Project:

Internet Worm Attacks and Stochastic Agent Models

Jeremy Bradley

Email: jb@doc.ic.ac.uk

Department of Computing, Imperial College London

Produced with prosper and LaTeX

Contention...

Passage or response times are useful in an agent modelling setting

Presentation

- Internet worms at work!
- A formal agent description
- Our existing techniques
- Other solutions methods

Lifecycle of an Internet Worm

Sequence of events:

1. Malicious person infects an initial server

Lifecycle of an Internet Worm

Sequence of events:

1. Malicious person infects an initial server
2. Each infected computer selects next victim at random

Lifecycle of an Internet Worm

Sequence of events:

1. Malicious person infects an initial server
2. Each infected computer selects next victim at random
3. Worm copies itself into the next running system using one or more security loopholes in victim's defence mechanism

Lifecycle of an Internet Worm

Sequence of events:

1. Malicious person infects an initial server
2. Each infected computer selects next victim at random
3. Worm copies itself into the next running system using one or more security loopholes in victim’s defence mechanism
4. Worm optionally discharges some malicious payload

Lifecycle of an Internet Worm

Sequence of events:

1. Malicious person infects an initial server
2. Each infected computer selects next victim at random
3. Worm copies itself into the next running system using one or more security loopholes in victim's defence mechanism
4. Worm optionally discharges some malicious payload
5. Repeat from (2) for each new infection

Lifecycle of an Internet server

On hearing of a worm attack:

- random time delay (exponentially distributed) before server's software is patched: *inoculation*

In which time...

Lifecycle of an Internet server

On hearing of a worm attack:

- random time delay (exponentially distributed) before server's software is patched: *inoculation*

In which time...

1. server may become infected

Lifecycle of an Internet server

On hearing of a worm attack:

- random time delay (exponentially distributed) before server's software is patched: *inoculation*

In which time...

1. server may become infected
2. server may be disabled by the worm infection

Lifecycle of an Internet server

On hearing of a worm attack:

- random time delay (exponentially distributed) before server's software is patched: *inoculation*

In which time...

1. server may become infected
2. server may be disabled by the worm infection
3. server may be repaired and returned to the network unpatched

Lifecycle of an Internet server

On hearing of a worm attack:

- random time delay (exponentially distributed) before server's software is patched: *inoculation*

In which time...

1. server may become infected
2. server may be disabled by the worm infection
3. server may be repaired and returned to the network unpatched
4. server may be repaired and patched

Lifecycle of an Internet server

On hearing of a worm attack:

- random time delay (exponentially distributed) before server's software is patched: *inoculation*

In which time...

1. server may become infected
2. server may be disabled by the worm infection
3. server may be repaired and returned to the network unpatched
4. server may be repaired and patched
5. server may be rolled back, thus removing patch

Internet worms

- Code Red, Nimbda, Code Red II (2001), SQL Slammer (January 2003), Nachi and MSBlast (August 2003), Sasser (1 May 2004)
- Usually malicious autonomous program that spreads without user intervention

Internet worms

- Code Red, Nimbda, Code Red II (2001), SQL Slammer (January 2003), Nachi and MSBlast (August 2003), Sasser (1 May 2004)
- Usually malicious autonomous program that spreads without user intervention
- **Emergent behaviour:** causes huge network bottlenecks – brings internet to standstill for many hours, or even days

Code Red II

- Example: Code Red II
  - 19th July 2001
  - 350,000 hosts infected in 14 hours
  - c.f. Sasser: 1–1.5 million hosts in 2 days
  - utilised *buffer overflow* in Microsoft IIS web server
  - infected machines would probe for other victims on port 80
  - 20th July 2001: mode changes from one of propagation to DDOS attack on the www.whitehouse.gov

Epidemiological model

- Good reasons to model Internet worms as biological agents:
  - inherently large scale dynamics

Epidemiological model

- Good reasons to model Internet worms as biological agents:
  - inherently large scale dynamics
  - computers can contact each other virtually randomly

Epidemiological model

- Good reasons to model Internet worms as biological agents:
  - inherently large scale dynamics
  - computers can contact each other virtually randomly

$\Rightarrow$ infected computers/computers susceptible to infection will mix homogeneously

Epidemiological model

- Good reasons to model Internet worms as biological agents:
  - inherently large scale dynamics
  - computers can contact each other virtually randomly
  - $\Rightarrow$ infected computers/computers susceptible to infection will mix homogeneously
  - potential to kill/disable a host, according to payload

Epidemiological model

- Good reasons to model Internet worms as biological agents:
  - inherently large scale dynamics
  - computers can contact each other virtually randomly
  - $\Rightarrow$ infected computers/computers susceptible to infection will mix homogeneously
  - potential to kill/disable a host, according to payload
  - Nicol et al use hybrid model of Internet worms: epidemiological/stochastic

Not quite Biology!

- Good news: We have an exact behavioural description of an individual worm in glorious detail

Not quite Biology!

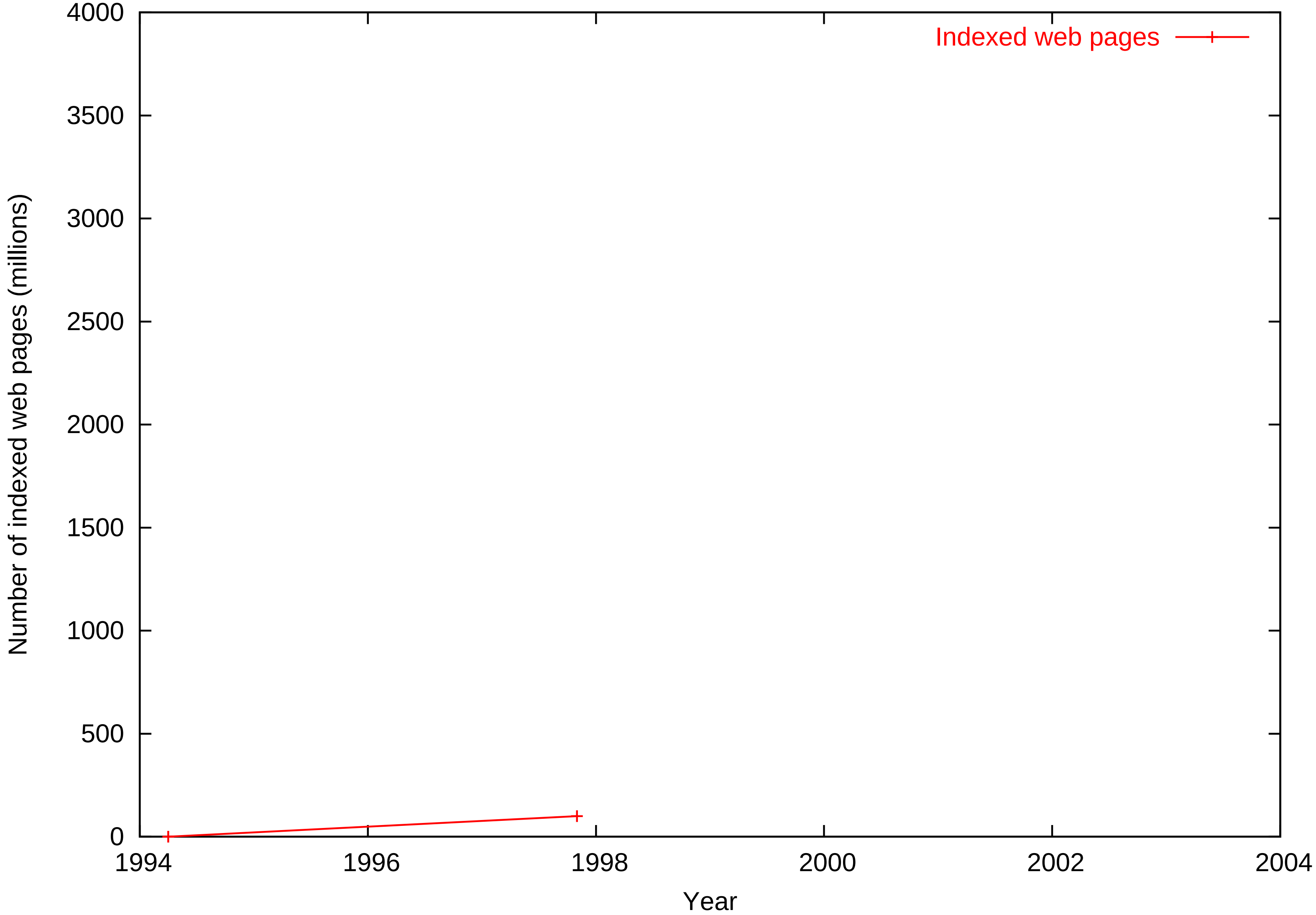
- Good news: We have an exact behavioural description of an individual worm in glorious detail
- Bad news: We have an exact behavioural description of an individual worm in glorious detail

Not quite Biology!

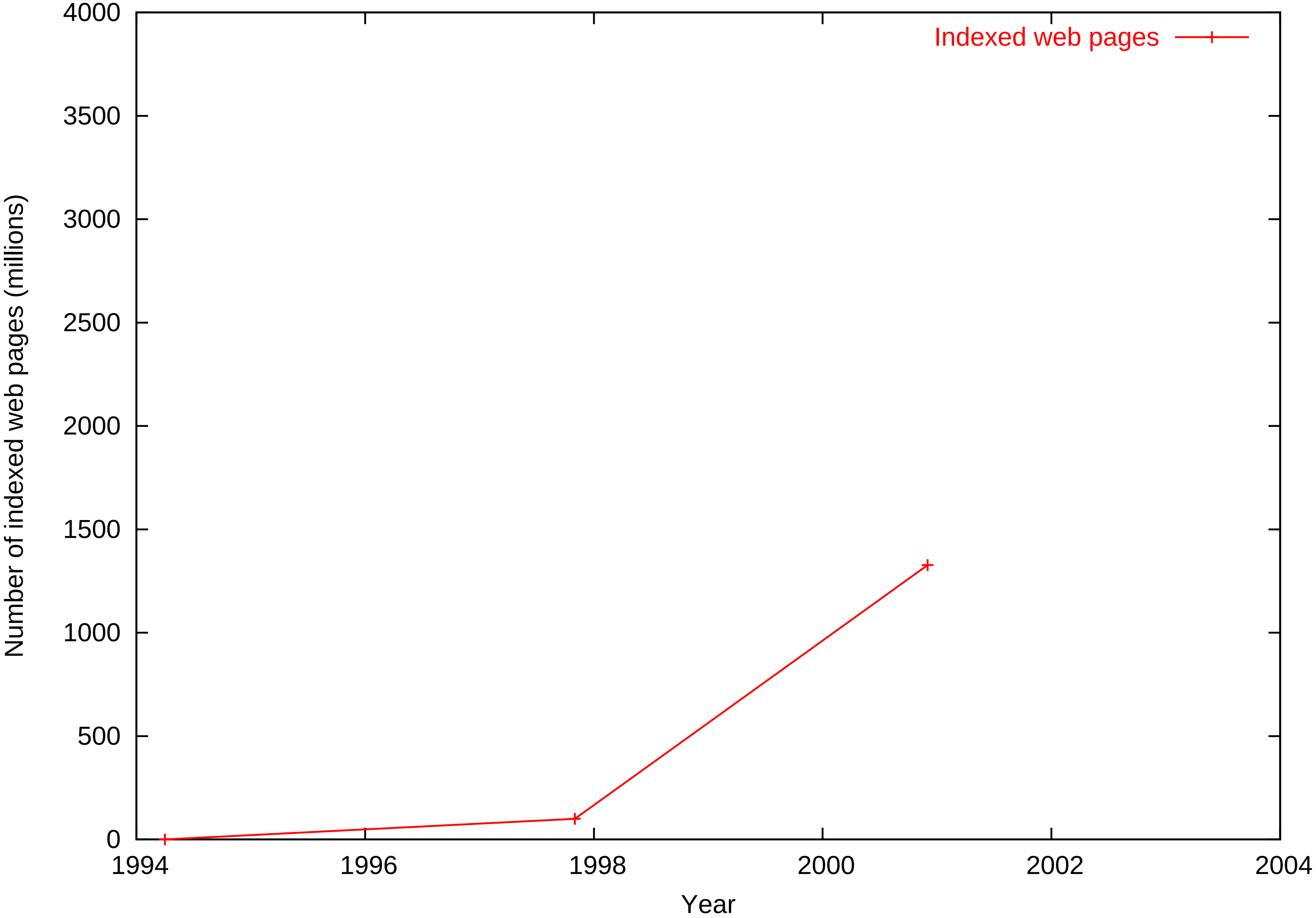
- Good news: We have an exact behavioural description of an individual worm in glorious detail
- Bad news: We have an exact behavioural description of an individual worm in glorious detail

$\Rightarrow$ We have to learn to prune *unimportant* behaviour

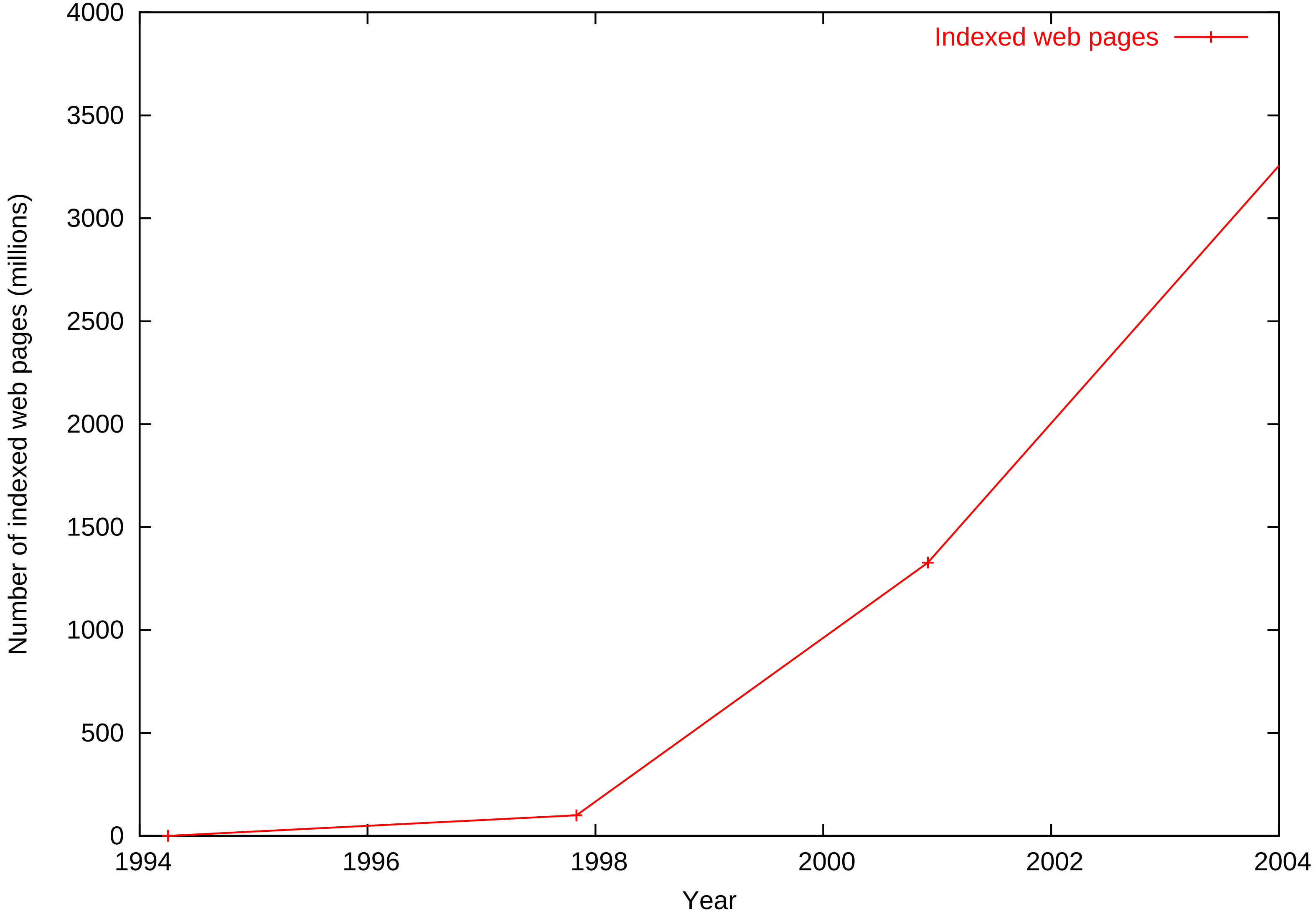
Internet Growth



Internet Growth



Internet Growth



Stochastic Process Algebra

PEPA syntax:

$$P \quad ::= \quad (a, \lambda).P \;\Big|\; P + P \;\Big|\; P \underset{L}{\bowtie} P \;\Big|\; P/L \;\Big|\; A$$

Stochastic Process Algebra

PEPA syntax:

$$P \quad ::= \quad (a, \lambda).P \;\Big|\; P + P \;\Big|\; P \underset{L}{\bowtie} P \;\Big|\; P/L \;\Big|\; A$$

- Action prefix: $(a, \lambda).P$

Stochastic Process Algebra

PEPA syntax:

$$P \quad ::= \quad (a, \lambda).P \;\Big|\; P + P \;\Big|\; P \underset{L}{\bowtie} P \;\Big|\; P/L \;\Big|\; A$$

- Action prefix: $(a, \lambda).P$
- Competitive choice: $P_1 + P_2$

Stochastic Process Algebra

PEPA syntax:

$$P \quad ::= \quad (a, \lambda).P \;\Big|\; P + P \;\Big|\; P \underset{L}{\bowtie} P \;\Big|\; P/L \;\Big|\; A$$

- Action prefix: $(a, \lambda).P$
- Competitive choice: $P_1 + P_2$
- Cooperation: $P_1 \underset{L}{\bowtie} P_2$

Stochastic Process Algebra

PEPA syntax:

$$P \quad ::= \quad (a, \lambda).P \;\Big|\; P + P \;\Big|\; P \underset{L}{\bowtie} P \;\Big|\; P/L \;\Big|\; A$$

- Action prefix: $(a, \lambda).P$
- Competitive choice: $P_1 + P_2$
- Cooperation: $P_1 \underset{L}{\bowtie} P_2$
- Action hiding: P/L

Stochastic Process Algebra

PEPA syntax:

$$P \quad ::= \quad (a, \lambda).P \;\Big|\; P + P \;\Big|\; P \underset{L}{\bowtie} P \;\Big|\; P/L \;\Big|\; A$$

- Action prefix: $(a, \lambda).P$
- Competitive choice: $P_1 + P_2$
- Cooperation: $P_1 \underset{L}{\bowtie} P_2$
- Action hiding: P/L
- Constant label: A

Biological PEPA Agent Modelling

Require a pairwise cooperation paradigm:

$$\mathrm{P} \;::=\; (\mathrm{a}, \lambda).\mathrm{P} \;\Big|\; \mathrm{P} + \mathrm{P} \;\Big|\; \mathrm{P} \underset{L}{\bigoplus} \mathrm{P} \;\Big|\; \mathrm{P}/\mathrm{L} \;\Big|\; \mathrm{A}$$

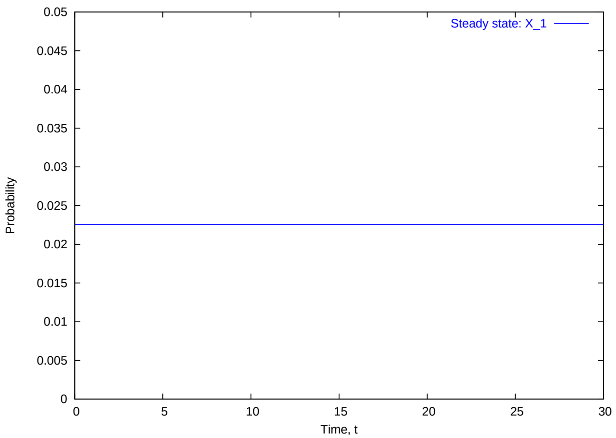
- Action prefix: $(\mathrm{a}, \lambda).\mathrm{P}$
- Competitive choice: $\mathrm{P}_1 + \mathrm{P}_2$
- Pairwise agent cooperation: $\mathrm{P}_1 \underset{L}{\bigoplus} \mathrm{P}_2$
- Action hiding: P/L
- Constant label: A

Types of Analysis

Steady-state and transient analysis in PEPA:

$$
\begin{aligned}
A1 &\stackrel{def}{=} (start, r_1).A2 + (pause, r_2).A3 \\
A2 &\stackrel{def}{=} (run, r_3).A1 + (fail, r_4).A3 \\
A3 &\stackrel{def}{=} (recover, r_1).A1 \\
AA &\stackrel{def}{=} (run, \top).(alert, r_5).AA \\
Sys &\stackrel{def}{=} AA \underset{\{run\}}{\bowtie} A1
\end{aligned}
$$

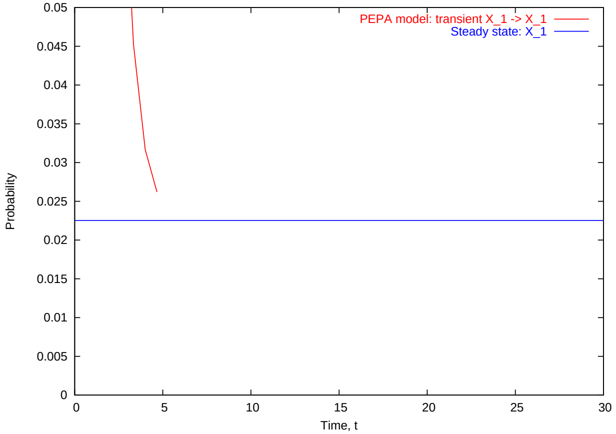
$\Rightarrow$



Types of Analysis

Steady-state and transient analysis in PEPA:

$$
\begin{aligned}
A1 &\stackrel{\mathrm{def}}{=} (start, r_1).A2 + (pause, r_2).A3 \\
A2 &\stackrel{\mathrm{def}}{=} (run, r_3).A1 + (fail, r_4).A3 \\
A3 &\stackrel{\mathrm{def}}{=} (recover, r_1).A1 \\
AA &\stackrel{\mathrm{def}}{=} (run, \top).(alert, r_5).AA \\
Sys &\stackrel{\mathrm{def}}{=} AA \underset{\{run\}}{\bowtie} A1
\end{aligned}
\quad \Rightarrow
$$

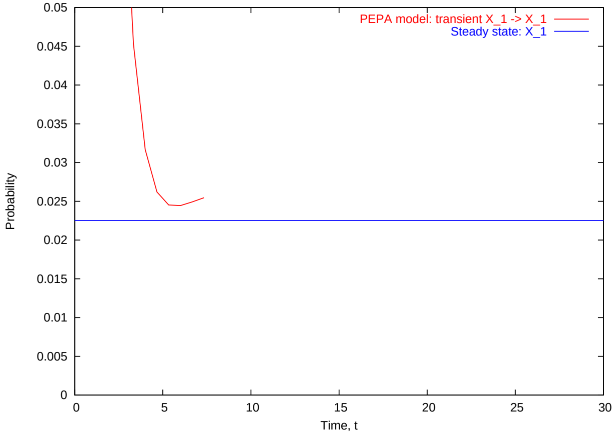


Types of Analysis

Steady-state and transient analysis in PEPA:

$$
\begin{aligned}
A1 &\stackrel{\text{def}}{=} (\mathit{start}, r_1).A2 + (\mathit{pause}, r_2).A3 \\
A2 &\stackrel{\text{def}}{=} (\mathit{run}, r_3).A1 + (\mathit{fail}, r_4).A3 \\
A3 &\stackrel{\text{def}}{=} (\mathit{recover}, r_1).A1 \\
AA &\stackrel{\text{def}}{=} (\mathit{run}, \top).(\mathit{alert}, r_5).AA \\
Sys &\stackrel{\text{def}}{=} AA \underset{\{run\}}{\bowtie} A1
\end{aligned}
$$

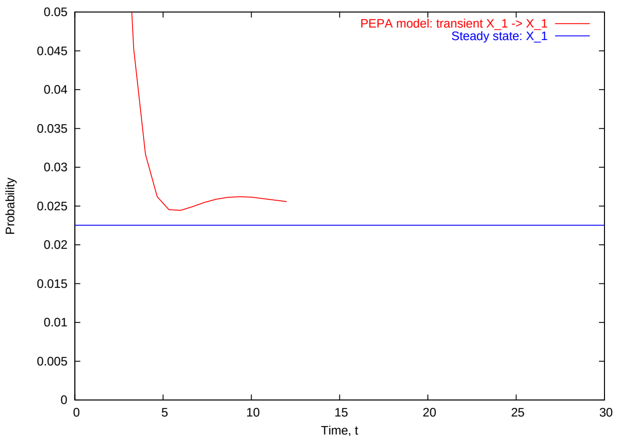
$\Rightarrow$



Types of Analysis

Steady-state and transient analysis in PEPA:

$$
\begin{array}{lcl}
A1 & \stackrel{def}{=} & (start, r_1).A2 + (pause, r_2).A3 \\
A2 & \stackrel{def}{=} & (run, r_3).A1 + (fail, r_4).A3 \\
A3 & \stackrel{def}{=} & (recover, r_1).A1 \\
AA & \stackrel{def}{=} & (run, \top).(alert, r_5).AA \\
Sys & \stackrel{def}{=} & AA \underset{\{run\}}{\bowtie} A1
\end{array}
\Rightarrow
$$

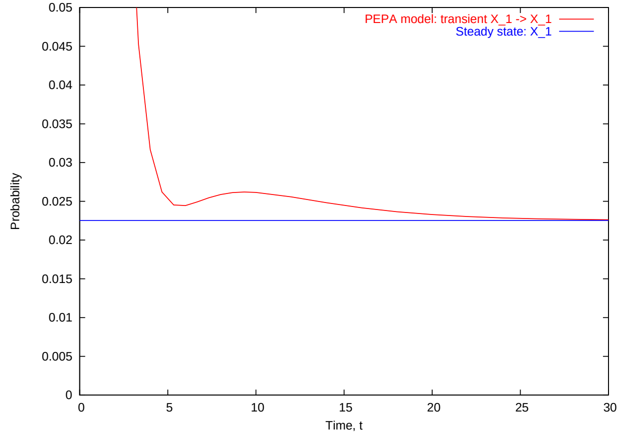


Types of Analysis

Steady-state and transient analysis in PEPA:

$$
\begin{aligned}
A1 &\stackrel{\text{def}}{=} (start, r_1).A2 + (pause, r_2).A3 \\
A2 &\stackrel{\text{def}}{=} (run, r_3).A1 + (fail, r_4).A3 \\
A3 &\stackrel{\text{def}}{=} (recover, r_1).A1 \\
AA &\stackrel{\text{def}}{=} (run, \top).(alert, r_5).AA \\
Sys &\stackrel{\text{def}}{=} AA \underset{\{run\}}{\bowtie} A1
\end{aligned}
$$

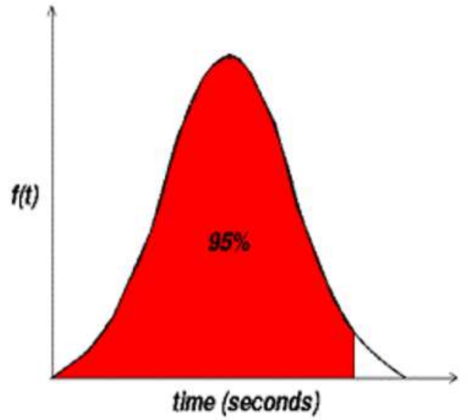
$\Rightarrow$



Passage-time Quantiles

Extract a passage-time density from a PEPA model:

$$
\begin{aligned}
A1 &\stackrel{\text{def}}{=} (\text{start}, r_1).A2 + (\text{pause}, r_2).A3 \\
A2 &\stackrel{\text{def}}{=} (\text{run}, r_3).A1 + (\text{fail}, r_4).A3 \\
A3 &\stackrel{\text{def}}{=} (\text{recover}, r_1).A1 \\
AA &\stackrel{\text{def}}{=} (\text{run}, \top).(\text{alert}, r_5).AA \\
Sys &\stackrel{\text{def}}{=} AA \underset{\{run\}}{\bowtie} A1
\end{aligned}
\quad \Longrightarrow
$$



State of the Art

- Good news
  - PEPA model: passage time/transient analysis - $O(10^8)$ states
  - Semi-Markov PEPA: passage time/transient analysis - $O(10^7)$ states

State of the Art

- Good news
  - PEPA model: passage time/transient analysis - $O(10^8)$ states
  - Semi-Markov PEPA: passage time/transient analysis - $O(10^7)$ states
- Bad news
  - This only represents 8 agents with 10 states each!

Ways forward for PEPA Agent modelling

- Either:
  - selective model aggregation

  ⇒ allows use of passage/transient

- Or:
  - development of approximate techniques

  ⇒ automated generation of MFA/ODE equations from PEPA model [c.f. Sumpter 2000, Hillston 2004]

SIR: Epidemiological model

- Consider fixed population of N computers
- Partition population into, computers that are:
  - susceptible to infection, $s(t)$
  - infected, $i(t)$
  - removed, $r(t)$
- Deterministic system:
  - $\frac{\mathrm{d}s(t)}{\mathrm{d}t} = -\beta s(t)i(t)$
  - $\frac{\mathrm{d}i(t)}{\mathrm{d}t} = \beta s(t)i(t) - \gamma i(t)$
  - $\frac{\mathrm{d}r(t)}{\mathrm{d}t} = \gamma i(t)$

Agent worm model

$$\begin{aligned}\text{Susceptible} \;&=\; (\text{infect}, \top).\text{Infected} \\ &\quad + (\text{patch}, \lambda_p).\text{Removed}\end{aligned}$$

Agent worm model

$$
\begin{aligned}
\text{Susceptible} &= (\text{infect}, \top).\text{Infected} \\
&\quad + (\text{patch}, \lambda_p).\text{Removed} \\
\text{Infected} &= (\text{infect}, \lambda_i).\text{Infected} \\
&\quad + (\text{repair}, \lambda_r).\text{Removed}
\end{aligned}
$$

Agent worm model

$$
\begin{aligned}
\text{Susceptible} &= (\text{infect}, \top).\text{Infected} \\
&\quad + (\text{patch}, \lambda_p).\text{Removed} \\
\text{Infected} &= (\text{infect}, \lambda_i).\text{Infected} \\
&\quad + (\text{repair}, \lambda_r).\text{Removed} \\
\text{Removed} &= (\text{rollback}, \lambda_s).\text{Susceptible}
\end{aligned}
$$

$$
\text{System} = \bigoplus_{\mathrm{i}=1}^{\mathrm{N}} \text{Susceptible}_{\mathrm{i}}
$$

Agent worm model

$$\begin{aligned}
\text{Susceptible} &= (\text{infect}, \top).\text{Infected} \\
&\quad + (\text{patch}, \lambda_p).\text{Removed} \\
\text{Infected} &= (\text{infect}, \lambda_i).\text{Infected} \\
&\quad + (\text{repair}, \lambda_r).\text{Removed} \\
\text{Removed} &= (\text{rollback}, \lambda_s).\text{Susceptible}
\end{aligned}$$

$$\begin{aligned}
\text{System}(\mathrm{p}, \mathrm{q}) &= \bigoplus_{i=1}^{p} \text{Infected}_{\mathrm{i}} \bigoplus_{\mathrm{i}=1}^{\mathrm{q}} \text{Susceptible}_{\mathrm{i}} \\
&\quad \bigoplus_{i=1}^{N-p-q} \text{Removed}_{\mathrm{i}}
\end{aligned}$$

Analysis possibilities

Sumpter Look to count numbers of agents, $A(t)$, in a given state by solving a derived mean field equation (MFE)

$$f(t,i) = \mathbb{E}(A(t+\Delta t) \mid A(t) = i)$$

Analysis possibilities

Sumpter Look to count numbers of agents, $A(t)$, in a given state by solving a derived mean field equation (MFE)

$$f(t,i) = \mathbb{E}(A(t+\Delta t) \mid A(t) = i)$$

Hillston Approximate number of components with a real numbered function

Analysis possibilities

Sumpter Look to count numbers of agents, $A(t)$, in a given state by solving a derived mean field equation (MFE)

$$f(t, i) = \mathbb{E}(A(t + \Delta t) \mid A(t) = i)$$

Hillston Approximate number of components with a real numbered function

- remodel using *bimodal assumption*

Agent Count-based Model

Bimodal characterisation of variables:

$$\begin{aligned} \mathrm{S_H} &\stackrel{\text{def}}{=} (\mathrm{infect}, \top).\mathrm{S_L} \\ &\quad + (\mathrm{patch}, r_2).\mathrm{S_L} \\ \mathrm{S_L} &\stackrel{\text{def}}{=} (\mathrm{rollback}, \lambda_s).\mathrm{S_H} \\ \vdots & \qquad \vdots \end{aligned} \quad \Rightarrow \quad \begin{aligned} \frac{\mathrm{d}s(t)}{\mathrm{d}t} &= -\beta s(t)i(t) \\ \frac{\mathrm{d}i(t)}{\mathrm{d}t} &= \beta s(t)i(t) - \gamma i(t) \\ \frac{\mathrm{d}r(t)}{\mathrm{d}t} &= \gamma i(t) \end{aligned}$$

Conclusion

- Internet worms have a reasonable biological analogy
- State spaces too large for traditional temporal modelling
- The answer: Selective aggregation/agent counting
- Passage times in an agent setting
  - a useful cost function for a model
  - probability of extinction within a given time