Parameterized Verification of Safety Properties in Ad Hoc Network Protocols

Giorgio Delzanno
University of Genova - Italy
delzanno@disi.unige.it

Arnaud Sangnier
LIAFA, University Paris 7 - France
sangnier@liafa.jussieu.fr

Gianluigi Zavattaro
University of Bologna - Italy
zavattar@cs.unibo.it

We summarize the main results proved in recent work on the parameterized verification of safety properties for ad hoc network protocols. We consider a model in which the communication topology of a network is represented as a graph. Nodes represent states of individual processes. Adjacent nodes represent single-hop neighbors. Processes are finite state automata that communicate via selective broadcast messages. Reception of a broadcast is restricted to single-hop neighbors. For this model we consider a decision problem that can be expressed as the verification of the existence of an initial topology in which the execution of the protocol can lead to a configuration with at least one node in a certain state. The decision problem is parametric both on the size and on the form of the communication topology of the initial configurations. We draw a complete picture of the decidability and complexity boundaries of this problem according to various assumptions on the possible topologies.

1 Introduction

Ad hoc networks consist of wireless hosts that, in the absence of a fixed infrastructure, communicate sending broadcast messages. In this context protocols are typically supposed to work independently from the communication topology and from the size (number of nodes) of the network. As suggested in [3, 4], the *control state reachability problem* (or *coverability problem*) seems a particularly adequate formalization of parameterized verification problems for ad hoc networks. A network is represented as a graph in which nodes are individual processes and edges represent communication links. Each node executes an instance of the same protocol. A protocol is described by a finite state communicating automaton. The control state reachability problem consists in checking whether there exists an initial graph (with unknown size and topology) that can evolve into a configuration in which at least one node is in a given error state. Since the size of the initial configuration is not fixed a priori, the state-space to be explored is in general infinite.

In this paper we summarize the main results that we have proved in two recent publications [3, 4]. The first result is negative: control state reachability is undecidable if we do not fix any restriction on the possible topologies. As for other communication models [16, 25], finding interesting classes of network topologies for which verification is, at least theoretically, possible is an important research problem. As a first positive result, we have proved in [3] that control state reachability turns out to be decidable for the class of *bounded path graphs*. Graphs have bounded path if there exists a value k such that all simple paths in the considered graph have length smaller than k. Although for a fixed k this class of graphs is infinite, it appears of limited interest as it does not include clique graphs. Cliques are appealing for at least two reasons. First, they represent the best possible scenario for optimizing broadcast communication (one broadcast to reach all nodes). Second, when restricting configurations only to cliques, control state reachability can be reduced to coverability in a Broadcast Protocol, i.e.,

L. Aceto and M.R. Mousavi (Eds.): PACO 2011
EPTCS 60, 2011, pp. 56–65, doi:10.4204/EPTCS.60.4

in a model in which configurations are multisets of processes defined by communicating automata [6]. Coverability is decidable in Broadcast Protocols in [8].

For these reasons, in [4] we have decided to investigate classes of graphs that at least include the clique graphs. More precisely, we have considered networks in which the underlying topology is in between the class of *cliques* and the strictly larger class of *bounded diameter graphs*. Graphs have bounded diameter if there exists a value k such that the minimal path between every pair of nodes of the same graph has length smaller than k. Graphs with bounded diameter (also called clusters) are particularly relevant for the domain of ad hoc networks. They are often used to partition a network in order to increase the efficiency of broadcast communication [10].

The restriction to bounded diameter follows the approach taken for point-to-point communication in [16, 25]. Differently from [16, 25] we have proved that for our model of selective broadcast control state reachability is undecidable when restricting the topologies to graphs whose diameter is bounded by k (for a fixed $k > 0$). Then, we have investigated further restrictions having in mind the constraint that they must allow at least cliques of arbitrary order. By using an original well-quasi ordering result, we have proved that control state reachability becomes decidable when considering a class of graphs in which the corresponding maximal cliques are connected by paths of bounded length. Furthermore, by exploiting a recent result of Schnoebelen [21] and a reduction to coverability in reset nets, we have shown that the resulting decision procedure is Ackermann-hard. Interestingly, the same complexity result already holds in the subclass of clique topologies.

Related Work Ethernet-like broadcast communication has been analyzed by Prasad [18] using the Calculus of Broadcasting Systems, in which all processes receive a broadcast message at once. A similar type of broadcast mechanism is used in the Broadcast Protocols of Emerson and Namjoshi [6]. In our setting, this is similar to the case in which all nodes share a common group (the underlying graph is a clique). Ene and Muntean presented the $b\pi$-calculus [7], an extension of the π-calculus [19] with a broadcast such that only nodes listening on the right channel can receive emitted messages. Wireless broadcast communication has been investigated in the context of process calculi by Nanz and Hankin [17], Singh, Ramakrishnan and Smolka [22, 23], Lanese and Sangiorgi [14], Godskesen [12], and Merro [15]. In particular Nanz and Hankin [17] consider a graph representation of node localities to determine the receivers of a message, while Godskesen [12] makes use of a neighbour relation. On the contrary, Lanese and Sangiorgi [14] and Merro [15] associate physical locations to processes so that the receivers depend on the location of the emitter and its transmission range. As already mentioned, we have been directly inspired by the ω-calculus of Singh, Ramakrishnan and Smolka [22, 23]. The ω-calculus is based on the π-calculus. The π-calculus [19] intermixes the communication and mobility of processes by expressing mobility as change of interconnection structure among processes through communication. In the ω-calculus mobility of processes is abstracted from their communication actions, i.e., mobility is spontaneous and it does not involve any communication. In [24] the same authors define a constraint-based analysis for configurations with fixed topologies and a fixed number of nodes. The authors also mention that checking reachability of a configuration from an initial one is decidable for the fragment without restriction. This property is an immediate consequence of the fact that there is no dynamic generation or deletion of processes (i.e. it boils down to a finite-state reachability problem). The symbolic approach in [24] seems to improve verification results obtained with more standard model checking techniques. For instance, in [9] model checking is used for automatic verification of finite-state and timed models of Ad Hoc Networks. In these works the number of nodes in the initial configurations is known and fixed a priori. In order to detect protocol vulnerabilities tools like Uppaal are executed on all possible topologies (modulo symmetries) for a given number of nodes. In [20] Saksena et al. define a symbolic procedure based on graph-transformations to analyze routing protocol for Ad Hoc Networks. The symbolic representation is based on upward closed sets of graphs ordered w.r.t. subgraph inclusion. Their procedure is not guaranteed to terminate. In our paper we consider a non trivial class of graphs (bounded path configurations) for which backward analysis with a similar symbolic representation (upward closure of graphs w.r.t. induced subgraph ordering) is guaranteed to terminate for finite-state descriptions of individual nodes.

Structure of the paper In Section 2 we formally introduce our model for ad hoc network protocols, we define the parametric version of the control state reachability problem, and we recall the result from [3], i.e. that control state reachability is undecidable if we do not impose any restriction on the class of possible topologies, while it turns out to be decidable when restricting to bounded path topologies. In Sections 3 and 4 we consider other restricted classes that include clique graphs: bounded diameter and bounded path on the maximal clique graph, respectively. For these classes we report the results proved in [4]: control state reachability is undecidable when restricted to graphs with a bounded diameter (but it turns out to be decidable if we additionally assume bounded degree), while for the class of graphs having a corresponding maximal clique graph with bounded path, the problem is decidable. Section 5 contains concluding remarks and directions for future work.

2 Ad Hoc Network Protocols

2.1 Preliminaries on Graphs

In this section we assume that Q is a finite set of elements. A *Q-labeled undirected graph* (shortly Q-graph or graph) is a tuple $G=(V,E,L)$, where V is a finite set of *vertices* (sometimes called *nodes*), and $E \subseteq V \times V$ is a finite set of *edges*, and $L : V \to Q$ is a labeling function. We consider here undirected graphs, i.e., such that $\langle u,v\rangle \in E$ iff $\langle v,u\rangle \in E$. We denote by $\mathscr{G}_Q$ the set of Q-graphs. For an edge $\langle u,v\rangle \in E$, u and v are called its *endpoints* and we say that u and v are adjacent vertices. For a node u we call *vicinity* the set of its adjacent nodes (neighbors). Given a vertex $v \in V$, the *degree* of v is the size of the set $\{u \in V \mid \langle v,u\rangle \in E\}$. The degree of a graph is the maximum degree of its vertices. We will sometimes denote $L(G)$ the set $L(V)$ (which is a subset of Q). A *path* π in a graph is a finite sequence $v_1,v_2,\ldots,v_m$ of vertices such that for $1 \leq i \leq m-1$, $\langle v_i,v_{i+1}\rangle \in E$ and the integer $m-1$ (i.e. its number of edges) is called the length of the path π, denoted by $|\pi|$. A path $\pi = v_1,\ldots,v_m$ is simple if for all $1 \leq i,j \leq m$ with $i \neq j$, $v_i \neq v_j$, in other words each vertex of the graph occurs at most once in π. A *cycle* is a path $\pi = v_1,\ldots,v_m$ such that $v_1 = v_m$. A graph $G = \langle V,E,L\rangle$ is *connected* if for all $u,v \in V$ with $u \neq v$, there exists a path from u to v in G. A *clique* in an undirected graph $G = \langle V,E,L\rangle$ is a subset $C \subseteq V$ of vertices, such that for every $u,v \in C$ with $u \neq v$, $\langle u,v\rangle \in E$. A clique C is said to be *maximal* if there exists no vertex $u \in V \setminus C$ such that $C \cup \{u\}$ is a clique. If the entire set of nodes V is a clique, we say that G is a clique graph. A *bipartite Q-graph* is a tuple $\langle V_1,V_2,E,L\rangle$ such that $\langle V_1 \cup V_2,E,L\rangle$ is a Q-graph, $V_1 \cap V_2 = \emptyset$ and $E \subseteq (V_1 \times V_2) \cup (V_2 \times V_1)$.

The *diameter* of a graph $G = \langle V,E,L\rangle$ is the length of the *longest shortest simple path* between any two vertices of G. Hence, the diameter of a clique graph is always one. We also need to define some graph orderings. Given two graphs $G = \langle V,E,L\rangle$ and $G' = \langle V',E',L'\rangle$, G is in the *subgraph* relation with G', written $G \preceq_s G'$, whenever there exists an injective function $f : V \to V'$ such that, for every $v,v' \in V$, if $\langle v,v'\rangle \in E$, then $\langle f(v),f(v')\rangle \in E'$ and for every $v \in V$, $L(v) = L'(f(v))$. Furthermore, G is in the *induced subgraph* relation with G', written $G \preceq_i G'$, whenever there exists an injective function $f : V \rightarrow V'$ such that, for every $v, v' \in V$, $\langle v, v' \rangle \in E$ if and only if $\langle f(v), f(v') \rangle \in E'$ and for every $v \in V$, $L(v) = L'(f(v))$. As an example, a path with three nodes is a subgraph, but not an induced subgraph, of a ring of the same order. Finally, we recall the notion of *well-quasi-ordering* (wqo for short). A quasi order $(A, \leq)$ is a wqo if for every infinite sequence of elements $a_1, a_2, \ldots, a_i, \ldots$ in A, there exist two indices $i < j$ s.t. $a_i \leq a_j$. Examples of wqo's are the sub-multiset relation, and both the subgraph and the induced subgraph relation over graphs with simple paths of bounded length [5].

2.2 Ad Hoc Networks

In our model of ad hoc networks a configuration is simply a graph and we assume that each node of the graph is a process that runs a common predefined protocol. A protocol is 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 in fact local to the vicinity of the sender. The initial configuration is any graph in which all the nodes are in an initial control state. Remark that even if Q is finite, there are infinitely many possible initial configurations. We next formalize the above intuition.

Individual Behavior The protocol run by each node is defined via a process $\mathscr{P} = \langle Q, \Sigma, R, Q_0 \rangle$, 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 $a \in \Sigma$.

Network Semantics An AHN associated to $\mathscr{P} = \langle Q, \Sigma, R, Q_0 \rangle$ is defined via a transition system $\mathscr{A}_{\mathscr{P}} = \langle \mathscr{C}, \Rightarrow, \mathscr{C}_0 \rangle$, where $\mathscr{C} = \mathscr{G}_Q$ (undirected graphs with labels in Q) is the set of configurations, $\mathscr{C}_0 = \mathscr{G}_{Q_0}$ (undirected graphs with labels in Q_0) is the subset of initial configurations, and $\Rightarrow \subseteq \mathscr{C} \times \mathscr{C}$ is the transition relation defined next. For $q \in Q$ and $a \in \Sigma$, we define the set $R_a(q) = \{q' \in Q \mid \langle q, ??a, q' \rangle \in R\}$ that contains states that can be reached from the state q upon reception of message a. For $G = \langle V, E, L \rangle$ and $G' = \langle V', E', L' \rangle$, $G \Rightarrow G'$ holds iff G and G' have the same underlying structure, i.e., $V = V'$ and $E = E'$, and one of the following conditions on L and L' holds:

- $\exists v \in V$ s.t. $(L(v), \tau, L'(v)) \in R$, and $L(u) = L'(u)$ for all u in $V \setminus \{v\}$;
- $\exists v \in V$ s.t. $(L(v), !!a, L'(v)) \in R$ and for every $u \in V \setminus \{v\}$
  - if $\langle v, u \rangle \in E$ and $R_a(L(u)) \neq \emptyset$ (reception of a in u is enabled), then $L'(u) \in R_a(L(u))$.
  - $L(u) = L'(u)$, otherwise.

An execution is a sequence $G_0 G_1 \ldots$ such that $G_0 \in \mathscr{G}_{Q_0}$ and $G_i \Rightarrow G_{i+1}$ 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 updates its state with a new state taken from $R(L(u))$. 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 a run.

Finally, for a set of Q-graphs $\mathscr{T} \subseteq \mathscr{G}_Q$, the AHN $A^{\mathscr{T}}_{\mathscr{P}}$ restricted to $\mathscr{T}$ is defined by the transition system $\langle \mathscr{C} \cap \mathscr{T}, \Rightarrow_{\mathscr{T}}, \mathscr{C}_0 \cap \mathscr{T} \rangle$ where the relation $\Rightarrow_{\mathscr{T}}$ is the restriction of $\Rightarrow$ to $(\mathscr{C} \cap \mathscr{T}) \times (\mathscr{C} \cap \mathscr{T})$.

2.3 Example of Ad Hoc Network Protocol

As an example of an ad hoc network protocol and of its semantics, consider a protocol consisting of the following rules: (A, τ, C), $(C, !!m, D)$, $(B, ??m, C)$, and $(A, ??m, C)$. As shown in Fig. 1, starting from a

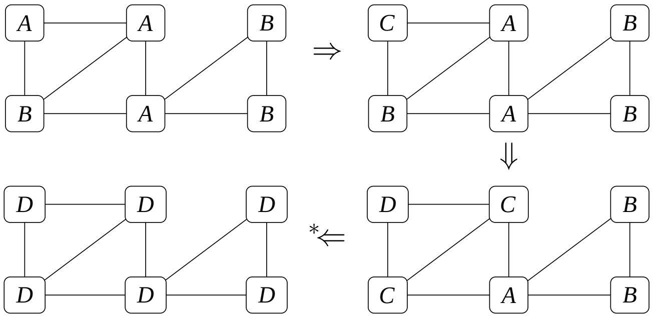

Figure 1: Example of execution

configuration with only A and B nodes, an A node first moves to C and then send m to his/her neighbors. In turn, they forward the message m to their neighbors, and so on.

2.4 Decision problem

We define the decision problem of *control state reachability* (COVER) as follows:

Input: A process $\mathscr{P} = \langle Q, \Sigma, R, Q_0 \rangle$ with $\mathscr{A}_{\mathscr{P}} = \langle \mathscr{C}, \Rightarrow, \mathscr{C}_0 \rangle$ and a control state $q \in Q$;

Output: Yes, if there exists $G \in \mathscr{C}_0$ and $G' \in \mathscr{C}$ such that $q \in L(G')$ and $G \Rightarrow^* G'$, no otherwise.

Control state reachability is strictly related to parameterized verification of safety properties. The input control state q can in fact be seen as an error state for the execution of the protocol in some node of the network. If the answer to COVER is yes, then there exists a sufficient number of processes, all executing the same protocol, and an initial topology from which we can generate a configuration in which the error is exposed. Under this perspective, COVER can be viewed as instance of a parameterized verification problem.

In [3] we have proved that COVER is undecidable. The proof is by reduction from the halting problem for two-counter Minsky machines. A Minsky machine manipulates two integer variables c_1 and c_2, which are called counters, and it is composed of a finite set of instructions. Each of the instuction is either of the form (1) $L : c_i := c_i + 1;$ `goto` L' or (2) $L :$ `if` $c_i = 0$ `then goto` L' `else` $c_i := c_i - 1;$ `goto` L'' where $i \in \{1, 2\}$ and L, L', L'' are labels preceding each instruction. Furthermore there is a special label L_F from which nothing can be done. The halting problem consists then in deciding whether or not the execution that starts from L_0 with counters equal to 0 reaches L_F.

The intuition behind the reduction is as follows. In a first phase we exploit an exploration protocol to impose a logical topology on top of the actual physical node connections. This logical topology is composed by a control node which is connected to two distinct lists of nodes used to simulate the content of the counters. Each node in the list associated to counter c_i is either in state Z_i or NZ_i. The current value of the counter c_i equals the number of NZ_i nodes in the list. The length of each list is guessed non-deterministically during the execution of the first phase (i.e. before starting the simulation) and it corresponds to the maximum value store in a counter for the simulation to succeed. Initially, all nodes must encode zero (state Z_i).

In the second phase the control node starts the simulation of the instructions. It operates by sending requests that are propagated back and forth a list by using broadcast sent by a node to its (unique) single-hop successor/predecessor node. The effect of these requests is to change the state of one node in zero state Z_i to the non-zero state NZ_i in case of increment, or the vice versa in the case of decrement. The

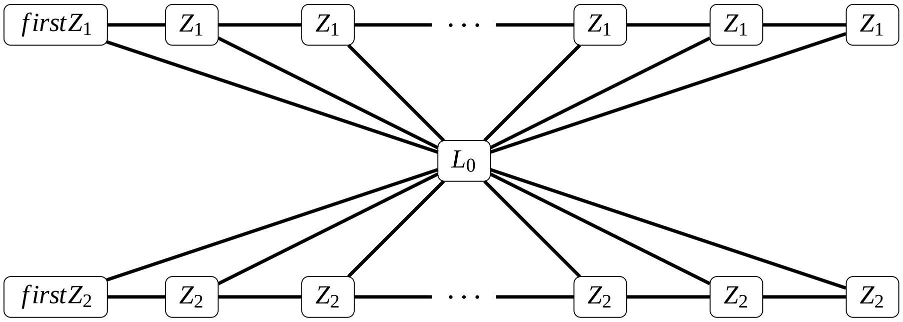

Figure 2: Butterfly-shaped induced subgraph needed to simulate a Minsky machine.

test-for-zero instruction on the counter c_i is simply simulated by checking whether there are no nodes in the zero state Z_i in the i-th list.

2.5 Configurations with Bounded Path

In [3] we have proved that COVER turns out to be decidable if we restrict the possible topologies to the class of graphs whose path is bounded by k (for a fixed $k > 0$). The proof is based on the theory of Well Structured Transition Systems [1, 2, 11] (WSTS). A WSTS is a transition system equipped with a well-quasi ordering on states and a monotonicity property: if a configuration c_1 smaller than a configuration c_2 has a transition to a configuration c'_1, then also c_2 has a transition to a configuration c'_2 which is greater than c'_1. Coverability turns out to be decidable in WSTSs by using backward analysis, if it is possible to compute the predecessors of a given state.

In [3] we have observed that ad hoc network protocols are monotonic with respect to the induced subgraph ordering relation, while this is not the case for the subgraph ordering relation. This is already an interesting observation that distinguishes selective broadcast from point-to-point communication, which is monotonic with respect to the usual subgraph ordering. The proof of decidability is completed by defining how to compute the predecessors, and by observing that the induced subgraph ordering is a wqo for the class of graphs for which the length of simple paths is bounded by a constant (i.e. bounded path graphs). This result is known as Ding's Theorem [5].

3 Configurations with Bounded Diameter

As mentioned in the introduction, restricting protocol analysis to configurations with bounded path seems to have a limited application in a communication model with selective broadcast. For these reasons, in [4] we have investige COVER for restricted classes of graphs that at least include the class of clique graphs. The first class we have consider is that of graphs with bounded diameter. Fixed $k > 0$, a graph G has a k-bounded diameter if and only if its diameter is smaller than or equal to k. Observe that for every $k > 0$, clique graphs belong to the class of graphs with a diameter bounded by k. Furthermore, given $k > 0$ the class of graphs with path bounded by k is included in the class of graphs with a diameter bounded by k. Graphs with k-bounded diameter coincide with the so called k-clusters used in partitioning algorithm for ad hoc networks [10]. Thus, this class is of particular relevance for the analysis of selective broadcast communication. Intuitively, the diameter corresponds to the minimal number of broadcasts (hops) needed to send a message to all nodes connected by a path with the sender.

The COVER problem restricted to configurations with k-bounded diameter turns out to be undecidable for $k > 1$. The proof is similar to the proof of undecidability for the general case reported in [3]: by reduction from the halting problem for two-counter Minsky machines.

The main difference is that the logical topology to be imposed in the first phase of the simulation of the Minsky machines should be with bounded diameter (namely, diameter 2). The topology that we have considered is a sort of butterfly (see Figure 2) consisting of two lists (to represent the counters) and in which all nodes in the lists are connected to a monitor node (to represent the program counter). The second phase of the simulation, i.e. the actual execution of the instructions, proceeds similarly to the protocol described above. The unique difference is that we use a distinct $firstZ_i$ node to distinguish the initial node of each list (this is needed as now all the list nodes are connected to the program counter node).

Note that if we restrict our attention to graphs with a diameter bounded by 1, the above encoding does not work anymore. The class of graphs with diameter 1 corresponds to the set of clique graphs and, as said above, COVER turns out to be decidable when restricting to clique topologies.

Bounded diameter and bounded degree.

From a non trivial result on bounded diameter graphs [13], we have obtained in [4] an interesting decidable subclass. Indeed, in [13] the authors show that, given two integers $k,d > 0$, the number of graphs whose diameter is smaller than k and whose degree is smaller than d is finite. The Moore bound $M(k,d) = (k(k-1)^d - 2)/(k-2)$ is an upper bound for the size of the largest undirected graph in such a class. It follows that, for $k,d > 0$, and an ad hoc protocol with n states, if we restrict to configurations with a diameter bounded by k and a degree bounded by d, the state space is bounded by $n^{M(k,d)}$, thus it is polynomial in the size of the protocol. Consequently we can conclude that COVER restricted to configurations with k-bounded diameter and d-bounded degree is in PSPACE.

4 Maximal Clique Graphs with Bounded Paths

In this section we describe classes of graphs that strictly increases both the classes of clique graphs and the classes of bounded path graphs, for which we have proved in [4] that COVER is decidable. We have called these classes of graphs BPC_n (n-Bounded Path maximal Cliques graphs). Namely, for $n > 0$ BPC_n contains both n-bounded path graphs and any clique graph, while being strictly contained in the class of graphs with $2n$-bounded diameter. These classes are defined on top of the notion of *maximal clique graphs* associated to a configuration.

Definition 4.1 Given a connected undirected graph $G = \langle V,E,L \rangle$ and $\bullet \notin L(V)$, the *maximal clique graph* K_G is the bipartite graph $\langle X,W,E',L' \rangle$ in which

- $X = V$;
- $W \subseteq 2^V$ is the set of maximal cliques of G;
- For $v \in V, w \in W$, $\langle v,w \rangle \in E'$ iff $v \in w$;
- $L'(v) = L(v)$ for $v \in V$, and $L'(w) = \bullet$ for $w \in W$.

Note that for each connected graph G there exists a unique maximal clique graph K_G. An example of construction is given by Figure 3. One can also easily prove that if G is a clique graph then in K_G there is no path of length strictly greater than 3. Furthermore, from the maximality of the cliques in W if two

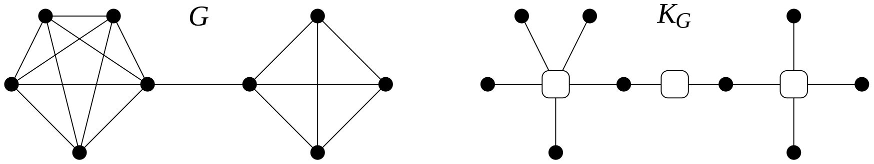

Figure 3: A graph G and its associated clique graph K_G.

nodes $v_1, v_2 \in V$ are connected both to w_1 and $w_2 \in W$, then w_1 and w_2 are distinct cliques. We use the notation $v_1 \sim_w v_2$ to denote that v_1, v_2 belong to the same clique w.

Definition 4.2 For $n \geq 1$, the class BPC_n consists of the set of configurations whose associate maximal clique graph has n-bounded paths (i.e. the length of the simple paths of K_G is at most n).

The proof of decidability of COVER for BPC_n graphs is based on an ordering defined on maximal clique graphs that corresponds to the induced subgraph ordering defined on the corresponding graphs. Such a new ordering is defined as follows.

Definition 4.3 Assume $G_1 = \langle V_1, E_1, L_1 \rangle$ with $K_{G_1} = \langle X_1, W_1, E'_1, L'_1 \rangle$, and $G_2 = \langle V_2, E_2, L_2 \rangle$ with $K_{G_2} = \langle X_2, W_2, E'_2, L'_2 \rangle$ with G_1 and G_2 both connected graphs. Then, $G_1 \sqsubseteq G_2$ iff there exist two injective functions $f : X_1 \to X_2$ and $g : W_1 \to W_2$, such that

(i) for every $v \in X_1$, and $C \in W_1$, $v \in C$ iff $f(v) \in g(C)$;

(ii) for every $v_1, v_2 \in X_1$, and $C \in W_2$, if $f(v_1) \sim_C f(v_2)$, then there exists $C' \in W_1$ s.t. $f(v_1) \sim_{g(C')} f(v_2)$;

(iii) for every $v \in X_1$, $L'_1(v) = L'_2(f(v))$;

(iv) for every $C \in W_1$, $L'_1(C) = L'_2(g(C))$.

The first condition ensures that (dis)connected nodes remain (dis)connected inside the image of g. Indeed, from point (i) it follows that, for every $v_1, v_2 \in X_1$, and $C \in W_1$, $v_1 \sim_C v_2$ iff $f(v_1) \sim_{g(C)} f(v_2)$. The second condition ensures that disconnected nodes remain disconnected outside the image of g.

By condition (i) in the definition of $\sqsubseteq$, we also have that $G_1 \sqsubseteq G_2$ (via f and g) implies that K_{G_1} is in the induced subgraph relation with K_{G_2} (via $f \cup g$). The relation between this new relation and the induced subgraph ordering is even stronger, in fact we have proved in [4] that the two coincide: $G_1 \sqsubseteq G_2$ iff G_1 is an induced subgraph of G_2.

The main theorem in [4] states that for any $n \geq 1$, $(BPC_n, \sqsubseteq)$ is a well-quasi ordering. In the light of the correspondance result between $\sqsubseteq$ and the induced subgraph ordering, and the monotonicity of ad hoc network protocol with respect to the induced subgraph ordering relation (and the computability of the predecessors) discussed the previous section, we have been able to prove in [4] the decidability of COVER for topologies restricted to graphs in BPC_n (for a fixed $n > 0$).

In [4] we have investigated also the complexity of the decision procedure for COVER restricted to topologies in BPC_n. We have found that this problem is not primitive recursive. The proof is by reduction from the coverability problem for reset nets, which is known to be an Ackermann-hard problem [21].

5 Conclusions

In this paper we have reported the main result that we have recently proved in [3, 4] about the decidability and complexity boundaries for the decidability and the complexity of the parametric verification of safety properties in ad hoc networks. Namely, given an ad hoc network protocol expressed as a finite state communicating automaton, we are interested in checking the existence of an initial network configuration that can generate a computation leading to a configuration in which at least one node is in a given (error) state.

The problem is undecidable if no restrictions are imposed to the possible initial configurations, but it turns out to be decidable for interesting classes of graphs in which the corresponding maximal cliques are connected by paths of bounded length. These graphs include both cliques and bounded path graphs. The problem returns to be undecidable for bounded diameter graphs.

As a future work, we plan to study decidability and complexity issues in presence of communication and node failures. In particular, an interesting case of communication failure in the context of ad hoc networks is due to *conflicts* deriving form the contemporaneous emission of signals from two distinct nodes that share some neighbors. We plan to move to a truly concurrent semantics for ad hoc network protocols in order to faithfully represent this specific phenomenon.

References

[1] P. A. Abdulla, C. Čer̄ans, B. Jonsson & Y.-K. Tsay (1996): *General decidability theorems for infinite-state systems*. In: *LICS'96*, IEEE Computer Society, pp. 313–321.

[2] P. A. Abdulla, C. Čer̄ans, B. Jonsson & Tsay. Y.-K. (2000): *Algorithmic analysis of programs with well quasi-ordered domains*. *Inf. Comput.* 160(1-2), pp. 109–127.

[3] G. Delzanno, A. Sangnier & G. Zavattaro (2010): *Parameterized Verification of Ad Hoc Networks*. In: *CONCUR'10*, *Lecture Notes in Computer Science* 6269, Springer, pp. 313–327. doi:10.1007/978-3-642-15375-4_22

[4] G. Delzanno, A. Sangnier & G. Zavattaro (2011): *On the Power of Cliques in the Parameterized Verification of Ad Hoc Networks*. In: *FOSSACS'11*, *Lecture Notes in Computer Science* 6604, Springer, pp. 441–455. doi:10.1007/978-3-642-19805-2_30

[5] G. Ding (1992): *Subgraphs and well quasi ordering*. *J. of Graph Theory* 16(5), pp. 489 – 502.

[6] E. A. Emerson & K. S. Namjoshi (1998): *On Model Checking for Non-Deterministic Infinite-State Systems*. In: *LICS'98*, IEEE Computer Society, pp. 70–80.

[7] C. Ene & T. Muntean (2001): *A Broadcast based Calculus for Communicating Systems*. In: *IPDPS '01*, p. 149.

[8] J. Esparza, A. Finkel & R. Mayr (1999): *On the Verification of Broadcast Protocols*. In: *LICS'99*, IEEE Computer Society, pp. 352–359.

[9] A. Fehnker, L. van Hoesel & A. Mader (2007): *Modelling and verification of the LMAC protocol for wireless sensor networks*. In: *IFM'07*, *Lecture Notes in Computer Science* 4591, Springer, pp. 253–272. doi:10.1007/978-3-540-73210-5_14

[10] Y. Fernandess & D. Malkhi (2002): *K-clustering in wireless ad hoc networks*. In: *POMC'02*, ACM, pp. 31–37. doi:10.1145/584490.584497

[11] A. Finkel & P. Schnoebelen (2001): *Well-structured transition systems everywhere!* *Theoretical Computer Science* 256(1-2), pp. 63–92. doi:10.1016/S0304-3975(00)00102-X

[12] J.C. Godskesen (2007): *A Calculus for Mobile Ad Hoc Networks*. In: *COORDINATION '07*, pp. 132–150.

[13] A.J. Hoffman & R.R. Singleton (1960): *On Moore graphs with diameter 2 and 3*. *IBM J. Res. Develop.* 4, pp. 497–504. doi:10.1147/rd.45.0497

[14] Ivan Lanese & Davide Sangiorgi (2010): *An operational semantics for a calculus for wireless systems*. *Theoretical Computer Science* 411(19), pp. 1928–1948. doi:10.1016/j.tcs.2010.01.023

[15] M. Merro (2009): *An Observational Theory for Mobile Ad Hoc Network*. *Inf. Comput.* 207(2), pp. 194–208. doi:10.1016/j.ic.2007.11.010

[16] R. Meyer (2008): *On boundedness in depth in the pi-calculus*. In: *IFIP TCS'08, IFIP* 477–489, Springer, pp. 477–489.

[17] S. Nanz & C. Hankin (2006): *A Framework for Security Analysis of Mobile Wireless Networks*. *TCS* 367(1–2), pp. 203–227. doi:10.1016/j.tcs.2006.08.036

[18] K.V.S. Prasad (1995): *A Calculus of Broadcasting Systems*. *Sci. of Comp. Prog.* 25(2-3), pp. 285–327. doi:10.1016/0167-6423(95)00017-8

[19] Milner R. (1999): *Communicating and Mobile Systems: the Pi-Calculus*. Cambridge Univ. Press.

[20] M. Saksena, O. Wibling & B. Jonsson (2008): *Graph grammar modeling and verification of Ad Hoc Routing Protocols*. In: *TACAS'08*, *Lecture Notes in Computer Science* 4963, Springer, pp. 18–32. doi:10.1007/978-3-540-78800-3_3

[21] P. Schnoebelen (2010): *Revisiting Ackermann-Hardness for Lossy Counter Machines and Reset Petri Nets*. In: *MFCS'10*, *Lecture Notes in Computer Science* 6281, Springer, pp. 616–628. doi:10.1007/978-3-642-15155-2_54

[22] A. Singh, C. R. Ramakrishnan & S. A. Smolka (2006): *Modeling the AODV routing protocol in omega-calculus*. In: *LISAT '06*.

[23] A. Singh, C. R. Ramakrishnan & S. A. Smolka (2008): *A Process Calculus for Mobile Ad Hoc Networks*. In Springer, editor: *COORDINATION '08*, *Lecture Notes in Computer Science* 5052, pp. 296–314.

[24] A. Singh, C. R. Ramakrishnan & S. A. Smolka (2009): *Query-Based model checking of Ad Hoc Network Protocols*. In: *CONCUR'09*, *Lecture Notes in Computer Science* 5710, Springer, pp. 603–61.

[25] T. Wies, D Zufferey & T. A. Henzinger (2010): *Forward analysis of depth-bounded processes*. In: *FOSSACS'10*, *Lecture Notes in Computer Science* 6014, Springer, pp. 94–108. doi:10.1007/978-3-642-12032-9_8