Modelling network performance with a spatial stochastic process algebra

Vashti Galpin
Laboratory for Foundations of Computer Science
University of Edinburgh

26 May 2009

Introduction

- model network performance

Introduction

- model network performance
- spatial concepts in a stochastic process algebra

Introduction

- model network performance
- spatial concepts in a stochastic process algebra
- location can affect time taken

Introduction

- model network performance
- spatial concepts in a stochastic process algebra
- location can affect time taken
- analysis using continuous time Markov chains (CTMCs)

Introduction

- model network performance
- spatial concepts in a stochastic process algebra
- location can affect time taken
- analysis using continuous time Markov chains (CTMCs)
- no unnecessary increase in state space

Introduction

- model network performance
- spatial concepts in a stochastic process algebra
- location can affect time taken
- analysis using continuous time Markov chains (CTMCs)
- no unnecessary increase in state space
- related research

Introduction

- model network performance
- spatial concepts in a stochastic process algebra
- location can affect time taken
- analysis using continuous time Markov chains (CTMCs)
- no unnecessary increase in state space
- related research
  - PEPA nets (Gilmore *et al*)

Introduction

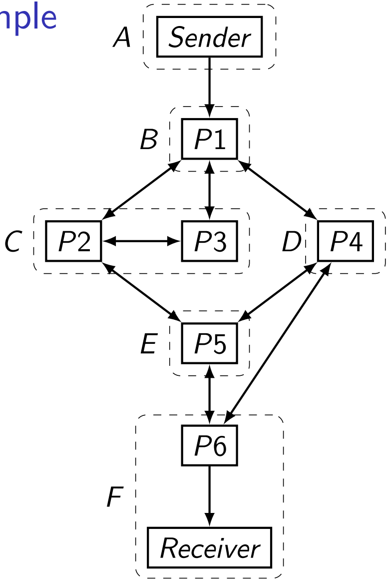
- model network performance
- spatial concepts in a stochastic process algebra
- location can affect time taken
- analysis using continuous time Markov chains (CTMCs)
- no unnecessary increase in state space
- related research
  - PEPA nets (Gilmore *et al*)
  - StoKlaim (de Nicola *et al*)

Introduction

- model network performance
- spatial concepts in a stochastic process algebra
- location can affect time taken
- analysis using continuous time Markov chains (CTMCs)
- no unnecessary increase in state space
- related research
  - PEPA nets (Gilmore *et al*)
  - StoKlaim (de Nicola *et al*)
  - biological models – BioAmbients, attributed π-calculus

Outline

Motivating example



Motivation

- we want to model the performance of a network

Motivation

- we want to model the performance of a network
- we know how to do this with stochastic process algebra

Motivation

- we want to model the performance of a network
- we know how to do this with stochastic process algebra
- PEPA [Hillston 1996]

Motivation

- we want to model the performance of a network
- we know how to do this with stochastic process algebra
- PEPA [Hillston 1996]
  - compact syntax, rules of behaviour

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P + Q \xrightarrow{(\alpha,r)} P'}$$

Motivation

- we want to model the performance of a network
- we know how to do this with stochastic process algebra
- PEPA [Hillston 1996]
  - compact syntax, rules of behaviour

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P + Q \xrightarrow{(\alpha,r)} P'}$$

  - transitions labelled with $(\alpha, r) \in \mathcal{A} \times \mathbb{R}^+$

Motivation

- we want to model the performance of a network
- we know how to do this with stochastic process algebra
- PEPA [Hillston 1996]
  - compact syntax, rules of behaviour

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P + Q \xrightarrow{(\alpha,r)} P'}$$

  - transitions labelled with $(\alpha, r) \in \mathcal{A} \times \mathbb{R}^+$
  - interpret as continuous time Markov chain

Motivation

- we want to model the performance of a network
- we know how to do this with stochastic process algebra
- PEPA [Hillston 1996]
  - compact syntax, rules of behaviour

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P + Q \xrightarrow{(\alpha,r)} P'}$$

  - transitions labelled with $(\alpha, r) \in \mathcal{A} \times \mathbb{R}^+$
  - interpret as continuous time Markov chain
  - analyses to understand performance

Motivation

- we want to model the performance of a network
- we know how to do this with stochastic process algebra
- PEPA [Hillston 1996]
  - compact syntax, rules of behaviour

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P + Q \xrightarrow{(\alpha,r)} P'}$$

  - transitions labelled with $(\alpha, r) \in \mathcal{A} \times \mathbb{R}^+$
  - interpret as continuous time Markov chain
  - analyses to understand performance
- new ingredient: general notion of location

Locations

- $\mathcal{L}$ locations

Locations

- $\mathcal{L}$ locations
  - location names, cities

Locations

- $\mathcal{L}$ locations
  - location names, cities
  - points in n-dimensional space

Locations

- $\mathcal{L}$ locations
  - location names, cities
  - points in n-dimensional space
- collections of locations

Locations

- $\mathcal{L}$ locations
  - location names, cities
  - points in *n*-dimensional space
- collections of locations
  - $\mathcal{P}_{\mathcal{L}} = 2^{\mathcal{L}}$, powerset

Locations

- $\mathcal{L}$ locations
  - location names, cities
  - points in n-dimensional space
- collections of locations
  - $\mathcal{P}_{\mathcal{L}} = 2^{\mathcal{L}}$, powerset
  - $\mathcal{P}_{\mathcal{L}} = P \cup (P \times P)$, singletons and ordered pairs

Locations

- $\mathcal{L}$ locations
  - location names, cities
  - points in n-dimensional space
- collections of locations
  - $\mathcal{P}_{\mathcal{L}} = 2^{\mathcal{L}}$, powerset
  - $\mathcal{P}_{\mathcal{L}} = P \cup (P \times P)$, singletons and ordered pairs
- structure over locations, weighted graph $G = (\mathcal{L}, E, w)$

Locations

- $\mathcal{L}$ locations
  - location names, cities
  - points in n-dimensional space
- collections of locations
  - $\mathcal{P}_\mathcal{L} = 2^\mathcal{L}$, powerset
  - $\mathcal{P}_\mathcal{L} = P \cup (P \times P)$, singletons and ordered pairs
- structure over locations, weighted graph $G = (\mathcal{L}, E, w)$
  - undirected hypergraph or directed graph

Locations

- $\mathcal{L}$ locations
  - location names, cities
  - points in n-dimensional space
- collections of locations
  - $\mathcal{P}_{\mathcal{L}} = 2^{\mathcal{L}}$, powerset
  - $\mathcal{P}_{\mathcal{L}} = P \cup (P \times P)$, singletons and ordered pairs
- structure over locations, weighted graph $G = (\mathcal{L}, E, w)$
  - undirected hypergraph or directed graph
  - $E \subseteq \mathcal{P}_{\mathcal{L}}$

Locations

- $\mathcal{L}$ locations
  - location names, cities
  - points in n-dimensional space
- collections of locations
  - $\mathcal{P}_{\mathcal{L}} = 2^{\mathcal{L}}$, powerset
  - $\mathcal{P}_{\mathcal{L}} = P \cup (P \times P)$, singletons and ordered pairs
- structure over locations, weighted graph $G = (\mathcal{L}, E, w)$
  - undirected hypergraph or directed graph
  - $E \subseteq \mathcal{P}_{\mathcal{L}}$
  - $w : E \rightarrow \mathbb{R}$

Locations

- $\mathcal{L}$ locations
  - location names, cities
  - points in n-dimensional space
- collections of locations
  - $\mathcal{P}_{\mathcal{L}} = 2^{\mathcal{L}}$, powerset
  - $\mathcal{P}_{\mathcal{L}} = P \cup (P \times P)$, singletons and ordered pairs
- structure over locations, weighted graph $G = (\mathcal{L}, E, w)$
  - undirected hypergraph or directed graph
  - $E \subseteq \mathcal{P}_{\mathcal{L}}$
  - $w : E \rightarrow \mathbb{R}$
  - weights modify rates on actions between locations

Syntax

- $\mathcal{L}$ locations, $\mathcal{P}_\mathcal{L}$ collection of locations

Syntax

- $\mathcal{L}$ locations, $\mathcal{P}_{\mathcal{L}}$ collection of locations
- $L \in \mathcal{P}_{\mathcal{L}} \quad \alpha \in \mathcal{A} \quad M \subseteq \mathcal{A} \quad r > 0$

Syntax

- $\mathcal{L}$ locations, $\mathcal{P}_{\mathcal{L}}$ collection of locations
- $L \in \mathcal{P}_{\mathcal{L}} \quad \alpha \in \mathcal{A} \quad M \subseteq \mathcal{A} \quad r > 0$
- sequential components

$$S ::= (\alpha@L, r).S \mid S + S \mid C_s@L$$

Syntax

- $\mathcal{L}$ locations, $\mathcal{P}_{\mathcal{L}}$ collection of locations
- $L \in \mathcal{P}_{\mathcal{L}} \quad \alpha \in \mathcal{A} \quad M \subseteq \mathcal{A} \quad r > 0$
- sequential components

$$S ::= (\alpha@L, r).S \mid S + S \mid C_s@L$$

- sequential constant definition

$$C_s@L \stackrel{def}{=} S$$

Syntax

- $\mathcal{L}$ locations, $\mathcal{P}_{\mathcal{L}}$ collection of locations
- $L \in \mathcal{P}_{\mathcal{L}} \quad \alpha \in \mathcal{A} \quad M \subseteq \mathcal{A} \quad r > 0$
- sequential components
$$S ::= (\alpha@L, r).S \mid S + S \mid C_s@L$$
- sequential constant definition
$$C_s@L \stackrel{def}{=} S$$
- model components
$$P ::= P \underset{M}{\bowtie} P \mid P/M \mid C$$

Syntax

- $\mathcal{L}$ locations, $\mathcal{P}_{\mathcal{L}}$ collection of locations
- $L \in \mathcal{P}_{\mathcal{L}} \quad \alpha \in \mathcal{A} \quad M \subseteq \mathcal{A} \quad r > 0$
- sequential components

$$S ::= (\alpha@L, r).S \mid S + S \mid C_s@L$$

- sequential constant definition

$$C_s@L \stackrel{\mathit{def}}{=} S$$

- model components

$$P ::= P \underset{M}{\bowtie} P \mid P/M \mid C$$

- model constant definition

$$C \stackrel{\mathit{def}}{=} P$$

Parameterised operational semantics

- transitions labelled with $\mathcal{A} \times \mathcal{P}_{\mathcal{L}} \times \mathbb{R}^{+}$

Parameterised operational semantics

- transitions labelled with $\mathcal{A} \times \mathcal{P}_{\mathcal{L}} \times \mathbb{R}^+$
- Prefix

$$\frac{}{(\alpha @ L, r).S \xrightarrow{(\alpha @ L', r)} S} \quad L' = \mathit{apref}((\alpha @ L, r).S)$$

Parameterised operational semantics

- transitions labelled with $\mathcal{A} \times \mathcal{P}_{\mathcal{L}} \times \mathbb{R}^+$
- Prefix

$$\frac{}{(\alpha@L, r).S \xrightarrow{(\alpha@L', r)} S} \quad L' = \mathit{apref}((\alpha@L, r).S)$$

- Cooperation $$\frac{P_1 \xrightarrow{(\alpha@L_1, r_1)} P_1' \quad P_2 \xrightarrow{(\alpha@L_2, r_2)} P_2'}{P_1 \underset{M}{\bowtie} P_2 \xrightarrow{(\alpha@L, R)} P_1' \underset{M}{\bowtie} P_2'} \quad \alpha \in M$$

$$L = \mathit{async}(P_1, P_2, L_1, L_2) \quad R = \mathit{rsync}(P_1, P_2, L_1, L_2, r_1, r_2)$$

Parameterised operational semantics

- transitions labelled with $\mathcal{A} \times \mathcal{P}_{\mathcal{L}} \times \mathbb{R}^+$
- Prefix

$$\frac{}{(\alpha@L, r).S \xrightarrow{(\alpha@L', r)} S} \quad L' = \textit{apref}((\alpha@L, r).S)$$

- Cooperation $\dfrac{P_1 \xrightarrow{(\alpha@L_1, r_1)} P_1' \quad P_2 \xrightarrow{(\alpha@L_2, r_2)} P_2'}{P_1 \underset{M}{\bowtie} P_2 \xrightarrow{(\alpha@L, R)} P_1' \underset{M}{\bowtie} P_2'} \quad \alpha \in M$

$$L = \textit{async}(P_1, P_2, L_1, L_2) \quad R = \textit{rsync}(P_1, P_2, L_1, L_2, r_1, r_2)$$

- parameterised by three functions

Parameterised operational semantics

- transitions labelled with $\mathcal{A} \times \mathcal{P}_{\mathcal{L}} \times \mathbb{R}^+$
- Prefix

$$\frac{}{(\alpha @ L, r).S \xrightarrow{(\alpha @ L', r)} S} \quad L' = \mathit{apref}((\alpha @ L, r).S)$$

- Cooperation $\dfrac{P_1 \xrightarrow{(\alpha @ L_1, r_1)} P_1' \quad P_2 \xrightarrow{(\alpha @ L_2, r_2)} P_2'}{P_1 \underset{M}{\bowtie} P_2 \xrightarrow{(\alpha @ L, R)} P_1' \underset{M}{\bowtie} P_2'} \quad \alpha \in M$

$$L = \mathit{async}(P_1, P_2, L_1, L_2) \quad R = \mathit{rsync}(P_1, P_2, L_1, L_2, r_1, r_2)$$

- parameterised by three functions
- other rules defined in the obvious manner

Parameterised operational semantics (cont.)

- instantiation of functions gives concrete process algebra

Parameterised operational semantics (cont.)

- instantiation of functions gives concrete process algebra
- determines what transitions are possible

Parameterised operational semantics (cont.)

- instantiation of functions gives concrete process algebra
- determines what transitions are possible
- different choices for $\mathcal{P}_{\mathcal{L}}$ give different semantics

Parameterised operational semantics (cont.)

- instantiation of functions gives concrete process algebra
- determines what transitions are possible
- different choices for $\mathcal{P}_{\mathcal{L}}$ give different semantics
  - locations associated with processes and/or actions

Parameterised operational semantics (cont.)

- instantiation of functions gives concrete process algebra
- determines what transitions are possible
- different choices for $\mathcal{P}_{\mathcal{L}}$ give different semantics
  - locations associated with processes and/or actions
  - singleton locations versus multiple locations

Parameterised operational semantics (cont.)

- instantiation of functions gives concrete process algebra
- determines what transitions are possible
- different choices for $\mathcal{P}_{\mathcal{L}}$ give different semantics
  - locations associated with processes and/or actions
  - singleton locations versus multiple locations
- longer terms aims

Parameterised operational semantics (cont.)

- instantiation of functions gives concrete process algebra
- determines what transitions are possible
- different choices for $\mathcal{P}_{\mathcal{L}}$ give different semantics
  - locations associated with processes and/or actions
  - singleton locations versus multiple locations
- longer terms aims
  - prove results for parametric process algebra

Parameterised operational semantics (cont.)

- instantiation of functions gives concrete process algebra
- determines what transitions are possible
- different choices for $\mathcal{P}_{\mathcal{L}}$ give different semantics
  - locations associated with processes and/or actions
  - singleton locations versus multiple locations
- longer terms aims
  - prove results for parametric process algebra
  - then apply to concrete process algebra

Concrete process algebra for modelling networks

- networking performance

Concrete process algebra for modelling networks

- networking performance
- scenario

Concrete process algebra for modelling networks

- networking performance
- scenario
  - arbitrary topology

Concrete process algebra for modelling networks

- networking performance
- scenario
  - arbitrary topology
  - single packet traversal through network

Concrete process algebra for modelling networks

- networking performance
- scenario
  - arbitrary topology
  - single packet traversal through network
  - processes can be colocated

Concrete process algebra for modelling networks

- networking performance
- scenario
  - arbitrary topology
  - single packet traversal through network
  - processes can be colocated
- want to model different topologies and traffic

Concrete process algebra for modelling networks

- networking performance
- scenario
  - arbitrary topology
  - single packet traversal through network
  - processes can be colocated
- want to model different topologies and traffic
- choose functions to create process algebra

Concrete process algebra for modelling networks

- networking performance
- scenario
  - arbitrary topology
  - single packet traversal through network
  - processes can be colocated
- want to model different topologies and traffic
- choose functions to create process algebra
  - each sequential component must have single fixed location

Concrete process algebra for modelling networks

- networking performance
- scenario
  - arbitrary topology
  - single packet traversal through network
  - processes can be colocated
- want to model different topologies and traffic
- choose functions to create process algebra
  - each sequential component must have single fixed location
  - communication must be pairwise and directional

Functions for concrete process algebra

- functions

Functions for concrete process algebra

- functions

$$apref(S) = \begin{cases} \ell & \text{if } ploc(S) = \{\ell\} \\ \perp & \text{otherwise} \end{cases}$$

Functions for concrete process algebra

- functions

$$apref(S) = \begin{cases} \ell & \text{if } ploc(S) = \{\ell\} \\ \perp & \text{otherwise} \end{cases}$$

$$async(P_1, P_2, L_1, L_2) = \begin{cases} (\ell_1, \ell_2) & \text{if } L_1 = \{\ell_1\}, L_2 = \{\ell_2\}, (\ell_1, \ell_2) \in E \\ \perp & \text{otherwise} \end{cases}$$

Functions for concrete process algebra

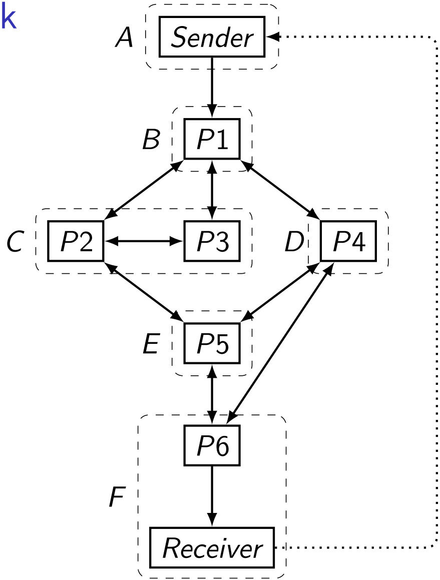
- functions

$$apref(S) = \begin{cases} \ell & \text{if } ploc(S) = \{\ell\} \\ \perp & \text{otherwise} \end{cases}$$

$$async(P_1, P_2, L_1, L_2) = \begin{cases} (\ell_1, \ell_2) & \text{if } L_1 = \{\ell_1\}, L_2 = \{\ell_2\}, (\ell_1, \ell_2) \in E \\ \perp & \text{otherwise} \end{cases}$$

$$rsync(P_1, P_2, L_1, L_2, r_1, r_2)$$
$$= \begin{cases} \dfrac{r_1}{r_\alpha(P_1)} \dfrac{r_2}{r_\alpha(P_2)} \min(r_\alpha(P_1), r_\alpha(P_2)) \cdot w((\ell_1, \ell_2)) & \\ \qquad \text{if } L_1 = \{\ell_1\}, L_2 = \{\ell_2\}, (\ell_1, \ell_2) \in E & \\ \perp \qquad \text{otherwise} & \end{cases}$$

Example network



PEPA model

$$\begin{aligned}
Sender@A &\stackrel{def}{=} (prepare, \rho).Sending@A \\
Sending@A &\stackrel{def}{=} \textstyle\sum_{i=1}^{6} (c_{Si}, r_S).(ack, r_{ack}).Sender@A
\end{aligned}$$

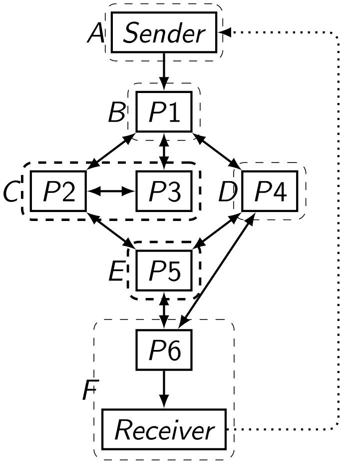
$$\begin{aligned}
Receiver@F &\stackrel{def}{=} \textstyle\sum_{i=1}^{6} (c_{iR}, r_6).Receiving@F \\
Receiving@F &\stackrel{def}{=} (consume, \gamma).(ack, r_{ack}).Receiver@F
\end{aligned}$$

$$\begin{aligned}
P_i@\ell_i &\stackrel{def}{=} (c_{Si}, \top).Q_i@\ell_i + \textstyle\sum_{j=1, j\neq i}^{6} (c_{ji}, r).Q_i@\ell_i \\
Q_i@\ell_i &\stackrel{def}{=} (c_{iR}, \top).P_i@\ell_i + \textstyle\sum_{j=1, j\neq i}^{6} (c_{ij}, r).P_i@\ell_i
\end{aligned}$$

$$\begin{aligned}
Network \stackrel{def}{=} \; & (Sender@A \underset{*}{\bowtie} (P1@B \underset{*}{\bowtie} (P2@C \underset{*}{\bowtie} (P3@C \underset{*}{\bowtie} \\
& (P4@D \underset{*}{\bowtie} (P5@E \underset{*}{\bowtie} (P6@F \underset{*}{\bowtie} Receiver@F)))))))
\end{aligned}$$

Graphs

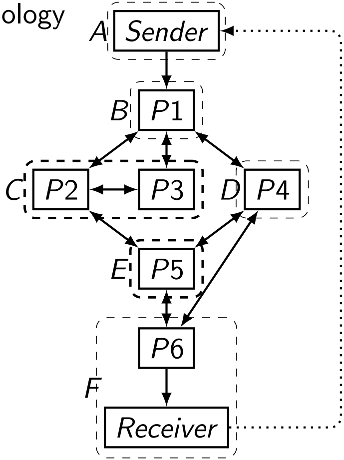
- rates: $r = r_R = r_S = 10$



Graphs

- rates: $r = r_R = r_S = 10$
- the weighted graph G describes the topology

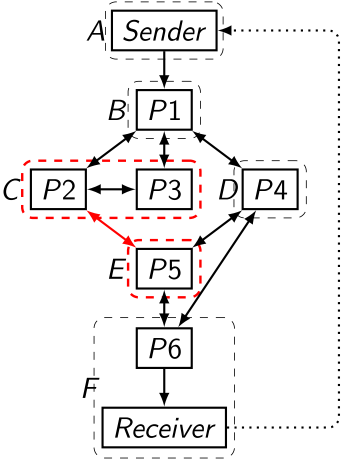
	A	B	C	D	E	F
A	1	1				
B			1	1		
C		1	1		1	
D		1			1	1
E			1	1		1
F	1			1	1	1



Graphs

- G_1 represents heavy traffic between C and E

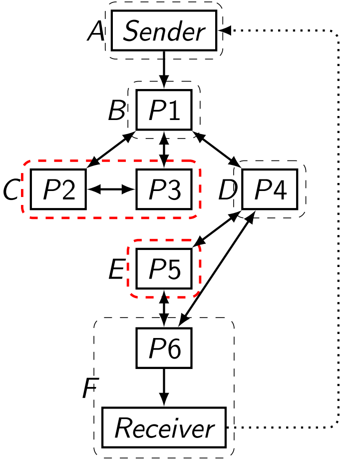
	A	B	C	D	E	F
A	1	1				
B			1	1		
C		1	1		0.1	
D		1			1	1
E			0.1	1		1
F	1			1	1	1



Graphs

- G_2 represents no connectivity between C and E

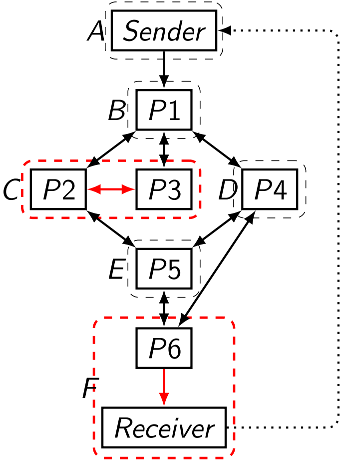
	A	B	C	D	E	F
A	1	1				
B			1	1		
C		1	1		0	
D		1			1	1
E			0	1		1
F	1			1	1	1



Graphs

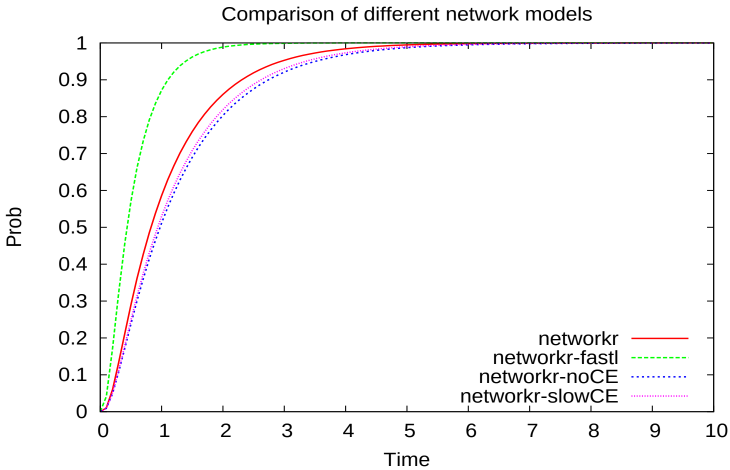
- G_3 represents high connectivity between colocated processes

	A	B	C	D	E	F
A	1	1				
B			1	1		
C		1	10		1	
D		1			1	1
E			1	1		1
F	1			1	1	10



Analysis

- cumulative density function of passage time



Further work and conclusions

- further work

Further work and conclusions

- further work
  - other specific applications

Further work and conclusions

- further work
  - other specific applications
  - theoretical results

Further work and conclusions

- further work
  - other specific applications
  - theoretical results
  - semantic equivalences

Further work and conclusions

- further work
  - other specific applications
  - theoretical results
  - semantic equivalences
  - translation into PEPA

Further work and conclusions

- further work
  - other specific applications
  - theoretical results
  - semantic equivalences
  - translation into PEPA
  - results from graph theory

Further work and conclusions

- further work
  - other specific applications
  - theoretical results
  - semantic equivalences
  - translation into PEPA
  - results from graph theory
- conclusions

Further work and conclusions

- further work
  - other specific applications
  - theoretical results
  - semantic equivalences
  - translation into PEPA
  - results from graph theory
- conclusions
  - presentation of a very general stochastic process algebra with locations

Further work and conclusions

- further work
  - other specific applications
  - theoretical results
  - semantic equivalences
  - translation into PEPA
  - results from graph theory
- conclusions
  - presentation of a very general stochastic process algebra with locations
  - use for modelling network performance in a general way

Thank you

This research was funded by the EPSRC SIGNAL Project