Performance modelling

with PEPA nets and PRISM

S. Gilmore, J. Hillston, L. Kloul, M. Ribaudo

Outline

- **PEPA nets: informal introduction**
- **PEPA nets: few formal definitions**
- **Simple example: mobile agent**
- **From PEPA nets to PRISM**
- **Complex example: mobile IP**

PEPA nets: informal introduction

Stochastic (coloured) Petri Nets

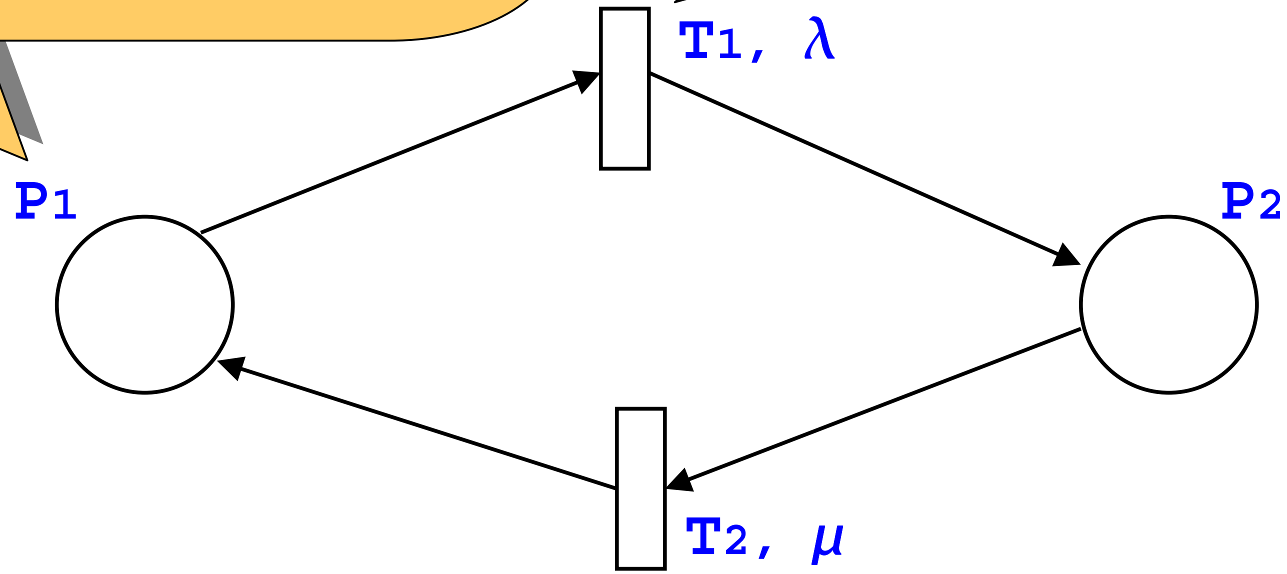
PEPA

PEPA nets

PEPA nets: informal introduction

The tokens are PEPA components that perform local activities and can move from one place to another …

Transition names are labelled with activity names



PEPA nets: informal introduction

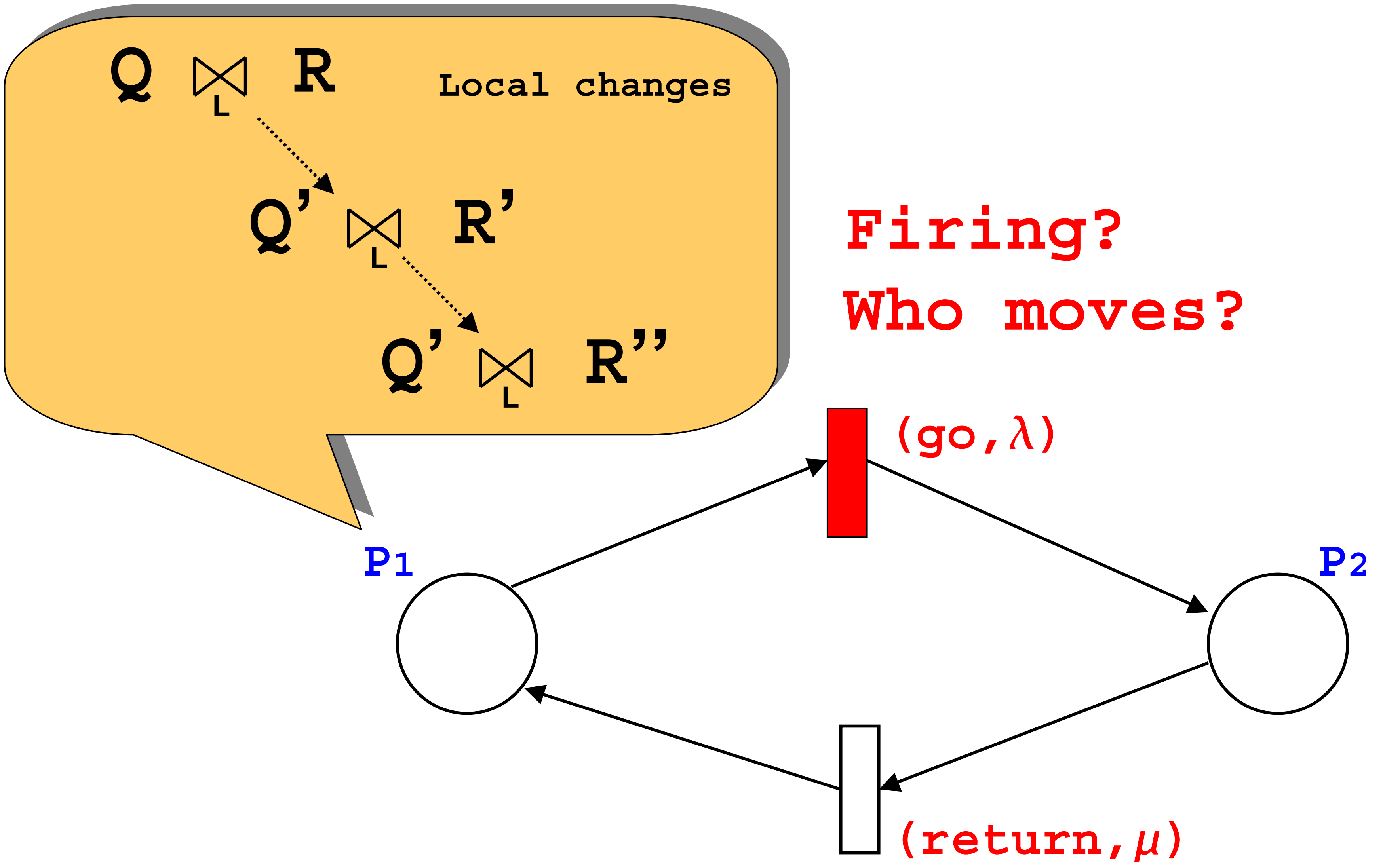
- PEPA components perform activities to represent state changes

- … in PEPA nets we distinguish between two types of changes …

  ✓ "local" changes
  (transitions of PEPA components)

  ✓ "global" changes
  (net firings)

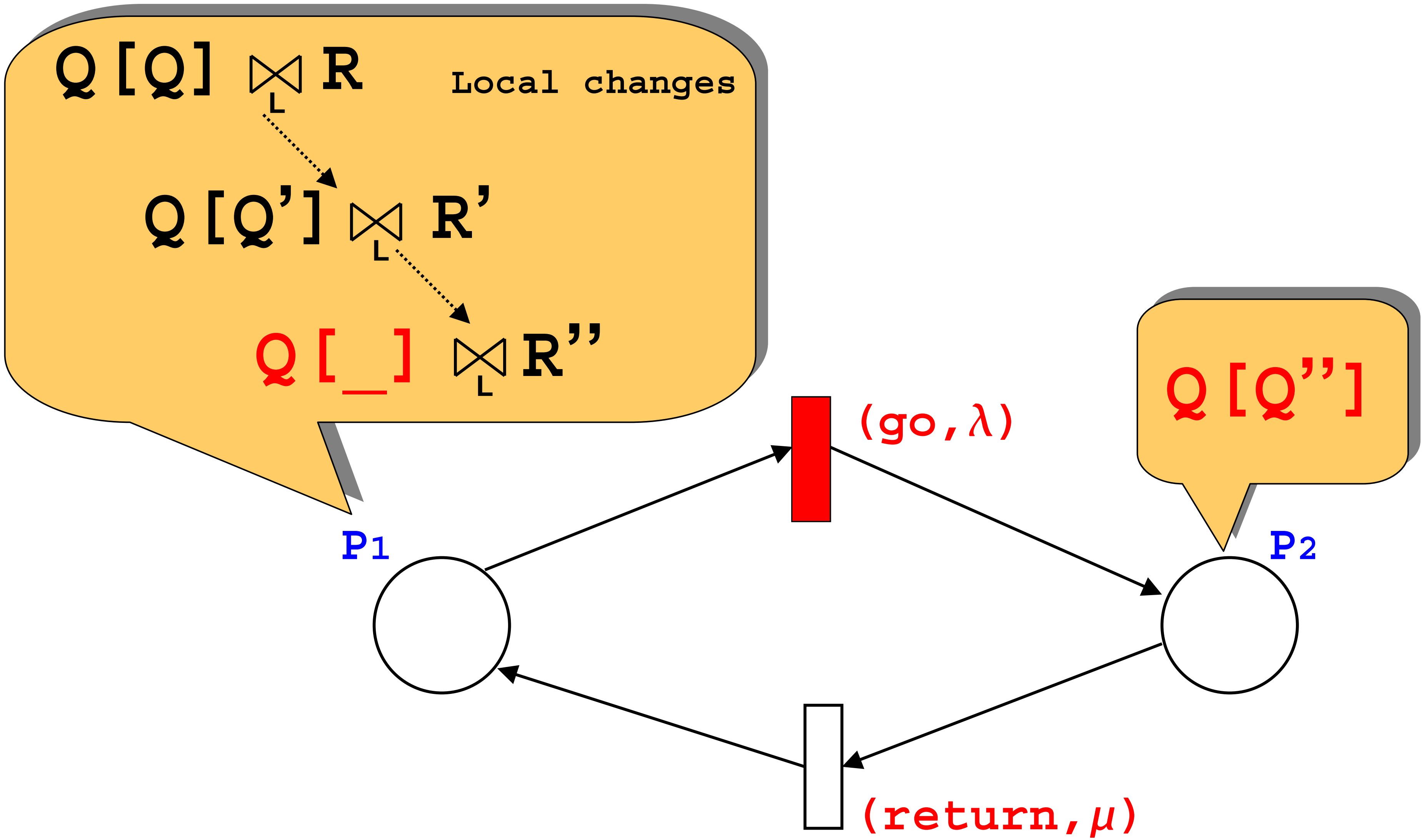
PEPA nets: informal introduction



PEPA nets: informal introduction

- There is a PEPA context at each place of the net
- A PEPA context consists of
  - ✓ static components
  - ✓ cells [ ]

PEPA nets: informal introduction



PEPA nets: informal introduction

- Some assumptions

  ✓ Components can cooperate only when they are in the same place

  ✓ It is NOT possible for one component to cooperate with another component AND transfer to another place

PEPA nets: syntax

$$
\begin{array}{lcll}
S & ::= & (\alpha, r).S & \text{(prefix)} \\
  & \mid & S + S & \text{(choice)} \\
  & \mid & I & \text{(identifier)}
\end{array}
$$

$$
\begin{array}{lcll}
P & ::= & P \underset{L}{\bowtie} P & \text{(cooperation)} \\
  & \mid & P/L & \text{(hiding)} \\
  & \mid & P[C] & \\
  & \mid & I &
\end{array}
$$

$$
\begin{array}{lcll}
C & ::= & `\_\text{'} & \text{(empty)} \\
  & \mid & S & \text{(full)}
\end{array}
$$

PEPA nets: markings and places

$$M \quad ::= \quad (M_{\mathbf{P}}, \ldots) \qquad \text{(marking)}$$

$$M_{\mathbf{P}} \quad ::= \quad \mathbf{P}[C, \ldots] \qquad \text{(place marking)}$$

$$\mathbf{P}[C, \ldots] \stackrel{def}{=} P[C] \underset{L}{\bowtie} P \qquad \text{(place defn)}$$

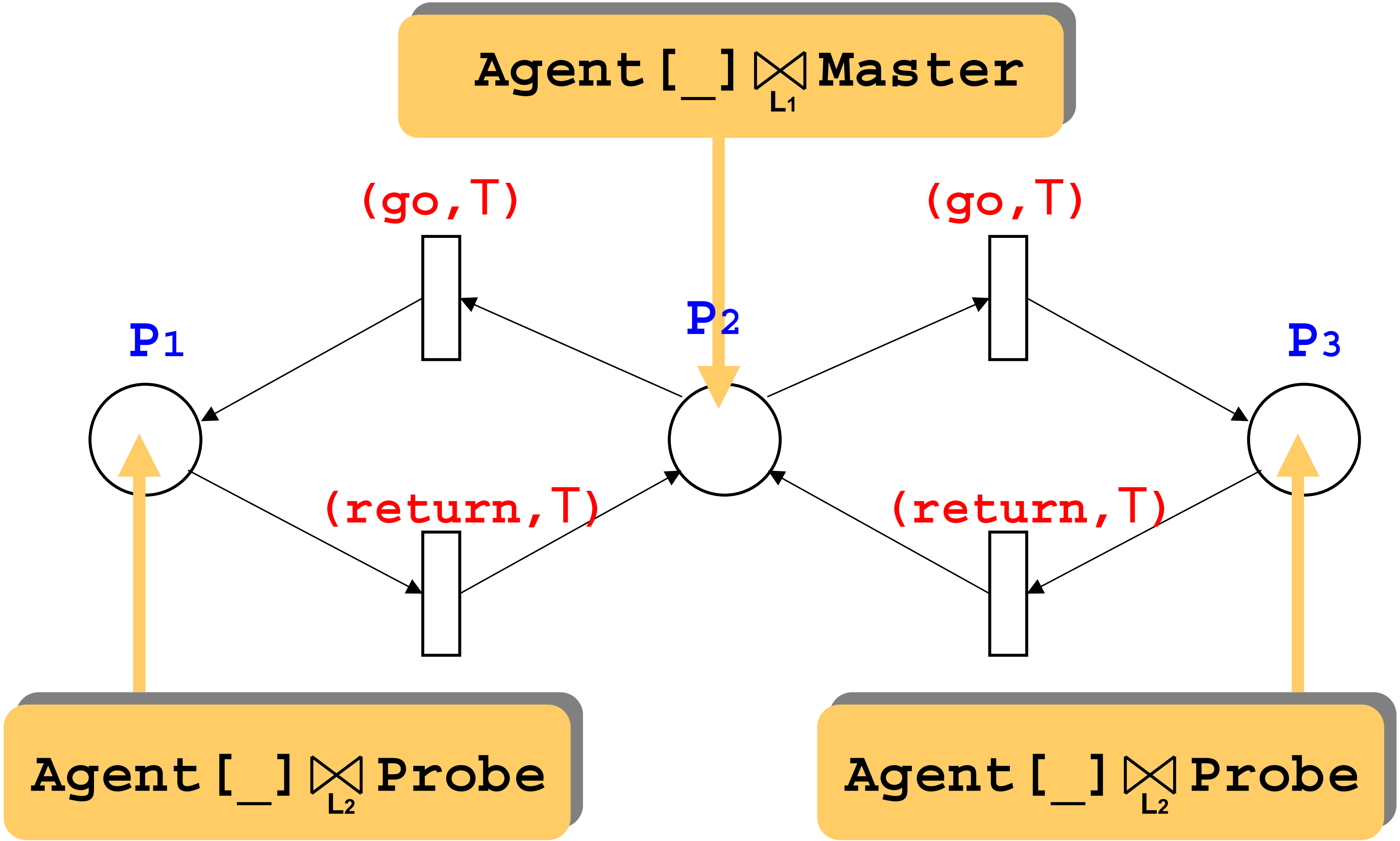
Simple example: mobile agent

- **A mobile software agent visits three sites, where it interacts with static software components**

- **In two sites the agent interrogates a network sensor for data** (on recent patterns of network traffic)

- **In the other site, the agent dumps the data to a master sensor**

Simple example: Mobile agent



Simple example: Mobile agent

$$
\begin{aligned}
\mathbf{Agent} &\stackrel{def}{=} (go,\lambda).\mathbf{Agent'} \\
\mathbf{Agent'} &\stackrel{def}{=} (interrogate,r_i).\mathbf{Agent''} \\
\mathbf{Agent''} &\stackrel{def}{=} (return,\mu).\mathbf{Agent'''} \\
\mathbf{Agent'''} &\stackrel{def}{=} (dump,r_d).\mathbf{Agent}
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{Master} &\stackrel{def}{=} (dump,\top).\mathbf{Master'} \\
\mathbf{Master'} &\stackrel{def}{=} (analyse,r_a).\mathbf{Master}
\end{aligned}
$$

$$
\mathbf{Probe} \stackrel{def}{=} (monitor,r_m).\mathbf{Probe} + (interrogate,\top).\mathbf{Probe}
$$

PRISM

- **Probabilistic model checker**
  probabilistic temporal logic, PCTL and CSL

- **Supports three models**
  DTMC, MDP, CTMC

- **Compact state representation** (BDD)

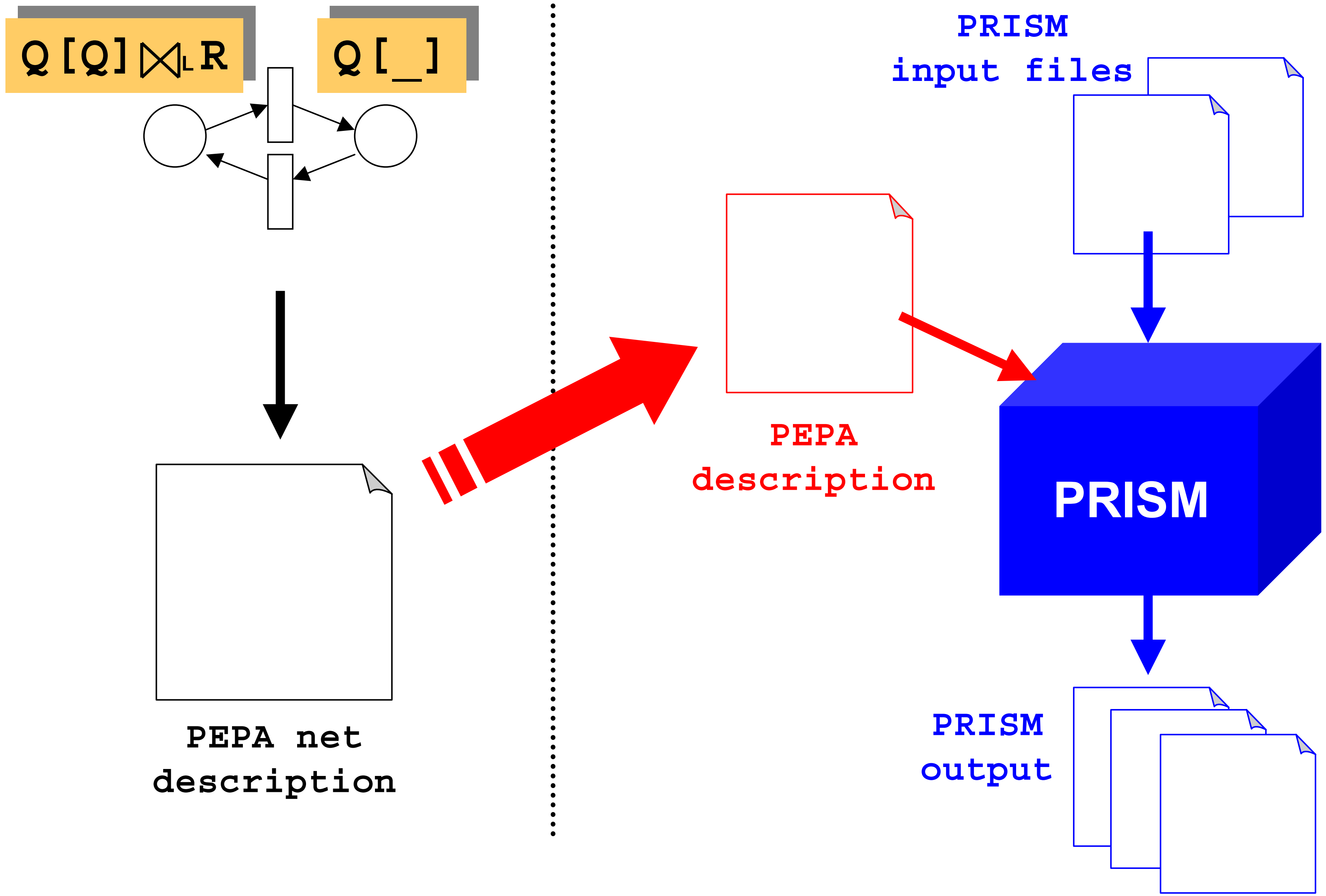
- **Input to the PRISM tool**
  1. **description of the system**
  2. **set of properties to be checked**

From PEPA to PRISM

- **A compiler exists for translating PEPA models** (a subset of PEPA) **into PRISM models**

- **… then the models can be analysed with the PRISM tool**
  - ✓ **The steady-state probability distribution for the underlying CTMC can be automatically derived**
  - ✓ **Properties can be verified**

From PEPA nets to PRISM



From PEPA nets to PEPA

- We need to map the net structure into (standard) PEPA components

- Problems

  ✓ What happens to different transitions with the same associated label?

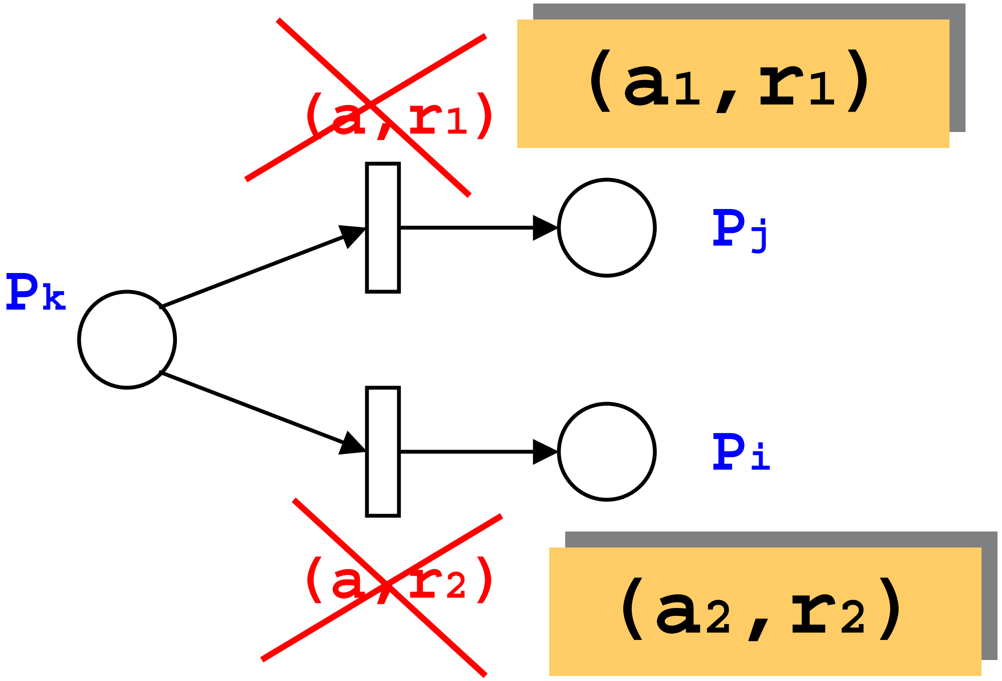
  ✓ What happens to replica of the same static component, resident in different places

  ✓ What about cells?

From PEPA nets to PEPA

- Steps of the translation algorithm

0. Preprocessing



From PEPA nets to PEPA

1. Translation of static components

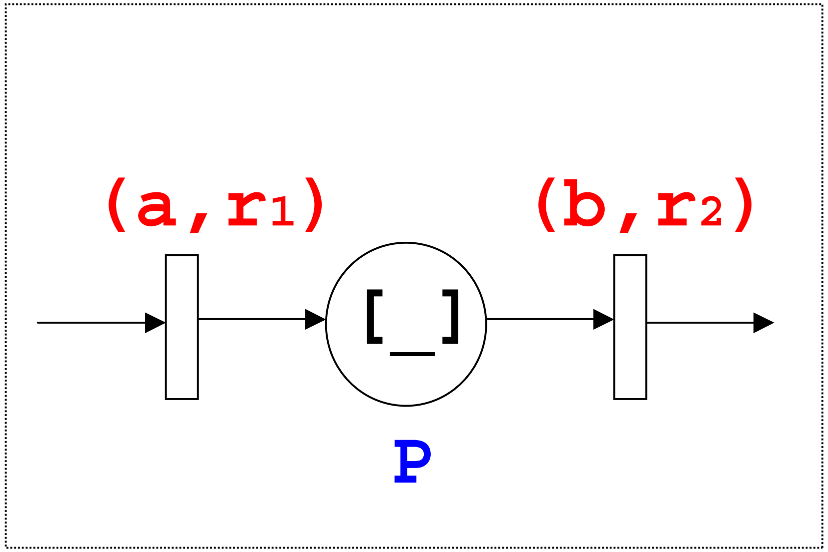
In order to avoid wrong synchronisations we need to distinguish replicas of the same static component.

This is done by renaming action types and derivatives

From PEPA nets to PEPA

2. Translation of cells

A new PEPA component need to be defined for each cell i within each place P



$$Cell_{i0} \stackrel{def}{=} (a, r_1).Cell_{i1}$$

$$Cell_{i1} \stackrel{def}{=} (b, r_2).Cell_{i0}$$

From PEPA nets to PEPA

3. Translation of tokens

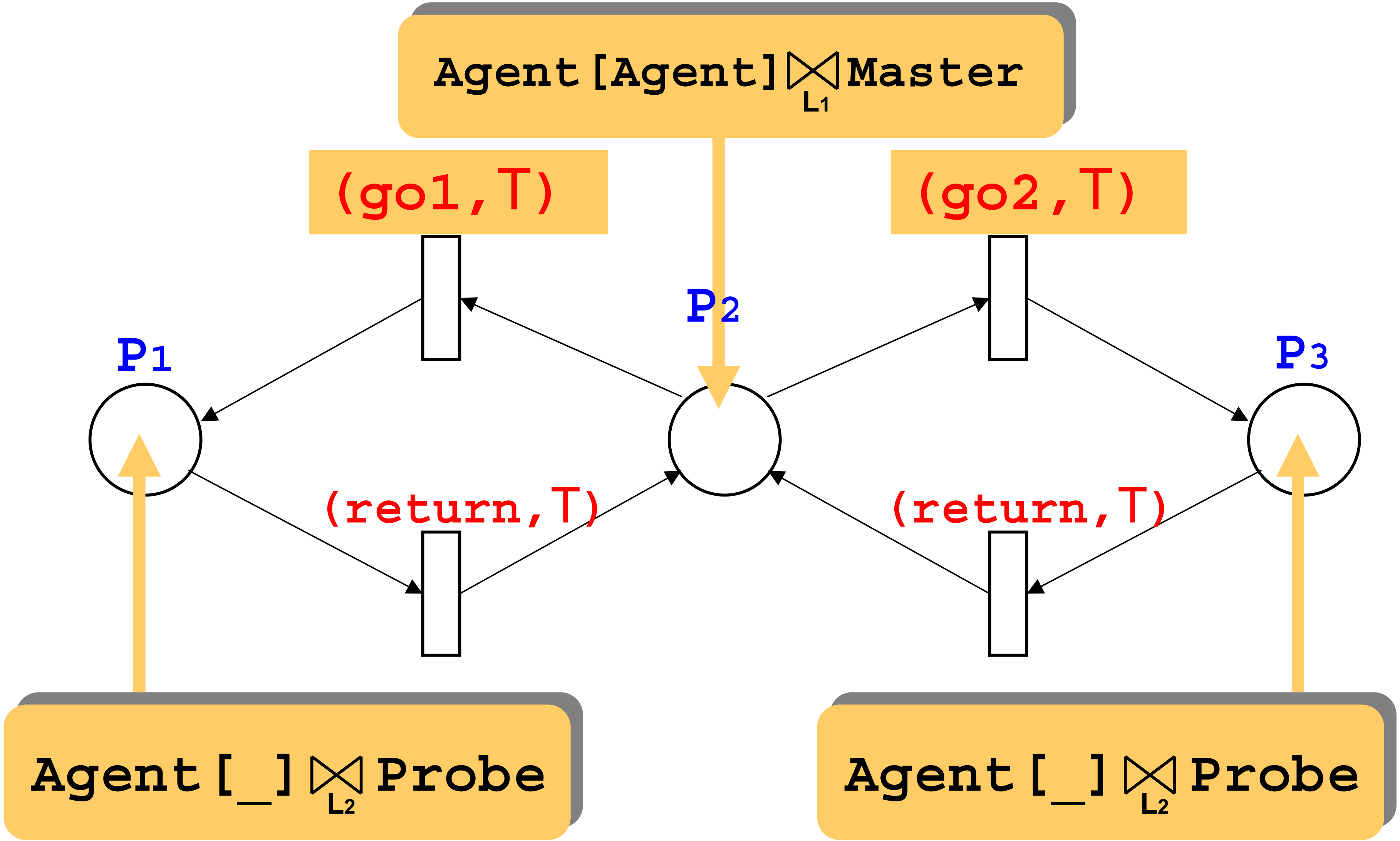
The movement of a token in a new place and its interaction with static components are considered

To allow correct synchronisations the new names introduced in the previous steps are are introduced in the token as well

4. Building the system equation

All PEPA components built in the previous steps are put in parallel and forced to synchronise on common action types

A simple example: preprocessing



A simple example: static components

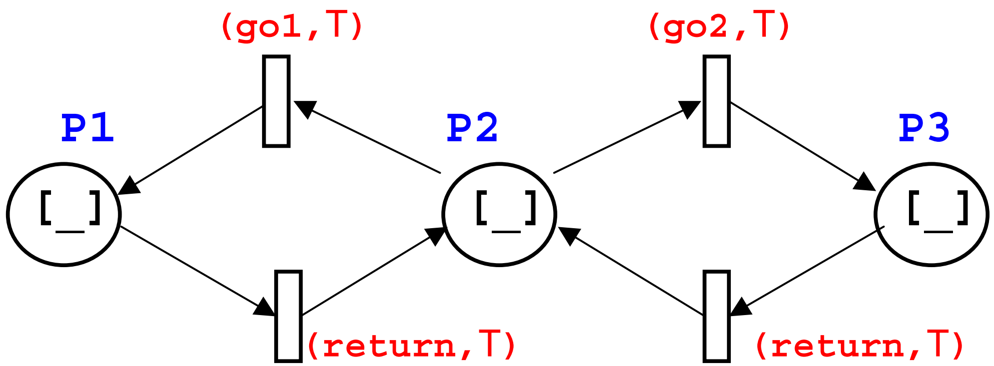
$$Master \stackrel{def}{=} (dump, \top).Master' + (analyse, r_a).Master$$

$$Master' \stackrel{def}{=} (analyse, r_a).Master$$

$$Probe_1 \stackrel{def}{=} (monitor_1, r_m).Probe_1 + (interrogate_1, \top).Probe_1$$

$$Probe_2 \stackrel{def}{=} (monitor_2, r_m).Probe_2 + (interrogate_2, \top).Probe_2$$

A simple example: cells



$$Cell_{10} \stackrel{def}{=} (go_1,T).Cell_{11}$$
$$Cell_{11} \stackrel{def}{=} (return,T).Cell_{10}$$

$$Cell_{20} \stackrel{def}{=} (return,T).Cell_{21}$$
$$Cell_{21} \stackrel{def}{=} (go_1,T).Cell_{20} + (go_2,T).Cell_{20}$$
$$Cell_{30} \stackrel{def}{=} (go_2,T).Cell_{31}$$
$$Cell_{31} \stackrel{def}{=} (return,T).Cell_{30}$$

A simple example: tokens

$$Agent \overset{def}{=} (go_1,\lambda).Agent_1' + (go_2,\lambda).Agent_2'$$

$$Agent_1' \overset{def}{=} (interrogate_1,r_i).Agent''$$

$$Agent_2' \overset{def}{=} (interrogate_2,r_i).Agent''$$

$$Agent'' \overset{def}{=} (return,\mu).Agent'''$$

$$Agent''' \overset{def}{=} (dump,r_d).Agent$$

A simple example: model equation

$$\mathbf{System} \stackrel{def}{=}$$

$$(\mathrm{Cell}_{10} \bowtie_{K_1} (\mathrm{Probe}_1 \bowtie_{K_2} (\mathrm{Agent} \bowtie_{K_3} (\mathrm{Cell}_{21} \bowtie_{K_4}$$

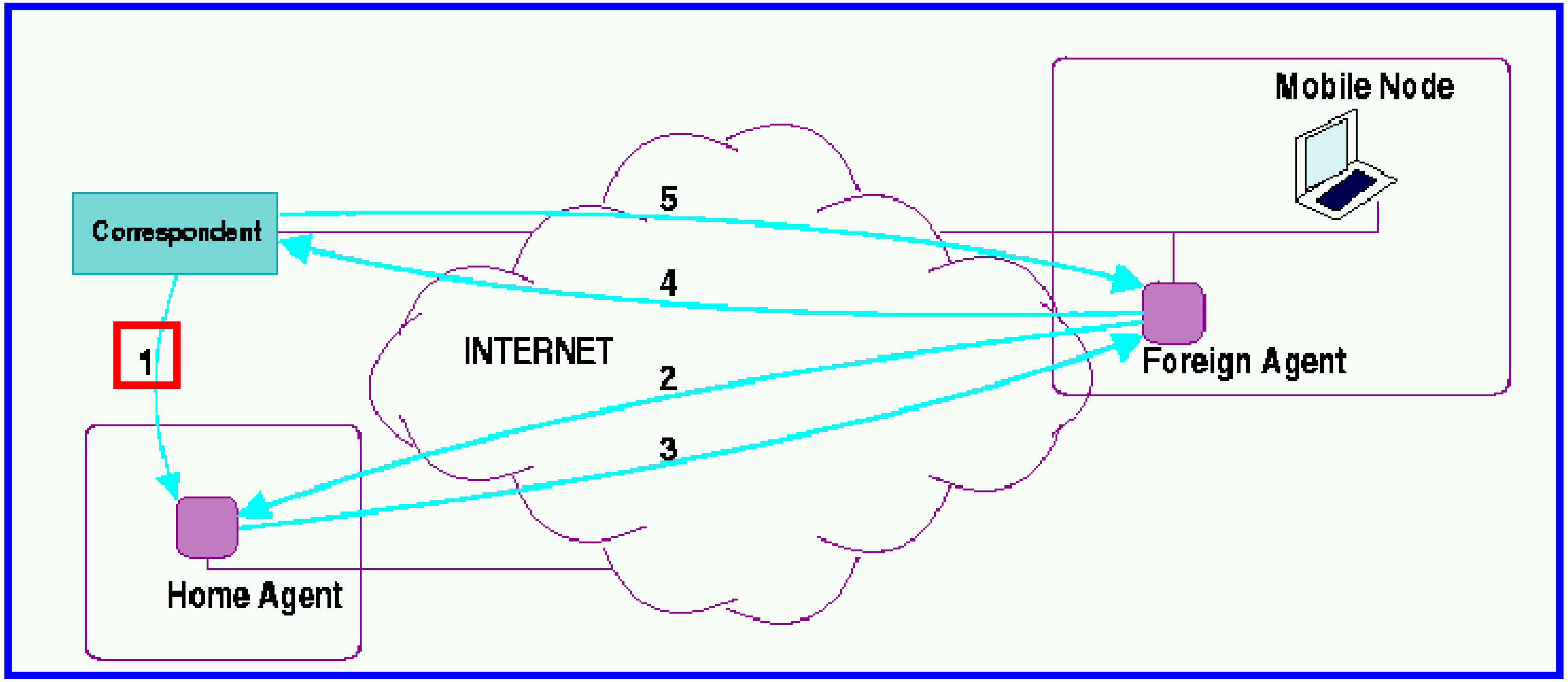
$$(\mathrm{Master} \bowtie_{K_5} (\mathrm{Probe}_2 \bowtie_{K_6} \mathrm{Cell}_{30})))))))$$

$K_1 = \{go_1,\ return\}$ $K_2 = \{interrogate_1\}$

$K_3 = \{go_1,\ go_2,\ return\}$ $K_4 = \{dump\}$

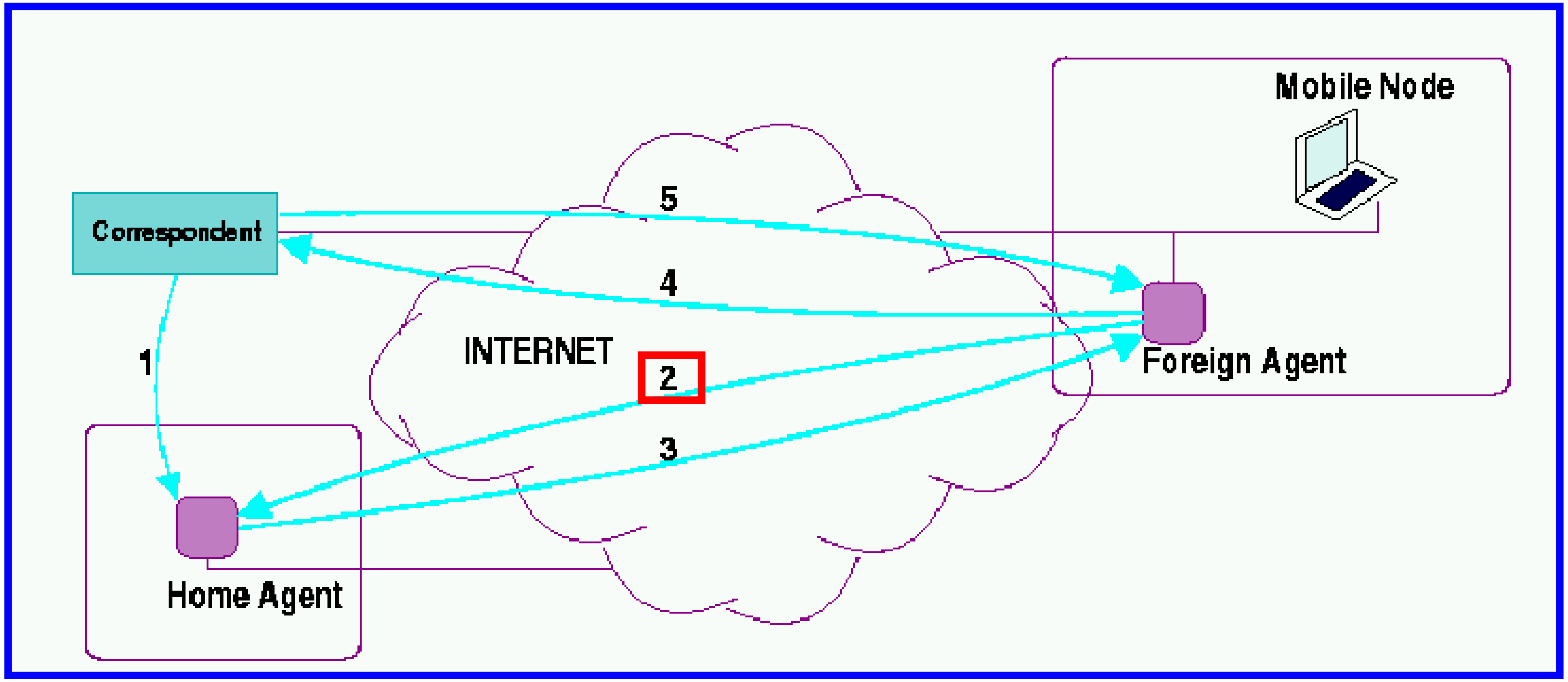
$K_6 = \{go_2,\ return\}$ $K_5 = \{interrogate_2\}$

Complex example: mobile IP



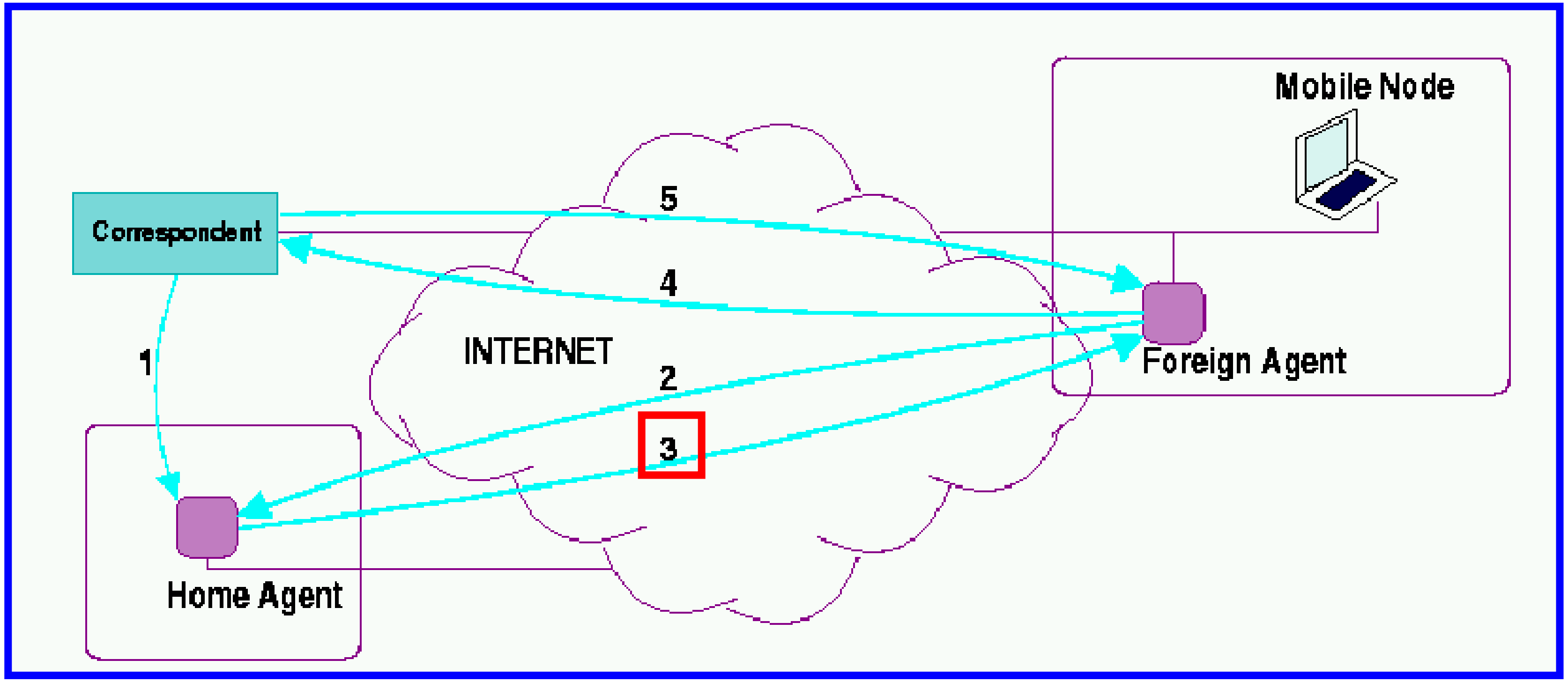
1. Correspondent sends IP packets to the mobile node at his home address

Complex example: mobile IP



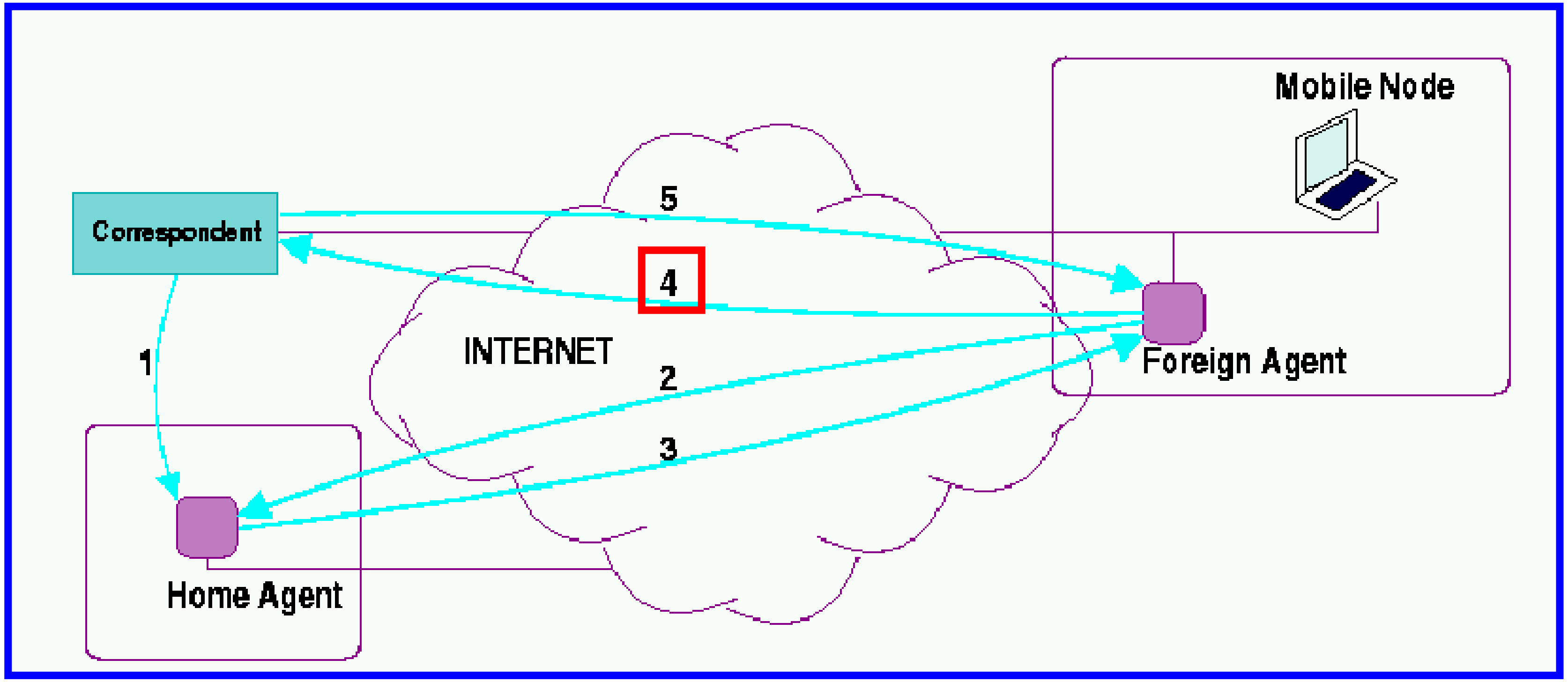
2. The Mobile Node sends its new IP address to the Home Agent

Complex example: mobile IP



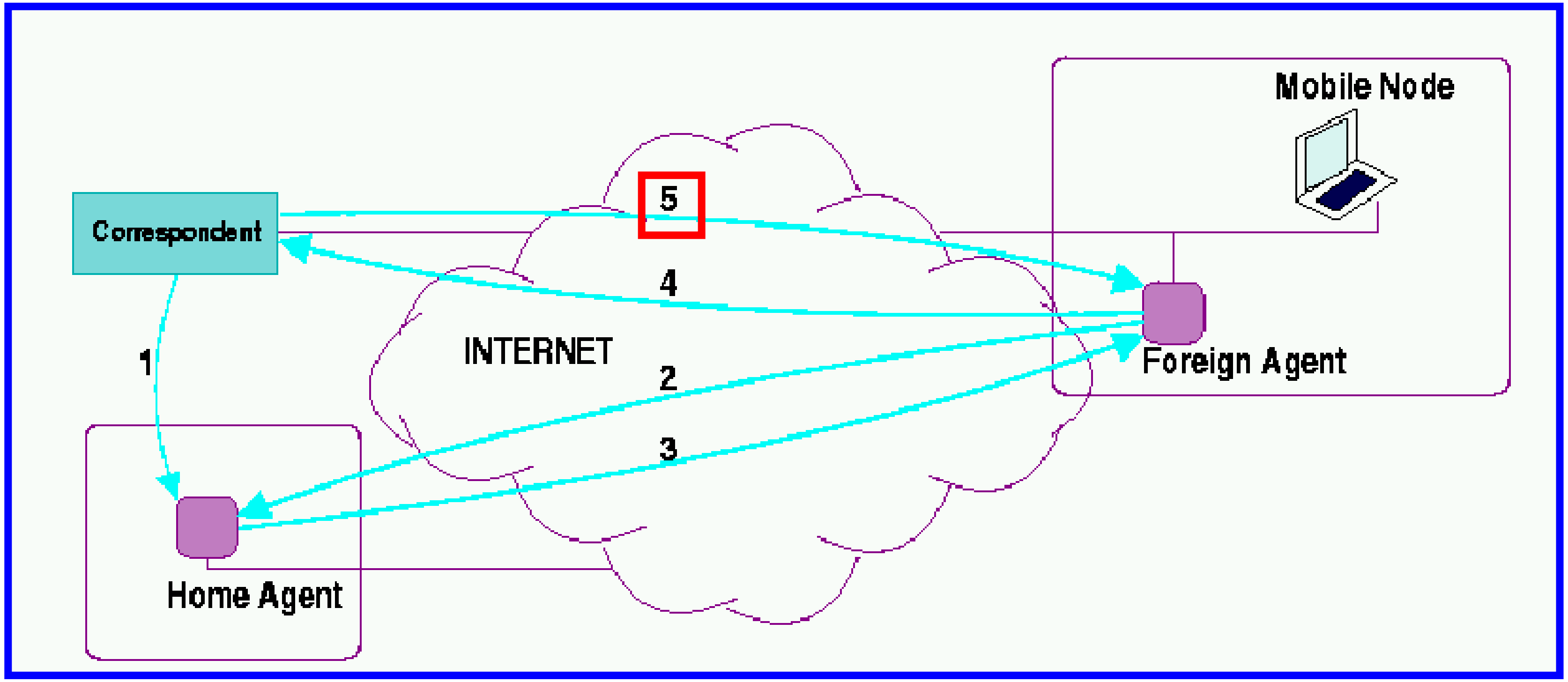
3. The Home Agent forwards packets
   to the Mobile Node

Complex example: mobile IP

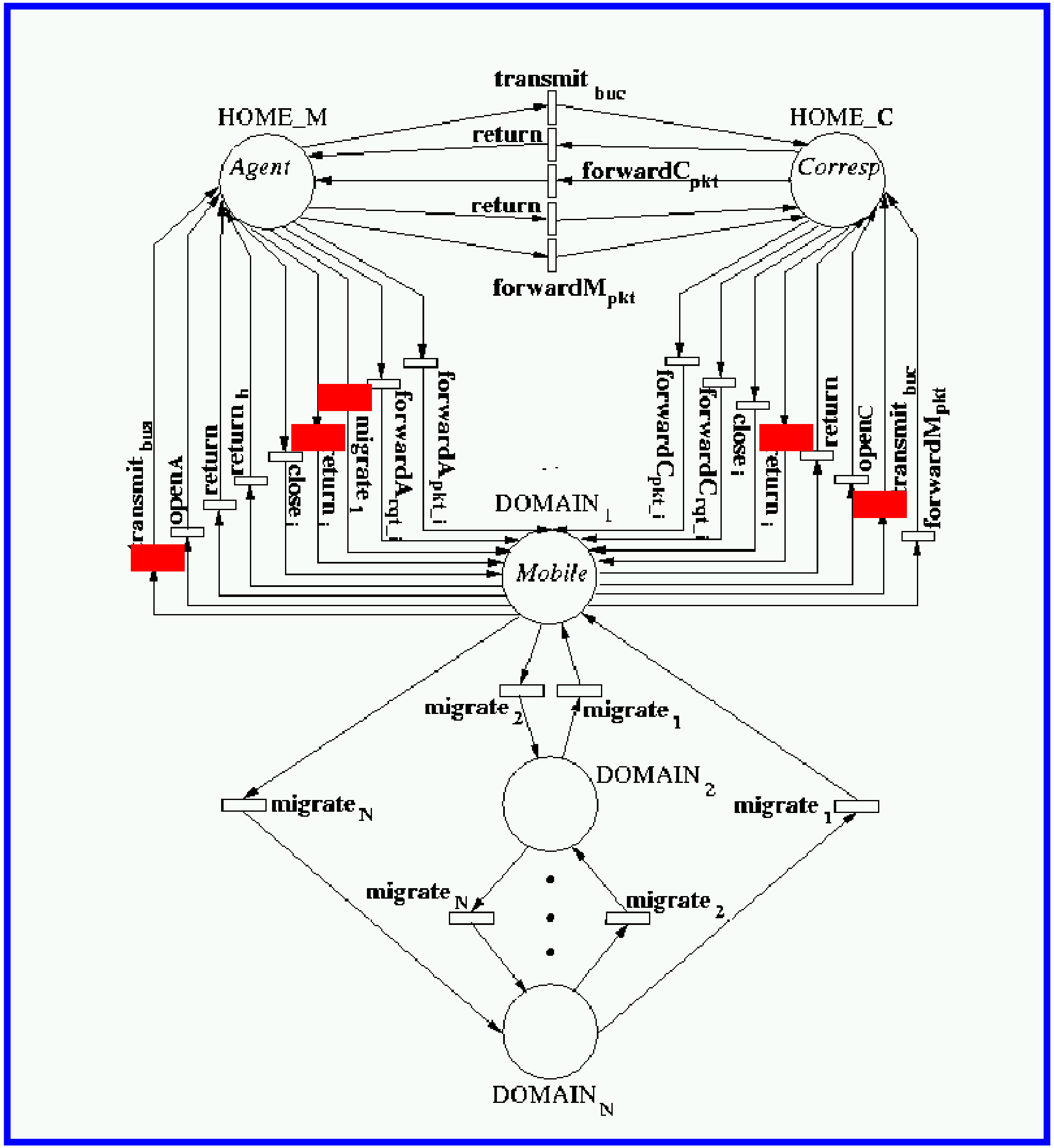


4. The Mobile Node sends its new IP address to the Correspondent

Complex example: mobile IP



5. The Correspondent sends packets directly to the Mobile Node


transmit buc
HOME_M
HOME_C
return
Agent
forwardC pkt
Corresp
return
forwardM pkt
transmit bua
openA
return
return h
close i
eturn i
migrate 1
forwardA rqt_i
forwardA pkt_i
DOMAIN 1
forwardC pkt_i
forwardC rqt_i
close i
eturn i
return
openC
ransmit buc
forwardM pkt
Mobile
migrate 2
migrate 1
DOMAIN 2
migrate N
migrate 1
migrate N
migrate 2
DOMAIN N

Complex example: mobile IP

- **1 Mobile Node, 1 Correspondent, 1 Domain**
  - ✓ 2.8 million of states
  - ✓ 16 million of transitions
  - ✓ in 13.2 seconds, 1.6GHz Pentium IV with 256 MB of RAM

Conclusions

- **PEPA nets is relatively new but we think that it can provide a framework for modelling systems characterised by some mobility**

Future work

- ✓ **Synchronisation over net transitions**
- ✓ **Movement of more than one token**

- ✓ **Graphical interface (done!)**