Theoretical Computer Science 612 (2016) 1–22

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Parameterized verification of time-sensitive models of ad hoc network protocols ☆

Parosh Aziz Abdulla [a], Giorgio Delzanno [b,*], Othmane Rezine [a], Arnaud Sangnier [c], Riccardo Traverso [d]

[a] Uppsala University, Sweden
[b] University of Genova, Italy
[c] LIAFA, Univ. Paris Diderot, CNRS, France
[d] FBK, Trento, Italy

ARTICLE INFO

Article history:
Received 1 February 2014
Received in revised form 8 July 2015
Accepted 25 July 2015
Available online 7 August 2015
Communicated by J.-F. Raskin

Keywords:
Parameterized verification
Timed automata
Ad hoc networks
Graphs
Decidability
Well structured transition systems

ABSTRACT

We study decidability and undecidability results for parameterized verification of a formal model of timed Ad Hoc network protocols. The communication topology is defined by an undirected graph and the behaviour of each node is defined by a timed automaton communicating with its neighbours via broadcast messages. We consider parameterized verification problems formulated in terms of reachability. In particular we are interested in searching for an initial configuration from which an individual node can reach an error state. We study the problem for dense and discrete time and compare the results with those obtained for (fully connected) networks of timed automata.

1. Introduction

In recent years there has been an increasing interest in automated verification methods for ad hoc networks, see e.g. [18,24,23,11,12]. Ad Hoc Networks (AHN) consist of wireless hosts that, in absence of a fixed infrastructure, communicate sending broadcast messages. In this context, protocols are supposed to work independently from a specific configuration of the network. Indeed, discovery protocols are often applied in order to identify the vicinity of a given node. In the AHN model proposed in [11] undirected graphs are used to represent a network in which each node executes an instance of a fixed (untimed) interaction protocol based on broadcast communication. Since individual nodes are not aware of the network topology, in the ad hoc setting it is natural to consider verification problems that are parametric in the size and shape of the initial configuration as in [11].

In this paper we introduce a new model of distributed systems obtained by enriching the AHN model of [11] with time-sensitive specification of individual nodes. In the resulting model, called Timed Ad Hoc Networks (TAHN), the connection

☆ This work is partially supported by the ANR national research program ANR-14-CE28-0002 PACS and by the MIUR PRIN Italian research program CINA.
* Corresponding author.
E-mail addresses: parosh@it.uu.se (P.A. Abdulla), giorgio.delzanno@unige.it (G. Delzanno), sangnier@liafa.univ-paris-diderot.fr (A. Sangnier), rtraverso@fbk.eu (R. Traverso).

http://dx.doi.org/10.1016/j.tcs.2015.07.048
0304-3975/

topology is still modelled as a graph in which nodes communicate via broadcast messages but the behaviour of a node is now defined as a timed automaton. More in detail, each node has a finite set of clocks which all advance at the same rate and transitions describing the behaviour of the nodes are guarded by conditions on clocks and have also the ability to reset clocks.

Following [11,12], we study the decidability status of the parameterized reachability problem taking as parameters the initial configuration of a TAHN, i.e., we aim at checking the existence of an initial configuration that can evolve using continuous and discrete steps into a configuration exposing a given local state (usually representing an error). Our model presents similarities with Timed Networks introduced in [2]. A major difference between TAHN and Timed Networks lies in the fact that in the latter model the connection topology is always a fully-connected graph, i.e., broadcast communication is not selective since a message sent by a node always reaches all other nodes. For Timed Networks, it is known that reachability of a configuration containing a given control location is undecidable in the case of two clocks per node, and decidable in the case of one clock per node.

When constraining communication via a complex connection graph, the decidability frontier becomes much more complex. More specifically, our technical results are as follows:

- For nodes equipped with a single clock, parameterized reachability becomes undecidable in a very simple class of graphs in which nodes are connected so as to form stars with diameter five.
- The undecidability result still holds in the more general class of bounded path graphs, i.e., graphs in which the length of maximal simple paths is bounded by a constant. In our proof we consider a bound $N \geq 5$ on the length of simple paths. Since nodes have no information about the shape of the network topology, the undecidability proof is not a direct consequence of the result for stars. Indeed the undecidability construction requires a preliminary step aimed at discovering a two-star topology in a graph of arbitrary shape but simple paths of at most five nodes.
- The problem turns out to be undecidable in the class of cliques of arbitrary order (that contains graphs with arbitrarily long paths) in which each timed automaton has at least two clocks.
- Decidability holds for special topologies like stars with diameter three and cliques of arbitrary order assuming that the process running in each node is equipped with a single clock (as in Timed Networks).
- Finally when considering discrete time, e.g. to model time-stamps, instead of continuous time, we show that the local state reachability problem becomes decidable for processes with any number of clocks in the class of graphs with bounded path. The same result holds for cliques of arbitrary order.

2. Preliminaries

Let $\mathbb{N}$ be the set of natural numbers and $\mathbb{R}^{\geq 0}$ the set of non-negative real numbers. For sets A and B, we use $f : A \mapsto B$ to denote that f is a total function that maps A to B. For $a \in A$ and $b \in B$, we write $f[a \hookleftarrow b]$ to denote the function f' defined as follows: $f'(a) = b$ and $f'(a') = f(a')$ for all $a' \neq a$. We denote by $[A \mapsto B]$ the set of all total functions from A to B.

We now recall the notion of well-quasi-ordering (which we abbreviate as wqo). A quasi-order $(A, \preceq)$ is a wqo if for every infinite sequence of elements $a_1, a_2, \ldots$ in A, there exist two indices $i < j$ such that $a_i \preceq a_j$. Given a set A with an ordering $\preceq$ and a subset $B \subseteq A$, the set B is said to be *upward closed* in A if $a_1 \in B$, $a_2 \in A$ and $a_1 \preceq a_2$ implies $a_2 \in B$. Given a set $B \subseteq A$, we define the upward closure $\uparrow B$ to be the set $\{a \in A \mid \exists a' \in B \text{ such that } a' \preceq a\}$. For a quasi-order $(A, \preceq)$, an element a is minimal for $B \subseteq A$ if for all $b \in B$, $b \preceq a$ implies $a \preceq b$. If $(A, \preceq)$ is a wqo and if B is upward closed in A, then the set of minimal elements of B is finite. If $\{b_1, \ldots, b_k\}$ is the set of minimal elements of B, then $\uparrow\{b_1, \ldots, b_k\} = B$; hence B can be represented finitely.

3. Timed ad hoc networks

3.1. Syntax

A Timed Ad Hoc Network (TAHN) consists of a graph where the nodes represent processes that run a common predefined protocol defined by a communicating timed automaton. The values of the clocks manipulated by the automaton inside each process are incremented all at the same rate. In addition, processes may perform discrete transitions which are either local transitions or communication events. When firing a *local* transition, a single process changes its local state without interacting with the other processes. For what concerns communication, it is performed by means of *selective broadcast*, a process sends a broadcast message which can be received only by its neighbours in the network. We choose to represent the communication relation as a graph. Finally, transitions are guarded by conditions on values of clocks and may also reset clocks.

We now provide the formal definition of the model. We assume that each process operates on a set of clocks X. A *guard* is a boolean combination of predicates of the form $k \lhd x$ for $k \in \mathbb{N}$, $\lhd \in \{=, <, \leq, >, \geq\}$, and $x \in X$. We denote by $\mathcal{G}(X)$ the set of guards over X. A reset R is a subset of X. The guards will be used to impose conditions on the clocks of processes that participate in transitions and the resets to identify the clocks that will be reset during the transition. A *clock valuation* is a mapping $F : X \mapsto \mathbb{R}^{\geq 0}$. For a guard g and a clock valuation F, we write $F \models g$ to indicate that the formula obtained by replacing in the guard g each clock x by $F(x)$ is valid. For a clock valuation F and a subset of clocks $Y \subseteq X$, we denote by $F[Y]$ the clock valuation such that $F[Y](x) = 0$ for all $x \in Y$ and $F[Y](x) = F(x)$ for all $x \in X \setminus Y$.