Boosting the Efficiency of Byzantine-tolerant Reliable Communication *

Silvia Bonomi[1], Giovanni Farina[2,1] ✉, and Sébastien Tixeuil[2]

[1] Sapienza Università di Roma, Rome, Italy
bonomi@diag.uniroma1.it
[2] Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
giovanni.farina@lip6.fr, sebastien.tixeuil@lip6.fr

Abstract. Reliable communication is a fundamental primitive in distributed systems prone to Byzantine (*i.e.* arbitrary, and possibly malicious) failures to guarantee integrity, delivery and authorship of messages exchanged between processes. Its practical adoption strongly depends on the system assumptions. One of the most general (and hence versatile) such hypothesis assumes a set of processes interconnected through an unknown communication network of reliable and authenticated links, and an upper bound on the number of Byzantine faulty processes that may be present in the system, known to all participants.
To this date, implementing a reliable communication service in such an environment may be expensive, both in terms of message complexity and computational complexity, unless the topology of the network is known. The target of this work is to combine the Byzantine fault-tolerant topology reconstruction with a reliable communication primitive, aiming to boost the efficiency of the reliable communication service component after an initial (expensive) phase where the topology is partially reconstructed. We characterize the sets of assumptions that make our objective achievable, and we propose a solution that, after an initialization phase, guarantees reliable communication with optimal message complexity and optimal delivery complexity.

Keywords: Reliable communication · Byzantine fault tolerance · Topology reconstruction

1 Introduction

Reliable communication primitives are fundamental building blocks for a distributed system, guaranteeing the eventual delivery of all messages sent by correct processes to their intended receivers. Their employment is particularly relevant when a fraction of processes may suffer arbitrary failures i.e., they are

* This work was performed within Project ESTATE (Ref. ANR-16-CE25-0009-03), supported by French state funds managed by the ANR (Agence Nationale de la Recherche) and it has been partially supported by the INOCS Sapienza Ateneo 2017 Project (protocol number RM11715C816CE4CB). Giovanni Farina wishes to thank *Université Franco-Italienne/Universitá Italo-Francese* (UFI/UIF) for supporting his mobility through the Vinci grant 2018.

Byzantine and may deviate from the protocol by dropping messages, altering their content, or generating spurious messages.

The availability of a reliable communication primitive strongly depends on the system behavior and on its capability to match the set of assumptions required to ensure the correctness of the reliable communication specification. In particular, it has been shown that such a primitive can be efficiently implemented when every process can directly exchange messages with every other [5], also in presence of a bounded and known number of Byzantine processes. However, full connectivity is a strong assumption in large networks and it results impractical whenever scalability is envisioned.

When considering multi-hop networks i.e., systems where every process can communicate directly only with a subset of the others, several results exist to build a Byzantine-tolerant reliable communication primitive. In this paper, we are interested in the solutions designed for multi-hop networks where the topology is not known to the participants. In this context, [8] defined a solution working under the assumption that processes are sufficiently connected. However, providing a reliable communication service in such a general environment may mandate a huge amount of messages and may require very high computational power. Those complexity issues can somewhat be reduced to a tractable problem when either the entire network topology is known to all the processes [8] or it satisfies specific topological requirements [17]. Thus, a naive approach to build an efficient reliable communication primitive is to act in two steps: (i) run a topology reconstruction algorithm to infer the network graph and (ii) use an efficient reliable communication protocol for known network on the reconstruction just inferred. Unfortunately, Byzantine fault-tolerant topology reconstruction has been proved difficult [16], and the final topology inferred does not perfectly match the real one. Besides, correct peers may end the topology reconstruction algorithm by obtaining different network graphs.

Given a network topology G unknown to processes, our goal in this paper is to detail how properly combine the two steps of the described naive approach and to study the set of conditions that G must satisfy to correctly support it. The rationale of this work is that the high topology reconstruction overhead only needs to be paid once, afterwards, reliable communication can be achieved efficiently (otherwise, it would have remained always extremely expensive). The main difficulty is to ensure that discrepancies in the topology reconstructions do not hinder the proper functioning of the reliable communication system. Our work builds upon two reliable communication protocols (`DolevR` and `DolevU` [8]), and a topology reconstruction one (`Explorer` [16]). In more detail, we characterize the sets of assumptions that make our objective achievable, and we propose a solution that, after an initialization phase, guarantees reliable communication with optimal message complexity and delivery complexity.

Due to space constraints, minor proofs of Properties, Lemmas and Theorems have been reported in the technical report version of this paper [3].

2 Related Works

Several solutions have been proposed in the literature to build Byzantine-tolerant reliable communication primitives. A seminal contribution was provided by Dolev [8], assuming (i) processes interconnected through a possibly multi-hop communication network (ii) and an upper bound f on the number of Byzantine faulty processes present in the system (*globally bounded failure model*). Dolev proved that a $(2f+1)$-connected network is required to build a reliable communication primitive in presence of f Byzantine participants i.e., the node connectivity of the communication network must be greater than twice the maximum estimated number of faulty processes. Dolev proposed two protocols working with different assumptions on the knowledge of the network topology by participating processes. More precisely, the lack of topology knowledge impacts both the message complexity (*i.e.*, the number of messages exchanged in the system during a reliable communication instance) and the delivery complexity (*i.e.*, the computational complexity of the procedure used to validate a message) of the protocol. The Dolev's protocol for unknown networks was recently revised to reduce its message complexity [4]. To the best of our knowledge, no other contribution addressed the reliable communication problem in the globally bounded failure model without considering stronger assumptions. When moving to the *locally bounded failure model* (where every process is linked to at most f Byzantine peers) other approaches have been defined [18]. The *Certified Propagation Algorithm* (CPA) was proposed as a reliable communication protocol by Pelc and Peleg, and it has been proven optimal, for the number of faulty processes that can be simultaneously tolerated [17]. Let us note that, either assuming a globally or locally bounded failure model, a dense communication network is required to enable reliable communication in a distributed system. For this reason, weaker primitives have been defined, allowing a (small) part of correct processes to either deliver spurious messages (i.e. messages not generated by their claimed author) or to never deliver a valid message [13,12,14]. These weaker versions enable almost reliable communication also on sparse communication networks.

All the aforementioned solutions do not necessarily rely on digital signatures or other cryptographic primitives. Indeed, the goal of Byzantine-tolerant algorithms is to withstand (computationally) unbounded adversaries that are potentially able to solve (computationally hard) problems on which cryptographic primitives are based upon. Nevertheless, links are assumed to be authenticated and reliable, so if u and v are linked, every message v received from u has been previously sent by u. Notice that cryptographic primitives are not necessary to implement authenticated links [19].

The Byzantine fault-tolerant topology reconstruction problem has been analyzed by Nesterenko and Tixeuil [16] assuming the globally bounded failure model. Then, temporary arbitrary faults have been considered by Dolev et al., defining a self-stabilizing Byzantine-tolerant solution [9].

3 Preliminaries

3.1 System Model

We consider an asynchronous distributed system [5] composed by a set P of n processes, each associated with a unique identifier i.e., $P = \{p_1, p_2, \ldots, p_n\}$.

Failure Model. We consider the *globally bounded Byzantine failure model*, namely we assume that inside the system there might be at most f Byzantine faulty processes. All other peers are assumed *correct*. Let us note that the identity of Byzantine processes is not known to correct ones.

Messages and Communication. Processes communicate by exchanging messages on top of a communication network made of *reliable* and *authenticated* links [5]. It means that messages cannot be lost or altered during their transmission and that the identity of their sender cannot be forged. Such communication network is abstracted by an undirected graph $G = (P, E)$ where the set of nodes V corresponds the set of processes participating in the system and the set of edges E contains an element $e_{i,j}$ for each existing link connecting two processes p_i and p_j. We assume the node connectivity k of G greater than twice the number of the potentially faulty processes i.e., $k > 2f$ [3].

On top of the communication network, two alternative primitives are available: *unicast* (*UL*) and *local broadcast* (*LBL*) links [1,2,11]: the former interconnect single pairs of processes p_i, p_j; the latter attach a process p_i to many others, such that if a message is sent by p_i then it is received by all of its neighbors, thus preventing a faulty process to *equivocate* (i.e., to transmit conflicting messages to different neighbors).

We assume that processes are unaware about the topology of the communication network, namely the graph G: they either know the identifier of the peers they have a link with (*known neighborhood* i.e., *KN* assumption) or they have no knowledge about (*unknown neighborhood* i.e., *UN* assumption).

We refer with *source* to the advertised author of a message, and with *sender* to the process that is sending a message through a link.

3.2 Problem Specification: Reliable Communication

We investigate the *reliable communication* problem between a *source* process p_s and a *target* process p_t. Informally, when addressing this problem, the goal is to define a distributed protocol able to deliver only the messages generated by a correct source to every correct process in the system.

Let us note that, in the literature, the term *message* is commonly used instead of *content* when formalizing a problem specification based on message exchanges. However, several messages carrying the same payload can be diffused in a system to solve the reliable communication problem. Therefore, for ease of presentation, we will refer with *content* to the payload diffused by a process and with *message*

[3] It is not possible otherwise to achieve reliable communication in the system model we are considering [8].

to union of a content and the protocol specific overhead.
More formally, we will say that a protocol solves the reliable communication problem if, for every pair of processes p_s and p_t in the system, both the following conditions are satisfied:

- (**Safety**) if p_t is correct and it delivers a content m from p_s, then p_s previously sent m;
- (**Liveness**) if p_s and p_t are both correct and p_s sends a content m to p_t, then p_t eventually delivers m from p_s.

We refer with *spurious* content to one not sent by its claimed source (i.e. a content initially diffused by some Byzantine processes).

3.3 Evaluation Metrics

We will evaluate the solutions to the reliable communication problem in terms of (i) *message complexity* i.e., the number of messages that the protocol generates to solve the problem and (ii) *delivery complexity* i.e., the computational complexity of the procedure that allows a target process p_t to decide if a content can be delivered or not.

3.4 The Topology Reconstruction Problem

Given an unknown network G of correct and Byzantine faulty processes, the aim of a distributed protocol addressing the topology reconstruction problem is to enable every correct process p_i to reconstruct a subset of the topology of the communication network G. Such a reconstruction G_i is expected to be as complete as possible. The nodes of the communication network G can be partitioned in *correct* and *faulty*, and its edges in *correct*, *one-faulty* and *two-faulty*, respectively interconnecting two correct processes, a correct process and a faulty one, and two Byzantine processes. Likewise, the nodes and edges of a topology reconstruction G_i can be either *real* or *spurious*, respectively mapping or not nodes and edges in G.

3.5 Basic Definitions

For the sake of presentation, this section recalls some definitions and results coming from graph theory [6] that will be employed in this work.

Let us consider an undirected graph $G = (V, E)$. A *path* $\mathcal{P}$ is a sequence of nodes with no repetition i.e., $\mathcal{P} = [v_1, v_2, \ldots, v_m]$ (with $v_i \in V$) such that for each pair of adjacent elements v_i, v_{i+1} there exists an edge $e_{i,i+1} \in E$. The first and last elements of a path are referred with *endpoints*.

A pair of nodes $v_i, v_j \in V$ is *connected* if there exists at least one path $\mathcal{P}_{i,j}$ between them in G, it is *disconnected* otherwise. Given two nodes v_i and v_j, many paths between them may exist. Given a set of paths $\mathcal{P}^1_{i,j}, \mathcal{P}^2_{i,j}, \ldots, \mathcal{P}^x_{i,j}$ between two nodes v_i and v_j they are said *node disjoint* if they share no vertex except for their endpoints.

We refer with $\Pi_{i,j}$ to a *disjoint paths solution* between nodes v_i and v_j, i.e. a set of node disjoint paths having v_i and v_j as endpoints. The *local node connectivity* $\kappa_{i,j}$ between two nodes v_i, v_j is the minimum number of nodes that has to be removed from G to disconnect v_i from v_j. The *node connectivity* of a graph is the minimum value k for the local node connectivity $\kappa_{i,j}$ (i.e., $k = \mathsf{min}(\kappa_{i,j}), \forall v_i, v_j \in V$). A graph having node connectivity greater or equal than k is said *k-connected* graph. The local node connectivity between two nodes is equal to the maximum number of disjoint paths that exist between them (Menger theorem [15]). It is possible to compute a disjoint paths solution $\Pi_{i,j}$ between two nodes v_i, v_j of maximum size (namely $\kappa_{i,j}$) with a deterministic algorithm with computational complexity polynomial in the size of the graph [10,7]. In the following, we will consider every disjoint paths solution $\Pi_{i,j}$ always of maximum size $\kappa_{i,j}$.

4 Dolev Protocols

Dolev [8] identified the necessary and sufficient conditions to solve reliable communication in the system model we consider.

Remark 1. The reliable communication problem can be solved if and only if the node connectivity of the communication graph is greater than $2f$ i.e., $k > 2f$.

Dolev provided two protocols that work under different assumptions on the (partial) knowledge that processes have about the network topology.

4.1 Dolev's Routed Protocol (`DolevR`)

`DolevR` is a protocol solving reliable communication in routed-networks [8], i.e. systems where all messages are relayed over (and only) fixed and known paths. Specifically, processes employing `DolevR` route contents between each pair of process p_i, p_j over $2f+1$ disjoint paths $\Pi_{i,j}$. The reliable and authenticated links restrict the capabilities of faulty processes, allowing them to diffuse spurious contents through at most f paths of any $\Pi_{i,j}$. The assumption of a $(2f+1)$-connected network guarantees that at least $f+1$ paths of any $\Pi_{i,j}$ are fault-free (i.e. they do not pass through any faulty processes). A process p_j employing `DolevR` delivers a content m from a process p_i if it is received through at least $f+1$ routes of $\Pi_{i,j}$.

Protocol Complexity. The message complexity of `DolevR` is linear in the size of the network, whereas its delivery complexity is linear in the number of maximum assumed faults, as detailed in the following Lemmas.

Lemma 1. *`DolevR` solves the reliable communication problem with $O(n)$ message complexity.*

Lemma 2. *`DolevR` solves the reliable communication problem with $O(f)$ delivery complexity.*

The delivery complexity and message complexity of `DolevR` are optimal solving the reliable communication problem in the system model we consider.

Theorem 1. *`DolevR` solves the reliable communication problem in routed networks assuming the globally bounded Byzantine failure model with optimal message complexity and optimal delivery complexity.*

Proof. Given Lemmas 1 and 2, we show that no algorithm can solve the reliable communication problem, in the settings considered in this paper, with an asymptotically lower complexity without considering additional assumptions.

Let us consider two processes p_s and p_t, not connected by a link, respectively as source and target of a reliable communication instance.

The target process relies on the messages it receives from its neighbors to deliver a content. Nevertheless, up to f of its neighbors could be Byzantine faulty and process p_t cannot identify them. Thus, a $O(f)$ procedure is required.

Given that p_s and p_t are not linked, a content must be relayed over fault-free paths (i.e. not including any faulty process) to achieve liveness of reliable communication. In the worst-case scenario the length of the longest fault-free path is $n - k$. □

4.2 Dolev's Topology Unaware Protocol (`DolevU`)

`DolevU` protocol solves the reliable communication problem in unknown networks [8], where contents are flooded in the system. Specifically, `DolevU` spreads messages $\langle m, path \rangle$, in which m is the content and *path* is a list data structure collecting the identifier of processes that are traversed by m. The source process starts the communication multicasting to all of its neighbors the content m with an empty *path*. Then, every process p_i that receives a message $\langle m, path \rangle$ from a neighbor p_j adds the identifier of p_j to *path*, it stores $\langle m, path + \{j\} \rangle$ and it relays such a message to all of its neighbors not yet included in $path + \{j\}$. Every process that succeeds identifying $f + 1$ disjoint *path* among the ones it received with a content m delivers m.

Protocol Complexity. The message complexity of `DolevU` is factorial in the size of the network, whereas a NP-Complete problem has to be solved verifying every content, as detailed in the following Lemmas.

Lemma 3. *`DolevU` solves the reliable communication problem with a message complexity factorial in the number of processes.*

Lemma 4. *`DolevU` solves the reliable communication problem with a NP delivery complexity.*

The `DolevU` protocol has been recently reviewed to reduce its message complexity [4]. It has been proven that modifications can be adopted in the protocol preventing some messages to be generated. Nonetheless, it is still an open problem whether it is always possible to solve reliable communication in unknown networks, under the weakest assumptions identified by Dolev (Remark 1), with a protocol having polynomial message complexity and/or polynomial delivery complexity. For sake of simplicity, we do not employ the reliable communication protocol defined in [4], given that its worst-case delivery complexity and message complexity is unchanged with respect `DolevU`.

The `DolevU` protocol provides the following additional guarantee in case local broadcast links are assumed.

Theorem 2. *Let `DolevU` solve reliable communication in a network G with local broadcast links. Then, a content m is delivered by every correct process if it is delivered by any correct one.*

Proof. When the reliable communication necessary correctness condition is met (Remark 1), the `DolevU` protocol guarantees that if the source p_s of a content m is correct, then any correct target eventually delivers m. This is not guaranteed in case of a faulty source: it may diverge from the protocol and it may prevent some targets from delivering its contents. The local broadcast links provide an additional guarantee: every message a process sends is received by all its neighbors. A correct source in `DolevU` multicast message $\langle m, \emptyset \rangle$ to all of its neighbors. It follows that if a correct process delivered m, then message $\langle m, \emptyset \rangle$ has been sent to all neighbors of p_s, given the local broadcast links, and the claim follows. □

5 `Explorer`

Nesterenko and Tixeuil analyzed the Byzantine fault-tolerant topology reconstruction problem [16]. Among the results they provided, two impossibilities have been identified.

Remark 2. No algorithm can decide whether a two-faulty edge exists [16].

Remark 3. No algorithm can compute a reconstruction of only real nodes and edges while including both all correct and all one-faulty edges [16].

They also defined `Explorer`, an algorithm that enables processes to partially reconstruct the topology of G in the globally bounded failure model assuming KN. It is specified only by the following two procedures: every process p_i *1)* broadcasts its neighborhood $\Gamma(i)$ (namely it broadcasts the identifier of processes it has a link with) and *2)* it stores all neighborhoods $\Gamma(j)$ delivered with a reliable communication primitive in a dictionary data structure $cTop_i := \bigcup \langle j, \Gamma(j) \rangle$.

We introduce a simple neighborhood discovery procedure to cope with the unknown neighborhood scenario, defined by the following actions: *1)* every process multicasts a *HELLO* message (basically a message with no payload), and *2)* every process that receives a *HELLO* message adds the identifier of the sender to its neighborhood.
Then, every process p_i broadcasts with a reliable communication primitive its neighborhood $\Gamma(i)$ every time that it changes, and it updates the entry $\langle j, \Gamma(j) \rangle \in cTop_i$ if $\langle j, \Gamma(j)' \rangle$, such that $\Gamma(j) \subset \Gamma(j)'$, is delivered.
Additionally, if local broadcast links are assumed, every process p_i that delivers

two neighborhood $\Gamma(j)$ and $\Gamma(j)'$ from p_j, such that $\Gamma(j)' \not\subset (\Gamma(j) \in cTop_i)$ and $(\Gamma(j) \in cTop_i) \not\subset \Gamma(j)'$, do not consider j for the reconstruction.

Every process p_i computes the reconstruction $G_i(P_i, E_i)$ from $cTop_i$ as follows:

- $\forall \langle u, \Gamma(u) \rangle \in cTop_i \Rightarrow \exists u \in P_i$;
- $\forall \langle v, \Gamma(v) \rangle, \langle u, \Gamma(u) \rangle \in cTop_i,\ u \in \Gamma(v) \Rightarrow\ \exists (v, u) \in E_i$.
- $\forall v \in \Gamma(u), \langle u, \Gamma(u) \rangle \in cTop : X \leftarrow \bigcup u, |X| > f \Rightarrow \exists v \in P_i$.

We report some properties of any reconstructed topology G_i computed with the defined protocol.

Property 1. (From [16]) $j \notin P \Rightarrow j \notin P_i$ (no G_i contains non-existent nodes).

Property 2. Assuming the *unknown neighborhood* assumption (UN), some reconstruction G_i may never include some Byzantine processes.

Property 3. (From [16]) Assuming the *known neighborhood* assumption (KN), the reconstruction G_i eventually guarantees the following property: $j \in P_i \Leftrightarrow p_j \in P$ (Property 1 + all real nodes are eventually detected).

Property 4. $\forall \langle u, v \rangle \in E,\ u, v \in Correct \Rightarrow \exists \langle u, v \rangle \in E_i$ (all correct edges are eventually contained in G_i).

Property 5. $\forall \langle u, v \rangle \in E_i,\ \langle u, v \rangle \notin E \Rightarrow u \in Byzantine$ (every spurious edge contains at least one Byzantine process).

Property 6. (From [16]) Assuming the *known neighborhood* assumption (KN): $\forall \langle u, v \rangle \in E,\ u \in Correct,\ v \in Byzantine \Rightarrow\ \exists \langle u, v \rangle \in E_i$ (all one-faulty edges will eventually be present in any G_i).

Property 7. Assuming local broadcast links (LBL), all one-faulty edges between a Byzantine process and all of its correct neighbors are eventually either all or none present in every G_i.

Property 8. Assuming local broadcast links (LBL), all correct processes eventually share the same topology reconstruction.

Property 9. No reconstruction G_i computed assuming local broadcast links (LBL) will ever contain more real edges than one obtained assuming the known neighborhood assumption (KN).

5.1 Protocol Complexity Analysis

All correct processes p_i in `Explorer` broadcast their neighborhood $\Gamma(i)$. Supposing the know neighborhood assumption (KN), every process broadcasts such information only once. It follows that `Explorer` requires $\mathcal{O}(n)$ reliable communication executions to enable all correct processes to compute G_i. Considering the unknown neighborhood (UN) assumption, every process has to perform the neighborhood discovery and then to broadcast its $\Gamma(i)$. Unfortunately, no process p_i knows how many nodes have to be detected before diffusing $\Gamma(i)$, and thus, they may broadcast their neighborhood many times, $n - f - 1$ in the worst-case scenario. It follows that `Explorer` with neighborhood discovery executes $\mathcal{O}(n^2)$ reliable communication instances to enable all correct processes to compute G_i.

5.2 Fault-free Disjoint Path Solution

The `Explorer` protocol enables processes to partially reconstruct the topology of G. We showed that different sets of assumptions provide more or less accurate reconstructions (Properties 1-9). We reported the `DolevR` protocol, that it leverages disjoint routes defined between all pairs of processes to achieve reliable communication. We highlighted how f Byzantine faulty processes may compromise at most f paths of any disjoint path solution $\Pi_{i,j}$ in `DolevR`, and that the liveness of such a protocol is guaranteed by the existence of disjoint path solutions of size greater than $2f$ between all pairs of processes, where at least $f+1$ paths cannot be compromised. It follows that, if every pair of correct processes is able to identify a disjoint path solution interconnecting them where at least $f+1$ paths are *faults-free* (i.e. they do not include any Byzantine faulty process), *real* and *disjoint* (*FF_R_D*), then they are able to achieve reliable communication.

We analyze several sets of assumptions that enable all pairs of correct processes p_i, p_j to compute a disjoint path solutions $\Pi_{i,j}$ in G_i containing at least $f+1$ FF_R_D paths.

Theorem 3. *The set of assumptions a)* $\mathbf{k > 3f}$*, b)* ***unicast links*** *and c)* ***unknown neighborhood enables*** *every correct process* p_i *to compute a disjoint paths solution* $\Pi_{i,j}$ *toward any correct process* p_j *that contains at least* $f+1$ *faults-free, real and disjoint paths.*

Proof. Let us assume processes employing `Explorer` and that all messages it generates have been already delivered by the peers. The unknown neighborhood assumption and unicast links allow Byzantine faulty processes to decide which one-faulty and two-faulty edges to declare (Remarks 2,3), thus the local connectivity between any two processes p_i, p_j in the reconstructed topology may be reduced by at most f. It follows that any disjoint paths solution $\Pi_{i,j}$ will contain more than $2f$ paths (Property 4). Given that at most f paths of any $\Pi_{i,j}$ may contain faults the claim follows. □

Theorem 4. *The set of assumptions a)* $\mathbf{k \leq 3f}$*, b)* ***unicast*** *link and c)* ***unknown neighborhood is not sufficient*** *to enable every correct process* p_i *to compute a disjoint paths solution* $\Pi_{i,j}$ *toward every correct process* p_j *containing at least* $f+1$ *faults-free, real and disjoint paths with any protocol.*

Proof. The unknown neighborhood assumption and the unicast links allow faulty processes to decide which one-faulty and two-faulty edges are detectable by correct processes (Remarks 2,3). It follows that the faulty processes may potentially be able to reduce the local connectivity between some pairs of correct processes p_i, p_j by f: the local connectivity $\kappa_{i,j}$ in G_i may be lower than $2f$ and at most $2f-1$ disjoint path $\Pi_{i,j}$ will be identifiable between p_i and p_j, whatever algorithm is envisioned for the reconstruction. Then, up to f paths in $\Pi_{i,j}$ may include faulty processes and the claim follows. □

Theorem 5. *The set of assumptions a)* $\mathbf{k > 2f + \lfloor f/2 \rfloor}$*, b)* ***local broadcast*** *links and c)* ***unknown neighborhood enables*** *every correct process* p_i *to compute a disjoint paths solution* $\Pi_{i,j}$ *toward any correct process* p_j *containing at least* $f+1$ *faults-free, real and disjoint paths.*

Proof. Given Property 7, let us suppose that $f_d \leq f$ Byzantine processes decide to be detected by their neighbors and they send the *HELLO* message, whereas $f - f_d$ ones do not. Let us assume that all messages exchanged by `Explorer` have been already delivered and let us consider $\Pi_{i,j}$ as the disjoint path solution computed on G_i between a pair of correct processes p_i and p_j. The assumption on the node connectivity of G guarantees that at least $2f + \lfloor f/2 \rfloor + 1$ disjoint paths exist between p_i and p_j in the communication network. The undeclared Byzantine processes may reduce the local connectivity between p_i and p_j by $f - f_d$ in G_i. Let us temporarily assume, for the purpose of the proof, that the declared Byzantine processes behave as correct ones. It follows, from Property 4 and 7 of `Explorer`, that the size of $\Pi_{i,j}$ would be at least equal to:

$$2f + \lfloor f/2 \rfloor + 1 \ \ -(\mathbf{f} - \mathbf{f_d}) = f + \lfloor f/2 \rfloor + 1 + f_d$$

Specifically, all paths between p_i and p_j that contain only correct or declared Byzantine processes existing in G are present in G_i.

Let us now consider the declared Byzantine processes not reporting the edges existing between them (i.e. the two-faulty edges). It follows that the paths in G containing two-faulty edges may not be present in G_i (Remark 2). Therefore, pairs of Byzantine processes may potentially cause a reduction to the maximum size of $\Pi_{i,j}$: every couple may decrease the number of available disjoint paths in G_i between p_i and p_j by one. It follows that the size of $\Pi_{i,j}$ would be at most reduced to:

$$f + \lfloor f/2 \rfloor + 1 + f_d - \lfloor \mathbf{f_d}/\mathbf{2} \rfloor$$

namely, f_d declared Byzantine faulty processes may reduce the local connectivity between p and q in G_i by at most $\lfloor f_d/2 \rfloor$. The f_d declared Byzantine processes may also be selected in the paths $\Pi_{i,j}$. Specifically, in the worst case scenario f_d paths in $\Pi_{i,j}$ may contain Byzantine processes. It follows that at most f_d paths would not be fault-free, and thus the remaining fault-free ones in $\Pi_{i,j}$ would be:

$$f + \lfloor f/2 \rfloor + 1 + f_d - \lfloor f_d/2 \rfloor - \mathbf{f_d} = f + 1 + \lfloor f/2 \rfloor - \lfloor f_d/2 \rfloor$$

Thus, at least $f+1$ paths in $\Pi_{i,j}$ are faults-free, real and disjoint. □

Notice that, given Property 9, the Theorem 5 extends substituting local broadcast links with the unicast ones and assuming the known neighborhood assumption.

6 `CombinedRC`, Reliable Communication Protocol

We combine `Explorer`, `DolevU` and `DolevR` protocols to design a new reliable communication primitive. We call such a protocol `CombinedRC`, that aims to set up an efficient reliable communication service.

The `Explorer` protocol is used to partially reconstruct the network topology, and then to enable processes to compute disjoint paths solutions through which relay contents. The `DolevU` protocol is adopted as reliable communication subprimitive by `Explorer` and `CombinedRC` during the initialization. Lastly, the `DolevR` protocol is employed as actual reliable communication primitive in `CombinedRC`, leveraging the routes computed and communicated using `Explorer` and `DolevU`.

We showed in Section 5.2 that `Explorer`, under certain conditions, enables every correct process p_i to identify a disjoint paths solution $\Pi_{i,j}$ interconnecting it with any other correct process p_j, such that at least $f+1$ paths in the solution are faults-free, real and disjoint. Once that the solution $\Pi_{i,j}$ is known to both p_i and p_j, they can efficiently communicate. We claimed in Property 8 that all correct processes eventually obtain the same topology reconstruction in case local broadcast links are employed. Thus, under such an assumption, processes p_i and p_j eventually compute the same solution $\Pi_{i,j}$. Under the weaker condition of unicast links, the reconstructed topologies may differ on distinct processes, thus a source process p_i has additionally to communicate the computed solution $\Pi_{i,j}$ to a target process p_j using `DolevU`.

Any source process p_i routes contents through the computed $\Pi_{i,j}$ and any target process p_j waits for messages over $f+1$ paths among the ones in $\Pi_{i,j}$.

The pseudo-code of `CombinedRC` is presented in Algorithm 1.
Every process relays its contents over the computed routes if available, otherwise, they are queued for subsequent transmission (lines 1-5).
Every process p_i attempts to compute a solution $\Pi_{i,j}$ toward every other process p_j of the system. In the case of local broadcast links, the reconstructed topology G_i is eventually the same in every process. Therefore, a source process has to relay its contents over the computed disjoint routes every time they change (a finite number of times). In case of unicast links, once that the local connectivity toward a target p_j reaches a value greater than $2f$, the source process p_i communicates the computed solution $\Pi_{i,j}$ via `DolevU` (lines 6-21).
Every process relays contents or computed disjoint solution following the path attached to messages (lines 22-33).
Every process that delivers a disjoint paths solution with `DolevU` adopts it to verify contents (lines 34-35) using `DolevR` (lines 36-37).

6.1 `CombinedRC` Correctness

Theorem 6. *`CombinedRC` provides safety of reliable communication.*

Theorem 7. *`CombinedRC` provides liveness of reliable communication in all cases where `Explorer` succeeds in identifying a disjoint path solution between two processes i, j that contains $f+1$ FF_R_D paths.*

Proof. Let us assume that all messages exchanged by `Explorer` have been already delivered and that a process p_i aims to reliably communicate with a correct process p_j. In case of local broadcast links, processes p_i and p_j eventually share

Algorithm 1 `CombinedRC`

```
upon RC_send(m, target) do
    Sent ← Sent ∪ ⟨m, target⟩
    if Π_{i,target} ≠ ∅ then
        for path ∈ Π_{i,target} do
            send(⟨CNT, i, target, m, path⟩, path[1])

upon G_i changes do
    for j ∈ G_i such that i ≠ j do
        if LB then
            if local_conn(G_i, i, j) > f + ⌊f/2⌋ and disj_paths(G_i, i, j) ≠ |Π_{i,j}| then
                Π_{i,j} ← disj_paths(G_i, i, j)
                for path ∈ Π_{i,j} do
                    for ⟨m, target⟩ ∈ Sent such that j = target do
                        send(⟨CNT, i, j, m, path⟩, path[1])
                Π_{j,i} ← disj_paths(G_i, j, i)
        else if UC then
            if Π_{i,j} = ∅ and local_conn(G_i, i, j) > 2f then
                Π_{i,j} ← disj_paths(G_i, i, j)
                for path ∈ Π_{i,j} do
                    send(⟨ROU, i, j, Π_{i,j}, path⟩, path[1])
                    for ⟨m, target⟩ ∈ Sent such that j = target do
                        send(⟨CNT, i, j, m, path⟩, path[1])

upon receive(⟨CNT, s, t, m, path⟩, j) do
    if predecessor(path, i) = j then
        if t = i then
            Paths_cnt[⟨m, s⟩] ← Paths_cnt[⟨m, s⟩] ∪ {path}
        else
            send(⟨CNT, s, t, m, path⟩, successor(path, i))

upon receive(⟨ROU, s, t, Π, path⟩, j) do
    if predecessor(path, i) = j then
        if t = i then
            Paths_rou[⟨s, Π⟩] ← Paths_uRts[⟨s, Π⟩] ∪ {path}
        else
            send(⟨ROU, s, t, Π, path⟩, successor(path, i))

upon DolevU_deliver(Paths_rou[⟨s, Π⟩], s) do
    Π_{s,i} ← Π

upon DolevR_deliver(Paths_cnt[⟨m, s⟩], s) do
    RC_deliver(m, s)
```

the same topology reconstruction G_i, thus also the disjoint path solution $\Pi_{i,j}$ will eventually be the same both on p_i and p_j. Process p_i relays the contents through $\Pi_{i,j}$ every time such a solution changes. The assumption of $f+1$ FF_R_D paths in $\Pi_{i,j}$ guarantees reliable communication. In case of unicast channels, the solution $\Pi_{i,j}$ is diffused via `DolevU` and contents are routed over $\Pi_{i,j}$. The assumption of $f+1$ FF_R_D paths in $\Pi_{i,j}$ guarantees reliable communication. □

6.2 Protocol Complexity Analysis

`CombinedRC` provides reliable communication with optimal message complexity and delivery complexity (Theorem 1). Specifically, it routes contents over computed disjoint routes as `DolevR`, thus $\mathcal{O}(n)$ messages per content are exchanged, and an $\mathcal{O}(f)$ procedure is executed to verify any content.

`CombinedRC` requires an initialization phase where the network topology is partially reconstructed and the solutions containing $f+1$ FF_R_D paths are computed between every pair of correct processes. We showed in Section 5 that `Explorer` requires at most $\mathcal{O}(n^2)$ reliable communication instances to partially reconstruct the network topology. The same solution $\Pi_{i,j}$ is eventually computed by both p_i and p_j, assuming local broadcast channels, without additional message exchanges, because the topology reconstruction will eventually be the same on every process and the disjoint paths solutions can be computed through a deterministic algorithm. On the other hand, employing unicast links, every couple of processes has to agree on a solution $\Pi_{i,j}$. Thus, an additional content exchange (with payload $\Pi_{i,j}$) using a reliable communication primitive has to be performed for each pair of correct processes. It follows that the initialization phase of `CombinedRC` requires the execution of $\mathcal{O}(n^2)$ `DolevU` instances. Notice that, in the case of known neighborhood and local broadcast links, the cost of the initialization phase reduces to $\mathcal{O}(n)$ `DolevU` instances, indeed each process diffuses its neighborhood only once and all correct processes eventually share the same reconstruction.

7 Conclusion

We demonstrated how to boost the efficiency of reliable communication despite some of the participants being Byzantine faulty, when the network topology is unknown to the participants, assuming reliable authenticated links. Our solution combines a costly topology reconstruction process, that is executed once, and an efficient reliable communication scheme that is optimal both in terms of exchanged messages and of local computation complexity. Without leveraging the topology reconstruction, the cost of every reliable communication instance in the same scenario would have been factorial in message complexity and NP in delivery complexity.

An interesting path for future research is to decrease the adversary capabilities. A noteworthy candidate is the computationally bounded adversary, that enables solutions based on cryptography.

References

1. Bhandari, V., Vaidya, N.H.: Implementing a reliable local broadcast primitive in wireless ad hoc networks. https://disc.georgetown.domains/publications/rbcast-tech.pdf
2. Bhandari, V., Vaidya, N.H.: On reliable broadcast in a radio network. In: Aguilera, M.K., Aspnes, J. (eds.) Proceedings of the Twenty-Fourth Annual ACM Symposium on Principles of Distributed Computing, PODC 2005, Las Vegas, NV, USA, July 17-20, 2005. pp. 138–147. ACM (2005)
3. Bonomi, S., Farina, G., Tixeuil, S.: Boosting the Efficiency of Byzantine-tolerant Reliable Communication. Technical report, https://hal.archives-ouvertes.fr/hal-02960087
4. Bonomi, S., Farina, G., Tixeuil, S.: Multi-hop byzantine reliable broadcast with honest dealer made practical. J. Braz. Comp. Soc. **25**(1), 9:1–9:23 (2019)
5. Cachin, C., Guerraoui, R., Rodrigues, L.E.T.: Introduction to Reliable and Secure Distributed Programming (2. ed.). Springer (2011)
6. Diestel, R.: Graph Theory. Springer Berlin Heidelberg (2017)
7. Dinic, E.A.: Algorithm for solution of a problem of maximum flow in networks with power estimation. In: Soviet Math. Doklady. vol. 11, pp. 1277–1280 (1970)
8. Dolev, D.: Unanimity in an unknown and unreliable environment. In: 22nd Annual Symposium on Foundations of Computer Science, Nashville, Tennessee, USA, 28-30 October 1981. pp. 159–168 (1981)
9. Dolev, S., Liba, O., Schiller, E.M.: Self-stabilizing byzantine resilient topology discovery and message delivery - (extended abstract). In: Networked Systems - First International Conference, NETYS 2013, Marrakech, Morocco, May 2-4, 2013, Revised Selected Papers. pp. 42–57 (2013)
10. Edmonds, J., Karp, R.M.: Theoretical improvements in algorithmic efficiency for network flow problems. J. ACM **19**(2), 248–264 (1972)
11. Khan, M.S., Naqvi, S.S., Vaidya, N.H.: Exact byzantine consensus on undirected graphs under local broadcast model. In: Robinson, P., Ellen, F. (eds.) Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019. pp. 327–336. ACM (2019)
12. Maurer, A., Tixeuil, S.: Byzantine broadcast with fixed disjoint paths. J. Parallel Distrib. Comput. **74**(11), 3153–3160 (2014)
13. Maurer, A., Tixeuil, S.: Containing byzantine failures with control zones. IEEE Trans. Parallel Distrib. Syst. **26**(2), 362–370 (2015)
14. Maurer, A., Tixeuil, S.: Tolerating random byzantine failures in an unbounded network. Parallel Processing Letters **26**(1) (2016)
15. Menger, K.: Zur allgemeinen kurventheorie. Fundamenta Mathematicae **10**(1), 96–115 (1927)
16. Nesterenko, M., Tixeuil, S.: Discovering network topology in the presence of byzantine faults. IEEE Trans. Parallel Distrib. Syst. **20**(12), 1777–1789 (2009)
17. Pagourtzis, A., Panagiotakos, G., Sakavalas, D.: Reliable broadcast with respect to topology knowledge. Distributed Computing **30**(2), 87–102 (2017)
18. Pelc, A., Peleg, D.: Broadcasting with locally bounded byzantine faults. Inf. Process. Lett. **93**(3), 109–115 (2005)
19. Zeng, K., Govindan, K., Mohapatra, P.: Non-cryptographic authentication and identification in wireless networks. IEEE Wireless Commun. **17**(5), 56–62 (2010)