The impact of the failure model above can be described in terms of an adversary attempting to cause a protocol to fail by manipulating the system within the bounds of the model. Such an adversary has these capabilities and restrictions:

—An adversary cannot use knowledge of future probabilistic outcomes, interfere with random coin tosses made by processes, cause correlated (nonindependent) failures to occur, or do anything not enumerated below.

—The adversary has complete knowledge of the history of the current run.

—At the beginning of a run of the protocol, it has the ability to individually set process failure rates, within the bounds $[0..\tau]$

—For messages, it has the ability to individually set message failure probabilities within the bounds of $[0..\varepsilon]$ and can arbitrarily select the "point" at which messages are lost.

Note, that, although probabilities may be manipulated by the adversary, it may only make the system "more reliable" than the bounds, ε and τ.

Over this system model, we layer protocols with strong probabilistic convergence properties. The probabilistic analysis of these properties is, necessarily, only valid in runs of the protocol in which the system obeys the model. The independence properties of the system model are quite strong and are not likely to be continuously realizable in the actual system. For example, partition failures are correlated communication failures and do not occur in this model. Partitions can be "simulated" by the independent failures of several processes, but are of vanishingly low probability. Similarly, the model gives little insight into how a system might behave during and after a brief networkwide communication outage. Both types of failures are realistic threats, which is why we resorted to experiments to explore their impact on the protocol.

A.2 Pbcast Protocol

The version of the protocol used in our analysis is simplified, as follows. We will assume that a run of the pbcast protocol consists of a fixed number of rounds, after which a multicast vanishes from the system because the corresponding message is garbage-collected. A process initiates a pbcast by unreliably multicasting the message, and it is received by a random subset of the processes. These gossip about the message, causing it to reach processes that did not previously have a copy, which gossip about it in turn. For our analysis, we consider just a single multicast event, and we adopt the view that a process gossips about a multicast message only during the round in which it first receives a copy of that message. Processes choose the destinations for their gossip by tossing a weighted random coin for each other process to determine whether to send a gossip message to that process. Thus, the parameters of the protocol studied in the analysis are

—P: the set of processes in the system. $N = |P|$.

—R: the number of rounds of gossip to run.

—β: the probability tht a process gossips to each other process (the weighting of the coin mentioned above). We define the *fanout* of the protocol to be β^*N: this is the expected number of processes to which a participant gossips.

Described in this manner, the behavior of the gossip protocol mirrors a class of disease epidemics which nearly always infect either almost all of a population or almost none of it. The pbcast bimodal delivery distribution, mentioned earlier, will stem from the "epidemic" behavior of the gossip protocol. The normal case for the protocol is one in which gossip floods the network in a random but exponential fashion.

A.3 Pbcast Analysis

Our analysis will show how to calculate the bimodal pbcast delivery distribution for a given setting, and how to bound the probability of a pbcast "failure" using a definition of failure provided by the application designer in the form of a predicate on the final system state. It would be preferable to present a closed-form solution; however, doing so for non-trivial epidemics of the kind seen here is an open problem in epidemic theory. In the absence of closed-form bounds, the approach of this analysis will be to derive a recurrence relation between successive rounds of the protocol, which will then be used to calculate an upper bound on the chance of a failed pbcast run.

A.4 Notation and Probability Background

The following analysis uses standard probability theory. We use three types of random variables. Lowercase variables, such as f, r and s, are integral random variables; uppercase variables, such as X, are binary random variables (they take values from $\{0,1\}$); and uppercase bold variables, such as $\mathbf{X}$, are integral random variables corresponding to sums of binary variables of the same letter: $\mathbf{X} = \Sigma X_i$.

$P\{v = k\}$ refers to the probability of the random variable v having the value k. For binary variables, $P\{X\} = P\{X = 1\}$. With lowercase integral random variables, in $P\{r\}$ the variable serves both to specify a random variable and as a binding occurrence for a variable of the same name.

The distributions of sums of independent, identically distributed binary variables are called binomial distributions. If $\forall 0 \leq i < n : P\{X_i\} = p$, then

$$P\{X = k\} = \binom{n}{k}(p)^k(1-p)^{n-k}.$$

We use relations among random variables to derive bounds on the distributions of the weighted and unweighted sums of the variables. Let X_i,

Y_i, and Z_i form finite sets of random variables, and let $g(i)$ be a nonnegative real-valued function defined over integers. If

$$\forall 0 \le i < n : P\{X_i\} \le P\{Y_i\} \le P\{Z_i\}$$

then

$$P\{\mathbf{Y} = k\} \le P\{\mathbf{Z} \ge k\} - P\{\mathbf{X} \ge k + 1\} \quad (1)$$

$$\sum_{0 \le i < n} P\{\mathbf{X} = i\} g\{i\} \le \sum_{0 \le i < n} P\{\mathbf{Y} = i\} \max_{0 \le j \le i} g(j) \quad (2)$$

These equations will be applied later in the analysis.

A.5 A Recurrence Relation

The first step is to derive a recurrence relation that bounds the probability of protocol state transitions between successive rounds. We describe the state of a round using three integral random variables: s_t is the number of processes that may gossip in round t (or in epidemic terminology the infectious processes); r_t is the number of processes in round t that have not received a gossip message yet (the susceptible processes); and f_t is the number of infectious processes in the current round which are faulty.

Recall from the outset of this chapter that our analysis is pessimistic, assuming that the initial unreliable broadcast fails and reaches none of the destinations, leaving an initial state in which a single process has a copy of the message while all others are susceptible:

$$s_0 = 1, r_0 = N - 1, f_0 = 0$$

$$r_{t+1} + s_{t+1} = r_t$$

$$\sum_{0 \le t \le R} s_t + r_R = N$$

The recurrence relation we derive, $R(s_t, r_t, f_t, s_{t+1})$, is a bound on the conditional probability, given the current state described by (s_t, r_t, f_t), that s_{t+1} of the r_t susceptible processes receive a gossip message from this round. Expressed as a conditional probability, this is $P\{s_{t+1} | s_t, r_t, f_t\}$.

For each of the r_t processes, we introduce a binary random variable, X_i, corresponding to whether a particular susceptible process receives gossip this round. s_{t+1} is equal to the sum of these variables, ΣX_i or equivalently $\mathbf{X}$. In order to calculate $R(s_t, r_t, f_t, s_{t+1})$, we will derive bounds on the distribution of $\mathbf{X}$. Our derivation will be in four steps. First we consider $P\{X_i\}$ in the absence of faulty processes and with fixed message failures. Then we introduce, separately, generalized message failures and faulty processes, and finally we combine both failures. Then we derive bounds on $P\{\mathbf{X} = k\}$ for the most general case.

A.5.1 *Fixed Message Failures.* The analysis begins by assuming (1) that there are no faulty processes and (2) that message delay failures occur with exactly ε probability, no more and *no less*. This assumption limits the system from behaving with a more reliable message failure rate. In the absence of these sort of failures, the behavior of the system is the same as a well-known (in epidemic theory) epidemic model, called the chain-binomial epidemic. The literature on epidemics provides a simple method for calculating the behavior of these epidemics when there are an unlimited number of rounds and no notion of failures [Bailey 1975]. We introduce constants $p = \beta(1 - \varepsilon)$ and $q = 1 - p$. p is the probability that both an infectious process gossips to a particular susceptible process and that the message does not experience a send omission failure under the assumption of fixed message failures. (Note that this use of p is unrelated to the reliability parameter p employed elsewhere in the article; the distinction is clear from context.)

For each of the r_t susceptible processes and corresponding variable, X_i, we consider the probability that at least one of the s_t infectious processes sends a gossip message which gets through. Expressed differently, this is the probability that not all infectious processes fail to send a message to a particular susceptible process:

$$P\{X_i\} = 1 - (1 - p)^{s_t} = 1 - q^{s_t}$$

A.5.2 *Generalized Message Failures.* A potential risk in the analysis of pbcast is to assume, as may be done for many other protocols, that the worst case occurs when message loss is maximized. Pbcast's failure mode occurs when there is a partial delivery of a pbcast. A pessimistic analysis must consider the case where local increases in the message delivery probability decrease the reliability of the overall pbcast protocol. We extend the previous analysis to get bounds on $P\{X_i\}$, but where the message failure rate may be anywhere in the range of $[0..\varepsilon]$

Consider every process i that gossips, and consider every process j that i sends a gossip message to. With generalized message failures, there is a probability ε_{ij} that the message experiences a send omission failure, such that $0 \leq \varepsilon_{ij} \leq \varepsilon$. This gives bounds $[p_{lo}..p_{hi}]$ on p_{ij} the probability that process i both gossips to process j and the probability that the message is delivered: $\beta(1 - \varepsilon) = p_{lo} \leq \beta(1 - \varepsilon_{ij}) = p_{ij} \leq p_{hi} = \beta$ (we also have $q_{lo} = 1 - p_{lo}$ and $q_{hi} = 1 - p_{hi}$).

This in turn gives bounds on the probability of each of the r_t processes being gossiped to, expressed using the variables X_{hi} and X_{lo} which correspond to a fixed message failure rate model:

$$1 - q_{lo}^{s_t} = P\{X_{lo}\} \leq P\{X_j\} \leq P\{X_{hi}\} = 1 - q_{hi}^{s_t}$$

A.5.3 *Process Failures.* Introducing process failures into the analysis is done in a similar fashion to that of generalized message failures. For

simplicity in the following discussion, we again fix the probability of message failure to ε.

We assume that f_t of the s_t infectious processes that are gossiping in the current round are faulty. For the purposes of analyzing pbcast, there are three ways in which processes can fail. They can crash before, during, or after the gossip stage of the pbcast protocol. Regardless of which case applies, a process always sends a subset of the messages it would have sent had it not been faulty: a faulty process never introduces spurious messages. If all f_t processes crash before sending their gossip messages, then the probability of one of the susceptible processes receiving gossip message, $P\{X_i\}$, will be as though there were exactly $s_t - f_t$ correct processes gossiping in the current round. If all crash after gossiping then the probability will be as though all s_t processes gossiped, while none of the f_t processes had failed. All other cases cause the random variables, X_i, to behave with some probability in between:

$$1 - q^{s_t - f_t} = P\{X_{lo}\} \leq P\{X_i\} \leq P\{X_{hi}\} = 1 - q^{s_t}$$

A.5.4 *Combined Failures.* The bounds from the two previous sections are "combined" to arrive at

$$1 - q_{lo}^{s_t - f_t} = P\{X_{lo}\} \leq P\{X_i\} \leq P\{X_{hi}\} = 1 - q_{hi}^{s_t}$$

Then we apply Eq. (1) to get bounds on $P\{\Sigma X_j = k\}$, or $P\{X = k\}$:

$$P\{\mathbf{X} = k\} \leq P\{\mathbf{X}_{hi} \geq k\} - P\{\mathbf{X}_{lo} \geq k + 1\}$$

Expanding terms, we get the full recurrence relation:

$$P\{s_{t+1} | s_t, r_t, f_t\} \leq \sum_{s_{t+1} \leq i \leq N} \binom{r_t}{i} (1 - q_{hi}^{s_t})^i (q_{hi}^{s_t})^{r_t - i} - \sum_{s_{t+1} \leq i \leq N} \binom{r_t}{i} (1 - q_{lo}^{s_t - f_t})^i (q_{lo}^{s_t - f_t})^{r_t - i} \quad (3)$$

We define the right hand side of relation (3) to be $R(s_t, r_t, f_t, s_{t+1})$, "an upper bound on the probability that with s_t gossiping processes of which f_t are faulty, and with r_t processes that have not yet received the gossip, that s_{t+1} processes will receive the gossip this round."

A.6 Predicting Latency to Delivery

Still working within the same model[10] we can compute the distribution of latency between when a message is sent and when it is delivered. For the

[10]Actually, we differ in one respect: the analysis of this subsection explicitly treats gossip to b processes during each round. The previous analysis treated all gossip as occuring in the first round.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.