An Inductive Technique for Parameterised Model Checking of Degenerative Distributed Randomised Protocols

Douglas Graham,[a,1,2] Muffy Calder [a] and Alice Miller[a]

[a] *Department of Computing Science, University of Glasgow, Glasgow, UK*

Abstract

We present a technique to tackle the parameterised probabilistic model checking problem for a particular class of randomised distributed systems, which we model as Markov Decision Processes. These systems, termed *degenerative*, have the property that a model of a system with some communication graph will eventually behave like a model of a system with a *reduced* graph. We describe an induction schema for reasoning about models of a degenerative system over arbitrary graphs. We thereby show that a certain class of quantitative LTL properties will hold for a model of a system with any communication graph if it holds for all models of a system with some base graph. We demonstrate our technique via a case study (a randomised leader election protocol) specified using the PRISM modelling language.

Keywords: Probabilistic model checking, parameterised model checking, degenerative systems, PRISM.

1 Introduction

Model checking of distributed systems is restricted to verifying systems with a fixed number of processes. Proving a property for a system with N identical processes, for any $N > 0$, is known as the *parameterised model checking problem* (PMCP). This problem is undecidable in general [2] but techniques can be used to solve it for certain types of system.

Probabilistic model checking augments traditional model checking, enabling *quantitative* as well as *qualitative* analysis. Probabilistic model checking has become an important area of research due to the increased use of probabilistic algorithms and the requirement for analysis of not just system correctness but also system performance. Probabilistic model checkers, such as PRISM [14], enable properties such as *"the system will fail with probability less than 0.01"* and *"with probability 1, the system will terminate"* to be verified. Probabilistic model checking tools vary in the

[1] Supported by a University of Glasgow scholarship

[2] Email:doug@dcs.gla.ac.uk

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

type of underlying model that they support. We focus on probabilistic model checking of randomised distributed systems, models of which exhibit both probabilistic and non-deterministic choice, therefore we restrict our attention to reasoning over MDPs.

In this paper we tackle the PMCP for randomised distributed systems by extending an inductive proof for a non-probabilistic parameterised distributed system [16]. We generalise this proof for a class of probabilistic systems, described as *degenerative* – they have the property that a system configuration of a given size eventually behaves like a smaller configuration. The proof employs induction over the topology of the system in order to show that any property in a class of properties that holds for a model of a base system topology will hold for a model of a system of any size and configuration. The induction relies on determining that any behaviour of a model of the system of a given size is equivalent to a behaviour in a model of a smaller system. To illustrate our technique we consider a family of models of the IEEE 1394 Firewire tree identify protocol [11] specified using PRISM.

2 Background

2.1 Markov Decision Processes

In the sequel, for a set Y, $Dist(Y)$ denotes the set of all discrete probability distributions over Y i.e. the set of all functions $\mu : Y \to [0,1]$ such that $\sum_{y \in Y} \mu(y) = 1$.

We model randomised distributed systems as Markov Decision Processes (MDPs). In particular, we consider state-labelled MDPs, where the states are augmented with a set of (atomic) propositions true in that state.

Definition 2.1 (See, for example, [18]). A (labelled) Markov Decision Process is a tuple $\mathcal{M} = (S, s_0, Steps, Act, L)$ where S is a finite set of states, $s_0 \in S$ is the initial state, Act is a set of actions, $Steps : S \to 2^{\mathrm{Act} \times \mathrm{Dist}(S)}$ is the probabilistic transition function such that, $\forall s \in S, Steps(s) \neq \emptyset$ and $L : S \to 2^{AP}$ is a labelling function over a set of propositions AP.

For an MDP, $\mathcal{M} = (S, s_0, Steps, Act, L)$, the function $Steps$ maps each state in S to a non-empty subset of $Act \times Dist(S)$. Intuitively, for $s \in S$, $Steps$ makes a non-deterministic choice over $|Steps(s)|$ action, distribution pairs, choosing action a and distribution μ, say. A probabilistic choice is made over S where the probability of moving to a state s' is given by $\mu(s')$. We say that a is *enabled* from s. If $\mu(s') > 0$ for some state s' we say there is a transition from s to s', written $s \xrightarrow{a,\mu} s'$. Action $a \in Act$ is *non-probabilistic* iff, $\forall s \in S$, $\forall (a, \mu) \in Steps(s)$, $\mu(s') = 1$ for some $s' \in S$ and is a *stutter* action iff, $\forall s \in S$, $\forall (a, \mu) \in Steps(s)$, $\mu(s') > 0 \implies L(s) = L(s')$. An infinite *path*, α in $\mathcal{M}$ is a non-empty sequence $s_0 \xrightarrow{a_0,\mu_0} s_1 \xrightarrow{a_1,\mu_1} \ldots$ where for $i \geq 0$, $s_i \in S$, $(a_i, \mu_i) \in Steps(s_i)$, $\mu(s_{i+1}) > 0$. Similarly, a finite path is a non-empty sequence, $s_0 \xrightarrow{a_0,\mu_0} s_1 \xrightarrow{a_1,\mu_1} \ldots \xrightarrow{a_{n-1},\mu_{n-1}} s_n$ for some $n \geq 0$. For a finite or infinite path, α, $|\alpha|$ denotes the length (the number of actions) of the path (with $|\alpha| = \infty$ for an infinite path), and $tr^{AP}(\alpha)$ the sequence given by the labelling of the states in α restricted to the set of propositions in AP. For a finite path, $\alpha = s_0 \xrightarrow{a_0,\mu_0} s_1 \xrightarrow{a_1,\mu_1} \ldots \xrightarrow{a_{n-1},\mu_{n-1}} s_n$, let $last(\alpha) = s_n$ and $\mathbf{P}(\alpha) = \mu_0(s_1).\mu_1(s_2)\ldots\mu_{n-1}(s_n)$

(with $\mathbf{P}(\alpha) = 1$ if $\alpha = s_0$). For two paths, α and α' with α finite, if α is a prefix of α' we write $\alpha \leq \alpha'$ (and $\alpha < \alpha'$ if it is a strict prefix). The set of all infinite paths starting at state s is given by $Path(s)$ and the set of all finite paths starting at s by $Path_{fin}(s)$.

2.2 Adversaries

In order to analyse an MDP we need to resolve the non-determinism. This is done by considering *adversaries*, constructs that make a choice over $Steps(s)$ for each state s of an MDP, based on the history of choices made up to state s. Formally, an adversary A of an MDP $\mathcal{M} = (S, s_0, Steps, Act, L)$ maps every finite path α of $\mathcal{M}$ onto an element $A(\alpha)$ of the set $Steps(last\ (\alpha))$ [19]. An adversary produces an infinite-state Markov chain, with each state given by the history of states so far visited. An adversary uniquely determines a Markov chain of this form, so in the sequel it will be convenient to refer to an adversary of an MDP when describing the Markov chain induced by it. Also, $Adv_{\mathcal{M}}$ denotes the set of adversaries for MDP $\mathcal{M}$ and, for adversary A and state s, $Path^A(s)$ denotes the subset of $Path_s$ which corresponds to A and similarly, $Path^A_{fin}(s)$, the subset of $Path_{fin}(s)$ that corresponds to A [19]. For path $\alpha \in Path^A_{fin}(s)$, define the *path cylinder*, $\mathcal{C}(\alpha) = \{\omega \in Path^A(s) | \alpha \leq \omega\}$. The probability measure, $Prob^A_s$, is defined on the smallest σ-algebra that contains all the sets $\mathcal{C}(\alpha)$ for all $\alpha \in Path^A_{fin}(s)$, such that, $Prob^A_s(\mathcal{C}(\alpha)) = \mathbf{P}(\alpha)$ (for more detail see, for example, [13]).

2.3 Cuts

Definition 2.2 Let $\mathcal{M}$ =(S,s_0,$Steps$,Act,L) be an MDP and let $A \in Adv_{\mathcal{M}}$. Define $Cut(A)$ to be a family of sets s.t. for $D \in Cut(A)$, $D \subseteq Path^A_{fin}(s_0)$ where, for all $\alpha \in D$, $\alpha \not\leq \alpha'$ and $\alpha' \not\leq \alpha$ for any $\alpha' \in D$, $\alpha' \neq \alpha$ and $\sum_{\alpha \in D} Prob^A_{s_0}(\mathcal{C}(\alpha)) = 1$.

Intuitively, a *cut* (a simplification of a *fringe* as defined for probabilistic automata by Segala [20]) represents a finite portion of the Markov chain induced by an adversary. Given an adversary A of an MDP, for $n \geq 0$, let $cut^A(n) \in Cut(A)$ be defined such that for all $\alpha \in cut^A(n)$, $|\alpha| = n$. For $C \in Cut(A)$ we say that C is a *cut* of A. Furthermore, we describe $cut^A(n)$ as a cut of A *at depth* n.

2.4 Quantitative Linear Time Logic

To specify properties of MDPs we employ Linear Time Logic (LTL). LTL formulae are defined in terms of paths of an MDP and have a formal syntax $\phi ::$ true $\mid a \mid \neg\phi_1 \mid \phi_1 \wedge \phi_2 \mid \phi_1 \mathcal{U} \phi_2 \mid \mathcal{X}\phi_1$ where a is an atomic proposition, and $\mathcal{U}$ and $\mathcal{X}$ are the standard until and next-time operators. See for example, [7] for a full description. $\text{LTL}_{\setminus\mathcal{X}}$ is defined as for LTL but without the next-time operator (the exclusion of this operator is not a great hardship since one seldom reasons about exactly the next state in a distributed algorithm).

A *quantitative* LTL (QLTL) formula, is defined over states of an MDP with syntax $\phi ::= \mathcal{P}_{\bowtie p}[\psi]$, where $\bowtie \in \{\leq, <, >, \geq\}, p \in [0, 1]$ and ψ is a LTL path formula (similarly for $\text{QLTL}_{\setminus\mathcal{X}}$). For MDP, $\mathcal{M}$, state s of $\mathcal{M}$, adversary A of $\mathcal{M}$ and LTL

path formula ψ, by abuse of notation, in the sequel, we let $Prob_s^A(\psi) = Prob_s^A(\{\alpha \in Path^A(s) | \alpha \models \psi\})$. For QLTL property, $\phi \equiv \mathcal{P}_{\bowtie p}[\psi]$, s satisfies ϕ, denoted $s \models \phi$, iff, $\forall\, A \in Adv_{\mathcal{M}}$, $Prob_s^A(\psi) \bowtie p$. $\mathcal{M}$ satisfies ϕ, ($\mathcal{M} \models \phi$) iff $s_0 \models \phi$ where s_0 is the initial state of $\mathcal{M}$.

2.5 Stuttering equivalence

For any string v, the stuttering removal operator $\#$ applied to v replaces every maximal finite subsequence of identical elements by a single copy of this element. Let $\mathcal{M}$ and $\mathcal{M}'$ be MDPs with propositions AP and AP' respectively. A path α of $\mathcal{M}$ is said to be *stuttering equivalent* to a path α' in $\mathcal{M}'$ (denoted $\alpha \simeq \alpha'$) with respect to $AP'' \subseteq AP \cap AP'$ if and only if $\#tr^{AP''}(\gamma) = \#tr^{AP''}(\beta)$. We extend stuttering equivalence of paths to adversaries by considering *trace cylinders* over sequences of sets of atomic propositions. Our definitions are based on those given in [5].

Definition 2.3 Let AP be a set of propositions. The trace cylinder $\mathcal{C}(l_0^+, l_1^+, \ldots, l_n^+)$ (for $l_0, l_1, \ldots, l_n \in 2^{AP}$ pairwise distinct, $n \geq 0$) is defined by $\mathcal{C}(l_0^+, l_1^+, \ldots, l_n^+) = \{t \in (2^{AP})^\omega) | t = l_0^{k_0}, l_1^{k_1}, \ldots, l_n^{k_n}, \ldots \text{ for some } k_0, k_1, \ldots, k_n \geq 1\}$ where, for $k \geq 1$, $l^k = \underbrace{l, l, \ldots, l}_{k}$ for $l \in 2^{AP}$.

For an adversary A of an MDP $\mathcal{M}$ with initial state s_0, and set of propositions AP, by abuse of notation in the sequel let $Prob_{s_0}^A(\mathcal{C}(l_0^+, l_1^+, \ldots, l_n^+)) = Prob_{s_0}^A(\{\alpha \in Path^A(s_0) | tr^{AP}(\alpha) \in \mathcal{C}(l_0^+, l_1^+, \ldots, l_n^+)\})$.

Definition 2.4 Given two MDPs, $\mathcal{M} = (S, s_0, Steps, Act, L)$ and $\mathcal{M}' = (S', s_0', Steps', Act', L')$, with propositions AP and AP' respectively, two adversaries $A \in Adv_{\mathcal{M}}$, $A' \in Adv_{\mathcal{M}'}$ are *probabilistic stuttering equivalent* (denoted $A \simeq A'$) w.r.t. $AP'' \subseteq AP \cap AP'$ if and only if, $Prob_{s_0}^A(\mathcal{C}(l_0^+, l_1^+, \ldots, l_n^+)) = Prob_{s_0'}^{A'}(\mathcal{C}(l_0^+, l_1^+, \ldots, l_n^+))$ for all pairwise disjoint $l_0, l_1, \ldots, l_n \in 2^{AP''}, n \geq 0$.

For convenience, and consistency with [5], we henceforth use the shorthand *stuttering equivalence* for *probabilistic stuttering equivalence* when it is clear that we are referring to equivalence between adversaries.

Let S, T be sets, $R \subseteq S \times T$ and $\mu \in Dist(S)$, $\nu \in Dist(T)$. A weight function for μ and ν with respect to R is a function $w : S \times T \to [0, 1]$ such that $w(s, t) > 0 \Rightarrow sRt$, $\mu(s) = \sum_{t \in T} w(s, t)$ for any $s \in S$ and $\nu(t) = \sum_{s \in S} w(s, t)$ for any $t \in T$. We write $\mu \sqsubseteq_R \nu$ iff there is a weight function for μ and ν with respect to R.

We now give conditions on a pair of adversaries that allow us to show stuttering equivalence without considering trace cylinders and examining only finite paths. The proof of Lemma 2.5 is given in [20] for a more general case.

Lemma 2.5 *Let $\mathcal{M} = (S, s_0, Steps, Act, L)$ and $\mathcal{M}' = (S', s_0', Steps', Act', L')$ be MDPs with sets of propositions AP and AP' respectively. Let $AP'' \subseteq AP \cap AP'$. Let A and A' be adversaries of $\mathcal{M}$ and $\mathcal{M}'$ respectively. $A \simeq A'$ if there exists cuts $D_0, D_1, \ldots$ with, $\forall i \geq 0 D_i \in Cut(A')$, such that*

(i) *$\forall i \geq 0$, $\forall \alpha \in D_{i+1}$, $\alpha \in D_i$ or $\alpha = \beta.a, \mu, s$ and $\beta \in D_i$,*

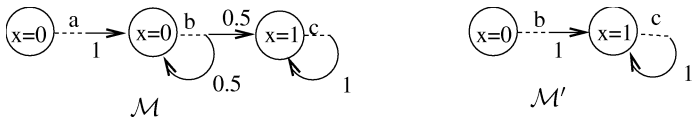

Fig. 1. Two MDPs, $\mathcal{M}$ and $\mathcal{M}'$, with stuttering equivalent adversaries

(ii) *For every* $\alpha \in Path^A_{fin}(s_0)$, $\lim_{i\to\infty} \sum_{\beta\in D_i, \alpha\leq\beta} \mathbf{P}(\beta) = Prob^A_{s_0}(\mathcal{C}(\alpha))$,

(iii) *For each* $i \geq 0$, *define* $\mu_i : cut^A_i \to [0,1]$, $\mu'_i : D_i \to [0,1]$ *such that for* $\alpha \in cut^A$, $\alpha' \in D_i$, $\mu_i(\alpha) = \mathbf{P}(\alpha)$, $\mu'_i(\alpha') = \mathbf{P}(\alpha')$. *Then* $\mu_i \sqsubseteq_R \mu'_i$ *where for* $\alpha \in cut^A_i$, $\alpha' \in D_i$, $R(\alpha,\alpha')$ *iff* $\alpha \simeq \alpha'$ *w.r.t.* AP''.

$\mathrm{LTL}_{\setminus\mathcal{X}}$ properties induce stutter-invariant measurable languages [22] and so, by standard arguments of measure theory, it follows that

Lemma 2.6 *If* $\mathcal{M}$ *and* $\mathcal{M}'$ *are MDPs with propositions* AP *and* AP' *and adversaries* A *and* A', *respectively, then for any* $\mathrm{LTL}_{\setminus\mathcal{X}}$ *path formula* ψ *with propositions in* $AP'' \subseteq AP \cap AP'$, *if* $A \simeq A'$ *w.r.t.* AP'' *then* $Prob^A_{s_0}(\psi) = Prob^{A'}_{s'_0}(\psi)$.

Example 2.7 In Figure 1 we give an example of MDPs, $\mathcal{M}$ and $\mathcal{M}'$, both over set of propositions $AP = \{x = 0, x = 1\}$, with initial states s_0 and s'_0 respectively and action sets $\{a, b, c\}$ and $\{b, c\}$ respectively. There is only one adversary associated with each MDP: let these be A and A', then $A \simeq A'$ w.r.t. AP since,

$$Prob^A_{s_0}(\mathcal{C}(\{x=0\}^+)) = Prob^{A'}_{s'_0}(\mathcal{C}(\{x=0\}^+)) = 1,$$

$$Prob^A_{s_0}(\mathcal{C}(\{x=0\}^+,\{x=1\}^+)) = Prob^{A'}_{s'_0}(\mathcal{C}(\{x=0\}^+,\{x=1\}^+)) = 1$$

and the probability measure over all other trace-cylinders is zero. If ψ is $\mathrm{LTL}_{\setminus\mathcal{X}}$ property, $(true\ \mathcal{U}\ (x=1))$, $Prob^A_{s_0}(\psi) = Prob^{A'}_{s'_0}(\psi) = 1$. Thus, $\mathcal{M}$ and $\mathcal{M}'$ satisfy $\mathcal{P}_{\geq 1}[\psi]$.

2.6 Isomorphism between Adversaries

Isomorphic adversaries must have exactly the same structural behaviour (up to labelling of states). Definition 2.8 and Lemma 2.9 are adapted from [9].

Definition 2.8 Let $\mathcal{M} = (S, s_0, Steps, Act, L)$ and $\mathcal{M}' = (S', s'_0, Steps', Act', L')$ be MDPs with adversaries A and A' respectively. Let $\rho : Path^A_{fin}(s_0) \to Path^{A'}_{fin}(s'_0)$ be a bijection with $\rho(s_0) = s'_0$. Suppose, for all $\alpha \in Path^A_{fin}(s_0)$, if $A(\alpha) = (a, \mu)$ and $\rho(\alpha \xrightarrow{(a,\mu)} t) = \alpha' \xrightarrow{(a',\mu')} t'$ then $\mu(t) = \mu'(t')$ for all t s.t. $\mu(t) > 0$. Then ς is an isomorphism from A to A', and A and A' are isomorphic (denoted $A = A'$).

Lemma 2.9 *Let* $\mathcal{M} = (S, s_0, Steps, Act, L)$ *and* $\mathcal{M}' = (S', s'_0, Steps', Act', L')$ *be MDPs with propositions* AP *and* AP', *respectively and let* $\Sigma : AP \to AP'$ *be a bijection. For LTL property* ψ *with propositions in* AP, $\Sigma(\psi)$ *is the LTL formula obtained from* ψ *by replacing every proposition* a *with* $\Sigma(a)$. *Let* ς *be an isomorphism between adversaries* A *(of* $\mathcal{M}$*) and* A' *(of* $\mathcal{M}'$*) such that, for all* $\alpha \in Path^A_{fin}(s_0)$ *and* $a \in AP$, $a \in L(last(\alpha)) \iff \Sigma(a) \in L'(last(\varsigma(\alpha)))$. *Then for any LTL formula* ψ *with propositions from* AP, $Prob^A_{s_0}(\psi) = Prob^{A'}_{s'_0}(\Sigma(\psi))$.

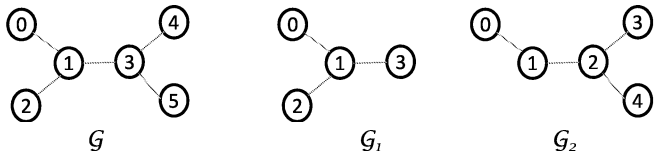

Fig. 2. Communication Topology Γ for Example 3.2

Note that in Example 2.7, A and A' are not isomorphic.

2.7 Graphs

We define a *graph*, $\mathcal{G} = (E, V, I)$, to be a tuple with V a set of vertices, E, a set of edges between pairs of vertices and I, a labelling of vertices with each vertex $v \in V$ uniquely labelled by a value $I(v) \in \{0, 1, \ldots, |\mathcal{G}| - 1\}$ (where $|\mathcal{G}| = |V|$ is the size of the graph). By abuse of notation, i denotes the vertex v with $I(v) = i$. Given a permutation, σ on $\{0, 1, \ldots, |\mathcal{G}| - 1\}$, we define the permuted graph under σ as $\sigma(\mathcal{G}) = (E, V, I')$ where $I'(v) = \sigma(I(v))$ and describe σ as a permutation on $\mathcal{G}$. For a graph, $\mathcal{G} = (E, V, I)$ and $V' \subseteq V$, $\mathcal{G}[V'] = (E', V', I')$ is the subgraph induced by V' obtained by deleting the vertices in $V \setminus V'$ and the associated edges from $\mathcal{G}$.

3 Parameterised Model Checking for Randomised Degenerative Systems

3.1 Communication Graphs and Reductions

In the sequel we use the term *communication graph* to describe a vertex-labelled, non-empty, finite, simple, connected graph (by abuse of notation, we refer to a communication graph simply as a graph). Also, for a communication graph $\mathcal{G}$ we refer to vertex v, with $I(v) = i$ as *process* i and describe i as a process *index*. If there is an edge (v, w) of $\mathcal{G}$, with $I(v) = i$, $I(w) = j$, we say process i and process j *communicate*. A set of communication graphs is defined as a *communication topology* (or simply a topology).

Informally a system is degenerative if it eventually behaves as a 'smaller' system. We formalise the notion of 'smaller' in terms of the topology of the system and define a set of 'least' elements of a topology as follows.

Definition 3.1 Let Γ be a topology and $\mathcal{G} = (E, V, I) \in \Gamma$. Let σ be a permutation of $\mathcal{G}$ and let $W \subset V$. Then $\mathcal{R} = (W, \sigma)$ is a reduction of $\mathcal{G}$ in Γ iff the graph $\mathcal{R}(\mathcal{G}) = \sigma(\mathcal{G})[W]$ belongs to Γ. We describe $\mathcal{R}(\mathcal{G})$ as the reduced communication graph of $\mathcal{G}$ in Γ under $\mathcal{R}$ or simply a reduced communication graph of $\mathcal{G}$.

Example 3.2 Consider the topology Γ consisting of graphs $\mathcal{G}$, $\mathcal{G}_1$ and $\mathcal{G}_2$, illustrated in Figure 2. Define sets W_1 and W_2 thus: $W_1 = \{0, 1, 2, 3\}$ and $W_2 = \{0, 1, 2, 3, 4\}$, and let permutations σ_1 and σ_2 be the identity permutation and a permutation that fixes 0 and 1 and maps 3 to 2, 4 to 3 and 5 to 4 respectively. Then, if $R_1 = (W_1, \sigma_1)$ and $R_2 = (W_2, \sigma_2)$, $R_1(\mathcal{G}) = \mathcal{G}_1$ and $R_2(\mathcal{G}) = \mathcal{G}_2$. Hence R_1 and R_2 are reductions of $\mathcal{G}$ in Γ.

Definition 3.3 Let Φ and Γ be topologies such that $\Phi \subset \Gamma$ and let $\mathcal{Q}_\Gamma = \{Q_\mathcal{G} | \mathcal{G} \in \Gamma\}$ be a family of sets of reductions for communication graphs in Γ such that for all $\mathcal{G} \in \Phi$, $Q_\mathcal{G} = \emptyset$. Then Γ is reducible to Φ under $\mathcal{Q}_\Gamma$ iff, for all $\mathcal{G} \in \Gamma \setminus \Phi$, there exists a sequence of reductions, $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n$ (for some $n \geq 1$) such that, for all $1 \leq i \leq n$, $\mathcal{R}_i \in Q_{\mathcal{R}_{i-1}(\mathcal{R}_{i-2}(\ldots(\mathcal{R}_1(\mathcal{G}))))}$ and $\mathcal{R}_n(\mathcal{R}_{n-1}(\ldots(\mathcal{R}_1(\mathcal{G}))\ldots)) \in \Phi$.

3.2 Specifying Sets of Models Over a Communication Topology

We consider MDPs defined with respect to some variable set. For a communication graph $\mathcal{G}$, we consider variable sets over $\mathcal{G}$, $X_\mathcal{G} = \cup_{i=0}^{N-1} X_\mathcal{G}^i \cup G_\mathcal{G} \cup C_\mathcal{G}$ where, for $0 \leq i \leq N-1$, each $X_\mathcal{G}^i$ is a set of *local* variables associated with process i. These are the same (up to indexing) for each process. The set $G_\mathcal{G}$ are the *global* variables that are common to all processes. The channel variables, $C_\mathcal{G} = \{c_{j,k} | j \text{ and } k \text{ communicate}\}$, are used to send messages between a pair of processes. For $x \in X_\mathcal{G}$, $D(x)$ denotes the domain of x and $D(X_\mathcal{G})$ the cross-product of the domains of the variables in $X_\mathcal{G}$. We assume that $D(c) = D(c')$ for all $c, c' \in C_\mathcal{G}$. We define the set of propositions over $X_\mathcal{G}$ as, $AP_\mathcal{G} = \{x = d | x \in X_\mathcal{G}, d \in D(x)\}$.

In the sequel, we distinguish between *indexed* and *unindexed* variables in $X_\mathcal{G}$. A variable is indexed if it is subscripted with a process index (all local and channel variables are indexed), or if its domain is the set of process indices plus the unassigned value, $\perp$ (otherwise it is unindexed). The *elected* variable in the example described in Section 4 is an indexed variable. For the same example, a local variable *mymsg* (say) storing the most recent message received by a given process would have domain $\{\perp, bmp, bmc, ack\}$ and would therefore be unindexed.

We can extend this definition to the set of propositions $AP_\mathcal{G}$ over $X_\mathcal{G}$. A proposition $x = d$ ($x \in X_\mathcal{G}$, $d \in D(x)$) is indexed if x is indexed and $d \neq \perp$ (otherwise it is unindexed). A LTL or QLTL property is unindexed if it contains only unindexed propositions.

We also assume for an MDP that there is a set of actions over a graph $\mathcal{G}$, $Act_\mathcal{G} = \cup_{i=0}^{N-1} Act_i$, such that each action is defined with respect to a process and that the sets of 'local' actions, Act_i, are isomorphic (up to process indexing).

Definition 3.4 Let $\mathcal{G} = (E, V, I)$ be a communication graph, $X_\mathcal{G}$ a variable set over $\mathcal{G}$ and $Act_\mathcal{G}$ an action set over $\mathcal{G}$. If the initial value of the variables in $X_\mathcal{G}$ is given by the tuple $init(X_\mathcal{G})$ then a model over $\mathcal{G}$ is an MDP, $\mathcal{M}_\mathcal{G} = (D(X_\mathcal{G}), init(X_\mathcal{G}), Steps_\mathcal{G}, Act_\mathcal{G}, L_\mathcal{G})$ such that $L_\mathcal{G}$ labels states with the set of propositions AP where AP is defined over $X_\mathcal{G}$. Given a topology, Γ, let $\mathcal{M}_\Gamma = \{\mathcal{M}_\mathcal{G} | \mathcal{G} \in \Gamma\}$ denote a set of models over Γ.

3.3 Mappings Induced by the Permutation of a Communication Graph

Given graph $\mathcal{G}$ and a permutation of $\mathcal{G}$, σ, let $\mathcal{M}_\mathcal{G}$ be a model over $\mathcal{G}$ and $\mathcal{M}_{\sigma(\mathcal{G})}$ be a model over $\sigma(\mathcal{G})$. For adversaries A of $\mathcal{M}_\mathcal{G}$ and A' of $\mathcal{M}_{\sigma(\mathcal{G})}$ we define the *index map on A induced by σ*, $\rho : Path_{fin}^A(s_0^\mathcal{G}) \rightarrow Path_{fin}^{A'}(s_0^{\sigma(\mathcal{G})})$ that maps the process indices associated with any indexed variables and any actions, according to σ. Similarly, we can define the *propositional index map induced by σ*, Σ, between the propositions AP over variable set $X_\mathcal{G}$ and AP' over $X_{\sigma(\mathcal{G})}$. Since Σ respects ρ, from Lemma 2.9

we can show that, for an unindexed LTL property ψ with propositions in AP, if ρ is an isomorphism then, $Prob^{A}_{s_0}(\psi) = Prob^{A'}_{s'_0}(\psi)$.

3.4 Degenerative Families of Models

We now turn to our main definition that gives conditions for a family of models (over a topology) to be *degenerative*. The key condition is that the communication graphs of the topology are reduced such that every adversary of a model over some graph is stuttering equivalent to an adversary of a model over a reduced graph.

Definition 3.5 Let Γ be a topology that is reducible to Φ under a family of sets of reductions, $\mathcal{Q}_\Gamma = \{Q_\mathcal{G} | \mathcal{G} \in \Gamma\}$. Suppose $\mathcal{M}_\Gamma = \{\mathcal{M}_\mathcal{G} | \mathcal{G} \in \Gamma\}$ is a set of models over Γ. For each $\mathcal{G} \in \Gamma$ let $X_\mathcal{G}$ be a set of variables over $\mathcal{G}$ and let $AP_\mathcal{G}$ be the propositions over $X_\mathcal{G}$. For each $\mathcal{R} \in Q_\mathcal{G}$, define a set of variables $X'_{\mathcal{R}(\mathcal{G})} \subseteq X_{\mathcal{R}(\mathcal{G})}$ (with $AP'_{\mathcal{R}(\mathcal{G})} \subseteq AP_{\mathcal{R}(\mathcal{G})}$, the set of propositions over $X'_{\mathcal{R}(\mathcal{G})}$). $\mathcal{M}_\Gamma$ is degenerative with base Φ under $\mathcal{Q}_\Gamma$ iff,

(i) (**Reduced Variables and Actions:**) For $\mathcal{G} \in \Gamma$ and $\mathcal{R} = (W, \sigma) \in Q_\mathcal{G}$,

$$X_{\sigma(\mathcal{G})} \setminus C_\mathcal{G} = X_\mathcal{G} \setminus C_\mathcal{G}, D(X_{\sigma(\mathcal{G})}) = D(X_\mathcal{G}), Act_{\sigma(\mathcal{G})} = Act_\mathcal{G},$$
$$X_{\mathcal{R}(\mathcal{G})} \subseteq X_{\sigma(\mathcal{G})}, D(X_{\mathcal{R}(\mathcal{G})}) \subseteq D(X_{\sigma(\mathcal{G})}), Act_{\mathcal{R}(\mathcal{G})} \subseteq Act_{\sigma(\mathcal{G})},$$

(ii) (**Matching Adversaries:**) For $\mathcal{G} \in \Gamma \setminus \Phi$, there exists $\mathcal{R} = (W, \sigma) \in Q_\mathcal{G}$ such that, for every adversary A of $\mathcal{M}_\mathcal{G}$, there exists an adversary A' of $\mathcal{M}_{\sigma(\mathcal{G})}$ that is isomorphic to A under the index map induced by σ, with A' stuttering equivalent to some adversary A'' of $\mathcal{M}_{\mathcal{R}(\mathcal{G})}$ with respect to $AP'_{\mathcal{R}(\mathcal{G})}$.

The establishment of a set of models, parameterised by a topology, that is degenerative provides an inductive basis (over the topology) with which to establish properties of the models.

Theorem 3.6 *Let Γ be a topology that is reducible to Φ under the family of sets of reductions, $\mathcal{Q}_\Gamma$ and let $\mathcal{M}_\Gamma$ be a set of models over Γ. Suppose, for each $\mathcal{G} \in \Gamma$, $\mathcal{R} \in Q_\mathcal{G}$, there is a set of variables $X'_{\mathcal{R}(\mathcal{G})} \subseteq X_{\mathcal{R}(\mathcal{G})}$ (with $AP'_{\mathcal{R}(\mathcal{G})} \subseteq AP_{\mathcal{R}(\mathcal{G})}$, the set of propositions over $X'_{\mathcal{R}(\mathcal{G})}$) such that $\mathcal{M}_\Gamma$ is degenerative with base Φ under $\mathcal{Q}_\Gamma$. Then for any unindexed QLTL$_{\setminus\mathcal{X}}$ property ϕ with propositions in $\bigcap_{\mathcal{G}\in\Gamma\setminus\Phi} \bigcap_{\mathcal{R}\in Q_\mathcal{G}} AP'_{\mathcal{R}(\mathcal{G})}$, if $\mathcal{M}_\mathcal{F} \models \phi$ for all $\mathcal{F} \in \Phi$, $\mathcal{M}_\mathcal{G} \models \phi$ for all $\mathcal{G} \in \Gamma$.*

Proof. Let $\mathcal{G} \in \Gamma$ and suppose ϕ is an unindexed QLTL$_{\setminus\mathcal{X}}$ property with propositions in $\bigcap_{\mathcal{R}\in Q_\mathcal{G}} AP'_{\mathcal{R}(\mathcal{G})}$. Assume $\mathcal{M}_{\mathcal{R}(\mathcal{G})} \models \phi$, for every $\mathcal{R} \in Q_\mathcal{G}$. We can show $\mathcal{M}_\mathcal{G} \models \phi$, as follows. Let $A \in Adv_{\mathcal{M}_\mathcal{G}}$. Choose $\mathcal{R} = (W, \sigma) \in Q_\mathcal{G}$ such that A is isomorphic to some $A' \in Adv_{\mathcal{M}_{\sigma(\mathcal{G})}}$ under ρ, the index map on A induced by σ, with A' stuttering equivalent to some $A'' \in Adv_{\mathcal{M}_{\mathcal{R}(\mathcal{G})}}$ w.r.t. $AP'_{\mathcal{R}(\mathcal{G})}$. Property ϕ has the form $\mathcal{P}_{\bowtie p}[\psi]$. Let Σ be the proposition index map induced by σ. For every adversary B of $\mathcal{M}_{\mathcal{R}(\mathcal{G})}$, $Prob^{B}_{s''_0}(\psi) \bowtie p$. If $\mathcal{M}_\mathcal{G}$, $\mathcal{M}_{\sigma(\mathcal{G})}$ and $\mathcal{M}_{\mathcal{R}(\mathcal{G})}$ have initial states s_0, s'_0 and s''_0 respectively then,

$Prob^{A}_{s_0}(\psi) = Prob^{A'}_{s'_0}(\psi)$ from Section 3.3 since $A = A'$ under ρ

$= Prob^{A''}_{s''_0}(\psi)$ since $A' \simeq A''$ w.r.t. $AP'_{\mathcal{R}(\mathcal{G})}$

$\bowtie p$ by the above.

Since the above is true for every adversary of $\mathcal{M}_{\mathcal{G}}$, $\mathcal{M}_{\mathcal{G}} \models \phi$.

Let ϕ be an unindexed $\mathrm{QLTL}_{\setminus\mathcal{X}}$ formula with propositions in $\bigcap_{\mathcal{G}\in\Gamma}\bigcap_{\mathcal{R}\in Q_{\mathcal{G}}} AP'_{\mathcal{R}(\mathcal{G})}$. Let $\mathcal{G} \in \Phi$ then, by the statement of the theorem, $\mathcal{M}_{\mathcal{G}} \models \phi$. Assume $\mathcal{G} \in \Gamma \setminus \Phi$. ϕ is defined over $\bigcap_{\mathcal{R}\in Q_{\mathcal{G}}} AP'_{\mathcal{R}(\mathcal{G})}$ and is unindexed, so by the above, $\mathcal{M}_{\mathcal{G}} \models \phi$ if $\mathcal{M}_{\mathcal{R}(\mathcal{G})} \models \phi$ for all $\mathcal{R} \in Q_{\mathcal{G}}$. For each $\mathcal{R} \in Q_{\mathcal{G}}$, either $\mathcal{R}(\mathcal{G}) \in \Phi$ or it can be reduced further. Since Γ is reducible to Φ under $\mathcal{Q}_\Gamma$, continuing in this way, we can construct a tree of graphs in which every terminal node is a graph in Φ. Finally, by statement of the theorem, each of the models associated with the graphs at these terminal nodes satisfy ϕ and, by propagation up the tree of graphs, it follows that $\mathcal{M}_{\mathcal{G}} \models \phi$. □

4 Model Checking the IEEE 1394 (Firewire) Tree Identify Protocol

We illustrate our technique with a case study. The IEEE 1394 (Firewire) Tree Identify Protocol (TIP) [11], is designed to elect a leader from a set of processes arranged in an acyclic topology. A process may send one of three messages to a neighbouring process: *be_my_parent* (*bmp*), *be_my_child* (*bmc*) or *acknowledge* (*ack*). Any process that has received *bmp* messages from all or all but one of its neighbours responds with *bmc* messages and, if necessary, sends a *bmp* to the remaining neighbour. The neighbouring processes will send an *ack* upon receiving a *bmc*, from which point the processes play no further part in the protocol (and hence the protocol is degenerative). In this manner the protocol builds a spanning tree with the root process elected as leader.

It is possible for two neighbouring processes to attempt to become leader by sending *bmp* requests to each other simultaneously. In order to resolve this contention, each process probabilistically chooses to wait for a long or short amount of time, before attempting to send a request again. If a process then receives a request before it has sent one, it will be elected leader. Otherwise, another contention situation ensues and the "back-off" procedure must be repeated. Much work has been done on proving correctness of root contention in the TIP [21]. Appealing to these results, in earlier work [6] we modelled the TIP with non-deterministic contention resolution. Here we consider a family of MDP models for the TIP in which contention is resolved probabilistically. We model contention with a contending process (the one with the smallest index) making a simple probabilistic choice: with probability $\frac{1}{4}$, the process loses and the other process sends its *bmp*; with probability $\frac{1}{4}$, the process wins and transmits its request to the other process; or with probability $\frac{1}{2}$ contention is not resolved and the process must choose again.

We have modelled the TIP and verified a suite of properties for all configurations of systems with three, four and five processes, using PRISM. For reasons of space we do not give our PRISM specifications or all of our properties here. We concentrate on one property which we refer to throughout the rest of the paper. Here *elected* is a global variable (see the subsequent section) that is initially equal to $\perp$ and is set to the value of the index of any process that is elected leader.

Property 1. A leader will almost surely be elected: $\mathcal{P}_{\geq 1}[\mathit{true}\,\mathbf{U}\,\neg(\mathit{elected} = \perp)]$.

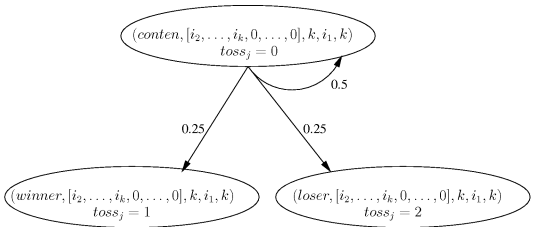

Fig. 3. Transition in $\mathcal{M}_\mathcal{G}$ corresponding to contention resolution between processes j and i_1 ($j < i_1$).

Table 1
Transitions in $\mathcal{M}_\mathcal{G}$ made by process j when it receives requests from all of its neighbours

1. Process j receives *bmp* from all its neighbours.	$(start, [\bot, \ldots, \bot], k, \bot, 0), [bmp]_{i_1,j}, \ldots, [bmp]_{i_k,j}$ $1 \downarrow a^j$ $(child, [i_1, i_2, \ldots, i_k, \bot, \ldots, \bot], k, \bot, k), []_{i_1,j}, \ldots, []_{i_k,j}$
2. Process j responds to its neighbours with *bmc* requests.	$(child, [i_1, i_2, \ldots, i_k, \bot, \ldots, \bot], k, \bot, k), []_{i_1,j}, \ldots, []_{i_k,j}$ $1 \downarrow b^j$ $(parent, [i_1, i_2, \ldots, i_k, \bot, \ldots, \bot], k, \bot, k), [bmc]_{i_1,j}, \ldots, [bmc]_{i_k,j}$
3. Process j receives *ack* from all its neighbours and becomes leader.	$(parent, [i_1, i_2, \ldots, i_k, \bot, \ldots, \bot], k, \bot, k), [ack]_{i_1,j}, \ldots, [ack]_{i_k,j}$ $1 \downarrow c^j$ $(finish, [\bot, \ldots, \bot], k, \bot, 0), []_{1,0}, \ldots, []_{N-1,0}$ $elected = j$

4.1 A Family of Models of the TIP over Acyclic Communication Graphs

Using the PRISM specifications for small configurations of the TIP as a basis, we have defined a script for automatically generating PRISM specifications of the TIP for any topology. We can view this script as specifying a family of models for the TIP system, $\mathcal{M}_\Gamma = \{\mathcal{M}_\mathcal{G} | \mathcal{G} \in \Gamma\}$ over the topology Γ, the set of communication graphs that are acyclic. Given $\mathcal{G} \in \Gamma$, with $|\mathcal{G}| = N$, model $\mathcal{M}_\mathcal{G}$ over $\mathcal{G}$ has variable set $X_\mathcal{G}$ over $\mathcal{G}$ with, for $i \in \{0, \ldots, N-1\}$,

$$G_\mathcal{G} = \{elected, toss_0, toss_1, \ldots, toss_{N-1}\}, C_\mathcal{G} = \{c_{g,h}, c_{h,g} | (g,h) \in E\},$$
$$X^i_\mathcal{G} = \{state_i, child_{i,0}, child_{i,1}, \ldots, child_{i,N-1}, adj_i, remaining\_partner_i, no\_of\_requests_i\},$$

The variable domains are, for $i, j \in \{0, 1, \ldots, N-1\}, c_{g,h} \in C_\mathcal{G}$,

$$D(state_i) = \{start, child, parent, conten, response, complete, winner, loser, b\_child, finish\}, D(no\_of\_requests_i) = D(adj_i) = \{0, 1, \ldots, N-1\},$$
$$D(remaining\_partner_i) = D(child_{i,j}) = D(elected) = \{\bot, 0, 1, \ldots, N-1\},$$
$$D(c_{g,h}) = \{empty, bmp, bmc, ack\}, D(toss_i) = \{0, 1, 2\}.$$

The set of actions over $\mathcal{G}$ is given by $Act_\mathcal{G} = \cup_{i=0}^{N-1} Act^i_\mathcal{G}$. A sample of the non-probabilistic actions in $Act^i_\mathcal{G}$ are shown in Tables 1 and 2 (for reasons of space we do not provide them all). The sole probabilistic action that a process can make, that of resolving contention, is shown in Figure 3. The conditions for an action to occur and the result of that action are given in each as the value of the local variables of process j along with some of the channel and global variables. The local variables are presented as a tuple, $(s, [ch_0, \ldots, ch_{N-1}], a, r, n)$, representing the values of $state_j$, $child_{j,0}, \ldots, child_{j,N-1}$, adj_j, $remaining\_partner_j$, $no\_of\_requests_j$ respectively. The value of a channel variable $c_{h,i}$ is represented by $[msg]_{h,i}$ (where msg is bmp, bmc, ack) or $[]_{h,i}$ if $c_{h,i} = empty$. If a variable is not presented then its value is not considered for that action. We assume process j has k neighbours,

Table 2
Transitions in $\mathcal{M}_{\mathcal{G}}$ made by process j when it receives requests from all but one of its neighbours (i_1).

4. Process j receives *bmp* from all its neighbours except i_1.	$(start, [\bot, \ldots, \bot], k, \bot, 0), []_{i_1,j}, [bmp]_{i_2,j}, \ldots, [bmp]_{i_k,j}$ $1 \downarrow a^j$ $(child, [i_2, \ldots, i_k, \bot, \ldots, \bot], k, i_1, k), []_{i_1,j}, []_{i_2,j}, \ldots, []_{i_k,j}$
$\vdots$	$\vdots$
10. Process j receives *bmp* from i_1 and enters contention.	$(response, [i_2, \ldots, i_k, \bot, \ldots, \bot], k, i_1, k), [bmp]_{i_1,j}$ $1 \downarrow e^j$ $(conten, [i_2, \ldots, i_k, \bot, \ldots, \bot], k, i_1, k), []_{i_1,j}$
11. $i_1 < j$ and process j has won contention.	$(conten, [i_2, \ldots, i_k, \bot, \ldots, \bot], k, i_1, k), toss_j = 1$ $1 \downarrow g^j$ $(winner, [i_2, \ldots, i_k, \bot, \ldots, \bot], k, i_1, k), toss_j = 0$
$\vdots$	$\vdots$
15. Process j has lost contention, receives *ack* from i_1 and becomes leader.	$(parent, [i_1, \bot, \ldots, \bot], 1, \bot, 1), [ack]_{i_1,j}$ $1 \downarrow c^j$ $(finish, [\bot, \ldots, \bot], 1, \bot, 0), []_{i_1,j}, elected = j$

$i_1, i_2, \ldots, i_k$ and for Table 2 and Figure 3, that i_1 is the neighbour that does not initially send a *bmp* to j.

5 Parameterised Model Checking of the TIP

We demonstrate our proof technique for tackling the PMCP for probabilistic degenerative systems, describing how it can be applied to the TIP and showing that,

Theorem 5.1 *Let ϕ be Property 1. Then $\mathcal{M}_{\mathcal{G}} \models \phi$ for all $\mathcal{G} \in \Gamma$.*

We tackle Theorem 5.1 by showing that $\mathcal{M}_{\mathcal{G}}$ is degenerative with base $\Phi \subset \Gamma$, the set of stars, under some family of sets of reductions. We do so by considering each of the conditions given in Definition 3.5 in turn, having defined an appropriate set of reductions and corresponding 'reduced' set of variables for each model in $\mathcal{M}_{\Gamma}$. First, however, we show that *Property 1* is satisfied by all models in $\mathcal{M}_{\Phi}$. Note that in order to show all models in $\mathcal{M}_{\Phi}$ satisfy *Property 1*, we have to tackle another instance of the PMCP. However, the topology Φ is regular, making the problem easier. We prove Lemma 5.2 by showing that $\mathcal{M}_{\Phi}$ is degenerative with base $\{\mathcal{M}_{\mathcal{G}(2)}\}$ where $\mathcal{G}(2)$ is the star with two vertices. The proof, omitted, is a simpler version of that for Theorem 5.1.

Lemma 5.2 *For all $\mathcal{G} \in \Phi$, $\mathcal{M}_{\mathcal{G}} \models \phi$, where ϕ is Property 1.*

5.1 Clipping Reductions

The main decision in defining a set of reductions is how vertices are removed from communication graphs. In the TIP example we define *clipping* reductions where we remove sets of leaf vertices that are connected to a particular kind of non-leaf vertex. These non-leaf vertices, termed *level-1* vertices, are the set of vertices for which all but one of the neighbouring vertices are leaves (the non-leaf neighbour is the *inner vertex* of j). They are guaranteed to exist in acyclic communication graphs that are not stars [16]. Given a level-1 vertex j of a graph $\mathcal{G}$, $clip^j(\mathcal{G})$ is the set of vertices of $\mathcal{G}$ excluding leaf vertices of j. We also have to identify a permutation of the vertex labels that ensures the reduced graph is labelled correctly. In the sequel, we let $\sigma_j^{\mathcal{G}}$ be a permutation (for $\mathcal{G} \in \Gamma$ and level-1 vertex j) on the vertex labels of $\mathcal{G}$, which

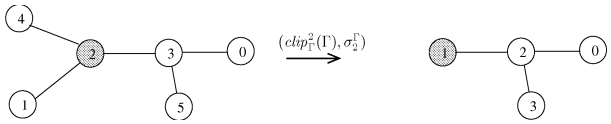

Fig. 4. An example of graph $\mathcal{G}$ (left) and the graph $\sigma^{\mathcal{G}}_2(\mathcal{G})[clip^2(\mathcal{G})]$ (right) obtained under a clipping reduction, with respect to level-1 vertex, vertex 2.

permutes the indices such that the leaves of j have the largest indices and the order of the indices of the remaining vertices is preserved (see [16] for a formal definition). We now define a set of reductions on a graph $\mathcal{G} \in \Gamma$.

Definition 5.3 For $\mathcal{G} \in \Gamma$, let $J_{\mathcal{G}} = \{j_1, j_2, \ldots, j_m\}$ be the set of all level-1 vertices in $\mathcal{G}$. Let $Clip^j_{\mathcal{G}} = (clip^j(\mathcal{G}), \sigma^{\mathcal{G}}_j)$ and define the set of clipping reductions of $\mathcal{G}$ as $Clip_{\mathcal{G}} = \{Clip^j_{\mathcal{G}} | j \in J^{\mathcal{G}}\}$. Furthermore, define the family of sets of clipping reductions, $Clip_\Gamma = \{Clip_{\mathcal{G}} | \mathcal{G} \in \Gamma\}$.

An example of a graph obtained under a clipping reduction for the level-1 vertex, vertex 2, is shown in Figure 4. We now show that we can reduce any communication graph in Γ to a star under a sequence of clipping reductions. The proof, by induction over the number of level-1 vertices, is omitted.

Lemma 5.4 *The topology, Γ, is reducible to the set of stars, Φ under $Clip_\Gamma$.*

5.2 Reduced variable Sets

We define a subset of the variable set of a model of a clipping reduced graph. Let $\mathcal{G} \in \Gamma$ and let j be a level-1 vertex. For variable set, $X_{\mathcal{G}}$ and a clipping reduction, $Clip^j_{\mathcal{G}} = (clip^j(\mathcal{G}), \sigma^{\mathcal{G}}_j)$ (with $j' = \sigma^{\mathcal{G}}_j(j)$) we remove variables associated with the leaf processes of j. Specifically, we define $X^{j'}_{Clip^j_{\mathcal{G}}(\mathcal{G})}$ to be equivalent to $X_{Clip^j_{\mathcal{G}}(\mathcal{G})}$ but excluding the variables $child_{j',0}$, $child_{j',1}$, ..., $child_{j',N-1}$, $adj_{j'}$, $no\_of\_requests_{j'}$. In the sequel $AP^{j'}_{Clip^j_{\mathcal{G}}(\mathcal{G})}$ is the set of propositions over $X^{j'}_{Clip^j_{\mathcal{G}}(\mathcal{G})}$.

5.3 Matching Adversaries

To show that $\mathcal{M}_\Gamma$, with clipping reductions, is degenerative we demonstrate that the conditions of Definition 3.5 are fulfilled. Here we establish condition (ii).

We partition the adversaries of a model $\mathcal{M}_{\mathcal{G}}$ over $\mathcal{G} \in \Gamma$ according to their behaviour in terms of the level-1 vertices. Specifically, we classify them according to which level-1 vertex receives *bmp* requests from all its leaf vertices, but not its inner vertex, first. If, under $A \in Adv_{\mathcal{M}_{\mathcal{G}}}$, j is such a vertex then A is *first-full* with respect to j. The leaf neighbours of j are then guaranteed to terminate without being elected leader and their effect under the adversary can be ignored (this is key to showing that $\mathcal{M}_\Gamma$ is degenerative). In the sequel, $Adv^j_{\mathcal{M}_{\mathcal{G}}} \subseteq Adv_{\mathcal{M}_{\mathcal{G}}}$ denotes the set of adversaries that are first-full with respect to j. The proof of Lemma 5.5 is as for the proof given in [16] for the non-probabilistic case. Intuitively, at the initialisation of the protocol only leaf processes can progress beyond their starting state. Thus a state must be reached where a level-1 process receives *bmp* requests from all of its leaf neighbours but not its inner vertex. The adversary corresponding

Table 3
Result of applying $\Sigma_j^{\mathcal{G}}$ to a proposition, a, for $0 \leq h, k \leq N$ and $0 \leq i \leq N-1$ (σ abbreviates $\sigma_j^{\mathcal{G}}$).

a	$\Sigma_j^{\mathcal{G}}(a)$
$position_h = x$	$position_{\sigma(h)} = x$
$child_{h,i} = x$	$child_{\sigma(h),i} = \sigma(x) + 1$
$adj_h = x$	$adj_{\sigma(h)} = x$
$remaining\_partner_h = x$	$remaining\_partner_{\sigma(h)} = \sigma(x)$
$no\_of\_requests_h = x$	$no\_of\_requests_{\sigma(h)} = x$
$elected_h = x$	$elected_{\sigma(h)} = \sigma(x)$
$toss_h = x$	$toss_{\sigma(h)} = x$
$c_{h,k} = x$	$c_{\sigma(h),\sigma(k)} = x$

to this scheduling must therefore be first-full with respect to a level-1 process.

Lemma 5.5 *For $\mathcal{G} \in \Gamma \setminus \Phi$, let $J^{\mathcal{G}} = \{j_1, j_2, \ldots, j_k\}$, be the set of level-1 vertices. Then, $\bigcup_{j \in J^{\mathcal{G}}} Adv^j_{\mathcal{M}_{\mathcal{G}}} = Adv_{\mathcal{M}_{\mathcal{G}}}$.*

Let $\mathcal{G} \in \Gamma \setminus \Phi$ and for level-1 vertex j let $(clip^j(\mathcal{G}), \sigma_j^{\mathcal{G}})$ be a clipping reduction. $\Sigma_j^{\mathcal{G}}$, the proposition index map induced by $\sigma_j^{\mathcal{G}}$ is shown in Figure 3 (by abuse of notation we let $\sigma_j^{\mathcal{G}}(\bot) = \bot$). In Lemma 5.6 we show every adversary A_j of $\mathcal{M}_{\mathcal{G}}$ (first-full with respect to j) is isomorphic to an adversary of $\mathcal{M}_{\sigma_j^{\mathcal{G}}(\mathcal{G})}$ under the index map induced by $\sigma^j_{\mathcal{G}}$. The proof (omitted) is by considering transitions under A_j.

Lemma 5.6 *Let $\mathcal{G} \in \Gamma \setminus \Phi$. Let j be a level-1 vertex and let $Clip^j_{\mathcal{G}} = (clip^j(\mathcal{G}), \sigma^j_{\mathcal{G}})$ be the clipping reduction for j. For every adversary A_j of $\mathcal{M}_{\mathcal{G}}$ that is first-full w.r.t. j, there exists an adversary $A_{j'}$ of $\mathcal{M}_{\sigma^j_{\mathcal{G}}(\mathcal{G})}$ that is first-full w.r.t. $j' = \sigma_j^{\mathcal{G}}(j)$ such that $\rho_j^{\mathcal{G}}$, the index map induced by $\sigma^j_{\mathcal{G}}$, is an isomorphism between A_j and $A_{j'}$.*

In Lemma 5.7 we show that, for every adversary of the model of a permuted graph, first full with respect to j, say, there exists a stuttering equivalent adversary of the model of the clipping reduced graph.

Lemma 5.7 *Let $\mathcal{G} \in \Gamma \setminus \Phi$, j a level-1 vertex and $Clip^j_{\mathcal{G}} = (clip^j(\mathcal{G}), \sigma^j_{\mathcal{G}})$ be the clipping reduction for j. For every adversary A of $\mathcal{M}_{\sigma_j^{\mathcal{G}}(\mathcal{G})}$ that is first-full w.r.t. j there exists an adversary A' of $\mathcal{M}_{Clip^j_{\mathcal{G}}(\mathcal{G})}$ s.t. $A \simeq A'$ w.r.t. $AP^j_{Clip^j_{\mathcal{G}}(\mathcal{G})}$.*

Proof. (Sketch) Let $\mathcal{M}_{\sigma_j^{\mathcal{G}}(\mathcal{G})} = (S, s_0, Steps, Act, L)$ and $\mathcal{M}_{Clip^j_{\mathcal{G}}(\mathcal{G})} = (S', s'_0, Steps', Act', L')$. Let A be an adversary of $\mathcal{M}_{\sigma_j^{\mathcal{G}}(\mathcal{G})}$, $AP* = AP^{j'}_{\sigma_j^{\mathcal{G}}(\mathcal{G})}[clip^j(\mathcal{G})]$ and $H \subseteq Path_{fin}(s_0) \times Path_{fin}(s'_0)$ be the relation given by $\forall \alpha \in Path_{fin}(s_0)$, $\alpha' \in Path_{fin}(s'_0)$, $H(\alpha, \alpha')$ iff $tr^{AP*}(\alpha) \simeq tr^{AP*}(\alpha')$. We define an adversary, A' of $\mathcal{M}_{Clip^j_{\mathcal{G}}(\mathcal{G})}$ and sets, D_0, D_1, D_2, ... s.t. $\forall n \geq 0$, $D_n \subseteq Path_{fin}(s'_0)$, by induction over the cuts of A at depth i. We show $\forall n \geq 0$,

IH1 For every $\alpha \in cut^A_{s_0}$, $\alpha' \in D_n$, if $H(\alpha, \alpha')$ then for every $m < n$ there exists prefixes $\beta \leq \alpha$ and $\beta' \leq \alpha'$ such that $\beta \in cut^A(m)$, $\beta' \in D_m$ and $H(\beta, \beta')$.

IH2 If μ_n, μ'_n are the distributions over $cut^n(A)$ and D_n, respectively, defined by, for $\alpha \in cut^A(n)$, $\alpha' \in D_n$, $\mu_n(\alpha) = \mathbf{P}(\alpha)$ and $\mu'_n(\alpha') = \mathbf{P}(\alpha')$ then $\mu_n \sqsubseteq_H \mu'_n$.

IH3 For every $\alpha \in cut^A_{s_0}$, $\alpha' \in D_n$, if $H(\alpha, \alpha')$ then for every $\beta \in cut^A_{s_0}$, $\beta' \in D_n$ such that $\beta \neq \alpha$ and $\beta' \neq \alpha'$, $(\beta, \alpha') \notin H$ and $(\alpha, \beta') \notin H$.

Base case: Clearly $cut^A(0) = \{s_0\}$. Let $D_0 = \{s'_0\}$. Immediately, IH1 and IH3 hold. By definition $\mathbf{P}(s_0) = \mathbf{P}(s'_0) = 1$. Therefore, $\mu_0 \sqsubseteq_H \mu'_0$ and so IH2 holds.

Induction step: Assume IH1, IH2 and IH3 hold for some $n \geq 0$. Suppose $\alpha \in cut^A(n+1)$. Then for $\gamma \in Path^A_{fin}(s_0)$, $(a, \mu) \in Steps(last(\gamma))$, $\alpha = \gamma \xrightarrow{a,\mu} s$. Since $|\gamma| = n$, $\gamma \in cut^A(n)$ and since $\mu_n \sqsubseteq_H \mu'_n$ by IH2, there exists $\gamma' \in D_n$ such that $H(\gamma, \gamma')$ and by IH3 no other path is related to γ' or γ. We now define D_{n+1} by considering transition $last(\gamma) \xrightarrow{a,\mu} s$. We need to consider four cases (we consider just one here).

Case (i): Let $leaf(j)$ denote the leaf vertices of level-1 vertex j. Suppose $a \in \cup_{i \in leaf(j)} Act_i$. Notice that for process j to send a *bmp* request to one of its leaves (k say) it must have received a *bmp* request from its inner vertex and all its other leaves. This would imply, however, that A is not first-full with respect to j. Therefore, process j cannot send a *bmp* request to any of its leaves and so none of the leaves can reach a contention state with j. Thus, we only need consider non-probabilistic stutter actions w.r.t. $AP*$ i.e. for which $\mu(s) = 1$ and $L(last(\gamma)) \cap AP* = L(s) \cap AP*$. Thus, since we also have that γ is stuttering equivalent to γ' w.r.t. $AP*$, α and γ' are stuttering equivalent w.r.t. $AP*$. Let $\alpha' = \alpha$.

We let D_{n+1} be the set of finite paths, $\{\alpha' | \alpha \in cut^A(n)$, and α' is derived from α as described above$\}$ and extend A' by these paths. By the definition of this set, IH1, IH2 and IH3 are satisfied. We can show that the conditions of Lemma 2.5 are satisfied by A and A' and it follows that $A \simeq A'$ w.r.t. $AP^j_{Clip^j_{\mathcal{G}}(\mathcal{G})}$. □

5.4 Proof of Theorem 5.1

Proof. From Lemma 5.4, Γ is reducible to Φ under the clipping reductions. Condition (i) of Definition 3.5 follows by definition of the action sets, variable sets and variable domains for $\mathcal{M}_\Gamma$. From Lemmas 5.5, 5.6 and 5.7, it follows that condition (ii) of Definition 3.5 is satisfied for $\mathcal{M}_\Gamma$. Thus, $\mathcal{M}_\Gamma$ is deterministically degenerative with base Φ under $Clip_\Gamma$. By Lemma 5.2, *Property 1* holds for all models over stars. Since *Property 1* is unindexed with appropriately defined propositions, by Theorem 3.6, it is satisfied by $\mathcal{M}_\mathcal{G}$ for all $\mathcal{G} \in \Gamma$. □

6 Related Work

Certain classes of probabilistic systems have been verified for arbitrary number of processes [17] e.g. Arons et al. [3] present two methods for verifying liveness properties with probability 1 over parameterised probabilistic systems by converting the probabilistic system to an 'equivalent' non-deterministic one. Duflot et. al. [10] consider the convergence of self-stabilising randomised protocols for a ring topology. They show that given a non-increasing measure on the state space of the model, if there exists a 'distance' measure between states and an ordering relation on the distance metric that satisfies certain conditions then it can be deduced that the protocol will converge to some *legitimate* set of states with probability 1. The methods described above have only been applied to verification of *qualitative* properties i.e. properties that hold with probability 0 or 1. Parameterised model checking of *quantitative* properties has not been widely addressed, although some manual proofs of

quantitative properties have been devised. For example, Aspnes and Herlihy [4], by appealing to results from random walk theory, give a lower bound for the probability of all processes returning heads in a weak shared coin protocol.

Much work has been carried out on analysing the TIP (see for example, [15]). We mention [1] since it describes an inductive proof for a protocol that is similar to the TIP. The authors observe that only a leaf can initially transmit an "up" (*bmp*) message and it will then move to a "dead" state after which the protocol behaves as if started in the graph with that leaf deleted. They note that, continuing in this manner, eventually a graph with only one or two vertices will be reached. The protocol is not specified formally, whereas we use state-based verification. Our work extends that described above as it allows us to formally reason about *quantitative* properties over parameterised systems.

7 Conclusion and Future Work

We have described an inductive proof technique for a class of randomised distributed systems (modelled as MDPs) described as *degenerative*. The technique is an induction schema over the underlying communication topology, represented by a set of graphs. The key idea is that topologies are *reduced* such that every adversary of a model of a system over some graph is stuttering equivalent to an adversary of a model of a system over a reduced graph. Reduction involves the removal of one or more vertices from the communication graph. The base case(s) are those graphs that are not reduced. We applied this technique to the IEEE 1394 (Firewire) tree identify protocol showing that a class of $\mathrm{QLTL}_{\setminus\mathcal{X}}$ properties that are true of the systems with a star topology will hold for a system with *any* acyclic topology. In this case, reduction is by removal of the leaf vertices of level-1 vertices.

Our technique is only applicable to degenerative protocols. These are, however, widespread in distributed systems e.g gossip-style multicast protocols such as [8], the weak shared coin protocol of Aspnes and Herlihy [4], the Itai Rodeh leader election protocol for rings [12]. These systems present further challenges because the protocols degenerate *probabilistically* (whereas the TIP degenerates *deterministically*). This necessitates extending our induction schema. This is work in progress.

References

[1] Angluin, D., *Local and global properties in networks of processors (extended abstract)*, in: *Proc. ToC'80* (1980), pp. 82–93.

[2] Apt, K. R. and D. C. Kozen, *Limits for automatic verification of finite-state concurrent systems*, Information Processing Letters **22** (1986), pp. 307–309.

[3] Arons, T., A. Pnueli and L. D. Zuck, *Parameterized verification by probabilistic abstraction.*, in: *FoSSaCS'03*, LNCS **2620** (2003), pp. 87–102.

[4] Aspnes, J. and M. Herlihy, *Fast randomized consensus using shared memory*, Journal of Algorithms **11** (1990), pp. 441–461.

[5] Baier, M., C. Grösser and M. a. Ciesinski, *Partial order reduction for probabilistic systems*, in: *Proceedings of the 1st International Conference on quantitative and qualitative evaluation of systems (QEST'04)* (2004), pp. 230–239.

[6] Calder, M. and A. Miller, *Using SPIN to analyse the tree identification phase of the IEEE 1394 high performance serial bus (FireWire) protocol*, in: Maharaj et al. [15], pp. 247–266.

[7] Clarke, E. M., O. Grumberg and D. Peled, "Model Checking," The MIT Press, Cambridge, Masachusetts, 1999.

[8] Demers, A., D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart and D. Terry, *Epidemic algorithms for replicated database maintenance*, in: *Proc. PoDC'87* (1987), pp. 1–12.

[9] Donaldson, A. F. and A. Miller, *Symmetry reduction for probabilistic model checking using generic representatives.*, in: *ATVA 2006*, 2006, pp. 9–23.

[10] Duflot, M., L. Fribourg and C. Picaronny, *Randomized finite-state distributed algorithms as Markov chains*, in: J. L. Welch, editor, *Distributed algorithms*, LNCS **2180**, 2001, pp. 240–254.

[11] IEEE 1394-1995, "IEEE Standard for a High Performance Serial Bus Std 1394-1995," Institute of Electrical and Electronic Engineers (1995).

[12] Itai, A. and M. Rodeh, *Symmetry breaking in distributed networks*, Information and Computation **88** (1990), pp. 60–87.

[13] Kemeny, J. G., J. L. Snell and A. W. Knapp, "Denumerable Markov Chains," Graduate Texts in Mathematics **40**, Springer-Verlag, New York, 1976, second edition.

[14] Kwiatkowska, M., G. Norman and D. Parker, *Probabilistic symbolic model checking with PRISM: A hybrid approach*, International Journal on Software Tools for Technology Transfer (STTT) **6** (2004), pp. 128–142.

[15] Maharaj, S., J. Romijn and C. Shankland, editors, "Formal specification of the IEEE 1394 Tree identify protocol," Formal Aspects of Computing **14(3)**, Springer-Verlag, 2003.

[16] Miller, A. and M. Calder, *Two verification results for networks of arbitrary size*, Technical Report TR2006-220, University of Glasgow,Department of Computing Science (2006).

[17] Norman, G., *Analyzing randomized distributed algorithms*, in: *Validation of Stochastic Systems: A Guide to Current Research*, LNCS (Tutorial Volume) **2925** (2004), pp. 384–418.

[18] Puterman, M. L., "Markov Decision Processes: Discrete Stochastic Dynamic Programming," John Wiley and Sons, New York, 1994, first edition.

[19] Rutten, J., M. Kwiatkowska, G. Norman and D. Parker, "Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems," CRM Monograph Series **23**, American Mathematical Society, Providence, Rhode Island, 2004.

[20] Segala, R., "Modeling and Verification of Randomized Distributed Real-Time Systems," Ph.D. thesis, MIT, Dept. of Electrical Engineering and Computer Science (1995).

[21] Stoelinga, M., *Fun with firewire: A comparative study of formal verification methods applied to the IEEE 1394 root contention protocol.*, Formal Asp. Comput. **14** (2003), pp. 328–337.

[22] Vardi, M. Y., *Automatic verification of probabilistic concurrent finite-state programs*, in: *Proceedings of FoCs'85*, 1985, pp. 327–338.