



A Rewriting Based Model for Probabilistic Distributed Object Systems

Koushik Sen

University of Illinois at Urbana-Champaign

Joint work with: Nirman Kumar, José Meseguer, Gul Agha



Introduction

- New model for Probabilistic and Non-deterministic systems.
- Based on rewriting.
- General semantic framework to model various systems.



Motivation

- High level model
- Can achieve semantic unification.
- Inter-operation of tools.



Background

- Rewrite theories.
- Probability theory
 - σ -algebra on a set.
 - Measure function and probability space.
- \mathcal{F} -cover for a σ -algebra.



Presentation Outline

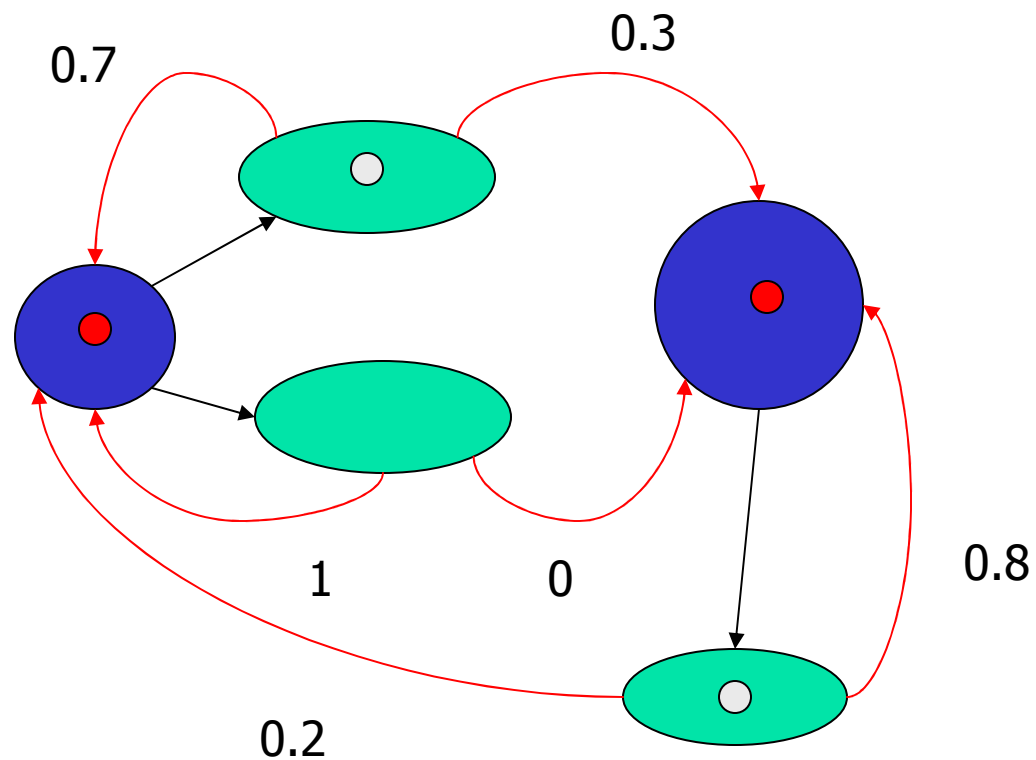
- Introduction
- Motivation
- Background
- **Semantics**
- Example
- Finitary Theories and Pmaude
- Future Work



Operational Semantics of Rewriting

- Steps in rewriting:
 1. Reduce term u to canonical form $[u]_A$
 2. Choose a rewrite rule
 1. $l : t(X) \rightarrow t'(X)$ if $C(X)$
 3. Match pattern $t(X)$ with $[u]_A$ to get substitution $[\theta]_A$ and context $[C]_A$
 4. Put $t'(\theta(X))$ into C to get the term v

Probabilistic Nondeterministic Systems





Operational Semantics of PRwTh

- Rules of the form:
 - $l : t(X) \rightarrow t'(X, Y)$ if $C(X)$ with probability $\pi(X)$
- Steps in probabilistic rewriting:
 1. Reduce term u to canonical form $[u]_A$
 2. Choose an R/A-match
 3. Choose substitution for new variables (probabilistic choice)
 4. Obtain new term v



What is an R/A-match

- Given canonical term $[u]_A$ its R/A-matches are triples
 - $([C]_A, r, [\theta]_A)$ so that lhs of rule r matches $[u]_A$ with substitution $[\theta]_A$ in context $[C]_A$
- R/A-matches represent **nondeterministic choices!**



The probabilistic choice

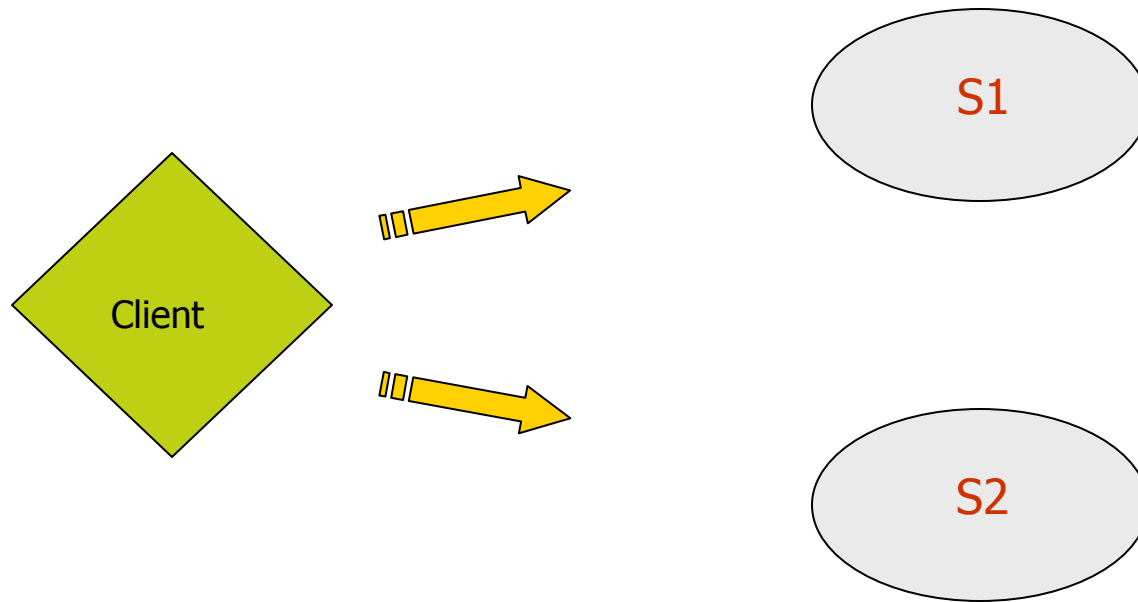
- Choice of the new variables $\mathcal{Y} - [\rho]_A$
- Distribution determined by $\pi([\theta]_A)$
- Transition probability defined by \mathcal{F} - cover : $\alpha_r([\rho]_A)$



Presentation Outline

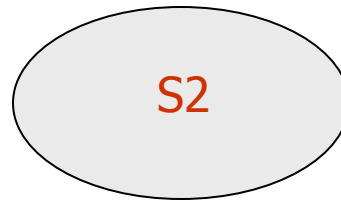
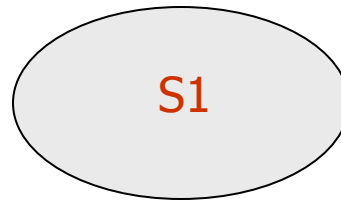
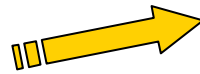
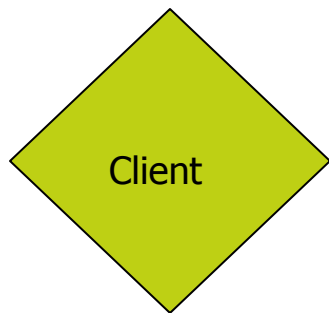
- Introduction
- Motivation
- Background
- Semantics
- **Example**
- Finitary Theories and Pmaude
- Future Work

An example Finally!

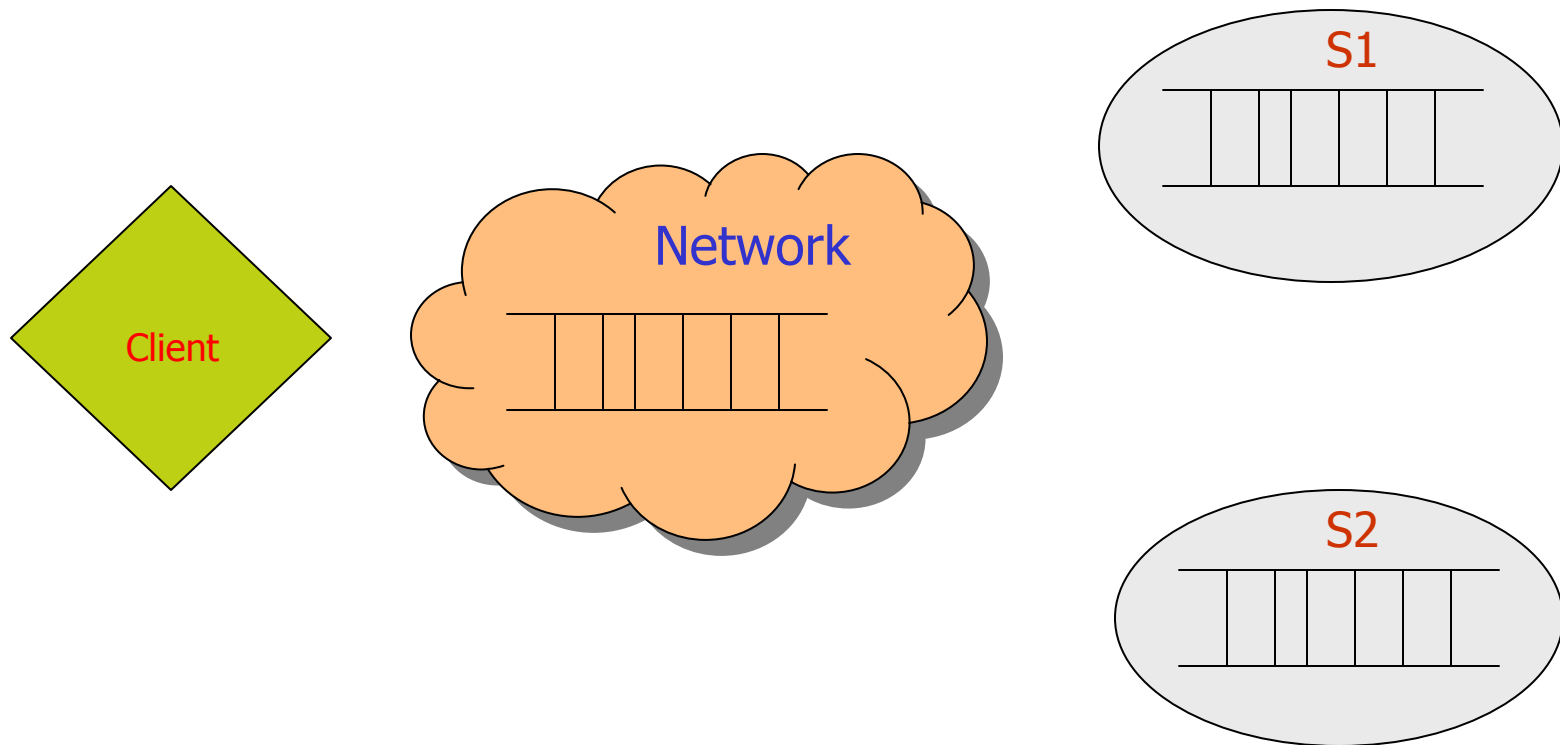


OR

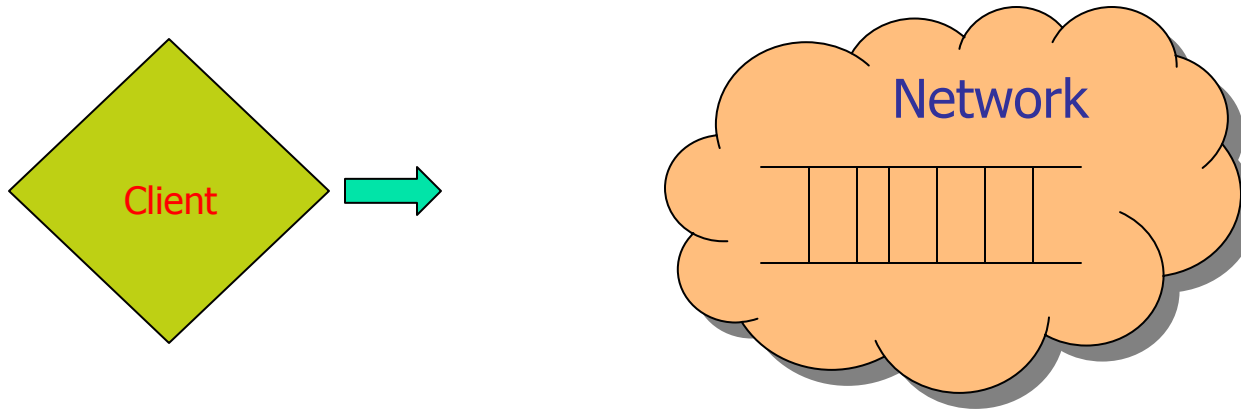
Example contd.



Modeling of the situation



Rules to model sending of packets

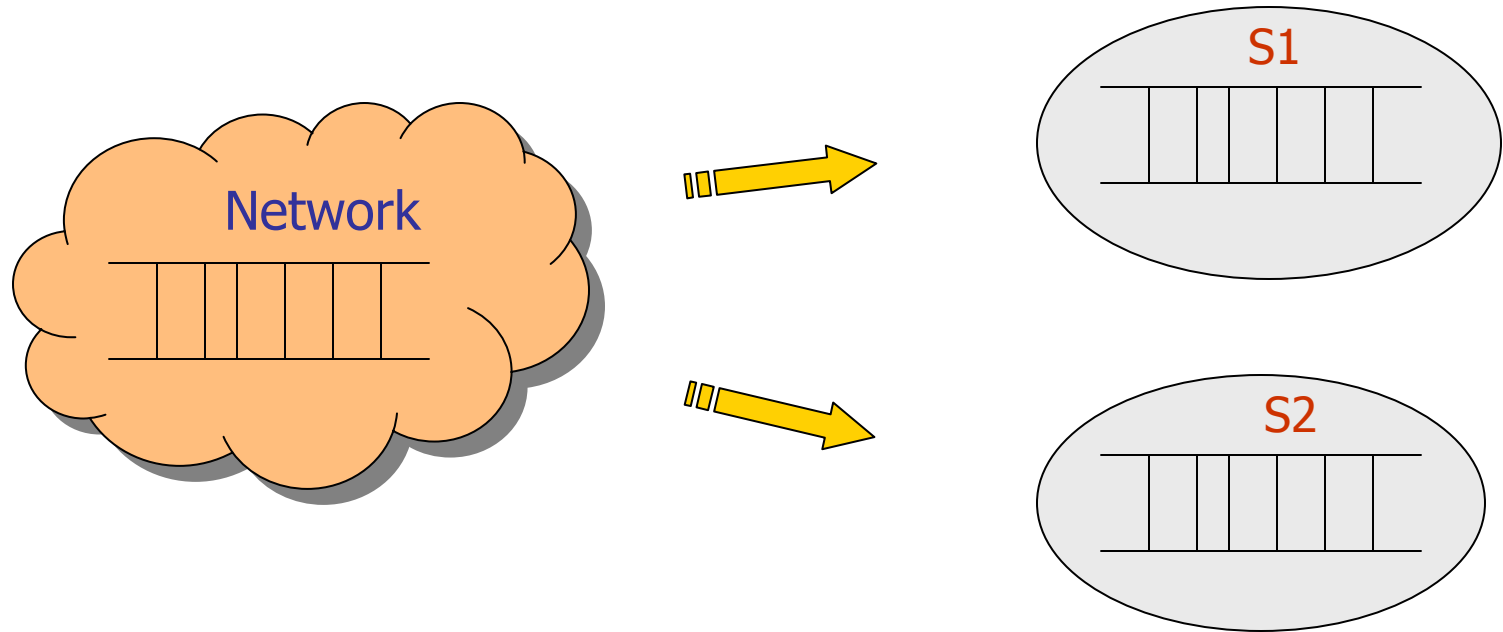


$\text{pri } [\text{req}] : \langle \text{Cl} : \text{Client} \mid \text{sent} : N, \text{svc}_1 : m_1, \text{svc}_2 : m_2 \rangle \langle \text{Nw} : \text{Network} \mid \text{soup} : C \rangle$

\Rightarrow

$\langle \text{Cl} : \text{Client} \mid \text{sent} : N + 1, \text{svc}_1 : m_1, \text{svc}_2 : m_2 \rangle \langle \text{Nw} : \text{Network} \mid \text{soup} : C (\text{Sr} \leftarrow L) \rangle$

Rules to model the delivery of packets

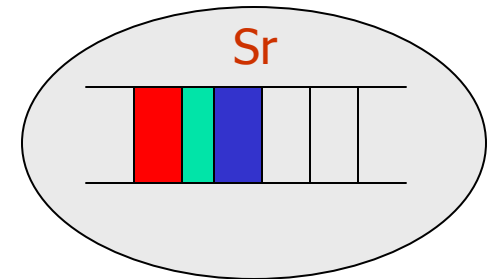
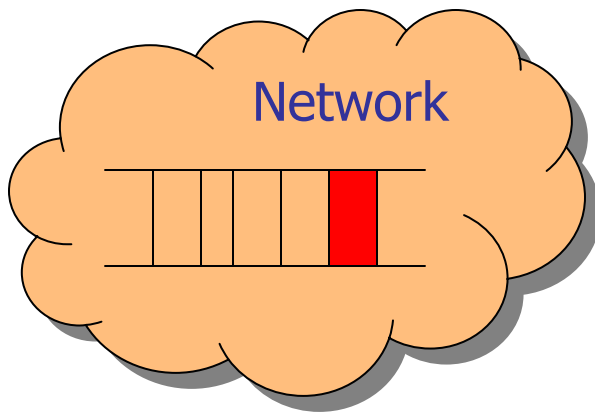
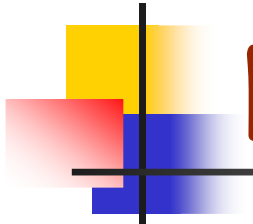


prl [deliver] : $\langle \text{Nw} : \text{Network} \mid \text{soup} : (\text{Sr} \leftarrow \text{L}) \text{C} \rangle \langle \text{Sr} : \text{Server} \mid \text{queue} : \text{Q} \rangle$

\Rightarrow

$\langle \text{Nw} : \text{Network} \mid \text{soup} : \text{C} \rangle \langle \text{Sr} : \text{Server} \mid \text{queue} : \text{Q} \text{ M} \rangle$

Modeling the processing of packets

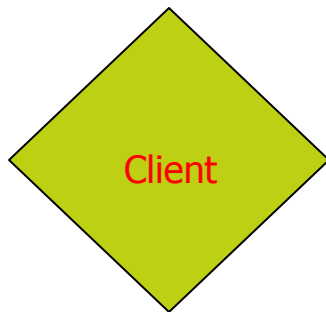


prl [process] : $\langle Sr : Server \mid queue : (Sr \leftarrow L) Q \rangle \langle Nw : Network \mid soup : C \rangle$

\Rightarrow

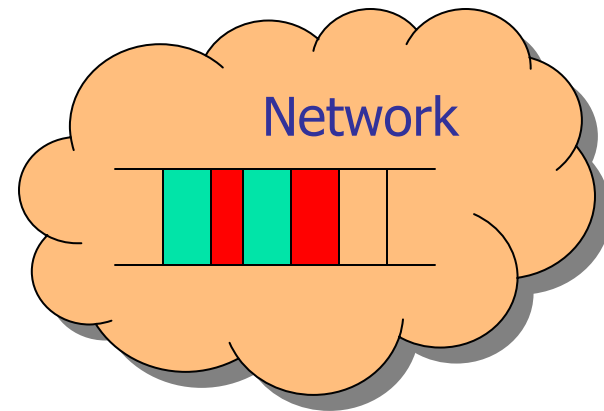
$\langle Sr : Server \mid queue : Q \rangle \langle Nw : Network \mid soup : C M \rangle$

Modeling acknowledgements



acq1

acq2



$\text{cprl} [\text{acq}] : \langle \text{Cl} : \text{Client} \mid \text{svc}_1 : m_1, \text{svc}_2 : m_2 \rangle \langle \text{Nw} : \text{Network} \mid \text{soup} : M \ C \rangle$

\Rightarrow

$\langle \text{Cl} : \text{Client} \mid \text{svc}_1 : m_1 + \delta(i, M, 1), \text{svc}_2 : m_2 + \delta(i, M, 2) \rangle \langle \text{Nw} : \text{Network} \mid \text{soup} : C \rangle$

if $\text{acq}(M)$.



Salient Features of PRwThs

- Expressive power
 - PNS
 - GSMP
 - Object Based distributed systems
- General Semantic framework
 - Interoperation of tools!
- Can model probabilistic actor systems.



Presentation Outline

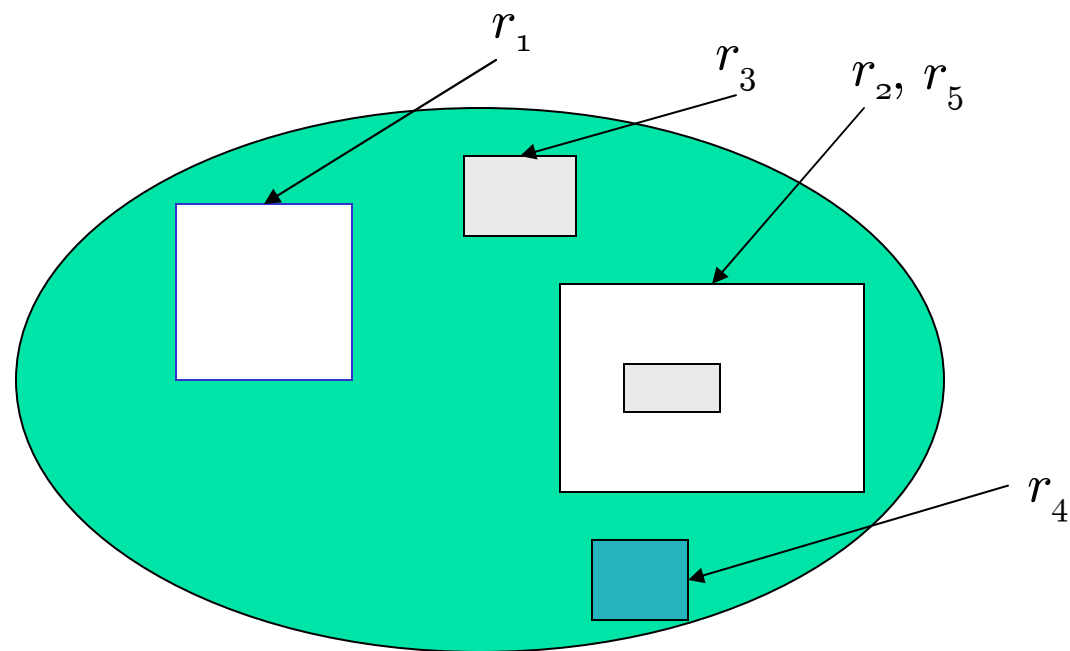
- Introduction
- Motivation
- Background
- Semantics
- Example
- **Finitary Theories and Pmaude**
- Future Work



Finitary PRwThs

- Nondeterminism = choice of $([C]_A, r, [\theta]_A)$
 - can we simplify?
- Design theory so that a rewrite rule describes computation of one component.

Finitary PRwThs contd.



1. Choose the subsystem first
2. Fix a substitution and context
3. Different possible Rhs give different probability choices
4. Normalize rate functions giving probability values

Semantics of Finitary PRwThs

$t(X) \rightarrow t_1(X)$ if $C_1(X)$ [rate $R_1(X)$]

...

$t(X) \rightarrow t_n(X)$ if $C_n(X)$ [rate $R_n(X)$]

$R_i(X)$ are non-negative!



Semantics of Finitary PRwThs contd.

$t(X) \rightarrow \text{proj}(i, (t_1(X), \dots, t_n(X)))$ if $C_1(X)$ or
 $C_2(X) \dots C_n(X)$ with probability $\pi_r(X)$

where

$$\pi_\theta(\{i\}) = ? R_i(\theta(X)) / \sum ? R_j(\theta(X))$$



The Pmaude Simulator

- Implemented as an extension of Full-Maude
- Simulator for a specific adversary for FPRwThs.
- <http://maude.cs.uiuc.edu/pmaude/pmaude.html>
- Web site contains examples and full code



Simulations

- Packets : loads 5 and 10
- Request messages dropped for S1 and S2 with probabilities $2/7$ and $1/6$
- Acknowledgements dropped with probabilities $1/6$ and $1/11$
- S1 : packet dropped with probability $1/7$ when load ≥ 100 and with $3/23$ otherwise.
- S2 : $1/7$ when load ≥ 40 and $3/19$ otherwise.
- Quality = $(svc_1 + svc_2) / \text{sent}$
- Strategy : Sending rate to S2 = twice that of S1



Current and Future Work

- Rewriting with concurrent probabilistic actions.
- Deductive and analytic methods of property verification of probabilistic systems.
- Variations on the theory itself : expressive power.
- Probabilistic validation : Monte-Carlo simulations, acceptance sampling.