Reliable Communication in a Dynamic Network in the Presence of Byzantine Faults

Alexandre Maurer, Sébastien Tixeuil, Xavier Défago

Reliable Communication in a Dynamic Multihop Network in the Presence of Byzantine Faults

Alexandre Maurer[1], Xavier Defago[2], and Sébastien Tixeuil[1,3]
[1] Sorbonne Universités, UPMC Univ. Paris 06 - LIP6 CNRS UMR 7606
[2] Japan Advanced Institute of Science and Technology (JAIST)
[3] Institut Universitaire de France
E-mail: Alexandre.Maurer@lip6.fr, Defago@jaist.ac.jp, Sebastien.Tixeuil@lip6.fr,
Contact author's address: Laboratoire d'Informatique de Paris 6,
4 place Jussieu, 75005 Paris, France
Phone: +33 1 44 27 87 62, +33 1 44 27 87 75

February 11, 2014

Abstract

We consider the problem of transmitting information reliably from a source node to a sink node in a dynamic multihop network, in spite of the presence of Byzantine nodes. Byzantine nodes behave arbitrarily, and can tamper with messages or forward spurious ones. Previous work has shown that, in static multihop networks, reliable communication is possible in the presence of k Byzantine faults if and only if there are $2k+1$ node-disjoint paths from the source to the sink. However, this result relies on Menger's theorem (the equivalence between node cut and connectivity), which only holds in *static* networks.

In this paper, we prove a necessary and sufficient condition for reliable communication in *dynamic* networks, where the topology can vary over time and nodes can be subject to arbitrary Byzantine failures. The positive side of the condition is constructive, as we provide a Byzantine tolerant protocol for multihop communication in dynamic networks. Then, we assess the significance of this condition for several case studies (synthetic movements on agents, actual movements of participants interacting in a conference, movements based on the schedule of the Paris subway) and demonstrate the benefits of our protocol in various contexts.

1 Introduction

As modern networks grow larger, they become more likely to fail, sometimes in unforeseen ways. Indeed, nodes can be subject to crashes, attacks, transient bit flips, etc. Many failure and attack models have been proposed, but one of the most general is the *Byzantine* model proposed by Lamport et al. [16]. The model assumes that faulty nodes can behave arbitrarily. In this paper, we study the problem of reliable communication in a multihop network despite the presence of Byzantine faults. The problem proves difficult since even a single Byzantine node, if not neutralized, can lie to the entire network.

Related works. Many Byzantine-robust protocols are based on *cryptography* [5, 9]. In such protocols, nodes rely on a public key infrastructure and digital signatures to authenticate the sender across multiple hops. However, there are several drawbacks associated with the use of digital signatures. First, digital signatures schemes typically rely on asymmetric cryptography, which is notoriously prohibitive in terms of computation overhead. Second, this approach limits scalability since, in order to be able to verify a signature, every destination must ideally store (or securely get read access to) the public key of every potential source. Third, cryptography-based approaches relies on the assumption that cryptographic secrets of correct nodes have not been compromised, which implies limiting the strength of the Byzantine adversary. Fourth, key distribution generally requires a trusted infrastructure that may come up as a single point of failure / attacks, or as another weakening of the Byzantine adversary if considered safe.

In principle, cryptography-free protocols do not exhibit these shortcomings, but this generally comes at the price of a higher redundancy in communication infrastructure. Following the setting of the seminal paper of Lamport et al. [16], many subsequent papers focusing of Byzantine tolerance [1, 17, 18, 24] study agreement and reliable communication primitives using cryptography-free protocols in networks that are both *static* and *fully connected*. A recent exception to fully connected topologies in Byzantine agreement protocols is the recent work of Tseung, Vaidya and Liang [29, 30], which considers specific classes of *static* directed graphs (*i.e.*, graphs with a particularly high clustering coefficient) and considers *approximate* and *iterative* versions of the agreement problem.

In general multihop networks, two notable classes of algorithms use some locality property to tolerate Byzantine faults: space-local and time-local algorithms. Space-local algorithms [20, 25, 28] try to contain the fault (or its effect) as close to its source as possible. This is useful for problems where information from remote nodes is unimportant (such as vertex coloring, link coloring, or dining philosophers). Time-local algorithms [10, 11, 12, 13, 19] try to limit over time the effect of Byzantine faults. Time-local algorithms presented so far can tolerate the presence of at most a single Byzantine node, and are unable to mask the effect of Byzantine actions. Thus, neither approach is suitable to reliable communication.

In dense multihop networks, a first line of work assumes that there is a bound on the fraction of Byzantine nodes among the neighbors of each node. Protocols have been proposed for nodes organized on a lattice [2, 15], and later generalized to other topologies [27], with the assumption that each node knows the global topology. Since this approach requires all nodes to have a large degree, it may not be suitable for every multihop networks. The case of sparse networks was studied under the assumption that Byzantine failures occur uniformly at random [21, 23, 22], an assumption that holds, *e.g.*, in structured overlay networks where the identifier (a.k.a. position) of a new node joining the network is assigned randomly, but not necessarily in various actual communication networks.

Most related to our work is the line of research that assume the existence of $2k + 1$ node-disjoint paths from source to destination, in order to provide reliable communication in the presence of up to k Byzantine failure [7, 26, 8]. The initial solution [7] assumes that each node is aware of the global network topology, but this hypothesis was dropped in subsequent work [26]. Nevertheless, these results assume a *static* network: the topology remains the same for the duration of the entire execution of an instance of the protocol. This hypothesis is fundamental for Menger's theorem [3] to ensure equivalence between minimal node-cut and connectivity.

None of the aforementioned papers considers genuinely dynamic networks, *i.e.*, where the topology evolves while the protocol executes.

In this paper, our objective is to design cryptography-free communication protocols that can withstand Byzantine nodes that are arbitrarily located, in a strongly *dynamic* network [4], where only few communication channels may be available at any given time. The main obstacle to face is that Menger's theorem cannot be generalized to this dynamic setting [14]. A simple counterexample is given in Figure 1, where at least two nodes must be removed in order to disconnect the source from the sink. However, it is impossible to find two node-disjoint paths between the source and the sink. Therefore, one can only reason in terms of node cuts to expect solving this problem in dynamic networks.

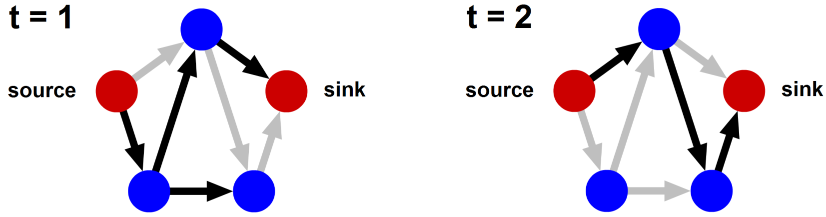

Figure 1: Counterexample to Menger's theorem in dynamic graphs. Black arrows represent arcs that are present at that time. Minimal node cut is 2, whereas connectivity is 1.

Our contribution. In this paper, we consider a dynamic multihop network that is subject to up to k Byzantine failures. We first prove necessary and sufficient conditions for reliable communication between two given nodes p and q (Theorem 1). Our characterization is based on a dynamic version of $\text{MinCut}(p,q)$ that takes into account both the presence of particular paths and their duration with respect to the delay that is necessary to actually transmit a message over a path. No such protocol can exist if $\text{MinCut}(p,q)$ is lower or equal to $2k$. We provide a protocol that is correct whenever $\text{MinCut}(p,q)$ is strictly greater than $2k$.

Another contribution is the application of our main Theorem to various case studies. Some are synthetic and are analytically described (robots moving on a grid) while others are based on actual data defining a dynamic network of interactions (participants interacting in a conference, user movements based on the Paris subway schedule). In both deterministic and probabilistic cases, we show that our solution enables an important performance and feasibility gain compared to the naive approach (waiting that the source meets the sink).

Organization of the paper. The paper is organized as follows. In Section 2, we present the model and give basic definitions. In Section 3, we describe our Byzantine-resilient broadcast protocol, then prove the necessary and sufficient condition for reliable communication. We present deterministic and probabilistic cases study in Section 4, both using synthetic toy networks and dynamic networks obtained from real world data. Section 5 concludes the paper.

2 Preliminaries

Network model We consider a continuous temporal domain $\mathbb{R}^+$ where dates are positive real numbers. We model the system as a time varying graph, as defined by Casteigts, Flocchini, Quattrociocchi and Santoro [4], where vertices represent the processes and edges represent the communication links (or channels). A time varying graph is a dynamic graph represented by a tuple $\mathcal{G} = (V, E, \rho, \zeta)$ where:

- V is the set of *nodes*.
- $E \subseteq V \times V$ is the set of *edges*.
- $\rho : E \times \mathbb{R}^+ \to \{0, 1\}$ is the *presence* function: $\rho(e,t) = 1$ indicates that edge e is present at date t.
- $\zeta : E \times \mathbb{R}^+ \to \mathbb{R}^+$ is the *latency* function: $\zeta(e,t) = T$ indicates that a message sent at date t takes T time units to cross edge e.

The discrete time model is a special case, where time and latency are restricted to integer values.

Hypotheses We make the same hypotheses as previous work on the subject [2, 7, 15, 21, 22, 23, 26, 27]. First, each node has a unique identifier. Then, we assume *authenticated channels* (or *oral model*), that is, when a node q receives a message through channel (p, q), it knows the identity of p. Now, an omniscient adversary can select up to k nodes as *Byzantine*. These nodes can have a totally arbitrary and unpredictable behavior defined by the adversary (including tampering or dropping messages, or simply crashing). Finally, other nodes are *correct* and behave as specified by the algorithm. Of course, correct nodes are unable to know *a priori* which nodes are Byzantine. We also assume that a correct node u is aware of its *local topology* at any given date t (that is, u knows the set of nodes v such that $\rho((u, v), t) = 1$).

Dynamicity-related definitions Informally, a *dynamic path* is a sequence of nodes a message can traverse, with respect to network dynamicity and latency.

Definition 1 (Dynamic path). *A sequence of distinct nodes* $(u_1, \ldots, u_n)$ *is a* dynamic path *from* u_1 *to* u_n *if and only if there exists a sequence of dates* $(t_1, \ldots, t_n)$ *such that,* $\forall i \in \{1, \ldots, n-1\}$ *we have:*

- $e_i = (u_i, u_{i+1}) \in V$*: there exists an edge connecting* u_i *to* u_{i+1}*.*
- $\forall t \in [t_i, t_i + \zeta(e_i, t_i)]$, $\rho(e_i, t) = 1$*:* u_i *can send a message to* u_{i+1} *at date* t_i*.*
- $\zeta(e_i, t_i) \leq t_{i+1} - t_i$*: the aforementioned message is received by date* t_{i+1}*.*

We now define the *dynamic minimal cut* between two nodes p and q as the minimal number of nodes (besides p and q) one has to remove from the network to prevent the existence of a dynamic path between p and q. Formally:

- Let $Dyn\,(p, q)$ be the multiset of node sets $\{u_1, \ldots, u_n\}$ such that $(p, u_1, \ldots, u_n, q)$ is a dynamic path.
- For a multiset of node sets $\Omega = \{S_1, \ldots, S_n\}$, let $Cut\,(\Omega)$ be a node set C such that, $\forall i \in \{1, \ldots, n\}$, $C \cap S_i \neq \emptyset$ (C contains at least one node from each set S_i).
- Let $MinCut\,(\Omega) = \min_{C \in Cut(\Omega)} card(C)$ (the size of the smallest element of $Cut\,(\Omega)$). If $Cut(\Omega)$ is empty, we assume that $MinCut\,(\Omega) = +\infty$.

Definition 2 (Dynamic minimal cut). *The set* $MinCut\,(p, q) = MinCut\,(Dyn\,(p, q))$ *is a* dynamic minimal cut *between two nodes* p *and* q*.*

Problem specification We say that a node *multicasts* a message m when it sends m to all nodes in its current local topology. Now, a node u *accepts* a message m from another node v when it considers that v is the author of this message. We now define our problem specification, that is, *reliable* communication.

Definition 3 (Reliable communication). *Let* p *and* q *be two correct nodes. An algorithm ensures* reliable communication *from* p *to* q *when the following two conditions are satisfied:*

- *When* q *accepts a message from* p*,* p *is necessarily the author of this message.*
- *When* p *sends a message,* q *eventually receives and accepts this message from* p*.*

3 Our Algorithm

In this section, we describe our Byzantine-resilient multihop broadcast protocol. This algorithm is used as a constructive proof for the sufficient condition for reliable communication. We then prove the necessary and sufficient condition for reliable communication.

Informal description Consider that each correct node p wants to broadcast a message m_0 to the rest of the network. Let us first discuss why the naive flood-based solution fails. A naive first idea would be to send a tuple (p, m_0) through all possible dynamic paths: thus, each node receiving m_0 knows that p broadcast m_0. Yet, Byzantine nodes may forward false messages, *e.g.*, a Byzantine node could forward the tuple (p, m_1), with $m_1 \neq m_0$, to make the rest of the network believe that p broadcast m_1.

To prevent correct nodes from accepting false message, we attach to each message the set of nodes that have been visited by this message since it was sent (that is, we use (p, m, S), where S is a set of nodes already visited by m since p sent it). Now, a correct node accepts a message when it has been received through a collection of dynamic paths that cannot be cut by k nodes (where k is a parameter of the algorithm, and supposed to be an upper bound on the total number of Byzantine nodes in the network). The fact that this simple scheme enables reliable communication is demonstrated in Lemma 4. This scheme shares the same underlying principle of that of Nesterenko and Tixeuil [26] for static graphs (traversed paths are attached to messages, and correct nodes accept a message when this message has been received by at least $k+1$ node-disjoint paths) but is now robust to high network dynamicity while the algorithm is operating due to focusing on dynamic minimal cut computations *vs.* node-disjoint path computations (whose equivalence is only valid in static networks).

Variables Each correct node u maintains the following variables:

- $u.m_0$, the message that u wants to broadcast.
- $u.\Omega$, a dynamic set registering all tuples (s, m, S) received by u.
- $u.Acc$, a dynamic set of confirmed tuples (s, m). We assume that whenever $(s, m) \in u.Acc$, u accepts m from s.

Initially, $u.\Omega = \{(u, u.m_0, \emptyset)\}$ and $u.Acc = \{(u, u.m_0)\}$.

Algorithm Each correct node u obeys the following rules:

1. Initially, and whenever $u.\Omega$ or the local topology of u change: multicast $u.\Omega$.
2. Upon reception of Ω' through channel (v, u): $\forall (s, m, S) \in \Omega'$, if $v \notin S$ then append $(s, m, S \cup \{v\})$ to $u.\Omega$.
3. Whenever there exist s, m and $\{S_1, \ldots, S_n\}$ such that $\forall i \in \{1, \ldots, n\}$, $(s, m, S_i \cup \{s\}) \in u.\Omega$ and $MinCut(\{S_1, \ldots, S_n\}) > k$: append (s, m) to $u.Acc$.

Main theorem Let us consider a given dynamic graph, and two given correct nodes p and q. Our main result is as follows:

Theorem 1. *A k-Byzantine tolerant reliable communication from p to q is feasible if and only if $MinCut(p, q) > 2k$.*

Proof. See Lemma 1 and 4. □

Lemma 1 (Necessary condition)**.** *Let us suppose that there exists an algorithm ensuring reliable communication from p to q. Then, we necessarily have $MinCut(p, q) > 2k$.*

Proof. Let us suppose the opposite: there exists an algorithm ensuring reliable communication from p to q, and yet, $MinCut(p,q) \leq 2k$. Let us show that it leads to a contradiction.

As we have $MinCut(p,q) = MinCut(Dyn(p,q)) \leq 2k$ and $MinCut(Dyn(p,q)) = \min_{C \in Cut(Dyn(p,q))} card(C)$, there exists an element C of $Cut(Dyn(p,q))$ such that $card(C) \leq 2k$. Let C_1 be a subset of C containing k' elements, with $k' = \min(k, card(C))$. Let $C_2 = C - C_1$. Thus, we have $card(C_1) \leq k$ and $card(C_2) \leq k$.

According to the definition of $Cut(Dyn(p,q))$, C contains a node of each possible dynamic path from p to q. Therefore, the information that q receives about p are completely determined by the behavior of the nodes in C.

Let us consider two possible placements of Byzantine nodes, and show that they lead to a contradiction:

- First, suppose that all nodes in C_1 are Byzantine, and that all other nodes are correct. This is possible since $card(C_1) \leq k$.

  Suppose now that p broadcasts a message m. Then, according to our hypothesis, since the algorithm ensures reliable communication, q eventually accepts m from p, regardless of what the behavior of the nodes in C_1 may be.

- Now, suppose that all nodes in C_2 are Byzantine, and that all other nodes are correct. This is also possible since $card(C_2) \leq k$.

  Then, suppose that p broadcasts a message $m' \neq m$, and that the Byzantine nodes have exactly the same behavior as the nodes of C_2 had in the previous case.

  Thus, as the information that q receives about p is completely determined by the behavior of the nodes of C, from the point of view of q, this situation is indistinguishable from the previous one: the nodes of C_2 have the same behavior, and the behavior of the nodes of C_1 is unimportant. Thus, similarly to the previous case, q eventually accepts m from p.

Therefore, in the second situation, p broadcasts m, and q eventually accepts $m' \neq m$. Thus, according to Definition 3, the algorithm does not ensure reliable communication, which contradicts our initial hypothesis. Hence, the result. □

Lemma 2 (Safety)**.** *Let us suppose that all correct nodes follow our algorithm. If $(p,m) \in q.Acc$, then $m = p.m_0$.*

Proof. As $(p,m) \in q.Acc$, according to rule 3 of our algorithm, there exists $\{S_1, \dots, S_n\}$ such that, $\forall i \in \{1, \dots, n\}$, $(p, m, S_i \cup \{p\}) \in q.\Omega$, and $MinCut(\{S_1, \dots, S_n\}) > k$.

Suppose that each node set $S \in \{S_1, \dots, S_n\}$ contains at least one Byzantine node. If C is the set of Byzantine nodes, then $C \in Cut(\{S_1, \dots, S_n\})$ and $card(C) \leq k$. This is impossible because $MinCut(\{S_1, \dots, S_n\}) > k$. Therefore, there exists $S \in \{S_1, \dots, S_n\}$ such that S does not contain any Byzantine node.

Now, let us use the correct dynamic path corresponding to S to show that $m = m_0$. Let $n' = card(S \cup \{p\})$. Let us show the following property $\mathcal{P}_i$ by induction, $\forall i \in \{0, \dots, n'\}$: there exists a correct node u_i and a set of correct nodes X_i such that $(p, m, X_i) \in u_i.\Omega$ and $card(X_i) = card(S \cup \{p\}) - i$.

- As $S \in \{S_1, \dots, S_n\}$, $(p, m, S \cup \{p\}) \in q.\Omega$. Thus, $\mathcal{P}_0$ is true if we take $u_0 = q$ and $X_0 = S \cup \{p\}$.

- Let us now suppose that $\mathcal{P}_{i+1}$ is true, for $i < n'$. As $(p, m, X_i) \in u_i.\Omega$, according to rule 2 of our algorithm, it implies that u_i received Ω' from a node v, with $(p, m, X) \in \Omega'$, $v \notin X$ and $X_i = X \cup \{v\}$. Thus, $card(X) = card(X_i) - 1 = card(S \cup \{p\}) - (i+1)$.

As $v \in X_i$ and X_i is a set of correct nodes, v is correct and behaves according to our algorithm. Then, as v sent Ω', according to rule 1 of our algorithm, we necessarily have $\Omega' \subseteq v.\Omega$. Thus, as $(p, m, X) \in \Omega'$, we have $(p, m, X) \in v.\Omega$. Hence, $\mathcal{P}_{i+1}$ is true if we take $u_{i+1} = v$ and $X_{i+1} = X$.

By induction principle, $\mathcal{P}_{n'}$ is true. As $card(X_{n'}) = 0$, $X_{n'} = \emptyset$ and $(p, m, \emptyset) \in u_{n'}$. As $u_{n'}$ is a correct node and follows our algorithm, the only possibility to have $(p, m, \emptyset) \in u_{n'}.\Omega$ is that $u_{n'} = p$ and $m = p.m_0$. Thus, the result. □

Lemma 3 (Communication)**.** *Let us suppose that $MinCut\,(p, q) > 2k$, and that all correct nodes follow our algorithm. Then, we eventually have $(p, p.m_0) \in q.Acc$.*

Proof. Let $\{S_1, \dots, S_n\}$ be the set of node sets $S \in Dyn\,(p, q)$ that only contain correct nodes. Similarly, let $\{X_1, \dots, X_{n'}\}$ be the set of node sets $X \in Dyn\,(p, q)$ that contain at least one Byzantine node.

Let us suppose that $MinCut\,(\{S_1, \dots, S_n\}) \leq k$. Then, there exists $C \in Cut\,(\{S_1, \dots, S_n\})$ such that $card(C) \leq k$. Let C' be the set containing the nodes of C and the Byzantine nodes. Thus, and $C' \in Cut\,(\{S_1, \dots, S_n\} \cup \{X_1, \dots, X_{n'}\}) = Cut\,(Dyn\,(p, q))$, and $card(C') \leq 2k$. Thus, $MinCut\,(Dyn\,(p, q)) \leq 2k$, which contradicts our hypothesis. Therefore, $MinCut\,(\{S_1, \dots, S_n\}) > k$.

In the following, we show that $\forall S \in \{S_1, \dots, S_n\}$, we eventually have $(p, p.m_0, S \cup \{p\}) \in q.\Omega$, ensuring that q eventually accepts $p.m_0$ from p.

Let $S \in \{S_1, \dots, S_n\}$. As $S \in Dyn\,(p, q)$, let $(u_1, \dots, u_N)$ be the dynamic path such that $p = u_1$, $q = u_N$ and $S = \{u_2, \dots, u_{N-1}\}$. Let $(t_1, \dots, t_N)$ be the corresponding dates, according to Definition 1. Let us show the following property $\mathcal{P}_i$ by induction, $\forall i \in \{1, \dots, N\}$: at date t_i, $(p, p.m_0, X_i) \in u_i.\Omega$, with $X_i = \emptyset$ if $i = 1$ and $\{u_1, \dots, u_{i-1}\}$ otherwise.

- $\mathcal{P}_1$ is true, as we initially have $(p, p.m_0, \emptyset) \in p.\Omega$.
- Let us suppose that $\mathcal{P}_i$ is true, for $i < N$. According to Definition 1, $\forall t \in [t_i, t_i + \zeta(t_i, u_i)]$, $\rho(e_i, t) = 1$, e_i being the edge connecting u_i to u_{i+1}.
  - Let $t_A \leq t_i$ be the earliest date such that, $\forall t \in [t_A, t_i + \zeta(t_i, u_i)]$, $\rho(e_i, t) = 1$.
  - Let $t_B \leq t_i$ be the date where (p, m, X_i) is added to $u_i.\Omega$.
  - Let $t_C = max(t_A, t_B)$.

  Then, at date t_C, either $u_i.\Omega$ or the local topology topology of u_i changes. Thus, according to rule 1 of our algorithm, u_i multicasts $\Omega' = u_i.\Omega$ at date t_C, with $(p, p.m_0, X_i) \in \Omega'$.

  As $\zeta(e_i, t_i) \leq t_{i+1} - t_i \leq t_{i+1} - t_C$, u_{i+1} receives Ω' from u_i at date $t_C + \zeta(e_i, t_i) \leq t_{i+1}$. Then, according to rule 2 of our algorithm, $(p, p.m_0, X_i \cup \{u_i\})$ is added to $u_{i+1}.\Omega$.

  Thus, $\mathcal{P}_{i+1}$ is true if we take $X_{i+1} = X_i \cup \{u_i\}$.

By induction principle, $\mathcal{P}_N$ is true. As $u_1 = p$, $X_N = \{u_1, \dots, u_{N-1}\} = S \cup \{p\}$, and we eventually have $(p, p.m_0, S \cup \{p\}) \in q.\Omega$.

Thus, $\forall S \in \{S_1, \dots, S_n\}$, we eventually have $(p, p.m_0, S \cup \{p\}) \in q.\Omega$. Then, as $MinCut\,(\{S_1, \dots, S_n\}) > k$, according to rule 3 of our algorithm, $(p, p.m_0)$ is added to $q.Acc$. □

Lemma 4 (Sufficient condition)**.** *Let p and q be two correct nodes, and k denote the maximum number of Byzantien nodes. If $MinCut\,(p, q) > 2k$, our algorithm ensures reliable communication from p to q.*

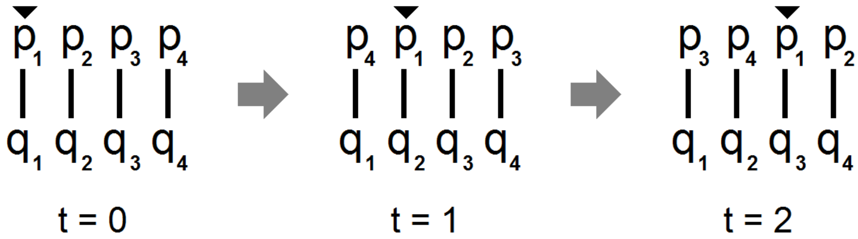

Figure 2: Case study: a deterministic dynamic toy network $\mathcal{T}_4$

Proof. Let us suppose that the correct nodes follow our algorithm, as described in Section 3. First, according to Lemma 2, if $(p, m) \in q.Acc$, then $m = p.m_0$. Thus, when q accepts a message from p, p is necessarily the author of this message. Then, according to Lemma 3, we eventually have $(p, p.m_0) \in q.Acc$. Thus, q eventually receives and accepts the message broadcast by p. Therefore, according to Definition 3, our algorithm ensures reliable communication from p to q. □

4 Case Studies

4.1 A deterministic dynamic toy network

Let $n > 0$, and let $(p_1, \ldots, p_n)$ and $(q_1, \ldots, q_n)$ be two sequences of nodes. We consider the dynamic network $\mathcal{T}_n$ where, at date $t \in \{0, 1, 2, \ldots\}$, p_i is connected to $q_{i+t \mod n}$. This is illustrated in Figure 2. Using our main theorem (Theorem 1), we are able to exactly characterize the Byzantine resilience of $\mathcal{T}_n$.

Theorem 2. *In $\mathcal{T}_n$, to ensure reliable communication between any two pairs of correct nodes, it is necessary and sufficient that $n > 2k$ and $t \geq 2k + n - 1$, where k denote the maximum number of Byzantine nodes in the network.*

Proof. Let $P = \{p_1, \ldots, p_n\}$ and $Q = \{q_1, \ldots, q_n\}$. Let u and v be two nodes.

- If $u \in P$ and $v \in Q$, let i and d be such that $u = p_i$ and $v = q_{i+d \mod n}$. Thus, $MinCut(u, v) = 0$ if $t < d$, and $+\infty$ otherwise. The same holds if $u \in Q$ and $v \in P$ (by symmetry).
- If $u \in Q$ and $v \in Q$, let i and d be such that $u = q_i$ and $v = q_{i+d \mod n}$. Thus, $MinCut(u, v) = 0$ if $t < d$, and $min(t-d+1, n)$ otherwise. The same holds if $u \in P$ and $v \in P$ (by symmetry).

Thus, as the maximal value of d is $n - 1$ (*e.g.*, when $u = q_1$ and $v = q_n$), $m = min_{(u,v)\in V\times V}$ $MinCut(u, v) = 0$ if $t < n - 1$, and $min(t - n + 2, n)$ otherwise. Now, according to Theorem 1, $m > 2k$ is necessary and sufficient to enable reliable communication between any pair of correct nodes.

First, let us show that the condition of Theorem 2 is necessary. Let us suppose the opposite: $n \leq 2k$ or $t < 2k + n - 1$, and $m > 2k$. Then, if $n \leq 2k$, as $m \leq n$, we get $m \leq 2k$: a contradiction. If $t < 2k + n - 1$ and $k = 0$, then $t < n - 1$ and $m = 0$: a contradiction. Hence, the condition is necessary.

Then, let us show that the condition of Theorem 2 is sufficient. As $t \geq 2k + n - 1 \geq n - 1$, we have $m = min(t - n + 2, n)$. Besides, as $t \geq 2k + n - 1$, it follows that $t - n + 2 \geq 2k$. Thus, as $n > 2k$, we have $m > 2k$, and the condition is sufficient. □

In particular, with $t = 2n$, we can tolerate roughly one fourth of Byzantine nodes.

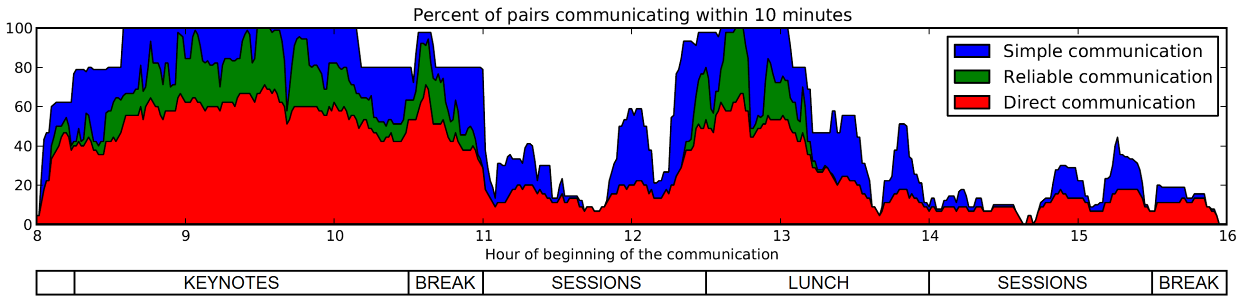

Figure 3: Reliable communication between 10 most sociable nodes of the Infocom 2005 dataset

4.2 A real-life dynamic network: the Infocom 2005 dataset

In this section, we consider the Infocom 2005 dataset [6], which is obtained in a conference scenario by iMotes capturing contacts between participants. This dataset can represent a dynamic network where each participant is a node and where each contact is a (temporal) edge. We consider an 8-hour period during the second day of the conference. In this period, we consider the dynamic network formed by the 10 most "sociable" nodes (our criteria of sociability is the total number of contacts reported). We assume that at most one on these nodes may be Byzantine (that is, $k = 1$).

Let p and q be two correct nodes. Let us suppose that p wants to transmit a message to q within a period of 10 minutes. After 10 minutes, three types of communication can be achieved:

- *Simple communication*: there exists a dynamic path from p to q.
- *Reliable communication*: the condition for reliable communication from p to q identified in Theorem 1 is satisfied.
- *Direct communication*: p meets q directly.

If we want to ensure reliable communication despite one Byzantine node, the simplest strategy is to wait until p meets q directly. Let us show now that relaying the message (*e.g.* using our algorithm as presented in Section 3) is usually beneficial and that our approach realizes a significant gain of performance.

Figure 3 represents the percentage of pairs of nodes (p, q) that communicate within 10 minutes, according to the date of beginning of the communication. We can correlate the peaks with the program of the conference: the first period corresponds to morning arrivals during the keynotes; the peak between 10:30 and 11:00 corresponds to the morning break; the peak starting at 12:30 corresponds to the end of parallel sessions and the departure for lunch. As it turns out, many pairs of nodes are able to communicate reliably, even though they are unable to meet directly. For instance, at 9:30, all pairs of nodes are effectively able to reliably exchange information, even though only two thirds of them come into direct contact. This means that relaying the information is actually effective and desirable.

4.3 Probabilistic mobile robots on a grid

We consider a network of 10 mobile robots that are initially randomly scattered on a square grid.

Definition 4 (Grid). *An* $N \times N$ grid *is a topology such that:*

- *Each vertex has a unique identifier* (i, j)*, with* $1 \leq i \leq N$ *and* $1 \leq j \leq N$*.*
- *Two vertices* (i_1, j_1) *and* (i_2, j_2) *are neighbors if and only if:* $|j_1 - j_2| + |i_1 - i_2| = 1$

At each time unit, a robot randomly moves to a neighbor vertex, or does not move (the new position is chosen uniformly at random among all possible choices). Let $position(u,t)$ be the current vertex of the robot u at date t. We consider that two robots can communicate if and only if they are on the same vertex. Our setting induces the following dynamic graph $\mathcal{G} = (V, E, \rho, \zeta)$: $V = \{u_1, \ldots, u_{10}\}$, $E = V \times V$, $\rho((u,v),t) = 1$ when $position(u,t) = position(v,t)$ and $\zeta((u,v),t) = 0$.

Let p and q be two correct robots, and suppose that up to k other robots are Byzantine. We aim at evaluating the *communication time*, that is: the mean time to have $MinCut\,(p,q) > 2k$ (Our condition for reliable communication established in Theorem 1). For this purpose, we ran more than 10000 simulations, and represented the results on Figure 4, 5, 6 and 7. Let us comment on these results.

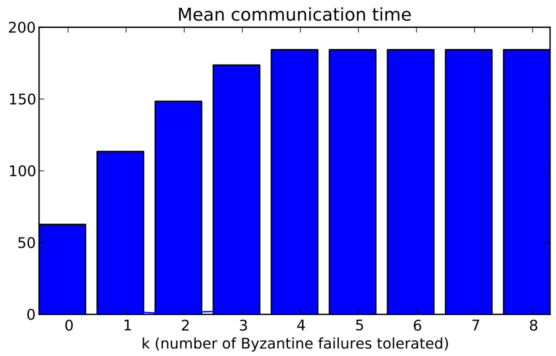

Figure 4: Mean communication time (10×10 grid)

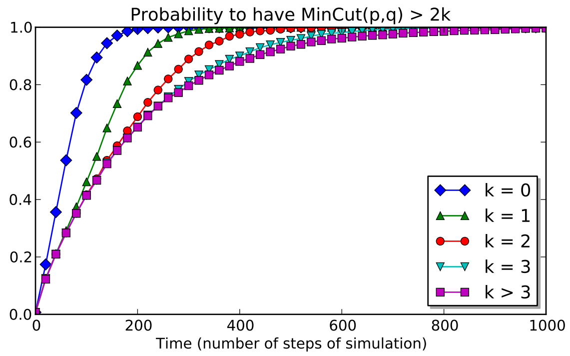

Figure 5: Probability to satisfy our condition for reliable communication (10×10 grid)

First, Figure 4 represents the mean communication time on a 10×10 grid, for all possible values of k. The time first increases with k, then stabilizes for $k > 3$. Indeed, for $k > 3$, due to the number of robots, the condition $MinCut\,(p,q) > 2k$ is satisfied if and only if p and q are on the same vertex: reliable *multihop* communication is impossible and only source to destination *direct* communication is feasible.

Then, Figure 5 represents the cumulative probability to satisfy our reliable communication condition on a 10×10 grid, with respect to time. As expected, this probability decreases when k increases. We also notice that this probability increases linearly at first with respect to time.

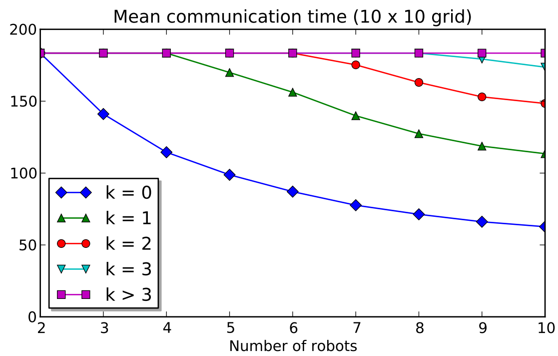

Figure 6: Mean communication time depending on the number of robots

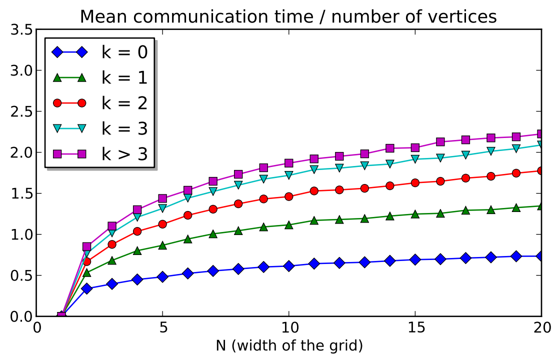

Figure 7: Mean communication time divided by the number of vertices

Also, Figure 6 represents the mean communication time according to the number of robots. With only 2 robots, we must wait for the source to meet the sink directly. However, when the number of robots increases, reliable multihop communication becomes increasingly more interesting. Also, we notice that, for every two robots that we add, it becomes possible to tolerate one more Byzantine

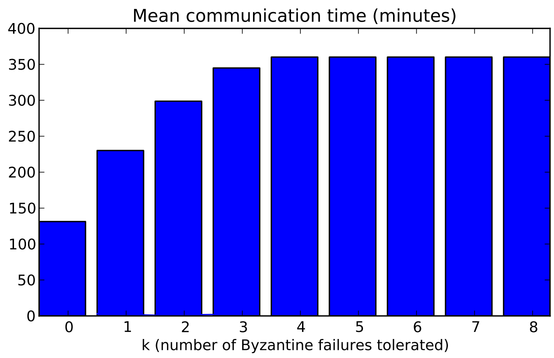

Figure 8: Mean communication time (subway)

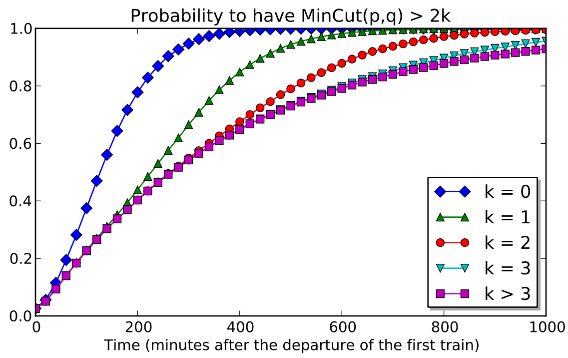

Figure 9: Probability to satisfy the condition for reliable communication (subway)

fault in multihop communication. This illustrates the condition $MinCut\,(p,q) > 2k$.

Finally, we study the influence of the size of the grid. We observe that the mean communication is roughly proportional to the number of vertices in the grid (that is, N^2 for a grid of width N). Figure 7 represents the ratio between the communication time and the number of vertices. This value seems to converge, or at least to increase very slowly with the size of the grid.

As we can see, the reliable multihop communication approach can be an interesting compromise. For instance, let us consider a 10×10 grid. The basic communication time is 63 time units. Now, let us suppose that we want to tolerate one Byzantine failure. If we wait for the source to meet directly with the sink, the mean communication times increases by 194% from the fault-free case. If we use our algorithm instead, it increases by only 81%.

4.4 Mobile agents in the Paris subway

We consider a dynamic network consisting of 10 mobile agents randomly moving in the Paris subway. The agents can use the classical subway lines (we exclude tramways and regional trains). Each agent is initially located at a randomly chosen junction station – that is, a station that connects at least two lines. Then, the agent randomly choses a neighbor junction station, waits for the next train, moves to this station, and repeats the process. We use the train schedule provided by the local subway company (`http://data.ratp.fr`). The time is given in minutes from the departure of the first train (*i.e.*, around 5:30). We consider that two agents can communicate in the two following cases:

1. They are staying together at the same station.

2. They cross each other in trains. For instance, if at a given time, one agent is in a train moving from station A to station B while the other agent moves from B to A, then we consider that they can communicate.

We provide the same plots as in 4.3: the mean communication time (see Figure 8) and the probability to satisfy the condition for reliable communication (see Figure 9). The results are very similar to those of 4.3, which suggests that the topology used for the simulations has only a minor qualitative influence.

The basic communication time is 131 minutes. Again, let us suppose that we want to tolerate one Byzantine failure. If we wait for the source to meet the sink directly, the mean communication time increases by 174%. If we use our algorithm, it increases only by 75%, which shows that there is a clear benefit in terms of latency.

5 Conclusion

In this paper, we gave a necessary and sufficient condition for reliable communication in a dynamic network that is subject to Byzantine failures. The sufficiency part of our condition is constructive, as we provide an algorithm for optimally broadcasting a message in this context (with respect to the number of Byzantine nodes tolerated). We demonstrated the benefits of this protocol in several case studies, both in synthetic example and in real dynamic networks.

Our result implicitly considers a worst-case placement of the Byzantine nodes, which is the classical approach when studying Byzantine failures in a distributed setting. Studying variants of the Byzantine node placement, and the associated necessary and sufficient condition for enabling multihop reliable communication, constitutes a interesting path for future research.

References

[1] H. Attiya and J. Welch. *Distributed Computing: Fundamentals, Simulations, and Advanced Topics*. McGraw-Hill Publishing Company, New York, May 1998. 6.

[2] Vartika Bhandari and Nitin H. Vaidya. On reliable broadcast in a radio network. In Marcos Kawazoe Aguilera and James Aspnes, editors, *PODC*, pages 138–147. ACM, 2005.

[3] T. Böhme, F. Göring, and J. Harant. Menger's theorem. *Journal of Graph Theory*, 37(1):35–36, 2001.

[4] Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408, 2012.

[5] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *OSDI*, pages 173–186, 1999.

[6] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on opportunistic forwarding algorithms. *TMC*, 6(6):606–620, 2007.

[7] D. Dolev. The Byzantine generals strike again. *Journal of Algorithms*, 3(1):14–30, 1982.

[8] Danny Dolev, Cynthia Dwork, Orli Waarts, and Moti Yung. Perfectly secure message transmission. *J. ACM*, 40, January 1993.

[9] Vadim Drabkin, Roy Friedman, and Marc Segal. Efficient byzantine broadcast in wireless ad-hoc networks. In *DSN*, pages 160–169. IEEE Computer Society, 2005.

[10] Swan Dubois, Toshimitsu Masuzawa, and Sébastien Tixeuil. The impact of topology on byzantine containment in stabilization. In *Proceedings of DISC 2010*, Lecture Notes in Computer Science, Boston, Massachusetts, USA, September 2010. Springer Berlin / Heidelberg.

[11] Swan Dubois, Toshimitsu Masuzawa, and Sébastien Tixeuil. On byzantine containment properties of the min+1 protocol. In *Proceedings of SSS 2010*, Lecture Notes in Computer Science, New York, NY, USA, September 2010. Springer Berlin / Heidelberg.

[12] Swan Dubois, Toshimitsu Masuzawa, and Sébastien Tixeuil. Bounding the impact of unbounded attacks in stabilization. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 2011.

[13] Swan Dubois, Toshimitsu Masuzawa, and Sébastien Tixeuil. Maximum metric spanning tree made byzantine tolerant. In David Peleg, editor, *Proceedings of DISC 2011*, Lecture Notes in Computer Science (LNCS), Rome, Italy, September 2011. Springer Berlin / Heidelberg.

[14] David Kempe, Jon Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. *Journal of Computer and System Sciences*, 64(4):820–842, 2002.

[15] Chiu-Yuen Koo. Broadcast in radio networks tolerating byzantine adversarial behavior. In Soma Chaudhuri and Shay Kutten, editors, *PODC*, pages 275–282. ACM, 2004.

[16] Leslie Lamport, Robert E. Shostak, and Marshall C. Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.

[17] D. Malkhi, Y. Mansour, and M.K. Reiter. Diffusion without false rumors: on propagating updates in a Byzantine environment. *Theoretical Computer Science*, 299(1–3):289–306, April 2003.

[18] D. Malkhi, M. Reiter, O. Rodeh, and Y. Sella. Efficient update diffusion in byzantine environments. In *The 20th IEEE Symposium on Reliable Distributed Systems (SRDS '01)*, pages 90–98, Washington - Brussels - Tokyo, October 2001. IEEE.

[19] Toshimitsu Masuzawa and Sébastien Tixeuil. Bounding the impact of unbounded attacks in stabilization. In Ajoy Kumar Datta and Maria Gradinariu, editors, *SSS*, volume 4280 of *Lecture Notes in Computer Science*, pages 440–453. Springer, 2006.

[20] Toshimitsu Masuzawa and Sébastien Tixeuil. Stabilizing link-coloration of arbitrary networks with unbounded byzantine faults. *International Journal of Principles and Applications of Information Science and Technology (PAIST)*, 1(1):1–13, December 2007.

[21] Alexandre Maurer and Sébastien Tixeuil. Limiting byzantine influence in multihop asynchronous networks. In *Proceedings of the 32nd IEEE International Conference on Distributed Computing Systems (ICDCS 2012)*, pages 183–192, June 2012.

[22] Alexandre Maurer and Sébastien Tixeuil. On byzantine broadcast in loosely connected networks. In *Proceedings of the 26th International Symposium on Distributed Computing (DISC 2012)*, volume 7611 of *Lecture Notes in Computer Science*, pages 183–192. Springer, 2012.

[23] Alexandre Maurer and Sébastien Tixeuil. A scalable byzantine grid. In *Proceedings of the 14th International Conference on Distributed Computing and Networking (ICDCN 2013)*, volume 7730 of *Lecture Notes in Computer Science*, pages 87–101. Springer, 2013.

[24] Y. Minsky and F.B. Schneider. Tolerating malicious gossip. *Distributed Computing*, 16(1):49–68, 2003.

[25] Mikhail Nesterenko and Anish Arora. Tolerance to unbounded byzantine faults. In *21st Symposium on Reliable Distributed Systems (SRDS 2002)*, pages 22–29. IEEE Computer Society, 2002.

[26] Mikhail Nesterenko and Sébastien Tixeuil. Discovering network topology in the presence of byzantine faults. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 20(12):1777–1789, December 2009.

[27] Andrzej Pelc and David Peleg. Broadcasting with locally bounded byzantine faults. *Inf. Process. Lett.*, 93(3):109–115, 2005.

[28] Yusuke Sakurai, Fukuhito Ooshita, and Toshimitsu Masuzawa. A self-stabilizing link-coloring protocol resilient to byzantine faults in tree networks. In *Principles of Distributed Systems, 8th International Conference, OPODIS 2004*, volume 3544 of *Lecture Notes in Computer Science*, pages 283–298. Springer, 2005.

[29] Lewis Tseng and Nitin H. Vaidya. Iterative approximate Byzantine consensus under a generalized fault model. In *Distributed Computing and Networking, 14th International Conference, ICDCN 2013*, pages 72–86, January 2013.

[30] Nitin H. Vaidya, Lewis Tseng, and Guanfeng Liang. Iterative approximate Byzantine consensus in arbitrary directed graphs. In *Proc. ACM Symp. on Principles of Distributed Computing, PODC'12*, pages 365–374, July 2012.