On the Power of Cliques in the Parameterized Verification of Ad Hoc Networks

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

[1] University of Genova - Italy
[2] LIAFA, University Paris 7, CNRS - France
[3] University of Bologna - Italy

Abstract. We study decision problems for parameterized verification of protocols for ad hoc networks. The problem we consider is control state reachability for networks of arbitrary size. We restrict our analysis to topologies that approximate the notion of cluster (graphs with bounded diameter) often used in ad hoc networks for optimizing broadcast communication. In particular we are interested in classes of graphs that include at least cliques of arbitrary order. We show that, although decidable, control state reachability over cliques is already Ackermann-hard and study more sophisticated topologies for which the problem remains decidable.

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], 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 in [3] 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. As proved in [3], control state reachability is undecidable for graphs with unrestricted topology. As in other communication models [12,20], finding interesting classes of network topologies for which verification is, at least theoretically, possible is an important research problem.

Moving along this line, in this paper we consider networks in which the underlying topology is in between the class of *cliques* and the strictly larger class of *bounded diameter graphs*. Cliques represent the best possible topology for minimizing the number of hops needed for diffusing data. Furthermore, control

state reachability in clique graphs reduces to coverability in a Broadcast Protocol (with unstructured configurations), a problem proved to be decidable in [7]. 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 [9].

Our first result is negative. Indeed, we prove that control state reachability is undecidable for networks in which configurations have bounded diameter. We investigate then 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 prove that coverability 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 [18] and a reduction to coverability in reset nets, we show that the resulting decision procedure is Ackermann-hard. Interestingly, the same complexity result already holds in the subclass of clique topologies. Finally, we introduce a unicast mechanism inspired by rendezvous communication in other concurrency models. Having the two mechanisms in the same model allows us to compare them, with complexity measures, with respect to the coverability problem. Specifically, coverability for unicast communication is easier than for selective broadcast. Indeed, it turns out to be in EXPSPACE for unrestricted graphs. To the best of our knowledge, this discrimination result is novel compared to the existing literature on concurrency models with selective broadcast and unicast communication.

Related Work Model checking has been applied to verify protocols for ad hoc networks with a fixed number of nodes in [8,19]. A possibly non-terminating procedure for the verification of routing protocols in ad hoc networks of arbitrary size is described in [17]. In [3] we have introduced and studied the (repeated) control state reachability problem described in the introduction for the ad hoc network model of [19]. Specifically, we have shown that the problem is undecidable when the topology is unrestricted and that it becomes decidable when the initial network has a topology taken from the class of graphs with bounded paths (the maximal length of a path is bounded by a constant). However this class does not include cliques of arbitrary order. In contrast, we extend here the decidability result to a larger class of graphs, and we investigate the problem for graphs with bounded diameter. Graphs with bounded paths have also been considered in verification problems with point-to-point (unicast in the ad hoc setting) communication in [12,16,20].

Due to lack of space, omitted proofs can be found in [4].

2 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 $\mathcal{G}_Q$ 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. 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 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 injection $f : V \to V'$ 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 injection $f : V \to V'$ 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].

3 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 $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$, where Q is a finite set of control states, Σ is a finite alphabet, $R \subseteq Q \times (\{\tau\} \cup \{!!a, ??a \mid a \in \Sigma\}) \times Q$ is the transition relation, and $Q_0 \subseteq Q$ is a set of initial control states. The label τ represents the capability of performing an internal action, and the label $!!a$ ($??a$) represents the capability of broadcasting (receiving) a message $a \in \Sigma$.

Network Semantics An AHN associated to $\mathcal{P}$ is defined via a transition system $\mathcal{A}_\mathcal{P} = \langle \mathcal{C}, \Rightarrow, \mathcal{C}_0 \rangle$, where $\mathcal{C} = \mathcal{G}_Q$ (undirected graphs with labels in Q) is the set of configurations, $\mathcal{C}_0 = \mathcal{G}_{Q_0}$ (undirected graphs with labels in Q_0) is the subset of initial configurations, and $\Rightarrow \subseteq \mathcal{C} \times \mathcal{C}$ is the transition relation defined next. For $u \in V$, we first define the set $R_a(u) = \{q \in Q \mid \langle L(u), ??a, q \rangle \in R\}$ that contains states that can be reached from the state $L(u)$ 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 $R_a(u) \neq \emptyset$ (reception of a in u is enabled), then $L'(u) \in R_a(u)$.
  - $L(u) = L'(u)$, otherwise.

An execution is a sequence $G_0 G_1 \ldots$ such that $G_0 \in \mathcal{G}_{Q_0}$ 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 $R(u)$. 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 $\mathcal{T} \subseteq \mathcal{G}_Q$, the AHN $A^\mathcal{T}_\mathcal{P}$ restricted to $\mathcal{T}$ is defined by the transition system $\langle \mathcal{C} \cap \mathcal{T}, \Rightarrow_\mathcal{T}, \mathcal{C}_0 \cap \mathcal{T} \rangle$ where the relation $\Rightarrow_\mathcal{T}$ is the restriction of $\Rightarrow$ to $(\mathcal{C} \cap \mathcal{T}) \times (\mathcal{C} \cap \mathcal{T})$.

Decision problem

Given a process $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$ with an associated AHN $\mathcal{A}_\mathcal{P} = \langle \mathcal{C}, \Rightarrow, \mathcal{C}_0 \rangle$, we define the *control state reachability* (COVER) as follows:

> Given a control state $q \in Q$, does there exist $G \in \mathcal{C}_0$ and $G' \in \mathcal{C}$ such that $q \in L(G')$ and $G \Rightarrow^* G'$?

Control state reachability is strictly related to parameterized verification of safety properties. The input control state q can 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.

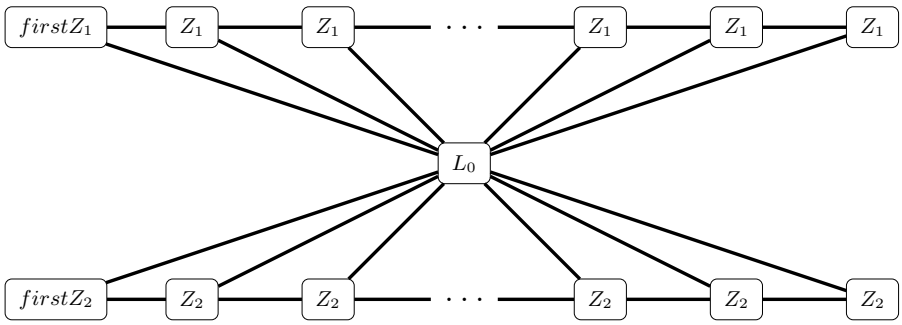

Fig. 1. Butterfly-shaped induced subgraph needed to simulate a Minsky machine.

4 Configurations with Bounded Diameter

As mentioned in the introduction, COVER is undecidable for configurations with unrestricted topology [3]. The problem becomes decidable when configurations are restricted to graphs with *k-bounded paths* (BP_k) for any $k \geq 0$. k-bounded path graphs are graphs in which there exist no simple path with length strictly greater than k. The class BP_k is infinite for any $k > 0$. As an example, with $k = 2$ it includes star-shaped graphs of any order.

Unfortunately, restricting protocol analysis to configurations in BP_k seems to have a limited application in a communication model with selective broadcast. Indeed, we first observe that BP_k does not include the class K consisting of clique graphs of any order. Cliques however 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 graphs in the class K, COVER can be reduced to coverability in a Broadcast Protocol, i.e., in a model in which configurations are multisets of processes defined by communicating automata [6]. Coverability is decidable in Broadcast Protocols in [7]. For these reasons, in this paper we investige COVER in restricted classes of graphs that at least include the class K. The first class we 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 BP_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 [9]. 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 $k > 1$. Indeed, we show next that AHNs working over this class of configurations can be used to simulate the behavior of a deterministic Minsky machine. A deterministic 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. Instructions are 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. There is also a special halting label L_F. The halting problem consists in deciding whether or not the execution that starts from L_0, with both counters set to 0, reaches L_F. The halting problem for deterministic two counter machines is undecidable [14]. The encoding is built in two steps.

We first need to run a protocol that terminates successfully only when the projection of the configuration on an appropriate set of control state is a sort of butterfly (see Figure 1) 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).

To reach such a configuration, we use a process $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$ with $\{L_0, firstZ_1, Z_1, firstZ_2, Z_2, error\} \subseteq Q$ such that if G_0 is an initial configuration in $\mathcal{A}_\mathcal{P} = \langle \mathcal{C}, \Rightarrow, \mathcal{C}_0 \rangle$ and if $G_0 \Rightarrow^* G$ for a configuration $G = \langle V, E, L \rangle$ verifying $L_0 \in L(V)$, then the graph $\theta = \langle V', E', L' \rangle$ represented in Figure 1 is an induced subgraph of G. Furthermore, all vertices $v \in V \setminus V'$ adjacent to a vertex of θ in G are labeled with $error$. We also have that all the nodes labeled with L_0 in G are connected as in θ (when abstracting away the nodes in state $error$). In these graphs θ of Figure 1, the number of nodes is guessed nondeterministically and represents the maximum value reached by the counters during the simulation. Thus, the number of Z_1 and Z_2 can be different. However, there is at least one node labeled Z_1 and one labeled Z_2. The diameter of θ is equal to 2 no matters how many nodes there are labelled with Z_1 or Z_2. The protocol for the first step is described in detail in [4].

Once the configuration is in the desired form, the second step consists in the simulation of the instructions of the encoded Minsky machine. The protocol for this step is shown in Figure 2 (as far as the simulation of the counters is concerned) and 3 (for the simulation of the instructions). More precisely, we build a process $\mathcal{P}'$ obtained by completing the process $\mathcal{P}$ with the processes shown on the Figures 2 and 3.

The simulation works as follows: first if the Minsky machine is at the line labelled with L and m is the value of the first counter (the same reasoning holds for the second counter) then in the corresponding configuration of the AHN there is one node labelled with L which is neighbor of $m - 1$ nodes labelled by NZ_1 and if $m > 0$ this node has also a neighbor labelled by $firstNZ_i$, if $m = 0$ then this same neighbor is labelled by $firstZ_i$. To simulate an increment of the form $L_1 : c_i := c_i + 1$; `goto` L_2, the node of the AHN labelled with L_1 sends an inc_i and the unique node labelled by $nextZ_i$ receives it, acknowledges it by sending an $ackinc_i$ and updates its unique neighbor labelled by Z_i to $nextZ_i$. The decrement works in the same manner except that if the value of the counter c_i is equal to 0 the node labelled with a label of the Minsky machine receives a $zero_i$ otherwise it receives a dec_i. We have then that in $\mathcal{P}'$ there is an execution from an initial configuration which reaches a configuration where at least one

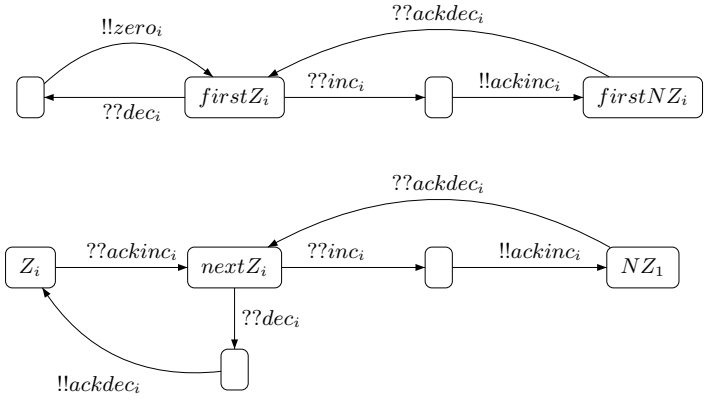

Fig. 2. Simulation of the instructions for counter c_i.

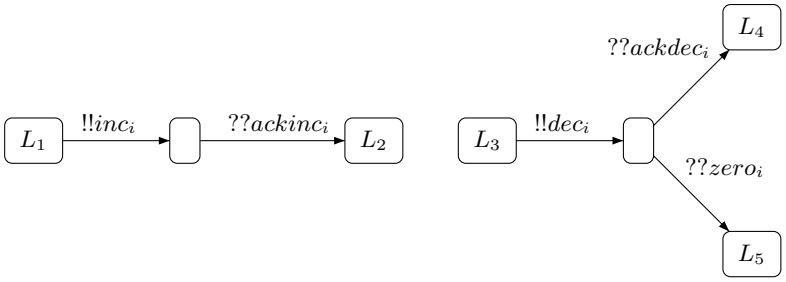

Fig. 3. Encoding ($L_1 : c_i := c_i + 1$; `goto` L_2) and (L_3 : `if` $c_i = 0$ `then goto` L_5 `else` $c_i := c_i - 1$; `goto` L_4)

node is labelled by L_F if and only if the corresponding Minsky machine halts. This allows us to deduce:

Theorem 1. *For $k > 1$,* COVER *restricted to configurations with k-bounded diameter is undecidable.*

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 [11], we obtain an interesting decidable subclass. Indeed, in [11] the authors show that, given two integers $k, d > 0$, the number of graphs whose diameter is smaller than k and whose degree (max number of neighbors) is smaller than d is finite. The Moore bound $M(k, d) = (k(k-1)^d - 2)/(k-2)$

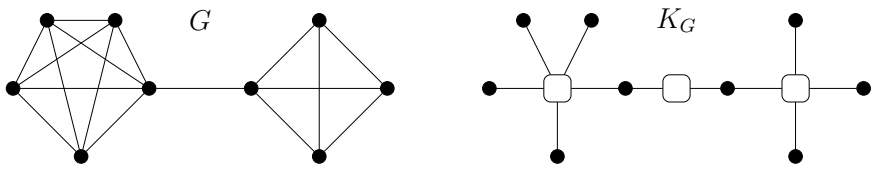

Fig. 4. A graph G and its associated clique graph K_G.

is an upper bound for the size of the largest undirected graph in such a class. The following property then holds.

Theorem 2. *For fixed $k, d > 0$ and given a process $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$,* COVER *restricted to configurations with k-bounded diameter and d-bounded degree is in* PSPACE *in the size of P.*

Proof. From [11], it follows that the number of possible configurations is finite. Thus, COVER is decidable. Since k and d are two fixed constants, the constant $N = M(k, d)$ gives us an upper bound on the number of nodes of the largest graph to be considered. Notice that we only need polynomial space in the size of P to store a graph with size smaller or equal than N. To solve the COVER problem, we define a non deterministic algorithm that first guesses the initial graph G_0 and then explore all possible successor configurations in search for an error state. Since the topology never changes, in the worst case we have to consider all possible relabelings of the initial graph. Thus, the size of the state space is bounded by $|Q|^N$ and it is still polynomial in the size of P.

5 Maximal Clique Graphs with Bounded Paths

In this section we prove decidability for COVER restricted to the class of graphs we call BPC_n (n-Bounded Path maximal Cliques graphs). BPC_n contains both n-bounded path graphs and any clique graph, while being strictly contained in the class of graphs with k-bounded diameter. The class is defined on top of the notion of *maximal clique graphs* associated to a configuration.

Definition 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 4. 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 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 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).*

Let us now study the properties of this class of graphs. We first introduce the following ordering on BPC_n graphs.

Definition 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 injections $f : X_1 \to X_2$ 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 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 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 $f \cup g$). Furthermore, we also have the following property:

Lemma 1. *$G_1 \sqsubseteq G_2$ iff $G_1 \preceq_i G_2$ (G_1 is an induced subgraph of G_2).*

We are now interested in the property of being wqo for the above defined graph orderings. Lemma 1 shows that the ordering $\sqsubseteq$, defined on the maximal clique graph, is equivalent to the induced subgraph ordering on the original graphs. It is well known that the induced subgraph relation is not a wqo for generic graphs (e.g. consider the infinite sequence of rings of increasing size). There are however interesting classes of graphs for which the induced subgraph ordering is wqo. For instance, induced subgraphs is a wqo for the class of graphs for which the length of simple paths is bounded by a constant (bounded path graphs). This result is known as Ding's Theorem [5]. Now observe that given $n \geq 1$ the class BPC_n we are interested in contains cliques of arbitrary order and it also strictly contains the class of $n/2$-bounded path graphs. Interestingly, Ding's result can be extended to the BPC_n class for every $n \geq 1$.

Lemma 2. *For any $n \geq 1$, $(BPC_n, \sqsubseteq)$ is a well-quasi ordering.*

The proof, given in [4], follows Ding's induction method and exploits a decomposition property of bounded path graphs due to Robertson and Seymour.

Given a subset $S \subseteq BPC_n$, we now define its *upward closure* $S \uparrow = \{G' \in BPC_n \mid G \in S \text{ and } G \sqsubseteq G'\}$, i.e., $S \uparrow$ is the set of configurations generated by those in S via $\sqsubseteq$. A set $S \subseteq BPC_n$ is an *upward closed set* w.r.t. to $(BPC_n, \sqsubseteq)$ if $S \uparrow = S$. Since $(BPC_n, \sqsubseteq)$ is a wqo, we obtain that every set of configurations

that is upward closed w.r.t. $(BPC_n, \sqsubseteq)$ has a finite basis, i.e., it can be finitely represented by a finite number of graphs. We can exploit this property to define a decision procedure for the coverability problem. For this purpose, we apply the methodology proposed in [1]. The first property we need is that the transition relation induced by our model is compatible with $\sqsubseteq$.

Lemma 3. *Fixed $n \geq 1$, for every $G_1, G_2, G_1' \in BPC_n$ such that $G_1 \Rightarrow_{BPC_n} G_2$ and $G_1 \sqsubseteq G_1'$, there exists $G_2' \in BPC_n$ such that $G_1' \Rightarrow_{BPC_n} G_2'$ and $G_2 \sqsubseteq G_2'$.*

For a fixed $n \geq 1$, monotonicity ensures that if S is an upward closed set of configurations, then the set of predecessors of S according to $\Rightarrow$, defined as $pre(S) = \{G \mid G \Rightarrow_{BPC_n} G' \text{ and } G' \in S\}$, is still upward closed. Furthermore, we can effectively compute a finite representation of $S \cup pre(S)$.

Lemma 4. *Given a finite basis B of an upward closed set $S \subseteq BPC_n$, there exists an algorithm to compute a finite basis B' of $S \cup pre(S)$ s.t. $S \cup pre(S) = B' \uparrow$.*

This allows us to state the main theorem of this section.

Theorem 3. *Given $n \geq 1$,* COVER *restricted to BPC_n configurations is decidable.*

Proof. It follows from Lemmas 2, 3, and 4 and from the general properties of well structured transition systems [1,2,10]. □

6 Ackermann-hardness of COVER in BPC_n

In the previous section we have proved that, despite COVER is undecidable for AHNs, it becomes decidable when imposing the configurations to be in BPC_n and this for every $n \geq 1$. We prove here, that even if decidable, 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 [18].

A reset net RN is a tuple $\langle P, T, \boldsymbol{m_0} \rangle$ such that P is a finite set of places, T is a finite set of transitions, and $\boldsymbol{m_0}$ is a marking, i.e. a mapping from P to $\mathbb{N}$ that defines the initial number of tokens in each place of the net. A transition $t \in T$ is defined by a mapping ${}^\bullet t$ (preset) from P to $\mathbb{N}$, a mapping $t^\bullet$ (postset), and by a set of reset arcs $t{\downarrow} \subseteq P$. A configuration is a marking $\boldsymbol{m}$. Transition t is enabled at marking $\boldsymbol{m}$ iff ${}^\bullet t(p) \leq \boldsymbol{m}(p)$ for each $p \in P$. Firing t at $\boldsymbol{m}$ leads to a new marking $\boldsymbol{m'}$ defined as $\boldsymbol{m'}(p) = \boldsymbol{m}(p) -^\bullet t(p) + t^\bullet(p)$ if $p \notin t{\downarrow}$, and $\boldsymbol{m'}(p) = 0$ otherwise. We assume that if $p \in t{\downarrow}$ then $t^\bullet(p) = 0$. A marking $\boldsymbol{m}$ is reachable from $\boldsymbol{m_0}$ if it is possible to produce it after firing finitely many times transitions in T. Given a reset net $\langle P, T, \boldsymbol{m_0} \rangle$ and a marking $\boldsymbol{m}$, the coverability problem consists in checking for the existence of a reachable marking $\boldsymbol{m'}$ such that $\boldsymbol{m'}(p) \geq \boldsymbol{m}(p)$ for every $p \in P$. In [18] it is proved that the coverability problem for reset nets is Ackermann-hard.

We start by showing a linear reduction of the coverability problem for reset nets to COVER for the class of AHNs with clique topologies, denoted with K. Note

that K corresponds to BPC_n with $2 \leq n < 4$. Then, we show how to generalize the presented reduction to AHNs with topologies in BPC_n, with $n \geq 4$.

Let $RN = \langle P, T, \boldsymbol{m_0} \rangle$ be a reset net, and let $\boldsymbol{m}$ be a marking. We construct a process $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$ with a control state $q \in Q$ such that $\boldsymbol{m}$ is coverable in RN iff the control state q is reachable in $A^K_{\mathcal{P}}$. We assume, without loss of generality, that both $\boldsymbol{m_0}$ and $\boldsymbol{m}$ contain only one token, i.e. there exist two places p_s and p_e such that $\boldsymbol{m}_0(p_s) = 1$ (resp. $\boldsymbol{m}(p_e) = 1$) and $\boldsymbol{m_0}(p) = 0$ (resp. $\boldsymbol{m}(p) = 0$) for every $p \neq p_s$ (resp. $p \neq p_e$).

We now describe the corresponding process definition $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$. We define $Q_0 = \{q_0\}$, i.e. all the processes are initially in the state q_0. At the beginning the processes perform a simple protocol (depicted in Figure 5) that elects one node as the *master*, and the other nodes become *slaves*.

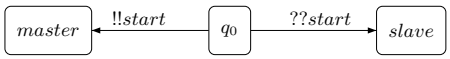

Fig. 5. The *initialization* phase.

The master will control the simulation of the reset net, while the slaves will be used to represent tokens in the net markings. Namely, when a slave process is in the state $q_p \in Q$, it represents one token in the place $p \in P$. For instance, in order to represent the initial marking it is necessary for the master to move one slave in the state q_{p_s}. This is achieved by the protocol in Figure 6.

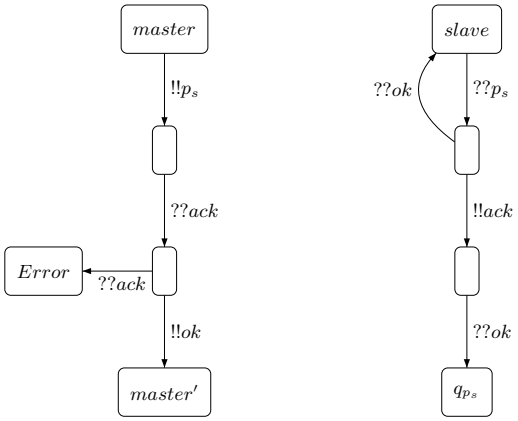

Fig. 6. Generating the initial marking with only one token in p_s.

Note that the protocol can deadlock in two possible cases: either when there is no slave node, or two of them reply with the *ack* message before the master closes the protocol with the *ok* message. If the master completes the protocol by entering in the state *master′*, exactly one slave moved to the state q_{p_s}.

At this stage, the simulation of the net transitions starts. The master in state $master'$ nondeterministically selects one of the possible transitions t, with ${}^\bullet t = \{p_1, \ldots, p_n\}$, $t\downarrow = \{p'_1, \ldots, p'_m\}$, and $t^\bullet = \{p''_1, \ldots, p''_l\}$, and it starts its simulation by performing the following protocol. It first tries to consume the tokens in the preset ${}^\bullet t$ by performing in sequence protocols similar to the one in Figure 6: in this case, it moves processes from the states q_{p_i} to *slave* to simulate the consumption of tokens in the places p_i. After, the reset actions are performed simply by emitting the messages $resetp'_i$, whose effect is to move all nodes in the states $q_{p'_i}$ to the *slave* state. Finally, by performing in sequence the same protocol of Figure 6, it simulates the production of tokens in the places p''_i.

Lemma 5. *Given a marking $\boldsymbol{m}$ containing only one token in p_e, we have that $\boldsymbol{m}$ can be covered in RN iff $A^K_{\mathcal{P}}$ satisfies* COVER *for the state q_{p_e}.*

Proof. The *if* part follows from the fact that every ad hoc network in $A^K_{\mathcal{P}}$ correctly reproduces computations of the reset net (it simply introduces deadlocks that are not relevant as far as the coverability problem is concerned). The *only-if* part is a consequence of the fact that every finite computation of the reset net can be simulated by at least one ad hoc network in $A^K_{\mathcal{P}}$ having a sufficient number of nodes. □

It is easy to see that the above construction does not work for topologies different from the clique. For instance, in the topologies in BPC_n with $n \geq 4$, there are nodes belonging to two distinct maximal cliques. If such nodes are connected to two distinct masters, they could generate interferences among them. In order to cope with this problem, we build another process $\mathcal{P}'$ obtained by replacing the trivial initialization protocol of Figure 5 with the most sophisticated one depicted in Figure 7.

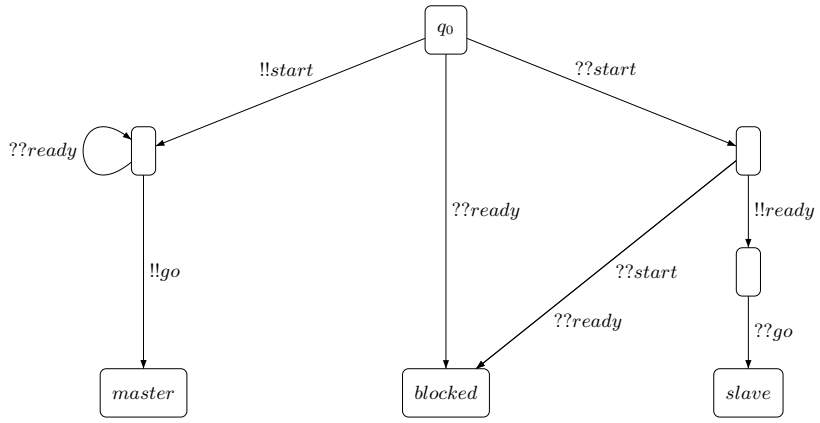

Fig. 7. The *initialization* phase for BPC_n with $n \geq 4$.

After the execution of this initialization protocol, we have the guarantee that *slave* processes do not generate interferences between two distinct *master* nodes.

In fact, at the end of the protocol we have the guarantee that every *slave* node is connected to exactly one *master*, and all of its other neighbors are *blocked*.

Lemma 6. *Given a marking $\boldsymbol{m}$ containing only one token in p_e and $n \geq 4$, we have that $\boldsymbol{m}$ can be covered in RN iff $A_{\mathcal{P}'}^{BPC_n}$ satisfies* COVER *for the state q_{p_e}.*

We can conclude with the main result of this section.

Theorem 4. *For every $n \geq 2$,* COVER *restricted to BPC_n configurations is non-primitive recursive.*

Proof. The result follows from the Ackermann-hardness of reset nets [18], and from Lemma 5 and Lemma 6. It is sufficient to note that K coincides with BPC_n with $2 \leq n < 4$, and to observe that $\mathcal{P}$ is obtained in linear time from the reset net RN.

7 Broadcast vs Unicast Communication

Although broadcast communication is specifically devised for networks in which nodes have no complete knowledge of the surrounding topology, unicast (point-to-point) communication is often provided, e.g., to exchange information after the acquisition of the information on the vicinity of a node. We investigate here the relationship between the coverability problem for unicast and broadcast communication. Specifically, we show that the two problems can be kept separated (i.e. the problem is more difficult for broadcast) in all the classes of graphs studied in [3] and in the present paper.

For this analysis we first introduce two primitives $!a$ and $?a$ for unicast communication. As in CCS [13], when a process sends a message a (action $!a$) it synchronizes with only one process that is in a state in which it is ready to receive a (action $?a$). The receiving process is nondeterministically chosen among those ready to receive a. For unicast communication the definition of a process $\mathcal{P} = \langle Q, \Sigma, R, Q_0 \rangle$ is modified in the component R that is now a subset of $Q \times (\{\tau\} \cup \{!a, ?a\} \mid a \in \Sigma) \times Q$. The operational semantics is obtained via a transition relation $\Rightarrow$ defined as follows: given two configurations $G = \langle V, E, L \rangle$ and $G' = \langle V', E', L' \rangle$, we have $G \Rightarrow G'$ iff G and G' have the same underlying structure, i.e., $V = V'$ and $E = E'$, and, in addition to the conditions for τ of Section 3, the following condition on L and L' defines a case in which a transition may take place:

- there exists $v \neq w \in V$ such that $(L(v), !a, L'(v))$ and $(L(w), ?a, L'(w))$ are both in R, and $L(u) = L'(u)$ for all u in $V \setminus \{v, w\}$.

For the sake of clarity, in the rest of the section we name AHN^b the model with broadcast (and no unicast) and AHN^u the model with unicast (and no broadcast).

The coverability problem for AHN^u can be reduced to the corresponding problem for AHN^b. Indeed, unicast communication can be simulated via broadcast messages via a protocol like the one in Figure 6. The encoding introduces

deadlocks that are not relevant as far as the COVER problem is concerned. The following theorem then holds.

Theorem 5. *The control state reachability problem for AHN^u is in* EXPSPACE.

Proof. We first show that we can restrict our attention to clique graphs only. Indeed, given a state q, if there exist G_0 and G_1 with n nodes s.t. $G_0 \Rightarrow_G^* G_1$ and q is a label in G_1, then there exist two cliques K_0 and K_1 with order n s.t. $K_0 \Rightarrow_G^* K_1$ and q is a label in K_1. This property follows from the observation that for any graph G with $n' \leq n$ nodes, there exists a clique graph with n nodes such that $G \preceq_s K_n$. Let K_0 be the clique such that $G_0 \preceq_s K_0$. Since $G_0 \Rightarrow_G^* G_1$, by exploiting the monotonicity of unicast communication w.r.t. subgraph ordering, we have that there exists K_1 s.t. $K_0 \Rightarrow_G^* K_1$ and q is a label in K_1. Now we observe that control state reachability in the class of clique graphs can be reduced to coverability in a Petri net in which each place corresponds to a state in Q. The initial marking is produced by firing transitions that (a nondeterministically chosen number of) tokens in the places in Q_0. For each unicast communication step involving a pair of nodes in state q and q', we add a transition with q and q' in the preset, and the corresponding target states in the postset. It follows then from classical results on Petri nets [15] that we can use an EXPSPACE decision procedure for deciding COVER for AHN^u.

From this property and from the undecidability result for coverability in AHN^b (for unrestricted topologies), we observe that there cannot be any recursive encoding of coverability in AHN^b into the corresponding problem in AHN^u. Furthermore, in the case of cliques with bounded paths, from Theorem 4, we have that there is no primitive recursive encoding of coverability in AHN^b. This way we separate the difficulty of the coverability problem in the two types of communication schemes.

8 Conclusions

In this paper we have extended the decidability result for verification of networks with bounded path topology presented in [3] to a larger and more interesting class of graphs. The new class consists of topologies in which the corresponding maximal cliques are connected by paths of bounded length. Furthermore, we have characterized the complexity of the corresponding decision procedure and compared the expressiveness of broadcast and unicast communication. As a future work, we plan to study decidability issues in presence of communication and node failure and to consider extensions of the ad hoc network model with features like timing information and structured messages.

References

1. P. A. Abdulla, C. Čerāns, B. Jonsson, and Y.-K. Tsay. General decidability theorems for infinite-state systems. In *LICS'96*, pages 313–321. IEEE Computer Society, 1996.

2. P. A. Abdulla, C. Čerāns, B. Jonsson, and T. Y.-K. Algorithmic analysis of programs with well quasi-ordered domains. *Inf. Comput.*, 160(1-2):109–127, 2000.
3. G. Delzanno, A. Sangnier, and G. Zavattaro. Parameterized Verification of Ad Hoc Networks. In *CONCUR'10*, volume 6269 of *LNCS*, pages 313–327. Springer, 2010.
4. G. Delzanno, A. Sangnier, and G. Zavattaro. On the Power of Cliques in the Parameterized Verification of Ad Hoc Networks. Technical Report DISI-TR-11-01, DISI-University of Genova, 2011.
5. G. Ding. Subgraphs and well quasi ordering. *J. of Graph Theory*, 16(5):489 – 502, 1992.
6. E. A. Emerson and K. S. Namjoshi. On model checking for non-deterministic infinite-state systems. In *LICS'98*, pages 70–80. IEEE Computer Society, 1998.
7. J. Esparza, A. Finkel, and R. Mayr. On the verification of broadcast protocols. In *LICS'99*, pages 352–359. IEEE Computer Society, 1999.
8. A. Fehnker, L. van Hoesel, and A. Mader. Modelling and verification of the LMAC protocol for wireless sensor networks. In *IFM'07*, volume 4591 of *LNCS*, pages 253–272. Springer, 2007.
9. Y. Fernandess and D. Malkhi. K-clustering in wireless ad hoc networks. In *POMC'02*, pages 31–37. ACM, 2002.
10. A. Finkel and P. Schnoebelen. Well-structured transition systems everywhere! *Theoret. Comp. Sci.*, 256(1-2):63–92, 2001.
11. A. Hoffman and R. Singleton. On Moore graphs with diameter 2 and 3. *IBM J. Res. Develop.*, 4:497–504, 1960.
12. R. Meyer. On boundedness in depth in the pi-calculus. In *IFIP TCS'08*, volume 477–489 of *IFIP*, pages 477–489. Springer, 2008.
13. R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
14. M. Minsky. *Computation: finite and infinite machines*. Prentice Hall, 1967.
15. C. Rackoff. The covering and boundedness problems for vector addition systems. *Theoret. Comp. Sci.*, 6:223–231, 1978.
16. F. Rosa Velardo. Depth boundedness in multiset rewriting systems with name binding. In *RP*, volume 6227 of *Lecture Notes in Computer Science*, pages 161–175. Springer, 2010.
17. M. Saksena, O. Wibling, and B. Jonsson. Graph grammar modeling and verification of Ad Hoc Routing Protocols. In *TACAS'08*, volume 4963 of *LNCS*, pages 18–32. Springer, 2008.
18. P. Schnoebelen. Revisiting Ackermann-Hardness for Lossy Counter Machines and Reset Petri Nets. In *MFCS'10*, volume 6281 of *LNCS*, pages 616–628. Springer, 2010.
19. A. Singh, C. R. Ramakrishnan, and S. A. Smolka. Query-Based model checking of Ad Hoc Network Protocols. In *CONCUR'09*, volume 5710 of *LNCS*, pages 603–61. Springer, 2009.
20. T. Wies, D. Zufferey, and T. A. Henzinger. Forward analysis of depth-bounded processes. In *FOSSACS'10*, volume 6014 of *LNCS*, pages 94–108. Springer, 2010.