An Introduction to Network Simulation Using Ptolemy Software Tool

Nazy Alborz

nalborz@sfu.ca

Communication Networks Laboratory

Simon Fraser University

Road Map:

- History
- Introduction to Ptolemy, its architecture and its applications
- Ptolemy environment
- Running a Ptolemy application (Leaky bucket mechanism)
- Simulation results

History:

- Created by Department of Computer and Electrical Engineering, U.C. Berkeley
- Named after the second century Greek astronomer and mathematician, Ptolemaeus
- Since 1990, seven C++ based versions, 0.1 to 0.7 released
- In 1999 new Java based version (Ptolemy II) was released
- Now four Java based versions, 0.1 to 0.4 are released

What is Ptolemy?

- Flexible foundation upon which many prototyping environments can be built.

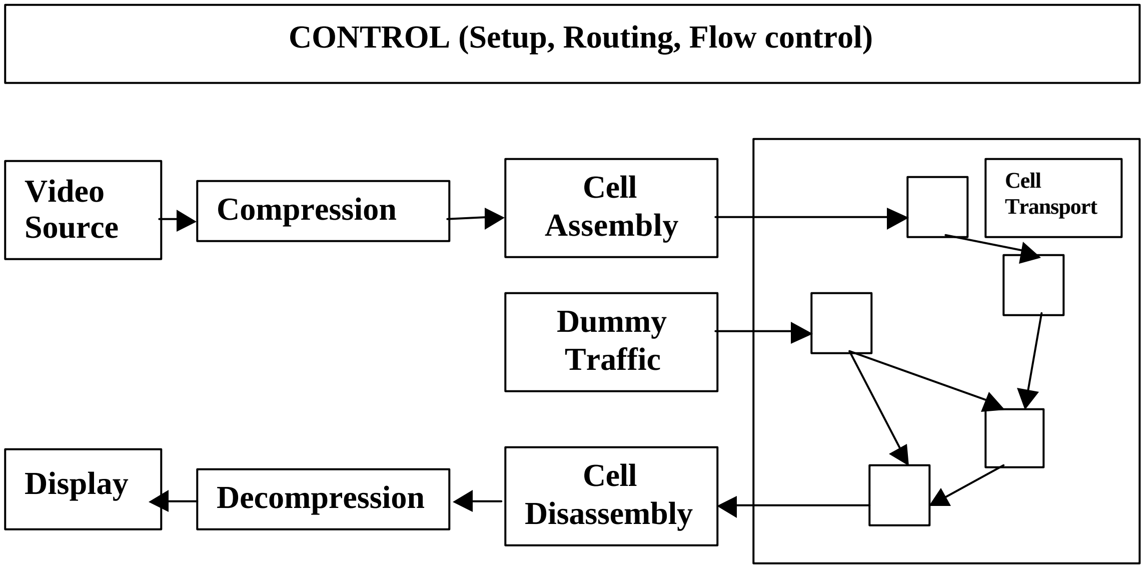
Examples:

- Graphical programming for signal processing
- Modeling of communication networks
- Circuit design
- Design assistance for hardware/software co design

- Capable of combining theses environments for modeling and simulation of complex heterogeneous systems

- Uses Object Oriented programming methodology to model and integrate these subsystems of the complex systems.

An example of a complex system: Video transmission through ATM network



These subsystems are interacting together:

- Video compression (Signal processing)
- Transport (Networking)
- Control (Software)

Ptolemy architecture:

- Ptolemy consists of a core (*kernel*) upon which special design environments (*domains*) are built.
  - **Kernel:** A family of object-oriented classes that makes some assumptions about the system.
  - **Domain:** Is defined by new object oriented classes based on Kernel classes.Each domain is an appropriate computational model for a particular type of subsystem. Domains can interact with one another.
- Use of object-oriented software technology permits the domains to interact with one another without having knowledge about eachother
- Domains can model each subsystem of a complex, heterogeneous system in an efficient manner
- These different subsystems can be nested to form a tree of subsystems

Some existing domains in Ptolemy:

Simulation domains: Interpreters that run an excecutable specification of a system on a local workstation

- **DDF (Dynamic Data Flow)**
  - Special case of data flow model
  - For graphs that flow of control is predictable at compile time
- **SDF (Synchronous Data Flow)**
  - Used for dynamic (run-time) scheduling
- **FSM (Finite State Machine)**
  - Used for describing control oriented systems
- **DE (Discreet Event)**
  - Use event-driven computational model
  - Particles carry time stamps to show events at a particular time
  - Events happen in chronological order

Some existing domains in Ptolemy:

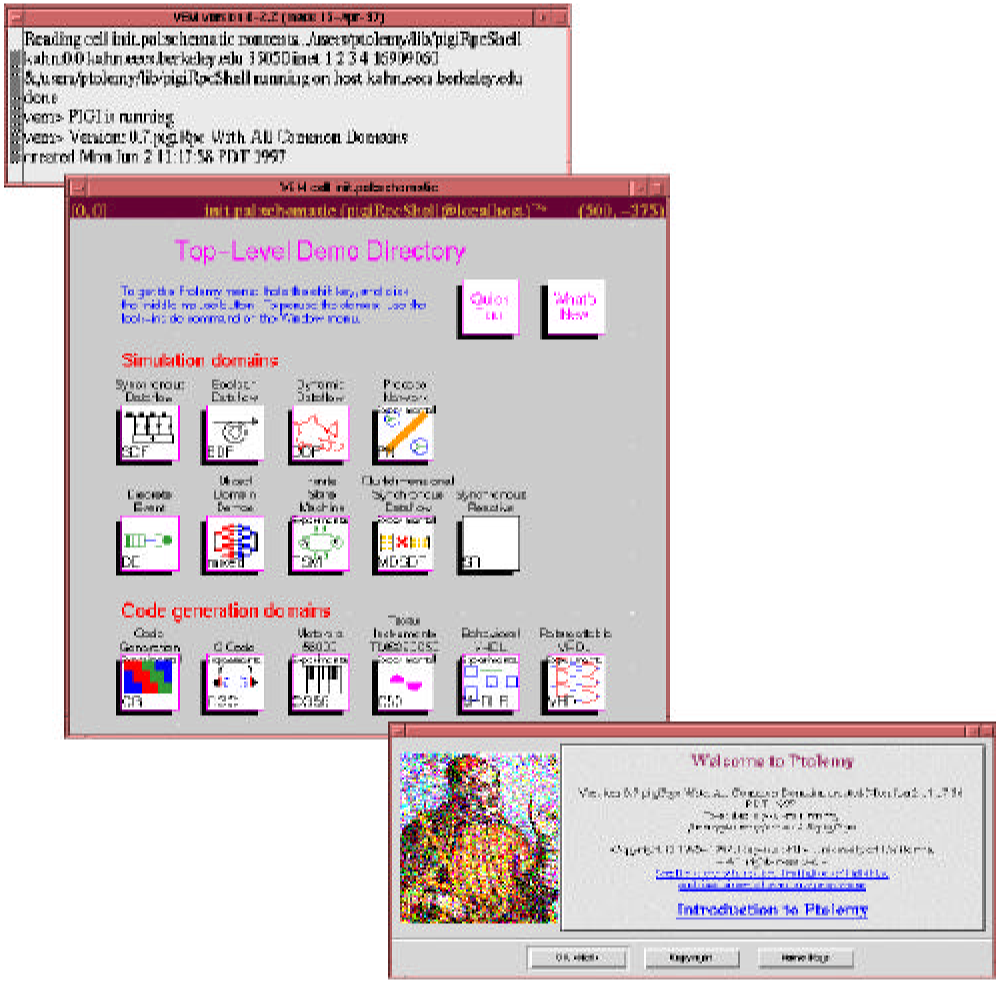
Code generation domains: Translator domain that translates the specifications into some programming languages.

- CG (Code generation): Base for all other CG domains, Codes are generated, compiled and executed.
- CGC (Code generation in C)
- VHDL (code generation in VHDL)

Ptolemy Interfaces:

- Textual Interface *(ptcl)*
  - Works on dumb terminals
  - Accepts input commands from keyboard
  - Based on Tcl (Tool command language), Extends Tcl by new adding new commands
- Graphical Interface *(pigi)*
  - Graphical editor based on Berkeley CAD tools.
  - Applications are constructed graphically by interconnecting icons

Running pigi:



```
Reading cell init.pal:schematic contents../users/ptolemy/lib/pigiRpcShell
kahn.0:0 kahn.eecs.berkeley.edu 39050 inet 1 2 3 4 16909060
&,/users/ptolemy/lib/pigiRpcShell running on host kahn.eecs.berkeley.edu
done
vem> PIGI is running
vem> Version: 0.7.pigiRpc With All Common Domains
created Mon Jun 2 11:17:58 PDT 1997
```

To get the Ptolemy menu: hold the shift key, and click the middle mouse button. To peruse the demos, use the look-inside command or the Window menu.

- Vem console

(upper left)

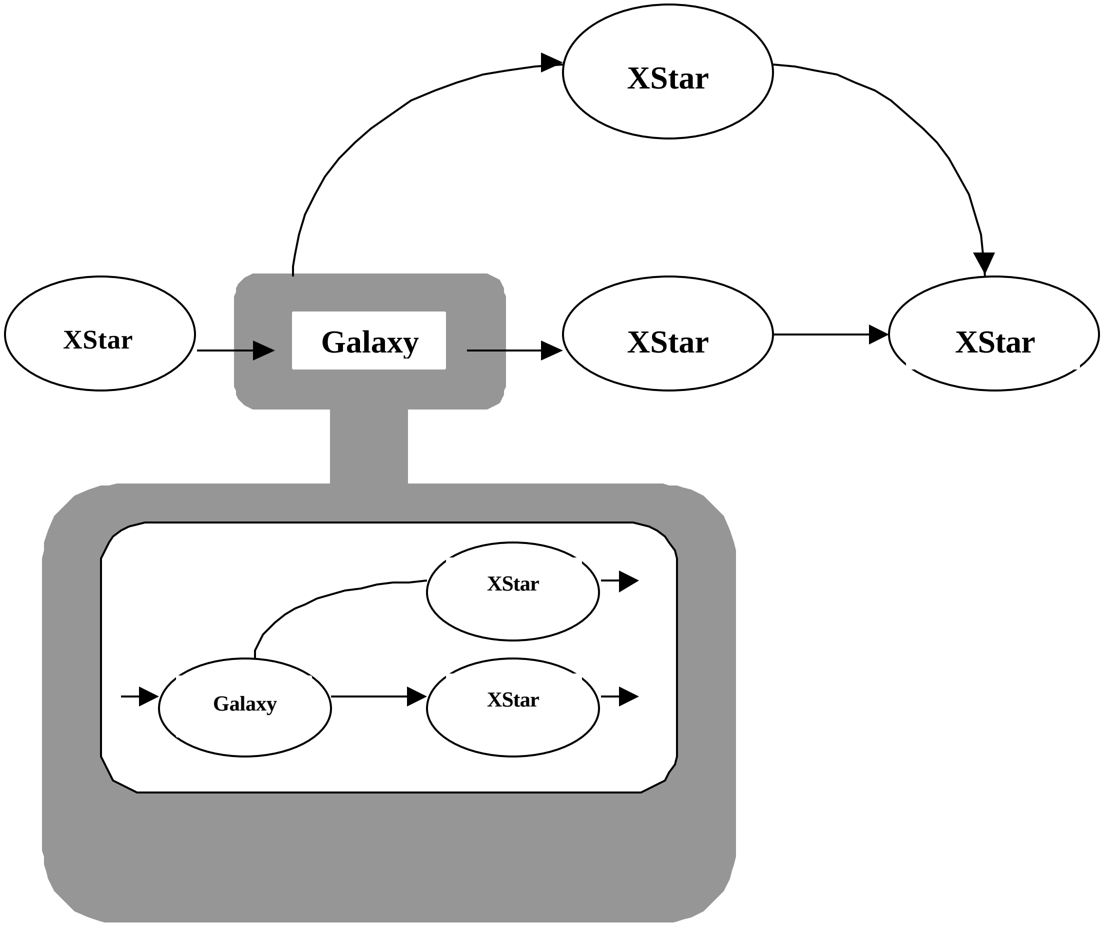
- Icon screen (Palette) representing icon directories (middle)
- Message window

Welcome (lower right)

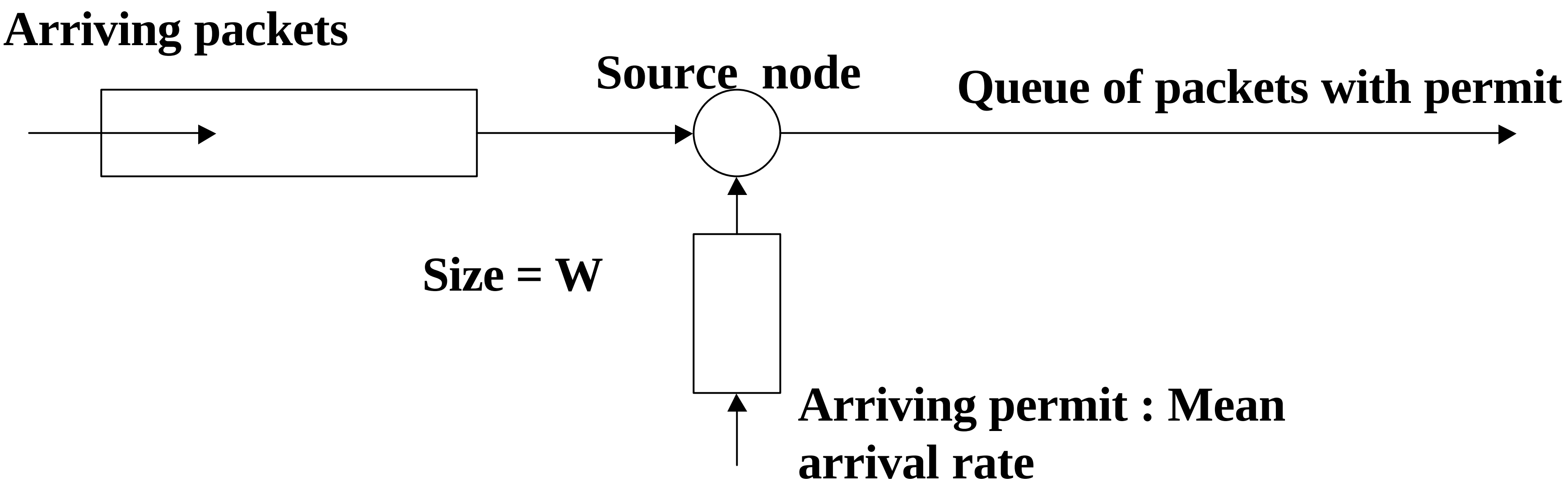
Ptolemy Objects:

- **Block:** The base class in Kernel for Stars, Galaxies and Universes.
- **Star:** Lowest level block, has functionality and defined in C++.
- **Galaxy:** A block containing a network of other blocks.
- **Universe:** An entire Ptolemy application. A type of Galaxy.
- **Palette:** An icon containing a library of other icons.
- **Scheduler:** Associated with each domain to determine the order of execution of Blocks inside the domain.
- **Target:** Manages the execution of a simulation or code generation process.

A complete Ptolemy application(Universe):



Leaky bucket controller:

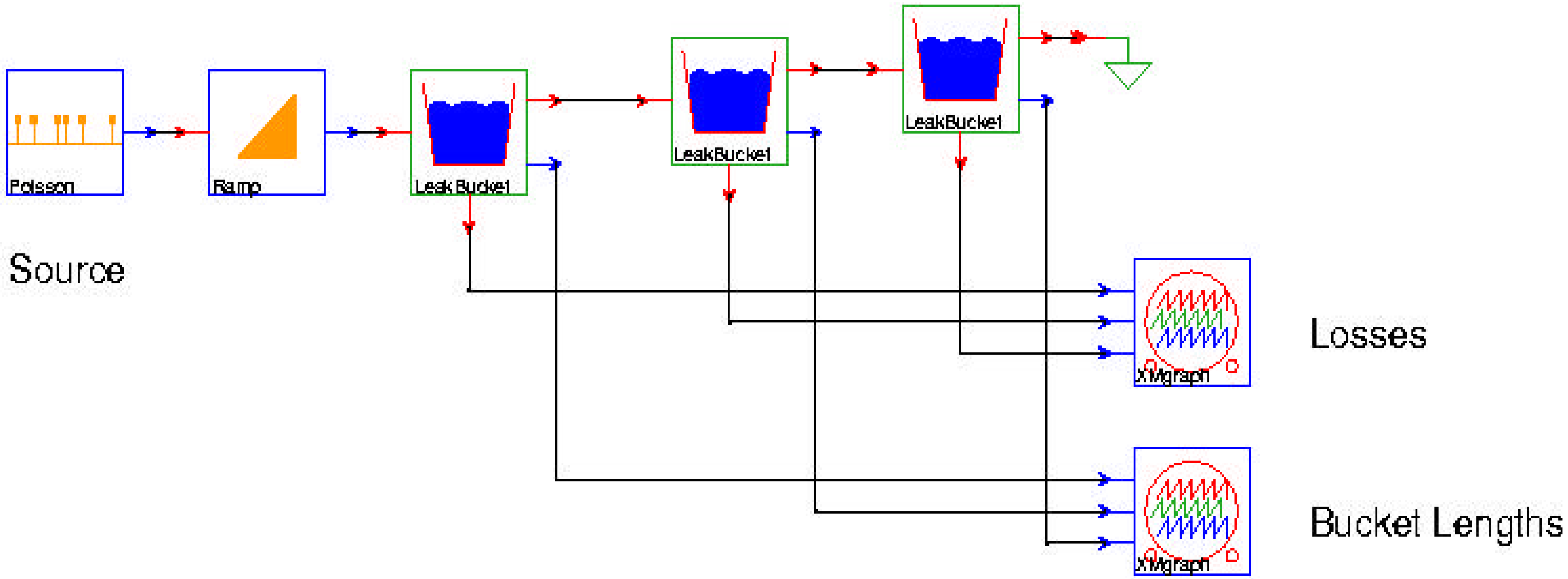

Arriving packets
Source node
Queue of packets with permit
Size = W
Arriving permit : Mean arrival rate

- A flow control rate mechanism.
- Arriving packets get a permit to enter network, else wait in queue.
- Permits generated at a constant rate and stored in a queue (bucket).
- Buckets have a max. capacity = W.
- Packets arriving while bucket size = W are discarded.
- Keep packet flow within a bound.

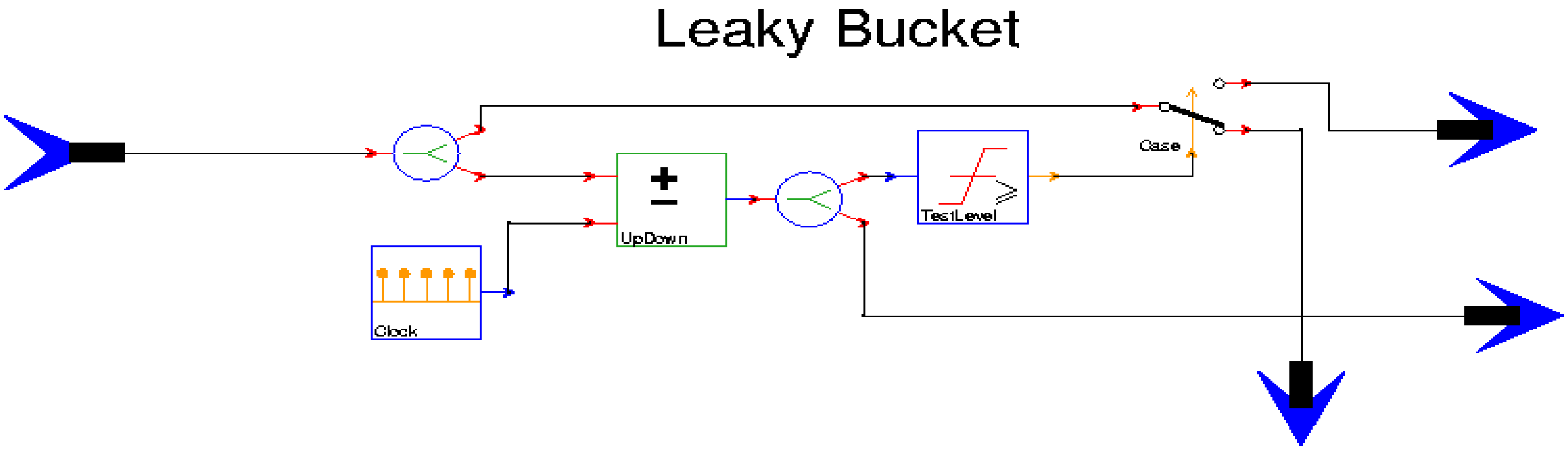
Leaky Bucket Universe

Leaky Bucket Monitors

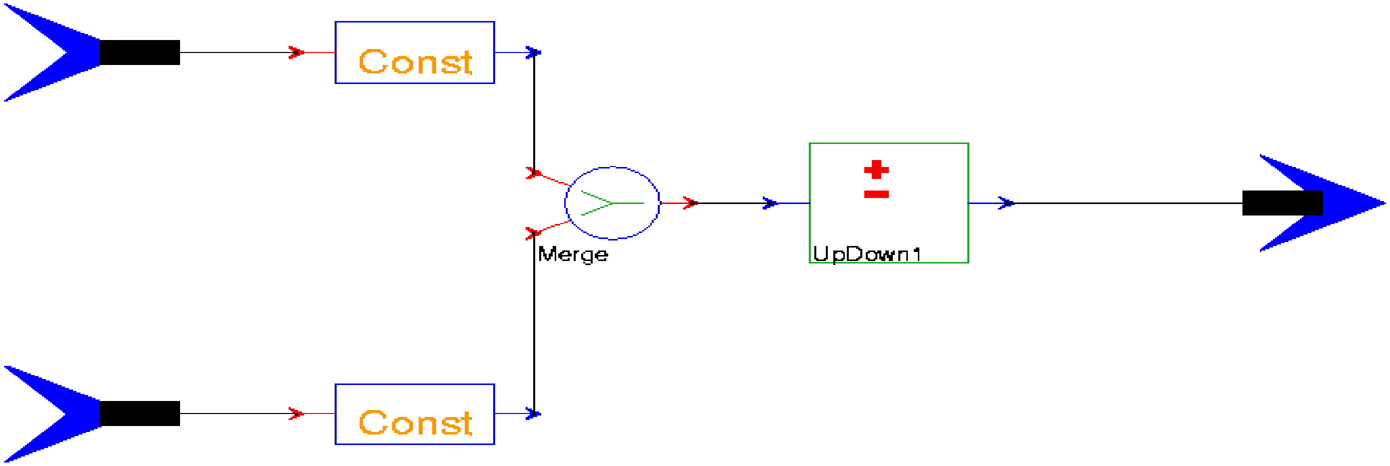
Ensure that a source's rate does not exceed specified bounds.



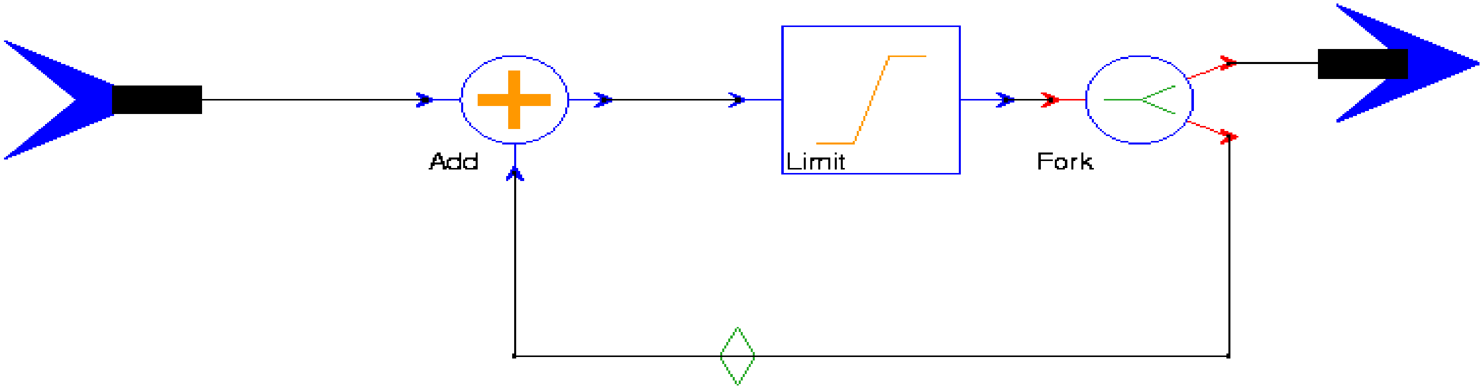
Leaky Bucket Galaxy



Updown Galaxy

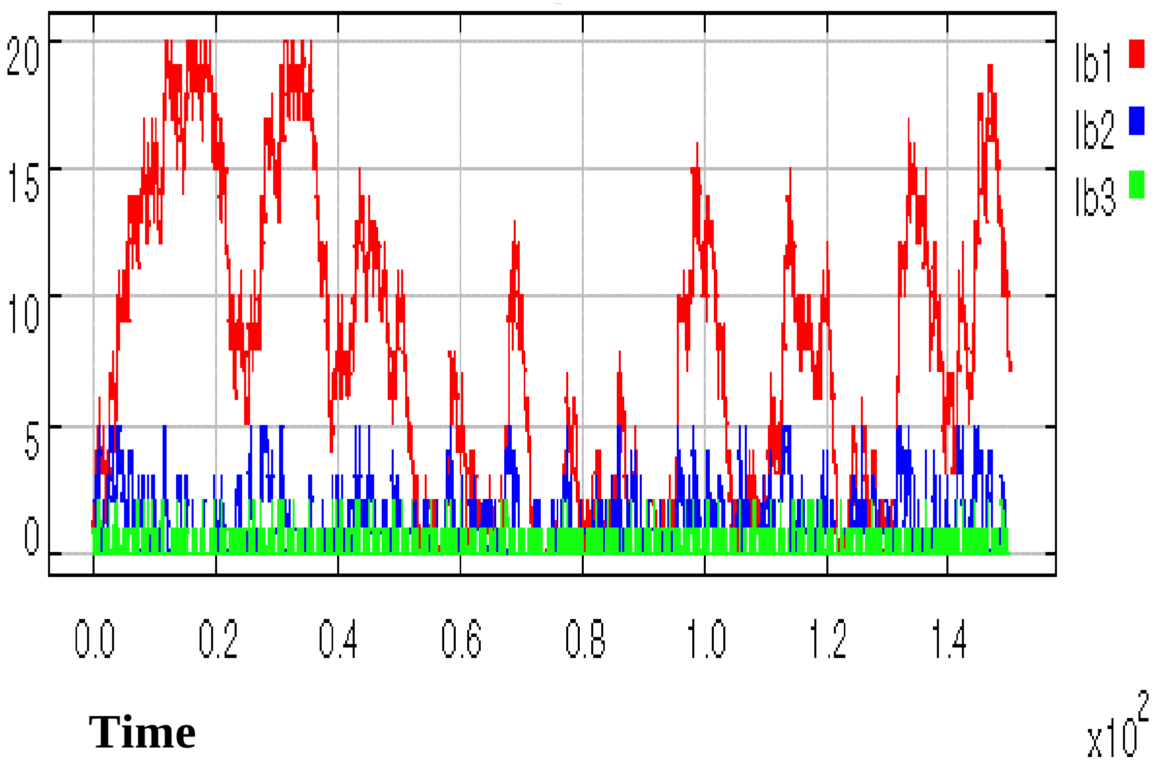


Updown1 Galaxy: SDF in DE Domain



Bucket Length vs. time

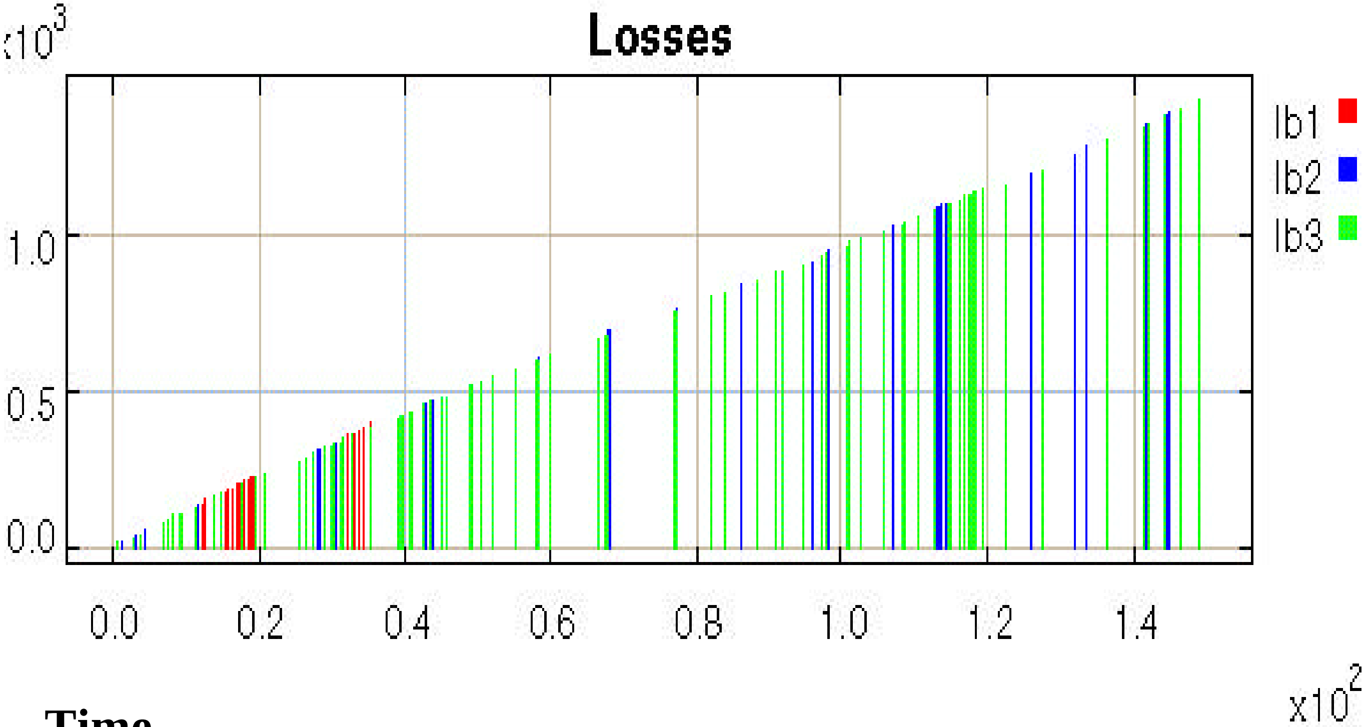
Bucket Length



Time

Packet loss vs. time

Packet Loss



Time

Conclusion

- Helps design of complex systems.
- Decreases complexity by modeling each subsystem individually.
- Different aspects of subsystems can be modeled in detail.
- Can be used in different design groups.
- Efforts of groups can be merged together.