Degradable Agreement with Hybrid Faults

(An Algorithm and Reliability-Safety Analysis)

Nitin H. Vaidya
Department of Computer Science
Texas A&M University
College Station, TX 77843-3112
Phone: 409-845-0512
E-mail: vaidya@cs.tamu.edu

Technical Report 93-037

August 1993

Abstract

Achieving traditional Byzantine agreement in the presence of arbitrary faults requires that the total number of nodes be larger than three times the number of faulty nodes. Two approaches have been proposed in the literature to circumvent this requirement: (i) hybrid fault model approach [4, 7] considers three types of faults of varying degree of severity, and (ii) degradable agreement approach [10, 11] relaxes the definition of agreement in the presence of excessive faults. This report combines the above two approaches and presents a degradable agreement algorithm for the hybrid fault model.

The report defines and evaluates reliability and safety measures for the proposed degradable agreement algorithm and concludes that degradable agreement can be used to trade reliability with safety. It is shown that the hybrid fault model results in better algorithms (with respect to reliability and/or safety) only when the probability of arbitrary faults is not very small.

1 Introduction

Consider a system consisting of a sender that wants to send a value to certain receivers. Various *agreement* algorithms have been proposed for this purpose (e.g [1, 2, 3]). that tolerate arbitrary (possibly malicious) failures. The requirement is typically that the fault-free receivers must all agree on the same value [2, 3]. Dolev [1] analyzes a weaker form of agreement. Prior work has shown that such agreements are impossible if a third of the nodes (or more) are faulty. In other words, the number of nodes in the system must be larger than three times the number of faulty nodes. Two very different approaches have been presented to circumvent this requirement. These approaches either weaken the fault model or weaken the definition of *agreement*.

- Various researchers [4, 5, 7] have presented agreement protocols that tolerate a mix of Byzantine failures and other less severe types of failures. As all the faults in the system are not necessarily Byzantine, these protocols are often able to tolerate more than a third of the nodes being faulty. We will use the *hybrid* fault model [4] that divides faults into three classes: arbitrary, symmetric and manifest (elaborated later).
- We recently presented the *degradable* agreement [11] approach for Byzantine faults. The degradable agreement protocol achieves traditional Byzantine agreement[1] [3] (hereafter referred to as Lamport's Byzantine agreement) up to a certain number of Byzantine faults and a degraded form of agreement with a higher number of faults. The degraded form of agreement allows the fault-free receivers to agree on at most two different values, one of which is necessarily the default value.[2] Note that Lamport's agreement requires all nodes to agree on an identical value. By weakening the definition of agreement when excessive faults exist, our approach can tolerate more than a third of the nodes being faulty.

[1] Although Byzantine agreement was defined by Lamport, Shostak and Pease [3], for brevity we refer to it as Lamport's Byzantine agreement.

[2] Default value, denoted V_d, is distinguishable from all other values.

This report presents an agreement protocol that combines the above two approaches. Specifically, we present a degradable agreement algorithm for the hybrid fault model presented in [4]. Also, we define safety and reliability measures and show that the proposed algorithm effectively trades reliability with safety. This report is organized as follows. Section 2 presents the hybrid fault model and Section 3 defines degradable agreement. An algorithm for achieving degradable agreement with hybrid fault model is presented in Section 4. Section 5 evaluates reliability and safety measures for the proposed algorithm, and also shows that the hybrid fault model is useful only when arbitrary faults occur with non-negligible probability. Conclusions are presented in Section 6.

2 Hybrid Fault Model

In the hybrid fault model [4, 7], the faults are divided into three classes (in order of increasing severity): *manifest* faults, *symmetric* faults and *arbitrary* faults. A manifest fault is one that produces detectably missing values or values that can be detected as bad by all non-faulty recipients [4]. Both symmetric and arbitrary faults can produce values that are not detectably bad. However, a symmetric fault delivers identical value to all recipients, whereas, an arbitrary fault behavior is unconstrained (e.g. may deliver different wrong values to different recipients). Thus, an arbitrary fault is equivalent to a Byzantine fault [3].

A value sent by a manifest-faulty node can always be detected to be erroneous by the recipient. Therefore, it is assumed that a manifest-faulty node always sends a value denoted by E; E is a value distinguishable from all other values including default value V_d. Default value V_d is also distinguishable from all other relevant values.

We use the following notation:

N = number of nodes in the system under consideration.

a = number of arbitrary-faulty nodes in the system.

s = number of symmetric-faulty nodes in the system.

c = number of manifest-faulty nodes in the system.

Note that symmetric fault is a restricted type of an arbitrary fault. This leads to the question of how to count the number of arbitrary and symmetric faults. For example consider a system containing 2 arbitrary-faulty and 3 symmetric-faulty nodes. It is clear that, for this case, we can count $a = 2$ and $s = 3$. As a symmetric-fault is less severe than an arbitrary-fault, in this case, an algorithm that can tolerate 3 arbitrary-faults and 2 symmetric-faults can tolerate the present fault situation. Therefore, for the purpose of analysis, one may classify one of the symmetric-faulty nodes as arbitrary-faulty, and assume $a = 3$ and $s = 2$ (even though in reality, $a = 2$ and $s = 3$). It turns out that this counter-intuitive classification is useful in accurately evaluating reliability and safety of the proposed approach. We will return to this subject in Section 5. Although a manifest-faulty node may also be classified as symmetric-faulty or arbitrary-faulty, our analysis does not gain from this.

The algorithm presented in this report achieves degradable agreement in the presence of a arbitrary-faults, s symmetric-faults and c manifest-faults, provided certain conditions are satisfied. The next section elaborates on these conditions as well as the definition of degradable agreement.

3 Degradable Agreement with Hybrid Fault Model

The system model can be described as follows. The system consists of a sender and some receivers. The sender wants to send its value to the receivers. The term "sender's value" means

- the value the sender wants to inform every receiver, when the sender is fault-free,
- the value the sender sends to every receiver, when the sender is symmetric-faulty, and
- value E, when the sender is manifest-faulty.

In the following, the term *node* may refer to the sender or a receiver. As noted earlier, V_d denotes the default value, which is distinguishable from all other relevant values.

Degradable Agreement with Hybrid Faults: The degradable agreement algorithm presented in this report satisfies the following two conditions. Note that the algorithm presented by Lincoln and Rushby [4] only satisfies a condition similar to first of the two conditions.

- if $N > 2(a+s)+c+u$ and $a \leq m$, then D.1 and D.2 stated below are satisfied.
- if $N > a+2m+2s+c$ and $a \leq u$, then D.3 and D.4 stated below are satisfied.

(D.1) If the sender is not arbitrary-faulty, then all the fault-free receivers must agree on the sender's value.

(D.2) If the sender is arbitrary-faulty, then the fault-free receivers must agree on an identical value.

(D.3) If the sender is not arbitrary-faulty, then the fault-free receivers may be partitioned into *at most* two classes. The fault-free receivers in one class must agree on the sender's value, and the fault-free receivers in the other class must all agree on the default value.

(D.4) If the sender is arbitrary-faulty, then the fault-free receivers may be partitioned into *at most* two classes. The fault-free receivers in one class must agree on the default value, and the fault-free receivers in the other class must all agree on an identical value.

Conditions D.1 and D.2 are identical to those satisfied by Lamport's Byzantine agreement [3]. Conditions D.3 and D.4 define degraded agreement. Observe that conditions D.3 and D.4 are strictly *weaker* than D.1 and D.2, respectively, i.e., satisfying D.1 (D.2) is sufficient to satisfy D.3 (D.4) but not vice-versa.

Note that when $m = u$, condition 1 above is equivalent to that satisfied by the algorithm in [7, 4]. Also, when only arbitrary failures occur, our algorithm achieves m/u-degradable agreement, as defined in [11].

Definition 1 *A system is said to be a conforming system, if at least one of the following conditions hold, where $m \leq u$:*

1. *$a \leq m$ and $N > 2(a+s)+c+u$, and*
2. *$a \leq u$ and $N > a+2m+2s+c$.*

4 An Algorithm for Degradable Agreement

We begin with definition of a special voting function to be used in our algorithm.

Definition 2 *σ-HVOTE (or σ-hybrid vote) of ν quantities $w_1, w_2, \cdots, w_\nu$ is α (where $\alpha \neq V_d$ and $\alpha \neq E$) if k quantities are α such that $k \geq \nu - k - c^{\#} + \sigma$, where $c^{\#}$ is the number of quantities that are equal to E. If no such α exists, then σ-hybrid vote is defined to be the default value V_d.*

For example, consider eight values $\alpha, \gamma, \beta, \alpha, \gamma, E, \gamma, \gamma$. 1-HVOTE of these eight values is γ, whereas 2-HOTE is V_d. Similarly, 1-HVOTE of $\alpha, E, E, E, E, \alpha, \beta, \beta$ is V_d. The following observations can be made from the definition of HVOTE.

1. HVOTE cannot equal E.
2. Provided $\sigma \geq 1$, if σ-HVOTE of some ν values is $\alpha \neq V_d$, then majority vote of the non-E values among these ν values is also α. The converse is not always true, however.
3. Provided $\sigma \geq 1$, the condition $k \geq \nu - k - c^{\#} + \sigma$ cannot hold true simultaneously for two different values, say α and β, such that $\alpha \neq V_d, E$ and $\beta \neq V_d, E$. A consequence of this observation is that, if the condition $k \geq \nu - k - c^{\#} + \sigma$ is true for some value γ ($\neq V_d, E$) then we can be certain that HVOTE is equal to γ.

The algorithm presented here uses two functions R and UnR, called the wrapper and unwrapper functions, respectively [4]. Function R and UnR are defined such that (i) for all $\alpha \neq V_d$, we have $R(\alpha) \neq E$ and $R(\alpha) \neq V_d$, (ii) $R(V_d) = V_d$ and (iii) $UnR(R(\alpha)) = \alpha$ for all

α. Lincoln and Rushby [4] suggest a simple implementation of R and UnR using bounded integers.[3] A similar implementation can be used here.

Algorithm HBYZ presented below may be viewed as a combination of the algorithms in [4] and [11]. HBYZ assumes that the nodes are fully connected. Following assumptions are made regarding messages when proving correctness of algorithm HBYZ:

(a) all messages are delivered correctly within a bounded delay,

(b) source of a received message can be identified, and

(c) presence or absence of a message can be correctly detected. Whenever a node detects a message to be absent, it assumes that the message contains value E. Detecting presence and absence of messages correctly requires that the clocks of various nodes be synchronized. This issue was discussed in [11].

We now present HBYZ(1) and HBYZ(t) (algorithm for $m = 0$ is omitted here). In these algorithms, the following notation is used. In HBYZ(1), n_1 is the number of nodes to which algorithm HBYZ(1) is applied. Similarly, in HBYZ(t), n_t is the number of nodes to which algorithm HBYZ(t) is applied. As N is the total number of nodes in the system, $n_m = N$. It can be seen that, $n_t = N - m + t$.

Algorithm HBYZ(1)

1. The sender sends its value to all the $(n_1 - 1)$ receivers.

2. For each i, let v_i be the value receiver i received from the sender. v_i is defined to be E if no value is received or a manifestly bad value is received from the sender. Receiver i broadcasts value $R(v_i)$ to all receivers, including itself. As there are $(n_1 - 1)$ receivers, each receiver now has $(n_1 - 1)$ values.

[3] Bounded integers can be used by observing that function R may be applied at most m times to any value in the agreement algorithm presented here, i.e. function R^{m+1} need not be well-defined.

3. Each receiver finds $(1+u-m)$-HVOTE of the (n_1-1) values, and uses the value obtained by applying UnR to the outcome of HVOTE.

Lemma 5 in the appendix proves some properties of algorithm HBYZ(1).

Algorithm HBYZ(t), $1 < t \leq m$

1. The sender sends its value to all the (n_t-1) receivers.
2. For each i, let v_i be the value receiver i received from the sender. v_i is defined to be E if no value is received or a manifestly bad value is received from the sender. Receiver i acts as the sender in algorithm HBYZ($t-1$) to send value $R(v_i)$ to each of the (n_t-1) receivers, including itself.
3. For receiver i, let w_j be the value receiver i obtained from receiver j in step 2 (using algorithm HBYZ($t-1$)). Thus, receiver i now has n_t-1 values $w_1, w_2, \cdots, w_{n_t-1}$. Receiver i finds $(t+u-m)$-HVOTE of the (n_t-1) values, and uses the value obtained by applying UnR to the outcome of HVOTE.

A fault-free sender's value is never equal to E (by definition of E). Additionally, in HBYZ(1) and HBYZ(t), observe that each fault-free receiver i transmits value $R(v_i)$ in step 2. As $R(v_i) \neq E$, $\forall v_i$, a fault-free node never transmits a message with value E.

Algorithm HBYZ(m) is used to achieve degradable agreement. The theorem below implies that HBYZ(m) achieves desired degradable agreement.

Theorem 1 *Given $u \geq m$ and a system with N nodes the following conditions hold for HBYZ(m).*

1. *if $N > 2(a+s)+c+u$ and $a \leq m$, then D.1 and D.2 are satisfied.*
2. *if $N > a+2m+2s+c$ and $a \leq u$, then D.3 and D.4 are satisfied.*

Proof: The appendix presents a proof of this theorem. □

5 Reliability and Safety Analysis

Degradable agreement is designed specifically to achieve a degraded version of agreement in the presence of excessive *arbitrary* faults. Clearly, in a system where arbitrary faults are much less likely to occur than symmetric and manifest faults, it may not be worthwhile to use degradable agreement. In fact, as seen later, it may not be worthwhile using even the Byzantine agreement algorithm. Although the likelihood of arbitrary hardware failures may be small in practice, arbitrary-failure model is of interest in *open distributed systems (ODS)* [6]. An ODS consists of *trusted* as well as *non-trusted* nodes. The non-trusted nodes may demonstrate arbitrary behavior, possibly in the hands of a malicious user. The degradable agreement algorithm would be suitable in such environments.

To evaluate degradable agreement, we define two parameters: reliability and safety. *Reliability* is defined as the probability that the system is in a *state* where it can be guaranteed that conditions D.1 and D.2 can be satisfied. The *state* of the system is defined by the 3-tuple (a, s, c). Our definition of reliability is identical to that used in [8]. *Safety* is defined as the probability that the system is in a state where it can be guaranteed that conditions D.3 and D.4 can be satisfied. As noted in Section 3, conditions D.3 and D.4 are strictly weaker than D.1 and D.2. This implies that *reliability* $\leq$ *safety.*

In Section 2 we stated that a symmetric fault may need to be classified as arbitrary to accurately evaluate reliability and safety. We now illustrate this with an example. Consider a system with $N = 8$ nodes such that $m = 1$ and $u = 4$. Let the faults be such that $a = 0$, $s = 2$ and $c = 2$. In this case, it may seem that the system is *not* conforming, as it does not satisfy either of the two conditions in Theorem 1. This may imply that degradable agreement cannot be achieved using our algorithm. However, now reclassify the two symmetric-faulty nodes as arbitrary-faulty, resulting in $a = 2$, $s = 0$ and $c = 2$. Now, the system satisfies condition 2 in Theorem 1. This implies that the system can achieve the degraded agreement with the present fault scenario.

From the above discussion, and Theorem 1, it follows that our degradable agreement algorithm satisfies:

- D.1 and D.2 if $a \leq m$ and $N > 2(a+s)+c+u$,
- D.3 and D.4 if $a \leq u$, $(a+s) \leq u$ and $N > (a+s)+2m+c$ (note: here all s symmetric-faults are reclassified as arbitrary-faults), and
- D.3 and D.4 if $a \leq u$, $(a+s) > u$ and $N > u+2m+2(a+s-u)+c$ (note: here $(u-a)$ symmetric-faults are reclassified as arbitrary-faults).

To evaluate reliability and safety, we use the following model. Lifetime of each node is governed by an exponential distribution with failure rate λ. Given that a fault has occurred, μ_a, μ_s and μ_c denote the conditional probability that the fault is an arbitrary-fault, a symmetric-fault and a manifest-fault, respectively.

$$\begin{aligned} \mathcal{R} &= \text{set of states in which conditions D.1 and D.2 can be satisfied} \\ \mathcal{S} &= \text{set of states in which conditions D.3 and D.4 can be satisfied} \end{aligned}$$

Then, from the discussion above, we have,

$$\begin{aligned} \mathcal{R} &= \{(a,s,c) \mid a \leq m,\ N > 2(a+s)+c+u\} \\ \mathcal{S} &= \{(a,s,c) \mid a \leq m,\ N > 2(a+s)+c+u\} \\ &\quad \cup \{(a,s,c) \mid a \leq u,\ (a+s) \leq u,\ N > (a+s)+2m+c\} \\ &\quad \cup \{(a,s,c) \mid a \leq u,\ (a+s) > u,\ N > u+2m+2(a+s-u)+c\} \\ &= \{(a,s,c) \mid a \leq m,\ N > 2(a+s)+c+u\} \\ &\quad \cup \{(a,s,c) \mid a \leq u,\ (a+s) \leq u,\ N > (a+s)+2m+c\} \\ &\quad \cup \{(a,s,c) \mid a \leq u,\ (a+s) > u,\ N > 2(a+s)+(2m-u)+c\} \end{aligned}$$

Now, the probability that a node has *not* failed till time t is $e^{-\lambda t}$. Therefore, it follows that the probability of being in state (a,s,c) at time t is given by

$$P(a,s,c) = \binom{N}{a}\binom{N-a}{s}\binom{N-a-s}{c} [\mu_a(1-e^{-\lambda t})]^a\ [\mu_s(1-e^{-\lambda t})]^s\ [\mu_c(1-e^{-\lambda t})]^c\ [e^{-\lambda t}]^{N-a-s-c}$$

$$= \binom{N}{a}\binom{N-a}{s}\binom{N-a-s}{c}(\mu_a)^a(\mu_s)^s(\mu_c)^c(1-e^{-\lambda t})^{a+s+c}e^{-\lambda t(N-a-s-c)}$$

Using $P(a,s,c)$, the following expressions for reliability and safety can be obtained.

$$\begin{aligned} \text{reliability} &= \sum_{(a,s,c)\in\mathcal{R}} P(a,s,c) \\ \text{safety} &= \sum_{(a,s,c)\in\mathcal{S}} P(a,s,c) \end{aligned}$$

It is a simple exercise to verify that, when $m = u$, $\mathcal{R} = \mathcal{S}$, and therefore, reliability is equal to safety.

Using the above expressions for reliability and safety, we present numerical results for some example parameters; similar results can be obtained for other parameters, as well. Two different types of results are obtained depending on the value of various parameters, as described below.

5.1 Reliability-safety trade-off using degradable agreement

It is clear from the definition of reliability and safety that both cannot, in general, be maximized simultaneously. Using degradable agreement, it is possible to trade reliability with safety. That is, it is possible to choose different values of m and u such that safety increases and reliability decreases, with increasing value of u, reliability being maximized when $m = u$. This is illustrated by the following numerical results, assuming $t = 10$.

$\lambda = 0.001$ $\mu_a = .2$ $\mu_s = .3$ $\mu_c = 0.5$

n	m	u	1−reliability	1−safety
6	1	1	6.677003e−05	6.677003e−05
	1	2	3.735889e−04	2.534725e−06
	1	3	1.089407e−03	1.447012e−07

$\lambda = 0.001$ $\mu_a = .15$ $\mu_s = .25$ $\mu_c = 0.6$

n	m	u	1−reliability	1−safety
6	1	1	3.896059e−05	3.896059e−05
	1	2	2.434319e−04	1.368393e−06
	1	3	9.324527e−04	1.447012e−07

$\lambda = 0.001 \quad \mu_a = .1 \quad \mu_s = .1 \quad \mu_c = 0.8$

n	m	u	1−reliability	1−safety
6	1	1	1.634273e−05	1.634273e−05
	1	2	6.654959e−05	2.976627e−07
	1	3	5.329331e−04	1.447012e−07

$\lambda = 0.001 \quad \mu_a = .01 \quad \mu_s = .05 \quad \mu_c = 0.94$

n	m	u	1−reliability	1−safety
6	1	1	3.731027e−07	3.731027e−07
	1	2	8.520649e−06	1.488311e−07
	1	3	1.853509e−04	1.447012e−07

$\lambda = 0.001 \quad \mu_a = .01 \quad \mu_s = .19 \quad \mu_c = 0.8$

n	m	u	1−reliability	1−safety
6	1	1	2.216854e−06	2.216854e−06
	1	2	6.654959e−05	2.976627e−07
	1	3	5.329331e−04	1.447012e−07

$\lambda = 0.001 \quad \mu_a = .01 \quad \mu_s = .01 \quad \mu_c = 0.98$

n	m	u	1−reliability	1−safety
6	1	1	1.770926e−07	1.770926e−07
	1	2	1.839864e−06	1.448541e−07
	1	3	7.576839e−05	1.447012e−07

The extent of increase or decrease in reliability and safety is measured in terms of percentage change in (1−reliability) and (1−safety), respectively. Observe that most significant gain in safety is obtained when u is increased from 1 to 2, further increasing u does not seem to result in significant trade-off (for the example parameters). For larger values of μ_a, relatively larger gain in safety is achieved with a small or comparable reduction in reliability. In general, the trade-off is more favorable for larger values of μ_a.

Reliability-safety trade-off is fundamental to many areas of fault tolerance. For results on reliability-safety trade-off in system diagnosis and modular redundant systems, refer to [9, 13] and [12], respectively.

5.2 Small values of μ_a

In the examples presented above, increasing value of u resulted in increasing safety and decreasing reliability. Thus, reliability was traded for safety. This trend holds only of the value of μ_a is not too small. Below we present numerical results, for $\mu = 0.001$, which indicate that degradable agreement results in poorer reliability *and* safety when μ_a (or probability of arbitrary failure) is too small. The next subsection presents numerical results to show that similar reliability degradation occurs for the algorithm presented in [4].

$\lambda = 0.001 \quad \mu_a = .001 \quad \mu_s = .019 \quad \mu_c = 0.98 \quad t = 10$

n	m	u	1−reliability	1−safety
6	1	1	3.583387e−08	3.583387e−08
	1	2	1.839864e−06	1.448541e−07

Note that when u is increased, both (1−reliability) and (1−safety) increase, implying that both reliability and safety decrease. In the above example, however, both μ_a and μ_s were chosen small. However, if we choose a larger μ_s keeping $\mu_a = 0.001$, increasing u results in a reliability-safety trade-off, as apparent from the table below. However, due to the small value of μ_a, relatively small gain in safety is achieved with a larger reduction in reliability.

$\lambda = 0.001 \quad \mu_a = .001 \quad \mu_s = .1 \quad \mu_c = 0.899 \quad t = 10$

n	m	u	1−reliability	1−safety
6	1	1	5.977259e−07	5.977259e−07
	1	2	1.992804e−05	1.644007e−07
	1	3	2.929344e−04	1.447012e−07

5.3 Reliability achieved by Byzantine agreement

We showed above that degradable agreement algorithm for hybrid fault model is useful primarily when μ_a is not negligible. We claim that the hybrid fault model itself is useful only when μ_a is not very small. To justify this claim we compare reliability of the Byzantine agreement algorithm OMH for hybrid faults presented in [4, 7] with the reliability of the agreement algorithm (named X) that tolerates only symmetric and manifest faults.

Algorithm X is simple and can be stated as follows: (step 1) The sender sends its value to all the $(N-1)$ receivers. (step 2) Receiver j, on receiving value v_j from the sender, agrees on v_j, or if no value or a manifestly bad value is received, then agrees on E.

It is clear that algorithm X achieves agreement among all fault-free nodes provided, $s + c < N$ and $a = 0$. Using this observation, we can find the reliability of algorithm X. Also, note that reliability of algorithm OMH (when $\mu_s, \mu_c > 0$ and $\mu_a = 0$) is the same as that of algorithm HBYZ when $m = u$.

$\lambda = 0.001 \quad t = 10$

n	m	u	μ_a	μ_s	μ_c	(1−reliability) for OMH	(1−reliability) for X
5	1	1	.00001	.01999	.98	1.000800e−06	4.976057e−07
6	1	1	5x10^{-7}	.0199995	.98	3.440701e−08	2.985147e−08

Note that in both examples above, algorithm X achieves higher reliability than algorithm OMH. This implies that, if reliability is the only parameter of interest, with low values of μ_a, it is beneficial to ignore the arbitrary-faults altogether. In other words, the hybrid fault model in [4, 7] is not necessarily appropriate. While this observation applies to both algorithm OMH as well as HBYZ, there is a quantitative difference: there exist (small) values of μ_a for which algorithm OMH achieves higher reliability than algorithm X, but algorithm HBYZ may not yield a good trade-off. In other words, algorithm HBYZ is effective for larger values of μ_a as compared to algorithm OMH. This is to be expected, however, as degradable agreement is designed for a larger number of arbitrary-faults.

6 Conclusions

Achieving traditional Byzantine agreement in the presence of arbitrary faults requires that the total number of nodes be larger than three times the number of faulty nodes. Two approaches have been proposed in the literature to circumvent this requirement: (i) hybrid fault model approach considers three types of faults of varying degree of severity, and (ii) degradable agreement approach relaxes the definition of agreement in the presence of excessive faults. This report combines the above two approaches and presents a degradable agreement algorithm for the hybrid fault model.

The report also defines and evaluates reliability and safety measures for the proposed degradable agreement algorithm and concludes that degradable agreement can be used to trade reliability with safety, particularly when probability of an arbitrary failure is *not* too small. Numerical results are presented for various parameter values to support this conclusion.

It is shown that, it does not pay to design algorithms for that tolerate arbitrary-faults, unless the probability of such faults is non-negligible. In other words, the hybrid fault model [4, 7] is effective only when arbitrary-faults are sufficiently likely to occur. Specifically, we show that (i) the reliability-safety trade-off achieved by HBYZ is effective only for non-negligible values of μ_a, and (ii) reliability achieved by the Byzantine agreement algorithms previously proposed for the hybrid fault model [4, 7] is worse than an algorithm that ignores arbitrary-faults, when the probability of arbitrary-faults is small.

A Proof of Correctness: Algorithm HBYZ

The proof presented here parallels the proof presented in [11], however, proofs of some of the lemmas below are more complicated than those in [11].

The correctness of algorithm HBYZ is being proved under assumptions (a) through (c) listed earlier in Section 4. When a message is detected to be absent by a node, that node considers the absent message to contain value E. Therefore, the following assumes that each node always sends a message when it is supposed to; however, a faulty node may send an incorrect message (possibly with value E).

Lemma 1 *When HBYZ(t) is called with $m \geq t \geq 1$, the following condition holds: $n_t = N - m + t$.*

Proof: Follows from the observations that $n_m = N$, and $n_{t-1} = n_t - 1$ for $m \geq t > 1$. □

Lemma 2 *For $m \geq t \geq 1$, at least one of the following conditions hold for a conforming system:*

1. *$a \leq m$ and $n_t > 2(a+s) + c + t + (u - m)$.*
2. *$a \leq u$ and $n_t > a + m + 2s + c + t$.*

Proof: Follows from Definition 1 and Lemma 1. □

Lemma 3 *For a conforming system, $(n_t - 1 - c + t + u - m)/2 > a + s$, provided $m \geq t \geq 1$.*

Proof: At least one of the conditions in Lemma 2 is true for a conforming system.

If condition 1 in Lemma 2 is true, then we have $n_t - 1 - c + t + u - m \geq 2a + 2s + 2t + 2(u - m)$. As $t \geq 1$ and $u \geq m$, this implies that $(n_t - 1 - c + t + u - m)/2 > a + s$.

If condition 2 in Lemma 2 is true, then we have $n_t - 1 - c + t + u - m \geq a + u + 2s + 2t$. As $t \geq 1$ and $u \geq a$, this implies that $(n_t - 1 - c + t + u - m)/2 > a + s$. □

Lemma 4 *Consider a conforming system. It is given that, in step 2 of HBYZ(t) ($t \geq 1$), E may be obtained only from faulty receivers and the value obtained from each manifest-faulty receiver is either E or V_d. If $(t + u - m)$-HVOTE of $n_t - 1$ quantities obtained by a receiver, say A, in step 3 of HBYZ(t) is $\gamma \neq V_d$, then receiver A must have obtained γ from at least one fault-free receiver.*

Proof: Consider receiver A. Let the total number of receivers from which E was obtained by receiver A (in step 2) be $c^{\#}$. Let the number of manifest-faulty receivers from which E is obtained be c^*. It follows that, $c^* \leq c^{\#}$ and $c^* \leq c$.[4] By Definition 2, receiver A must have obtained γ from k nodes such that, $k \geq (n_t - 1) - k - c^{\#} + (t + u - m)$. This implies that $k + (c^{\#} - c^*)/2 \geq (n_t - 1 - c^* + t + u - m)/2$. As $c^* \leq c$, this implies that $k + (c^{\#} - c^*)/2 \geq (n_t - 1 - c + t + u - m)/2$. Now, by Lemma 3, it follows that $k + (c^{\#} - c^*)/2 > a + s$, or $k > a + s - (c^{\#} - c^*)/2$. Now, as E is obtained only from faulty receivers, the number of symmetric-faulty and arbitrary-faulty receivers from which node A did not obtain value E is $f \leq a + s + c^* - c^{\#} = a + s - (c^{\#} - c^*)$. As $k > a + s - (c^{\#} - c^*)/2$ and $c^{\#} \geq c^*$, we have $f < k$. Therefore, receiver A must have obtained γ from at least one fault-free or manifest-faulty node. But only E or V_d could have been obtained from a manifest-faulty receiver. Therefore, the above implies that node A must have obtained γ from at least one fault-free receiver. □

Lemma 5 *Given a conforming system, following conditions hold true for HBYZ(1).*

[4] Recall that c is the total number of manifest-faulty nodes among the N nodes in the system.

1. *HBYZ(1) satisfies condition D.1 if* $a \leq m$ *and* $n_1 > 2(a + s) + c + 1 + (u - m)$.

2. *HBYZ(1) satisfies condition D.2 if the number of arbitrary-faulty nodes (among the* n_1 *nodes) is 1.*

3. *HBYZ(1) satisfies condition D.3 if* $a \leq u$ *and* $n_1 > a + m + 2s + c + 1$.

Proof:

Case 1: $a \leq m$, $n_1 > 2(a + s) + c + 1 + (u - m)$ and the sender in HBYZ(1) is not arbitrary-faulty.

Assume that a fault-free receiver receives value α from the sender in step 1 of HBYZ(1). If the sender is fault-free, then α is the sender's value; if the sender is symmetric-faulty, then α is the value it sent to all the receivers; if the sender is manifest faulty, then $\alpha = E$. In step 2, each fault-free receiver broadcasts $R(\alpha)$ to all the $(n_1 - 1)$ receivers including itself (note: $R(\alpha) \neq$ E, $\forall\alpha$). When the broadcasts in step 2 of HBYZ(1) are complete, each receiver will have $(n_1 - 1)$ values. If c^* receivers are manifest-faulty ($c^* \leq c$), then at least $n_1 - 1 - (a + s + c^*)$ values must be $R(\alpha)$, at least c^* values must be E, and at most $n_1 - 1 - c^*$ values may be non-E. Now, let $k^* = n_1 - 1 - (a + s + c^*)$. As $n_1 > 2(a+s)+c+1+(u-m)$ and $c^* \leq c$, we have $k^* \geq (n_1-1)-k^*-c^*+1+u-m$. Consider any receiver, say A. Assume that receiver A obtained $R(\alpha)$ from k nodes. Then, $k \geq k^*$. As $k^* \geq (n_1-1)-k^*-c^*+1+u-m$ and $k \geq k^*$, we have, $k \geq (n_1-1)-k-c^*+1+u-m$. Now consider Definition 2. It is clear that $c^{\#}$ in the definition is such that $c^{\#} \geq c^*$. Therefore, by Definition 2 and the inequality for k it follows that $(1 + u - m)$-HVOTE of the $n_1 - 1$ quantities must be $R(\alpha)$. Therefore, each fault-free receiver must obtain α in step 3 of HBYZ(1). Thus, item 1 of the lemma is proved.

Case 2: The number of arbitrary-faulty nodes is one and the sender is arbitrary-faulty.

As there is only one arbitrary-faulty node among the n_1 nodes and the sender is arbitrary-faulty, none of the $(n_1 - 1)$ receivers is arbitrary-faulty. Therefore, in step 2 of

HBYZ(1), each fault-free receiver must obtain the same set of $(n_1 - 1)$ values. This implies that in step 3, each fault-free receiver will obtain the same value using $(1+u-m)$-HVOTE. Thus, item 2 in the lemma is proved.

Case 3: $a \leq u$, $n_1 > a+m+2s+c+1$ and the sender (in HBYZ(1)) is not arbitrary-faulty.

The sender sends its value, say α, to the receivers in step 1 of HBYZ(1). All receivers receive the same value, as the sender is not arbitrary-faulty. Each fault-free receiver broadcasts $R(\alpha)$ to all $n_1 - 1$ receivers, including itself. When the broadcasts in step 2 of HBYZ(1) are complete, each fault-free receiver will have $n_1 - 1$ values of which values received from all non-faulty receivers will all be $R(\alpha)$ (note: $R(\alpha) \neq E$, $\forall \alpha$). Additionally, values received from all manifest-faulty receivers must be E. If a receiver, say A, obtains $(1+u-m)$-HVOTE in step 3 equal to $\beta \neq V_d$, then by Lemma 4, β must be equal to $R(\alpha)$. In other words, each fault-free receiver must obtain $(1+u-m)$-HVOTE equal to either $R(\alpha)$ or V_d. Therefore, each fault-free receiver will agree on α or V_d (note: $UnR(V_d) = V_d$). Thus, item 3 in the lemma is proved. □

Lemmas 6 through 9 below together prove that HBYZ(m) achieves desired degradable agreement. It is implicitly assumed that $1 \leq t \leq m$.

Lemma 6 *Given a conforming system, algorithm HBYZ(t) satisfies condition D.1 provided* $a \leq m$ *and* $n_t > 2(a+s)+c+t+(u-m)$.

Proof: Condition D.1 assumes that the sender is not arbitrary-faulty. The proof is by induction on t. The lemma is true for $t = 1$ by Lemma 5. We now assume that the lemma is true for HBYZ($t-1$) where $2 \leq t \leq m$, and prove it for HBYZ(t). In step 1 of HBYZ(t), the sender sends a value, say α, to all the $(n_t - 1)$ receivers. All receivers receive the same value, as the sender is not arbitrary-faulty. In step 2, each fault-free receiver acts as a sender in HBYZ($t-1$) to send value $R(\alpha)$ to all the $(n_t - 1)$ receivers, including itself. As $n_t > 2(a+s)+c+t+(u-m)$, $n_t - 1 > 2(a+s)+c+(t-1)+(u-m)$. Also, $n_{t-1} = n_t - 1$. Therefore, the induction hypothesis holds for HBYZ($t-1$). Therefore, at the end of step

2, every fault-free receiver gets $w_j = R(\alpha)$ for each fault-free receiver j. If c^* receivers are manifest-faulty ($c^* \leq c$), then at least $n_t - 1 - (a+s+c^*)$ values must be $R(\alpha)$. Also, as c^* receivers are manifest-faulty, by the induction hypothesis, at least c^* values must be equal to E, and at most $n_t - 1 - c^*$ values may be non-E. Now, let $k^* = n_t - 1 - (a+s+c^*)$. As $n_t > 2(a+s)+c+t+(u-m)$ and $c^* \leq c$, we have $k^* \geq (n_t - 1) - k^* - c^* + (t+u-m)$. Consider any receiver, say A. Assume that receiver A obtained $R(\alpha)$ from k nodes. Then, $k \geq k^*$. As $k^* \geq (n_t-1) - k^* - c^* + 1 + u - m$ and $k \geq k^*$, we have, $k \geq (n_t - 1) - k - c^* + 1 + u - m$. Now consider Definition 2. It is clear that $c^{\#}$ in the definition is such that $c^{\#} \geq c^*$. Therefore, by Definition 2 and the inequality for k it follows that $(t+u-m)$-HVOTE of the $(n_t - 1)$ quantities must be $R(\alpha)$. Therefore, each fault-free receiver must obtain α in step 3 of HBYZ(t). Thus, the lemma is proved. □

Lemma 7 *Given a conforming system, algorithm HBYZ(t) satisfies condition D.2 provided at most t nodes (among the n_t nodes) are arbitrary-faulty and $n_t > 2(a+s)+c+t+(u-m)$.*

Proof: As at most t nodes are arbitrary-faulty and $t \leq m$, we have $a \leq m$.

The proof is by induction on t. The lemma is true for $t = 1$ by Lemma 5. We now assume that the lemma is true for HBYZ$(t-1)$ where $2 \leq t \leq m$, and prove it for HBYZ(t).

The number of arbitrary-faulty nodes is at most t. Condition D.2 assumes that the sender is arbitrary-faulty. Therefore, at most $(t-1)$ of the $(n_t - 1)$ receivers are arbitrary-faulty. In step 2, a receiver uses HBYZ$(t-1)$ to send a value to all the $n_t - 1$ receivers, including itself. As at most $(t-1)$ of the $(n_t - 1)$ receivers are arbitrary-faulty, and $n_{t-1} = (n_t - 1) > 2(a+s)+c+(t-1)+(u-m)$, we can apply the induction hypothesis to conclude that HBYZ$(t-1)$ satisfies condition D.2. As observed earlier, $a \leq m$. Therefore, by Lemma 6, HBYZ$(t-1)$ satisfies condition D.1 as well. Hence, at the end of step 2 of HBYZ(t), any two fault-free receivers must obtain the same vector $w_1, w_2, \cdots, w_{n_t - 1}$. Therefore, all fault-free receivers must obtain the same value using $(t+u-m)$-HVOTE in step 3 of HBYZ(t). This, in turn, implies that all fault-free receivers agree on the same value. Thus, the lemma is proved for HBYZ(t). □

Lemma 8 *Given a conforming system, algorithm HBYZ(t) satisfies condition D.3 provided* $a \leq u$ *and* $n_t > a + m + 2s + c + t$.

Proof: The proof is by induction on t. Condition D.3 assumes that the sender is not arbitrary-faulty. The lemma is true for $t = 1$ by Lemma 5. We now assume that the lemma is true for HBYZ($t-1$) where $2 \leq t \leq m$, and prove it for HBYZ(t).

In step 1 of HBYZ(t), the sender sends a value, say α, to all the $(n_t - 1)$ receivers. All receivers receive the same value, as the sender is not arbitrary-faulty. In step 2, each fault-free receiver applies HBYZ($t-1$) to send value $R(\alpha)$ to all the $(n_t - 1)$ receivers, including itself (note: $R(\alpha) \neq E$, $\forall \alpha$). Also, each manifest-faulty receiver applies[5] HBYZ($t-1$) to send value E to all the $(n_t - 1)$ receivers. We can apply the induction hypothesis to conclude that every fault-free receiver gets $w_j = R(\alpha)$ or V_d from each fault-free receiver j (using HBYZ($t-1$)), and E or V_d from each manifest-faulty receiver. If a receiver, say A, obtains $(t + u - m)$-HVOTE in step 3 equal to $\beta \neq V_d$, then by Lemma 4, β must be equal to $R(\alpha)$. In other words, each fault-free receiver must obtain $(t + u - m)$-HVOTE equal to either $R(\alpha)$ or V_d. Therefore, each fault-free receiver will agree on α or V_d (note: $UnR(V_d) = V_d$). Thus, the lemma is proved. □

Lemma 9 *Given a conforming system, algorithm HBYZ(m) satisfies condition D.4 provided* $a \leq u$ *and* $n_m > a + 2m + 2s + c$.

Proof: In this proof, we assume that $m > 1$. The proof for $m = 1$ is analogous.

Condition D.4 assumes that the sender is faulty. Therefore, at most $(u - 1)$ of the receivers are arbitrary-faulty. As at most $(u - 1)$ receivers are arbitrary-faulty, at least $(n_m - u)$ are not arbitrary-faulty.

In step 2 of HBYZ(m), each receiver sends a value to all the $(n_m - 1)$ receivers using HBYZ($m-1$). Thus, at the end of step 2 each receiver obtains $(n_m - 1)$ values.

[5] Recollect that, by definition of a manifest-fault, any message transmitted by a manifest-faulty receiver, effectively, contains value E.

As $n_{m-1} = n_m - 1 > a + m + 2s + c + (m-1)$ and $a \leq u$, by Lemma 8, we know that HBYZ($m-1$) satisfies condition D.3. Therefore, the values obtained from manifest-faulty receivers must be E or V_d. Also, as a receiver j sends $R(v_j)$ in step 2 using HBYZ($m-1$) and $R(v_j) \neq E, \forall v_j$, by Lemma 8, the value obtained from a fault-free receiver cannot be E.

Assume that, in step 3, receivers A and B obtain ($m+u-m$)-HVOTE (i.e. u-HVOTE) equal to β and γ respectively, where $\beta \neq \gamma$, $V_d \neq \beta$ and $V_d \neq \gamma$. (Also, by definition of HVOTE, $\beta \neq E$ and $\gamma \neq E$.)

Define the following:

Z_a (Z_b) = set of receivers from which A (B) obtained value β (γ) using HBYZ($m-1$).
S_a (S_b) = set of symmetric-faulty receivers from which A (B) obtained E but B (A) did not.
A_a (A_b) = set of arbitrary-faulty receivers from which A (B) obtained E but B (A) did not.
C_a (C_b) = set of manifest-faulty receivers from which A (B) obtained E but B (A) obtained V_d.
S_{ab} = set of symmetric-faulty receivers from which both A and B obtained value E.
A_{ab} = set of arbitrary-faulty receivers from which both A and B obtained value E.
C_{ab} = set of manifest-faulty receivers from which both A and B obtained value E.
$Z_a^* = Z_a \cup S_a \cup A_a \cup A_{ab}$
$Z_b^* = Z_b \cup S_b \cup A_b \cup A_{ab}$
e_a (e_b) = number of E values (among the $n_m - 1$ values) obtained by A (B)

This implies that,[6] by Definition 2, $|Z_a| \geq (n_m - 1) - |Z_a| - e_a + u$. This, in turn, implies that $|Z_a^*| \geq (n_m - 1) - |Z_a| + |S_a| + |A_a| + |A_{ab}| - e_a + u$. Now, as E is not obtained from any fault-free receiver, $e_a = |S_a| + |A_a| + |C_a| + |S_{ab}| + |A_{ab}| + |C_{ab}|$. Therefore, $|Z_a^*| \geq (n_m - 1) - |Z_a| - |S_{ab}| - |C_a| - |C_{ab}| + u$. As $|Z_a^*| \geq |Z_a|$, we have, $|Z_a^*| \geq (n_m - 1 - |S_{ab}| - |C_a| - |C_{ab}| + u)/2$. By following similar step, we also have, $|Z_b^*| \geq (n_m - 1 - |S_{ab}| - |C_b| - |C_{ab}| + u)/2$. Adding the inequalities for $|Z_a^*|$ and $|Z_b^*|$, we get $|Z_a^*| + |Z_b^*| \geq n_m - 1 - |S_{ab}| - (|C_a| + 2|C_{ab}| + |C_b|)/2 + u$. Now, $|C_a| + |C_{ab}| \leq c$ and $|C_b| + |C_{ab}| \leq c$. Therefore, $|Z_a^*| + |Z_b^*| \geq n_m - 1 - |S_{ab}| - c + u$. As $a \leq u$, we have

$$|Z_a^*| + |Z_b^*| \geq n_m - 1 - |S_{ab}| - c + a.$$

As the sender in HBYZ(m) is arbitrary-faulty and total number of arbitrary-faults is a, $(a-1)$ receivers are arbitrary-faulty. Also, as c nodes are manifest-faulty, $(n_m - 1 - c)$

[6] Recall that receiver A obtained u-HVOTE equal to β, in step 3 of HBYZ(m).

receivers are not manifest-faulty, and $(n_m - c - a)$ receivers are either fault-free or symmetric-faulty. Also, observe the following:

1. As noted earlier, in step 2, either E or V_d is obtained from each manifest-faulty receiver. Therefore, Z_a and Z_b cannot contain any manifest-faulty receivers. Therefore, Z_a^* and Z_b^* also do not contain any manifest-faulty receivers.

2. $Z_a \cap Z_b$ cannot contain a fault-free receiver, because (i) Lemma 8 holds for BYZ$(m-1)$, and (ii) we assumed that $\beta \neq \gamma$, $V_d \neq \beta$ and $V_d \neq \gamma$. This, in turn, implies that $Z_a^* \cap Z_b^*$ cannot contain a fault-free receiver.

3. By an argument similar to (2) above, $Z_a \cap Z_b$ cannot contain a symmetric-faulty receiver. Therefore, as $S_a \cap S_b = \emptyset$, $Z_a^* \cap Z_b^*$ cannot contain a symmetric-faulty receiver.

4. $(Z_a^* \cup Z_b^*) \cap S_{ab} = \emptyset$, i.e., no receivers in S_{ab} belong to $Z_a^* \cup Z_b^*$.

5. As $A_{ab} \subseteq Z_a^* \cap Z_b^*$, an arbitrary-faulty receiver may belong to both Z_a^* and Z_b^*. (Recollect that $a - 1$ receivers are arbitrary-faulty.)

Observations 2 and 3 imply that sets Z_a^* and Z_b^* do not have any fault-free or symmetric-faulty nodes in common. Thus, by the above observations, we have $|Z_a^*| + |Z_b^*| \leq 2(a-1) + (n_m - c - a - |S_{ab}|) = n_m - c - |S_{ab}| + a - 2$, i.e.,

$$|Z_a^*| + |Z_b^*| < n_m - 1 - |S_{ab}| - c + a$$

But this inequality contradicts one derived earlier. Therefore, our assumption that $\beta \neq \gamma$, $V_d \neq \beta$ and $V_d \neq \gamma$ must be incorrect. In other words, at least one of the following must be true: $\beta = \gamma$, or $V_d = \beta$ or $V_d = \gamma$. Thus, the lemma is proved. □

Theorem 1 *Given $u \geq m$ and a system with N nodes the following conditions hold for HBYZ(m).*

1. *if* $N > 2(a+s)+c+u$ *and* $a \leq m$, *then D.1 and D.2 are satisfied.*

2. *if* $N > a+2m+2s+c$ *and* $a \leq u$, *then D.3 and D.4 are satisfied.*

Proof: The proof follows from Lemmas 6 through 9 by choosing $t = m$. □

References

[1] D. Dolev, "The Byzantine generals strike again," *J. Algo.*, pp. 14–30, 1982.

[2] L. Lamport, "The weak Byzantine generals problem," *J. ACM*, vol. 30, pp. 668–676, July 1983.

[3] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Trans. Prog. Lang. Syst.*, vol. 4, pp. 382–401, July 1982.

[4] P. Lincoln and J. Rushby, "A formally verified algorithm for interactive consistency under a hybrid fault model," in *Digest of papers: The* 23^{rd} *Int. Symp. Fault-Tolerant Comp.*, pp. 402–411, 1993.

[5] F. J. Meyer and D. K. Pradhan, "Consensus with dual failure modes," *IEEE Trans. Par. Distr. Syst.*, vol. 2, pp. 214–222, April 1991.

[6] K. Rothermel, "An open commit protocol preserving consistency in the presence of commission failures," in *International Conf. Distributed Computing Systems*, pp. 168–177, May 1993.

[7] P. Thambidurai and Y.-K. Park, "Interactive consistency with multiple failure modes," in *7th Symposium on Reliable Distributed Systems*, pp. 93–100, October 1988.

[8] P. Thambidurai, Y.-K. Park, and K. S. Trivedi, "On reliability modelling of fault-tolerant distributed systems," in *International Conf. Distributed Computing Systems*, pp. 136–142, 1989.

[9] N. H. Vaidya and D. K. Pradhan, "System level diagnosis: Combining detection and location," in *Digest of papers: The* 21^{st} *Int. Symp. Fault-Tolerant Comp.*, pp. 488–495, 1991.

[10] N. H. Vaidya, "Degradable agreement in the presence of Byzantine faults," Tech. Rep. 92-020, Computer Science Department, Texas A&M University, College Station, 1992.

[11] N. H. Vaidya and D. K. Pradhan, "Degradable agreement in the presence of Byzantine faults," in *International Conf. Distributed Computing Systems*, May 1993.

[12] N. H. Vaidya and D. K. Pradhan, "Fault-tolerant design strategies for high reliability and safety," *IEEE Trans. Computers*, accepted for publication, 1993.

[13] N. H. Vaidya and D. K. Pradhan, "Safe system level diagnosis," *IEEE Trans. Computers*, accepted for publication, 1993.