Mean-field analysis for large-scale GSMPs and interacting fluid models

Richard A. Hayden
rh@doc.ic.ac.uk
AESOP Seminar
November 29, 2012
Mean-field analysis for CTMCs

- $N$ interacting agents each with $D$ local states
- $N \rightarrow \infty$
- System of $D$ ODEs (independent of $N$)
Simple example: peer-to-peer software update
Simple example: peer-to-peer software update

\[ \dot{v}_y(t) = -\rho v_y(t) - (1/\gamma)v_y(t) - \frac{\beta v_y(t)v_u(t)}{N} + \lambda v_x(t) \]
Simple example: peer-to-peer software update

\[
\dot{\nu}_y(t) = -\rho \nu_y(t) - \frac{1}{\gamma} \nu_y(t) - \frac{\beta \nu_y(t) \nu_u(t)}{N} + \lambda \nu_x(t)
\]

Rate leaving state \(y\)
Simple example: peer-to-peer software update

\[ \dot{v}_y(t) = -\rho v_y(t) - \frac{1}{\gamma} v_y(t) - \frac{\beta v_y(t) v_u(t)}{N} + \lambda v_x(t) \]

Rate leaving state $y$

Rate entering state $y$
Simple example: peer-to-peer software update

Old node

Updated node

\[ \frac{1}{\gamma} \quad \lambda \quad \rho \]

\( \beta U / N \)

\( \lambda \quad \rho \quad \times 0.9N \)

\( \times 0.1N \)

Nodes in state \( y \)

Nodes in state \( z \)

Nodes in state \( x \)

Nodes in state \( u \)

Nodes in state \( v \)

Rescaled component count

Time, \( t \)
Simple example: peer-to-peer software update

\[ \frac{1}{\gamma} \xrightarrow{} \text{on}_y \xrightarrow{} \text{on}_z' \xrightarrow{} \text{off}_x \xrightarrow{} \text{on}_u \xrightarrow{} \text{off}_v \xrightarrow{} \beta U \frac{\lambda}{N} \xrightarrow{} \rho \times 0.9N \]

Old node

Updated node

\[ N = 20 \]

\begin{align*}
\text{Nodes in state } y & \quad \text{Nodes in state } z \\
\text{Nodes in state } x & \quad \text{Nodes in state } u \\
\text{Nodes in state } v & \quad \beta U \frac{\lambda}{N} \end{align*}

\begin{align*}
\text{Rescaled component count} & \quad \text{Rescaled component count} \\
\text{Time, } t & \quad \text{Time, } t
\end{align*}
Simple example: peer-to-peer software update

\[ \frac{1}{\gamma} \quad \frac{\beta U}{N} \]

Old node

Updated node

\[ N = 50 \]

\[ \lambda \]

\[ \rho \]

\[ \times 0.9N \]

\[ \times 0.1N \]

Nodes in state y
Nodes in state z
Nodes in state x
Nodes in state u
Nodes in state v
Simple example: peer-to-peer software update

\[ \frac{1}{\gamma} \quad \text{on}_y \quad \frac{\beta U}{N} \]  
\[ \lambda \quad \rho \quad \text{off}_x \times 0.9N \]  
\[ \lambda \quad \rho \quad \text{off}_v \times 0.1N \]  
Old node  
Updated node

\[ N = 100 \]
Simple example: peer-to-peer software update

\[ \frac{1}{\gamma} \]

\[ \lambda \]

\[ \rho \]

\[ \beta U \]

\[ N = 1000 \]

\[ N \times 0.9N \]

\[ \times 0.1N \]

\[ \text{Nodes in state } y \]

\[ \text{Nodes in state } z \]

\[ \text{Nodes in state } x \]

\[ \text{Rescaled component count} \]

\[ \text{Time, } t \]
Generally timed transitions
Generally timed (non-Markovian) transitions

Necessary to model realistic computer systems, e.g.:

- **Deterministic durations**: timeouts in communication protocols, networks with fixed-length packets, time to reset/reboot a server;

- **Generally distributed durations**: many (most?) uncertain activity durations do not follow exponential distributions!
Generally timed (non-Markovian) transitions

Necessary to model realistic computer systems, e.g.:

- **Deterministic durations**: timeouts in communication protocols, networks with fixed-length packets, time to reset/reboot a server;

- **Generally distributed durations**: many (most?) uncertain activity durations do not follow exponential distributions!

Generalised semi-Markov process (GSMP); underlying Markov process requires continuous state to track residual times
Generally timed (non-Markovian) transitions

Necessary to model realistic computer systems, e.g.:

- **Deterministic durations:** timeouts in communication protocols, networks with fixed-length packets, time to reset/reboot a server;
- **Generally distributed durations:** many (most?) uncertain activity durations do not follow exponential distributions!

**Generalised semi-Markov process (GSMP);** underlying Markov process requires continuous state to track residual times

- For large number $N$ of interacting components, even more expensive to solve exactly than the CTMC case;
- Phase-type approximation of generally distributed durations can require large local state spaces — e.g. the $k$-stage Erlang has coefficient of variation $1/\sqrt{k}$

**To keep things simple(r), we consider here just the deterministic case**
Peer-to-peer software update with deterministic timeout

\[ \dot{v}_y(t) = -\rho v_y(t) - \left(\frac{1}{\gamma}\right)v_y(t) - \frac{\beta v_y(t)v_u(t)}{N} + \lambda v_x(t) \]
Peer-to-peer software update with deterministic timeout

\[ \dot{v}_y(t) = -\rho v_y(t) - \frac{\beta v_y(t)v_u(t)}{N} + \lambda v_x(t) \]

Markovian transitions

\[ -\mathbf{1}_{\{t \geq \gamma\}} \lambda v_x(t - \gamma) \exp \left( -\int_{t-\gamma}^{t} \frac{\beta v_a(s)}{N} \, ds - \rho \gamma \right) \]
Peer-to-peer software update with deterministic timeout

\[ \dot{v}_y(t) = -\rho v_y(t) - \frac{\beta v_y(t)v_u(t)}{N} + \lambda v_x(t) \]

Markovian transitions

\[ - \mathbf{1}_{\{t \geq \gamma\}} \lambda v_x(t - \gamma) \exp \left( - \int_{t-\gamma}^{t} \frac{\beta v_a(s)}{N} ds - \rho \gamma \right) \]

Rate of deterministic clocks starting at \( t - \gamma \)
Peer-to-peer software update with deterministic timeout

\[ \dot{v}_y(t) = -\rho v_y(t) - \frac{\beta v_y(t)v_u(t)}{N} + \lambda v_x(t) \]

Markovian transitions

\[-\mathbf{1}_{\{t \geq \gamma\}} \lambda v_x(t - \gamma) \exp \left( - \int_{t-\gamma}^{t} \frac{\beta v_a(s)}{N} \, ds - \rho \gamma \right) \]

Rate of deterministic clocks starting at \( t - \gamma \)

\approx \text{timeout probability}
Peer-to-peer software update with deterministic timeout

Old node

Updated node

Rescaled component count

Nodes in state y
Nodes in state z
Nodes in state x

Nodes in state u
Nodes in state v

Time, $t$

$\text{Nodes in state y}$
$\text{Nodes in state z}$
$\text{Nodes in state x}$
Peer-to-peer software update with deterministic timeout

\[ N = 20 \]
Peer-to-peer software update with deterministic timeout

\[ N = 50 \]
Peer-to-peer software update with deterministic timeout

Old node

Updated node

$N = 100$
Peer-to-peer software update with deterministic timeout

\[ N = 1000 \]
Peer-to-peer software update with deterministic timeout

Old node

Updated node
Peer-to-peer software update with deterministic timeout

\[ \dot{v}_y(t) = -\rho v_y(t) - \frac{\beta v_y(t) v_u(t)}{N} + \lambda v_x(t) \]

\[- 1_{\{t \geq \gamma\}} \lambda v_x(t - \gamma) \exp \left( - \int_{t-\gamma}^{t} \frac{\beta v_a(s)}{N} ds - \rho \gamma \right)\]
Peer-to-peer software update with deterministic timeout

\[ v_y(t) = v_{y,0} + \int_0^t -\rho v_y(s) - \frac{\beta v_y(s)v_u(s)}{N} + \lambda v_x(s) \, ds \]
\[ - \int_0^{(t-\gamma)\vee 0} \lambda v_x(s) \exp \left( - \int_s^{s+\gamma} \frac{\beta v_a(u)}{N} \, du - \rho \gamma \right) \, ds \]
Peer-to-peer software update with deterministic timeout

\[
v_y(t) = v_{y,0} + \int_0^t \left( -\rho v_y(s) - \frac{\beta v_y(s)v_u(s)}{N} + \lambda v_x(s) \right) ds \\
- \int_0^{(t-\gamma)\vee 0} \lambda v_x(s) \exp \left( -\int_s^{s+\gamma} \frac{\beta v_a(u)}{N} du - \rho \gamma \right) ds \\
- \mathbf{1}_{\{t \geq \gamma\}} v_{y,0} \exp \left( -\int_0^{\gamma} \frac{\beta v_a(s)}{N} ds - \rho \gamma \right)
\]
Peer-to-peer software update with deterministic timeout

Old node

Updated node

Nodes in state $y$

Nodes in state $z$

Nodes in state $x$

Nodes in state $u$

Nodes in state $v$

Jump at $t = \gamma$

Rescaled component count

Time, $t$
Peer-to-peer software update with deterministic timeout

\[
\begin{align*}
\text{Old node} & : 0.9N \times \text{on}_y & \frac{\beta U}{N} \\
\text{Updated node} & : \gamma \quad \lambda \quad \rho \\
\text{on}_z' & \quad \text{off}_x \\
\text{on}_u & \quad \text{off}_v \times 0.1N
\end{align*}
\]

\[N = 20\]

Graphical representation of the system dynamics with nodes in states \(y, z, x, u, v\) and transitions governed by \(\lambda, \rho, \beta U/N\). The rescaled component counts over time, showing transitions at \(t = \gamma\).
Peer-to-peer software update with deterministic timeout

Old node

Updated node

\( N = 50 \)
Peer-to-peer software update with deterministic timeout

\[ N = 100 \]
Peer-to-peer software update with deterministic timeout

$$N = 1000$$

**Old node**

- **on**
- **off**

**Updated node**

- **on**
- **off**

Indicator graphs:

- Nodes in state y
- Nodes in state z
- Nodes in state x
- Nodes in state u
- Nodes in state v

Jump at $t = \gamma$
Peer-to-peer software update with deterministic timeout

\[ \frac{\beta U}{N} \]

\[ 0.9N \times \]

\[ \gamma \]

\[ \lambda \]

\[ \rho \]

\[ \rho \] \[ \rho \]

\[ \times 0.1N \]

\[ N = 10000 \]

\[ \text{Nodes in state y} \]

\[ \text{Nodes in state z} \]

\[ \text{Nodes in state x} \]

\[ \text{Nodes in state u} \]

\[ \text{Nodes in state v} \]

\[ \text{Jump at } t = \gamma \]
Population generalised semi-Markov processes (PGSMPs)
Population generalised semi-Markov processes (PGSMPs)

- Interacting components each inhabiting a *local state* in $S$
Population generalised semi-Markov processes (PGSMPs)

- Interacting components each inhabiting a *local state* in $S$
- Population process $\mathbf{x} = (x_s)_{s \in S}$ living in *global state space* $X$

Restrictions for mean-field analysis:

- At most one event clock active in any local state: $\forall s \in S: |\{ e \in E : s \in A_e \}| \leq 1$
- No components with uninterrupted cycles of deterministically timed behaviour
Population generalised semi-Markov processes (PGSMPs)

- Interacting components each inhabiting a \textit{local state} in $S$
- Population process $\mathbf{x} = (x_s)_{s \in S}$ living in \textit{global state space} $\mathcal{X}$
- \textit{Markovian transitions} $c \in \mathcal{C}$ each specified by \textit{multiset of tuples} $L_c \subseteq S \times S$ and a \textit{rate function} $r_c : \mathcal{X} \to \mathbb{R}_+$
Population generalised semi-Markov processes (PGSMPs)

- Interacting components each inhabiting a local state in $S$
- Population process $\mathbf{x} = (x_s)_{s \in S}$ living in global state space $X$
- Markovian transitions $c \in \mathcal{C}$ each specified by multiset of tuples $L_c \subseteq S \times S$ and a rate function $r_c : X \rightarrow \mathbb{R}_+$
- Deterministically timed transitions specified by event clocks $e \in \mathcal{E}$, with set of active states $A_e \subseteq S$, transition probability $p_e : A_e \times S \rightarrow [0, 1]$ and clock duration $d_e \in \mathbb{R}_{>0}$
Population generalised semi-Markov processes (PGSMPs)

- Interacting components each inhabiting a local state in \( S \)
- Population process \( \mathbf{x} = (x_s)_{s \in S} \) living in global state space \( \mathcal{X} \)
- Markovian transitions \( c \in \mathcal{C} \) each specified by multiset of tuples \( L_c \subseteq S \times S \) and a rate function \( r_c : \mathcal{X} \to \mathbb{R}_+ \)
- Deterministically timed transitions specified by event clocks \( e \in \mathcal{E} \), with set of active states \( A_e \subseteq S \), transition probability \( p_e : A_e \times S \to [0, 1] \) and clock duration \( d_e \in \mathbb{R}_{>0} \)

Restrictions for mean-field analysis:
Population generalised semi-Markov processes (PGSMPs)

- Interacting components each inhabiting a local state in $S$
- Population process $\mathbf{x} = (x_s)_{s \in S}$ living in global state space $\mathcal{X}$
- Markovian transitions $c \in C$ each specified by multiset of tuples $L_c \subseteq S \times S$ and a rate function $r_c : \mathcal{X} \to \mathbb{R}_+$
- Deterministically timed transitions specified by event clocks $e \in \mathcal{E}$, with set of active states $A_e \subseteq S$, transition probability $p_e : A_e \times S \to [0, 1]$ and clock duration $d_e \in \mathbb{R}_{>0}$

Restrictions for mean-field analysis:

- At most one event clock active in any local state:
  $\forall s \in S : |\{e \in \mathcal{E} : s \in A_e\}| \leq 1$
Interacting components each inhabiting a *local state* in $S$

Population process $\mathbf{x} = (x_s)_{s \in S}$ living in *global state space* $\mathcal{X}$

*Markovian transitions* $c \in \mathcal{C}$ each specified by *multiset of tuples* $L_c \subseteq S \times S$ and a *rate function* $r_c : \mathcal{X} \to \mathbb{R}_+$

*Deterministically timed* transitions specified by event clocks $e \in \mathcal{E}$, with set of *active states* $A_e \subseteq S$, *transition probability* $p_e : A_e \times S \to [0, 1]$ and *clock duration* $d_e \in \mathbb{R}_{>0}$

**Restrictions for mean-field analysis:**

- At most one event clock active in any local state:
  \[ \forall s \in S : |\{ e \in \mathcal{E} : s \in A_e \}| \leq 1 \]

- No components with *uninterrupted cycles of deterministically timed behaviour*
General mean-field differential equation
General mean-field differential equation

\[ \dot{v}_s(t) = \sum_{c \in C} l^c_s r_c(v(t)) + \ldots \]

where:

\[ l^c_s := |\{(s', s) \in L_c\}| - |\{(s, s') \in L_c\}| \]
General mean-field differential equation

\[ \dot{v}_s(t) = \sum_{c \in C} l^c_s r_c(v(t)) + \sum_{e \in E} \mathbf{1}_{\{t \geq d_e\}} \sum_{z \in A_e} \left( \ldots \right) \]

Markovian transitions

\[ \sum_{s' \in A_e} \sum_{c \in C} \left| \left\{ (y, z) \in L_c : y \notin A_e \right\} \right| r_c(v(t - d_e)) \]

Rate of \( e \) clocks starting at \( t - d_e \) by expo. transits. \( \rightarrow z \)

\[ \times \left[ Y_e^{t-d_e}(d_e, v) \right]_{z,s'} \times p_e(s', s) - \ldots \]

Probability that \( e \) completes at time \( t \) ending in state \( s' \) via \( s \)

where:

\[ l^c_s := \left| \left\{ (s', s) \in L_c \right\} \right| - \left| \left\{ (s, s') \in L_c \right\} \right| \]
General mean-field differential equation

\[ \dot{v}_s(t) = \sum_{c \in C} l^c_s r_c(v(t)) + \sum_{e \in \mathcal{E}} 1_{\{t \geq d_e\}} \sum_{z \in \mathcal{A}_e} \left( \right) \]

Markovian transitions

\[ \sum_{s' \in \mathcal{A}_e} \sum_{c \in C} \frac{1}{|L_c|} \left| \{(y, z) \in L_c : y \notin \mathcal{A}_e\} \right| r_c(v(t - d_e)) \]

Rate of \( e \) clocks starting at \( t - d_e \) by expo. transits. \( \rightarrow z \)

\[ \times \left[ Y^{t-d_e}_{e}(d_e, v) \right]_{z, s'} \times p_e(s', s) - \ldots \]

Probability that \( e \) completes at time \( t \) ending in state \( s' \) via \( s \)

where:

\[ l^c_s := |\{(s', s) \in L_c\}| - |\{(s, s') \in L_c\}| \]

with auxiliary time-inhomogeneous linear IVP:

\[ Y^{t_0}_{e}(0, v) = I \]

\[ \dot{Y}^{t_0}_{e}(u, v) = Y^{t_0}_{e}(u, v)Q_e(v(t_0 + u)) \]
General mean-field differential equation

\[
\dot{v}_s(t) = \sum_{c \in C} l^c_s r_c(v(t)) + \sum_{e \in \mathcal{E}} \mathbf{1}_{\{t \geq d_e\}} \sum_{z \in \mathcal{A}_e} \left( \ldots \right)
\]

Markovian transitions

\[- \mathbf{1}_{\{s \in \mathcal{A}_e\}} \sum_{s' \in \mathcal{S}} \sum_{c \in C} \left| \{ (y, z) \in \mathcal{L}_c : y \notin \mathcal{A}_e \} \right| r_c(v(t - d_e))
\]

Rate of \(e\) clocks starting at \(t - d_e\) by expo. transits. \(\rightarrow z\)

\[
\times \left[ Y_{e}^{t-d_e}(d_e, v) \right]_{z, s} \times p_e(s, s')
\]

Probability that \(e\) completes at time \(t\) ending in state \(s\) via \(s'\)

where:

\[
l^c_s := \left| \{ (s', s) \in \mathcal{L}_c \} \right| - \left| \{ (s, s') \in \mathcal{L}_c \} \right|
\]

with auxiliary time-inhomogeneous linear IVP:

\[
Y_{e}^{t_0}(0, v) = I
\]

\[
\dot{Y}_{e}^{t_0}(u, v) = Y_{e}^{t_0}(u, v)Q_e(v(t_0 + u))
\]
Mean-field convergence theorem (FLLN)

- Sequence of models indexed by $N$ with initial conditions $x_N^N(0) = N x_0$
- There exists some locally Lipschitz $r_c(x) := (1/N)r_c^N(Nx)$ independently of $N$ (density dependence à la Kurtz)
- $r_c^N(x) \leq R(\|x\| + 1)$ for all $x \in \mathcal{X}^N$ where $R \in \mathbb{R}_+$ is independent of $N$

**Theorem: PGSMP FLLN**

For $T, \epsilon > 0$: $\lim_{N \to \infty} \mathbb{P} \left\{ \sup_{t \leq T} \|(1/N)x_N^N(t) - \bar{v}(t)\| > \epsilon \right\} = 0$

**Proof.**

- In the delay-only case: trace-wise representation of the processes by random time changes of Poisson processes and time-delayed Poisson processes. Subsequent technical details then fairly standard.[1]
- In the general case: seems the continuous state-space elements must be accommodated explicitly . . . cf. the second part of the talk.

---

Delay-only PGSMPs

Additional restrictions:
Delay-only PGSMPs

Additional restrictions:

- Markovian transitions cannot be enabled *locally* concurrently with deterministic ones
Delay-only PGSMPs

Additional restrictions:

- Markovian transitions cannot be enabled *locally* concurrently with deterministic ones
- When any deterministic transition completes, the component always jumps into a fixed state that does not immediately enable another deterministic clock
Delay-only PGSMPs

Additional restrictions:

- Markovian transitions cannot be enabled *locally* concurrently with deterministic ones
- When any deterministic transition completes, the component always jumps into a fixed state that does not immediately enable another deterministic clock

Delay-only mean-field equation:

$$\dot{v}_s(t) = \sum_{c \in C} l^c_s r_c(v(t))$$

Markovian transitions
Delay-only PGSMPs

Additional restrictions:

- Markovian transitions cannot be enabled *locally* concurrently with deterministic ones
- When any deterministic transition completes, the component always jumps into a fixed state that does not immediately enable another deterministic clock

Delay-only mean-field equation:

\[
\dot{v}_s(t) = \sum_{c \in C} l^c_r_c(v(t)) + \sum_{e \in E} \sum_{c \in C} 1_{\{t \geq d_e\}}
\]

Markovian transitions
Delay-only PGSMPs

Additional restrictions:

- Markovian transitions cannot be enabled *locally* concurrently with deterministic ones
- When any deterministic transition completes, the component always jumps into a fixed state that does not immediately enable another deterministic clock

Delay-only mean-field equation:

\[
\dot{v}_s(t) = \sum_{c \in C} l_s^c r_c(v(t)) + \sum_{e \in E} \sum_{c \in C} 1\{t \geq d_e\}
\]

Markovian transitions

\[
\left( \sum_{s' \in A_e} p_e(s', s) l_{s'}^c r_c(v(t - d_e)) - 1\{s \in A_e\} l_s^c r_c(v(t - d_e)) \right)
\]

Rate of \( e \) clocks starting at \( t - d_e \) by expo. transits. \( \rightarrow s' \) ending in \( s \)

Rate of \( e \) clocks starting at \( t - d_e \) by expo. transits. \( \rightarrow s \)
Delay-only PGSMPs

Additional restrictions:

- Markovian transitions cannot be enabled \textit{locally} concurrently with deterministic ones
- When any deterministic transition completes, the component always jumps into a fixed state that does not immediately enable another deterministic clock

Delay-only mean-field equation:

\[
v_s(t) = x_{s,0} + \sum_{c \in C} l_s^c \int_0^t r_c(v(u)) \, du \\
+ \sum_{e \in E} \sum_{c \in C} \left( \sum_{s' \in A_e} p_e(s', s) l_{s'}^c - 1_{\{s \in A_e\}} l_s^c \right) \int_0^{(t-d_e) \vee 0} r_c(v(u)) \, du
\]
Delay-only PGSMPs

Delay-only mean-field equation:

\[ v_s(t) = x_{s,0} + \sum_{c \in C} l_s^c \int_0^t r_c(v(u)) \, du \]

\[ + \sum_{e \in E} \sum_{c \in C} \left( \sum_{s' \in A_e} p_e(s', s) l_{s'}^c - 1_{\{s \in A_e\}} l_s^c \right) \int_0^{(t-d_e) \lor 0} r_c(v(u)) \, du \]
Delay-only PGSMPs

Delay-only mean-field equation:

\[ v_s(t) = x_{s,0} + \sum_{c \in C} l^c_s \int_0^t r_c(v(u)) \, du \]

\[ + \sum_{e \in E} \sum_{c \in C} \left( \sum_{s' \in A_e} p_e(s', s) l^c_{s'} - 1_{\{s \in A_e\}} l^c_s \right) \int_0^{(t-d_e) \vee 0} r_c(v(u)) \, du \]

Poisson process delayed random time change representation of counting processes:

\[ x_s(t) = x_{s,0} + \sum_{c \in C} l^c_s P_c \left( \int_0^t r_c(v(u)) \, du \right) \]

\[ + \sum_{e \in E} \sum_{c \in C} \left( \sum_{s' \in A_e} p_e(s', s) l^c_{s'} - 1_{\{s \in A_e\}} l^c_s \right) P_c \left( \int_0^{(t-d_e) \vee 0} r_c(v(u)) \, du \right) \]

where \( \{P_c\}_{c \in C} \) are mutually indep. rate-1 Poisson processes
More general mixed discrete–continuous local state spaces
Nodes are in one of two discrete states: active (a) or idle (i)
Nodes are in one of two discrete states: active (a) or idle (i)

Idle nodes are awaiting stimulus, active nodes are exchanging collected data with a neighbour
Wireless sensor network model

Nodes are in one of two discrete states: *active* (a) or *idle* (i)
- Idle nodes are awaiting stimulus, active nodes are exchanging collected data with a neighbour
- Each has own battery, drains at a state-dependent rate

\[ \mathbb{1}_{\{B_t > 0\}} \lambda \]

\[ B_t \]

\[ 1 \{ B_t > 0 \} \lambda \]

\[ ? \]
Nodes are in one of two discrete states: \textit{active} (a) or \textit{idle} (i)

- Idle nodes are awaiting stimulus, active nodes are exchanging collected data with a neighbour
- Each has own battery, drains at a state-dependent rate
- Threshold control — wireless radios operate at two different power levels: $0 < B_t \leq B^*$ (low) or $B_t > B^*$ (high)
Nodes are in one of two discrete states: active \( (a) \) or idle \( (i) \)

Idle nodes are awaiting stimulus, active nodes are exchanging collected data with a neighbour.

Each has own battery, drains at a state-dependent rate.

Threshold control — wireless radios operate at two different power levels: \( 0 < B_t \leq B^* \) (low) or \( B_t > B^* \) (high)

\[
 r(A_l(t), A_h(t)) := (\mathbf{1}_{0 < B_t \leq B^*} \beta_l + \mathbf{1}_{B_t > B^*} \beta_h) \frac{A_l(t) + A_h(t) - 1}{N}
\]
Wireless sensor network model

- Nodes are in one of two discrete states: *active* (a) or *idle* (i)
- Idle nodes are awaiting stimulus, active nodes are exchanging collected data with a neighbour
- Each has own battery, drains at a state-dependent rate
- Threshold control — wireless radios operate at two different power levels: $0 < B_t \leq B^* \text{ (low)}$ or $B_t > B^* \text{ (high)}$

\[ r(A_i(t), A_h(t)) := 1_{\{0 < B_t \leq B^*\}} \beta_l \frac{\tau_l(A_i(t) - 1) + \tau_h A_h(t)}{N} + 1_{\{B_t > B^*\}} \beta_h \frac{\tau_l A_i(t) + \tau_h (A_h(t) - 1)}{N} \]

Presumably $\beta_l \leq \beta_h$, $\tau_l \leq \tau_h$
Wireless sensor network model

- Nodes are in one of two discrete states: *active* (a) or *idle* (i)
- Idle nodes are awaiting stimulus, active nodes are exchanging collected data with a neighbour
- Each has own battery, drains at a state-dependent rate
- Threshold control — wireless radios operate at two different power levels: \(0 < B_t \leq B^*\) (low) or \(B_t > B^*\) (high)

\[
r(A_l(t), A_h(t)) := 1_{\{0 < B_t \leq B^*\}} \beta_l \frac{\tau_l (A_l(t) - 1) + \tau_h A_h(t)}{N}
+ 1_{\{B_t > B^*\}} \beta_h \frac{\tau_l A_l(t) + \tau_h (A_h(t) - 1)}{N}
\]

Presumably \(\beta_l \leq \beta_h\), \(\tau_l \leq \tau_h\) and \(\gamma_h \leq \gamma_l \leq \gamma_i \leq 0\)
Mean-field PDEs

\[
\begin{align*}
\gamma_i & : \text{state } i, \quad 0 < B_t \\
\gamma_l & : \text{state } a, \quad 0 < B_t \leq B^* \\
\gamma_h & : \text{state } a, \quad B_t > B^* \\
0 & : \text{otherwise}
\end{align*}
\]

\[
F_a(t, z), F_i(t, z) : \text{proportion of nodes in } a \text{ or } i, \text{ battery } \leq z
\]
Mean-field PDEs

\[ r(A_l(t), A_h(t)) := \mathbbm{1}_{0 < B_t \leq B^*} \beta_l \frac{\tau_l(A_l(t) - 1) + \tau_h A_h(t)}{N} + \mathbbm{1}_{B_t > B^*} \beta_h \frac{\tau_l A_l(t) + \tau_h (A_h(t) - 1)}{N} \]

\[ \frac{d B_t}{dt} = \begin{cases} 
\gamma_i : \text{state } i, 0 < B_t \\
\gamma_l : \text{state } a, 0 < B_t \leq B^* \\
\gamma_h : \text{state } a, B_t > B^* \\
0 : \text{otherwise}
\end{cases} \]

▶ \( f_a(t, z) := \frac{\partial}{\partial z} F_a(t, z), \ f_i(t, z) := \frac{\partial}{\partial z} F_i(t, z) \) for \( z \in (0, 1] \)
Mean-field PDEs

\[ r(A_l(t), A_h(t)) := 1_{\{0 < B_t \leq B^*\}} \beta_l \frac{\gamma_l(A_l(t) - 1) + \tau_h A_h(t)}{N} + 1_{\{B_t > B^*\}} \beta_h \frac{\tau_l A_l(t) + \tau_h (A_h(t) - 1)}{N} \]

\[
\begin{cases}
\gamma_i : \text{state } i, 0 < B_t \\
\gamma_l : \text{state } a, 0 < B_t \leq B^* \\
\gamma_h : \text{state } a, B_t > B^* \\
0 : \text{otherwise}
\end{cases}
\]

\[
\frac{dB_t}{dt} = \begin{cases}
\lambda & \text{Discharging of batteries in } [t, t+\delta t] \\
\beta_l \tau_l (A_l(t) - 1) + \beta_h \tau_h (A_h(t) - 1) & \text{Discrete transitions } i \rightarrow a \text{ in } [t, t+\delta t] \\
\beta_l \tau_l (A_l(t) - 1) + \beta_h \tau_h (A_h(t) - 1) & \text{Discrete transitions } a \rightarrow i \text{ in } [t, t+\delta t]
\end{cases}
\]

\[
\begin{align*}
f_a(t, z) &:= \frac{\partial}{\partial z} F_a(t, z), \quad f_i(t, z) := \frac{\partial}{\partial z} F_i(t, z) \text{ for } z \in (0, 1) \\
fa(t + \delta t, z) &\approx f_a(t, z) + \left( f_a \left( t, z + [1_{\{z \leq B^*\}} \gamma_l + 1_{\{z > B^*\}} \gamma_h] \delta t \right) - f_a(t, z) \right) - (1_{\{z \leq B^*\}} \beta_l + 1_{\{z > B^*\}} \beta_h) \delta tf_a(t, z) \\
+ \lambda \delta tf_i(t, z) &\quad \text{Discharging of batteries in } [t, t+\delta t] \\
- (1_{\{z \leq B^*\}} \beta_l + 1_{\{z > B^*\}} \beta_h) \delta tf_a(t, z) \left( \tau_l \int_0^{B^*} f_a(t, v) \, dv + \tau_h \int_{B^*}^{1} f_a(t, v) \, dv \right) + o(\delta t)
\end{align*}
\]
Mean-field PDEs

\[
\begin{align*}
    r(A_l(t), A_h(t)) := & \ 1_{\{0 < B_t \leq B^*\}} \beta_l \frac{\tau_l (A_l(t) - 1) + \tau_h A_h(t)}{N} \\
    & + 1_{\{B_t > B^*\}} \beta_h \frac{\tau_l A_l(t) + \tau_h (A_h(t) - 1)}{N}
\end{align*}
\]

\[ dB_t = \begin{cases} 
    \gamma_i : \text{state } i, 0 < B_t \\
    \gamma_l : \text{state } a, 0 < B_t \leq B^* \\
    \gamma_h : \text{state } a, B_t > B^* \\
    0 : \text{otherwise}
\end{cases} \]

\[ f_a(t, z) := \frac{\partial}{\partial z} F_a(t, z), \ f_i(t, z) := \frac{\partial}{\partial z} F_i(t, z) \text{ for } z \in (0, 1] \]

\[
\begin{align*}
    \frac{\partial f_a(t, z)}{\partial t} - (1_{\{z \leq B^*\}} \gamma_l + 1_{\{z > B^*\}} \gamma_h) \frac{\partial f_a(t, z)}{\partial z} = \\
    \lambda f_i(t, z) - (1_{\{z \leq B^*\}} \beta_l + 1_{\{z > B^*\}} \beta_h) f_a(t, z) \left( \tau_l \int_0^{B^*} f_a(t, v) \, dv + \tau_h \int_{B^*}^1 f_a(t, v) \, dv \right)
\end{align*}
\]
Mean-field PDEs

\[
\frac{dB_t}{dt} = \begin{cases} 
\gamma_i : \text{state } i, \ 0 < B_t \\
\gamma_l : \text{state } a, \ 0 < B_t \leq B^* \\
\gamma_h : \text{state } a, \ B_t > B^* \\
0 : \text{otherwise}
\end{cases}
\]

\[
r(A_l(t), A_h(t)) := 1_{\{0 < B_t \leq B^*\}} \beta_l \frac{\tau_l(A_l(t) - 1) + \tau_h A_h(t)}{N} + 1_{\{B_t > B^*\}} \beta_h \frac{\tau_l A_l(t) + \tau_h (A_h(t) - 1)}{N}
\]

\[
\lambda 1_{\{B_t > 0\}} r(A_l(t), A_h(t))
\]

\[
\begin{align*}
e_a(t), e_i(t) & : \text{proportion of nodes in } a \text{ or } i \text{ with empty battery}
\end{align*}
\]

\[
e_a(t + \delta t) \approx e_a(t) + \int_0^{\gamma_l \delta t} f_a(t, \nu) d\nu + o(\delta t)
\]

Discharging of batteries in \([t, t + \delta t]\)
Mean-field PDEs

\[ r(A_l(t), A_h(t)) := 1_{0 < B_t \leq B^*} \beta_l \frac{\tau_l(A_l(t) - 1) + \tau_h A_h(t)}{N} + 1_{B_t > B^*} \beta_h \frac{\tau_l A_l(t) + \tau_h (A_h(t) - 1)}{N} \]

\[ dB_t = \begin{cases} 
\gamma_i : \text{state } i, 0 < B_t \\
\gamma_l : \text{state } a, 0 < B_t \leq B^* \\
\gamma_h : \text{state } a, B_t > B^* \\
0 : \text{otherwise}
\end{cases} \]

- \( e_a(t), e_l(t) \): proportion of nodes in a or l with empty battery

\[ \frac{de_a(t)}{dt} = \gamma_l f_a(t, 0) \]
Mean-field PDEs

\[ r(A_l(t), A_h(t)) := 1_{0 < B_t \leq B^*} \beta_l \frac{\tau_l(A_l(t) - 1) + \tau_h A_h(t)}{N} + 1_{B_t > B^*} \beta_h \frac{\tau_l A_l(t) + \tau_h (A_h(t) - 1)}{N} \]

\[ 1_{B_t > 0} \lambda r(A_l(t), A_h(t)) \]

\[ \frac{dB_t}{dt} = \left\{ \begin{array}{ll}
\gamma_i : \text{state } i, 0 < B_t \\
\gamma_l : \text{state } a, 0 < B_t \leq B^* \\
\gamma_h : \text{state } a, B_t > B^* \\
0 : \text{otherwise}
\end{array} \right. \]

- System of two non-linear partial (functional) differential equations with ordinary differential equations capturing the mass at zero
Mean-field PDEs

\[ r(A_l(t), A_h(t)) := 1_{0 < B_t \leq B^*} \beta_l \frac{\tau_l(A_l(t) - 1) + \tau_h A_h(t)}{N} + 1_{B_t > B^*} \beta_h \frac{\tau_l A_l(t) + \tau_h (A_h(t) - 1)}{N} \]

1\{B_t > 0\} \lambda \xrightarrow{\text{r}(A_l(t), A_h(t))} \n
\[ \frac{dB_t}{dt} = \begin{cases} \gamma_i : \text{state } i, \ 0 < B_t \\ \gamma_l : \text{state } a, \ 0 < B_t \leq B^* \\ \gamma_h : \text{state } a, \ B_t > B^* \\ 0 : \text{otherwise} \end{cases} \]

- System of two non-linear partial (functional) differential equations with ordinary differential equations capturing the mass at zero
- Specify initial conditions at \( t = 0 \) and also boundary conditions \( f_a(t, 1) = f_i(t, 1) = 0 \) for \( t > 0 \)
Mean-field PDEs

\[
r(A_l(t), A_h(t)) := 1_{\{0 < B_t \leq B^*\}} \beta_l \frac{\gamma_l(A_l(t)-1) + \tau_h A_h(t)}{N} + 1_{\{B_t > B^*\}} \beta_h \frac{\gamma_h A_l(t) + \tau_h (A_h(t)-1)}{N}
\]

- System of two non-linear partial (functional) differential equations with ordinary differential equations capturing the mass at zero
- Specify initial conditions at \( t = 0 \) and also boundary conditions \( f_a(t, 1) = f_i(t, 1) = 0 \) for \( t > 0 \)
- Can be solved inexpensively using standard finite difference techniques
Example solutions

\[ r(A_i(t), A_h(t)) := 1_{0 < B_t \leq B^*} \beta_l \frac{\tau_l(A_i(t) - 1) + \tau_h A_h(t)}{N} + 1_{B_t > B^*} \beta_h \frac{\tau_l A_i(t) + \tau_h (A_h(t) - 1)}{N} \]

\[ 1_{B_t > 0} \lambda \]

\[ r(A_i(t), A_h(t)) \]

\[ dB_t \quad \begin{cases} 
\gamma_i : \text{state } i, \quad 0 < B_t \\
\gamma_l : \text{state } a, \quad 0 < B_t \leq B^* \\
\gamma_h : \text{state } a, \quad B_t > B^* \\
0 : \text{otherwise}
\end{cases} \]
Example solutions

\[ r(A_l(t), A_h(t)) := 1_{\{0 < B_t \leq B^*\}} \beta_l \frac{\tau_l(A_l(t) - 1) + \tau_h A_h(t)}{N} + 1_{\{B_t > B^*\}} \beta_h \frac{\tau_l A_l(t) + \tau_h (A_h(t) - 1)}{N} \]

\[ \frac{dB_t}{dt} = \begin{cases} 
\gamma_i : \text{state } i, 0 < B_t \\
\gamma_l : \text{state } a, 0 < B_t \leq B^* \\
\gamma_h : \text{state } a, B_t > B^* \\
0 : \text{otherwise} 
\end{cases} \]

\[ N = 10 \]
Example solutions

\[ r(A_i(t), A_h(t)) := 1_{\{0 < B_t \leq B^*\}} \beta_l \frac{\tau_l(A_i(t)-1)+\tau_h A_h(t)}{N} + 1_{\{B_t > B^*\}} \beta_h \frac{\tau_l A_i(t)+\tau_h (A_h(t)-1)}{N} \]

\[ 1_{\{B_t > 0\}} \lambda \circlearrowright r(A_i(t), A_h(t)) \]

\[ \frac{dB_t}{dt} = \begin{cases} 
\gamma_i : \text{state } i, 0 < B_t \\
\gamma_l : \text{state } a, 0 < B_t \leq B^* \\
\gamma_h : \text{state } a, B_t > B^* \\
0 : \text{otherwise}
\end{cases} \]

\[ N = 20 \]
Example solutions

\[ r(A_l(t), A_h(t)) := 1_{0 < B_t \leq B^*} \beta_l \frac{\tau_l(A_l(t) - 1) + \tau_h A_h(t)}{N} + 1_{B_t > B^*} \beta_h \frac{\tau_l A_l(t) + \tau_h (A_h(t) - 1)}{N} \]

\[ 1_{\{B_t > 0\}} \lambda B_t \]

\[ \frac{dB_t}{dt} = \begin{cases} 
\gamma_i : \text{state i, } 0 < B_t \\
\gamma_l : \text{state a, } 0 < B_t \leq B^* \\
\gamma_h : \text{state a, } B_t > B^* \\
0 : \text{otherwise}
\end{cases} \]

\[ N = 100 \]
Example solutions

\[ r(A_l(t), A_h(t)) := 1_{0 < B_t \leq B^*} \beta_l \frac{\tau_l(A_l(t)-1) + \tau_h A_h(t)}{N} + 1_{B_t > B^*} \beta_h \frac{\tau_l A_l(t) + \tau_h (A_h(t)-1)}{N} \]

\[ \frac{dB_t}{dt} = \begin{cases} 
\gamma_i : \text{state i, } 0 < B_t \\
\gamma_l : \text{state a, } 0 < B_t \leq B^* \\
\gamma_h : \text{state a, } B_t > B^* \\
0 : \text{otherwise}
\end{cases} \]

\[ N = 1000 \]
Example solutions

\[ r(A_i(t), A_h(t)) := \begin{cases} 1_{\{0 < B_t \leq B^*\}} \beta_l \frac{\tau_l(A_i(t)-1)+\tau_h A_h(t)}{N} + 1_{\{B_t > B^*\}} \beta_h \frac{\tau_l A_i(t)+\tau_h (A_h(t)-1)}{N} \\ 1 \{B_t > 0\} \lambda \end{cases} \]

\[
\begin{aligned}
\frac{dB_t}{dt} &= \\
\gamma_i &\text{: state } i, \ 0 < B_t \\
\gamma_l &\text{: state } a, \ 0 < B_t \leq B^* \\
\gamma_h &\text{: state } a, \ B_t > B^* \\
0 &\text{: otherwise}
\end{aligned}
\]

<table>
<thead>
<tr>
<th>Compared to single trace</th>
<th>Pop. size</th>
<th>Avg. error</th>
<th>Max. error</th>
</tr>
</thead>
<tbody>
<tr>
<td>(N = 10)</td>
<td>0.1425</td>
<td>0.3540</td>
<td></td>
</tr>
<tr>
<td>(N = 100)</td>
<td>0.0264</td>
<td>0.1212</td>
<td></td>
</tr>
<tr>
<td>(N = 1000)</td>
<td>0.0138</td>
<td>0.0345</td>
<td></td>
</tr>
<tr>
<td>(N = 10000)</td>
<td>0.0042</td>
<td>0.0336</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compared to mean of 1000 traces</th>
<th>Pop. size</th>
<th>Avg. error</th>
<th>Max. error</th>
</tr>
</thead>
<tbody>
<tr>
<td>(N = 10)</td>
<td>0.0307</td>
<td>0.0577</td>
<td></td>
</tr>
<tr>
<td>(N = 100)</td>
<td>0.0042</td>
<td>0.0234</td>
<td></td>
</tr>
<tr>
<td>(N = 1000)</td>
<td>Simulation too costly</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(N = 10000)</td>
<td>Simulation too costly</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Mean-field convergence

Tricks with Poisson processes do not work in general
Mean-field convergence

Tricks with Poisson processes do not work in general

- Again we consider a sequence of models indexed by $N$ with increasing population
Mean-field convergence

Tricks with Poisson processes do not work in general

► Again we consider a sequence of models indexed by $N$ with increasing population

► For a given $t \in \mathbb{R}_+$, $F^N_a(t, \cdot)$ and $F^N_i(t, \cdot)$ induce, respectively, empirical (random) measures $\mu^N_a(t)$ and $\mu^N_i(t)$ living in $\mathcal{P}(\mathbb{R}_+)$
Mean-field convergence

Tricks with Poisson processes do not work in general

Again we consider a sequence of models indexed by $N$ with increasing population.

For a given $t \in \mathbb{R}_+$, $F^N_a(t, \cdot)$ and $F^N_i(t, \cdot)$ induce, respectively, empirical (random) measures $\mu^N_a(t)$ and $\mu^N_i(t)$ living in $\mathcal{P}(\mathbb{R}_+)$.

We view $(\mu^N_a, \mu^N_i)$ as a stochastic process living in $D_{\mathcal{P}(\mathbb{R}_+)^2}[0, \infty)$.
Mean-field convergence

Tricks with Poisson processes do not work in general

- Again we consider a sequence of models indexed by \( N \) with increasing population
- For a given \( t \in \mathbb{R}_+ \), \( F_a^N(t, \cdot) \) and \( F_i^N(t, \cdot) \) induce, respectively, empirical (random) measures \( \mu_a^N(t) \) and \( \mu_i^N(t) \) living in \( \mathcal{P}(\mathbb{R}_+) \)
- We view \((\mu_a^N, \mu_i^N)\) as a stochastic process living in \( D_{\mathcal{P}(\mathbb{R}_+)^2}[0, \infty) \)
- And we show that \((\mu_a^N, \mu_i^N) \Rightarrow (\mu_a, \mu_i)\) where the convergence is weak on the metric space \( D_{\mathcal{P}(\mathbb{R}_+)^2}[0, \infty) \)
Mean-field convergence

Tricks with Poisson processes do not work in general

► Again we consider a sequence of models indexed by \( N \) with increasing population

► For a given \( t \in \mathbb{R}_+ \), \( F_a^N(t, \cdot) \) and \( F_i^N(t, \cdot) \) induce, respectively, empirical (random) measures \( \mu_a^N(t) \) and \( \mu_i^N(t) \) living in \( \mathcal{P}(\mathbb{R}_+) \)

► We view \((\mu_a^N, \mu_i^N)\) as a stochastic process living in \( D_{\mathcal{P}(\mathbb{R}_+)^2}[0, \infty) \)

► And we show that \((\mu_a^N, \mu_i^N) \Rightarrow (\mu_a, \mu_i)\) where the convergence is weak on the metric space \( D_{\mathcal{P}(\mathbb{R}_+)^2}[0, \infty) \)

► Finally we show that the limit measures \( \mu_a \) and \( \mu_i \) are non-random and that their CDFs satisfy the mean-field PDEs
Ongoing work

- General framework for mean-field limits of interacting piecewise deterministic Markov processes (PDMPs)
Ongoing work

- General framework for mean-field limits of interacting piecewise deterministic Markov processes (PDMPs)

Steady-state mean-field limits:
Ongoing work

- General framework for mean-field limits of interacting piecewise deterministic Markov processes (PDMPs)

Steady-state mean-field limits:

- Much harder to prove convergence here
Ongoing work

- General framework for mean-field limits of interacting piecewise deterministic Markov processes (PDMPs)

Steady-state mean-field limits:

- Much harder to prove convergence here

- Will not always hold — likely to depend on intricate stability properties of the limiting PDEs
Thank you, questions?