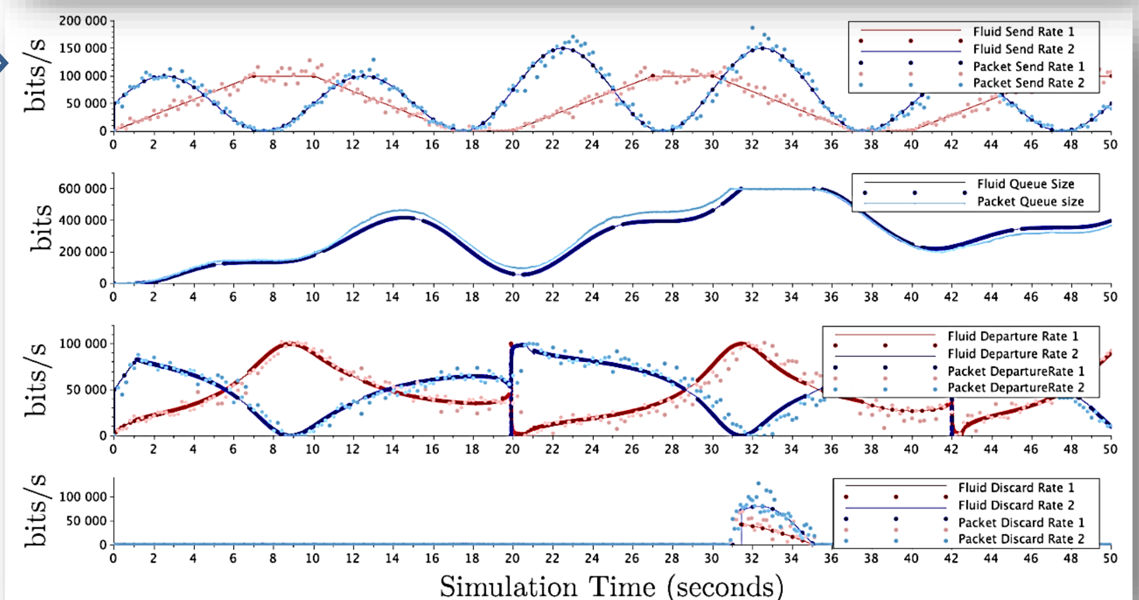


Discrete system

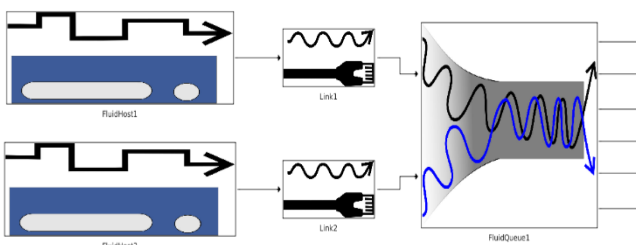


Packet-level model

sinusoidal and linear input rates
(stochastic packet size)



Continuous system

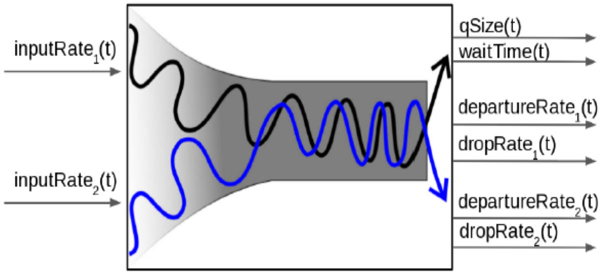


Fluid-level model

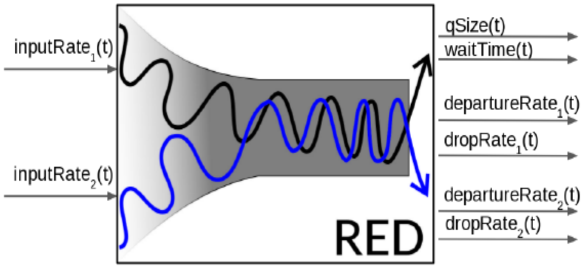
TCP dynamics

Multidomain network models

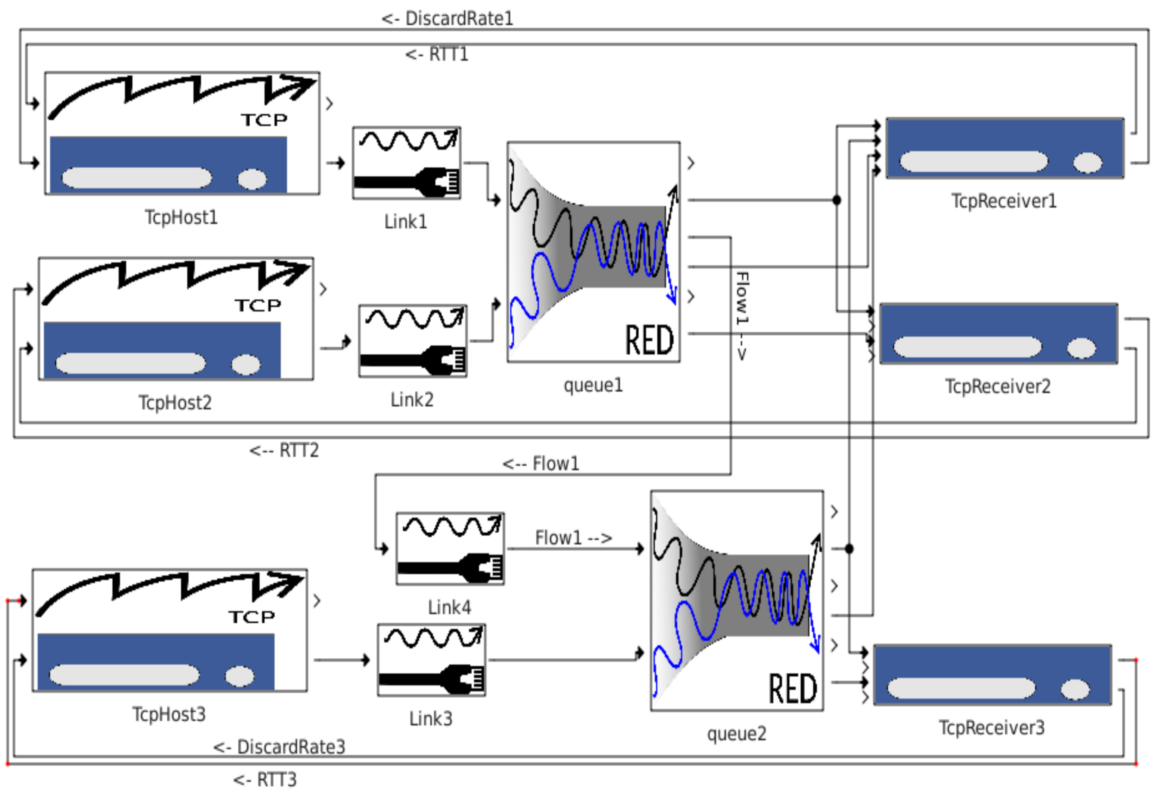
- TCP protocol
- Network of queues
- Random Early Discards

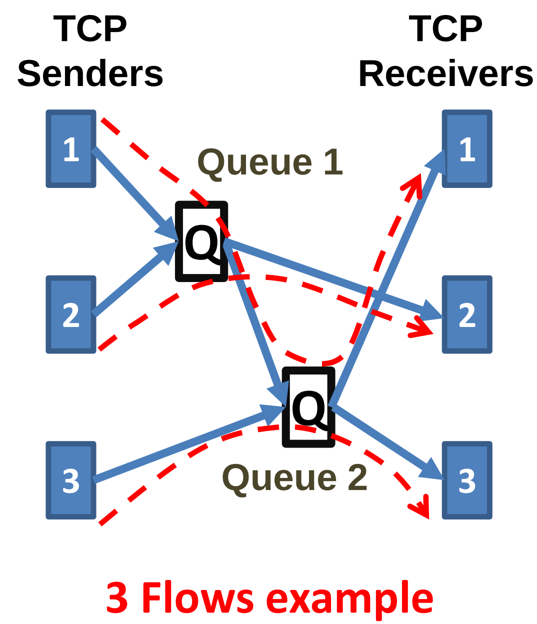


(a) Tail-drop queue

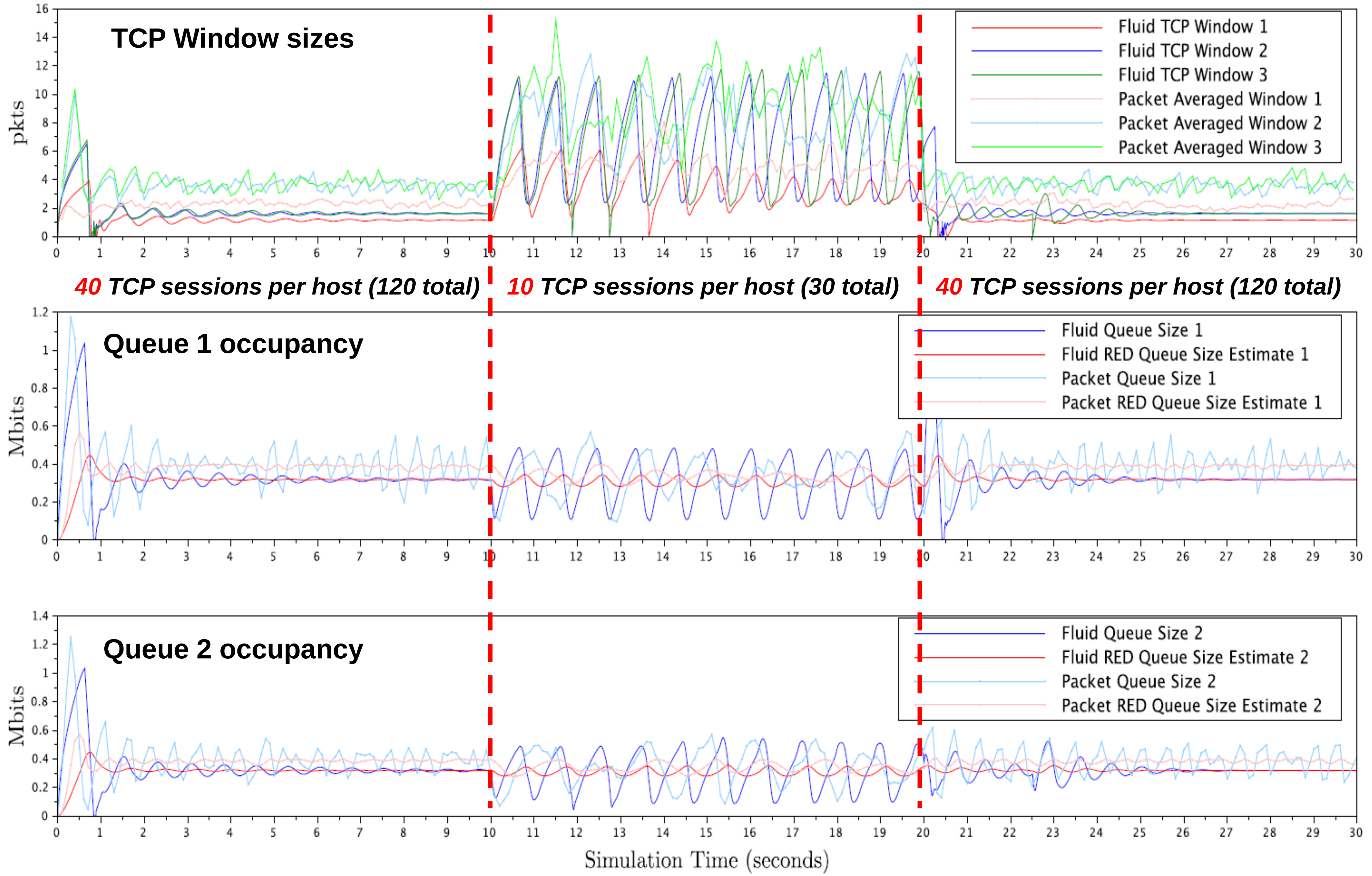


(b) RED queue



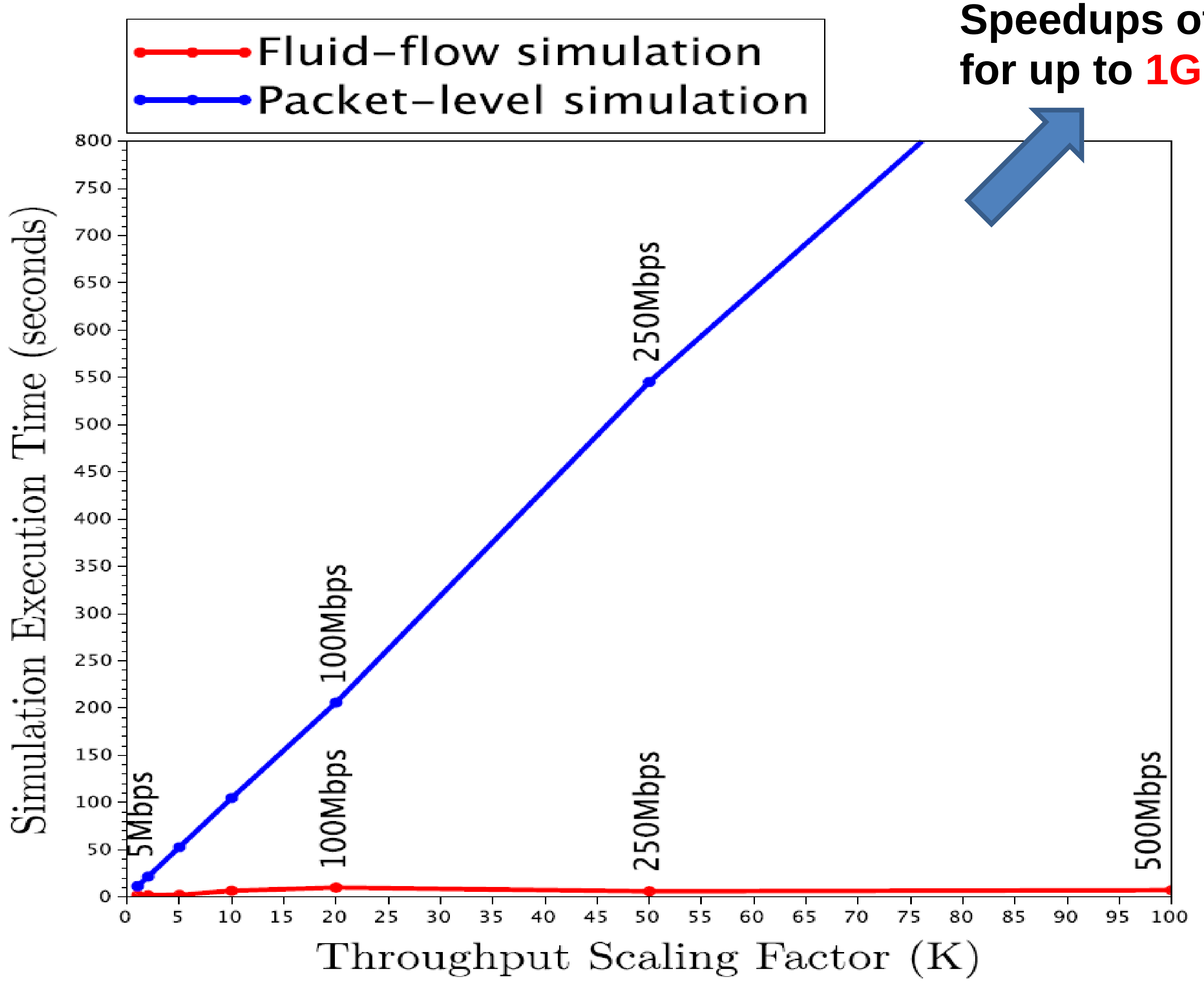


Multidomain network models



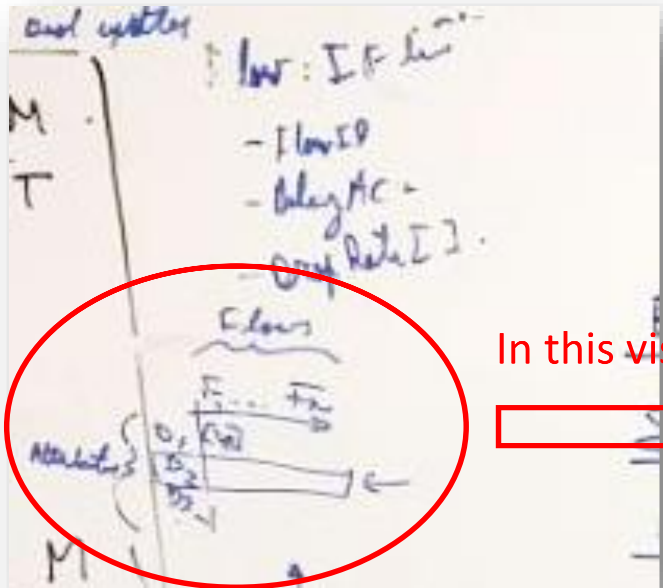
Multidomain network models

Speedups of up to 200x for up to 1Gbps links

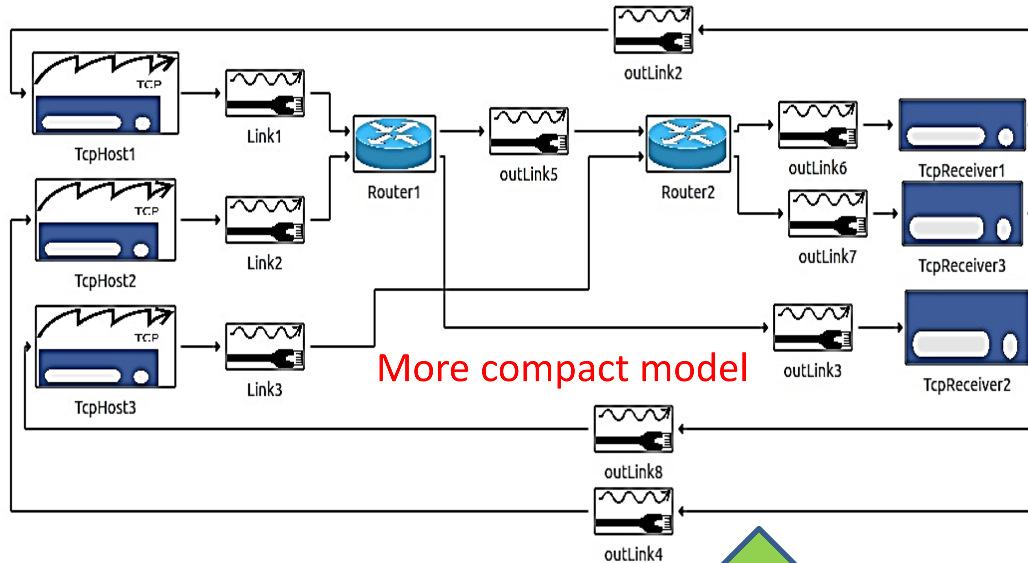


New: Multiflows

Multidomain network models

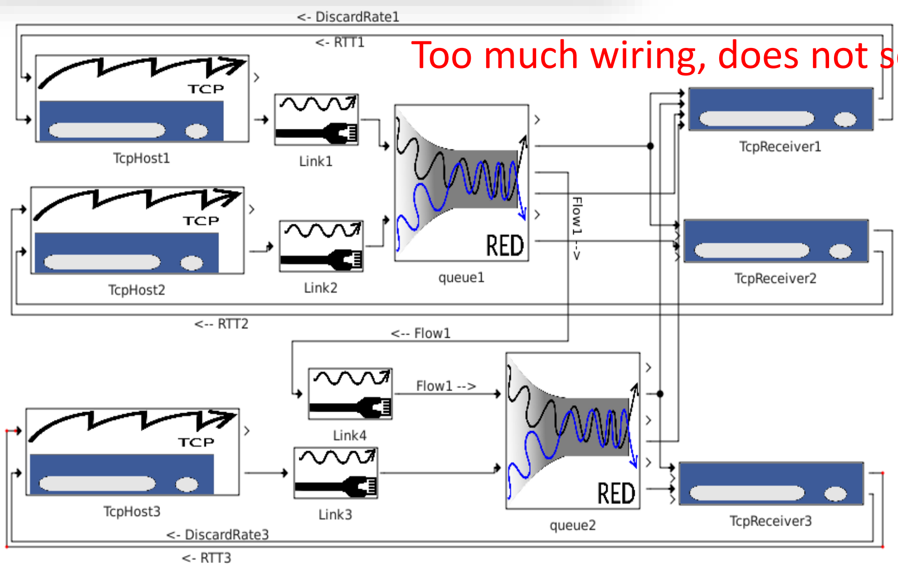


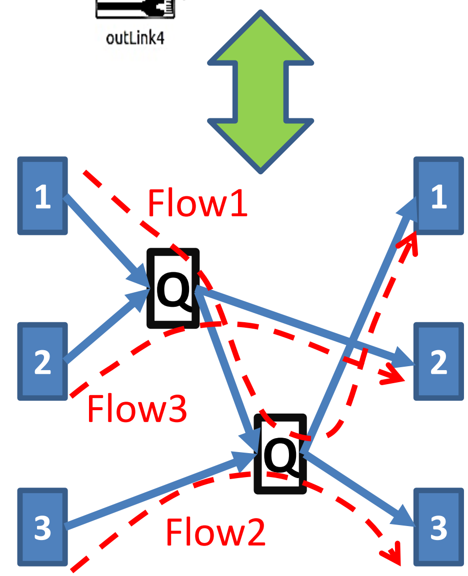
In this visit…



More compact model

Too much wiring, does not scale up

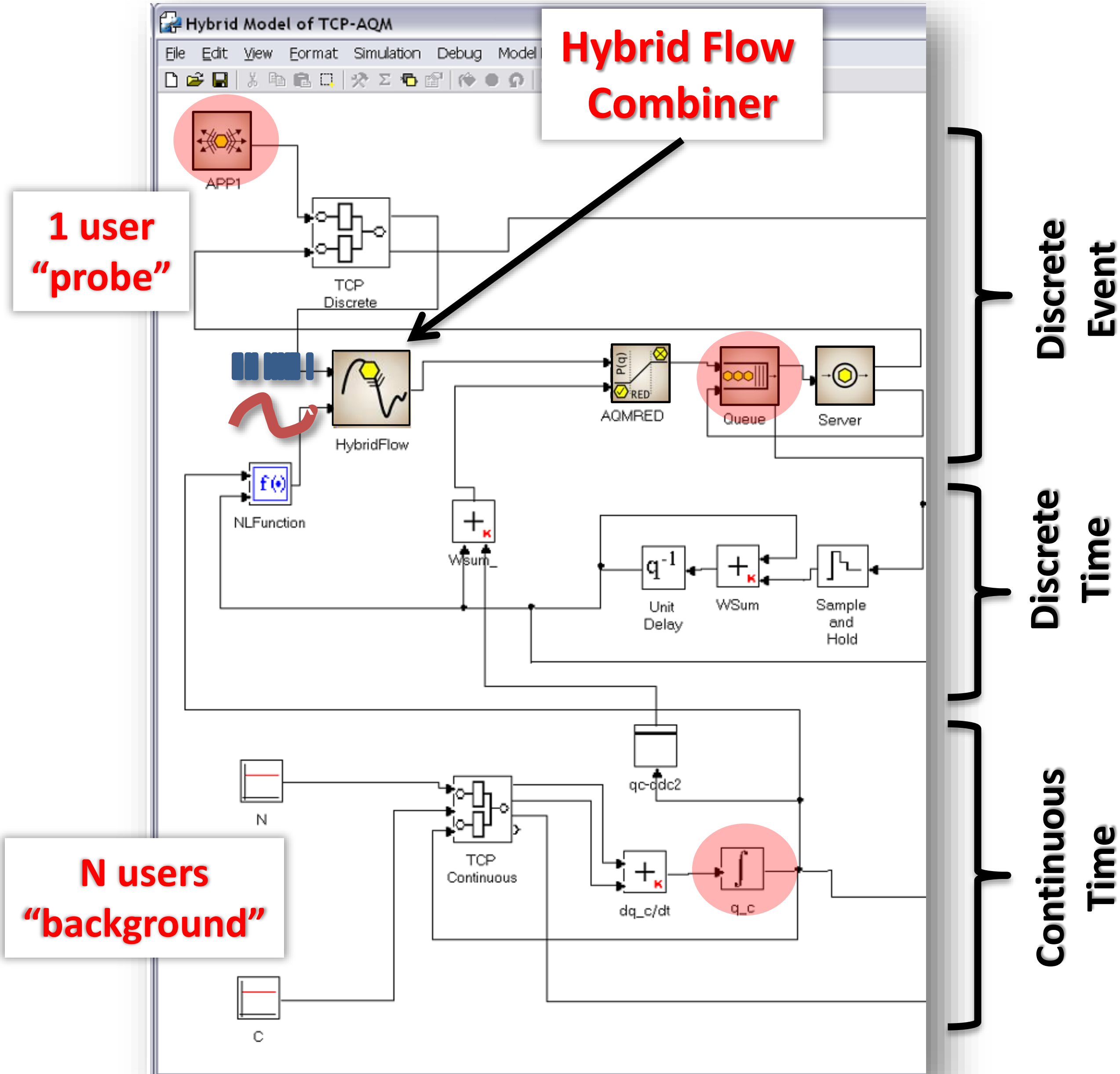




Basic topology

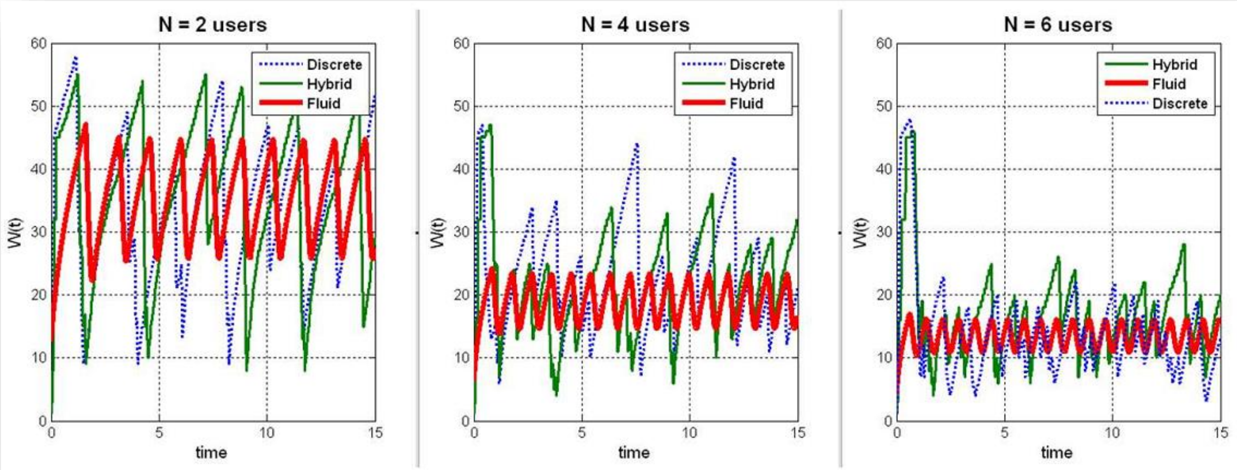
Hybrid TCP flows

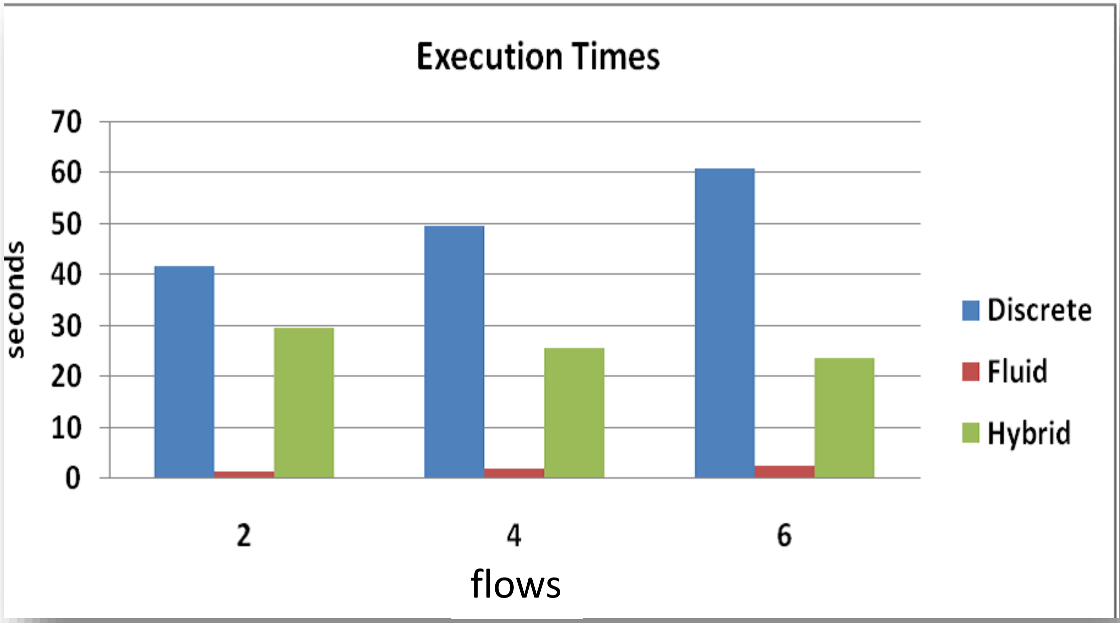
- Proof-of-concept
- Hybrid Discrete/Continuous Flows
- **1 discrete** "probe" flow
- **N continuous** "background traffic" flows



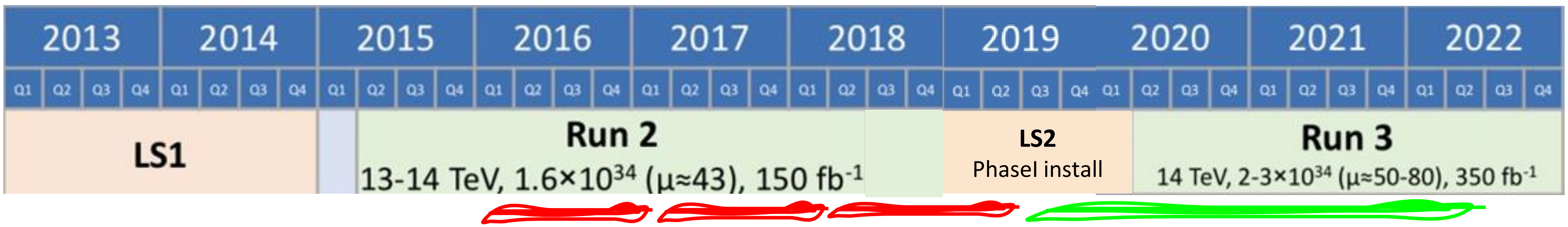
Speedups and validation

- Very acceptable qualitative results
- Preserves the level of detail of the discrete flow
- Speedup of hybrid vs. pure discrete models
  - From **2x to 3x** for **less than 10 flows**
- **Ongoing research:**
  - **increase speedups**
  - **scale up to thousands of flows**





Particle Tracking for HEP detector simulation
CSD, Fermilab



QSS methods for HEP Applications (CMS)

GQLink Interface Geant4 with QSS methods

GQLink optimization & other detectors

Advanced Geometry handling Parallel/Distributed Simulation

Motivation

- Simulation in HEP involves the numerical solution of ODE systems
  - Determine the trajectories described by charged particles in a magnetic field.
  - As a particle moves through a detector, each volume crossing interrupts the underlying numerical solver.
  - Traditional methods can invest considerable computational efforts to handle very frequent discontinuities accurately (detection of intersection points).

Goals

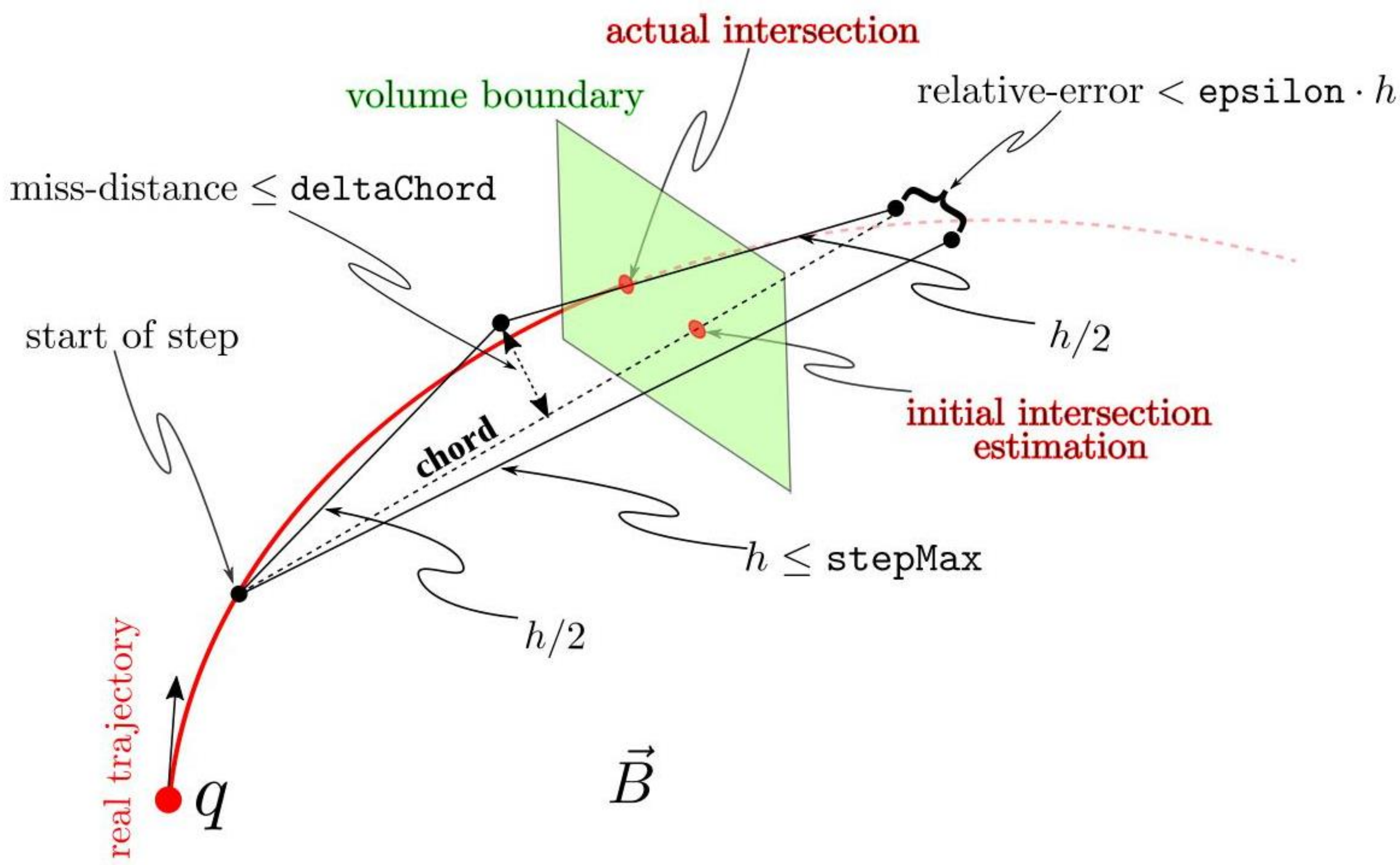
- Quantized State System methods (QSS)
  - Methods exhibiting attractive features for this type of HEP simulation scenarios
- Our goals in this line of research are:
  - Develop an implementation of QSS for the Geant4 toolkit
  - Characterize its performance in varied realistic HEP applications

The Geant4 toolkit

- Geant4: the most widely used simulation toolkit in contemporary HEP experiments.
  - Provides classical numerical methods based on time discretization (variations of the Runge-Kutta family of numerical solvers)
  - Uses custom iterative algorithms to approximate the event times of each spatial discontinuity
  - When events are very frequent, they can dominate the CPU time within the integration method

Geant4: particle transport

Transportation of a charged particle q along a step of length h proposed by a physics process:



$\Rightarrow$ a total of 11 RHS evaluations involved for the 4th order Runge-Kutta.

QSS for Lorentz Eqs.

Lorentz equations

$$\begin{cases} \dot{x} = v_x & \dot{v}_x = \frac{q\,c^2}{m\,\gamma} \cdot (v_y\,B_z - v_z\,B_y) \\ \dot{y} = v_y & \dot{v}_y = \frac{q\,c^2}{m\,\gamma} \cdot (v_z\,B_x - v_x\,B_z) \\ \dot{z} = v_z & \dot{v}_z = \frac{q\,c^2}{m\,\gamma} \cdot (v_x\,B_y - v_y\,B_x) \end{cases}$$

- x, y, z, v_x, v_y, v_z are the state variables

$\Downarrow$

Quantized approximation

$$\begin{cases} \dot{x} = q_{v_x} & \dot{v}_x = \frac{q\,c^2}{m\,\gamma} \cdot (q_{v_y}\,B_z - q_{v_z}\,B_y) \\ \dot{y} = q_{v_y} & \dot{v}_y = \frac{q\,c^2}{m\,\gamma} \cdot (q_{v_z}\,B_x - q_{v_x}\,B_z) \\ \dot{z} = q_{v_z} & \dot{v}_z = \frac{q\,c^2}{m\,\gamma} \cdot (q_{v_x}\,B_y - q_{v_y}\,B_x) \end{cases}$$

- Each state variable s is approximated by the quantized variable q_s

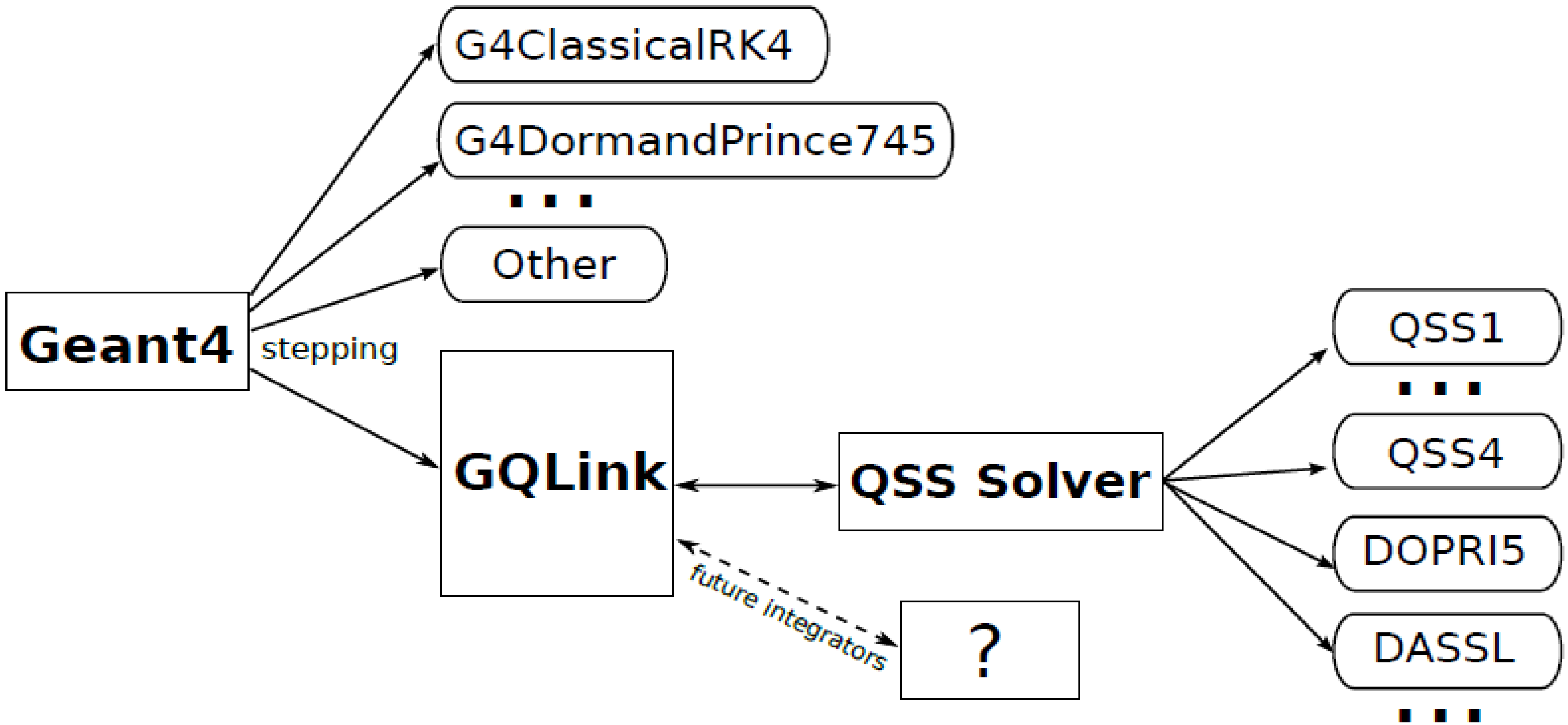
QSS Solver – Define model

- Models are mathematically described by differential, algebraic and discrete equations.

Example: Lorentz equations in GQLink

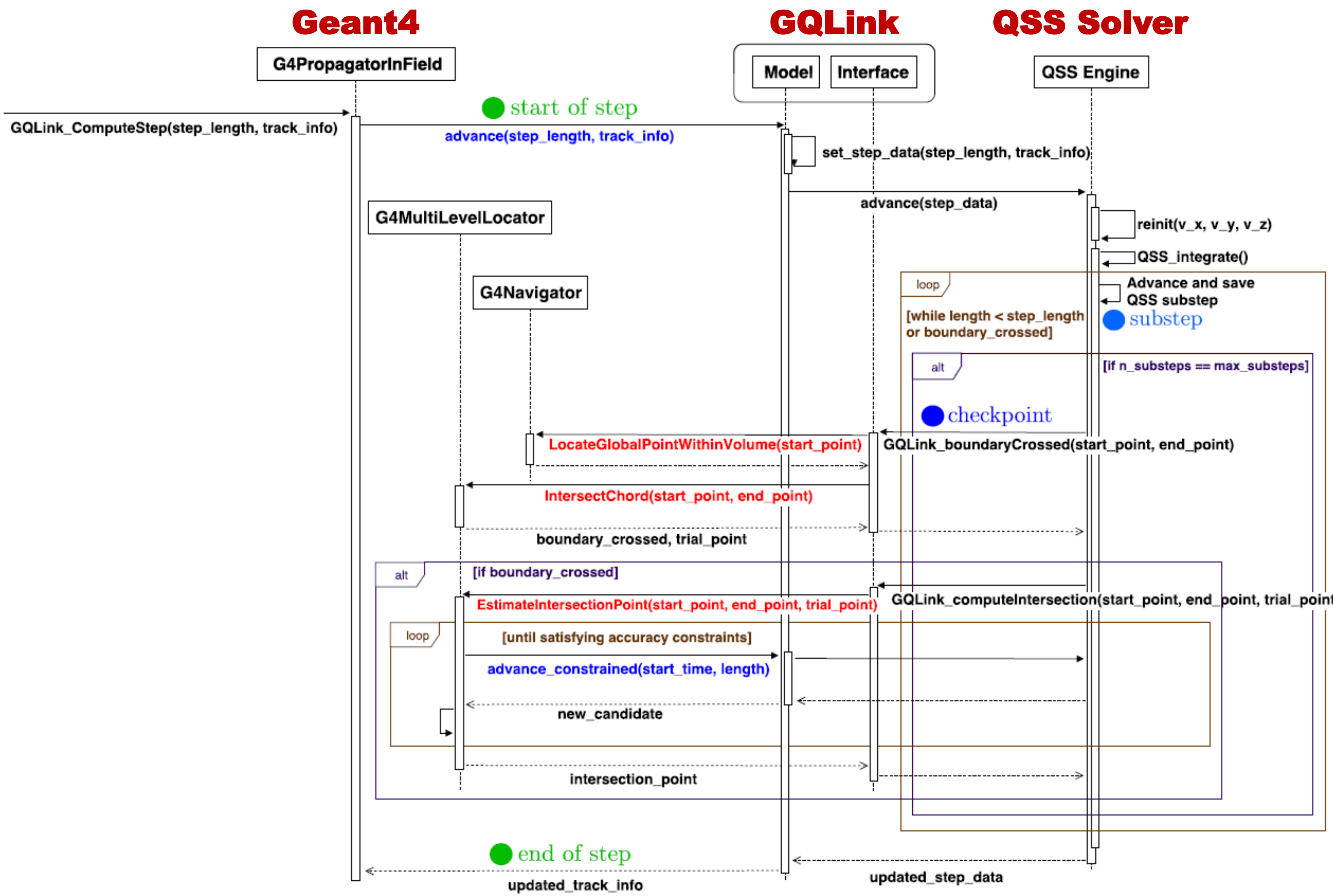
```
model UsualEq
...
parameter Real invMGamma = 1 / (m * gamma);
parameter Real coeff     = q*c*c * invMGamma;
...
equation
(Bx,By,Bz) = GQLink_GetB(x,y,z);
der(x) = vx;   der(vx) = coeff * (Bz*vy - By*vz);
der(y) = vy;   der(vy) = coeff * (Bx*vz - Bz*vx);
der(z) = vz;   der(vz) = coeff * (By*vx - Bx*vy);
...
end UsualEq;
```

GQLink high-level diagram



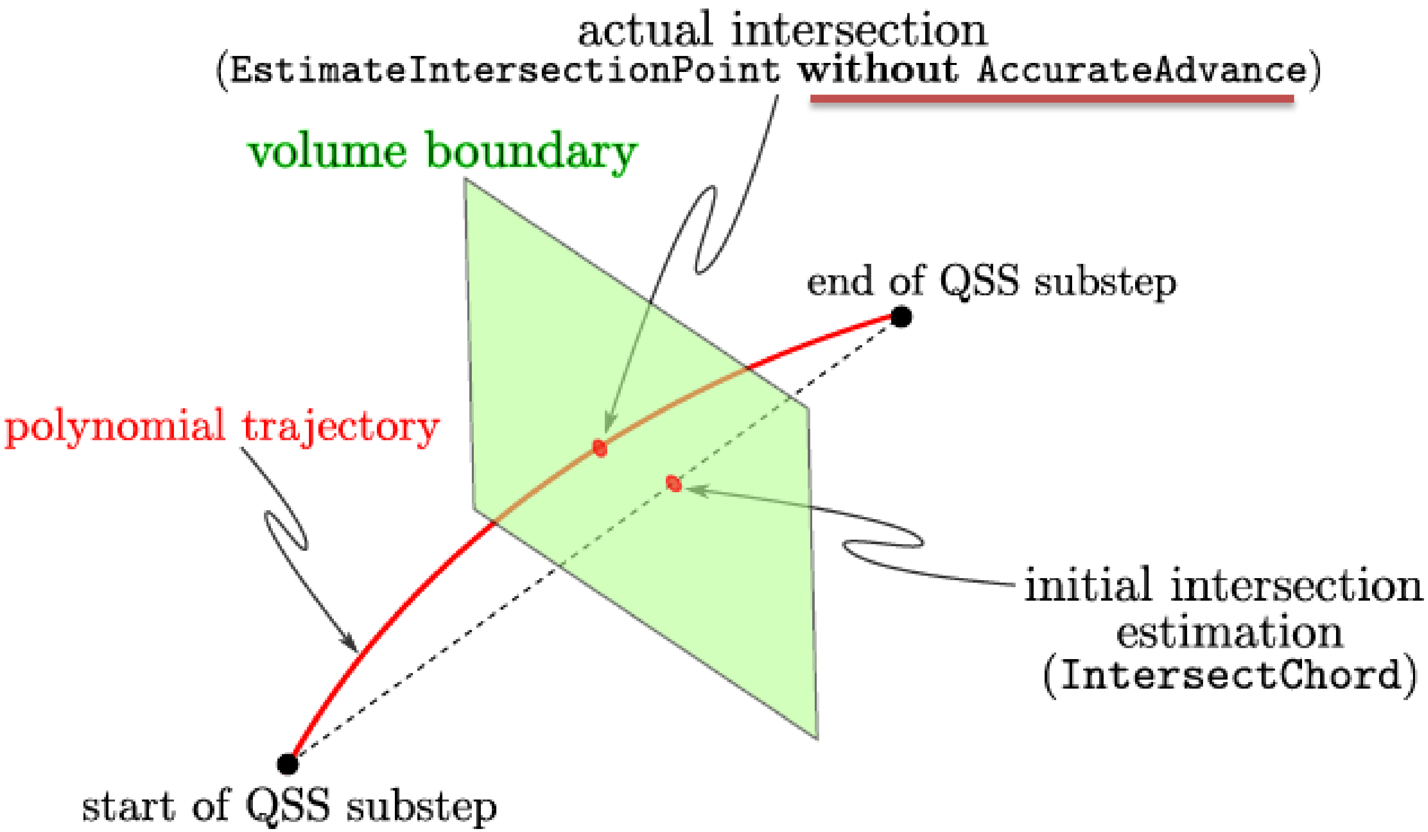
- GQLink: not another Geant4 *stepper*.
- An abstract, clean, single entry point interface to the QSS Solver family of numerical integration methods.

GQLink detailed interface



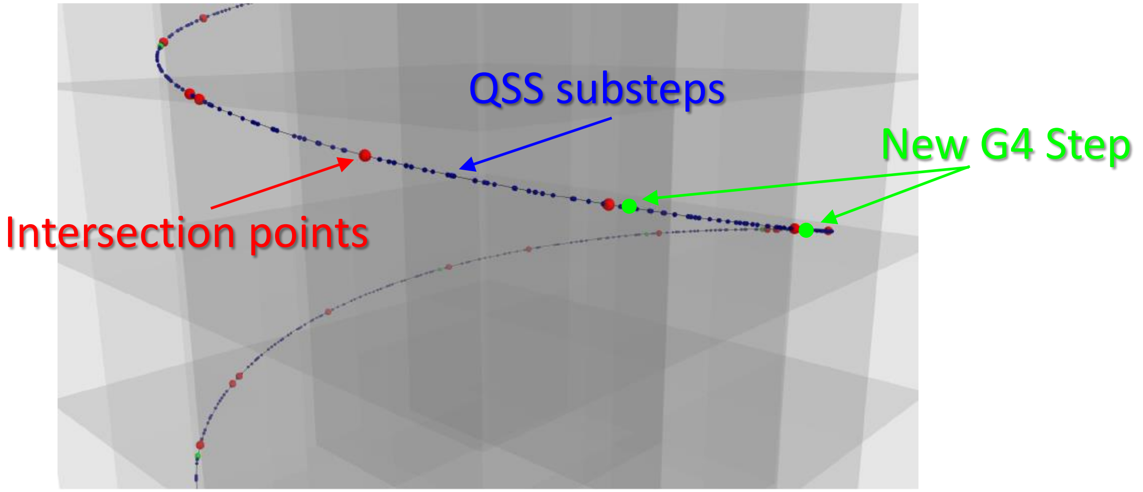
Boundary crossing

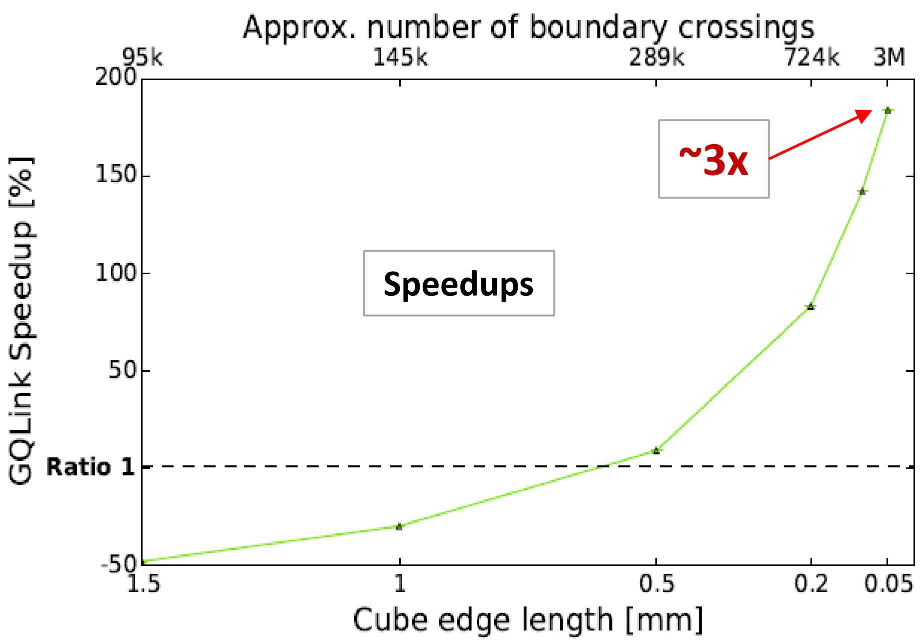
Cheaper particle transport until the crossing point using QSS polynomial dense output instead of iterative procedures:

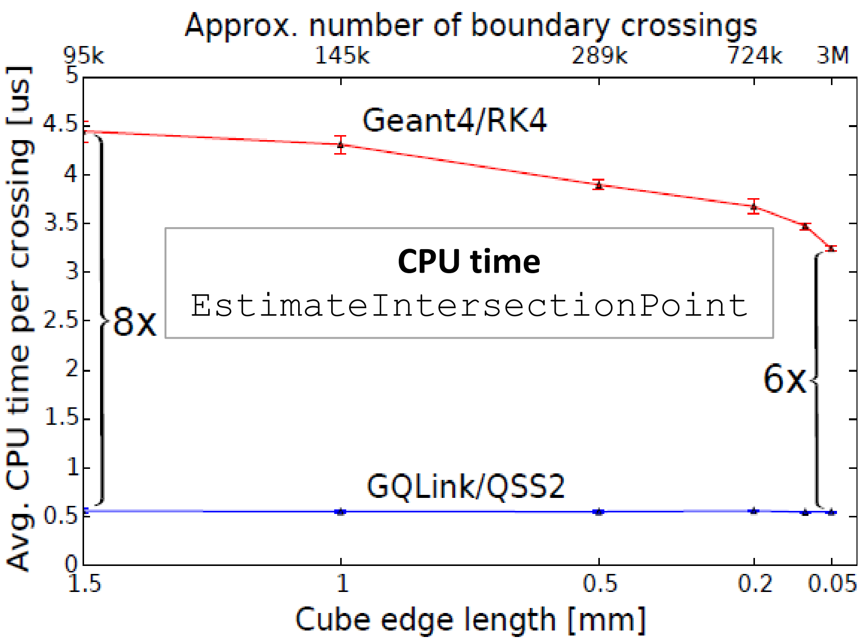


Cubes and Helix

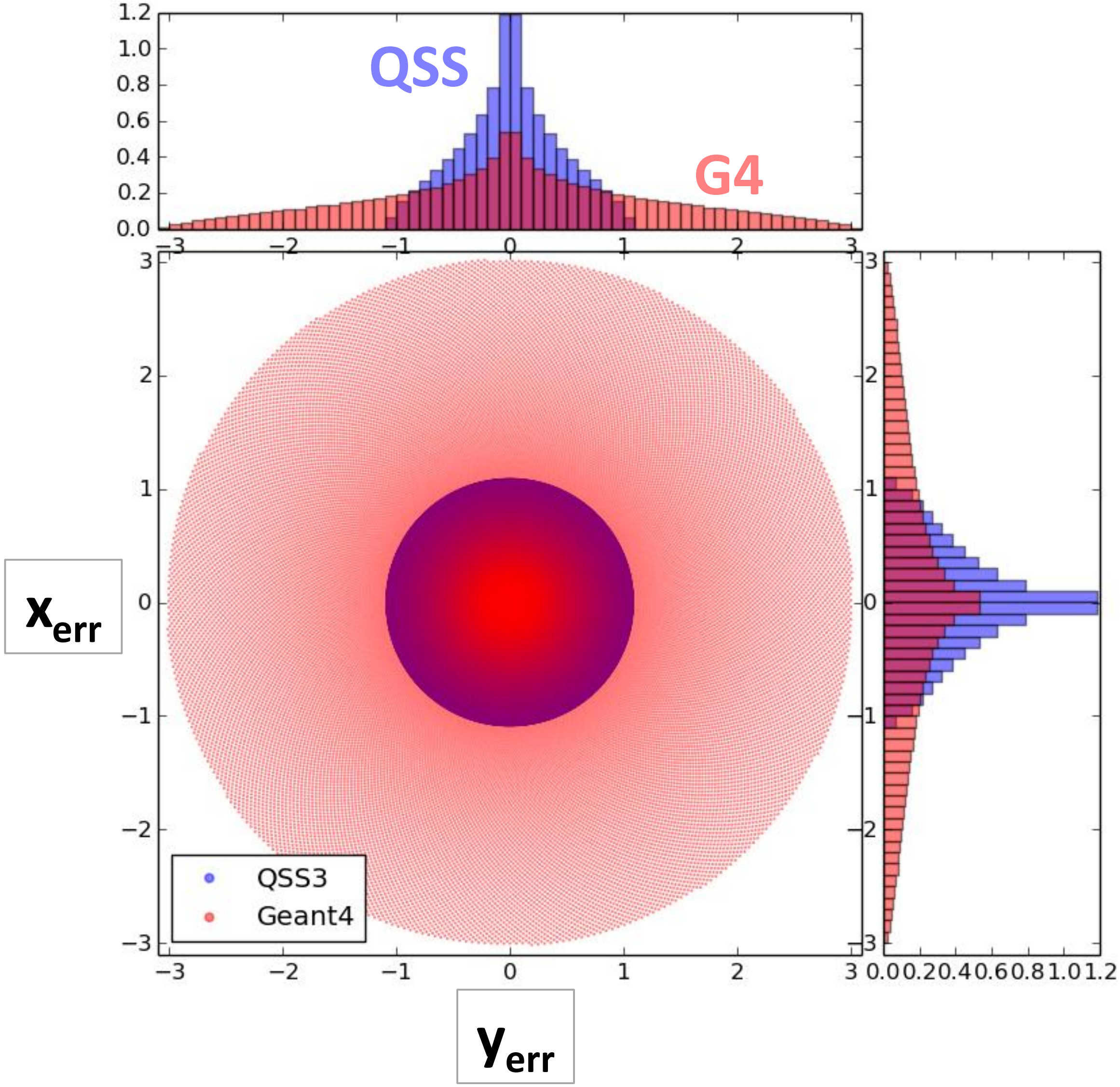
- Synthetic baseline
  - Physics off, Constant B in z
  - Harmonic 2D oscillation in x-y
  - Linear motion in z
  - Known exact solution
  - G4ClassicalRK4 (eps= 1E-5)
  - QSS2 (dQrel=1E-5)





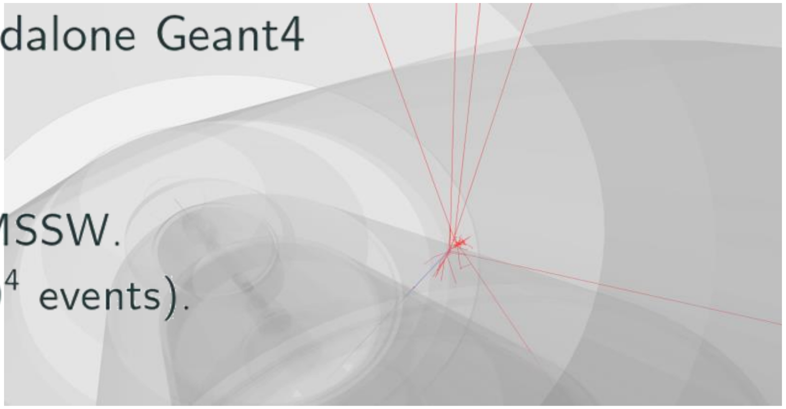


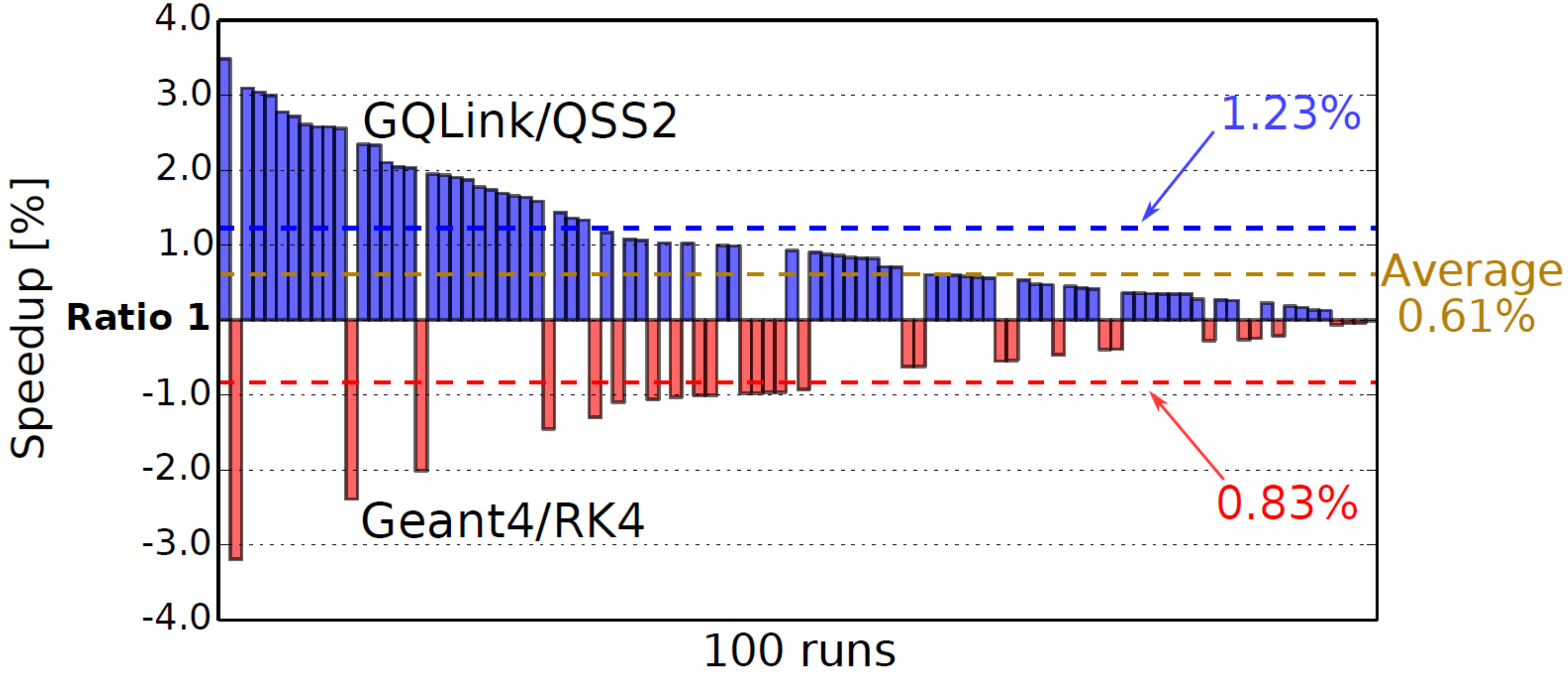
Error control



CMS – Particle gun

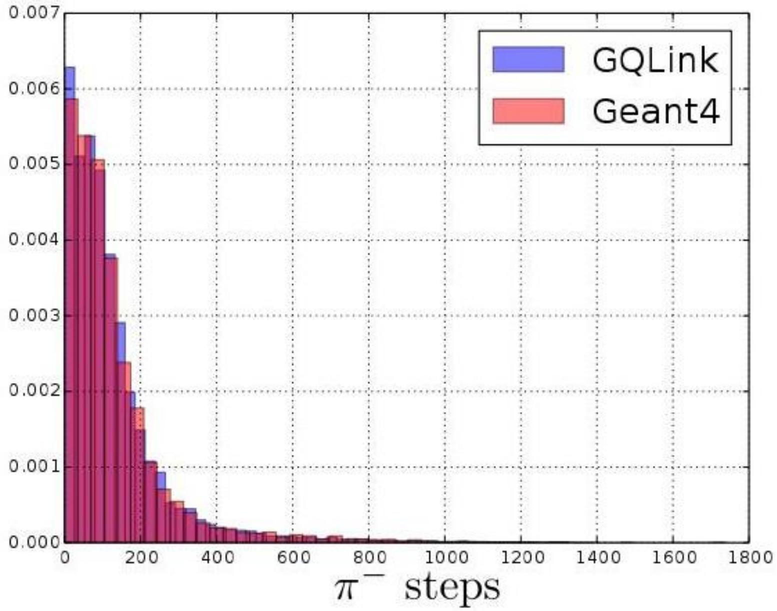
- GQLink validation was performed against a standalone Geant4 application featuring:
  - Full CMS (RunI) detector geometry.
  - Volume base magnetic field excerpted from CMSSW.
  - Particle gun shooting π^- particles (10 GeV, 10^4 events).
    - 2000 events
    - ~62000 secondaries per event

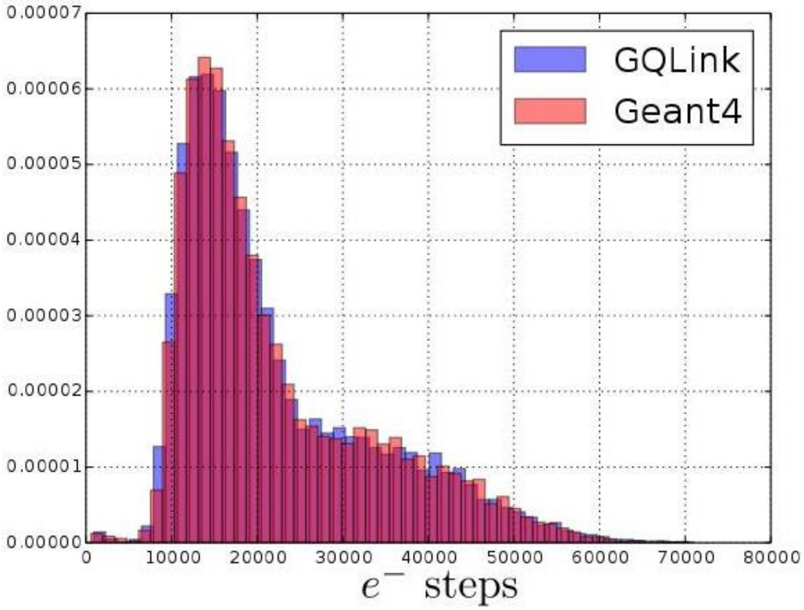




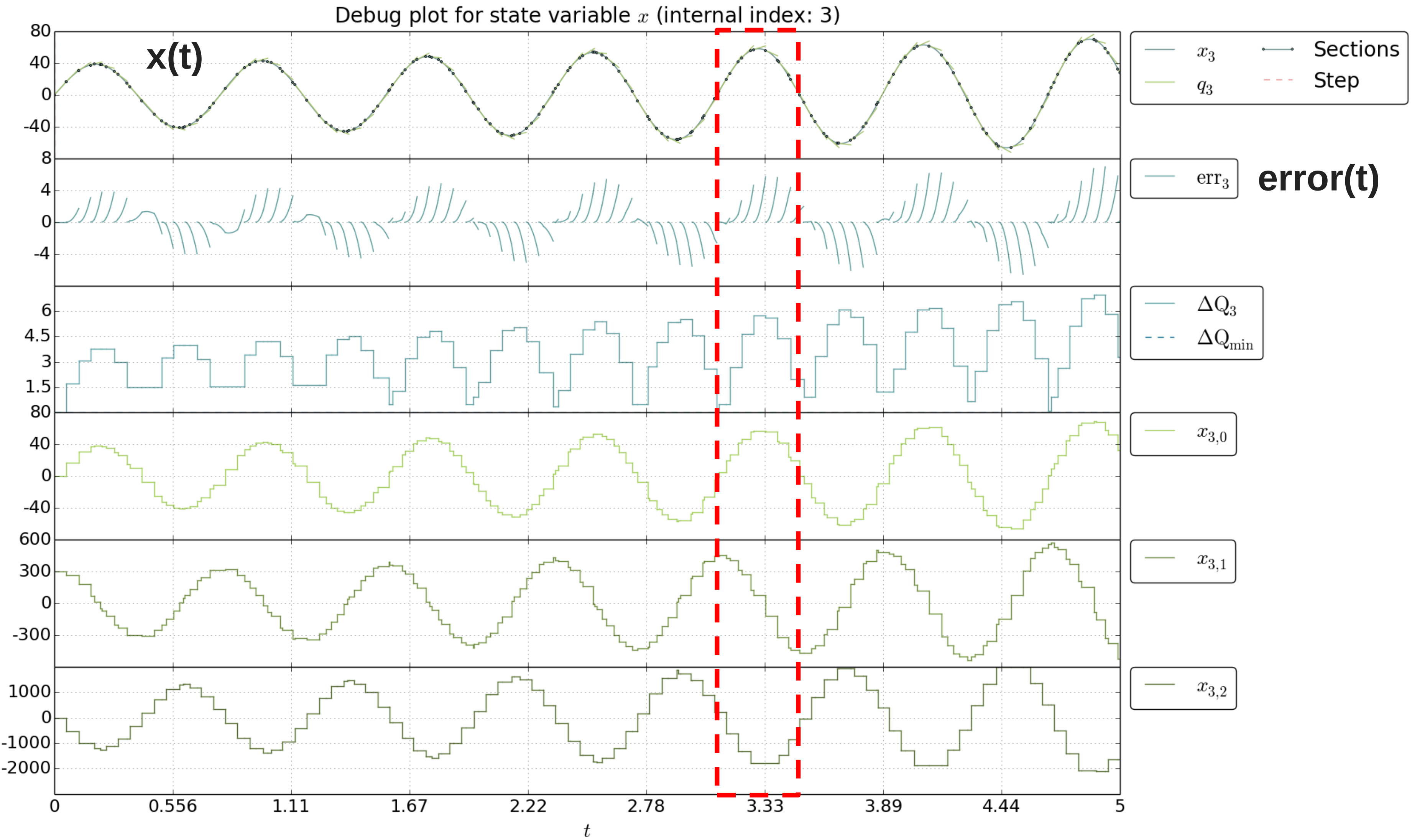
CMS Validation

- Step count distribution for π^- (left) and secondary electrons (right) for 10^4 single π^- events, showing equivalency of GQLink simulations:



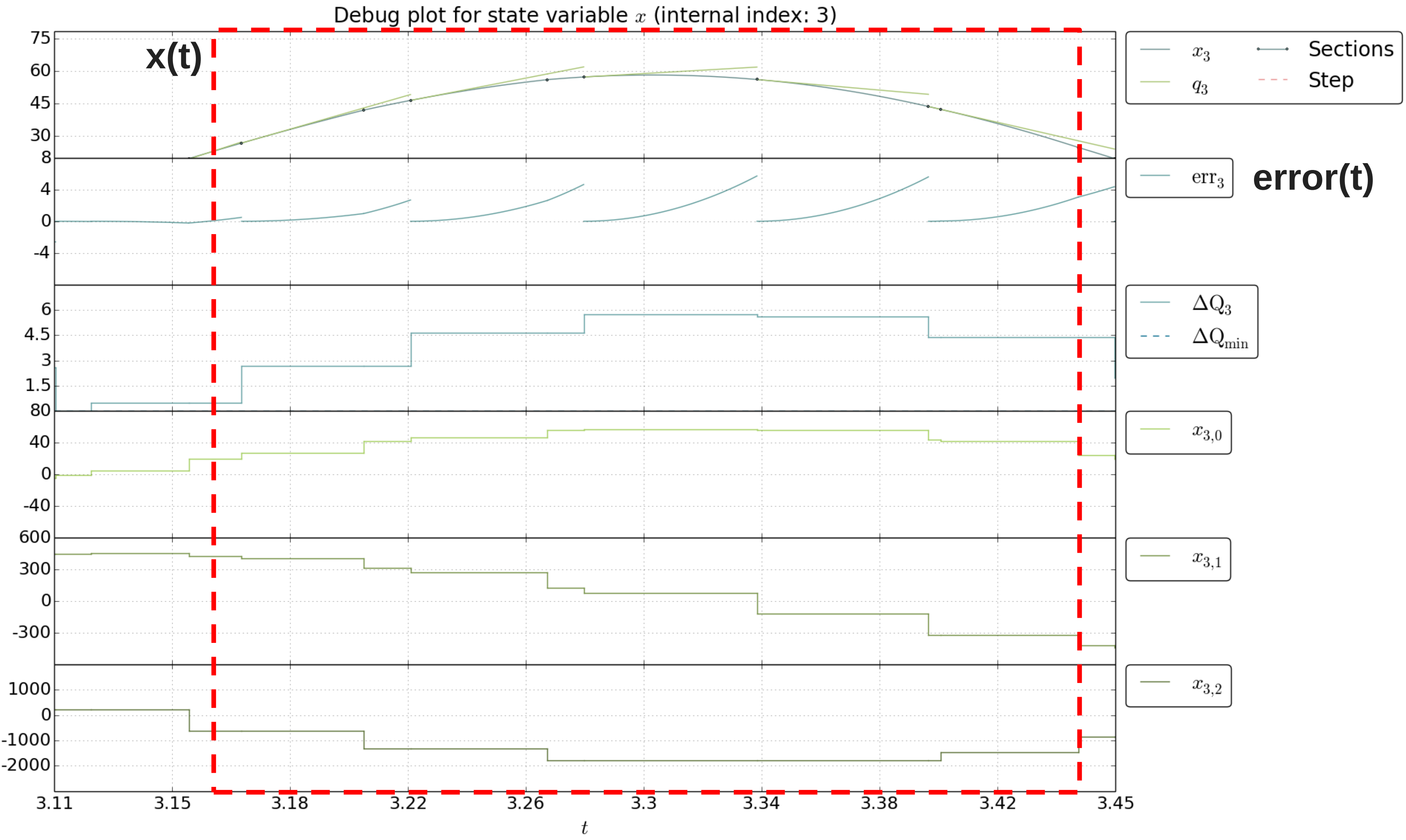


QSS Plot - details



QSS2, dQRel = 0.1, dQMin = 0.01

QSS Plot - details



QSS2, dQRel = 0.1, dQMin = 0.01

- DEVS-based Hybrid M&S technologies
  - Proved successful for dealing with effective treatment of continuous/discrete interactions
  - Proved effective to obtain speedups in applications where:
    - Continuous models present frequent discontinuities
    - Discrete models can be approximated by continuous equations
  - So far our "core technologies" were positively pushed forward driven by applications relevant to CERN
    - Network of data flows
    - Particle tracking in HEP
  - There is still huge room for performance gains

$$\delta_{\text{int}}(s) = \text{"Thanks !"}$$

rcastro@dc.uba.ar
rodrigo.daniel.castro@cern.ch