Modeling Power Systems of the Future

Information Technology and Power System Dynamics

James Nutaro, Phani Teja Kuruganti, Vladimir Protopopescu, and Mallikarjun Shankar,

Computational Sciences and Engineering Division

Oak Ridge National Laboratory

nutarojj@ornl.gov

Two Fallacies about Networked Systems

- **The communication network can be characterized independently of the load that is offered to it.**
  - "[the probability of packet being successfully delivered is modeled by a] Bernoulli random variable…", a para-phrasing from *Foundations of Control and Estimation Over Lossy Networks* by Schenato, *et. al.* which appeared in Proceedings of the IEEE, Vol. 95, No. 1, January 2007.

- **The offered load can be characterized independently of the communication network.**
  - "In this paper, a single server G/D/1 queuing system with infinite buffer is simulated with the consideration of three input traffic sources: exponential, weibull, and normal distributions." from the abstract of *Network Traffic Characterization for High-Speed Networks Supporting Multimedia* by Elleithy andAl-Suwaivan which appeared in Proceedings of the 34th Annual Simulation Symposium, 2001.

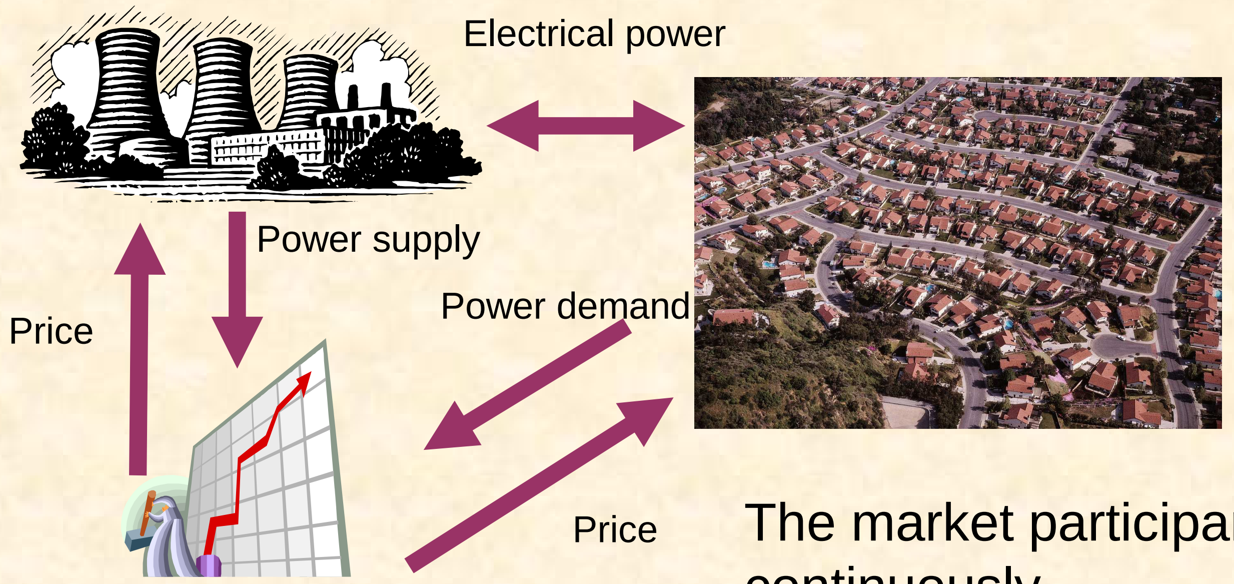
Why do these fallacies persist?

- **The mathematical foundations of control systems and communication networks evolved, and continue to evolve, independently of each other.**
- **Simulation tools that readily and simultaneously accommodate models of complicated plants, controllers, and communication networks are not available.**

Two Examples of Complicated, Hybrid Dynamics

- The impact of discrete, delayed price signals on the stability of a real-time market for electric power.
- The interplay of the dynamics of electro-mechanical and communication sub-systems in system for automatic load shedding and restoration.

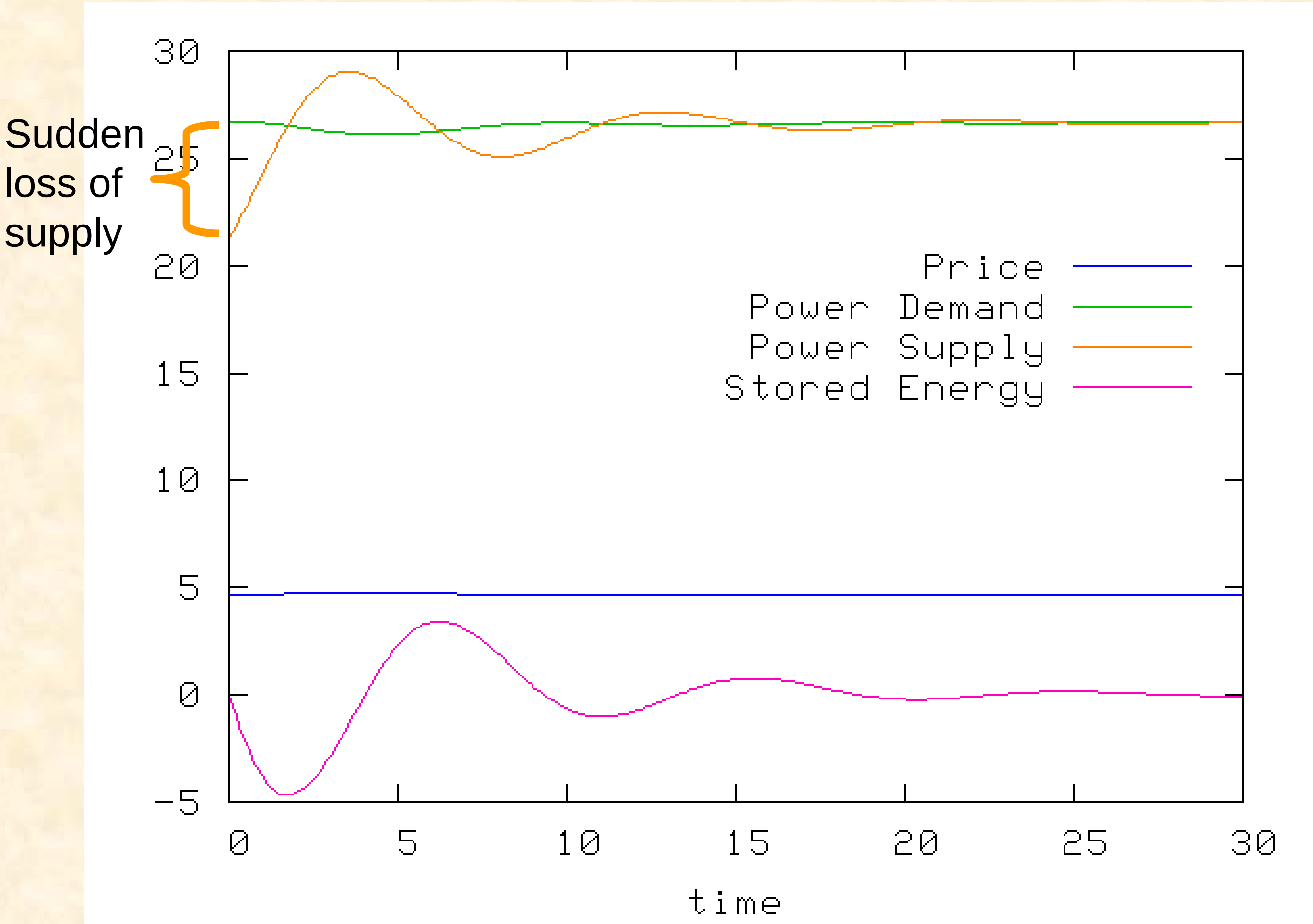
Market Stability: Continuous view



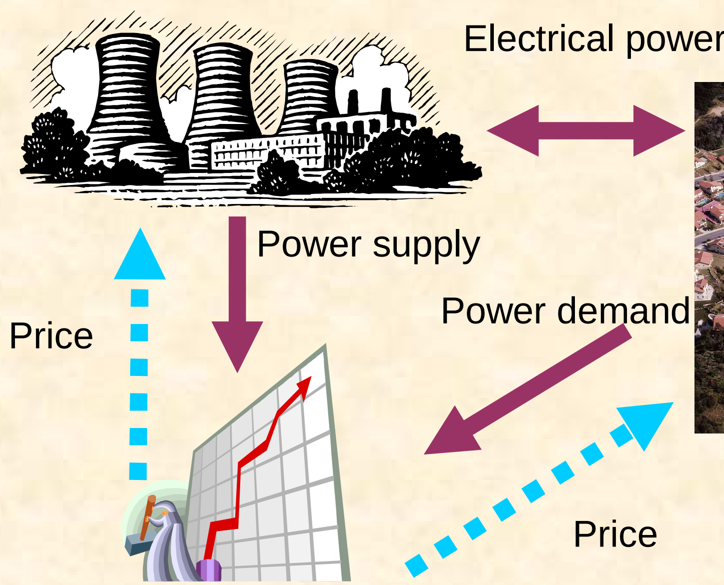
The market participants interact continuously.

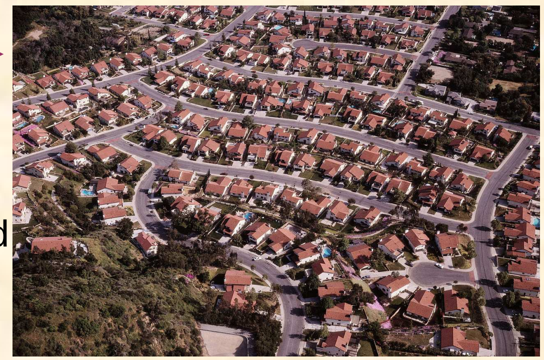
Stability depends, essentially, on how much the supplier pays for brief mismatches in supply and demand.

Continuous market following a loss of generation



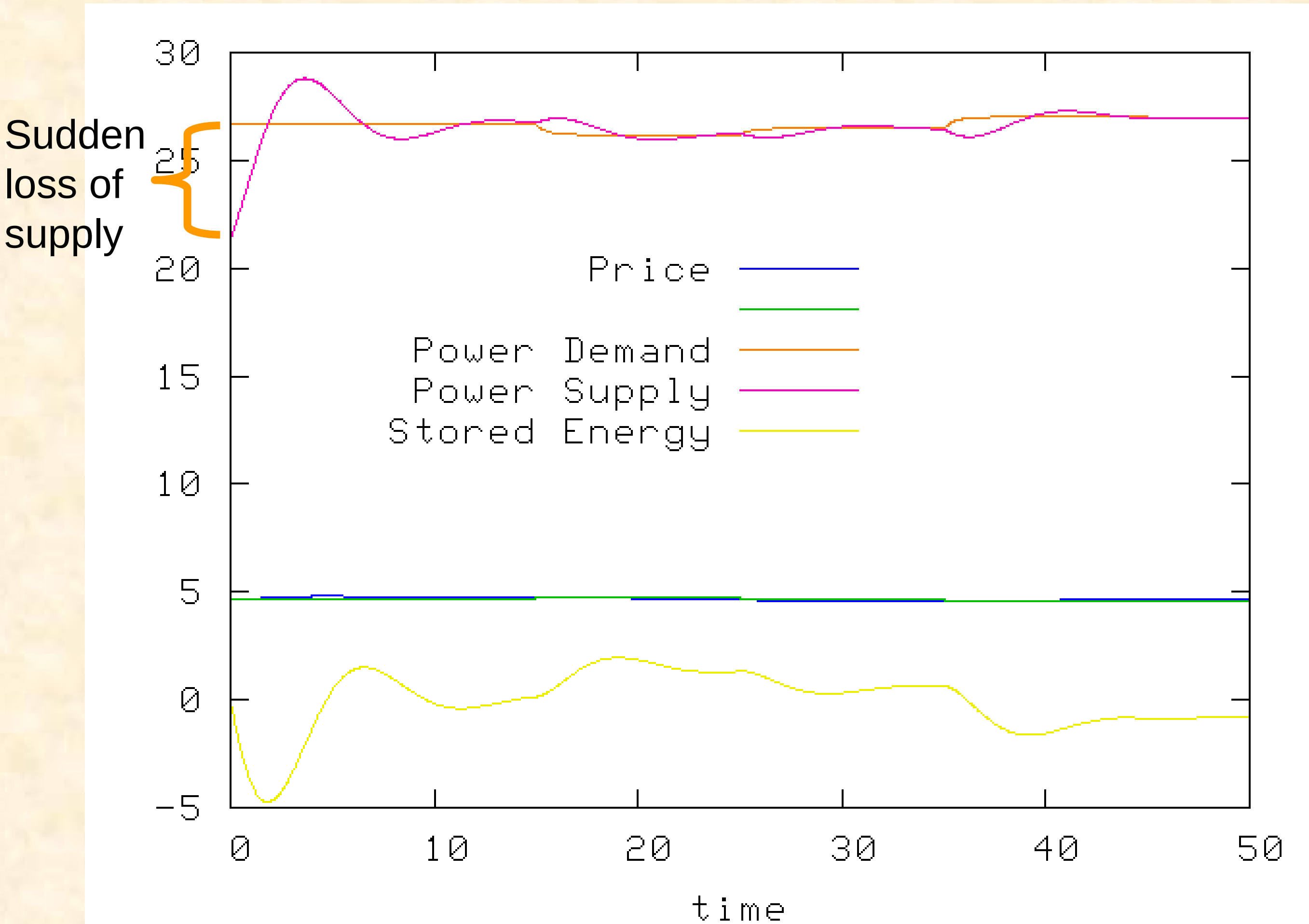
Market Stability: Hybrid view





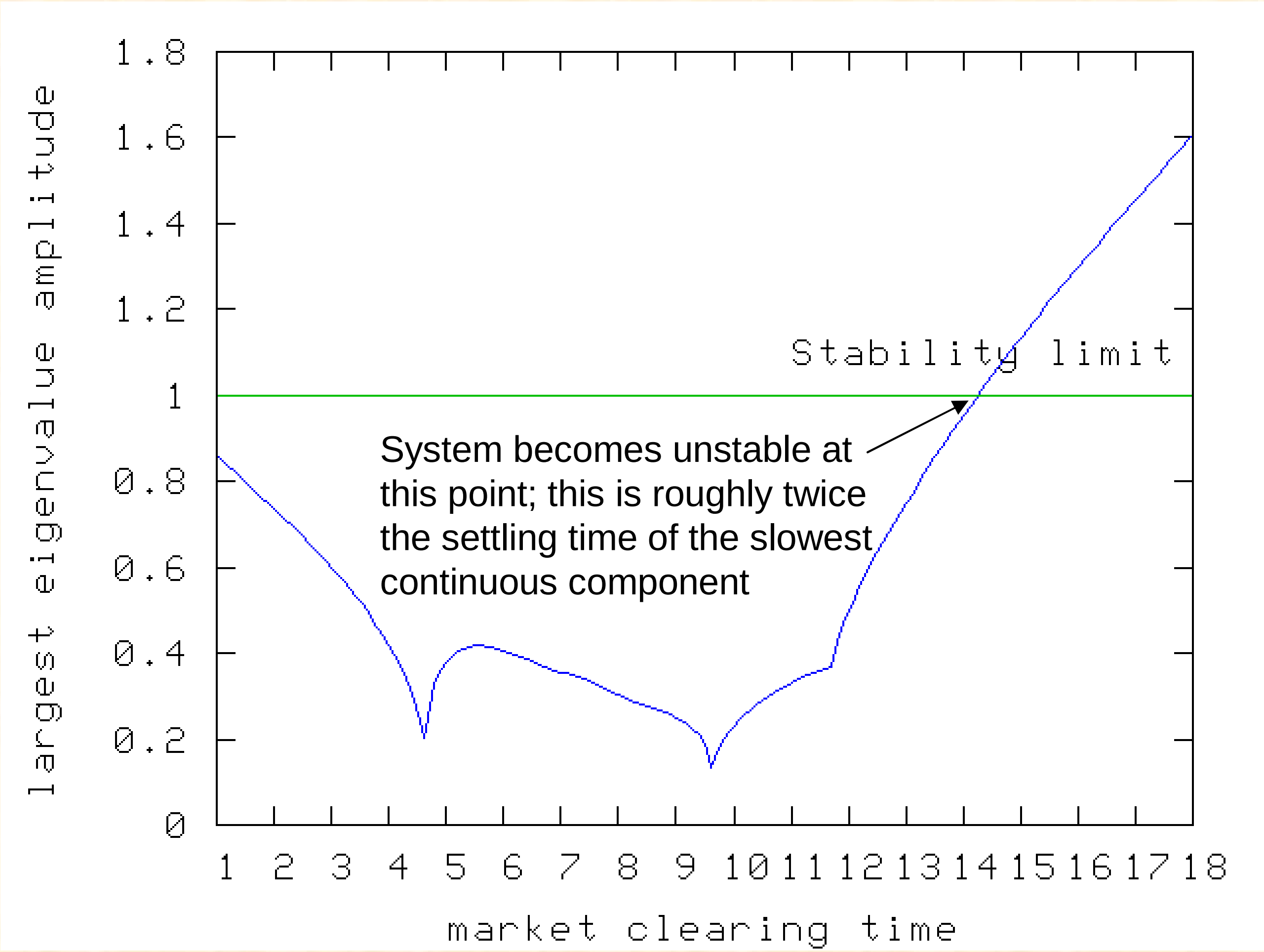
The market participants interact through a market that issues discrete price signals. The price signals experience a transmission and processing delay.

Discrete market following a loss of generation



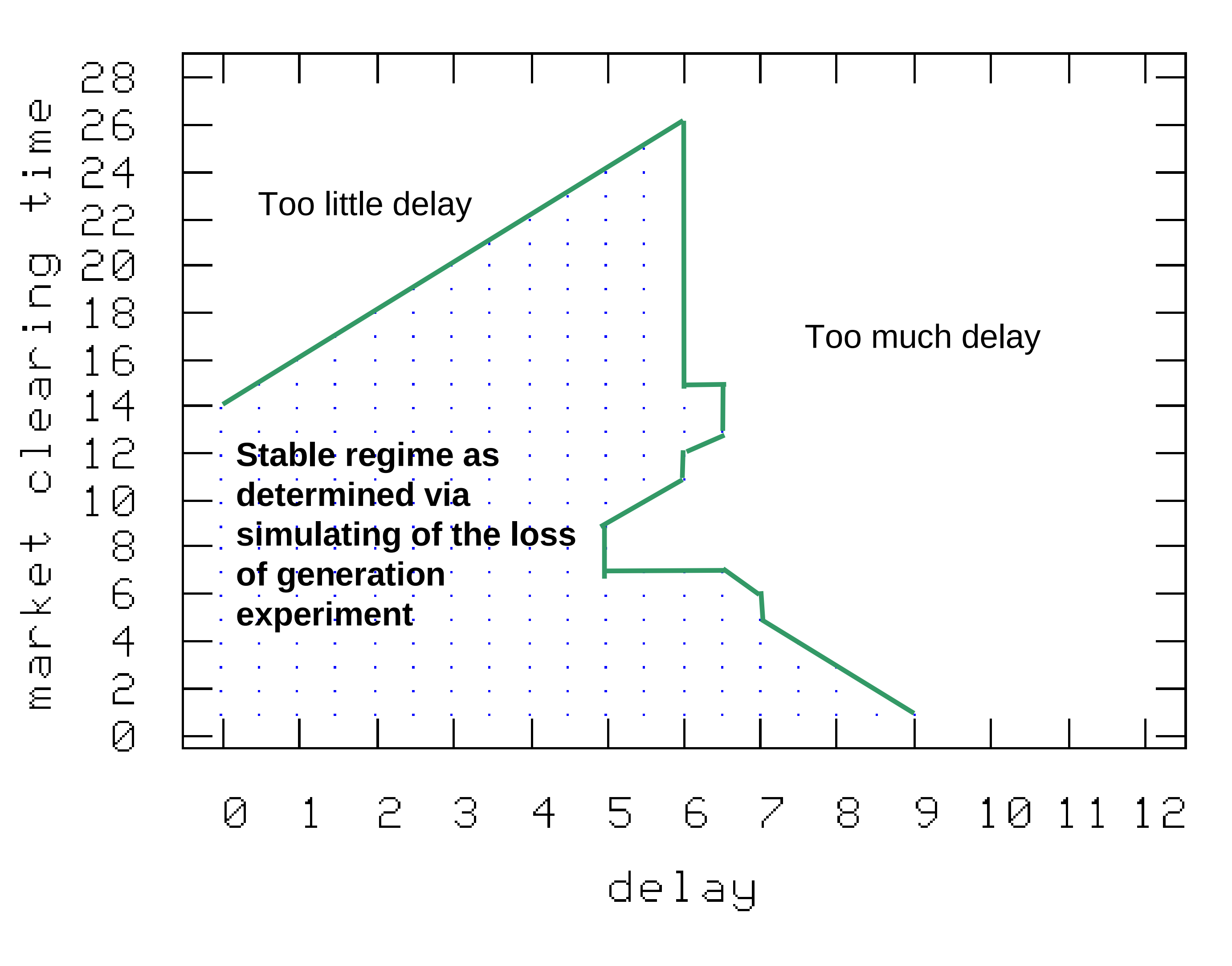
delay is 5 units, market clear time is 10 units

Market Stability Due Market Clearing Time Alone

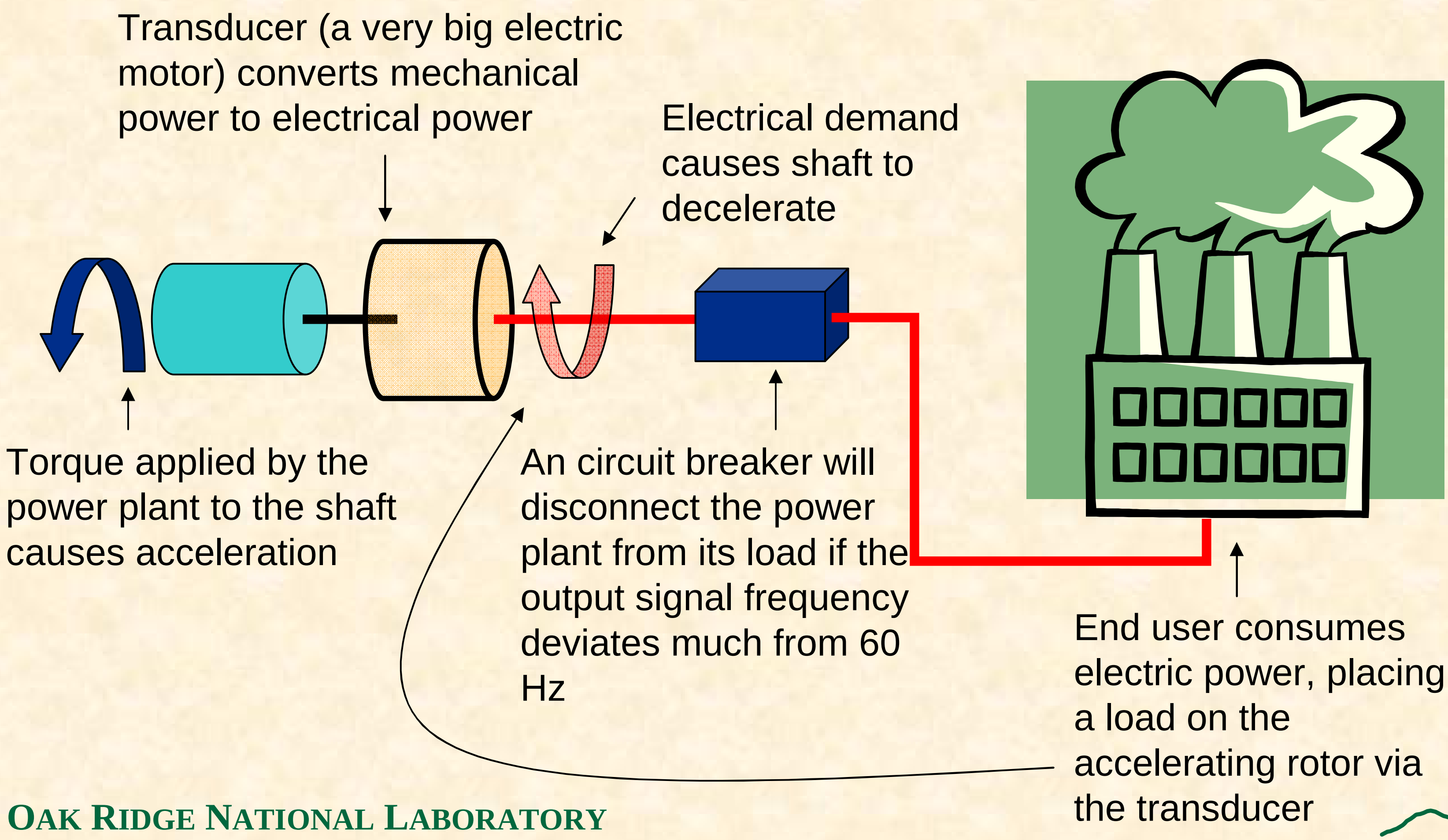


In the absence of delay, there is an upper limit on the tolerable market clearing time.

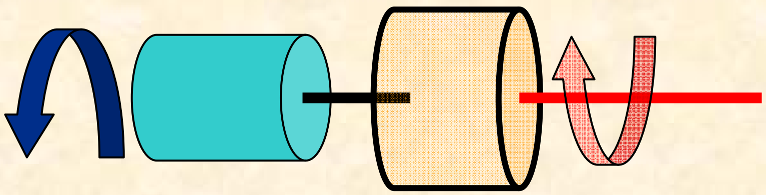
Impact of the Combined Market Clearing Time and Delay



Generator disconnect due to frequency excursions



Cause of frequency related failures



Power plants are large machines with a lot of inertia – they need time to adjust the amount of torque applied to the rotor

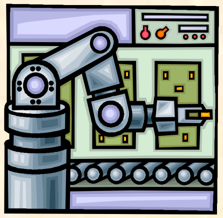
Industrial loads can switch large electrical loads on and off quickly

Frequency changes when there is difference between electrical load and mechanical power

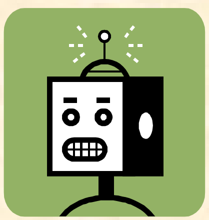
Objective of an under-frequency loading shedding system

- **Problem: Prevent circuit breakers from disconnecting due to the line frequency exceeding 60 Hz by more than their safety margin**
- **Solution: If the load is too high for too long, then cut some of the load**
  - **Bring the cut load back on-line as soon as possible!**

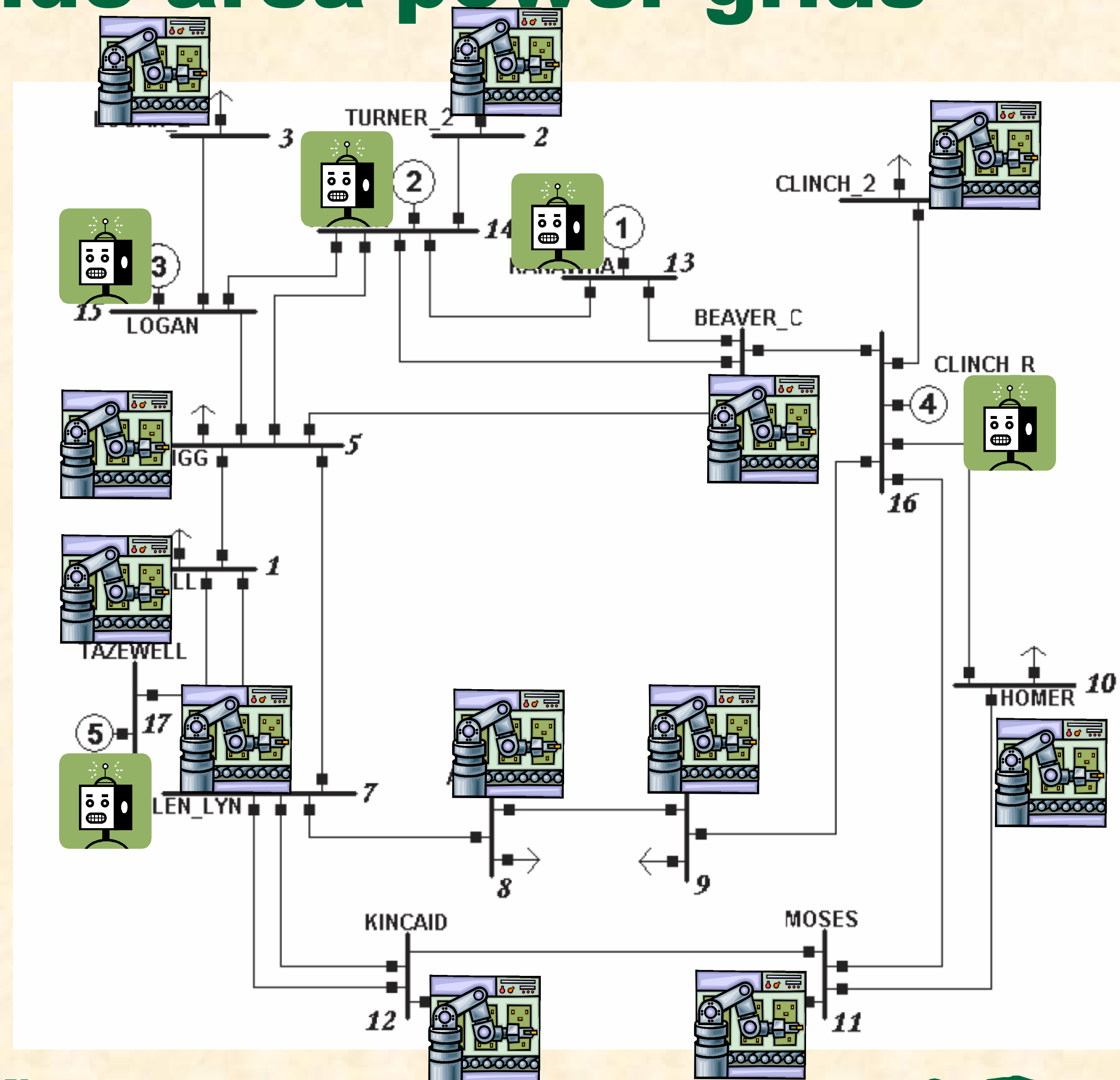
Example: Automatic, coordinated control of wide area power grids



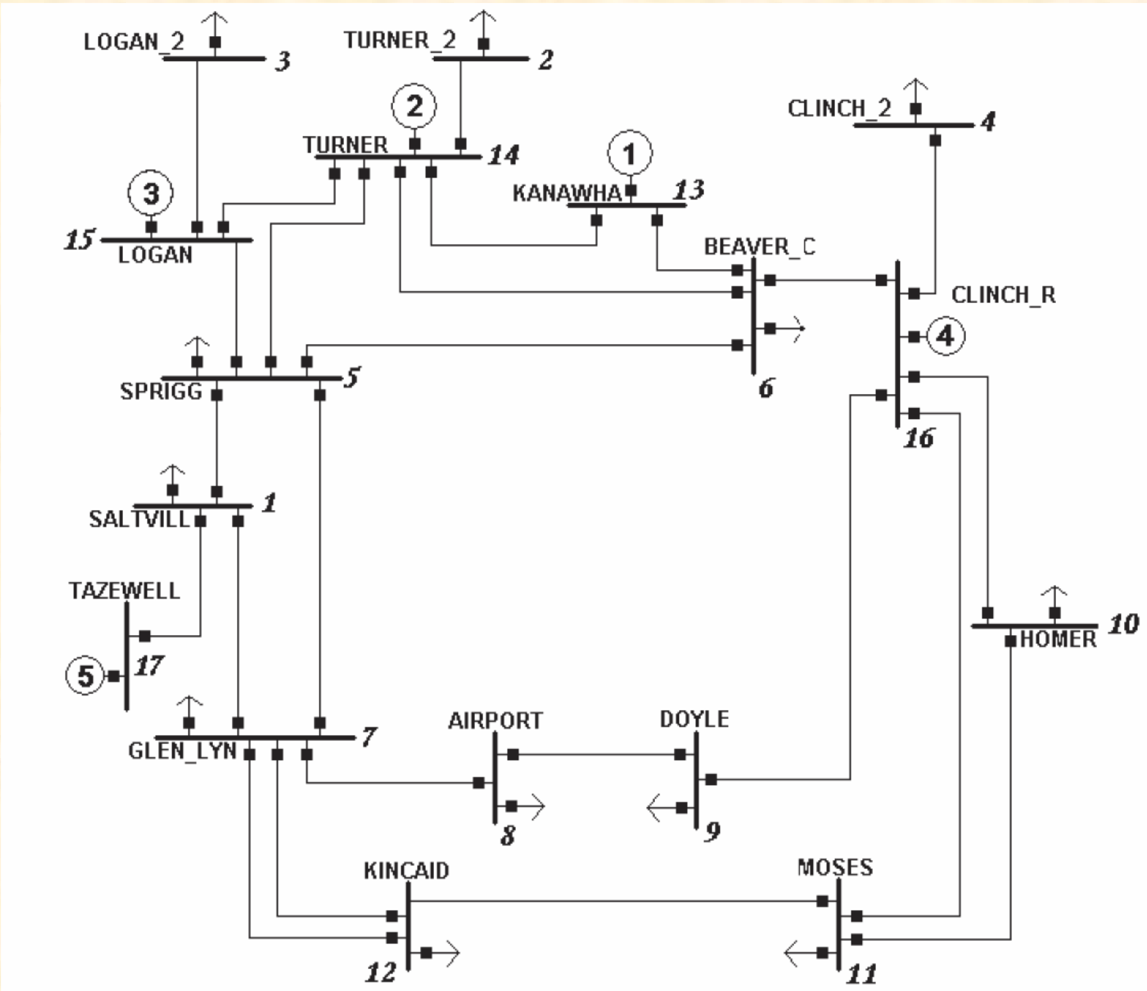
Networked actuators at electrical substations can connect and disconnected loads in fixed increments

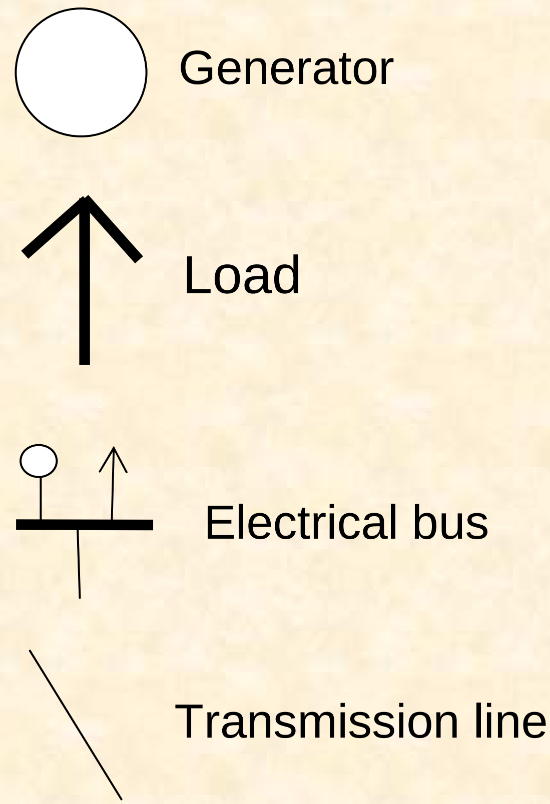


Monitor and control software attached to the generators control the actuators through an IP-based, wide area network

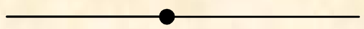


IEEE 14 Bus Test Case - Structure





State event: Generator disconnect



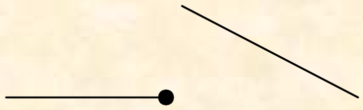
Frequency protection breaker closed

$$\begin{bmatrix} Y_{ll} & Y_{lg} \\ Y_{gl} & Y_{gg} \end{bmatrix}$$

|60 Hz - ω| > 0.1 Hz

Generator output frequency exceeds safe threshold

Disconnect causes an immediate change in the admittance matrix



Frequency protection breaker open

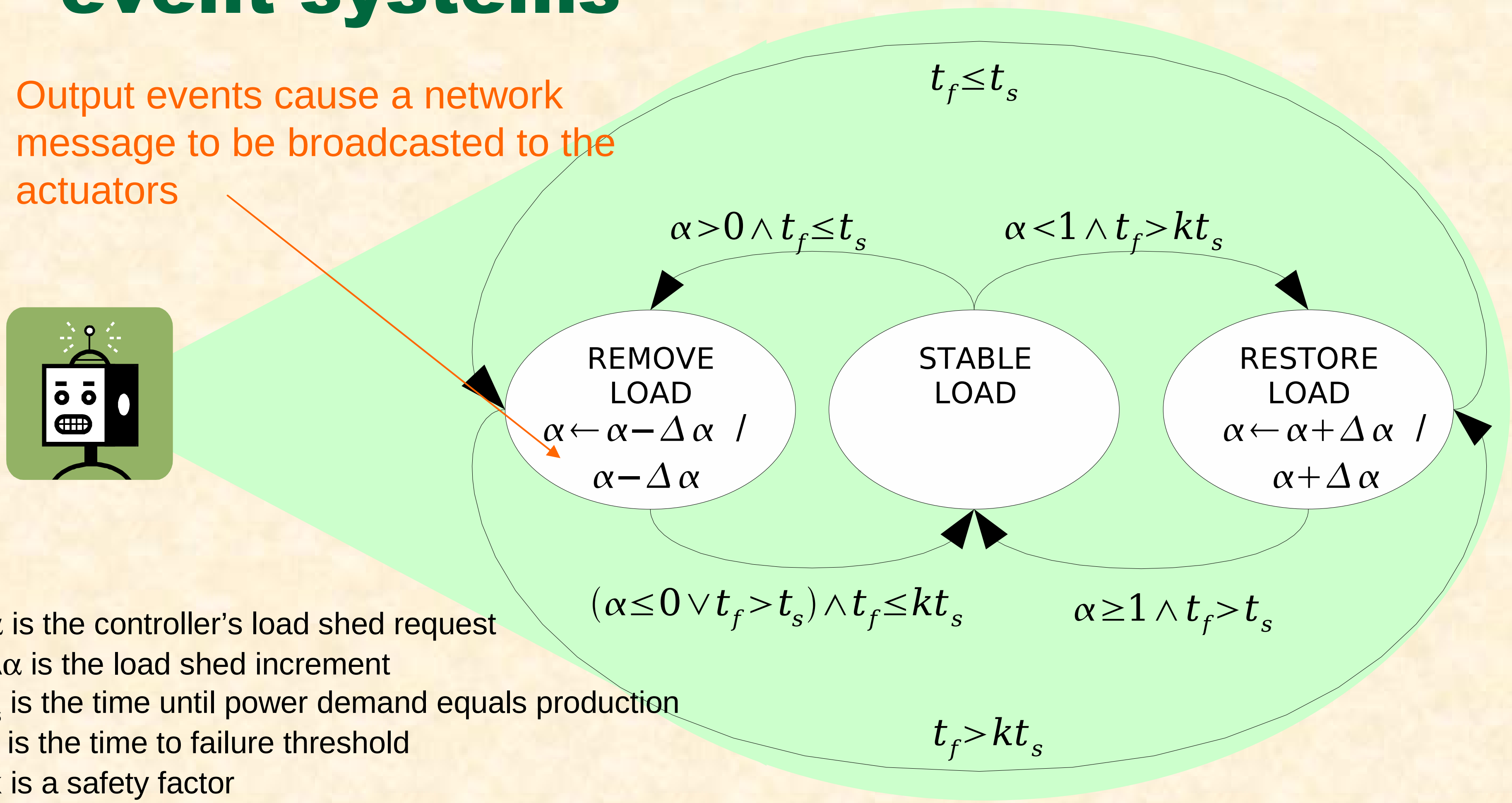
$$\begin{bmatrix} Y_{ll} & Y'_{lg} \\ Y'_{gl} & Y'_{gg} \end{bmatrix}$$

Basic controller concept

- **Monitor power production and demand at the generator terminals and the generator angular acceleration**
- **Monitor produces an input for the controller that is attached to the generator whenever the frequency of the power signal changes by 0.0025 Hz**
- **On receiving an input, the monitor calculates an estimate of time to under-frequency failure**
- **If that time is too short, send a network message asking actuators to shed a load increment**
- **If that time is very long, send a network message telling actuators to reconnect a load increment (if any are disconnected)**

Controllers are modeled as discrete event systems

Output events cause a network message to be broadcasted to the actuators



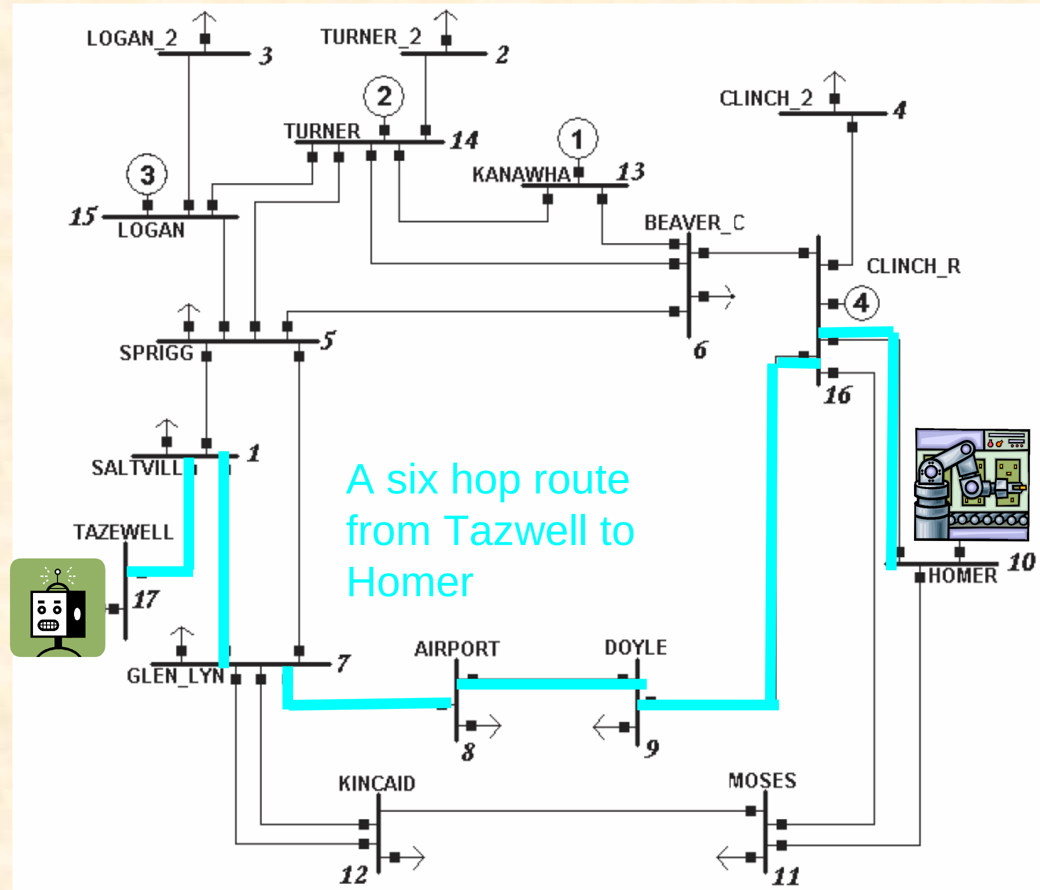
α is the controller's load shed request
$\Delta\alpha$ is the load shed increment
t_s is the time until power demand equals production
t_f is the time to failure threshold
k is a safety factor

The communication network is modeled as a discrete event system

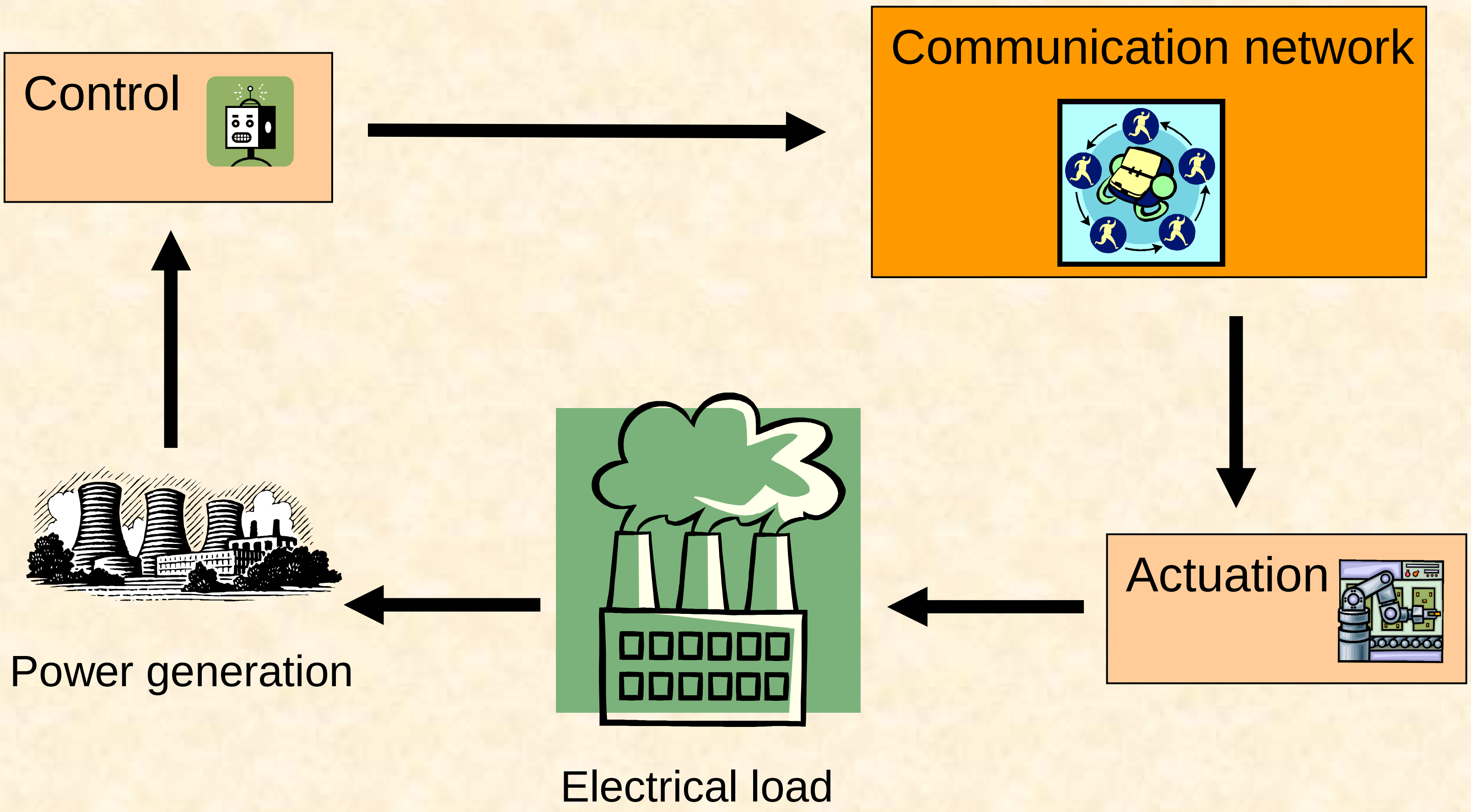
Network communication lines follow the electrical transmission lines

Broadcasts are implemented with a flooding protocol when UDP/IP is used

When TCP/IP is used, all communication is peer to peer



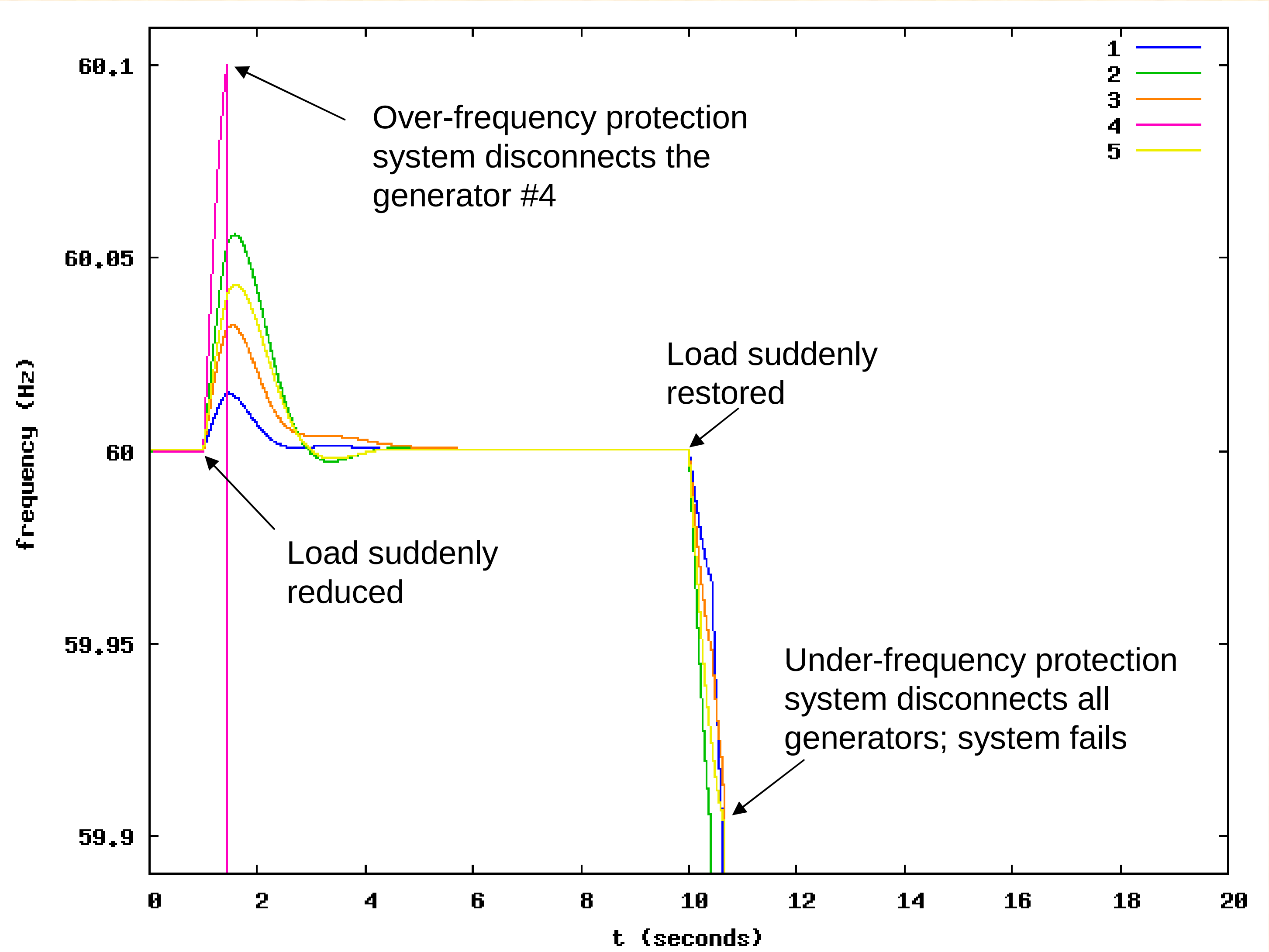
Discrete event and continuous subsystems are tightly coupled



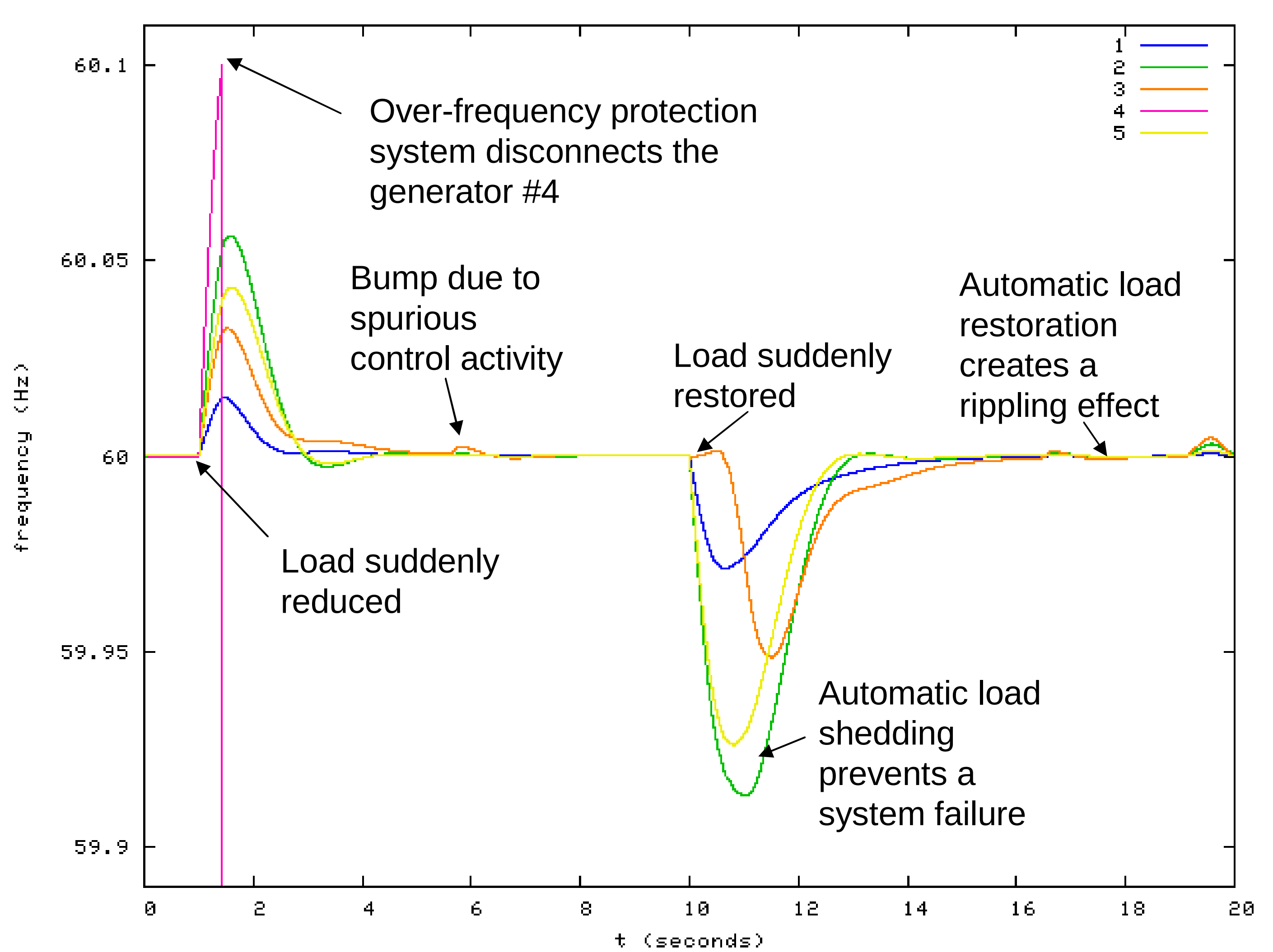
Experimental Setup

- At t = 1 second the electrical demand drops suddenly. The electrical system recovers.
- At t = 10 seconds the lost load is suddenly restored plus a little extra.
- Vary the communication protocol and communication line characteristics and observe the consequences.
  - A single line has 1 ms based delay and up to 10 Mbps throughput until stated otherwise, default NS2 limit on queue size (~50)

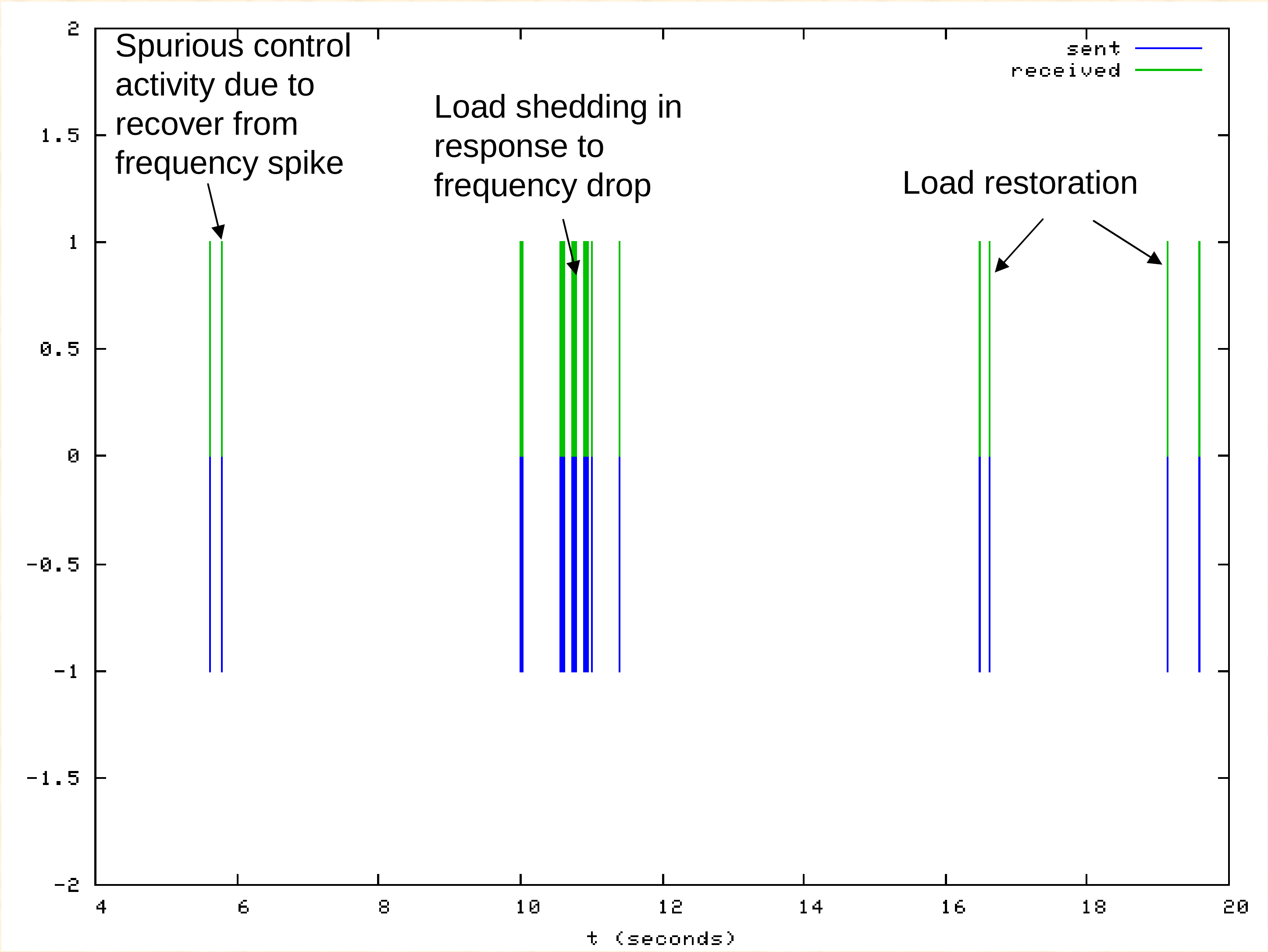
Simulation without load control



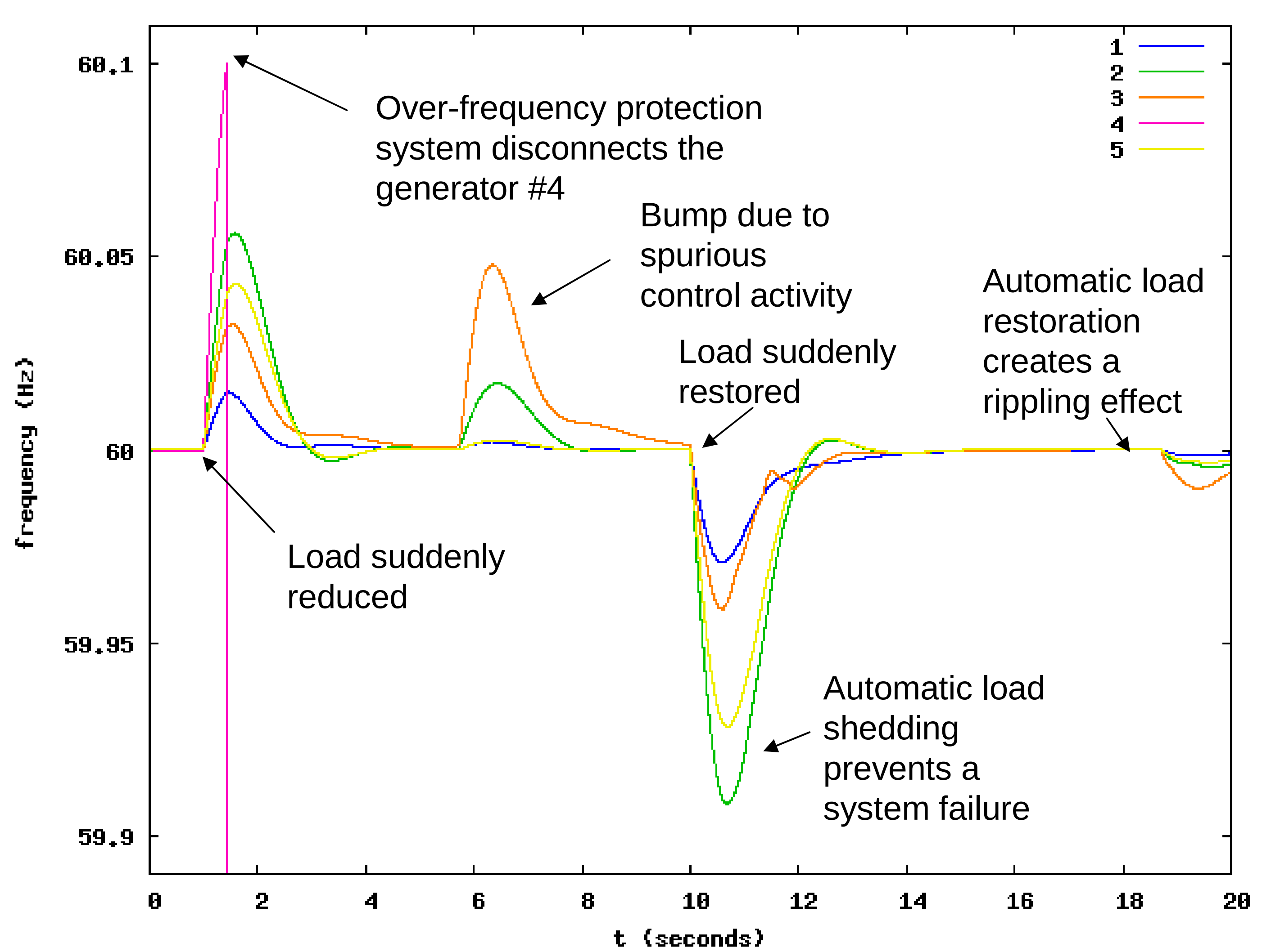
Frequency: ideal communication



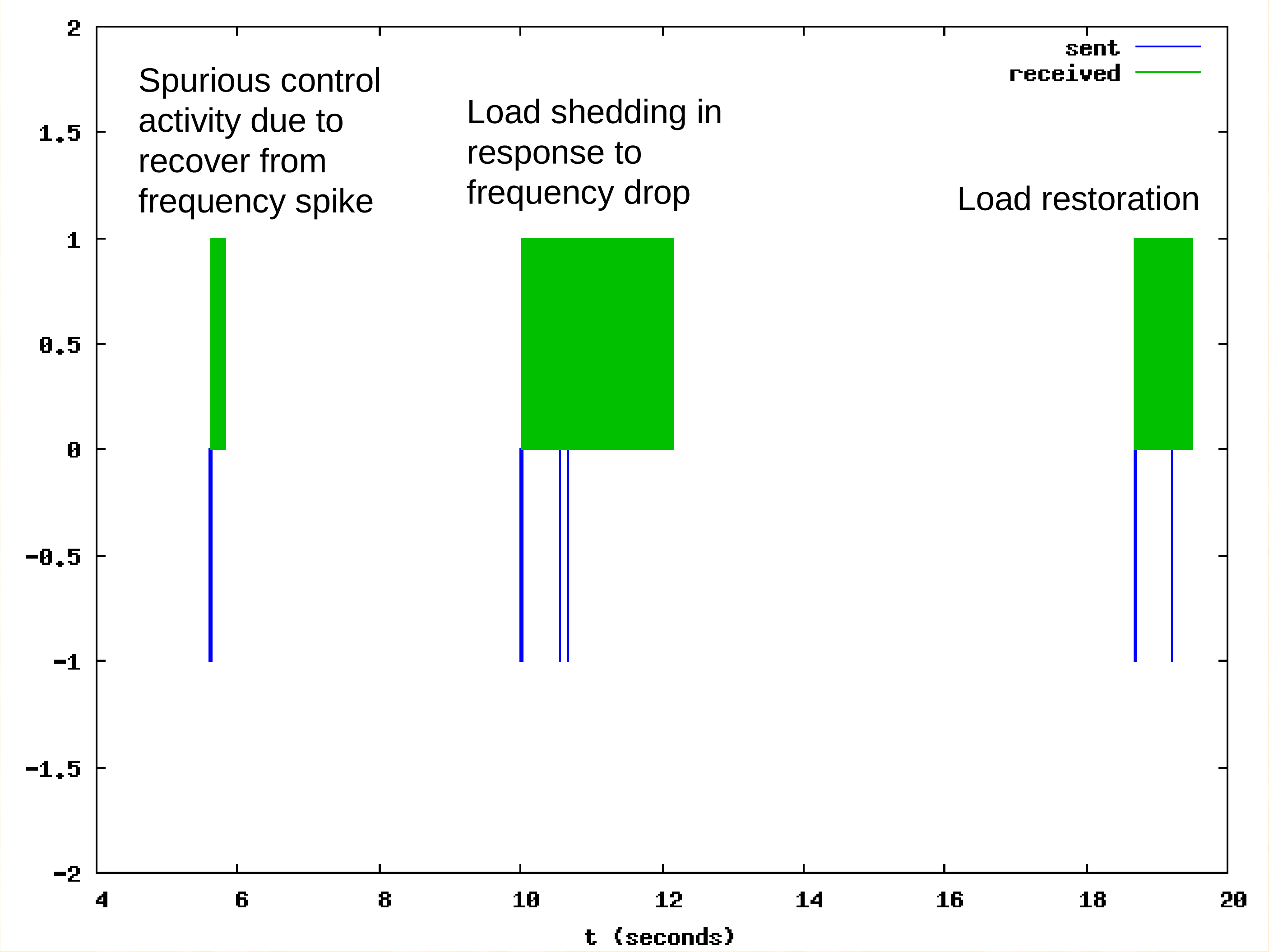
Messages: ideal communication



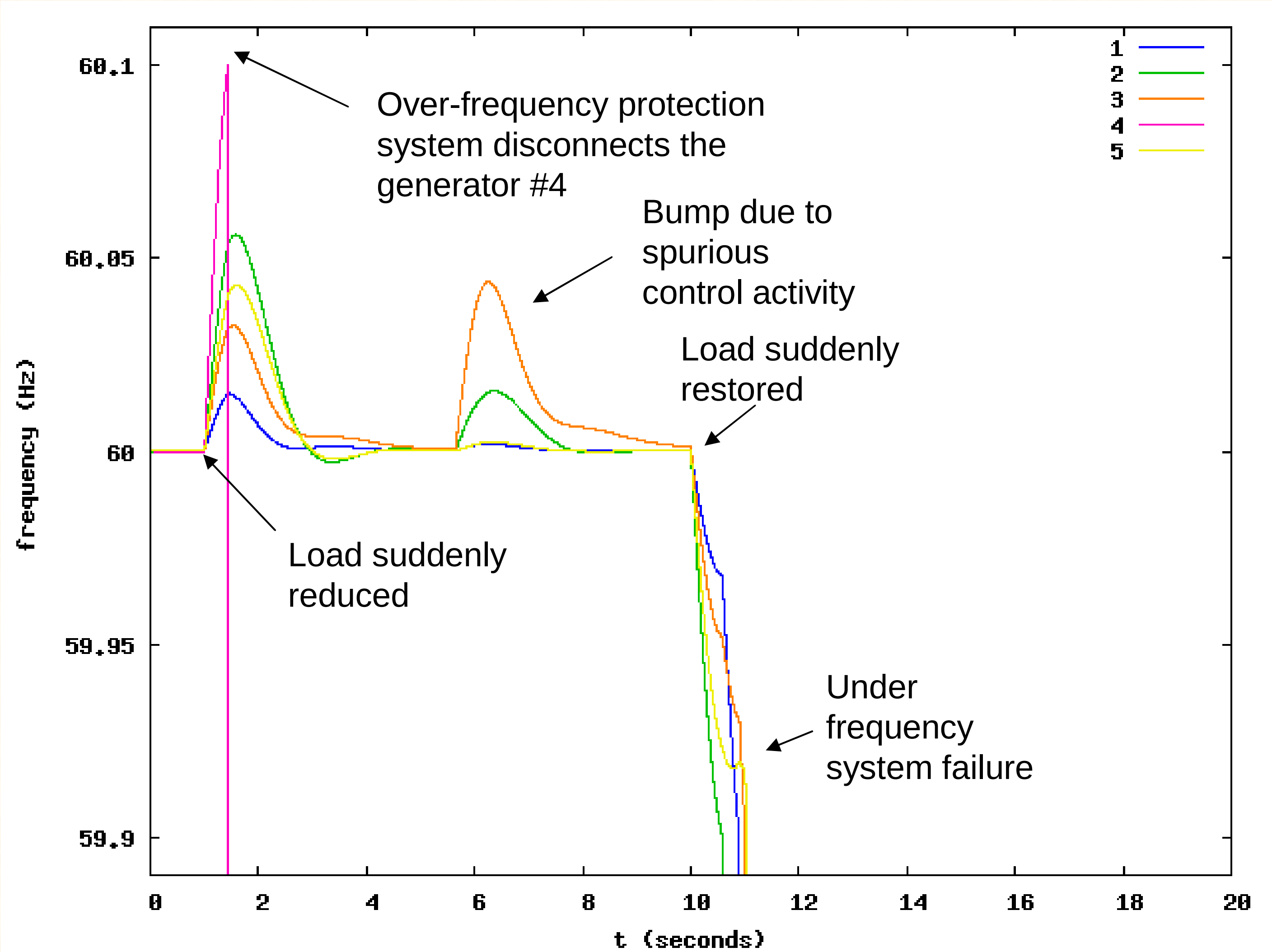
Frequency: UDP flooding, communication lines follow transmission lines



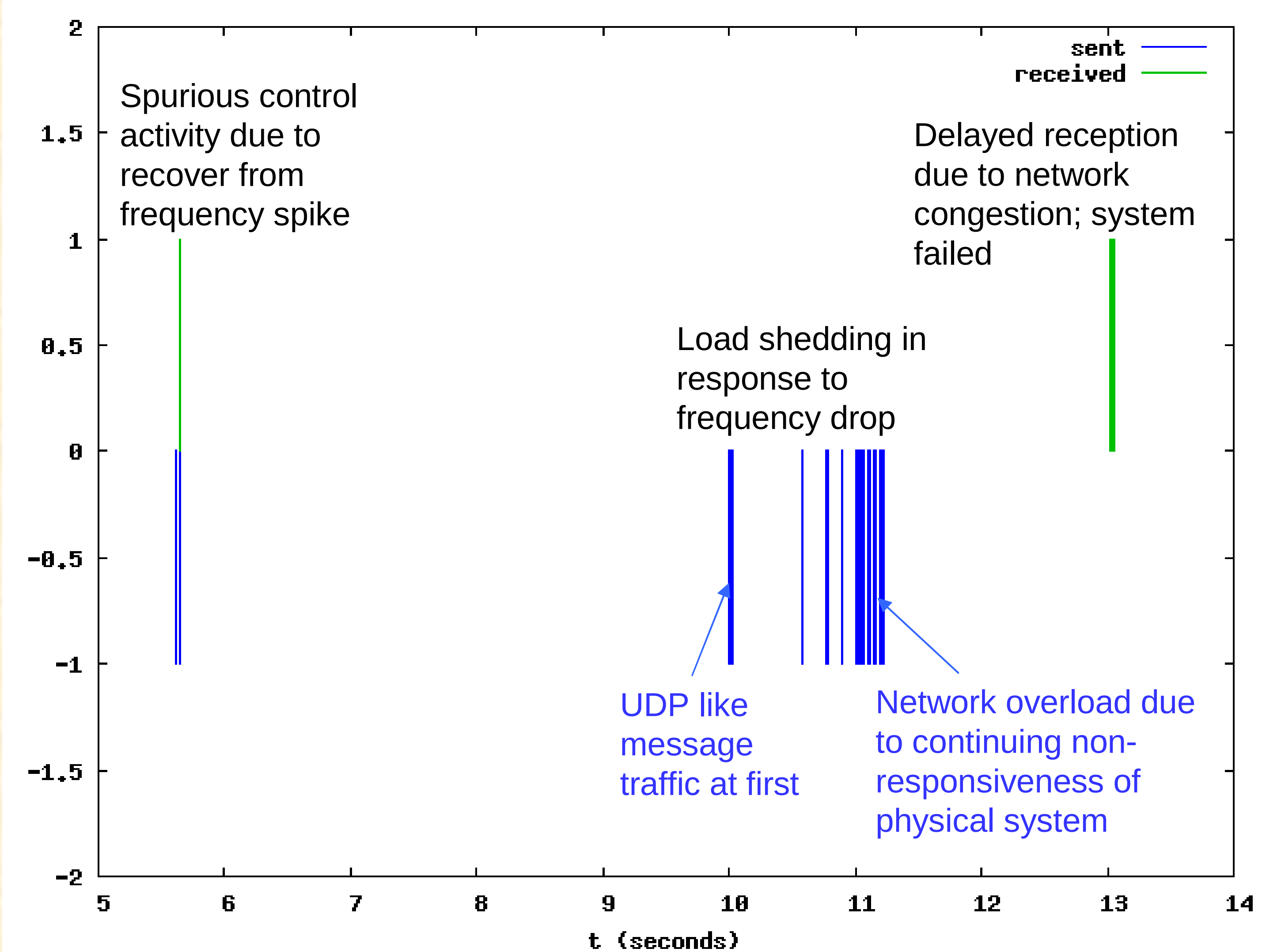
Messages: UDP flooding, communication lines follow transmission lines



Frequency: TCP, communication lines follow transmission lines



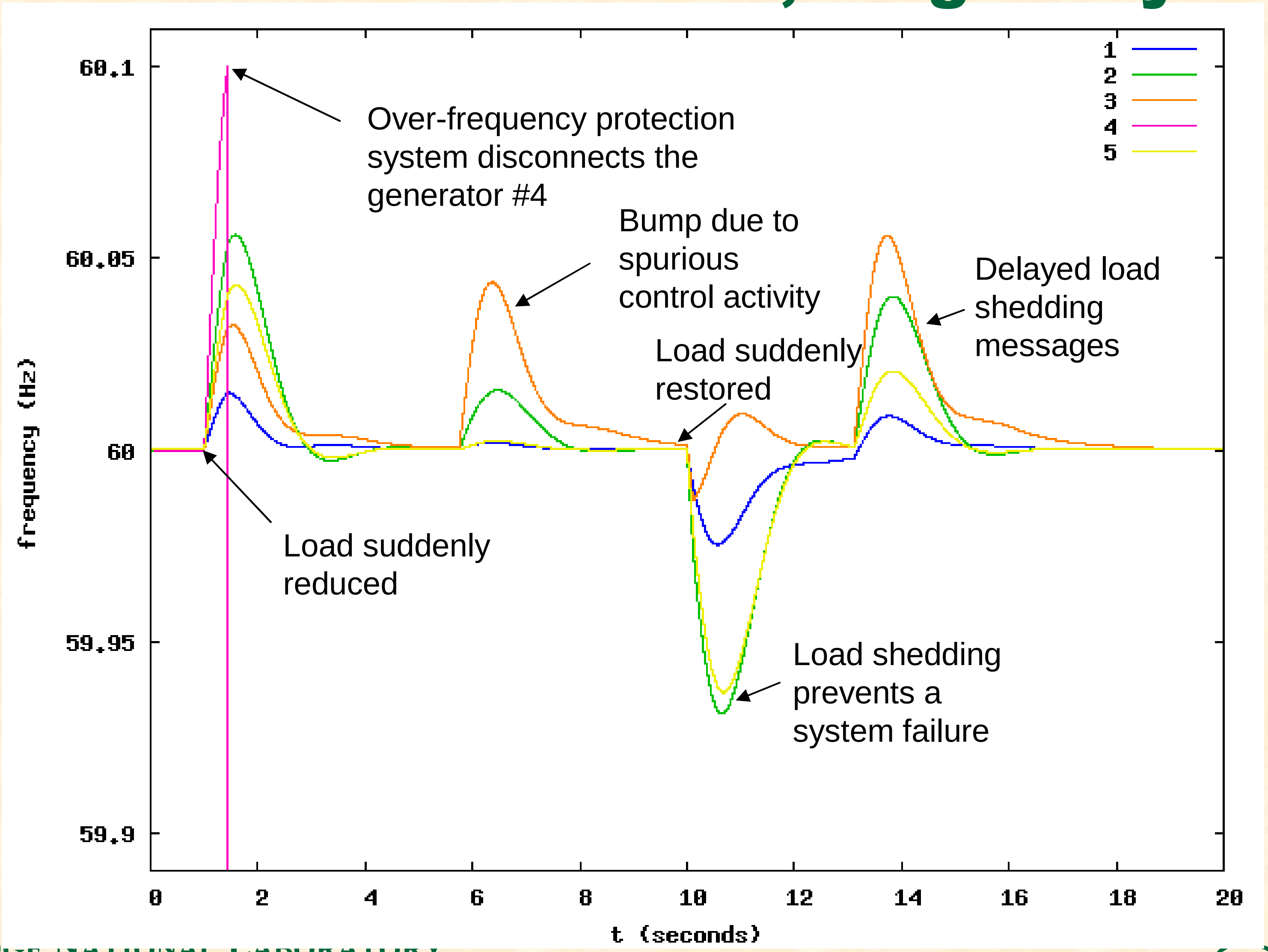
Messages: TCP, communication lines follow transmission lines



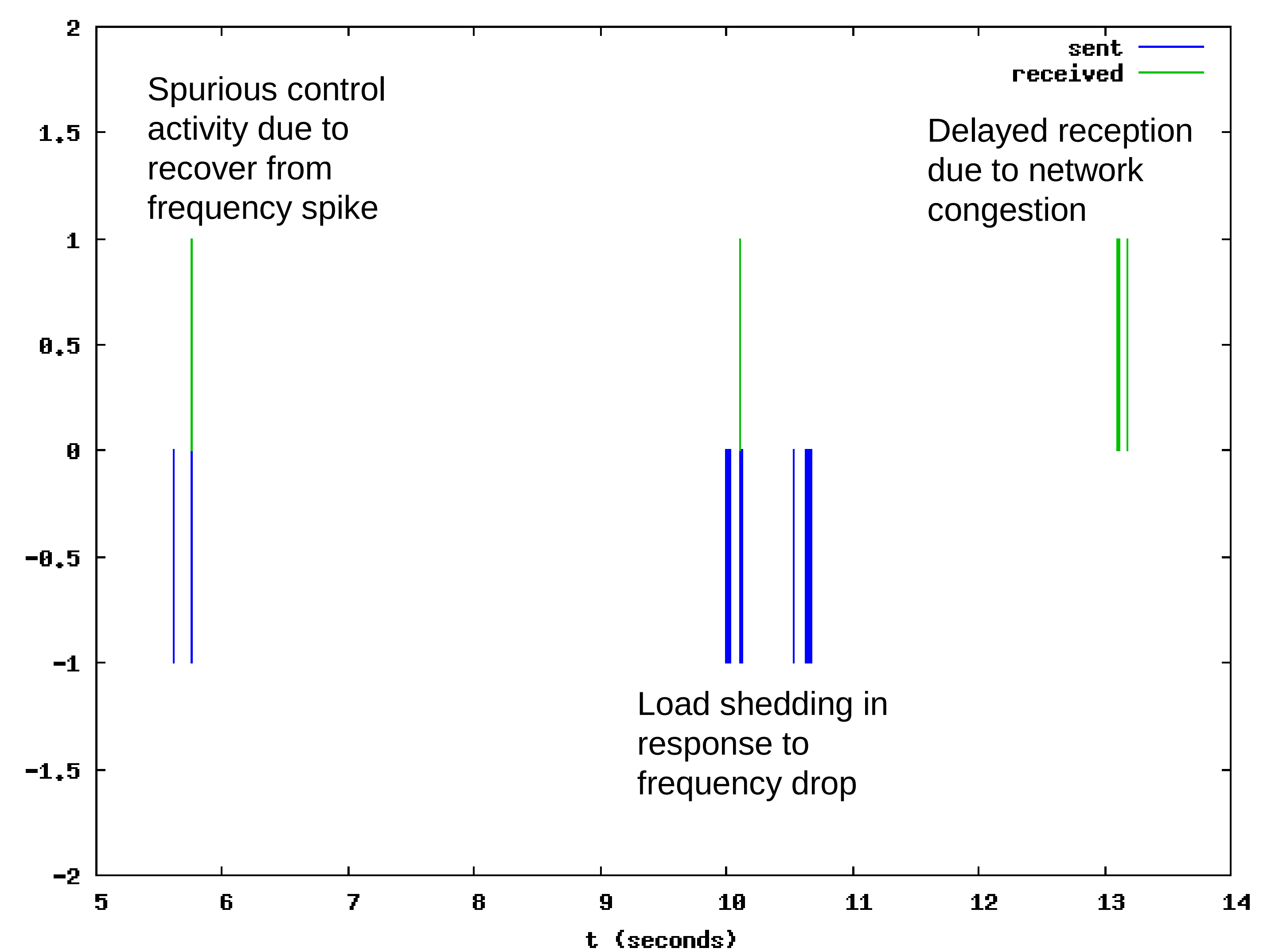
Longer delays – better system response?

- **This TCP case used a 1 millisecond base line latency**
- **Increasing base line latency to 10 milliseconds has a dramatic positive effect**
- **Suggests that there must be a lower *and* upper bound on tolerable latency for the TCP case**
- **This effect is similar, at least in appearance, to the model of market stability when the market clearing time is large**

Frequency: TCP, communication lines follow transmission lines, long delay



Messages: TCP, communication lines follow transmission lines, long delay



Interactive dynamics matter!

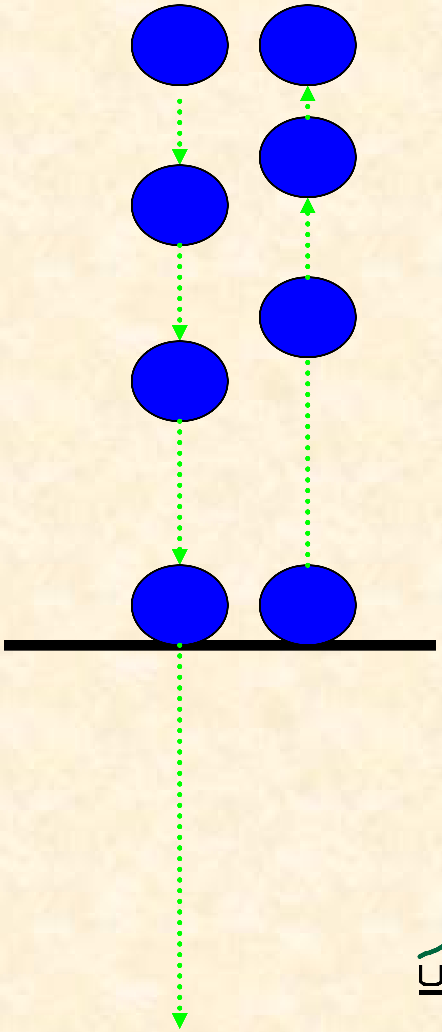
- **The communication network can not be characterized independently of the load that is offered to it.**
- **The offered load can not be characterized independently of the communication network.**
- **A new framework for analysis is needed that integrates existing, but disconnected, analysis frameworks for queuing networks and linear control systems.**
- **New simulation tools are needed to study these problems.**

Models contain continuous systems with state events

$$\frac{dv}{dt} = -g$$

$$\frac{dh}{dt} = v$$

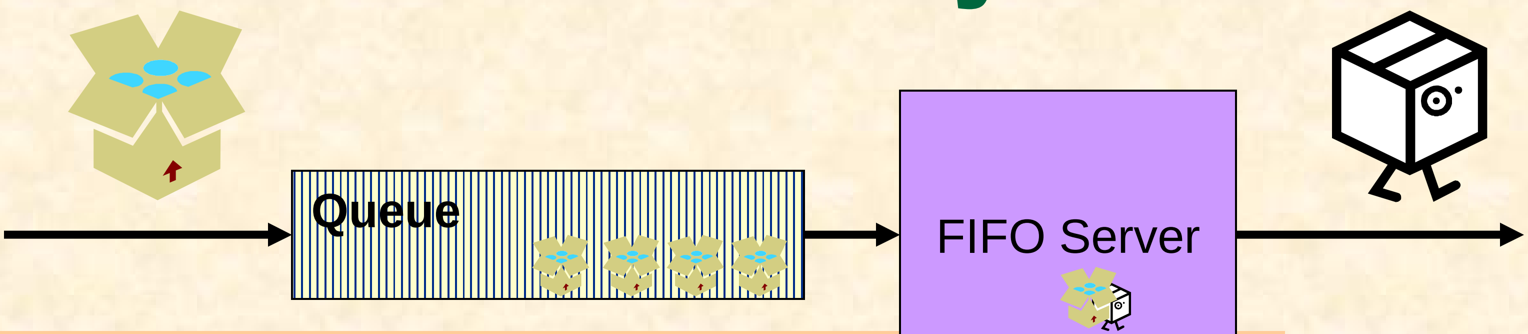
Continuous dynamic model of a falling ball



$$\textbf{\textit{when}}\ h = 0\ \textbf{\textit{set}}\ v \leftarrow -v$$

A *state event* occurs when the ball hits the floor, this event causes an instantaneous change in velocity

And discrete event systems



EntityArrives at time t
If the server is busy
add the entity to the back of the queue
Else
schedule an EntityServed event time at t + ServiceTime
Endif
End EntityArrives

EntityServed at time t
Eject the finished entity
If the queue is not empty
remove the first entity in the queue
schedule an EntityServed event at time t + ServiceTime
Endif
End EntityServed

The Goal

To build simulation software for *complex* hybrid systems that

- **Incorporates non-trivial continuous models with state events**
- **Includes large discrete event sub-components**
  - **Packet level network models**
- **Is numerical robust**
- **Is computationally efficient**

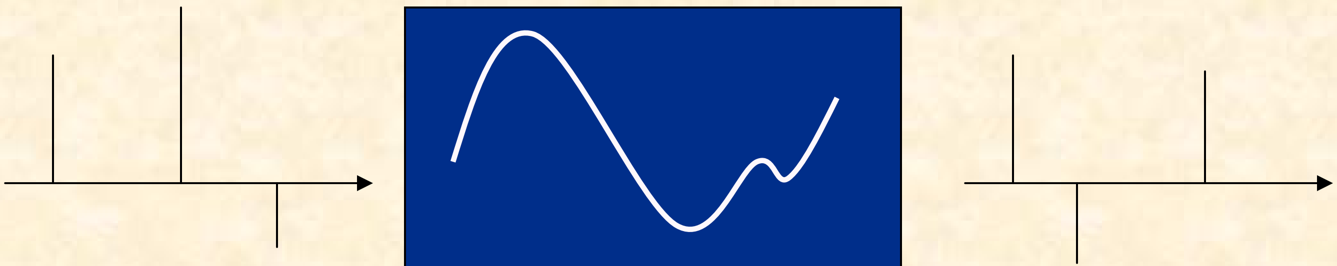
Three main computational problems

- Precise simulation of discrete events
  - Events that depend on continuous variables (e.g., tripping of the breakers)
  - Events that depend only on time (e.g., pre-scheduled load changes)
  - Events that depend on previous events (e.g., simulating the UDP/IP protocol stack)
- Accurate approximation of the continuous dynamics between events
  - Numerical integration of the ODE set

~~$2+1 \approx 10$~~

- Computational efficiency
  - Reasonable performance of the event scheduler
  - Reasonable step-size selections for the numerical integration scheme

The means: Continuous sub-systems with discrete event I/O



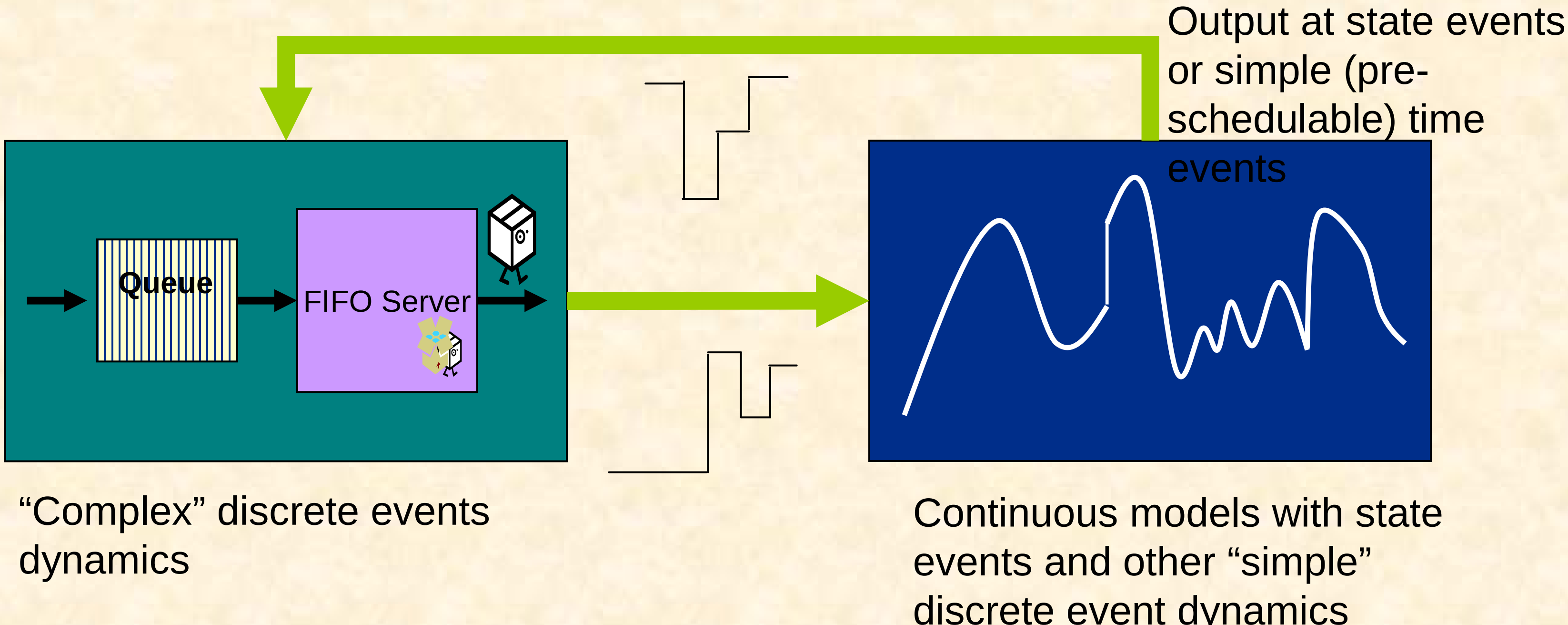
Discrete input arrives from discrete event sub-systems

Discrete output specified by state and time events
Event condition: g(x,t) = 0

Internal dynamics described by DAE or ODE with zero crossing function for state event detection

Integrating discrete/continuous models

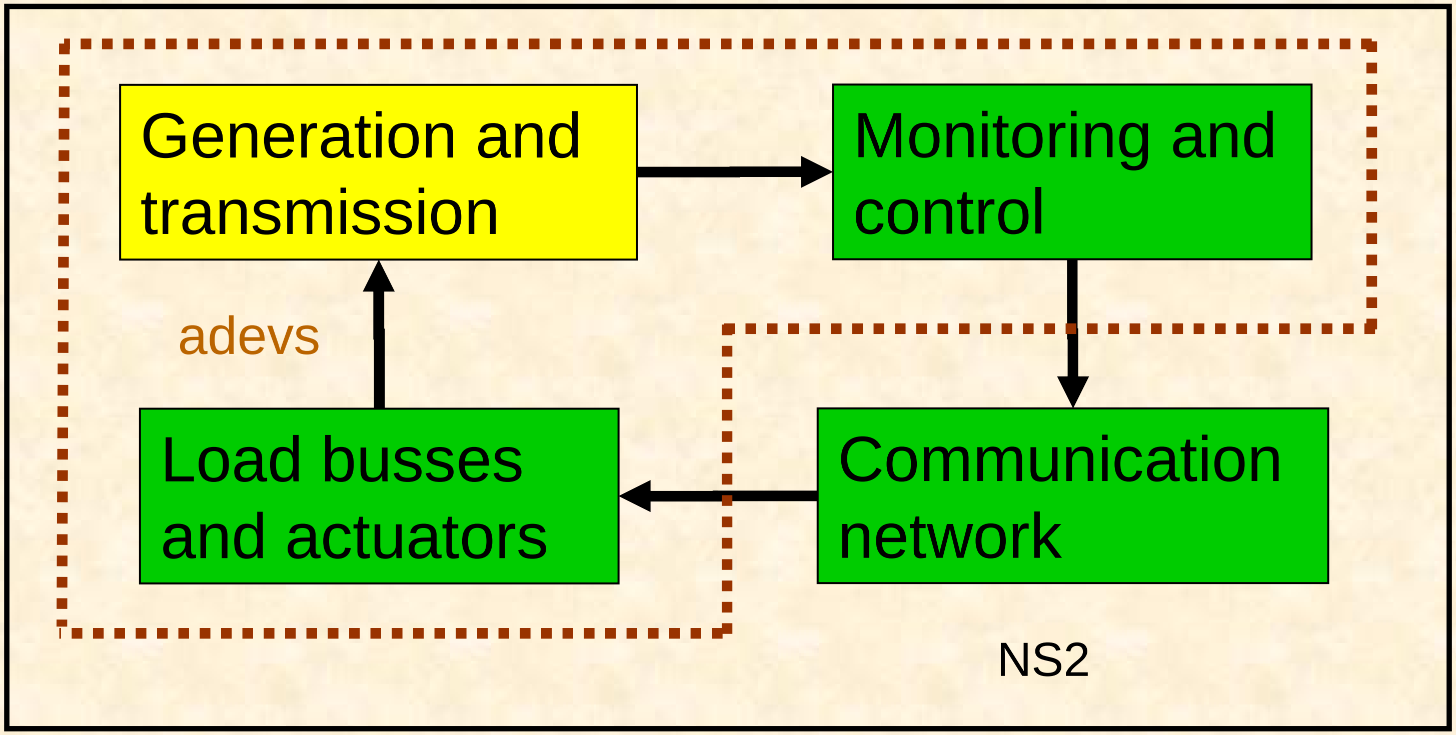
- **Overarching simulation is discrete event**
- **Continuous sub-systems are simulated with embedded numerical algorithms**



Building the simulation software

- **Implement the discrete event system using a suitable discrete event simulation package**
  - **NS2 for the network simulation**
  - **DEVS based simulation tool (*adevs)* for the controller models and load schedule**
  - **Integrated these using the simulation control API provided by adevs – NS2 was the 'main' simulation driver**
- **Wrap a continuous system solver in a discrete event module**
  - **RK45 with interval bisection**
  - **Implemented as a DEVS atomic model**

Overview of the software architecture



In Conclusion

- The discrete and continuous parts of a complicated hybrid system can not be characterized independently
  - The communication network can not be characterized independently of the load that is offered to it.
  - The offered load can not be characterized independently of the communication network.
- New simulation frameworks are needed to study models that integrate complicated continuous and discrete event dynamics
- Our software is available at http://www.ornl.gov/~1qn/thyme