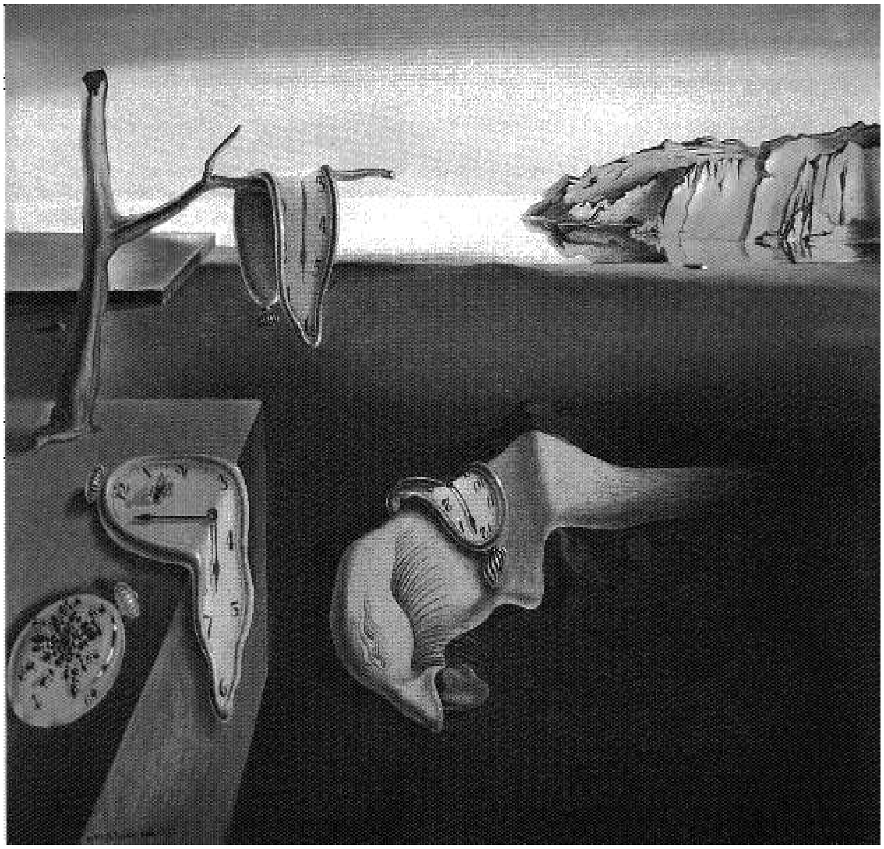
Institut für Automation
Abt. für Automatisierungssysteme

Technische
Universität
Wien

Projektbericht Nr. 183/1-110
February 2001

Consensus with Oral/Written Messages: Link Faults Revisited

Ulrich Schmid and Bettina Weiss



Salvador Dali, "Die Beständigkeit der Erinnerung"

Consensus with Oral/Written Messages: Link Faults Revisited

ULRICH SCHMID, BETTINA WEISS
Technische Universität Wien
Department of Automation
Treitlstraße 1, A-1040 Vienna
Email: {s, bw}@auto.tuwien.ac.at
Phone: ++43-1-58801-18325, FAX: ++43-1-58801-18391

Abstract

This paper[1] *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 result by moderately restricting the inconsistency that link faults may cause system-wide. Relying upon a novel perception-based hybrid fault model that provides different classes of faults for both nodes and links, we prove that the* $m+1$*-round Byzantine agreement algorithms OMH (Lincoln & Rushby, 1993) and its authenticated variants OMHA, ZA (Gong, Lincoln & Rushby, 1995) require*

$$\begin{aligned} n &> 2f_\ell^s + f_\ell^r + f_\ell^{ra} + 2(f_a+f_s) + f_o + f_m + m \\ n &> 2f_\ell^s + f_\ell^r + 2(f_a+f_s) + f_o + f_m + m \\ n &> f_\ell^s + f_\ell^r + f_a + f_s + f_o + f_m + 1 \end{aligned}$$

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* $m \geq f_a + f_o + 1$. *If signatures are broken, OMHA degrades to OMH, whereas ZA can be made tolerant to* f_b *broken signatures by increasing* f_a *accordingly. An analysis of the assumption coverage in systems where links fail independently with probability* p *reveals that adding nodes for tolerating link faults yields a vanishing probability of violating the fault model as long as* $np < 1$. *A number of theoretical results, including tight lower bounds for the number of nodes in presence of link faults and a precise characterization of what makes a node fault Byzantine, establish a sound theoretical foundation for our framework as well.*

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

1 Motivation

Although process[2] 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 fault models that capture only node faults.

Indeed, due to the high reliability of modern processors, communication-related faults like receiver overruns (run out of buffers), unrecognized packets (synchronization errors), and CRC errors (data reception problems) in high-speed wireline and, in particular, all sorts of wireless networks are increasingly dominating node faults. Such *link faults*[3] occur on the communication channel or in the network interface and can cause any data packet to be lost or even faulty. The resulting error, however, cannot reasonably be attributed to the innocent sender node. Declaring the receiver node as faulty would be overly conservative either, since a packet error does not usually imply a node failure (after all, its processor executes the particular algorithm correctly). Consequently, link faults should be a category of their own in a more realistic fault model.

Unfortunately, we do not know of any fault model for synchronous systems that adequately captures both node and link faults. We will hence provide a suitable one in this paper, which is a generalization of the *perception-based fault model* developed in [28] for single round (approximate) agreement algorithms. Belonging to the class of hybrid fault models, it will distinguish different types of node and link faults to ensure maximum fault-tolerance degree under realistic operating conditions, something that is particularly important for small n.

Still, there is a discouraging general impossibility result for deterministic consensus in presence of link faults, which

[1] 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.

[2] We will use the term *node* instead of the more abstract term *process* throughout this paper.

[3] Since sender-caused link faults, which affect more or less all the recipients, can reasonably be considered as node faults, we will use the term *link fault* exclusively for those that affect a single message reception only.

goes back to Gray's 1978 paper [12] on atomic commitment in distributed databases:

Theorem 1 (Gray's Impossibility [16, Thm. 5.1]) *There is no deterministic algorithm that solves the coordinated attack problem in a synchronous two-node system with lossy links.*

Due to this result, almost all the work on deterministic consensus developed during the past 20+ years deals with node faults only. Link faults have been addressed by randomized consensus algorithms like the one of [35], however, which circumvent the impossibility result by adding non-determinism (coin tossing) to the computations. Still, apart from sacrificing the simplicity of—and compatibility with—deterministic solutions, randomized algorithms are not suitable for all applications due to the inherent non-zero probability of failure/non-termination within a fixed number of rounds.

In order to save deterministic algorithms, one must address the question of whether and how some of the pivotal assumptions of the impossibility result could be relaxed. The present paper is the first one to show that this can indeed be done: If the power of link faults is moderately restricted with respect to the inconsistency that 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. We will establish detailed formulas for the family of *Hybrid Oral* and *Written Messages* algorithms[4] for Byzantine agreement and show that they are optimal. The correctness of our proofs has rigorously been justified by means of a formal verification using PVS in a companion paper [23]. An analysis of the assumption coverage in systems where links fail independently with probability p reveals that our approach indeed decreases the probability of violating the fault model if $np < 1$. Therefore, our algorithms can reasonably be employed even in wireless systems, where link loss probabilities up to $p = 10^{-2}$ are common.

The remaining sections of our paper are organized as follows:

- *Section 2:* Generalization of the perception-based hybrid fault model of [28] to the consensus framework.
- *Section 3:* Analysis of the *Hybrid Oral Messages* algorithm (OMH) of [15].
- *Section 4:* Introduction of authentication issues for the *Hybrid Written Messages* algorithms OMHA and ZA of [11].
- *Section 5 and 6:* Analysis of OMHA and ZA, respectively.
- *Section 7:* Analysis of the assumption coverage in typical system architectures.
- *Section 8:* Discussion of some consequences of our results, including the costs of tolerating link faults (Section 8.1), OMHA's and ZA's behavior in case of broken signatures (Section 8.2), and application in systems with a broadcast network (Section 8.3).
- *Section 9:* Conclusions and directions of further research.

2 Perception-based Fault Model

Deterministic fault models, like the one that at most f nodes may behave Byzantine in each execution, usually rest upon the total number of faults in the entire system. Channel or receiver-originating link faults, however, are difficult to accommodate in such models for synchronous[5] systems: The practical considerations of Section 1 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 [11], however, this assumption would generate many fault patterns where all $f = n$ nodes must be considered faulty; Figure 1 shows an example for $n = 4$ and $f_\ell = 1$. This suggests that consensus is not solvable in this model.

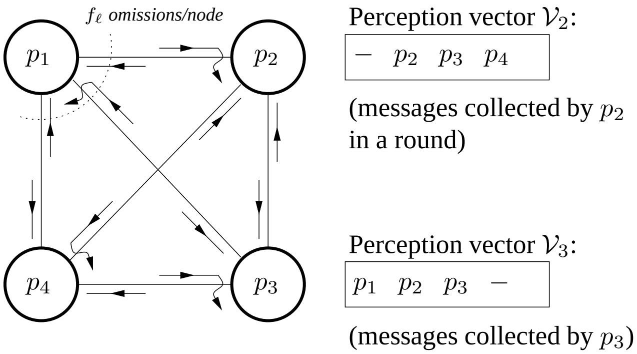

f_ℓ omissions/node
p_1
p_2
p_3
p_4
Perception vector $\mathcal{V}_2$:
$-\ p_2\ p_3\ p_4$
(messages collected by p_2 in a round)
Perception vector $\mathcal{V}_3$:
$p_1\ p_2\ p_3\ -$
(messages collected by p_3)

Figure 1. *Example of a 4-node system with $f_\ell = 1$ receive faults per node in each round, where all nodes must be considered faulty in existing fault models.*

A similar argument applies to the more detailed send/receive-omission fault model of [19], 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, agree-

[4] Note that we are aware of the fact that those algorithms suffer from an exponential number of messages. Given that they are hybrid instances of the most well-researched algorithm [14] for Byzantine agreement, however, they are certainly the most suitable candidates for introducing our fairly general approach. More efficient consensus algorithms are treated in [8].

[5] Note that it is relatively easy to handle link faults in asynchronous systems satisfying the "fair loss" property: If sending an infinite number of messages over a link causes an infinite number of messages to be received, a perfect link can be simulated by suitable retransmission schemes, see e.g. [3, 7, 39]. Clearly, this approach cannot be used in synchronous systems without unduly increasing the duration of the rounds, according to the maximum number of successive message losses that are to be tolerated.

ment has only been shown to hold for a node that did not commit either type of fault. Hence, in the example of Figure 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.

In this paper, however, we will show that the resilience of consensus algorithms is much better than the above discussion suggests: For example, Theorem 4 will establish that ZA solves Byzantine agreement in the above setting (where, in addition to node faults, every node may lose messages from up to f_ℓ arbitrary senders per round), provided that the number of nodes n_0 required for masking node faults is increased by $2f_\ell$ (and one round is added); ZA would easily achieve consensus among all nodes—viewed as receivers—in the example of Figure 1.

Since f_ℓ could be as much as $\mathcal{O}(n)$, any of our algorithms can cope with an impressive number of $\mathcal{O}\big((m+1)n^2\big)$ link faults in the system during the whole execution. This dramatically outperforms the $\lfloor (n-2)/2 \rfloor = \mathcal{O}(n)$ result of the few instances of related work on Byzantine agreement under link faults [20, 24, 32] known to us, which basically relies upon the well-known $2f+1$ connectivity lower bound of [9]. Note that our result reveals that Byzantine agreement algorithms can in fact cope with about the same number $\mathcal{O}(n^2)$ of link faults as leader election algorithms [2, 31], without, however, sacrificing the ability to deal with Byzantine faulty nodes.

The key to our results is a *perception-based hybrid fault model* for synchronous systems, which is a generalization of the one introduced for our analysis of clock synchronization and single-round agreement algorithms in [27, 28]. 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 "perceptions" of the system. Formally, node r's *perception vector*

$$\mathcal{V}_r = (V_r^1, V_r^2, \dots, V_r^n), \tag{1}$$

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. For approximate agreement algorithms, for example, the V_r^s are real values that represent the receiver's opinion about the sender's local value V^s.

In case of the single-round algorithms analyzed in [27, 28], we found it sufficient to just 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

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

Recalling the example from Figure 1, it is apparent that any two perception vectors can differ only in at most $2f_\ell = 2$ perceptions, namely, the ones where either receiver node experienced its omission. Moreover, only at most $f_\ell = 1$ of the non-faulty perceptions present at some non-faulty node can be missing or faulty at any other non-faulty node. Last but not least, 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. Hence, all existing lower bound and impossibility results remain valid.

Depending upon the type of the fault of a perception, e.g., missing or value faulty, several different classes of faults (manifest/omission/symmetric/asymmetric) can be distinguished. This leads to a *hybrid fault model* [5, 6, 15, 17, 22, 26, 29, 32, 34, 36, 37], which allows 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, cf. [21]. 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[6] obtained by setting some model parameters to 0.

For Srikanth & Toueg's consistent broadcasting primitive [33] in asynchronous systems, for example, our analysis in [28] revealed that

$$n \geq 4f_{ia} + 3f_a + 2(f_s + f_{is} + f_o + f_{io}) + f_m + 1$$

nodes are sufficient for tolerating at most f_m, f_o, f_s, f_a crash, omission, symmetric, and arbitrary node faults and f_{io}, f_{is}, and f_{ia} additional omission, symmetric, and arbitrary link faults per receiver node.

In view of these encouraging results, 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 by Theorem 1, 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, there is a suitable perception-based fault model that allows e.g. consensus with oral messages [14]

[6] Note, however, that the general hybrid analysis might be too conservative for certain restricted cases, see Remark 2 on Theorem 2 for an example.

even in wireless systems with high link failure rates, see Section 7. Recalling the "matrix" of perceptions above, it 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 [28]), see Figure 2,

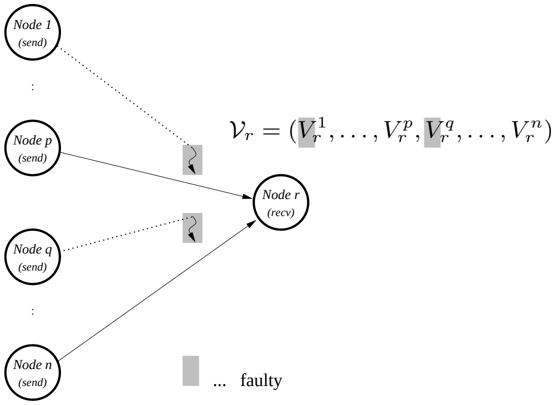


Figure 2. *Example of a receive fault that involves the messages from two senders.*

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, see Figure 3.

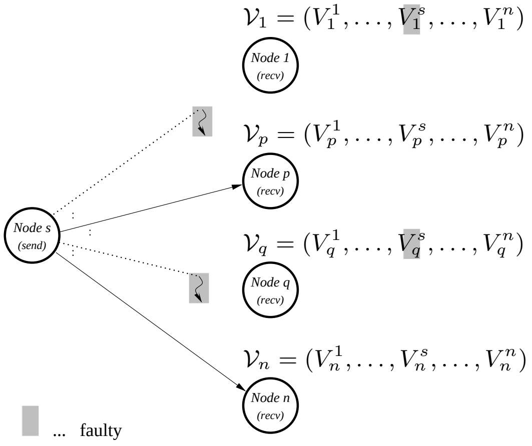

$\mathcal{V}_1 = (V_1^1, \ldots, V_1^s, \ldots, V_1^n)$
Node 1 (recv)
$\mathcal{V}_p = (V_p^1, \ldots, V_p^s, \ldots, V_p^n)$
Node p (recv)
Node s (send)
$\mathcal{V}_q = (V_q^1, \ldots, V_q^s, \ldots, V_q^n)$
Node q (recv)
$\mathcal{V}_n = (V_n^1, \ldots, V_n^s, \ldots, V_n^n)$
Node n (recv)
... faulty

Figure 3. *Example of a broadcast fault that affects the messages to two recipients.*

To align the present paper with the existing literature, the above perception-based model is recast into a simple modification of the original oral messages assumption of [14]. Node faults will be modeled according to a generalized version of the hybrid fault model of [15]. Our contribution here is to add the important class of *omission faults* to the already present manifest, symmetric, and arbitrary node faults. In doing so, those frequently encountered faults [5] need not be counted as arbitrary any more, which further decreases the required number of nodes n (but see Remark 2 on Theorem 2).

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.

- *A* manifest *faulty node p produces (detectably) missing messages or a received value that all non-faulty recipients q can detect as obviously bad; they all deliver the value $V_q^p = E$ in this case.*
- *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.*
- *A* symmetric *faulty node p sends the same wrong—but not usually detectably bad—value X^p to every non-faulty receiver. All receivers q deliver $V_q^p = X^p$—the value* "actually sent"*—in this case.*
- *An* arbitrary (asymmetric) *faulty node may inconsistently send any value to any non-faulty receiver.*

(A1^s) If a single non-faulty node p broadcasts (= successively sends) a message containing V^p to some set of non-faulty or omission faulty receiver nodes $\mathcal{R}$, at most f_ℓ^s of the delivered values $V_{q_i}^p$ may differ from V^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 non-faulty sender nodes send a message containing V^{p_i} to some non-faulty or omission faulty receiver node q, at most f_ℓ^r of the delivered values $V_q^{p_i}$ may differ from V^{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 lock-step 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. Our hybrid node faults can easily be mapped to the standard terminology [4]: A *clean crash* is equivalent to a manifest fault that reappears in every round after its first occurrence. A *crash* can be viewed as an omission fault that reappears as a clean crash in every round after its first occurrence. Finally, a *send/receive omission* [19] is equivalent to our omission fault.

3. Faulty nodes must not change their fault mode, i.e., must be counted in f_a, f_s, f_o or f_m according to their most severe behavior. A node that behaves symmetric faulty in one round and omission or manifest faulty in another should be considered arbitrary faulty.

4. A sender node that suffers from link faults according to (A1^s) is said to commit a *broadcast fault*, recall Figure 3, whereas a receiver node that experiences link faults according to (A1^r) is said to commit a *receive fault*, recall Figure 2. 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. Note that our model assumes links that are (virtually) made up of a pair of unidirectional channels, which can be hit by faults independently.

5. 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

$$f_\ell^s \leq f_\ell^r \quad \text{and} \quad f_\ell^{sa} \leq f_\ell^{ra}, \qquad (2)$$

since at most $n \cdot f_\ell^s$ messages may be faulty systemwide in any round's broadcasts according to (A1^s). By (A1^r), they must be spread over all message receptions in a way that no node experiences more than f_ℓ^r faulty messages. This is only possible if $n \cdot f_\ell^r \geq n \cdot f_\ell^s$, because otherwise there would be at least one message reception with more than f_ℓ^r faulty messages. Note that (2) implies $f_\ell^r = 0 \Rightarrow f_\ell^s = 0$.

   It also follows that, in case of $f_\ell^r > f_\ell^s$, only at most $\lfloor n f_\ell^s / f_\ell^r \rfloor$ nodes can experience a message reception that exhausts its full budget f_ℓ^r of link faults. This in turn shows that $f_\ell^r = f_\ell^s$ is in fact the optimal choice, although $f_\ell^r > f_\ell^s$ also makes sense if an accumulation of more than f_ℓ^s receive faults must be tolerated at certain nodes e.g. for safety reasons. Note carefully, however, that we will treat f_ℓ^r and f_ℓ^s as independent in the subsequent analysis (unless otherwise specified).The analysis of our algorithms will confirm that this assumption makes sense, see Remark 4 on Theorem 2.

6. In (A1^s) and (A1^r), we assume that link faults hit only messages from non-faulty senders/receivers. In practice, however, it is more realistic to consider node and link faults as being completely independent of each other. It could then happen that a link fault hits a message from a faulty node, which probably ends up in a node fault of increased severity.

   Fortunately, it can be shown[7] that the results obtained under the model of Definition 1 are valid for the more general independent model as well. This is particularly easy in the usual case where both the implementation of an algorithm and its analysis results do not depend upon the values of the individual model parameters $f_a, f_s, \ldots$ and $f_\ell^s, f_\ell^{sa}, \ldots$, but rather on a suitable (weighted) sum of those. For example, the algorithm of Section 3 is completely independent of those parameters and requires $n > 3f_a + 2f_s + 2f_o + f_m + 2f_\ell^s + f_\ell^r + f_\ell^{ra}$ nodes for guaranteeing the agreement and validity property for Byzantine agreement. Since all changes to the individual parameter settings caused by collapsing e.g. a symmetric node fault and a separate link fault into an arbitrary node fault ($f_s \rightarrow f_s - 1, f_\ell^r \rightarrow f_\ell^r - 1, f_a \rightarrow f_a + 1$) preserve validity of such expressions, Theorem 2 holds under the general independent model as well.

7. Since the consequences of an incomplete communication graph can be viewed as link omission faults [32], any analysis under our model provides results that are valid for partially connected networks as well.

3 The Hybrid Oral Messages Algorithm

In this section, we will show that the *Hybrid Oral Messages* algorithm (OMH) derived from [14] in [15] 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*. Eventually, every non-faulty receiver p must *deliver* a value v_p ascribed to the transmitter that satisfies the agreement and validity properties (B1) and (B2), as specified below. 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, $R(E)$, $R(R(E))$, $R(R(R(E)))$, $\ldots$ must actually be distinguishable here; for each legitimate value v, we can allow $R(v) = R^{-1}(v) = v$.

[7] Consult [30] for a direct proof and its history, which consisted of a sequence of manual and formal verification efforts.

- 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. Note that this particular default value is required for securing validity in presence of an omission faulty transmitter, see the proof of Lemma 1.

Consult [15] for a detailed discussion of the above primitives and the operation of OMH in general.

Definition 2 (Algorithm OMH [15]) *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[8] *receivers [including itself].*
3. *For every p and q, 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 applies R^{-1} to that value. The result is delivered as the transmitter's value v_p.*

In the above description of OMH, we did not explicitly address the question of how to uniquely assign received messages to the particular recursive instances of OMH they belong to. The uniqe *id* of the transmitter node is appended to each message for this purpose, which in fact produces a string of ids that uniquely determines the particular recursive instance. Note carefully that this string must be reconstructed upon reception of an E value and included in the $R(E)$-message prior to submitting it to further recursive instances.

By adopting and extending the analysis of [15] for our perception-based fault model, we will show that OMH 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.

Following the line of reasoning in [15], we start with the validity property 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 (non-faulty) 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

$$n' \geq 2f_\ell^s + f_\ell^r + f_\ell^{ra} + 2(f_a + f_s) + f_o + f_m + m, \quad (3)$$

with $f_m' \leq f_m$ manifest faulty ones among those, there must be at least $n' - f_\ell^s - f_a - f_s - f_m'$ non-faulty or omission faulty 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

$$n_q' = n' - f_\ell^s - f_a - f_s - f_m' - f_o' - f_\ell^{ra\prime} - f_\ell^{ro\prime}, \quad (4)$$

[8] 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.

where $f_\ell^{ro\prime}$ 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 $f_o' \leq f_o$ is the number of omission faulty nodes that actually caused an omission at node q. Note carefully that (4) is also valid for the transmitter ($q = p$), which must be non-faulty if at all considered here and must hence have "sent" itself the correct value $R(\nu)$. Since we assumed exactly f_m' manifest faulty receivers, our receiver q gets a minimum of $f_o' + f_m' + f_\ell^{ro\prime}$ values equal to E, hence at most $n_q'' = n' - f_o' - f_m' - f_\ell^{ro\prime}$ values different from E.

Recalling (4), we thus find

$$\begin{aligned} 2n_q' - n_q'' &= n' - 2f_\ell^s - 2f_a - 2f_s - f_o' - f_m' \\ &\qquad -2f_\ell^{ra\prime} - f_\ell^{ro\prime} \\ &\geq f_\ell^r + f_\ell^{ra} - 2f_\ell^{ra\prime} - f_\ell^{ro\prime} \\ &\qquad +(f_o - f_o') + (f_m - f_m') + m \\ &> 0 \end{aligned} \tag{5}$$

since $m = 1$ and both $f_\ell^{ro\prime} + f_\ell^{ra\prime} \leq f_\ell^r$ 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 (4) 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 (5) 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 $n' - f_\ell^s - f_a - f_s - f_m'$ non-faulty or omission faulty 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

$$\overline{n}_q' = n' - f_\ell^s - f_a - f_s - f_o' - f_m'$$

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, $f_o' \leq f_o$ gives the number of E-values delivered to receiver q by OMH($m-1$) when p was omission faulty. Since $m > 1$,

$$\begin{aligned} 2\overline{n}_q' - \overline{n}_q'' &= n' - 2f_\ell^s - 2f_a - 2f_s - f_o' - f_m' \\ &\geq f_\ell^r + f_\ell^{ra} + (f_o - f_o') + (f_m - f_m') + m \\ &> 0 \end{aligned} \tag{6}$$

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. □

With the help of Lemma 1, it is not too difficult to show the major Theorem 2.

Theorem 2 (Agreement and Validity) *For any $m \geq f_a + f_o + \min\{1, f_\ell^s\}$ and any f_a, f_o, f_s, f_m, f_ℓ^s, f_ℓ^r, f_ℓ^{ra}, the algorithm OMH(m) satisfies agreement (B1) and validity (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 and is an extension of the one of [15]. In the base case $m = \min\{1, f_\ell^s\}$, we must have $f_a = f_o = 0$ since $m \geq f_a + f_o + \min\{1, f_\ell^s\}$ by assumption. Hence, the transmitter must not be arbitrary or omission faulty here, such that Lemma 1 already implies both (B2) and (B1).

We can therefore assume that our theorem is true for OMH($m-1$) with $m - 1 \geq \min\{1, f_\ell^s\}$, and prove it for OMH($m$). Again, it suffices to consider the case where the transmitter is arbitrary or omission faulty, since Lemma 1 implies both (B2) and (B1) in the other cases. Since we have at most $f_a + f_o$ arbitrary or omission faulty nodes and the transmitter is one of those, either (1) at most $f_a - 1$ arbitrary faulty nodes or (2) at most $f_o - 1$ omission faulty ones remain among the strictly more than $2f_\ell^s + f_\ell^r + f_\ell^{ra} + 2(f_a + f_s) + f_o + f_m + m - 1$ receivers. Since obviously

$$\begin{aligned} &2f_\ell^s + f_\ell^r + f_\ell^{ra} + 2(f_a + f_s) + f_o + f_m + m - 1 \;> \\ &\qquad 2f_\ell^s + f_\ell^r + f_\ell^{ra} + 2([f_a - 1] + f_s) + f_o + f_m + [m - 1] \end{aligned}$$

as well as

$$\begin{aligned} &2f_\ell^s + f_\ell^r + f_\ell^{ra} + 2(f_a + f_s) + f_o + f_m + m - 1 \;> \\ &\qquad 2f_\ell^s + f_\ell^r + f_\ell^{ra} + 2(f_a + f_s) + [f_o - 1] + f_m + [m - 1], \end{aligned}$$

we can apply the induction hypothesis to conclude that any OMH($m-1$) satisfies (B1) and (B2). Hence, for any q, any two non-faulty receivers deliver the same value for w_p^q in step (3). Note carefully that this follows from (B2) if one of the two receivers is node q, and from (B1) otherwise. Hence, any two non-faulty receivers get the same vector of values and hence the same hybrid-majority, thereby proving (B1). □

Remarks:

1. It would be possible—and even makes perfect sense—to extend both properties (B2) and (B1) to apply to

omission faulty nodes p and q as well. In fact, since all that a omission faulty node could do is to omit sending the correct value to some recipients, it must participate in the algorithm like a non-faulty node in order to know the correct value. In the original version of this paper, we conjectured that our proofs remain valid for these extended validity and agreement properties if omission faulty nodes always send the correct value to itself, i.e., do not commit an omission in this internal "send". However, formal verification in [23, 30] showed that this is not true if OMH is used in its present form.

2. It is apparent from Theorem 2 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 [19] 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. If the latter were disallowed, however, i.e., if $f_a = f_s = f_\ell^{ra} = 0$, it should be possible to show—by a modified analysis—that OMH has optimal resilience. Note that this is a similar situation as encountered in [15, Thm. 2] for manifest and symmetric faults.

3. For OMH, receive faults (A1^r) resulting in an omission are easier to tolerate than those that produce a value fault, cf. Theorem 2, since $f_\ell^r + f_\ell^{ra} = f_\ell^{ro} + 2f_\ell^{ra}$ with f_ℓ^{ro} bounding the number of "pure" omission faults.

   Interestingly, this is not the case for broadcast faults (A1^s) here.

4. From the proof of Lemma 1, it is apparent that (A1^r) 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_ℓ^r according to (A1^r) is only required in the last round, where in turn (A1^s) is not explicitly used. This supports our approach of considering both types of faults independently from each other, recall Remark 5 on Definition 1.

4 Authentication

Consensus with written messages [14] assumes that no node can make undetectable modifications to messages and that the originator of a message is always known. It is generally agreed that electronic signatures can be used to achieve these goals, although there are some pitfalls [11]. The assumptions placed on the authentication scheme are:

(SA1) A node cannot change the contents of a message.

(SA2) A node cannot forge a signature.

(SA3) A valid signature cannot be mistaken for an invalid one (i.e., the signature does not introduce new errors).

In addition, we must also ensure that a node can detect whether a message belongs to the current execution run to avoid replay attacks.

Generally, every node p uses its signature σ_p to sign a message v, thereby generating the signed message $\sigma_p(v)$. The value v of the transmitter is consistently disseminated to all remaining nodes in the system by forwarding messages via paths of distinct processors in every round, such that a value v that is sent from node p_1 along the chain of nodes $p_2 \dots p_k$ arrives as a message $M = \sigma_{p_k} \cdots \sigma_{p_1}(v)$. This allows a node to recognize several manifest faults upon message reception:

(M1) If a message arrives in round $m-k$, then it must either bear $k+1$ signatures, the first of which is from the original transmitter, or it must contain the value E. All other messages are manifest faulty.

(M2) The message arrives on the link from node p_i but has not been signed by p_i last.

(M3) The message contains a signature at least twice (i.e., one node has signed the message twice).

(M4) Two messages bear the same signature chain and contain different values.

Of course, for (M1), all nodes must know who the initial transmitter is. The reaction of a node to these manifest faults depends on its own fault status. We assume the following:

- A non-faulty or omission faulty node recognizes (M1)-(M4) and discards the message received in (M1)-(M3), reporting E instead. In (M4), since the nodes wait until they receive all messages from one round of the algorithm before sending these values in the next round, the node will discard both messages and report E instead.

- A *manifest* faulty node produces a manifest fault at all receivers regardless of what it receives. However, in case of broken signatures we must assume that the node does not send two messages in (M4) in order to secure Lemma 6.

- A *symmetric* faulty node may ignore (M1)-(M3) and send the manifest faulty messages. If the signatures are secure, then it may also ignore (M4) and send both messages it has received. However, if we assume that signatures are broken, then we again must assume that a symmetric faulty node recognizes (M4) and does not send two different messages along, cf. Lemma 6.

- A *arbitrary* faulty node ignores (M1)-(M4) and sends along whatever it likes. In particular, it may send a message it should have recognized as manifest faulty.

Using signatures has a beneficial effect on node faults, because faulty nodes cannot introduce new values into the system, as will be proved in Lemma 6 in Section 8.2. A faulty node that relays a message can only choose not to relay it at all, or to report that it has experienced a manifest fault.

How does authentication affect link faults? Obviously, a link fault can either produce a detectable fault or replace the original message with some other valid message sent by the same node. In case of a non-arbitrary faulty node, replacing the message has no effect since all valid messages are the same, and in case of an arbitrary faulty node, the value sent is not important anyway. So authentication does have a positive effect on the severity of link faults as well, since it prohibits value faults.

5 Algorithm OMHA

In this section, we will analyze a variant of the algorithm OMHA developed in [11] under the system model of Section 2. The original algorithm OMHA is the same as OMH except that every message sent in OMHA(m) with $m > 0$ must be signed.

As we have argued in the previous section, faulty nodes cannot generate new values, so the only values that do occur are those originally sent by the transmitter, E, and various $R(E)$'s. In the hybrid fault model of [15], the fact that a faulty node can inject an $R(E)$ value is enough to make the performance of OMHA no better than that of OMH. But how does authentication affect link faults in OMHA? The previous section tells us that link faults do not introduce any new values if authentication is used. However, the original version of OMHA does not sign messages in OMHA(0), and at this stage, a link fault could insert a bogus $R(E)$ value. Therefore, we must assume that messages are signed even in OMHA(0).

Lemma 2 (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, algorithm OMHA(m) satisfies the validity property if there are strictly more than $2f_\ell^s + f_\ell^r + 2(f_a + f_s) + f_o + f_m + m$ participating nodes.*

Proof: The proof is virtually the same as for OMH in Lemma 1. Our initial number of participating receivers is

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

and again a non-faulty receiver obtains at least

$$n_q' = n' - f_\ell^s - f_a - f_s - f_m' - f_o' - f_\ell^{ra\prime} - f_\ell^{ro\prime}$$

identical values $R(\nu)$. The only difference to OMH is that due to the signatures, both omission ($f_\ell^{ro\prime}$) and value ($f_\ell^{ra\prime}$) link faults are detectable and result in E values. Hence, a non-faulty receiver q can get at most $n_q'' = n' - f_o' - f_m' - f_\ell^{ro\prime} - f_\ell^{ra\prime}$ values different from E, so instead of equation (5) we get

$$\begin{aligned} 2n_q' - n_q'' &= n' - 2f_\ell^s - 2f_a - 2f_s - f_o' - f_m' - \\ &\quad -f_\ell^{ro\prime} - f_\ell^{ra\prime} \\ &\geq f_\ell^r - f_\ell^{ro\prime} - f_\ell^{ra\prime} + (f_o - f_o') + \\ &\quad +(f_m - f_m') + m \\ &> 0 \end{aligned} \tag{7}$$

since $m = 1$ and $f_\ell^{ro\prime} + f_\ell^{ra\prime} \leq f_\ell^r$, which again implies that $R(\nu)$ wins the hybrid-majority at any non-faulty receiver. □

Theorem 3 (Agreement and Validity) *For any $m \geq f_a + f_o + min\{1, f_\ell^s\}$, algorithm OMHA(m) satisfies agreement and validity if there are strictly more than $2f_\ell^s + f_\ell^r + 2(f_a + f_s) + f_o + f_m + m$ participating nodes.*

Proof: As one can see in the proof for Theorem 2, agreement follows directly from validity and the fact that we use enough rounds to ensure that in at least one round the transmitter is neither arbitrary nor omission faulty. Link faults are not considered here. Therefore, the proof for Theorem 2 is also valid for OMHA. □

Remarks:

1. Note that the proof for agreement only uses the validity property and the fact that f_a and f_o are added to the m required by validity. Hence, every consensus algorithm of this type that achieves validity for a given m will also achieve agreement for $m' = m + f_a + f_o$.

2. We already mentioned that the original OMHA in [11] avoided signing the messages sent by OMHA(0). Recall that (A2) in Definition 1 assumes a point-to-point network where the transmitter of a message can be uniquely identified. If a link fault could only cause an omission or a manifest fault, Theorem 3 would remain valid for the original algorithm as well. However, if a link fault can substitute an $R(E)$ value for the real message, then the original algorithm performs no better than OMH. Hence, by Theorem 2, we would need strictly more than $2f_\ell^s + f_\ell^r + f_\ell^{ra} + 2(f_a + f_s) + f_o + f_m + m$ nodes for this variant of OMHA.

Since OMHA just adds signatures to OMH, both algorithms send and receive the same messages. Therefore, the results of OMH's assumption coverage analysis in Section 7 (Theorem 5 and 6 as well as Tables 3–8) will also be valid for OMHA.

6 Algorithm ZA

In this section, we will analyze the authenticated algorithm ZA of [11] under our perception-based fault model.

The algorithm ZA has been derived from the flawed algorithm Z of [34] and provides a much better resilience than OMHA. However, its correctness depends critically upon Assumptions (SA1)–(SA3)—but see Section 8.2—and upon the fact that the transmitter must be known.

Definition 3 (Algorithm ZA [11]) *The algorithm ZA is defined recursively as follows (we assume that E is assigned whenever a message was not received or manifest faulty or incorrectly signed):*

ZA(0):

1. *The transmitter sends its value to every receiver.*
2. *Every receiver delivers the value obtained from the transmitter, or some fixed value E.*

ZA(m), $m > 0$:

1. *The transmitter signs and sends its value to every receiver.*
2. *For every node p, let w_p be the value p has obtained from the transmitter, or E. Every receiver p acts as the transmitter in algorithm ZA($m-1$) to send the value w_p to the $n-1$ receivers [including itself].*
3. *For every node p and q, let w_p^q be the value p has obtained from receiver q in step (2) using algorithm ZA($m-1$), or E if no such value of a manifest faulty one was delivered. Every receiver p calculates the majority value among all non-E values w_p^q it has received; if no non-E value exists, E is delivered, if no majority exists, some arbitrary but fixed value is used.*

Note that the strength of the signed algorithm lies in the fact that the only values that can occur are the values sent by the original transmitter and E. So if the transmitter is not arbitrary faulty, then every node can only receive the original value and E.

Lemma 3 (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, algorithm ZA(m) satisfies the validity property if there are strictly more than $f_\ell^s + f_\ell^r + f_a + f_s + f_o + f_m + 1$ participating nodes.*

Proof: Let us assume a not arbitrary and not omission faulty transmitter that sends the value ν. Then we only have to show that every good receiver obtains at least one ν in the first $min\{1, f_\ell^s\} + 1$ rounds, because once it has obtained the value, it will also deliver it. Recall that any transmitter "sends" its value to itself in step 2 of ZA(m) as well.

If $f_\ell^s = 0$, then we allow $m = 0$. However, since the transmitter is non-faulty and the links are non-faulty as well (recall that $f_\ell^s = 0$ implies $f_\ell^r = 0$), every good receiver will obtain ν in ZA(0) and will deliver it.

Now let $m \geq 1$. In ZA(m), the transmitter signs and sends its value ν to all $n-1$ receivers, $n' \geq n - 1 - f_a - f_s - f_o - f_m$ of which are non-faulty. At least $n' - f_\ell^s$ of these will receive ν. In round $m-1$, the $n' - f_\ell^s$ non-faulty receivers will broadcast ν. So by the end of the second round, every receiver gets ν from at least $n' - f_\ell^s - f_\ell^r$ nodes and all we have to do is to ensure that the number of non-faulty nodes is $n' > f_\ell^s + f_\ell^r$, so the number of nodes must be $n > f_\ell^s + f_\ell^r + f_a + f_s + f_o + f_m + 1$. As soon as a node has obtained at least one ν, it will deliver it, so for all $m \geq 1$, any non-faulty receiver will deliver ν.

Now assume that the transmitter is omission faulty. Let n_v^i be the number of non-faulty nodes which receive ν in ZA($m-i$), and n_E^i the number of non-faulty nodes which receive E in ZA($m-i$). If the transmitter is omission faulty, then only some non-faulty nodes n_v^0 will initially get the value ν, and the other non-faulty nodes n_E^0 will get E. If $n_v^0 > f_\ell^r$, then every non-faulty node will receive at least $n_v^0 - f_\ell^r > 0$ values ν in the next round. If, however, $n_v^0 \leq f_\ell^r$, then we have to distinguish two cases:

If $n_E^0 \leq f_\ell^s$, then the nodes in n_E^0 may never get ν, regardless of the number of rounds we spend. For $n_E^0 > f_\ell^s$, however, in the next round all good nodes except at most f_ℓ^s ones will have received ν, i.e., $n_v^1 \geq n - 1 - f_a - f_s - f_o - f_m - f_\ell^s$. If we require $n_v^1 > f_\ell^r$, i.e., $n > f_\ell^s + f_\ell^r + f_a + f_s + f_o + f_m + 1$, then we can again ensure that all good nodes will receive ν in the second round. In any case, every good node will either receive and deliver ν, or it will not receive anything and therefore deliver E. □

Theorem 4 (Agreement and Validity) *For any $m \geq f_a + f_o + min\{1, f_\ell^s\}$, algorithm ZA($m$) satisfies agreement and validity if there are strictly more than $f_\ell^s + f_\ell^r + f_a + f_s + f_o + f_m + 1$ participating nodes.*

Proof: As argued in Remark 1 on Theorem 3, the proof is the same as that for agreement in OMHA. □

Remarks:

1. We have changed the definition of ZA so that every receiver relays the message to all other receivers including itself in step 2 of ZA(m), since it needs its own value in step 3. In the original paper [11], the message was only relayed to the other $n-2$ receivers.

2. Since ZA does not distinguish between E and $R(E)$ as OMHA does, using signatures means that the only possible values a node ever encounters are those originally sent by the transmitter and E values. Link faults are also recognized as manifest faults in all subsequent stages of the algorithm. Contrary to OMHA, where the algorithm benefits from signing messages in OMHA(0), link faults can only insert the values E or a valid signed message ν in ZA(0), so we do not require signatures in this last stage if the message has been signed at least once (hence, we require $m \geq min\{1, f_\ell^s\}$).

3. In case of validity with an omission faulty transmitter, we cannot guarantee that every non-faulty node

delivers ν if the transmitter has sent at least one ν. However, if we require that $m \geq 2$, then we can at least ensure that as many nodes as possible deliver ν.

4. Although ZA is defined recursively like OMHA, its execution develops quite differently: ZA(m) achieves validity after the first two (or three, see the previous remark) rounds regardless of the number of rounds m actually employed. Validity of OMHA(m), however, is achieved only after its full number $m+1$ of rounds. Moreover, m needs to be included into n for OMHA, since the faulty nodes can inject $R(E)$'s in the additional rounds that must be balanced. This is not true for ZA, since the latter deals with ν and E only.

Like OMHA, ZA also sends and receives the same messages as OMH. The results of OMH's assumption coverage analysis in Section 7, namely, Theorem 5 and 6, will hence remain valid for ZA as well. Note carefully, however, that the numerical results in Tables 3–8 will not apply since they assume $n = 4f_\ell + 3m + 1$ and not ZA's minimum setting $n = 2f_\ell + m + 1$.

7 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. Note carefully that this is a mandatory step for any algorithm where safety[9] 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 our algorithms send 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 this section, we will derive an upper bound on the probability of failure Q_m of OMH(m)—valid for OMHA and ZA as well, since they employ the same communications pattern—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. Despite of its simplicity, this model is commonly used in practice, see e.g. [10,18], 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.

Since $f_\ell^s = f_\ell^r = f_\ell$ is the only reasonable choice in presence of independent link faults, recall Remark 5 on Definition 1, the success probabilities for a single message broadcast/reception, namely,

$$\begin{aligned} p^s_{n-k} &= \textbf{Prob}\{\leq f_\ell^s \text{ faults in a single broadcast to } \\ &\qquad\qquad n-k \text{ receivers}\} \\ p^r_{n-k} &= \textbf{Prob}\{\leq f_\ell^r \text{ faults in a single reception from } \\ &\qquad\qquad n-k \text{ senders}\} \end{aligned}$$

for $0 \leq k \leq n-1$ are the same $p^s_{n-k} = p^r_{n-k} = p_{n-k}$ and follow a binomial distribution:

$$p_{n-k} = \sum_{l=0}^{f_\ell} \binom{n-k}{l} p^l (1-p)^{n-k-l}$$

The total *probability of success* $P_m = 1 - Q_m$ that there is no violation of our assumption of at most f_ℓ link faults in any message broadcast/reception during the execution of OMH(m) is given by

$$\begin{aligned} P_m \;=\; & \textbf{Prob}\{\text{All broadcasts in OMH}(m),\ldots,\text{OMH}(1) \\ & \text{have} \leq f_\ell \text{ link faults each} \wedge \text{all receptions} \\ & \text{in OMH(0) have} \leq f_\ell \text{ link faults each}\}. \qquad (8) \end{aligned}$$

Recall from the proof of Lemma 1 that (A1^r) in Definition 1 is required in the base case of the induction only, i.e., in OMH(0).

It is immediately apparent from step 1 of Definition 2 that the execution of OMH(m) evolves as shown in Table 1.

OMH(m)	**# instances**	**# receivers**
m	1	$n-1$
$m-1$	$n-1$	$n-2$
$m-2$	$(n-1)(n-2)$	$n-3$
$\vdots$	$\vdots$	$\vdots$
1	$(n-1)\cdots(n-m+1)$	$n-m$
0	$(n-1)(n-2)\cdots(n-m)$	$n-m-1$

Table 1. *Recursive instances in the execution of algorithm OMH.*

With the notation

$$\begin{aligned} [n]_k &= n(n-1)\ldots(n-k+1) \qquad \text{for } k > 0 \\ [n]_0 &= 1 \end{aligned}$$

it is apparent that, for $k < m$, there are $[n-1]_k$ instances of OMH($m-k$) that each issue a single broadcast [where (A1^s) applies] to $n-k-1$ receivers. For $k = m$, on the other hand, we have to consider message receptions [where (A1^r) applies] only: There are $[n-1]_m$ instances of OMH(0), and every receiver of a particular instance of OMH(0) should receive a message from all $n-m$ recipients in the prior instance of OMH(1). Not counting the

[9] We note that this is not true for certain randomized algorithms, which violate only liveness but not safety in this case.

"self-reception" by the transmitter of OMH(0), there remain $n-m-1$ "true" message receptions by any receiver of OMH(0).

Abbreviating $n_k = [n-1]_k$, (8) translates to

$$\begin{aligned} P_m &= \prod_{k=0}^{m-1} p_{n-k-1}^{n_k} \cdot p_{n-m-1}^{n_m} = \prod_{k=0}^{m} p_{n-k-1}^{n_k} \\ &= \prod_{k=0}^{m} (1-q_{n-k-1})^{n_k} \quad \text{with } q_{n-k} = 1-p_{n-k} \\ &= \prod_{k=0}^{m} \Big(1 - \frac{n_k q_{n-k-1}}{n_k}\Big)^{n_k} \\ &\geq e^{-\sum_{k=0}^{m} n_k q_{n-k-1}} \prod_{k=0}^{m} \Big(1 - \frac{\big(n_k q_{n-k-1}\big)^2}{n_k}\Big), \end{aligned} \tag{9}$$

where we used the relation [38, p. 242]

$$e^{-t} \geq (1-t/n)^n \geq e^{-t}(1-t^2/n) \tag{10}$$

valid for $t < n$; since q_{n-k-1} is some probability < 1, this condition is of course satisfied. Assuming

$$\sum_{k=0}^{m} n_k q_{n-k-1} < 1, \tag{11}$$

the application of the well-known facts (1) $\log(1-x) = -\sum_{j\geq 1} x^j/j$ for $|x| < 1$, (2) $\sum_{i\in I} a_i^j \leq \big(\sum_{i\in I} a_i\big)^j$ for $a_i \geq 0$ and integer $j \geq 1$, and (3) $e^{-x} \geq 1-x$ for $0 \leq x < 1$ yields

$$\begin{aligned} P_m &\geq e^{-\sum_{k=0}^{m} n_k q_{n-k-1} + \sum_{k=0}^{m} \log\Big(1 - \frac{\big(n_k q_{n-k-1}\big)^2}{n_k}\Big)} \\ &\geq e^{-\sum_{k=0}^{m} n_k q_{n-k-1} - \sum_{k=0}^{m}\sum_{j\geq 1} \frac{(\sqrt{n_k} q_{n-k-1})^{2j}}{j}} \\ &\geq e^{-\sum_{k=0}^{m} n_k q_{n-k-1} - \sum_{j\geq 1} \frac{\big(\sum_{k=0}^{m} \sqrt{n_k} q_{n-k-1}\big)^{2j}}{j}} \\ &\geq \Big(1 - \sum_{k=0}^{m} n_k q_{n-k-1}\Big)\Big(1 - \big(\sum_{k=0}^{m} \sqrt{n_k} q_{n-k-1}\big)^2\Big) \\ &\geq 1 - \sum_{k=0}^{m} n_k q_{n-k-1} - \Big(\sum_{k=0}^{m} \sqrt{n_k} q_{n-k-1}\Big)^2 \qquad (12) \\ &\geq 1 - \sum_{k=0}^{m} n_k q_{n-k-1} - \Big(\sum_{k=0}^{m} n_k q_{n-k-1}\Big)^2. \qquad (13) \end{aligned}$$

To obtain an upper bound on the overall probability of failure $Q_m = 1 - P_m$, we hence need an upper bound on

$$\sum_{k=0}^{m} [n-1]_k q_{n-k-1} = (n-1)! \sum_{k=0}^{m} \frac{q_{n-k-1}}{(n-k-1)!} \tag{14}$$

and, if the more accurate lower bound (12) is addressed,

$$\sum_{k=0}^{m} \sqrt{[n-1]_k} q_{n-k-1} = \sqrt{(n-1)!} \sum_{k=0}^{m} \frac{q_{n-k-1}}{\sqrt{(n-k-1)!}}. \tag{15}$$

The required bound for the dominating term (14) follows from the following Lemma 4.

Lemma 4 (Upper Bound) *For* $n - m - f_\ell - 2 \geq 1$,

$$\begin{aligned} G_m &= \sum_{k=0}^{m} \frac{q_{n-k-1}}{(n-k-1)!} \\ &\leq \Big(1 + \frac{1}{n-m-f_\ell-2}\Big) \cdot \frac{q_{n-m-1}}{(n-m-1)!} \qquad (16) \\ &\leq \Big(1 + \frac{1}{n-m-f_\ell-2}\Big) \cdot \\ &\qquad \frac{1}{(n-m-f_\ell-2)!} \cdot \frac{p^{f_\ell+1}}{(f_\ell+1)!}. \qquad (17) \end{aligned}$$

Proof: According to [1, Eq. 26.5.24], q_{n-k} equals the incomplete Beta function $I_p(f_\ell+1, n-k-f_\ell)$, i.e.,

$$\begin{aligned} q_{n-k} &= \sum_{l=f_\ell+1}^{n-k} \binom{n-k}{l} p^l (1-p)^{n-k-l} \qquad (18) \\ &= \frac{(n-k)!}{(f_\ell)!\,(n-k-f_\ell-1)!} \cdot \\ &\qquad \int_0^p t^{f_\ell} (1-t)^{n-k-f_\ell-1}\, dt. \qquad (19) \end{aligned}$$

Hence,

$$G_m = \frac{1}{(f_\ell)!} \int_0^p t^{f_\ell} \sum_{k=0}^{m} \frac{(1-t)^{n-k-f_\ell-2}}{(n-k-f_\ell-2)!}\, dt, \tag{20}$$

which involves

$$\begin{aligned} S &= \sum_{k=0}^{m} \frac{(1-t)^{n-k-f_\ell-2}}{(n-k-f_\ell-2)!} \\ &= \frac{(1-t)^{n-m-f_\ell-2}}{(n-m-f_\ell-2)!}\Big(1 + \frac{1-t}{n-m-f_\ell-1} + \\ &\qquad + \cdots + \frac{(1-t)^m}{(n-m-f_\ell-1)\cdots(n-f_\ell-2)}\Big) \\ &\leq \frac{(1-t)^{n-m-f_\ell-2}}{(n-m-f_\ell-2)!} \sum_{j=0}^{m} \Big(\frac{1-t}{n-m-f_\ell-1}\Big)^j \\ &\leq \frac{(1-t)^{n-m-f_\ell-2}}{(n-m-f_\ell-2)!} \cdot \frac{1}{1 - \frac{1-t}{n-m-f_\ell-1}} \\ &\leq \frac{(1-t)^{n-m-f_\ell-2}}{(n-m-f_\ell-2)!} \cdot \frac{n-m-f_\ell-1}{n-m-f_\ell-2+t} \\ &\leq \frac{(1-t)^{n-m-f_\ell-2}}{(n-m-f_\ell-2)!} \cdot \Big(1 + \frac{1}{n-m-f_\ell-2}\Big) \end{aligned}$$

since $0 \le t \le p$. Plugging the above expression into (20), we obtain

$$G_m \le \frac{1+\frac{1}{n-m-f_\ell-2}}{(f_\ell)!\,(n-m-f_\ell-2)!}\int_0^p t^{f_\ell}(1-t)^{n-m-f_\ell-2}\,dt \tag{21}$$

from where the major result (16) of our theorem follows easily by recalling (19).

To finally establish (17), we use the definition (18) of q_{n-k-1} to find

$$\begin{aligned}
g &= \frac{q_{n-m-1}}{(n-m-1)!} \\
&= \sum_{l=f_\ell+1}^{n-m-1} \frac{p^l}{l!}\cdot\frac{(1-p)^{n-m-l-1}}{(n-m-l-1)!} \qquad (22)\\
&= \sum_{j=0}^{n-m-f_\ell-2} \frac{p^{j+f_\ell+1}}{(j+f_\ell+1)!}\cdot\frac{(1-p)^{n-m-f_\ell-2-j}}{(n-m-f_\ell-2-j)!} \\
&\le \frac{p^{f_\ell+1}}{(f_\ell+1)!}\sum_{j=0}^{n-m-f_\ell-2}\frac{p^j}{j!}\cdot\frac{(1-p)^{n-m-f_\ell-2-j}}{(n-m-f_\ell-2-j)!}.
\end{aligned}$$

Applying the binomial theorem $(p+1-p)^{n-m-f_\ell-2}=1$, we finally get

$$\frac{q_{n-m-1}}{(n-m-1)!} \le \frac{1}{(n-m-f_\ell-2)!}\cdot\frac{p^{f_\ell+1}}{(f_\ell+1)!} \tag{23}$$

which completes the proof of our lemma. □

Remarks:

1. Lemma 4 reveals that the sum (14) is dominated by the term $k=m$, which just reflects the intuitively clear fact that the many messages from OMH(0) in the last round determine OMH(m)'s overall probability of failure.

2. By subtracting $q_{n-m-1}/(n-m-1)!$ from both sides of (16), and multiplying by $(n-1)!$ according to (14), it is easy to see that (16) also implies monotonicity of

$$\begin{aligned}
\frac{q_{n-k-1}}{(n-k-1)!} &\le \frac{q_{n-m-1}}{(n-m-1)!} \qquad (24)\\
[n-1]_k q_{n-k-1} &\le [n-1]_m q_{n-m-1} \qquad (25)
\end{aligned}$$

   for any $0 \le k \le m$.

3. The bound given by (17) is reasonably small—and also accurate, cp. the derivation starting with (22)—only if $np < 1$ is sufficiently small, since the ultimately required quantity $(n-1)!G_m$ that must be < 1 according to (11) has order $\mathcal{O}\big(n^m(np)^{f_\ell+1}/(f_\ell+1)!\big)$.

By a very similar proof, it is not difficult to show a similar Lemma 5 related to the square-rooted sum (15). Since it is only used to improve the remainder $\mathcal{O}$-term in Theorem 5 below, its proof will be left as an exercise to the reader.

Lemma 5 (Upper Bound $\sqrt{\ }$) *For $n-m-f_\ell-2 \ge 1$,*

$$\begin{aligned}
H_m &= \sum_{k=0}^{m}\frac{q_{n-k-1}}{\sqrt{(n-k-1)!}} \\
&\le \Big(1+\frac{1}{\sqrt{n-m-f_\ell-2}}\Big)\cdot \\
&\qquad \sqrt{\frac{[n-1]_{f_\ell+1}}{[n-m-1]_{f_\ell+1}}}\cdot\frac{q_{n-m-1}}{\sqrt{(n-m-1)!}} \\
&\le \Big(1+\frac{1}{\sqrt{n-m-f_\ell-2}}\Big)\cdot \\
&\qquad \sqrt{\frac{[n-1]_{f_\ell+1}}{(n-m-f_\ell-2)!}}\cdot\frac{p^{f_\ell+1}}{(f_\ell+1)!}.
\end{aligned}$$

□

By virtue of those results, we can establish the following Theorem 5.

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

$$Q_m \le Q'_m + \mathcal{O}\Big(\frac{(Q'_m)^2}{[n-f_\ell-2]_m}\Big) = \mathcal{O}\Big(n^m\frac{(np)^{f_\ell+1}}{(f_\ell+1)!}\Big), \tag{26}$$

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: Recalling (14) resp. (15), the result of Lemma 4 resp. 5 immediately yields $(n-1)!G_m \le Q'_m$ resp.

$$\begin{aligned}
R''_m &= (n-1)!H_m^2 \\
&\le \Big(1+\frac{1}{\sqrt{n-m-f_\ell-2}}\Big)^2\cdot \\
&\qquad [n-1]_{f_\ell+1}\cdot[n-1]_{m+f_\ell+1}\cdot\Big(\frac{p^{f_\ell+1}}{(f_\ell+1)!}\Big)^2 \\
&\le \mathcal{O}\Big(\frac{(Q'_m)^2}{[n-f_\ell-2]_m}\Big), \qquad (27)
\end{aligned}$$

where the last bound is easily confirmed by comparing R''_m with $(Q'_m)^2$. Recalling the lower bound (12) on the probability of success, (26) is established by straightforward upper bounding. Note that (12) is only guaranteed to hold when (11) holds, which is secured by $np < 1$ sufficiently small according to Remark 3 on Lemma 4. □

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 (26), 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:

$$\begin{aligned}
Q_m &= \mathcal{O}\Big(n_0^m\frac{(n_0p)^{f_\ell+\ell+1}}{(f_\ell+\ell+1)!}\big(1+\frac{c\ell}{n_0}\big)^{m+f_\ell+\ell+1}\Big) \\
&= \mathcal{O}\Big(n_0^m\frac{(n_0p)^{f_\ell+1}}{(f_\ell+1)!}\cdot\frac{(n_0p\cdot e^c)^\ell}{[f_\ell+\ell+1]_\ell}\Big) \qquad (28)
\end{aligned}$$

In the last step, we employed the well-known relation $\big(1 + t/(k+j)\big)^k \leq e^t$ for $t \geq 0$ in conjunction with $k + j = n_0 \geq m + f_\ell + \ell + 1 = k$.

It is hence apparent from (28) that, as long as $np < 1$ sufficiently small, the probability of failure Q_m

- 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 (28) 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.

In Tables 3–6, we give 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 2.

f_ℓ	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
1	8	11	14	17	20	23
2	12	15	18	21	24	27
3	16	19	22	25	28	31
5	24	27	30	33	36	39
7	32	35	38	41	44	47
10	44	47	50	53	56	59
15	64	67	70	73	76	79
20	84	87	90	93	96	99

Table 2. *Value of $n = 4f_\ell + 3m + 1$ for different m, f_ℓ.*

f_ℓ	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
1	0.64	1.	1.	1.	1.	1.
2	0.59	1.	1.	1.	1.	1.
3	0.52	1.	1.	1.	1.	1.
5	0.36	1.	1.	1.	1.	1.
7	0.22	1.	1.	1.	1.	1.
10	0.095	1.0	1.	1.	1.	1.
15	0.019	0.86	1.	1.	1.	1.
20	0.0036	0.37	1.	1.	1.	1.

Table 3. *Value of (exact) probability of failure Q_m for $p = 0.1$.*

Whereas the probability of failure of OMH(m) given in Tables 3–6 is not bad, even in case of a typical "wireless" loss probability $p = 0.01$, it is nevertheless clear that an algorithm that uses less messages is preferable with respect to our fault model. As an example, we consider the algorithm $\overline{\text{OMH}}$ that results from combining all messages that a

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 4. *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	$1. 10^{-6}$	0.00003	0.0009	0.03	1	1
2	$2. 10^{-9}$	$4. 10^{-8}$	$2. 10^{-6}$	0.00007	0.004	0.2
3	$2. 10^{-12}$	$6. 10^{-11}$	$3. 10^{-9}$	$1. 10^{-7}$	$7. 10^{-6}$	0.0004
5	$2. 10^{-18}$	$9. 10^{-17}$	$5. 10^{-15}$	$3. 10^{-13}$	$2. 10^{-11}$	$2. 10^{-9}$
7	$2. 10^{-24}$	$1. 10^{-22}$	$9. 10^{-21}$	$7. 10^{-19}$	$6. 10^{-17}$	$5. 10^{-15}$
10	$2. 10^{-33}$	$2. 10^{-31}$	$2. 10^{-29}$	$2. 10^{-27}$	$2. 10^{-25}$	$2. 10^{-23}$
15	$2. 10^{-48}$	$2. 10^{-46}$	$3. 10^{-44}$	$4. 10^{-42}$	$5. 10^{-40}$	$8. 10^{-38}$
20	$2. 10^{-63}$	$2. 10^{-61}$	$4. 10^{-59}$	$7. 10^{-57}$	$1. 10^{-54}$	$2. 10^{-52}$

Table 5. *Value of (approximate) probability of failure Q'_m for $p = 0.0001$.*

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

node sends during OMH in a round into a single message. According to Table 1, such a combined message consists of exactly $[n-1]_k/(n-k) = [n-1]_{k-1}$ single messages—corresponding to the instances of OMH($m-k$) at any of the $n-k$ originating nodes—that are broadcast to $n-k-1$ receivers. Clearly, during the whole execution of $\overline{\text{OMH}}$(m), any node broadcasts only m messages, except for the initial transmitter, which broadcasts only one message.

It is not difficult to show that the proofs of correctness for OMH are also valid for $\overline{\text{OMH}}$. In fact, the only difference lies in the fact that the receivers in $\overline{\text{OMH}}$ experience a link fault in a correlated fashion: If f_ℓ^s of the combined messages are lost in the broadcast of a single sender, any affected receiver looses the round message for all instances of OMH($m-k$). This situation, however, could also occur when link faults are independent for all instances of OMH($m-k$).

By the same devices as used before, the probability of success $\overline{P}_m$ for $\overline{\text{OMH}}$(m) evaluates to

$$\begin{aligned}\overline{P}_m &= p_{n-1} \prod_{k=1}^{m} p_{n-k-1}^{n-k} \\ &\geq \prod_{k=0}^{m} \Big(1 - \frac{(n-k)q_{n-k-1}}{n-k}\Big)^{n-k}\end{aligned}$$

where the bound is even valid if all nodes (and not only the initial transmitter) send an initial message in $\overline{\text{OMH}}$(m). Due to that simplification, we just have to substitute $n_k = n-k$ in (13) and use the same line of reasoning as before to show the following Theorem 6.

Theorem 6 (Assumption Coverage $\overline{\text{OMH}}$) *For $n-m-f_\ell-2 \geq 1$ and $np < 1$ sufficiently small, the probability of failure $\overline{Q}_m$ of $\overline{\text{OMH}}$(m) satisfies*

$$\overline{Q}_m \leq \overline{Q}'_m + \mathcal{O}\big((\overline{Q}'_m)^2\big) = \mathcal{O}\Big(\frac{n}{f_\ell+3} \cdot \frac{(np)^{f_\ell+1}}{(f_\ell+1)!}\Big), \quad (29)$$

where

$$\overline{Q}'_m = \frac{[n+1]_{f_\ell+3} - [n-m]_{f_\ell+3}}{f_\ell+3} \cdot \frac{p^{f_\ell+1}}{(f_\ell+1)!}. \quad (30)$$

Proof: Applying (12) with $n_k = n-k$ reveals that $\overline{P}_m$ and hence the probability of failure $\overline{Q}_m$ is dominated by

$$\overline{Q}''_m = \sum_{k=0}^{m} (n-k)q_{n-k-1} = \sum_{k=0}^{m} (n-k)! \frac{q_{n-k-1}}{(n-k-1)!}, \quad (31)$$

Using the upper bound (23) with $m = k$ established in the proof of Lemma 4, we find

$$\begin{aligned}\overline{Q}''_m &\leq \sum_{k=0}^{m} \frac{(n-k)!}{(f_\ell+1)!\,(n-k-f_\ell-2)!} \cdot p^{f_\ell+1} \\ &\leq (f_\ell+2)p^{f_\ell+1} \sum_{k=0}^{m} \binom{n-k}{f_\ell+2} \\ &\leq (f_\ell+2)p^{f_\ell+1} \sum_{k=n-m}^{n} \binom{k}{f_\ell+2} \\ &\leq (f_\ell+2)\left[\binom{n+1}{f_\ell+3} - \binom{n-m}{f_\ell+3}\right]p^{f_\ell+1} \\ &\leq \frac{[n+1]_{f_\ell+3} - [n-m]_{f_\ell+3}}{f_\ell+3} \cdot \frac{p^{f_\ell+1}}{(f_\ell+1)!},\end{aligned}$$

where we employed the well-known identity [13, p.54.(11)] $\sum_{k=0}^{n} \binom{k}{m} = \binom{n+1}{m+1}$. Recalling (13), which is again valid for $np < 1$ sufficiently small, and applying some simple majorizations on (30) that consider the fact that the coefficient of $n^{f_\ell+3}$ in both $[n+1]_{f_\ell+3}$ and $[n-m]_{f_\ell+3}$ is 1 and hence cancels out, (29) follows. □

Comparing (26) and (29) clearly shows that the probability of failure $\overline{Q}_m$ no longer grows with m. Tables 7 and 8 contain a few numerical values for $\overline{Q}'_m$ for different values of m and f_ℓ and the same $n = 4f_\ell + 3m + 1$ used before, which ensures compatibility with Tables 3 and 4. We should note, however, that the messages sent by $\overline{\text{OMH}}$ are much longer than the ones of OMH—it is perhaps not really fair to consider the same values for the loss probability p here.

f_ℓ	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
1	0.88	1.	1.	1.	1.	1.
2	0.85	1.	1.	1.	1.	1.
3	0.80	0.99	1.	1.	1.	1.
5	0.62	0.93	1.	1.	1.	1.
7	0.42	0.77	0.96	1.	1.	1.
10	0.20	0.43	0.71	0.91	0.99	1.
15	0.041	0.10	0.21	0.37	0.58	0.78
20	0.0078	0.019	0.041	0.08	0.14	0.24

Table 7. *Value of (exact) probability of failure $\overline{Q}_m$ for $p = 0.1$.*

f_ℓ	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
1	0.04	0.1	0.4	0.9	1.	1.
2	0.004	0.02	0.04	0.1	0.2	0.4
3	0.0004	0.002	0.004	0.01	0.02	0.04
5	$5.\,10^{-6}$	0.00002	0.00005	0.0001	0.0002	0.0005
7	$5.\,10^{-8}$	$2.\,10^{-7}$	$5.\,10^{-7}$	$1.\,10^{-6}$	$3.\,10^{-6}$	$5.\,10^{-6}$
10	$5.\,10^{-11}$	$2.\,10^{-10}$	$4.\,10^{-10}$	$1.\,10^{-9}$	$3.\,10^{-9}$	$6.\,10^{-9}$
15	$4.\,10^{-16}$	$1.\,10^{-15}$	$4.\,10^{-15}$	$1.\,10^{-14}$	$2.\,10^{-14}$	$5.\,10^{-14}$
20	$4.\,10^{-21}$	$1.\,10^{-20}$	$3.\,10^{-20}$	$9.\,10^{-20}$	$2.\,10^{-19}$	$5.\,10^{-19}$

Table 8. *Value of (approximate) probability of failure $\overline{Q}'_m$ for $p = 0.01$.*

8 Discussion of Results

In this section, we will discuss some consequences of the results of the previous sections.

8.1 Costs of Tolerating Link Faults

The results of Theorems 2, 3 and 4 allow us to compare the costs for tolerating link faults. Table 9 summarizes the relevant figures for $f_\ell = f_\ell^r = f_\ell$: The second resp. third column gives the number of nodes required for tolerating f_ℓ omission resp. f_ℓ arbitrary link faults; n_0 denotes the number of nodes required for $f_\ell = 0$, where only node faults are present. The last column gives the number of rounds for either type of link fault; m_0 is the number of rounds for $f_\ell = 0$.

Algorithm	**omissions**	**arbitrary**	**# rounds**
OMH	$n_0 + 3f_\ell$	$n_0 + 4f_\ell$	$m_0 + 1$
OMHA	$n_0 + 3f_\ell$	$n_0 + 3f_\ell$	$m_0 + 1$
ZA	$n_0 + 2f_\ell$	$n_0 + 2f_\ell$	$m_0 + 1$

Table 9. *Additional costs of tolerating $f_\ell^r = f_\ell^s = f_\ell$ omission resp. arbitrary link faults in terms of number of nodes and number of rounds.*

The best algorithm, ZA, needs only $2f_\ell$ additional nodes (and one additional round) to cope with $f_\ell \cdot n$ link faults per node in each round; for $f_\ell = 1$, only $n = 4$ nodes are required in the absence of node faults.

Since f_ℓ could be as much as $\mathcal{O}(n)$, each of our algorithms can cope with an impressive number of $\mathcal{O}\big((m + 1)n^2\big)$ link faults during the whole execution. It is important to note, though, that this does not mean that those algorithms are resilient to link faults *per se*. After all, we had to add $\mathcal{O}(f_\ell)$ nodes to n_0 in order to mask f_ℓ link faults per node, which means that we added $\mathcal{O}(n^2)$ links. What we really gained, however, is that any link—and not just the ones added—may experience a fault. Moreover, the bound (28) on the probability Q_m of violating the fault model reveals that adding sufficiently many nodes is always beneficial as long as $np < 1$ is sufficiently small. In this case, the disadvantage of increasing the number of links that could be faulty is more than compensated by the ability to mask additional link faults per node. This ultimately confirms that

1. limiting the power of link faults according to our fault model is not an undue restriction,

2. our algorithms can even be employed in wireless systems, where link fault probabilities p up to 10^{-2} are common,

which was the ultimate goal for starting this research at all.

8.2 Broken Signatures

In the original Byzantine Generals paper [14], it was assumed that only messages from non-faulty nodes cannot be forged. If we translate that to the hybrid fault model, then messages from arbitrary faulty nodes can be forged, i.e., their signature is not secure. If we take this one step further, we can assume that the signatures of all arbitrary faulty nodes as well as the signatures of at most f_b not arbitrary faulty nodes are common knowledge. Knowing a signature allows a node to generate a message in the name of some other node, although this does not imply that it is able to eventually generate a valid chain of signatures.

Lemma 6 (Signatures) *At the end of OMHA's and ZA's execution run, there are no two messages $M = \sigma_{p_k} \ldots \sigma_{p_1}(v)$ and $M' = \sigma_{p_k} \ldots \sigma_{p_1}(v')$ with $v \neq v'$, if at least one signature σ_{p_i} in M, M' is from a not arbitrary faulty node p_i with an unbroken signature.*

Proof: Let us assume that two such messages M and M' exist. If we consider one node p_i which has signed both messages, then this node must have received $\sigma_{p_{i-1}} \ldots \sigma_{p_1}(v)$ and $\sigma_{p_{i-1}} \ldots \sigma_{p_1}(v')$. But according to (M4) in Section 4, every not arbitrary faulty node at most signs the first message, but not the second one. So if p_i has signed both messages, it must be arbitrarily faulty. If the node has not signed the messages, then someone else must have done so in its name, so its signature must have been broken. □

Recalling the operation of our Byzantine agreement algorithms, it is apparent that in OMHA(1) and ZA(1) every non-faulty node applies the hybrid-majority to an input set whose chain of signatures only differs in the last one. The chains may have different lengths, though, but if a chain has less than $m + 1$ signatures, then it must contain the value E. For all chains with length $m + 1$, we can deduce from Lemma 6 that each node will act upon the same input set if there is at least one not arbitrary faulty node in the chain whose signature has not been broken. So we can solve the problem of broken signatures by treating nodes with compromised signatures like arbitrary faulty nodes.

Theorem 7 (ZA with Broken Signatures) *For any $m \geq f_a + f_b + min\{1, f_\ell^s\}$, algorithm ZA(m) satisfies agreement and validity if there are strictly more than $f_\ell^s + f_\ell^r + f_a + f_b + f_s + f_m + 1$ participating nodes.*

Proof: For validity, it is easy to see that if the transmitter is not arbitrary faulty (and thus due to our assumption has no broken signature), then the only values that a non-faulty node considers are the value ν sent by the transmitter and E. Therefore, the argument of Lemma 3 still holds.

For agreement, we now use enough rounds to ensure that every message received in ZA(0) has been signed by at least one not arbitrary faulty node or contains E. Therefore, Lemma 6 guarantees that no fictive messages can occur, and the algorithm is still the same as without broken signatures. So the proof of Theorem 4 still holds if we count broken signatures as arbitrary faults. □

Algorithm OMHA could be made tolerant to broken signatures in the same fashion as ZA. However, for OMHA it is probably cheaper with respect to the required number of nodes to simply let it degrade to OMH. So in fact, if there is a possibility that signatures might be compromised, then one should spend additional f_ℓ^{ra} nodes according to Theorem 2 or, preferably, simply dispose of authentication and use OMH instead.

8.3 Broadcast Networks

The consensus algorithms analyzed in this paper assume a point-to-point network, which implies that the sender of a message is always known. If we use those algorithms on a broadcast network, however, the sender is not necessarily known: If we do not sign messages, we obviously loose the ability to identify the sender of a message, thus allowing faulty nodes to impersonate non-faulty nodes. So the oral messages algorithms would not work in this case. Written messages algorithms, however, should reasonably[10] work because they do not allow impersonation. To be precise, any oral messages algorithm that achieves consensus under the system model of Definition 1 will achieve consensus in a broadcast network if authentication is added. This is due to the fact that using a broadcast network only violates assumption (A2), i.e., the sender is not known. With authentication, (A2) is ensured again and the system model remains the same. In fact, without link faults, written messages algorithms would even benefit from the broadcast network, because neither arbitrary nor omission faulty nodes are possible anymore. Since every node sends only one message, which is automatically broadcast to all nodes, every receiver must get the same value. So we can in fact set $f_a = f_o = 0$ and count all arbitrary faults as symmetric faults and all omission faults as manifest faults for any written messages algorithm analyzed under the hybrid fault model.

If link faults are possible, however, we find that they now have a lot more power than before. Whereas they can simply be caught by adding an appropriate multiple of f_ℓ^r and f_ℓ^s to the number of nodes in the point-to-point case, we experience the unpleasant effect that they make arbitrary (but not omission) faults possible in the broadcast case [25]. However, the behavior of arbitrary nodes is restricted:

Consider a message from an arbitrary faulty node which is not received by f_ℓ^s receivers. If that node sends a second message containing a different value, which is not received by another f_ℓ^s receivers, then at most $2f_\ell^s$ receivers will only get one message and will assume that the message is valid. The other nodes do detect the second message from the same sender and will use the value E due to the manifest fault. So the obvious solution is either to count arbitrary faults again, or to count sender link faults twice, i.e., require $4f_\ell^s$ instead of $2f_\ell^s$ additional nodes.

When analyzing OMHA in broadcast networks, we can exploit the restricted behavior of arbitrary nodes. First, it is easily seen that the validity proof can be taken over unchanged: Since the transmitter is not arbitrary faulty, and since arbitrary faulty nodes do not occur in the last but one line of equations (5) and (6) in the proof of Lemma 1, their different behavior in the broadcast network has no impact on validity and the proof of Lemma 1 still holds. As far as omission faulty nodes are concerned, they either behave non-faulty or like crashed nodes. In any case, all receivers will deliver the same value for them.

Theorem 8 (Agreement and Validity) *For any f_a, f_s, f_o, f_m, f_ℓ^s, f_ℓ^r and any $m \geq \min\{1, f_\ell^s\}$, OMHA(m) satisfies agreement (B1) and validity (B2) if there are strictly more than $4f_\ell^s + f_\ell^r + 2(f_a + f_s) + f_o + f_m + m$ participating nodes.*

Proof: Again, we only look at the case where the transmitter behaves arbitrary faulty as described above.

Let $m = 1$. Abbreviating the number of initially participating receivers with $n' \geq 4f_\ell^s + f_\ell^r + 2(f_a + f_s) + f_o + f_m + m$, the transmitter sends ν, ν', which is received and turned into E by at least $n' - 2f_\ell^s - (f_a - 1) - f_s - f'_o - f'_m$ non-faulty receivers, whereas at most f_ℓ^s nodes receive only ν and f_ℓ^s only ν'. Here, $f'_o \leq f_o$ is the number of omission faulty receivers that will commit a crash fault in OMHA(0) (all others will appear like non-faulty nodes) and $f'_m \leq f_m$ is the number of actual crashed nodes.

In OMHA(0), a non-faulty receiver gets $R(E)$ from at least $n'_q = n' - 2f_\ell^s - (f_a - 1) - f_s - f'_o - f'_m - f_\ell^{r\prime}$ nodes, with $f_\ell^{r\prime} \leq f_\ell^r$ link faults according to (A1^r), and it receives at most $n''_q = n' - f_\ell^{r\prime} - f'_o - f'_m$ values different from E. Therefore, we have

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

and $R(E)$ will win the hyrid-majority on every non-faulty node.

Let us now consider $m > \min\{1, f_\ell^s\}$. At least $\overline{n}'_q = n' - 2f_\ell^s - (f_a - 1) - f_s - f'_m$ non-faulty or omission faulty receivers will receive E in OMHA(m). From these, all but at most f'_o will disseminate $R(E)$ in OMHA($m-1$), and validity ensures that every non-faulty node will deliver $R(E)$ for them. The f'_o omission faulty nodes will appear crashed and cause E. Therefore, every non-faulty node will deliver at most $\overline{n}''_q = n' - f'_o - f'_m$ values different from E. Since

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

$R(E)$ will again win the hybrid-majority on all non-faulty nodes, hence all non-faulty nodes will deliver E. □

[10] Besides of the problem of jamming.

The algorithm ZA, however, cannot exploit the different behavior of arbitrary faults so easily. Here, arbitrary faulty nodes must still be counted in m. The reason is obvious from the proof of Theorem 8: We have utilized the fact that every non-faulty node receives enough $R(E)$ values to win the hybrid-majority. This has saved us from using enough rounds to ensure that all nodes get exactly the same input set for the hybrid-majority. In ZA, however, only E does exist, which is not considered in the majority function. Therefore, we will again have to ensure that all nodes work with the same input set in ZA(0), effectively requiring $m \geq f_a + \min\{1, f_\ell^s\}$ again.

Remarks:

1. Note that an arbitrary faulty node can do the worst damage by sending two messages. With a third message, again only f_ℓ^s receivers might not detect a manifest fault.

2. When executing OMHA on a broadcast network, the tradeoff between the point-to-point algorithm of Theorem 3 and the broadcast version of Theorem 8 is f_a vs. $2f_\ell^s$ additional nodes and $f_a + 1$ vs. 1 rounds.

3. When executing ZA on a broadcast network, there is the slight tradeoff f_a vs. $2f_\ell^s$. However, since f_ℓ^s will probably be larger than f_a anyway, ZA does not benefit from the broadcast network.

Acknowledments

We are grateful to Martin Biely for his comments on an earlier version of this paper.

9 Conclusions

In this paper, we showed that deterministic consensus in presence of link faults is possible, despite the impossibility result of [12]. The latter is avoided by limiting the maximum number of link faults in the broadcast resp. reception of any node. We introduced a novel perception-based hybrid fault model for this purpose, 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. For $m \geq f_a + f_o + 1$, we analyzed three existing $m+1$-round Byzantine agreement algorithms under this fault model, namely, the non-authenticated OMH as well as its authenticated variants OMHA and ZA. Their respective number of nodes was shown to be

$$\begin{aligned} n &> 2f_\ell^s + f_\ell^r + f_\ell^{ra} + 2(f_a + f_s) + f_o + f_m + m, \\ n &> 2f_\ell^s + f_\ell^r + 2(f_a + f_s) + f_o + f_m + m, \\ n &> f_\ell^s + f_\ell^r + f_a + f_s + f_m + 1. \end{aligned}$$

We also provided an impossibility result and associated lower bounds for the required number of nodes in presence of omission and arbitrary link faults, which are matched by ZA and OMH and are therefore tight. Moreover, our investigations led to a precise characterization of what makes a node fault arbitrary/omissive. 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. We computed the probability Q of violating the link fault assumption $f_\ell^r = f_\ell^s = f_\ell$, which shows that our approach of adding nodes in order to tolerate additional link faults per node always decreases Q as long as $np < 1$. Consequently, for reasonably small m, our algorithms can be used even in wireless systems, where link faults with loss probabilities up to $p = 10^{-2}$—as well as intrusions—are the dominating source of errors. Given the limited bandwidth usually available in wireless systems, the excessive communication requirements may be prohibitive, though.

Our results also reveal that the usefulness of authentication depends heavily upon the particular algorithm used. In fact, a consensus algorithm should be specifically designed for using written messages and not simply adapted from an oral messages solution: Whereas OMHA did not profit much from authentication, ZA benefits considerably— but also depends critically upon its strength. It turned out, however, that both algorithms can withstand intrusions to some extent: In case of broken signatures, OMHA degrades to OMH and hence requires an additional f_ℓ^{ra} in the number of nodes. For ZA, a node with a compromised signature must be considered as arbitrary faulty and therefore counted in f_a. As far as link faults are concerned, authentication serves to identify and tolerate link value faults: Any algorithm that requires $f_\ell^r + f_\ell^{ra}$ nodes to tolerate link faults will only require f_ℓ^r nodes in the authenticated version. Apart from that, authentication is the only means to (more or less) safely employ algorithms like OMHA on top of broadcast networks.

There are two primary directions of current/future research in this area: First, we have already made progress in the analysis of less message-costly consensus algorithms [8] under the perception-based fault model, which even includes algorithms for asynchronous systems. Second, there are many applications like intrusion-tolerant admission control, on-line diagnosis, distributed database commitment, etc. that could benefit from the possibility to achieve consensus in systems with link faults. A major part of our future research will hence be devoted to the exploration of such applications.

References

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions*. Dover Publications, Inc., New York, 1970.

[2] H. Abu-Amara and J. Lokre. Election in asynchronous complete networks with intermittent link failures. *IEEE Transactions on Computers*, 43(7):778–788, July 1994.

[3] M. K. Aguilera, W. Chen, and S. Toueg. Failure detection and consensus in the crash-recovery model. *Distributed Computing*, 13(2):99–125, Apr. 2000.

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

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

[6] M. H. Azadmanesh and R. M. Kieckhafer. New hybrid fault models for asynchronous approximate agreement. *IEEE Transactions on Computers*, 45(4):439–449, 1996.

[7] A. Basu, B. Charron-Bost, and S. Toueg. Crash failures vs. crash + link failures. In *Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing*, page 246. ACM Press, 1996.

[8] M. Biely and U. Schmid. Message-efficient consensus in presence of hybrid node and link faults. Technical Report 183/1-116, Department of Automation, Vienna University of Technology, August 2001.

[9] D. Dolev. The Byzantine generals strike again. *Journal of Algorithms*, 3(1):14–30, 1982.

[10] P. T. Eugster, R. Guerraoui, S. Handurukande, A.-M. Kermarrec, and P. Kouznetsov. Lightweight probabilistic broadcast. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN'01)*, pages 443–452, Göteborg, Sweden, July 1–4, 2001.

[11] 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.

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

[13] D. E. Knuth. *Fundamental Algorithms*, volume 1 of *The Art of Computer Programming*. Addison-Wesley, Reading, Massachusetts, 2nd edition, 1973.

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

[15] 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.

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

[17] 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.

[18] S. E. Nikoletseas and P. G. Spirakis. Expander properties in random regular graphs with edge faults. In *12th Annual Symposium on Theoretical Aspects of Computer Science (STACS'95)*, pages 421 – 432, Munich, Germany, 1995.

[19] 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.

[20] 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.

[21] D. Powell. Failure mode assumptions and assumption coverage. In *Proc. 22nd IEEE Int. Symp. on Fault-Tolerant Computing (FTCS-22)*, pages 386–395, Boston, MA, USA, 1992. (Revised version available as LAAS-CNRS Research Report 91462, 1995).

[22] 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.

[23] 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.

[24] 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.

[25] U. Schmid. Synchronized Universal Time Coordinated for distributed real-time systems. *Control Engineering Practice*, 3(6):877–884, 1995. (Reprint from Proceedings 19th IFAC/IFIP Workshop on Real-Time Programming (WRTP'94), Lake Reichenau/Germany, 1994, p. 101–107.).

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

[27] U. Schmid. A perception-based fault model for single-round agreement algorithms. Technical Report 183/1-108, Vienna University of Technology, Department of Automation, Oct. 2000.

[28] 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.

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

[30] U. Schmid, B. Weiss, and J. Rushby. Formally verified byzantine agreement in presence of link faults, 2001. (submitted).

[31] G. Singh. Leader election in the presence of link failures. *IEEE Transactions on Parallel and Distributed Systems*, 7(3):231–236, Mar. 1996.

[32] 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.

[33] T. K. Srikanth and S. Toueg. Optimal clock synchronization. *Journal of the ACM*, 34(3):626–645, July 1987.

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

[35] G. Varghese and N. A. Lynch. A tradeoff between safety and liveness for randomized coordinated attack protocols. In *Proceedings of the 11th Annual ACM Symposium on Pprinciples of Distributed Computing*, pages 241–250, Vancouver, British Columbia, Canada, August 1992.

[36] 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).

[37] C. J. Walter, N. Suri, and M. M. Hugue. Continual on-line diagnosis of hybrid faults. In *Proceedings DCCA-4*, Jan. 1994.

[38] E. Whittaker and G. Watson. *A Course of Modern Analysis*. Cambridge University Press, Cambridge, 1927.

[39] L. Zhou, F. B. Schneider, and R. van Renesse. COCA: A secure distributed on-line certification authority. Technical Report TR2000-1828, Computer Science Department, Cornell University, Dec. 2000.