Tolerating Corrupted Communication*

Martin Biely
TU Wien, Vienna
biely@ecs.tuwien.ac.at

Bernadette Charron-Bost
Ecole Polytechnique, Paris
charron@polytechnique.fr

Antoine Gaillard
Ecole Polytechnique, Paris
gaillard@lix.polytechnique.fr

Martin Hutle
EPFL, Lausanne
martin.hutle@epfl.ch

André Schiper
EPFL, Lausanne
andre.schiper@epfl.ch

Josef Widder
TU Wien, Vienna
widder@ecs.tuwien.ac.at

ABSTRACT

Consensus encalpsulates the inherent problems of building fault tolerant distributed systems. In this context, the classic model of *Byzantine* faulty processes can be restated such that messages from a subset of processes can be arbitrarily corrupted (including addition and omission of messages).

We consider the case of *dynamic* and *transient* faults, that may affect all processes and that are not permanent, and we model them via corrupted communication. For corrupted communication it is natural to distinguish between the safety of communication, which is concerned with the number of altered messages, and the liveness of communication, which restricts message loss.

We present two consensus algorithms, together with sufficient conditions on the system to ensure correctness. Our first algorithm needs strong conditions on safety but requires weak conditions on liveness in order to terminate. Our second algorithm tolerates a lower degree of communication safety at the price of stronger liveness conditions.

Our algorithms allow us to circumvent the resilience lower bounds from Santoro/Widmayer and Martin/Alvisi.

Categories and Subject Descriptors: C.2.4 [Computer-Communication Networks]: Distributed Systems; F.1.1 [Computation by Abstract Devices]: Models of Computation F.2.m [Analysis of Algorithms and Problem Complexity]: Miscellaneous.

General Terms: Algorithms, Theory.

Keywords: Byzantine Fault Tolerance, Consensus, Dynamic Faults, Transient Faults.

*Rearch funded by the Swiss National Science Foundation under grant number 200021-111701, by the Austrian BM:vit FIT-IT project *TRAFT* (proj. no. 812205), and by the Austrian FWF project *Theta* (proj. no. P17757).

PODC'07, August 12–15, 2007, Portland, Oregon, USA.

1. INTRODUCTION

When looking at distributed systems with (non-benign) corruption faults, we can distinguish between (i) *state faults*, that hit the state of the processes, and (ii) *transmission faults*, that affect the information that is exchanged among processes. The first class received more attention in literature, one noteworthy exception is the seminal work of Santoro and Widmayer [18]. Moreover, due to quite discouraging impossibility results regarding agreement tasks (like consensus) in the presence of unreliable communication [9, 18, 21], communication faults were often mapped to process faults in order to be able to define and solve representative problems. In case the erroneous message stems from a corrupted state, this modelling of transmission faults as faulty processes is somehow natural.

In contrast to this classical approach, in this paper we explicitly look at the second type of faults, i.e., we consider only transmission faults, and no state corruptions. In this context the notion of a "faulty process" becomes much more problematic: In fact, the scenario, where a process p should send a message m to a process q, but q receives a message m' different from m may have the following causes:

- p has sent m' instead of m (send value fault),
- m was correctly transmitted, but q erroneously delivers m' (receive value fault),
- p and q behaved correctly, but the channel from p to q has corrupted m to m' (faulty channel).

A benign fault is the special case of the former example with m' being just not received. Here, Charron-Bost and Schiper [6] have shown that it is not only unnecessary to distinguish these three types of faults, but also may be harmful.

In this paper, we extend the round-based approach of [6] to value faults.[1] In this approach, we reason about faults only as being *transmission faults*, without looking for a "culprit" for the fault. As in [6], we are thus able — in the classical terminology — to deal with both *dynamic* and *transient* faults. A *transient* fault is a non-permanent fault; a *dynamic* fault is a fault that can affect any process/channel in the system — as opposed to *static* faults that affect at most f out of n processes per run.

Transmission faults are addressed in this paper in the context of the HO model (which stands for Heard-Of) defined

[1]We use here the term *value fault* for non-benign transmission faults, since the term *Byzantine* makes only sense in the context of process faults. Nevertheless, in general, value faults can be arbitrarily malicious.

for the benign case in [6]. The HO model is a communication-closed round model inspired by [7], [8], and [18]. Note that the round structure of our model does not imply limits on the asynchrony of the system. An algorithm $\mathcal{A}$ is specified by sending functions S_p^r and transition functions T_p^r for each round r and process p. Then, for each round the *discrepancy between what should be sent (with respect to S_p^r) and what is actually received* is characterized by two sets $HO(p,r)$ and $SHO(p,r)$. The former — which is already defined in the benign case — gives the set of processes from which p receives a message. Additionally, the latter, the *safe heard-of* set $SHO(p,r)$ denotes the set of processes from which p receives an uncorrupted message at round r. Together with $\mathcal{A}$, by specifying a *communication predicate* $\mathcal{P}$ over the collection of all HO and SHO sets, we get an HO machine $\langle \mathcal{A}, \mathcal{P} \rangle$. The communication predicate $\mathcal{P}$ hereby characterizes all assumptions on the system, like synchrony and failures in a unified way.

Informally, predicates on the SHO and HO collections specify the safety of the communication, whereas those on the HO collections alone characterize liveness of the communication. For the benign case, we always have $SHO(p,r) = HO(p,r)$.

Above all, this paper is related to theoretical aspects of consensus in the presence of non-benign and dynamic faults, following the spirit of Santoro and Widmayer's approach [18, 19]. We are aware that some causes of value faults can be reduced with error correcting codes or signed messages. They can be used to transform some value faults into benign faults, i.e., omissions. Despite the theoretical problem that such techniques are hard to formalize properly [15], their use is based on the widespread assumption that undetected corruptions have a "neglectable probability". Our approach consists of weakening this assumption by allowing a certain number of undetected corruptions (value faults) per round.

Contribution. We define a novel computational model for dynamic and transient value faults that leads to new algorithmic solutions for consensus.

We present two algorithms that solve consensus in the presence of transient and dynamic faults, the $\mathcal{A}_{T,E}$ and the $\mathcal{U}_{T,E,\alpha}$ algorithms. The first algorithm requires a weaker condition for safety of communication than the second one, while the second algorithm requires a weaker condition for liveness of communication than the first one.

We show that these two algorithms require a predicate $\mathcal{P}_\alpha$ for safety of consensus that is weaker than some classical assumptions about faulty processes. Moreover, these algorithms allow us to "circumvent" the lower bounds by Santoro and Widmayer [18] ($\lfloor n/2 \rfloor$ value transmission faults per round) and by Martin and Alvisi [16] (less than $n/5$ Byzantine processes for fast consensus). Note that our algorithms do not contradict these bounds as we distinguish between liveness and safety predicates. Our algorithms need a stronger condition for liveness, a condition that makes sense in the context of transient faults. This shows that the lower bounds for permanent faults do not hold with transient faults. For suitable choices of parameters our algorithms attain the lower bound on Byzantine consensus claimed by Lamport [11].

Organization of the paper. In Section 2 we introduce the HO model for value faults. In Sections 3 and 4 we present two consensus algorithms for this new fault model. We also give expressions on the number of faults that can be tolerated in order to ensure safety as well as communication predicates which ensure liveness. In Section 5 we compare our results to related work.

2. MODEL

Computations in our model are structured in rounds, that are communication-closed layers in the sense that any message sent in a round can be received only at that round. We give a definition of our round-based model, and introduce the notions of the *heard-of sets* (HO), which handles omissions, i.e., captures communication liveness properties, and of the *safe heard-of sets* (SHO), which handles corruptions, i.e., captures communication safety properties. These collections of sets allow us to specify sufficient conditions for solving consensus with our algorithms in Sections 3 and 4.

2.1 Heard-Of Sets

Let Π be a finite non-empty set of cardinality n, and let M be a set of messages (optionally including a *null* placeholder indicating the empty message). To each p in Π, we associate a *process*, which consists of the following components: a set of states denoted by $states_p$, a subset $init_p$ of initial states, and for each positive integer r called *round number*, a message-sending function S_p^r mapping $states_p \times \Pi$ to a unique message from M, and a state-transition function T_p^r mapping $states_p$ and partial vectors (indexed by Π) of elements of M to $states_p$. The collection of processes is called an *algorithm on* Π. In each round r, a process p:

1. applies S_p^r to the current state, and emits the "messages" to be sent (according to its sending function S_p^r) to each process;
2. determines the partial vector $\vec{\mu}_p^r$, formed by the messages that p receives at round r;
3. applies T_p^r to its current state and $\vec{\mu}_p^r$.

The partial vector $\vec{\mu}_p^r$ is called the *reception vector of p at round r*.

Computation evolves in an infinite sequence of rounds. Each *run* is entirely determined by the initial configuration (i.e., the collection of process initial states), and the collection of the reception vectors

$$\left(\vec{\mu}_p^r\right)_{p\in\Pi,\ r>0}.$$

For each process p and each round r, we introduce two subsets of Π:

- The *heard-of* set, denoted $HO(p,r)$, which is the support of $\vec{\mu}_p^r$, i.e.,
$$HO(p,r) = \{q \in \Pi : \vec{\mu}_p^r[q] \text{ is defined}\}.$$
- The *safe heard-of* set, denoted $SHO(p,r)$, and defined by
$$SHO(p,r) = \{q \in \Pi : \vec{\mu}_p^r[q] = S_q^r(s_q, p)\},$$
where s_q is q's state at the beginning of round r.

Both sets specify the discrepancy between *what should be sent* and *what is actually received*. As for the benign case [6], we make no assumption on the reason why $\vec{\mu}_p^r[q] \neq S_q^r(s_q, p)$: it may be due to an incorrect sending by q, an incorrect

receipt by p, or due to the corruption by the link. Obviously, we have

$$SHO(p,r) \subseteq HO(p,r).$$

In contrast to $HO(p,r)$, process p is not able to determine $SHO(p,r)$.

From the sets $HO(p,r)$ and $SHO(p,r)$, we form the *altered heard-of set* denoted $AHO(p,r)$ as follows:

$$AHO(p,r) = HO(p,r) \setminus SHO(p,r).$$

For any round r, we further define the *kernel*, resp. *safe kernel*, of a round:

$$K(r) = \bigcap_{p\in\Pi} HO(p,r) \qquad SK(r) = \bigcap_{p\in\Pi} SHO(p,r)$$

The kernel of round r consists of all processes which are heard by all at round r, whereas the safe kernel consists of all processes whose messages were received correctly by all processes. We can generalize this definition to the whole computation:

$$K = \bigcap_{r>0} K(r) \qquad SK = \bigcap_{r>0} SK(r)$$

Similarly, the *altered span* (of round r) denotes the set of processes from which at least one process received a corrupted message (at round r):

$$AS(r) = \bigcup_{p\in\Pi} AHO(p,r) \qquad AS = \bigcup_{r>0} AS(r)$$

2.2 HO Machines

A *heard-of machine* for a set of processes Π is a pair $\langle \mathcal{A}, \mathcal{P}\rangle$, where $\mathcal{A}$ is an algorithm on Π, and $\mathcal{P}$ is a *communication predicate*, i.e., a predicate over $(HO(p,r))_{p\in\Pi,\, r>0}$ and $(SHO(p,r))_{p\in\Pi,\, r>0}$. Predicates over $(HO(p,r))_{p\in\Pi,\, r>0}$ characterize the liveness properties of communication, whereas predicates over $(SHO(p,r))_{p\in\Pi,\, r>0}$ characterize the safety properties of communication.

As an example, we can model the classical assumption, that no more than α processes may send a corrupted information in a computation:[2]

$$\mathcal{P}_\alpha^{perm} \;::\; |AS| \leq \alpha \tag{1}$$

For our algorithms we will consider the weaker predicate $\mathcal{P}_\alpha$ that restricts the number of corrupted messages only per round and per process:

$$\mathcal{P}_\alpha \;::\; \forall r > 0, \forall p \in \Pi : |AHO(p,r)| \leq \alpha \tag{2}$$

with $\alpha \in \mathbb{R}$ such that $0 \leq \alpha \leq n$, and we say that a computation has α-*safe communication* when $\mathcal{P}_\alpha$ holds. Note that $\mathcal{P}_\alpha^{perm}$ implies $\mathcal{P}_\alpha$.

We will consider only communication predicates, that are *time-invariant*: A communication predicate $\mathcal{P}$ is time-invariant if it has the same truth value for all heard-of collections $(HO(p,r+i))_{p\in\Pi,r>0}$ and $(SHO(p,r+i))_{p\in\Pi,r>0}$ for any $i \in \mathbb{N}$. This allows processes to start in their execution in any round r, while preserving the truth value of $\mathcal{P}$. The predicates $\mathcal{P}_\alpha^{perm}$ and $\mathcal{P}_\alpha$ are trivially time-invariant, since they are *permanent* predicates. Time-invariant predicates that characterize *eventual properties* have typically the form

$$\forall r > 0\ \exists r_0 > r :\ \mathcal{P}_{r_0},$$

where $\mathcal{P}_{r_0}$ is a communication predicate over the collection $(HO(p,r_0);\, SHO(p,r_0))_{p\in\Pi}$.

HO machines for the benign case [6] can be seen as a special case of the above definition, with $AS = \emptyset$, which is equivalent to assuming the predicate

$$\mathcal{P}_{benign} :: \forall p \in \Pi\ \forall r > 0 :\ SHO(p,r) = HO(p,r).$$

2.3 Consensus

In this paper, we concentrate on the *consensus* problem in V, where V is a (non-empty) totally ordered set. For this problem, every process has an initial value $v_p \in V$ and decides irrevocably on a decision value, fulfilling:

Integrity: If all processes have the same initial value this is the only possible decision value.

Agreement: No two processes may decide differently.

Termination: All processes eventually decide.

Since, contrary to classical approaches, there is no deviation according to T_p^r, and thus we do not have the notion of a faulty process, the upper specification makes no exemption: all processes should decide the initial value in the Integrity clause, and all processes must make a decision by the Termination clause.

Formally, an HO machine $\langle \mathcal{A}, \mathcal{P}\rangle$ solves consensus, if any run for which $\mathcal{P}$ holds, satisfies Integrity, Agreement, and Termination. To make this definition non-trivial, we assume that the set of *HO* and *SHO* collections for which $\mathcal{P}$ holds is non-empty.

The implementation of predicates is not discussed in this paper; it can be done in the spirit of [10].

3. WHEN COMMUNICATION IS SAFE BUT NOT SO LIVE

In this section we present the first of our two algorithms, that we call the $\mathcal{A}_{T,E}$ algorithm. The code of $\mathcal{A}_{T,E}$ is given as Algorithm 1 and is designed to work under the predicates $\mathcal{P}_\alpha$ and $\mathcal{P}^{\mathcal{A},live}$, given in (2) and (3) in Figure 1.

Basically, it consists in a parametrization of the two receive thresholds T and E of the *OneThirdRule* algorithm given in [6] for solving consensus in the presence of benign failures (there both are equal to $\frac{2n}{3}$). The main features of *OneThirdRule* are: (i) the algorithm is always safe, whatever the number of benign transmission faults is, and (ii) the algorithm is *fast* in the sense that it requires two rounds to terminate in every fault free run, and ensures termination at the end of the first round if, in fault free runs, all initial values are equal. In the sequel, we show that, for appropriate choices of T and E, the $\mathcal{A}_{T,E}$ algorithm both tolerates a large number of benign faults (omission) and non-benign faults (corruption), while retaining the main features of *OneThirdRule*.

3.1 The $\mathcal{A}_{T,E}$ algorithm

In $\mathcal{A}_{T,E}$, each process p maintains a variable x_p initially equal to p's initial value. At each round, process p broadcasts x_p, and then updates x_p if it receives more than T

[2]Note that in the classical Byzantine setting, also the state of a "faulty" process can be corrupted, which is not the case in our model (see Section 5.2 for a discussion).

Algorithm 1 The $\mathcal{A}_{T,E}$ algorithm

```
1: Initialization:
2:   x_p ∈ V, initially v_p                    /* v_p is the initial value of p */

3: Round r:
4:   S_p^r:
5:     send ⟨ x_p ⟩ to all processes
6:   T_p^r:
7:     if |HO(p, r)| > T then
8:       x_p := the smallest most often received value in this round
9:     if more than E values received are equal to v then
10:      DECIDE(v)
```

("Threshold") messages: p sets x_p to the smallest value that it has received most frequently. If p has received more than E ("Enough") times the same value v, then it decides on v.

3.2 Correctness of the $\mathcal{A}_{T,E}$ algorithm

First we introduce some piece of notation. For any variable x local to process p, we denote $x_p^{(r)}$ the value of x_p at the end of round r. For any value $v \in V$ and any process p, at any round $r > 0$, we define the sets $R_p^r(v)$ and $Q_p^r(v)$ as follows:

$$R_p^r(v) := \{q \in \Pi : \vec{\mu}_p^r[q] = v\}$$
$$Q_p^r(v) := \{q \in \Pi : S_q^r(p, s_q) = v\}.$$

where s_q denotes q's state at the beginning of round r. The set $R_p^r(v)$ (resp. $Q_p^r(v)$) represents the set of processes from which p receives v (resp. which ought to send v to p) at round r. Since at each round of the $\mathcal{A}_{T,E}$ algorithm, every process sends the same message to all, the sets $Q_p^r(v)$ do not depend on p, and so can be just denoted by $Q^r(v)$ without any ambiguity.

We start our correctness proof with a general basic lemma

LEMMA 1. *For any process p and any value v, at any round r, we have:*

$$|R_p^r(v)| \leq |Q^r(v)| + |AHO(p, r)|$$

PROOF. Suppose that process p receives a message with value v at round $r > 0$ from process q. Then, either the code of q prescribes it to send v to p at round r, i.e., q belongs to $Q^r(v)$ and thus q is also in $SHO(p, r)$, or the message has been corrupted and q is in $AHO(p, r)$. It follows that $R_p^r(v) \subseteq Q^r(v) \cup AHO(p, r)$, which implies $|R_p^r(v)| \leq |Q^r(v)| + |AHO(p, r)|$. □

Lemma 1 naturally leads to consider the communication predicate $\mathcal{P}_\alpha$ from (2): $\mathcal{P}_\alpha$ bounds the size of $AHO(p, r)$, i.e., it limits the discrepancy between the sets R_p^r and Q^r.

Our second lemma shows that choosing $E \geq \frac{n}{2}$ renders the decision rule in the $\mathcal{A}_{T,E}$ algorithm "deterministic".

LEMMA 2. *If $E \geq \frac{n}{2}$, then the guard in line 9 of the $\mathcal{A}_{T,E}$ algorithm is true for at most one value v.*

PROOF. Assume by contradiction that there exist a process p and a round r, so that the guard in line 9 is true for two distinct values v and v'. The code implies that $|R_p^r(v)| > E$ and $|R_p^r(v')| > E$. Since v and v' are different, $R_p^r(v)$ and $R_p^r(v')$ are disjoint sets, and so

$$|R_p^r(v) \cup R_p^r(v')| = |R_p^r(v)| + |R_p^r(v')|.$$

From $E \geq \frac{n}{2}$ we have $|R_p^{r_0}(v) \cup R_p^{r_0}(v')| > n$, a contradiction. □

As an intermediate step to argue agreement, our next lemma shows that a stronger condition on E ensures no two processes can decide differently *at the same round*:

LEMMA 3. *If $E \geq \frac{n}{2} + \alpha$ then in any run of the HO machine $\langle \mathcal{A}_{T,E}, \mathcal{P}_\alpha \rangle$ there is at most one possible decision value per round.*

PROOF. Assume by contradiction that there exist two processes p and q that decide on different values v and v' in some round $r > 0$. From the code of $\mathcal{A}_{T,E}$, we deduce that $|R_p^r(v)| > E$ and $|R_q^r(v')| > E$. Then Lemma 1 ensures that $|Q^r(v)| > E - \alpha$ and $|Q^r(v')| > E - \alpha$ when $\mathcal{P}_\alpha$ holds.

Since each process sends the same value to all at each round r, the sets $Q^r(v)$ and $Q^r(v')$ are disjoint if v and v' are distinct values . Hence $|Q^r(v) \cup Q^r(v')| = |Q^r(v)| + |Q^r(v')|$. Consequently, since $E \geq \frac{n}{2} + \alpha$, we derive that $|Q^r(v) \cup Q^r(v')| > n$, a contradiction. □

The next lemma ensures that for sufficiently large T, once a process has decided, other processes may learn only the decision value *in that round*:

LEMMA 4. *If $T \geq 2(n + 2\alpha - E)$, then in any run of the HO machine $\langle \mathcal{A}_{T,E}, \mathcal{P}_\alpha \rangle$ such that process p decides value v at round r_0, every process q that updates its variable x_q at round r_0 sets it to v.*

PROOF. Suppose that process p decides value v at round $r_0 > 0$. Let q be any process such that $|HO(q, r_0)| > T$, i.e., q modifies x_q at round r_0. Let $Q^{r_0}(\bar{v})$ denote the set of processes that, according to their sending functions, ought to send messages different from v at round r_0, and let $R_q^{r_0}(\bar{v})$ denote the set of processes from which q receives values different from v at round r_0. Since each process sends a message to all at each round, $Q^{r_0}(\bar{v}) = \Pi \setminus Q^{r_0}(v)$, and thus $|Q^{r_0}(\bar{v})| = n - |Q^{r_0}(v)|$. Similarly, we have $R_q^{r_0}(\bar{v}) = HO(q, r_0) \setminus R_q^{r_0}(v)$, and since $R_q^{r_0}(v) \subseteq HO(q, r_0)$, it follows that $|R_q^{r_0}(\bar{v})| < T - R_q^{r_0}(v)$.

Since p makes a decision at round r_0, by line 7, $|R_p^{r_0}(v)| > E$. Then Lemma 1 implies $|Q^{r_0}(v)| > E - \alpha$, from which $|Q^{r_0}(\bar{v})| < n - (E - \alpha)$ follows. With an argument similar to the one used in the proof of Lemma 1, we derive that $|R_q^{r_0}(\bar{v})| \leq |Q^{r_0}(\bar{v})| + |AHO(q, r_0)|$. When $\mathcal{P}_\alpha$ holds, we obtain $|R_q^{r_0}(\bar{v})| < n + 2\alpha - E$.

It follows that because of $T \geq 2(n + 2\alpha - E)$, $|R_q^{r_0}(v)| > |R_q^{r_0}(\bar{v})|$. This implies that v is the most frequent value received by q at round r_0. Then the code entails q to set x_q to v. □

We now extend the statement of Lemma 4 to hold also for any round after the decision:

LEMMA 5. *If $T \geq 2(n + 2\alpha - E)$, then in any run of the HO machine $\langle \mathcal{A}_{T,E}, \mathcal{P}_\alpha \rangle$ such that process p decides some value v at some round $r_0 > 0$, every process q that updates its variable x_q at some round $r \geq r_0$ necessarily sets it to v.*

PROOF. Assume process p decides value v at round $r_0 > 0$. First we prove by induction on r that:

$$\forall r \geq r_0,\ |\{p' \in \Pi : x_{p'}^{(r-1)} = v\}| > E - \alpha.$$

Basic case: $r = r_0$. Since p decides v at round r_0, then $|R_p^{r_0}(v)| > E$. By Lemma 1, we have $|Q^{r_0}(v)| > E - \alpha$ when $\mathcal{P}_\alpha$ holds. From the code of $\mathcal{A}_{T,E}$, we have $Q^{r_0}(v) = \{p' \in \Pi : x_{p'}^{(r_0-1)} = v\}$, and so the basic case follows.

$$\begin{array}{c} \forall r_0 > 0,\ \exists r \geq r_0,\ \exists \Pi_r^1, \Pi_r^2 \subseteq \Pi \text{ s.t.} (|\Pi_r^1| > E - \alpha) \wedge (|\Pi_r^2| > T) \wedge (\forall p \in \Pi_r^1,\ HO(p,r) = SHO(p,r) = \Pi_r^2) \\ \wedge \\ \forall r > 0,\ \forall p \in \Pi,\ \exists r_p > r\ :\ |HO(p, r_p)| > T \\ \wedge \\ \forall r > 0,\ \forall p \in \Pi,\ \exists r_p > r\ :\ |SHO(p, r_p)| > E \end{array} \tag{3}$$

Figure 1: Predicate $\mathcal{P}^{\mathcal{A},live}$

Inductive step: $r > r_0$. From the inductive assumption, we know that $|\{p' \in \Pi : x_{p'}^{(r-1)} = v\}| > E - \alpha$. The same argument as in Lemma 4, because we have $T \geq 2(n+2\alpha-E)$, yields $|\{p' \in \Pi : x_{p'}^{(r)} = v\}| > E - \alpha$.

Let q be some process that updates x_q at some round $r \geq r_0$. Since $|Q^r(v)| = |\{p' : x_{p'}^{(r-1)} = v\}| > E - \alpha$, the same argument as in Lemma 4 applies, and so the code entails q to set x_q to v at round r. □

From the above lemmas, we derive a sufficient condition on E and T which ensures that the HO machine $\langle \mathcal{A}_{T,E}, \mathcal{P}_\alpha \rangle$ satisfies the Agreement clause of consensus.

Proposition 1 (Agreement). *If $E \geq \frac{n}{2} + \alpha$ and $T \geq 2(n + 2\alpha - E)$, then there is at most one possible decision value in any run of the HO machine $\langle \mathcal{A}_{T,E}, \mathcal{P}_\alpha \rangle$.*

Proof. Let $r_0 > 0$ be the first round at which some process p makes a decision, and let v be p's decision value. Assume that process q decides v' at round r. By definition of r_0, we have $r \geq r_0$.

We proceed by contradiction, and assume that $v \neq v'$. By Lemma 3, we derive that $r > r_0$. Since p decides v at round r_0 and q decides v' at round r, Lemma 1 ensures that $|Q^{r_0}(v)| > E-\alpha$ and $|Q^r(v')| > E-\alpha$ when $\mathcal{P}_\alpha$ holds. Since $T \geq 2(n+2\alpha-E)$, Lemma 5 implies that $Q^{r_0}(v)$ and $Q^r(v')$ are disjoint sets. Therefore, $|Q^{r_0}(v) \cup Q^r(v')| = |Q^{r_0}(v)| + |Q^r(v')|$. Because of $E \geq \frac{n}{2} + \alpha$, we have $|Q^{r_0}(v) \cup Q^r(v')| > n$, a contradiction. □

Similarly, we derive sufficient conditions on E and T which ensures that the HO machine $\langle \mathcal{A}_{T,E}, \mathcal{P}_\alpha \rangle$ satisfies the Integrity clause of consensus.

Proposition 2 (Integrity). *If $E \geq \alpha$ and $T \geq 2\alpha$, then in any run of the HO machine $\langle \mathcal{A}_{T,E}, \mathcal{P}_\alpha \rangle$ where all the initial values are equal to some value v_0, the only possible decision value is v_0.*

Proof. Consider a run of the HO machine $\langle \mathcal{A}_{T,E}, \mathcal{P}_\alpha \rangle$ such that all the initial values are equal to v_0.

First, by induction on r, we show that:

$$\forall r > 0 :\ Q^r(v_0) = \Pi$$

Note that according to the code of $\mathcal{A}_{T,E}$, p belongs to $Q^r(v_0)$ if and only if $x_p^{(r-1)} = v_0$, and so $Q^r(v_0) = \{p \in \Pi : x_p^{(r-1)} = v_0\}$.

Basic case: $r = 1$. All the initial values are equal to v_0. Therefore, every process sends a message with value v_0 at round 1.

Inductive step: Let $r > 1$, and suppose that $Q^{r-1}(v_0) = \Pi$. Let p be a process that updates its variable x_p at round $r-1$. Since $AHO(p, r-1) \leq \alpha$, each process p receives at most α values distinct from v_0 at round $r-1$. Therefore, either p does not modify x_p at the end of round r which remains equal to v_0, or p receives strictly more than T messages at round r, and thus strictly more than $T-\alpha$ messages with value v_0 and at most α values different from v_0. In the latter case, p sets x_p to v_0 since $T \geq 2\alpha$. This shows that definitely, $x_p^{(r-1)} = v_0$. Therefore, $Q^r(v_0) = \Pi$.

Let p be a process that makes a decision at some round $r_0 > 0$. We have just shown that $Q^{r_0}(v_0) = \Pi$. When $|AHO(p, r_0)| \leq \alpha$ holds, p receives at most α messages with value different to v_0. Since $E \geq \alpha$, the code entails p to decide v_0 at round r. □

For liveness, we introduce the time-invariant communication predicate $\mathcal{P}^{\mathcal{A},live}$, given in Figure 1, which (i) ensures all x_q to eventually be identical, and (ii) guarantees that each process then hears of sufficiently many processes to make a decision.

Proposition 3 (Termination). *If $n > E \geq \frac{n}{2} + \alpha$ and $n > T \geq 2(n + 2\alpha - E)$, then any run of the HO machine $\langle \mathcal{A}_{T,E}, \mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{A},live} \rangle$ satisfies the Termination clause of consensus.*

Proof. Since we have $n > E$ and $n > T$, the set of all heard-of collections $(HO(p,r); SHO(p,r))_{p \in \Pi,\ r>0}$ that fulfill $\mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{A},live}$ is non-empty, i.e., there exist runs that fulfill $\mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{A},live}$.

Let r be a round in a run of the HO machine $(\mathcal{A}_{T,E}, \mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{A},live})$ such that:

$$\begin{array}{c} \exists \Pi_r^1, \Pi_r^2 \subseteq \Pi \ s.t.\ (|\Pi_r^1| > E - \alpha) \wedge (|\Pi_r^2| > T) : \\ (\forall p \in \Pi_r^1,\ HO(p,r) = SHO(p,r) = \Pi_r^2) \end{array}$$

Because of $E \geq \frac{n}{2} + \alpha$, the code implies that all x_p for p in Π_r^1 are equal to some common value $v \in V$. Since $|\Pi_r^1| > E - \alpha$, it follows that $|Q^{r+1}(v)| > E - \alpha$. Because of $T \geq 2(n + 2\alpha - E)$, a similar argument as the one used in Lemma 5 shows that every process q that updates x_q at round $r' > r$ definitely sets it to v. Moreover, from $\mathcal{P}^{\mathcal{A},live}$ we have $\forall r > 0,\ \forall q \in \Pi,\ \exists r_q > r\ :\ |HO(q, r_q)| > T$. Therefore, there exist a round $r' > r$ such that every process q in $\Pi \setminus \Pi_r^1$ updates x_q after round r and by the end of round r'. Then we deduce that for each process $q \in \Pi$, we have $x_q^{(r')} = v$. Finally, since

$$\forall r > 0,\ \forall p \in \Pi,\ \exists r_p > r\ :\ |SHO(p, r_p)| > E,$$

we know that every process $p \in \Pi$ eventually receives strictly more than E messages with value v at some round $r_p > r'$, and so decides v. □

Combining Propositions 1, 2, and 3, we spin-off the following theorem:

Theorem 1. *If $n > E$ and $n > T \geq 2(n + 2\alpha - E)$, then the HO machine $\langle \mathcal{A}_{T,E}, \mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{A},live} \rangle$ solves consensus.*

PROOF. Obviously, $n > T \geq 2(n + 2\alpha - E)$ implies $E \geq \frac{n}{2} + \alpha$. The Agreement clause is a straightforward consequence of Proposition 1. For Integrity, we just check that $E \geq \alpha$ and $T \geq 2\alpha$ are both ensured by $n \geq E \geq \frac{n}{2} + \alpha$ and $T \geq 2(n + 2\alpha - E)$, respectively. Indeed for all $\alpha \geq 0$, we have:

$$E \geq \frac{n}{2} + \alpha \;\Rightarrow\; E \geq \alpha.$$

Moreover

$$E \leq n \;\wedge\; T \geq 2(n + 2\alpha - E) \;\Rightarrow\; T \geq 2\alpha.$$

Termination directly follows from Proposition 3. □

3.3 Consensus with $\mathcal{A}_{T,E}$ solutions

At this point, we have to examine whether for any integer α, $0 \leq \alpha \leq n$, there exist T and E such that $\langle \mathcal{A}_{T,E}, \mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{A},live} \rangle$ solves consensus. In order to answer this question, Theorem 1 shows that it is sufficient to solve the following inequations:

$$n > E \tag{4}$$

$$n > T \geq 2(n + 2\alpha - E) \tag{5}$$

Obviously, (4) and (5) imply $\alpha < \frac{n}{4}$. Conversely, assume $0 \leq \alpha < \frac{n}{4}$. We let $\alpha = \frac{n}{4} - \epsilon$ with $\frac{n}{4} \geq \epsilon > 0$.

If we choose $E = n - \epsilon$, then we have $2(n + 2\alpha - E) = n - 2\epsilon$, and so $2(n + 2\alpha - E) < n$. It follows that if $\alpha < \frac{n}{4}$, then there exist T and E that satisfy (4) and (5), and so such that $\langle \mathcal{A}_{T,E}, \mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{A},live} \rangle$ solves consensus.

This naturally leads us to question what are the "best choices" for T and E, for any given integer α, $0 \leq \alpha < \frac{n}{4}$. Regarding the predicate $\mathcal{P}^{\mathcal{A},live}$, the best choices for T and E is taking them as small as possible. However, T and E are not independent of each other because of the requirement $T \geq 2(n + 2\alpha - E)$. The "best choices" for T and E are paradoxical under the latter constraint, and so there is no best choice for T and E without specifying additional prerequisites on T and E. Note that roughly speaking, in $\mathcal{P}^{\mathcal{A},live}$, T plays the role of a minimal "liveness communication threshold" whereas E plays the role of a minimal "safety communication threshold" guaranteeing $\mathcal{A}_{T,E}$'s correctness.

Without specific assumptions on communication, we look for T and E such that $E = T$. This yields the following inequality with $\alpha = \frac{n}{4} - \epsilon$:

$$n > E \geq 3n - 4\epsilon - 2E \tag{6}$$

Obviously, inequality (6) can be replaced by $E \geq n - \frac{4}{3}\epsilon$. We easily check that $E = n - \frac{4}{3}\epsilon$ is a solution of the above system, since $\epsilon \leq \frac{n}{4}$. In this case, we have $E = T = \frac{2}{3}(n + 2\alpha)$. The above discussion can be summarized as follows:

PROPOSITION 4. *For any integer $0 \leq \alpha < \frac{n}{4}$, the HO machine $\langle \mathcal{A}_{E,E}, \mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{A},live} \rangle$ with $E = \frac{2}{3}(n + 2\alpha)$ solves consensus.*

Note that in the benign case (i.e., $\alpha = 0$), we get $E = T = \frac{2n}{3}$, and that $\mathcal{A}_{\frac{2n}{3},\frac{2n}{3}}$ exactly coincides with the *OneThirdRule* algorithm in [6].

Interestingly, Theorem 1 shows that with $\mathcal{A}_{T,E}$, we do not lose any of the basic properties of the *OneThirdRule* algorithm even in the presence of corrupted communication. Indeed, like *OneThirdRule*, $\mathcal{A}_{T,E}$ is quite resilient since, to be safe, it just requires $\mathcal{P}_\alpha$. Furthermore, from each initial configuration, there is at least one run of $\mathcal{A}_{T,E}$ that achieves consensus in two rounds, and in the case all the initial values are equal, there is a run that achieves consensus just in one round. As the *OneThirdRule* algorithm, $\mathcal{A}_{T,E}$ is thus *fast* in the sense of [12].

Algorithm 2 The $\mathcal{U}_{T,E,\alpha}$ algorithm

```
1: Initialization:
2:   x_p ∈ V; initially x_p = v_p      /* v_p is the initial value of p */
3:   vote_p ∈ V; initially vote_p = ?

4: Round r = 2φ − 1:
5:   S_p^r:
6:     send ⟨ x_p ⟩ to all processes
7:   T_p^r:
8:     if received > T values equal to v with v ∈ V then
9:       vote_p := v

10: Round r = 2φ:
11:   S_p^r:
12:     send ⟨ vote_p ⟩ to all processes
13:   T_p^r:
14:     if received at least α + 1 messages with value v ≠ ? then
15:       x_p := v
16:     else
17:       x_p := v_0                                  /* default value */
18:     if received > E messages with value v then
19:       DECIDE(v)
20:     vote_p := ?
```

4. WHEN COMMUNICATION IS LIVE BUT NOT SO SAFE

We now describe a second consensus algorithm tolerating more corruptions, denoted $\mathcal{U}_{T,E,\alpha}$ and given as Algorithm 2, that is designed to work under the predicates $\mathcal{P}_\alpha$, $\mathcal{P}^{\mathcal{U},safe}$, and $\mathcal{P}^{\mathcal{U},live}$, given in (2), (7), and (8) in Figure 2. We use the same approach as for $\mathcal{A}_{T,E}$, namely we parametrize the various thresholds that occur in the *UniformVoting* algorithm given in [6].

4.1 The $\mathcal{U}_{T,E,\alpha}$ algorithm

Informally, the algorithm works as follows. The $\mathcal{U}_{T,E,\alpha}$ algorithm is organized into phases, each composed of two rounds. Each process p maintains a variable x_p, initialized to p's initial value. At the first round of each phase ϕ, every process p sends x_p to all. Then, provided that p receives sufficiently many messages with some proper value $v \in V$, it votes for v (if so, we shall say that p casts a *true vote*); otherwise, p votes for "?". At the second round of ϕ, each process p sends its vote (in $V \cup \{?\}$). Then it updates x_p by setting it to some value $v \in V$ if p can be sure that at least one process has voted for v. Otherwise, p adopts a default value $v_0 \in V$ as new estimate x_p. When messages may be corrupted, being sure that at least one process voted for v, is no more guaranteed by the reception of just one vote for v as in the *UniformVoting* algorithm. This leads us to consider the communication predicate $\mathcal{P}_\alpha$, and then to parametrize the threshold for updating the x_p (line 14), substituting $\alpha+1$ for 1. Indeed, when $\mathcal{P}_\alpha$ holds, if p receives at least $\alpha + 1$ messages with value $v \in V$, then at least one process has definitely voted for v. At the end of phase ϕ, a process that has received sufficiently votes for some value v decides v.

4.2 Correctness of the $\mathcal{U}_{T,E,\alpha}$ algorithm

As for arguing the correctness of $\mathcal{A}_{T,E}$, we use $Q^r(v)$ and $R_p^r(v)$, representing the set of processes that ought to send

v, resp. from which p receives v at round r. First we give an obvious but useful technical lemma:

LEMMA 6. *Let A and B be two subsets of Π. Then from $|A| + |B| > n + \alpha$ follows $|A \cap B| > \alpha$.*

PROOF. Because of $|A \cap B| = |A| + |B| - |A \cup B|$ and $|A \cup B| \leq n$ we have $|A \cap B| \geq |A| + |B| - n$. Further, since $|A| + |B| > n + \alpha$, we have $|A \cap B| > n + \alpha - n = \alpha$, as needed. □

Like for $\mathcal{A}_{T,E}$, choosing $E \geq \frac{n}{2}$ renders the decision rule "deterministic", as expressed in the following lemma.

LEMMA 7. *If $E \geq \frac{n}{2}$, then at most one value can be decided per process per round.*

Now we are going to prove that if $T \geq \frac{n}{2} + \alpha$, then the basic argument that ensures the safety of *UniformVoting* — namely, there is at most one true vote per round — still holds for $\mathcal{U}_{T,E,\alpha}$.

LEMMA 8. *Suppose that $T \geq \frac{n}{2} + \alpha$. Under the predicate $\mathcal{P}_\alpha$, if a process votes for $v \neq ?$ at round r, then every process votes for v or $?$ at round r. Moreover, the only possible values for any x_q at the end of round $r+1$ is v or v_0, i.e., $x_q^{(r+1)} \in \{v, v_0\}$.*

PROOF. Suppose that at some round r, two processes p and q cast true votes with values v and v', respectively. The code implies that $|R_p^r(v)| > T$ and $|R_q^r(v')| > T$. Under the communication predicate $\mathcal{P}_\alpha$, Lemma 1 implies $|Q^r(v)| > T - \alpha$ and $|Q^r(v')| > T - \alpha$. Suppose by contradiction that $v \neq v'$. Then $Q^r(v)$ and $Q^r(v')$ are disjoints sets, and so $|Q^r(v) \cup Q^r(v')| = |Q^r(v)| + |Q^r(v')|$. Since $T \geq \frac{n}{2} + \alpha$ it follows $|Q^r(v) \cup Q^r(v')| > n$. Thereby, there is at most one possible true vote value per round.

Now assume that process p votes for v at round r. The code enforces any process q to update x_q at the end of round $r+1$. Suppose q updates x_q by setting it to v' with $v' \notin \{v, ?\}$. This implies $|R_q^{r+1}(v')| \geq \alpha + 1$. Under the communication predicate $\mathcal{P}_\alpha$, we derive that at least one process hast voted for v' at round r, which contradicts that v is the only true vote at round r. □

Contrary to $\mathcal{A}_{T,E}$, the predicate $\mathcal{P}_\alpha$ is not sufficient to guarantee the Agreement clause for $\mathcal{U}_{T,E,\alpha}$. Indeed, the following lemma leads us to introduce the additional communication predicate $\mathcal{P}^{\mathcal{U},safe}$:

$$\forall p \in \Pi,\ \forall r > 0 : \\ |SHO(p,r)| > max(n + 2\alpha - E - 1, T, \alpha) \qquad (7)$$

Note that $\mathcal{P}^{\mathcal{U},safe}$ involves both safety and liveness requirements for communication.

LEMMA 9. *Suppose that $T \geq \frac{n}{2} + \alpha$. Under the predicate $\mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{U},safe}$, if a process p decides v at round r, then each process q necessarily sets x_q to v at the end of round r.*

PROOF. Consider an arbitrary run of $\langle \mathcal{U}_{T,E,\alpha}, \mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{U},safe} \rangle$ in which some process p decides value v at round r. The code implies that $|R_p^r(v)| > E$. By Lemma 1, it follows $|Q^r(v)| > E - \alpha$. Hence, under the predicate $\mathcal{P}^{\mathcal{U},safe}$, we have:

$$\forall q \in \Pi\ :\ |Q^r(v)| + |SHO(q,r)| > n + \alpha.$$

Lemma 6 ensures that $|Q^r(v) \cap SHO(q,r)| > \alpha$. Therefore, q receives at least $\alpha + 1$ messages with value v at round r. Since $T \geq \frac{n}{2} + \alpha$, the second part of Lemma 8 shows that q definitely sets x_q to v. □

From the above lemmas, we derive a sufficient condition on E, T and communication which ensures that the HO machine $\langle \mathcal{U}_{T,E,\alpha}, \mathcal{P}_\alpha \rangle$ satisfies the Agreement clause of consensus.

PROPOSITION 5 (AGREEMENT). *If $E \geq \frac{n}{2} + \alpha$ and $T \geq \frac{n}{2} + \alpha$, then any run of the HO machine $\langle \mathcal{U}_{T,E,\alpha}, \mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{U},safe} \rangle$ satisfies the Agreement clause of consensus.*

PROOF. Let $\phi_0 > 0$ be the first phase at which a process p makes a decision, and let v be the value that p decides at round $2\phi_0$. We first show, by induction on ϕ, that under the predicate $\mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{U},safe}$, we have:

$$\forall \phi \geq \phi_0 :\ |Q^{2\phi}(v)| > E - \alpha.$$

$\phi = \phi_0$: Since p decides v at round $2\phi_0$, the code implies that $|R_p^{2\phi_0}(v)| > E$. Hence, under $\mathcal{P}_\alpha$, Lemma 1 ensures that $|Q^{2\phi_0}(v)| > E - \alpha$, as needed.

Inductive step: Let $\phi > \phi_0$, and assume $|Q^{2(\phi-1)}(v)| > E - \alpha$. Since $T \geq \frac{n}{2} + \alpha$, under $\mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{U},safe}$ the same argument as the one in Lemma 9 ensures that each process q sets x_q to v at round $2(\phi - 1)$. Moreover, $\mathcal{P}^{\mathcal{U},safe}$ then guarantees that every process votes for v at round $2\phi - 1$. Hence, $|Q^{2\phi}(v)| = n > E - \alpha$.

Let p' be a process that decides v' at round 2ϕ. We proceed by contradiction and assume $v \neq v'$. On the one hand, the code implies that $|R_{p'}^{2\phi}(v')| > E$. Under $\mathcal{P}_\alpha$, Lemma 1 implies $|Q^{2\phi}(v')| > E - \alpha$. On the other hand, we have just shown that $|Q^{2\phi}(v)| > E - \alpha$. Since $v \neq v'$, $Q^{2\phi}(v)$ and $Q^{2\phi}(v')$ are disjoint sets. It follows from $E \geq \frac{n}{2} + \alpha$ that $|Q^{2\phi}(v) \cup Q^{2\phi}(v')| > n$, a contradiction. □

Similarly, we derive a sufficient condition on communication which ensures that the HO machine $\langle \mathcal{U}_{T,E,\alpha}, \mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{U},safe} \rangle$ to satisfies the Integrity clause of consensus.

PROPOSITION 6 (INTEGRITY). *If $E \geq \frac{n}{2} + \alpha$, then in any run of the HO machine $\langle \mathcal{U}_{T,E,\alpha}, \mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{U},safe} \rangle$ such that all the initial values are equal to some value v, v is the only possible decision value.*

PROOF. Consider a run of the HO machine $\langle \mathcal{U}_{T,E,\alpha}, \mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{U},safe} \rangle$ such that all the initial values are equal to v.

Let p be a process that decides value v' at phase ϕ_0. Suppose for contradiction that $v \neq v'$. The code gives $|R_p^{2\phi_0}(v')| > E$. By Lemma 1, it follows that under $\mathcal{P}_\alpha$, $|Q^{2\phi_0}(v')| > E - \alpha$. Besides, using the same inductive argument as above, we show that under $\mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{U},safe}$, at every phase $\phi > 0$, $|Q^{2\phi}(v)| > E - \alpha$. In particular, we have $|Q^{2\phi_0}(v)| > E - \alpha$. Since $v \neq v'$, $Q^{2\phi_0}(v)$ and $Q^{2\phi_0}(v')$ are disjoint sets. It follows that because of $E \geq \frac{n}{2} + \alpha$, $|Q^{2\phi_0}(v) \cup Q^{2\phi_0}(v')| > n$, a contradiction. □

For liveness, we exhibit the communication predicate given in Figure 2. The first part forces all processes to hear of the same subset of processes at round $2\phi_0$, and precludes any corruption at this round. It ensures all x_p to be identical (all equal to some value $v \in V$) at the end of phase ϕ_0. The second part in $\mathcal{P}^{\mathcal{U},live}$ limits the number of corruption at the

$$\forall \phi,\ \exists \phi_0 \geq \phi,\ \exists \Pi_0 \subseteq \Pi, \forall p \in \Pi :$$
$$HO(p, 2\phi_0) = SHO(p, 2\phi_0) = \Pi_0 \wedge |SHO(p, 2\phi_0 + 1)| > T \wedge |SHO(p, 2\phi_0 + 2)| > max(E, \alpha) \quad (8)$$

Figure 2: Predicate $\mathcal{P}^{\mathcal{U},live}$

next phase, and thus guarantees that at phase $\phi_0 + 1$, all processes vote for v. The last part of the predicate ensures that every process definitely hears of vote v sufficiently often to make a decision at the end of phase $\phi_0 + 1$.

Thus, combining this remark with Propositions 5 and 6, we spin-off the following theorem:

THEOREM 2. *If $n > E \geq \frac{n}{2} + \alpha$ and $n > T \geq \frac{n}{2} + \alpha$ and $n > \alpha$, then the HO machine $\langle \mathcal{U}_{T,E,\alpha}, \mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{U},safe} \wedge \mathcal{P}^{\mathcal{U},live} \rangle$ solves consensus.*

PROOF. Since $n > E$, $n > T$ and $n > \alpha$, then the set of all heard-of collections $(HO(p, r); SHO(p, r))_{p \in \Pi,\ r > 0}$ that fulfill $\mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{U},safe} \wedge \mathcal{P}^{\mathcal{U},live}$ is non-empty when $E \geq \frac{n}{2} + \alpha$.

By Proposition 5 and 6, the Agreement and Integrity clauses are guaranteed when $\mathcal{P}^{\mathcal{U},safe}$ holds.

Now, we show that any run ρ of the HO machine $\langle \mathcal{U}_{T,E,\alpha}, \mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{U},safe} \wedge \mathcal{P}^{\mathcal{U},live} \rangle$ satisfies the Termination clause of consensus. Let ϕ_0 be a phase for which there exists a subset $\Pi_0 \subseteq \Pi$ such that:

$$\forall p \in \Pi,\ HO(p, 2\phi_0) = SHO(p, 2\phi_0) = \Pi_0,$$
$$\wedge$$
$$|SHO(p, 2\phi_0 + 1)| > T \text{ and } |SHO(p, 2\phi_0 + 2)| > max(E, \alpha).$$

The code implies that at the end of phase ϕ_0, all the x_p are equal to some unique value $v \in V$. Because of $T \geq \frac{n}{2} + \alpha$, every process casts a true vote at round $2\phi_0 + 1$, and by Lemma 8, each vote is for v. Hence every process receives strictly more than E messages with value v at round $2(\phi_0+1)$. Since $E \geq \frac{n}{2}$, Lemma 7 implies that every process decides v at this round. □

4.3 Consensus with $\mathcal{U}_{T,E,\alpha}$ solutions

At this point, we have to examine whether for any given integer α, $0 \leq \alpha \leq n$, there exist T and E such that $\langle \mathcal{U}_{T,E,\alpha}, \mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{U},safe} \wedge \mathcal{P}^{\mathcal{U},live} \rangle$ solves consensus. From the expression of $\mathcal{P}^{\mathcal{U},live}$, we deduce that this is equivalent to solve the following inequations:

$$n > T \geq \frac{n}{2} + \alpha \quad (9)$$
$$n > n + 2\alpha - E - 1 \quad (10)$$
$$n > E \geq \frac{n}{2} + \alpha \quad (11)$$

Obviously, (11) implies $\alpha < \frac{n}{2}$, and so (11) implies (10). Therefore, the latter three inequalities are equivalent to only (9) and (11).

As mentioned above, (11) implies $\alpha < \frac{n}{2}$. Conversely, assume $0 \leq \alpha < \frac{n}{2}$. Trivially, $E = T = \frac{n}{2} + \alpha$ satisfy (9) and (11). It follows that if $\alpha < \frac{n}{2}$, then there exist T and E such that $\langle \mathcal{U}_{T,E,\alpha}, \mathcal{P}_\alpha \wedge \mathcal{P}^{\mathcal{U},safe} \wedge \mathcal{P}^{\mathcal{U},live} \rangle$ solves consensus.

Note that in the case $E = T = \frac{n}{2} + \alpha$, the predicate $\mathcal{P}^{\mathcal{U},safe}$ guarantees that all the safe heard-of sets $SHO(p, r)$, and so all the $HO(p, r)$, are of cardinality greater than $\frac{n}{2}+\alpha$. Hence contrary to $\mathcal{A}_{T,E}$, the safety of $\mathcal{U}_{T,E,\alpha}$ is guaranteed by some permanent "liveness" properties of communication. Besides, $\mathcal{U}_{T,E,\alpha}$ tolerates more corruptions than $\mathcal{A}_{T,E}$ ("$\alpha < \frac{n}{2}$" instead of "$\alpha < \frac{n}{4}$").

5. RELATED WORK

Most previous work on consensus algorithms in the presence of non-benign faults relies on the paradigm of Byzantine processes, i.e., on *permanent* and *static* value faults. For the synchronous case the seminal papers are [13, 17] (which also include lower bounds). The partial synchronous case is considered in [2, 7].

Byzantine variants for the famous Paxos algorithm include [1,4,14,22]. An algorithm for Fast Byzantine Paxos has been proposed in [16]. The Byzantine Paxos definition is in fact closer related to Byzantine agreement than to consensus, and thus it is related to the notion of faulty processes as Integrity only restricts delivery of messages in the case of a *correct* broadcaster.

There exists only little work on dynamic and transient value faults: Santoro and Widmayer [18,19] as well as Schmid *et al.* [20] provide lower bounds on agreement problems in the presence of dynamic value faults. We will discuss the relation to the literature on lower bounds before we present some general discussion of the modelling of non-benign faults.

5.1 Correspondence to Lower Bounds

Santoro and Widmayer [18, 19] show that agreement is already impossible to solve given a quite strong predicate, with as few as $\lfloor n/2 \rfloor$ faulty transmissions per round. The problematic case is when these $\lfloor n/2 \rfloor$ value faults occur in blocks, i.e., in every round the outgoing links of one process are affected. In every round this may happen to a different process. On the other hand, our algorithms allow up to $n^2/4$ (for $\mathcal{A}_{T,E}$), resp. $n^2/2$ (for $\mathcal{U}_{T,E,\alpha}$), transmission faults per round in general. However, our result is not contradicting the lower bound: It comes from the fact that we distinguish between the safety and liveness of the consensus algorithm, and assume transient failures. So, e.g. for $\mathcal{A}_{T,E}$, for safety it is sufficient that less than $n/4$ corruptions per process and round occur. But in order to ensure termination, two rounds are necessary where the assumptions are much higher than those given by the lower bound of Santoro and Widmayer (cf. to predicate $\mathcal{P}^{\mathcal{A},live}$).

Another way to circumvent the impossibility of Santoro and Widmayer is given for synchronous systems by Schmid *et al.* [20]. They restrict the number of the transmission faults for each round that correct processes may experience (both for outgoing and incoming messages), which prevents the blocks of faults of [18]. The synchronous system considered in [20] translates into a strong predicate, i.e., a large number of messages must be transmitted correctly in every round. This contrasts our work that separates predicates for safety and liveness, which leads our algorithms to require quite weak safety predicates but stronger ones for termination. They also provide lower bounds in a failure model which restricts both benign and non-benign transmission failures both on sender and receiver side. One special instance of their results where only value faults are considered yields that at most $n/4$ faults may occur in each round per sender and receiver. The predicates of $\mathcal{U}_{T,E,\alpha}$ reveal a trade-off.

$\mathcal{U}_{T,E,\alpha}$ allows that in most rounds more than $n/4$ (up to nearly $n/2$) received messages may be corrupted given that there exist rounds where processes experience much less corruptions.

The $\mathcal{A}_{T,E}$ algorithm is *fast* in the sense that for each initial configuration there is a run where all processes decide in two rounds [5, 12]. For this, Martin and Alvisi [16] have established $(4n+1)/5$ as a lower bound for the number of correct processes. For a similar reason as for the Santoro/Widmayer lower bound, here we have also no contradiction: Our model is more fine grained in two dimensions than theirs: First in the spacial dimension, where we reason in a per-link basis instead of a per-process basis. And, more important for this comparison, it is finer in the temporal dimension, since the quorums are measured in a per-round basis instead of assuming permanent faults. Thus, although we have a fast algorithm, we can have in each round, up to $(n-1)/4$ processes that may emit corrupted information (cf. Section 3.3). On the other hand, for deciding we need at least one round where no process emits corrupted information. Further, we do not rely on separate recovery protocol and signatures, where the latter is not only computationally expensive but also questionable from a theoretical point of view: To the best of our knowledge, there exists no satisfactory rigorous definition of signatures in the context of Byzantine-tolerant distributed algorithms.

Lamport [11] has conjectured the lower bound $N > 2Q + F + 2M$ for Byzantine consensus, where N is the number of acceptors, F is the maximum number of Byzantine acceptors despite which liveness is ensured, M is the maximum number of Byzantine acceptors despite which consensus safety is ensured, and Q is the number of Byzantine acceptors despite which the protocol is fast. We attain this bound with both of our algorithms: In the case the algorithm should just be safe (that is, it is not necessarily fast and does not satisfy the liveness conditions; corresponding to $F = Q = 0$), $\mathcal{U}_{T,E,\alpha}$ achieves safety with $\alpha = (n-1)/2$. If the algorithm should be safe and fast (but does not necessarily satisfy the liveness conditions, i.e., $F = 0$), we present such an algorithm via $\mathcal{A}_{T,E}$ with $\alpha = (n-1)/4$. For both algorithms, $F = 0$ as we have stronger conditions for liveness, i.e., our algorithms cannot tolerate classic Byzantine (process) failures. Note, however, that we assume dynamic faults while Lamport's lower bound consider static Byzantine faults.

5.2 Relation to other non-benign fault models

Figure 3 shows, how an HO machine, in general, can suffer from corruption. The model at the top of the figure, where no value faults occur, is the benign case.

The model at the bottom, where in an execution transitions might deviate from what is prescribed by transition functions and transmissions might deviate from what is prescribed by sending functions, corresponds to the classical Byzantine model assumption [13, 17].

Naturally, there are two models that lie in between these two extreme cases. Based on the left, where transmissions always follow sending functions, but state transitions might not follow transition functions (thus also state corruptions may occur), one could design (broadcast based) algorithms that send the same message to all other processes. Given that transmissions follow the sending function, it is not possible that two different values from the same sender are received. Consequently, faulty behavior is restricted to "symmetrical failures" [20] (also termed "identical Byzantine" in [3]). Similar behavior can be implemented (if not a priori present) e.g. with *signed messages*. Thus, signatures are an important implementation concept in order to ensure such behavior. Given the lack of a formal definition of signatures, we believe that they should be studied in the context of predicate implementations but not as part of consensus algorithms. This also allows to separate between the mechanisms of consensus algorithms on the one hand and signatures on the other hand.

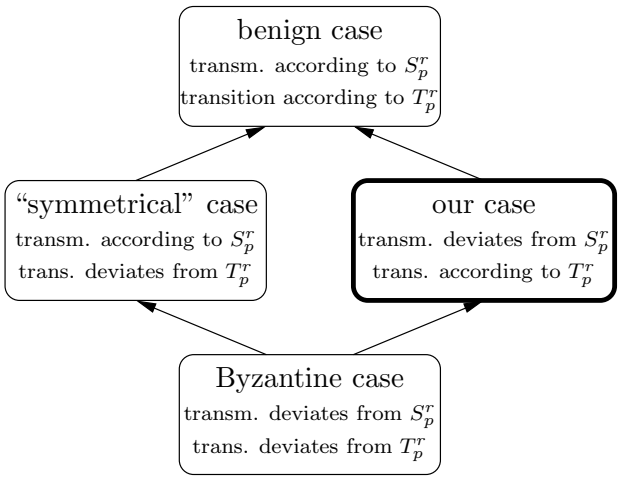

Figure 3: Possible types of corruption

Finally, the right side case is the one considered in this paper, which is also the one of Santoro and Widmayer [18, 19]. One could argue that this approach is only of theoretical, but not of practical interest, given that techniques like *signatures* or *error correcting codes* could be used to transform value into benign failures, and consequently the right side model to the benign case. Since there are applications where this is not feasible due to the involved coverage (error correcting codes cannot correct all errors) or due to extensive computational cost, we provide solutions for the cases where such techniques cannot be used to eliminate *all* value faults. However, such techniques can be used to increase the coverage of our predicates.

Although we assume that T_p^r is always followed, the classical Byzantine assumptions can be expressed in our model. This stems from the fact that Byzantine processes are *static* and *permanent* faults. Thus, from the perspective of an outside observer it is indistinguishable whether such a process has a corrupted state or not.[3] Consequently, we can give, e.g., a predicate for a synchronous system with reliable links and at most f Byzantine processes:

$$|SK| \geq n - f$$

and a predicate for an asynchronous system with reliable links and at most f Byzantine processes:

$$\forall p \in \Pi : \ \forall r > 0 : \ |HO(p,r)| \geq n - f \ \wedge \ |AS| \leq f.$$

6. CONCLUSION

In this paper we investigated consensus in the presence of transient and dynamic value faults. To this end we generalized the round based HO model of [5, 6] which only consid-

[3]This was also observed in [14].

Alg.	Predicate for Safety	Predicate for Liveness	Conditions
$\mathcal{A}_{T,E}$	$\forall r > 0, \forall p \in \Pi :$ $\lvert AHO(p,r)\rvert \leq \alpha$	$\forall r_0 > 0,\ \exists r \geq r_0,\ \exists \Pi_r^1, \Pi_r^2 \subseteq \Pi$ s.t.$(\lvert\Pi_r^1\rvert > E - \alpha) \wedge (\lvert\Pi_r^2\rvert > T)$ $\wedge(\forall p \in \Pi_r^1,\ HO(p,r) = SHO(p,r) = \Pi_r^2)$ $\forall r > 0,\ \forall p \in \Pi,\ \exists r_p > r\ :\ \lvert HO(p,r_p)\rvert > T$ $\forall r > 0,\ \forall p \in \Pi,\ \exists r_p > r\ :\ \lvert SHO(p,r_p)\rvert > E$	$n > E$ $T \geq 2(n + 2\alpha - E)$
$\mathcal{U}_{T,E,\alpha}$	$\forall r > 0, \forall p \in \Pi :$ $\lvert AHO(p,r)\rvert \leq \alpha$ $\lvert SHO(p,r)\rvert > max(n + 2\alpha - E - 1, T, \alpha)$	$\forall \phi,\ \exists \phi_0 \geq \phi,\ \exists \Pi_0 \subseteq \Pi, \forall p \in \Pi :$ $HO(p, 2\phi_0) = SHO(p, 2\phi_0) = \Pi_0 \wedge$ $\lvert SHO(p, 2\phi_0 + 1)\rvert > T \wedge\ \lvert SHO(p, 2\phi_0 + 2)\rvert > max(E, \alpha)$	$n > E \geq \frac{n}{2} + \alpha$ $n > T \geq \frac{n}{2} + \alpha$

Table 1: Summary of results

ered benign faults. Our novel framework includes predicates that allow also to express value faults like the classic Byzantine assumption [13, 17] as well as the dynamic fault model found in [18].

In [18], Santoro and Widmayer prove an impossibility result for agreement problems in the dynamic setting. In contrast, we were interested in exploring system properties that would allow positive results: We introduced two consensus algorithms (derived from the benign case) that are suitable for systems in which the communication predicates given in Table 1 can be guaranteed. These predicates can be separated into safety and liveness conditions. Informally, the liveness conditions allow us to circumvent the impossibility of [18]. In contrast to classic literature, liveness of our algorithms does not rely on conditions that hold from some stabilization on, but only sporadically.

7. REFERENCES

[1] I. Abraham, G. Chockler, I. Keidar, and D. Malkhi. Byzantine disk paxos: optimal resilience with Byzantine shared memory. *Distributed Computing*, 18(5):387–408, 2006.

[2] M. K. Aguilera, C. Delporte-Gallet, H. Fauconnier, and S. Toueg. Consensus with Byzantine failures and little system synchrony. In *Dependable Systems and Networks (DSN 2006)*, pages 147–155, 2006.

[3] H. Attiya and J. Welch. *Distributed Computing*. John Wiley & Sons, 2nd edition, 2004.

[4] M. Castro and B. Liskov. Practical Byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems*, 20(4):398–461, 2002.

[5] B. Charron-Bost and A. Schiper. Improving fast paxos: being optimistic with no overhead. In *Pacific Rim Dependable Computing, Proceedings*, pages 287–295, 2006.

[6] B. Charron-Bost and A. Schiper. The Heard-Of model: Computing in distributed systens with benign failures. Technical report, EPFL, 2007.

[7] C. Dwork, N. Lynch, and L. Stockmeyer. Consensus in the presence of partial synchrony. *Journal of the ACM*, 35(2):288–323, Apr. 1988.

[8] E. Gafni. Round-by-round fault detectors (extended abstract): unifying synchrony and asynchrony. In *Proc. 16th Annual ACM Symposium on Principles of Distributed Computing (PODC'98)*, pages 143–152, Puerto Vallarta, Mexico, 1998. ACM Press.

[9] J. N. Gray. Notes on data base operating systems. In G. S. R. Bayer, R.M. Graham, editor, *Operating Systems: An Advanced Course*, volume 60 of *Lecture Notes in Computer Science*, chapter 3.F, page 465. Springer, New York, 1978.

[10] M. Hutle and A. Schiper. Communication predicates: A high-level abstraction for coping with transient and dynamic faults. In *Dependable Systems and Networks (DSN 2007)*, 2007.

[11] L. Lamport. Lower bounds for asynchronous consensus. In *Future Directions in Distributed Computing*, number 2584 in Lecture Notes in Computer Science, pages 22–23. Springer-Verlag, 2003.

[12] L. Lamport. Fast paxos. Technical Report MSR-TR-2005-12, Microsoft Research, 2005.

[13] L. Lamport, R. Shostak, and M. Pease. The Byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.

[14] B. Lampson. The ABCD's of paxos. In *Proc. 19th Annual ACM Symposium on Principles of Distributed Computing (PODC'01)*, page 13, New York, NY, USA, 2001. ACM Press.

[15] N. Lynch. *Distributed Algorithms*. Morgan Kaufman, 1996.

[16] J.-P. Martin and L. Alvisi. Fast Byzantine consensus. *Transactions on Dependable and Secure Computing*, 3(3):202–214, 2006.

[17] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *Journal of the ACM*, 27(2):228–234, 1980.

[18] N. Santoro and P. Widmayer. Time is not a healer. In *Proc. 6th Annual Symposium on Theor. Aspects of Computer Science (STACS'89)*, volume 349 of *LNCS*, pages 304–313, Paderborn, Germany, Feb. 1989. Springer-Verlag.

[19] N. Santoro and P. Widmayer. Distributed function evaluation in the presence of transmission faults. In *SIGAL International Symposium on Algorithms*, pages 358–367, 1990.

[20] U. Schmid, B. Weiss, and J. Rushby. Formally verified Byzantine agreement in presence of link faults. In *22nd International Conference on Distributed Computing Systems (ICDCS'02)*, pages 608–616, Vienna, Austria, July 2–5, 2002.

[21] G. Varghese and N. A. Lynch. A tradeoff between safety and liveness for randomized coordinated attack. *Inf. Comput.*, 128(1):57–71, 1996.

[22] P. Zieliński. Paxos at war. Technical Report UCAM-CL-TR-593, University of Cambridge, 2004.