# Decidability and Complexity Results for Verification of Asynchronous Broadcast Networks

Giorgio Delzanno and Riccardo Traverso

DIBRIS, Università di Genova
via Dodecaneso 35, 16146 Genova, Italy
{Giorgio.Delzanno,Riccardo.Traverso}@unige.it

**Abstract.** We study decidability and complexity of verification problems for networks in which nodes communicate via asynchronous broadcast messages. This type of communication is achieved by using a distributed model in which nodes have a local buffer. We consider here safety properties expressed as a coverability problem with an arbitrary initial configuration. This formulation naturally models the search of an initial topology that may lead to an error state in the protocol.

**Keywords:** Infinite-state Systems, Broadcast Protocols, Verification.

## 1 Introduction

We present (un)decidability and complexity results for the coverability problem of Asynchronous Broadcast Networks (ABN), a mathematical model of distributed systems in which processes interact via topology-dependent and asynchronous communication. Our formal model of asynchronous broadcast communication combines three main features: a graph representation of a network configuration decoupled from the specification of individual process behaviour, a topology-dependent semantics of synchronization, and the use of local mailboxes to deliver messages to individual nodes. Our main abstraction comes from considering protocols defined via a communicating finite-state automaton replicated on each node of the network.

In our setting the coverability problem is formulated as follows. We first define an initial configuration as any graph in which nodes have labels that represent the initial state of the protocol (and no constraints on edges). Coverability consists then in checking whether there exists an initial configuration that can reach a target configuration that contains a given process state. A similar decision problem is considered in [8] for a mathematical model with synchronous communication and dynamic reconfiguration of the topology called Reconfigurable Broadcast Network (RBN).

Our analysis is carried out with different policies to handle buffers, namely unordered bags (an abstraction of a tuple space), and perfect or lossy FIFO channels. Our technical contribution is as follows. We first show that, in contrast with the synchronous case discussed in [9,10], coverability is decidable when

A.-H. Dediu, C. Martín-Vide, and B. Truthe (Eds.): LATA 2013, LNCS 7810, pp. 238–249, 2013.

local buffers are unordered. For the proof, we first give a reduction to the restricted case of fully connected topologies. We then solve the coverability problem through a reduction to the Cardinality Reachability Problem for Reconfigurable Broadcast Networks, a PTIME-complete problem [8]. The resulting algorithm is based on a forward labelling procedure described in detail in [8].

When mailboxes are ordered buffers, we obtain undecidability already in the case of fully connected topologies. The undecidability proof is based on a non-trivial encoding of the set of operations of a two counter machine in form of a cooperation protocol between distinct nodes. The protocol consists of different phases, each one is defined over a distinct set of control messages. The difficulty of the encoding comes from the fact that it is not possible to infer well-formedness properties for the content of the mailbox of an individual node. Thus it is not possible to encode the current value of the counters using the current content of a set of mailboxes. The current value of the counters is represented however in the flow of messages consumed by a pair of nodes elected in a preliminary phase of the protocol, which, in turn, completes successfully only under certain conditions on the sequence of consumed control messages. The coverability problem is decidable when introducing non-deterministic message losses. The results again follows from a reduction to the Cardinality Reachability Problem for RBN.

In an extended model in which a node can test if its mailbox is empty, we obtain undecidability with unordered bags and both arbitrary or fully-connected topologies. For this reduction we need to control the interferences due to the simultaneous communication with several neighbours. We exploit here the emptiness test in order to enforce the well-formedness of the mailboxes of nodes involved in the simulation of counter machines.

To our knowledge, the present work shows the first complexity analysis for (parameterized) coverability in formal models of asynchronous broadcast communication.

Detailed proofs and encodings are presented in the technical report [13].

## 2 Asynchronous Broadcast Network (ABN)

In this section we formally define our asynchronous model for broadcast communication. A configuration is defined as a labelled graph. Nodes correspond to processes running a common, pre-defined protocol. Each node has a local message buffer used to collect messages sent by neighbours.

A protocol is specified via a finite-state automaton with send and receive operations that correspond to write [resp. read] on remote [resp. local] buffers. Communication is topology-dependent, anonymous and asynchronous: when a process at node $n$ sends a message $a$, the process does not block, and the message is added to the local mailbox of all of its neighbours without explicit information about the sender (i.e. messages do not contain node identifiers).

Formally, we consider a finite set $\Sigma$ of messages, and different disciplines for handling the mailbox (message buffer), e.g., unordered mailboxes that we represent as bags over $\Sigma$, and ordered mailboxes that we represent as words over $\Sigma$.

In order to deal in a uniform way with different mailbox types we define a transition system parametric on the data structures used to model mailboxes. More specifically, we consider a mailbox structure $\mathbb{M} = \langle \mathcal{M}, del?, add, del, [] \rangle$, where $\mathcal{M}$ is a denumerable set of elements denoting possible mailbox contents; for $a \in \Sigma$ and $m \in \mathcal{M}$, $add(a, m)$ denotes the mailbox obtained by adding $a$ to $m$, $del?(a, m)$ is true if $a$ can be removed from $m$; $del(a, m)$ denotes the mailbox obtained by removing $a$ from $m$ when possible, undefined otherwise. Finally, $[] \in \mathcal{M}$ denotes the empty mailbox. We call an element $a$ of $m$ *visible* when $del?(a, m) = true$. Their specific semantics and corresponding properties change with the type of mailbox considered.

**Definition 1.** *A protocol is defined by a process $\mathcal{P} = \langle Q, \Sigma, R, q_0 \rangle$, where $Q$ is a finite set of control states, $\Sigma$ is a finite message alphabet, $Act = \{\tau\} \cup \{!!a, ??a \mid a \in \Sigma\}$, $R \subseteq Q \times Act \times Q$ is a set of transition rules, $q_0 \in Q$ is an initial control state.*

The label $\tau$ represents the capability of performing an internal action, and the label $!!a$ [$??a$] represents the capability of broadcasting [receiving] a message $a \in \Sigma$.

**Definition 2.** *Configurations are undirected $(Q \times \mathcal{M})$-graphs. A $(Q \times \mathcal{M})$-graph $\gamma$ is a tuple $\langle V, E, L \rangle$, where $V$ is a finite set of nodes, $E \subseteq V \times V$ is a finite set of edges (such that $E$ is symmetric and $\forall v \in V.(v, v) \notin E$), and $L : V \to (Q \times \mathcal{M})$ is a labelling function.*

In the rest of the paper, 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 one another in $\gamma$. We omit $\gamma$, and simply write $u \sim v$, when it is made clear by the context. We use $L(\gamma)$ to represent the set of labels in $\gamma$. The set of all configurations is denoted $\Gamma$, while $\Gamma_0 \subseteq \Gamma$ is the set of all initial configurations, in which nodes always have the same label $\langle q_0, [] \rangle$.

Given the labelling $L$ and the node $v$ s.t. $L(v) = \langle q, m \rangle$, we define $L_s(v) = q$ (state component of $L(v)$) and $L_b(v) = m$ (buffer component of $L(v)$). Furthermore, for $\gamma = \langle V, E, L \rangle \in \Gamma$, we use $L_s(\gamma)$ to denote the set $\{L_s(v) \mid v \in V\}$.

**Definition 3.** *For $\mathbb{M} = \langle \mathcal{M}, del?, add, del, [] \rangle$, an Asynchronous Broadcast Network (ABN) associated to $\mathcal{P}$ is a tuple $\mathcal{T}(\mathcal{P}, \mathbb{M}) = \langle \Gamma, \Rightarrow_{\mathbb{M}}, \Gamma_0 \rangle$, where $\Rightarrow_{\mathbb{M}} \subseteq \Gamma \times \Gamma$ is the transition relation defined next. For $\gamma = \langle V, E, L \rangle$ and $\gamma' = \langle V, E, L' \rangle$, $\gamma \Rightarrow_{\mathbb{M}} \gamma'$ holds iff one of the following conditions on $L$ and $L'$ holds:*

**Local.** *There exists $v \in V$ such that $(L_s(v), \tau, L'_s(v)) \in R$, $L_b(v) = L'_b(v)$, and $L(u) = L'(u)$ for each $u \in V \setminus \{v\}$.*

**Broadcast.** *There exists $v \in V$ and $a \in \Sigma$ such that $(L_s(v), !!a, L'_s(v)) \in R$, $L_b(v) = L'_b(v)$ and for every $u \in V \setminus \{v\}$*

- *if $u \sim v$ then $L'_b(u) = add(a, L_b(u))$ and $L_s(u) = L'_s(u)$,*
- *otherwise $L(u) = L'(u)$.*

**Receive.** *There exists* $v \in V$ *and* $a \in \Sigma$ *such that* $(L_s(v), ??a, L'_s(v)) \in R$, $del?(a, L_b(v))$ *is satisfied,* $L'_b(v) = del(a, L_b(v))$, *and* $L(u) = L'(u)$ *for each* $u \in V \setminus \{v\}$.

A local transition only affects the state of the process that executes it, while a broadcast also adds the corresponding message to the mailboxes of all the neighbours of the sender. Notice that broadcast is never blocking for the sender. Receivers can read the message in different instants. This models asynchronous communication. A reception of a message $a$ is blocking for the receiver whenever the buffer is empty or the visible elements are all different from $a$. If $a$ is visible in the mailbox, the message is removed and the process moves to the next state. Furthermore, it is easy to show that, when needed, a set $Q_0 \subseteq Q$ of initial states for $\mathcal{P}$ can be modelled by introducing a fresh initial state with outgoing local transitions to each $q \in Q_0$.

An *execution* is a sequence $\gamma_0\gamma_1 \ldots$ such that $\gamma_0$ is an initial configuration, and $\gamma_i \Rightarrow_{\mathbb{M}} \gamma_{i+1}$ for $i \geq 0$. We use $\Rightarrow^*_{\mathbb{M}}$ to denote the reflexive and transitive closure of $\Rightarrow_{\mathbb{M}}$. We drop $\mathbb{M}$ when the mailbox type is clear from the context.

**Decision Problem.** The *Coverability Problem* parametric on the mailbox structure $\mathbb{M}$, abbreviated as $COV(\mathbb{M})$, is defined as follows.

**Definition 4.** *Given a protocol* $\mathcal{P}$ *with transition system* $\mathcal{T}(\mathcal{P}, \mathbb{M}) = \langle \Gamma, \Rightarrow_{\mathbb{M}}, \Gamma_0 \rangle$ *and a control state* $q$, *the coverability problem* $COV(\mathbb{M})$ *states: are there two configurations* $\gamma_0 \in \Gamma_0$ *and* $\gamma_1 \in \Gamma$ *such that* $\gamma_0 \Rightarrow^*_{\mathbb{M}} \gamma_1$ *and* $q \in L_s(\gamma_1)$*?*

In other words we require that a graph $\gamma_q$ with a singleton node labelled $q$ covers a reachable configuration $\gamma_1$, i.e., $\gamma_q$ is a subgraph of $\gamma_1$. We often use the terminology $\gamma_0$ reaches state $q$ as an abbreviation for $\gamma_0 \Rightarrow^*_{\mathbb{M}} \gamma_1$ and $q \in L_s(\gamma_1)$ for some configuration $\gamma_1$. Besides being parametric on the mailbox structure, our decision problem is parametric on the shape of the initial configuration. As mentioned in the introduction, this feature models in a natural way verification problems for protocols with partial information about the structure of the network.

### 2.1 ABN vs RBN

In the rest of the paper we will often refer to the semantics of RBN models [8]. Protocols in RBN adhere to the same syntax as ABN. Configurations are simply $Q$-graphs, i.e., graphs in which nodes have labels in $Q$ via the labelling function $L$. The semantics of broadcast communication however is synchronous instead of asynchronous. Furthermore, the topology of the network may nondeterministically change. Formally, given $R_a(q) = \{q' \in Q \mid \langle q, ??a, q' \rangle \in R\}$ and two $Q$-graphs $\theta, \theta'$ with $\theta = \langle V, E, L \rangle$, we have $\theta \to \theta'$ iff $\theta' = \langle V, E', L' \rangle$ and one of the following conditions holds:

**Synch Broadcast.** $E' = E$ and $\exists v \in V$ s.t. $\langle L(v), !!a, L'(v) \rangle \in R$ and $L'(u) \in R_a(L(u))$ for every $u \sim v$, and $L(w) = L'(w)$ for any other node $w$.

**Reconfiguration.** $E' \subseteq V \times V \setminus \{\langle v, v \rangle \mid v \in V\}$ and $L = L'$.

## 3 Unordered Mailboxes

In this section we study the coverability problems for ABNs in which mailboxes are unordered buffers modelled as bags over the finite message alphabet $\Sigma$. The mailbox structure $Bag$ is defined as follows: $\mathcal{M}$ is the denumerable set of bags over $\Sigma$, $add(a, m) = [a] \oplus m$ (multiset sum of the singleton $[a]$ and $m$), $del?(a, m) = true$ iff $m(a) > 0$, $del(a, m) = m \ominus [a]$ (multiset removal of $[a]$ from $m$), and $[] \in \mathcal{M}$ is the empty bag $[]$. The operational semantics follows from the general definitions.

Let us consider the instance $COV(Bag)$ of the coverability problem. For synchronous broadcast, coverability is undecidable for arbitrary topologies [9]. We show next that coverability is in PTIME for unordered mailboxes.

For the ease of notation, we use $\mathcal{T}^{\mathcal{K}}(\mathcal{P}, \mathbb{M})$ [resp. $COV_{fc}(\mathbb{M})$] to denote the restriction of $\mathcal{T}(\mathcal{P}, \mathbb{M})$ [resp. $COV(\mathbb{M})$] to fully connected configurations only, i.e., configurations such that $u \sim_\gamma v$ for each pair of distinct nodes $u, v \in V$. We prove the results in two different steps. We first show that, for the purpose of deciding $COV(Bag)$, we can focus on fully connected topologies only. We then show a reduction from $COV_{fc}(Bag)$ to the Cardinality Reachability Problem for Reconfigurable Broadcast Networks, that, for short, we will refer to as CRP. The reduction requires reachability queries of the form $\#q \geq 1$ (at least one occurrence of control state $q$). The latter problem is PTIME-complete [8].

For asynchronous communication with unordered mailboxes, coverability for arbitrary topologies case can be reduced to the fully connected case. The following lemma indeed holds.

**Lemma 5.** *Given an ABN protocol $\mathcal{P} = \langle Q, \Sigma, R, q_0 \rangle$ and a state $q \in Q$, if there exists an arbitrary topology from which we can reach state $q$, then there exists a fully connected topology from which we can also reach $q$.*

One side of the property is immediate. If there exists a fully connected initial configuration that reaches a configuration in which state $q$ occurs, then coverability is solved. In order to prove the other implication, the intuition is that we can exploit the fact that mailboxes are unordered to ignore messages sent along links that are not present in a given topology.

The following lemma relates coverability in ABN to the cardinality reachability problem in RBN.

**Lemma 6.** *Given an ABN protocol $\mathcal{P} = \langle Q, \Sigma, R, q_0 \rangle$ and a state $q \in Q$ let $\mathcal{P}'$ be the RBN protocol with the same rules but with $\{q_0\}$ as singleton set of initial states. Then, there exists an execution of $\mathcal{P}'$ that satisfies CRP if and only if there exists an execution of $\mathcal{P}$ satisfying $COV_{fc}(Bag)$.*

In the proof we can delay message receptions to simulate deletions of links. Vice versa, we can exploit reconfigurations and the possibility of adding nodes to the initial configuration to simulate asynchronous receipts using dynamically created links and synchronous messages. The previous reduction is done in constant time, since there is no need of modifying the protocol specification. We can therefore conclude that the following property holds.

**Theorem 7.** *$COV(Bag)$ is* PTIME*-complete.*

*Proof.* Thanks to Lemmas 5 and 6 and to the PTIME algorithm for coverability in RBN [8], we know that $COV(Bag)$ is in PTIME. Completeness follows from a reduction of the Circuit Value Problem (CVP) [19] to $COV(Bag)$. Given an acyclic circuit $G$ composed by a finite set of gates and a fixed evaluation of its inputs, CVP consists in evaluating $G$ in the inputs. The reduction is based on a protocol in which a special node broadcasts the evaluation of a single input (in form of a message with label $true/false$ and an index associated to the corresponding variable). Gates (i.e. Boolean operations like and/or/not) are simulated by processes running on nodes. For each gate, we have nodes that receive the inputs, evaluate the gate, and broadcast their output to the other nodes. A special node intercepts the *true* message corresponding to the output of the whole circuit and moves in an acceptance state. Regardless the type of communication topology, number of nodes simulating each gate, and possible delays, coverability of the acceptance state corresponds to satisfiability of the circuit $G$ w.r.t. the given assignment. □

## 4 FIFO Mailboxes

In this section we move to ABN with perfect FIFO buffers as communication media. In this context we instantiate the mailbox structure $FIFO$ as follows: $\mathcal{M}$ is defined as $\Sigma^*$; $add(a, m) = m{\cdot}a$ (concatenation of $a$ and $m$); $del?(a, m) = true$ iff $m = a{\cdot}m'$; $del(a, m)$ is the string $m'$ whenever $m = a{\cdot}m'$, undefined otherwise; finally, $[] \in \mathcal{M}$ is the empty string $\epsilon$.

**Theorem 8.** *$COV(FIFO)$ and $COV_{fc}(FIFO)$ are undecidable.*

*Proof.* The proof is based on a reduction of the halting problem for two-counter machines – a well known undecidable problem – to $COV(FIFO)$. A two-counter machine is defined by a pair $\langle Loc, Inst\rangle$ where $Loc$ is a finite set of control locations and $Inst \subseteq Loc \times Op \times Loc$ is a finite set of instructions such that $Op = \{c{+}{+}, c{-}{-}, c == 0 \mid c \in \{x_1, x_2\}\}$ is a set of operators over the counters $x_1$ and $x_2$, and $\ell_0 \in Loc$ is the initial location. Configurations are tuples $\langle \ell, v_1, v_2\rangle$ such that $\ell \in L$ is the current location and $v_1, v_2$ are natural numbers that denote the current value of $x_1$ and $x_2$, respectively. The operational semantics is defined in a standard way: the execution of increment and decrement updates the control location and the current value of the corresponding counter, a zero-test updates the location whenever the test is satisfied in the current state of the counter.

The rationale behind the reduction of coverability to the halting problem of two-counter machines is as follows. We first use an election protocol that assigns fixed roles (controller/slave) to a pair of adjacent nodes. Since the initial configuration is not fixed a priori our election protocol does not forbid the election of multiple pairs of controller/slave nodes, but we only require that at least one pair is elected in order to succeed. The controller/slave nodes set up their mailboxes in order to use them as overlapping circular queues. Messages represent

the current value (in unary) of the counters. The simulation is guided by the controller. The slave forwards all received messages back to the controller. As an example, to check that $x_1$ is zero, the controller reads all messages in the mailbox and checks that in between two successive reads of the marker for $x_1$ there are no units. We use interference to denote an unwanted message occurring in the mailbox of a controller/slave node. Since the network topology is not fixed a priori, a key point of the whole construction is the capability of controlling interferences with other nodes, e.g., avoiding the adjacency between multiple controllers and slaves. For this purpose, we use special control messages to coordinate the different phases and exploit the FIFO mailboxes in order to enforce the simulation to get into a deadlock state whenever the same control message is received more than once. A detailed description of the protocol is in [13]. The same construction can be used for the fully connected case. □

## 5 Lossy FIFO Mailboxes

We now consider coverability for ABNs in which mailboxes are lossy FIFO channels, i.e., channels in which messages may non-deterministically be lost. Given a protocol $\mathcal{P}$, a configuration $\gamma$ of $\mathcal{T}^{\mathcal{K}}(\mathcal{P}, LFIFO)$ is a multiset of pairs $\langle q, m\rangle$ where $q \in Q$ and $m \in \Sigma^*$. To model non-deterministic loss of messages, we modify the operational semantics by introducing lossy steps.

We first need to define the ordering $\preccurlyeq$ between configurations. For $\gamma = \langle V, V \times V, L\rangle$ and $\gamma' = \langle V', V' \times V', L'\rangle$ $\gamma \preccurlyeq \gamma'$ iff there exists an injection $h : V \to V'$ s.t. $L_s(v) = L_s(h(v))$ and $L_b(v) \prec L_b(h(v))$ for each $v \in V$, where $\prec$ denotes the subword relation, namely, for $w, w' \in \Sigma^*$, $w \prec w'$ iff there exists an injective and strictly monotone mapping $h : |w| \to |w'|$ s.t. $w_i = w'_{h(i)}$ for $i : 1, \ldots, |w|$, where $v_i$ denotes the $i$-th symbol in the word $v$. Intuitively, $\gamma \preccurlyeq \gamma'$ means that $\gamma$ is obtained from $\gamma'$ by removing nodes (and all corresponding edges) and messages from the buffers. We modify the transition relation $\Rightarrow$ to include lossy steps before and after each transition in the original system as follows: $\gamma \longmapsto \gamma'$ iff there exists $\eta$ and $\nu$ s.t. $\eta \preccurlyeq \gamma$, $\eta \Rightarrow \nu$, and $\gamma' \preccurlyeq \nu$.

The ordering $\preccurlyeq$ is a simulation relation and is also a well-quasi ordering. These two properties pave the way for a possible application of the theory of well-structured transition systems [12] to solve coverability. In the rest of the section we use a reduction to RBN-coverability to obtain better complexity results. As for unordered mailbox we first show that we can focus our attention on fully connected topologies, only.

**Lemma 9.** *There exists an execution of $\mathcal{P}$ that satisfies $COV(LFIFO)$ if and only if there is one of $\mathcal{P}$ satisfying $COV_{fc}(LFIFO)$.*

We are now at the most tricky part of the proof that consists in proving that $COV_{fc}(LFIFO)$ can be reduced to CRP. Let $\mathcal{P}$ be an ABN protocol, and let $\mathcal{P}'$ be the corresponding RBN protocol derived as in Section 3.

**Lemma 10.** *There exists an execution of $\mathcal{P}'$ that satisfies CRP if and only if there is one of $\mathcal{P}$ satisfying $COV_{fc}(LFIFO)$.*

The coverability problem for lossy FIFO mailboxes has a property in common with the one for bags, that is in both cases processes are able to ignore incoming messages indefinitely; this is achieved by either leaving the message in the multiset or by deleting it from the lossy FIFO queue. We can therefore take again advantage of this property to obtain the following theorem.

**Theorem 11.** *$COV_{fc}(LFIFO)$ is* PTIME*-complete.*

*Proof.* Membership to PTIME follows from the reduction to CRP for RBNs. Hardness follows again from a reduction of CVP to COV for ABN lossy FIFO queues. The encoding protocol is the same as for unordered mailboxes. □

## 6 ABN with Emptiness Test

In this section we enrich the ABN model with a new type of transitions in order to enable nodes to test whether their mailbox is empty. We call the resulting model $\text{ABN}_\epsilon$. The set $Act$ of action labels is extended to include $\epsilon$, i.e., $Act = \{\tau, \epsilon\} \cup \{!!a, ??a \mid a \in \Sigma\}$. The transition systems associated to an $\text{ABN}_\epsilon$ are changed accordingly to take $\epsilon$ into account; given two configurations $\gamma = \langle V, E, L\rangle$ and $\gamma' = \langle V, E, L'\rangle$, $\gamma \Rightarrow \gamma'$ holds also if the following condition is met.

**Emptiness Test.** There exists a $v \in V$ such that $(L_s(v), \epsilon, L'_s(v)) \in R$, $L_b(v) = L'_b(v) = []$, and $L(u) = L'(u)$ for each $u \in V \setminus \{v\}$.

The only difference w.r.t. the semantics of $\tau$-transitions consists in the $L_b(v) = []$ condition, that ensures that $\epsilon$-transitions only fire when the mailbox is empty.

The introduction of $\epsilon$-transitions affects the different instances of the coverability problem in different ways. The simplest case is for $COV_{fc}(FIFO)$ and $COV(FIFO)$, which of course are still undecidable: the possibility to test the emptiness of the mailbox does not have any effect on the reduction from two-counter machines. The reduction from $COV_{fc}(LFIFO)$ to CRP of Lemma 10 has to be modified in order to consider also $\epsilon$-transitions. Given two configurations $\gamma, \gamma' \in \Gamma$ such that $\gamma \preccurlyeq \gamma'$ (see Section 5 for the definition of the $\preccurlyeq$ ordering), if $\epsilon$ is enabled in $\gamma$ then it can be fired starting from $\gamma'$ too, through a preliminary lossy step that empties the relevant mailbox. This means that $\epsilon$-transitions are almost the same as internal transitions in case of $LFIFO$ mailboxes. Therefore, given a protocol $\mathcal{P} = \langle Q, \Sigma, R, q_0\rangle$ and a target state $q \in Q$, we derive an RBN protocol $\mathcal{P}' = \langle Q, \Sigma, R', \{q_0\}\rangle$ where $R'$ is the set of rules $R$ where all occurrences of $\epsilon$ have been replaced by $\tau$, and then we solve CRP for the target state $q$. Thanks to the previously mentioned property of $\epsilon$-transitions, one could adapt easily enough the proof of Lemma 10 to this case. From these observations we can therefore derive that both $COV(LFIFO)$ and $COV_{fc}(LFIFO)$ are decidable even with $\epsilon$-transitions.

We incur in a completely different case when considering bags: as it can be shown, the extended semantics traces indeed a sharp boundary between decidability and undecidability. Without the emptiness test, both reachability problems $COV(Bag)$ and $COV_{fc}(Bag)$ are decidable; we prove that the operator $\epsilon$

introduced with the extended model is sufficient to make them undecidable. The proof proceeds by building a reduction from the control state reachability problem for two-counter machines to $COV(Bag)$. The reduction encodes a counter machine $\mathcal{M}$ with an ABN protocol $\mathcal{P} = \langle Q, \Sigma, R, q_0 \rangle$ where, like before, each location $\ell \in Loc$ and each instruction $i \in Inst$ corresponds respectively to a state $\mathcal{P}(\ell) \in Q$ and to a set of intermediate states and rules. The protocol is split in two phases. In the first phase processes follow a distributed election protocol to identify who takes care of which role and who is excluded from the simulation. The second phase is the simulation of $\mathcal{M}$. The alphabet is partitioned in two sets, $\Sigma_e$ for the election and $\Sigma_s$ for the simulation. Since we do not make any particular assumption on the connectivity graph, the proof works for both $COV_{fc}(Bag)$ and $COV(Bag)$.

*Election.* A simulation must be carried out by three nodes: a controller and two slaves, one per counter. Figure 1 shows the protocol used to choose such roles. We say that a node is *in simulation* if it reaches (at least once) $\mathcal{P}(\ell_0)$, $q_{S_1}$, or $q_{S_2}$. The election guarantees minimal connectivity requirements, as stated in the following Lemma.

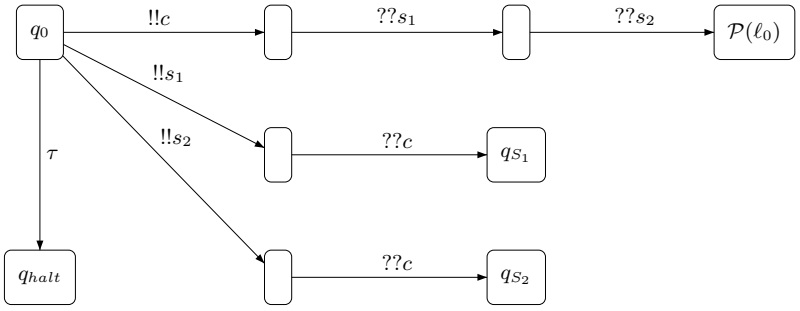


**Fig. 1.** $COV(Bag)$: Election protocol

**Lemma 12.** *If a node is in state $\mathcal{P}(\ell_0)$, then at least two of its neighbours are already respectively in state $q_{S_1}$ and $q_{S_2}$ or they can possibly move only those states. If a node is in state $q_{S_1}$ or $q_{S_2}$, then at least one of its neighbours is already in state $\mathcal{P}(\ell_0)$ or it can possibly move only to $\mathcal{P}(\ell_0)$.*

*Simulation.* Each slave $S_j$ keeps in its mailbox a number of $u_j$ messages equal to the current value of counter $x_j$. The controller sends messages $sub_j$ or $tz_j$ to give orders depending on the instruction $(\ell, op, \ell')$ that is going to be simulated by the system and waits for the slave which manages the involved counter to react accordingly (see Figure 2). Once the slave is done, the same control message is sent back to the controller as acknowledgement and the controller is able to proceed. When $A$ is a set we write $??A$ to mean that for every $a \in A$ the protocol has a reception rule $??a$ with the same endpoints. Again, the increment can be done directly by the controller with a single broadcast $!!u_j$. In order to be able

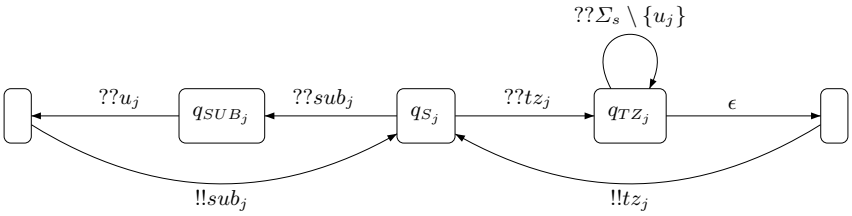


**Fig. 2.** $COV(Bag)$: Slave process

to prove the correctness of the reduction, we first state some properties of the simulation phase.

**Lemma 13.** *Any $m \in \Sigma_e$ received by a node in simulation will persist in its mailbox forever. Such a node is said to be* in interference*.*

*Proof.* By construction, for all $m \in \Sigma_e$, there are no receptions of $m$ starting from any state which may be reached by simulating nodes. □

**Lemma 14.** *At any time, the value of the counter $i$ is equal to the number of occurrences of units $u_i$ in the mailbox of the corresponding slave, provided that no simulating node is in interference. We say then that the counters are* valid*.*

We remark that the notion of validity of the counters does not have anything to do with the compliance of their values w.r.t. the ones of the two-counter machine being simulated. Moreover, since the simulation may proceed even with invalid counters, the reduction does not compute reachability of the encoding $\mathcal{P}(\ell_f)$ of the target state $\ell_f$, but instead it checks for the reachability of a fresh state $q_{target}$ added according to Figure 3. This is needed in order to ensure the correctness of the simulation. It is straightforward to check that the instructions added to

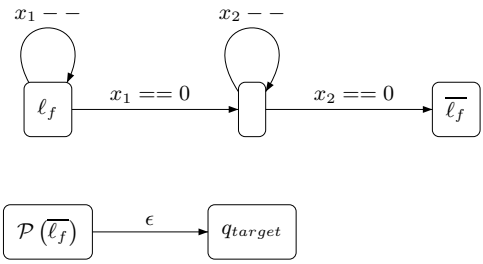


**Fig. 3.** $COV(Bag)$: Interference detection

$\mathcal{M}$ do not have any impact on the reachability of the target location, as they just decrement down to zero both counters before reaching the destination. We are now ready to prove that the reduction is indeed a correct simulation of the given two-counter machine.

**Theorem 15.** *$COV(Bag)$ [$COV_{fc}(Bag)$] is undecidable in $ABN_\epsilon$.*

The correctness of the reduction can be proved by first demonstrating by induction on the number of simulated instructions that for any number of steps, either the counters will be valid and consistent w.r.t. the corresponding state of the two-counter machine or they will be (and remain) invalid. Given this property, we can exploit Lemma 13 in order to show that the added, final transitions from Figure 3 ensure that the controller will deadlock before reaching the target state when the counters are invalid. A detailed description of the protocol is in [13].

## 7 Related Work

Formal models of broadcast protocols in fully connected topologies have been defined in [14]. The model is based on extensions of Petri nets with whole place operations, used to model cache coherence protocols [6]. The coverability problem for broadcast protocols is decidable in fully connected graphs [16,7]. This problem is strictly related to marking coverability in Petri nets with transfer or reset arcs [1]. When individual processes are distributed over graphs of arbitrary shape, coverability becomes undecidable as shown in [9]. Decidability holds for special classes like bounded path graphs under the induced graph ordering [9,10], in presence of communication failures or interferences [11], and with dynamic reconfiguration of the communication topology [8]. The PTIME decision procedure in [8] is similar to the labelling algorithms used for parameterized verification of synchronous systems in [18]. In the timed case coverability becomes undecidable already in special types of star topologies [2].

Other formal models of broadcast communication have been proposed in [20,22,15,17]. Verification of unreliable communicating FIFO systems have been studied in [3,4]. In [5] the authors consider different classes of topologies with mixed lossy and perfect channels. The complexity of the verification procedures for lossy FIFO channel systems and broadcast protocols (transfer and reset nets) is discussed in [21]. A classification of the expressive power of different infinite-state models including lossy FIFO channel systems and broadcast protocols is discussed in [1].

Differently from all the previous works, we consider here coverability for parametric initial configurations for a distributed model with asynchronous broadcast. Furthermore, we also consider different policies to handle the message buffers as well as unreliability of the communication media. Finally, our new complexity results improve the preliminary analysis presented in the extended abstract [12], where we used well-structured transition systems for evaluating decidability for bags and lossy FIFO systems.

## References

1. Abdulla, P.A., Delzanno, G., Begin, L.V.: A classification of the expressive power of well-structured transition systems. Inf. Comput. 209(3), 248–279 (2011)
2. Abdulla, P.A., Delzanno, G., Rezine, O., Sangnier, A., Traverso, R.: On the Verification of Timed Ad Hoc Networks. In: Fahrenberg, U., Tripakis, S. (eds.) FORMATS 2011. LNCS, vol. 6919, pp. 256–270. Springer, Heidelberg (2011)

3. Abdulla, P.A., Jonsson, B.: Undecidable verification problems for programs with unreliable channels. Inf. Comput. 130(1), 71–90 (1996)
4. Cécé, G., Finkel, A., Iyer, S.P.: Unreliable channels are easier to verify than perfect channels. Inf. Comput. 124(1), 20–31 (1996)
5. Chambart, P., Schnoebelen, P.: Mixing Lossy and Perfect Fifo Channels. In: van Breugel, F., Chechik, M. (eds.) CONCUR 2008. LNCS, vol. 5201, pp. 340–355. Springer, Heidelberg (2008)
6. Delzanno, G.: Constraint-based verification of parameterized cache coherence protocols. FMSD 23(3), 257–301 (2003)
7. Delzanno, G., Esparza, J., Podelski, A.: Constraint-Based Analysis of Broadcast Protocols. In: Flum, J., Rodríguez-Artalejo, M. (eds.) CSL 1999. LNCS, vol. 1683, pp. 50–66. Springer, Heidelberg (1999)
8. Delzanno, G., Sangnier, A., Traverso, R., Zavattaro, G.: On the complexity of parameterized reachability in reconfigurable broadcast networks. In: FSTTCS 2012, vol. 18, pp. 289–300. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (2012)
9. Delzanno, G., Sangnier, A., Zavattaro, G.: Parameterized Verification of Ad Hoc Networks. In: Gastin, P., Laroussinie, F. (eds.) CONCUR 2010. LNCS, vol. 6269, pp. 313–327. Springer, Heidelberg (2010)
10. Delzanno, G., Sangnier, A., Zavattaro, G.: On the Power of Cliques in the Parameterized Verification of Ad Hoc Networks. In: Hofmann, M. (ed.) FOSSACS 2011. LNCS, vol. 6604, pp. 441–455. Springer, Heidelberg (2011)
11. Delzanno, G., Sangnier, A., Zavattaro, G.: Verification of Ad Hoc Networks with Node and Communication Failures. In: Giese, H., Rosu, G. (eds.) FMOODS/FORTE 2012. LNCS, vol. 7273, pp. 235–250. Springer, Heidelberg (2012)
12. Delzanno, G., Traverso, R.: A formal model of asynchronous broadcast communication (preliminary results). In: ICTCS 2012 (2012), `http://ictcs.di.unimi.it/papers/paper_29.pdf`
13. Delzanno, G., Traverso, R.: On the coverability problem for asynchronous broadcast networks (extended and revised version). Tech. rep., TR-12-05, DIBRIS, University of Genova (November 2012), `http://verify.disi.unige.it/publications/`
14. Emerson, E.A., Namjoshi, K.S.: On model checking for non-deterministic infinite-state systems. In: LICS, pp. 70–80 (1998)
15. Ene, C., Muntean, T.: A broadcast-based calculus for communicating systems. In: IPDPS 2001, p. 149 (2001)
16. Esparza, J., Finkel, A., Mayr, R.: On the verification of broadcast protocols. In: LICS 1999, pp. 352–359 (1999)
17. Fehnker, A., van Glabbeek, R., Höfner, P., McIver, A., Portmann, M., Tan, W.L.: A Process Algebra for Wireless Mesh Networks. In: Seidl, H. (ed.) ESOP 2012. LNCS, vol. 7211, pp. 295–315. Springer, Heidelberg (2012)
18. German, S.M., Sistla, A.P.: Reasoning about systems with many processes. J. ACM 39(3), 675–735 (1992)
19. Ladner, R.E.: The circuit value problem is log space complete for p. SIGACT News 7(1), 18–20 (1975)
20. Prasad, K.V.S.: A calculus of broadcasting systems. Sci. Comput. Program. 25(2-3), 285–327 (1995)
21. Schnoebelen, P.: Revisiting Ackermann-Hardness for Lossy Counter Machines and Reset Petri Nets. In: Hliněný, P., Kučera, A. (eds.) MFCS 2010. LNCS, vol. 6281, pp. 616–628. Springer, Heidelberg (2010)
22. Singh, A., Ramakrishnan, C.R., Smolka, S.A.: A process calculus for mobile ad hoc networks. Sci. Comput. Program. 75(6), 440–469 (2010)