Verification of Ad Hoc Networks with Node and Communication Failures

Giorgio Delzanno, Arnaud Sangnier, Gianluigi Zavattaro

Verification of Ad Hoc Networks with Node and Communication Failures

Giorgio Delzanno[1], Arnaud Sangnier[2], and Gianluigi Zavattaro[3]

[1] University of Genova, Italy
[2] LIAFA, Univ Paris Diderot, Sorbonne Paris Cité, CNRS, France
[3] University of Bologna, INRIA - FOCUS Research Team, Italy

Abstract. We investigate the impact of node and communication failures on the decidability and complexity of parametric verification of a formal model of ad hoc networks. We start by considering three possible types of node failures: intermittence, restart, and crash. Then we move to three cases of communication failures: nondeterministic message loss, message loss due to conflicting emissions, and detectable conflicts. Interestingly, we prove that the considered decision problem (reachability of a control state) is decidable for node intermittence and message loss (either nondeterministic or due to conflicts) while it turns out to be undecidable for node restart/crash, and conflict detection.

1 Introduction

Broadcast communication is often used in networks in which individual nodes have no precise information about the underlying connection topology (e.g. ad hoc wireless networks). As shown in [13,10,11,16,17,4], this type of communication can naturally be specified in models in which a network configuration is represented as a graph and in which individual nodes run an instance of a given protocol specification. A protocol typically specifies a sequence of control states in which a node can either send a message (emitter role), waits for a message (receiver role), or performs an update of its internal state. Broadcast communication can be represented here as a simultaneous update of the state of the emitter node and of the states of its neighbors. This semantics of broadcast is often termed *selective* in contrast with broadcast messages that simultaneously reach all nodes of a network.

Already at this level of abstraction, verification of ad hoc network protocols turns out to be a very difficult task. A formal account of this problem is given in [3,4], where the *control state reachability problem* is proved to be undecidable for selective broadcast communication. The control state reachability problem consists in verifying the existence of an initial network configuration (with unknown size and topology) that may evolve into a configuration in which at least one node is in a given control state. If such a control state represents a protocol error, then this problem naturally expresses (the complement of) a safety verification task in a setting in which nodes have no information a priori about the

size and connection topology of the underlying network. The analysis in [3,4] works under the assumption that the underlying network and communication model are both reliable. This is a quite strong assumption since ad hoc networks have several sources of unreliability: from node failures to conflicts caused by interferences among different transmissions.

In this paper we study the impact of node and communication failures on the control state reachability problem for ad hoc network protocols. We start our analysis by introducing node failures in a model of selective broadcast. For this purpose, we consider an intermittent semantics in which a node can be (de)activated at any time. As a first result, we show that control state reachability becomes decidable under the intermittent semantics. Decidability seems strictly related to the assumption that nodes have cannot directly take decisions that depend on the current activation state (e.g. change state when the node is turned on). We then consider two restricted types of node failure, i.e., node crash (a node can only be deactivated) and node restart (when it is activated, it restarts in a special restart state). We show that for these two semantics, the verification task becomes undecidable.

We consider then different types of communication failures. We first consider a semantics in which a broadcast is not guaranteed to reach all neighbors of the emitter nodes (message loss). Control state reachability is again decidable in this case. We then introduce a semantics for selective broadcast specifically designed to capture possible conflicts during a transmission. Basically, a transmission of a broadcast message is split into two different phases: a starting and an ending phase. During the starting phase, receivers connected to the emitter move to a transient state. While being in the transient state, a reception from another node generates a conflict. In the ending phase an emitter always moves to the next state whereas connected receivers move to their next state only when no conflicts have been detected. Time-out can be modeled here by allowing receivers to abandon a transmission at any time. In our model we also allow several emitters to simultaneously start a transmission. Decidability holds only when receivers ignore corrupted messages by remaining in their original state. Moreover, for the verification task in the decidable variants we show that it is possible to resort to the polynomial time reachability algorithm that we have presented for a model of ad hoc networks with nondeterministic mobility presented in [2].

Related Work. Formal models of broadcast communication have been considered in several work in the literature such as [14,16,17,6,5,8,10,11,12]. Perfect synchronous semantics for broadcast communication in mobile and ad hoc networks have been proposed in [14,16,17,5]. Verification problems for broadcast protocols has been studied in the different context of hardware protocols [6]. In all the above mentioned works a transmission is modelled as an atomic step in which the emitter node and the connected receiver nodes simultaneously update their current state. Decidability of reachability problems like those we consider here (coverability) is not considered only in the case of synchronous broadcast for fully connected networks [6].

Delays in between the instant in which the emitter starts a transmission and the instant in which the transmission ends have been considered in a timed semantics [10,11] in which every message has an associated non-zero transmission time, or in form of non-atomic transitions (start and end phase are kept distinct) as in [12]. In all these approaches a broadcast communication is split into several phases to model scenarios in which different transmission periods of different emitters overlap. Following [12] in the present paper we consider an untimed semantics for explicitly representing conflicts. Differently from other models, our semantics allows multiple nodes to start a communication in the same instant, a model that seems closer to real scenarios.

In [3,4] we have studied decision problems for verification of models of ad hoc networks with seective broadcast communication with perfect semantics and no conflicts. In this paper we lift our studies to unreliable networks and communication models and consider semantics for broadcast communication with conflicts. Communication failures (e.g. message loss and insertion) are commonly considered when facing verification problems for communication protocols as in the case of unreliable FIFO channels [1]. Differently from works like [1], we evaluate here the impact of communication failures in a communication model with broadcast communication restricted to neighbour nodes and in which reachability is formulated for an initial configuration with arbitrary size and topology.

2 Ad Hoc Networks

Definition 1. *A Q-graph is a labeled undirected graph $\gamma = \langle V, E, L\rangle$, where V is a finite set of* nodes*, $E \subseteq V \times V$ is a symmetric relation representing a finite set of* edges*, and L is a labeling function from V to a set of labels Q (in our setting they represent control states).*

We use $L(\gamma)$ to represent all the labels present in γ (i.e. the image of the function L). The nodes belonging to an edge are called the *endpoints* of the edge. For an edge $\langle u, v\rangle$ in E, we use the notation $u \sim_\gamma v$ and say that the vertices u and v are adjacent to each other in the graph γ. We omit γ, and simply write $u \sim v$, when it is made clear by the context.

A configuration is a Q-graph and we assume that each node of the graph is a process that runs a common predefined protocol defined by a communicating automaton with a finite set Q of control states. Communication is achieved via selective broadcast: the effect of a broadcast is local to the vicinity of the sender. The initial configuration is any graph in which all the nodes are labeled by an initial control state. Note that even if Q is finite, there are infinitely many possible configurations (the number of Q-graphs). We next formalize the above intuition.

Definition 2. *A process is a tuple $\mathcal{P} = \langle Q, \Sigma, R, Q_0\rangle$, where Q is a finite set of control states, Σ is a finite alphabet, $R \subseteq Q \times (\{\tau\} \cup \{!!a, ??a \mid a \in \Sigma\}) \times Q$ is the transition relation, and $Q_0 \subseteq Q$ is a set of initial control states.*

The label τ represents the capability of performing an internal action, and the label $!!a$ ($??a$) represents the capability of broadcasting (receiving) a message $a \in \Sigma$. For $q \in Q$ and $a \in \Sigma$, we define the set $R_a(q) = \{q' \in Q \mid \langle q, ??a, q' \rangle \in R\}$ which contains states that can be reached from the state q when receiving the message a.

The network semantics associated to a process $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$ is given by the transition system $AHN(\mathcal{P}) = \langle \mathcal{C}, \Rightarrow, \mathcal{C}_0 \rangle$, where $\mathcal{C}$ is the set of Q-graphs (network configurations), $\mathcal{C}_0$ is the set of Q_0-graphs (initial configurations), and $\Rightarrow \subseteq \mathcal{C} \times \mathcal{C}$ is the transition relation defined as follows: for $\gamma = \langle V, E, L \rangle$, we have $\gamma \Rightarrow \gamma'$ iff $\gamma' = \langle V, E, L' \rangle$ and one of the following conditions holds:

Local: $\exists v \in V$ s.t. $(L(v), \tau, L'(v)) \in R$, and $L(u) = L'(u)$ for all u in $V \setminus \{v\}$;
Broadcast: $\exists v \in V$ s.t. $(L(v), !!a, L'(v)) \in R$ and for every $u \in V \setminus \{v\}$, we have:
- if $u \sim v$ and $R_a(L(u)) \neq \emptyset$ (u can receive a), then $L'(u) \in R_a(L(u))$,
- $L(u) = L'(u)$, otherwise.

An execution in $AHN(\mathcal{P})$ is a sequence $\gamma_0\gamma_1 \ldots$ such that $\gamma_0 \in \mathcal{C}_0$ and $\gamma_i \Rightarrow \gamma_{i+1}$ for $i \geq 0$. We use $\Rightarrow^*$ to denote the reflexive and transitive closure of $\Rightarrow$.

Observe that a broadcast message a sent by v is delivered only to the subset of neighbors interested in it; such a neighbor u has then to update its state with a new state taken from $R_a(L(u))$. All the other nodes (including neighbors not interested in a) simply ignore the message. Also notice that the topology is static, i.e., the set of nodes and edges remain unchanged during an execution.

As an example of an ad hoc network and of its semantics, consider a process consisting of the following rules: (A, τ, C), $(C, !!m, D)$, $(B, ??m, C)$, and $(A, ??m, C)$. As shown in Figure 1, starting from a configuration with only A and B nodes, an A node first moves to C and then sends m to his/her neighbors. In turn, they forward the message m to their neighbors, and so on.

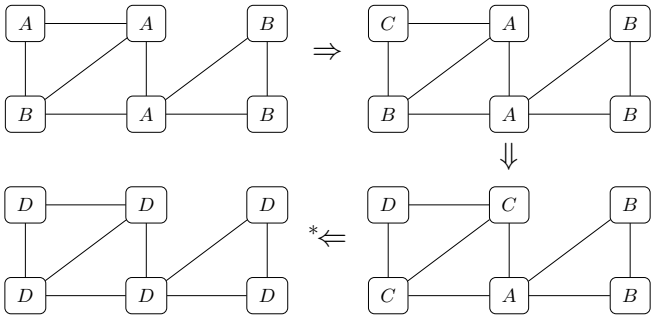

Fig. 1. Example of normal execution

The network semantics formalized by the transition system $\Rightarrow$ assumes fixed topology. Formally, if $\gamma \Rightarrow \gamma'$ then $\gamma = \langle V, E, L \rangle$ and $\gamma' = \langle V, E, L' \rangle$ share

the same nodes and edges and can differ only in the labeling function. In [3] we have formalized also nondeterministic mobility as follows. Given a process $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$ the mobile network semantics is given by the transition system $MAHN(\mathcal{P}) = \langle \mathcal{C}, \rightsquigarrow, \mathcal{C}_0 \rangle$, where $\mathcal{C}$ and $\mathcal{C}_0$ are as in the definition of $AHN(\mathcal{P})$ and $\rightsquigarrow \subseteq \mathcal{C} \times \mathcal{C}$ is the transition relation defined as follows: for $\gamma = \langle V, E, L \rangle$, we have $\gamma \rightsquigarrow \gamma'$ iff $\gamma' = \langle V, E', L' \rangle$ and one of the following conditions holds:

State transition: $\gamma \Rightarrow \gamma'$;
Mobility: $E' \subseteq V \times V$ and $L' = L$.

Observe that all the transitions of the original $AHN(\mathcal{P})$ transition system are included by the state transition rule, while the mobility rule adds transitions that modify the edges arbitrarily while preserving the labeling function.

2.1 Safety Analysis: the Control State Reachability Problem

Following [3,4] we consider decision problems related to verification of safety properties. We remark that in our formulation the size and topology of the initial configurations is not fixed a priori. The problem that we consider is *control state reachability* (COVER) defined as follows:

Input: A process $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$ with $AHN(\mathcal{P}) = \langle \mathcal{C}, \Rightarrow, \mathcal{C}_0 \rangle$ and a control state $q \in Q$.
Output: Yes, if $\exists \gamma \in \mathcal{C}_0$ and $\gamma' \in \mathcal{C}$ s.t. $\gamma \Rightarrow^* \gamma'$ and $q \in L(\gamma')$; no, otherwise.

If q represents an error state, COVER amounts at checking whether there exists an initial configuration (among the infinitely many possible ones) from which a configuration containing a node in the error state is reachable.

In [3], we prove the following result.

Theorem 1. COVER *is undecidable.*

In the following we will also consider COVER for the mobile network semantics: in that case the transitions $\gamma \rightsquigarrow \gamma'$ will be taken into account instead of $\gamma \Rightarrow \gamma'$. In [3] we have proved that COVER turns out to be decidable with spontaneous (i.e. non-deterministic) mobility. Indeed, in this setting the topology of the network cannot be exploited to build structures that could be applied to model an unbounded storage. In a more recent work [2], we have characterized its complexity.

Theorem 2. COVER *for mobile ad hoc networks is* PTIME*-complete.*

We will also study different semantics for ad hoc networks and we will consider COVER for these semantics. However, sometimes the labelled graphs representing the configurations will have more information in their labels than only the control state of the process, for these cases, COVER will correspond to the reachability of a configuration in which there exists a node whose label contains the desired control state.

3 Node Failures

3.1 Intermittent Nodes

We start our analysis from a semantic variant that models intermittent nodes. We modify the network semantics by using a flag, which is set to $\mathtt{A}$ [resp. to $\mathtt{D}$] to denote an active [resp. deactivated] node.

Definition 3. *Given a process $\mathcal{P} = \langle Q, \Sigma, R, Q_0\rangle$, an i-configuration is a $(Q \times \{\mathtt{A}, \mathtt{D}\})$-graph and an initial i-configuration is a $(Q_0 \times \{\mathtt{A}, \mathtt{D}\})$-graph.*

We use $\mathcal{C}^{int}$ [resp. $\mathcal{C}_0^{int}$] to denote the set of i-configurations [resp. initial i-configurations] associated to a process definition $\mathcal{P}$. Given a process $\mathcal{P} = \langle Q, \Sigma, R, Q_0\rangle$, the semantics of the corresponding ad hoc network with intermittent nodes is given by the transition system $AHN_i(\mathcal{P}) = \langle \mathcal{C}^{int}, \dashrightarrow, \mathcal{C}_0^{int}\rangle$ where the transition relation $\dashrightarrow \subseteq \mathcal{C}^{int} \times \mathcal{C}^{int}$ is defined as follows: for $\gamma = \langle V, E, L\rangle$, we have $\gamma \dashrightarrow \gamma'$ iff $\gamma' = \langle V, E, L'\rangle$ and one of the following conditions holds:

Local: $\exists v \in V$ s.t. $L(v) = \langle q, \mathtt{A}\rangle$, $L'(v) = \langle q', \mathtt{A}\rangle$, $(q, \tau, q') \in R$, and $L(u) = L'(u)$ for all u in $V \setminus \{v\}$;

Broadcast: $\exists v \in V$ s.t. $L(v) = \langle q, \mathtt{A}\rangle$, $(q, !!a, q') \in R$, $L'(v) = \langle q', \mathtt{A}\rangle$, and for every u in $V \setminus \{v\}$:

- if $u \sim v$ and $L(u) = \langle q'', \mathtt{A}\rangle$ and $R_a(q'') \neq \emptyset$, then $L'(u) = \langle q''', \mathtt{A}\rangle$ with $q''' \in R_a(u)$;
- $L(u) = L'(u)$, otherwise.

Intermittence: $\exists v \in V$ s.t. $L(v) = \langle q, \mathtt{A}\rangle$ [resp. $L(v) = \langle q, \mathtt{D}\rangle$], $L'(v) = \langle q, \mathtt{D}\rangle$ [resp. $L(v) = \langle q, \mathtt{A}\rangle$] , and $L(u) = L'(u)$ for all u in $V \setminus \{v\}$.

Note that the transition relation is defined as in the previous section with only two differences: the transitions already present in the previous definition now apply only to active nodes (i.e. those with the flag $\mathtt{A}$); additional transitions allow one node to move from the active to the passive state, and vice versa. We denote by $\dashrightarrow^*$ the reflexive and transitive closure of $\dashrightarrow$.

An example of ad hoc network protocol and of its semantics under node intermittence, consider the following protocol: $(A, !!m, D)$, $(C, !!m, D)$, $(B, ??m, C)$, and $(A, ??m, C)$. As shown in Figure 2, the top-left node is initially deactivated. It then activates, sends a message, and only active neighbors react, and so on.

We now prove that COVER is PTIME-complete also for ad hoc networks with intermittent nodes. This result follows from a the correspondence between $AHN_i(\mathcal{P})$ and $MAHN(\mathcal{P})$ formalized by the following proposition.

Proposition 1. *Consider a process definition $\mathcal{P}$ and a control state q. A configuration γ s.t. $q \in L(\gamma)$ is reachable from an initial configuration in $AHN_i(\mathcal{P})$* if and only if *a configuration γ' s.t. $q \in L(\gamma')$ is reachable from an initial configuration in $MAHN(\mathcal{P})$.*

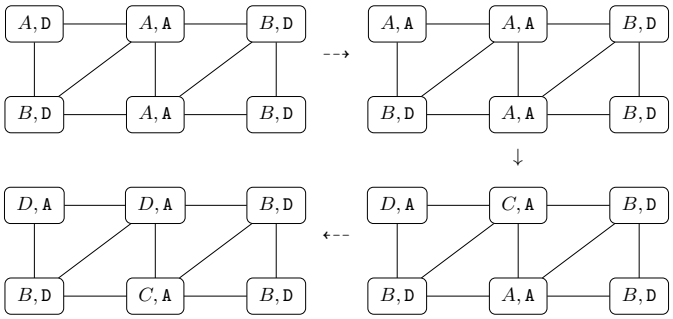

Fig. 2. Example of execution with intermittent nodes

Proof. We start from the *only if* part. Consider the initial state $\gamma_0 = \langle V, E, L_0 \rangle$ and the execution $\gamma_0 \dashrightarrow^* \gamma$ in $AHN_i(\mathcal{P})$ with $q \in L(\gamma)$. A similar execution can be reproduced also in $MAHN(\mathcal{P})$. Consider the initial configuration $\gamma'_0 = \langle V, E, L'_0 \rangle$ with, for every $v \in V$, $L'_0(v) = q_v$ assuming $L_0(v) = \langle q_v, \mathtt{A} \rangle$ or $L_0(v) = \langle q_v, \mathtt{D} \rangle$. Consider now the following execution $\gamma'_0 \rightsquigarrow^* \gamma'$ constructed from the above execution $\gamma_0 \dashrightarrow^* \gamma$ as follows. All the **Local** and **Broadcast** transitions are faithfully reproduced, while the **Intermittence** transitions are mimicked by a **Mobility** transition: in case of deactivation of one node the **Mobility** transition disconnects such node from its neighbors, while in case of node activation the **Mobility** transition restores the previously removed edges. It is easy to see that $q \in L(\gamma')$.

We now move to the *if* part. Consider the initial state $\gamma'_0 = \langle V', E', L'_0 \rangle$ and the execution $\gamma'_0 \rightsquigarrow^* \gamma'$ in $MAHN(\mathcal{P})$ with $q \in L(\gamma')$. A similar execution can be reproduced also in $AHN_i(\mathcal{P})$. Consider the initial configuration $\gamma_0 = \langle V', E, L_0 \rangle$ with $E = V' \times V'$ (i.e. γ_0 is a complete graph) and, for every $v \in V'$, $L_0(v) = \langle q_v, \mathtt{A} \rangle$ assuming $L'_0(v) = q_v$. Consider now the following execution $\gamma_0 \dashrightarrow^* \gamma$ constructed from the above execution $\gamma'_0 \rightsquigarrow^* \gamma'$ as follows. All the **Local** transitions are faithfully reproduced; the **Broadcast** transitions are reproduced by a protocol that first deactivates the nodes that are not neighbors of the emitter in the corresponding mobile network execution, then the broadcast actions is mimicked, and then the previously deactivated nodes are re-activated; the **Mobility** transitions are not reproduced. It is easy to see that $q \in L(\gamma)$. □

As a simple corollary of the above Proposition and Theorem 2 we obtain the following.

Theorem 3. COVER *for ad hoc networks with intermittent nodes is* PTIME*-complete.*

3.2 Node Crash and Restart

We now consider two variants of the semantics with intermittence. In the first one, modelling node crash, nodes can only be deactivated. In the second one,

modelling node restart, nodes can also be reactivated but then they restart from a given special state.

Given process $\mathcal{P}$, its transition system with node crash denoted by $AHN_{cr}(\mathcal{P})$, is defined as the transition system $AHN_i(\mathcal{P})$ where the **Intermittence** transitions are replaced by the following **Crash** transitions:

Crash: $\exists v \in V$ s.t. $L(v) = \langle q, \mathtt{A}\rangle$, $L'(v) = \langle q, \mathtt{D}\rangle$, and $L(u) = L'(u)$ for all u in $V \setminus \{v\}$.

Note that with this semantics, nodes that have been turned off (or deactivated) cannot be activated again.

The variant with restart requires the indication of the restart state in the process. So a process $\mathcal{P} = \langle Q, \Sigma, R, Q_0, q_r\rangle$ now includes a restart state $q_r \in Q$. The transition system $AHN_r(\mathcal{P})$ with node restart for $\mathcal{P}$, is defined as the transition system $AHN_i(\langle Q, \Sigma, R, Q_0\rangle)$ where the **Intermittence** transitions are replaced by the following **Restart** transitions:

Restart: $\exists v \in V$ s.t. $L(v) = \langle q, \mathtt{A}\rangle$ [resp. $L(v) = \langle q, \mathtt{D}\rangle$], $L'(v) = \langle q, \mathtt{D}\rangle$ [resp. $L'(v) = \langle q_r, \mathtt{A}\rangle$] and $L(u) = L'(u)$ for all u in $V \setminus \{v\}$.

In this case, besides the transitions turning off nodes, there are also transitions that turn on one node by changing its internal state to the restart state q_r. The following theorem then holds.

Theorem 4. COVER *with node crash [resp. with node restart] is undecidable.*

Proof. The proof is by reduction from the undecidability of COVER for ad hoc networks (Theorem 1). We first consider the model with node crash. Let $\mathcal{P}$ be a process. It is trivial to see that a computation leading to a configuration that exposes the control state q in $AHN(\mathcal{P})$ has a corresponding computation in $AHN_{cr}(\mathcal{P})$ (in which no **Crash** transition is performed).

Consider now a computation in $AHN_{cr}(\mathcal{P})$ leading to a configuration that exposes the control state q. It is not restrictive to assume that the state q is exposed by a node that did not crash during the computation (we can always consider the last step in q before the node crashes). Consider now a computation in $AHN(\mathcal{P})$ that performs the same **Local** and **Broadcast** transitions (but not the **Crash** transitions). It is easy to see that the nodes that did not crash during the computation in $AHN_{cr}(\mathcal{P})$ are in the same state also in the computation of $AHN(\mathcal{P})$. Hence also the latter computation leads to a configuration exposing the control state q.

The undecidability can be proved as in [3] where we present how to translate a two counter machine (a Turing powerful formalism) into a protocol $\mathcal{P}$ for ad hoc network without failures. Such protocol $\mathcal{P}$ should be slightly modified as follows to work also under intermittence. Let $\mathcal{P} = \langle Q, \Sigma, R, Q_0\rangle$; the modified protocol is defined as $\mathcal{P}' = \langle Q', \Sigma', R', \{q_0\}, q_0\rangle$ where $q_0 \notin Q$ and R' is obtained from R by adding the following rules: $(q_0, !!init, q_0')$ and (q_0', τ, q) for all $q \in Q_0$ and $(q, ??init, q_{err})$ for all $q \in Q$ and this assuming that $q_0', q_{err} \in Q' \setminus Q$. The idea of this encoding is that the unique initial state and the restart state are

the same, but when a node comes back to the initial state while simulating the protocol $\mathcal{P}$, if it goes to q_0' it sends all his neighbors (which are in state belonging to Q) into the deadlock state q_{err}. This ensures that if a node is turned off and is reactivated, it cannot play a role in the simulation of the protocol $\mathcal{P}$ by $\mathcal{P}'$. □

4 Communication Failures

4.1 Message Loss

The first type of failures corresponds to nondeterministic message loss: when a message is broadcasted, some of the receivers could not receive it.

A process $\mathcal{P}$ is defined as usual. The corresponding transition system $AHN_l(\mathcal{P})$ is defined as $AHN(\mathcal{P})$ where the **Broadcast** transitions are replaced by the following **Message loss** transitions:

Message loss: $\exists v \in V$ s.t. $(L(v), !!a, L'(v)) \in R$ and for every $u \in V \setminus \{v\}$
- if $u \sim v$ and $R_a(L(u)) \neq \emptyset$ (reception of a in u is enabled), then $L'(u) \in R_a(L(u))$ or $L'(u) = L(u)$,
- $L(u) = L'(u)$, otherwise.

The main difference with the transition system $AHN(\mathcal{P})$ is that during the performance of a broadcast, some of the potential receivers could remain in their internal state. This is similar to what happens in the model with intermittent nodes when one is deactivated. Starting from this observation it is easy to show that there exists a computation leading to a configuration that exposes the control state q in $AHN_l(\mathcal{P})$ iff there exists a corresponding computation in $AHN_i(\mathcal{P})$. From this consideration, we deduce the following theorem.

Theorem 5. COVER *for ad hoc networks with message loss is* PTIME*-complete.*

Proof. Consider a process definition $\mathcal{P}$. As in Theorem 3 we show that there exists an execution in $AHN_l(\mathcal{P})$ leading to a configuration exposing the control state q *if and only if* there exists an execution in $AHN_i(\mathcal{P})$ leading to a configuration exposing q.

Consider an execution leading to a configuration that exposes the control state q in $AHN_l(\mathcal{P})$. It has the following corresponding execution in $AHN_i(\mathcal{P})$: it is sufficient to mimic **Broadcast** transitions by executing before the broadcast a sequence of **Intermittence** transitions that switch off the nodes that do not receive the message, and by performing after the broadcast the **Intermittence** transitions on the same nodes.

Consider now an execution in $AHN_i(\mathcal{P})$ leading to a configuration that exposes the control state q. This execution can be mimicked in $AHN_l(\mathcal{P})$ simply by assuming that the nodes that are deactivated during a specific phase of the execution in $AHN_i(\mathcal{P})$, lose the messages that are broadcasted in that phase in the corresponding execution in $AHN_l(\mathcal{P})$. □

4.2 Conflict

The second type of failures we consider corresponds to transmission conflicts. Here we consider conflicts due to the contemporaneous emission of messages: if a node has (at least two) neighbors that contemporaneously broadcast a message, then such a node is unable to correctly receive the emitted messages. The modeling of this phenomenon requires a significant modification of the formal semantics. First of all we need to introduce a notion of internal state.

Internal State. The internal state of a node is characterized by the current state according to the process behavior, and by two additional flags indicating whether the node is currently emitting or receiving a message. Formally, given a process $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$ we define the set of states $\mathcal{S} = \big\{[q, x, y] \mid q \in Q, x \in \{\bot\} \cup \Sigma, y \in \{\bot, \texttt{rcv}, \texttt{cnfl}\}\big\}$. The field denoted with x represents whether the node is or is not in a transmission state ($\bot$ means no transmission, while $a \in \Sigma$ denotes transmission of message a). The field y represents whether the node is not receiving ($\bot$) or it is currently receiving correctly a message (`rcv`) or the reception has been damaged due to a conflict (`cnfl`). The initial states are defined as follows: $\mathcal{S}_0 = \{[q, \bot, \bot] \mid q \in Q_0\}$. Notice that nodes in their initial state are neither receiving nor emitting.
The notation based on triples is useful to simplify the definition of the semantics. In the figures we also use a more compact notation without distinction between transmission and reception state, e.g., $[q, \bot, \bot]$ is simplified as q, $[q, a, \bot]$ as $[q, a]$, $[q, \bot, \texttt{rcv}]$ as $[q, \texttt{rcv}]$, etc.
Network Semantics. The semantics of a process $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$ with conflicts is given by the transition system $AHN_{co}(\mathcal{P}) = \langle \mathcal{C}^{co}, \rightrightarrows, \mathcal{C}_0^{co} \rangle$ where $\mathcal{C}^{co}$ is the set of $\mathcal{S}$-graphs and the set of initial configurations $\mathcal{C}_0^{co}$ is the set of $\mathcal{S}_0$-graphs.

Before giving the formal definition of the transition relation $\rightrightarrows \subseteq \mathcal{C}^{co} \times \mathcal{C}^{co}$, we define the function *emitter* which associates to a $\mathcal{S}$-graph $\gamma = \langle V, E, L \rangle$ and to a node $u \in V$, the set $emitter(\gamma, u) = \{v \mid u \sim v$ and $L(v) = [q, a, y]$ for some $a \in \Sigma$ and $y \in \{\bot, \texttt{rcv}, \texttt{cnfl}\}\}$ of nodes adjacent to u in γ which are in a transmission state.

Given a configuration $\gamma = \langle V, E, L \rangle$, we have that $\gamma \rightrightarrows \gamma'$ iff $\gamma' = \langle V, E, L' \rangle$ and one of the following conditions holds:

Local/Time-out: $\exists v \in V$ s.t. $L(v) = [q, \bot, y]$, $y \in \{\bot, \texttt{cnfl}, \texttt{rcv}\}$, $(q, \tau, q') \in R$, $L'(v) = [q', \bot, \bot]$, and $L(u) = L'(u)$ for all $u \in V \setminus \{v\}$;

Start broadcast: $\exists v_1, \ldots, v_l \in V$ s.t. $\cup_{j \in \{1 \ldots l\}} emitter(\gamma, v_j) = \emptyset$, $L(v_i) = [q_i, \bot, \bot]$, $(q_i, !!a_i, q_i') \in R$, $L'(v_i) = [q_i', a_i, \bot]$ $\forall i \in \{1 \ldots l\}$ and the following conditions hold:

- $\forall u \in V \setminus \{v_1, \ldots, v_l\}$ s.t. $u \sim v_i$ for some $i \in \{1 \ldots l\}$ and $L(u) = [r, \bot, y]$ with $y \in \{\texttt{rcv}, \bot\}$ we have:
  - if $y = \texttt{rcv}$ then $L'(u) = [r, \bot, \texttt{cnfl}]$;
  - if $y = \bot$ and $u \not\sim v_j$ $\forall j \in \{1 \ldots l\} \setminus \{i\}$ then $L'(u) = [r, \bot, \texttt{rcv}]$;
  - if $y = \bot$ and $u \sim v_j$ for some $j \in \{1 \ldots l\} \setminus \{i\}$ then $L'(u) = [r, \bot, \texttt{cnfl}]$;

– $L(u) = L'(u)$ otherwise;

End broadcast: $\exists\ v \in V$ s.t. $L(v) = [q, a, \bot]$, $L'(v) = [q, \bot, \bot]$ and we have:

– $\forall u \in V$ s,t. $u \sim v$ and $L(u) = [r, \bot, y]$, with $y \in \{\texttt{rcv}, \texttt{cnfl}\}$, and $emitter(\gamma, u) = \{v\}$ we have:
  - if $y = \texttt{rcv}$ and $\exists\ r'$ s.t. $(r, ??a, r') \in R$ then $L'(u) = [r', \bot, \bot]$;
  - if $y = \texttt{rcv}$ and $\nexists\ r'$ s.t. $(r, ??a, r') \in R$ or $y = \texttt{cnfl}$ then $L'(u) = [r, \bot, \bot]$;

– $L(u) = L'(u)$ otherwise.

The local rule models internal and time-out steps (a node non-deterministically decides to abandon a transmission). In the start rule we select a set of node that have the capability of sending a broadcast and check that no other node in their vicinity is currently transmitting. The selected emitters simultaneously start transmitting. Receiving nodes connected to a single emitter move to the `rcv` state, and to the `cnfl` state in case of connection with more than one emitter (e.g. a selected node and an emitter that started transmitting in a previous step). In the ending rule an emitter moves to its next state. A receiver connected to such a node moves to the next state only if it is still in the `rcv` state (no conflicts occurred in between the start and end phases).

As an example of ad hoc networks and of its semantics in the model with conflicts, consider the process $(S, !!m, T), (R, ??m, Q)$, and the execution in Figure 3. In the initial configuration we have three senders in state S (a, b, c from left to right), and three receivers in state R (d, e, f from left to right). Nodes a and b can simultaneously start transmitting m, since no other node is currently transmitting in their vicinity. Node d simultaneously moves to a conflict state (it is connected to both emitters), while node e moves to a reception state. When c starts transmitting m (again there are no other emitters in its vicinity), node e is forced to enter a conflict state, whereas node f goes to a reception state. When a stops transmitting, d goes back to the original state (a conflict occurred). If now c stops transmitting, f receives the message and moves to its next state Q (no conflicts occurred). Finally when b stops transmitting, e goes back to the original state (a conflict occurred). Other possible executions are obtained, e.g., by selecting only one of the nodes a, b for starting a transmission (the other node has to remain silent since it is connected to an active emitter) and by nondeterministically allowing receiver nodes to abandon a transmission.

Theorem 6. COVER *for ad hoc networks with conflicts is* PTIME*-complete.*

Proof. Consider a process $\mathcal{P}$. Following our usual proof technique, we show that there exists an execution in $AHN_{co}(\mathcal{P})$ leading to a configuration exposing the control state q *if and only if* there exists an execution in $AHN_i(\mathcal{P})$ leading to a configuration exposing q.

It is easy to see that a computation leading to a configuration that exposes the control state q in $AHN_{co}(\mathcal{P})$ has a corresponding computation in $AHN_i(\mathcal{P})$: the **Local** transitions are faithfully reproduced, the **Start broadcast** transitions are not mimicked, and the **End broadcast** transitions are simulated via a protocol

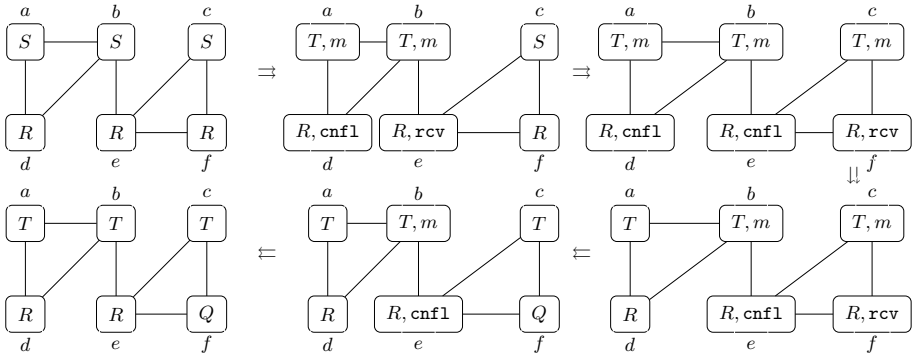

Fig. 3. Example of execution with conflicts

that first turns off the nodes that do not receive the message or that detect a conflict, then executes the broadcast, and then turns on the samenodes.

It is more complex to show that a computation in $AHN_i(\mathcal{P})$ that leads to a configuration that exposes the control state q can be reproduced in $AHN_{co}(\mathcal{P})$.

We first assume, without loss of generality, that in the process $\mathcal{P}$ there is at least one state with an outgoing broadcast transition which is reachable from an initial state $q_0 \in Q_0$ doing only internal steps. If this is not the case, there is no communication in the system and the analysis of COVER can be trivially done by checking whether the target state q is reachable from an initial state in the automaton defining the process behavior doing only internal steps. Consider now the computation in $AHN_i(\mathcal{P})$ that leads to a configuration that exposes the control state q. Let γ_0 be the initial configuration in the considered computation, and let $loss(u)$ be the number of messages that the node u loses during the computation when it was turned off.

We now show the existence of an initial configuration in $AHN_{co}(\mathcal{P})$ able to reproduce such computation. This initial configuration contains γ_0 plus a set of additional nodes used to generate conflicts.

Namely, we connect to each node u of the initial configuration $loss(u)$ additional nodes $Noise(u)$: each node in $Noise(u)$ is connected only with its corresponding node u.

Each node u simulates the behavior of the corresponding node in the computation in $AHN_i(\mathcal{P})$. The nodes in $Noise(u)$ are initially in the state q_0. The simulation of the transitions in the computation in $AHN_i(\mathcal{P})$ is as follows. First of all, for every node u we consider local transitions for nodes in $Noise(u)$ in state q_0 leading them to a state ready to perform a broadcast. Then the transitions are simulated as follows.

- **Local** transitions are faithfully reproduced.
- **Intermittence** transitions are not mimicked.
- To simulate **Broadcast** transitions performed by one node, say v, we proceed as follows: we partition the potential receivers in two groups, (i) those that actually receive the message and (ii) those that do not receive it as they are turned off. For each node u in group (ii) we take an attacker node

$n \in Noise(u)$ ready to start a transmission and let n perform a **Start broadcast** transition. Simultaneously node u moves to the `rcv`-state. Node v performs then a broadcast (it executes both the **Start** and the **End broadcast** transitions). Since u and v are connected, u detects a conflicting transmission and moves to the `cnfl`-state. Finally, node n ends the transmission.
Note that the nodes corresponding to (i) receive the broadcast messages, while those corresponding to (ii) do not receive it, due to the conflict generated by the interferring transmissions generated by the attacker node n.

By assumption on the cardinality of $Nodes(u)$, therefore an attack can be executed every time node u is switched off in the computation with intermittent semantics. □

4.3 Conflict detection

We now define a variant of the semantics in order to capture the notion of conflict detection. In fact, even though a node that receives overlapping signal emissions is unable to reconstruct the emitted messages, it can infer that (at least) two neighbors have contemporaneously emitted their messages. This can be considered in our model of ad hoc networks by adding *conflict detection* transitions to the processes. Such transitions can be executed by nodes at the end of a receive phase during which more than one neighbor has performed a broadcast. Formally, we slightly modify the definition of the *Internal State* and of the *Network Semantics* of the previous section.

Internal State. The new definition of $\mathcal{P}$ is as usual with the unique difference that we can have transitions of the form (q, ρ, q') in R, representing conflict detection (where ρ is a new symbol).

Network Semantics. Given a process $\mathcal{P}$, the transition system $AHN_{cd}(\mathcal{P})$ characterizing the semantics with conflict detection is defined as $AHN_{co}(\mathcal{P})$ except that the **End broadcast** transitions are replaced by the following **End broadcast II** transitions:

End broadcast II: $\exists v \in V$ s.t. $L(v) = [q, a, \bot]$, $L'(v) = [q, \bot, \bot]$ and we have:
- $\forall u \in V$ s.t. $u \sim v$, $L(u) = [r, \bot, y]$, with $y \in \{\texttt{rcv}, \texttt{cnfl}\}$, and $emitter(\gamma, u) = \{v\}$:
  - if $y = \texttt{rcv}$ and $\exists r'$ s.t. $(r, ??a, r') \in R$ then $L'(u) = [r', \bot, \bot]$;
  - if $y = \texttt{cnfl}$ and $\exists r'$ s.t. $(r, \rho, r') \in R$ then $L'(u) = [r', \bot, \bot]$;
  - if $y = \texttt{rcv}$ and $\nexists r'$ s.t. $(r, ??a, r') \in R$, or $y = \texttt{cnfl}$ and $\nexists r'$ s.t. $(r, \rho, r') \in R$, then $L'(u) = [r, \bot, \bot]$;
- $L(u) = L'(u)$ otherwise.

As an example of ad hoc networks and of its semantics with conflict detection, consider the process $(S, !!m, T), (R, ??m, Q), (R, \rho, Er)$, and the execution in Figure 4. It consists of the same steps as those in Figure 3 up to ending phases of broadcast messages. Receiver that detect a conflict move here to the special

Er states. Note that in the step from the fourth to the fifth configuration only the node in the leftmost down corner detects a conflict. The other receiver R is connected to two different emitters, so it will apply the detection only in the next step.

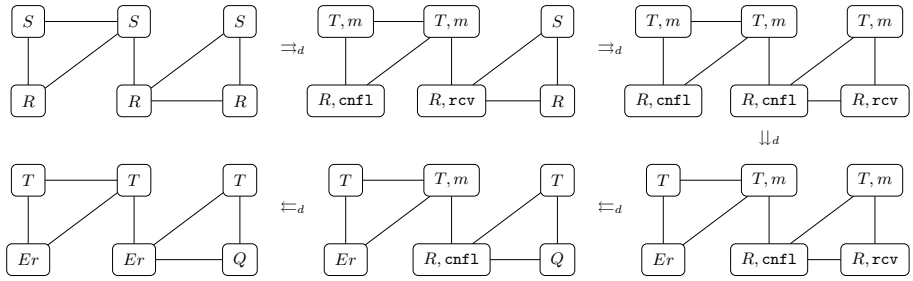

Fig. 4. Example of execution with conflict detections (indicated as $\rightrightarrows_d$)

Theorem 7. COVER *for ad hoc networks with conflict detection is undecidable.*

Proof. The proof is by reduction from the undecidability of COVER for ad hoc networks with node restart (Theorem 4). Consider a process $\mathcal{P} = \langle Q, \Sigma, R, Q_0, q_r \rangle$ for ad hoc networks with node restart (q_r being the restart state). Consider now the process $\mathcal{P}' = \langle Q \cup \{q_i\}, \Sigma, R', Q_0 \rangle$, for ad hoc networks with conflict detection, defined as $\mathcal{P}$ with the following additional transitions: for each node $q \in Q$ we have a transition labeled with ρ leading to the additional state q_i, from which there is only one outgoing transition labeled with τ leading to the restart state q_r.

We first show that given a computation in $AHN_r(\mathcal{P})$ leading to a configuration that exposes the control state q, there exists a corresponding computation in $AHN_{cd}(\mathcal{P}')$. As in Theorem 6 we make the nonrestrictive assumption that in the process $\mathcal{P}$ there is at least one state with an outgoing broadcast transition which is reachable from an initial state $q_0 \in Q_0$ doing only internal steps. Let γ be the initial configuration of the considered computation in $AHN_r(\mathcal{P})$. For each node u in γ we denote with $restart(u)$ the number of restarts performed by u during the computation. We now show the existence of an initial configuration γ' of $AHN_{cd}(\mathcal{P}')$ from which the computation is simulated. The configuration γ' is as γ with the difference that each node u has exactly $restart(n) \times 2$ additional neighbors that are used to generate conflicts. These additional nodes are connected only to the corresponding node u. The simulation of the computation proceeds as follows. At the beginning the additional nodes in state q_0 perform the local transitions leading them to a state ready to perform a broadcast. Then the simulation starts.

– **Local** transitions are reproduced faithfully.

– A transition that deactivates the node u is simulated via the following protocol: two of the additional nodes connected to u perform a **Start broadcast** transition and then execute the **End broadcast II**. Due to the emission conflict, the node u moves to the internal state q_i.
– A transition that activates the node u is reproduced by an internal transition from the state q_i of u to the restart state q_r.
– Finally, **Broadcast** transitions are mimicked by performing in sequence a **Start** and an **End broadcast II** transition.

We now show that a computation in $AHN_{cd}(\mathcal{P}')$ leading to a configuration that exposes the control state q has a corresponding computation in $AHN_r(\mathcal{P})$. In the simulated computation the **Local** transitions are reproduced faithfully, the **Start broadcast** transitions are not mimicked, while **End broadcast II** transitions are simulated by the following protocol.

Assume that the node that completes its signal emission in the **End broadcast II** transition is u, and let a be the emitted message. The neighbors of u able to receive a can be partitioned in three groups:

(i) those that correctly receive message a,
(ii) those that perform a conflict detection transition during the execution of the **End broadcast II** transition,
and (iii) those that do not change their internal state because they are still under the effect of another signal emission.

The simulation of the transition in $AHN_r(\mathcal{P})$ proceeds as follows. The nodes, corresponding to those in (ii) and (iii), that are not currently crashed perform a **Crash** transition, then the **Broadcast** transition is executed. Notice that at the end of this protocol the nodes in (ii) are in the intermediary state q_i in the computation in $AHN_{cd}(\mathcal{P}')$, while they are crashed in the corresponding computation in $AHN_r(\mathcal{P})$. The **Local** transitions that move the nodes form the state q_i to q_r are reproduced in $AHN_r(\mathcal{P})$ by **Restart** transitions. □

5 Conclusion

In this paper we have compared different types of semantics for modelling unreliability in protocols based on broadcast communication. The comparison is based on the study of decidability and undecidability of the coverability problem (reachability of a network with at least a node in an error state for an initial configuration of unknown size and shape). Coverability is commonly used to formulate violations of properties like mutual exclusion (and more in general to locally reason on errors generated by a fixed set of processes independently from the global configuration). Coverability turns out to be undecidable for models in which individual nodes have special transition to the detect the occurrence of a failure (e.g. crash with restart, conflict detection). Removing this feature from the model completely change the corresponding expressive power, often making coverability decidable. Decidability results are obtained by means of reduction to a coverability in a model with spontaneous movement, for which we

have given a PTIME algorithm in [2]. Among possible future direction we plan to investigate the impact of node and communication failures in richer models of broadcast communication that could be used to model for instance routing strategy or time division protocols.

References

1. Abdulla, P. A., Jonsson, B.: Verifying programs with unreliable channels. Inf. Comput. 127(2): 91–101 (1996)
2. Delzanno, G., Sangnier, A., Traverso, R.,Zavattaro, G.: Reachability Problems in Mobile Ad Hoc Networks. Technical report available on arXiv.
3. Delzanno, G., Sangnier, A., Zavattaro, G.: Parameterized verification of Ad Hoc Networks. CONCUR '10: 313–327
4. Delzanno, G., Sangnier, A., Zavattaro, G.: On the Power of Cliques in the Parameterized verification of Ad Hoc Networks. FOSSACS '11: 441–455
5. Ene, C., Muntean, T.: A broadcast based calculus for Communicating Systems. IPDPS '01: 149
6. Esparza, J., Finkel, A., Mayr, R.: On the verification of Broadcast Protocols. LICS '99: 352–359
7. Fehnker, A., van Hoesel, L., Mader, A.: Modelling and verification of the LMAC protocol for wireless sensor networks. IFM '07: 253–272
8. Godskesen, J.C.: A calculus for Mobile Ad Hoc Networks. Coordination '07: 132–150
9. Ladner, R. E.: The circuit value problem is logspace complete for P. SIGACT News: 18–20, 1977
10. Merro, M.: An observational theory for Mobile Ad Hoc Networks. Inf. Comput. 207(2): 194–208 (2009)
11. Merro, M., Ballardin, F., Sibilio, E. A Timed Calculus for Wireless Systems FSEN '09: 228-243, 2010.
12. Lanese, I., Sangiorgi, D.: An operational semantics for a calculus for wireless systems. TCS, 411(19): 1928-1948 (2010)
13. Nanz, S., Hankin, C.: A Framework for security analysis of mobile wireless networks. TCS, 367(1–2):203-227 (2006)
14. Prasad, K.V.S.: A Calculus of Broadcasting Systems. SCP, 25(2–3): 285–327 (1995).
15. Saksena, M., Wibling, O., Jonsson, B.: Graph grammar modeling and verification of Ad Hoc Routing Protocols. TACAS '08: 18–32
16. Singh, A., Ramakrishnan, C. R., Smolka, S. A.: A process calculus for Mobile Ad Hoc Networks. COORDINATION '08: 296–314
17. Singh, A., Ramakrishnan, C. R., Smolka, S. A.: Query-Based model checking of Ad Hoc Network Protocols. CONCUR '09: 603–619