Optimal Unconditional Information Diffusion

Dahlia Malkhi, Elan Pavlov, and Yaron Sella

School of Computer Science and Engineering
The Hebrew University of Jerusalem, Jerusalem 91904, Israel
{dalia,elan,ysella}@cs.huji.ac.il

Abstract. We present an algorithm for propagating updates with information theoretic security that propagates an update in time logarithmic in the number of replicas and linear in the number of corrupt replicas. We prove a matching lower bound for this problem.

I cannot tell how the truth may be; I say the tale as 'twas said to me. –Sir Walter Scott

1 Introduction

In this paper, we consider the problem of secure information dissemination with information theoretic guarantees. The system we consider consists of a set of *replica* servers that store copies of some information, e.g., a file. A concern of deploying replication over large scale, highly decentralized networks is that some threshold of the replicas may become (undetectably) corrupt. Protection by means of cryptographic signatures on the data might be voided if the corruption is the action of an internal intruder, might be impossible if data is generated by low powered devices, e.g., replicated sensors, or might simply be too costly to employ. The challenge we tackle in this work is to spread *updates* to the stored information in this system efficiently and with unconditional security, while preventing corrupted information from contaminating good replicas. Our model is relevant for applications that employ a client-server paradigm with replication by the servers, for example distributed databases and quorum-systems.

More specifically, our problem setting is as follows. Our system consists of n replica servers, of which strictly less than a threshold b may be arbitrarily *corrupt*; the rest are *good* replicas. We require that each pair of good servers is connected by an authenticated, reliable, non-malleable communication channel. In order to be able to distinguish correct updates from corrupted (spurious) ones, we postulate that each update is initially input to an *initial set* of α good replicas, where α is at least b, the presumed threshold on the possible number of corrupt replicas. In a client-server paradigm, this means that the client's protocol for submitting an update to the servers addresses all the replicas in the initial set. The initial set is not known apriori, nor is it known to the replicas themselves at the outset of the protocol, or even during the protocol. Multiple updates are being continuously introduced to randomly designated initial sets,

and the diffusion of multiple updates actually occurs simultaneously. This is done by packing several updates in each message. Because we work with information theoretic security, the only criterion by which an update is accepted through diffusion by a good replica is when b different replicas independently vouch for its veracity. It should be stressed that we do not employ cryptographic primitives that are conditioned on any intractability assumptions, and hence, our model is the full Byzantine model without signatures.

The problem of secure information dissemination in a full Byzantine environment was initiated in [MMR99] and further explored in [MRRS01]. Because of the need to achieve information theoretic security, the only method to ascertain the veracity of updates is by replication. Consequently, those works operated with the following underlying principle: A replica is initially *active* for an update if it is input to it, and otherwise it is *passive*. Active replicas participate in a diffusion protocol to disseminate updates to passive replicas. A passive replica becomes active when it receives an update directly from b different sources, and consequently becomes *active* in its diffusion. For reasons that will become clear below, we call all algorithms taking this approach *conservative*. More formally:

Definition 1. *A diffusion algorithm in which a good replica p sends an update u to another replica q only if p is sure of the update's veracity is called* conservative.

In contrast, we call non-conservative algorithms *liberal*. Conservative algorithms are significantly limited in their performance. To illustrate this, we need to informally establish some terminology. First, for the purpose of analysis, we conceive of propagation protocols as progressing in synchronous *rounds*, though in practice, the rounds need not occur in synchrony. Further, for simplicity, we assume that in each round a good replica can send out at most one message (i.e., the *Fan-out*, F^{out}, is one); more detailed treatment can relate to F^{out} as an additional parameter. The two performance measures introduced in [MMR99] are as follows (precise definitions are given in the body of the paper):

- Let *Delay* denote the expected number of communication rounds from when an update is input to the system and until it reaches all the replicas;
- Let *Fan-in* (F^{in}) denote the expected maximum number of messages received by any replica from good replicas in a round (intuitively, the F^{in} measures the "load" on replicas).

In [MMR99] a lower bound is shown on conservative algorithms of $Delay * F^{in} = \Omega((nb/\alpha)^{1-\frac{1}{b}})$. This linear lower bound is discouraging, especially compared with the cost of epidemic-style diffusion of updates in benign-failure environments[1], which has $Delay * F^{in} = O(\log n)$. Such efficient diffusion would have been possible in a Byzantine setting if signatures were utilized to distinguish correct from spurious updates, but as already discussed, deploying digital signatures is ruled out in our setting. It appears that the advantages achieved by avoiding digital signatures come at a grave price.

[1] In epidemic-style diffusion we refer to a method whereby in each round, each active replica chooses a target replica independently at random and sends to it the update.

Fortunately, in this paper we propose an approach for diffusion in full Byzantine settings that is able to circumvent the predictions of [MMR99] using a fundamentally different approach. Our proposed liberal algorithm has $Delay*F^{in} = O(b + \log n)$ and enjoys the same simplicity of epidemic-style propagation. The main price paid is in the size of messages used in the protocol. Although previous analyses ignored the size of messages, we note that our method requires additional communication space of $n^{O(\log(b+\log n))}$ per message. In terms of delay, we prove our algorithm optimal by showing a general lower bound of $\Omega(b\frac{n-\alpha}{n}+\log\frac{n}{\alpha})$ on the delay for the problem model.

Our liberal approach works as follows. As before, a replica starts the protocol as *active* if it receives an update as input. Other replicas start as *passive*. Active replicas send copies of the update to other replicas at random. When a passive replica receives a copy of an update through another replica, it becomes *hesitant* for this update. A hesitant replica sends copies of the update, along with information about the paths it was received from, to randomly chosen replicas. Finally, when a replica receives copies of an update over b vertex-disjoint paths, it believes its veracity, and becomes active for it.

It should first be noted that this method does not allow corrupt updates to be accepted by good replicas. Intuitively, this is because when an update reaches a good replica, the last corrupt replica it passed through is correctly expressed in its path. Therefore, a spurious update cannot reach a good replica over b disjoint paths.

It is left to analyze the diffusion time and message complexity incurred by the propagation of these paths. Here, care should be taken. Since we show that a lower bound of $\Omega(b\frac{n-\alpha}{n} + \log\frac{n}{\alpha})$ holds on the delay, then if path-lengthening proceeds uncontrolled throughout the algorithm, then messages might carry up to $O(b^b)$ paths. For a large b, this would be intolerable, and also too large to search for disjoint paths at the receivers. Another alternative that would be tempting is to try to describe the paths more concisely by simply describing the graph that they form, having at most $O(nb)$ edges. Here, the problem is that corrupt replicas can in fact create spurious updates that appear to propagate along b vertex-disjoint paths in the graph, despite the fact that there were no such paths in the diffusion.

Our solution is to limit all paths to length $\log\frac{n}{b}$. That is, a replica that receives an update over a path of length $\log\frac{n}{b}$ does not continue to further propagate this path. Nevertheless, we let the propagation process run for $O(b+\log\frac{n}{b})$ rounds, during which paths shorter than $\log\frac{n}{b}$ continue to lengthen. This process generates a dense collection of limited length paths. Intuitively, the diffusion process then evolves in two stages.

1. First, the diffusion of updates from the α active starting points is carried as an independent epidemic-style process, so each one of the active replicas establishes a group of hesitant replicas to a vicinity of logarithmic diameter.
2. Each log-diameter vicinity of active replicas now directly targets (i.e., with paths of length 1) the remaining graph. With careful analyses it is shown that it takes additional $O(b)$ rounds for each replica to be targeted directly

by some node from b out of the α disjoint vicinities of active replicas, over b disjoint paths.

Throughout the protocol, each replica diffuses information about up to $O((b+\log\frac{n}{b})^{\log\frac{n}{b}})$ different paths, which is the space overhead on the communication.

1.1 Related work

Diffusion is a fundamental mechanism for driving replicated data to a consistent state in a highly decentralized system. Our work optimizes diffusion protocols in systems where arbitrary failures are a concern, and may form a basis of solutions for disseminating critical information in this setting.

The study of Byzantine diffusion was initiated in [MMR99]. That work established a lower bound for conservative algorithms, and presented a family of nearly optimal conservative protocols. Our work is similar to the approach taken in [MMR99] in its use of epidemic-style propagation, and consequently in its probabilistic guarantees. It also enjoys similar simplicity of deployment, especially in real-life systems where partially-overlapping universes of replicas exist for different data objects, and the propagation scheme needs to handle multiple updates to different objects simultaneously. The protocols of [MMR99] were further improved, and indeed, the lower bound of [MMR99] circumvented to some extent, in [MRRS01], but their general worst case remained the same.

The fundamental distinction between our work and the above works is in the liberal approach we take. With liberal approach, we are able to completely circumvent the lower bound of [MMR99], albeit at the cost of increased message size. An additional advantage of liberal methods is that in principle, they can provide update diffusion in any b-connected graph (though some topologies may increase the delay of diffusion), whereas the conservative approach might simply fail to diffuse updates if the network is not fully connected. The investigation of secure information diffusion in various network topologies is not pursued further in this paper however, and is a topic of our ongoing research. The main advantage of the conservative approach is that spurious updates generated by corrupt replicas cannot cause good replicas to send messages containing them; they may however inflict load on the good replicas in storage and in receiving and processing these updates. Hence, means for constraining the load induced by corrupt replicas must exist in both approaches.

While working on this paper, we learned that our liberal approach to secure information diffusion has been independently investigated by Minsky and Schneider [MS01]. Their diffusion algorithms use age to decide which updates to keep and which to discard, in contrast to our approach which discards based on the length of the path an update has traversed. Also, in the algorithms of [MS01], replicas pull updates, rather than push messages to other replicas, in order to limit the ability of corrupt hosts to inject bogus paths into the system. Simulation experiments are used in [MS01] to gain insight into the performance of those protocols; a closed-form analysis was sought by Minsky and Schneider but could not be obtained. Our work provides the foundations needed to analyze

liberal diffusion methods, provides general lower bounds, and proves optimality of the protocol we present.

Prior to the above works, previous work on update diffusion focused on systems that can suffer benign failures only. Notably, Demers et al. [DGH+87] performed a detailed study of epidemic algorithms for the benign setting, in which each update is initially known at a single replica and must be diffused to all replicas with minimal traffic overhead. One of the algorithms they studied, called *anti-entropy* and apparently initially proposed in [BLNS82], was adopted in Xerox's Clearinghouse project (see [DGH+87]) and the Ensemble system [BHO+99]. Similar ideas also underly IP-Multicast [Dee89] and MUSE (for USENET News propagation) [LOM94]. The algorithms studied here for Byzantine environments behave fundamentally differently from any of the above settings where the system exhibits benign failures only.

Prior studies of update diffusion in distributed systems that can suffer Byzantine failures have focused on single-source broadcast protocols that provide reliable communication to replicas and replica agreement on the broadcast value (e.g., [LSP82,DS83,BT85,MR97]), sometimes with additional ordering guarantees on the delivery of updates from different sources (e.g., [Rei94,CASD95,MM95,KMM98,CL99]). The problem that we consider here is different from these works in the following ways. First, in these prior works, it is assumed that one replica begins with each update, and that this replica may be faulty—in which case the good replicas can agree on an arbitrary update. In contrast, in our scenario we assume that at least a threshold $\alpha \geq b$ of good replicas begin with each update, and that only these updates (and no arbitrary ones) can be accepted by good replicas. Second, these prior works focus on reliability, i.e., *guaranteeing* that all good replicas (or all good replicas in some agreed-upon subset of replicas) receive the update. Our protocols diffuse each update to all good replicas only with some probability that is determined by the number of rounds for which the update is propagated before it is discarded. Our goal is to devise diffusion algorithms that are efficient in the number of rounds until the update is expected to be diffused globally and the load imposed on each replica as measured by the number of messages it receives in each round.

2 Preliminaries

Following the system model of [MMR99], our system consists of a universe S of n *replicas* to which updates are input. Strictly less than some known threshold b of the replicas could be *corrupt*; a corrupt replica can deviate from its specification arbitrarily (Byzantine failures). Replicas that always satisfy their specifications are *good*. We do not allow the use of digital signatures by replicas, and hence, our model is the full information-theoretic Byzantine model.

Replicas can communicate via a completely connected point-to-point network. Communication channels between good replicas are reliable and authenticated, in the sense that a good replica p_i receives a message on the communi-

cation channel from another good replica p_j if and only if p_j sent that message to p_i.

Our work is concerned with the diffusion of *updates* among the replicas. Each update u is introduced to an *initial set* I_u containing at least $\alpha \geq b$ good replicas, and is then diffused to other replicas via message passing. Replicas in I_u are considered *active* for u. The goal of a diffusion algorithm is to make all good replicas *active* for u, where a replica p is active for u only if it can guarantee its veracity.

Our diffusion protocols proceed in synchronous rounds. For simplicity, we assume that each update arrives at each replica in I_u simultaneously, i.e., in the same round at each replica in I_u. This assumption is made purely for simplicity and does not impact on either the correctness or the speed of our protocol. In each round, each replica selects one other replica to which it sends information about updates as prescribed by the diffusion protocol. That is, the *Fan-out*, F^{out}, is assumed to be 1.[2] A replica receives and processes all the messages sent to it in a round before the next round starts.

We consider the following three measures of quality for diffusion protocols:

Delay: For each update, the delay is the worst-case expected number of rounds from the time the update is introduced to the system until all good replicas are active for update. Formally, let η_u be the round number in which update u is introduced to the system, and let τ_p^u be the round in which a good replica p becomes active for update u. The delay is $E[\max_{p,C}\{\tau_p^u\} - \eta_u]$, where the expectation is over the random choices of the algorithm and the maximization is over good replicas p, all failure configurations C containing fewer than b failures, and all behaviors of those corrupt replicas. In particular, $\max_{p,C}\{\tau_p^u\}$ is reached when the corrupt replicas send no updates, and our delay analysis applies to this case.

Fan-in: The fan-in measure, denoted by F^{in}, is the expected maximum number of messages that any good replica receives in a single round from good replicas under all possible failure scenarios. Formally, let ρ_p^i be the number of messages received in round i by replica p from good replicas. Then the fan-in in round i is $E[\max_{p,C}\{\rho_p^i\}]$, where the maximum is taken with respect to all good replicas p and all failure configurations C containing fewer than b failures. *Amortized fan-in* is the expected maximum number of messages received over multiple rounds, normalized by the number of rounds. Formally, a k-amortized fan-in starting at round l is $E[\max_{p,C}\{\sum_{i=l}^{l+k} \rho_p^i/k\}]$. We emphasize that fan-in and amortized fan-in are measures only for messages from good replicas.

Communication complexity: The maximum amount of information pertaining to a specific update, that was sent by a good replica in a single message. The maximum is taken on all the messages sent (in any round), and with respect to all good replicas and all failure configurations C containing fewer than b failures.

[2] We could expand the treatment here to relate to F^{out} as a parameter, but chose not to do so for simplicity.

Note that what interests us is the *expected* value of the measures. When we make statements of the type "within an expected $f(r)$ rounds, $P(r)$" (for some predicate P, and function f), we mean that if we define X as a random variable that measures the time until $P(r)$ is true then $E(X) = f(r)$.

The following bound presents an inherent tradeoff between delay and fan-in for conservative diffusion methods (Definition 1), when the initial set I_u is arbitrarily designated:

Theorem 1 ([MMR99]). *Let there be a conservative diffusion algorithm. Denote by D the algorithm's delay, and by F^{in} its D-amortized fan-in. Then $DF^{in} = \Omega(bn/\alpha)$, for $b \geq 2\log n$.*

One contribution of the present work is to show that the lower bound of Theorem 1 for conservative diffusion algorithms, does not hold once inactive replicas are allowed to participate in the diffusion.

3 Lower Bounds

In this section we present lower bounds which apply to any diffusion method in our setting. Our main theorem sets a lower bound on the delay. It states that the propagation time is related linearly to the number of corrupt replicas and logarithmically to the total number of replicas.

We start by showing the relation between the delay and the number of corrupt players.

Lemma 1. *Let there be any diffusion algorithm in our setting. Let D denote the algorithm's delay. Then $D = \Omega(b\frac{n-\alpha}{n})$.*

Proof. Since it is possible that there are $b-1$ corrupt replicas, each good replica who did not receive the update initially as input must be targeted directly by at least b different other replicas, as otherwise corrupt replicas can cause it to accept an invalid update. Since only α replicas receive the update initially, at least $b(n-\alpha)$ direct messages must be sent. As $F_{out} = 1$ and there are n replicas, at most n messages are sent in each round. Therefore it takes at least $b\frac{n-\alpha}{n}$ rounds to have $b(n-\alpha)$ direct messages sent.

We now show the relationship of the delay to the number of replicas.

Lemma 2. *Let there be any diffusion algorithm in our setting. Let D denote the algorithm's delay. Then $D = \Omega(\log\frac{n}{\alpha})$.*

Proof. Each replica has to receive a copy of the update. Since $F_{out} = 1$, the number of replicas who receive the update up to round t is at most twice the number of replicas who received the update up to round $t-1$. Therefore at the final round t_{end}, when all replicas received the update, we have that $2^{t_{end}}\alpha = n$ or $t_{end} = \log\frac{n}{\alpha}$.

The following theorem immediately follows from the previous two lemmas:

Theorem 2. *Let there be any diffusion algorithm in our setting. Let D denote the algorithm's delay. Then $D = \Omega(b\frac{n-\alpha}{n} + \log\frac{n}{\alpha})$.*

Remark 1. We will deal primarily in the case where $\alpha \leq \frac{n}{2}$ as otherwise the diffusion problem is relatively simple. In particular, if $\alpha > \frac{n}{2}$, then we can use the algorithm of [MMR99] to yield delay of $O(b)$, which is optimal for $F^{out} = 1$. When $\alpha \leq \frac{n}{2}$ our lower bound is equal to $\Omega(b + \log\frac{n}{\alpha})$, which is met by the propagation algorithm presented below.

Remark 2. We note that in order for an update to propagate successfully we must have that $\alpha > b$. From this, it immediately follows that $b < \frac{n}{2}$. However, below we shall have a tighter constraint on b that stems from our diffusion method. We note that throughout this paper no attempt is made to optimize constants.

4 The propagation algorithm

In this section we present an optimal propagation algorithm that matches the lower bound shown in section 3.

In our protocol, each replica can be in one of three states for a particular update: *passive, hesitant or active*. Each replica starts off either in the active state, if it receives the update initially as input, or (otherwise) in the passive state. In each round, the actions performed by a replica are determined by its state. The algorithm performed in a round concerning a particular update is as follows:

- An active replica chooses a random replica and sends the update to it. (Compared with the actions of hesitant replicas below, the lack of any paths attached to the update conveys the replica's belief in the update's veracity.)
- A passive or hesitant replica p that receives the update from q, with various (possibly empty) path descriptions attached, appends q to the end of each path and saves the paths. If p was passive, it becomes hesitant.
- A hesitant replica chooses a random replica and sends to it all vertex-minimal paths of length $< \log\frac{n}{b}$ over which the update was received.
- A hesitant replica that has b vertex disjoint paths for the update becomes active.

A couple of things are worth noting here. First, it should be clear that the algorithm above executes simultaneously for all concurrently propagating updates. Second, any particular update is propagated by replicas for a limited number of rounds. The purpose of the analysis in the rest of the paper is to determine the number of rounds needed for the full propagation of an update. Finally, some

optimizations are possible. For example, a hesitant replica p that has b vertex disjoint paths passing through a single vertex q (i.e., disjoint between q and p) can unify the paths to be equivalent to a direct communication from the vertex q.

We now prove that our algorithm is correct.

Lemma 3. *If a good replica becomes active for an update then the update was initially input to a good replica.*

Proof. There are two possible ways in which a good replica can become active for an update. The first possibility is when the replica receives the update initially as input. In this case the claim certainly holds.

The second possibility is when the replica receives the update over b vertex disjoint paths. We say that a corrupt replica *controls* a path if it is the last corrupt replica in the path. Note that for any invalid update which was generated by corrupt replica(s), there is exactly one corrupt replica controlling any path (since by definition the update was created by the corrupt replicas). Since good replicas follow the protocol and do not change the path(s) they received, the corrupt controlling replica will not be removed from any path by any subsequent good replica receiving the update. As there are less than b corrupt replicas and the paths are vertex disjoint there are less than b such paths. As a good replica becomes active for an update when it receives the update over b disjoint paths, at least one of the paths has only good replicas in it. Therefore the update was input to a good replica.

The rest of this paper will prove the converse direction. If an update was initially input to $\alpha \geq b$ good replicas then within a relatively small number of rounds, all good replicas will receive the update with high probability.

5 Performance analysis

In this section, we proceed to analyze the performance of our algorithm. Our treatment is based on a communication graph that gradually evolves in the execution. We introduce some notation to be used in the analysis below. At every round r, the communication graph $G_r = (V, E_r)$ is defined on (good) vertices V such that there is a (directed) edge between two vertices if one sent any message to the other during round r. We denote by $N_G(I)$ the neighborhood of I (singleton or set) in G. We denote by $\| p, q \|_G$ the shortest distance between p and q in G. In the analysis below, we use vertices and replicas interchangeably.

Our proof will make use of *gossip-circles* that gradually evolve around active replicas. Intuitively, the gossip-circle $C(p, d, r)$ of a good active replica is the set of good replica that heard the update from p over good paths (comprising good replicas) of length up to d in r rounds. Formally:

Definition 2. *Let p be some good replica which is active for the update u. Let $\{G_j = (V, E_j)\}_{j=1..r}$ be the set of communication graphs of r rounds of the execution of vertices in V. Recall that $N_G(I)$ denotes the set of all neighbors in a graph G of nodes in I. We then define* gossip circles *of p in r rounds inductively as follows:*

$C_V(p, 0, r) = \{p\}$

$\forall 1 \leq d \leq r$:

$$C_V(p, d, r) = C_V(p, d-1, r) \cup \{q \in N_{G_d}(C_V(p, d-1, r)) : \| p, q \|_{C_V(p,d-1,r)} \leq \min\{d-1, \log \tfrac{n}{b} - 1\}\}$$

When V is the set of good replicas, we omit it for simplicity. Note that the gossip circle $C(p, d, r)$ is constrained by definition to have radius $\leq \min\{d, \log \frac{n}{b}\}$.

The idea behind our analysis is that any b initial active good replicas spread paths that cover *disjoint* low-diameter gossip-circles of size $\frac{n}{4b}$. Hence, it is sufficient for any replica to be directly targeted by some replica from each one of these sets in order to have b vertex-disjoint paths from initial replicas.

We first show a lemma about the spreading of epidemic style propagation with bounded path length. Without bounding paths, the analysis reduces to epidemic-style propagation for benign environment, as shown in [DGH+87].

Lemma 4. *Let $p \in I_u$ be a good replica, and let $d \leq \log \frac{n}{b}$. Assume there are no corrupt replicas. Then within an expected $r > d$ rounds, $|C(p, d, r)| \geq \min\{(\frac{3}{2})^d + (r-d)(\frac{3}{2})^{d-4}, \frac{n}{2}\}$.*

Proof. The proof looks at an execution of r rounds of propagation in two parts. The first part consists of d rounds. In this part, the set of replicas that received a copy of u (equivalently, received a copy of u over paths of length $\leq d$), grows exponentially. That is, in d rounds, the update propagates to $(\frac{3}{2})^d$ replicas. The second part consists of the remaining $r - d$ rounds. This part makes use of the fact that at the end of the first part, an expected $(\frac{3}{2})^{d-4}$ replicas receive a copy of u over paths of length $< d$. Hence, in the second part, a total of $(r-d) \times (\frac{3}{2})^{d-4}$ replicas receive u.

Formally, let m_j denote the number of replicas that received u from p over paths of length $\leq d$ by round j, i.e., $m_j = |C(p, d, j)|$.

Let $j \leq d$. So long as the number of replicas reached by paths of length $\leq d$ does not already exceed $\frac{n}{2}$, then in round $j+1$ each replica in $C(p, d, j)$ targets a new replica with probability $\geq \frac{1}{2}$. Therefore, the expected number of messages sent until $\frac{m_j}{2}$ new replicas are targeted is at most m_j. Furthermore, since at least m_j messages are sent in round j, this occurs within an expected one round. We therefore have that the expected time until $(\frac{3}{2})^d$ replicas receive u over paths of length $\leq d$ is at most d.

From round $d+1$ on, we note that at least half of m_d received u over paths of length strictly less than d. Therefore, in each round $j > d$, there are at least $\frac{1}{2} \times (\frac{3}{2})^d$ replicas forwarding u over paths of length $< d$. So long as $m_j \leq \frac{n}{2}$, then in round j each of these replicas targets a new replica with probability $\geq \frac{1}{2}$. Therefore, the expected number of messages sent until $(\frac{3}{2})^{d-4} < \frac{1}{2} \times \frac{1}{2} \times (\frac{3}{2})^d$

new replicas are targeted is at most $\frac{1}{2} \times (\frac{3}{2})^d$, which occurs in an expected one round.

Putting the above together, we have that within an expected r rounds, $(\frac{3}{2})^d + (r-d) \times (\frac{3}{2})^{d-4}$ replicas are in $C(p,d,r)$.

Since the choice of communication edges in the communication graph is made at random, we get as an immediate corollary:

Corollary 1. *Let $V' \subseteq V$ be a set of vertices, containing all corrupt ones, chosen independently from the choices of the algorithm, such that $|V'| \leq \frac{n}{3}$. Let $p \in I_u$ be a good replica, and let $d \leq \log \frac{n}{b}$. Then within an expected $3r > d$ rounds, $|C_{(V \setminus V')}(p,d,3r)| \geq \min\{(\frac{3}{2})^d + (r-d)(\frac{3}{2})^{d-4}, \frac{n}{2}\}$.*

We now use corollary 1 to build b disjoint gossip circles of initial replicas, and wish to proceed with the analysis of the number of rounds it takes for replicas to be targeted by these disjoint sets. As edges in the communication graph are built at random, a tempting approach would be to treat this as a simple coupon collector problem on the b gossip-circles where each replica wishes to "collect a member" of each of these sets by being targeted with an edge from it. With this simplistic analysis, it would take each replica $O(b \log b)$ rounds to collect all the coupons, and an additional logarithmic factor in n for all replicas to complete. The resulting analysis would provide an upper bound of $O(b(\log b)(\log n))$ on the delay. Although this is sufficient for small b, for large b we wish to further tighten the analysis on the number of rounds needed for diffusion.

The approach we take is to gradually adapt the size of the disjoint gossip-circles as the process evolves, and to show that the expected amount of time until all sets are connected to a replica remains constant. More precisely, we show that in an expected $O(b)$ rounds, a replica has edges to half of b gossip-circles of size $\frac{n}{4b}$. We then look at the communication graph with all of the vertices in the paths of the previous step(s) removed. We show that in time $O(b/2)$, a replica has edges to gossip-circles of size $\frac{2n}{4b}$ of half of the $\frac{b}{2}$ remaining initial replicas. And so on. In general, we have an inductive analysis for $k = 0..\log b$. For each k, we denote $b_k = \frac{b}{2^k}$. For step k of the analysis, we show that in time $O(b_k)$, a replica has disjoint paths of length $\leq \log \frac{n}{b \times b_k}$ to $\frac{b_k}{2}$ of the initial replicas. Hence, in total time $O(b)$, a replica connects to b initial replicas over disjoint paths, all of length $\leq \log \frac{n}{b}$ (and hence, not exceeding the algorithm's path limit).

Our use of Corollary 1 is as follows. Let $b_k = \frac{b}{2^k}$, and let V' denote a set of vertices we wish to exclude from the graph, where $|V'| \leq \frac{n}{3}$. Then we have that within an expected $3r = 3(b + 2\log \frac{n}{b \times b_k})$ rounds, each initial good replica has a gossip circle of diameter $d = \max\{1, 2\log \frac{n}{b \times b_k}\}$ whose size is at least $(b+d-d)(\frac{3}{2})^{(d-4)} \geq \frac{n}{4b_k}$.

We now use this fact to designate disjoint low-diameter gossip circles around b good replicas in I_u.

Lemma 5. *Let $I \subseteq I_u$ be a subset of initial good replicas of size b_k. Let W' be a subset of replicas with $|W'| \leq \frac{n}{12}$. Denote by $d = \max\{1, 2\log\frac{n}{b\times b_k}\}$. Then within an expected $3r = 3(b+d)$ rounds there exist disjoint subsets $\{C_i\}_{i\in I}$ containing no vertices of W', such that each $C_i \subseteq C_{(V\setminus W')}(i,d,3r)$, and such that each $|C_i| = \frac{n}{4b_k}$.*

Proof. The proof builds these sets for I inductively. Suppose that $C_1, ..., C_{i-1}$, for $0 < i \leq b_k$, have been designated already, such that for all $1 \leq j \leq i-1$, we have that $C_j \subseteq C(j,d,3r)$ and $|C_j| = \frac{n}{4b_k}$. Denote by $C = \bigcup_{j=1..i-1} C_j$. Then the total number of vertices in $V' = C \cup W'$ is at most $\frac{n}{12} + (i-1)\frac{n}{4b_k} \leq \frac{n}{12} + b_k\frac{n}{4b_k} \leq \frac{n}{3}$. From Corollary 1, we get that within an expected $3r$ rounds, and without using any vertex in V', the gossip circle $C_{V\setminus V'}(i,d,3r)$ contains at least $\left(b + 2\log\frac{n}{b\times b_k} - 2\log\frac{n}{b\times b_k}\right)\left(\frac{3}{2}\right)^{\left(2\log\frac{n}{b\times b_k}-4\right)} \geq \frac{n}{4b_k}$. Hence, we set C_i to be a subset of $C(i,d,3r)$ of size $\frac{n}{4b_k}$ and the lemma follows.

We now analyze the delay until a vertex has direct edges to these b_k disjoint sets.

Lemma 6. *Let $v \in V$ be a good replica. Let $b_k = \frac{b}{2^k}$ as before and let $\{C_i\}_{i=1..b_k}$ be disjoint sets, each of size $\frac{n}{4b_k}$ and diameter $2\log\frac{n}{b\times b_k}$ (as determined by Lemma 5). Then within an expected $4b_k$ rounds there are edges from $\frac{b_k}{2}$ of the sets to v.*

Proof. The proof is simply a coupon collector analysis of collecting $\frac{b_k}{2}$ out of b_k coupons, where in epoch i, for $1 \leq i \leq \frac{b_k}{2}$, the probability of collecting the i'th new coupon in a round is precisely the probability of v being targeted by a new set, i.e., $\frac{(b_k-i)\frac{n}{4b_k}}{n}$. The expected number of rounds until completion is therefore $\sum_{i=1..(b_k/2)} \frac{4b_k}{b_k-i} \leq 4b_k$.

We are now ready to put these facts together to analyze the delay that a single vertex incurs for having disjoint paths to b initial replicas.

Lemma 7. *Let $v \in V$ be a good replica. Suppose that $b < \frac{n}{60}$. Then within an expected $5(b + \log\frac{n}{b})$ rounds there are b vertex disjoint paths of length $\leq \log\frac{n}{b}$ from I_u to v.*

Proof. We prove by induction on $b_k = \frac{b}{2^k}$, for $k = 0..(\log b - 1)$. To begin the induction, we set $b_0 = b$. By Corollary 1, within an expected $b + 2\log\frac{n}{b\times b_0}$ stages, there are $b_0 = b$ disjoint sets (of radius $2\log\frac{n}{b\times b_0}$) whose size is $\frac{n}{4b_0}$. By Lemma 6, within $4b_0$ rounds, v has direct edges to $\frac{b_0}{2}$ of these sets. Hence, it has disjoint paths of length $\leq 2\log\frac{n}{b\times b_0} + 1$ to $\frac{b_0}{2}$ initial replicas. These paths comprise at most $\frac{b_0}{2}(2\log\frac{n}{b\times b_0} + 1)$ good vertices.

For step $0 \leq k < (\log b)$ of the analysis, we set $b_k = \frac{b}{2^k}$. The set of vertices used in paths so far, together with all the corrupt vertices, total less than

$$b+\sum_{k'<k}\frac{b_{k'}}{2}\left(2\log\frac{n}{b\times b_{k'}}+1\right)\leq b+\sum_{k'<k}\frac{b}{2^{k'}}\left(\log\frac{2^{k'}n}{b^2}+1\right)\leq b+2b(1+\log\frac{n}{b^2})\,.$$

By our assumption that $b<\frac{n}{60}$, we get that the total number of vertices used until step k is less than $\frac{n}{12}$. Hence, in each step $0\leq k<\log b$, we apply Corollary 1 to form b_k disjoint sets (of radius $2\log\frac{n}{b\times b_k}$) whose size is $\frac{n}{4b_k}$ each. By Lemma 6, half of these sets have direct edges to v within an expected $4b_k$ rounds.

In total, we showed that in expected $\max_{0\leq k<\log b}\{4b_k+b+2\log\frac{n}{b\times b_k}\}$ rounds, v has disjoint paths (of length at most $\log\frac{n}{b}$) to b initial replicas.

We now wish to bound the time when **all** of the nodes have b vertex disjoint paths to I_u. A tempting approach would be to use a Chernoff bound, but the analysis would then require an additional logarithmic factor in n. This factor can be avoided by utilizing the fact that after a $O(\log n+b)$ rounds there exist a fraction of the replicas who are active for the update. Finally, propagation from a linear set is easily done.

Lemma 8. *Let $c>1$ be a constant. The expected time until $(n-b)\left(1-\frac{1}{c}\right)$ replicas become active is $O(b+\log n)$.*

Proof. By Lemma 7, the expected time for a replica to become active is $5(b+\log\frac{n}{b})$. Hence, the probability that a replica becomes active in $c\times 5(b+\log\frac{n}{b})$ rounds or more is less than $\frac{1}{c}$. Hence, within an expected $c\times 5(b+\log\frac{n}{b})$ rounds the number of active replicas is at least $(n-b)\left(1-\frac{1}{c}\right)$.

We now choose a particular value for c in the previous lemma. We note that we choose an arbitrary value without attempting to minimize the constants.

For $c=2$, within an expected $10(b+\log\frac{n}{b})$ rounds there are $\frac{1}{2}(n-b)$ replicas who are active for the update. By reusing the supposition $b<\frac{n}{60}$ from Lemma 7, we get that $\frac{1}{2}(n-b)>\frac{1}{2}(n-\frac{n}{60})>\frac{2}{5}n$. This means that there are at least $\frac{2}{5}n$ good replicas who are active for the update.

Lemma 9. *If at least $\frac{2}{5}n$ good replicas are active for the update then within an expected $O(b+\log n)$ rounds all of the replicas become active for the update.*

Proof. Fix any replica and let Y_i be the number of updates from active replicas that the replica receives in round i. Let Y be the number of updates that the replica receives in r rounds, i.e., $Y=\sum_{i=1}^{r}Y_i$. By the linearity of expectation, $E(Y)=\sum_{i=1}^{r}E(Y_i)\geq\frac{2}{5}r$. Using a Chernoff bound we have that $Pr[Y\leq\frac{r}{10}]\leq e^{-\frac{r}{48}}$. Therefore if $r=48\log n+2b$ we have that $Pr[Y\leq\frac{r}{10}]\leq\frac{1}{n^2}$.

Theorem 3. *The algorithm terminates in an expected $O(\log n+b)$ rounds.*

Proof. By corollary 7, and lemma 8 it follows that within $O(\log n+b)$ rounds 0.8 of the replicas become active. From Lemma 9 within an additional $O(\log n+b)$ rounds all of the replicas become active.

Therefore, our delay matches the lower bound of theorem 2.

We conclude the analysis with a log amortized F^{in} analysis and a communication complexity bound. The $\log n$ amortized F^{in} of our algorithm as shown in [MMR99] is 1.

In order to finish the analysis the communication complexity (which also bounds the required storage size) must be addressed. Each vertex $v \in V$ receives at most $O(b + \log \frac{n}{b})$ sets of paths. Paths are of length at most $\log \frac{n}{b}$. Therefore, the communication overhead per message can be bounded by $O(b + \log \frac{n}{b})^{\log \frac{n}{b}} = (\frac{n}{b})^{O(\log(b + \log n))}$.

This communication complexity can be enforced by good replicas even in the presence of faulty replicas. A good replica can simply verify that (a) the length of all paths in any incoming message does not exceed $\log \frac{n}{b}$, and that (b) the out-degree of any vertex does not exceed $O(b + \log \frac{n}{b})$. Any violation of (a) or (b) indicates that the message was sent by a faulty replica, and can be safely discarded.

6 Conclusions and future work

This paper presented a round-efficient algorithm for disseminating updates in a Byzantine environment. The protocol presented propagates updates within an expected $O(b + \lg n)$ rounds, which is shown to be optimal. Compared with previous methods, the efficiency here was gained at the cost of an increase in the size of messages sent in the protocol. Our main direction for future work is to reduce the communication complexity, which was cursorily addressed in the present work.

References

[BHO+99] K. P. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budio and Y. Minsky. Bimodal multicast. *ACM Transactions on Computer Systems* 17(2):41–88, 1999.

[BLNS82] A. D. Birrell, R. Levin, R. M. Needham, and M. D. Schroeder. Grapevine, An exercise in distributed computing. *Communications of the ACM* 25(4):260–274, 1982.

[BT85] G. Bracha and S. Toueg. Asynchronous consensus and broadcast protocols. *Journal of the ACM* 32(4):824–840, October 1985.

[CL99] M. Castro and B. Liskov. Practical Byzantine fault tolerance. In *Proceedings of the 3rd Symposium on Operating Systems Design and Implementation*, 1999.

[CASD95] F. Cristian, H. Aghili, R. Strong, and D. Dolev. Atomic broadcast: From simple message diffusion to Byzantine agreement. *Information and Computation* 18(1), pages 158–179, 1995.

[Dee89] S. E. Deering. Host extensions for IP multicasting. SRI Network Information Center, RFC 1112, August 1989.

[DGH+87] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. In *Proceedings of the 6th ACM Symposium on Principles of Distributed Computing*, pages 1–12, 1987.

[DS83] D. Dolev and R. Strong. Authenticated algorithms for Byzantine agreement. *SIAM Journal of Computing* 12(4):656–666, 1983.

[KMM98] K. P. Kihlstrom, L. E. Moser and P. M. Melliar-Smith. The SecureRing protocols for securing group communication. In *Proceedings of the 31st IEEE Annual Hawaii International Conference on System Sciences*, vol. 3, pages 317–326, January 1998.

[LOM94] K. Lidl, J. Osborne and J. Malcome. Drinking from the firehose: Multicast USENET news. In *Proceedings of the Usenix Winter Conference*, pages 33–45, January 1994.

[LSP82] L. Lamport, R. Shostak, and M. Pease. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems* 4(3):382–401, July 1982.

[MM95] L. E. Moser and P. M. Melliar-Smith. Total ordering algorithms for asynchronous Byzantine systems. In *Proceedings of the 9th International Workshop on Distributed Algorithms*, Springer-Verlag, September 1995.

[MMR99] D. Malkhi, Y. Mansour, and M. K. Reiter. On diffusing updates in a Byzantine environment. In *Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems*, pages 134–143, October 1999.

[MR97] D. Malkhi and M. Reiter. A high-throughput secure reliable multicast protocol. *Journal of Computer Security* 5:113–127, 1997.

[MRRS01] D. Malkhi, M. Reiter, O. Rodeh and Y. Sella. Efficient update diffusion in Byzantine environments. To appear in *Proceedings of the 20th IEEE Symposium on Reliable Distributed Systems*, 2001.

[MS01] Y. Minsky and F. B. Schneider. Tolerating malicious gossip. Private communication.

[Rei94] M. K. Reiter. Secure agreement protocols: Reliable and atomic group multicast in Rampart. In *Proceedings of the 2nd ACM Conference on Computer and Communications Security*, pages 68–80, November 1994.