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 $\pi$-calculus
Outline

Introduction
Motivation
Locations
Syntax
Semantics
Measuring performance
Conclusion

Vashti Galpin
Modelling network performance with a spatial stochastic process algebra
AINA 2009
Motivating example

Vashti Galpin
Modelling network performance with a spatial stochastic process algebra
AINA 2009
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

\[
\begin{align*}
P &\xrightarrow{(\alpha,r)} P' \\
\frac{P + Q}{P + Q} &\xrightarrow{(\alpha,r)} P'
\end{align*}
\]
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

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

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

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
  \[ P \xrightarrow{\alpha, r} P' \]
  \[ P + Q \xrightarrow{\alpha, r} P' \]
  - transitions labelled with \((\alpha, r) \in A \times \mathbb{R}^+\)
  - 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
    
    \[
    P \xrightarrow{\,(\alpha,r)\,} P' \\
    P + Q \xrightarrow{\,(\alpha,r)\,} P'
    \]
  - transitions labelled with \((\alpha, r) \in A \times \mathbb{R}^+\)
  - 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
    \[
    P \xrightarrow{(\alpha, r)} P' \\
    P + Q \xrightarrow{(\alpha, r)} P'
    \]
  - 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
  - \( \mathcal{P}_\mathcal{L} = \mathcal{P} \cup (\mathcal{P} \times \mathcal{P}) \), 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
  - \( \mathcal{P}_\mathcal{L} = \mathcal{P} \cup (\mathcal{P} \times \mathcal{P}) \), 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
  - $E \subseteq \mathcal{P}_\mathcal{L}$
  - $w : E \to \mathbb{R}$
  - 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
- $L \in \mathcal{P}_{\mathcal{L}}$, $\alpha \in \mathcal{A}$, $M \subseteq \mathcal{A}$, $r > 0$
Syntax

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

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

- $\mathcal{L}$ locations, $\mathcal{P}_L$ collection of locations
- $L \in \mathcal{P}_L$, $\alpha \in \mathcal{A}$, $M \subseteq \mathcal{A}$, $r > 0$
- sequential components
  
  $S ::= (\alpha @ L, r).S \mid S + S \mid C_s @ L$

- sequential constant definition
  
  $C_s @ L \overset{def}{=} S$
Syntax

- $\mathcal{L}$ locations, $\mathcal{P}_L$ collection of locations
- $L \in \mathcal{P}_L \quad \alpha \in \mathcal{A} \quad M \subseteq \mathcal{A} \quad r > 0$
- sequential components
  $S ::= (\alpha@L, r).S \mid S + S \mid C_s@L$
- sequential constant definition
  $C_s@L \overset{\text{def}}{=} S$
- model components
  $P ::= P \boxtimes_M P \mid P/M \mid C$
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
  \[
  C_s@L \triangleq S
  \]
- model components
  \[
  P ::= P \parallel_M P \mid P/M \mid C
  \]
- model constant definition
  \[
  C \triangleq P
  \]
Parameterised operational semantics

- transitions labelled with $\mathcal{A} \times \mathcal{P}_L \times \mathbb{R}^+$
Parameterised operational semantics

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

$$
(\alpha \otimes L, r).S \xrightarrow{(\alpha \otimes L', r)} S \quad L' = \text{apref}( (\alpha \otimes L, r).S )
$$
Parameterised operational semantics

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

$$
(\alpha \otimes L, r).S \xrightarrow{(\alpha \otimes L', r)} S
\quad \Rightarrow \quad L' = \text{apref}((\alpha \otimes L, r).S)
$$

- Cooperation

$$
P_1 \xrightarrow{(\alpha \otimes L_1, r_1)} P'_1 \quad P_1 \otimes M P_2 \xrightarrow{(\alpha \otimes L, R)} P'_1 \otimes M P'_2
\quad \Rightarrow \quad \alpha \in M
$$

$$
L = \text{async}(P_1, P_2, L_1, L_2) \quad R = \text{rsync}(P_1, P_2, L_1, L_2, r_1, r_2)
$$
Parameterised operational semantics

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

- Prefix

  $$(\alpha \otimes L, r).S \xrightarrow{(\alpha \otimes L', r)} L' = \text{apref}\left((\alpha \otimes L, r).S\right)$$

- Cooperation

  $$P_1 \xrightarrow{(\alpha \otimes L_1, r_1)} P'_1 \quad P_2 \xrightarrow{(\alpha \otimes L_2, r_2)} P'_2 \quad \alpha \in M$$

  $$P_1 \Join^M P_2 \xrightarrow{(\alpha \otimes L, R)} P'_1 \Join^M P'_2$$

- $L = \text{async}(P_1, P_2, L_1, L_2)$  \hspace{1cm} $R = \text{rsync}(P_1, P_2, L_1, L_2, r_1, r_2)$

- parameterised by three functions
Parameterised operational semantics

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

- Prefix

$$
(\alpha \circ \mathcal{L}, r).S \xrightarrow{(\alpha \circ \mathcal{L}', r)} S
$$

- Cooperation

$$
P_1 \xrightarrow{(\alpha \circ \mathcal{L}_1, r_1)} P_1' \quad P_2 \xrightarrow{(\alpha \circ \mathcal{L}_2, r_2)} P_2' \quad \alpha \in \mathcal{M}
$$

$$
P_1 \Join^M P_2 \xrightarrow{(\alpha \circ \mathcal{L}, R)} P_1' \Join^M P_2'
$$

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

- parameterised by three functions

- other rules defined in the obvious manner

Vashti Galpin

Modelling network performance with a spatial stochastic process algebra

AINA 2009
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 $\mathcal{P}_L$ 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 $\mathcal{P}_L$ 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 $P_L$ 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 $\mathcal{P}_L$ 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

Vashti Galpin
Modelling network performance with a spatial stochastic 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
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 collocated
- 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

\[ \text{apref}(S) = \begin{cases} \ell & \text{if } ploc(S) = \{\ell\} \\ \perp & \text{otherwise} \end{cases} \]
Functions for concrete process algebra

Functions

\[ \text{apref}(S) = \begin{cases} \ell & \text{if } ploc(S) = \{\ell\} \\ \bot & \text{otherwise} \end{cases} \]

\[ \text{async}(P_1, P_2, L_1, L_2) = \begin{cases} (\ell_1, \ell_2) & \text{if } L_1 = \{\ell_1\}, L_2 = \{\ell_2\}, (\ell_1, \ell_2) \in E \\ \bot & \text{otherwise} \end{cases} \]
Functions for concrete process algebra

▶ functions

\[ \text{apref}(S) = \begin{cases} \ell & \text{if } \text{ploc}(S) = \{\ell\} \\ \bot & \text{otherwise} \end{cases} \]

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

\[ \text{rsync}(P_1, P_2, L_1, L_2, r_1, r_2) = \begin{cases} \frac{r_1}{r_\alpha(P_1)} \frac{r_2}{r_\alpha(P_2)} \min(r_\alpha(P_1), r_\alpha(P_2)) \cdot w((\ell_1, \ell_2)) & \text{if } L_1 = \{\ell_1\}, L_2 = \{\ell_2\}, (\ell_1, \ell_2) \in E \\ \bot & \text{otherwise} \end{cases} \]
Example network

Sender

P1

P2

P3

P4

P5

P6

Receiver

A

B

C

D

E

F

Vashti Galpin

Modelling network performance with a spatial stochastic process algebra

AINA 2009
PEPA model

\[
\begin{align*}
\text{Sender}@A & \stackrel{\text{def}}{=} (\text{prepare}, \rho).\text{Sending}@A \\
\text{Sending}@A & \stackrel{\text{def}}{=} \sum_{i=1}^{6}(c_{Si}, r_{S}).(\text{ack}, r_{ack}).\text{Sender}@A \\
\text{Receiver}@F & \stackrel{\text{def}}{=} \sum_{i=1}^{6}(c_{iR}, r_{6}).\text{Receiving}@F \\
\text{Receiving}@F & \stackrel{\text{def}}{=} (\text{consume}, \gamma).(\text{ack}, r_{ack}).\text{Receiver}@F \\
\text{P}_i@\ell_i & \stackrel{\text{def}}{=} (c_{Si}, \top).\text{Q}_i@\ell_i + \sum_{j=1, j \neq i}^{6}(c_{ji}, r).\text{Q}_i@\ell_i \\
\text{Q}_i@\ell_i & \stackrel{\text{def}}{=} (c_{iR}, \top).\text{P}_i@\ell_i + \sum_{j=1, j \neq i}^{6}(c_{ij}, r).\text{P}_i@\ell_i \\
\text{Network} & \stackrel{\text{def}}{=} (\text{Sender}@A \boxtimes (\text{P}_1@\ell_1 \boxtimes (\text{P}_2@\ell_2 \boxtimes (\text{P}_3@\ell_3 \boxtimes (\text{P}_4@\ell_4 \boxtimes (\text{P}_5@\ell_5 \boxtimes (\text{P}_6@\ell_6 \boxtimes \text{Receiver}@\ell_7))))))))
\end{align*}
\]
Graphs

rates: \( r = r_R = r_S = 10 \)
Graphs

- rates: \( r = r_R = r_S = 10 \)
- the weighted graph \( G \) describes the topology

\[
\begin{array}{ccccccc}
A & B & C & D & E & F \\
A & 1 & 1 &  &  &  &  \\
B &  & 1 & 1 &  &  &  \\
C &  &  & 1 & 1 &  &  \\
D &  &  &  & 1 & 1 &  \\
E &  &  &  &  & 1 & 1 \\
F &  &  &  &  &  & 1 \\
\end{array}
\]

▶ the weighted graph \( G \) describes the topology.
Graphs

- $G_1$ represents heavy traffic between $C$ and $E$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td>1 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>0.1</td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Vashti Galpin
Modelling network performance with a spatial stochastic process algebra
AINA 2009
### Graphs

- $G_2$ represents no connectivity between $C$ and $E$

<table>
<thead>
<tr>
<th></th>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$D$</th>
<th>$E$</th>
<th>$F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$B$</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C$</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$D$</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$E$</td>
<td></td>
<td>0</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$F$</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Sender

Receiver

Vashti Galpin

Modelling network performance with a spatial stochastic process algebra

AINA 2009
Graphs

- $G_3$ represents high connectivity between colocated processes

\[
\begin{array}{ccccccc}
A & B & C & D & E & F \\
A & 1 & 1 & & & \\
B & & 1 & 1 & & \\
C & 1 & & 10 & 1 & \\
D & 1 & & 1 & 1 & \\
E & 1 & 1 & & 1 & \\
F & 1 & & 1 & 1 & 10 \\
\end{array}
\]
Analysis

- cumulative density function of passage time

Comparison of different network models

Vashti Galpin
Modelling network performance with a spatial stochastic process algebra
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