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

Alexandre Maurer[1], Sébastien Tixeuil[2,3] and Xavier Defago[3]
[1] École Polytechnique Fédérale de Lausanne
[2] Sorbonne Universités, UPMC Univ. Paris 06, LIP6 CNRS UMR 7606
[3] Institut Universitaire de France
[4] Japan Advanced Institute of Science and Technology (JAIST)
E-mail: Alexandre.Maurer@epfl.ch, Sebastien.Tixeuil@lip6.fr, Defago@jaist.ac.jp

February 17, 2015

Abstract

We consider the following problem: two nodes want to reliably communicate in a dynamic multihop network where some nodes have been compromised, and may have a totally arbitrary and unpredictable behavior. These nodes are called *Byzantine*. We consider the two cases where cryptography is available and not available.

We prove the necessary and sufficient condition (that is, the weakest possible condition) to ensure reliable communication in this context. Our proof is constructive, as we provide Byzantine-resilient algorithms for reliable communication that are optimal with respect to our impossibility results.

In a second part, we investigate the impact of our conditions in three case studies: participants interacting in a conference, robots moving on a grid and agents in the subway. Our simulations indicate a clear benefit of using our algorithms for reliable communication in those contexts.

1 Introduction

As modern networks grow larger, their components become more likely to fail, sometimes in unforeseen ways. As opportunistic networks become more widespread, the lack of global control over individual participants makes those networks particularly vulnerable to attacks. Many failure and attack models have been proposed, but one of the most general is the *Byzantine* model proposed by Lamport et al. [17]. 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

A common way to solve this problem is to use *cryptography* [6, 10]: the nodes use digital signatures to authenticate the sender across multiple hops. However, cryptography *per se* is not unconditionally reliable, as shown by the recent Heartbleed bug [1] discovered in the widely deployed OpenSSL software. The *defense in depth* paradigm [19] advocates the use of multiple layers of security controls, including non-cryptographic ones. For instance, if the cryptography-based security layer is compromised by a bug, a virus, or intentional tampering, a cryptography-free communication layer can be used to safely broadcast a patch or to update cryptographic keys. Thus, it is interesting to develop both cryptographic and non-cryptographic strategies.

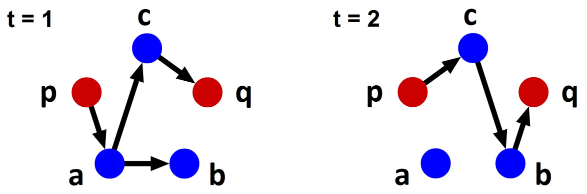

Figure 1: Counterexample to Menger's theorem in dynamic graphs.

Following the setting of the seminal paper of Lamport et al. [17], many subsequent papers focusing of Byzantine tolerance [2, 20, 21, 27] 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 Tseng, Vaidya and Liang [32, 33], 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 [23, 28, 31] 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 [11, 12, 13, 14, 22] 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 grid [3, 16] (but with much more than 4 neighbors), and later generalized to other topologies [30], 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 [24, 26, 25], 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 [8, 29, 9]. The initial solution [8] assumes that each node is aware of the global network topology, but this hypothesis was dropped in subsequent work [29, 18].

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

Our contribution

In this paper, our objective is to determine the condition for reliable communication in the presence of up to k Byzantine failures in a *dynamic* network, where the topology can vary with time. The proof technique used in [8, 29, 9] implicitly relies on Menger's theorem [4], which can be expressed as follows: there exists x disjoint paths between two nodes p and q if and only if x nodes must removed to disconnect p and q.

However, Menger's theorem does not generalize to dynamic networks [15]. To illustrate this, let us consider the simple dynamic network of Figure 1. This network is in two steps ($t = 1$ and $t = 2$). There exists three dynamic paths connecting p to q: (p, a, c, q), (p, c, b, q) and (p, a, b, q). To cut these three paths, at least two nodes must be removed: either $\{a, b\}$, $\{b, c\}$ or $\{a, c\}$. Yet, it is impossible to find two disjoint paths among the three dynamic paths. Therefore, Menger's theorem cannot be used to prove the condition in dynamic networks.

In this paper, we prove the necessary and sufficient condition for reliable communication in dynamic networks, in the presence of up to k Byzantine failures. We consider the two cases where cryptography is available and not available. Our characterization is based on a dynamic version of a minimal cut between p and q, denoted by $\text{DynMinCut}(p, q)$, 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. Then condition is that $\text{DynMinCut}(p, q)$ is lower or equal to $2k$ (without cryptography) or k (with cryptography). The proof is constructive, as we provide algorithms to prove the sufficiency of the condition.

In a second part, we apply these conditions to three case studies: participants interacting in a conference, robots moving on a grid and agents moving in the subway. We thus show the benefit of this multihop approach for reliable communication, instead of waiting that the source meets the sink directly (if this event is to occur).

Organization of the paper

The paper is organized as follows. In Section 2, we present the model and give basic definitions. In Section 3 (resp. 4), we give the algorithm and prove the condition for the non-cryptographic (resp. cryptographic) case. We present the case studies in Section 5.

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 [5], 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 [3, 8, 16, 24, 25, 26, 29, 30]. 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:*

- $e_i = (u_i, u_{i+1}) \in E$*, i.e. there exists an edge connecting* u_i *to* u_{i+1}*.*
- $\forall t \in [t_i, t_i + \zeta(e_i, t_i)]$*,* $\rho(e_i, t) = 1$*, i.e.* u_i *can send a message to* u_{i+1} *at date* t_i*.*
- $\zeta(e_i, t_i) \leq t_{i+1} - t_i$*, i.e. 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 $\mathit{Dyn}\,(p,q)$ be the set of node sets $\{u_1, \ldots, u_n\}$ such that $(p, u_1, \ldots, u_n, q)$ is a dynamic path.
- For a set of node sets $\Omega = \{S_1, \ldots, S_n\}$, let $\mathit{Cut}\,(\Omega)$ be the set of node sets 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 $\mathit{MinCut}\,(\Omega) = \min_{C\mathit{Cut}(\Omega)} |C|$ (the size of the smallest element of $\mathit{Cut}\,(\Omega)$). If $\mathit{Cut}(\Omega)$ is empty, we assume that $\mathit{MinCut}\,(\Omega) = +\infty$.
- Let $\mathit{DynMinCut}\,(p,q) = \mathit{MinCut}\,(\mathit{Dyn}\,(p,q))$.

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 2 (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 Non-cryptographic reliable communication

In this section, we consider that cryptography is not available. We first provide a 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 of the algorithm

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). As the Byzantine nodes can send any message, in particular, they can forward false tuples (p, m, S). Therefore, a correct node only 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).

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 three 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 $u.Acc$.

Condition for reliable communication

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

Theorem 1. *For a given dynamic graph, a k-Byzantine tolerant reliable communication from p to q is feasible if and only if $DynMinCut\,(p, q) > 2k$.*

Proof. The proof of the "only if" part is in Lemma 1. The proof of the "if" is in Lemma 4. □

Lemma 1 (Necessary condition)**.** *For a given dynamic graph, let us suppose that there exists an algorithm ensuring reliable communication from p to q. Then, we necessarily have $DynMinCut\,(p, q) > 2k$.*

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

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

According to the definition of $\mathit{Cut}\,(\mathit{Dyn}\,(p,q))$, 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 $|C_1| \leq k$.

  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 $|C_2| \leq k$.

  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 2, 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 $(p,m) \in q.\mathit{Acc}$, then $m = p.m_0$.*

Proof. As $(p,m) \in q.\mathit{Acc}$, 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 $\mathit{MinCut}\,(\{S_1, \dots, S_n\}) > k$.

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 $C \in \mathit{Cut}\,(\{S_1, \dots, S_n\})$ and $|C| \leq k$. This is impossible because $\mathit{MinCut}\,(\{S_1, \dots, S_n\}) > k$. 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 $n' = |S \cup \{p\}|$. 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 $|X_i| = |S \cup \{p\}| - i$.

- 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, $|X| = |X_i| - 1 = |S \cup \{p\}| - (i+1)$.

  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 $|X_{n'}| = 0$, $X_{n'} = \emptyset$ 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 DynMinCut $(p, q) > 2k$, and that all correct nodes follow our algorithm. Then, we eventually have $(p, p.m_0) \in q.Acc$.*

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

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

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

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

  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 $u_1 = p$, $X_N = \{u_1, \ldots, u_{N-1}\} = S \cup \{p\}$, and we eventually have $(p, p.m_0, S \cup \{p\}) \in q.\Omega$.

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

Lemma 4 (Sufficient condition). *Let there be any dynamic graph. Let p and q be two correct nodes, and k denote the maximum number of Byzantine nodes. If DynMinCut $(p, q) > 2k$, our algorithm ensures reliable communication from p to q.*

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 2, our algorithm ensures reliable communication from p to q. □

4 Cryptographic reliable communication

If cryptography is available, then it becomes possible to authenticate the sender of a message across multiple hops.

The setting is now the following. Each node p has a private key $priv_p$ (only known by p) and a public key pub_p (known by all nodes). The node p can encrypt a message m with the function $crypt(priv_p, m)$. Any node q can decrypt a message from p with the function $decrypt(pub_p, m)$. This function returns NULL if the message was not correctly encrypted. We assume that the Byzantine nodes do not know the private keys of correct nodes.

Then, we modify the previous algorithm as follows. Initially, $u.\Omega = \{(u, crypt(priv_u, u.m_0))\}$ and $u.Acc = \{(u, u.m_0)\}$. Then, each correct node u obeys to the three following rules:

1. Initially, and whenever $u.\Omega$ or the local topology of u change: multicast $u.\Omega$.
2. Upon reception of Ω' from a neighbor node: $u.\Omega = u.\Omega \cup \Omega'$.
3. Whenever there exists $(s, m) \in u.\Omega$ such that $m' = decrypt(pub_s, m) \neq NULL$: append (s, m') to $u.Acc$.

Theorem 2. *If cryptography is available, for a given dynamic graph, a k-Byzantine tolerant reliable communication from p to q is feasible if and only if DynMinCut* $(p, q) > k$.

Proof. If $DynMinCut\,(p, q) \leq k$, then it is possible to cut all dynamic paths between p and q with Byzantine nodes. Thus, q never receives any message from p. Thus, the condition is necessary. Now, let us show that the condition is sufficient.

First, q cannot accept a message (p, m) with $m \neq p.m_0$. Indeed, let us suppose the opposite. According to step 3 of the algorithm, it implies that we have $(p, m') \in q.\Omega$, with $decrypt(pub_p, m') = m$. Implying that $m' = crypt(priv_p, m)$. Let v be the first node to have $(p, m') \in v.\Omega$. According to steps 1 and 2 of the algorithm, v cannot be a correct node. Thus, v is Byzantine, implying that a Byzantine node knows $priv_p$: contradiction.

Besides, if $DynMinCut\,(p, q) > k$, then there exists at least one dynamic path from p to q. Thus, for the same argument as in Lemma 3, we eventually have $(p, crypt(priv_p, p.m_0)) \in q.\Omega$. Thus, according to step 3 of the algorithm, $(p, p.m_0)$ is added to $q.Acc$, and the condition is sufficient.

□

5 Case Studies

In this section, we apply our conditions for reliable communication to several case studies: participants interacting in a conference, robots moving on a grid and agents moving in the subway. We show the interest of multihop reliable communication.

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

In this section, we consider the Infocom 2005 dataset [7], 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, $k = 1$).

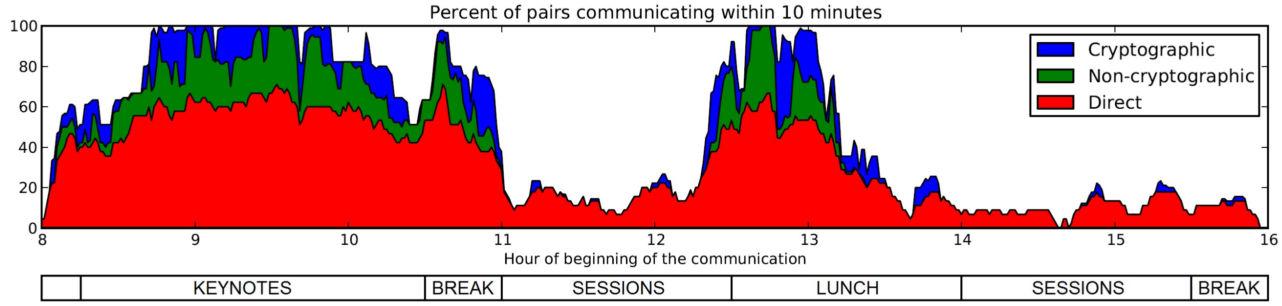

Figure 2: Reliable communication between 10 most sociable nodes of the Infocom 2005 dataset

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. Within 10 minutes, three types of communication can be achieved:

- *Direct* communication: p meets q directly.
- *Non-cryptographic* communication: the condition for reliable non-cryptographic communication (Theorem 1) is satisfied.
- *Cryptographic* communication: the condition for reliable cryptographic communication (Theorem 2) is satisfied.

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 is usually beneficial and that our approach realizes a significant gain of performance.

Figure 2 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 9:15, 60% of pairs of nodes meet directly, 80% can communicate reliably without cryptography, and 100% can communicate reliably with cryptography. This means that relaying the information is actually effective and desirable.

5.2 Probabilistic mobile robots on a grid

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

Definition 3 (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 $\mathit{position}(u, t)$ 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 $\mathit{position}(u, t) = \mathit{position}(v, t)$ and $\zeta((u, v), t) = 0$.

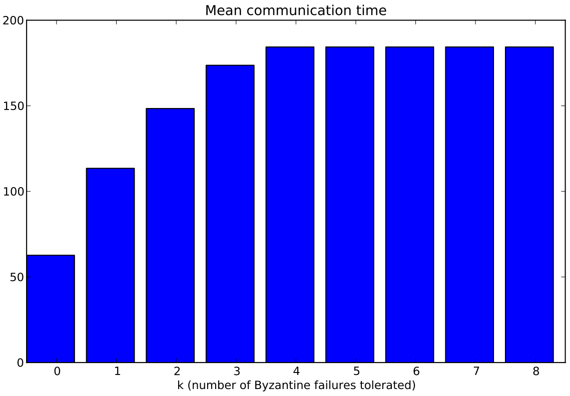

Figure 3: Mean communication time without cryptography (robots)

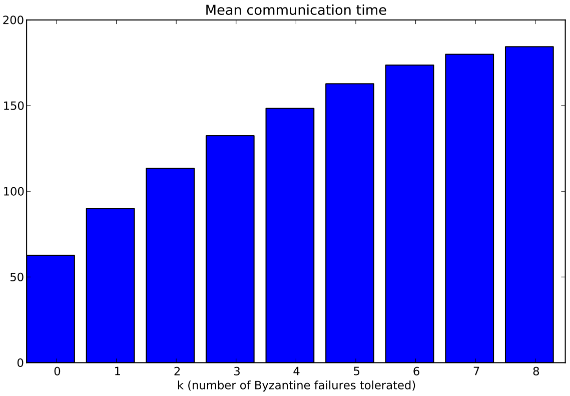

Figure 4: Mean communication time with cryptography (robots)

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 satisfy the condition for reliable communication with cryptography (Theorem 2) and without cryptography (Theorem 1). For this purpose, we ran more than 10000 simulations, and represented the results on Figure 3 and 4. Let us comment on these results.

In Figure 3, we represented the mean communication time varying the maximal number of Byzantine failures k when cryptography is available. This time increases regularly. The case $k = 8$ corresponds to the case where all the nodes (except p and q) are Byzantine. In this limit case, the only possibility for p and q to communicate is to meet directly.

In Figure 3, we represented the case where cryptography is not available. Here, the aforementioned limit case is reached for $k = 4$, as the condition for non-cryptographic reliable communication is harder to satisfy.

As we can see, the reliable multihop communication approach can be an interesting compromise. For instance, let us suppose that we want to tolerate one Byzantine failure ($k = 1$). Let us consider the mean time for p and q to meet directly. If we use our algorithms, this time decreases by 38% without cryptography, and by 51% with cryptography.

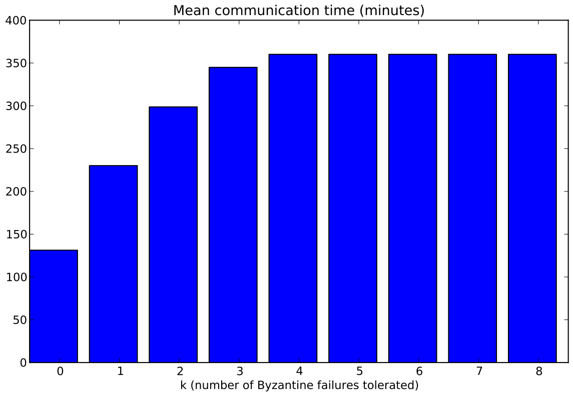

Figure 5: Mean communication time without cryptography (subway)

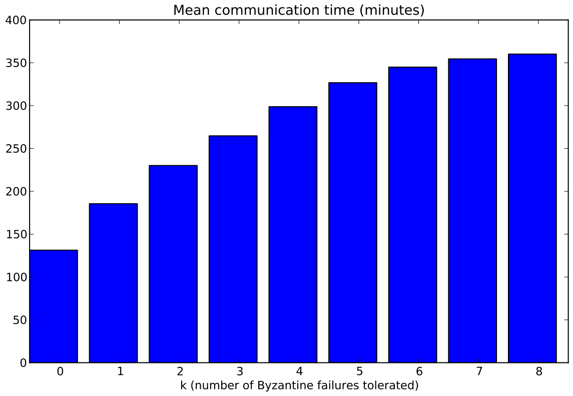

Figure 6: Mean communication time with cryptography (subway)

5.3 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 chooses 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.

Similarly to the previous case study, we represented the mean communication time with and without cryptography (see Figure 5 and 6). The qualitative observations are the same.

Again, let us suppose that we want to tolerate one Byzantine failure ($k = 1$). Let us consider the mean time for p and q to meet directly. If we use our algorithms, this time decreases by 36% without cryptography, and by 49% with cryptography.

6 Conclusion

In this paper, we gave the necessary and sufficient condition for reliable communication in a dynamic multihop network that is subject to Byzantine failures. We considered both cryptographic and non-cryptographic cases, and provided algorithms to show the sufficient condition. We then demonstrated the benefits of these algorithms in several case studies.

Our experiments explicitly quantify the benefits of a cryptographic infrastructure (fewer dynamic paths are required, less computations are necessary at each node for accepting genuine messages), but additional tradeofs are worth examining. In practice, ensuring hop by hop integrity through cryptography requires every node on the (dynamic) path to collect the public key of the sender (as it is unlikely that all public keys are initially bundled into the node, for memory size reasons and inclusion/exclusion node dynamics). Actually reaching a trusted authority from a gueninely dynamic network to obtain this public key raises both bootstrapping and performance issues.

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 (*e.g.* a random placement according to a particular distribution), and the associated necessary and sufficient condition for enabling multihop reliable communication, constitutes an interesting path for future research.

References

[1] The Heartbleed Bug (http://heartbleed.com).

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

[3] 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.

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

[5] 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.

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

[7] 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.

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

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

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

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

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

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

[18] Omri Liba. Erratum (http://vega.cs.kent.edu/˜mikhail/Research/topology .errata.html).

[19] R. Lippmann, K. Ingols, C. Scott, and K. Piwowarski. Validating and restoring defense in depth using attack graphs. *IEEE Military Communications Conference*, 2006.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

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

[28] 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.

[29] 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.

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

[31] 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.

[32] 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.

[33] 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.