Distrib. Comput. (2017) 30:87–102
DOI 10.1007/s00446-016-0279-6

Reliable broadcast with respect to topology knowledge

Aris Pagourtzis[1] · **Giorgos Panagiotakos**[2] · **Dimitris Sakavalas**[1]

Received: 3 June 2015 / Accepted: 7 July 2016 / Published online: 16 July 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract We study the Reliable Broadcast problem in incomplete networks against a Byzantine adversary. We examine the problem under the *locally bounded adversary model* of Koo (Proceedings of the 23rd annual ACM symposium on principles of distributed computing, PODC '04, St. John's, Newfoundland, Canada, 25–28 July 2004, ACM New York pp 275–282, 2004) and the *general adversary model* of Hirt and Maurer (Proceedings of the 16th annual ACM symposium on principles of distributed computing, PODC '97, Santa Barbara, California, USA, August 21–24, 1997 ACM, New York pp 25–34, 1997) and explore the tradeoff between the level of topology knowledge and the solvability of the problem. In order to explore this tradeoff we introduce the *partial knowledge model* which captures the situation where each player has arbitrary topology knowledge. We refine the local pair-cut technique of Pelc and Peleg (Inf Process Lett 93(3):109–115, 2005) in order to obtain impossibility results for every level of topology knowledge and any type of corruption distribution. On the positive side we devise protocols that match the obtained bounds, and thus, exactly characterize the classes of graphs in which Reliable Broadcast is possible. Among others, we show that Koo's Certified Propagation Algorithm (CPA) is *unique*, against locally bounded adversaries in ad hoc networks, among all *safe algorithms*, i.e., algorithms which never cause a node to decide on an incorrect value. This means that CPA can tolerate as many local corruptions as any other safe algorithm; this settles an open question posed by Pelc and Peleg. We also provide an adaptation of CPA achieving reliable broadcast against general adversaries and prove that this algorithm too is unique under this model. To the best of our knowledge this is the first optimal algorithm for Reliable Broadcast in generic topology ad hoc networks against general adversaries.

Keywords Partial knowledge · Reliable broadcast · Byzantine adversary · Locally bounded adversary · General adversary

1 Introduction

A fundamental problem in distributed networks is Reliable Broadcast (Byzantine Generals), in which the goal is to distribute a message correctly despite the presence of Byzantine faults. That is, an adversary may control several nodes and be able to make them deviate from the protocol arbitrarily by blocking, rerouting, or even altering a message that they should normally relay intact to specific nodes. Even in this case, a Reliable Broadcast protocol must guarantee that all non-corrupted (honest) nodes *decide* on the correct value. The *decision* of a player can be typically modeled as the output of this player by the end of the execution. In general, agreement problems have been primarily studied under the threshold adversary model, where a fixed upper bound t is set for the number of corrupted players and broadcast can be achieved if and only if $t < n/3$, where n is the total number of players. The Broadcast problem has been extensively studied in complete networks under the threshold adversary model mainly in the period from 1982, when it was introduced by

✉ Dimitris Sakavalas
sakaval@corelab.ntua.gr

Aris Pagourtzis
pagour@cs.ntua.gr

Giorgos Panagiotakos
g.panagiotakos@di.uoa.gr

[1] School of Electrical and Computer Engineering, National Technical University of Athens, 15780 Athens, Greece

[2] Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, 15784 Athens, Greece

Lamport, Shostak and Pease [11], to 1998, when Garay and Moses [5] presented the first fully polynomial Broadcast protocol optimal in resilience and round complexity.

The case of Reliable Broadcast under a threshold adversary in incomplete networks has been studied to a much lesser extent, in a study initiated in [1,2,10], mostly through protocols for Secure Message Transmission which, combined with a Broadcast protocol for complete networks, yield Broadcast protocols for incomplete networks. Naturally, connectivity constraints are required to hold in addition to the $n/3$ bound. Namely, at most $t < c/2$ corruptions can be tolerated, where c is the network connectivity, and this bound is tight [1].

In the case of an honest dealer, particularly meaningful in wireless networks, the impossibility threshold of $n/3$ does not hold; for example, in complete networks with an honest dealer the problem becomes trivial regardless of the number of corrupted players. However, in incomplete networks the situation is different. A small number of corrupted players may manage to block the entire protocol if they control a critical part of the network, e.g. if they form a separator of the graph. It therefore makes sense to define criteria (or parameters) depending on the structure of the graph, in order to bound the number or restrict the distribution of corruptions that can be tolerated.

An approach in this direction is to consider topological restrictions on the adversary's corruption capacity. We will first focus on local restrictions, the importance of which comes, among others, from the fact that they may be used to derive criteria which can be employed in ad hoc networks. Such a paradigm is the *t-locally bounded adversary model*, introduced in [9], in which at most a certain number t of corruptions are allowed in the neighborhood of every node.

The locally bounded adversarial model is particularly meaningful in real-life applications and systems. For example, in social networks it is more likely for an agent to have a quite accurate estimation of the maximum number of malicious agents that may appear in its neighborhood, than having such information, as well as knowledge of connectivity, for the whole network. In fact, this scenario applies to all kinds of networks, where each node is assumed to be able to estimate the number of traitors in its close neighborhood. It is also natural for these traitor bounds to vary among different parts of the network. Motivated by such considerations, in this work we will introduce a generalization of the t-locally bounded model.

1.1 Related work

Considering t-locally bounded adversaries, Koo [9] proposed a simple, yet powerful protocol, namely the *Certified Propagation Algorithm* (CPA) (a name coined by Pelc and Peleg in [15]), and applied it to networks of specific topology. CPA is based on the idea that a set of $t + 1$ neighbors of a node always contain an honest one. Pelc and Peleg [15] considered the t-locally bounded model in generic graphs and gave a sufficient topological condition for CPA to achieve Broadcast. They also provided an upper bound on the number of corrupted players t that can be locally tolerated in order to achieve Broadcast by any protocol, in terms of an appropriate graph parameter; they left the deduction of tighter bounds as an open problem. To this end, Ichimura and Shigeno [8] proposed an efficiently computable graph parameter which implies a more tight, but not exact, characterization of the class of graphs on which CPA achieves Broadcast. It had remained open until very recently to derive a tight parameter revealing the maximum number of traitors that can be locally tolerated by CPA in a graph G with dealer D. Such a parameter is implicit in the work of Tseng et al. [16], who gave a necessary and sufficient condition for CPA Broadcast. Finally, in [12] such a graph parameter was presented explicitly, together with an efficient 2-approximation algorithm for computing its value.

A more general approach regarding the adversary structure was initiated by Hirt and Maurer in [7] where they studied the security of multiparty computation protocols with respect to an *adversary structure*, i.e. a family of sets of players, such that the adversary may entirely corrupt any set in the family. This line of work has yielded results on Broadcast against a general adversary in complete networks [4] but, to the best of our knowledge, the case of Broadcast against general adversaries in incomplete networks has not been studied as such.[1] A study on the related problem of Iterative Approximate Byzantine Consensus against general adversaries can be seen in [17] where a similar model for the ad hoc case is considered.

1.2 Our Results

In this work we study the tradeoff between the level of topology knowledge and the solvability of the problem, under various adversary models. In the course of this study we consider the natural class of *safe* Broadcast algorithms, i.e., algorithms that never cause a player to decide on an incorrect value. The importance of safeness is pointed out in [15], where it is regarded as a basic requirement of a Broadcast algorithm; it guarantees that even if all players do not have sufficient information to decide on the dealer's value, no one will eventually decide on an incorrect value or accept false data.

We first consider a natural generalization of the t-locally bounded model, namely the *non-uniform t-locally bounded model* which subsumes the (uniform) model studied so far.

[1] Some related results are implicit in [10], but in the problem studied there, namely Secure Message Transmission, additional secrecy requirements are set which are out of the scope of our study.

The new model allows for a varying bound on the number of corruptions in each player's neighborhood. We address the issue of locally resilient Broadcast in the non-uniform model. We present a new necessary and sufficient condition for CPA to be t-locally resilient by extending the notion of *local pair cut* of Pelc and Peleg [15] to the notion of *partial local pair cut*. Note that although equivalent conditions exist [12,16], the simplicity of the new condition allows to settle the open question of CPA Uniqueness [15] in the affirmative: we show that if any safe algorithm achieves Broadcast in an ad hoc network then so does CPA. We next prove that computing the validity of the condition is NP-hard and observe that the latter negative result also has a positive aspect, namely that a polynomially bounded adversary is unable to design an optimal attack unless P = NP.

We next shift focus on networks of known topology and devise an optimal resilience protocol, which we call *Path Propagation Algorithm* (PPA). Using PPA we prove that a topological condition which was shown in [15] to be necessary for the existence of a Broadcast algorithm is also sufficient. Thus, we manage to exactly characterize the class of networks for which there exists a solution to the Broadcast problem. On the downside, we prove that it is NP-hard to compute an essential decision rule of PPA, rendering the algorithm impractical. However, we are able to provide an indication that probably no efficient protocol of optimal resilience exists, by showing that efficient algorithms through which players always take the same decisions as they would if they ran PPA do not exist if P $\neq$ NP.

We then take one step further, by considering a hybrid between ad hoc and known topology networks: each node knows a part of the network, namely a connected subgraph containing itself. We propose a protocol for this setting as well, namely the *Generalized Path Propagation Algorithm* (GPPA). We use GPPA to show that this *partial knowledge* model allows for Broadcast algorithms of increased resilience.

Finally, we study the general adversary model and show that an appropriate adaptation of CPA is unique against general adversaries in ad hoc networks. To the best of our knowledge this is the first algorithm for Reliable Broadcast in generic topology ad hoc networks against a general adversary. We show an analogous result for known topology networks, which however can be obtained implicitly from [10] as mentioned above.

We conclude by discussing how to extend our results to the case of a corrupted dealer by simulating Broadcast protocols for complete networks.

A central tool in our work is a refinement of the local pair-cut technique of Pelc and Peleg [15] which proves to be adequate for the exact (in most cases) characterization of the class of graphs for which Broadcast is possible for any level of topology knowledge and type of corruption distribution. A useful by-product of practical interest is that the refined cuts can be used to determine the exact subgraph in which Broadcast is possible.

For clarity we have chosen to present our results for the t-local model first (Sects. 3, 4, 5), for which proofs and protocols are somewhat simpler and more intuitive, and then for the more involved general adversary model (Sect. 6).

2 Problem and model definition

As we previously mentioned, the goal of Reliable Broadcast is to have some designated player, called the dealer, consistently send an input value to all other players of the network even in the presence of a central adversary which corrupts some players and controls them in some extend. Therefore the effectiveness of a Reliable Broadcast protocol should be considered w.r.t. the capacity of the adversary, i.e. the adversary model.

Adversary model $\mathcal{T}$ An adversary model $\mathcal{T}$ defines the sets of players that can be corrupted by the $\mathcal{T}$-adversary (possible/admissible corruption sets) as well as the possible behavior of the corrupted players, i.e., all the possible actions that the corrupted players can execute. The adversarial behavior in an execution of a distributed protocol can be described exactly by the set and the actions of the corrupted players. We consider the byzantine adversary model which imposes no restrictions on the behavior of the corrupted players. Regarding the possible corruption sets we consider the t-locally bounded model and the *general adversary model* which will be defined in the following.

The network model that we use in this paper is defined below.

Network model We assume that the players V are arranged in a communication network which is represented by a graph $G = (V, E)$ where E is a set of undirected, authenticated channels of communication between pairs of players.

In this paper we address the problem of *Reliable Broadcast with an honest dealer* in generic (possibly incomplete) networks. For brevity we will refer to it simply as the *Broadcast problem*. The problem is trivial in complete networks; we will consider the case of incomplete networks here. As we will see in Sect. 7, the case of an honest dealer in incomplete networks essentially captures the difficulty of the general problem, where even the dealer may be corrupted. A protocol for the general case can be devised by simulating the message exchange of Broadcast protocols in complete networks, which have been extensively studied. We consider deterministic protocols for the solution of the problem.

Definition 1 (*Reliable broadcast with honest dealer/broadcast*) Let $V = \{v_1, \ldots, v_n\}$ be the set of n play-

ers arranged in a communication network $G = (V, E)$ as described above and X be a finite domain. Consider a distributed protocol Π among players V, where player $D \in V$ (called the *dealer*) holds an input value $x_D \in X$ and every player $v \in V$ finally decides on a single output value $y_v \in X$. Also assume any adversary model $\mathcal{T}$ s.t. the dealer can not be corrupted. Protocol Π achieves Broadcast in (G, D) under the adversary model $\mathcal{T}$ if for any possible corruption set T and any adversarial behavior of this set conforming to $\mathcal{T}$, all honest players decide on the dealer's input value, i.e., $\forall v \in V \setminus T, y_v = x_D$.

Termination is also required by the standard definition of Broadcast, i.e., it must be guaranteed that all correct players eventually terminate the protocol. As usual in the related literature, we omit the termination study, which is often implied directly by the algorithm's correctness. We discuss this issue briefly in Sect. 8.

In the sequel, we will informally use the term *Broadcast protocol* (or algorithm) for any distributed algorithm that aims to achieve Broadcast, no matter if it is successful or not.

We will now formally define the adversary model by generalizing the notions originally developed in [9,15]. We will also define basic notions and terminology that we will use throughout the paper. We refer to the participants of the protocol by using the terms *node* and *player* interchangeably.

Corruption function Taking into account that each player might be able to estimate her own upper bound on the corruptions of its neighborhood, as discussed earlier, we introduce a model in which the maximum number of corruptions in each player's neighborhood may vary from player to player. We thus generalize the standard t-locally bounded model [9] in which a uniform upper bound on the number of local corruptions was assumed. Here we consider $t : V \to \mathbb{N}$ to be a *corruption function* over the set of players V.

Non-uniform t-locally bounded adversary model The network is represented by a graph $G = (V, E)$ and one player $D \in V$ is the dealer (sender) as explained before. A corruption function $t : V \to \mathbb{N}$ is also given, implying that an adversary may corrupt at most $t(u)$ nodes in the neighborhood $\mathcal{N}(u)$ of each node $u \in V$. The family of *t-local* sets (defined below) plays an important role in our study since it coincides with the family of admissible corruption sets.

Definition 2 (*t-local set*) Given a graph $G = (V, E)$ and a function $t : V \to \mathbb{N}$ a *t-local set* is a set $C \subseteq V$ for which $\forall u \in V,\ |\mathcal{N}(u) \cap C| \leq t(u)$. For $V' \subseteq V$ a *t-local w.r.t. V' set* is a set $C \subseteq V$ for which $\forall u \in V',\ |\mathcal{N}(u) \cap C| \leq t(u)$.

Uniform vs non-uniform model Obviously the original t-locally bounded model corresponds to the special case of t being a constant function. Hereafter we will refer to the original t-locally bounded model as the *Uniform Model* as opposed to the *Non-Uniform Model* which we introduce here. Hereafter we will also refer to the Non-Uniform Model simply as the t-locally bounded model.

In our study we will often make use of node-cuts which separate some players from the dealer, i.e., node-cuts that do not include the dealer. From here on we will simply use the term *cut* to denote such a node-cut. The notion of *t-local pair cut* was introduced in [15] and is crucial in defining the bounds for which correct dissemination of information in a network is possible.

Definition 3 (*t-local pair cut*) Given a graph $G = (V, E)$ and a function $t : V \to \mathbb{N}$, a pair of t-local sets C_1, C_2 s.t. $C_1 \cup C_2$ is a cut of G is called a *t-local pair cut*.

The next definition extends the notion of t-local pair cut and is particularly useful in describing capability of achieving Broadcast in networks of unknown topology (ad hoc networks) where each player's knowledge of the topology is limited in its own neighborhood.

Definition 4 (*t-partial local pair cut*) Let C be a cut of G, partitioning $V \setminus C$ into sets $A, B \neq \emptyset$ s.t. $D \in A$. C is a *t-partial local pair cut* (*t-plp cut)* if there exists a partition $C = C_1 \cup C_2$ where C_1 is t-local and C_2 is t-local w.r.t. B.

In the uniform model the *Local Pair Connectivity* (LPC(G, D)) [15] parameter of a graph G with dealer D, was defined to be the minimum integer t s.t. G has a t-local pair cut. To define the corresponding notion in the non-uniform model we need to define a (partial) order among corruption functions. Nevertheless, as implied by Theorems 2 and 3, for reasoning about the feasibility of Broadcast it suffices to consider the following decision problem:

Definition 5 ($pLPC$) Given a graph G, a dealer D and a corruption function t determine whether there exists a t-plp cut in G.

2.1 Protocol properties

We next define some protocol properties, introduced in [15], that facilitate our study.

Definition 6 (*t-locally resilient algorithm for*(G, D)) An algorithm which achieves Broadcast in a given graph G with dealer D for any t-local corruption set T and any behavior of T is called *t-locally resilient* for (G, D).

According to the definition Broadcast, a t-locally resilient algorithm for (G, D) is an algorithm which achieves Broadcast in (G, D) under the t-locally bounded adversary model.

Definition 7 (*safe / t-locally safe algorithm*) An algorithm which never causes an honest node to decide on (output) an incorrect value, for any graph-dealer pair (G, D) under any corruption set and any behavior of it (that is, under any adversary model), is called *safe*.

An algorithm which never causes an honest node to decide on an incorrect value under any t-local corruption set and any behavior of it, for any graph-dealer pair (G, D), is called *t-locally safe*.

Note that a safe algorithm might still fail, particularly by not correctly delivering the message to all nodes of the network. By not correctly we mean that the information received by a player is not sufficient for it to decide. Essentially, a safe Broadcast algorithm ensures that a player will decide on a value only in the case she can undoubtedly deduce from her view (input and exchanged messages) that this is the actual value of the dealer.

Observe that an algorithm is t-locally safe if it satisfies the desired property for every instance (G, D). On the other hand, the algorithm is t-locally resilient for (G, D) if it satisfies the property for the specific instance (G, D). Therefore, it might be the case that an algorithm is t-locally resilient for (G, D) but not t-locally safe, even if the first trivially implies that the safeness property holds for (G, D).

Definition 8 (*Uniqueness of Algorithm*) Let $\mathcal{A}$ be a family of algorithms. An algorithm A is unique (for Broadcast) among algorithms in $\mathcal{A}$ if the existence of an algorithm of family $\mathcal{A}$ which achieves Broadcast in an instance (G, D) implies that A also achieves Broadcast in (G, D).

A unique algorithm A among $\mathcal{A}$, naturally defines the class of instances (G, D) in which the problem is solvable by $\mathcal{A}$-algorithms, namely the ones that A achieves Broadcast in.

3 Ad Hoc networks

3.1 Certified Propagation Algorithm (CPA)

The Certified Propagation algorithm [9] uses only local information and thus is particularly suitable for ad hoc networks. CPA is probably the only safe Broadcast algorithm known up to now for the t-locally bounded model, which does not require knowledge of the network topology or use topology discovery subroutines.

Probably another, more complex, algorithm for this setting could be devised by employing a topology discovery algorithm (e.g. variation of [13]), and then use the topology knowledge obtained to execute some known Broadcast algorithm which requires topology knowledge (e.g. RPA presented in [15]). CPA does not use any topology discovery subroutine; despite its simplicity and minimal propagation (a player only propagates the value she decides to all her neighbors) it proves to be of optimal resilience (unique). The latter means that one cannot achieve better solvability of the problem by employing more complex schemes. Moreover the combination of the results of the current section with those of Sects. 4, 5 imply that there are instances in which the problem is not solvable under the Ad Hoc model but is solvable assuming higher level of topology knowledge. This suggests that employing any topology discovery topology algorithm in the ad hoc model does not provide any useful information which will affect the solvability of the problem.

Protocol 1, presented here, is a modification of the original CPA that can be employed under the generalized corruption model introduced here. Namely a node v, upon reception of $t(v) + 1$ messages with the same value x from $t(v) + 1$ distinct neighbors, decides on x, sends it to all neighbors and terminates. The description of the protocol follows:

Protocol 1: ***Certified Propagation Algorithm (CPA) for the Non-uniform model***

Input (for each node v): Dealer's label D, labels of v's neighbors, corruption bound $t(v)$.
Message format: A single value $x \in X$.

Code for D: send value $x_D \in X$ to all neighbors, decide on x_D and terminate.

Code for $v \in \mathcal{N}(D)$: upon reception of x_D from the dealer, decide on x_D, send it to all neighbors and terminate.

(* *certified propagation rule* *)

Code for $v \notin \mathcal{N}(D) \cup D$: upon reception of $t(v) + 1$ messages with the same value x from $t(v)+1$ distinct neighbors, decide on x, send it to all neighbors and terminate.

As shown in [9], CPA is a t-locally safe Broadcast algorithm. The proof is given for completeness.

Theorem 1 *CPA is t-locally safe.*

Proof We will show that if a player decides on a value x through CPA then $x = x_D$. Assume on contrary that there is a set of players $V' \subseteq V$ that decide on values different than x_D. Let v be the player of V' that decides in the earliest round among all players in V', i.e., the first player to make an incorrect decision, and assume that v decides on $x \neq x_D$. v cannot be a neighbor of the dealer since all neighbors of the dealer only decide on x_D as can be shown in the respective decision rule of CPA. Therefore v has received $t(v)+1$ copies of x from $t(v) + 1$ distinct neighbors. Since at most $t(v)$ neighbors can be corrupted, at least one honest player has decided in $x \neq x_D$ before v. A contradiction to the fact that v is the first player to make an incorrect decision. □

3.2 CPA uniqueness in ad hoc networks

Based on the above definitions we can now prove the *CPA uniqueness conjecture* for ad hoc networks, which was posed as an open problem in [15]. The conjecture states that no algorithm can locally tolerate more corrupted nodes than CPA in networks of unknown topology.

We consider only the class of *t-locally safe* Broadcast algorithms. We assume the ad hoc network model, as described e.g. in [15]. In particular we assume that nodes know only their own labels, the labels of their neighbors and the label of the dealer. We call a distributed Broadcast algorithm that operates under these assumptions an *ad hoc Broadcast algorithm*.

Theorem 2 (Sufficient Condition) *Given a graph G, a corruption function t and a dealer D, if no t-plp cut exists, then CPA is t-locally resilient for* (G, D).

Proof Suppose that no t-plp cut exists in G. Assume an execution of CPA where the actual corruption set is T. By definition, T is t-local, since we are in the t-locally bounded adversary model; clearly $T \cup \mathcal{N}(D)$ is a cut on G as defined before (i.e. not including node D). Since T is t-local and $T \cup \mathcal{N}(D)$ is not a t-plp cut there must exist $u_1 \in V \setminus (T \cup \mathcal{N}(D) \cup D)$ s.t. $|\mathcal{N}(u_1) \cap (\mathcal{N}(D) \setminus T)| \geq t(u_1) + 1$. Since u_1 is honest and all players in $\mathcal{N}(D) \setminus T$ will trivially decide on the correct value x_D through CPA as direct neighbors of the dealer, u_1 will receive $t(u_1)$ copies of x_D and decide on the *correct* dealer's value x_D. Let us now use the same argument inductively to show that every honest node will eventually decide on the *correct* value x_D through CPA. Let $C_k = (\mathcal{N}(D) \setminus T) \cup \{u_1, u_2, \ldots, u_{k-1}\}$ be the set of the honest nodes that have decided until a certain round of the protocol, and assume that they decided on the correct value x_D. Then $C_k \cup T$ is a cut. Since T is t-local, by the same argument as before there exists a node u_k s.t. $|C_k \cap \mathcal{N}(u_k)| \geq t(u_k) + 1$ and u_k will decide *correctly* on x_D. Eventually all honest players will *correctly* decide on x_D. Thus CPA is t-locally resilient in G. □

Observe that the latter proof does not explicitly use the fact that CPA is t-locally safe. Instead, we inductively show that in every step (before all terminate), there are some nodes which decide and that all of them decide correctly. A slight modification of the proof can be used as an alternative proof for CPA's t-local safety since in the induction hypothesis we assume that all decided nodes have decided on the correct value.

Theorem 3 (Necessary Condition) *Let $\mathcal{A}$ be a t-locally safe ad hoc Broadcast algorithm. Given a graph G, a corruption function t and a dealer D, if a t-plp cut exists, then $\mathcal{A}$ is not t-locally resilient in* (G, D).

Proof Assume the partition of set V in the sets A, B, T, H such that $C = T \cup H$ is a t-plp cut in graph G with dealer D which disconnects the node sets A, B. Let T be the t-local set of the cut partition and H the t-local w.r.t. to B set (Fig. 1). Let G' be a graph that results from G if we remove some edges that connect nodes in $A \cup T \cup H$ with nodes in H so that the set H becomes t-local in G' (e.g. we can remove all edges that connect nodes in $A \cup T \cup H$ with nodes in H). Note that the existence of a set of edges that guarantees such a property is implied by the fact that H is t-local w.r.t. B.

The proof is by contradiction. Suppose that there exists a t-locally safe Broadcast algorithm $\mathcal{A}$ which is t-locally resilient in graph G with dealer D. We consider the following executions σ and σ' of $\mathcal{A}$:

- Execution σ is on the graph G with dealer D, with dealer's value $x_D = 0$, and corruption set T; in each round, each corrupted player in T performs the actions that its corresponding player performs in the respective round of execution σ' (where T is a set of honest players).
- Execution σ' is on the graph G' with dealer D, with dealer's value $x_D = 1$, and corruption set H; in each round, each corrupted player in H performs the actions that its corresponding player performs in the respective round of execution σ (where H is a set of honest players).

Although the above definitions of σ, σ' may seem circular, in fact the actions of players are well defined as is explained in the note after the proof. Note that T, H are admissible corruption sets in G, G' respectively due to their t-locality. It is easy to see that $H \cup T$ is a cut which separates D from B in both G and G' and that actions of every node of this cut are identical in both executions σ, σ'. Consequently, the actions of any honest node $w \in B$ must be identical in both

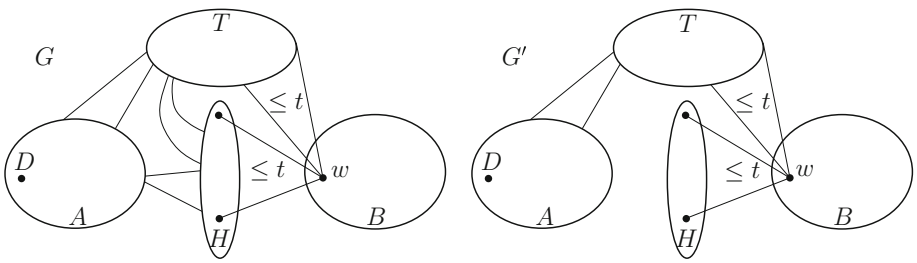

Fig. 1 Graphs G and G'

executions. Since, by assumption, algorithm $\mathcal{A}$ is t-locally resilient on G with dealer D, w must decide on the dealer's message 0 in execution σ on G with dealer D, and must do the same in execution σ' on G' with dealer D. However, in execution σ' the dealer's message is 1. Therefore $\mathcal{A}$ makes w decide on an incorrect message in (G', D). This contradicts the assumption that $\mathcal{A}$ is locally safe. □

Note on the proof of Theorem 3 Although the argument of the two simultaneous executions σ, σ' is standard in the literature (e.g. [1,9,10,15]), it may seem that the definition of the actions of the corrupted players is circular and thus are not well defined. For ease of presentation we denote with T, H the sets of the execution σ and with T', H' their respective sets in the execution σ'. The circularity of the definition may (falsely) appear in the following example; the actions of T depend on the actions of T' which may in turn depend on the messages they receive from H' which depend on the actions of H in σ which may lastly depend on the actions of T in the same execution. To overcome this obstacle we observe that the actions of all players are uniquely defined in an inductive manner, i.e., in the first round of both executions the actions of honest players in the sets H, T' are uniquely defined by the deterministic protocol $\mathcal{A}$ and their initial values due to the fact that no messages have been received. Therefore, the actions of the first round that the respective corruption sets H', T take are uniquely defined by the actions of H, T'. Assuming that the actions (exchanged messages) of all players are uniquely defined until the end of round k, one can observe that the actions of all players are uniquely defined in round $k+1$ due to the fact that the exchanged messages of round $k+1$ are completely determined by actions taken until round k.

We can show that if we drop the requirement for t-local safety, then Theorem 3 does not hold. Intuitively, the reason is that an ad hoc protocol that assumes certain topological properties for the network may be t-locally resilient in a family of graphs that have the assumed topological properties. Indeed, Pelc and Peleg [15] introduced another algorithm for the uniform model, the *Relaxed Propagation Algorithm* (RPA) which uses knowledge of the topology of the network and they proved that there exists a graph G'' with dealer D for which RPA is 1-locally resilient and CPA is not. So if we use RPA in an ad hoc setting assuming that the network is G'' then this algorithm will be t-locally resilient for (G'', D) while CPA will not. Non-t-local safety of RPA follows from the fact that the decisions depend on the assumed topology and therefore they could be incorrect if the topological assumptions do not hold. More specifically, a player, running RPA, could decide on a message which she receives from $2t+1$ disjoint paths and for which she can verify, from the assumed topology, that at most t may contain corrupted nodes. However, if the topology is actually not as assumed, then it could even be the case that all $2t+1$ paths contain corrupted nodes and thus the decision value is incorrect. The fact that the non-safe algorithm RPA is resilient in instances where CPA is not, shows that there exist non-safe algorithms of higher resilience than CPA.

Corollary 1 *(CPA Uniqueness) Given a graph G and dealer D, if there exists an ad hoc Broadcast algorithm which is t-locally resilient in (G, D) and t-locally safe, then CPA is t-locally resilient in (G, D).*

Proof Immediate from Theorems 2,3. □

This, according to the definition of uniqueness means that, CPA is unique among t-locally safe algorithms.

3.3 Hardness of $pLPC$

Ichimura and Shigeno in [8] prove that the *set splitting* problem, known as NP-hard [6], can be reduced to the problem of computing the minimum integer t such that a t-local pair cut exists in a graph G. By generalizing the notion of the t-local pair cut to that of t-plp cut and defining the $pLPC$ problem analogously one can use a nearly identical proof to that of [8] and show that the $pLPC$ problem is NP-hard.

Theorem 4 *$pLPC$ is* NP*-hard.*

Proof We first consider a different (general) version of the pLPC problem which asks if there is a t-plp cut in the graph where no dealer is specified, i.e., if there exists a t-plp cut for any possible dealer-node in the node set. Concluding the proof we will show that if the general pLPC problem is NP-hard then so is our original pLPC problem (with specified dealer).

We first show that the *set splitting* problem known as NP-hard [6] can be reduced to the general pLPC problem. Given a collection S of 3-element subsets of a finite set X, the set splitting problem asks whether there is a partition of X into two subsets X_1 and X_2 such that no subset in S is entirely contained in either X_1 or X_2. An instance of this problem is shown in Fig. 2.

We propose the following reduction. Let $S+$ be a multiple collection adding dummy subsets $\{v\}$ to S such that the cardinality of $\{s \in S+ : v \in s\}$ is at least six for each $v \in X$. A complete graph with node set $S+$ and a copy of it are denoted by K_{S+} and K'_{S+}, respectively. We denote with $s' \in V(K'_{S+})$ the copy of node $s \in S+$. We construct a graph G_{SSP} (Fig. 3) with vertex set $V(G_{SSP}) = V(K_{S+}) \cup V(K'_{S+}) \cup X$ and edge set

$$E(G_{SSP}) = E(K_{S+}) \cup E(K'_{S+}) \cup \{(v, s), (v, s') : v \in X, s \in S+, v \in s\}$$

where s' is a copy of s as mentioned above.

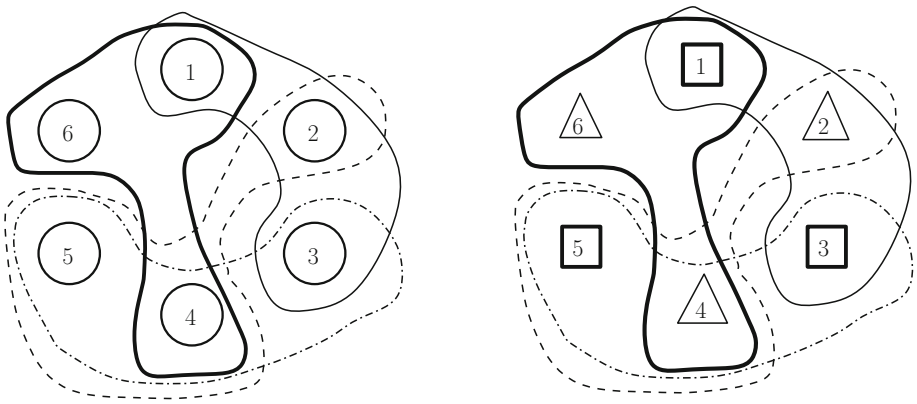

Fig. 2 An instance and the solution of a set splitting problem with $X = \{1, 2, 3, 4, 5, 6\}$ and $A = \{\{1, 2, 3\}, \{3, 4, 5\}, \{1, 4, 6\}, \{2, 4, 5\}\}$. The solution is depicted by the the two sets $X_1 = \{1, 3, 5\}$ and $X_2 = \{2, 4, 6\}$ the elements of which are marked with *squares* and *triangles* respectively. Notice that all sets in A have at least one node of both shapes

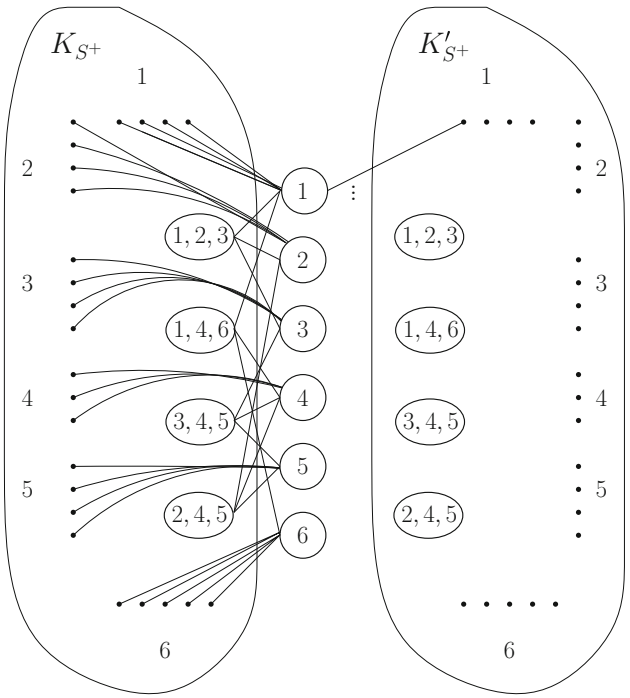

Fig. 3 The graph G_{SSP} for the set splitting problem in Fig. 2. Edges on the *right side* are formed symmetrically to those on the *left side* and are omitted for simplicity

We next prove that there is a set splitting of X if and only if there is a 2-plp cut C in G_{SSP}.

For the "only-if" direction it suffices to observe that a partition $X = X_1 \cup X_2$ for which no subset in S is entirely contained in either X_1 or X_2, implies that each of the sets X_1, X_2 will contain at most 2 nodes (elements) that appear in the neighborhood of every node (set) in K_{S+} and K'_{S+} and thus $X = X_1 \cup X_2$ is a 2-plp cut.

For the "if" direction we argue as follows. Considering a 2-plp cut C on G_{SSP} we distinguish between two cases, the case $X \setminus C \neq \emptyset$ and the case $X \setminus C = \emptyset$. In the first case we observe that if a subgraph of G_{SSP} obtained by removing C from G_{SSP} consists of at least two connected components, then C must contain $\mathcal{N}(v) \cap V(K_{S+})$ or $\mathcal{N}(v) \cap V(K'_{S+})$ for each $v \in X \setminus C$. Since each $v \in X$ has at least six neighbors in both $V(K_{S+})$ and $V(K'_{S+})$, for any possible partition of C, either each node in $V(K_{S+}) \setminus C$ or each node in $V(K'_{S+}) \setminus C$ has at least 3 neighbors in some set of the partition. Therefore, since C is a 2-plp cut the case $X \setminus C \neq \emptyset$ cannot hold.

It remains to consider the case of a 2-plp cut $C = C_1 \cup C_2$ where $X \setminus C = \emptyset$, which implies that $X \subseteq C$; note that C_1, C_2 are in fact 2-local due to symmetry. Observe that X also constitutes a cut in G_{SSP}; moreover, in this case both sets $X_i = C_i \cap X, i = 1, 2$, are 2-local (being subsets of the 2-local sets $C_i, i = 1, 2$), hence $X = X_1 \cup X_2$ is a 2-plp cut. Therefore, no set in $s \in S$ can be entirely contained in some $X_i, i = 1, 2$, because $|s| = 3$, hence the corresponding vertex s in K_{S+} (and s' in K'_{S+}) would have three neighbors in X_i contradicting the fact that X_i is a 2-local set. Thus the set splitting instance (S, X) has a solution $X = X_1 \cup X_2$.

We conclude the proof by showing that NP-hardness for $pLPC(G, t)$ without a dealer (general case) implies NP-hardness for the case with a dealer D, i.e., problem $pLPC(G, t, D)$. Indeed, if $pLPC(G, t, D)$ could be solved with a polynomial-time algorithm then solving $pLPC(G, t, v)$ for every node $v \in V$ would suffice to build a polynomial algorithm for $pLPC(G, t)$. Therefore to compute $pLPC(G, t, D)$ is NP-hard. □

We have thus established that computing the necessary and sufficient condition for CPA to work is NP-hard. Observe that this negative result also has a positive aspect, namely that a polynomially bounded adversary is unable to always compute an optimal attack unless P = NP.

4 Known topology networks

4.1 The path Propagation Algorithm

Considering only safe Broadcast algorithms, the uniqueness of CPA in the ad hoc model implies that an algorithm that achieves Broadcast in cases where CPA does not, must

operate under a weaker model e.g., assuming additional information on the topology of the network. It thus makes sense to consider the setting where players have full knowledge of the topology of the network. In this section we propose the *Path Propagation Algorithm* (PPA) and show that is of optimal resilience in the full-knowledge model.

For convenience we will use the following notions: a set $S \subseteq V \setminus D$ is called a *cover* of a set of paths $\mathcal{P}$ if and only if $\forall p \in \mathcal{P}$, $\exists s \in S$ s.t. $s \in p$ (s is a node of p). As one can see in the algorithm, each path which is propagated, is transmitted along with a value which it carries. This value corresponds to the value initially sent by the first node of the path (*source* of the path). The other endpoint of the path, i.e., the last node of path p will be denoted with $tail(p)$. When a node v acts as a relay of a value which has reached to it through path p, it appends its id v to the last node of p and thus it creates a new path p' with $tail(p') = v$, whereas the source of p and p' remains the same. The description of PPA follows.

Protocol 2: ***Path Propagation Algorithm (PPA)***

Input (for each node v): dealer's label D, graph G, $t(v) =$ max #corruptions in $\mathcal{N}(v)$.
Message format: pair (x, p), where $x \in X$ (message space), and p is a path of G (message's propagation trail).

Code for D: send message (x_D, D) to all neighbors, decide on x_D and terminate.

Code for $v \neq D$: Initialize $decision_v := \perp$.
upon reception of (x, p) from node u `do:`

`if` $(v \in p) \vee (tail(p) \neq u)$ `then discard` the message `else send` $(x, p||v)$[2] to all neighbours.

`if` $(decision_v = \perp) \wedge (\texttt{decision}(v) \neq \perp)$ `then`
$decision_v := \texttt{decision}(v)$;
send message $(decision_v, v)$ to all neighbors;
decide on $decision_v$.

function `decision`(v)

(* *dealer propagation rule* *)

`if` $v \in \mathcal{N}(D)$ and v receives (x_D, D) `then return` x_D.

(* *honest path propagation rule* *)

`if` v receives messages $(x, p_1), \ldots, (x, p_m)$ and $\exists \mathcal{P} \subseteq \{p_1, \ldots, p_m\}$ that does not have a t-local cover.

`then return` x `else return` $\perp$.

[2] By $p||v$ we denote the path consisting of path p and node v, with the last node of p connected to v.

Concerning the honest path propagation rule of PPA, note that $\mathcal{P}$ is not the whole set of collected paths received by v but rather any subset of the paths through which v receives a value x. Also observe that the criterion is existential, and thus the existence of one subset $\mathcal{P}$ with the desired property suffices for the player to decide on the corresponding value. Observe that each player can check the validity of the honest path propagation rule only if it has knowledge of the corruption function t and the network's topology. Next, we argue about the safeness of PPA.

Theorem 5 *PPA is t-locally safe.*

Proof We will show that if a player decides on a value x through PPA then $x = x_D$. Assume on the contrary that there is a set of players $V' \subseteq V$ that decide on values different than x_D. Let v be the player of V' that decides in the earliest round among all players in V', i.e., the first player to make an incorrect decision, and assume that v decides on $x \neq x_D$. Player v cannot be a neighbor of the dealer since all neighbors of the dealer only decide on x_D as can be seen in the respective decision rule of PPA. Therefore v has decided on x through the honest path propagation rule. This means that v received value x from a set of paths $\mathcal{P}$ such that there does not exist a t-local cover of $\mathcal{P}$. Moreover, through the check $tail(p) \neq u$, we ensure that at least one corrupted node will be included in a path which contains faulty nodes. Due to the latter, we avoid the case where all the corrupted nodes hide their identity in a path by changing the actual propagation trail; this is a commonly used idea which was first presented in [1].

Since there does not exist a t-local cover for $\mathcal{P}$, it is now obvious that at least one path p of $\mathcal{P}$ is entirely corruption free and if p is entirely corruption free, then value x, which is relayed through p, is the actual value that the source-node w of p has decided on. Thus, at least one honest player has decided in $x \neq x_D$ before v. A contradiction to the fact that v is the first player to make an incorrect decision. □

4.2 A necessary and sufficient condition

We will now show that the non-existence of a t-local pair cut is a sufficient condition for PPA to achieve Broadcast in the t-locally bounded model in networks of known topology.

Theorem 6 (Sufficiency) *Given a graph G with dealer D and corruption function t, if no t-local pair cut exists in (G, D) then all honest players will decide through PPA on x_D.*

Proof All players in $\mathcal{N}(D)$ decide on x_D due to the *dealer propagation rule*, since the dealer is honest. We next show the rest of the players will decide on x_D due to the *honest path propagation rule*. Observe that since PPA is t-locally safe, it suffices to show that, at some step, every player will receive

the correct value x_D through a set of paths $\mathcal{P}$ which will allow her to decide on x_D through the honest path propagation rule (if she has not aldready decided on it in a previous step).

Let v be any player in $V \setminus \mathcal{N}(D)$ and assume that no t-local pair cut exist in (G, D). Let T be a t-local set and consider an execution σ_T of PPA where T is the corruption set. Let $\mathcal{P}_{D,v}$ be the set of all paths connecting D with v that are composed entirely by nodes in $V \setminus T$ (honest nodes). Observe that $\mathcal{P}_{D,v} \neq \emptyset$, otherwise T is a cut separating D from v and T is trivially a t-local pair cut, a contradiction. Since paths in $\mathcal{P}_{D,v}$ are entirely composed by honest nodes it is easy to see that v will receive the correct value x_D through all paths in $\mathcal{P}_{D,v}$ i.e. the path set $\mathcal{P}_{D,v}$ is a subset of all the paths that propagate the same value x_D to v.

We next prove that there does not exist a t-local cover of $\mathcal{P}_{D,v}$. Assume that $\exists T'$: t-local cover of $\mathcal{P}_{D,v}$. Then obviously $T \cup T'$ is a cut separating D from v, since every path that connects D with v contains at least a node in $T \cup T'$. Moreover the cut $T \cup T'$ can be partitioned in the sets $T \setminus T', T'$ which are trivially t-local and thus, $T \cup T'$ is a t-local pair cut, a contradiction. Hence, there does not exist a t-local cover of $\mathcal{P}_{D,v}$. This means that there exists a path set which propagates the correct value x_D to v and does not have a t-local cover, namely $\mathcal{P}_{D,v}$ and thus, the honest path propagation rule is verified, for instance by taking $\mathcal{P} = \mathcal{P}_{D,v}$.

Consequently, in execution σ_T, node v will receive the correct value, in some step k, through every path in $\mathcal{P}_{D,v}$ along with the corresponding propagation trail. If player v has already decided before step k then her decision will certainly be on x_D due to the t-local safety of PPA. In any other case, v will also decide on the correct value x_D by the end of step k due to the honest path propagation rule, because $\mathcal{P}_{D,v}$ is not covered by any t-local set. □

Using the same arguments as in the proof of the necessity of condition $t < LPC(G, D)$ [15] it can be seen that the non-existence of a t-local pair cut is a necessary condition for any algorithm to achieve Broadcast under the non-uniform model. The proof uses similar arguments with that of Theorem 3 but is much simpler; the different executions are considered in the same graph. One cannot consider executions in two different graphs since the topology is known to all the players and the players would be able to distinguish the two scenaria. For completeness the proof is presented below.

Theorem 7 (Necessity) *Given a graph G with dealer D and corruption function t, if there exists a t-local pair cut in (G, D) then there is no t-locally resilient algorithm for (G, D).*

Proof Assume that there exists a t-local pair cut $C = C_1 \cup C_2$ in (G, D) partitioning $V \setminus C$ into sets $A, B \neq \emptyset$ such that $D \in A$. Also let $\mathcal{A}$ be a t-locally resilient algorithm for (G, D). We will show that $\mathcal{A}$ does not allow any $v \in B$ to correctly decide on the value of the dealer x_D in all possible executions, which contradicts its t-local resilience. Consider the following two executions σ and σ' of $\mathcal{A}$ on the instance (G, D).

- In execution σ the dealer's value is $x_D = 0$ and the corrupted players are precisely those in C_1. In each round $t \geq 1$ of the execution σ, every player in C_1 performs the action that she is instructed to perform in round t of execution σ' (where she is honest).
- In execution σ' the dealer's value is $x_D = 1$ and the corrupted players are precisely those in C_2. In each round $t \geq 1$ of the execution σ', every player in C_2 performs the action that she is instructed to perform in round t of execution σ (where she is honest).

The same standard argument of the two simultaneous executions is used here. Its correctness regarding the unambiguous definition of the players' actions is proved in Sect. 3.2 in the paragraph "*Note on the proof of Theorem 3*".

Similarly with the proof of Theorem 3, it follows that any player $v \in B$ performs identical actions in executions σ and σ' of $\mathcal{A}$. Hence v decides on the same value in σ and σ', which cannot be correct in both executions, since D has a different initial value in each of them. □

Thus the non-existence of a t-local pair cut proves to be a necessary and sufficient condition for the existence of a t-locally resilient algorithm in both the uniform and the non-uniform model. Therefore PPA is of optimal resilience.

4.3 On the hardness of broadcast in known networks

In order to run PPA we have to be able to deduce whether a corruption-free path exists among a set of paths broadcasting the same value. Formally, given a graph $G(V, E)$, a set of paths $\mathcal{P}$, a node u (the one that executes `decision`(u)) and a corruption function t we need to determine whether there exists a t-local cover T of $\mathcal{P}$. We call this problem the Local Path Cover Problem, $LPCP(G, D, u, t, \mathcal{P})$ and show that is NP-hard.

Theorem 8 *It is* NP*-hard to compute $LPCP(G, D, u, t, \mathcal{P})$.*

Proof We will describe a reduction from $3SAT$ to $LPCP(G, D, u, t, \mathcal{P})$. For every variable x_i we construct a gadget G_{x_i} shown at the left of Fig. 4. We will make use of a parameter μ that will serve as a corruption function of constant value (that is, our hardness result holds even for the uniform model). We will use several copies of the complete graphs $K_{\mu+1}$ and $K_{2\mu}$. Node D is connected to every vertex of a $K_{\mu+1}$ copy. Every vertex of that $K_{\mu+1}$ copy is connected to all the 'upper' μ vertices of a $K_{2\mu}$ copy; let us call this

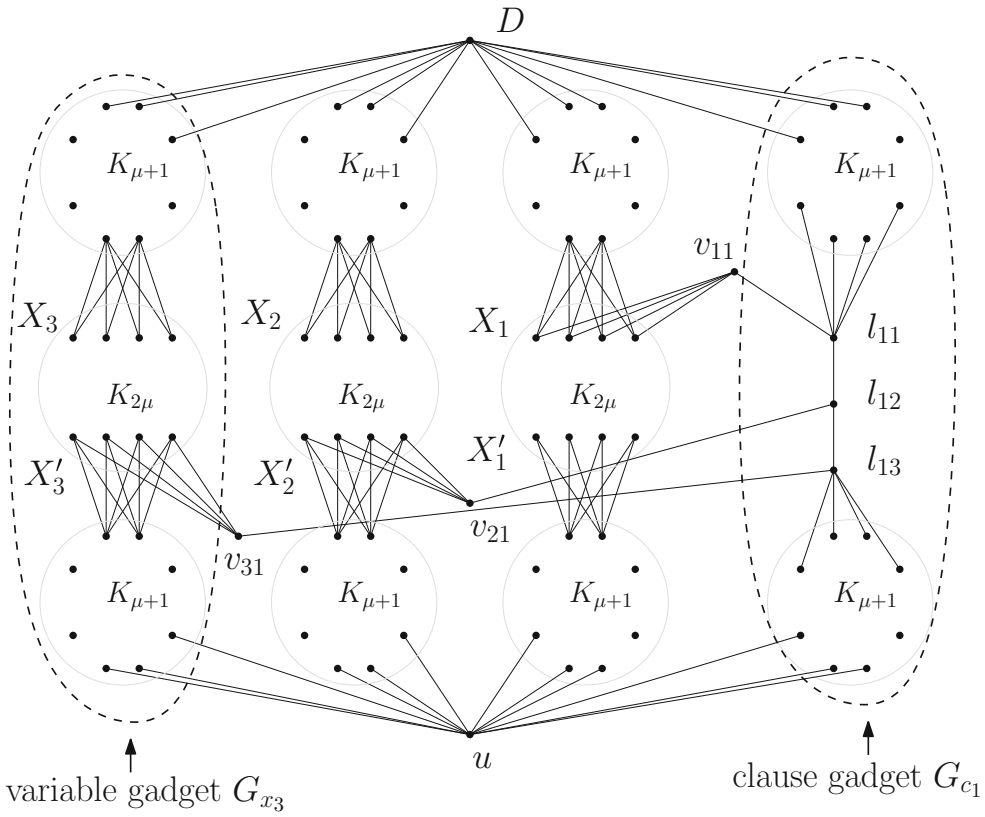

Fig. 4 An instance of the reduction graph G for variables $\{x_1, x_2, x_3\}$ and clause $c_1 = \{\neg x_1 \vee x_2 \vee x_3\}$. Literals $\neg x_1, x_2, x_3$ of clause c_1 are represented by nodes l_{11}, l_{12}, l_{13} respectively

'upper' node set X_i. Symmetrically for the lower part, node u is connected to every vertex of a 'bottom' $K_{\mu+1}$ copy and every vertex of that $K_{\mu+1}$ copy is connected to all the 'lower' μ vertices of $K_{2\mu}$; let us denote by X'_i this 'lower' part of $K_{2\mu}$. Now assuming that $\mathcal{P}$ contains those paths in G_{x_i} that are of length 5 and connect D to u (and no other path in G_{x_i}) it is easy to show that :

Lemma 1 *If $LPCP(G, D, u, \mu, \mathcal{P}) = 1$ with μ-local cover T, then either $X_i \subseteq T$ or $X'_i \subseteq T$.*

$T \cap G_{x_i}$ is a cut of G_{x_i}. Since the only possible μ-local cuts in G_{x_i} are X_i and X'_i, the claim is immediate.

Now for every clause $c_i = l_{i_1} \vee l_{i_2} \vee l_{i_3}$ in C we construct the gadget shown on the right of Fig. 4. Node D is connected to every vertex of $K_{\mu+1}$. Every vertex of $K_{\mu+1}$ is connected to the first literal of the clause, say l_{i_1}. Literal l_{i_1} is connected to l_{i_2}, and l_{i_2} to l_{i_3}. Symmetrically, node u is connected to every vertex of another copy of $K_{\mu+1}$, and every vertex of $K_{\mu+1}$ is connected to l_{i_3}. Let us call this subgraph of G, G_{c_i}. Assuming that all paths from D to u of length 6 that go through G_{c_i} are contained in $\mathcal{P}$ we show that:

Lemma 2 *if $LPCP(G, D, u, \mu, \mathcal{P}) = 1$ with μ-local cover T, then $l_{i_1} \in T$ or $l_{i_2} \in T$ or $l_{i_3} \in T$.*

The proof is by contradiction: if no l_{i_j} node belongs to T, then it must be $K_{\mu+1} \subseteq T$, contradicting the t-locality of T.

The last thing we need to establish is that if $X_i \subseteq T$ (respectively $X'_i \subseteq T$), no $\neg x_i$ (resp. x_i) literal of G_{c_j} is in T. We achieve this by adding a node v_{ij} connecting X_i (resp. X'_i) to $\neg x_i$ (resp. x_i) for each appearance of these literals in some G_{c_j}. The following holds because If both X_i and $\neg x_i$ are in T, then T is not μ-local since $|\mathcal{N}(v_{ij}) \cap T| = \mu + 1$.

Lemma 3 *If $LPCP(G, D, u, \mu, \mathcal{P}) = 1$ with μ-local cover T, then $X_i \subseteq T$ (resp. $X'_i \subseteq T$) $\Rightarrow \neg x_i \notin T$ (resp. $x_i \notin T$).*

So for graph G that is constructed as described above and for path set $\mathcal{P}$ consisting of the paths used for proving Lemmata 1 and 2 we have that $LPCP(G, D, u, \mu, \mathcal{P}) = 1$ iff there exists a truth assignment A which makes every clause in C true. The '$\Rightarrow$' direction follows from the lemmata proved above. The truth assignment A is constructed as follows: if $X_i \subseteq T$ (resp. $X'_i \subseteq T$) then $\neg x_i$ (resp. x_i) is true in A. The '$\Leftarrow$' direction comes naturally by setting T contain X_i if x_i is true by A, otherwise T contains X'_i; T also contains all literals in G_{c_j} that are set true by A. Then T is a μ-local cover of $\mathcal{P}$ and $LPCP(G, D, u, \mu, \mathcal{P}) = 1$. □

The above theorem implies that PPA may not be practical in some cases, since its decision rule cannot be always

checked efficiently. It remains to show whether any other algorithm which has the same resiliency as PPA can be efficient. The following theorem provides an indication that the answer is negative, by showing that algorithms which behave exactly as PPA w.r.t. decision are unlikely to be efficient.

Theorem 9 *Assuming* P $\neq$ NP, *no safe fully polynomial protocol* Π *can satisfy the following: for any graph G, dealer D, corruption function t, and admissible corruption set C executing protocol* Π_C*, a node u decides through PPA on a value x iff u will decide on x by running* Π *on* (G, D, t, C, Π_C).

Proof We will show that if such Π existed then it would be a polynomial time solver for the 3-SAT problem. Let us consider what happens when Π is run on the graph G that we used in the proof of Theorem 8, with dealer D and the corrupted nodes being the ones that connect the "clause" gadgets with the "variable" gadgets (e.g. $C = \{v_1, v_2, v_3\}$ in Fig. 4). The adversary protocol Π_C is: the corrupted nodes don't send or relay any messages.

The 3-SAT instance used to make G has a solution iff $LPCP(G, D, u, t, \mathcal{P}) = 1$, i.e. a μ-local cover C_1 on $\mathcal{P}$ exists, where $\mathcal{P}$ is the set of paths we used in the proof of Theorem 8. It can be seen from the decision rule of PPA that, while running PPA on G, u will not decide on any value iff a μ-local cover C_1 on $\mathcal{P}$ exists. Moreover a node u does not decide through PPA on a value x iff u does not decide on x by running Π on (G, D, t, C, Π_C).

So u decides on x_D while running Π on G, with dealer D and corruption set C which runs the Π_C protocol iff 3-SAT does not have a solution. Apparently if Π existed then 3-SAT would have a polynomial time solver. □

5 Partial knowledge

Until now we have presented optimal resilience algorithms for Broadcast in two extreme cases, with respect to the knowledge over the network topology: the ad hoc model and the full-knowledge model. A natural question arises: is there any algorithm that works well in settings where nodes have partial knowledge of the topology?

To address this question we introduce the *partial knowledge model*, where each player has restricted knowledge over the topology of the network and devise a new, generalized version of PPA that can run with partial knowledge of the topology of the network. More specifically we assume that each player v only has knowledge of the topology of a certain connected subgraph G_v of G which includes v. Namely if we consider the family $\mathcal{G}$ of connected subgraphs of G we use the *topology view function* $\gamma : V \to \mathcal{G}$, where $\gamma(v)$ represents the subgraph over which player v has knowledge of the topology. We also define the *joint view* of a set S as the subgraph $\gamma(S)$ of G with node-set $V(\gamma(S)) = \bigcup_{u \in S} V(\gamma(u))$ and edge-set $E(\gamma(S)) = \bigcup_{u \in S} E(\gamma(u))$. We will call an algorithm which achieves Broadcast for any t-local corruption set in graph G with dealer D and view function γ, (γ, t)-*locally resilient* for (G, D).

Now given a corruption function t and a view function γ we define the Generalized Path Propagation Algorithm (GPPA) to work exactly as PPA apart from a natural modification of the path propagation rule.

Generalized path propagation rule: Player v receives the same value x from a set $\mathcal{P}$ of paths that are completely inside $\gamma(v)$ and is able to deduce (from the topology) that no t-local cover of $\mathcal{P}$ exists.

Remark Note that GPPA generalizes both CPA and PPA. Indeed, if $\forall v \in V$, $\gamma(v) = \mathcal{N}(v)$, then GPPA$(G, D, t, \gamma)$ coincides with CPA(G, D, t). If, on the other hand, $\forall v \in V$, $\gamma(v) = G$ then GPPA(G, D, t, γ) coincides with PPA(G, D, t).

We also notice that, quite naturally, as γ provides more information for the topology of the graph, resilience increases, with CPA being of minimal resilience in this family of algorithms, and PPA achieving maximal resilience.

To prove necessary and sufficient conditions for GPPA being t-locally resilient we need to generalize the notion of t-plp cut as follows:

Definition 9 (*type 1* (γ, t)*-partial local pair cut*) Let C be a cut of G, partitioning $V \setminus C$ into sets $A, B \neq \emptyset$ s.t. $D \in A$. C will be called a *type 1* (γ, t)*-partial local pair cut (plp1 cut)* if there exists a partition $C = C_1 \cup C_2$ s.t. C_1 is t-local and C_2 is t-local in the graph $\gamma(B)$.

Definition 10 (*type 2* (γ, t)*-partial local pair cut*) Let C be a cut of G, partitioning $V \setminus C$ into sets $A, B \neq \emptyset$ s.t. $D \in A$. C will be called a *type 2* (γ, t)*-partial local pair cut (plp2 cut)* if there exists a partition $C = C_1 \cup C_2$ s.t. C_1 is t-local and $\forall u \in B$, $C_2 \cap \mathcal{N}(u)$ is t-local in the graph $\gamma(u)$.

We can now show the following two theorems. The proofs build on the techniques presented for CPA and PPA.

Theorem 10 (sufficient condition) *Let t be corruption function and* γ *be a view function, if no* (γ, t)*-plp2 cut exists in G with dealer D then GPPA*(G, D, t, γ) *is* (γ, t)*-locally resilient for G, D.*

Proof Suppose no (γ, t)-plp2 cut exists. Assume an execution of GPPA where the actual corruption set is T. By definition, T is t-local, since we are in the t-locally bounded adversary model; clearly $T \cup \mathcal{N}(D)$ is a cut on G as defined before (i.e. not including node D). Since T is t-local and $T \cup \mathcal{N}(D)$ is not a (γ, t)-plp2 cut there must exist $u_1 \in V \setminus (T \cup \mathcal{N}(D) \cup D)$ s.t. $\mathcal{N}(D) \cap \mathcal{N}(u_1)$ is not t-local on $\gamma(u_1)$. But since all the honest nodes in $\mathcal{N}(D) \cap \mathcal{N}(u_1)$

have decided correctly as neighbors of the dealer, u_1 will receive the value x_D from paths of length 1, starting from these nodes. Finding a t-local corruption set covering these paths is impossible since it would have to include all these nodes, and from the above, it would not be t-local. So u_1 will decide on the dealer's value x_D. We can use the same argument inductively to show that every honest node will eventually decide on the correct value x_D through GPPA. Let $C_k = (\mathcal{N}(D) \setminus T) \cup \{u_1, u_2, \ldots, u_{k-1}\}$ be the set of the nodes that have decided until a certain round of the protocol and assume that they have decided correctly on x_D. Then $C_k \cup T$ is a cut. Since T is t-local by the same argument as before there exists an undecided node u_k s.t. $C_k \cap \mathcal{N}(u_k)$ is not t-local on $\gamma(u_k)$. Using the same argument as before u_k will decide on the correct value. Eventually all honest players will decide on x_D. Thus $GPPA$ is t-locally resilient in G. □

Again, as in the proof of Theorem 2, observe that in this proof we do not use the fact that GPPA is safe but rather prove inductively that in the case discussed all nodes will decide correctly.

Theorem 11 (necessary condition) *Let t be a corruption function, γ be a view function and $\mathcal{A}$ be a t-locally safe Broadcast algorithm. If a (γ, t)-plp1 cut exists in graph G with dealer D, then $\mathcal{A}$ is not (γ, t)-locally resilient for G, D.*

Proof Assume that there exists a (γ, t)-plp1 cut $C = T \cup H$ in graph G with dealer D and with T being the t-local set of the partition (Fig. 1). $\gamma(B)$ is the joint view of the nodes in B. G' is the graph that results from G if we remove edges from $A \setminus \gamma(B)$ s.t. the set H becomes t-local in G'. The existence of a set of edges that guarantees such a property is implied by the second property of the (γ, t)-plp1 cut. Suppose that there exists a t-locally safe Broadcast algorithm $\mathcal{A}$ which is t-locally resilient in graph G with dealer D. We can argue the same way we did on Theorem 3 which leads to a contradiction. □

One can argue that increased topology knowledge implies increased resilience for GPPA compared to CPA; for example, the sufficient condition of GPPA holds in settings where the sufficient condition of CPA does not hold. An overview of our results concerning the t-local model with respect to the level of topology knowledge appears in Fig. 5.

Notice that the reason for which GPPA is not optimal is that nodes in $\gamma(v)$ do not share their knowledge of topology. An optimal resilience protocol would probably include exchange of topological knowledge among players.

6 General adversary

Hirt and Maurer in [7] study the security of multiparty computation protocols with respect to an *adversary structure*, that is, a family of subsets of the players; the adversary is able to corrupt one of these subsets. More formally,

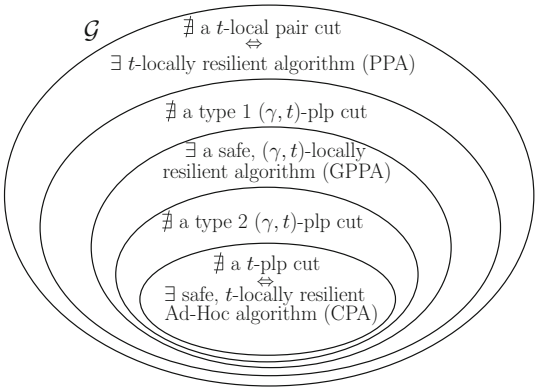

Fig. 5 Overview of conditions concerning the existence of t-locally resilient algorithms with respect to the level of topology knowledge. Note that $\mathcal{G}$ refers to the family of pairs (G, D)

A structure $\mathcal{Z}$ for the set of players V is a monotone family of subsets of V, i.e. $\mathcal{Z} \subseteq 2^V$, where all subsets of a set $Z \in \mathcal{Z}$ are in $\mathcal{Z}$ too, (alternatively, $\forall Z \in \mathcal{Z}$, if $Z' \subseteq Z$ then it holds that $Z' \in \mathcal{Z}$).

Let us now redefine some notions that we have introduced in this paper in order to extend our results to the case of a general adversary. We will call an algorithm that achieves Broadcast for any corruption set $T \in \mathcal{Z}$ in graph G with dealer D, *$\mathcal{Z}$-resilient*. A cover $S \in \mathcal{Z}$ of a set of paths $\mathcal{P}$ will be called a $\mathcal{Z}$-cover. We next generalize the notion of a t-local pair cut.

Definition 11 (*$\mathcal{Z}$-pair cut*) A cut C of G for which there exists a partition $C = C_1 \cup C_2$ and $C_1, C_2 \in \mathcal{Z}$ is called a *$\mathcal{Z}$-pair cut* of G.

6.1 Known topology networks

We adapt PPA in order to address the Broadcast problem under a general adversary. The Generalized $\mathcal{Z}$-PPA algorithm can be obtained by a modification of the path propagation rule of PPA (Protocol 2).

$\mathcal{Z}$-PPA Honest Path Propagation Rule player v receives the same value x from a set $\mathcal{P}$ of paths and is able to deduce that for any $T \in \mathcal{Z}$, T is not a cover of $\mathcal{P}$.

Moreover, the following theorems hold and their proofs are essentially the same as the proofs of Theorems 6, and 7. The only technical modification in the proofs is that one should replace the notions of t-local pair cut, t-local set, t-local cover, with that of $\mathcal{Z}$-pair cut, admissible corruption set (or set which belongs to $\mathcal{Z}$) and $\mathcal{Z}$-cover respectively.

Theorem 12 (Sufficiency) *Given a graph G, dealer D, and an adversary structure $\mathcal{Z}$, if no $\mathcal{Z}$-pair cut exists, then all honest players will decide on x_D through $\mathcal{Z}$-PPA.*

Theorem 13 (Necessity) *Given a graph G dealer D, and an adversary structure $\mathcal{Z}$, if there exists a $\mathcal{Z}$-pair cut then there exists no $\mathcal{Z}$-resilient Broadcast algorithm for (G, D).*

6.2 Ad hoc networks

Since in the ad hoc model the players know only their own labels, the labels of their neighbors and the label of the dealer it is reasonable to assume that a player has only local knowledge on the actual adversary structure $\mathcal{Z}$. Specifically, given the actual adversary structure $\mathcal{Z}$ we assume that each player v knows only the *local adversary structure* $\mathcal{Z}_v = \{A \cap \mathcal{N}(v) : A \in \mathcal{Z}\}$.

As in known topology networks, we can describe a generalized version $\mathcal{Z}$-CPA of CPA, which is an ad hoc Broadcast algorithm for the general adversary model. In particular, we modify step (3) of CPA (Protocol 1) in the following way.
$\mathcal{Z}$-CPA Certified Propagation Rule: if a node v is not a neighbor of the dealer, then upon receiving the same value x from all its neighbors in a set $\mathcal{N} \subseteq \mathcal{N}(v)$ s.t. $N \notin \mathcal{Z}_v$, it decides on value x.

In order to argue about the topological conditions which determine the effectiveness of $\mathcal{Z}$-CPA we generalize the notion of partial t-local pair cut.

Definition 12 (*$\mathcal{Z}$-partial pair cut*) Let C be a cut of G partitioning $V \setminus C$ into sets A, $B \neq \emptyset$ s.t. $D \in A$. C is a *$\mathcal{Z}$-partial pair cut* (*$\mathcal{Z}$-pp cut)* if there exists a partition $C = C_1 \cup C_2$ with $C_1 \in \mathcal{Z}$ and $\forall u \in B,\ \mathcal{N}(u) \cap C_2 \in \mathcal{Z}_u$.

Analogously to CPA Uniqueness, we can now prove $\mathcal{Z}$-CPA Uniqueness in the general adversary model.

Theorem 14 (Sufficient Condition) *Given a graph G, dealer D, and an adversary structure $\mathcal{Z}$, if no $\mathcal{Z}$-pp cut exists, then $\mathcal{Z}$-CPA is $\mathcal{Z}$-resilient.*

Proof Suppose that $\mathcal{Z}$-CPA is not $\mathcal{Z}$-resilient. Then there exists a scenario where C are the corrupted nodes, A are the honest and decided nodes, and B are the honest undecided nodes. All nodes in A have decided on the correct value because $\mathcal{Z}$-CPA is safe. Since every node in B is undecided we have that $\forall u \in B : \mathcal{N}(u) \cap A \in \mathcal{Z}_u$, otherwise u would have decided because a set of nodes that are not in Z_u would have sent him the same broadcast value. But then $C \cup A$ is a $\mathcal{Z}$-pp cut which is a contradiction. Hence, $\mathcal{Z}$-CPA is $\mathcal{Z}$-resilient. □

Theorem 15 (Necessary Condition) *Let $\mathcal{A}$ be a safe ad hoc Broadcast algorithm. Given a graph G, dealer D, and an adversary structure $\mathcal{Z}$, if a $\mathcal{Z}$-pp cut exists then $\mathcal{A}$ is not $\mathcal{Z}$-resilient for G, D.*

Proof Let $C = C_1 \cup C_2$ be the $\mathcal{Z}$-pp cut which partitions $V \setminus C$ in sets $A, B \neq \emptyset$ s.t. $D \in A$. Let $\mathcal{Z}' = \{\bigcup_{u \in B} Z \cap \mathcal{N}(u) : Z \in \mathcal{Z}\} \cup \{C_2\}$.

For every node u in B we have:

$$\begin{aligned}
\mathcal{Z}'_u &= \left\{Z \cap \mathcal{N}(u) : Z \in \mathcal{Z}'\right\} \cup \{C_2 \cap \mathcal{N}(u)\} \\
&= \left\{ (\bigcup_{v \in B} Z \cap \mathcal{N}(v)) \cap \mathcal{N}(u) : Z \in \mathcal{Z} \right\} \cup \{C_2 \cap \mathcal{N}(u)\} \\
&= \{Z \cap \mathcal{N}(u) : Z \in \mathcal{Z}\} \cup \{C_2 \cap \mathcal{N}(u)\} \\
&= \mathcal{Z}_u
\end{aligned}$$

since $\forall u \in B : \mathcal{N}(u) \cap C_2 \in \mathcal{Z}_u$.

So far we have established that (a) nodes in B cannot tell whether $\mathcal{Z}$ or $\mathcal{Z}'$ is the adversary structure since $\forall u \in B : \mathcal{Z}_u = \mathcal{Z}'_u$ and (b) C_2 is an admissible corruption set in $\mathcal{Z}'$.

Suppose a node in B could decide on some value in the scenario where $\mathcal{Z}$ is the adversary structure. Then using the standard argument employed in Theorem 3, an attack on the safeness of the algorithm would be possible in a different scenario where $\mathcal{Z}'$ is the adversary structure. The details of the proof are similar and are based on the difficulty of the honest players in B to distinguish which scenario they participate in, with respect to the adversary structure: the one with $\mathcal{Z}$ or the one with $\mathcal{Z}'$. □

Complexity of $\mathcal{Z}$-CPA

We will now make a simple but practical observation on the complexity of $\mathcal{Z}$-CPA. We measure the complexity of $\mathcal{Z}$-CPA with respect to the size of the graph $|G|$ only, because it is interesting to consider if CPA is *fully-polynomial* (of polynomial round, communication and local computations complexity) regardless of the size of the adversary structure description. We consider its complexity on the instances where Broadcast is solvable, i.e., there does not exist a $\mathcal{Z}$-pp cut.

Since $\mathcal{Z}$-CPA is trivially of polynomial round and communication complexity it holds that $\mathcal{Z}$-CPA is fully polynomial if its local computations complexity is polynomial. Observe that the local computations of every node essentially are comprised of membership checks dictated by the $\mathcal{Z}$-CPA Certified Propagation rule. Thus given any instance $(G, D, \mathcal{Z})$ of a family of instances $\mathcal{I}$, if there exists a polynomial algorithm $\mathcal{B}$ which given a set $S \subseteq \mathcal{N}(v)$ decides whether $S \in \mathcal{Z}_v$, for every player v, then $\mathcal{Z}$-CPA, with subroutine $\mathcal{B}$ for membership checks, is fully polynomial in $\mathcal{I}$. Practically, if the description of the adversary structure allows polynomial membership checks on the local adversary structures of all players then $\mathcal{Z}$-CPA is fully polynomial. Such an example is the t-locally bounded adversary model described is the first sections. In that model the description of the adversary structure is $\mathcal{Z} = \{S \in V : \forall v \in V, |S \cap \mathcal{N}(v)| \leq t\}$, which allows efficient local membership checks which

essentially constitute of comparing the cardinality of a set with t.

7 Dealer corruption

We have studied the problem of Broadcast in the case where the dealer is honest. In order to address the general case in which the dealer may also be corrupted one may observe that for a given adversary structure $\mathcal{Z}$ and graph G, $\mathcal{Z}$-resilient Broadcast in *ad hoc* networks can be achieved if the following conditions both hold:

(1) $\nexists Z_1, Z_2, Z_3 \in \mathcal{Z}$ *s.t.* $Z_1 \cup Z_2 \cup Z_3 = V$.
(2) $\forall v \in V$ there does not exist a $\mathcal{Z}$-pp cut for G with dealer v.

Condition 1 was proved by Hirt and Maurer [7] sufficient and necessary for the existence of secure multiparty protocols in complete networks. $\mathcal{Z}$-resilient Broadcast in the general case where the network is incomplete can be achieved by simulating any protocol for complete graphs (e.g. the protocol presented in [4]) as follows: each one-to-many transmission is replaced by an execution of $\mathcal{Z}$-CPA. It is not hard to see that the conjunction of the above two conditions is necessary and sufficient for Broadcast in incomplete networks in the case of corrupted dealer. Similarly in networks of known topology, there exists a $\mathcal{Z}$-resilient Broadcast algorithm if condition 1 holds and for every $v \in V$ a $\mathcal{Z}$-pair cut does not exist for graph G with dealer v. Naturally, the above observations hold also in the special case of a locally bounded adversary.

8 Conclusions and open questions

As already mentioned in Sect. 5, GPPA is not of optimal resilience regarding the partial knowledge model. A necessary and sufficient condition for achieving Broadcast together with a unique protocol for the partial knowledge model with a general adversary were very recently presented in [14]; however, the latter algorithm is not efficient. Devising an efficient unique algorithm, or showing that such an algorithm is unlikely to exist, would be of great interest. To this end, a different approach from that of [14] would be to consider discovering network topology under a Byzantine adversary, as studied in [3,13].

We have shown that necessary and sufficient criteria for Broadcast on known topology and ad-hoc networks are NP-hard to compute. So what is the best attack a polynomially bounded adversary could deploy? Similar issues may be raised from the point of view of system designers. Defining an appropriate meaningful optimization objective on the network resilience is essential in answering such questions.

We have provided an indication that no safe, fully polynomial algorithm can achieve optimal resilience in the known topology locally bounded setting. It remains to prove or disprove this conjecture.

Finally, regarding protocol termination, we note that in the ad hoc case all honest players terminate since termination follows decision. In the full knowledge case it is easy to adapt the protocol since the number of nodes $|V|$ in the network is known by each player and no path can have length more than $|V| - 1$. Hence, each player can terminate and stop the propagation of messages after $|V - 1|$ rounds. The same applies to the partial knowledge case provided that the size of the network $|V|$ is known. If this is not the case then it is not obvious how to adapt GPPA in order to ensure termination; we leave this as an open question.

Acknowledgements This research was supported in part by the European Union (European Social Fund ESF) and Greek national funds through the Operational Program Education and Lifelong Learning of the National Strategic Reference Framework (NSRF) Research Funding Program ALGONOW: THALIS-NTUA (MIS 379414). We would also like to acknowledge support from the National Technical University of Athens through a scholarship to Dimitris Sakavalas (Special Account for Research Funds, NTUA) and a travel grant to Giorgos Panagiotakos (Thomaidion award). A large part of this work was done while Giorgos Panagiotakos was an undergraduate student at the National Technical University of Athens. Finally, we are grateful to the anonymous referees for their detailed comments and suggestions which signicantly improved the presentation and clarity of the paper.

References

1. Dolev, D.: The byzantine generals strike again. J. Algorithms **3**(1), 14–30 (1982)
2. Dolev, D., Dwork, C., Waarts, O., Yung, M.: Perfectly secure message transmission. J. ACM **40**(1), 17–47 (1993)
3. Dolev, S., Liba, O., Schiller, E.M.: Self-stabilizing byzantine resilient topology discovery and message delivery. In: Higashino, T., Katayama, Y., Masuzawa, T., Potop-Butucaru, M., Yamashita, M. (eds.) Stabilization, Safety, and Security of Distributed Systems—15th International Symposium, SSS '13, Osaka, Japan, November 13–16, 2013. Proceedings of Lecture Notes in Computer Science, vol. 8255, pp. 351–353. Springer, 2013
4. Fitzi, M., Maurer, U.: Efficient byzantine agreement secure against general adversaries. In: Kutten, S. (ed.) Distributed Computing, 12th International Symposium, DISC '98, Andros, Greece, September 24–26, 1998, Proceedings of Lecture Notes in Computer Science, vol. 1499 pp. 134–148. Springer, 1998
5. Garay, J.A., Moses, Y.: Fully polynomial byzantine agreement for $n > 3t$ processors in $t+1$ rounds. SIAM J. Comput. **27**(1), 247–290 (1998)
6. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, New York (1979)
7. Martin H., Ueli, M.: Complete characterization of adversaries tolerable in secure multi-party computation (extended abstract). In: James E.B., Hagit A., (eds.), Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing, PODC '97, Santa Barbara, California, USA, August 21–24, 1997, pp. 25–34. ACM, 1997

8. Ichimura, A., Shigeno, M.: A new parameter for a broadcast algorithm with locally bounded byzantine faults. Inf. Process. Lett. **110**(12–13), 514–517 (2010)
9. Koo, C-Y.: Broadcast in radio networks tolerating byzantine adversarial behavior. In: Chaudhuri, S., Kutten S., (eds.), Proceedings of the 23rd Annual ACM Symposium on Principles of Distributed Computing, PODC '04, St. John's, Newfoundland, Canada, July 25–28, 2004, pp. 275–282. ACM, 2004
10. Ashwin Kumar, M.V.N., Goundan, P.R., Srinathan, K, Rangan, C.P.: On perfectly secure communication over arbitrary networks. In: Proceedings of the 21st Annual Symposium on Principles of Distributed Computing, PODC '02, Monterey, California, USA, July 21–24, 2002, pp. 193–202, New York, NY, USA, ACM (2002)
11. Lamport, L., Shostak, R.E., Pease, M.C.: The byzantine generals problem. ACM Transact. Progr. Lang. Syst. **4**(3), 382–401 (1982)
12. Litsas, C., Pagourtzis, A., Sakavalas, D.: A graph parameter that matches the resilience of the certified propagation algorithm. In: Cichon, J., Gebala, M., Klonowski, M. (eds.) Ad-hoc, Mobile, and Wireless Network—12th International Conference, ADHOC-NOW '13, Wrocław, Poland, July 8–10, 2013. Proceedings of Lecture Notes in Computer Science, vol. 7960, pp. 269–280. Springer, (2013)
13. Nesterenko, M., Tixeuil, S.: Discovering network topology in the presence of byzantine faults. IEEE Transact. Parallel Distrib. Syst. **20**(12), 1777–1789 (2009)
14. Pagourtzis, A., Panagiotakos, G., Sakavalas, D.: Reliable message transmission under partial knowledge. IACR Cryptol. ePrint Arch. **2015**, 243 (2015)
15. Pelc, A., Peleg, D.: Broadcasting with locally bounded byzantine faults. Inf. Process. Lett. **93**(3), 109–115 (2005)
16. Tseng, L., Vaidya, N., Bhandari, V.: Broadcast using certified propagation algorithm in presence of byzantine faults. Inf. Process. Lett. **115**(4), 512–514 (2015)
17. Tseng, L., Vaidya, N.H.: Iterative approximate byzantine consensus under a generalized fault model. In: Frey, D., Raynal, M., Sarkar, S., Shyamasundar, R.K., Sinha, P. (eds.) Distributed Computing and Networking, 14th International Conference, ICDCN '13, Mumbai, India, January 3–6, 2013. Proceedings of Lecture Notes in Computer Science, vol. 7730, pp. 72–86. Springer, (2013)