Simulation Method for Estimation of Security Overhead of Grid Applications

Wojciech Rząsa[1], Marian Bubak[2,3]
Bartosz Baliś[2,3], Tomasz Szepieniec[2]

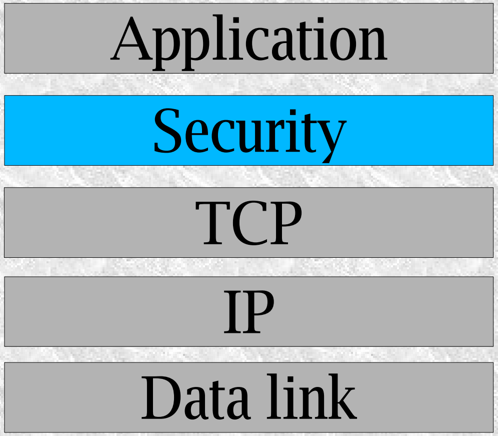
[1]Rzeszow University of Technology [2]Academic Computer Centre – CYFRONET
[3]Institute of Computer Science, AGH

Outline

- Introduction
  - Secure communication in the Grid
  - Security overhead
- Simulation method
  - Model parameters
  - Simulation results
  - Results accuracy
  - Petri Nets – enabling model execution
- Related work
- Future work
- Summary

Secure communication

- GSI [Foster, Kesselman, Tsudik, Tuecke 1998]
  - Solution based on existing standards (eg. TLS, X.509)
  - Introduces communication layer



- Leads to additional consumption of resources
  - CPU
  - RAM
  - Network bandwidth

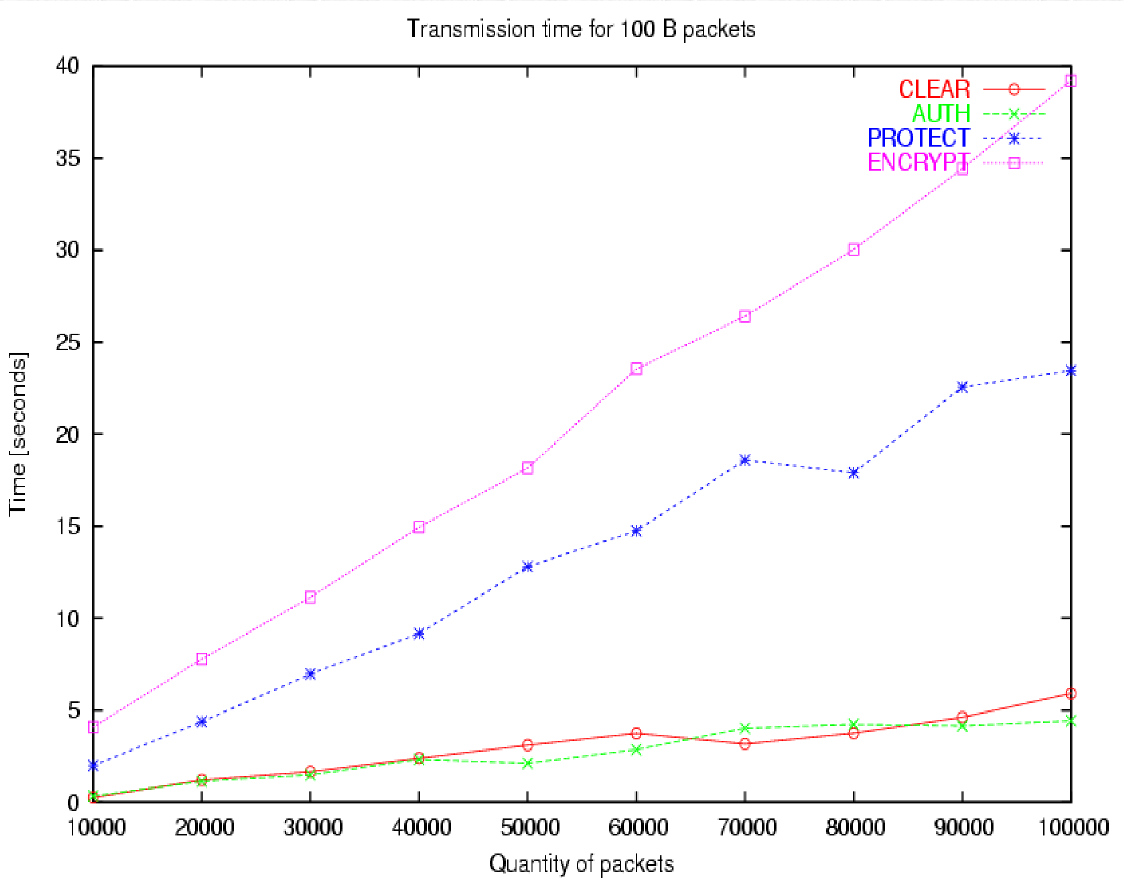
Security overhead

[Baliś, Bubak, Rząsa, Szepieniec 2004]

- Secured connection enables
  - Authentication
  - Data integrity
  - Confidentiality
- Connection establishment

Connections	Requested in 1 second	Established in 1 second	Failed
Secured	896	30	4
Clear	1692	1691	0

- Data transmission



Simulate the application

- Enable application modeling to verify
  - Behavior in time
  - Resource consumption

  depending on communication overhead
- Useful while
  - Application development
  - Legacy software adaptation

Model components

Model of resources

- Parameters of resources
  - Nodes
    - Performance of CPUs
    - RAM
  - Network links
    - Bandwidth
- Topology of resources

Model of application

- Processes/components allocated on the nodes
- Network connections between application processes
  - Security level for individual network connections
- Consumption of resources by application logic
  - Communication dependent
  - Communication independent
- Should not include algorithms

Results of simulation

- Whole application statistics concerning
  - Resources usage
  - Execution time
  - Data transmission
- Accuracy of results
  - Lack of detailed information about application logic
  - Accuracy results from **proper model of communication and interactions** being part of simulation method

Modeling and simulation concept

- *High level application model* provided by the user
- Enable simulation by converting the *High level model* into an **executable formalism**, that is
  - flexible enough to let us model required entities and activities
  - capable of providing required statistics
  - precise enough to provide accurate results

Coloured Petri Nets (CPN) with time

- Formal model
- Capable of modeling
  - Concurrency
  - Synchronization
  - Mutual exclusion
  - Conflict
  - Time
- Moreover
  - CPN are hierarchical
  - CPN allows both: interactive and non-interactive simulation

Petri net

[Murata 1989]

- Defined as five-tuple

$$PN = (P, T, F, W, M_0)$$

Where

P – finite set of places

T – finite set of trasitions

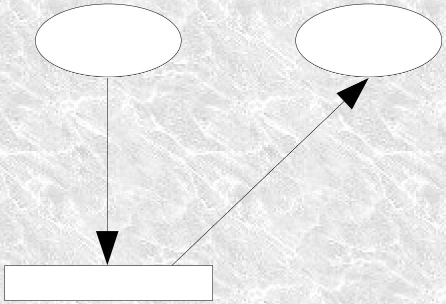
$F \subseteq (P \times T) \cup (T \times P)$ – set of arcs

$W: F \rightarrow \{1,2,3,...\}$ – weight function

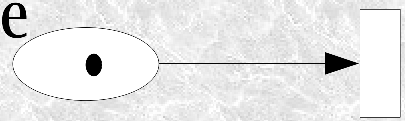
$M_0: P \rightarrow \{0,1,2,...\}$ – initial marking

$P \cap T = \phi$ and $P \cup T \neq \phi$

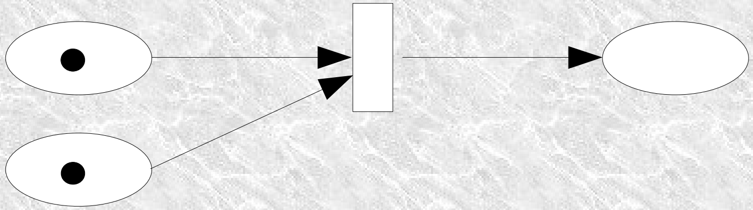
- Graph consists of
  - Places
  - Arcs
  - Transitions



- Tokens reside in places



- Tokens enable transitions
- Enabled transitions can fire



Petri net

[Murata 1989]

- Defined as five-tuple

$$PN = (P, T, F, W, M_0)$$

Where

P – finite set of places

T – finite set of trasitions

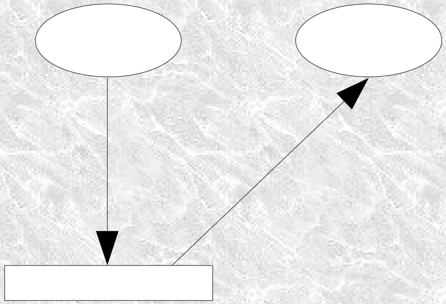
$F \subseteq (P \times T) \cup (T \times P)$ – set of arcs

$W: F \rightarrow \{1,2,3,\ldots\}$ – weight function

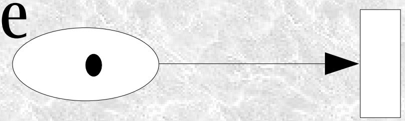
$M_0: P \rightarrow \{0,1,2,\ldots\}$ – initial marking

$P \cap T = \phi$ and $P \cup T \neq \phi$

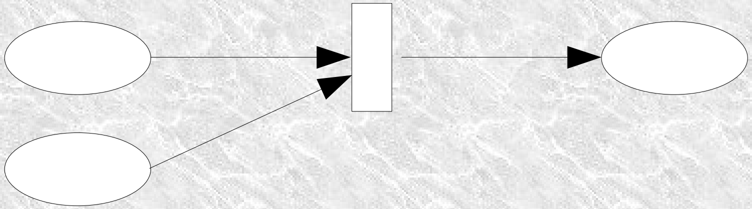
- Graph consists of
  - Places
  - Arcs
  - Transitions



- Tokens reside in places



- Tokens enable transitions
- Enabled transitions can fire



Petri net

[Murata 1989]

- Defined as five-tuple

$$PN = (P, T, F, W, M_0)$$

Where

P – finite set of places

T – finite set of trasitions

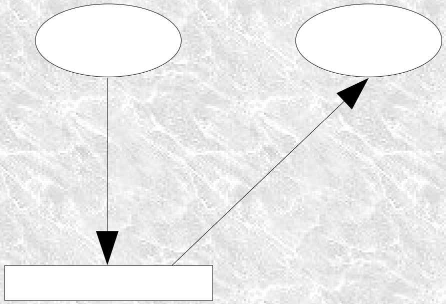
$F \subseteq (P \times T) \cup (T \times P)$ – set of arcs

$W: F \rightarrow \{1,2,3,\ldots\}$ – weight function

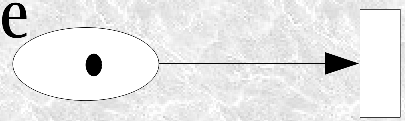
$M_0 : P \rightarrow \{0,1,2,\ldots\}$ – initial marking

$P \cap T = \phi$ and $P \cup T \neq \phi$

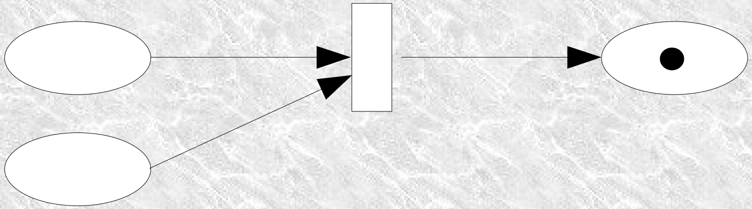
- Graph consists of
  - Places
  - Arcs
  - Transitions



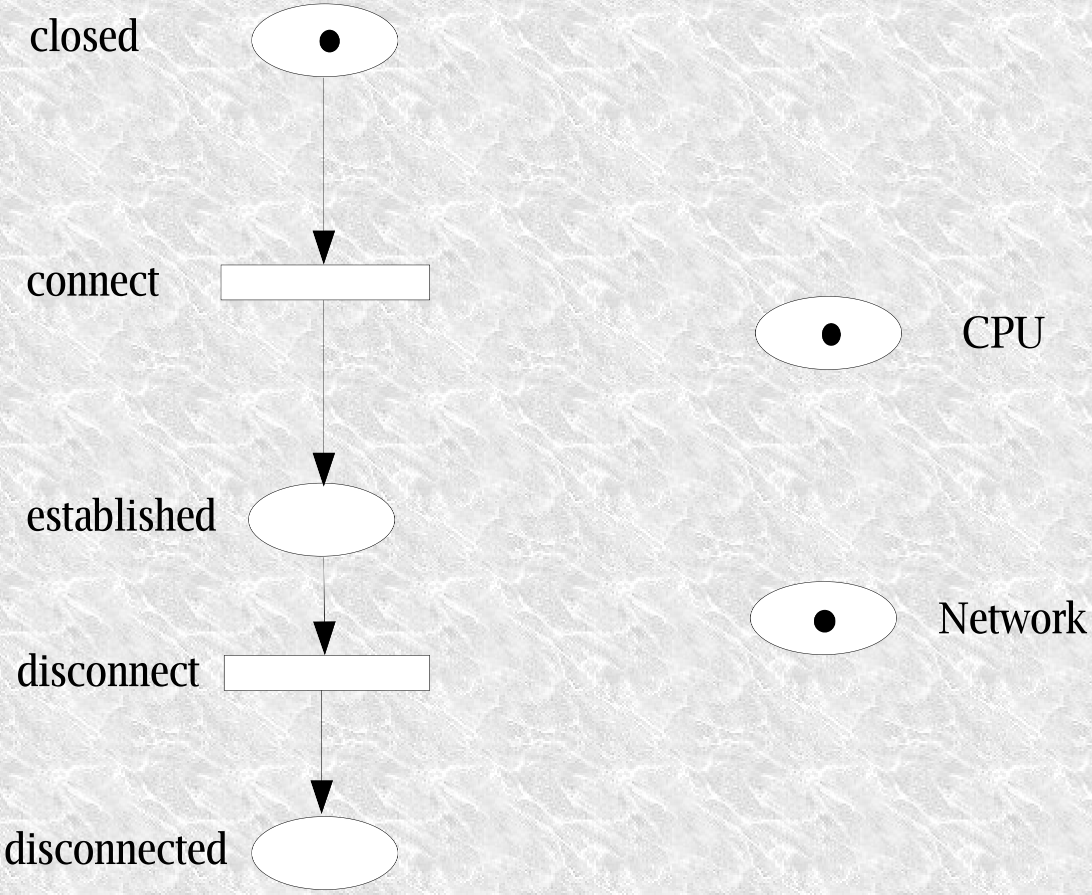
- Tokens reside in places



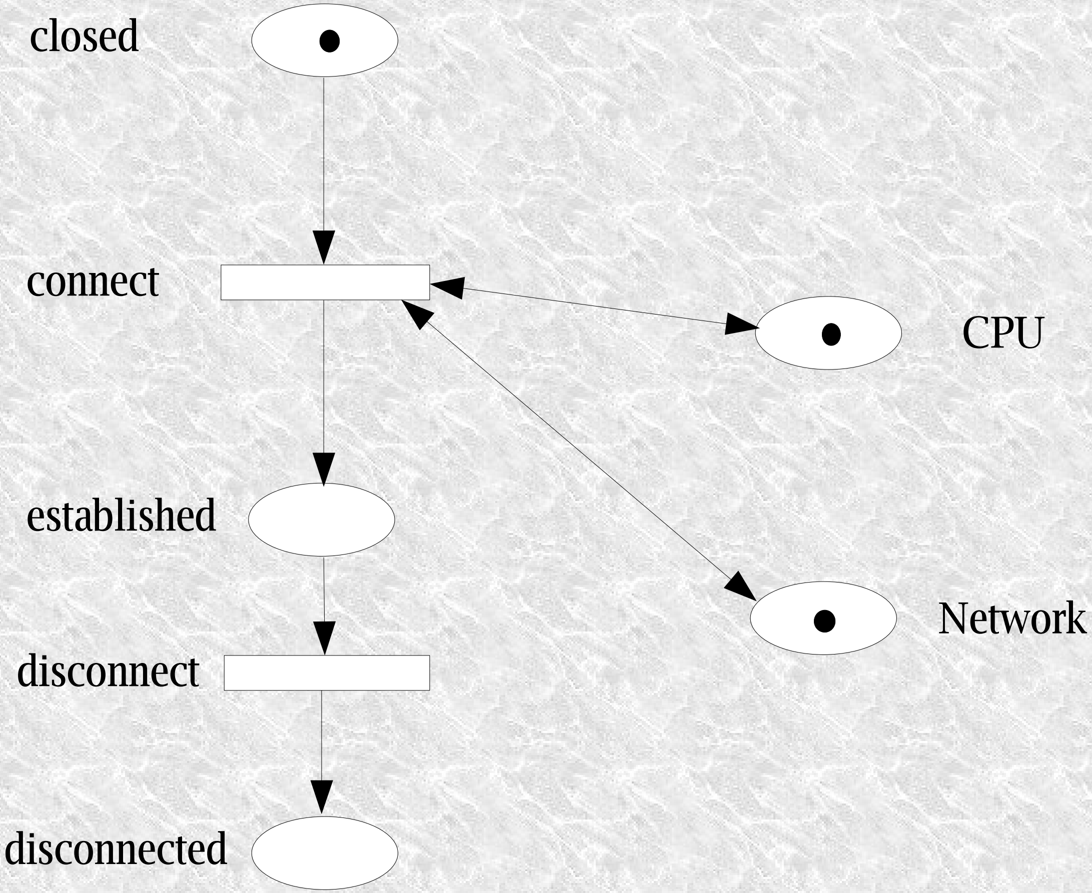
- Tokens enable transitions
- Enabled transitions can fire



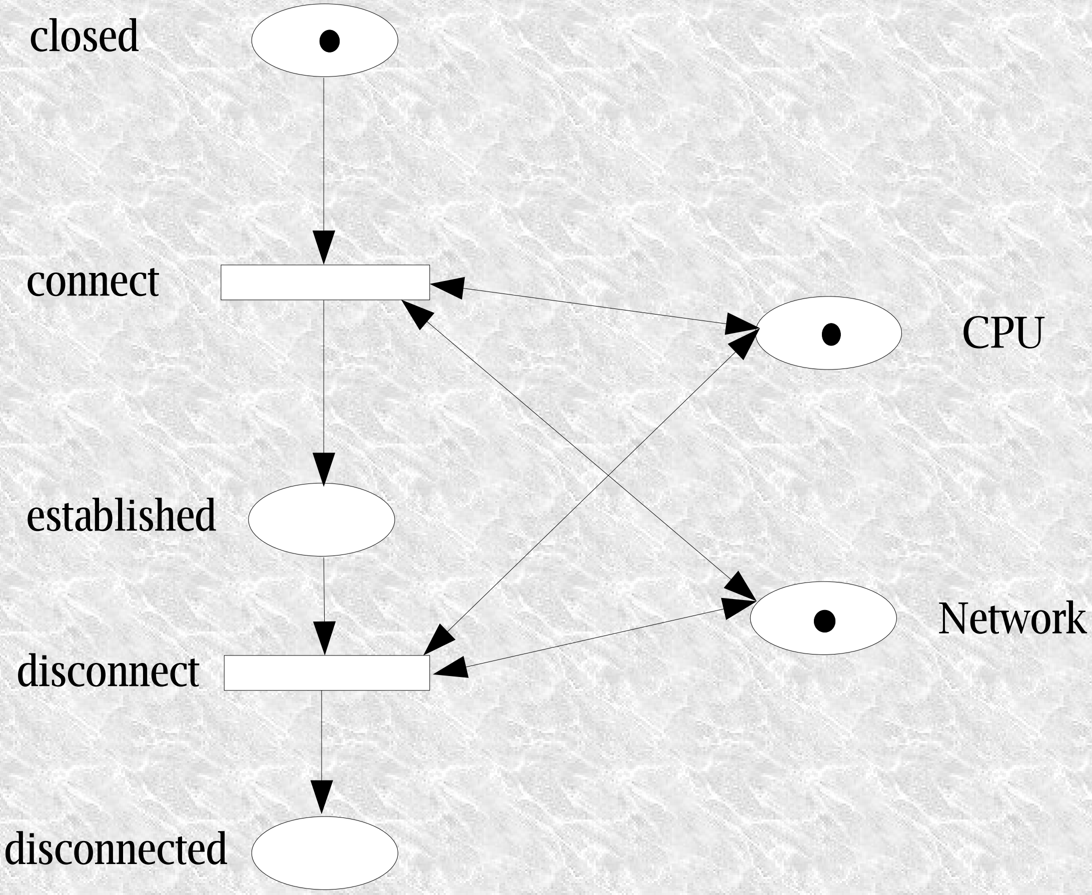
PN Examples - resources



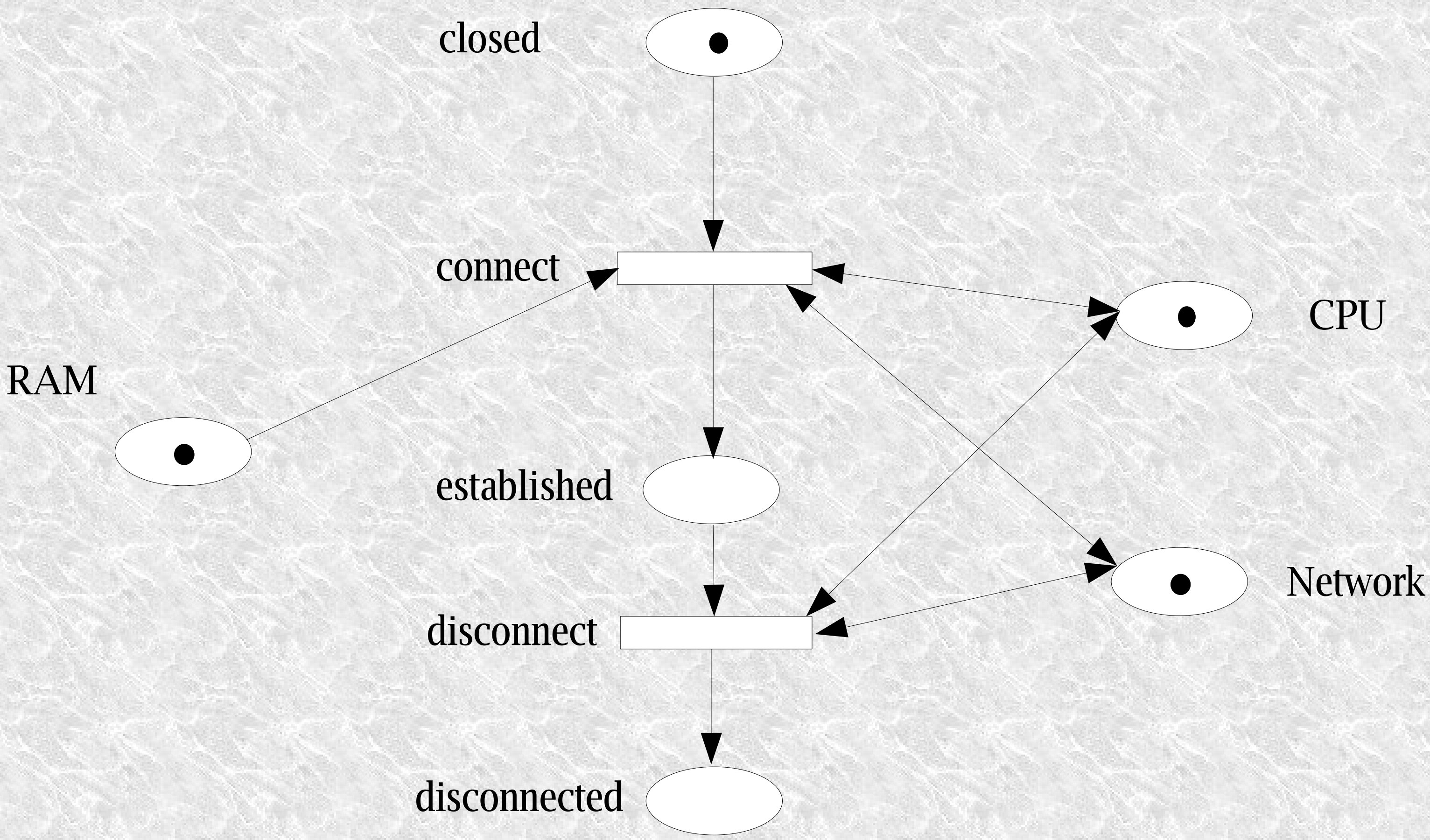
PN Examples - resources



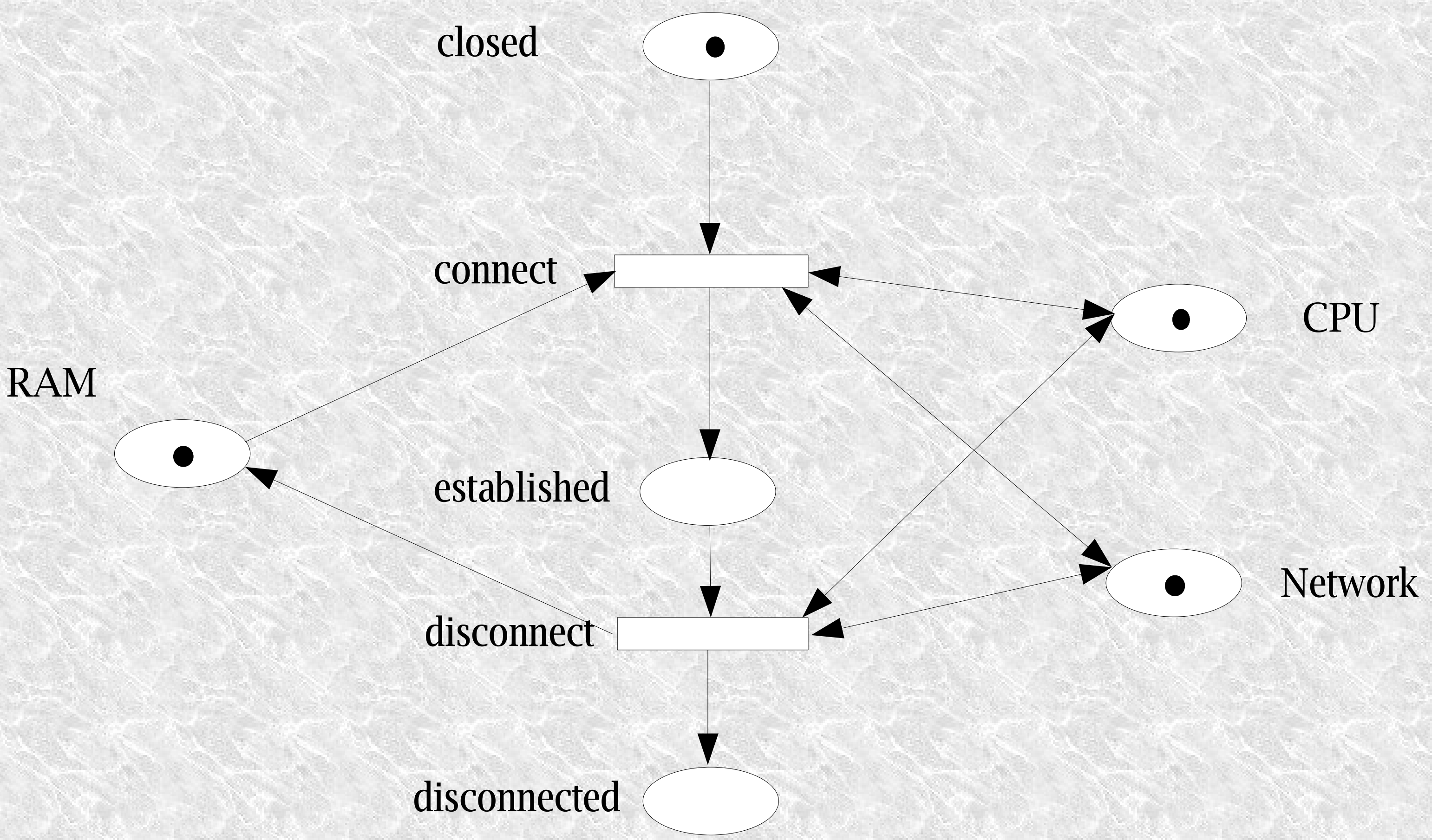
PN Examples - resources



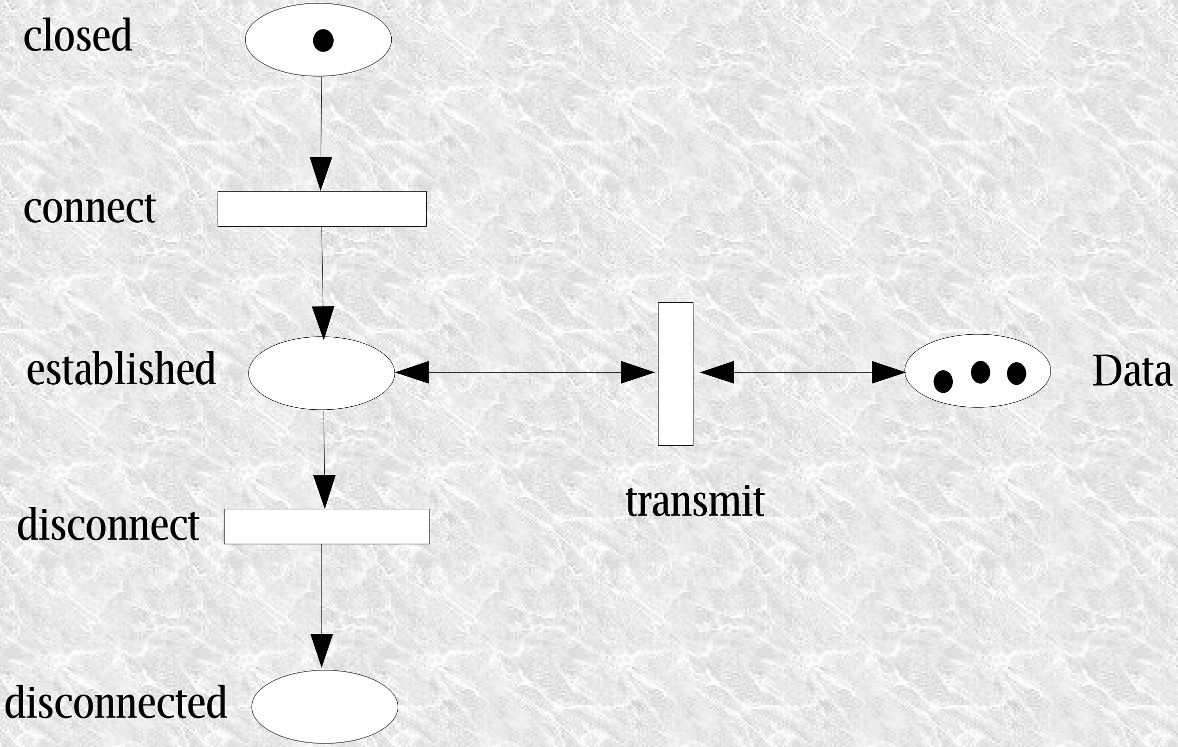
PN Examples - resources



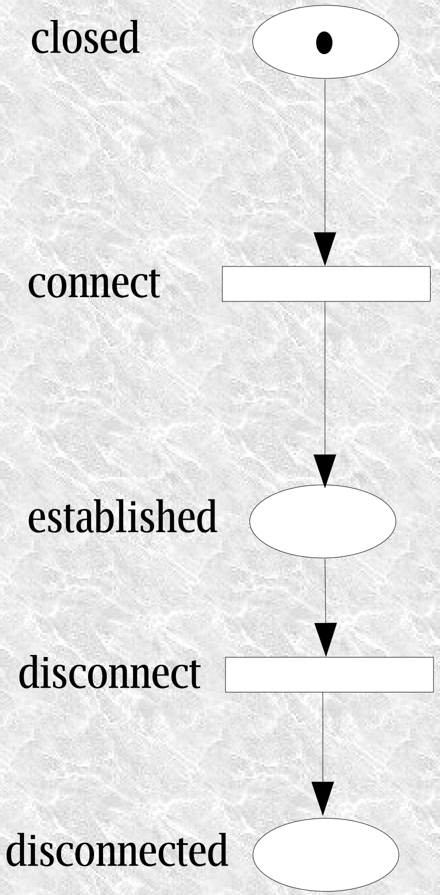
PN Examples - resources



PN Examples – data transmission

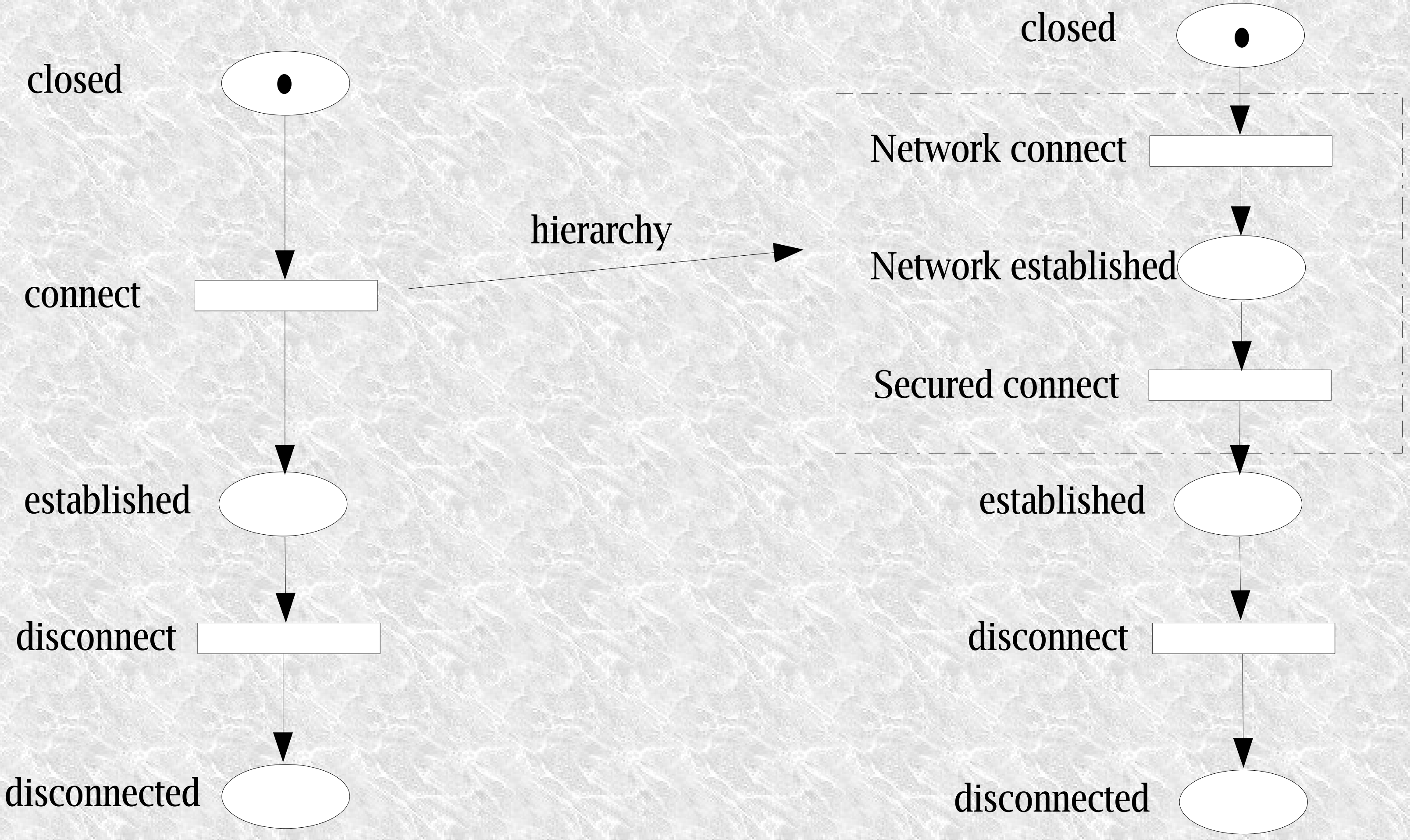


PN Examples - hierarchy



Network connection model

PN Examples - hierarchy



Network connection model

Secured connection model

Coloured Petri Net

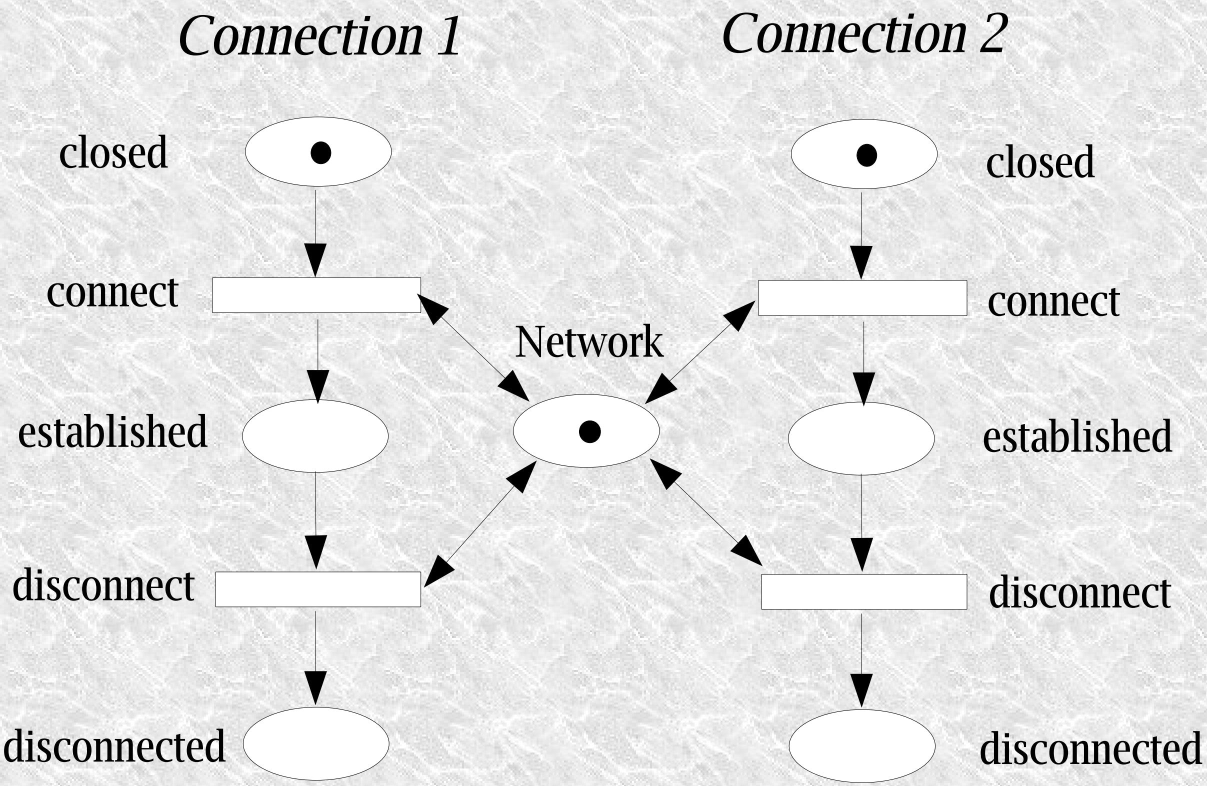
[Jensen 1994]

- Classical PN extended by
  - Colour sets – data types
  - Colours of tokens – values
  - Guards defined for transitions
  - Arc expressions
- More compact and transparent model

Coloured Petri Net - example

More compact and transparent graph

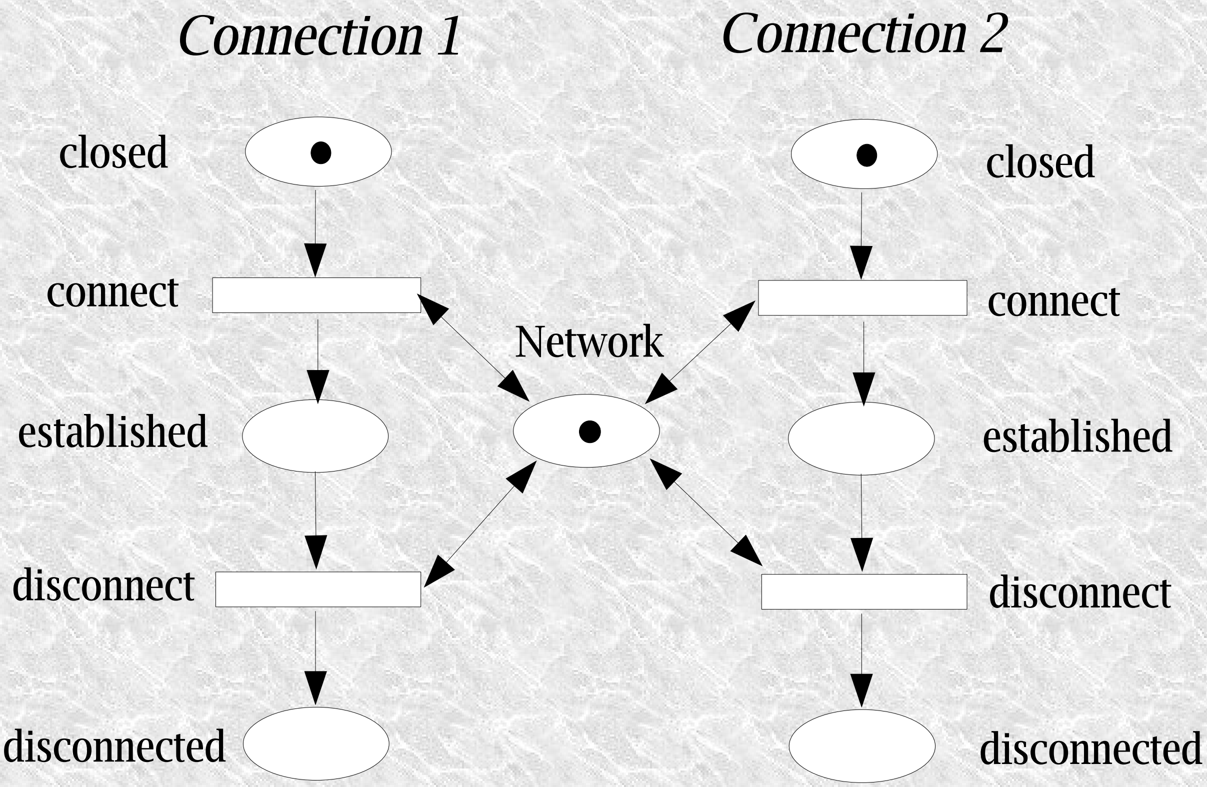
Petri Net



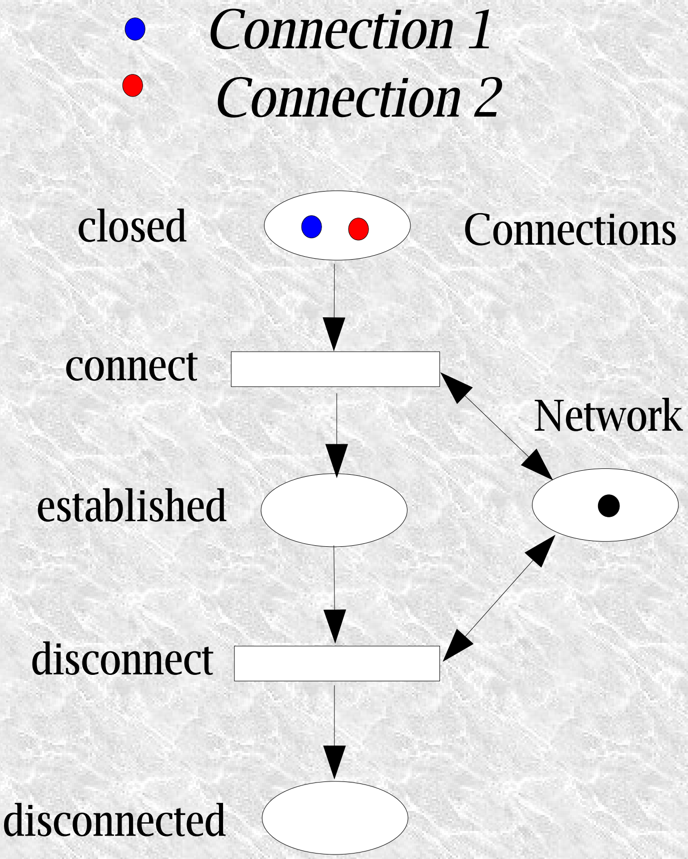
Coloured Petri Net - example

More compact and transparent graph

Petri Net

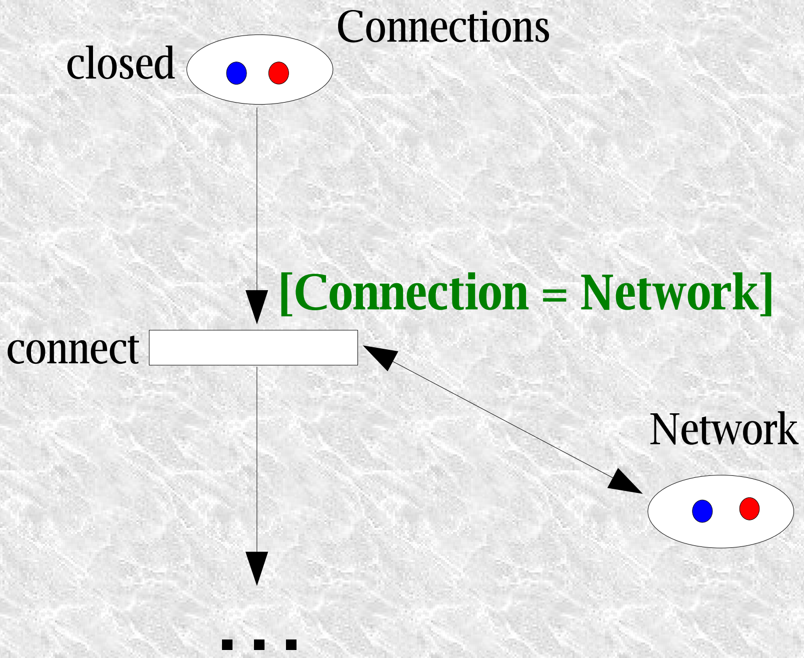


Coloured Petri Net

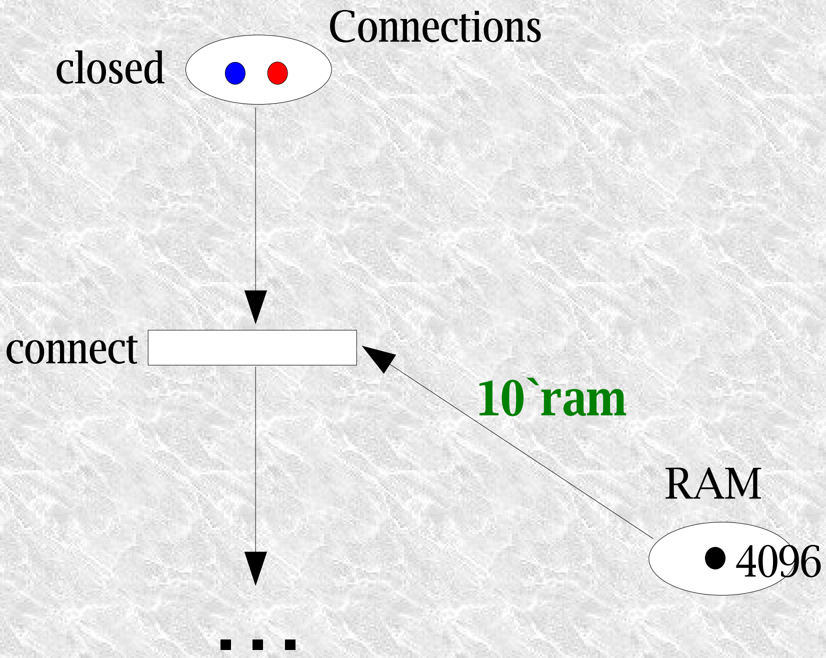


CPN – Guards and Inscriptions

Guard



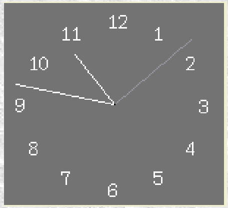
Inscription



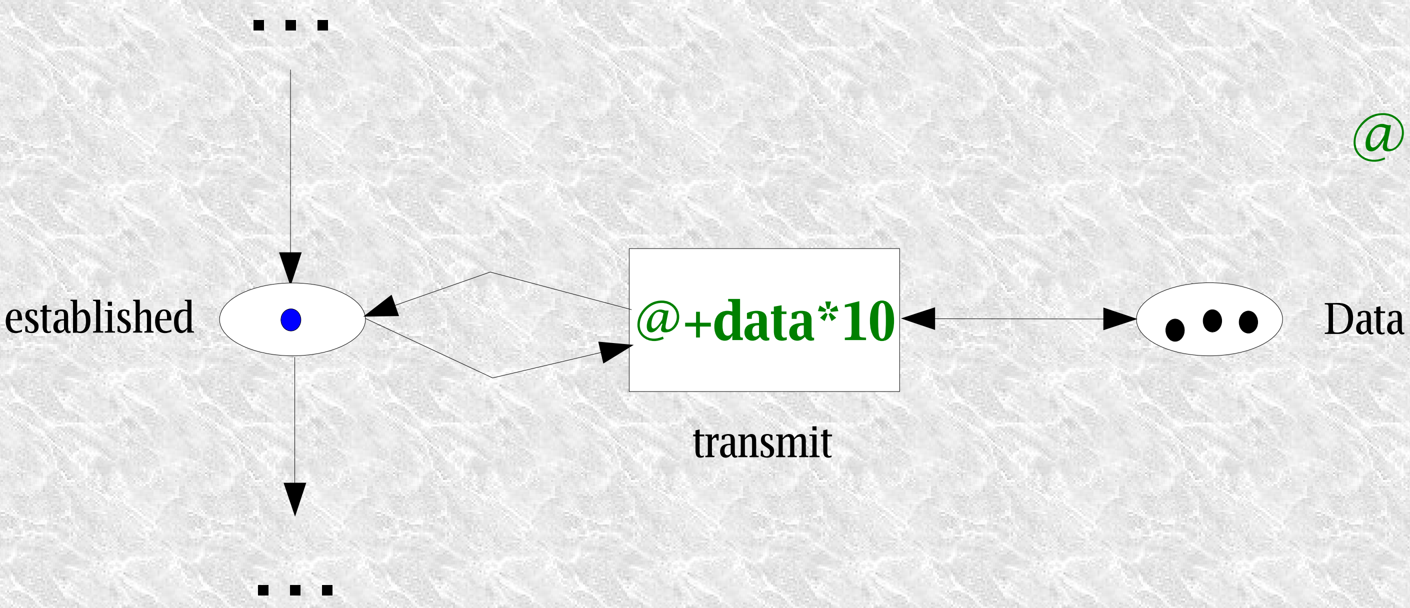
CPN with time

[Jensen 1995/96]

- Tokens with timestamps
- Timestamps affects availability of tokens



@ – global clock



Related work (examples)

- Focus on wide application area
  - Simulink (Matlab)
  - CSIM [Schwetman 2001]
  - C++SIM, JavaSIM [Little, McCue 1994]
- Network protocols simulators
  - cnet http://www.csse.uwa.edu.au/cnet/
  - ns2 http://www.isi.edu/nsnam/ns/
- Distributed systems and Grid simulators
  - GridSim based on SimJava [Buyya 2002]
  - Prophet [Fahringer]
    - UML model transformed to CSIM

Future work

- Distributed application and infrastructure model (CPN with time)
  - nearly finished
- *High level application model* description (XML?)
  - guidelines formulated
- *High level model* to CPN model transformation
- Case study
  - Model calibration
  - Verification of simulation results

Summary

- The goal is to

Enable simulation of distributed applications

in order to estimate security overhead

using

limited information about the application logic and

precise model of communication and interactions

- To facilitate
  - distributed applications development
  - adaptation of legacy software
- Method should provide possibly detailed statistics about
  - resource consumption
  - execution and communication time