Message-Efficient Consensus Algorithms in Presence of Hybrid Node and Link Faults

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

Abstract

In a recent paper (Schmid & Weiss, 2001), we showed how to avoid the impossibility of deterministic consensus in presence of link faults (Gray, 1978) for the well-known Oral and Written Messages Byzantine agreement algorithms (Lamport, Shostak & Pease, 1982). In this paper[1]*, we analyze several alternative algorithms that do not require messages of exponential size: An improved version of the Phase Queen and Phase King consensus algorithms of (Berman, Garay & Perry, 1992) and the non-authenticated Byzantine agreement algorithm of (Srikanth & Toueg, 1987) are shown to require just one additional round and*

$$\begin{aligned} n &> 2f_\ell^s + 2f_\ell^r + 2f_\ell^{r,a} + 4f_a + 2f_s + 2f_o + f_m \\ n &> 2f_\ell^s + 2f_\ell^r + 2f_\ell^{r,a} + 3f_a + 2f_s + 2f_o + f_m \\ n &> f_\ell^s + f_\ell^{s,a} + 2f_\ell^r + 2f_\ell^{r,a} + 3f_a + 2f_s + 2f_o + f_m \end{aligned}$$

nodes, respectively, for tolerating at most f_ℓ^s broadcast and f_ℓ^r receive link faults (including at most $f_\ell^{s,a}$ and $f_\ell^{r,a}$ 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. Being sub-optimal with respect to link faults, those resiliences can be considered as the price for the algorithms' message efficiency. An analysis of the assumption coverage in systems where links fail independently with probability p reveals that adding nodes for tolerating link faults results in a vanishing probability of violating the fault model as long as $np < 1$. This confirms that our algorithms can reasonably be employed even in wireless networks, which suffer from low bandwidth and link failure probabilities up to $p = 10^{-2}$.

Keywords: *Fault-tolerant distributed systems, hybrid fault models, link faults, consensus, Byzantine agreement.*

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

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* 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.

In [25, 26], we provided a comprehensive hybrid fault model that captures both node and link faults, and showed that it allows consensus with *Oral* and *Written Messages* (algorithms OMH, OMHA & ZA [11, 14]) in synchronous distributed systems, despite of the impossibility result of [12, 15]. The key for escaping from the impossibility result is to limit the inconsistency caused by link faults in the broadcasts of a single sender and the receptions at a single receiver. Limiting the power of link faults that way, their effects can be masked transparently, provided that sufficiently many nodes are available. Still, the algorithms considered in [25] suffer from exponential message size. In this paper, we show that several message-efficient algorithms can also

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

be adopted for and analyzed under our fault model. It is hence possible to achieve consensus even in systems with low communication bandwidth and high link failure rates.

The remaining sections of our paper are organized as follows: In Section 2, we review the perception-based hybrid fault model introduced in [25]. Section 3 resp. 4 contains the description and analysis of the Phase Queen resp. Phase King consensus algorithm of [9] under our fault model, Section 5 is devoted to Srikanth & Toueg's Byzantine agreement algorithm of [28]. Section 6 contains the analysis of the assumption coverage in typical system architectures. A discussion of our results and some directions of further research in Section 7 eventually conclude the paper.

2 System 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[3] systems: The practical considerations of Section 1 suggest to grant each receiving 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 solveable in this model.

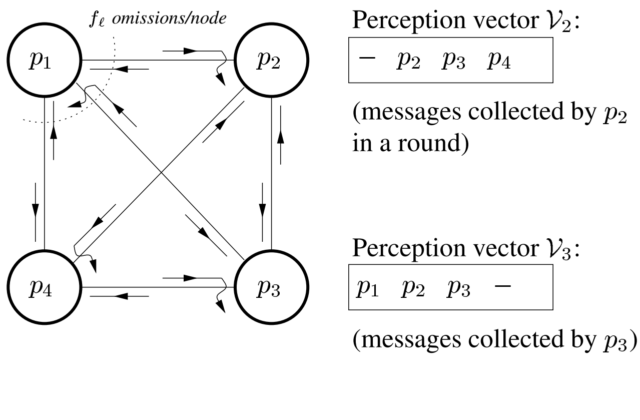

Figure 1. *Example of a 4-node system with $f_\ell = 1$ receive link faults per node in each round, where all nodes must be considered faulty in existing fault models.*

[3]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. [1, 5, 36]. 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.

A similar argument applies to the more detailed send/receive-omission fault model of [18], where receive omissions are mapped to receiver node faults. Although it has been observed in this paper that only the number of nodes that commit a send omission (but not the number of nodes committing a receive omission) needs to be counted in f, agreement is only 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 due to the fact that any receiver's f_ℓ receive omissions 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.

The key to the accomplishments of [25] is a generalization of the *perception-based hybrid fault model* for single-round clock synchronization and (approximate) agreement algorithms developed in [23]. In this model, we replaced the usual global, i.e., system-wide number of faults by the number of faults that are observable in the nodes' local "perceptions" of the system. Belonging to the class of hybrid fault models [3, 4, 14, 16, 20, 22, 24, 30, 32, 33], our model (restated in Definition 1 below) distinguishes different classes of faults, namely, manifest, omission, symmetric, and arbitrary. This allows us to exploit the fact that less severe faults can be handled with fewer nodes than more severe ones. For example, it will turn out that masking f symmetric faults requires only $n \geq 2f+1$ nodes, whereas $n \geq 3f+1$ is needed if all faults are asymmetric (Byzantine) ones. Since a large number of asymmetric faults is quite unlikely in practice, this effectively leads to a smaller n for tolerating a given number of faults. This, in turn, has a positive effect upon dependability by reducing the number n of components that could be faulty, cf. [19]. 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.

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 $V_q^p = \emptyset$ 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 $V_{q_i}^p = \emptyset$ 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 $f_\ell^{s,a} \leq f_\ell^s$ 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 $f_\ell^{r,a} \leq f_\ell^r$ 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 $V_q^p = \emptyset$ for some distinguished value $\emptyset$.

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. 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.

3. A sender node that suffers from link faults according to (A1^s) is said to commit a *broadcast fault*, whereas a receiving node that experiences link faults according to (A1^r) is said to commit a *receive fault*. Each node's receive resp. broadcast fault has its own "budget" f_ℓ^r resp. f_ℓ^s of individual link faults, which are independent of node faults. 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.

4. 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
$$f_\ell^s \leq f_\ell^r \quad \text{and} \quad f_\ell^{s,a} \leq f_\ell^{r,a}, \tag{1}$$
see [25, 26] for details. Note that this implies $f_\ell^r = 0 \Rightarrow f_\ell^s = 0$.

5. In (A1^s) and (A1^r), we assume that link faults hit only messages from non-faulty senders/receivers. In practice, however, it would be 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 usually turns out that the results obtained under the model of Definition 1 are valid for the more general independent model as well. This is particularly easy to show for (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 $f_\ell^s, f_\ell^{s,a}, \ldots$, but rather on a suitable (weighted) sum of those.

   For example, the implementation of the Phase Queen algorithm in Figure 2 depends only upon $f_a + f_s + f_o + f_c$ and requires $n > 4f_a + 2f_s + 2f_o + f_m + 2f_\ell^s + 2f_\ell^r + 2f_\ell^{r,a}$ nodes according to Theorem 1. 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 ($f_s \to f_s - 1, f_\ell^r \to f_\ell^r - 1, f_a \to f_a + 1$) do not affect the above expressions, Theorem 1 (as well as Theorem 2 and 5) is in fact also valid for the general independent model.

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

In [21,25,26,34], we studied the fundamental problem of reaching consensus among the nodes of a distributed system under the fault model of Definition 1. Apart from developing suitable impossibility and lower bounds results, we showed that the resilience of consensus algorithms is much better than the results under traditional fault models suggest: For example, the (optimal) authenticated Byzantine agreement[4] algorithm ZA of [11], which can cope with f arbitrary node faults as long as $n > f$, can be made tolerant to f_ℓ additional lost/faulty messages at every node per round just by setting $n > f + 2f_\ell$; this algorithm would hence achieve Byzantine agreement in the example of Figure 1. In the limit, ZA tolerates up to $nf_\ell = \mathcal{O}(n^2)$ link faults system-wide during each round. In the subsequent sections, we will show that similar results can be derived for message-efficient algorithms as well.

[4]Byzantine agreement can be viewed as reaching consensus among the values disseminated by a single transmitter.

3 Phase Queen Algorithm

Assuming that every node p is provided with some *initial value* x_p, a consensus algorithm computes a *decision value* v_q at every node q that satisfies the following properties:

(C1) *Agreement*: If nodes p and q are both non-faulty, then both compute the same $v_p = v_q$.

(C2) *Validity*: If all non-faulty nodes start with the same initial value $\forall p : x_p = v$, then any non-faulty node q computes $v_q = v$.

Note that we will only consider deterministic algorithms for synchronous systems in this paper, so every non-faulty node must compute its decision value within a fixed number of rounds.

The *Phase Queen Protocol* introduced in [9] is a simple algorithm that solves consensus for binary values in systems with $n > 4f$ nodes, at most f of which may be arbitrary faulty, and no link faults. It assumes unique node identifiers $\in \{1, \ldots, n\}$, uses constant size (1-bit) messages and takes $f + 1$ rounds with 2 phases each. Our contribution is a modified version of the original algorithm that can cope with hybrid node and link faults according to the system model of Definition 1.

Figure 2 shows the pseudocode of our hybrid algorithm, which works as follows: In the first phase of each round, every node p broadcasts its current preference value v_p to all nodes in the system (including itself) and collects the resulting messages from its peers; we will call this a *full message exchange*. Node p then counts how many nodes sent preference for 0 and 1, respectively, and updates its v_p accordingly. In the second phase of each round, only one specific node (the *Phase Queen*, whose identifier is equal to the current round number) broadcasts its new preference. This value is used by all nodes in the system to break ties.

Note that we made only two non-trivial changes to the original algorithm: First, the bound in the decision when to use the queen's broadcast (line 14 in Figure 2) had to be adopted to our hybrid fault model. Second, in order to improve the resilience with respect to non-arbitrary faults, the original condition $C[1] > n/2$ in the update of the preference value (line 7) had to be replaced by the simple majority test $C[1] > C[0]$.

In order to show that the above algorithm satisfies the agreement (C1) and validity property (C2) of consensus, the analysis of the Phase Queen algorithm from [2] will be adopted. We start with some simple properties of full message exchanges: The first lemma bounds the difference in the number of messages with a particular value received at two different nodes.

Lemma 1 (Difference in Perceptions) *Let $C_r[v]$ and $C_q[v]$ be the number of messages containing v received at two non-faulty nodes r and q in a full message exchange of a system complying to Definition 1. Then,*

$$|C_q[v] - C_r[v]| \leq f_a + f_o + f_\ell^r + f_\ell^{r,a}.$$

---

for k:=1 **to** $f_a + f_s + f_o + f_m + 2$ **do begin**
    / Phase 1: full message exchange */*
    send(v) **to all nodes** [*including itself*]
    receive(v_q) **from all nodes**
    $C[0] := |\{v_q : v_q = 0\}|$
    $C[1] := |\{v_q : v_q = 1\}|$
    if $C[1] > C[0]$ **then** $v = 1$ **else** $v = 0$ **fi**
    / Phase 2: queen's broadcast */*
    if $k = p$ **then**
        send(v) **to all nodes** [*including itself*]
    fi
    receive(v_{queen})
    if $v_{queen} = \emptyset$ **then** $v_{queen} := 0$ **fi**
    if $C[v] \leq C[1-v] + 2f_a + f_o + f_\ell^r + f_\ell^{r,a}$ **then**
        $v := v_{queen}$
    fi
done

Figure 2. *Hybrid Phase Queen algorithm, code for node p*

Proof: Assuming w.l.o.g. $C_q[v] > C_r[v]$, at most f_a arbitrary faulty senders could have sent v to node q and $x \neq v$ to node r, and at most f_o nodes could have sent v to q but no message to r. Node faults hence contribute at most $f_a + f_o$ to $|C_q[v] - C_r[v]|$. The remaining terms originate from link faults: Node q could have received at most $f_\ell^{r,a}$ messages containing v from nodes that actually sent $x \neq v$, and at most f_ℓ^r messages containing v could have been lost in transit to node r. □

When using a majority test after a full message exchange as in Figure 2, inconsistent reception could produce a majority for the value v at some node q but a majority for $x \neq v$ at some node $r \neq q$. The question arises how big the lead of v must be at node q to be able to guarantee that r will compute the same majority? This is answered by the following Lemma 2.

Lemma 2 (Deviation of Differences) *Let $C_r[v]$ and $C_q[v]$ be the number of messages containing $v \in \{0, 1\}$ received at two non-faulty nodes r and q in a binary full message exchange of a system complying to Definition 1. For $\Delta_q[v] = C_q[v] - C_q[1-v]$ and $\Delta_r[v] = C_r[v] - C_r[1-v]$, we obtain*

$$|\Delta_q[v] - \Delta_r[v]| \leq 2f_a + f_o + f_\ell^r + f_\ell^{r,a}.$$

Proof: Abbreviating $\Delta = |\Delta_q[v] - \Delta_r[v]|$, each asymmetric faulty node could change Δ by at most 2, by sending v to q but $1 - v$ to r, while each omission faulty node could change Δ by at most 1, by failing to send a message to either q or r; the remaining symmetric and manifest faults cannot change Δ. Node faults could hence grow/shrink Δ by at most $2f_a + f_o$. The worst case deviation due to link faults occurs when both q and r experience $f_\ell^{r,a}$ arbitrary ones (which flip the value sent) and $f_\ell^r - f_\ell^{r,a}$ omissive ones from different sender nodes; the appropriate contributions sum up to $2f_\ell^{r,a} + (f_\ell^r - f_\ell^{r,a}) = f_\ell^r + f_\ell^{r,a}$. □

We are now ready for the first major lemma, which states that once sufficiently many nodes have the same preference value, it will not change anymore. Lemma 3 will subsequently be used to show the validity property (C2).

Lemma 3 (Persistence of Agreement) *If at least $n - f_a - f_s - f_o - f_m - f_\ell^s$ non-faulty nodes prefer $v \in \{0,1\}$ at the beginning of round $k \in \{1,\dots,f_a+f_s+f_m+2\}$ in a system with $n > 4f_a+2f_s+2f_o+f_m+2f_\ell^s+2f_\ell^r+2f_\ell^{r,a}$, then all non-faulty nodes prefer v at the end of round k.*

Proof: Since $n - f_a - f_s - f_o - f_m - f_\ell^s$ non-faulty nodes prefer v at the beginning of phase k, every non-faulty node q receives $C_q[v] \geq n - f_a - f_s - f_o - f_m - f_\ell^s - f_\ell^r$ preferences for v (including its own) in the first phase of round k. Moreover, since the preference of f_ℓ^s non-faulty nodes was left unspecified by Lemma 3 and at most $f_a + f_s$ nodes and $f_\ell^{r,a}$ links may cause erroneous values, node q could also get $C_q[1-v] \leq f_a + f_s + f_\ell^s + f_\ell^{r,a}$ preferences for $1 - v$. This implies

$$\begin{aligned} C_q[v] &\geq n - f_a - f_s - f_o - f_m - f_\ell^s - f_\ell^r \\ &> 3f_a + f_s + f_o + f_\ell^s + f_\ell^r + 2f_\ell^{r,a} \\ &\geq C_q[1-v] + 2f_a + f_o + f_\ell^r + f_\ell^{r,a}, \end{aligned}$$

so every non-faulty node q updates its preference v_q to v in line 7 in Figure 2 and ignores the queen's broadcast in line 14. □

The second major lemma is a prerequisite for showing that agreement can be reached by means of a non-faulty queen breaking ties.

Lemma 4 (Agreement) *Let g be a round whose queen is non-faulty. Then, at least $n - f_a - f_s - f_o - f_m - f_\ell^s$ non-faulty nodes finish round g with the same preference.*

Proof: Since the queen g is non-faulty, it sends the same v_g to all receivers. Assume first that all non-faulty nodes use the value v_{queen} received from g as their new preference in line 15 of Figure 2. According to (A1^s), at most f_ℓ^s of those could have received a queen's preference $v_{queen} \neq v_g$,[5] such that at least $n - f_a - f_s - f_o - f_m - f_\ell^s$ non-faulty nodes set their preference to v_g.

If, on the other hand, some non-faulty node p ignores the queen's broadcast and uses its own majority value v as its new preference, $C_p[v] - C_p[1-v] > 2f_a + f_o + f_\ell^r + f_\ell^{r,a}$ according to line 14 in Figure 2. Hence, for every other non-faulty node q (including the queen g), $C_q[v] - C_q[1-v] \geq C_p[v] - C_p[1-v] - 2f_a - f_o - f_\ell^r - f_\ell^{r,a} > 0$ by Lemma 2, so g must have set its preference to v in line 7 as well. Node p can therefore safely use its majority value instead of the queen's in this case, since they are the same. □

[5] At most $f_\ell^{s,a}$ nodes could receive the opposite value $1 - v_g$, and the remaining $f_\ell^s - f_\ell^{s,a}$ ones could suffer from omissions, that is, deliver $\emptyset$.

By means of Lemma 3 and 4, it is not difficult to prove the following major theorem.

Theorem 1 (Phase Queen) *Under the system model of Definition 1 with $f_a, f_s, f_o, f_m, f_\ell^r, f_\ell^{r,a}, f_\ell^s, f_\ell^{s,a} \geq 0$ and $n > 4f_a + 2f_s + 2f_o + f_m + 2f_\ell^s + 2f_\ell^r + 2f_\ell^{r,a}$, the Phase Queen algorithm of Figure 2 solves binary consensus in $2(f_a + f_s + f_o + f_m + 2)$ phases with a total of $(f_a + f_s + f_o + f_m + 2)(n+1)$ 1-bit message broadcasts from non-faulty nodes.*

Proof: Lemma 3 already implies the validity property (C2): If all non-faulty nodes start out with the same value v, they will continue to prefer v until the algorithm terminates after $f_a + f_s + f_o + f_m + 2$ rounds, since there are $n - f_a - f_s - f_o - f_m > n - f_a - f_s - f_o - f_m - f_\ell^s$ non-faulty nodes.

To show agreement (C1), we note that all but at most f_ℓ^s non-faulty nodes will have the same preference at the end of a non-faulty queen's round g by Lemma 4. Definition 1 ensures that at least one round $g \in \{1,\dots,f_a+f_s+f_o+f_m+1\}$ has a non-faulty queen. Since the algorithm takes $f_a + f_s + f_o + f_m + 2$ rounds, Lemma 3 eventually assures that all non-faulty nodes will have the same (persistent) preference by the end of round $g+1$, no matter how many rounds with faulty queens follow.

To justify the claimed time and communication complexity of our algorithm, we note that there is one full message exchange consisting of n broadcasts in phase 1 and one additional queen's broadcast in phase 2 of every round. Hence, at most $(f_a+f_s+f_o+f_m+2)(n+1)$ 1-bit message broadcasts are performed during the whole execution by nodes that faithfully[6] follow the algorithm in Figure 2. □

As a final remark, we note that the number of message broadcasts of the Phase Queen algorithm could be reduced by one by omitting the queen's broadcast in the last round: According to the proof of Theorem 1, no non-faulty node uses v_{queen} in the last round at all.

4 Phase King Algorithm

The *Phase King Protocol* of [9] improves the Phase Queen algorithm by adding one phase in every round, in which the nodes' preferences from the first phase are exchanged systemwide. This eventually reduces the sub-optimal fault tolerance degree $n > 4f$ of the Phase Queen algorithm to $n > 3f$. We provide a hybrid variant of this algorithm in Figure 3, which can cope with node and link faults according to Definition 1.

[6] Clearly, there is no way to restrict the number of message broadcasts initiated by an arbitrary faulty node.

for k:=1 **to** $f_a + f_s + f_o + f_m + 2$ **do begin**
  / Phase 1: initial full message exchange */*
  send(v) **to all nodes** [*including itself*]
  receive(v_q) **from all nodes**
  $C[0] := |\{v_q : v_q = 0\}|$
  $C[1] := |\{v_q : v_q = 1\}|$
  / Phase 2: C[j] full message exchange */*
  for j:=0 **to** 1 **do begin**/* *simultaneously* */
    if $C[j] > C[1-j] + f_a + f_o + f_\ell^r + f_\ell^{r,a}$ **then**
      M[j]:=1 **else** M[j]:=0
    fi
    send($M[j]$) **to all nodes** [*including itself*]
    receive($M_q[j]$) **from all nodes**
    $D[j] := |\{M_q[j] : M_q[j] = 1\}|$
  done
  if $D[1] > f_a + f_s + f_\ell^{r,a}$ **then** v:=1 **else** v:=0 **fi**
  / Phase 3: kings's broadcast */*
  if $k = p$ **then**
    send(v) **to all nodes** [*including itself*]
  fi
  receive(v_{king})
  if $v_{king} = \emptyset$ **then** $v_{king} := v$ **fi**
  if $D[v] \leq 2f_a + f_s + f_o + f_\ell^r + 2f_\ell^{r,a}$ **then**
    $v := v_{king}$
  fi
done

Figure 3. *Hybrid Phase King algorithm, code for node p*

The proof for the Phase King Protocol is similar to the one for the Phase Queen Protocol: The first Lemma 5 will show that once sufficiently many nodes have the same preference value, it will not change anymore.

Lemma 5 (Persistence of Agreement) *If at least $n - f_a - f_s - f_o - f_m - f_\ell^s$ non-faulty nodes prefer v at the beginning of round $k \in \{1, \ldots, f_a + f_s + f_o + f_m + 2\}$ in a system with $n > 3f_a + 2f_s + 2f_o + f_m + 2f_\ell^s + 2f_\ell^r + 2f_\ell^{r,a}$ nodes, then all non-faulty nodes prefer v at the end of round k.*

Proof: Since at least $n - f_a - f_s - f_o - f_m - f_\ell^s$ non-faulty nodes broadcast v in the first full message exchange, every non-faulty node q obtains $C_q[v] \geq n - f_a - f_s - f_o - f_m - f_\ell^s - f_\ell^r$ and $C_q[1-v] \leq f_a + f_s + f_\ell^s + f_\ell^{r,a}$ according our fault model in Definition 1. Similar to the proof of Lemma 3, this leads to

$$C_q[v] > C_q[1-v] + f_a + f_o + f_\ell^r + f_\ell^{r,a},$$

which implies that every non-faulty node will send the same $M[j]$ in the second phase. Therefore, every non-faulty node r obtains $D_r[v] \geq n - f_a - f_s - f_o - f_m - f_\ell^r$ and $D_r[1-v] \leq f_a + f_s + f_\ell^{r,a}$ and thus sets its local preference to the same value v in line 16 in Figure 3; the king's value v_{king} is ignored in line 23 since $D_r[v] > 2f_a + f_s + f_o + f_\ell^r + 2f_\ell^{r,a}$. □

The second major lemma is again a prerequisite for showing that agreement can be reached by means of a non-faulty king breaking ties.

Lemma 6 (Agreement) *Let g be a round whose king g is non-faulty. If $n > 3f_a + 2f_s + 2f_o + f_m + 2f_\ell^s + 2f_\ell^r + 2f_\ell^{r,a}$, then at least $n - f_a - f_s - f_o - f_m - f_\ell^s$ non-faulty nodes finish round g with the same preference.*

Proof: At the end of round g, one of the following cases applies in line 23 of Figure 3:

(1) $D_p[v_p] \leq 2f_a + f_s + f_o + f_\ell^r + 2f_\ell^{r,a}$ for every non-faulty node p, which thus assigns the value v_{king} to v_p in line 24. Since at most f_ℓ^s nodes could have received a value v_{king} different from the value v_g actually sent, at least $n - f_a - f_s - f_o - f_m - f_\ell^s$ non-faulty nodes end up with the same preference v_g as asserted.

(2) $D_p[v_p] > 2f_a + f_s + f_o + f_\ell^r + 2f_\ell^{r,a}$ for some non-faulty node p. We have to show that p's preference at the end of the round is equal to the king's value v_g in this case, which will again imply that all non-faulty nodes except at most f_ℓ^s will end up with a preference for the king's value v_g — either by receiving it, or by being convinced of it in the first place. We distinguish the two possible cases here: If $v_p = 1$, it must be that $D_g[1] > f_a + f_s + f_\ell^{r,a}$ and hence $v_g = 1$, since Lemma 1 implies $D_g[1] \geq D_p[1] - f_a - f_o - f_\ell^r - f_\ell^{r,a} > f_a + f_s + f_\ell^{r,a}$. If, on the other hand, $v_p = 0$, we can use the following argument to show that v_g cannot be 1:

We first show that if a non-faulty node q sends $M_q[b] = 1$ for some $b \in \{0, 1\}$ in the second phase, then no non-faulty node r can send $M_r[1-b] = 1$. Assuming the contrary, line 9 in Figure 3 would require both

$$\Delta_q = C_q[b] - C_q[1-b] > f_a + f_o + f_\ell^r + f_\ell^{r,a}$$
$$-\Delta_r = C_r[1-b] - C_r[b] > f_a + f_o + f_\ell^r + f_\ell^{r,a}$$

and hence $\Delta_q - \Delta_r > 2f_a + 2f_o + 2f_\ell^r + 2f_\ell^{r,a}$, which would contradict Lemma 2.

Now, $D_p[0] > 2f_a + f_s + f_o + f_\ell^r + 2f_\ell^{r,a}$ implies that at least one non-faulty node sent $M[0] = 1$ in the second phase. Consequently, as we have just shown, no non-faulty node can have sent $M[1] = 1$. Since this implies $D_q[1] \leq f_a + f_s + f_\ell^{r,a}$ for any non-faulty node q (including the king g), v_g is set to 0 in line 16, just before the kings broadcast. This eventually completes the proof of Lemma 6. □

We can now show the following major Theorem 2.

Theorem 2 (Phase King) *Under the system model of Definition 1 with* $f_a, f_s, f_o, f_m, f_\ell^r, f_\ell^{r,a}, f_\ell^s, f_\ell^{s,a} \geq 0$ *and* $n > 3f_a + 2f_s + 2f_o + f_m + 2f_\ell^s + 2f_\ell^r + 2f_\ell^{r,a}$*, the Phase King algorithm of Figure 3 solves binary consensus in* $3(f_a + f_s + f_o + f_m + 2)$ *phases with a total of* $(f_a + f_s + f_o + f_m + 2)(3n + 1)$ *1-bit message broadcasts from non-faulty nodes.*

Proof: Since Lemma 5 and 6 are the same as Lemma 3 and 4, respectively, the proof of the agreement (C1) and validity property (C2) is literally the same as in Theorem 1. To justify the claimed time and message complexity, Figure 3 reveals that every node broadcasts three 1-bit messages per round in Phase 1 and 2; note that the broadcasts of $M[0]$ and $M[1]$ occur simultaneously in the same phase. Adding the single message broadcast by every round's king, $(f_a + f_s + f_o + f_m + 2)(3n + 1)$ messages are broadcast during the whole execution by not arbitrary faulty nodes. □

As for the Phase Queen algorithm, it would again be possible to omit the king's broadcast in the last round of this algorithm as well. Moreover, 2-bit messages could be used in the second phase, which would reduce the number of message broadcasts to $(2n + 1)(f_a + f_s + f_o + f_m + 2) - 1$.

5 Srikanth & Toueg Algorithm

In this section, we will analyze a hybrid version of the simple binary *Byzantine agreement* algorithm of [28] under the system model of Definition 1. A Byzantine agreement algorithm is used to consistently disseminate a single *transmitter's* value m to a set of *receivers*.[7] Within a fixed number of rounds, every non-faulty receiver p must *deliver* a value v_p ascribed to the transmitter that satisfies the following properties:

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

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

- m, if the transmitter is non-faulty,
- $\emptyset$, if the transmitter is manifest faulty,
- m or $\emptyset$, if the transmitter is omission faulty,
- the value actually sent, if the transmitter is symmetric faulty,
- unspecified, if the transmitter is arbitrary faulty.

Note that we will assume $\emptyset := 0$ here, since binary Byzantine agreement does not allow for a unique value $\emptyset \notin \{0, 1\}$.

The algorithm of [28] is built upon a *broadcast primitive* **broadcast**(p, m, r) and the corresponding **accept**(p, m, k), which are used to exchange messages (p, m, k) consisting of the identifier of the sending node p, some value $m \in \{0, 1\}$, and the round number k. The semantics of the broadcast primitive is fully captured by the following three properties:

(C) *Correctness:*
If a non-faulty node p executes **broadcast**(p, m, k) in round k, then every non-faulty node **accepts**(p, m, k) in the same round.

(U) *Unforgeability:*
If a non-faulty node p never executes **broadcast**(p, m, k), then no non-faulty node ever **accepts**(p, m, k).

(R) *Relay:*
If a non-faulty node **accepts**(p, m, k) in round $r \geq k$, then every other non-faulty node also **accepts**(p, m, k) in round $r + 1$ or earlier.

The usual way to implement the above broadcast primitive is by means of signed messages in conjunction with relaying, i.e., forwarding accepted messages to all other nodes in the system. An alternative solution that circumvents cryptography will be described later in Subsection 5.2. By plugging in this hybrid simulated broadcast primitive into the "generic" hybrid algorithm introduced in Subsection 5.1, a non-authenticated algorithm for Byzantine agreement will be obtained that tolerates hybrid node and link faults. Note that cryptographic implementations of the broadcast primitive cannot usually tolerate link-faults, since f_ℓ^s messages could get lost in every broadcast, thereby violating correctness (C).

5.1 Hybrid Byzantine Agreement

Figure 4 shows our hybrid version of the algorithm for binary Byzantine Agreement of [28]. It differs from the original algorithm only in that the number of faulty nodes t has been replaced by $f_a + f_s + f_o$ and that we hid relaying, which was done explicitly in [28, Fig. 1], within **broadcast**( ) for simplicity.

The algorithm proceeds in $f_a + f_s + f_o + f_c + 1$ rounds as defined by the broadcast primitive, where the only value broadcast by non-faulty nodes is 1; the value 0 is decided upon by default. A non-faulty transmitter broadcasts 1 in round 1 if agreement is to be reached on the value $m = 1$. Any non-faulty receiver p sets $v := 1$ at the end of round r and hence delivers 1, if it has accepted r messages broadcast by different nodes $P_r = \{p_1, \ldots, p_r\}$, one of which is the transmitter. Note that $p \notin P_r$ since p still has $v = 0$ during round r. The correctness and relay properties of the broadcast primitive ensure that all other correct nodes accept all messages $\in P_r$ plus the one broadcast by p by round $r + 1$, hence they will also deliver 1 by then.

If $m = 0$ is the value to be agreed upon, a non-faulty transmitter never broadcasts any message. By the unforgeability property, no non-faulty receiver ever accepts any mes-

[7] In this paper, all n nodes in the system are considered receivers; one specific node t also acts as the transmitter. In [25, 26], we considered the slightly different asymmetric "Byzantine Generals" setting [13] consisting of one transmitter-only node and $n - 1$ receivers.

if $p = t$ **then** $v := m$ **else** $v := 0$ **fi**
for r:=1 **to** $f_a + f_s + f_o + f_c + 1$ **do**
    / Round r */*
    if $v = 1$ **and** not yet **broadcast**(p,_,_) **then**
        broadcast$(p, 1, r)$
    fi
    if accept$(p_k, 1, r_k)$ from r-th distinct p_k, including t, **then**
        $v := 1$
    fi
done

Figure 4. *The hybrid binary Byzantine agreement algorithm of Srikanth & Toueg, code for node p*

sage originating from the transmitter. Consequently, no non-faulty node sets $v := 1$ and all must hence deliver 0 by default. Note that a manifest faulty transmitter is indistinguishable from a correct transmitter that tries to communicate $m = 0$, which explains why we assumed $\emptyset := 0$ for the validity property (B2).

By formalizing the above line of reasoning, the following Theorem 3 shows that our hybrid algorithm achieves agreement (B1) and validity (B2), provided that the broadcast primitive employed in the algorithm can cope with the node and link faults of Definition 1.

Theorem 3 (Hybrid Authenticated Algorithm) *Given an implementation of the broadcast primitive that guarantees (C), (U) and (R) under the system model of Definition 1, the hybrid algorithm of Figure 4 achieves Byzantine agreement within $f_a + f_s + f_o + f_c + 1$ rounds, where every non-faulty node calls* **broadcast**( ) *at most once during the whole execution. At least one non-faulty node calls* **accept**( ) *by round $f_a + f_s + f_o + f_c$.*

Proof: We first show that validity (B2) is achieved. Since (B2) is void in case of an arbitrary and omission faulty transmitter (recall that we are considering binary Byzantine agreement here), we only have to deal with the following cases:

(1) If the transmitter t is non-faulty and agreement is to be reached on 0, then t does not call **broadcast**( ). Hence, no non-faulty node will ever **accept**$(t, 1, \_)$ due to (U) and all non-faulty receivers will decide upon the default value $v = 0$. If agreement is to be reached on 1, a non-faulty transmitter faithfully calls **broadcast**$(t, 1, 1)$. By (C), all non-faulty nodes will accept this message in round 1 and set $v := 1$ as required.

(2) If the transmitter is manifest faulty, it either does not call **broadcast**( ) or, if it does, all of the resulting messages are discarded. Hence, those two cases are indistinguishable for the receivers and no non-faulty node will ever **accept**$(t, 1, \_)$ due to (U). Consequently, all non-faulty receivers will decide upon the default value $v = 0$ as required.

(3) If the transmitter is symmetric faulty, it can only[8] appear like a non-faulty transmitter that broadcasts the opposite value. Hence, case (1) above applies.

In order to show the agreement property (B1), it suffices to show that if one non-faulty node sets $v := 1$, then all other non-faulty nodes will do so as well; agreement on 0 occurs per default. Let q be the first non-faulty node that executes $v := 1$, and l be the round in which this happens. Then, q must have accepted at least l different messages containing a value of 1 in round l. We distinguish two cases: If $l < f_a + f_s + f_o + f_c + 1$, then any non-faulty node p will accept these messages due to (R) at most one round later as well. Since p will also accept q's broadcast of $(q, 1, l+1)$ in round $l+1$ due to (C), it will eventually end up with $v := 1$ as required.

If, on the other hand, $l = f_a + f_s + f_o + f_c + 1$, then the first node q that executed $v := 1$ does so in the last round $f_a + f_s + f_o + f_c + 1$, where it accepted at least $f_a + f_s + f_o + f_c + 1$ messages with a value of 1. However, at least one of these messages originates from a non-faulty node, which must have executed $v := 1$ in some round $l' < l$. This contradicts our assumption of q being the first one, so at least one non-faulty node must call **accept**( ) strictly before the last round. ☐

We finally note that it would be sufficient to use only $f_a + f_s + f_o + 1$ instead of $f_a + f_s + f_o + f_c + 1$ rounds in the algorithm of Figure 4, since the above proof of the agreement property (B1) deals only with messages that non-faulty node q received; they cannot come from one of the f_c manifest faulty nodes. However, our overly conservative choice secures validity of Theorem 3 also for the more general independent link fault model described in Remark 5 following Definition 1, where arbitrary link faults could create spurious messages even from a manifest faulty node.

5.2 Hybrid Simulated Broadcast Primitive

Algorithms based on authenticated messages, like the one of Figure 4, are simple, easy to understand and have superior fault-tolerance properties as well. However, cryptographic methods (see e.g. [27] for an overview) typically used for authentication are expensive operations, which cannot always be afforded in power/bandwidth-limited systems. Moreover, since there is always a non-zero probability of

[8] Recall from Definition 1 that a symmetric faulty node consistently sends information that is not detectably bad. A wrong transmitter identifier p or round number r in a message (p, m, r) would be detected at any receiver.

guessing or forging a node's signature, no present cryptographic authentication technique is unconditionally secure. Our algorithm's fault-tolerance properties, however, depend critically upon an uncompromised cryptosystem.

Those deficiencies make the non-authenticated implementation of the broadcast primitive developed by Srikanth & Toueg in [28] attractive. Instead of using cryptography, their *simulated broadcast primitive* relies upon the idea of letting all nodes witness a node's message broadcast. Using this approach, the algorithm of [28, Fig. 2] unconditionally guarantees the correctness (C), unforgeability (U) and relay (R) properties, provided that $n > 3f$ and at most f nodes are arbitrary faulty.

Figure 5 shows our hybrid variant of this simulated broadcast primitive, which works under the fault model of Definition 1. It proceeds in consecutive rounds consisting of two phases each, where two kinds of messages are exchanged: When calling **broadcast**(p, m, k) in round k, node p enters the code in phase $2k$. It first sends an message $(init, p, m, k)$ to all nodes in the system, which is answered by every witness with a message $(echo, p, m, k)$ in phase $2k+1$. A node calls **accept**(p, m, k) when it received the $n - f_a - f_s - f_o - f_m - f_\ell^s - f_\ell^r$-th $(echo, p, m, k)$ from different nodes.

Multiple rounds can occur if the initial broadcaster p is faulty. The additional rounds consist of two identical phases each, where a node that did not yet send $(echo, p, m, k)$ could get sufficient evidence that it should do so, which could in turn convince some nodes to accept in the following phase.

In the original algorithm [28, Fig. 2], any node q can terminate its instance of **broadcast**(p, m, k) after having called **accept**(p, m, k), since any other still active node in the system must have seen q's echo message by then. However, in order to deal with link faults, we had to change the original algorithm in a subtle but important way: Link faults can produce $f_\ell^{r,a}$ erroneous $(echo, p, m, k)$ messages per round at any non-faulty node, which would accumulate over multiple rounds. In Figure 4, we hence retransmit $(echo, p, m, k)$ in every additional phase, instead of sending it only once and remembering an earlier reception at all active nodes. Since other nodes may accept one round later than q according to the relay property (R), however, any node must retransmit $(echo, p, m, k)$ up to and including the phase following acceptance.

By adopting the analysis of [28], with guidance from [23], where we analyzed the closely related asynchronous consistent broadcasting primitive of [29], we can prove the following Theorem 4.

Theorem 4 (Simulated Broadcast) *Under the system model of Definition 1 with* $f_a, f_s, f_o, f_m, f_\ell^r, f_\ell^{r,a}, f_\ell^s, f_\ell^{s,a} \geq 0$ *and* $n > 3f_a + 2f_s + 2f_o + f_m + f_\ell^s + f_\ell^{s,a} + 2f_\ell^r + 2f_\ell^{r,a}$*, the simulated broadcast primitive of Figure 5 guarantees correctness (C), unforgeability (U), and relay (R). During* $0 < r \leq R$ *rounds, where* R *is an a priori upper bound upon the maximum number of rounds the algorithm is allowed to run, at most* $(2r-1)n + 1$ *broadcasts of* $(\log n + \log R + 2)$*-bit messages are performed by non-faulty nodes.*

/ Round k */*
/ Phase $2k$: Node p only (entered by* **broadcast**( )*) */*
send$(init, p, m, k)$ **to all nodes** [*including itself*];
/ Phase $2k+1$: All nodes */*
if received$(init, p, m, k)$ from p in phase $2k$ **then**
  send$(echo, p, m, k)$ **to all nodes** [*including itself*]
fi
if received$(echo, p, m, k)$ in phase $2k+1$
  $\geq n - f_a - f_s - f_o - f_m - f_\ell^s - f_\ell^r$ **then**
  accept(p, m, k)
fi
/ Round $r \geq k+1$ */*
/ Phase $2r$, $2r+1$ (same code): All nodes */*
if received$(echo, p, m, k)$ in previous phase
  $\geq n - 2f_a - f_s - 2f_o - f_m - f_\ell^s - 2f_\ell^r - f_\ell^{r,a}$
or sent$(echo, p, m, k)$ in previous phase **then**
  send$(echo, p, m, k)$ **to all nodes** [*including itself*]
fi
if accepted(p, m, k) in previous phase **then**
  terminate
fi
if received$(echo, p, m, k)$ in this phase
  $\geq n - f_a - f_s - f_o - f_m - f_\ell^r$ **then**
  accept(p, m, k)
fi

Figure 5. *The hybrid version of the simulated broadcast primitive of Srikanth & Toueg*

Proof: We show each of the three properties seperatly:

Correctness: Since p is non-faulty, $n - f_a - f_s - f_o - f_m - f_\ell^s$ non-faulty nodes receive the message $(init, p, m, k)$ in phase $2k$ and **send**$(echo, p, m, k)$ to all nodes in phase $2k+1$. Hence, every non-faulty node receives $(echo, p, m, k)$ from at least $n - f_a - f_s - f_o - f_m - f_\ell^s - f_\ell^r$ distinct nodes in phase $2k+1$ and thus **accepts**(p, m, k) in phase $2k+1$, i.e., in round k.

Unforgeability: The proof is by contradiction: Assume that some non-faulty node q **accepts**(p, m, k) in some round $r \geq k$, although the non-faulty node p did not **broadcast**(p, m, k) and hence did not send any $(init, p, m, k)$ message in phase $2k$. Nevertheless, $f_\ell^{s,a}$ non-faulty nodes might have received $(init, p, m, k)$ as the result of a "foreign" broadcast link fault at p, resulting in at most $f_a + f_s + f_\ell^{r,a} + f_\ell^{s,a}$ $(echo, p, m, k)$ messages that might arrive at any non-faulty node. Since $f_a + f_s + f_\ell^{r,a} + f_\ell^{s,a} < n - 2f_a - f_s - 2f_o - f_m - f_\ell^s - 2f_\ell^r - f_\ell^{r,a}$, no more than $f_\ell^{s,a}$ non-faulty node will ever execute **send**(echo,p,m,k) in line 17 of Figure 5. Since q executed **accept**(p, m, k), however, it must have received $(echo, p, m, k)$ from at least $n - f_a - f_s - f_o - f_m -$

$f_\ell^s - f_\ell^r > f_a + f_s + f_\ell^{r,a} + f_\ell^{s,a}$ nodes, which provides the required contradiction.

Relay: Let l be the phase in which the first node q accepts (p, m, k) in round r. Node q must have received at least $n - f_a - f_s - f_o - f_m - f_\ell^s - f_\ell^r$ *echo*-messages from different nodes in phase l, which implies that every other non-faulty node must have received at least $n - 2f_a - f_s - 2f_o - f_m - f_\ell^s - 2f_\ell^r - f_\ell^{r,a}$ *echo*-messages as well by Lemma 1. According to line 15 of Figure 5, every non-faulty node thus emits an *echo*-message in phase $l + 1$. Therefore, by phase $l + 1$ in round $r' \leq r + 1$, every non-faulty node will receive $(echo, p, m, k)$ at least $n - f_a - f_s - f_o - f_m - f_\ell^r$ times and will hence **accept**(p, m, k).

As far as the claimed message complexity of **broadcast**( ) is concerned, it is of course again impossible to bound the number of message broadcasts by (arbitrary) faulty nodes. Every node that faithfully executes the algorithm of Figure 5, however, executes one broadcast of $(echo, p, m, k)$ in each phase following the initial one, where only node p broadcasts $(init, p, m, k)$. Hence, during $0 < r \leq R$ rounds, at most $1 + n + 2(r - 1)n = (2r - 1)n + 1$ broadcasts of $(1 + \log n + 1 + \log R)$-bit messages are performed. This eventually completes the proof of Theorem 4. □

Note that the bound R upon the number of rounds required by Theorem 4 must be enforced externally, as in the Byzantine agreement algorithm of Figure 4. After all, the collusion of a faulty broadcaster with faulty witnesses could let the number of required rounds grow without bounds, as already noted in [28]. Still, this can only happen if no non-faulty node has accepted yet: By the above proof of the relay property, all non-faulty nodes accept by the end of the phase following acceptance of the first non-faulty node. Since any non-faulty node terminates the algorithm of Figure 5 in the phase after acceptance, all non-faulty nodes must have terminated by the end of the second phase following acceptance of the first non-faulty node, i.e., at most one round later.

With those preparations, it is not difficult to prove our major Theorem 5.

Theorem 5 (Hybrid Srikanth & Toueg) *Under the system model of Definition 1 with* $f_a, f_s, f_o, f_m, f_\ell^r, f_\ell^{r,a}, f_\ell^s, f_\ell^{s,a} \geq 0$ *and* $n > 3f_a + 2f_s + 2f_o + f_m + f_\ell^s + f_\ell^{s,a} + 2f_\ell^r + 2f_\ell^{r,a}$, *the algorithm of Figure 4 in conjunction with the simulated broadcast primitive of Figure 5 achieves binary Byzantine agreement in at most* $2(f_a + f_s + f_o + f_c + 1)$ *phases with at most* $(2(f_a + f_s + f_o + f_c + 1) - 1)n^2 + n$ *broadcasts of* $(2 \log n + 2)$*-bit messages from non-faulty nodes.*

Proof: The major part of our theorem follows immediately by combining Theorem 3 with Theorem 4.

As far as the claimed time and message complexity is concerned, we know from Theorem 3 that the first accept of a non-faulty node happens in round $f_a + f_s + f_o + f_c$ or earlier. Hence, any instance of the simulated broadcast primitive at a non-faulty node must terminate by round $R = f_a + f_s + f_o + f_c + 1 \leq n$ and could generate at most $(2(f_a + f_s + f_o + f_c + 1) - 1)n + 1$ broadcasts of $(2 \log n + 2)$-bit messages by Theorem 4. Since each of the at most n non-faulty nodes calls **broadcast**( ) at most once by Theorem 3, the proof of Theorem 5 is completed. □

Theorem 3 in conjunction with the lower bound $n > f_\ell^r + f_\ell^{r,a} + f_\ell^s + f_\ell^{s,a}$ for consensus developed in [25] finally allows us to derive a lower bound for the number of nodes required for establishing *Correctness, Unforgeability* and *Relay*, cp. [28]: If there was a broadcast primitive that needs only $f_\ell^r + f_\ell^{r,a} + f_\ell^s + f_\ell^{s,a}$ nodes, it would be possible to solve Byzantine agreement by means of the algorithm of Figure 4, thereby violating the lower bound for consensus. We thus have the following Theorem 6, which shows that the hybrid simulated broadcast primitive of Figure 5 is not optimal.

Theorem 6 (Lower Bound) *Achievement of Correctness (C), Unforgeability (U), and Relay (R) under the system model of Definition 1 requires* $n > f_\ell^r + f_\ell^{r,a} + f_\ell^s + f_\ell^{s,a}$ *nodes.* □

6 Assumption Coverage

To apply a deterministic fault model like the one of Definition 1 in practice, one has to address the question of assumption coverage. More specifically, for the particular system in mind, the *probability of failure* Q implied by a possible violation of the fault assumptions (f_a, f_s, f_o, f_m, f_ℓ^s, f_ℓ^r) needs to be evaluated. Note carefully that this is a mandatory step for any algorithm where safety depends upon non-violation of the fault model. It is particularly important for link faults, however, since Q increases with every message broadcast during the execution of the algorithm here: According to (A1^s) resp. (A1^r), no message broadcast resp. reception may suffer from more than f_ℓ^s resp. f_ℓ^r link faults. Given the fact that our algorithms send 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.

As in [25], we will derive an upper bound on the probability of failure Q_m of our algorithms 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, 17], 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.

In order to avoid a specific analysis for each of our algorithms, we will develop an expression for the total probability of failure Q_m of a generic m-round algorithm. We assume that this generic algorithm performs

- γ full message exchanges (= broadcasts of all nodes) in every round,
- β broadcasts in every round,
- α additional broadcasts,

where each broadcast is a full one, i.e., involves all $n-1$ remote nodes (the transmission to itself is assumed to be fault-free). So we have a total of

$$B_{tot} = \alpha + m \cdot (\beta + \gamma n) = \mathcal{O}(mn) \tag{2}$$

broadcasts during the execution of the generic algorithm. In any of those broadcasts resp. receptions in a full message exchange, it must be secured that the appropriate budget f_ℓ^s resp. f_ℓ^r of link faults is not exceeded. Since $f_\ell^s = f_\ell^r = f_\ell$ is the only reasonable choice in presence of independent link faults, the success probabilities for a single message broadcast/reception, namely,

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

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

$$p_{n-k} = \sum_{l=0}^{f_\ell} \binom{n-k}{l} p^l (1-p)^{n-k-l} \tag{3}$$

Abbreviating $q_{n-1} = 1 - p_{n-1}$, $M = \alpha + \beta m$, and $N = \gamma \cdot m \cdot n$ such that $B_{tot} = N + M$, the probability of success $P_m = 1 - Q_m$ reads

$$\begin{aligned} P_m &= p_{n-1}^{B_{tot}} = (1-q_{n-1})^{N+M} \\ &= \big(1-q_{n-1}\big)^M \cdot \Big(1 - \frac{Nq_{n-1}}{N}\Big)^N \\ &\geq \big(1-q_{n-1}\big)^M \cdot e^{-Nq_{n-1}} \Big(1 - \frac{N^2 q_{n-1}^2}{N}\Big), \end{aligned} \tag{4}$$

where we used the relation [35, p.242]

$$e^{-t} \geq (1 - t/n)^n \geq e^{-t}(1 - t^2/n) \tag{5}$$

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

$$Nq_{n-1} = \gamma m n q_{n-1} < 1, \tag{6}$$

which will be verified later, the application of the well-known facts $e^{-x} \geq 1 - x$ for $0 \leq x < 1$ and $(1-x)^k \geq 1 - kx$ for $0 \leq x \leq 1$ and $k \geq 1$ to (4) yields

$$\begin{aligned} P_m &\geq \big(1 - Mq_{n-1}\big) \cdot \big(1 - Nq_{n-1}\big) \cdot \big(1 - Nq_{n-1}^2\big) \\ &\geq 1 - (M+N)q_{n-1} - Nq_{n-1}^2 - MN^2 q_{n-1}^4. \end{aligned}$$

Consequently, the probability of failure Q_m can be bounded by

$$Q_m \geq B_{tot} q_{n-1} + N q_{n-1}^2 + MN^2 q_{n-1}^4, \tag{7}$$

so it only remains to derive a bound upon q_{n-1}. It is provided by the following Lemma 7.

Lemma 7 (Upper Bound) *For any p, n, f_ℓ,*

$$q_{n-1} \leq \binom{n-1}{f_\ell + 1} p^{f_\ell + 1} = \mathcal{O}\left(\frac{(np)^{f_\ell+1}}{(f_\ell+1)!}\right). \tag{8}$$

Proof: Using definition (3) in $q_{n-1} = 1 - p_{n-1}$, we find

$$\begin{aligned} G &= \frac{q_{n-1}}{(n-1)!} \\ &= \sum_{l=f_\ell+1}^{n-1} \frac{p^l}{l!} \cdot \frac{(1-p)^{n-l-1}}{(n-l-1)!} \\ &= \sum_{j=0}^{n-f_\ell-2} \frac{p^{j+f_\ell+1}}{(j+f_\ell+1)!} \cdot \frac{(1-p)^{n-f_\ell-2-j}}{(n-f_\ell-2-j)!} \\ &\leq \frac{p^{f_\ell+1}}{(f_\ell+1)!} \sum_{j=0}^{n-f_\ell-2} \frac{p^j}{j!} \cdot \frac{(1-p)^{n-f_\ell-2-j}}{(n-f_\ell-2-j)!}. \end{aligned} \tag{9}$$

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

$$\frac{q_{n-1}}{(n-1)!} \leq \frac{1}{(n-f_\ell-2)!} \cdot \frac{p^{f_\ell+1}}{(f_\ell+1)!}. \tag{10}$$

Combining this with

$$\frac{(n-1)!}{(n-f_\ell-2)!} = (n-1)(n-2)\cdots(n-f_\ell-1) = \mathcal{O}\big(n^{f_\ell+1}\big), \tag{11}$$

the result of our lemma follows. □

We can hence put everything together to obtain the following final Theorem 7.

Theorem 7 (Assumption Coverage) *For $np < 1$ sufficiently small, the probability of failure Q_m of the generic algorithm with B_{tot} broadcasts during the execution satisfies*

$$Q_m \leq Q'_m + \mathcal{O}\left(\frac{(Q'_m)^2}{B_{tot}}\right) = \mathcal{O}\left(\frac{nm \cdot (np)^{f_\ell+1}}{(f_\ell+1)!}\right), \tag{12}$$

where

$$Q'_m = B_{tot} \binom{n-1}{f_\ell+1} p^{f_\ell+1} \tag{13}$$

Proof: The result of Lemma 7 implies that condition (6), which was required for validity of (7), is satisfied since $np < 1$ is sufficiently small. We may hence plug in (8) into (7), which justifies the expression for (13). Recalling (2), the remainder term in (12) follows from $Nq_{n-1}^2 + MN^2 q_{n-1}^4 = \mathcal{O}(B_{tot} q_{n-1}^2)$. □

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 (12). Herein, $c \cdot \ell$ gives the number of nodes that must be added to cope with (a sufficiently small number) ℓ of additional link faults per node.

$$
\begin{aligned}
Q_m &= \mathcal{O}\left(n_0 m \frac{(n_0 p)^{f_\ell+\ell+1}}{(f_\ell+\ell+1)!}\big(1+\frac{c\ell}{n_0}\big)^{f_\ell+\ell+2}\right) \\
&= \mathcal{O}\left(n_0 m \frac{(n_0 p)^{f_\ell+1}}{(f_\ell+1)!} \cdot \frac{(n_0 p \cdot e^c)^\ell}{(f_\ell+2)\cdots(f_\ell+\ell+1)}\right)
\end{aligned}
$$

In the last step, we employed the well-known relation $(1 + t/(k+j))^k \leq e^t$ for $t \geq 0$ and $j \geq 0$ in conjunction with $k + j = n_0 \geq f_\ell + \ell + 2 = k$.

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

- marginally grows with n and hence with the number of any kind of node faults,
- decreases with the number of tolerated link faults f_ℓ (and with decreasing p, of course), since the last factor in the above $\mathcal{O}(.)$ 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 order to map the specific algorithms analyzed in the previous sections to the generic one, we summarize the relevant parameters in Table 1. To simplify the presentation, we restrict ourselves to the case $f_s = f_o = f_m = 0$, $f_a > 0$ and $f_\ell^r = f_\ell^{r,a} = f_\ell^s = f_\ell^{s,a} = f_\ell > 0$. The required number of nodes n as well as the required number of rounds m and the total number of broadcasts during the execution B_{tot} are taken from Theorems 1, 2 and 4.

Algorithm	$\boldsymbol{n}$	$\boldsymbol{m}$	$\boldsymbol{B_{tot}}$
Phase Queen	$6f_\ell + 4f_a + 1$	$2(f_a + 2)$	$(f_a + 2)(n + 1)$
Phase King	$6f_\ell + 3f_a + 1$	$3(f_a + 2)$	$(f_a + 2)(3n + 1)$
Sri. & Toueg	$6f_\ell + 3f_a + 1$	$2(f_a + 1)$	$(2f_a + 1)n^2 + n$

Table 1. *Parameters of the algorithms analyzed in the previous sections for $f_s = f_o = f_m = 0$, $f_a > 0$ and $f_\ell^r = f_\ell^{r,a} = f_\ell^s = f_\ell^{s,a} = f_\ell > 0$.*

In Tables 2 and 4, we give numerical values for the required number of nodes n of our algorithms for different values of f_a and f_ℓ according to Table 1. Tables 3, 5 and 6 provide the corresponding values of the (approximate) probability of failure Q'_m for $p = 10^{-2}$. Needless to say, much smaller values are obtained for smaller p, cp. [25,26].

7 Conclusions

This paper makes two contributions to the state of the art: We showed that (1) message-efficient deterministic consen-

f_ℓ	$f_a = 1$	$f_a = 2$	$f_a = 3$	$f_a = 4$	$f_a = 5$	$f_a = 6$
1	11	15	19	23	27	31
2	17	21	25	29	33	37
3	23	27	31	35	39	43
5	35	39	43	47	51	55
7	47	51	55	59	63	67
10	65	69	73	77	81	85

Table 2. *Minimal number of nodes $n = 6f_\ell + 4f_a + 1$ for the hybrid Phase Queen algorithm for different f_a, f_ℓ.*

f_ℓ	$f_a = 1$	$f_a = 2$	$f_a = 3$	$f_a = 4$	$f_a = 5$	$f_a = 6$
1	0.2	0.6	1	1	1	1
2	0.03	0.1	0.3	0.6	1	1
3	0.005	0.02	0.04	0.1	0.2	0.4
5	0.0001	0.0004	0.001	0.003	0.006	0.01
7	3.10^{-6}	0.00001	0.00003	0.00007	0.0002	0.0003
10	1.10^{-8}	4.10^{-8}	1.10^{-7}	3.10^{-7}	6.10^{-7}	1.10^{-6}

Table 3. *Value of (approximate) probability of failure Q'_m for $p = 0.01$ for the Phase Queen algorithm with minimal number of nodes.*

f_ℓ	$f_a = 1$	$f_a = 2$	$f_a = 3$	$f_a = 4$	$f_a = 5$	$f_a = 6$
1	10	13	16	19	22	25
2	16	19	22	25	28	31
3	22	25	28	31	34	37
5	34	37	40	43	46	49
7	46	49	52	55	58	61
10	64	67	70	73	76	79

Table 4. *Minimal number of nodes $n = 6f_\ell + 3 + f_a + 1$ for the hybrid Phase King and the Srikanth & Toueg algorithm for different f_a, f_ℓ.*

f_ℓ	$f_a = 1$	$f_a = 2$	$f_a = 3$	$f_a = 4$	$f_a = 5$	$f_a = 6$
1	0.3	1	1	1	1	1
2	0.07	0.2	0.4	0.9	1	1
3	0.01	0.03	0.07	0.2	0.3	0.5
5	0.0003	0.0009	0.002	0.004	0.008	0.01
7	9.10^{-6}	0.00002	0.00005	0.0001	0.0002	0.0004
10	4.10^{-8}	9.10^{-8}	2.10^{-7}	4.10^{-7}	8.10^{-7}	1.10^{-6}

Table 5. *Value of (approximate) probability of failure Q'_m for $p = 0.01$ for the Phase King algorithm with minimal number of nodes.*

f_ℓ	$f_a = 1$	$f_a = 2$	$f_a = 3$	$f_a = 4$	$f_a = 5$	$f_a = 6$
1	1	1	1	1	1	1
2	0.35672	1	1	1	1	1
3	0.09	0.3	1	1	1	1
5	0.004	0.01	0.04	0.09	0.2	0.4
7	0.0001	0.0005	0.001	0.003	0.006	0.01
10	8.10^{-7}	2.10^{-6}	6.10^{-6}	0.00001	0.00003	0.00006

Table 6. *Value of (approximate) probability of failure Q'_m for $p = 0.01$ for the Srikanth & Toueg algorithm with minimal number of nodes.*

sus in synchronous systems is possible even in presence of link faults, and (2) improved and adopted three existing algorithms and their analysis for hybrid node and link faults [as well as incomplete communication graphs]. More specifically, using the novel perception-based hybrid fault model of [25], which grants every node at most f_ℓ^r independent receive link faults (with at most $f_\ell^{r,a}$ non-omission faults among those) and f_ℓ^s broadcast link in each round, in addition to at most f_a, f_s, f_o, f_m arbitrary, symmetric, omission, and manifest node faults, the required number of nodes for the hybrid versions of the Phase Queen and Phase King consensus algorithms [9] and the Byzantine agreement algorithm of Srikanth & Toueg [28] were shown to satisfy

$$\begin{aligned} n &> 2f_\ell^s + 2f_\ell^r + 2f_\ell^{r,a} + 4f_a + 2f_s + 2f_o + f_m, \\ n &> 2f_\ell^s + 2f_\ell^r + 2f_\ell^{r,a} + 3f_a + 2f_s + 2f_o + f_m, \\ n &> f_\ell^s + f_\ell^{s,a} + 2f_\ell^r + 2f_\ell^{r,a} + 3f_a + 2f_s + 2f_o + f_m. \end{aligned}$$

Comparison with the (optimal) non-authenticated algorithm OMH of [25] reveals that those algorithms are less resilient to arbitrary link faults. This can be considered as the price for the algorithms' message efficiency. The resiliencies are equivalent, however, if only omission link faults are allowed ($f_\ell^{s,a} = f_\ell^{r,a} = 0$), although the resulting fault-tolerance degree $n > f_\ell^s + 2f_\ell^r$ does not match the lower bound $n > f_\ell^s + f_\ell^r$ developed in [25, 26] for this restricted case.

We also 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 decreases Q as long as $np < 1$. Consequently, all our algorithms can be used even in bandwidth-limited wireless systems, where link faults with loss probabilities up to $p = 10^{-2}$ are the dominating source of errors.

Part of our current/future work in this area is devoted to the few open issues in the area of synchronous consensus we are aware of: Apart from a hybrid version of the multi-valued consensus framework of Turpin & Coan [31], we are looking out for a message-efficient algorithm that has optimal resilience w.r.t. arbitrary link faults. Another major direction of our current research is the problem of asynchronous consensus under link faults.

References

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

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

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

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

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

[6] P. Berman and J. A. Garay. Asymptotically optimal consensus. In *Proc. 16th International Colloquium on Automata, Languages and Programming*, pages 80–94, July 1989.

[7] P. Berman and J. A. Garay. Efficient distributed consensus with $n = (3 + \epsilon)t$ processors. In *Proceedings of the 5th International Workshop on Distributed Algorithms*, pages 129–142, Oct. 1991.

[8] P. Berman, J. A. Garay, and K. J. Perry. Towards optimal distributed consenus. In *Proceedings of the 30th IEEE Symposium on the Foundations of Computer Science*, pages 410–415. IEEE Computer Society Press, 1989.

[9] P. Berman, J. A. Garay, and K. J. Perry. Asymptotically optimal distributed consensus. http://www.bell-labs.com/user/garay/#distributed-pub, 1992. (A combination of results from ICALP'89 [6], FOCS'89 [8], and WDAG'91 [7]).

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

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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).

[20] 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.

[21] 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`.

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

[23] 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.

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

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

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

[27] B. Schneier. *Applied Cryptography*. John Wiley & Sons, second edition, 1996.

[28] T. Srikanth and S. Toueg. Simulating authenticated broadcasts to derive simple fault-tolerant algorithms. *Distributed Computing*, 2:80–94, 1987.

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

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

[31] R. Turpin and A. B. Coan. Extending binary byzantine agreement to multivalued byzantine agreement. *Information Processing Letters*, 18(2):73–6, February 1984.

[32] 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).

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

[34] B. Weiss and U. Schmid. Consensus with written messages under link faults. In *20th Symposium on Reliable Distributed Systems (SRDS'01)*, Oct. 28–31, 2001. (to appear).

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

[36] 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.