Proc. 22nd Intl. Conf. on Distributed Computer Systems (ICDCS'02), p. 608–616, July 2002, Vienna, Austria July 4, 2002

Formally Verified Byzantine Agreement in Presence of Link Faults

ULRICH SCHMID, BETTINA WEISS*
Technische Universität Wien
Department of Automation
Treitlstraße 1, A-1040 Vienna
{*s, bw*}*@auto.tuwien.ac.at*

JOHN RUSHBY†
SRI International
Computer Science Laboratory
Menlo Park CA 94025
Rushby@csl.sri.com

Abstract

This paper shows that deterministic consensus in synchronous distributed systems with link faults is possible, despite the impossibility result of (Gray, 1978). Instead of using randomization, we overcome this impossibility by moderately restricting the inconsistency that link faults may cause system-wide. Relying upon a novel hybrid fault model that provides different classes of faults for both nodes and links, we provide a formally verified proof that the $m+1$-round Byzantine agreement algorithm OMH (Lincoln & Rushby, 1993) requires $n > 2f_\ell^s + f_\ell^r + f_\ell^{ra} + 2(f_a + f_s) + f_o + f_m + m$ nodes for transparently masking at most f_ℓ^s broadcast and f_ℓ^r receive link faults (including at most f_ℓ^{ra} arbitrary ones) per node in each round, in addition to at most f_a, f_s, f_o, f_m arbitrary, symmetric, omission, and manifest node faults, provided that $m \geq f_a + f_o + 1$. Our approach to modeling link faults is justified by a number of theoretical results, which include tight lower bounds for the required number of nodes and an analysis of the assumption coverage in systems where links fail independently with some probability p.

Keywords: *Fault-tolerant distributed systems, fault models, link faults, consensus, Byzantine agreement, formal verification, assumption coverage, impossibility results, lower bounds.*

1 Motivation

Although process/node fault models, like the one that at most f of the n nodes of a distributed system may be faulty during a particular execution, have always been applied most successfully in the analysis of fault-tolerant distributed algorithms, they do have limitations. In fact, given the steadily increasing dominance of communication over computation in modern distributed systems, it becomes increasingly difficult to apply node-centric fault models that provide no explicit treatment of *link faults*.

For more than 20 years, however, it has been considered impossible to deterministically solve the important problem of *consensus* (*Byzantine agreement*) in systems with high link failure rates: A general impossibility result, going back to Gray's 1978 paper [8] on atomic commitment in distributed databases, says that there is no deterministic algorithm solving the coordinated attack problem in a two-node system with lossy links [13, Thm. 5.1]. We recently discovered, however, that this result does not really prohibit consensus in such systems: If the power of link faults is moderately restricted with respect to the inconsistency they might cause system-wide, most existing consensus algorithms can be made resilient to a large number of link faults if the number of nodes n is increased appropriately.

The remaining sections of our paper[1] are organized as follows: In Section 2, we introduce the perception-based hybrid fault model that will be used in the analysis of OMH in Section 3. Section 4 is devoted to the formal verification of our proofs, Section 5 contains some of our theoretical results. In Section 6, we investigate our model's assumption coverage in typical system architectures. A discussion of our results, including a relation to existing work, in Section 7 and some conclusions and directions of further research in Section 8 eventually round off the paper.

2 Perception-based Fault Model

The key to our results is a novel *perception-based hybrid fault model* for synchronous systems, which is a generalization of the one introduced for the analysis of clock synchronization algorithms in [22]. In this model, the global, i.e., system-wide, number of faults is replaced by the number of faults that are observable in the nodes' local "percep-

*This research is part of our W2F-project, which targets a wireline/wireless fieldbus based upon spread-spectrum (CDMA) communications, see *http://www.auto.tuwien.ac.at/Projects/W2F/* for details. W2F is supported by the Austrian START programme Y41-MAT.

†This research was partially supported by by NASA Langley Research Center under contract NAS1-20334 and Cooperative Agreement NCC-1-377 with Honeywell Incorporated.

[1]Due to space limitations, we can only give a very brief overview—usually without proofs—of our many results in this paper. Consult the comprehensive report [24] and its formal verification [19] for all the details.

tions" of the system. Formally, node r's *perception vector* $\mathcal{V}_r = (V_r^1, V_r^2, \ldots, V_r^n)$ is considered, where every *perception* $V_r^s \in \mathcal{V}_r$ represents the message node r received from node s in some specific round; type and value(s) depend upon the particular algorithm considered.

In case of the single-round algorithms analyzed in [22], we found it sufficient to impose a bound upon the maximum number of faults in any *pair* of perception vectors $\{\mathcal{V}_p, \mathcal{V}_q\}$, i.e., p's and q's lines in the "matrix" of perceptions on the right-hand side of

$$\begin{aligned} \mathcal{V}_1 &= (V_1^1, V_1^2, \ldots, V_1^n) \\ \mathcal{V}_2 &= (V_2^1, V_2^2, \ldots, V_2^n) \\ &\vdots \\ \mathcal{V}_n &= (V_n^1, V_n^2, \ldots, V_n^n). \end{aligned}$$

In fact, if we grant every node at most f_ℓ receive link faults, any two perception vectors can differ only in at most $2f_\ell$ perceptions, namely, the ones where either receiver node experienced its link faults. In addition, since f node faults can produce at most f faulty perceptions in any $\mathcal{V}_r$, our perception-based model is compatible with traditional node fault models as well. Hence, all existing lower bound and impossibility results remain valid.

In view of these observations, it was natural to consider the question of whether a perception-based fault model can also be used to attack deterministic consensus in presence of link faults. More specifically, as the general problem is unsolvable [8], one might ask whether there is a meaningful restriction of the power of link faults that can be expressed in a perception-based manner. And indeed, using the "matrix" of perceptions above, a suitable solution is based upon limiting both

1. the number of columns with wrong perceptions in any single line, which puts a limit upon the number of sender nodes that may appear faulty to a single receiver (already employed in [22]).
2. the number of lines with wrong perceptions in any single column, which puts a limit upon the number of receiving nodes that may obtain a wrong (or no) message in the broadcast of a single sender.

Depending upon the type of fault of a perception, e.g., missing or value faulty, different classes of faults (manifest/omission/symmetric/asymmetric) can be distinguished. This leads to a *hybrid fault model* [2, 12, 14, 18, 21, 23, 25–27], which allows us to exploit the fact that less severe faults can be handled with fewer nodes than more severe ones. For example, it will turn out that masking f symmetric faults requires only $n \geq 2f+1$ nodes, whereas $n \geq 3f+1$ is needed if all faults are asymmetric (Byzantine) ones. Since a large number of asymmetric faults is quite unlikely in practice, this effectively leads to a smaller n for tolerating a given number of faults. This, in turn, has a positive effect upon dependability by reducing the number n of components that could be faulty. System designers will hence appreciate our very detailed hybrid fault model for getting the maximum fault-tolerance out of a given—and usually quite small—n. Obviously, an algorithm's node requirements in standard models (like all-Byzantine) are easily obtained by setting some model parameters to 0.

In order to cleanly specify the semantics of restricted faults, we will need the notion of *obedient* nodes: An obedient node is an alive node that faithfully executes the particular algorithm. It must hence get its inputs and perform its computations like a non-faulty node, but it may fail in specific ways to communicate its value to the outside world. We will subsequently use this term instead of non-faulty whenever a node acts as a receiver.

Definition 1 (System Model) *We consider a distributed system of n nodes interconnected by a fully connected point-to-point network, which has the following properties:*

(P1) In any execution, there may be at most f_a, f_s, f_o, and f_m arbitrary, symmetric, omission, and manifest faulty nodes. The fault modes are defined via the set rvals(V_p, p) of admissible values delivered by obedient receivers when node p attempts to send some value V_p in a single broadcast:

- *A* manifest *faulty node p fails to send a message to any receiver or sends a value that all receivers q can detect as obviously bad. They all deliver the value $V_q^p = E$ in this case, i.e., rvals$(V_p, p) = \{E\}$.*
- *An* omission *faulty node p may fail to send the correct value V^p to some of its receivers q_i, which deliver $V_{q_i}^p = E$ instead of V^p in this case. Hence, rvals$(V_p, p) = \{V_p, E\}$.*
- *A* symmetric *faulty node p sends the same wrong (but not usually detectably bad) value X^p to every receiver q. They all deliver $V_q^p = X^p$—the value* "actually sent"*—in this case, such that rvals$(V_p, p) = \{X^p\}$.*
- *An* arbitrary (asymmetric) *faulty node may inconsistently send any value to any receiver, so rvals(V_p, p) is the set of all possible values, including E.*

A node that is manifest or omission faulty is called benign *faulty and is assumed to be obedient (if not crashed).*

(A1^s) If a single [faulty or non-faulty] node p broadcasts (= successively sends) a value V^p to some set of obedient receiver nodes $\mathcal{R}$, at most f_ℓ^s of the delivered values $V_{q_i}^p$ may differ from the admissible receive values in rvals(V^p, p). Let $f_\ell^{sa} \leq f_\ell^s$ be the maximum number of non-omissive, i.e., non-empty and hence value faulty, $V_{q_i}^p$ among those.

(A1^r) If all nodes $p_i \in \mathcal{S}$ of a set of [faulty or non-faulty] nodes send a message containing V^{p_i} to some obedient receiver node q, at most f_ℓ^r of the delivered values $V_q^{p_i}$ may differ from the admissible receive values in rvals(V^{p_i}, p_i).

Let $f_\ell^{ra} \leq f_\ell^r$ be the maximum number of non-omissive, i.e., non-empty and hence value faulty, $V_q^{p_i}$ among those.

(A2) The receiver of a message knows who sent it.

(A3) The absence of a message from sender p can be detected at any receiver q, which leads to $V_q^p = E$ for some distinguished value E.

Remarks:

1. Assumption (A3) ultimately implies a synchronous system, where all non-faulty nodes operate in lockstep rounds. Any node's round consists of some local computation based upon the messages received in the previous round, the broadcast of the resulting messages to all nodes (including itself) in the system (A1^s), and the reception of those messages (A1^r).

2. A sender node that suffers from link faults according to (A1^s) is said to commit a *broadcast fault*, whereas a receiver node that experiences link faults according to (A1^r) is said to commit a *receive fault*. Each node's receive resp. broadcast fault has its own "budget" f_ℓ^r resp. f_ℓ^s of individual link faults, which are independent of node faults, and the particular links actually hit are usually different for any two message broadcasts resp. receptions.

3. The model parameters f_ℓ^r and f_ℓ^s are not independent of each other: If a message from node p to q is hit by a fault in p's message broadcast, it contributes a fault in node q's message reception as well. In fact, (A1^s) and (A1^r) can only be guaranteed unconditionally if

$$f_\ell^s \leq f_\ell^r \quad \text{and} \quad f_\ell^{sa} \leq f_\ell^{ra}. \tag{1}$$

Note that (1) implies $f_\ell^r = 0 \Rightarrow f_\ell^s = 0$, and $f_\ell^r = f_\ell^s$ is obviously the most natural choice.

As a final remark, we note the obvious fact that the consequences of an incomplete communication graph can be viewed as (persistent) link omission faults [25]. Consequently, any analysis under the model of Definition 1 provides results that are valid for partially connected networks as well. Note carefully, however, that our lower bound Theorem 2 requires $n > f_\ell^r + f_\ell^s$ for solving consensus. Hence, every node's broadcast must be correctly received by at least f_ℓ^r nodes here.

3 The Hybrid Oral Messages Algorithm

In this section, we will show that the *Hybrid Oral Messages* algorithm (OMH) derived from [10] in [12] solves consensus under the system model of Section 2. To retain compatibility with the existing papers, it will be presented in its original "Byzantine generals" style, where the value v of a dedicated *transmitter* is to be disseminated to the remaining $n-1$ *receivers* (the transmitter already knows its value). Eventually, every non-faulty node p—including the transmitter—must *deliver* a value v_p ascribed to the transmitter that satisfies the following properties:

(B1) (*Agreement*): If nodes p and q are both non-faulty, then both deliver the same $v_p = v_q$.

(B2) (*Validity*): If node p is non-faulty, the value v_p delivered by p is

- v, if the transmitter is non-faulty,
- E, if the transmitter is manifest faulty,
- v or E, if the transmitter is omission faulty,
- the value actually sent, if the transmitter is symmetric faulty,
- unspecified, if the transmitter is arbitrary faulty.

A fully-fledged consensus algorithm is obtained by using a separate instance of Byzantine agreement for disseminating any node's local value and using a suitable choice function (majority) for the consensus result.

The algorithm OMH as specified in Definition 2 below uses two primitives:

- The *wrapper function* $R(v)$ encodes a statement "I am reporting v" as a unique value. Reporting is undone by means of the inverse function $R^{-1}(v)$, which must guarantee $R^{-1}(R(v)) = v$. Note that only E, $R(E)$, $R(R(E))$, $R(R(R(E)))$, $\dots$ must actually be distinguishable here; for each legitimate value v, we can allow $R(v) = R^{-1}(v) = v$.

- The *hybrid-majority* of a set $\mathcal{V}$ of values provides the majority of all non-E values in $\mathcal{V}$. If no majority exists, the default value $R(E)$ is returned.

Consult [12] for a detailed discussion of the above primitives and the operation of OMH in general.

Definition 2 (Algorithm OMH [12]) *The Hybrid Oral Message algorithm OMH is defined recursively as follows:*

OMH(0):

1. *The transmitter sends its value v to every receiver.*
2. *Every receiver p delivers the value v_p received from the transmitter, or the value E if a missing or manifestly erroneous value was received.*

OMH(m), $m > 0$:

1. *The transmitter sends its value v to every receiver.*
2. *For every p, let w_p be the value receiver p receives from the transmitter, or E if no value or a manifestly bad value was received.*

   Every receiver p acts as the transmitter in Algorithm OMH($m-1$) to communicate the value $R(w_p)$ to all[2] receivers.

[2]There are $n-1$ receivers in the first instance OMH(m) of the algorithm; the transmitter does not participate in any way in further recursive instances. Our n-node, $m+1$-round Byzantine agreement algorithm OMH(m) can hence be viewed as an initial broadcast of the transmitter's value to all receivers combined with an $n-1$-node, m-round consensus algorithm.

3. *For every p and $q \neq p$, let w_p^q be the value receiver p delivers as the result of OMH($m-1$) initiated by receiver q in step 2 above, or else E if no w_p^q or a manifestly bad value was delivered. Every receiver p calculates the hybrid-majority value among all values w_p^q and its own value $w_p^p = R(w_p)$, and applies R^{-1} to that value. The result is delivered as the transmitter's value v_p.*

By adopting and extending the analysis of [12] for our perception-based fault model, it is possible to show that OMH satisfies (B1) and (B2). We start with the validity property (B2), which is secured by Lemma 1. Note carefully that the lemma is void in case of an arbitrary faulty transmitter, since (B2) does not say anything about the value a receiver ascribes to the transmitter in this case.

Lemma 1 (Validity) *For any $m \geq \min\{1, f_\ell^s\}$ and any f_a, f_s, f_o, f_m, f_ℓ^s, f_ℓ^r, f_ℓ^{ra}, algorithm OMH(m) satisfies the validity property (B2) if there are strictly more than $2f_\ell^s + f_\ell^r + f_\ell^{ra} + 2(f_a + f_s) + f_o + f_m + m$ participating nodes.*

Proof: The proof is by induction on m. For the base case, assume that the transmitter sends some value ν ($\nu = v$ if the transmitter is non-faulty) to all receivers. Ignoring link faults for the moment, we only have to distinguish the following cases: For any non-faulty receiver p,

(1) $v_p = \nu = v$, if the transmitter is non-faulty,

(2) $v_p = \nu = E$, if the transmitter is manifest faulty,

(3) $v_p = v$ or $v_p = E$, if the transmitter is omission faulty.

(4) $v_p = \nu$, if the transmitter is symmetric faulty.

Therefore, if $f_\ell^s = 0$, i.e., if there are no link faults, receiver p can simply deliver its received value v_p according to OMH(0) to ensure (B2). The induction starts at $m = 0$ in this case.

If $f_\ell^s > 0$, however, induction must start with $m = 1$ as the base case: According to the definition of OMH(1), every receiver p of step 1 of OMH(1) uses OMH(0) to disseminate its w_p to all other receivers q. Let us first consider the cases where the transmitter is not omission faulty, i.e., (1), (2) and (4): Abbreviating the number of initially participating receivers by

$$n' \geq 2f_\ell^s + f_\ell^r + f_\ell^{ra} + 2(f_a + f_s) + f_o + f_m + m, \quad (2)$$

there must be at least $n' - f_\ell^s - f_a - f_s$ obedient receivers p of step 1 of OMH(1) that get the same $w_p = \nu$ (recall that $\nu = E$ in case of a manifest faulty transmitter), despite the at most f_ℓ^s link faults according to (A1^s).

It hence follows that any non-faulty receiver q of step 1 of OMH(0) obtains at least n'_q identical values $R(\nu)$ with

$$\begin{aligned} n'_q &= n' - f_\ell^s - f_a - f_s - f'_m - f'_o \\ &\quad -(f_\ell^{ra\prime} - f_o^{a\prime} - f_m^{a\prime}) - f_\ell^{ro\prime}, \end{aligned} \quad (3)$$

where $f'_m \leq f_m$ resp. $f'_o \leq f_o$ is the actual number of manifest resp. omission faulty nodes that did not deliver the correct value $R(\nu)$ to node q but rather E (due to an omission node/link fault) or even some arbitrary value (due to an arbitrary link fault). Furthermore, $f_\ell^{ro\prime}$ resp. $f_\ell^{ra\prime} \leq f_\ell^{ra}$ with $f_\ell^{ro\prime} + f_\ell^{ra\prime} \leq f_\ell^r$ denotes the number of omission resp. value faults caused by link faults according to (A1^r), and $f_o^{a\prime}$ and $f_m^{a\prime}$ is the number of arbitrary link faults that hit omission and manifest faulty senders, respectively. Note carefully that (3) is also valid for any OMH(0)'s transmitter ($q = p$), which must be non-faulty if at all considered here and must hence own the correct value $R(\nu)$. Since we assumed exactly f'_m manifest faulty receivers, our receiver q gets a minimum of $f'_o - f_o^{a\prime} + f'_m - f_m^{a\prime} + f_\ell^{ro\prime}$ values equal to E, hence at most $n''_q = n' - f'_o + f_o^{a\prime} - f'_m + f_m^{a\prime} - f_\ell^{ro\prime}$ values different from E. Note carefully that we need not consider non-E-values from nodes outside the n' participating receivers here, since q knows this set and can hence restrict its attention only to legitimate ones.

Recalling (3), we thus find

$$\begin{aligned} 2n'_q - n''_q &= n' - 2f_\ell^s - 2f_a - 2f_s - f'_o - f'_m \\ &\quad -2f_\ell^{ra\prime} - f_\ell^{ro\prime} + f_o^{a\prime} + f_m^{a\prime} \\ &\geq f_\ell^r + f_\ell^{ra} - 2f_\ell^{ra\prime} - f_\ell^{ro\prime} \\ &\quad +(f_o - f'_o) + (f_m - f'_m) + m \\ &> 0 \end{aligned} \quad (4)$$

since $m = 1$, $f_\ell^{ro\prime} + f_\ell^{ra\prime} \leq f_\ell^r$ and $f_\ell^{ra\prime} \leq f_\ell^{ra}$, which implies that $R(\nu)$ wins the hybrid-majority at any non-faulty receiver. Since R^{-1} is applied to the result, the final value ν is obtained as required.

Turning to the remaining case (3) for $m = 1$, where the transmitter is omission faulty, n'_q given by (3) is a lower bound on the number of values obtained by a non-faulty receiver q that are either $R(E)$ or else $R(\nu)$, depending upon whether its source node encountered an omission from the transmitter or not. We cannot hope to get a majority for either of those values in the general case, but since (4) still holds, it is also clear that if there is no majority for either $R(E)$ or $R(\nu)$, then there cannot be a majority for any other non-E-value as well. Hence, by our default value for the hybrid-majority primitive, $R(E)$ is returned here. In any case, (B2) is also satisfied for omission faulty transmitters.

Assuming now that the lemma is already true for $m - 1 \geq \min\{1, f_\ell^s\}$, we will show that it is also true for m: The proof is almost the same as for the base case; we only have to replace the application of OMH(0) by OMH($m-1$) with n' participants: For case (1), (2) and (4), we have at least $n' - f_\ell^s - f_a - f_s$ obedient receivers p of step 1 of OMH(m) that apply OMH($m-1$) to consistently disseminate their $R(w_p) = R(\nu)$. Since both m and the number of participants decreased by one, we can apply the induction hypothesis to OMH($m-1$) to conclude that any non-faulty receiver q actually delivers $R(\nu)$ in this step. Consequently, any non-faulty receiver q must have at least

$\overline{n}'_q = n' - f^s_\ell - f_a - f_s - f'_o - f'_m$ values equal to $R(\nu)$ among the at most $\overline{n}''_q = n' - f'_o - f'_m$ non-E values it may have got at all. Herein, $f'_m \leq f_m$ resp. $f'_o \leq f_o$ gives the number of E-values delivered to receiver q by OMH$(m-1)$ when p was manifest resp. omission faulty. Since $m > 0$, $2\overline{n}'_q - \overline{n}''_q > 0$ as before, so $R(\nu)$ wins the hybrid-majority at any non-faulty receiver and the final value $\nu = R^{-1}(R(\nu))$ follows.

For the remaining case (3), exactly the same reasoning as for $m = 1$ reveals that only the default value $R(E)$ can be returned by hybrid-majority if no majority of either $R(\nu)$ or $R(E)$ exists. This eventually completes the proof of Lemma 1. □

Using Lemma 1, an induction proof similar to the one of [12] can be used to show the following major result:

Theorem 1 (Agreement and Validity) *For any $m \geq f_a + f_o + \min\{1, f^s_\ell\}$ and any f_a, f_o, f_s, f_m, f^s_ℓ, f^r_ℓ, f^{ra}_ℓ, the algorithm OMH(m) satisfies agreement (B1) and validity (B2) if there are strictly more than $2f^s_\ell + f^r_\ell + f^{ra}_\ell + 2(f_a + f_s) + f_o + f_m + m$ participating nodes.*

Proof: See [24, Thm. 5].

Remarks:

1. It is apparent from Theorem 1 that $2f_o$ nodes are required by OMH to tolerate f_o omission faulty nodes, which is definitely sub-optimal: Algorithms like the one of [16] require only $f_o < n - 1$. This is not due to an overly conservative analysis, but rather the price paid for the ability to mask additional symmetric and asymmetric faults.

2. For OMH, receive faults (A1^r) resulting in an omission are easier to tolerate than those that produce a value fault, cf. Theorem 1, since $f^r_\ell + f^{ra}_\ell = f^{ro}_\ell + 2f^{ra}_\ell$ with f^{ro}_ℓ bounding the number of "pure" omission faults. Interestingly, this is not the case for the f^s_ℓ broadcast faults (A1^s) here.

3. From the proof of Lemma 1, it is apparent that (A1^r)—and one additional round—is only required to eventually rule out the inconsistencies caused by (A1^s). This is solely done in the base case $m = 1$ of the induction, which implies that limiting the number of link faults for a single receiver by f^r_ℓ according to (A1^r) is <u>only</u> required in the last round, where in turn (A1^s) is not explicitly used.

4 Formal Verification

Theorem 1 and its supporting Lemma 1 were developed through an iterative process that interleaved manual proof and formal verification.

The main technical challenge in developing Theorem 1 is to formulate the fault model in such a way that node and link faults are independent, yet the proof is tractable. Formal verification identifies the exact assumptions used in a proof, and thereby revealed that earlier formulations of the fault model finally expressed in Definitions (A1^s) and (A1^r) did not achieve the independence desired. This led to a reformulation of the fault model and a new formal verification, and then to a final iteration that produced the formulation presented here.

The formal verification used the PVS verification system [15] and builds on previous formal analysis of the OMH algorithm under a simple hybrid fault model [11]. Verification was performed by specifying the algorithm and its fault model in the specification language of PVS, and then using the PVS theorem prover to develop mechanically-checked proofs for a number of lemmas, culminating in those corresponding to Lemma 1 and Theorem 1. The formal verification proved to be quite a challenging exercise: its development took about two weeks of full-time effort, and a similar amount of part-time preparation and experiment. Some of the challenges were of a technical nature, due to difficulties in finding a reasonably automated way to reason over expressions involving finite cardinalities, but most were due to the numerous cases that must be considered and the delicacy with which assumptions must be stated and matched up across various lemmas and axioms. Full details, and the PVS specification and proof script, are available on the Web [19].

Formal verification has recognized value in providing strong assurance for correct results and revealing the flaws of incorrect ones. Less widely recognized, but no less valuable, is its ability to help refine the statements of lemmas and theorems and their supporting assumptions—for once a mechanized formal proof has been developed, it is relatively straightforward to experiment with generalizations and alternative formulations. We derived benefits of all three types during development of this particular formal verification.

First, we are able to provide strong assurance for the results stated in Lemma 1 and Theorem 1. The proofs of these results involve extensive case analysis: there are separate cases $m = 0$, $m = 1$, and $m > 1$, each considered for the cases of non-faulty, manifest, omission, symmetric, and arbitrary faulty transmitters, and all further compounded by the presence of broadcast and receive link faults of both omissive and non-omissive kinds. The human tendency is to group some cases together, and to relegate others to a terse statement of the kind "similarly,..." This approach has obvious vulnerabilities, but is very hard to overcome without assistance—because human patience seldom extends to independent scrutiny of numerous similar cases. A mechanized theorem prover examines each variation with the same untiring scepticism.

Second, we were able to refine the assumptions; in particular, we can now treat link faults independently of node faults. As originally stated in [24, Def. 1], definition (A1^s) and (A1^r) were restricted to non-faulty sender nodes. Examination of the proof to Lemma 1, however, revealed that the constraint was implicitly applied to sender nodes that are non-faulty, manifest faulty, or symmetric faulty. In our new formal treatment, we extend this treatment to apply to all kinds of senders, thereby allowing node faults and broadcast link faults to occur independently. We also made explicit that receive link faults are of concern only in Step 2 of OMH(1), see [19] for all the details.

Third, we were able to detect the erroneous conjecture in [24] related to *uniform* [9] validity (B2) and agreement (B1): Although we were able to formally verify that extending validity to obedient nodes is indeed possible, we failed to show this for agreement. A simple counterexample is an omission faulty transmitter that sends v to itself but E to all other nodes. The receivers will all agree on E but the transmitter will have v. There are applications where such disagreement could be catastrophic. An example is distributed brake-by-wire in automobiles, where the values exchanged are the braking forces applied at each wheel and non-faulty nodes compensate for failed nodes (which are assumed to fail with their wheels free-running). All receivers will assume that the transmitter has failed and will adjust their braking force to compensate for its presumably free-running wheel, whereas the transmitter will be unaware of any fault and will apply the full force v to its wheel.

5 Theoretical Results

In order to be able to compare our algorithm's fault-tolerance degree to the optimal one, we need suitable[3] lower bound results. For pure omission link faults, it is not too difficult to derive a (tight) lower bound by means of a generalization of the proof of the impossibility result of [8] according to [13, Thm. 5.1]:

Theorem 2 (Lower Bound 1) *Any deterministic algorithm that solves consensus under the system model of Definition 1 for* $f_\ell^r \geq f_\ell^s$ *needs* $n > f_\ell^r + f_\ell^s$.

Proof: See [24, Thm. 6]. □

Note carefully that this result allows a broadcast fault to affect at most a minority f_ℓ^s of nodes. The original impossibility result of [8] considered the case $n = 2$ only, however, where no non-empty minority of nodes exists. This might be one of the reasons why our fairly simple escape from the impossibility result was not discovered earlier during the past 20 years.

Unfortunately, the proof of Theorem 2 cannot easily be generalized to the case of arbitrary link faults, where a link could produce an arbitrary faulty value as well. However, a new instance of the "easy impossibility proof techniques" of [6] can be applied here:

Theorem 3 (4-Node Impossibility) *There is no deterministic algorithm that solves consensus under the system model of Definition 1 for a single arbitrary link fault in a 4-node system.*

Proof: We show that any deterministic algorithm violates agreement if every node in a 4-node system may see an inconsistent value from one of its neighbors. Suppose that our 4 nodes execute a distributed algorithm consisting of specific programs A, B, C, D, which solve consensus under the system model of Definition 1 with $f_\ell^{ra} = f_\ell^{sa} = f_\ell^s = f_\ell^r = 1$. In order to derive a contradiction, we arrange 8 nodes in a cube as shown in Figure 1. For example, the lower leftmost node labeled $A[0]$ executes program A starting with initial value 0 (the $\boxed{0}$ on its left displays this node's decision value, as explained below). Note carefully that all nodes and all links are assumed to be non-faulty here.

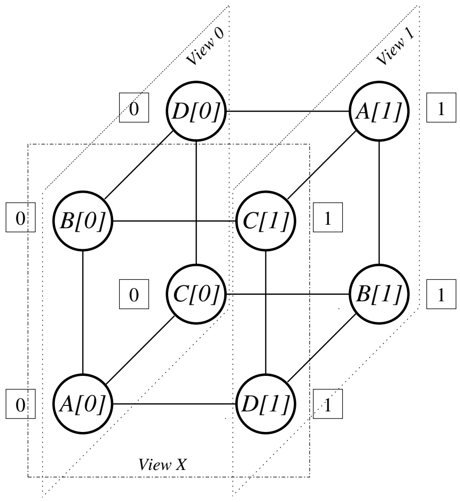

Figure 1. *Topology used for proving the violation of agreement in a 4-node system.*

Of course, dealing with a solution for a 4-node system, we cannot expect to achieve consensus in the 8-node system of Figure 1. However, due to the special assignment of programs to nodes, each node observes a neighborhood as in a 4-node system. More specifically, the 4 nodes at any side of the cube (we call it a *view*) can be interpreted as an instance of a legitimate 4-node system. In fact, as can be checked easily, our assignment ensures that any node in a view is connected to exactly one node outside this view. Since we assumed that every node may see an arbitrary faulty input from at most one neighbor, the input from the node outside the view may be arbitrary—it just appears like an arbitrary faulty node.

Now consider the nodes in view 0, which all have initial value 0. By the validity property for consensus, all nodes must decide 0 here (the initial value 1 of the nodes outside view 0 do not matter, as they are considered arbitrary faulty w.r.t. view 0). Similarly, in view 1, all nodes must decide 1 since they have initial value 1. But now the nodes in view X

[3]Note that we will entirely focus on consensus in this section, since the impossibility results and lower bounds carry over to Byzantine agreement (as the latter can be used to implement consensus), recall our discussion at the beginning of Section 3 and Footnote 2. The validity property for consensus requires that if all obedient nodes start with the same input value, then their decision must be that value; the agreement property is the same as for Byzantine agreement.

face a problem: Since e.g. the lower leftmost node $A[0]$ observes exactly the same messages in view X as in view 0, it must decide 0 as observed above. Similarly, as the lower rightmost node $D[1]$ observes exactly the same messages in view X as in view 1, it must decide 1—but this would violate agreement in the 4-node system corresponding to view X. We hence established the required contradiction, thereby completing the proof of Theorem 3. □

Using Theorem 3, it is not too difficult to prove the following tight lower bound for arbitrary link faults (matched by OMH for $f_\ell^s = f_\ell^{sa}$, cf. Theorem 1).

Theorem 4 (Lower Bound 2) *Any deterministic algorithm that solves consensus under the system model of Definition 1 with $f_\ell^r \geq f_\ell^s$ and $f_\ell^{ra} \geq f_\ell^{sa}$ needs $n > f_\ell^r + f_\ell^{ra} + f_\ell^s + f_\ell^{sa}$.*

Proof: See [24, Thm. 8].

There are some implications of Theorem 1, which are also interesting from a theoretical point of view. From Definition 1, it is apparent that an arbitrary broadcast fault ($A1^s$) can also be viewed as the consequence of a *restricted node fault* with inconsistency limited to f_ℓ^s. Since $f_\ell^s < n$, the inconsistency caused by such a broadcast fault is in fact strictly less than that of an arbitrary faulty node, which is not restricted in the number of recipients that may experience a faulty message.

Using this alternative interpretation, the result of Theorem 1 allows us precisely to characterize what makes a node fault an arbitrary (Byzantine) one for OMH: Setting $f_a = f_s = f_o = f_m = 0$, $f_\ell^r = f_\ell^{ra} > 0$ arbitrary and fixing some $n > 2f_\ell^{ra} + 1$, it turns out that OMH(1) can tolerate f_ℓ^{ra} restricted arbitrary node faults with inconsistency limited to

$$f_\ell^{sa} \leq \lfloor n/2 \rfloor - 1 - f_\ell^{ra}. \tag{5}$$

For example, $n = 10$ nodes are required for tolerating one restricted arbitrary fault with inconsistency $f_\ell^{sa} = 3$. Due to (5), at least $\lceil n/2 \rceil + f_\ell^{ra} + 1$ nodes, i.e., a majority of the non-faulty nodes, get the correct message even in the broadcast of a restricted arbitrary faulty node. Therefore, provided that n is chosen appropriately, any f_ℓ^{ra} can be tolerated by the two-round algorithm OMH(1), i.e., *in a number of rounds that does not depend upon the number of faults f_ℓ^{ra}!*

It hence follows that a node must be considered arbitrary faulty and accounted for in f_a only if it can supply wrong information to more than a minority of the non-faulty nodes. Note that our observations do not contradict the lower-bound $f + 1$ for the number of rounds required for consensus in presence of f faulty nodes, since its proof [1, Sec. 5.1.4] relies heavily upon the fact that a faulty node can disseminate inconsistent information to as many nodes as desired.

6 Assumption Coverage

To apply a deterministic fault model like the one of Definition 1 in practice, one has to address the question of assumption coverage. More specifically, for the particular system in mind, the *probability of failure Q* implied by a possible violation of the fault assumptions (f_a, f_s, f_o, f_m, f_ℓ^s, f_ℓ^r) needs to be evaluated. This is a mandatory step for any algorithm where safety depends upon non-violation of the fault model. It is particularly important for link faults, however, since Q increases with every message broadcast during the execution of the algorithm here: According to ($A1^s$) resp. ($A1^r$), no message broadcast resp. reception may suffer from more than f_ℓ^s resp. f_ℓ^r link faults. Given the fact that OMH sends many, many messages, the question arises whether Q can eventually be made as small a desired by choosing suitable values of f_ℓ^s and f_ℓ^r.

In Theorem 5 below, we will present an upper bound on the probability of failure Q_m of OMH(m) for a simple probabilistic model of link faults: We assume that the probability of losing or corrupting a single message on the link or in the network interface is $0 < p < 1$, and that those link faults occur independently of each other. Note that $f_\ell^s = f_\ell^r = f_\ell$ is the only meaningful choice in this setting. Despite of its simplicity, this model is commonly used in practice, since it is analytically tractable and facilitates easy comparison of results. It is in fact a quite accurate and realistic model for uncorrelated transient channel/network interface faults in homogeneous system architectures. Persistent and, in particular, correlated faults are of course beyond its scope.

Theorem 5 (Assumption Coverage OMH) *For $n - m - f_\ell - 2 \geq 1$ and $np < 1$ sufficiently small, the probability of failure Q_m of OMH(m) satisfies*

$$Q_m \leq Q'_m + \mathcal{O}\Big(\frac{(Q'_m)^2}{[n - f_\ell - 2]_m}\Big) = \mathcal{O}\Big(n^m \frac{(np)^{f_\ell+1}}{(f_\ell+1)!}\Big), \tag{6}$$

where

$$Q'_m = \Big(1 + \frac{1}{n - m - f_\ell - 2}\Big)[n-1]_{m+f_\ell+1} \frac{p^{f_\ell+1}}{(f_\ell+1)!}.$$

Proof: See [24, Thm. 8]. □

In Tables 1 and 2, we give a few numerical values for Q'_m for different values of m and f_ℓ in case of $n = 4f_\ell + 3m + 1$, which allows e.g. $f_\ell^s = f_\ell^r = f_\ell$, $f_a = m - 1$, $f_o = 0$, and $f_s = f_m = 1$ by Theorem 1.

In order to assess the dependency of the probability of failure Q_m upon the model parameters n, m, f_ℓ, we substitute $n = n_0 + c\ell$ in (6), where $c \cdot \ell$ gives the number of nodes that must be added to cope with a (sufficiently small) number ℓ of additional link faults per node:

$$Q_m = \mathcal{O}\Big(n_0^m \frac{(n_0 p)^{f_\ell+1}}{(f_\ell+1)!} \cdot \frac{(n_0 p \cdot e^c)^\ell}{[f_\ell + \ell + 1]_\ell}\Big) \tag{7}$$

It is hence apparent from (7) that, as long as $np < 1$ sufficiently small, the probability of failure Q_m

f_ℓ	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
1	0.01	0.3	1	1	1	1
2	0.002	0.04	1	1	1	1
3	0.0002	0.006	0.3	1	1	1
5	2.10^{-6}	0.00009	0.005	0.3	1	1
7	2.10^{-8}	1.10^{-6}	0.00009	0.007	0.6	1
10	2.10^{-11}	2.10^{-9}	2.10^{-7}	0.00002	0.002	0.2
15	2.10^{-16}	2.10^{-14}	3.10^{-12}	4.10^{-10}	5.10^{-8}	8.10^{-6}
20	2.10^{-21}	2.10^{-19}	4.10^{-17}	7.10^{-15}	1.10^{-12}	2.10^{-10}

Table 1. *Value of (approximate) probability of failure Q'_m for $p = 0.01$.*

f_ℓ	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
1	1.10^{-10}	3.10^{-9}	9.10^{-8}	3.10^{-6}	0.0001	0.007
2	2.10^{-15}	4.10^{-14}	2.10^{-12}	7.10^{-11}	4.10^{-9}	2.10^{-7}
3	2.10^{-20}	6.10^{-19}	3.10^{-17}	1.10^{-15}	7.10^{-14}	5.10^{-12}
5	2.10^{-30}	9.10^{-29}	5.10^{-27}	3.10^{-25}	2.10^{-23}	2.10^{-21}
7	2.10^{-40}	1.10^{-38}	9.10^{-37}	7.10^{-35}	6.10^{-33}	5.10^{-31}
10	2.10^{-55}	2.10^{-53}	2.10^{-51}	2.10^{-49}	2.10^{-47}	2.10^{-45}
15	2.10^{-80}	2.10^{-78}	3.10^{-76}	4.10^{-74}	5.10^{-72}	8.10^{-70}
20	2.10^{-105}	2.10^{-103}	4.10^{-101}	7.10^{-99}	1.10^{-96}	2.10^{-94}

Table 2. *Value of (approximate) probability of failure Q'_m for $p = 0.000001$.*

- rapidly grows with m and hence with the number $f_a + f_o$ of arbitrary and omission faults,
- marginally grows with n and hence with the number of any kind of faults,
- decreases with the number of tolerated link faults f_ℓ (and with decreasing p, of course), since the last factor in (7) is < 1 for any suitably chosen ℓ.

Note carefully that the latter implies that increasing f_ℓ is always beneficial for reasonable parameter settings, which actually justifies our whole approach.

7 Comparison with Existing Work

Most work on deterministic consensus we are aware of employs a node-centric fault model, like the one that at most f nodes may behave Byzantine during the execution. Channel or receiver-originating link faults, however, are difficult to accommodate in such models in case of synchronous[4] systems: Practical considerations suggest to grant every receiver node its own budget of f_ℓ link faults per round, which may hit any of the incoming links. If those link faults are simply mapped to (sender) node faults, as in the model of [7], however, even $f_\ell = 1$ would generate many fault patterns where all $f = n$ nodes must be considered faulty.

A similar argument applies to the more detailed send/receive-omission fault model of [16], where receive omissions are mapped to receiver node faults. Although it has been observed in this paper that only the number of nodes that commit a send omission (but not the number of nodes committing a receive omission) needs to be counted in f, agreement has only been shown to hold for a node that did not commit either type of fault. Hence, in the previous example $f_\ell = 1$, no node would remain that could be guaranteed to reach agreement.

In both models, the situation gets worse by the fact that the f_ℓ receive omissions of a single receiver node could hit different inbound links in different rounds of the execution. Since node faults are usually considered persistent during an execution, the "exhaustion" of non-faulty nodes would progress rapidly with every round, which makes any attempt to solve consensus in such models even more hopeless.

There are also a few papers on Byzantine agreement that explicitly model link faults [17, 20, 25]. However, all these works allow only a very limited number $\lfloor (n-2)/2 \rfloor = \mathcal{O}(n)$ of link faults system-wide during the whole execution. Moreover, only very specific algorithms are considered, and neither the model nor the analysis seems to be generally applicable.

8 Conclusions

In this paper, we showed that deterministic consensus in presence of link faults is possible, despite the impossibility result of [8]: We introduced a novel perception-based hybrid fault model, which grants every node at most f_ℓ^r independent receive link faults (with at most f_ℓ^{ra} non-omission faults among those) and f_ℓ^s broadcast link faults in each round, in addition to at most f_a, f_s, f_o, f_m arbitrary, symmetric, omission, and manifest node faults. By means of a formally verified proof, we showed that the $m+1$-round *Hybrid Oral Messages* Byzantine agreement algorithm OMH(m) with $m \geq f_a + f_o + 1$ satisfies validity and agreement of under this fault model as long as $n > 2f_\ell^s + f_\ell^r + f_\ell^{ra} + 2(f_a + f_s) + f_o + f_m + m$. This reveals that the resilience of consensus algorithms is much better than previous results suggested: Since f_ℓ^s, f_ℓ^r could be as much as $\mathcal{O}(n)$, OMH can in fact cope with up to $\mathcal{O}\big((m+1)n^2\big)$ link faults in the system during the whole execution.

We also provided an impossibility result and associated lower bound for the required number of nodes in presence of

[4]Note that it is relatively easy to handle link faults in asynchronous systems satisfying the "fair loss" property: A perfect link can be simulated by suitable retransmission schemes [3]. In synchronous systems, however, the duration of the rounds must be increased according to the maximum number of successive message losses that are to be tolerated. In sharp contrast, our approach exploits resource (node) redundancy only and is therefore content with pure datagram communications without retransmissions.

An alternative approach has been suggested for partially synchronous systems [5], where it is assumed that an arbitrary number of link faults may occur up to some unknown stabilization time t_s. After time t_s, however, all the links in the system must behave perfect. It is immediately apparent from setting $f_\ell = 0$ in Theorem 5, though, that this approach has much worse coverage in real systems than ours.

arbitrary link faults, which is matched by OMH and therefore tight. Moreover, our investigations led to a precise characterization of what makes a node fault arbitrary. Last but not least, we conducted an analysis of the assumption coverage for a simple probabilistic system model, where link faults occur with a fixed probability p independently of each other. It shows that adding nodes in order to tolerate additional link faults per node always decreases the probability of violating the link fault assumption $f_\ell^r = f_\ell^s = f_\ell$ as long as $np < 1$ sufficiently small. Consequently, for reasonably small m, even OMH—not to speak of the polynomial consensus algorithms analyzed in [4]—could be used even in wireless systems, where link failure rates up to $p = 10^{-2}$ are common.

References

[1] H. Attiya and J. Welch. *Distributed Computing*. McGraw-Hill, 1998.

[2] M. Azadmanesh and R. M. Kieckhafer. Exploiting omissive faults in synchronous approximate agreement. *IEEE Transactions on Computers*, 49(10):1031–1042, Oct. 2000.

[3] A. Basu, B. Charron-Bost, and S. Toueg. Simulating reliable links with unreliable links in the presence of process crashes. In *Proceedings of the 10th International Workshop on Distributed Algorithms (WDAG'96)*, volume 1151 of *LNCS*, pages 105–122. Springer, Oct. 1996.

[4] M. Biely and U. Schmid. Message-efficient consensus in presence of hybrid node and link faults. Technical Report 183/1-116, Department of Automation, TU Vienna, August 2001. (submitted).

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

[6] M. Fischer, N. Lynch, and M. Merrit. Easy impossibility proofs for the distributed consensus problem. *Distributed Computing*, 1(1):26–39, 1986.

[7] L. Gong, P. Lincoln, and J. Rushby. Byzantine agreement with authentication: Observations and applications in tolerating hybrid and link faults. In *Proceedings Dependable Computing for Critical Applications-5*, pages 139–157, Champaign, IL, Sept. 1995.

[8] J. 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.

[9] V. Hadzilacos and S. Toueg. Fault-tolerant broadcasts and related problems. In S. Mullender, editor, *Distributed Systems*, chapter 5, pages 97–145. Addison-Wesley, 2nd edition, 1993.

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

[11] P. Lincoln and J. Rushby. Formal verification of an algorithm for interactive consistency under a hybrid fault model. In C. Courcoubetis, editor, *Computer-Aided Verification, CAV '93*, volume 697 of *Lecture Notes in Computer Science*, pages 292–304, Elounda, Greece, June/July 1993. Springer-Verlag.

[12] P. Lincoln and J. Rushby. A formally verified algorithm for interactive consistency under a hybrid fault model. In *Proceedings Fault Tolerant Computing Symposium 23*, pages 402–411, Toulouse, France, June 1993.

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

[14] F. J. Meyer and D. K. Pradhan. Consensus with dual failure modes. In *In Digest of Papers of the 17th International Symposium on Fault-Tolerant Computing*, pages 48–54, Pittsburgh, July 1987.

[15] S. Owre, S. Rajan, J. Rushby, N. Shankar, and M. Srivas. PVS: Combining specification, proof checking, and model checking. In R. Alur and T. A. Henzinger, editors, *Computer-Aided Verification, CAV '96*, volume 1102 of *Lecture Notes in Computer Science*, pages 411–414, New Brunswick, NJ, July/Aug. 1996. Springer-Verlag.

[16] K. J. Perry and S. Toueg. Distributed agreement in the presence of processor and communication faults. *IEEE Transactions on Software Engineering*, SE-12(3):477–482, March 1986.

[17] S. S. Pinter and I. Shinahr. Distributed agreement in the presence of communication and process failures. In *Proceedings of the 14th IEEE Convention of Electrical & Electronics Engineers in Israel*. IEEE, Mar. 1985.

[18] J. Rushby. A formally verified algorithm for clock sychronization under a hybrid fault model. In *Proceedings ACM Principles of Distributed Computing (PODC'94)*, pages 304–313, Los Angeles, CA, Aug. 1994.

[19] J. Rushby. Formal verification of hybrid Byzantine agreement under link faults. Technical report, Computer Science Laboratory, SRI International, Menlo Park, CA, 2001. Available at http://www.csl.sri.com/~rushby/abstracts/byzlinks01.html.

[20] H. M. Sayeed, M. Abu-Amara, and H. Abu-Amara. Optimal asynchronous agreement and leader election algorithm for complete networks with Byzantine faulty links. *Distributed Computing*, 9(3):147–156, 1995.

[21] U. Schmid. Orthogonal accuracy clock synchronization. *Chicago Journal of Theoretical Computer Science*, 2000(3):3–77, 2000.

[22] U. Schmid. How to model link failures: A perception-based fault model. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN'01)*, pages 57–66, Göteborg, Sweden, July 1–4, 2001.

[23] U. Schmid and K. Schossmaier. How to reconcile fault-tolerant interval intersection with the Lipschitz condition. *Distributed Computing*, 14(2):101 – 111, May 2001.

[24] U. Schmid and B. Weiss. Consensus with oral/written messages: Link faults revisited. Technical Report 183/1-110, Department of Automation, TU Vienna, Feb. 2001.

[25] H.-S. Siu, Y.-H. Chin, and W.-P. Yang. Byzantine agreement in the presence of mixed faults on processors and links. *IEEE Transactions on Parallel and Distributed Systems*, 9(4):335–345, Apr. 1998.

[26] P. M. Thambidurai and Y. K. Park. Interactive consistency with multiple failure modes. In *Proceedings 7th Reliable Distributed Systems Symposium*, Oct. 1988.

[27] C. J. Walter and N. Suri. The customizable fault/error model for dependable distributed systems. *Theoretical Computer Science*, 2000. (Special issue on Dependable Computing, to appear).