

Broadcast CONGEST Algorithms against Adversarial Edges

Yael Hitron

Weizmann Institute of Science, Rehovot, Israel

Merav Parter

Weizmann Institute of Science, Rehovot, Israel

Abstract

We consider the corner-stone broadcast task with an adaptive adversary that controls a fixed number of t edges in the input communication graph. In this model, the adversary sees the *entire* communication in the network and the random coins of the nodes, while maliciously manipulating the messages sent through a set of t edges (unknown to the nodes). Since the influential work of [Pease, Shostak and Lamport, JACM'80], broadcast algorithms against plentiful adversarial models have been studied in both theory and practice for over more than four decades. Despite this extensive research, there is no round efficient broadcast algorithm for *general* graphs in the CONGEST model of distributed computing. Even for a *single* adversarial edge (i.e., $t = 1$), the state-of-the-art round complexity is polynomial in the number of nodes.

We provide the first round-efficient broadcast algorithms against adaptive edge adversaries. Our two key results for n -node graphs of diameter D are as follows:

- For $t = 1$, there is a deterministic algorithm that solves the problem within $\tilde{O}(D^2)$ rounds, provided that the graph is 3 edge-connected. This round complexity beats the natural barrier of $O(D^3)$ rounds, the existential lower bound on the maximal length of 3 edge-disjoint paths between a given pair of nodes in G . This algorithm can be extended to a $\tilde{O}(tD^{O(t)})$ -round algorithm against t adversarial edges in $(2t + 1)$ edge-connected graphs.
- For expander graphs with edge connectivity of $\Omega(t^2 \log n)$, there is a considerably improved broadcast algorithm with $O(t \log^2 n)$ rounds against t adversarial edges. This algorithm exploits the connectivity and conductance properties of G -subgraphs obtained by employing the Karger's edge sampling technique.

Our algorithms mark a new connection between the areas of fault-tolerant network design and reliable distributed communication.

2012 ACM Subject Classification Networks → Network algorithms

Keywords and phrases CONGEST, Fault-Tolerant Network Design, Edge Connectivity

Digital Object Identifier 10.4230/LIPIcs.DISC.2021.23

Related Version *Full Version*: <https://arxiv.org/abs/2004.06436>

Funding *Merav Parter*: This project is funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 949083).

Acknowledgements We are very grateful to David Peleg and Eylon Yogev for many useful discussions.

1 Introduction

Guaranteeing the uninterrupted operation of communication networks is a significant objective in network algorithms. The area of resilient distributed computation has been receiving a growing attention over the last years as computer networks grow in size and become more vulnerable to byzantine failures. Since the introduction of this setting by Pease et al. [50] and Lamport et al. [41, 50] distributed broadcast algorithms against various adversarial models have been studied in theory and practice for over more than four decades. Resilient distributed algorithms have been provided for broadcast and consensus [19, 20, 24, 12, 57,



© Yael Hitron and Merav Parter;
licensed under Creative Commons License CC-BY 4.0
35th International Symposium on Distributed Computing (DISC 2021).
Editor: Seth Gilbert; Article No. 23; pp. 23:1–23:19



Leibniz International Proceedings in Informatics
LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

55, 8, 56, 7, 23, 27, 25, 40, 52, 37, 21, 43, 34, 17, 38], as well as for the related fundamental problems of gossiping [9, 6, 13], and agreement [20, 50, 11, 16, 27]. See [51] for a survey on this topic. A key limitation of many of these algorithms is that they assume that the communication graph is the complete graph.

Our paper is concerned with communication graphs of arbitrary topologies. In particular, it addresses the following basic question, which is still fairly open, especially in the CONGEST model of distributed computing [54]:

► **Question 1.** *What is the cost (in terms of the number of rounds) for providing resilience against adversarial edges in distributed networks with arbitrary topologies?*

An important milestone in this regard was made by Dolev [19] who showed that the $(2t+1)$ node-connectivity of the graph is a necessary condition for guaranteeing the correctness of the computation in the presence of t adversarial nodes. Since then, byzantine broadcast algorithms for general graph topologies have been addressed mostly under simplified settings [52], e.g., probabilistic faulty models [53], cryptographic assumptions [26, 2, 1, 44], or under bandwidth-*free* settings (e.g., allowing neighbors to exchange exponentially large messages) [19, 43, 38, 40, 21, 38, 15]. A more in-depth comparison to the previous work can be found in the full paper [32]. In this paper, we consider the following extension of the standard CONGEST model to the adversarial setting:

The Adversarial CONGEST Model: The network is abstracted as an n -node graph $G = (V, E)$, with a processor on each node. Each node has a unique identifier of $O(\log n)$ bits. Initially, the processors only know the identifiers of their incident edges^a, as well as a polynomial estimate on the number of nodes n .

There is a *computationally unbounded* adversary that controls a fixed set of at most t edges in the graph, denoted hereafter as *adversarial* edges. The nodes do not know the identity of the adversarial edges, but they know the bound t . The adversary knows the graph topology and the random coins of the nodes. In each round, it is allowed to send $O(\log n)$ bit messages on each of the adversarial edges F (possibly a distinct message on each edge direction). It is *adaptive* as it can determine its behavior in round r based on the messages exchanged throughout the entire network up to round r .

^a This is known as the standard KT1 model [5].

The definition naturally extends to adversarial *nodes* $F \subseteq V$ for which the adversary can send in each round, arbitrarily bad $O(\log n)$ -bit messages on each of the edges incident to F . The primary complexity measure of this model is the *round* complexity. In contrast to many prior works in the adversarial setting, in our model, the nodes are *not* assumed to know the graph's topology, and not even its *diameter*. To address Question 1, we provide a comprehensive study of the adversarial broadcast problem, defined as follows:

The adversarial broadcast task: Given is a $(2t+1)$ edge-connected graph $G = (V, E)$ and a set $F \subset E$ of $|F| \leq t$ edges controlled by the adversary. There is a designated source node $s \in V$ that holds a message m_0 . It is then required for all the nodes to output m_0 , while ignoring all other messages.

To this date, all existing broadcast algorithms in the adversarial CONGEST model require a polynomial number of rounds, even when handling a single adversarial edge! Recently, Chlebus, Kowalski, and Olkowski [15] extended the result of Garay and Moses [27] to general

$(2t + 1)$ node-connected graphs with minimum degree $3t$. Their algorithms, however, use *exponentially* large communication. Their message size can be improved to polynomial only when using authentication schemes (which we totally avoid in this paper). It is also noteworthy that the existing protocols for *node* failures might still require polynomially many rounds for general graphs, even for a single adversarial edge and for *small* diameter graphs.

A natural approach for broadcasting a message m_0 in the presence of t adversarial edges is to route the message along $(2t + 1)$ edge-disjoint paths between the source node s , and each target node v . This allows each node to deduce m_0 by taking the majority message. This approach has been applied in previous broadcast algorithms (e.g., [19]) under the assumption that the nodes know *the entire graph*, and therefore can compute these edge disjoint paths. A recent work of [33] demonstrated that there are D -diameter $(2t + 1)$ edge-connected graphs for which the maximal length of any collection of t edge-disjoint paths between a given pair of nodes might be as large as $(D/t)^{\Theta(t)}$. For $t = 3$, the length lower bound becomes $\Omega(D^3)$. Providing round efficient algorithms in the adversarial CONGEST model calls for a new approach.

Our approach in a nutshell. Our approach is based on combining the perspectives of fault-tolerant (FT) network design, and distributed graph algorithms. The combined power of these points of views allows us to characterize the round complexity of the adversarial broadcast task as a function of the graph diameter D , and the number of adversarial edges t . This is in contrast to prior algorithms that obtain a polynomial round complexity (in the number of nodes). On a high level, one of the main tools that we borrow from FT network design is the FT sampling technique [3, 60, 18, 30, 46, 49, 14], and its recent derandomization by [10, 36]. For a given graph G and a bound on the number of faults k , the FT sampling technique defines a small family $\mathcal{G} = \{G_i \subseteq G\}$ of G -subgraphs denoted as *covering family*, which is formally defined as follows:

► **Definition 2** ((L, k) covering family). *For a given graph G , a family of G -subgraphs $\mathcal{G} = \{G_1, \dots, G_\ell\}$ is an (L, k) covering family if for every $\langle u, v, F \rangle \in V \times V \times E^{\leq k}$ and any L -length u - v path $P \subseteq G \setminus F$, there exists a subgraph G_i such that (P1) $P \subseteq G_i$ and (P2) $F \cap G_i = \emptyset$.*

As the graph topology is unknown, one cannot hope to compute a family of subgraphs that are completely known to the nodes. Instead, we require the nodes to *locally* know the covering family in the following manner.

► **Definition 3** (Local Knowledge of a Subgraph Family). *A family of ordered subgraphs $\mathcal{G} = \{G_1, \dots, G_\ell\}$ where each $G_i \subseteq G$, is locally known if given the identifier of an edge $e = (u, v)$ and an index i , the nodes u, v can locally determine if $e \in G_i$.*

In the context of $(2t + 1)$ edge-connected graphs with t adversarial edges, we set $L = O(tD)$ and $k = O(t)$. The randomized FT sampling technique [60, 18] provides an (L, k) covering family \mathcal{G} of cardinality $O(L^k \log n)$. [36] provided a deterministic construction with $O((L \text{ poly}(\log n))^{k+1})$ subgraphs.

One can show that by the properties of the covering family, exchanging the message m_0 over all subgraphs in \mathcal{G} (in the adversarial CONGEST model) guarantees that all nodes successfully receive m_0 . This holds since for every $v \in V$ and a fixed set of adversarial edges F , the family \mathcal{G} contains a subgraph G_i which contains a short s - v path (of length L) and does not contain any of the adversarial edges. Our challenge is two folds:

1. provide a round-efficient algorithm for exchanging m_0 over all \mathcal{G} -subgraphs simultaneously,
2. guarantee that each node outputs the message m_0 while ignoring the remaining messages.

To address the first challenge, we show that the family of subgraphs obtained by this technique has an additional key property of *bounded width*. Informally, a family \mathcal{G} of G -subgraphs has a bounded width if each G -edge appears in all but a bounded number of subgraphs in \mathcal{G} . The bounded width of \mathcal{G} allows us to exchange messages in all these subgraphs simultaneously, in a nearly optimal number of rounds. The round complexity of this scheme is based on a very careful analysis which constitutes the key technical contribution in this paper. To the best of our knowledge, the bounded width property of the FT sampling technique has been used before only in the context of data structures [60, 31]. We find the fact that it finds applications in the context of reliable distributed communication to be quite exceptional. The second challenge is addressed by performing a second phase which filters out the corrupted messages. The round complexities of our broadcast algorithms for general graphs are dominated by the cardinality of covering families (which are nearly tight by [36]).

We also consider the family of expander graphs, which received a lot of attention in the context of distributed resilient computation [22, 58, 39, 4]. For these graphs, we are able to show covering families¹ of considerably smaller cardinality that scales linearly with the number of the adversarial edges. This covering family is obtained by using Karger's edge sampling technique, and its conductance-based analysis by Wulff-Nilsen [61]. We hope this result will also be useful in the context of FT network design. We next describe our key contribution in more detail.

1.1 Our Results

We adopt the gradual approach of fault tolerant graph algorithms, and start by studying broadcast algorithms against a single adversarial edge. Perhaps surprisingly, already this case has been fairly open. We show:

► **Theorem 4** (Broadcast against a Single Adversarial Edge). *Given a D -diameter, 3 edge-connected graph G , there exists a deterministic algorithm for broadcast against a single adversarial edge that runs in $\tilde{O}(D^2)$ adversarial-CONGEST rounds. In addition, at the end of the algorithm, all nodes obtain a linear estimate for the diameter of the graph.*

This improves considerably upon the (implicit) state-of-the-art $n^{O(D)}$ bound obtained by previous algorithms (e.g., by [43, 15]). In addition, in contrast to many previous works (including [43, 15]), our algorithm does not assume global knowledge of the graph or any estimate on the graph's diameter. In fact, at the end of the broadcast algorithm, the nodes also obtain a linear estimate of the graph diameter.

Using the covering family obtained by the standard FT-sampling technique, it is fairly painless to provide a broadcast algorithm with a round complexity of $\tilde{O}(D^3)$. Our main efforts are devoted to improving the complexity to $\tilde{O}(D^2)$ rounds. Note that the round complexity of D^3 appears to be a natural barrier for this problem for the following reason. There exists a 3 edge-connected D -diameter graph $G = (V, E)$ and a pair of nodes s, v such that in any collection of 3 edge-disjoint s - v paths P_1, P_2, P_3 , the length of the longest path is $\Omega(D^3)$ (By Corollary 40 of [33]). The improved bound of $\tilde{O}(D^2)$ rounds is obtained by exploiting another useful property of the covering families of [36]. One can show that, in

¹ Using a somewhat more relaxed definition of these families

our context, each G -edge appears on all but $O(\log n)$ many subgraphs in the covering family. This plays a critical role in showing that the simultaneous message exchange on all these subgraphs can be done in $\tilde{O}(D^2)$ rounds (i.e., linear in the number of subgraphs).

Multiple adversarial edges. We consider the generalization of our algorithms to support t adversarial edges. For $t = O(1)$, we provide broadcast algorithms with $\text{poly}(D)$ rounds.

► **Theorem 5 (Broadcast against t -Adversarial Edges).** *There exists a deterministic broadcast algorithm against t adversarial edges, for every D -diameter $(2t + 1)$ edge-connected graph, with round complexity of $(tD \log n)^{O(t)}$. Moreover, this algorithm can be implemented in $O(tD \log n)$ LOCAL rounds (which is nearly optimal).*

We note that we did not attempt to optimize for the constants in the exponent in our results for multiple adversarial edges. The round complexity of the algorithm is mainly dominated by the number of subgraphs in the covering family (extended to support t faults).

Improved broadcast algorithms for expander graphs. We then turn to consider the family of expander graphs, which has been shown to have various applications in the context of resilient distributed computation [22, 58, 39, 4]. Since the diameter of expander graphs is logarithmic, the algorithm of Theorem 5 yields a round complexity of $(t \log n)^{O(t)}$. In the full paper [32], we provide a considerably improved solution using a combination of tools. The improved broadcast algorithm is designed for $\Theta(t^2 \log n / \phi)$ edge-connected ϕ -expander graphs.

► **Theorem 6.** *Given an n -node ϕ -expander graph with edge connectivity $\Omega(t^2 \log n / \phi)$, there exists a randomized broadcast algorithm against t adversarial edges with round complexity of $O(t \cdot \log^2 n / \phi)$ rounds.*

To obtain this result, we use the edge sampling technique by Karger [35]. Karger showed that given a k edge-connected graph, when sampling each edge with a probability of $p = \Theta(1/t) = \Omega(\log n / \phi \cdot k)$, in the sampled subgraph all cuts are concentrated around their expectation. Wulff-Nilsen [61] showed that this sampled subgraph is an expander as well, w.h.p. We combine these properties to obtain a broadcast algorithm using $O(t \cdot \log^2 n)$ rounds. We believe that this technique might have other applications in the contexts of fault-tolerant network design and distributed secure computation (e.g., for the works of [48]).

Improved broadcast algorithms with a small shared seed. Finally, in the full paper [32], we show (nearly) optimal broadcast algorithms given that all nodes have a shared seed of $\tilde{O}(1)$ bits.

► **Theorem 7 (Nearly Optimal Broadcast with Shared Randomness).** *There exists a randomized broadcast algorithm against a single adversarial edge that runs in $\tilde{O}(D)$ rounds, provided that all nodes are given $\text{poly}(\log n)$ bits of shared randomness.*

This result is obtained by presenting a derandomization for the well-known fault-tolerant (FT) sampling technique [60]. The FT-sampling technique is quite common in the area of fault-tolerant network design [18], and attracted even more attention recently [46, 14, 10, 36]. While it is relatively easy to show that one can implement the sampling using $\tilde{O}(D)$ random bits, we show that $\tilde{O}(1)$ bits are sufficient. This is obtained by using the pseudorandom generator (PRG) of Gopalan [29] and its recent incarnation in distributed settings [47]. We

note that for a large number of faults t , the complexity is unlikely to improve from $D^{O(t)}$ to $\tilde{O}(D)$ even when assuming shared randomness, i.e., the complexity can be improved only by a factor of D . Using the framework of *pseudorandom generator* [45, 59], we provide an improved broadcast algorithm for $\Omega(t \cdot \log n / \phi)$ edge-connected expander graphs that can tolerate t adversarial edges.

► **Lemma 8.** *Given an n -node ϕ -expander graph with edge connectivity $\Omega(t \cdot \log n / \phi)$, there exists a randomized broadcast algorithm against t adversarial edges, with a round complexity of $O(t \log^2 n / \phi)$, provided that all nodes have a shared seed of $O(\log n)$ bits.*

We hope that this work will motivate the study of additional distributed graph algorithms in the presence of adversarial edges and nodes.

Road Map. The broadcast algorithm with a single adversarial edge and the proof of Theorem 4 are given in Section 2. In Section 3 we consider multiple adversarial edges. Other results appear in the full paper [32].

Preliminaries. For a subgraph $G' \subseteq G$ and nodes $u, v \in V(G')$, let $\pi(u, v, G')$ be the unique u - v shortest path in G' where shortest-path ties are decided in a consistent manner. For a path $P = [u_1, \dots, u_k]$ and an edge $e = (u_k, v)$, let $P \circ e$ denote the path obtained by concatenating e to P . Given a path $P = [u_1, \dots, u_k]$ denote the sub-path from u_i to u_ℓ by $P[u_i, u_\ell]$. The asymptotic term $\tilde{O}(\cdot)$ hides poly-logarithmic factors in the number of nodes n . Given a graph G , throughout we will use the following observation.

► **Observation 9.** *Consider an n -node D -diameter graph $G = (V, E)$ and let u, v be a pair of nodes that are connected in $G \setminus F$ for some $F \subseteq E$. It then holds that $\text{dist}_{G \setminus F}(u, v) \leq 2(|F| + 1) \cdot D + |F|$.*

Proof. Let T be a BFS tree in G rooted at some source s . The forest $T \setminus F$ contains at most $|F| + 1$ trees of diameter $2D$. Then, the u - v shortest path P in $G \setminus F$ can be transformed into a path P' containing at most $|F|$ edges of P as well as $|F| + 1$ tree subpaths of the forest $T \setminus F$. Therefore, $|P'| \leq 2(|F| + 1) \cdot D + |F|$ as desired. ◀

2 Broadcast Algorithms against an Adversarial Edge

In this section, we prove Theorem 4. We first assume, in Section 2.1 that the vertices have a linear estimate $c \cdot D$ on the diameter of the graph D , for some constant $c \geq 1$. A-priori, obtaining the diameter estimation seems to be just as hard as the broadcast task itself. In Section 2.2, we then show how this assumption can be removed. Throughout, we assume that the message m_0 consists of a single bit. In order to send a $O(\log n)$ bit message, the presented algorithm is repeated for each of these bits (increasing the round complexity by a $O(\log n)$ factor).

2.1 Broadcast with a Known Diameter

We first describe the adversarial broadcast algorithm assuming that the nodes have a linear estimate on the diameter D . In Section 2.2, we omit this assumption. The underlying objective of our broadcast algorithms is to exchange messages over reliable communication channels that avoid the adversarial edge e' . There are two types of challenges: making sure that all the nodes first *receive* the message m_0 , and making sure that each node correctly

distinguish between the true bit and the false one. Our algorithm `BroadcastKnownDiam` has two phases, a *flooding* phase and an *acceptance* phase, which at the high level, handles each of these challenges respectively.

The first phase propagates the messages over an ordered collection of G -subgraphs $\mathcal{G} = \{G_1, \dots, G_\ell\}$ where each $G_i \subseteq G$ has several desired properties. Specifically, \mathcal{G} is an (L, k) covering family (see Def. 2) for $L = O(D)$ and $k = 1$. An important parameter of \mathcal{G} which determines the complexity of the algorithm is denoted as the *width*.

► **Definition 10** (Width of Covering Family). *The width of a collection of subgraphs $\mathcal{G} = \{G_1, \dots, G_k\}$, denoted by $\omega(\mathcal{G})$, is the maximal number of subgraphs avoiding a fixed edge in G . That is,*

$$\omega(\mathcal{G}) = \max_{e \in G} |\{G_i \in \mathcal{G} \mid e \notin G_i\}|.$$

The broadcast algorithm starts by applying a 0-round procedure that provides each node in the graph with a local knowledge of an $(O(D), 1)$ covering family with bounded width. By [36], we have the following (see [32] for the proof):

► **Fact 11** ([36]). *Given a 3 edge-connected graph G , there exists a 0-round algorithm that allows all nodes to locally know an $(L, 1)$ -covering family $\mathcal{G} = \{G_1, \dots, G_\ell\}$ for $L = 7D$, such that $\ell = \tilde{O}(D^2)$. The width of \mathcal{G} is $\tilde{O}(D)$.*

In the following we present a broadcast algorithm whose time complexity depends on several parameters of the covering family. This will establish the case where all nodes know a linear bound on the diameter D .

► **Theorem 12.** *Given a 3 edge-connected graph G of diameter D . Assuming that the nodes locally know an $(L, 1)$ covering family \mathcal{G} for $L = 7D$, there exists a deterministic broadcast algorithm against an adversarial edge with $O(\omega(\mathcal{G}) \cdot L + |\mathcal{G}|)$ rounds.*

Broadcast with a known diameter (Proof of Theorem 12). Given a locally known $(L, 1)$ covering family $\mathcal{G} = \{G_1, \dots, G_\ell\}$ for $L = 7D$, the broadcast algorithm has two phases. The first phase has $O(L \cdot \omega(\mathcal{G}) + |\mathcal{G}|)$ rounds and the second phase has $O(L)$ rounds.

Phase 1: Flooding phase. The flooding phase consists of $\ell = |\mathcal{G}|$ sub-algorithms A_1, \dots, A_ℓ , where in each algorithm A_i , the nodes propagate messages on the underlying subgraph $G_i \in \mathcal{G}$ that is defined locally by the nodes. The algorithm runs the subalgorithms A_1, \dots, A_ℓ in a pipeline manner, where in the i 'th round of sub-algorithm A_i , the source node s sends the message (m_0, i) to all its neighbors. For every i , a node $u \in V$, upon receiving a message (m', i) from a neighbor w for the first time, stores the message (m', i) and sends it to all its neighbors if the following conditions hold: (i) $(w, u) \in G_i$ and (ii) u did not receive a message (m', i) in a prior round². For a node u and messages $(m_1, i_1), \dots, (m_k, i_k)$ waiting to be sent in some round τ , u sends the messages according to the order of the iterations i_1, \dots, i_k (note that potentially $i_j = i_{j+1}$, and specifically, there might be at most two messages with index i_j , namely, $(0, i_j)$ and $(1, i_j)$).

² If it receives several (m', i) messages in the same round from different neighbors, it will be considered as only one.

Phase 2: Acceptance phase. The second phase consists of $O(L)$ rounds, in which *accept* messages are sent from the source s to all nodes in the graph as follows. In the first round, the source node s sends an *accept*(m_0) message to all its neighbors. Then every other node $u \in V$ accepts the message m' as its final output, and sends an *accept*(m') message to all neighbors, provided that the following conditions hold: (i) there exists $i \in \{1, \dots, \ell\}$, such that u stored a message (m', i) in Phase 1; (ii) u received an *accept*(m') message in Phase 2 from a neighbor w_2 , such that $(u, w_2) \notin G_i$. Since \mathcal{G} is locally known, u can locally verify that $(u, w_2) \notin G_i$. This completes the description of the algorithm.

Correctness. We next prove the correctness of the algorithm. Missing proofs are deferred to the full paper [32]. We begin with showing that no node accepts a wrong message.

▷ **Claim 13.** No node $u \in V$ accepts a false message $m' \neq m_0$.

Proof. Assume towards contradiction there exists at least one node which accepts a message $m' \neq m_0$ during the second phase. Let u be the *first* node that accepts m' . By first we mean that any other node that accepted m' , accepted the message in a later round than u , breaking ties arbitrarily. Hence, according to the algorithm, u received an *accept*(m') message from a neighbor w_1 , and stored a message (m', i) in Phase 1, where $(u, w_1) \notin G_i$. Since u is the first node that accepts m' , the node w_1 did not accept m' in the previous round. We conclude that the edge (w_1, u) is the adversarial edge and all other edges are reliable. Because the adversarial edge (w_1, u) was not included in the i 'th graph G_i , all messages of the form (m', i) sent by the adversarial edge in Phase 1 are ignored. Since all other edges are reliable, no node received (and did not ignore) the false message (m', i) during the first phase - in contradiction to the assumption that u stored a message (m', i) in Phase 1, and therefore received the message (m', i) in Phase 1. ◁

From Claim 13 we can conclude that in the case where the adversarial edge initiates a false broadcast, it will not be accepted by any of the nodes.

► **Corollary 14.** *In case $e' = (v_1, v_2)$ initiates the broadcast, no node accepts any message.*

Proof. Since no node initiated the broadcast, in the second phase the only nodes that can receive *accept*(m) messages are v_1 and v_2 over the edge e' . In addition, since e' also initiates the first phase, for every node storing a message (m, i) in Phase 1 it must hold that $e' \in G_i$. Hence, we can conclude that neither v_1 nor v_2 accepts any of the false messages. Consequently, no node in $V \setminus \{v_1, v_2\}$ receives an *accepts*(m) message for any m , as required. ◀

So far, we showed that if a node v accepts a message, it must be the correct one. It remains to show that each node indeed accepts a message during the second phase. Towards that goal, we will show the collection of ℓ sub-algorithms executed in Phase 1 can be simulated in $O(\omega(\mathcal{G}) \cdot L + |\mathcal{G}|)$ rounds. This argument holds regardless of the power of the adversary.

► **Lemma 15.** *Consider an $(L, 1)$ covering family $\mathcal{G} = \{G_1, \dots, G_\ell\}$ for G that is locally known by all the nodes. For a fixed node v , an edge e , and an L -length s - v path $P \subseteq G \setminus \{e, e'\}$, let $G_i \in \mathcal{G}$ be the subgraph containing P where $e \notin G_i$. Then, v receives the message (m_0, i) in Phase 1 within $O(L \cdot \omega(\mathcal{G}) + |\mathcal{G}|)$ rounds of that phase.*

We note that by Observation 9 taking $L = 7D$ yields that for every node v and edge e , it holds that $\text{dist}_{G \setminus \{e, e'\}}(s, v) \leq L$. Hence, by the properties of the covering family \mathcal{G} , for every node v and an edge e there exists an L -length s - v path $P \subseteq G \setminus \{e, e'\}$ and a subgraph G_i that contains P and avoids e . The proof of Lemma 15 is one of the most technical parts

in this paper. Whereas pipeline is a very common technique, especially in the context of broadcast algorithms, our implementation of it is quite nontrivial. Unfortunately, since our adversary has a full knowledge of the randomness of the nodes, it is unclear how to apply the random delay approach of [28, 42] in our setting. We next show that our pipeline approach works well thanks to the bounded width of the covering family.

Proof of Lemma 15. Let $P = (s = v_0, \dots, v_\eta = v)$ be an s - v path in G_i where $\eta \leq L$. For simplicity, we consider the case where the only message propagated during the phase is m_0 . The general case introduces a factor of 2 in the round complexity. This holds since there could be at most two messages of the form $(0, i)$ and $(1, i)$. We also assume, without loss of generality, that each node v_j receives the message (m_0, i) for the *first* time from v_{j-1} . If v_j received (m_0, i) for the first time from a different neighbor in an earlier round, the time it sends the message can only decrease.

In order to show that $v_\eta = v$ receives the message (m_0, i) within $O(L \cdot \omega(\mathcal{G}) + |\mathcal{G}|)$ rounds, it is enough to bound the total number of rounds the message (m_0, i) spent in the *queues* of the nodes of P , waiting to be sent. That is, for every node $v_j \in P$ let r_j be the round in which v_j received the message (m_0, i) for the first time, and let s_j be the round in which v_j sent the message (m_0, i) . In order to prove Lemma 15, our goal is to bound the quantity $T = \sum_{j=1}^{\eta-1} (s_j - r_j)$.

For every $k < i$ we denote the set of edges from P that are not included in the subgraph G_k by $N_k = \{(v_{j-1}, v_j) \in P \mid (v_{j-1}, v_j) \notin G_k\}$, and define $\mathcal{N} = \{(k, e) \mid e \in N_k, k \in \{1, \dots, i-1\}\}$. By the definition of family \mathcal{G} , it holds that:

$$|\mathcal{N}| = \sum_{k=1}^{i-1} |N_k| \leq \eta \cdot \omega(\mathcal{G}) = O(\omega(\mathcal{G}) \cdot L).$$

For every node $v_j \in P$ let Q_j be the set of messages (m_0, k) that v_j sent between rounds r_j and round s_j . By definition, $|Q_j| = s_j - r_j$, and $T = \sum_{j=1}^{\eta-1} |Q_j|$.

Thus, to prove Lemma 15 it's enough to show that $\sum_{j=1}^{\eta-1} |Q_j| \leq |\mathcal{N}|$. This is shown next in two steps. First we define a set I_j consisting of certain (m_0, k) messages such that $|Q_j| \leq |I_j|$, for every $j \in \{1, \dots, \eta-1\}$. Then, we show that $\sum_{j=1}^{\eta-1} |I_j| \leq |\mathcal{N}|$.

Step one. For every node $v_j \in P$, let I_j to be the set of messages (m_0, k) satisfying the following three properties: (1) $k < i$, (2) v_j sent the message (m_0, k) before sending the message (m_0, i) in round s_j , and (3) v_j did not receive the message (m_0, k) from v_{j-1} before receiving the message (m_0, i) in round r_j . In other words, the set I_j includes messages received by v_j , with a graph index at most $i-1$, that are either received from v_{j-1} between round r_j and round s_j , or received by v_j from another neighbor $w \neq v_{j-1}$ by round s_j (provided that those messages were not received additionally from v_{j-1}). Note that it is not necessarily the case that $Q_j \subseteq I_j$, but for our purposes, it is sufficient to show the following.

▷ **Claim 16.** For every $1 \leq j \leq \eta-1$ it holds that $|Q_j| \leq |I_j|$.

Step two. We next show that $\sum_{j=1}^{\eta-1} |I_j| \leq |\mathcal{N}|$ by introducing an injection function f from $\mathcal{I} = \{(v_j, k) \mid (m_0, k) \in I_j\}$ to \mathcal{N} , defined as follows. For $(v_j, k) \in \mathcal{I}$, set $f((v_j, k)) = (k, (v_{h-1}, v_h))$ such that (v_{h-1}, v_h) is the closest edge to v_j on $P[v_0, v_j]$ where $(v_{h-1}, v_h) \in N_k$ (i.e., $(v_{h-1}, v_h) \notin G_k$).

$$f((v_j, k)) = (k, (v_{h-1}, v_h)) \mid h = \min_{\tau < i} \{t \mid (v_{\tau-1}, v_\tau) \in N_k\}.$$

23:10 Broadcast CONGEST Algorithms against Adversarial Edges

We begin by showing the function is well defined.

▷ **Claim 17.** The function $f : \mathcal{I} \rightarrow \mathcal{N}$ is well defined.

Next, we show the function f is an injection.

▷ **Claim 18.** The function f is an injection.

Proof. First note that by the definition of the function f , for every $k_1 \neq k_2$, and $1 \leq j_1, j_2 \leq \eta - 1$ such that $(v_{j_1}, k_1), (v_{j_2}, k_2) \in \mathcal{I}$ it holds that $f((v_{j_1}, k_1)) \neq f((v_{j_2}, k_2))$. Next, we show that for every $k < i$ and $1 \leq j_1 < j_2 \leq \eta - 1$ such that $(v_{j_1}, k), (v_{j_2}, k) \in \mathcal{I}$, it holds that $f((v_{j_1}, k)) \neq f((v_{j_2}, k))$. Denote $f((v_{j_2}, k)) = (k, (v_{h_2-1}, v_{h_2}))$ and $f((v_{j_1}, k)) = (k, (v_{h_1-1}, v_{h_1}))$. We will now show that $(v_{h_2-1}, v_{h_2}) \in P[v_{j_1}, v_{j_2}]$. Since $(v_{h_1-1}, v_{h_1}) \in P[v_0, v_{j_1}]$, it will then follow that $(v_{h_1-1}, v_{h_1}) \neq (v_{h_2-1}, v_{h_2})$.

Assume towards contradiction that $(v_{h_2-1}, v_{h_2}) \in P[v_0, v_{j_1}]$. By the definition of f , we have $P[v_{i_1}, v_{i_2}] \subseteq G_k$. Since $(v_{j_1}, k) \in \mathcal{I}$, the node v_{j_1} sent the message (m_0, k) before sending (m_0, i) . Additionally, since by our assumption, every node $v_t \in P$ receives the message (m_0, i) for the first time from node v_{t-1} (its incoming neighbor on the path P), and all the edges on P are reliable, it follows that v_{j_2} receives the message (m_0, k) from v_{j_2-1} *before* receiving the message (m_0, i) in round r_{j_2} . This contradicts the assumption that $(v_{j_2}, k) \in \mathcal{I}$, as by property (3) of the definition of I_{j_2} , the node v_{j_2} did not receive the message (m_0, k) from v_{j_2-1} before round r_{j_2} . ◁

This completes the proof of Lemma 15. Finally, we show that all nodes accept the message m_0 during the second phase using Lemma 15. This concludes the proof of Theorem 12.

▷ **Claim 19.** All nodes accept m_0 within $O(L)$ rounds from the beginning of Phase 2.

► **Remark.** Our broadcast algorithm does not need to assume that the nodes know the identity of the source node s . By Corollary 14, in case the adversarial edge e' initiates a false broadcast execution, no node will accept any of the messages sent.

Moreover, the same round complexity also holds in the case where there are multiple sources holding the same broadcast message m_0 . This fact will play a role in the final broadcast algorithm where the nodes do not have an estimate on the diameter D .

2.2 Broadcast without Knowing the Diameter

We next show how to remove the assumption that the nodes know an estimate of the diameter; consequently, our broadcast algorithm also computes a linear estimate of the diameter. This increases the round complexity by a logarithmic factor and establishes Theorem 4.

We first describe the algorithm under the simultaneous wake-up assumption and then explain how to remove it.

Algorithm Broadcast. The algorithm applies Algorithm BroadcastKnownDiam of Section 2 for $k = O(\log D)$ iterations in the following manner. Every iteration $i \in \{1, \dots, k\}$ consists of two steps. In the first step, the source node s initiates BroadcastKnownDiam(D_i) with diameter estimate $D_i = 2^i$, and the desired message m_0 . Denote all nodes that accepted the message m_0 by A_i and let $N_i = V \setminus A_i$ be the nodes that did not accept the message.

In the second step, the nodes in N_i inform s that the computation is not yet complete in the following manner. All nodes in N_i broadcast *the same* designated message M by applying Algorithm BroadcastKnownDiam($9D_i$) with diameter estimate $9D_i$ and the message M . The second phase can be viewed as performing a single broadcast from $|N_i|$ multiple sources. If

the source node s receives and accepts the message M during the second step, it continues to the next iteration $i + 1$. If after $\tilde{O}(D_i^2)$ rounds s did not receive and accept the message M , it broadcasts a termination message M_t to all nodes in V using **BroadcastKnownDiam**($7D_i$) (with diameter estimate $7D_i$). Once a node $v \in V$ accepts the termination message M_t , it completes the execution with the output message it has accepted so far. Additionally, for an iteration i in which v accepted the termination message, D_i can be considered as an estimation of the graph diameter.

Analysis. We begin with noting that no node $v \in V$ accepts a wrong message $m' \neq m_0$ as its output. This follows by Claim 13 and the correctness of Algorithm **BroadcastKnownDiam**.

► **Observation 20.** *No node $v \in V$ accepts a wrong message $m' \neq m_0$.*

Fix an iteration i . Our next goal is to show that if $N_i \neq \emptyset$, then s will accept the message M by the end of the iteration. Consider the second step of the algorithm where the nodes in N_i broadcast the message M toward s using **BroadcastKnownDiam**($9D_i$). Since all nodes in N_i broadcast *the same* message M , we refer to the second step as a *single* execution of **BroadcastKnownDiam**($9D_i$) with multiple sources. We begin with showing that the distance between the nodes in A_i and s is at most $14D_i$.

▷ **Claim 21.** For every node $v \in A_i$, it holds that $\text{dist}(s, v, G \setminus \{e'\}) \leq 14D_i$.

Proof. Recall that Algorithm **BroadcastKnownDiam** proceeds in two phases. In the first phase, the source node propagates messages of the form (M, k) , and in the second phase, the source node propagates *accept* messages. For a node v that accepts the message m_0 in the i 'th iteration, according to Algorithm **BroadcastKnownDiam**(D_i), it receives an *accept*(m_0) message from a neighbor w in Phase 2, and stored a message (m_0, k) in Phase 1, such that $(v, w) \notin G_k$. Let P_1 be the path on which the message *accept*(m_0) propagated toward v in Phase 2 of **BroadcastKnownDiam**(D_i). Since the second phase is executed for $7D_i$ rounds, it holds that $|P_1| \leq 7D_i$. In the case where $e' \notin P_1$, since s is the only node initiating *accept*(m_0) messages (except maybe e'), P_1 is a path from s to v in $G \setminus \{e'\}$ as required.

Assume that $e' \in P_1$, and denote it as $e' = (v_1, v_2)$. Without loss of generality, assume that on the path P_1 , the node v_1 is closer to v than v_2 . Hence, v_1 received an *accept*(m_0) message from v_2 during Phase 2, and because v_1 also sent the message over P_1 , it accepted m_0 as its output. Therefore, during the execution of **BroadcastKnownDiam**(D_i), the node v_1 stored a message (m_0, j) during the first phase, where $e' \notin G_j$. As all edges in G_j are reliable, we conclude that G_j contains a s - v_1 path P of length $\eta \leq 7D_i$ such that $e' \notin P$. Thus, the concatenated path $P \circ P_1[v_1, v]$ is a path of length at most $14D_i$ from s to v in $G \setminus \{e'\}$ as required. ◁

We now show that if $N_i \neq \emptyset$ then s accepts the message M during the second step and continues to the next iteration. The proof is very similar to the proof of Claim 19 and follows from the following observation.

► **Observation 22.** *For every $u \in A_i$ and an edge $e = (v, u)$, it holds that $\text{dist}_{G \setminus \{e', e\}}(N_i, u) \leq 7 \cdot 9D_i$.*

Proof. Let T be a BFS tree rooted at s in $G \setminus \{e'\}$ restricted³ to the nodes in A_i . By Claim 21 the depth of T is at most $14D_i$. It follows that the forest $T \setminus \{e\}$ contains at most

³ The tree T might also contain internal nodes in N_i , but it is required to span only the nodes in A_i .

2 trees of diameter $2 \cdot 14D_i$. Since G is 3 edge-connected, there exists a path from some node in N_i to u in $G \setminus \{e, e'\}$.

Hence, the shortest path from N_i to u in $G \setminus \{e, e'\}$ denoted as P can be transformed into a path P' containing at most two edges of P as well as two tree subpaths of the forest $T \setminus \{e\}$. Therefore, $\text{dist}_{G \setminus \{e, e'\}}(N_i, u) \leq |P'| \leq 4 \cdot 14D_i + 2 \leq 7 \cdot 9D_i$. \blacktriangleleft

\triangleright **Claim 23.** If $N_i \neq \emptyset$, s accepts the message M by the end of Step 2 of the i 'th iteration.

Proof. Let $P = (u_0, u_1, \dots, u_\eta = s)$ be a shortest path from some node $u_0 \in N_i$ to the source node s . As P is the shortest such path, for every $j \neq 0$ $u_j \in A_i$, and due to Claim 21 $|P| \leq 14D_i + 1$. In order to prove Claim 23 we will show that every node $u_j \in P$ accepts the message M by round j of Phase 2 in the execution of **BroadcastKnownDiam**($9D_i$) (in Step 2), by induction on j .

Base case: as $u_0 \in N_i$, it accepts the message M at the beginning of the phase. Assume the claim holds for u_j and consider the node u_{j+1} . By Observation 22 there exists a path P_{j+1} from some node in N_i to u_{j+1} in $G \setminus \{e', (u_j, u_{j+1})\}$ of length $|P_{j+1}| \leq 7 \cdot 9D_i$. Hence, by Lemma 15 combined with the covering property of the covering subgraphs family used in **BroadcastKnownDiam**($9D_i$), we conclude that u_{j+1} stored a message (M, τ) in Phase 1 where $(u_j, u_{j+1}) \notin G_\tau$ for some neighbor $w \in N(u_{j+1})$.

By the induction assumption, u_j sends u_{j+1} an *accept*(M) message by round j of Phase 2, and since $(u_{j+1}, u_j) \notin G_\tau$, it follows that the node u_{j+1} accepts the message M by round $j + 1$ as required. \blacktriangleleft

Recall that since the second phase of **BroadcastKnownDiam**(D_i) is executed for $7D_i$ rounds. Hence, when $D_i < D/7$ there must exist a node $w \in V$ that did not accept the message m_0 during the execution of **BroadcastKnownDiam**(D_i) in the first step, and therefore $N_i \neq \emptyset$. On the other hand, when $D_i \geq D$, all nodes in V receive and accept m_0 during the first step of the i 'th iteration and therefore $N_i = \emptyset$. Hence, for an iteration i^* in which no node broadcasts the message M (and therefore s decides to terminate the execution), it holds that $D_{i^*} \in [D/7, D]$. Since s broadcasts the termination message M_t by applying Algorithm **BroadcastKnownDiam**($7D_{i^*}$) with diameter estimate $7D_{i^*}$, we conclude that all nodes in V will finish the execution as required. To omit the wake-up assumption, in the second step of each iteration, the broadcast of message M is initiated by nodes in N_i with neighbors in A_i .

3 Broadcast against t Adversarial Edges

In this section, we consider the broadcast problem against t adversarial edges and prove Theorem 5. The adversarial edges are fixed throughout the execution but are unknown to any of the nodes. Given a D -diameter, $(2t + 1)$ edge-connected graph G , and at most t adversarial edges $F \subseteq E$, the goal is for a source node s to deliver a message m_0 to all nodes in the graph. At the end of the algorithm, each node is required to output the message m_0 . Our algorithm is again based on a locally known family \mathcal{G} with several desired properties. The algorithm floods the messages over the subgraphs of \mathcal{G} . The messages exchanged over each subgraph $G_i \in \mathcal{G}$ contains also the path information along which the message has been received. As we will see, the round complexity of the algorithm is mostly dominated by the cardinality of \mathcal{G} .

We use the following fact from [36], whose proof follows by the proof of Fact 11.

\blacktriangleright **Fact 24** (Implicit in [36]). *Given a graph G and integer parameters L and k , there exists a 0-round algorithm that allows all nodes to locally know an (L, k) covering family $\mathcal{G} = \{G_1, \dots, G_\ell\}$ such that $\ell = ((Lk \log n)^{k+1})$.*

Towards proving Theorem 5, we prove the following which will become useful also for the improved algorithms for expander graphs.

► **Theorem 25.** *Given a $(2t + 1)$ edge-connected graph G of diameter D , and a parameter L satisfying that for every $u, v \in V$, and every set $E \subseteq E$ of size $|E| \leq 2t$, it holds that $\text{dist}_{G \setminus E}(u, v) \leq L$. Then assuming that the nodes locally know an $(L, 2t)$ covering family \mathcal{G} , there exists a deterministic broadcast algorithm $\text{BroadcastKnownCovFamily}(\mathcal{G}, L, t)$ against at most t adversarial edges F with round complexity $O(L \cdot |\mathcal{G}|)$.*

We note that by Observation 9, every $(2t + 1)$ edge-connected graph G with diameter D satisfies the promise of Theorem 25 for $L = (6t + 2)D$. Finally, our algorithm makes use also of the following definition for a minimum s - v cut defined over a collection of s - v paths.

► **Definition 26** (Minimum (Edge) Cut of a Path Collection). *Given a collection of s - v paths \mathcal{P} , the minimum s - v cut in \mathcal{P} , denoted as $\text{MinCut}(s, v, \mathcal{P})$, is the minimal number of edges appearing on all the paths in \mathcal{P} . I.e., letting $\text{MinCut}(s, v, \mathcal{P}) = x$ implies that there exists a collection of x edges E' such that for every path $P \in \mathcal{P}$, it holds that $E' \cap P \neq \emptyset$.*

We are now ready to describe the broadcast algorithm given that the nodes know an $(L, 2t)$ covering family \mathcal{G} (along with the parameters L and t) as specified by Theorem 25. Later, we explain the general algorithm that omits this assumption.

Broadcast Algorithm $\text{BroadcastKnownCovFamily}(\mathcal{G}, L, t)$. Similarly to the single adversarial edge case, the algorithm has two phases, a flooding phase and an acceptance phase. In the first phase of the algorithm, the nodes exchange messages over the subgraphs of \mathcal{G} , that contain also the path information, along which the bit $\{0, 1\}$ is received. In addition, instead of propagating the messages of distinct G_i 's subgraphs in a pipeline manner, we run the entire i 'th algorithm (over the edges of the graph G_i) after finishing the application of the $(i - 1)$ algorithm⁴.

In the first phase, the nodes flood *heard bundles* over all the $G_i \in \mathcal{G}$ subgraphs, defined as follows.

Heard bundles. A *bundle* of heard messages sent from node v to u consists of:

1. A header message $\text{heard}(m, \text{len}, P)$, where P is an s - v path of length len along which v received the message m .
2. A sequence of len messages specifying the edges of P , one by one.

This bundle contains $\text{len} + 1$ messages that will be sent in a pipeline manner in the following way. The first message is the header $\text{heard}(m, \text{len}, P)$ sent in round τ . Then in the next consecutive len rounds, v sends the edges of P in reverse order (from the edge incident to v to s).

Phase 1: Flooding. The first phase consists of $\ell = |\mathcal{G}|$ iterations, where each iteration is implemented using $O(L)$ rounds. At the first round of the i 'th iteration, the source node s sends the message $\text{heard}(m_0, 1, \emptyset)$ to all neighbors. Every node v , upon receiving the *first* bundle message $\text{heard}(m', x, P)$ over an edge in G_i from a neighbor w , stores the bundle $\text{heard}(m', x + 1, P \cup \{w\})$ and sends it to all neighbors. Note that each node stores and sends at most *one* heard bundle $\text{heard}(m', x, P)$ in each iteration (and not one per message m').

⁴ One might optimize the $O(t)$ exponent by employing a pipeline approach in this case as well.

Phase 2: Acceptance. The second phase consists of $O(L)$ rounds, in which *accept* messages are propagated from the source s to all nodes as follows. In the first round s sends $accept(m_0)$, to all neighbors. Every node $v \in V \setminus \{s\}$ decides to accept a message m' if the following two conditions hold: (i) v receives $accept(m')$ from a neighbor w , and (ii) $\text{MinCut}(s, v, \mathcal{P}) \geq t$, where

$$\mathcal{P} = \{P \mid v \text{ stored a } heard(m', len, P) \text{ message and } (v, w) \notin P\}. \quad (1)$$

Note that since the decision here is made by computing the minimum cut of a path collection, it is indeed required (by this algorithm) to send the path information.

Correctness. We begin with showing that no node accepts a false message.

▷ **Claim 27.** No node $v \in V$ accepts a message $m' \neq m_0$ in the second phase.

Proof. Assume by contradiction there exists a node that accepts a false message m' , and let v be the *first* such node. By first we mean that any other node that accepted m' accepted the message in a later round than v breaking ties arbitrarily. Hence, v received a message $accept(m')$ from some neighbor w . Because v is the first such node, the edge (w, v) is adversarial. Let $E' = F \setminus \{(w, v)\}$ be the set of the remaining $t - 1$ adversarial edges, and let \mathcal{P} be given as by Eq. (1). We next claim that $\text{MinCut}(s, v, \mathcal{P}) \leq t - 1$ and thus v does not accept m' .

To see this, observe that any path P such that v received a message $heard(m', len, P)$ must contain at least one edge in E' . This holds even if the content of the path P is corrupted by the adversarial edges. Since there are at most $t - 1$ edges in E' and all the paths in \mathcal{P} are passing through them, it holds that $\text{MinCut}(s, v, \mathcal{P}) \leq t - 1$ as required. ◁

Finally, we show that all nodes in V accept the message m_0 during the second phase. This completes the proof of Theorem 25.

▷ **Claim 28.** All nodes accept m_0 within $O(L)$ rounds from the beginning of Phase 2.

Proof. We will show that all nodes accept the message m_0 by induction on the distance from the source s in the graph $G \setminus F$. Let T be some BFS tree rooted at s in $G \setminus F$. The base case holds vacuously, as s accepts the message m_0 in round 0. Assume all nodes at distance at most i from s in $G \setminus F$ accepted the message by round i . Consider a node v at distance $i + 1$ from s in $G \setminus F$. By the induction assumption on layer i , v receives the message $accept(m_0)$ from a neighbor w in round $j \leq i$ over a reliable edge (w, v) . We are left to show that $\text{MinCut}(s, v, \mathcal{P}) \geq t$, where \mathcal{P} is as given by Eq. (1). Alternatively, we show that for every edge set $E' \subseteq E \setminus \{(w, v)\}$ of size $t - 1$, the node v stores a heard bundle containing m_0 and a path P_k such that $P_k \cap (E' \cup \{(v, w)\}) = \emptyset$ during the first phase. This necessary implies that the minimum cut is at least t .

For a subset $E' \subseteq E$ of size $t - 1$, as $|F \cup E' \cup \{(w, v)\}| \leq 2t$, by the promise on L in Theorem 25, $\text{dist}_{G \setminus (F \cup E' \cup \{(w, v)\})}(s, v) \leq L$. By the covering property of the covering family \mathcal{G} it follows that there exists a subgraph G_k such that $G_k \cap (F \cup E' \cup \{(v, w)\}) = \emptyset$, and $\text{dist}_{G_k}(s, v) \leq L$. Hence, all edges in G_k are reliable, and the only message passed through the heard bundles during the k 'th iteration is the correct message m_0 . Additionally, as $\text{dist}_{G_k}(s, v) \leq L$, the node v stores a heard bundle $heard(m_0, x, P_k)$ during the k 'th iteration, for some s - v path P_k of length $x = O(L)$. As $P_k \subseteq G_k$ it also holds that $P_k \cap (E' \cup \{(v, w)\}) = \emptyset$. We conclude that $\text{MinCut}(s, v, \mathcal{P}) \geq t$, and by the definition of Phase 2, v accepts m_0 by round $j + 1 \leq i + 1$. The claim follows as the diameter of T is $O(L)$ by the promise on L . ◁

Algorithm Broadcast (Proof of Theorem 5). We now describe the general broadcast algorithm. Our goal is to apply Algorithm BroadcastKnownCovFamily(\mathcal{G}, L, t) over the $(L, 2t)$ covering family \mathcal{G} for $L = O(tD)$, constructed using Fact 24. Since the nodes do not know the diameter D (or a linear estimate of it), we make $O(\log D)$ applications of Algorithm BroadcastKnownCovFamily($\widehat{\mathcal{G}}, \widehat{L}, t$) using the $(\widehat{L}, 2t)$ covering family $\widehat{\mathcal{G}}$, for $\widehat{L} = O(t\widehat{D})$ where $\widehat{D} = 2^i$ is the diameter guess for the i 'th application.

Specifically, at the beginning of the i 'th application, the source s initiates Algorithm BroadcastKnownCovFamily(\mathcal{G}_i, L_i, t) with the desired message m_0 over the $(L_i, 2t)$ covering family \mathcal{G}_i constructed using Fact 24 with $L_i = O(tD_i)$ and $D_i = 2^i$. Denote all nodes that accepted the message m_0 at the end of Algorithm BroadcastKnownCovFamily(\mathcal{G}_i, L_i, t) by A_i , and let $N_i = V \setminus A_i$ be the nodes that did not accept the message.

The algorithm now applies an additional step where the nodes in N_i inform s that they did not accept any message in the following manner. All nodes in N_i broadcast *the same* designated message M , by applying Algorithm BroadcastKnownCovFamily(\mathcal{G}'_i, ctL_i, t) over an $(ctL_i, 2t)$ covering family \mathcal{G}'_i , for some fixed constant $c > 0$ (known to all nodes). This can be viewed as performing a single broadcast execution (i.e., with the same source message) but from $|N_i|$ multiple sources. We next set $\tau_i = O(t \cdot D_i \log n)^{O(t)}$ as a bound on the waiting time for a node to receive any acknowledgment.

If the source node s accepts the message M at the end of this broadcast execution, it waits τ_i rounds, and then continues to the next application⁵ $i + 1$ (with diameter guess 2^{i+1}). In the case where s did not accept the message M within τ_i rounds from the beginning of that broadcast execution, it broadcasts a termination message M_T to all nodes in V . This is done by applying Algorithm BroadcastKnownCovFamily(\mathcal{G}'_i, ctL_i, t) over the $(ctL_i, 2t)$ covering family \mathcal{G}'_i . Once a node $v \in V$ accepts the termination message M_T , it completes the execution with the last message it has accepted so far (in the analysis part, we show that it indeed accepts the right message). A node v that did not receive a termination message M_T within τ_i rounds, continues to the next application of Algorithm BroadcastKnownCovFamily.

The correctness argument exploits the fact that for an application i such that $N_i \neq \emptyset$, the graph G' obtained by contracting⁶ all nodes in N_i into a single node a , satisfies the following: (i) it is $(2t + 1)$ edge-connected, (ii) it contains s , and (iii) it has diameter $O(L_i) = O(t \cdot D_i)$. A complete analysis of the algorithm can be found in the full paper [32].

We observe that our broadcast algorithm can be implemented in the LOCAL model using $O(tD \log n)$ many rounds.

► **Corollary 29.** *For every $(2t + 1)$ edge-connected graph, and a source node s , there is a deterministic broadcast algorithm against t adversarial edges that runs in $O(tD \log n)$ local rounds.*

Proof. The algorithm is the same as in the CONGEST model. However, since in the local model there are no bandwidth restrictions, the message propagation over the $|\mathcal{G}|$ subgraphs of the (L, t) covering family can be implemented simultaneously within $L = O(tD)$ rounds. ◀

⁵ We make the source node s wait since in the case where it actually sends a termination message, all nodes accept it within τ_i rounds. Therefore, we need to make sure that all nodes start the next $i + 1$ application at the same time.

⁶ I.e., we contract all edges with both endpoints in N_i .

References

- 1 Ittai Abraham, T.-H. Hubert Chan, Danny Dolev, Kartik Nayak, Rafael Pass, Ling Ren, and Elaine Shi. Communication complexity of byzantine agreement, revisited. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, pages 317–326, 2019.
- 2 Ittai Abraham, Srinivas Devadas, Danny Dolev, Kartik Nayak, and Ling Ren. Synchronous byzantine agreement with expected $O(1)$ rounds, expected $o(n^2)$ communication, and optimal resilience. In *Financial Cryptography and Data Security - 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18-22, 2019, Revised Selected Papers*, pages 320–334, 2019.
- 3 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *Journal of the ACM (JACM)*, 42(4):844–856, 1995.
- 4 John Augustine, Gopal Pandurangan, and Peter Robinson. Fast byzantine leader election in dynamic networks. In *Distributed Computing - 29th International Symposium, DISC 2015, Tokyo, Japan, October 7-9, 2015, Proceedings*, pages 276–291, 2015.
- 5 Baruch Awerbuch, Oded Goldreich, David Peleg, and Ronen Vainish. A trade-off between information and communication in broadcast protocols. *J. ACM*, 37(2):238–256, 1990.
- 6 Anindo Bagchi and S. Louis Hakimi. Information dissemination in distributed systems with faulty units. *IEEE Transactions on Computers*, 43(6):698–710, 1994.
- 7 Piotr Berman and Juan A. Garay. Cloture votes: $n/4$ -resilient distributed consensus in $t+1$ rounds. *Math. Syst. Theory*, 26(1):3–19, 1993.
- 8 Piotr Berman, Juan A. Garay, and Kenneth J. Perry. Towards optimal distributed consensus (extended abstract). In *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989*, pages 410–415. IEEE Computer Society, 1989.
- 9 Douglas M Blough and Andrzej Pelc. Optimal communication in networks with randomly distributed byzantine faults. *Networks*, 23(8):691–701, 1993.
- 10 Greg Bodwin, Michael Dinitz, and Caleb Robelle. Optimal vertex fault-tolerant spanners in polynomial time. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2924–2938. SIAM, 2021.
- 11 Gabriel Bracha. Asynchronous byzantine agreement protocols. *Information and Computation*, 75(2):130–143, 1987.
- 12 Gabriel Bracha and Sam Toueg. Asynchronous consensus and broadcast protocols. *J. ACM*, 32(4):824–840, 1985.
- 13 Keren Censor-Hillel and Tariq Toukan. On fast and robust information spreading in the vertex-congest model. *Theoretical Computer Science*, 2017.
- 14 Diptarka Chakraborty and Keerti Choudhary. New extremal bounds for reachability and strong-connectivity preservers under failures. In *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, pages 25:1–25:20, 2020.
- 15 Bogdan S. Chlebus, Dariusz R. Kowalski, and Jan Olkowski. Fast agreement in networks with byzantine nodes. In *34th International Symposium on Distributed Computing, DISC 2020, October 12-16, 2020, Virtual Conference*, pages 30:1–30:18, 2020.
- 16 Brian A. Coan and Jennifer L. Welch. Modular construction of a byzantine agreement protocol with optimal message bit complexity. *Inf. Comput.*, 97(1):61–85, 1992.
- 17 Ran Cohen, Iftach Haitner, Nikolaos Makriyannis, Matan Orland, and Alex Samorodnitsky. On the round complexity of randomized byzantine agreement. In *33rd International Symposium on Distributed Computing, DISC 2019, October 14-18, 2019, Budapest, Hungary*, pages 12:1–12:17, 2019.
- 18 Michael Dinitz and Robert Krauthgamer. Fault-tolerant spanners: better and simpler. In *Proceedings of the 30th annual ACM SIGACT-SIGOPS symposium on Principles of distributed computing*, pages 169–178. ACM, 2011.

- 19 Danny Dolev. The byzantine generals strike again. *J. Algorithms*, 3(1):14–30, 1982.
- 20 Danny Dolev, Michael J. Fischer, Robert J. Fowler, Nancy A. Lynch, and H. Raymond Strong. An efficient algorithm for byzantine agreement without authentication. *Information and Control*, 52(3):257–274, 1982.
- 21 Danny Dolev and Ezra N. Hoch. Constant-space localized byzantine consensus. In *Distributed Computing, 22nd International Symposium, DISC 2008, Arcachon, France, September 22-24, 2008. Proceedings*, pages 167–181, 2008.
- 22 Cynthia Dwork, David Peleg, Nicholas Pippenger, and Eli Upfal. Fault tolerance in networks of bounded degree. *SIAM J. Comput.*, 17(5):975–988, 1988.
- 23 Pesech Feldman and Silvio Micali. An optimal probabilistic protocol for synchronous byzantine agreement. *SIAM J. Comput.*, 26(4):873–933, 1997.
- 24 Michael J Fischer. The consensus problem in unreliable distributed systems (a brief survey). In *International Conference on Fundamentals of Computation Theory*, pages 127–140. Springer, 1983.
- 25 Mattias Fitzi and Ueli Maurer. From partial consistency to global broadcast. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 494–503, 2000.
- 26 Chaya Ganesh and Arpita Patra. Broadcast extensions with optimal communication and round complexity. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, pages 371–380, 2016.
- 27 Juan A. Garay and Yoram Moses. Fully polynomial byzantine agreement for $n > 3t$ processors in $t + 1$ rounds. *SIAM J. Comput.*, 27(1):247–290, 1998.
- 28 Mohsen Ghaffari. Near-optimal scheduling of distributed algorithms. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21 - 23, 2015*, pages 3–12, 2015.
- 29 Parikshit Gopalan, Raghu Meka, Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Better pseudorandom generators from milder pseudorandom restrictions. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 120–129, 2012.
- 30 Fabrizio Grandoni and Virginia Vassilevska Williams. Faster replacement paths and distance sensitivity oracles. *ACM Transactions on Algorithms (TALG)*, 16(1):1–25, 2019.
- 31 Fabrizio Grandoni and Virginia Vassilevska Williams. Faster replacement paths and distance sensitivity oracles. *ACM Trans. Algorithms*, 16(1):15:1–15:25, 2020.
- 32 Yael Hitron and Merav Parter. Broadcast CONGEST algorithms against adversarial edges. *CoRR*, abs/2004.06436, 2021.
- 33 Yael Hitron and Merav Parter. General CONGEST algorithms compilers against adversarial edges. In *DISC*, 2021.
- 34 Damien Imbs and Michel Raynal. Simple and efficient reliable broadcast in the presence of byzantine processes. *arXiv preprint arXiv:1510.06882*, 2015.
- 35 David R Karger. Random sampling in cut, flow, and network design problems. *Mathematics of Operations Research*, 24(2):383–413, 1999.
- 36 Karthik C. S. and Merav Parter. Deterministic replacement path covering. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 704–723, 2021.
- 37 Jonathan Katz and Chiu-Yuen Koo. On expected constant-round protocols for byzantine agreement. In *Annual International Cryptology Conference*, pages 445–462. Springer, 2006.
- 38 Muhammad Samir Khan, Syed Shalan Naqvi, and Nitin H. Vaidya. Exact byzantine consensus on undirected graphs under local broadcast model. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, pages 327–336, 2019.

- 39 Valerie King, Jared Saia, Vishal Sanwalani, and Erik Vee. Towards secure and scalable computation in peer-to-peer networks. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 87–98, 2006.
- 40 Chiu-Yuen Koo. Broadcast in radio networks tolerating byzantine adversarial behavior. In Soma Chaudhuri and Shay Kutten, editors, *Proceedings of the Twenty-Third Annual ACM Symposium on Principles of Distributed Computing, PODC 2004, St. John's, Newfoundland, Canada, July 25-28, 2004*, pages 275–282. ACM, 2004.
- 41 Leslie Lamport, Robert E. Shostak, and Marshall C. Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.
- 42 Frank Thomson Leighton, Bruce M Maggs, and Satish B Rao. Packet routing and job-shop scheduling ino (congestion+ dilation) steps. *Combinatorica*, 14(2):167–186, 1994.
- 43 Alexandre Maurer and Sébastien Tixeuil. On byzantine broadcast in loosely connected networks. In *Distributed Computing - 26th International Symposium, DISC 2012, Salvador, Brazil, October 16-18, 2012. Proceedings*, pages 253–266, 2012.
- 44 Kartik Nayak, Ling Ren, Elaine Shi, Nitin H. Vaidya, and Zhuolun Xiang. Improved extension protocols for byzantine broadcast and agreement. In *34th International Symposium on Distributed Computing, DISC 2020, October 12-16, 2020, Virtual Conference*, pages 28:1–28:17, 2020.
- 45 Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994.
- 46 Merav Parter. Small cuts and connectivity certificates: A fault tolerant approach. In *33rd International Symposium on Distributed Computing (DISC 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.
- 47 Merav Parter and Eylon Yogev. Congested clique algorithms for graph spanners. In *32nd International Symposium on Distributed Computing (DISC 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.
- 48 Merav Parter and Eylon Yogev. Distributed algorithms made secure: A graph theoretic approach. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1693–1710. SIAM, 2019.
- 49 Merav Parter and Eylon Yogev. Secure distributed computing made (nearly) optimal. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, pages 107–116, 2019.
- 50 Marshall Pease, Robert Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *Journal of the ACM (JACM)*, 27(2):228–234, 1980.
- 51 Andrzej Pelc. Fault-tolerant broadcasting and gossiping in communication networks. *Networks: An International Journal*, 28(3):143–156, 1996.
- 52 Andrzej Pelc and David Peleg. Broadcasting with locally bounded byzantine faults. *Inf. Process. Lett.*, 93(3):109–115, 2005.
- 53 Andrzej Pelc and David Peleg. Feasibility and complexity of broadcasting with random transmission failures. *Theoretical Computer Science*, 370(1-3):279–292, 2007.
- 54 David Peleg. *Distributed Computing: A Locality-sensitive Approach*. SIAM, 2000.
- 55 Nicola Santoro and Peter Widmayer. Time is not a healer. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 304–313. Springer, 1989.
- 56 Nicola Santoro and Peter Widmayer. Distributed function evaluation in the presence of transmission faults. In *Algorithms, International Symposium SIGAL '90, Tokyo, Japan, August 16-18, 1990, Proceedings*, pages 358–367, 1990.
- 57 Sam Toueg, Kenneth J Perry, and TK Srikanth. Fast distributed agreement. *SIAM Journal on Computing*, 16(3):445–457, 1987.
- 58 Eli Upfal. Tolerating a linear number of faults in networks of bounded degree. *Inf. Comput.*, 115(2):312–320, 1994.

- 59 Salil P. Vadhan. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science*, 7(1-3):1–336, 2012.
- 60 Oren Weimann and Raphael Yuster. Replacement paths and distance sensitivity oracles via fast matrix multiplication. *ACM Transactions on Algorithms (TALG)*, 9(2):1–13, 2013.
- 61 Christian Wulff-Nilsen. Fully-dynamic minimum spanning forest with improved worst-case update time. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1130–1143, 2017.