Co-Simulation Tools for Networked Control Systems

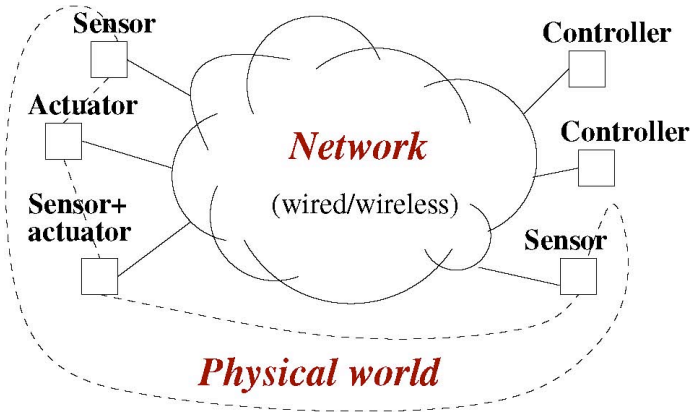
Ahmad T. Al-Hammouri[1], Michael S. Branicky[2], **and Vincenzo Liberatore[2]**

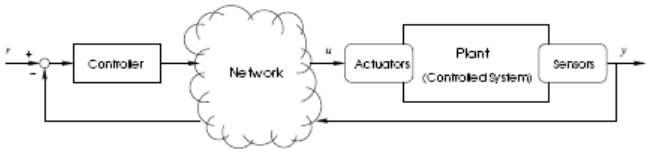
[1]Jordan University of Science & Technology, Irbid, Jordan
[2]Case Western Reserve University, Cleveland, OH, USA

HSCC'08, CPSWEEK
St. Louis, 22 April 2008

Supported in part by **NSF** CCR-0329910, **Department of Commerce** TOP 39-60-04003, and **Department of Energy** DE-FC26-06NT42853

Networked Control System (NCS)





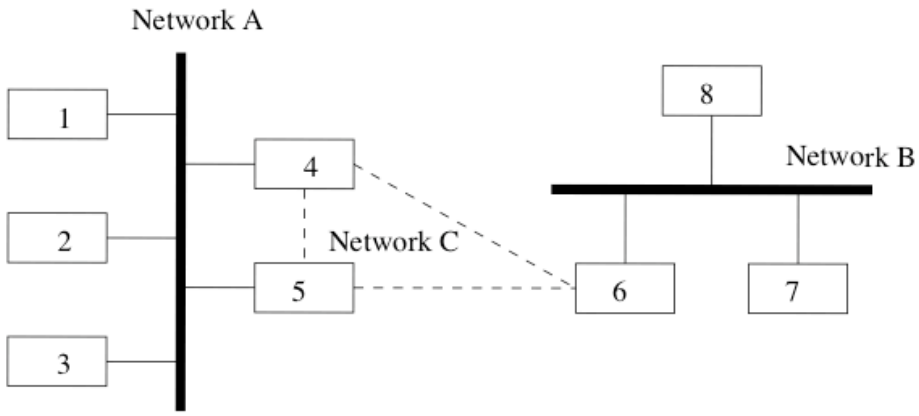


Figure 1: Multiple LANs Example: network architecture

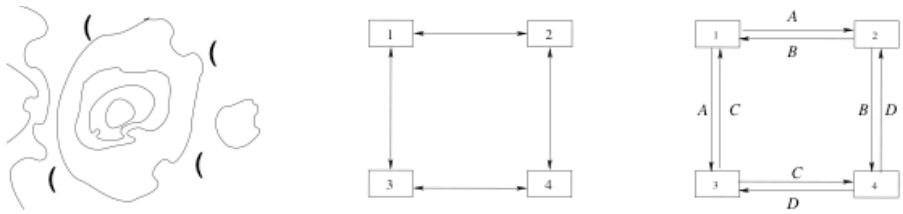
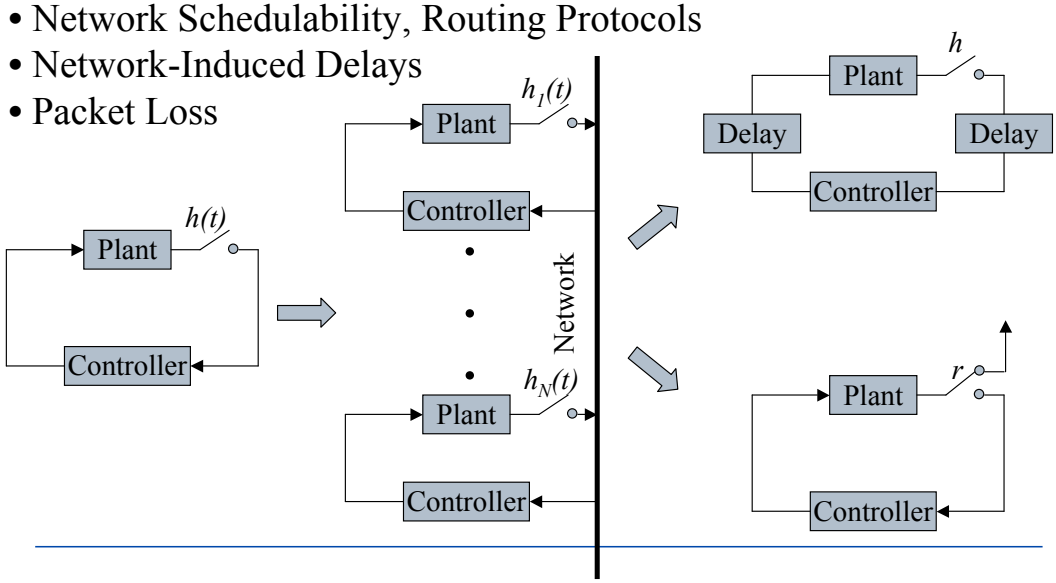


Figure 1: Formation example: physical situation (l), information flow graph (m), network multigraph (r)

Fundamental Control Issues

- Time-Varying Transmission Period
- Network Schedulability, Routing Protocols
- Network-Induced Delays
- Packet Loss

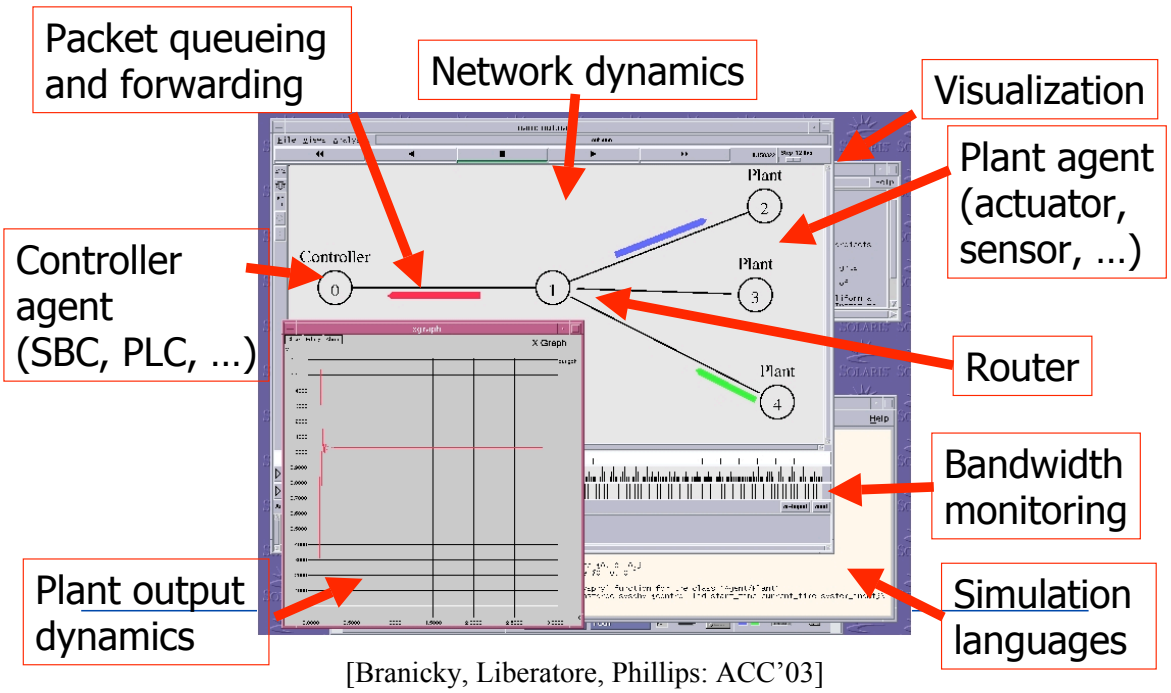


[Branicky, Phillips, Zhang: ACC'00, CSM'01, CDC'02]

Outline

- NCS Co-Simulation
- Background and Related Work
- Our Approach
  - Agent/Plant Extension
  - NSCSPlant, NSCSController Extensions
  - Modelica/ns-2 Integration

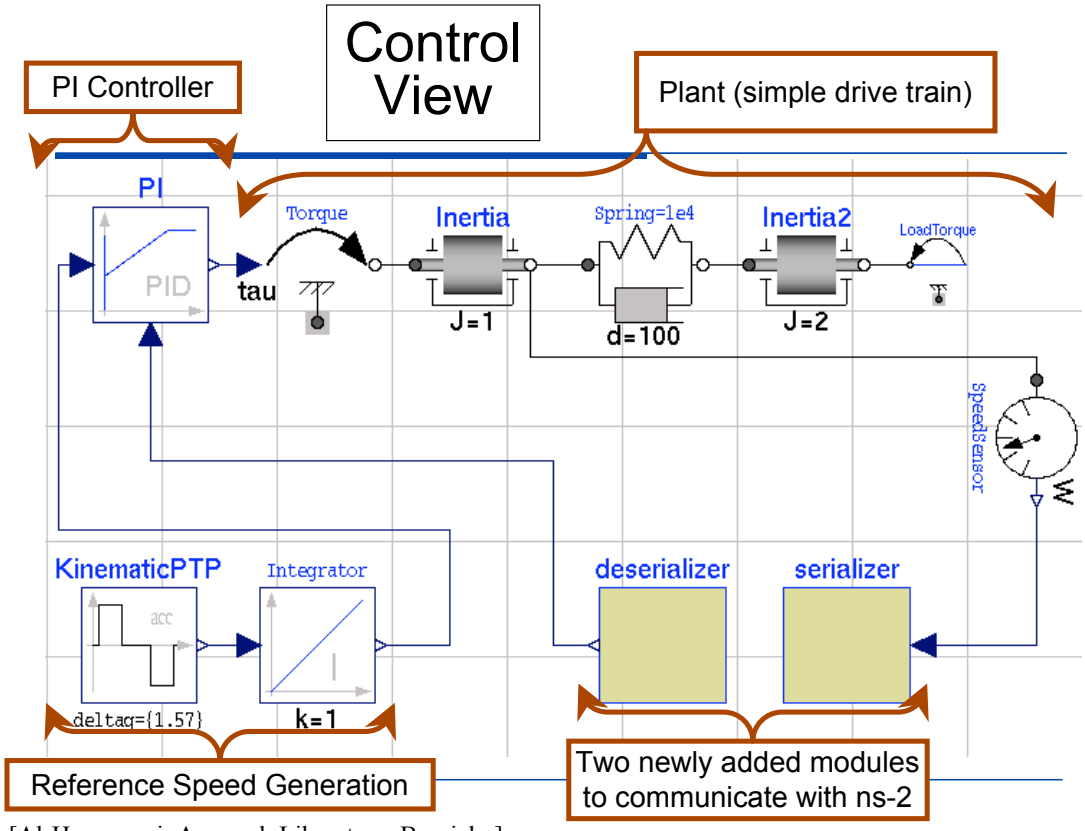
NCS Co-Simulation



[Branicky, Liberatore, Phillips: ACC'03]

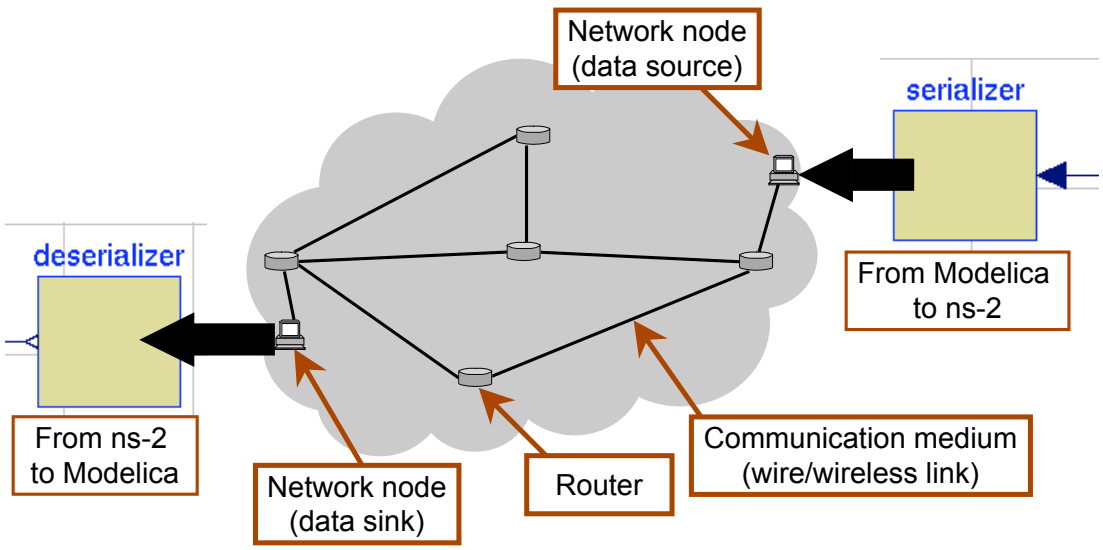
NCS Co-Simulation (2)

- Why Simulation?
  - A pure analytical approach is likely to be intricate (multiple NCSs, time-varying delays, packet losses **plus** heterogeneous networks, multiple time bases, intermitency)
- Why Co-Simulation?
  - Joint and simultaneous simulation of physical and communication networks dynamics
    - Network performance affects behavior, stability, and safety of the controlled physics
    - Demands of physical side influence the design of a communication network (e.g., bandwidth requirements)



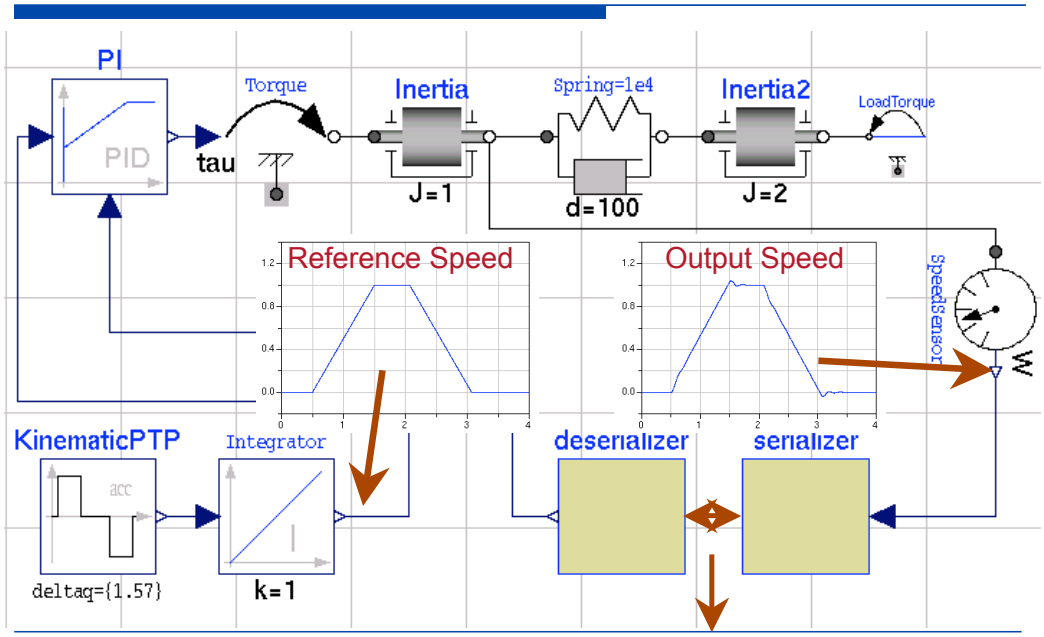
[Al-Hammouri, Agrawal, Liberatore, Branicky]

Network View



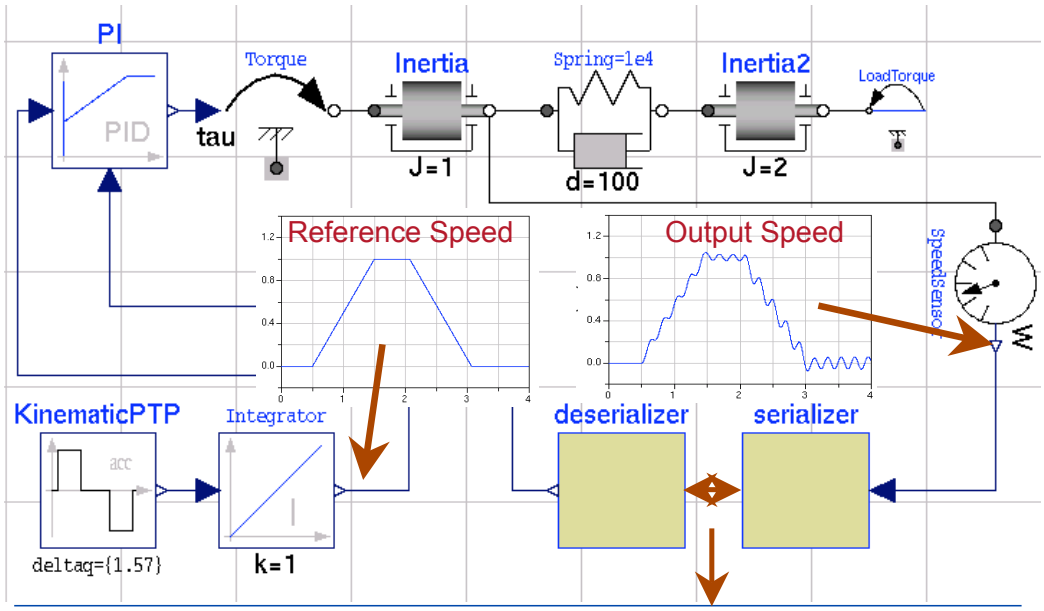
[Al-Hammouri, Agrawal, Liberatore, Branicky]

Example



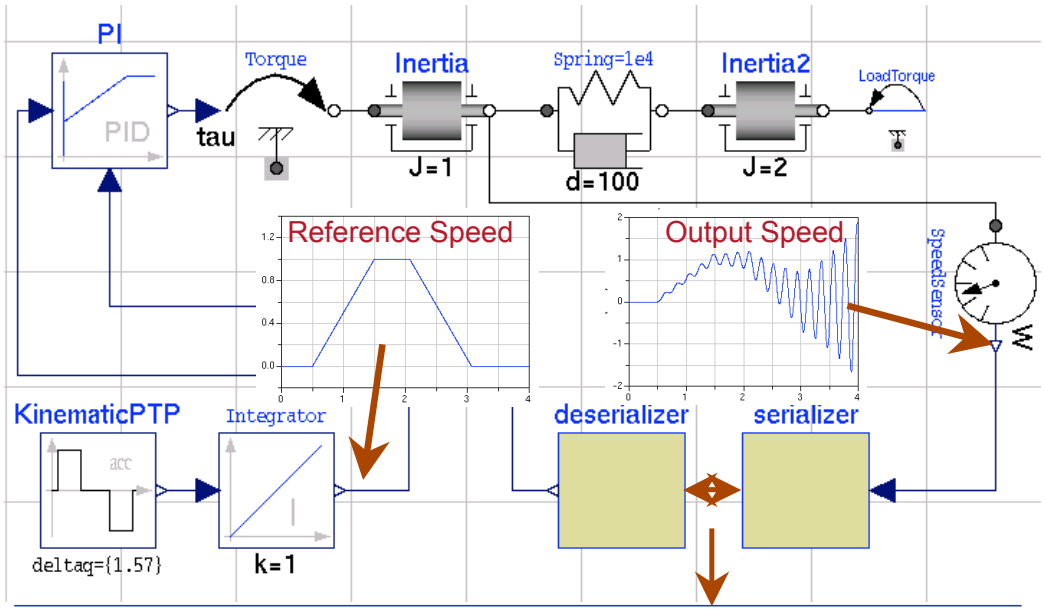
Source-to-sink network delay = 30 msec

Example (2)



Source-to-sink network delay = 42 msec

Example (3)



Source-to-sink network delay = 44 msec

NCS Co-Simulation (3)

- Co-simulation is critical to NCS research/practice
  - Allows a numerical analysis of the overall system
    - Obtain a complete snapshot of both the network behavior and the physical systems states
    - Phenomena beyond what theory can bear
  - Enables us to
    - Construct synthetic large-scale networks, workloads
    - Replay collected traces
    - Discover the design lessons
  - A tool for the verification, validation, and evaluation of different control and network algorithms

Background and Related Work

- There exist separate simulation tools for dynamical systems, communication networks
- NCS co-simulators can leverage them …
  - Extend dynamical systems simulators to also simulate the events and dynamics of communication networks. E.g.: TrueTime [Henriksson, Cervin, Arzen, *IFAC*'02]
  - Extend a network simulator to include capability for dynamical systems simulations. Example: Agent/Plant
  - Marry a full-fledged network simulator with a full-fledged dynamical systems simulator. E.g.:
    - ADEVS/ns-2 integrated tool [Nutaro *et al*., 2007]
    - Our Modelica/ns-2 integrated tool

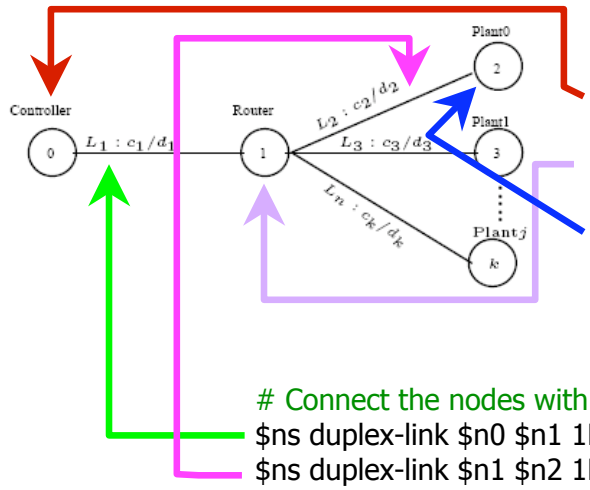
Our Approach

- Rely on and combine well-established simulation tools for networks and physical systems as much as possible
- Network side: ns-2 (`www.isi.edu/nsnam/ns/`)
  - Widespread, discrete-event, packet-level simulator
  - Routing, transport, application protocols
  - Wired, wireless, local- & wide-area networks

Our Approach (2)

- Physical systems side:
  - Agent/Plant Extension [*ACC*'03]
  - NSCSPlant, NSCSController Extensions [*INFOCOM*'05]
  - Modelica [*HSCC*'08]

ns-2 Example

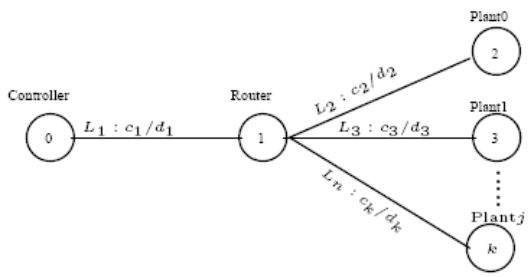



```
# Create nodes and label them
set n0 [$ns node]
$n0 label "Controller"
set n1 [$ns node]
$n1 label "Router"
set n2 [$ns node]
$n2 label "Plant0"

# n3-nk defined similarly here

# Connect the nodes with a star topology
$ns duplex-link $n0 $n1 1Mb 1ms DropTail
$ns duplex-link $n1 $n2 1Mb 1ms DropTail

# Links connecting n3-nk to n1 defined similarly here
```

Agent/Plant Extension

- ns-2 extension for physical dynamics, control laws
- Simulates the transmission of sensor and control packets between plants, controllers
- Role of sensor, controller interface, actuator (depending on simulation script)
- Any combination of time-driven, event-driven agents
- Various plant dynamics (CT/DT, linear/nonlinear)
- Various controller algorithms (e.g., P, PI)
- Dynamics solved numerically using Ode utility, “in-line” (e.g., Euler), or through calls to Matlab

Agent/Plant Demo



- Study impact of router's buffer size
- Vary # of NCSs from 1 to 39
- NCSs: scalar linear plants, P control

- Suppressed: code defining plant dynamics, control laws; NCSs 2-39

```
# Create a controller agent and
# attach it to node n0
set c0 [new Agent/Plant]
$ns attach-agent $n0 $c0

# Create a plant and attach it to
# node n2
set p0 [new Agent/Plant]
$ns attach-agent $n2 $p0

# Connect the two agents
$ns connect $c0 $p0

# Schedule events
$ns at 0.1 "$p0 sample"
$ns at 10.0 "finish"
```

Agent/Plant Demo: Results*

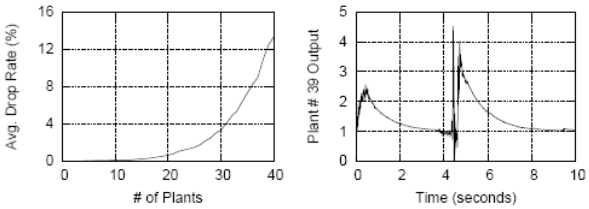
- For each NCS:
  $dx(t)/dt = ax(t) + u(t)$
  $u(t) = K(R(t) - x(t))$

a=100; K=101; R(t)=unit step; x(0)=0

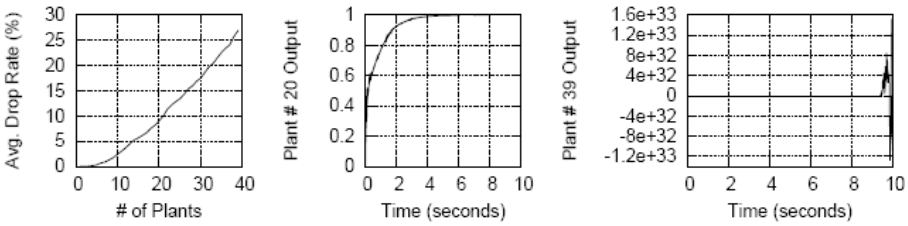
Sampling intervals ~ [5, 15] ms

- For the network:
c1 = 1.544Mbps; d1 = 1ms;
ci = 100Mbps, di = 120μs, i > 1;
packets sizes = 48B

Buffer size = 4



Buffer size = 2



* See [BLP, *ACC*'03; HBL, *ACC*'05] for more.

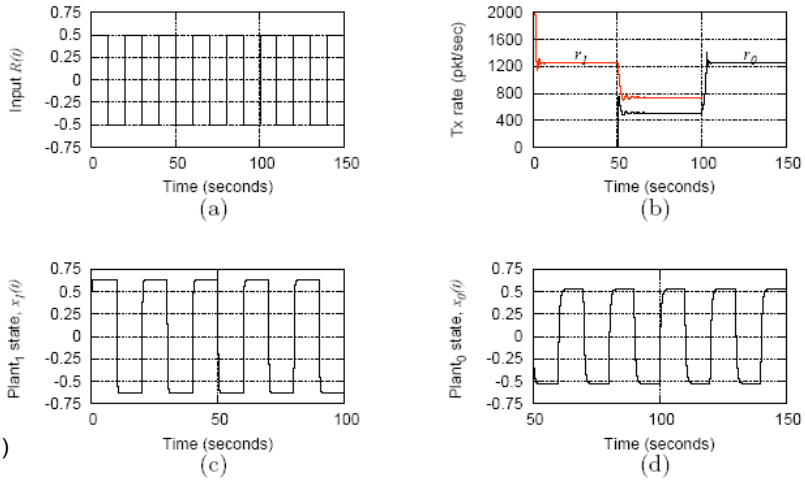
NSCSPlant and NSCSController Extensions

- New ns-2 agents (based on Agent/Plant framework)
- NSCSPlant
  - An abstract agent class
  - Used to instantiate several controlled systems, each of which simulates a physical system
  - Supports adaptive sampling policies

    E.g., sampling rate is adjusted based on a utility (performance) function associated with each plant
- NSCSController
  - Used to instantiate controllers

NSCSPlant and NSCSController Demo*

- For ncs0:
  a0 = 0.1; K0 = 2.0
  ts = 50 s; te = 150s
- For ncs1:
  a1 = 1.0; K1 = 5.0
  ts = 0 s; te = 100 s
- For the network:
  c1 = 1.0 Mbps
  d1 = 5.0 ms
  c2 = c3 = 10.0 Mbps
  d2 = d3 = 1.0ms
  packet size = 100B
- Ui(ri)=[(ai-Ki)/ai]exp(ai/ri)



* See [*INFOCOM*'05, *WPDRTS*'06, *ACM ENSS*'06] for more

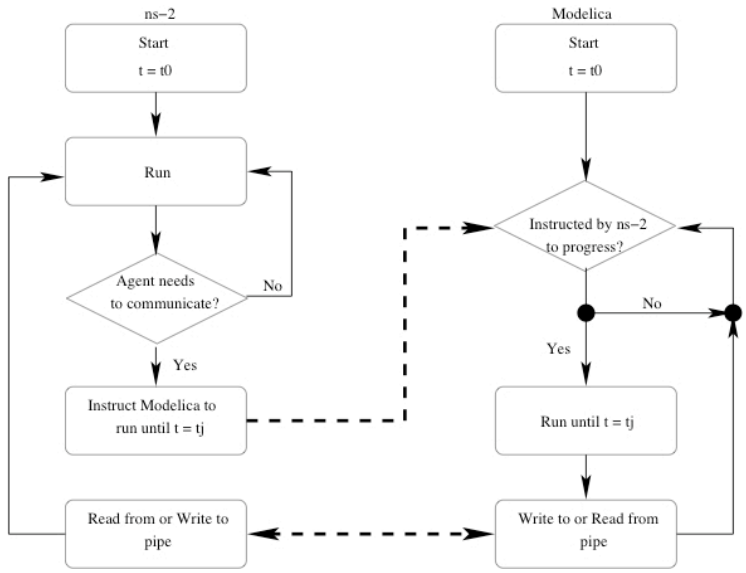
Modelica/ns-2 Integration

- A tool that combines the best features of the individual simulators
- Modelica (`www.modelica.org`)
  - Modeling language for large-scale physical systems
  - Supports model construction & reusability (O-O)
  - Allows acausal modeling
  - Many libraries (power train, power systems, …)
  - Commercial, open source simulation environments

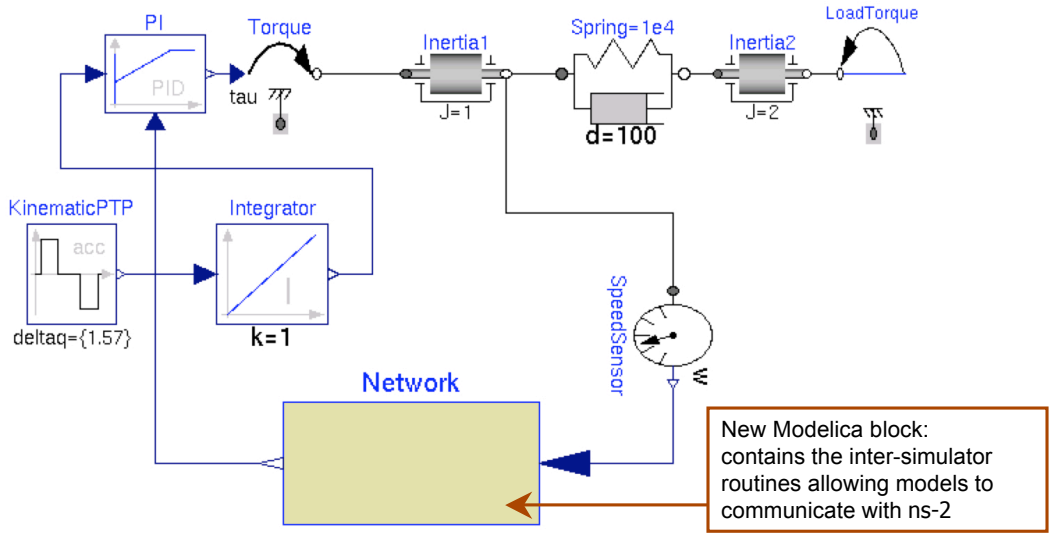
Modelica/ns-2 Integration (2)

- Technical Approach
  - Integration via newly developed inter-process communication interfaces in both Modelica & ns-2
  - IPC mechanics must guarantee synchronization between the two simulators
  - Synchronization is achieved by *enslaving* each Modelica module to a corresponding ns-2 agent

Modelica/ns-2 Integration (3)

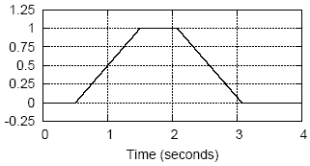


Modelica/ns-2 Integration Demo 1

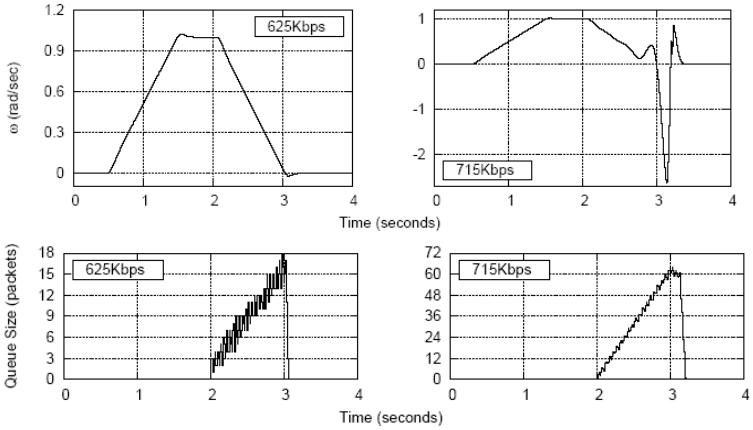


Modelica/ns-2 Integration Demo 1: Results

- Sensor packets share the same link with a multimedia application
- Network parameters as before
- Speed is sampled every 2ms

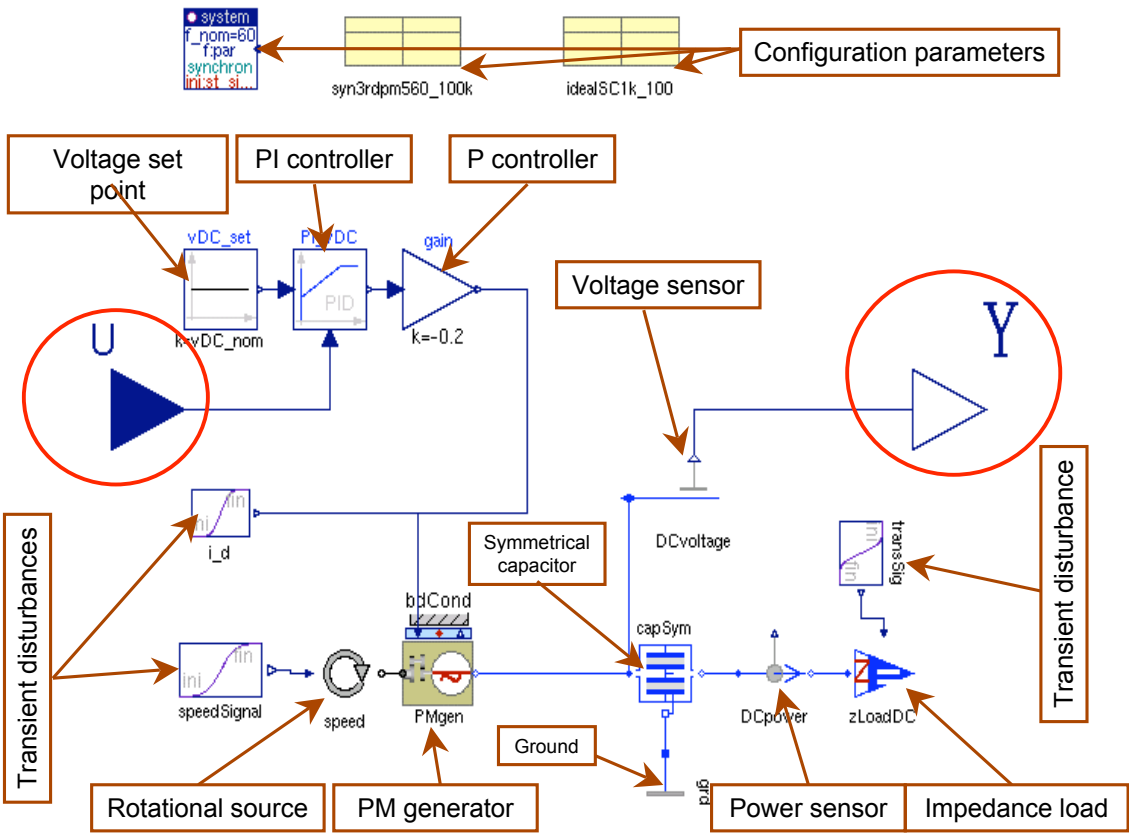


- For Multimedia App, packets = 1000B; Tx rate 625 & 715 Kbps; ts = 2s; te = 3s

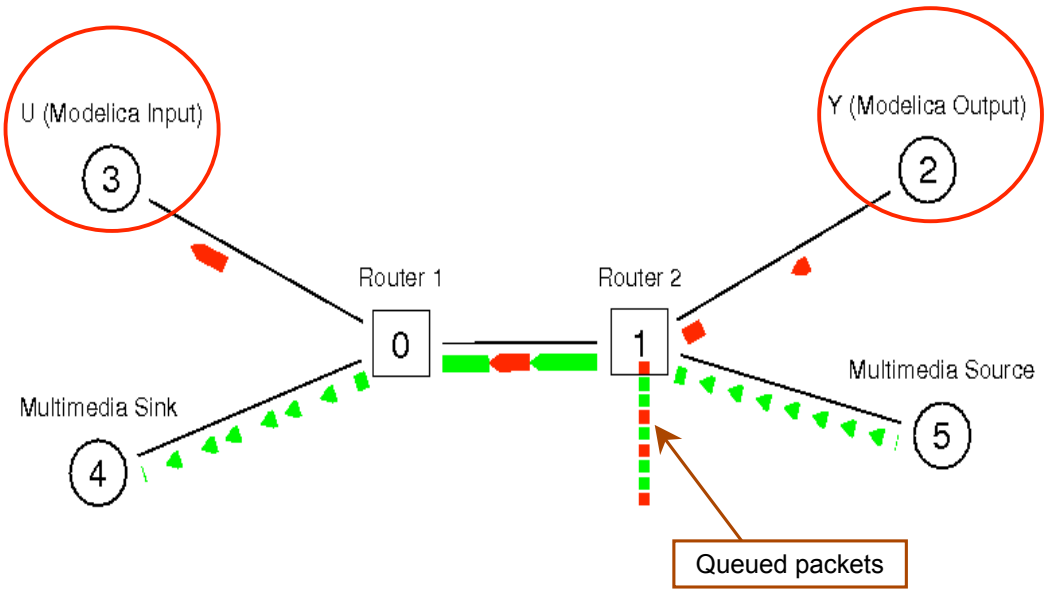


Modelica/ns-2 Integration Demo 2

- NCS consisting of a power transmission system controlled over a wide-area network
- The power system involved typical elements:
  - A permanent-magnet generator
  - A transmission line
  - Time-varying loads
- PID controller to regulate generator's output voltage under varying load
  - Using output measurements sent over the network

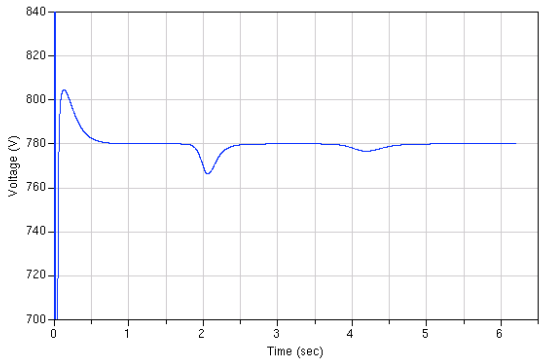


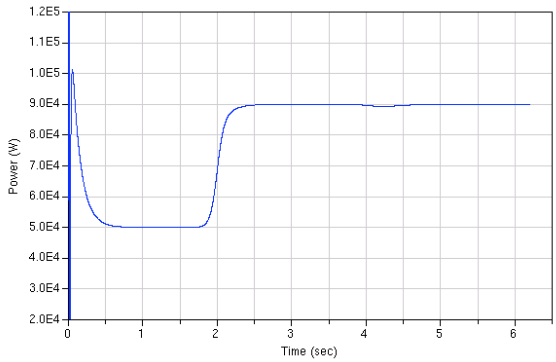
Modelica/ns-2 Integration Demo 2 (ns-2 View)



Modelica/ns-2 Integration Demo 2: Results

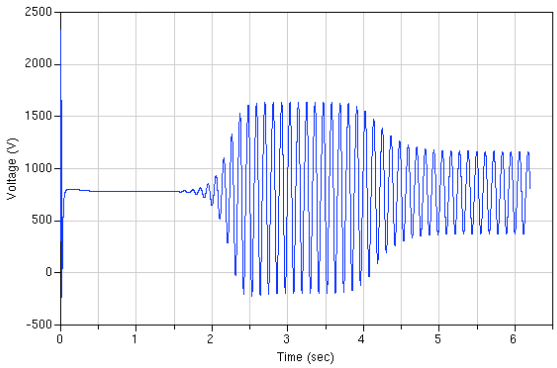
- Multimedia application consumes only ***0.25*** Mbps of the 3 Mbps bottleneck link

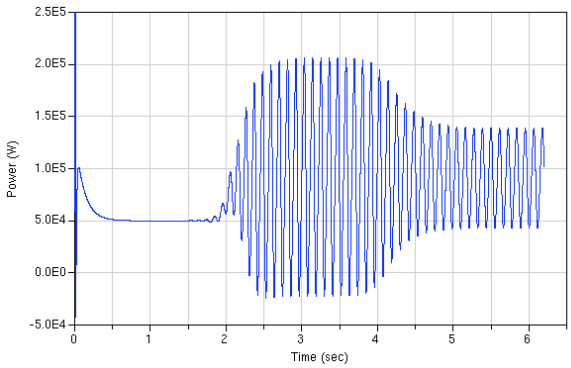




Modelica/ns-2 Integration Demo 2: Results (2)

- Multimedia application consumes ***2.5*** Mbps of the 3 Mbps bottleneck link





Conclusion

- NCS Definition
- NCS Co-Simulation
- 3 Instantiations
  - Agent/Plant Extension
  - NSCSPlant, NSCSController Extensions
  - Modelica/ns-2 Integration

Selected References

- Zhang, Branicky, Phillips: Stability of networked control systems. IEEE Control Systems Magazine 21(1): 84–99, 2001
- Modelica and Modelica Association. http://www.modelica.org
- Network Simulator—ns-2. http://www.isi.edu/nsnam/ns
- ADEVS: A Discrete EVent System simulator. http://www.ornl.gov/1qn/adevs/index.html
- Cervin, Ohlin, Henriksson: Simulation of networked control systems using TrueTime. Intl. Workshop Networked Control Systems: Tolerant to Faults, Nancy, France, 2007
- Nutaro, Kuruganti, Miller, Mullen, Shankar: Integrated hybrid-simulation of electric power and communications systems. IEEE Power Engineering Society General Meeting, 1–8, 2007
- Liberatore: Agent/Plant extension. http://vorlon.case.edu/vxl11/NetBots
- Hartman, Branicky, Liberatore: Time-dependent dynamics in networked sensing and control. Proc. American Control Conf., Portland, 2005
- Al-Hammouri, Liberatore, Al-Omari, Al-Qudah, Branicky, Agrawal: A co-simulation platform for actuator networks. ACM Conf. Embedded Networked Sensor Systems, Sydney, 2007

Networked Control Systems: Results

Stability and Control of Networked Control Systems

- Fundamental Issues, Experiments, Stability Regions [ZBP01*]
- Multiple Lyapunov Functions for Stability [BPZ00, 01]
- Control Strategies [HBL05, ABL08]
- Decentralized Bandwidth Allocation [ABLP06, ALBP06]

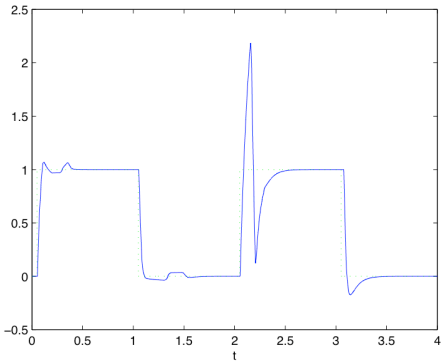
NCS Co-Design and Co-Simulation Tools

- Scheduling and feedback co-design [BPZ02]
- Co-simulation via ODE extensions to ns-2 [BLP03, 04]
- Co-simulation by combining Modelica and ns-2 [ABL08]

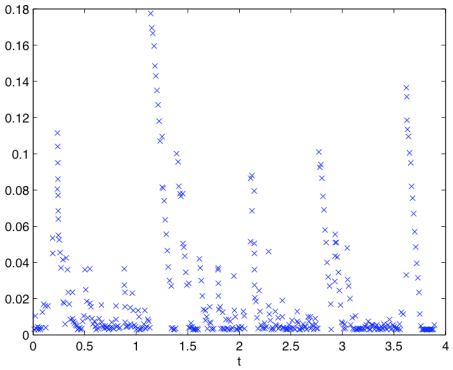
*Google Scholar and ISI Web of Science: highest-cited publication (relevant to the field) in a keyword/topic search for “networked control systems”

A Quick Example: PID NCS

[simulated in TrueTime; Henriksson, Cervin, Arzen, IFAC’02]



Step responses of plant
- First-order plant (time-driven)
- PI controller (event-driven)
- Connected by a network
- Interfering traffic (48% of BW)



Corresponding round-trip times (s)