Stochastic Restricted Broadcast Process Theory

Fatemeh Ghassemi[1,2], Mahmoud Talebi[1], Ali Movaghar[1,2], and Wan Fokkink[3]

[1] Sharif University of Technology, Tehran, Iran
[2] Institute for studies in Theoretical Physics and Mathematics, Tehran, Iran
[3] VU University Amsterdam, The Netherlands

Abstract. We provide a framework for modeling and analyzing both qualitative and quantitative aspects of mobile ad hoc network (MANET) protocols above the data-link layer. We extend Restricted Broadcast Process Theory [11,9]: delay functions are assigned to actions, while the semantics captures the interplay of a MANET protocol with stochastic behavior of the data-link and physical layer, and the dynamic topology. A continuous-time Markov chain is derived from our semantic model by resolving non-determinism, using the notion of weak Markovian network bisimilarity. The framework is applied to a leader election algorithm.

Keywords: Ad hoc protocol, Cross layer performance evaluation, Stochastic process algebra, Markovian model.

1 Introduction

In mobile ad hoc networks (MANETs), nodes communicate with each other using wireless transceivers which are unreliable. Nodes move arbitrarily and the topology of the network changes *dynamically*. MANET protocols should be able to tolerate faults that may arise due to unreliable wireless communication and changes in the underlying topology, while quality of service metrics in benchmarks should be satisfied. Therefore a unified framework for the verification and evaluation of MANET protocols can alleviate the complexity in the design process of such protocols.

We introduced *Restricted Broadcast Process Theory* (*RBPT*) in [11,9] to specify and verify MANET protocols, taking into account mobility. Here we extend this framework to *Stochastic RBPT* (*SRBPT*), to evaluate properties of MANET protocols above the data-link layer. Performance evaluation of MANET protocols depends on physical characteristics of the nodes, the underlying dynamic topology of the network, the protocol behavior itself, and its collaboration with data-link layer protocols. The physical characteristics of nodes and their underlying topology define whether two nodes can communicate, while data-link layer protocols define how fast nodes can communicate.

To study the cross-layer performance of protocols above the data-link layer, an abstract model of the MAC protocol is used. The MAC protocol at the data-link layer manages transmissions of a node to reduce their collisions with other possible ones occurring in the vicinity. This abstract model may specify the aggregated behavior of the MAC protocols of an arbitrary number of nodes in terms of some delay functions like [19] or the behavior of a single MAC protocol from the point view of upper-layer protocols as a queue with a limited capacity of K and service time equal to MAC response

© Springer-Verlag Berlin Heidelberg 2011

time T_{Mac} like [24]. To provide a compositional framework for evaluating MANETs, we choose the latter approach and then unfold the behavior of a MANET protocol deployed at a node, at the semantic level in terms of its interaction with its underlying MAC protocol [12]. To enhance the non-compositional framework of [12] with modular specification and analysis of MANETs, we provide a process algebra with more expressive power in defining the timing behavior of protocols. This process algebra follows the lines of *RBPT* [10] in syntax and the model of [12] in semantics.

The effect of physical and the dynamic underlying topology can be captured through the probability that a node receives a message successfully (P_{rcv}), and the probability that a communication link exists between two nodes (P_{UP}) with an identical mobility behavior. To remedy the effect of mobility on MAC layer performance factors like response time, we assume networks of nodes with an identical MAC layer and mobility model. Therefore, *SRBPT* semantic model is parameterized by mentioned four parameters: K, T_{Mac}, P_{rcv} and P_{UP}. We capture the interplay of all these parameters in the operational semantics. Due to the existence of internal immediate actions (with zero delay), probabilistic and non-deterministic choice coexist in the semantics. Non-deterministic choices of a model can be resolved by means of our weak Markovian network bisimilarity, a congruence relation on labeled multi-transition systems, while the performance measures of the model are preserved. We explain when and how a Markov chain can be derived from the specification of a MANET.

With this unified framework, one can first verify the correctness of a MANET specified in *RBPT*. Then one can specify appropriate action delays and semantics parameters to evaluate the quality of service metrics of a MANET protocol deployed on a specific data-link layer in a dynamic network. We illustrate the applicability of our framework by the analysis of a leader election protocol for MANETs [27].

Related works. PEPA, a stochastic process algebra, has been exploited to investigate the performance of backoff algorithms in ad hoc protocols [25] and the performance of WiFi protocol in configurations that the IEEE 802.11i does not guarantee the fairness for channel access [20]. PRISM have been used in the verification of MANET protocols like the Bluetooth device discovery [5], the MAC protocol of IEEE 802.11 [22], the IEEE 1394 root contention protocol [4], and the ZeroConf dynamic configuration protocol for IPv4 link-local addresses [21]. These protocols (except ZeroConf) mainly belong to the data-link layer, so the effect of the data-link layer and the dynamic topology are not considered. The wireless communication modeled in these works is either point-to-point or unreliable, while collisions that occur during a transmission are modeled and the underlying topology is considered fixed. However, modeling collisions (which are handled by MAC protocols) or point-to-point communication is not appealing for performance evaluation of MANET protocols (above the data-link layer). On the other hand, *non-blocking* and *topology-dependent* behaviors intrinsic in wireless communication, called local broadcast, cannot be modeled by *CSP*-like parallel composition in PEPA and PRISM in a modular way [7]. The only framework tailored to performance evaluation of wireless protocols (which can be used for MANETs) is [8]. That framework is more suitable for protocols beneath the data-link layer, since it only considers the physical characteristics of nodes and their underlying topology (which is static), and not the effect of the data-link layer.

2 Evaluation Factors

As explained in Sect. 1, performance evaluation of protocols (above the data-link layer) depends on data-link and physical layer protocols and the underlying dynamic topology of the network. These factors are taken into account in our semantic model in [12]:

- (T_{Mac}, K) specifying the abstract data-link layer model: when a protocol has data to be transmitted, it delivers its message to its underlying data-link layer. The message is inserted in the queue of the node's data-link layer if the queue is not full, and messages in this queue are transmitted with an average delay equal to the average response time T_{Mac} of the data-link layer, i.e., the average time a message spends in a data-link layer queue, waiting to be transmitted or being transmitted.
- P_{UP} abstracting the dynamic underlying topology: we assume nodes move under an identical mobility model. Therefore in the steady-state, the probability that a link exists between two arbitrary nodes can be computed as explained in [23] for a mobility model (see also [12]). Generally speaking, the higher P_{UP}, the more successful communication.
- P_{rcv} abstracting physical layer protocols: a node located in the transmission range of a sender will receive its messages successfully with a probability P_{rcv}. In [8], this probability is computed by taking distance, signal strength and interference of other nodes into account, while [28] incorporates channel and radio parameters such as the path loss exponent and shadowing variance of the channel, and modulation and encoding of the radio. So this parameter provides an abstraction from physical characteristics of nodes, physical layer protocols and noise from the environment.

It should be noted that when the data-link layer of a node like ℓ broadcasts, it communicates to ℓ' with probability $P_{rcv} \times P_{UP}$ if there is a communication link between them (the link is UP) and ℓ' successfully receives. Otherwise either despite the readiness of ℓ' to receive, there is no communication link between them (the link was DOWN), or the link was UP but ℓ' received noisy data. Therefore ℓ does not communicate to ℓ' with probability $1 - P_{rcv} \times P_{UP}$. If ℓ' was not ready to receive, ℓ does not communicate to ℓ' with probability 1. We encode these concerns in the semantics: to each transition a probability is assigned (which is multiplied with the rate of the transition).

3 Stochastic RBPT

Network protocols (in particular MANET protocols) rely on data. To separate the manipulation of data from processes, we make use of abstract data types [6]. Data is specified by equational specifications: one can declare data types (so-called *sorts*) and functions working upon these data types, and describe the meaning of these functions by equational axioms. Following of μCRL [15], *Stochastic Restricted Broadcast Process Theory* (*SRBPT*) is provided with equational abstract data types. The semantics of the data part (of a specification), denoted by $\mathbb{D}$, is defined as in [15]. It should contain $Bool$, the booleans, with distinct T and F constants. We assume the data sorts $Bool$ and Nat, the natural numbers, with the standard operations on them.

We mention some notations used in the definition of the formal syntax of *SRBPT*. Let D denote a data sort; u, v and d range over closed and open data terms of sort D, respectively. The functions $eq : D \times D$ and $if : Bool \times D \times D$ are defined for all data sorts D. The former only returns T if its two data parameters are semantically equal. The latter returns the first argument if the boolean parameter equals T, and otherwise the second argument. $d[d_1/d_2]$ denotes substitution of d_2 by d_1 in data term d; this extends to substitution in process terms. Loc denotes a finite set of addresses, ranged over by ℓ, which represent hardware addresses of nodes. Msg denotes a set of message types communicated over a network and ranged over by m. Messages are parameterized with data; w.l.o.g. we let all messages have exactly one such parameter. $\mathcal{A}_p$ denotes a countably infinite set of protocol names which are used in recursive process specifications.

3.1 Actions: Types and Rates

Each action in *SRBPT* is a pair (α, r) where α is the action name and r the rate. A process performs two types of actions: sending and receiving a message. The rate indicates the speed at which the action occurs from the viewpoint of the data-link layer. According to their rates, actions are classified as *active* and *passive*. *Active* actions have as rate either a positive real number or $\top$. A positive real number denotes the parameter of the exponentially distributed random variable specifying the duration of the action. Such actions are called *delayable*. The $\top$ rate denotes *immediate* actions; either they are irrelevant from a performance point of view, or they have duration zero. *Passive* actions have an undefined rate, denoted by $\perp$. The duration of a passive action is fixed only by synchronizing it with an active action. Send actions are active while receive actions are passive. Restriction of the duration of active actions to exponential distributions enables us to define in *SRBPT* the classical interleaving semantics for the parallel composition operator, i.e. local broadcast, and to derive a Markov chain from the semantic model.

3.2 Syntax

The syntax of *SRBPT* is:

$$t ::= \ 0 \mid (\alpha, r).t \mid t + t \mid [b]t \diamond t \mid \textstyle\sum_{d:D} t \mid A(d),\ A(v : D) \overset{def}{=} t \mid [\![t]\!]_\ell \mid t \parallel t.$$

A process can be a deadlock, modeled by 0. A process $(\alpha, r).t$ performs action α with rate $r \in \{\top, \perp\} \cup I\!R^{>0}$ and then behaves as t. An action α can be a send $snd(m(d))$ or a receive $rcv(m(d))$. A process $t_1 + t_2$ behaves non-deterministically as t_1 or t_2. A conditional command $[b]t_1 \diamond t_2$ defines process behavior based on the data term b of sort $Bool$; if it evaluates to T in the data semantics the process behaves as t_1, and otherwise as t_2. Summation $\sum_{d:D} t$, which binds the name d in t, defines a non-deterministic choice among $t[u/d]$ for all closed $u \in D$. A process is specified by $A(d : D) \overset{def}{=} t$ where $A \in \mathcal{A}_p$ is a protocol name and d a variable name that appears free in t, meaning that it is not within the scope of a sum operator in t. We restrict to process specifications in which each occurrence of an $A(d)$ in t is within the scope of an action prefix. The simplest form of a MANET is a node, represented by the deployment operator $[\![t]\!]_\ell$; it denotes process t deployed at network address ℓ. A MANET can be composed from MANETs using $\|$; the MANETs communicate with each other by restricted broadcast.

We only consider well-defined *SRBPT* terms, meaning that processes deployed at a network address, called protocols, are defined by action prefix, choice, summation, conditional, and protocol names:

- If $t \equiv rcv(m(d)).t'$, then it is in the context of a $\sum_{d:D}$, which is in the context of a deployment operator. Furthermore, t' should be well-defined.
- If $t \equiv snd(m(d)).t_1$ or $t \equiv t_1 + t_2$ or $t \equiv [b]t_1 \diamond t_2$ or $t \equiv \sum_{d:D} t_1$, then it occurs in the scope of a deployment operator, and t_1 and t_2 are well-defined.
- If $t \equiv A(d)$, then it occurs in the the scope of a deployment operator, and $A(d : D) \overset{def}{=} t'$ where t' doesn't contain a parallel or deployment operator.
- If $t \equiv t_1 \parallel t_2$ or $t \equiv [\![t_1]\!]_\ell$, then t isn't in the scope of a deployment operator, and t_1 and t_2 are well-defined.

3.3 Execution Mechanisms and Formal Semantics

Our semantics captures the interplay between network layer protocols and the underlying topology as explained in Sect. 2 such that compositionality is maintained. The state of a MANET node is defined by the state of its deployed protocol $[\![t]\!]_\ell$ and of its data-link layer Q_i, denoted by $[\![t]\!]_\ell : Q_i$. The latter denotes the state of the queue with capacity K, containing i messages: $Q_i = m_1(u_1) \cdot \ldots \cdot m_i(u_i)$ where $i \leq K$. A state of a MANET, called configuration and denoted by $\mathcal{T}$, results from the aggregate states of its nodes. In a configuration, the data-link layers of different nodes may compete on the shared communication medium to broadcast. To provide the formal semantics for *SRBPT*, we informally examine the behavior of three operators.

Prefix. The configuration $[\![(snd(m(u)), r).0]\!]_\ell : Q_0$ specifies that it delivers message $m(u)$ after a delay, exponentially distributed with rate r, to its underlying data-link layer to be transferred to the network. The configuration $[\![0]\!]_\ell : m(u)$ is reached, in which the data-link layer transfers $m(u)$ to the network after an average delay of T_{Mac}, and the configuration $[\![0]\!]_\ell : Q_0$ is reached. The semantic model of $[\![(snd(m(u)), r).0]\!]_\ell : Q_0$ has three states with two transitions; message delivery of a protocol to its data-link layer is represented by a transition with the internal action (τ, r), while sending a message from the data-link layer to the network is represented by a transition with the label $(nsnd(m(d), \ell), r_{Mac})$ where $r_{Mac} = 1/T_{Mac}$.

Choice. In the configuration $[\![(snd(m_1(u_1)), r_1).0 + (snd(m_2(u_2)), r_2).0]\!]_\ell : Q_0$, the protocol can deliver message $m_1(u_1)$ or $m_2(u_2)$. Following [2,18], we adopt the *race policy* mechanism for choosing the delayable action to execute: the action sampling the least duration wins the competition and therefore, the choice is resolved probabilistically. Hence the transitions in Fig. 1 are achieved for this configuration. The two transitions labeled by (τ, r_i) denote interaction of the protocol with its data-link layer, executed with the probability $r_i/(r_1 + r_2)$, while the transitions labeled by $(nsnd(m_i(u_i), \ell), r_{Mac})$ denote interaction of the data-link layer with its environment.

A consequence of the adoption of the race policy is that when a delayable and an immediate action compete, the immediate action is executed, since the delayable action

$$\begin{tikzcd}
& \llbracket (snd(m_1(u_1)), r_1).0 + (snd(m_2(u_2)), r_2).0 \rrbracket_\ell : Q_0 \arrow[dl, "(\tau, r_1)"'] \arrow[dr, "(\tau, r_2)"] & \\
\llbracket 0 \rrbracket_\ell : m_1(u_1)) \arrow[dr, "(nsnd(m_1(u_1), \ell), r_{Mac})"'] & & \llbracket 0 \rrbracket_\ell : m_2(u_2) \arrow[dl, "(nsnd(m_2(u_2), \ell), r_{Mac})"] \\
& \llbracket 0 \rrbracket_\ell : Q_0 &
\end{tikzcd}$$

Fig. 1. Transitions of $\llbracket (snd(m_1(u_1)), r_1).0 + (snd(m_2(u_2)), r_2).0 \rrbracket_\ell : Q_0$

cannot sample to zero from the associated exponential distribution, $F_X(0) = Pr[X \leq 0] = 0$. In other words, immediate actions take precedence over delayable actions. So stochastically speaking, configurations have the *maximal progress* property [16]. To achieve compositionality as explained in [16], this precedence is not reflected in the semantic rules of Table 1. Instead, we introduce a notion of equality in Sect. 4.1 to take care of it. We adopt non-determinism to choose between passive actions to execute in a state. Therefore the behavior of the choice operator is probabilistic (for delayable actions), prioritized (for immediate and delayable actions) or non-deterministic (for immediate or passive actions) according to its operands.

Parallel. $M_0 \equiv \llbracket (snd(m_1(u_1)), r_1).0 \rrbracket_A : Q_0 \parallel \llbracket (snd(m_2(u_2)), r_2).0 \rrbracket_B : Q_0$ consists of two nodes with addresses A and B. The transitions are achieved using the memoryless property of exponential distribution as shown in Fig. 2. In the initial configuration, the actions $(snd(m_1(u_1)), r_1)$ and $(snd(m_2(u_2)), r_2)$ compete to execute. If we assume that $snd(m_1(u_1))$ finishes before $snd(m_2(u_1))$, then the remaining time of $snd(m_2(u_1))$ still has a distribution with rate r_2. Therefore these two actions can be interleaved. Likewise, after delivery of message $m_1(u_1)$ by node A to its data-link layer, this data-link layer and action $snd(m_2(u_1))$ compete to execute. Since the probability of having the same delay for actions $(snd(m_1(u_1)), r_1)$ and $(snd(m_2(u_1)), r_2)$ is zero, there is no transition from M_0 to M_4.

$$\begin{tikzcd}
& & M_0 \arrow[dl, "\tau, r_1"'] \arrow[dr, "\tau, r_2"] & & \\
& M_1 \arrow[dl, "\eta_1, r_{Mac}"'] \arrow[dr, "\tau, r_2"] & & M_2 \arrow[dl, "\tau, r_1"'] \arrow[dr, "\eta_2, r_{Mac}"] & \\
M_3 \arrow[dr, "\tau, r_2"'] & & M_4 \arrow[dl, "\eta_1, r_{Mac}"'] \arrow[dr, "\eta_2, r_{Mac}"] & & M_5 \arrow[dl, "\tau, r_1"] \\
& M_6 \arrow[dr, "\eta_2, r_{Mac}"'] & & M_7 \arrow[dl, "\eta_1, r_{Mac}"] & \\
& & M_8 & &
\end{tikzcd}$$

M_0	$\llbracket (snd(m_1(u_1)), r_1).0 \rrbracket_A : Q_0 \parallel \llbracket (snd(m_2(u_2)), r_2).0 \rrbracket_B : Q_0$
M_1	$\llbracket 0 \rrbracket_A : m_1(d_1) \parallel \llbracket (snd(m_2(u_2)), r_2).0 \rrbracket_B : Q_0$
M_2	$\llbracket (snd(m_1(u_1)), r_1).0 \rrbracket_A : Q_0 \parallel \llbracket 0 \rrbracket_B : m_2(d_2)$
M_3	$\llbracket 0 \rrbracket_A : Q_0 \parallel \llbracket (snd(m_2(u_2)), r_2).0 \rrbracket_B : Q_0$
M_4	$\llbracket 0 \rrbracket_A : m_1(d_1) \parallel \llbracket 0 \rrbracket_B : m_2(d_2)$
M_5	$\llbracket (snd(m_1(u_1)), r_1).0 \rrbracket_A : Q_0 \parallel \llbracket 0 \rrbracket_B : Q_0$
M_6	$\llbracket 0 \rrbracket_A : Q_0 \parallel \llbracket 0 \rrbracket_B : m_2(d_2))$
M_7	$\llbracket 0 \rrbracket_A : m_1(d_1) \parallel \llbracket 0 \rrbracket_B : Q_0$
M_8	$\llbracket 0 \rrbracket_A : Q_0 \parallel \llbracket 0 \rrbracket_B : Q_0$

$\eta_1 = nsnd(m_1(u_1), A), \quad \eta_2 = nsnd(m_2(u_2), B)$

Fig. 2. Transitions of $\llbracket (snd(m_1(u_1)), r_1).0 \rrbracket_A : Q_0 \parallel \llbracket (snd(m_2(u_2)), r_2).0 \rrbracket_B : Q_0$

Consider the closed configuration $[\![t_1]\!]_{\ell_1} : Q^1_{i_1} \parallel \ldots \parallel [\![t_n]\!]_{\ell_n} : Q^n_{i_n}$ on data-link layer (T_{Mac}, K), where $i_1, \ldots i_n \le K$, and P_{UP} is defined. Following the approach of [18], its semantics is given by a labeled multi-transition system with the multi-set transition relation results from the multi-set union of transitions $\mathcal{T} \xrightarrow{(\eta,\lambda)} \mathcal{T}'$ induced by the rules in Table 1, where $\lambda \in \{\top, r\bot\} \cup \mathbb{R}^{>0}$ ($r\bot$ is a shorthand for $r \times \bot$) and η is the unobservable action τ or a network send or receive action $nsnd(m(d), \ell)$ or $nrcv(m(d))$. Let $t \not\xrightarrow{(rcv(m(d)),-)}$ denote there is no t' such that $t \xrightarrow{(rcv(m(d)),\bot)} t'$. Conversely we also use $t \xrightarrow{(m(d)?,-)}$. Let $\mid \{\!| t^* | t \xrightarrow{(rcv(m(d)),\bot)} t^* |\!\} \mid$ denote the number of transitions that t can make by performing $(rcv(m(d)), \bot)$, where $\{\!| \ |\!\}$ denotes a multi-set. In Table 1 the symmetric counterparts of $Choice$, $Sync_2$ and Par are omitted.

Table 1. Operational semantics of MANETs

$$\frac{}{(\alpha, r).t \xrightarrow{(\alpha,r)} t} : Pre \qquad \frac{t_1 \xrightarrow{(\alpha,r)} t_1'}{t_1 + t_2 \xrightarrow{(\alpha,r)} t_1'} : Choice \qquad \frac{t[u/d] \xrightarrow{(\alpha,r)} t'}{A(u) \xrightarrow{(\alpha,r)} t'} : Inv,\ A(d:D) \stackrel{def}{=} t$$

$$\frac{t[u/d] \xrightarrow{(\alpha,r)} t'}{\sum_{d:D} t \xrightarrow{(\alpha,r)} t'} : Sum \qquad \frac{t_1 \xrightarrow{(\alpha,r)} t_1' \quad \mathbb{D} \models b = T}{[b]t_1 \diamond t_2 \xrightarrow{(\alpha,r)} t_1'} : Then \qquad \frac{t_2 \xrightarrow{(\alpha,r)} t_2' \quad \mathbb{D} \models b = F}{[b]t_1 \diamond t_2 \xrightarrow{(\alpha,r)} t_2'} : Else$$

$$\frac{t \xrightarrow{(snd(m(d)),r)} t' \quad i < K}{[\![t]\!]_\ell : Q_i \xrightarrow{(\tau,r)} [\![t']\!]_\ell : Q_i \cdot m(d)} : Snd_1 \qquad \frac{t \xrightarrow{(snd(m(d)),r)} t' \quad i = K}{[\![t]\!]_\ell : Q_i \xrightarrow{(\tau,r)} [\![t']\!]_\ell : Q_i} : Snd_2$$

$$\frac{t \xrightarrow{(rcv(m(d)),\bot)} t'}{[\![t]\!]_\ell : Q \xrightarrow{(nrcv(m(d)),r(t)\bot)} [\![t']\!]_\ell : Q} : Rcv_1 \qquad \frac{t \xrightarrow{(rcv(m(d)),-)}}{[\![t]\!]_{\ell'} : Q \xrightarrow{(nrcv(m(d)),r_{\neg s}\bot)} [\![t]\!]_{\ell'} : Q} : Rcv_2$$

$$\frac{t \not\xrightarrow{(rcv(m(d)),-)}}{[\![t]\!]_\ell \xrightarrow{(nrcv(m(d)),\bot)} [\![t]\!]_\ell} : Rcv_3 \qquad \frac{\mathcal{T}_1 \xrightarrow{(nrcv(m(d)),r_1\bot)} \mathcal{T}_1' \quad \mathcal{T}_2 \xrightarrow{(nrcv(m(d)),r_2\bot)} \mathcal{T}_2'}{\mathcal{T}_1 \parallel \mathcal{T}_2 \xrightarrow{(nrcv(m(d)),r_1 \times r_2\bot)} \mathcal{T}_1' \parallel \mathcal{T}_2'} : Sync_1$$

$$\frac{\mathcal{T}_1 \xrightarrow{(\tau,\lambda)} \mathcal{T}_1'}{\mathcal{T}_1 \parallel \mathcal{T}_2 \xrightarrow{(\tau,\lambda)} \mathcal{T}_1' \parallel \mathcal{T}_2} : Par \qquad \frac{\mathcal{T}_1 \xrightarrow{(nsnd(m(d),\ell),\lambda)} \mathcal{T}_1' \quad \mathcal{T}_2 \xrightarrow{(nrcv(m(d)),r\bot)} \mathcal{T}_2'}{\mathcal{T}_1 \parallel \mathcal{T}_2 \xrightarrow{(nsnd(m(d),\ell),r \times \lambda)} \mathcal{T}_1' \parallel \mathcal{T}_2'} : Sync_2$$

$$\frac{}{[\![t]\!]_\ell : m(d) \cdot Q \xrightarrow{(nsnd(m(d),\ell),r_{Mac})} [\![t]\!]_\ell : Q} : Bro$$

Pre, Sum, $Choice$, Inv, $Then$ and $Else$ are standard rules for basic process algebras. Interactions between protocol t and its data-link layer are specified by $Snd_{1,2}$: when a protocol t transmits a message, it is delivered to the data-link layer, which inserts it at the end of its queue if there is space, else the message is dropped. This action is considered internal from the viewpoint of the data-link layer, and consequently it cannot be synchronized with other actions, as explained by Par. After this synchronization the data-link layer transmits the message with average time T_{Mac}. The sojourn time corresponding to a configuration that ends with a local broadcast (by a node) is an exponentially distributed random variable with rate r_{Mac}. Bro explains that the data-link layer of a node transmits messages in its queue with rate r_{Mac}, while the network address of the node is appended to this message. Rcv_1 specifies that a process t can receive a message successfully if it has a link to a sender (P_{UP}) and receives the message correctly (P_{rcv}). Therefore the probability of a successful receive action is $r(t) = r_s \times r_t$, where $r_s = P_{rcv} \times P_{UP}$ and $r_t = | \{\!|t^* | t \xrightarrow{(rcv(m(d)),\perp)} t^* |\!\} |$ is the normalization factor (since a protocol can non-deterministically execute multiple receive actions, the normalization factor maintains the rate of the sojourn time of a configuration ending with a local broadcast to r_{Mac}). However, if the node does not perform the receive action which was enabled, then the node was either disconnected with probability $1 - P_{UP}$, or it could not receive successfully with probability $P_{UP} \times (1 - P_{rcv})$. Therefore a node with an enabled receive action does not receive with probability $r_{\neg s} = 1 - P_{UP} \times P_{rcv}$. This behavior is explained in Rcv_2, by making the node perform the receive action while the state of its process is unchanged. A node with no enabled receive action can be synchronized with probability 1, while the state of its process is unchanged, as explained by Rcv_3. $Sync_1$ allows to group together nodes that are ready to receive the same message. In this case, the probability of receive actions is a product of all receive coefficients. $Sync_2$ explains what happens when a node broadcasts: the rate of synchronization is the rate of broadcast multiplied by the probability of receivers.

Example. Consider MANET $[\![P(A)]\!]_A \parallel [\![Q(B)]\!]_B \parallel [\![R]\!]_C$ with $P(adr : Loc) \overset{def}{=} (snd(elec(adr)), r).0$, $Q(adr : Loc) \overset{def}{=} \sum_{lx:Loc}(rcv(elec(lx)), \perp).0 + (snd(elec(adr), r).0$ and $R \overset{def}{=} \sum_{lx:Loc}(rcv(elec(x)), \perp).0$, on a data-link layer with $(T_{Mac}, 1)$. By Snd_1:

$$[\![P(A)]\!]_A : Q_0 \parallel [\![Q(B)]\!]_B : Q_0 \parallel [\![R]\!]_C : Q_0 \xrightarrow{(\tau,r)} [\![0]\!]_A : elec(A) \parallel [\![Q(B)]\!]_B : Q_0 \parallel [\![R]\!]_C : Q_0$$

If node A broadcasts $elec(A)$ and only B receives, then by Rcv_1, Rcv_2 for nodes B, C respectively and $Sync_2$:

$$[\![0]\!]_A : elec(A) \parallel [\![Q(B)]\!]_B : Q_0 \parallel [\![R]\!]_C : Q_0 \xrightarrow{(nsnd(elec(A),A),\mathfrak{r})} [\![0]\!]_A : Q_0 \parallel [\![0]\!]_B : Q_0 \parallel [\![R]\!]_C : Q_0$$

where $\mathfrak{r} = r_{Mac} \times r_s \times r_{\neg s}$. If B broadcasts and C does not receive, then by Rcv_3, Rcv_2 for nodes A, C respectively and $Sync_2$:

$$[\![0]\!]_A : elec(A) \parallel [\![Q(B)]\!]_B : Q_0 \parallel [\![R]\!]_C : Q_0 \xrightarrow{(\tau,r)}$$
$$[\![0]\!]_A : elec(A) \parallel [\![0]\!]_B : elec(B) \parallel [\![R]\!]_C : Q_0 \xrightarrow{(nsnd(elec(B),B)r_{Mac}\times r_{\neg s})}$$
$$[\![0]\!]_A : elec(A) \parallel [\![0]\!]_B : Q_0 \parallel [\![R]\!]_C : Q_0.$$

4 From Configuration to CTMC

Our framework aims at the evaluation of MANET protocols by means of CTMCs derived from the semantic model. Immediate actions give rise to non-determinism. To obtain a CTMC, we need to eliminate immediate actions by means of an appropriate congruence relation.

4.1 Weak Markovian Network Bisimilarity

We adapt the notion of weak Markovian bisimilarity from [17] for our framework, which behaves as weak bisimilarity [14] on immediate actions and as Markovian bisimilarity [18] on delayable and passive actions. It is called *weak Markovian network bisimilarity*: delayable and passive actions are treated as in Markovian bisimilarity, but they may be preceded and followed by internal immediate actions.

Let $Q(T_{Mac}, K)$ denote the set of states of the data-link layer, a queue with capacity K and response time T_{Mac}. We write $\xrightarrow{\tau}^{*}$ for the transitive closure of $\xrightarrow{(\tau,\top)}$ transitions, and $\mathcal{T} \overset{\tau}{\not\rightarrow}$ to denote there is no $\mathcal{T}'$ such that $\mathcal{T} \xrightarrow{(\tau,\top)} \mathcal{T}'$.

The definition of a weak Markovian network bisimulation is obtained in the same manner as [16,17]: a passive/delayable action must be simulated by a matching step, possibly preceded and followed by arbitrarily many immediate internal steps. Stochastically speaking, the cumulated rate of moving by a passive/delayable action to an equivalence class should be equal for each transition and its match. Since its match may be preceded by internal transitions, for an equivalence class $\mathcal{C}_{\mathcal{R}}$ we let $\mathcal{C}_{\mathcal{R}}^{\tau}$ denote the set $\{\mathcal{T}' \mid \exists \mathcal{T} \in \mathcal{C}_{\mathcal{R}} \cdot \mathcal{T}' \xrightarrow{\tau}^{*} \mathcal{T}\}$. An internal immediate step may be simulated, but can also be mimicked by taking no transition at all, provided the equivalence classes match.

Definition 1. *An equivalence relation $\mathcal{R}$ on configurations is a weak Markovian network bisimulation if $\mathcal{T}_1 \; \mathcal{R} \; \mathcal{T}_2$ implies for all equivalence classes $C_{\mathcal{R}} \in (SRBPT \times Q(T_{Mac}, K))/\mathcal{R}$:*

- *if $\mathcal{T}_1 \overset{\tau}{\not\rightarrow}$ then $\gamma(\mathcal{T}_1, \eta, C_{\mathcal{R}}) = \gamma(\mathcal{T}_2', \eta, C_{\mathcal{R}}^{\tau})$ for some $\mathcal{T}_2'$ with $\mathcal{T}_2 \xrightarrow{\tau}^{*} \mathcal{T}_2'$, $\mathcal{T}_2' \overset{\tau}{\not\rightarrow}$;*
- *if $\mathcal{T}_2 \overset{\tau}{\not\rightarrow}$ then $\gamma(\mathcal{T}_2, \eta, C_{\mathcal{R}}) = \gamma(\mathcal{T}_1', \eta, C_{\mathcal{R}}^{\tau})$ for some $\mathcal{T}_1'$ with $\mathcal{T}_1 \xrightarrow{\tau}^{*} \mathcal{T}_1'$, $\mathcal{T}_1' \overset{\tau}{\not\rightarrow}$;*
- *if $\mathcal{T}_1 \xrightarrow{(\tau,\top)} \mathcal{T}_1'$ then for some $\mathcal{T}_2'$, $\mathcal{T}_2 \xrightarrow{\tau}^{*} \mathcal{T}_2'$, $\mathcal{T}_1'\mathcal{R}\mathcal{T}_2'$;*
- *if $\mathcal{T}_2 \xrightarrow{(\tau,\top)} \mathcal{T}_2'$ then for some $\mathcal{T}_1'$, $\mathcal{T}_1 \xrightarrow{\tau}^{*} \mathcal{T}_1'$, $\mathcal{T}_1'\mathcal{R}\mathcal{T}_2'$.*

$\gamma(\mathcal{T}, \eta, C_{\mathcal{R}}) = \sum\{\!|\lambda \mid \mathcal{T} \xrightarrow{(\eta,\lambda)} \mathcal{T}', \mathcal{T}' \in \mathcal{C}_{\mathcal{R}}|\!\}$, i.e. the summation of all elements in this multiset. Since $r_1\bot + r_2\bot = (r_1 + r_2)\bot$ and $r_1\bot = r_2\bot$ if and only if $r_1 = r_2$, γ is well-defined. Configurations $\mathcal{T}_1$ and $\mathcal{T}_2$ are weak Markovian network bisimilar, denoted $\mathcal{T}_1 \approx_m \mathcal{T}_2$, if $\mathcal{T}_1\mathcal{R}\mathcal{T}_2$ with $\mathcal{R}$ a weak Markovian network bisimulation.

Theorem 1. *Markovian network bisimilarity is an equivalence relation, and a congruence for configurations.*

See [13] for the proof.

Example. The following equivalences hold:

$$\begin{array}{ll} [\![(\alpha, r).t_1 + (snd(m(d)), \top).t_2]\!]_\ell : Q & \approx_m [\![(snd(m(d)), \top).t_2]\!]_\ell : Q, r \in I\!R^{>0} \\ [\![(snd(m(d)), \top).t]\!]_\ell : Q & \approx_m [\![t]\!]_\ell : Q \cdot m(d) \end{array}$$

The first explains that immediate actions have precedence over delayable actions. The second explains how an immediate action $snd(m(d))$ can be removed while its impact, insertion of $m(d)$ at the end of the queue, is considered.

4.2 Markovian Semantics of MANETs

A Markov model can be derived from a MANET specification, if the non-deterministic choices can be resolved by application of Markovian network bisimilarity: each configuration equivalence class in the MANET is a state of the stochastic process, and the transitions are defined by collapsing transitions carrying active actions between corresponding configuration equivalence classes while adding up the rates; see Theorem. 2.

Definition 2. *The* derivative set $ds(\mathcal{T})$ *of a MANET model* $\mathcal{T}$ *is the smallest set of configurations such that:*

- $\mathcal{T} \in ds(\mathcal{T})$*; and*
- *if* $\mathcal{T}_i \in ds(\mathcal{T})$*, and* $\mathcal{T}_i \xrightarrow{(\eta,\lambda)} \mathcal{T}_j$ *with* $\lambda \in \{\top\} \cup I\!R^{>0}$ *and* $\eta \in \{\tau, nsnd(m(d), \ell)\}$*, then* $\mathcal{T}_j \in ds(\mathcal{T})$*.*

A Markov process can be derived from a configuration $\mathcal{T}$ if each equivalence class $[\mathcal{T}_i]_{/\approx_m}$ with $\mathcal{T}_i \in ds(\mathcal{T})$ cannot move to another equivalence class by an immediate action:

$$\forall \mathcal{T}_j \in [\mathcal{T}_i]_{/\approx_m} \cdot \mathcal{T}_j \xrightarrow{\tau} \mathcal{T}_j' \Rightarrow \mathcal{T}_j' \in [\mathcal{T}_i]_{/\approx_m}.$$

So immediate actions are removed by weak Markovian network bisimilarity. Such a configuration is called Markovian.

Theorem 2. *Given a finite closed Markovian configuration* $\mathcal{T}$*, let the stochastic process* $X(t)$ *be defined such that* $X(t) = [\mathcal{T}_i]_{/\approx_m}$ *for some* $\mathcal{T}_i \in ds(\mathcal{T})$*, indicating that the MANET is in a configuration belonging to* $[\mathcal{T}_i]_{/\approx_m}$ *at time t. Then* $X(t)$ *is a Markov process.*

The proof is straightforward (cf. [12,18]). If transition rates are independent of the time at which a transition occurs, the derived CTMC is time-homogeneous, so that it can be used to evaluate the performance of a MANET in terms of different data-link layer service quality, mobility models, and protocol parameter settings.

5 A Leader Election Algorithm for MANETs

In this section we illustrate how the *SRBPT* framework is applicable in analyzing MANET protocols. For this purpose we use the leader election algorithm for MANETs from [27].

5.1 Protocol Specification

Each node has a value associated with it. In the context of MANETs, the leader election algorithms aim at finding the highest-valued node within a connected component during a limited period of time, when the underlying topology is stable. For simplicity the value of a node is the same as its network address. Let $? \in Loc$ denote an unknown address. We assume a total order on network addresses, where ? is the minimum. Election is performed in three stages. In the first stage, a spanning tree is grown which (potentially) contains all the nodes within a connected component by broadcasting *elec* messages, which make nodes join the election and send it in turn. To this aim, each node initially sends a elec message after waiting $1/r_{heartbeat}$ in average to receive from a leader in its vicinity. After a node receives $\mathit{elec}(\mathit{xparent})$, it sets its parent to $\mathit{xparent}$ and then immediately relay the message. In the second stage, values are collected through ack messages, which contain the maximum value of a subtree under a node and are passed on to the parent in the spanning tree. Inner nodes of the spanning tree wait for an average time of $1/r_{child\_timeout}$ to gather ack messages from their potential children and inform their parent. On receiving ack, each node updates the maximum value it knows from its subtree.

$$
\begin{array}{l}
\mathit{Node}(s_n : \mathit{Nat}, \mathit{id}, \mathit{lid}, \mathit{max}, \mathit{parent} : \mathit{Loc}, \mathit{elec}, \mathit{ack} : \mathit{Bool}) \overset{def}{=} \\
\quad [\mathit{eq}(\mathit{lid}, \mathit{id})](\mathit{snd}(\mathit{leader}(\mathit{lid})), r_{heartbeat}).\mathit{Node}(s_n, \mathit{id}, \mathit{lid}, \mathit{max}, \mathit{parent}, \mathit{elec}, \mathit{ack}) \diamond 0 \\
\quad +[\mathit{eq}(s_n, 0)](\mathit{snd}(\mathit{elec}(\mathit{id})), r_{heartbeat}).\mathit{Node}(1, \mathit{id}, \mathit{lid}, \mathit{id}, \mathit{id}, T, T) \diamond 0 \\
\quad +[\mathit{eq}(s_n, 2)](\mathit{snd}(\mathit{elec}(\mathit{id})), \top).\mathit{Node}(1, \mathit{id}, \mathit{lid}, \mathit{id}, \mathit{parent}, T, T) \diamond 0 \\
\quad +[s_n < 2 \wedge \neg \mathit{elec}] \sum_{lx:Loc}(\mathit{rcv}(\mathit{elec}(\mathit{lx})), \bot).\mathit{Node}(2, \mathit{id}, \mathit{lid}, \mathit{max}, \mathit{lx}, \mathit{elec}, \mathit{ack}) \diamond 0 \\
\quad +[\mathit{eq}(s_n, 1) \wedge \mathit{ack}] \sum_{xmax:Loc} \sum_{xid}(\mathit{rcv}(\mathit{ack}(\mathit{xmax}, \mathit{xid})), \bot) \\
\qquad .\mathit{Node}(s_n, \mathit{id}, \mathit{lid}, \mathit{if}(\mathit{xmax} > \mathit{max}, \mathit{xmax}, \mathit{max}), \mathit{parent}, \mathit{elec}, \mathit{ack}) \diamond 0 \\
\quad +[\mathit{eq}(s_n, 1) \wedge \mathit{ack} \wedge \neg \mathit{eq}(\mathit{parent}, \mathit{id})](\mathit{snd}(\mathit{ack}(\mathit{max}, \mathit{parent})), r_{child_timeout}) \\
\qquad .\mathit{Node}(s_n, \mathit{id}, \mathit{lid}, \mathit{max}, \mathit{parent}, \mathit{elec}, F) \diamond 0 \\
\quad +[\mathit{eq}(s_n, 1) \wedge \mathit{eq}(\mathit{parent}, \mathit{id}) \wedge \mathit{ack}](\mathit{snd}(\mathit{leader}(\mathit{max})), r_{child_timeout}) \\
\qquad .\mathit{Node}(s_n, \mathit{id}, \mathit{max}, \mathit{max}, \mathit{parent}, F, F) \diamond 0 \\
\quad +[\mathit{eq}(s_n, 3)](\mathit{snd}(\mathit{leader}(\mathit{lid})), \top).\mathit{Node}(1, \mathit{id}, \mathit{lid}, \mathit{max}, \mathit{parent}, F, F) \diamond 0 \\
\quad +[s_n < 2 \wedge ((\mathit{elec} \wedge \neg \mathit{ack}) \vee \neg \mathit{elec})] \sum_{xlid:Loc}(\mathit{rcv}(\mathit{leader}(\mathit{xlid})), \bot) \\
\qquad .\mathit{Node}(\mathit{if}(\neg \mathit{elec} \vee (\neg \mathit{ack} \wedge \mathit{xlid} > \mathit{max}), 3, 1), \mathit{id}, \\
\qquad\qquad\quad \mathit{if}((\neg \mathit{elec} \wedge \mathit{xlid} > \mathit{lid}) \vee (\mathit{elec} \wedge \neg \mathit{ack} \wedge \mathit{xlid} > \mathit{max}), \mathit{xlid}, \mathit{lid}), \\
\qquad\qquad\quad \mathit{if}((\neg \mathit{elec} \wedge \mathit{xlid} > \mathit{lid}) \vee (\mathit{elec} \wedge \neg \mathit{ack} \wedge \mathit{xlid} > \mathit{max}), \mathit{xlid}, \mathit{max}), \mathit{parent}, \\
\qquad\qquad\quad \mathit{if}((\mathit{elec} \wedge \mathit{xlid} > \mathit{max}), F, \mathit{elec}), \mathit{ack}) \diamond 0 \\
\quad +[\mathit{eq}(s_n, 1) \wedge \neg \mathit{eq}(\mathit{lid}, ?) \wedge \neg \mathit{eq}(\mathit{lid}, \mathit{id}) \wedge \neg \mathit{elec}] \\
\qquad (\mathit{snd}(\mathit{elec}(\mathit{id})), r_{hb_timeout}).\mathit{Node}(s_n, \mathit{id}, ?, \mathit{id}, \mathit{id}, T, T) \diamond 0 \\
\quad +[\mathit{eq}(s_n, 1)](\mathit{snd}(\mathit{crash}), r_{crash_freq}).\mathit{Node}(0, \mathit{id}, ?, ?, ?, F, F) \diamond 0 \\
\quad + \sum_{xid:Loc}(\mathit{rcv}(\mathit{probe}(\mathit{xid})), \bot).\mathit{Node}(\mathit{if}(\mathit{eq}(\mathit{xid}, \mathit{id}), 5, s_n), \mathit{id}, \mathit{lid}, \mathit{max}, \mathit{parent}, \mathit{elec}, \mathit{ack}) \\
\quad +[\mathit{eq}(s_n, 5)](\mathit{snd}(\mathit{reply}(\mathit{xid})), \top). \\
\qquad \mathit{Node}(\mathit{if}(\mathit{elec} \vee \neg \mathit{eq}(\mathit{lid}, ?), 1, 0), \mathit{id}, \mathit{lid}, \mathit{max}, \mathit{parent}, \mathit{elec}, \mathit{ack}) \diamond 0 \\
\quad +[\mathit{eq}(s_n, 1) \wedge \neg \mathit{eq}(\mathit{parent}, \mathit{id}) \wedge \neg \mathit{ack} \wedge \mathit{elec}](\mathit{snd}(\mathit{probe}(\mathit{parent})), r_{probe_freq}) \\
\qquad .\mathit{Node}(4, \mathit{id}, \mathit{lid}, \mathit{max}, \mathit{parent}, \mathit{elec}, \mathit{ack}) \diamond 0 \\
\quad +[\mathit{eq}(s_n, 4)](\sum_{xparent:Loc}(\mathit{rcv}(\mathit{reply}(\mathit{xparent})), \bot). \\
\qquad\qquad\qquad \mathit{Node}(\mathit{if}(\mathit{eq}(\mathit{xparent}, \mathit{parent}), 1, s_n), \mathit{id}, \mathit{lid}, \mathit{max}, \mathit{parent}, \mathit{elec}, \mathit{ack}) \\
\qquad\qquad +(\mathit{snd}(\mathit{leader}(\mathit{max})), r_{reply_timeout}).\mathit{Node}(1, \mathit{id}, \mathit{max}, \mathit{max}, \mathit{id}, F, F)) \diamond 0
\end{array}
$$

Fig. 3. Specification of a leader election algorithm for MANETs

In the third stage, a node declares the maximum value by broadcasting the leader message, if it is a root (on expiration of a timer with rate $r_{child\_timeout}$ during which acks are gathered), or it has been disconnected from its parent (on expiration of a timer with rate $r_{reply\_timeout}$ to detect its parent). This message is then broadcast periodically with rate $r_{heartbeat}$ to reestablish leadership of a node, until it is challenged by a greater value. If a node does not hear from its leader for an average time of $1/r_{hb\_timeout}$, it initiates a leader election. The spanning tree can change during these stages, since nodes can move into or out of a connected component at will. To keep the tree connected and swiftly respond to changes, a node constantly checks the existence of its parent by sending/receiving *probe*/*reply* messages.

The leader election algorithm is specified by the protocol name $Node$ in Fig. 3. The list of variables maintained by each protocol are: $elec$, ack of type $Bool$, where $elec$ is T when the node is involved in an election, while ack is T if the node has not sent its ack message to its parent; lid, max, $parent$ of type Loc, where lid denotes the address of the leader (which is updated when the node receives a $leader$ message), max is the highest value the node is aware of in an election, and $parent$ is the address of the node's parent in the spanning tree. The protocol is initially (or after a crash) in a state with $eq(s_n, 0)$, $eq(elec, F)$, $eq(ack, F)$, $eq(parent, ?)$ and $eq(lid, ?)$.

5.2 Protocol Analysis

We focus on evaluating effects of some parameters like r_{Mac} and timer values on protocol performance. We construct configurations by composing several $[\![Node(0, \ell, ?, ?, ?, F, F)]\!]_\ell : Q_0$, and examine message overhead and the duration from the start of the election until all nodes have found a leader (called election time). It should be examined that configurations are Markovian, and consequently by Theorem. 2 a CTMC can be derived.

Each node immediately sends a message when it is in the state $eq(s_n, 2) \vee eq(s_n, 3) \vee eq(s_n, 5)$. It can be shown that the following equivalences hold (by constructing the Markovian network bisimulation relation $\mathcal{R} = \{(\mathcal{T}_1, \mathcal{T}_2) | \forall Q_i \cdot i < K\} \cup \{(\mathcal{T}, \mathcal{T})\}$ for each equivalence relation $\mathcal{T}_1 \approx_m \mathcal{T}_2$):

$$[\![Node(2, \theta, F, ack)]\!]_\ell : Q \approx_m [\![Node(1, \theta, F, ack)]\!]_\ell : Q \cdot elec(\ell)$$
$$[\![Node(3, \theta, elec, ack)]\!]_\ell : Q \approx_m [\![Node(1, \theta, F, F)]\!]_\ell : Q \cdot leader(lid)$$
$$[\![Node(5, \theta', F, ack)]\!]_\ell : Q \approx_m [\![Node(0, \theta', F, ack)]\!]_\ell : Q \cdot reply(\ell')$$
$$[\![Node(5, \theta, elec, ack)]\!]_\ell : Q \approx_m [\![Node(1, \theta, elec, ack)]\!]_\ell : Q \cdot reply(\ell'),\ \text{where } elec \vee \neg eq(lid, ?)$$

where θ abbreviates $id, lid, max, parent$ and θ' abbreviates $id, ?, max, parent$. Therefore, by congruence of $\approx_m$, in any configuration, nodes in the form of the left-hand side of an equation above can be replaced by the corresponding right-hand side.

We exploit PRISM to derive the overall CTMC resulting from a MANET configuration. To this aim, we cast the resulting CTMC of each node to PRISM such that its parallel composition with other nodes in the MANET results in the overall CTMC. See the [13] for how the encoding is managed. We note that the cast CTMC of a node in PRISM is dependent on all locations in the MANET (no modularity), and it is not straightforward to write the code in PRISM from the scratch. We implemented the Loc data sort using the integer type of PRISM (where $?, A, B, \ldots$ are denoted by $0, 1, 2, \ldots$), and so the cast of the configuration $[\![Node(0, \ell, ?, ?, ?, F, F)]\!]_\ell : Q_0$ is well-defined.[1]

[1] See `http://mehr.sharif.edu/~fghassemi/pcodes.zip`.

Then we can define desired properties in a well-known stochastic temporal logic, Continuous Stochastic Logic [1], which has been extended in PRISM with rewards and queries. By assigning a reward to each send action, we can compute the number of messages sent during an election:

$$R_{|messages|=?}[F lid_A \neq ? \wedge lid_B \neq 0 \wedge lid_C \neq 0 \wedge \ lid_D \neq 0 \wedge lid_E \neq 0)]$$

where the condition $F lid_A \neq ? \wedge lid_B \neq 0 \wedge lid_C \neq 0 \wedge lid_D \neq 0 \wedge lid_E \neq 0$ specifies that all nodes will finally have a leader. We can thus examine the message overhead for different implementation policies. We have also examined another implementation of the protocol; the node tries to participate in the election of a node having the highest value. Thus each node listens to the next election messages, and whenever it receives one with a greater value than its own $parent$, it immediately changes its parent. We use "single" and "multiple" election for the first and second implementation. In Fig. 4, the trend of the message overhead growth is illustrated with respect to the number of nodes for each implementation, which shows the nearly linear relationship between the increase in the number of nodes and the increase in the message overhead. We used the values for $r_{hearbeat} = 0.05$ and $r_{hb\_timeout} = 6 \times r_{hearbeat}$ given in [27] and $r_{child\_timeout} = 0.02$, $r_{crash\_freq} = 0.000028$, $r_{prob\_freq} = 0.1$, and $r_{reply\_timeout} = 0.1$. P_{UP} is 0.7, and r_{Mac} is 10.

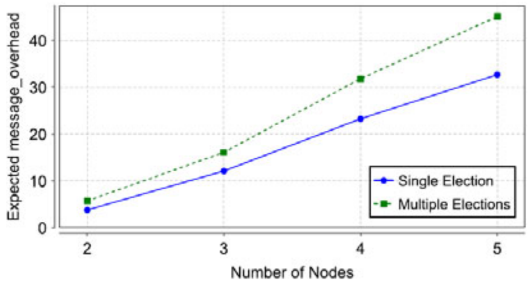


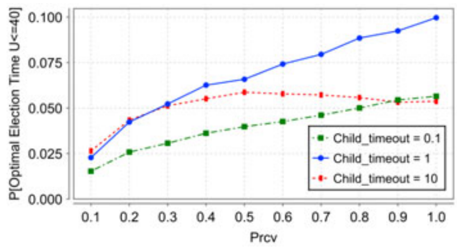
Fig. 4. Message overhead in MANETs of different size

We can compute the distribution function for the duration of an election (for a specific node) in which the highest-valued node is elected as the leader by

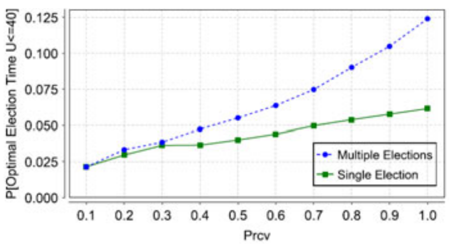
$$P_{=?}[lid_A = 0 \ U_T \ lid_A = max\{id_A, id_B, id_C, id_D, id_E\}.$$

where the until operator U_T computes the time between when the node has no leader and when its leader has the maximum value. In Fig. 5a, the probability that this election time is less than 40 seconds (for a MANET of five nodes with a data-link layer with capacity 2) is measured in terms of different values of P_{rcv} and $t_{child\_timeout} = 1/r_{child\_timeout}$. Fig. 5a shows that the optimal choice for child timeout in a network with varying P_{rcv} is $1.0s$ for the single election implementation. We can again examine the effect of parameters like P_{UP}.

This probability is compared for each implementation when $t_{child\_timeout} = 1$ in Fig. 5b: the election time for the multiple election implementation is reduced.



(a)



(b)

Fig. 5. Effect of P_{rcv} and $t_{child\_timeout}$ on election time (left). Probability that the election time is less than 40 sec. for a single and multiple elections (right).

Depending on the quality of service metrics, election time or message overhead, either one of the protocol implementations can be chosen.

6 Conclusion and Future Work

We have extended *RBPT*, an algebraic framework for the specification and verification of MANETs, with stochastic concepts. This is useful for the analysis of protocols in terms of environment (like data-link layer quality of service, mobility of nodes) and protocol parameters.

We plan to extend *SRBPT* following the approach of [9], to arrive at a sound and complete axiomatization. Then we can define a stochastic variant of linear process equations (SLPE) for configurations, which can be exploited in the analysis of MANETs with an arbitrary number of nodes. To this aim, we can also extract the embedded DTMC of an SLPE and then use symbolic confluence reduction [3] or the SCOOP tool [26] to reduce the SLPE. With an increase in the number of nodes, the effect of T_{Mac} on a MANET can be sensed more.

References

1. Baier, C., Haverkort, B., Hermanns, H., Katoen, J.: On the logical characterisation of performability properties. In: Welzl, E., Montanari, U., Rolim, J.D.P. (eds.) ICALP 2000. LNCS, vol. 1853, pp. 780–792. Springer, Heidelberg (2000)
2. Bernardo, M., Gorrieri, R.: A tutorial on EMPA: A theory of concurrent processes with nondeterminism, priorities, probabilities and time. Theoretical Computer Science 202(1-2), 1–54 (1998)
3. Blom, S., van de Pol, J.: State space reduction by proving confluence. In: Brinksma, E., Larsen, K.G. (eds.) CAV 2002. LNCS, vol. 2404, pp. 596–609. Springer, Heidelberg (2002)
4. Daws, C., Kwiatkowska, M., Norman, G.: Automatic verification of the IEEE 1394 root contention protocol with KRONOS and PRISM. Software Tools for Technology Transfer 5(2), 221–236 (2004)
5. Duflot, M., Kwiatkowska, M., Norman, G., Parker, D.: A formal analysis of Bluetooth device discovery. In: Proc. ISOLA 2004, pp. 268–275 (2004)
6. Ehrich, H., Loeckx, J., Wolf, M.: Specification of Abstract Data Types. John Wiley, Chichester (1996)

7. Ene, C., Muntean, T.: Expressiveness of point-to-point versus broadcast communications. In: Ciobanu, G., Păun, G. (eds.) FCT 1999. LNCS, vol. 1684, pp. 258–268. Springer, Heidelberg (1999)
8. Fehnker, A., Fruth, M., McIver, A.K.: Graphical modelling for simulation and formal analysis of wireless network protocols. In: Butler, M., Jones, C., Romanovsky, A., Troubitsyna, E. (eds.) Methods, Models and Tools for Fault Tolerance. LNCS, vol. 5454, pp. 1–24. Springer, Heidelberg (2009)
9. Ghassemi, F., Fokkink, W., Movaghar, A.: Equational reasoning on mobile ad hoc networks. Fundamenta Informaticae 103, 1–41 (2010)
10. Ghassemi, F., Fokkink, W., Movaghar, A.: Verification of mobile ad hoc networks: An algebraic approach. Theoretical Computer Science 412(28), 3262–3282 (2011)
11. Ghassemi, F., Fokkink, W.J., Movaghar, A.: Restricted broadcast process theory. In: Proc. SEFM 2008, pp. 345–354. IEEE, Los Alamitos (2008)
12. Ghassemi, F., Fokkink, W.J., Movaghar, A.: Towards performance evaluation of mobile ad hoc network protocols. In: Proc. ACSD 2010, pp. 98–105. IEEE, Los Alamitos (2010)
13. Ghassemi, F., Talebi, M.: Fokkink, W., Movaghar, A.: Stochastic restricted broadcast process theory. Tech. rep., Sharif University of Technology (2011), `http://mehr.sharif.edu/fghassemi/srbpt-web.pdf`
14. van Glabbeek, R., Weijland, W.: Branching time and abstraction in bisimulation semantics. Journal of the ACM 43(3), 555–600 (1996)
15. Groote, J.F., Ponse, A.: Syntax and semantics of μ-CRL. In: Proc. ACP 1994. Workshops in Computing, pp. 26–62. Springer, Heidelberg (1995)
16. Hermanns, H., Herzog, U., Katoen, J.P.: Process algebra for performance evaluation. Theoretical Computer Science 274(1-2), 43–87 (2002)
17. Hermanns, H., Rettelbach, M., Weiss, T.: Formal characterisation of immediate actions in spa with nondeterministic branching. The Computer Journal 38(7), 530–541 (1995)
18. Hillston, J.: A Compositional Approach to Performance Modelling. Ph.D. thesis, Cambridge University (1996)
19. Khabbazian, M., Kuhn, F., Kowalski, D.R., Lynch, N.A.: Decomposing broadcast algorithms using abstract mac layers. In: Proc. DIALM-PODC, pp. 13–22. ACM, New York (2010)
20. Kloul, L., Valois, F.: Investigating unfairness scenarios in manet using 802.11b. In: Proc. PE-WASUN, pp. 1–8. ACM, New York (2005)
21. Kwiatkowska, M., Norman, G., Parker, D., Sproston, J.: Performance analysis of probabilistic timed automata using digital clocks. Formal Methods in System Design 29, 33–78 (2006)
22. Kwiatkowska, M., Norman, G., Sproston, J.: Probabilistic model checking of the IEEE 802.11 wireless local area network protocol. In: Hermanns, H., Segala, R. (eds.) PROBMIV 2002, PAPM-PROBMIV 2002, and PAPM 2002. LNCS, vol. 2399, pp. 169–187. Springer, Heidelberg (2002)
23. Lin, T.: Mobile Ad-hoc Network Routing Protocols: Methodologies and Applications. Ph.D. thesis, Virginia Polytechnic Institute and State University (2004)
24. Oliveira, R., Bernardo, L., Pinto, P.: Modelling delay on IEEE 802.11 MAC protocol for unicast and broadcast nonsaturated traffic. In: Proc. WCNC 2007, pp. 463–467. IEEE, Los Alamitos (2007)
25. Razafindralambo, T., Valois, F.: Performance evaluation of backoff algorithms in 802.11 adhoc networks. In: Proc. PE-WASUN 2006, pp. 82–89. ACM, New York (2006)
26. Timmer, M.: Scoop: A tool for symbolic optimisations of probabilistic processes. In: QEST 2011 (to appear, 2011)
27. Vasudevan, S., Kurose, J., Towsley, D.: Design and analysis of a leader election algorithm for mobile ad hoc networks. In: Proc. ICNP 2004, pp. 350–360. IEEE, Los Alamitos (2004)
28. Zuniga, M., Krishnamachari, B.: Analyzing the transitional region in low power wireless links. In: Proc. SECON 2004, pp. 517–526. IEEE, Los Alamitos (2004)