Optimal Detection of a Corrupted Page in a Replicated File

Khaled A. S. Abdel-Ghaffar*
Department of Electrical
and Computer Engineering
University of California
Davis, CA 95616

Amr El Abbadi†
Department of Computer Science
University of California
Santa Barbara, CA 93106

Abstract

The problem of detecting a corrupted page in a file with multiple copies is addressed. A lower bound is derived on the communication overhead and a protocol is developed that requires exactly the amount of communication specified by the lower bound. The lower bound and the protocol are the first optimality results for the detection of a corrupted page in a file with more than two copies.

1 Introduction

One of the main paradigms for fault-tolerance in distributed systems is replication. By storing several copies of a file at different sites, the overall availability of the file can be greater than the availability of a single site. Another, equally important motivation for replicating a file is the need for fast and efficient access to the information in the file at the site where a copy is stored. In many scientific applications, e.g., geographic information systems, large files of images and maps are used by many geographically dispersed scientists. The usual mode of operation for these scientists is to acquire a copy of the data, and then use it to derive other more complex data. Hence in such applications, multiple copies of a large file are stored in different sites of a distributed network.

Recently, there has been increasing interest in developing techniques for detecting corrupted pages in the distributed copies of a large file. Since the files are large, efficient techniques must be developed that avoid the transmission of large copies across a network for comparison. Furthermore, since disk failures are usually localized to a small portion of the disk, the number of corrupted pages is usually quite small. In fact, the earliest work on this problem, by Fuchs, Wu and Abraham [5] considered the problem of detecting a single corrupted page when the file is implemented by two copies only. Barbara, Garcia-Molina and Feijoo [3] extended this work to detect a maximum of two corrupted pages in a file with two copies. Since pages are typically quite large, it is common to compute for each page, a concise representation called a signature [8]. Metzner and Kapturowski [8, 11, 10] have developed several protocols for detecting any number of corrupted pages for a file with two copies. Barbara and Lipton [4] presented a class of randomized strategies for identifying any given number of corrupted pages. Schwarz, Bowdidge and Burkhard [13] also developed a mechanism that is able to detect and identify missing and extraneous pages.

Several attempts have been made recently to derive lower bounds on the communication overhead and to design protocols that meet these lower bounds. Madej [6] presented a lower bound on the amount of communication needed to detect any given number of corrupted pages of a file with two copies. Deterministic and randomized protocols were also presented although they do not meet the lower bound. In [2] another lower bound was derived for the same problem. Furthermore, a protocol was developed that requires the minimum amount of communication specified in the lower bound, hence the protocol is optimal and the bound is tight. However, this bound is restricted to files that have two copies only and where it is known a priori that one of the two copies has no corrupted pages. Rangarajan and Fussell [12] explored the more

*This author was supported in part by the National Science Foundation under grant NCR-9115423.

†This author was supported in part by the National Science Foundation under grant IRI-9117094.

general case of a file with any number of copies. They derived lower bounds on the number of bits that need to be transmitted in order to detect corrupted pages within a certain probability of error that goes to zero as the number of pages in the file grows to infinity. They also presented randomized protocols that meet this lower bound, asymptotically, within a multiplicative constant. However, since the protocols are randomized, it may fail to detect the corrupted pages. Abdel-Ghaffar and El Abbadi [1] studied the multiple copy file problem in a restricted communication model, namely the primary site model. In this model, a particular site, referred to as the primary site, is in charge of coordinating and detecting all corrupted pages, and communication is strictly forbidden between other sites. A lower bound is derived and an optimal deterministic protocol for this model is presented. It is, however, shown that this bound does not hold in the more general and more realistic model where any two sites can communicate with each other. In fact, a protocol can be derived that requires less communication overhead than the lower bound of the primary site model.

In this paper, we address the problem of deriving a tight lower bound for the general, nonrestricted communication model for a multi-copy replicated file. As in [5], we restrict ourselves to the case of detecting at most one corrupted page. This case is quite important since failures corrupting more than one page are less frequent than those corrupting a single page. However, unlike [5] where only two copies are assumed, we consider a file that is implemented by any number of copies. We derive a lower bound on the communication required to detect such a corrupted page, and develop a protocol that achieves this bound. This protocol and the lower bound are the first optimality results in the case of general deterministic multi-copy file systems.

In the next section, we start by presenting a formal description of the problem statement and introduce some of the standard techniques used in the solutions. In Section 3, we derive the lower bound, and in Section 4, we develop an optimal protocol. A comparison with other related work is presented in Section 5. The paper concludes with a discussion of our results.

2 Problem statement

Consider a file which is divided into N pages: $P_1, \ldots, P_N$, and a distributed system of M sites referred to as $s_1, \ldots, s_M$. The file, which is denoted by $(P_1, \ldots, P_N)$, is replicated at the M different sites. We use $P_{n,m}$, where $1 \leq n \leq N$ and $1 \leq m \leq M$, to refer to the n^{th} page of the copy residing in site s_m. Ideally, all copies are identical, however, due to site failures, some pages may be corrupted. We assume that among the total number of NM pages that reside in all M sites, at most *one* of them is corrupted. The goal is to be able, through (error-free) communication between sites, to compare the copies of the file residing in the M sites in order to determine the corrupted page.

Since the pages may be quite large, it is common to compute for each page P_i a concise representation, called the page signature of P_i, and denoted by p_i [8]. Each page signature has length b bits, which is typically much less than the length of the page. Thus, two pages with different page signatures are not identical, but two different pages may have the same page signature. The probability that this happens can be made arbitrarily small by making b large. For practical purposes, we are often content if all pages with corrupted signatures are identified. Metzner [8] has developed a simple technique to assign signatures to pages using feedback shift registers. In this technique, it is assumed that the number of pages in the file, N, is less than the total number of page signatures 2^b, i.e., $b > \log_2 N$. We will assume that the condition $b > \log_2 N$ holds throughout this paper.

To identify the corrupted page signature, sites must exchange information via message passing. The amount of communication is measured by the number of transmitted bits. For normalization, the cost of communication is defined to be the number of sequences, each of length b bits, that need to be transmitted. For convenience, we refer to each such sequence as a signature. For example, a site may send the bit-wise exclusive-or of a number of page signatures. This contributes one signature to the cost of communication. It should be noted that the cost of communication may depend on the location of the corrupted page. Here, we are concerned with the number of signatures that need to be transmitted, in the worst case, in order to identify the page with the corrupted signature.

In this paper, we will consider the general case where the number of copies, M, is arbitrary. Furthermore, we will drop the assumption that an uncorrupted copy resides in a known site. In order to drop this assumption and still be able to identify the page with a corrupted signature, we will assume that $M \geq 3$. Indeed, if $M = 2$, then there is no way to determine which page is corrupted even if each site knows the page signatures of the other site.

3 Lower bound

In this section we derive a lower bound on the number of signatures that must be transmitted in order to detect one page with a corrupted signature in a file with $M \geq 3$ copies. Obviously, in order to accomplish this, a site with a corrupted signature should receive or transmit a message. Indeed, if this is not the case, then it is not possible to determine if the site is corrupted or not, and in case it is corrupted and $N \geq 2$, then there is no way to identify which page has an erroneous signature. Since the corrupted site is not known a priori, at least $\lceil M/2 \rceil$ signatures may need to be communicated in order for each site, and in particular the corrupted one, to receive or transmit a signature. This gives a lower bound on the number of communicated signatures. The following theorem refines this lower bound and shows that it can be improved in case $N = 1$ to $M/2+1$ if M is even, and in case $N \geq 2$ to $(M+3)/2$ if M is odd and to $M/2+2$ if M is even.

Theorem 1 *The minimum number of signatures $T_N(M)$ that need to be transmitted, in the worst case, in order to identify up to one page with corrupted signature in M copies of a file composed of N pages is lower bounded by*

$$T_N(M) \geq \begin{cases} \lfloor M/2 \rfloor + 1 & \text{if } N = 1 \\ \lfloor M/2 \rfloor + 2 & \text{if } N \geq 2 \end{cases}.$$

Proof. We first consider the case $N = 1$. Suppose, to get a contradiction, that it suffices to transmit $T \leq \lfloor M/2 \rfloor$ signatures. Assume that the first $T-1$ transmitted signatures are communicated between uncorrupted sites. Thus, there are $M - 2(T-1) \geq M - 2(\lfloor M/2 \rfloor - 1) \geq 2$ sites that do not transmit or receive any of these signatures. Let s and s' be two such sites. If the remaining signature is not transmitted from one of them to the other, then assume that this last signature is communicated between uncorrupted sites. Hence, there is a site that does not transmit or receive any signatures. It is impossible to determine whether or not this site is corrupted. Therefore, assume that the last signature is communicated between s and s'. If one of them is corrupted, there is no way to determine which one it is. This completes the proof in case $N = 1$.

Next, consider the case $N \geq 2$. First, we will show that if it is known that a corrupted page resides in site s, then it is impossible to determine the corrupted page by transmitting one signature. Indeed, if it is possible to determine the corrupted page from one signature, then this signature is either transmitted or received by s. Suppose that it is transmitted by s. Since each signature is composed of b bits, then the transmitted signature is one of 2^b possible signatures. On the other hand, there are $2^{2b} > 2^b$ choices for any pair of page signatures. Therefore, there are at least two tuples of distinct page signatures $\mathbf{p}' = (p'_1, p'_2, p_3, \ldots, p_N)$ and $\mathbf{p}'' = (p''_1, , p''_2, p_3, \ldots, p_N)$ such that if s has any of them, then it transmits the same signature. If the receiving site has the page signatures $\mathbf{p} = (p'_1, p''_2, p_3, \ldots, p_N)$, then it should be able to determine, based on the signature received from s, whether s has page signatures $\mathbf{p}'$ or $\mathbf{p}''$ since each of these tuples differs from $\mathbf{p}$ in one page only. This is impossible since the transmitted signature is the same in both cases. A similar argument holds in case the signature is received by s.

To complete the proof of the theorem in case $N \geq 2$, assume that it suffices to transmit $T \leq \lfloor M/2 \rfloor + 1$ signatures. Suppose that the first $T-2$ transmitted signatures are communicated between uncorrupted sites. Thus, there are $M - 2(T-2) \geq M - 2(\lfloor M/2 \rfloor - 1) \geq 2$ sites that do not transmit or receive any of these signatures. Let s and s' be two such sites. If the $T-1^{st}$ signature is not transmitted from one of them to the other, then assume that this signature is communicated between uncorrupted sites. Hence, there is a site, say s, that does not transmit or receive any of the first $T-1$ signatures. From the above argument, it is impossible to determine the corrupted page in s, if there is any, based on the T^{th} signature. Therefore, assume that the $T-1^{st}$ signature is communicated between s and s'. If one of these sites is corrupted, there is no way to determine which one it is based on the first $T-1$ transmitted signa-

tures. Therefore, the above argument implies that the last signature should be communicated between s and s' also. Since s and s' do not communicate with other sites, it is impossible to determine which one is corrupted. This completes the proof of the theorem in case $N \geq 2$. □

4 An optimal protocol

We present a protocol, to identify up to one page with corrupted signature in $M \geq 3$ copies of a file composed of N pages, that requires the transmission of T signatures, where

$$T = \begin{cases} \lfloor M/2 \rfloor + 1 & \text{if } N = 1 \\ \lfloor M/2 \rfloor + 2 & \text{if } N \geq 2 \end{cases}.$$

Based on the theorem, it follows that the protocol requires the transmission of $T = T_N(M)$ signatures, which is the minimum number of signatures that need to be communicated. Hence, this protocol is optimal.

First, we consider the case of $N = 1$ presented in Figures 1, 2, and 3. This case illustrates, in a simple manner, the basic operation of the protocol. The algorithm for site s_m, where m is even, is shown in Figure 1 in case the total number of copies M is even and in Figure 2 in case M is odd. Figure 3 illustrates the algorithm for site s_m, where m is odd. Site s_m, where $m = 2, 4, \ldots, 2\lfloor M/2 \rfloor$, requests and receives page signature $p_{1,m-1}$ from site s_{m-1}. Site s_m compares its page signature, $p_{1,m}$, with $p_{1,m-1}$. If $p_{1,m} \neq p_{1,m-1}$, then s_m knows that $p_{1,m}$ or $p_{1,m-1}$ is corrupted. Therefore, pages in all sites other than s_{m-1} and s_m are correct. In particular, there is a unique even m for which $p_{1,m} \neq p_{1,m-1}$. Site s_m requests from a third site (as specified below) its page signature which must be correct. Comparing this page signature with $p_{1,m}$ and $p_{1,m-1}$ determines which page is corrupt. The total number of transmitted signatures is $\lfloor M/2 \rfloor + 1$. The third site, which is different from both s_{m-1} and s_m, and from which s_m requests its page signature is specified as follows. In case M is even, then site s_1 is different from sites s_{m-1} and s_m if $m = M$ (recall that $M \geq 3$) and site s_{M-1} is different from the same sites if $m < M$ (recall that m and M are even). Therefore, we take the third site to be s_1 if $m = M$ and s_{M-1} if $m < M$ (see Figure 1). In case M is odd, then site s_M is different from sites s_{m-1} and s_m since $m < M$. We take the third site to be s_M (see Figure 2).

Next, we consider the case in which $p_{1,m} = p_{1,m-1}$ for all $m = 2, 4, \ldots, 2\lfloor M/2 \rfloor$. If M is even, then this implies that there are no pages with corrupted signatures. If M is odd, then the first $M - 1$ sites have no pages with corrupted signatures and no site requests $p_{1,M}$ from site s_M. If site s_M does not receive any requests from other sites, it sends its page signature to site s_{M-1}. Site s_{M-1} compares its page signature $p_{1,M-1}$ with $p_{1,M}$. If they are identical, then $p_{1,M}$ is correct. Otherwise $p_{1,M}$ is corrupt. The number of transmitted signatures is $\lfloor M/2 \rfloor + 1$ in this case also.

Next, we present the protocol in case $N \geq 2$ as shown in Figures 4, 5, and 6. The algorithm for site s_m, where m is even, is shown in Figure 4 in case the total number of copies M is even and in Figure 5 in case M is odd. Figure 6 illustrates the algorithm for site s_m, where m is odd. Each signature can be represented as an element in the finite field $GF(2^b)$. Define

$$sig_{0,m} = \sum_{n=1}^{N} p_{n,m} \quad \text{and} \quad sig_{1,m} = \sum_{n=1}^{N} p_{n,m}\alpha^n$$

where α is a primitive element in $GF(2^b)$, i.e., $\alpha^1, \alpha^2, \ldots, \alpha^N, \ldots, \alpha^{2^b-1}$ are distinct [7]. Site s_m, where $m = 2, 4, \ldots, 2\lfloor M/2 \rfloor$, requests and receives the signature $sig_{0,m-1}$ from site s_{m-1}. Site s_m compares $sig_{0,m}$ with $sig_{0,m-1}$. If $sig_{0,m} \neq sig_{0,m-1}$, then s_m knows that a page in either s_m or s_{m-1} is corrupted. Therefore, pages in all sites other than s_{m-1} and s_m are correct. Furthermore, there is a unique even m and a unique n for which $p_{n,m} \neq p_{n,m-1}$. To determine n, site s_m requests $sig_{1,m-1}$ from site s_{m-1}. From the definitions of $sig_{0,m}$ and $sig_{1,m}$, we have

$$\begin{aligned} sig_{0,m} - sig_{0,m-1} &= p_{n,m} - p_{n,m-1} \\ sig_{1,m} - sig_{1,m-1} &= \alpha^n(p_{n,m} - p_{n,m-1}) \end{aligned}$$

where α^n and $p_{n,m} - p_{n,m-1}$ are unknown. Solving these equations gives $\alpha^n = (sig_{1,m} - sig_{1,m-1})/(sig_{0,m} - sig_{0,m-1})$. From α^n, the value of n can be determined since $\alpha^1, \alpha^2, \ldots, \alpha^N$ are distinct. This can be easily accomplished by a look-up table. Thus, site s_m knows that the corrupted page is either $p_{n,m}$ or $p_{n,m-1}$. To determine which one is corrupt, site s_m requests from a third site s_i, as specified above in case $N = 1$, its page signature $p_{i,n}$ which must be correct. Comparing this page signature with $p_{n,m}$ and

```
SEND(REQUEST p_{1,m-1}) to site s_{m-1};
RECEIVE(p_{1,m-1}) from site s_{m-1};
IF p_{1,m} = p_{1,m-1}
    THEN p_{1,m} and p_{1,m-1} are not corrupted
    ELSE (* Either site s_m or site s_{m-1} has a page with a corrupted signature *)
        IF m = M
            THEN
                SEND(REQUEST p_{1,1}) to site s_1;
                RECEIVE(p_{1,1}) from site s_1;
                IF p_{1,m} = p_{1,1} THEN p_{1,m-1} is corrupted ELSE p_{1,m} is corrupted
            ELSE (* m ≠ M *)
                SEND(REQUEST p_{1,M-1}) to site s_{M-1};
                RECEIVE(p_{1,M-1}) from site s_{M-1};
                IF p_{1,m} = p_{1,M-1} THEN p_{1,m-1} is corrupted ELSE p_{1,m} is corrupted
```

Figure 1: **The algorithm for site s_m, where m is even, in case $N = 1$ and M is even**

```
SEND(REQUEST p_{1,m-1}) to site s_{m-1};
RECEIVE(p_{1,m-1}) from site s_{m-1};
IF p_{1,m} = p_{1,m-1}
    THEN p_{1,m} and p_{1,m-1} are not corrupted
    ELSE (* Either site s_m or site s_{m-1} has a page with a corrupted signature *)
        SEND(REQUEST p_{1,M}) to site s_M;
        RECEIVE(p_{1,M}) from site s_M;
        IF p_{1,m} = p_{1,M} THEN p_{1,m-1} is corrupted ELSE p_{1,m} is corrupted
IF RECEIVE(p_{1,M}) from site s_M (* m = M - 1 and pages in all sites other than s_M are correct *)
    THEN
        IF p_{1,m} = p_{1,M} THEN p_{1,M} is correct ELSE p_{1,M} is corrupted
```

Figure 2: **The algorithm for site s_m, where m is even, in case $N = 1$ and M is odd**

```
IF RECEIVE(REQUEST p_{1,m}) from site s_i THEN SEND(p_{1,m}) to site s_i;
IF m = M and site s_m has not received a message (REQUEST p_{1,m}) THEN SEND(p_{1,m}) to site s_{M-1};
```

Figure 3: **The algorithm for site s_m, where m is odd, in case $N = 1$ and all M**

```
SEND(REQUEST sig_{0,m-1}) to site s_{m-1};
RECEIVE(sig_{0,m-1}) from site s_{m-1};
IF sig_{0,m} = sig_{0,m-1}
    THEN sites s_m and s_{m-1} have no pages with corrupted signatures
    ELSE (* Either site s_m or site s_{m-1} has a page with a corrupted signature *)
        SEND(REQUEST sig_{1,m-1}) to site s_{m-1}
        RECEIVE(sig_{1,m-1}) from s_{m-1};
        SOLVE equations to determine the value of n
            sig_{0,m} - sig_{0,m-1} = p_{n,m} - p_{n,m-1}
            sig_{1,m} - sig_{1,m-1} = α^n(p_{n,m} - p_{n,m-1})
        IF m = M
            THEN
                SEND(REQUEST p_{n,1}) to site s_1;
                RECEIVE(p_{n,1}) from site s_1;
                IF p_{n,m} = p_{n,1} THEN p_{n,m-1} is corrupted ELSE p_{n,m} is corrupted
            ELSE (* m ≠ M *)
                SEND(REQUEST p_{n,M-1}) to site s_{M-1};
                RECEIVE(p_{n,M-1}) from site s_{M-1};
                IF p_{n,m} = p_{n,M-1} THEN p_{n,m-1} is corrupted ELSE p_{n,m} is corrupted
```

Figure 4: **The algorithm for site s_m, where m is even, in case $N \geq 2$ and M is even**

```
SEND(REQUEST sig_{0,m-1}) to site s_{m-1};
RECEIVE(sig_{0,m-1}) from site s_{m-1};
IF sig_{0,m} = sig_{0,m-1}
    THEN sites s_m and s_{m-1} have no pages with corrupted signatures
    ELSE (* Either site s_m or site s_{m-1} has a page with a corrupted signature *)
        SEND(REQUEST sig_{1,m-1}) to site s_{m-1};
        RECEIVE(sig_{1,m-1}) from s_{m-1};
        SOLVE equations to determine the value of n
            sig_{0,m} - sig_{0,m-1} = p_{n,m} - p_{n,m-1}
            sig_{1,m} - sig_{1,m-1} = α^n(p_{n,m} - p_{n,m-1})
        SEND(REQUEST p_{n,M}) to site s_M;
        RECEIVE(p_{n,M}) from site s_M;
        IF p_{n,m} = p_{n,M} THEN p_{n,m-1} is corrupted ELSE p_{n,m} is corrupted
IF RECEIVE(sig_{0,M}) from site s_M (* m = M - 1 and pages in all sites other than s_M are correct *)
    THEN
        IF sig_{0,m} = sig_{0,M}
            THEN site s_M has no page with a corrupted signature
            ELSE(* Site s_M has a corrputed page *)
                SEND(REQUEST sig_{1,M}) to site s_M;
                RECEIVE(sig_{1,M}) from site s_M;
                SOLVE equations to determine the value of n
                    sig_{0,m} - sig_{0,M} = p_{n,m} - p_{n,M}
                    sig_{1,m} - sig_{1,M} = α^n(p_{n,m} - p_{n,M})
                p_{n,M} is corrupt
```

Figure 5: **The algorithm for site s_m, where m is even, in case $N \geq 2$ and M is odd**

IF RECEIVE(REQUEST $sig_{j,m}$) from site s_i; **THEN SEND**($sig_{j,m}$) to site s_i;
IF RECEIVE(REQUEST $p_{n,m}$) from site s_i; **THEN SEND**($p_{n,m}$) to site s_i;
IF $m = M$ and site s_m has not received a message (REQUEST $p_{n,m}$)
THEN
SEND($sig_{0,m}$) to site s_{M-1};
IF RECEIVE(REQUEST $sig_{1,m}$) from site s_{M-1} **THEN SEND**($sig_{1,m}$) to site s_{M-1};

Figure 6: **The algorithm for site s_m, where m is odd, in case $N \geq 2$ and all M**

$p_{n,m-1}$ determines which page is corrupt. The total number of transmitted signatures is $\lfloor M/2 \rfloor + 2$.

Next, we consider the case in which $p_{1,m} = p_{1,m-1}$ for all $m = 2, 4, \ldots, 2\lfloor M/2 \rfloor$. If M is even, then this implies that there are no pages with corrupted signatures. If M is odd, then the first $M-1$ sites have no pages with corrupted signatures and no site requests a page signature from site s_M. If site s_M does not receive any requests from other sites, it sends $sig_{0,M}$ to site s_{M-1}. Site s_{M-1} compares $sig_{0,M-1}$ with $sig_{0,M}$. If they are identical, then site s_M has no pages with corrupted signatures. Otherwise site s_{M-1} knows that site s_M has a corrupted page signature $p_{n,M}$. It requests $sig_{1,M}$ from site s_M and solves the equations

$$\begin{aligned} sig_{0,M-1} - sig_{0,M} &= p_{n,M-1} - p_{n,M} \\ sig_{1,M-1} - sig_{1,M} &= \alpha^n (p_{n,M-1} - p_{n,M}) \end{aligned}$$

to determine n. The total number of transmitted signatures is $\lfloor M/2 \rfloor + 2$ in this case also.

5 Related work

Previous work on the detection of corrupted pages, was concerned with the special case of files with two copies, assumed restricted communication models, or was randomized in nature. In the case of two copy files, it is assumed that a known site, the coordinator, has no corrupted pages. The protocol of Fuchs, Wu and Abraham [5], which was designed to detect one corrupted page in a file with two copies requires $\lceil \log_2 N \rceil + 1$ signatures. When the model is restricted to a two copy file with a known uncorrupted copy and $N \geq 2$, our protocol requires the transmission of only 2 signatures; which represents a significant improvement for large N. In fact, under the restricted model, our protocol is optimal and is a special case of the protocol presented in [2]. However, all protocols designed for a two copy file do not scale well for files with more than two copies. Consider, for example, the optimal protocol for two copy files [2]. This protocol requires, when the maximum number of possible corrupted pages is one and $N \geq 2$, two signature transmissions to detect the corrupted page. Given M sites (for simplicity assume M is even), a straightforward protocol would divide the M sites into $M/2$ sets each with two sites. The optimal protocol would then be applied to each set. Such a naive approach would only identify a pair of sites with a corrupted page, and already requires $2(M/2) = M$ signatures, i.e., approximately double the number of signatures required by our protocol which is $(M/2) + 2$.

Now we compare our protocol with those protocols designed for multiple copy files, where the number of copies M is greater than two. The protocol of [1] restricted all sites to communicate with a particular site called the primary site. The protocol requires M signatures to be exchanged if $N \geq 2$, which although optimal if communication is restricted to the primary site, is obviously wasteful in terms of communication overhead. Our protocol takes advantage of the more relaxed communication model where all sites can communicate with each other and requires about half as many signature exchanges. In [1], a protocol is also presented which does not restrict the communication model. In the case of one corrupted signature and $N \geq 2$, this protocol requires at least $2M/3 + 1$ signature exchanges, while our optimal protocol requires at most $M/2 + 2$ signature exchanges.

6 Discussion

In this paper we presented an optimal protocol for detecting a corrupted page in a replicated file with any number of copies. We first derived a lower bound on the number of signatures needed to detect a corrupted page. In particular, if the file has only one page and M copies, then $\lfloor M/2 \rfloor + 1$ signatures must be transmitted,

while if the file has more than one page then $\lfloor M/2 \rfloor + 2$ signatures must be transmitted. We then presented a simple protocol that requires exactly this number of signatures, and hence the protocol is optimal in terms of number of transmitted signatures.

Although our presentation concentrated on the problem of detecting the page with the corrupt signature, the goal is to correct this page. This can be accomplished by transmitting a correct copy of the page to the site with the corrupt signature. Hence at an extra cost of one page transfer, the page with a corrupted page signature can be easily corrected.

Our protocol, as well as most previously proposed protocols, use signatures to detect corrupted pages in order to reduce communication overhead since signatures represent pages concisely. Although economical in terms of communication, there is a low probability that two different pages may have the same signature, and hence the corrupted page may not be detected. Our protocol can be easily modified to eliminate this possibility by identifying the signature of a page with the page itself. This approach has the advantage of guaranteeing detection of the corrupted page even though it may not have a corrupted signature. Furthermore, if the corrupted page belongs to a site s_m, where m is even, then the site can correct its page without the need for sending the extra message that contains a correct copy of the page from an uncorrupted site to site s_m. Indeed, in our protocol, site s_m always receives a correct page signature corresponding to its erroneous signature. If signatures are replaced by pages, then site s_m receives the correct page. However, this is at the significant communication overhead of exchanging page-sized messages instead of concise signatures.

Finally, we note that an open problem that still has not been solved is the determination of the minimum communication needed to detect any number of corrupted pages in a file with more than two copies. The present work solves this problem for the important case when only one page can be corrupt.

References

[1] K. A. S. Abdel-Ghaffar and A. El Abbadi, "Efficient detection of corrupted pages in a replicated file," in *Proc. Symp. Principles of Distributed Syst.*, Ithaca, NY, pp. 219–227, Aug. 1993.

[2] K. A. S. Abdel-Ghaffar and A. El Abbadi, "An optimal strategy for comparing file copies," *IEEE Trans. Parallel and Distributed Syst.*, vol. 5, no. 1, pp. 87–93, Jan. 1994.

[3] D. Barbará, H. Garcia-Molina, and B. Feijoo, "Exploiting symmetries for low-cost comparison of file copies," in *Proc. Int. Conf. Distributed Comput. Syst.*, San Jose, CA, pp. 471–479, June 1988.

[4] D. Barbará and R. J. Lipton, "A class of randomized strategies for low-cost comparison of file copies," *IEEE Trans. Parallel and Distributed Syst.*, vol. 2, pp. 160–170, Apr. 1991.

[5] W. Fuchs, K. L. Wu, and J. A. Abraham, "Low-cost comparison and diagnosis of large remotely located files," in *Proc. Symp. Reliability Distributed Software and Database Syst.*, Los Angeles, CA, pp. 67–73, Jan. 1986.

[6] T. Madej, "An application of group testing to the file comparison problem," in *Proc. Int. Conf. Distributed Comput. Syst.*, Newport Beach, CA, pp. 237–243, June 1989.

[7] R. J. McEliece, *Finite Fields for Computer Scientists and Engineers*. Boston: Kluwer, 1987.

[8] J. J. Metzner, "A parity structure for large remotely located replicated data files," *IEEE Trans. Comput.*, vol. C-32, pp. 727–730, Aug. 1983.

[9] J. J. Metzner, "Reliable and efficient broadcast of files to a group of locally interconnected stations," in *Proc. GLOBECOM '86*, Houston, TX, pp. 1762–1767, Dec. 1986.

[10] J. J. Metzner, "Efficient replicated remote file comparison," *IEEE Trans. Comput.*, vol. C-40, pp. 651–660, May 1991.

[11] J. J. Metzner and E. J. Kapturowski, "A general decoding technique applicable to replicated file disagreement location and concatenated code decoding," *IEEE Trans. Inform. Theory*, vol. IT-36, pp. 911–917, July 1990.

[12] S. Rangarajan and D. Fussell, "Rectifying corrupted files in distributed file systems," in *Proc. Int. Conf. Distributed Comput. Syst.*, Arlington, TX, pp. 446–453, May 1991.

[13] T. Schwarz, R. W. Bowdidge, and W. A. Burkhard, "Low cost comparisons of file copies," in *Proc. Int. Conf. Distributed Comput. Syst.*, Paris, France, pp. 196–202, May 1990.