# A Formal Model of Asynchronous Broadcast Communication

Giorgio Delzanno and Riccardo Traverso

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

We present a mathematical model, called Asynchronous Broadcast Networks (ABN), of distributed computation based on topology-dependent and asynchronous communication. Our model combines three main features: a graph representation of a network configuration decoupled from the specification of individual process behavior, a topology-dependent semantics of synchronization, the use of local mailboxes to deliver messages to individual nodes. The resulting communication layer is similar to that of languages like AWN [9]. As in other protocol models like $\omega$ [16,17] and AHN [5], our main abstraction comes from considering protocols defined via a communicating finite-state automaton replicated on each node of the network.

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 $\Sigma$. The initial configuration is any graph in which all the nodes are in the initial control state and all local buffers are empty. Even if the set of control states is finite, there are infinitely many possible initial configurations. We next formalize the above intuition.

A mailbox structure is a tuple $\mathbb{M} = \langle \mathcal{M}, del?, add, del, [] \rangle$, where $\mathcal{M}$ is a denumerable set of elements denoting possible mailbox contents on some fixed finite alphabet $\Sigma$, and, 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 $del?(a, m) = true$. The semantics and corresponding properties change with the type of mailbox considered.

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 the transition relation, $q_0 \in Q$ is an initial control state. The label $\tau$ represents the ability of performing an internal action, while $!!a$ [$??a$] represents the ability of broadcasting [receiving] a message $a \in \Sigma$. 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 (self-loops are forbidden to model half-duplex communication), and $L : V \to Q \times \mathcal{M}$ is a labeling function.

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 possible configurations is denoted $\mathcal{C}$, while $\mathcal{C}_0 \subseteq \mathcal{C}$ is the set of all initial configurations, in which nodes always have the same label $\langle q_0, [] \rangle$.

Given the labeling $L$ 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 $\gamma = \langle V, E, L \rangle \in \mathcal{C}$, we use $L_s(\gamma)$ to denote the set $\{L_s(v) \mid v \in V\}$.

For $\mathbb{M} = \langle \mathcal{M}, del?, add, del, [] \rangle$, an Asynchronous Broadcast Network (ABN) associated to $\mathcal{P}$ is defined by its associated transition system $\mathcal{T}(\mathcal{P}, \mathbb{M}) = \langle \mathcal{C}, \Rightarrow_{\mathbb{M}}, \mathcal{C}_0 \rangle$, where $\Rightarrow_{\mathbb{M}} \subseteq \mathcal{C} \times \mathcal{C}$ 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 neighbors 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.

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 $\Rightarrow_{\mathbb{M}}$. Furthermore, we define the set of immediate predecessors of a set $S$ of configurations as $pre(S) = \{\gamma \mid \gamma \Rightarrow_{\mathbb{M}} \gamma', \gamma' \in S\}$. We use $pre^*$ to indicate the reflexive-transitive closure of $pre$.

*Decision Problems* The coverability problem parametric on the mailbox structure $\mathbb{M}$ is defined as follows. Given a protocol $\mathcal{P}$ with transition system $\mathcal{T}(\mathcal{P}, \mathbb{M}) = \langle \mathcal{C}, \Rightarrow_{\mathbb{M}}, \mathcal{C}_0 \rangle$ and a control state $q$, the coverability problem $COVER(\mathbb{M})$ states: are there two configurations $\gamma_0 \in \mathcal{C}_0$ and $\gamma_1 \in \mathcal{C}$ such that $\gamma_0 \Rightarrow^*_{\mathbb{M}} \gamma_1$ and $q \in L_s(\gamma_1)$?

*Preliminary Results* When local buffers are treated as bags of messages the coverability problem is decidable. For the proof, it is first possible to consider the restricted case of fully connected topologies. For fully connected topologies, we can then resort to the theory of well-structured transition systems (wsts) [1,10] and show that reachability of a given control state can be solved via a symbolic backward search algorithm. When mailboxes are ordered buffers, we obtain undecidability already in the case of fully connected topologies. Indeed, by using FIFO mailboxes, we give nodes the possibility of recognizing communication with multiple neighbors with the same role. We cannot use this feature to define discovery protocols as for the undecidability proof of synchronous broadcast given in [5], but we can simulate a counter machine by using FIFO mailboxes as circular queues for encoding counters and to block computations which may lead to incorrect results. The coverability problem becomes decidable when introducing non-deterministic message losses. We can exploit again the theory of wsts for this positive result. In an extended model in which a node can test if its mailbox is empty, we obtain undecidability with unordered bags and fully-connected topologies. We cannot rely on queues anymore to distinguish bad computations, but the emptiness test allows us to do it anyway. Detailed proofs of these results are available in the technical report [6].

*Related Work* Our analysis completes previous work on verification and expressiveness (w.r.t. coverability) of broadcast communication. More specifically, for synchronous broadcast communication, the coverability problem is decidable for fully connected graphs [8] and undecidable for arbitrary graphs in the AHN model of [5]. Broadcast in AHN is topology-dependent. Synchronous communication is used here to implement a discovery protocol that, by a careful control of interferences, allows individual nodes to infer precise information about their vicinity (e.g. the existence of one and only one neighbor with a certain role). The discovery protocol is a building block for more complex computations. In this paper we use similar ideas but reductions of different nature to obtain undecidability (e.g. we encode counters using mailboxes and not by using linked structures).

For variations of the synchronous semantics like those proposed in [11], intermittent nodes and non-atomic broadcast, coverability becomes decidable. The decidability results exploit however different proof techniques. Indeed, coverability with intermittent nodes can be decided by using a weaker model than Petri nets, whereas we need to resort to the theory of wsts with nested data structures (bags of tuples containing multisets) to show decidability for the unordered case. There seems to be no direct reduction from one model to the other. Furthermore, by either introducing $\epsilon$-transitions or moving to the case of ordered mailboxes we obtain undecidability of the resulting model. Concerning other models of broadcast communication, we would like to mention the CBS process calculi by Prasad [14,15] for fully connected networks with synchronous broadcast communication, the $\omega$-calculus by Singh et al. [16,17] for fully connected networks with synchronous broadcast communication, and the model with topology-dependent broadcast by Ene and Muntean [7]. More recently, a process algebra for different types of communication, including asynchronous broadcast, called AWN, has been proposed in [9]. Semantics that take into consideration interferences and conflicts during a transmission have been proposed in [13,12]. Verification of unreliable communicating FIFO systems have been studied in [2,3]. In [4] the authors consider different classes of topologies with mixed lossy and perfect channels [4]. 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 (bags/queues) and as well as unreliability of the communication media.

Concerning possible refinement of the unordered case, we are currently considering an extension with identifiers where each node has a unique identifier that can be passed using broadcast messages and compared with equality. The introduction of the extended semantics with identifiers and value passing and the formal analysis of the coverability problem is left for an extended version of the work.

## References


1. Parosh Aziz Abdulla, Karlis Cerans, Bengt Jonsson, and Yih-Kuen Tsay. General decidability theorems for infinite-state systems. In *LICS*, pages 313–321, 1996.
2. Parosh Aziz Abdulla and Bengt Jonsson. Undecidable verification problems for programs with unreliable channels. *Inf. Comput.*, 130(1):71–90, 1996.
3. Gérard Cécé, Alain Finkel, and S. Purushothaman Iyer. Unreliable channels are easier to verify than perfect channels. *Inf. Comput.*, 124(1):20–31, 1996.
4. Pierre Chambart and Ph. Schnoebelen. Mixing lossy and perfect fifo channels. In *CONCUR*, pages 340–355, 2008.
5. Giorgio Delzanno, Arnaud Sangnier, and Gianluigi Zavattaro. Parameterized verification of ad hoc networks. In *CONCUR*, pages 313–327, 2010.
6. Giorgio Delzanno and Riccardo Traverso. On the coverability problem for asynchronous broadcast networks. Technical Report DISI-TR-12-05, Dip. Informatica e Scienze dell'Informazione, 2012.
7. Cristian Ene and Traian Muntean. A broadcast-based calculus for communicating systems. In *IPDPS*, page 149, 2001.
8. Javier Esparza, Alain Finkel, and Richard Mayr. On the verification of broadcast protocols. In *LICS*, pages 352–359, 1999.
9. Ansgar Fehnker, Rob J. van Glabbeek, Peter Höfner, Annabelle McIver, Marius Portmann, and Wee Lum Tan. A process algebra for wireless mesh networks. In *ESOP*, pages 295–315, 2012.
10. Alain Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *Theor. Comput. Sci.*, 256(1-2):63–92, 2001.
11. Gianluigi Zavattaro Giorgio Delzanno, Arnaud Sangnier. Verification of ad hoc networks with node and communication failures. In *FORTE*, 2012.
12. Massimo Merro, Francesco Ballardin, and Eleonora Sibilio. A timed calculus for wireless systems. *Theor. Comput. Sci.*, 412(47):6585–6611, 2011.
13. Nicola Mezzetti and Davide Sangiorgi. Towards a calculus for wireless systems. *Electr. Notes Theor. Comput. Sci.*, 158:331–353, 2006.
14. K. V. S. Prasad. A calculus of broadcasting systems. *Sci. Comput. Program.*, 25(2-3):285–327, 1995.
15. K. V. S. Prasad. Broadcasting in time. In *COORDINATION*, pages 321–338, 1996.
16. Anu Singh, C. R. Ramakrishnan, and Scott A. Smolka. Query-based model checking of ad hoc network protocols. In *CONCUR*, pages 603–619, 2009.
17. Anu Singh, C. R. Ramakrishnan, and Scott A. Smolka. A process calculus for mobile ad hoc networks. *Sci. Comput. Program.*, 75(6):440–469, 2010.