The Universal Gossip Fighter

Anastasiia Gorbunova, Rachid Guerraoui, Anne-Marie Kermarrec, Anastasiia Kucherenko, Rafael Pinot

École Polytechnique Fédérale de Lausanne

***Abstract*—The notion of adversary is a staple of distributed computing. An adversary typically models "hostile" assumptions about the underlying distributed environment, e.g., a network that can drop messages, an operating system that can delay processes or an attacker that can hack machines. So far, the goal of distributed computing researchers has mainly been to develop a distributed algorithm that can face a given adversary, the abstraction characterizing worst-case scenarios.**

This paper initiates the study of the somehow opposite approach. Given a distributed algorithm, the adversary is the abstraction we seek to implement. More specifically, we consider the problem of controlling the spread of messages in a large-scale system, conveying the practical motivation of limiting the dissemination of fake news or viruses. Essentially, we assume a general class of gossip protocols, called all-to-all gossip protocols, and devise a practical method to hinder the dissemination.

We present the *Universal Gossip Fighter* (UGF). Just like classical adversaries in distributed computing, UGF can observe the status of a dissemination and decide to stop some processes or delay some messages. The originality of UGF lies in the fact that it is *universal*, i.e., it applies to any all-to-all gossip protocol. We show that *any* gossip protocol attacked by UGF ends up exhibiting a quadratic message complexity (in the total number of processes) if it achieves sublinear time of dissemination. We also show that if a gossip protocol aims to achieve a message complexity α times smaller than quadratic, then the time complexity rises exponentially in relation to α. We convey the practical relevance of our theoretical findings by implementing UGF and conducting a set of empirical experiments that confirm some of our results.

***Index Terms*—Distributed Computing, Gossip Protocols, Adaptive Adversaries**

I. Introduction

Gossip protocols (a.k.a. epidemic or rumor-spreading protocols) represent efficient means to disseminate information in a large-scale distributed system and have been long used to model the spread of infectious diseases [18] or rumors in social networks [10], [15]. Gossip protocols have also led to significant advances in the fields of distributed databases [9], group communication systems [24], decentralized machine learning [25] and optimization [8], [11].

The goal of researchers studying these protocols has mainly been to design strategies to disseminate messages as efficiently as possible to the entire network, despite failures of processes or communication links [5], [23]. But when information travels fast and without any control, the network becomes vulnerable to the quick spread of poisoned messages such as fake news in social networks or viruses in epidemic protocols. In this paper, we open a line of research studying the exact opposite approach. In other words, we ask the following question:

Can we hamper the spread of gossips?

We take a first step toward addressing this challenging question by studying a general class of *all-to-all* gossip protocols. In these protocols, every process starts with a unique gossip (piece of information) that it seeks to communicate to all other processes in the network. Clearly, every process could send its gossip to all the other processes in only 1 communication round. But this amounts to sending N^2 messages, which is extremely costly in a large-scale distributed system (large in the number N of processes). An efficient gossip protocol has to balance the number of communication rounds (*time complexity*) and the total number of messages sent by the processes (*message complexity*). This kind of protocols have been widely studied in the distributed computing literature [3], [4], [7], [13], [14]; hence they represent a good starting point for our study.

In this paper, we explore strategies to delay the dissemination of an all-to-all gossip protocol, either by directly extending the communication rounds or by forcing processes to send too many messages (i.e., by overloading them to the point where they can no longer contribute to the dissemination). Essentially, we aim at designing an adversarial strategy that forces any time-efficient gossip protocol to send $\Omega\left(N^2\right)$ messages; making it therefore inefficient in terms of message complexity.

To better understand our approach, it is useful to relate it to a classical one in the theory of distributed computing: seeking lower bounds and impossibility results, as in [14]. There in particular, the goal is to show that for any all-to-all gossip protocol, an adversary with the power of failing (some) processes and delaying messages, can prevent the disseminating of gossips in a time- and communication-efficient manner. The key impossibility result of [14] basically says that, for an asynchronous distributed system of N processes where F processes can fail and for any all-to-all gossip protocol, there exists an adaptive adversary (specific to the gossip protocol being studied) that forces the time complexity of the dissemination to be linear in F *or* the message complexity to be quadratic in F.

However, that line of research only focuses on demonstrating the existence of worst-case scenarios in which the spread of gossips can be inefficient. It does not present any evidence of whether these scenarios are probable or even realizable in practice. Therefore, establishing whether adaptive adversaries constitute a genuine point of vulnerability with practical consequences or if they are simply a mathematical

artifact of gossip protocols was yet to be demonstrated. More specifically, existing works did not provide any indication on whether there existed a universal strategy that can be used to slow information dissemination without having prior knowledge about the underlying gossip.

Summary of our contributions

In this paper, we close this gap by showing that adaptive adversaries are a genuine and concrete vulnerability of all-to-all gossip protocols. More specifically, we construct an adaptive adversary which we call the *Universal Gossip Fighter* (UGF), and which we show can effectively slow down the propagation of messages *regardless* of the underlying gossip protocol and without any prior knowledge about it. To the best of our knowledge, this is the first work presenting such an adversary.

To ensure the universality of UGF, i.e., the ability to harm *any* gossip protocol, we had to address a major technical challenge. The gossip protocol can be itself adaptive: namely, processes can change their behavior during the dissemination to avoid ending up with high message or time complexities. Intuitively, we had to design UGF in such a way that it hides its strategy, at least for some time. One of the key element of our technical contribution is to implant randomization schemes in UGF to prevent adaptation on the protocol side, ensuring thereby the universality of UGF.

By carefully analysing how randomization impacts the universality of UGF, we induce some important results on the strength of our adversary. Essentially, we show that if a gossip protocol aims to achieve a message complexity α times less than quadratic, then its time complexity increases exponentially with respect to α. This not only matches the worst-case scenario studied by previous works (when $\alpha = 1$), but also provides a more general result. This highlights a fundamental trade-off between the time efficiency and message efficiency of gossip protocols under adverse manipulation by UGF, quantified by α.

We empirically validate our theoretical results with a set of experiments demonstrating the disruptive power of UGF on existing all-to-all gossip protocols.

Outline of the paper

In Section II, we present the model setting we consider and introduce the main concepts we use in the paper. In Section III, we present the adaptive adversary we designed: the Universal Gossip Fighter. Sections IV and V present our main technical contributions showing the universality and disruptive power of UGF both theoretically and experimentally. Finally, we discuss prior work and present some concluding remarks in Section VI.

II. Model setting

A. System model

We consider a set of N crash-prone processes denoted by Π. The processes communicate within a fully-connected network: each of them can communicate with any other one via sending messages. Every process holds a unique gossip it aims to propagate throughout the network. Assuming time proceeds in discrete global steps, the execution model is as follows.

1) Local steps: The gossip protocol for each process proceeds in *local steps*. At the beginning of any local step, the process verifies if any messages were received from other processes and delivers them to its local memory. During the rest of the local step, based on any information the process has in memory, it decides on further messages to send. At the end of each local step, the process can send a subset of the gossips it holds or any additional information it gathered to an arbitrary number of other processes. What messages are sent and to whom, depends on the gossip protocol being applied.

2) Local step time: The length of local steps can vary between different processes. We denote by δ_ρ the duration of the local steps (a.k.a., local step time) for any $\rho \in \Pi$, and δ the maximal length of a local step for the whole system. δ is calculated when all processes in the system have stopped sending messages, i.e., $\delta := \max_{\rho\in\Pi} \delta_\rho$.

3) Delivery time: After being sent by a process, a message takes several global steps before reaching another process. This *delivery time* may also depend on the process which sends the message. We denote by d_ρ the delivery time of messages sent by any $\rho \in \Pi$ and d the maximal delivery time of the system, i.e., $d := \max_{\rho\in\Pi} d_\rho$.

4) Partial synchrony: We consider a *partially synchronous* system, i.e. the processes neither have access to the global clock nor to d and δ. However, we always have $d < \infty$ and $\delta < \infty$.

Remark 1. *In general, the length of local steps and the delivery times of a single process could vary in time. However, for presentation simplicity, we proceed with a scenario where, for each process, the local steps have a fixed length and the delivery time do not change during the dissemination.*

B. All-to-all gossip protocols

A gossip protocol is a predefined set of properties that orchestrates the behavior of every process at each local step. In this paper, we consider the general class of *all-to-all gossip protocols* that respect the following criteria:

Definition II.1 (Rumor gathering)**.** *Every process that did not crash during the dissemination (a.k.a. a* correct *process) should receive the gossip of all the other correct processes.*

Definition II.2 (Quiescence)**.** *For every process, there exists a point in time when it either crashes, or it* completes *in the sense that it stops sending messages forever.*

More formally, let us consider an arbitrary gossip protocol P. At any global step t, we denote by P_t the *state* of the system at t, when running the protocol P. This characterizes the information held by every process, the state of their local computations, as well as the messages they send at global step t and their destination. Then, an *outcome* of a gossip protocol P can be characterized by a sequence $(P_t)_{t\in[T_{\text{end}}]}$, where $T_{\text{end}} < \infty$ and $P_{T_{\text{end}}}$ is such that all correct processes

completed and received all correct gossips. In the remaining, we denote by $\mathrm{O}(P)$ the set of all outcomes for P. Finally, note that our analysis also covers gossip protocols that can be random and adaptive. To formalize this, we use the notion of *execution*. The execution of a gossip protocol P, denoted by EXE, is a random variable with values in $\mathrm{O}(P)$, where the randomness might come from the protocol itself or any random process that could influence the dissemination (e.g., an adversary). This random variable assigns probabilities to each of the outcomes of the gossip protocol.

C. Communication complexities: message and time

Let P be a gossip protocol, EXE the random variable characterizing the execution of P, and $\mathrm{O} \in \mathrm{O}(P)$ an outcome for P. We define the notions of message and time complexity below (commonly denoted as communication complexities).

Definition II.3 (Message complexity)**.**

1) *Let $\rho \in \Pi$, the message complexity of O for process ρ, denoted $M_\rho(\mathrm{O})$, is the total amount of messages sent by ρ during O, without taking into account their size (i.e., a message can include several gossips at once).*
2) *The overall message complexity of the outcome O, denoted $M(\mathrm{O})$ is the sum of the message complexities for every $\rho \in \Pi$, i.e., $M(\mathrm{O}) := \sum_{\rho \in \Pi} M_\rho(\mathrm{O})$.*
3) *Finally, we define the average message complexity of* EXE *as the expectation of the message complexity over the set of its possible outcomes, i.e., $\mathbb{E}\left[M(\mathrm{EXE})\right]$.*

The idea behind time complexity is to measure the minimum number of communication rounds that the outcome takes before all correct processes complete. To compute it, we take the number of global time steps that have passed during O and normalize it by the maximum number of global steps that a message can take to pass from one process to another. More formally, we define time complexity as follows.

Definition II.4 (Time complexity)**.** *Let us denote by $T_{end}(\mathrm{O})$ the first global step at which all correct processes completed during the outcome O.*

1) *We define time complexity of O as $T(\mathrm{O}) := \frac{T_{end}(\mathrm{O})}{\delta + d}$, where d and δ respectively are the maximum delivery and local step times of the system during O.*
2) *Similarly to message complexity, the average time complexity of* EXE *is defined as $\mathbb{E}\left[T(\mathrm{EXE})\right]$.*

D. Adaptive adversary

To reason about the communication complexities of a gossip protocol in a crash-prone setting, the classical approach is to consider the notion of *adversary*: this is an abstract notion that models situations that can slow the diffusion of the message by delaying communications or crashing processes. In this work, we are particularly interested in the notion of adaptive adversary, defined below.

Definition II.5 (Adaptive adversary)**.** *An adaptive adversary can delay messages by modifying d_ρ and δ_ρ for any $\rho \in \Pi$ and can crash up to $F < N$ processes in an online fashion. This means that at each global step t, the adversary has access to the system state P_t and can decide accordingly which processes to crash and which messages to delay.*

In the distributed computing literature, the notion of adaptive adversary has been used to study the intrinsic efficiency of gossip protocols in crash-prone asynchronous systems (see e.g., [12], [14]). In this work, we take a new approach to this notion by studying the universality of an adaptive adversary, i.e., its ability to harm the effectiveness of any all-to-all gossip protocol. To this end, we introduce a new adversary (the *Universal Gossip Fighter*) and show that it can effectively slow down the dissemination of gossips regardless of the protocol it attacks.

III. THE UNIVERSAL GOSSIP FIGHTER

In this section, we design an adaptive adversary called the *Universal Gossip Fighter* (UGF). UGF is a centralized algorithm that can monitor the dissemination of the gossips in an *on-line fashion* by observing the state of the system at every step t. Using this information, UGF can modify the delivery or local step times of processes and crash up to $F \leq N$ of them. Before diving into the design of UGF let us first present some intuitions of what would be a desirable outcome for such an adversary.

A. Objective of the adversary

Essentially, the objective of our adversary is to render a gossip protocol inefficient. The rationale of what we mean by inefficiency in this paper, is illustrated in an example below.

Example 1. *Consider the gossip protocol P where every process sorts the other processes and sends its gossip to one process per step during $N-1$ steps (following the order it created). For any outcome O of this protocol (which is deterministic), we can show that $M(\mathrm{O}) = \Theta\left(N^2\right)$ and $T(\mathrm{O}) = \Theta\left(N\right)$.*

Note that, in the absence of an adversary, there exists a deterministic gossip protocol working with $\mathcal{O}(\log^3 N)$ time complexity and sending $\mathcal{O}(N \log^4 N)$ messages [7]. Hence, the above protocol is arguably inefficient both in terms of message and time complexity. Note also that there is no point in aiming for more than quadratic message complexity, as it can always be achieved by broadcasting gossips to everyone at the first communication round, as explained in Section I. Therefore, we hereafter consider quadratic message and linear time complexities as a basis for inefficiency of a dissemination. Specifically, we seek to design an adaptive adversary capable of forcing any all-to-all gossip protocol to have either linear time or quadratic message complexity.

B. Intuitions and algorithm design

The main challenge when designing UGF is to guarantee the universality of the algorithm, i.e., its applicability to *any* all-to-all gossip protocol. Since these protocols can be adaptive, they may change their behavior during the dissemination to avoid ending up with too high message or time complexities.

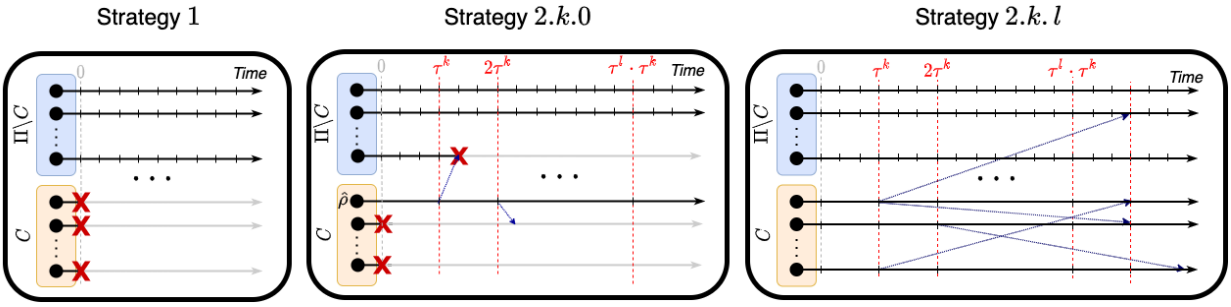

Fig. 1: Illustration of UGF's strategies.

To ensure the universality of our adversary even in this context, we decompose it into several adversarial strategies that we implement within a randomized scheme, illustrated in Figure 2. This scheme not only prevents adaptation, but also ensures success on average, as each implemented strategy is only useful against a specific type of gossip protocols.

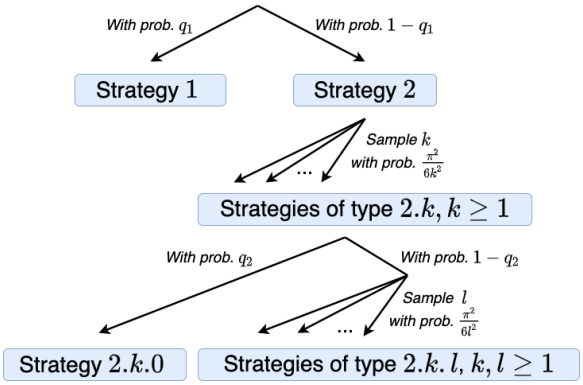

Fig. 2: Randomization scheme for UGF.

In short, UGF first divides Π into two disjoint sets: C and $\Pi \setminus$ C, where C has cardinality $\Theta(F)$. This step is necessary to separate processes over which UGF aims to have active control (C) from those it will not actively disrupt ($\Pi \setminus$ C). Then it implements three kind of strategies succinctly described below and illustrated in Figure 1.

(a) **Fig. 1 left:** *When processes in* $\Pi \setminus$ C *communicate slowly* (do not send a lot of messages per communication round). It is then sufficient to crash the processes in C (Strategy 1) to obtain a high time complexity.

(b) **Fig. 1 middle:** *When processes in* C *communicate slowly*. The adversary can then isolate a process in C by crashing all the processes it tries to communicate with (Strategy $2.k.0$ with $k \geq 1$). If the algorithm succeeds in isolating this process long enough, it forces a high time complexity.

(c) **Fig. 1 right:** *When processes in* C *communicate quickly*. In this context, the adversary cannot simply crash message receivers from C because it would quickly consume its crash budget F. Instead, it can delay messages from C in order to force the processes to send a large amount of messages, thus achieving high message complexity (Strategies $2.k.l$, with $k, l \geq 1$).

Algorithm 1: The Universal Gossip Fighter (UGF)

Require: Π {*Set of processes*}; F {*Maximal number of crashes*}; q_1, q_2 {*Probability parameters in* $(0,1)$}; τ {*Delay parameter in* $\mathbb{N}^*$}.

Correct $\leftarrow \Pi$; Crash$_{\text{count}} \leftarrow 0$;
C $\leftarrow$ a random sample of $F/2$ processes from Π;
for $\rho \in \Pi$ **do**
  $d_\rho \leftarrow 1$; $\delta_\rho \leftarrow 1$;
With *prob.* q_1 **do** // Str. 1
  for $\rho \in$ C **do**
    Crash(ρ);
Otherwise *with prob.* $1 - q_1$ **do**
  Sample k at random from $\mathbb{N}^*$ with prob. $\frac{6}{k^2 \cdot \pi^2}$;
  for $\rho \in$ C **do** // Choose k to apply Str. 2.k
    $\delta_\rho \leftarrow \tau^k$;
  With *prob.* q_2 **do** // Str. 2.k.0
    Sample $\hat{\rho}$ at random from C;
    for $\rho \in$ C $\setminus \hat{\rho}$ **do**
      Crash(ρ);
      Crash$_{\text{count}} \leftarrow$ Crash$_{\text{count}} + 1$;
    Correct $\leftarrow$ ($\Pi \setminus$ C) $\cup \hat{\rho}$;
    while (*Crash$_{count}$* $< F$) **do**
      if ($\hat{\rho}$ *sends a message to* ρ *at global step* t) ***and*** ($\rho \in$ *Correct*) **then**
        Crash(ρ);
        Crash$_{\text{count}} \leftarrow$ Crash$_{\text{count}} + 1$;
        Correct $\leftarrow$ Correct$\setminus \rho$;
  Otherwise *with prob.* $1 - q_2$ **do** // Str.2.k.l,l>0
    Sample l at random from $\mathbb{N}^*$ with prob. $\frac{6}{l^2 \cdot \pi^2}$;
    for $\rho \in$ C **do**
      $d_\rho \leftarrow \tau^k \cdot \tau^l$;

The parameters q_1, q_2 define the probabilities of applying strategies of type 1 (w.p. q_1), $2.k.0$ (w.p. $(1 - q_1)q_2$) or $2.k.l, l > 0$ (w.p. $(1 - q_1)(1 - q_2)$). Note that, we show in Section IV that UGF ensures disruption of the dissemination with any choice of q_1, q_2. However, one may tune these parameters to change the probability of applying some specific

strategies, e.g. if there is prior knowledge about the gossip protocol to attack. Without prior knowledge, the safe choice is to make all these strategies equiprobable, by taking $q_1 = 1/3$ and $q_2 = 1/2$, as we do in our experimental section.

The precise instantiating of these different strategies, as well as the randomization scheme used by UGF are formally described in Algorithm 1.

Remark 2. *The probabilities* $\{\frac{\pi^2}{6k^2}, k > 0\}$ *are used to guarantee indistinguishability (see the next section for more details) between the strategies* $\{2.k.l, k \geq 0, l \geq 0\}$. *In fact, any other infinite sequence summing to* 1 *would provide similar results.*

IV. Analysis

In this section, we present our main technical contributions. We first introduce, in Section IV-A, a formal framework to analyze how randomization can help hiding the changes that UGF operates on the system. Then, we present in Section IV-B our main result on the disruptive power of our adversary on any all-to-all gossip protocol. Note that UGF depend on two probability parameters (q_1, q_2) and a delay parameter τ. Hereafter, we refer to UGF without specifying these parameters when providing intuitions about the method and only discuss them explicitly when necessary.

A. Randomization prevents adaptation

As we previously mentioned in Section III-B, one of the main building block of our algorithm is the use of randomization to prevent the gossip protocol from adapting; hence guaranteeing the universality of the procedure. To formalize this idea, we introduce below the notion of *indistinguishability*.

Definition IV.1 (Indistinguishability)**.** *Let us consider* $\rho \in \Pi$, P *an all-to-all gossip protocol and* Exe *the execution of* P *under attack by* UGF.

1) *Two events* E *and* E' *are said to be indistinguishable to* ρ *during a global time frame* $[t_1, t_2]$ *if the actions of* ρ *during this time frame are equally likely to be conditioned on* E *or* E'.
2) *If two events are indistinguishable to every* ρ *in a group during the same period of time, then we say that these events are indistinguishable to this group.*

The notion of indistinguishability is particularly useful, especially to characterize the behavior of some processes when they do not receive any messages from the rest of the system. In this context, they can stop sending messages even if the system has not completed yet. To formalize this notion, we introduce below the concept of *falling asleep*.

Definition IV.2 (Falling asleep)**.** *A process falls asleep if it stops sending messages until it delivers a message from another process. If a sleeping process delivers a new message, it can* wake up *and start gossiping again. Otherwise, the moment it falls asleep is also the moment it completes.*

We now present some technical lemmas demonstrating the indistinguishability of several strategies applied by UGF. These lemmas will essentially be used in the proof of our main result to establish that in some specific time frames, the gossip protocol will not be able to modify its dissemination to counter UGF.

Lemma 1. *Let* P *be an all-to-all gossip protocol,* Exe *the execution of* P *under attack by* UGF *with delay parameter* $\tau > 1$ *and* ρ *a process in* $\Pi\backslash$C. *Then, for any integers* $k \geq 1$ *and* $l \geq 0$, *the events* $E =$"UGF *applies Strategy* 1" *and* $E' =$"UGF *applies Strategy* $2.k.l$" *are indistinguishable to* ρ *during the global time frame* $[1, \tau^k]$.

Proof. First, regardless of whether we condition on E or E', no message from C can be delivered to ρ before the global time step τ^k. In fact, C is either crashed (when conditioning to E) or no message from C can be delivered before the global step τ^k, i.e., before the end of the first local step for C (when conditioning on E').

Second, for both E and E', the local step and delivery times are equal to 1 for $\Pi\backslash$C during $[1, \tau^k]$ and no communication there is affected. Accordingly, E and E' have identical impact on $\Pi\backslash$C and in particular on ρ during the global time frame $[1, \tau^k]$. Hence ρ's actions during $[1, \tau^k]$ are equally likely to be conditioned on E or E'. □

Lemma 2. *Let* P *be an all-to-all gossip protocol,* Exe *the execution of* P *under attack by* UGF *with delay parameter* $\tau > 1$ *and* ρ *a process in* $\Pi\backslash$C. *Then for any integers* $k_1 \geq k_2 \geq 1$ *and* $i_1, i_2 \geq 0$, *the events* $E =$"UGF *applies Strategy* $2.k_1.i_1$" *and* $E' =$"UGF *applies Strategy* $2.k_2.i_2$" *are indistinguishable to* ρ *during the global time frame* $[1, \tau^{k_2}]$.

Proof. Let us consider the event $E^* =$"UGF applies Strategy 1". From Lemma 1 we know that ρ's actions during $[1, \tau^{k_1}]$ are equally likely to be conditioned on E or E^*. Still using Lemma 1 we get the same conclusion for E' or E^* during $[1, \tau^{k_2}]$. Finally, since $[1, \tau^{k_2}] \subset [1, \tau^{k_1}]$ we obtain that E and E' are indistinguishable to ρ during $[1, \tau^{k_2}]$. □

Lemma 3. *Let* P *be an all-to-all gossip protocol,* Exe *the execution of* P *under attack by* UGF *with delay parameter* $\tau > 1$ *and* ρ *a process in* C. *For any integers* $k \geq 1$ *and* $l \geq 1$, *let us define* $E =$"UGF *applies Strategy* $2.k.0$ *on* ρ", $E' =$"UGF *applies Strategy* $2.k.l$" *and* $E'' =$"$\Pi\backslash$C *falls asleep before global step* τ^k". *Then* $E \cap E''$ *and* $E' \cap E''$ *are indistinguishable to* ρ *during global time frame* $[1, \min\{t_{F/2}, \tau^{k+l}\}]$, *where* $t_{F/2}$ *is the the first global step at which* ρ *sent in total more than* $\frac{F}{2}$ *messages.*

Proof. When conditioning to $E \cap E''$, no message from C can be delivered before $t_{F/2}$. Indeed, ρ is the only non-crashed process from C and the receivers of its $F/2$ first messages are crashed by UGF. Similarly, when conditioning to $E' \cap E''$ no message from C can be delivered before $\tau^k \cdot \tau^l$ by construction of Strategy $2.k.l$. Therefore, both when conditioning to $E \cap E''$ and $E' \cap E''$, no process in Π delivers messages from C during the global time frame $[1, \min\{t_{F/2}, \tau^k \cdot \tau^l\}]$.

Now note that, thanks to Lemma 2, we know that messages coming to ρ from $\Pi\backslash$C are equally likely to be sent when

conditioning on E or E' during the global time frame $[1, \tau^k]$. Furthermore, in both $E \cap E''$ and $E' \cap E''$, after global step τ^k, processes in $\Pi \backslash \mathrm{C}$ are asleep and do not wake up since no message from C is delivered before $[1, \min\{t_{F/2}, \tau^l \cdot \tau^k\}]$.

Accordingly, messages send by ρ during the global time frame $[1, \min\{t_{F/2}, \tau^l \cdot \tau^k\}]$ are equally likely to be delivered when conditioning on $E \cap E''$ or $E' \cap E''$. Hence $E \cap E''$ and $E' \cap E''$ are indistinguishable to ρ during this time frame. □

To conclude this section, we compute the probability of UGF applying a strategy of type $2.k$ or $2.k.l$ below. To do so, and for the rest of the paper, we denote by O_i the set of outcomes of P where UGF applies Strategy i.

Lemma 4. *Let P be an all-to-all gossip protocol and* EXE *the execution of P under attack by* UGF *with probability parameters $q_1, q_2 \in (0,1)$ and delay parameter $\tau > 1$. Then for any integer $j \geq 0$ and $t > 1$ with probability at least $\frac{(1-q_1)\cdot 6}{\pi^2 \cdot \lceil \log_\tau t \rceil}$,* UGF *applies a strategy $2.k$ with $\tau^k \geq t$.*

Proof. By construction of UGF, the probability that UGF applies a strategy $2.k$ with $\tau^k \geq t$ is

$$\sum_{k \geq 1} \mathbb{P}\left[\mathrm{o} \in \mathrm{O}_{2.k} \cap \tau^k \geq t\right] = \sum_{k \geq \lceil \log_\tau t \rceil} \frac{(1-q_1)\cdot 6}{\pi^2 \cdot k^2} \tag{1}$$

$$= \frac{(1-q_1)\cdot 6}{\pi^2} \cdot \sum_{k \geq \lceil \log_\tau t \rceil} \frac{1}{k^2}. \tag{2}$$

Furthermore, as $k+1 \geq k$ we get

$$\sum_{k \geq 1} \mathbb{P}\left[\mathrm{o} \in \mathrm{O}_{2.k} \cap \tau^k \geq t\right]$$

$$\geq \frac{(1-q_1)\cdot 6}{\pi^2} \cdot \left(\sum_{k \geq \lceil \log_\tau t \rceil} \frac{1}{k} \cdot \frac{1}{k+1} \right) \tag{3}$$

$$= \frac{(1-q_1)\cdot 6}{\pi^2} \cdot \left(\sum_{k \geq \lceil \log_\tau t \rceil} \left(\frac{1}{k} - \frac{1}{k+1} \right) \right). \tag{4}$$

By developing the above telescopic sum, we obtain

$$\sum_{k \geq 1} \mathbb{P}[\mathrm{o} \in \mathrm{O}_{2.k} \cap \tau^k \geq t] \geq \frac{(1-q_1)\cdot 6}{\pi^2 \cdot \lceil \log_\tau t \rceil}. \tag{5}$$

□

Lemma 5. *Let P be an all-to-all gossip protocol and* EXE *the execution of P under attack by* UGF *with probability parameters $q_1, q_2 \in (0,1)$ and delay parameter $\tau > 1$. Then, for any integers $k \geq 1$ and $t > 1$, if we know that* UGF *applies a strategy $2.k$, then with probability at least $\frac{(1-q_2)\cdot 6}{\pi^2 \cdot \lceil \log_\tau t \rceil}$, it applies a strategy $2.k.l$ with $\tau^l \geq t$.*

Proof. Similarly to the proof of Lemma 4 we can write this probability as

$$\sum_{l \geq 1} \mathbb{P}\left[\mathrm{o} \in \mathrm{O}_{2.k.l} \cap \tau^l \geq t \middle| \, \mathrm{o} \in \mathrm{O}_{2.k}\right] = \sum_{l \geq \lceil \log_\tau t \rceil} \frac{(1-q_2)\cdot 6}{\pi^2 \cdot l^2}$$

And by decomposing as in (2) to (5) we get

$$\sum_{l \geq 1} \mathbb{P}\left[\mathrm{o} \in \mathrm{O}_{2.k.l} \cap \tau^l \geq t \middle| \, \mathrm{o} \in \mathrm{O}_{2.k}\right] \geq \frac{(1-q_2)\cdot 6}{\pi^2 \cdot \lceil \log_\tau t \rceil}.$$

□

B. The universal disruptive power of UGF

We now demonstrate our main result below. In a nutshell, we show that, for any α and for any all-to-all gossip protocol, UGF forces the average time complexity of the execution to be in $\Omega(\alpha F)$ or the average message complexity to be in $\Omega\left(N + {F^2}/{\log_\tau^2(\alpha F)}\right)$. When $F = \Theta(N)$, $\tau = F$, and $\alpha = 1$ it means that no time-efficient protocol (sublinear in N) can have message complexity less than quadratic in N.

Theorem 1. *Let P be an all-to-all gossip protocol, and* EXE *the execution of P under attack by* UGF *with probability parameters $q_1, q_2 \in (0,1)$ and delay parameter $\tau > 1$. Then for any integer $\alpha \geq 1$,* UGF *can force either*

$$\mathbb{E}[T(\textsc{Exe})] = \Omega(\alpha F) \text{ or } \mathbb{E}[M(\textsc{Exe})] = \Omega\left(N + {F^2}/{\log_\tau^2(\alpha F)}\right).$$

Proof. In the following, we develop a case-based reasoning that both depends on the strategy being applied and on the local behavior of the processes. In all cases, we either show that $\mathbb{E}[T(\textsc{Exe})] = \Omega(\alpha F)$ or $\mathbb{E}[M(\textsc{Exe})] = \Omega\left(N + {F^2}/{\log_\tau^2(\alpha F)}\right)$.

Part 1. For any outcome O, we denote by t^*_{O} the first global step at which all processes in $\Pi \backslash \mathrm{C}$ fall asleep for the first time. t^*_{O} exists for every O, because otherwise the quiescence property is violated. We also denote by

$$\mathrm{R}_1 := \mathbb{P}[T(\mathrm{O}) = \Omega(\alpha F)],$$

the probability that an outcome has time complexity in $\Omega(\alpha F)$. In this first part, we consider the following case.

Case (i). $\mathbb{P}[t^*_{\mathrm{O}} \geq \alpha F | \, \mathrm{O} \in \mathrm{O}_1] \geq 1/2$.

Observation 1. *Note that for any $\mathrm{O} \in \mathrm{O}_1$, the set C is crashed from the beginning; hence $t^*_{\mathrm{O}} = T_{end}(\mathrm{O})$ is also the first global step at which all processes completed. Furthermore, when $\mathrm{O} \in \mathrm{O}_1$, we have $d = \delta = 1$; hence $T(\mathrm{O}) = t^*_{\mathrm{O}}/2$.*

Using Observation 1 and the Bayes rule we get:

$$\begin{aligned} \mathrm{R}_1 &\geq \mathbb{P}[T(\mathrm{O}) = \Omega(\alpha F) \cap \mathrm{O} \in \mathrm{O}_1] \\ &\geq \mathbb{P}[t^*_{\mathrm{O}} = \Omega(\alpha F) \cap \mathrm{O} \in \mathrm{O}_1] \\ &\geq \mathbb{P}[t^*_{\mathrm{O}} \geq \alpha F | \, \mathrm{O} \in \mathrm{O}_1] \cdot \mathbb{P}[\mathrm{O} \in \mathrm{O}_1]. \end{aligned}$$

Finally, by construction of UGF, we have $\mathbb{P}[\mathrm{O} \in \mathrm{O}_1] = q_1$, hence using the definition of Case (i) we get $\mathrm{R}_1 \geq 1/2 \cdot q_1$.

Conclusion of part 1. In the Case (i), the average time complexity of EXE is greater than $1/2 \cdot q_1 \cdot \alpha F$ which is in $\Omega(\alpha F)$ because q_1 is a fixed constant.

Part 2. We now consider the case where (i) does not hold true, i.e., we consider the following:

Case (ii). $\mathbb{P}\left[t^*_{\mathrm{o}} < \alpha F | \, \mathrm{o} \in \mathrm{O}_1\right] \geq 1/2$.

When considering Case (ii), we focus on the outcomes where $\Pi \backslash \mathrm{C}$ fall asleep before the end of the first local step of C. This enable us to conclude that, during a given time-frame, no messages from $\Pi \backslash \mathrm{C}$ arrives to C; hence simplifying the analysis. To do so, for any integers $l \geq 0$ and $k \geq 1$ we define the event

$$E_{k.l} = \{t^*_{\mathrm{o}} < \tau^k \ \cap \mathrm{o} \in \mathrm{O}_{2.k.l} \cap \tau^k \geq \alpha F\}.$$

We can compute the probability of such an event for any $l \geq 0$ and $k \geq 1$ as follows:

$$\begin{aligned}\mathbb{P}\left[E_{k.l}\right] &= \mathbb{P}\left[t^*_{\mathrm{o}} < \tau^k | \, \mathrm{o} \in \mathrm{O}_{2.k.l} \cap \tau^k \geq \alpha F\right] \cdot \\ &\quad \cdot \mathbb{P}\left[\mathrm{o} \in \mathrm{O}_{2.k.l} \,|\, \mathrm{o} \in \mathrm{O}_{2.k}\right] \cdot \mathbb{P}\left[\mathrm{o} \in \mathrm{O}_{2.k} \cap \tau^k \geq \alpha F\right] \qquad (6)\\ &\geq \mathbb{P}\left[t^*_{\mathrm{o}} < \alpha F | \, \mathrm{o} \in \mathrm{O}_{2.k.l} \cap \tau^k \geq \alpha F\right] \\ &\quad \cdot \mathbb{P}\left[\mathrm{o} \in \mathrm{O}_{2.k.l} \,|\, \mathrm{o} \in \mathrm{O}_{2.k}\right] \cdot \mathbb{P}\left[\mathrm{o} \in \mathrm{O}_{2.k} \cap \tau^k \geq \alpha F\right]\end{aligned}$$

Finally, by using Lemma 1 we get

$$\begin{aligned}\mathbb{P}\left[E_{k.l}\right] &\geq \mathbb{P}\left[t^*_{\mathrm{o}} < \alpha F | \, \mathrm{o} \in \mathrm{O}_1\right] \\ &\quad \cdot \mathbb{P}\left[\mathrm{o} \in \mathrm{O}_{2.k.l} \,|\, \mathrm{o} \in \mathrm{O}_{2.k}\right] \cdot \mathbb{P}\left[\mathrm{o} \in \mathrm{O}_{2.k} \cap \tau^k \geq \alpha F\right].\end{aligned}$$

Furthermore, note that by construction of UGF, the probability $\mathbb{P}\left[\mathrm{o} \in \mathrm{O}_{2.k.l} \,|\, \mathrm{o} \in \mathrm{O}_{2.k}\right]$ does not depend on k. Then using the definition of Case (ii) and Lemma 4 we can lower bound the sum of these events, as follows

$$\sum_{k\geq 1} \mathbb{P}\left[E_{k.l}\right] \geq \frac{(1-q_1)\cdot 3}{\pi^2 \cdot \lceil \log_\tau \alpha F \rceil} \cdot \mathbb{P}\left[\mathrm{o} \in \mathrm{O}_{2.k.l} \,|\, \mathrm{o} \in \mathrm{O}_{2.k}\right]. \quad (7)$$

Now, for any $\rho \in \mathrm{C}$, $k \geq 1$ and $\mathrm{o} \in \mathrm{O}_{2.k}$, we denote by $t_{F/2}(\rho, \mathrm{o})$ the first global step at which ρ sent more than $F/2$ messages during o.

Observation 2. *Note that for any $\rho \in \mathrm{C}$, $k \geq 1$ and $\mathrm{o} \in \mathrm{O}_{2.k.0}$, if $t_{F/2}(\rho, \mathrm{o})$ is bigger than $\alpha F \lceil \log_\tau(\alpha F)\rceil \cdot \tau^k$, then no process will receive the gossip of ρ before the global step $\alpha F \cdot \lceil \log_\tau(\alpha F)\rceil \cdot \tau^k$. Hence we have $T_{end}(\mathrm{o}) \geq \alpha F \cdot \lceil \log_\tau(\alpha F)\rceil \cdot \tau^k$. Furthermore, when $\mathrm{o} \in \mathrm{O}_{2.k.0}$, we have $\delta = \tau^k$ and $d = 1$; hence we get $T(\mathrm{o}) = {}^{T_{end}(\mathrm{o})}/_{(\tau^k+1)} \in \Omega\left(\alpha F \cdot \lceil \log_\tau(\alpha F)\rceil\right)$.*

To quantify the probability of $t_{F/2}$ being as described in Observation 2, we introduce the following probability:

$$\mathrm{R}_{\rho,k} := \mathbb{P}\left[t_{F/2}(\rho, \mathrm{o}) \geq \alpha F \cdot \lceil \log_\tau \alpha F \rceil \cdot \tau^k | E_{k.0} \cap \hat{\rho} = \rho\right].$$

We also denote by G be the group of processes that spread their messages slowly under attack by a strategy $2.k.0$, i.e,

$$G := \left\{ \rho \ \text{ s.t.} \sum_{k\geq 1} \mathrm{R}_{\rho,k} \cdot \mathbb{P}\left[E_{k.0}\right] \geq 1/2 \cdot \sum_{k\geq 1} \mathbb{P}\left[E_{k.0}\right] \right\}.$$

Part 2.a. We first study the sub-case where G represents more than one half of C.

Sub-case $(ii.a)$. $|G| \geq {}^{|C|}/_2$.

This means that for any integer $k \geq 1$ the probability of sampling ρ from G when applying Strategy $2.k.0$ is at least $1/2$. Let us now compute a lower bound on the probability

$$\mathrm{R}_2 := \mathbb{P}\left[T(\mathrm{o}) = \Omega(\alpha F \cdot \lceil \log_\tau(\alpha F)\rceil)\right].$$

Thanks to Observation 2 we can lower bound R_2 by

$$\sum_{k\geq 1} \mathbb{P}\left[t_{F/2}(\hat{\rho}, \mathrm{o}) \geq \alpha F \cdot \lceil \log_\tau(\alpha F)\rceil \cdot \tau^k \cap E_{k.0} \cap \hat{\rho} \in G\right].$$

Using the Bayes rule, we also get

$$\begin{aligned}\mathrm{R}_2 &\geq \sum_{k\geq 1} \Big(\mathbb{P}[t_{F/2}(\hat{\rho}, \mathrm{o}) \geq \alpha F \cdot \lceil \log_\tau(\alpha F)\rceil \cdot \tau^k | E_{k.0} \cap \hat{\rho} \in G] \\ &\qquad \cdot \mathbb{P}[E_{k.0}] \cdot \mathbb{P}[\hat{\rho} \in G] \Big) \\ &\geq \sum_{k\geq 1} \mathrm{R}_{\rho,k} \cdot \mathbb{P}[E_{k.0}] \cdot \mathbb{P}[\hat{\rho} \in G] \geq \frac{1}{4} \sum_{k\geq 1} \mathbb{P}[E_{k.0}].\end{aligned}$$

Finally, by instantiating (7) with $l = 0$ we get

$$\mathrm{R}_2 \geq \frac{3/4 \cdot (1-q_1) \cdot q_2}{\pi^2 \cdot \lceil \log_\tau \alpha F \rceil}.$$

Conclusion of part 2.a. In the conjunction of Cases (ii) and $(ii.a)$ the average time complexity of EXE is greater than $\frac{3/4\cdot(1-q_1)\cdot q_2}{\pi^2 \cdot \lceil \log_\tau \alpha F \rceil} \cdot \alpha F \cdot \lceil \log_\tau(\alpha F)\rceil$ which is in $\Omega\left(\alpha F\right)$ because q_1 and q_2 are fixed constants.

Part 2.b. We now study the sub-case where G represents less than one half of C.

Sub-case $(ii.b)$. $|G| < {}^{|C|}/_2$.

Then for any ρ in $\mathrm{C} \setminus G$ we have

$$\begin{aligned}&\sum_{k\geq 1} \Big(\mathbb{P}\left[t_{F/2}(\rho, \mathrm{o}) < \alpha F \cdot \lceil \log_\tau(\alpha \cdot F)\rceil \cdot \tau^k | E_{k.0} \cap \hat{\rho} = \rho\right] \\ &\qquad \cdot \mathbb{P}[E_{k.0}] \Big) \\ &= \sum_{k\geq 1} (1 - \mathrm{R}_{\rho,k}) \cdot \mathbb{P}[E_{k.0}] \geq \frac{1}{2} \cdot \sum_{k\geq 1} \mathbb{P}[E_{k.0}]. \qquad (8)\end{aligned}$$

Using Lemma 3 and the fact that for any integers $k \geq 1$ and $l \geq 0$ the event "$\Pi \backslash \mathrm{C}$ fall asleep before global step τ^k" is included in $E_{k.l}$, we conclude that for any ρ the events $E_{k.0}$ and $E_{k.l}$ are indistinguishable during the time frame $[1, \min\{t_{F/2}(\rho, \mathrm{o}), \tau^{k+l}\}]$. Therefore

$$\begin{aligned}&\mathbb{P}\left[t_{F/2}(\rho, \mathrm{o}) < \tau^{k+l} \mid E_{k.l} \cap \tau^l \geq \alpha F \cdot \lceil \log_\tau(\alpha F)\rceil\right] \\ &\geq \mathbb{P}[t_{F/2}(\rho, \mathrm{o}) < \alpha F \cdot \lceil \log_\tau(\alpha \cdot F)\rceil \cdot \tau^k | \\ &\qquad E_{k.l} \cap \tau^l \geq \alpha F \cdot \lceil \log_\tau(\alpha F)\rceil] \\ &\geq \mathbb{P}\left[t_{F/2}(\rho, \mathrm{o}) < \alpha F \cdot \lceil \log_\tau(\alpha \cdot F)\rceil \cdot \tau^k | E_{k.0} \cap \hat{\rho} = \rho\right]. \qquad (9)\end{aligned}$$

Furthermore, thanks to (6) and by construction of UGF, we know that for any integer $l \geq 1$, the ratio ${}^{\mathbb{P}[E_{k.l}]}/_{\mathbb{P}[E_{k.0}]} = {}^{\mathbb{P}[\mathrm{o}\in \mathrm{O}_{2.k.l} \,|\, \mathrm{o}\in \mathrm{O}_{2.k}]}/_{\mathbb{P}[\mathrm{o}\in \mathrm{O}_{2.k.0} \,|\, \mathrm{o}\in \mathrm{O}_{2.k}]}$ does not depend on k.

Hence, combining (8) and (9) we get the following for any ρ in $\mathrm{C} \setminus G$:

$$\sum_{k\geq 1} \left(\mathbb{P}\left[t_{F/2}(\rho, \mathrm{O}) < \tau^{k+l} \mid E_{k.l} \cap \tau^l > \alpha F \cdot \lceil \log_\tau(\alpha F) \rceil\right] \cdot \mathbb{P}[E_{k.l}]\right) \geq \frac{1}{2} \sum_{k\geq 1} \mathbb{P}[E_{k.l}]. \quad (10)$$

Thanks to (10) we can now compute a lower bound on the average message complexity of EXE under Cases (ii) and $(ii.b)$. We first note that, by definition of message complexity we have

$$\mathbb{E}\left[M(\text{EXE})\right] = \sum_{\rho\in\Pi} \mathbb{E}\left[M_\rho(\text{EXE})\right] \geq \sum_{\rho\in \mathrm{C}\setminus G} \mathbb{E}\left[M_\rho(\text{EXE})\right].$$

Furthermore, we also have

$$\mathbb{E}\left[M(\text{EXE})\right] \geq \sum_{\substack{\rho\in \mathrm{C}\setminus G \\ k,l\geq 1}} \frac{F}{2} \cdot \mathbb{P}\left[M_\rho(\mathrm{O}) \geq F/2 \cap \mathrm{O} \in \mathrm{O}_{2.k.l}\right].$$

By definition of $t_{F/2}$, if $t_{F/2}(\rho, \mathrm{O}) \leq \infty$, then $M_\rho(\mathrm{O}) \geq F/2$. Furthermore, note that the event $\mathrm{O} \in \mathrm{O}_{2.k.l}$ is included in $E_{k.l}$. Hence $\mathbb{E}\left[M(\text{EXE})\right]$ is lower bounded by the following

$$\frac{F}{2} \cdot \sum_{\substack{\rho\in \mathrm{C}\setminus G \\ k,l\geq 1}} \mathbb{P}\left[t_{F/2}(\rho, \mathrm{O}) < \tau^{k+l} \cap E_{k.l} \cap \tau^l \geq \alpha F \cdot \lceil \log_\tau(\alpha F) \rceil\right]$$

Using Bayes rule and decomposing the sum gives the following lower bound for $\mathbb{E}\left[M(\text{EXE})\right]$:

$$\frac{F}{2} \cdot \sum_{\substack{\rho\in \mathrm{C}\setminus G \\ l\geq 1}} \Big(\mathbb{P}\left[\tau^l > \alpha F \cdot \lceil \log_\tau(\alpha F) \rceil\right] \cdot \sum_{k\geq 1} \big(\mathbb{P}[E_{k.l}] \cdot \mathbb{P}\left[t_{F/2}(\rho, \mathrm{O}) < \tau^{k+l} \mid E_{k.l} \cap \tau^l > \alpha F \cdot \lceil \log_\tau(\alpha F) \rceil\right] \big) \Big)$$

Using (10) and the fact that $\mid C \setminus G \mid > \frac{|C|}{2} > \frac{F}{4}$, we get

$$\geq \frac{F^2}{8} \cdot \sum_{l\geq 1} \left(\mathbb{P}\left[\tau^l > \alpha F \cdot \lceil \log_\tau(\alpha F) \rceil\right] \cdot \frac{1}{2} \sum_{k\geq 1} \mathbb{P}[E_{k.l}] \right)$$

Finally, using (7) and Lemma 5 we get

$$\begin{aligned} &\mathbb{E}\left[M(\text{EXE})\right] \\ &\geq \frac{F^2}{8} \cdot \frac{(1-q_1)\cdot 3}{\pi^2 \cdot \lceil \log_\tau \alpha F \rceil} \cdot \frac{(1-q_2)\cdot 6}{\pi^2 \cdot \lceil \log_\tau \left(\alpha F \cdot \lceil \log_\tau(\alpha F)\rceil\right)\rceil} \\ &\geq \frac{F^2}{8} \cdot \frac{9\cdot(1-q_1)\cdot(1-q_2)}{\pi^4 \lceil \log_\tau (\alpha F)\rceil^2}. \end{aligned}$$

Conclusion of part 2.b In the conjunction of Cases (ii) and $(ii.b)$, the average message complexity of EXE is in $\Omega\left(F^2/\log_\tau^2(\alpha F)\right)$ because q_1 and q_2 are fixed constants. Note also that at least N messages need to be send in order to achieve rumor gathering. Therefore, the average message complexity of EXE is also in $\Omega(N)$. Combining this two lower bounds we get $\mathbb{E}\left[M(\text{EXE})\right] = \Omega\left(N + F^2/\log_\tau^2(\alpha F)\right)$.

Conclusion of the proof To conclude, as EXE always has to satisfy either Cases (i), $(ii)\cap(ii.a)$ or $(ii)\cap(ii.b)$, UGF can always force either $\mathbb{E}\left[T(\text{EXE})\right] = \Omega\left(\alpha F\right)$ or $\mathbb{E}\left[M(\text{EXE})\right] = \Omega\left(N + F^2/\log_\tau^2(\alpha F)\right)$.

□

Theorem 1 not only demonstrates the disruptive power of UGF, but also provides a more general intuition on the impact of our algorithm. Indeed, it presents an interesting interplay between time- and message-efficiency of all-to-all gossip protocols under adversarial manipulation by UGF. Essentially, if the protocol aims for message complexity α times lower than quadratic, then its time complexity rises exponentially in α.

V. EXPERIMENTAL RESULTS

We report here on our empirical evaluation of UGF and experimentally convey the theoretical results presented in Section IV-B. For reproducibility purposes, our implementation is accessible online[1]

A. Experimental setup

To test whether our theoretical results are applicable in practice, we evaluate the impact of UGF on the time and message complexities of several gossip protocols in various system configurations.

1) System configurations: In order to show that UGF is able to enforce large communication complexities even when the network is small, we vary the total number of processes N in $\{10, 20, 30, 50, 70, 100, 200, 300, 400, 500\}$. We also vary F in $\{0.1N, 0.2N, 0.3N, 0.4N, 0.5N\}$ (smaller F seems irrelevant as we start at $N = 10$). As expected, the higher F, the stronger the adversary, i.e., the higher communication complexities. However, the main takeaway of the experiments is consistent across all the values of F; hence below we only present results for $F = 0.3N$.

2) Gossip protocols: For all system configurations, we consider three type of all-to-all gossip protocols described below, namely *Push-Pull*, *EARS* and *SEARS*. To the best of our knowledge, these protocols are the only currently existing all-to-all gossip protocols functioning in partial synchrony even with process crashes and communication delays.

(a) **Push-Pull** This protocol, inspired by [19], proceeds as follows. At each local step, each process randomly chooses another one in the set of processes for which it does not yet know the gossip and sends it a pull request. Upon receiving a pull request, a process sends all the gossips it knows to the process that made the request. In addition, each process chooses at random another process to whom it did not send its gossip yet and sends (pushes) all the gossips it knows to this process. Finally, a process ρ falls asleep if it either made a pull request to or already knows the gossip of every other process.

[1] https://gitlab.epfl.ch/kucheren/the-universal-gossip-fighter/-/tree/main

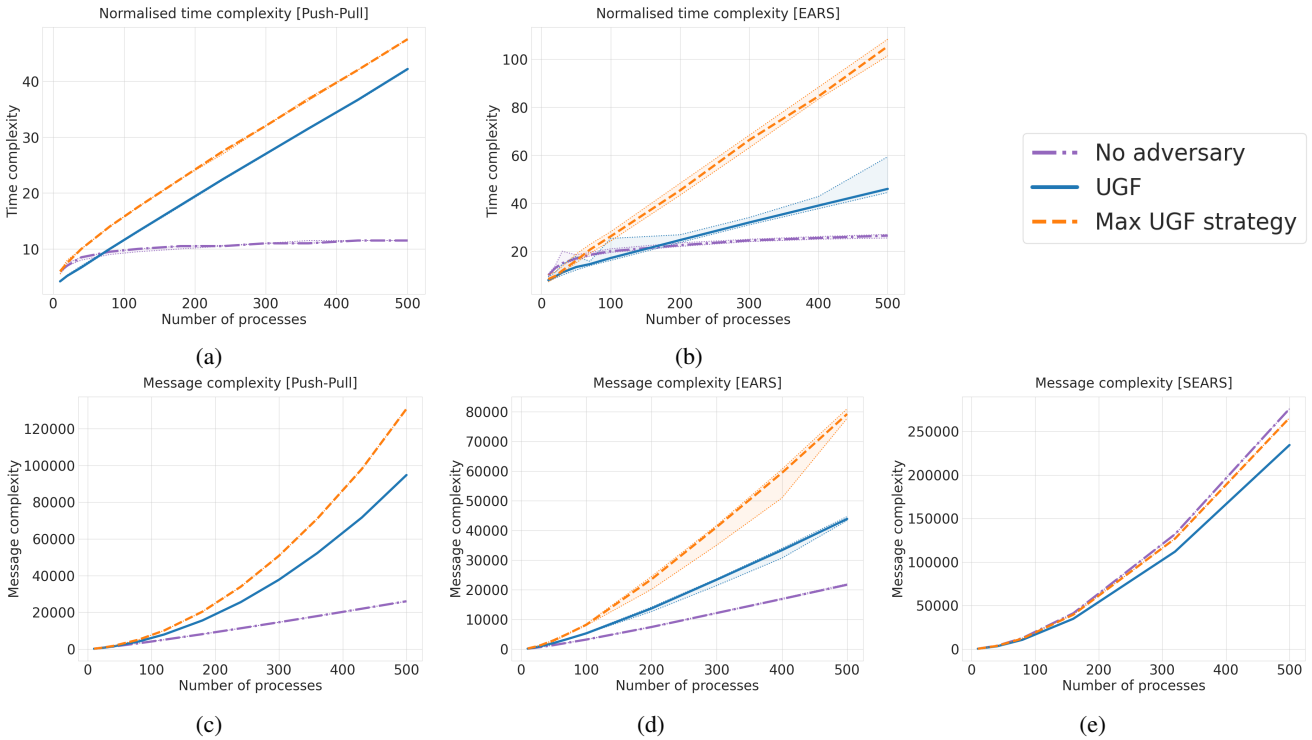

Fig. 3: Communication complexities of Push-Pull, EARS and SEARS (1) with no adversarial influence, (2) under attack by UGF, and (3) under attack by the strategy that have the most impact for each particular protocol (max UGF). For time complexity, Str. 1 is maximal for Push-Pull (see 3a) and Str. 2.1.0 for EARS (see 3b). For message complexity, Str. 2.1.1 has the most impact for all three protocols (see 3c, 3d, and 3e). The reported results represent a median over 50 runs. The dotted lines defining the shaded area around each curve represent the first and third quartiles observed during the runs.

(b) **Epidemic Asynchronous Rumor Spreading (EARS).** This protocol, first presented in [14], operates as follows. Each process ρ stores both a set of known gossips $G(\rho)$ and a set $I(\rho) = \{(\rho', g) : \text{Process } \rho' \text{ knows gossip } g\}$. At each step, each process sends a message containing both these sets to another process chosen at random. In turn, the receiver updates its sets G and I. A process ρ completes after not receiving any new message during $\frac{N}{N-F} \log N$ local steps and if there is no pair (ρ', g) such that $g \in G(\rho)$ and $(\rho', g) \notin I(\rho)$ (i.e., for every gossip ρ knows, it should also know that all other processes received it).

(c) **Spamming EARS (SEARS).** This is an adaptation of EARS also from [14]. Its objective is to ensure constant time complexity of the gossip dissemination. Similarly to EARS, each process in SEARS shares and updates sets $G(\rho)$ and $I(\rho)$. But processes do not only send one message per step. Instead, they share their sets to $c \cdot n^{\epsilon} \log n$ processes chosen at random. We set c to 1 and ϵ to 0.5 in our experiments.[2]

3) Implementation of UGF*:* We implemented UGF by applying Strategy 1, Strategy $2.k.0$ (aiming for high time complexity) and Strategy $2.k.l$ (aiming for high message complexity), each of them with probability $1/3$ (i.e. $q_1 = 1/3$ and $q_2 = 1/2$). For the sake of simplicity we set k and l to 1 and $\tau = F$.

4) Baseline: In order to measure the harm caused by UGF, we evaluate the time and message complexity when there is *no adversary*, i.e. when local step and delivery times are equal to 1 and no processes are crashed.

B. Results

Our experimental results are presented in Figure 3. It compares the average time and message complexity of the Push-Pull, EARS, and SEARS gossip protocols without attack (no adversary) and under attack by UGF. Besides, we also present the strategy that causes the most damage in terms of time or message complexity (max UGF). This illustrates which strategy is the most efficient one for each protocol.

1) Main takeaway: From Figure 3, we clearly observe that, in all protocols, UGF forces either linear time complexity, or quadratic message complexity, thus dramatically hampering the success of the dissemination. Moreover, while we have shown in Section IV that either time complexity or message complexity should be high, in practice UGF has a significant impact on both communication complexities simultaneously.

2) Comparison to the baseline: Our experimental results confirm that the impact of UGF on the communication complexities is extremely important when compared to the baseline (no-adversary). While the time complexity of the

[2]SEARS works for any value of $\epsilon \in [0, 1]$ and $c \in \mathbb{N}$ as explained in [14]

baseline is logarithmic in the Push-Pull and EARS protocols (see Figures 3a and 3b), it becomes linear under attack by UGF. We observe similar results for the message complexity, as UGF makes it quadratic in all protocols; hence matching our theoretical findings.

3) Interesting remark on SEARS: By construction, SEARS aims to achieve constant time complexity. Therefore, an adversary can only influence the message complexity of SEARS. For this reason, we only report this complexity in our experiments for SEARS. Moreover, we observe that even without attack, SEARS' message complexity is already quadratic in N, thus reaching the trivial logical limit on the message complexity of an all-to-all gossip protocol. In this sense, SEARS always sacrifices message complexity for time efficiency; hence automatically placing itself at one end of the interplay between time and message complexity under attack.

VI. Concluding remarks and related works

This paper initiates a new line of research in distributed computing consisting of studying the power and universality of adaptive adversaries. We introduce a new adversary called the *Universal Gossip Fighter* (UGF) and prove that UGF can force any all-to-all gossip protocol to have either a linear time or a quadratic message complexity on average.

The notion of adversary is central to the literature on distributed computing and has been studied in several contexts before. In particular, several papers studied gossip dissemination on a dynamic network with an adaptive adversary that can also affect the topology of the network according to its knowledge about the current location of the different gossips [1], [16], [17], [22]. In this work, we consider a fixed network and therefore it cannot be altered by the adversary. Another research line [2], [7], [13], have been studying gossips that propagate in a fixed network but in a synchronous manner: the adversary is only able to crash processes. We consider the more challenging setting where communication is partially synchronous and the adversary can change message delays.

The closest result to our work is the one presented in [14], considering a fixed network with partially synchronous communications. Two kinds of adversaries are compared: the oblivious and the adaptive adversary [2], [6], [16]. It is shown that while oblivious adversaries are not sufficiently powerful to harm the dissemination, for any gossip protocol, there exists an adaptive adversary such that, the dissemination has $\Omega(N+F^2)$ message complexity or $\Omega(F)$ time complexity on average. Our work however significantly differs from this one in two ways.

First, [14] only states a worst-case result. In short, it says that "for any gossip protocol, we can design a specific adversary that harms the execution on average". We present a much stronger result by demonstrating that "there is an adaptive adversary that, on average, hinders the execution of any gossip protocol". This first difference is fundamental because it means that we can design and implement a universal adversary that can be applied independently of context, and without prior knowledge of the protocol it attacks. To the best of our knowledge, we are the first to present such an adversary.

Second, our work also generalizes the impossibility result of [14]. Indeed, we show that, for any α, UGF can force either $\Omega(N+{F^2}/{\log^2_\tau(\alpha F)})$ message complexity or $\Omega(\alpha F)$ time complexity on average. When $\alpha = 1$ and $\tau = F$, we retrieve the same findings as in [14], but our result says much more than that. We highlight a trade-off between the time and the message complexity under attack by showing that achieving a message complexity α times smaller than quadratic, forces a time complexity that increases exponentially with respect to α. To our knowledge, this analysis is also new to the community.

VII. The future work

A question of further interest would be to evaluate whether some realistic additional information about the gossip could improve the performance of our algorithm. It would also be worth studying stronger adversaries, e.g. that can omit messages instead of simply delaying them [21] - would this kind of adversary harm the dissemination even more?

Another interesting future direction is the use of UGF for studying practical vulnerabilities of distributed systems. Our work could pave the way to new research fields in adversarial machine learning where all-to-all communication is important. For example, in collaborative learning, UGF could model an adversarial system provider that fights against the design of personalized machine learning models by slowing the network communications. Finally, we would like to study whether if UGF or a variant of it could be applied to other problems in partial synchrony model, for instant a related Do-All problem [20].

Acknowledgments

Rafael Pinot has been supported in part by Ecocloud, an EPFL research center (Postdoctoral Research Award).

References

[1] Ahmadi, M., Kuhn, F., Kutten, S., Molla, A. R., and Pandurangan, G. The communication cost of information spreading in dynamic networks. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (2019), pp. 368–378.

[2] Alistarh, D., Gilbert, S., Guerraoui, R., and Zadimoghaddam, M. How efficient can gossip be? (on the cost of resilient information exchange). In *Proceedings of the 37th International Colloquium Conference on Automata, Languages and Programming: Part II* (Berlin, Heidelberg, 2010), ICALP'10, Springer-Verlag, p. 115–126.

[3] Aspnes, J., and Hurwood, W. Spreading rumors rapidly despite an adversary. *Journal of Algorithms 26*, 2 (1998), 386–411.

[4] Augustine, J., Avin, C., Liaee, M., Pandurangan, G., and Rajaraman, R. Information spreading in dynamic networks under oblivious adversaries, 2016.

[5] Birman, K. P., Hayden, M., Ozkasap, O., Xiao, Z., Budiu, M., and Minsky, Y. Bimodal multicast. *ACM Trans. Comput. Syst. 17*, 2 (May 1999), 41–88.

[6] Chlebus, B. S., and Kowalski, D. R. Robust gossiping with an application to consensus. *Journal of Computer and System Sciences 72*, 8 (2006), 1262–1281.

[7] Chlebus, B. S., and Kowalski, D. R. Time and communication efficient consensus for crash failures. In *Distributed Computing* (Berlin, Heidelberg, 2006), S. Dolev, Ed., Springer Berlin Heidelberg, pp. 314–328.

[8] Colin, I., Bellet, A., Salmon, J., and Clémençon, S. Gossip dual averaging for decentralized optimization of pairwise functions. In *International Conference on Machine Learning* (2016).

[9] DEMERS, A., GREENE, D., HAUSER, C., IRISH, W., LARSON, J., SHENKER, S., STURGIS, H., SWINEHART, D., AND TERRY., D. Epidemic algorithms for replicated database maintenance. *Proceedings of the 6th Annual ACM Symposium on Principles of Distributed Computing 8* (1987), 1–12.

[10] DOERR, B., FOUZ, M., AND FRIEDRICH, T. Social networks spread rumors in sublogarithmic time. In *Proceedings of the forty-third annual ACM symposium on Theory of computing* (2011), pp. 21–30.

[11] DUCHI, J. C., AGARWAL, A., AND WAINWRIGHT, M. J. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control 57*, 3 (2011), 592–606.

[12] DWORK, C., LYNCH, N., AND STOCKMEYER, L. Consensus in the presence of partial synchrony. *Journal of the ACM (JACM) 35*, 2 (1988), 288–323.

[13] EVEN, S., AND MONIEN, B. On the number of rounds necessary to disseminate information. In *Proceedings of the First Annual ACM Symposium on Parallel Algorithms and Architectures* (New York, NY, USA, 1989), SPAA '89, Association for Computing Machinery, p. 318–327.

[14] GEORGIOU, C., GILBERT, S., GUERRAOUI, R., AND KOWALSKI, D. R. On the complexity of asynchronous gossip. In *Proceedings of the Twenty-Seventh ACM Symposium on Principles of Distributed Computing* (New York, NY, USA, 2008), PODC '08, Association for Computing Machinery, p. 135–144.

[15] GIAKKOUPIS, G., GUERRAOUI, R., JÉGOU, A., KERMARREC, A.-M., AND MITTAL, N. Privacy-conscious information diffusion in social networks. In *International Symposium on Distributed Computing* (2015), Springer, pp. 480–496.

[16] GIAKKOUPIS, G., SAUERWALD, T., AND STAUFFER, A. Randomized rumor spreading in dynamic graphs. In *Automata, Languages, and Programming* (Berlin, Heidelberg, 2014), J. Esparza, P. Fraigniaud, T. Husfeldt, and E. Koutsoupias, Eds., Springer Berlin Heidelberg, pp. 495–507.

[17] HAEUPLER, B., AND KUHN, F. Lower bounds on information dissemination in dynamic networks. In *Proceedings of the 26th International Conference on Distributed Computing* (Berlin, Heidelberg, 2012), DISC'12, Springer-Verlag, p. 166–180.

[18] HETHCOTE, H. W. The mathematics of infectious diseases. *SIAM review 42*, 4 (2000), 599–653.

[19] KARP, R., SCHINDELHAUER, C., SHENKER, S., AND VOCKING, B. Randomized rumor spreading. In *41st Annual Symposium on Foundations of Computer Science* (2000), IEEE, pp. 565–574.

[20] KOWALSKI, D. R., AND SHVARTSMAN, A. A. Performing work with asynchronous processors: Message-delay-sensitive bounds. *Inf. Comput. 203*, 2 (2005), 181–210.

[21] KOWALSKI, D. R., AND STROJNOWSKI, M. Gossiping by processors prone to omission failures. *Inf. Process. Lett. 109*, 6 (2009), 308–314.

[22] KUHN, F., LYNCH, N., AND OSHMAN, R. Distributed computation in dynamic networks. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing* (New York, NY, USA, 2010), STOC '10, Association for Computing Machinery, p. 513–522.

[23] MALKHI, D., MANSOUR, Y., AND REITER, M. K. On diffusing updates in a byzantine environment. In *Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems* (USA, 1999), SRDS '99, IEEE Computer Society, p. 134.

[24] VAN RENESSE, R., BIRMAN, K., AND VOGELS., W. Astrolabe: A robust and scalable technology for distributed systems monitoring, management, and data mining. *Proceedings of the 6th Annual ACM ACM Transactions on Computer Systems 821(3)* (2003).

[25] VANHAESEBROUCK, P., BELLET, A., AND TOMMASI, M. Decentralized collaborative learning of personalized models over networks. In *Artificial Intelligence and Statistics* (2017), pp. 509–517.