Research Report 2/2004, Technische Universität Wien, Institut für Technische Informatik, 2004
http://www.auto.tuwien.ac.at/Projects/W2F/papers.html 28 March 2008

Failure Model Coverage under Transient Link Failures

Ulrich Schmid [1],

Technische Universität Wien, Institut für Technische Informatik, Embedded Computing Systems Group (E182/2), Treitlstraße 3, A-1040 Vienna

Abstract

In a number of former papers, we analyzed several consensus and Byzantine agreement algorithms under a novel hybrid failure model for synchronous distributed systems. It extends traditional process failure models by allowing every process in the system to commit up to f_ℓ^s send link failures and experience up to f_ℓ^r receive link failures per round, without being considered (process-)faulty. The present paper shows that this model—and hence our algorithms—can also be applied in systems with high transient link failures rates: Assuming that every link may fail independently with some probability p in every round, we derive the probability that the link failure bounds f_ℓ^s, f_ℓ^r are respected during the entire execution of some communications-expensive Byzantine agreement algorithm. This probability can in fact be made (almost) as large as desired by increasing f_ℓ^r and f_ℓ^s appropriately, even though n and hence the number of links that could fail systemwide increases as well. It turns out that our algorithms could even be applied in wireless systems, where link loss probabilities up to $p = 10^{-2}$ are common.

Key words: Fault-tolerant distributed systems, hybrid failure models, Byzantine agreement, transient link failures, assumption coverage

1 Motivation

It is well-known that link failures, as caused e.g. by receiver overruns (run out of buffers), unrecognized packets (synchronization errors), and CRC er-

Email address: `s@ecs.tuwien.ac.at` (Ulrich Schmid).

[1] This research is part of our W2F-project, see *http://www.auto.tuwien.ac.at/Projects/W2F/* for details. W2F is supported by the Austrian START programme Y41-MAT.

rors (data reception problems) occur much more frequent than node/process[2] failures in state-of-the-art wireline and, in particular, wireless networks. Simply trying to model link failures as failures of the sending and/or receiving processes typically violates any reasonable process failure model, however: Even a single link failure per process includes scenarios where all n processes must be considered faulty [10], and the probability of violating assumptions like "at most f out of $n \geq 2f + 1$ processes may be faulty" is inacceptably high even for moderate link failure rates.

Perfect communications abstractions, as provided by retransmission schemes in data-link layer protocols, have been invented to cope with this problem. Apart from increasing the protocol's complexity, however, such solutions cannot deal with persistent link failures. In addition, the required retransmissions usually impair the real-time properties of distributed algorithms running atop of the protocol. This is particularly true for synchronous systems, where the (fixed) round duration must be chosen a priori according to the maximum allowed number of retransmissions — even if no link failure occurs at run-time.

In [12,10], we introduced a novel hybrid *perception-based failure model* for synchronous systems with high link failure rates, which allows to employ resource redundancy (more processes) instead of time redundancy (retransmissions) when dealing with link failures: In addition to process failures, every process in the system may commit up to f_ℓ^s send link failures and up to f_ℓ^r receive link failures in a round-by-round fashion [4] here, without being considered faulty. Typically, $f_\ell^s = f_\ell^r = f_\ell$, such that up to nf_ℓ link failures could occur system-wide in any round. Several consensus and Byzantine agreement algorithms were analyzed sucessfully under this failure model [10,2]. Unlike in case of unrestricted link failures [5] and mobile failures [9], where deterministic consensus is known to be impossible, increasing the number of processes n by some small integer multiples of f_ℓ^s and f_ℓ^r was found to be sufficient to deal with link failures here.

Still, in order to apply any failure model in real applications, one has to address the question of assumption coverage: Given some particular system implementation, what is the probability Q that some failure bound is violated at runtime? Computing this probability of violation is in fact a mandatory step for any algorithm where safety depends upon compliance to the failure model. Fortunately, since the probability of a component failure is usually independent of the particular execution (except of its duration), Q can be computed by classic reliability modeling techniques in case of process failure models.

This is not necessarily true for our link failure model, however: Consider wire-

[2] The terms process and node failures are used synonymously in this paper.

less systems, where individual links typically fail independently of each other due to local interference/fading, for example. Q increases with every message broadcast during the execution of a distributed algorithm here: According to our failure model, no correct process may suffer from more than f_ℓ^s resp. f_ℓ^r link failures in any message broadcast resp. reception *during the entire execution*. Given the fact that typical consensus algorithms send many messages, the question arises whether Q can eventually be made as small as desired—by choosing suitable values of f_ℓ^s and f_ℓ^r—at all. More generally, does it always pay off to increase f_ℓ^s, f_ℓ^r for decreasing Q, given that n and hence the total number of links in the system that could fail also increases? We do not know of any existing work that addresses such issues.

The present paper answers our questions in the affirmative: Following a brief overview of our perception-based failure model in Section 2 and the exponential algorithm OMH in Section 3, we compute the probability of violation Q for a system where links fail independently with some fixed probability p. Section 4 contains the derivation of Q for OMH, Section 5 is devoted to a modified version $\overline{\text{OMH}}$ that packs multiple concurrent messages of OMH into a single one. In Section 6, we derive generic expressions for Q that can be applied to any polynomial algorithm like the ones in [2]. A discussion of the consequences of our results in Section 7 eventually concludes our paper.

2 System and Failure Model

We consider a distributed system of n *nodes*, which execute one or more *processes* each that communicate via a fully connected point-to-point network. In case of multiple processes per node, it is assumed that every such process has a unique peer at every other node (which need not be the same in different rounds, however). All processes executed by a non-faulty node must be non-faulty; an arbitrary number of processes executed by a faulty node may be faulty as specified. All links between processes are bidirectional, consisting of two unidirectional channels (not necessarily FIFO) that may be hit by failures independently.

The entire system is synchronous in the usual sense, that is, the distributed computation evolves in a series of rounds at all processes in lock-step. In every round, all processes (or a subset of those) "broadcast" a single value to each other; the received values are used to compute the value to be broadcast in the next round. Broadcasting just means non-atomic, non-reliable sending of the same message to all receivers (including the transmitter) here. In case of failures, inconsistent reception of a broadcast message may occur. It is the purpose of the failure model to cleanly specify the failures that are to be tolerated by the distributed algorithm in question.

The perception-based failure model given in Definition 1 below is a hybrid one, which distinguishes several categories of both process and link failures. Consult [10] for a thorough discussion, including a detailed relation to existing work, of such failure models.

Definition 1 (System Model) *We consider a synchronous distributed system consisting of n nodes executing one or more processes, which are interconnected by a fully connected point-to-point network made up of pairs of unidirectional channels.*

(P1) In any round, there may be at most f_a, f_s, f_o, and f_m arbitrary, symmetric, omission, and manifest faulty nodes. The failure modes of their processes are defined via the set $rvals(V_p, p)$ of admissible values delivered by correct receivers when process p attempts to send them the value V_p:

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

A process that is manifest or omission faulty is called benign *faulty. A process is* consistent *if it is either non-faulty, manifest faulty, or symmetric faulty.*

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

(A1^r) If all processes $p_i \in \mathcal{S}$ of a set of [faulty or non-faulty] processes send a message containing V_{p_i} to some correct receiver process q in a round, at most f_ℓ^r of the delivered values $V_q^{p_i}$ may differ from the admissible receive values in $rvals(V_{p_i}, p_i)$. Let $f_\ell^{ra} \leq f_\ell^r$ be the maximum number of non-omissive, i.e., non-empty and hence value faulty, $V_q^{p_i}$ among those.

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

(A3) The absence of a message from sender p can be detected at any receiver q at the end of a round, which leads to $V_q^p = \emptyset$.

Remarks:

(1) A single round consists of (1) all the processes' local computations based upon the values received in the previous round, (2) the broadcasts (= successive sends) of the resulting messages according to (A1^s), and (3) the reception of those messages according to (A1^r).
(2) Every process has an individual "budget" f_ℓ^r (resp. f_ℓ^s) of link failures that may hit arbitrary inbound (resp. outbound) links. Note that they may be different in different rounds.
(3) Properties (A1^s) and (A1^r) must hold simultaneously in any execution and are considered independently of each other. Since every send link failure must be counted as a receive link failure at the respective receiver, however, we must have $nf_\ell^s = nf_\ell^r$ and $nf_\ell^{sa} = nf_\ell^{ra}$. Hence, without restricting link failure patterns, the only possible parameter settings are $f_\ell^s = f_\ell^r$ and $f_\ell^{sa} = f_\ell^{ra}$. An alternative setting is exploited in [10,11], however.

The coverage analysis in this paper will be based upon a very simple probabilistic model of link failures, which assumes that the probability of losing or corrupting a single message on a single link is $0 < p < 1$, and that individual link failures occur independently of each other and of process failures. Moreover, we consider only link omission failures, i.e., $f_\ell^{sa} = f_\ell^{ra} = 0$, and assume $f_\ell^s = f_\ell^r = f_\ell > 0$. Despite of its simplicity, this model is commonly used in practice, see e.g. [3,8], 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 failures in homogeneous system architectures. Persistent and, in particular, correlated failures are of course beyond its scope.

The question addressed in this paper is whether the deterministic failure model of Definition 1 can reasonably be applied in this setting, that is, has reasonable coverage. Note that our coverage analysis will consider link failures only: We do not account for violations of process failure assumptions in our analysis, i.e., we assume that the actual number of arbitrary, symmetric, omission and manifest faulty processes is always at most f_a, f_s, f_o and f_m, respectively. Link failures are hence viewed as additional incidents that happen to the messages sent on a link, irrespectively of whether they are correct messages from a non-faulty process or incorrect/missing ones from a faulty process.

3 Hybrid Oral Messages Algorithm

The Hybrid Oral Messages algorithm OMH [7] is a "Byzantine generals" algorithm, where the value v of a dedicated *transmitter* is to be disseminated to the remaining $n-1$ *receivers*; the transmitter already knows its value. Eventually, every non-faulty process p—including the transmitter—must *deliver* a

value v_p ascribed to the transmitter that satisfies the following properties:

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

(B2) (*Validity*): If process p is non-faulty, the value v_p delivered by p is
- v, if the transmitter is non-faulty,
- $\emptyset$, if the transmitter is manifest faulty,
- v 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.

A fully-fledged consensus algorithm is obtained by using a separate instance of Byzantine agreement for disseminating any process'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 $\emptyset$, $R(\emptyset)$, $R(R(\emptyset))$, $R(R(R(\emptyset)))$, ... must actually be distinguishable here; for each legitimate value v, we can allow $R(v) = R^{-1}(v) = v$.
- The *hybrid-majority* of a set $\mathcal{V}$ of values provides the majority of all non-$\emptyset$ values in $\mathcal{V}$. If no majority exists, the default value $R(\emptyset)$ is returned.

Consult [7] for a detailed discussion of the above primitives and the operation of OMH in general.

Definition 2 (Algorithm OMH [7]) *The Hybrid Oral Message algorithm OMH is defined recursively as follows:*

OMH(0):

(1) The transmitter t sends its value v to every receiver and delivers $v_t = v$.
(2) Every receiver p delivers the value v_p received from the transmitter, or the value $\emptyset$ if a missing or manifestly erroneous value was received.

OMH(m), $m > 0$**:**

(1) The transmitter t sends its value v to every receiver and delivers $v_t = v$.
(2) For every p, let w_p be the value receiver p receives from the transmitter, or $\emptyset$ 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 receivers.

(3) For every p and $q \neq p$, let w_p^q be the value receiver p delivers as the result of OMH($m-1$) initiated by receiver q in step 2 above. Every receiver p

calculates the hybrid-majority value of all values w_p^q and its own value $w_p^p = R(w_p)$, and applies R^{-1} to that value. This result is delivered as v_p.

Note that 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-process, $m+1$-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 $n-1$-process, m-round consensus algorithm.

By adopting and extending the analysis of [7] to the perception-based failure model of Definition 1, we showed in [12,10] that OMH satisfies (B1) and (B2) according to the following Theorem 3.

Theorem 3 (Agreement & Validity OMH) *For any $m \geq f_a + f_o + \min\{1, f_\ell^s\}$ and f_a, f_o, f_s, f_m, f_ℓ^s, f_ℓ^r, $f_\ell^{ra} \geq 0$, the algorithm OMH(m) satisfies agreement (B1) and validity (B2) if $n > 2f_\ell^s + f_\ell^r + f_\ell^{ra} + 2(f_a + f_s) + f_o + f_m + m$.*

4 Assumption Coverage OMH

In this section, we will derive a bound on the probability of violation Q_m of our failure model if the algorithm OMH(m) of Section 3 is employed in the probabilistic link failure setting of Section 2. Since OMH has exponential message complexity, this may be seen as a worst case scenario for assumption coverage; polynomial algorithms like the ones analyzed in [2] guarantee a considerably smaller probability of violation, see Section 6. Numerical results for a few parameter settings will show that Q_m can indeed be made arbitrarily small by sufficiently increasing n.

Our detailed analysis provides a bound on the probability $P_m = 1 - Q_m$ of non-violation of the failure bounds $f_\ell^s = f_\ell^r = f_\ell$ during a single execution of OMH(m) as follows: We start with computing the binomial probability p_{n-k} of a non-violating broadcast transmission/reception to/from $n-k$ peer processes. Next, we analyze the (exponential) communications requirements of OMH and compute the joint probability that all broadcasts/receptions during an execution are non-violating. This gives an expression for Q_m consisting of two sums involving p_{n-k}, which are evaluated in Lemma 4 and 5. This finally leads to the expression of Q_m given by Theorem 6.

The non-violation probabilities for a single message broadcast/reception, namely,

$$p_{n-k}^s = \mathbf{Prob}\{\leq f_\ell^s \text{ failures in a single broadcast to } n-k \text{ receivers}\}$$
$$p_{n-k}^r = \mathbf{Prob}\{\leq f_\ell^r \text{ failures in a single reception from } n-k \text{ senders}\}$$

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:

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

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

$$\begin{aligned} P_m = \mathbf{Prob}\{&\text{All broadcasts in OMH}(m),\ldots,\text{OMH}(1) \text{ have} \leq f_\ell \text{ link failures} \\ &\wedge \text{ all receptions in OMH}(0) \text{ have} \leq f_\ell \text{ link failures each}\}. \end{aligned} \tag{2}$$

Note from OMH's definition in Section 3 that (A1^r) in Definition 1 is required in OMH(0) only; see OMH's analysis in [10] for details.

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

OMH(m)	**# concurrent 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 OMH(1). Assuming that the "self-reception" by the transmitter of OMH(0) is always failure-free, there remain $n-m-1$ "true" message receptions by any receiver of OMH(0).

Abbreviating $n_k = [n-1]_k$, (2) translates to

$$
\begin{aligned}
P_m &= \prod_{k=0}^{m-1} p_{n-k-1}^{n_k} \cdot p_{n-m-1}^{n_m} = \prod_{k=0}^{m} p_{n-k-1}^{n_k} \\
&= \prod_{k=0}^{m} (1-q_{n-k-1})^{n_k} \quad \text{with } q_{n-k} = 1 - p_{n-k} \\
&= \prod_{k=0}^{m} \Big(1 - \frac{n_k q_{n-k-1}}{n_k}\Big)^{n_k} \\
&\geq e^{-\sum_{k=0}^{m} n_k q_{n-k-1}} \prod_{k=0}^{m} \Big(1 - \frac{(n_k q_{n-k-1})^2}{n_k}\Big),
\end{aligned}
\tag{3}
$$

where we used the relation [13, p. 242]

$$
e^{-t} \geq (1-t/n)^n \geq e^{-t}(1-t^2/n) \tag{4}
$$

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

$$
\sum_{k=0}^{m} n_k q_{n-k-1} < 1, \tag{5}
$$

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

$$
\begin{aligned}
P_m &\geq e^{-\sum_{k=0}^{m} n_k q_{n-k-1} + \sum_{k=0}^{m} \log\Big(1 - \frac{(n_k q_{n-k-1})^2}{n_k}\Big)} \\
&\geq e^{-\sum_{k=0}^{m} n_k q_{n-k-1} - \sum_{k=0}^{m} \sum_{j\geq 1} \frac{(\sqrt{n_k} q_{n-k-1})^{2j}}{j}} \\
&\geq e^{-\sum_{k=0}^{m} n_k q_{n-k-1} - \sum_{j\geq 1} \frac{(\sum_{k=0}^{m} \sqrt{n_k} q_{n-k-1})^{2j}}{j}} \\
&\geq \Big(1 - \sum_{k=0}^{m} n_k q_{n-k-1}\Big)\Big(1 - (\sum_{k=0}^{m} \sqrt{n_k} q_{n-k-1})^2\Big) \\
&\geq 1 - \sum_{k=0}^{m} n_k q_{n-k-1} - \Big(\sum_{k=0}^{m} \sqrt{n_k} q_{n-k-1}\Big)^2
\end{aligned}
\tag{6}
$$

$$\geq 1 - \sum_{k=0}^{m} n_k q_{n-k-1} - \Big(\sum_{k=0}^{m} n_k q_{n-k-1}\Big)^2. \tag{7}$$

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

$$\sum_{k=0}^{m} [n-1]_k q_{n-k-1} = (n-1)! \sum_{k=0}^{m} \frac{q_{n-k-1}}{(n-k-1)!} \tag{8}$$

and, if the more accurate lower bound (6) is addressed,

$$\sum_{k=0}^{m} \sqrt{[n-1]_k} q_{n-k-1} = \sqrt{(n-1)!} \sum_{k=0}^{m} \frac{q_{n-k-1}}{\sqrt{(n-k-1)!}}. \tag{9}$$

The required bound for the dominating term (8) follows from the following Lemma 4.

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

$$\begin{aligned} G_m &= \sum_{k=0}^{m} \frac{q_{n-k-1}}{(n-k-1)!} \\ &\leq \Big(1 + \frac{1}{n-m-f_\ell-2}\Big) \cdot \frac{q_{n-m-1}}{(n-m-1)!} \qquad (10) \\ &\leq \Big(1 + \frac{1}{n-m-f_\ell-2}\Big) \cdot \frac{1}{(n-m-f_\ell-2)!} \cdot \frac{p^{f_\ell+1}}{(f_\ell+1)!}. \qquad (11) \end{aligned}$$

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.,

$$\begin{aligned} q_{n-k} &= \sum_{l=f_\ell+1}^{n-k} \binom{n-k}{l} p^l (1-p)^{n-k-l} \qquad (12) \\ &= \frac{(n-k)!}{(f_\ell)!\,(n-k-f_\ell-1)!} \cdot \int_0^p t^{f_\ell} (1-t)^{n-k-f_\ell-1}\, dt. \qquad (13) \end{aligned}$$

Hence,

$$G_m = \frac{1}{(f_\ell)!} \int_0^p t^{f_\ell} \sum_{k=0}^{m} \frac{(1-t)^{n-k-f_\ell-2}}{(n-k-f_\ell-2)!}\, dt, \tag{14}$$

which involves

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

since $0 \le t \le p$. Plugging the above expression into (14), we obtain

$$
G_m \le \frac{1+\frac{1}{n-m-f_\ell-2}}{(f_\ell)!\,(n-m-f_\ell-2)!}\int_0^p t^{f_\ell}(1-t)^{n-m-f_\ell-2}\,dt \tag{15}
$$

from where the major result (10) of our theorem follows by recalling (13).

To establish (11), we use the definition (12) of q_{n-k-1} to find

$$
\begin{aligned}
g &= \frac{q_{n-m-1}}{(n-m-1)!} \\
&= \sum_{l=f_\ell+1}^{n-m-1}\frac{p^l}{l!}\cdot\frac{(1-p)^{n-m-l-1}}{(n-m-l-1)!} \\
&= \sum_{j=0}^{n-m-f_\ell-2}\frac{p^{j+f_\ell+1}}{(j+f_\ell+1)!}\cdot\frac{(1-p)^{n-m-f_\ell-2-j}}{(n-m-f_\ell-2-j)!} \\
&\le \frac{p^{f_\ell+1}}{(f_\ell+1)!}\sum_{j=0}^{n-m-f_\ell-2}\frac{p^j}{j!}\cdot\frac{(1-p)^{n-m-f_\ell-2-j}}{(n-m-f_\ell-2-j)!}.
\end{aligned} \tag{16}
$$

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

$$
\frac{q_{n-m-1}}{(n-m-1)!} \le \frac{1}{(n-m-f_\ell-2)!}\cdot\frac{p^{f_\ell+1}}{(f_\ell+1)!} \tag{17}
$$

which completes the proof of our lemma. □

Remarks:

(1) Lemma 4 reveals that the sum (8) 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 violation.

(2) By subtracting $q_{n-m-1}/(n-m-1)!$ from both sides of (10), and multiplying by $(n-1)!$ according to (8), it is easy to see that (10) also implies monotonicity of

$$\frac{q_{n-k-1}}{(n-k-1)!} \leq \frac{q_{n-m-1}}{(n-m-1)!} \tag{18}$$

$$[n-1]_k q_{n-k-1} \leq [n-1]_m q_{n-m-1} \tag{19}$$

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

(3) The bound given by (11) is reasonably small—and also accurate, cp. the derivation starting with (16)— only if $np < 1$ is sufficiently small, since the ultimately required quantity $(n-1)!G_m$ that must be < 1 according to (5) has order $\mathcal{O}(n^m (np)^{f_\ell+1}/(f_\ell+1)!)$.

By a very similar proof, it is not difficult to prove a similar Lemma 5 related to the square-rooted sum (9). Since it is only used to improve the remainder $\mathcal{O}$-term in Theorem 6 below, its proof will be left as an exercise to the reader.

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

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

□

By virtue of those results, we can establish the following Theorem 6.

Theorem 6 (Assumption Coverage OMH) *For $n - m - f_\ell - 2 \geq 1$ and $np < 1$ sufficiently small, the probability of violation Q_m of OMH(m) satisfies*

$$Q_m \leq Q'_m + \mathcal{O}\Big(\frac{(Q'_m)^2}{[n-f_\ell-2]_m}\Big) = \mathcal{O}\Big(n^m \frac{(np)^{f_\ell+1}}{(f_\ell+1)!}\Big), \tag{20}$$

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 (8) resp. (9), the result of Lemma 4 resp. 5 immediately yields $(n-1)!G_m \le Q'_m$ resp.

$$\begin{aligned} R''_m &= (n-1)!H_m^2 \\ &\le \Big(1 + \frac{1}{\sqrt{n-m-f_\ell-2}}\Big)^2 \cdot [n-1]_{f_\ell+1} \cdot [n-1]_{m+f_\ell+1} \cdot \Big(\frac{p^{f_\ell+1}}{(f_\ell+1)!}\Big)^2 \\ &\le \mathcal{O}\Big(\frac{(Q'_m)^2}{[n-f_\ell-2]_m}\Big), \end{aligned} \tag{21}$$

where the last bound is easily confirmed by comparing R''_m with $(Q'_m)^2$. Recalling the lower bound (6) on the probability of non-violation, (20) is established by straightforward upper bounding. Note that (6) is only guaranteed to hold when (5) holds, which is secured by $np < 1$ sufficiently small according to Remark 3 on Lemma 4. □

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 3.

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 violation Q_m for $p = 0.1$.

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 violation Q'_m for $p = 0.01$.

5 Assumption Coverage $\overline{\text{OMH}}$

Whereas the probability of violation 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 failure model. As an example, we consider the algorithm $\overline{\text{OMH}}$ that results from combining all messages that a process 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 processes—that are broadcast to $n-k-1$ receivers. Clearly, during the whole execution of $\overline{\text{OMH}}(m)$, any process broadcasts only m messages, except for the initial

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 violation 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 violation Q'_m for $p = 0.000001$.

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 failure in a correlated fashion: If f_ℓ^s of the combined messages are lost in the broadcast of a single sender, any affected receiver loses the message for all instances of $\text{OMH}(m-k)$. This situation, however, could also occur when link failures are independent for all instances of $\text{OMH}(m-k)$.

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

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

where the bound is even valid if all processes (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 (7) and use the same line of reasoning as before to show the following Theorem 7.

Theorem 7 (Assumption Coverage $\overline{\text{OMH}}$) *For $n - m - f_\ell - 2 \geq 1$ and $np < 1$ sufficiently small, the probability of violation $\overline{Q}_m$ of $\overline{OMH}(m)$ satisfies*

$$\overline{Q}_m \leq \overline{Q}'_m + \mathcal{O}((\overline{Q}'_m)^2) = \mathcal{O}\Big(\frac{n}{f_\ell + 3} \cdot \frac{(np)^{f_\ell+1}}{(f_\ell+1)!}\Big), \tag{22}$$

where

$$\overline{Q}'_m = \frac{[n+1]_{f_\ell+3} - [n-m]_{f_\ell+3}}{f_\ell + 3} \cdot \frac{p^{f_\ell+1}}{(f_\ell+1)!}. \tag{23}$$

Proof: Applying (6) with $n_k = n-k$ reveals that $\overline{P}_m$ and hence the probability of violation $\overline{Q}_m$ is dominated by

$$\overline{Q}''_m = \sum_{k=0}^{m} (n-k) q_{n-k-1} = \sum_{k=0}^{m} (n-k)! \frac{q_{n-k-1}}{(n-k-1)!}, \tag{24}$$

Using the upper bound (17) with $m = k$ established in the proof of Lemma 4, we find

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

where we employed the well-known identity [6, p.54.(11)] $\sum_{k=0}^{n} \binom{k}{m} = \binom{n+1}{m+1}$. Recalling (7), which is again valid for $np < 1$ sufficiently small, and applying some simple majorizations on (23) 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, (22) follows. □

Comparing (20) and (22) clearly shows that the probability of violation $\overline{Q}_m$ 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 larger than the ones of OMH—it is 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 violation $\overline{Q}_m$ 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 violation $\overline{Q}'_m$ for $p = 0.01$.

6 Assumption Coverage Polynomial Algorithms

In this section, we will derive an upper bound on Q_m for a "generic" m-round distributed algorithm, which covers all the polynomial consensus and Byzantine agreement algorithms analyzed in [2]. 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).

The probability P_s that none of the n processes in the system experiences more than f_ℓ link failures on its outgoing links in a single full broadcast is $P_s = p_{n-1}(f_\ell)^n$, since the failures on the outgoing links of different processes are independent. Similarly, the probability P_r that none of the n processes in a single full broadcast experiences more than f_ℓ link failures on its incoming links is also $P_r = p_{n-1}(f_\ell)^n$.

The probability P_{sr} that none of the n processes in the system experiences more than f_ℓ link failures on its outgoing links <u>and</u> no more than f_ℓ link failures on its incoming links is not just the product of P_s and P_r, however, since they are not independent. However, $P_{sr} = P_{r|s}P_s$, where $P_{r|s}$ denotes the conditional probability that no process perceives more than f_ℓ link failures on its incoming links, conditioned on the fact that no process experiences more than f_ℓ link failures on its outgoing links. Since trivially $P_{r|s} \geq P_r$, we obtain $P_{sr} \geq P_s P_r = p_{n-1}(f_\ell)^{2n}$.

For simplicity, we will account for this fact by just doubling the total number of broadcasts

$$B_{tot} = \alpha + m \cdot (\beta + \gamma n) = \mathcal{O}(mn) \tag{25}$$

performed during the m rounds of execution of the generic algorithm when using our formulas, see Table 9.

From the probability p_{n-1} that a single full broadcast does not violate our link failure assumptions, recall (1), it follows by independence that $P_m = p_{n-1}^{B_{tot}}$, where $P_m = 1 - Q_m$ denotes the probability that all B_{tot} broadcasts during the execution of the generic algorithm respect the link failure model. Hence, all that remains to be done in order to compute Q_m is to compute an expression for P_m.

Abbreviating $q_{n-1} = 1 - p_{n-1}$, $M = \alpha + \beta m$, and $N = \gamma \cdot m \cdot n$ such that $B_{tot} = N + M$, we obtain

$$\begin{aligned} P_m &= p_{n-1}^{B_{tot}} = (1 - q_{n-1})^{N+M} \\ &= (1 - q_{n-1})^M \cdot \Big(1 - \frac{N q_{n-1}}{N}\Big)^N \\ &\geq (1 - q_{n-1})^M \cdot e^{-N q_{n-1}} \Big(1 - \frac{N^2 q_{n-1}^2}{N}\Big), \end{aligned} \tag{26}$$

where we again used the relation [13, p.242]

$$e^{-t} \geq (1 - t/n)^n \geq e^{-t}(1 - t^2/n) \tag{27}$$

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

$$N q_{n-1} = \gamma m n q_{n-1} < 1, \tag{28}$$

which will be verified later, the application of the already 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 (26) yields

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

Consequently, the probability of failure $Q_m = 1 - P_m$ can be bounded by

$$Q_m \leq B_{tot} q_{n-1} + N q_{n-1}^2 + M N^2 q_{n-1}^4, \tag{29}$$

so it only remains to derive a bound upon q_{n-1}. It is provided by the following Lemma 4.

Lemma 8 (Upper Bound) *For any $0 < p < 1$, $n \geq 2$, $0 \leq f_\ell \leq n - 2$,*

$$q_{n-1} \leq \binom{n-1}{f_\ell + 1} p^{f_\ell + 1} = \mathcal{O}\left(\frac{(np)^{f_\ell + 1}}{(f_\ell + 1)!}\right). \tag{30}$$

Proof: Using expression (1) for $k = 1$ in $q_{n-1} = 1 - p_{n-1}$, we find

$$G = \frac{q_{n-1}}{(n-1)!}$$

$$
\begin{aligned}
&= \sum_{l=f_\ell+1}^{n-1} \frac{p^l}{l!} \cdot \frac{(1-p)^{n-l-1}}{(n-l-1)!} \qquad (31)\\
&= \sum_{j=0}^{n-f_\ell-2} \frac{p^{j+f_\ell+1}}{(j+f_\ell+1)!} \cdot \frac{(1-p)^{n-f_\ell-2-j}}{(n-f_\ell-2-j)!}\\
&\leq \frac{p^{f_\ell+1}}{(f_\ell+1)!} \sum_{j=0}^{n-f_\ell-2} \frac{p^j}{j!} \cdot \frac{(1-p)^{n-f_\ell-2-j}}{(n-f_\ell-2-j)!}.
\end{aligned}
$$

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

$$
\frac{q_{n-1}}{(n-1)!} \leq \frac{1}{(n-f_\ell-2)!} \cdot \frac{p^{f_\ell+1}}{(f_\ell+1)!}. \qquad (32)
$$

Combining this with

$$
\frac{(n-1)!}{(n-f_\ell-2)!} = (n-1)(n-2)\cdots(n-f_\ell-1) = \mathcal{O}(n^{f_\ell+1}), \qquad (33)
$$

the result of our lemma follows. □

We can hence put everything together to obtain the following Theorem 9.

Theorem 9 (Assumption Coverage Generic) *For $np < 1$ sufficiently small, the probability of failure Q_m of the generic algorithm with B_{tot} broadcasts during the execution satisfies*

$$
Q_m \leq Q'_m + \mathcal{O}\left(\frac{(Q'_m)^2}{B_{tot}}\right) = \mathcal{O}\left(\frac{nm \cdot (np)^{f_\ell+1}}{(f_\ell+1)!}\right), \qquad (34)
$$

where

$$
Q'_m = B_{tot}\binom{n-1}{f_\ell+1} p^{f_\ell+1} \qquad (35)
$$

Proof: The result of Lemma 8 implies that condition (28), which was required for validity of (29), is satisfied since $np < 1$ can be chosen sufficiently small. We may hence plug in (30) into (29), which justifies the expression for (35). Recalling (25), the remainder term in (34) follows easily from $Nq_{n-1}^2 + MN^2q_{n-1}^4 = \mathcal{O}(B_{tot}q_{n-1}^2)$. □

In order to evaluate Q_m for the specific polynomial consensus and Byzantine agreement algorithms analyzed in [2], we summarize the relevant parameters in Table 9. To simplify the presentation, we restrict ourselves to the case

$f_s = f_o = f_m = 0$, $f_a > 0$ and $f_\ell^r = f_\ell^{ra} = f_\ell^s = f_\ell^{sa} = f_\ell > 0$. Note that we simply doubled the algorithms' actual total number of rounds when computing B_{tot} to account for both send and receive link failures in full broadcasts.

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

Table 9
Parameters of the algorithms analyzed in the previous sections for $f_s = f_o = f_m = 0$, $f_a > 0$ and $f_\ell^r = f_\ell^{ra} = f_\ell^s = f_\ell^{sa} = f_\ell > 0$.

In Tables 10 and 12, 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 9. Tables 11, 13 and 14 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.

$\boldsymbol{f_\ell}$	$\boldsymbol{f_a = 1}$	$\boldsymbol{f_a = 2}$	$\boldsymbol{f_a = 3}$	$\boldsymbol{f_a = 4}$	$\boldsymbol{f_a = 5}$	$\boldsymbol{f_a = 6}$
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 10
Minimal number of nodes $n = 6f_\ell + 4f_a + 1$ for the hybrid Phase Queen algorithm for different f_a, f_ℓ.

7 Conclusions

We analyzed the link failure assumption coverage of our perception-based failure model: For both exponential and polynomial consensus and Byzantine agreement algorithms, we computed the probability Q_m of violating the link failure bound $f_\ell^r = f_\ell^s = f_\ell$ at least once during m ($m + 1$ in case of OMH) rounds of execution. Our numerical results reveal a very reasonable coverage, even for the worst exponential algorithm OMH, cp. Tables 4–6.

In order to answer the question of whether increasing f_ℓ always decreases Q_m, despite of the fact that n and hence the number of links that may fail is also

f_ℓ	$f_a = 1$	$f_a = 2$	$f_a = 3$	$f_a = 4$	$f_a = 5$	$f_a = 6$
1	0.4	1	1	1	1	1
2	0.06	0.2	0.6	1	1	1
3	0.01	0.04	0.08	0.2	0.4	0.8
5	0.0002	0.0008	0.002	0.006	0.01	0.02
7	$6.\,10^{-6}$	0.00002	0.00006	0.0001	0.0004	0.0006
10	$2.\,10^{-8}$	$8.\,10^{-8}$	$2.\,10^{-7}$	$6.\,10^{-7}$	$1.\,10^{-6}$	$2.\,10^{-6}$

Table 11
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 12
Minimal number of nodes $n = 6f_\ell + 3f_a + 1$ 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	0.6	1	1	1	1	1
2	0.1	0.4	0.8	1	1	1
3	0.02	0.06	0.1	0.4	0.6	1
5	0.0006	0.002	0.004	0.008	0.02	0.02
7	$2.\,10^{-6}$	0.00004	0.0001	0.0002	0.0004	0.0008
10	$8.\,10^{-8}$	$2.\,10^{-7}$	$4.\,10^{-7}$	$8.\,10^{-7}$	$2.\,10^{-6}$	$2.\,10^{-6}$

Table 13
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	0.7	1	1	1	1	1
3	0.2	0.6	1	1	1	1
5	0.008	0.02	0.08	0.2	0.4	0.8
7	0.0002	0.001	0.002	0.006	0.01	0.02
10	$2.\,10^{-6}$	$4.\,10^{-6}$	$1.\,10^{-5}$	0.00002	0.00006	0.0001

Table 14
Value of (approximate) probability of failure Q'_m for $p = 0.01$ for the Srikanth & Toueg algorithm with minimal number of nodes.

increased, we proceed as follows: Starting from system with n_0 processes that can withstand f_ℓ link failures, how does Q_m change if we add $c\ell$ processors ($c = 4$ in case of OMH, $c = 6$ for all polynomial algorithms) in order to cope with ℓ additional link failures per process? Comparison of (20), (22) and (34) reveals that the expressions for Q_m are very similar. It hence suffices to consider the worst one: Substituting $n = n_0 + c\ell$ in (20) in Theorem 6, we find

$$\begin{aligned} Q_m &= \mathcal{O}\Big(n_0^m \frac{(n_0 p)^{f_\ell+\ell+1}}{(f_\ell+\ell+1)!}(1+\frac{c\ell}{n_0})^{m+f_\ell+\ell+1}\Big) \\ &= \mathcal{O}\Big(n_0^m \frac{(n_0 p)^{f_\ell+1}}{(f_\ell+1)!} \cdot \frac{(n_0 p \cdot e^c)^\ell}{[f_\ell+\ell+1]_\ell}\Big) \end{aligned} \tag{36}$$

In the last step, we employed the well-known relation $(1+t/(k+j))^k \leq e^t$ for $t \geq 0$, $j \geq 0$: Since we must have $n_0 - m - f_\ell - 2 \geq 1$ for Theorem 6, it follows that $k + j = n_0 \geq m + f_\ell + \ell + 1 = k$.

It is hence apparent from (36) that, as long as $np < 1$ is still sufficiently small, the probability of violation Q_m

- rapidly grows with m and hence with the number $f_a + f_o$ of arbitrary and omission failures,
- marginally grows with n and hence with the number of any kind of failures,
- decreases with the number of tolerated link failures f_ℓ (and with decreasing p, of course), since the last factor in (36) is < 1 for any suitably chosen ℓ.

Hence, adding processes in order to tolerate more link failures 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 failures per process. This ultimately confirms that

(1) limiting the power of link failures according to our failure model is not an undue restriction,
(2) our algorithms can even be employed in wireless systems, where link failure probabilities p up to 10^{-2} are common,

which was the ultimate question to be answered by this paper.

References

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

[2] Martin Biely and Ulrich Schmid. Message-efficient consensus in presence of hybrid node and link faults. Technical Report 183/1-116, Department of Automation, Technische Universität Wien, August 2001.

[3] P. Th. Eugster, R. Guerraoui, S.B. 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.

[4] Eli Gafni. Round-by-round fault detectors (extended abstract): unifying synchrony and asynchrony. In *Proceedings of the Seventeenth Annual ACM Symposium on Principles of Distributed Computing*, pages 143–152. ACM Press, 1998.

[5] Jim N. Gray. Notes on data base operating systems. In G. Seegmüller 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.

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

[7] Patrick Lincoln and John 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.

[8] 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.

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

[10] Ulrich Schmid and Bettina Weiss. Synchronous Byzantine agreement under hybrid process and link failures. Technical Report 183/1-124, Department of Automation, Technische Universität Wien, November 2002. (replaces TR 183/1-110).

[11] Ulrich Schmid and Bettina Weiss. Impossibility results and lower bounds for consensus under link failures. Technical Report 183/1-129, Department of Automation, Technische Universität Wien, April 2003. (submitted).

[12] Ulrich Schmid, Bettina Weiss, and John Rushby. Formally verified byzantine agreement in presence of link faults. In *22nd International Conference on Distributed Computing Systems (ICDCS'02)*, pages 608–616, July 2-5, 2002.

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