A CLASS OF RANDOMIZED STRATEGIES FOR LOW-COST COMPARISON OF FILE COPIES

Daniel Barbara
Richard J. Lipton

CS-TR-176-88

September 1988

A CLASS OF RANDOMIZED STRATEGIES FOR LOW-COST COMPARISON OF FILE COPIES †

Daniel Barbará
Richard J. Lipton

Department of Computer Science
Princeton University
Princeton, NJ 08544

ABSTRACT

In this paper we present a class of algorithms for comparison of remotely located file copies that use randomized signatures. We are able to show a simple technique that sends on the order of $4^f log(n)$ bits, where f is the number of differing pages that we want to diagnose and n is the number of pages in the file. We later show how to improve the bound in the number of bits sent, making them grow with f as $f\, log(f)$ and with n as $log(n) log(log(n))$. A third class of algorithms is presented, in which the number of signatures grows with f as fr^f, where r can be made to approach 1. This class of techniques exhibit a worse asymptotic behavior, but they perform very well in practice. Previously published algorithms ([Fu86] and [Ba88]) were aimed to diagnose 1 and 2 differing pages by sending $O(log(n) log(log(n)))$ and $O(log^2(n) log(log(n)))$ bits respectively. Moreover, our techniques prove to be very competitive in practice sending less bits than those in [Fu86] and [Ba88] for the cases $f = 1$ and $f = 2$ respectively.

September 8, 1988

A CLASS OF RANDOMIZED STRATEGIES FOR LOW-COST COMPARISON OF FILE COPIES †

Daniel Barbará
Richard J. Lipton

Department of Computer Science
Princeton University
Princeton, NJ 08544

1. INTRODUCTION

File replication is used in distributed systems in order to improve reliability and performance. However, due to human errors or hardware failures file copies may diverge. It then becomes necessary to compare the remotely located files and identify the differences.

Such a case arises, for example, in a triple modular redundant (TMR) database system that has been built at Princeton [Pi86]. The system is implemented on three SUN 120 computers, each with a full copy of a database. Transactions are submitted to any of the three nodes, but before it is executed it is reliably broadcast to the other two nodes. Once it is certain that all three nodes have the transaction, each node executes the transaction independently on its local database. The three results are sent to the user, who then uses voting to select the correct one. The system continues to operate with two computers until a failed computer is fixed and restarted. The restarted machine must then identify the portion of the database that is corrupted (if any) and then request a copy of that portion from the operational machines. This is where the file compare algorithm is used. The system can tolerate one arbitrary failure of a machine (e.g., a head crash, or the processor writing the wrong balance into an account) and still guarantees correct data and transaction results.

The development of methods to test mutual consistency of replicated files in distributed systems is a key issue for applications like the one described above. The more sophisticated approaches try to minimize the number of messages sent and to maximize the number of diverging pages that can be located [Me83,Fu86,Ba88]. These parameters are important in

† This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and by the Office of Naval Research under Contracts Nos. N00014-85-C-0456 and N00014-85-K-0465, and by the National Science Foundation under Cooperative Agreement No. DCR-8420948. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

the cases where the files are large and the cost of identifying the differences must be kept low. These strategies are all based in sending a number of *signatures* or check sums of sets of pages and running an algorithm that compares such signatures with the locally computed ones to locate the differing pages. In [Me83], $O(logn)$ message exchanges are necessary to locate the differing pages, where n is the total number of pages in the file. Exchanging messages between the nodes may slow down the process of diagnosis. However, the real drawback in this technique is that $O(n)$ signatures must be stored in each site, making the cost of managing large files prohibitive. The strategies in [Fu86] (**FU**) and [Ba88] (**BA**) send only one message containing $O(logn)$ and $O(log^2 n)$ signatures respectively. The strategy **FU** is able to locate one differing page (giving a superset of the differing pages if there is more than one fault), while the **BA** locates up to two differing pages. If more than these number of pages are different, the algorithms give a solution where the set of differing pages is guaranteed to be properly included with a probability that can be made to approach 1 by increasing the number of bits that compose each signature. Only $O(logn)$ and $O(log^2 n)$ signatures need to be stored in each site respectively.

The purpose of this paper is threefold. First, we want to establish a theoretical framework in which these techniques can be compared. Second, we want to present some randomized techniques that achieve the best bounds found so far in the literature. We also show that these techniques are very competitive in practice, exhibiting a good performance in terms of the number of bits that one needs to send. This paper also elaborates on two important issues that were ignored by previous work in the area. First, the signatures used to compare pages are not perfect. It is possible that two pages that differ in their contents render the same signature. The event of this happening becomes more unlikely as the size of the signatures grows, but its probability is not zero. In all our analysis, we take this error into consideration and bound the expected number of pages that can be falsely diagnosed. (A page can be falsely diagnosed as ''good'' when in reality the two copies disagree or can pass as ''bad'' when the two copies agree.) Previous work in the area has only suggested that the number of bits per signature has to be ''sufficiently large''. However, the impact of this issue over the complexity of the protocol has not been taken into consideration. The second issue is that all previous analysis counted the number of signatures that the algorithm sends. We make all our comparisons in terms of bits sent,that being the only universal measure known to compare the traffic imposed over the communications lines. To carry out the algorithm comparison, we use the following definition:

Definition 1.1: For a given one message strategy, $\beta(n, f, \delta)$ is the number of bits sent to diagnose up to f differing pages in a file with n pages keeping the probability of false diagnose bounded by δ.

By a one-message strategy, we mean a technique in which both sites compute a set of signatures over the actual file pages, and site A sends one message containing its signatures to site B. Having received the signatures, site B runs an algorithm to diagnose the differing pages, i.e., to pinpoint the pages that are different in copies a and b of the file. The techniques are set up to diagnose up to a given number of differing pages f. If there are f or fewer differing pages, the algorithm will diagnose *exactly* these pages with a probability less than or equal to δ. If more than f pages differ, the algorithm will find a superset of the actual set of differing pages.

We begin in section 2 by presenting a brief review of the other published approaches for remote file comparison that use one message and the bounds achieved by those techniques using definition 1.1.

In section 3, we present our first technique which achieves a bound of

$$\beta(n,f,\delta) = O\,(4^f(log\,(n) + \log(\frac{1}{\delta}))).$$

For practical purposes, we would like the dependency on f to be less dramatic. For that reason, we show in section 4, how to improve this bound to

$$\beta(n,f,\delta) = O\,(f(log\,(n) + \log(\frac{1}{\delta}))(log\,(log\,(n)) + log\,(f) + log\,(\frac{1}{\delta}))).$$

In section 5, we present a class of algorithms which exhibit a worse asymptotic bound than the one in section 4, but have a very good performance in practice. Section 6 compares all the techniques presented, and section 7 presents an example. In section 8, we reason about the role of randomness in this class of algorithms and the price one must pay to make them deterministic. Finally, in section 9 we offer some conclusions and suggestions for future work.

2. OVERVIEW OF PREVIOUS FILE COMPARE STRATEGIES

All the strategies assume that the file is divided into a collection of pages $P_1, P_2, ..., P_n$. For simplicity the assumption that n is a power of 2, i.e., $n = 2^m$ is made. This assumption, however can be dropped easily. We assume the pages belong to one of the two following disjoint sets:

$$G = \{P_i \,/\, P_i \textit{ is the same in both copies }\}$$

$$B = \{P_i \,/\, P_i \textit{ is not the same in both copies }\}$$

That is, there is a group of differing pages that we consider to be in B (for ‘‘bad’’ pages) and a group of non-differing pages which are in G (for ‘‘good’’ pages).

For each page P_i we can compute a signature $sig(P_i)$. One can think of the signature as a check sum, although the are more sophisticated ways to compute them [Me83]. If the signature contains b bits, then the probability that two different pages have the same signature is 2^{-b}.

The signatures for a set of pages can be combined into one by performing an exclusive or of the individual signatures. If the original signatures have b bits, the combined signature will also have b bits. The combined signature can be used to compare the pages in the set to their copies in a single operation. If the combined signatures are identical, then the copy pages are equal with a probability that depends on the size of the signatures sent. The probability that one or more of the j pages are different but have the same combined signature is approximately 2^{-b} [Fu86]. If the combined signatures are different, then there are some differences among the pages. An important point to notice is that the combined signature does not by itself identify the pages that are different.

In mechanisms **FU** and **BA**, the signatures are organized in a two-dimensional array. This array has m rows and k columns in **FU** and $\frac{m(m+1)}{2} + 1$ rows and 2 columns in **BA**. In both cases, the diagnosing algorithm builds for each row a set with the pages that correspond to the differing signatures for the row. The solution set is computed as the intersection of these sets. It can be shown that the solution set contains exactly the differing pages if $|B|=1$ in the strategy of **FU** or $|B|=2$ in **BA**. If more pages are differing, the solution set identifies a superset of the pages that differ. We shall put the strategies **FU** and **BA** in the context of our metric, β, in order to establish a framework to compare the techniques.

In all our analysis, we will be interested in bounding the expected number of pages that are diagnosed incorrectly. In all our analysis we make use of the following trivial lemma.

Lemma 2.1 Let $\{\tilde{X_i}\}$ be random variables that can take values from $\{0,1\}$. If we define $\tilde{Y} = \sum_i X_i$, and $E[\tilde{Y}] \leq \varepsilon$, then $Prob[\tilde{Y} \geq 1] \leq \varepsilon$. ○

Lemma 2.1 says that proving that the expected value of the sum of a series of random variables that take values from $\{0,1\}$ is bounded from above implies that the probability of having one of them take the value 1 is also bounded.

Now, defining

$$\tilde{X_i} = \begin{cases} 1 & \text{if } P_i \textit{ is incorrectly diagnosed} \\ 0 & \textit{otherwise} \end{cases}$$

and $\tilde{Y} = \sum_i \tilde{X}_i$, then the expected number of pages that are falsely diagnosed is

$$E_F = E[\tilde{Y}] = \sum_i E[\tilde{X}_i] = n\ Prob[a\ page\ P_i\ is\ incorrectly\ diagnosed]$$

A page can be incorrectly diagnosed in one of two ways. Either the page is in G (a ''good'' page) and is included in the set of diagnosed pages, or the page is in B (a ''bad'' page) and is not included in the set of diagnosed pages. Thus, denoting by T the set of diagnosed pages we have:

$$E_F = n(Prob[P_i \in G \text{ and } P_i \in T] + Prob[P_i \in B \text{ and } P_i \notin T]) \qquad (1)$$

Now,

$$Prob[P_i \in G \text{ and } P_i \in T] = Prob[P_i \in T \,/\, P_i \in G]Prob[P_i \in G]$$

and,

$$Prob[P_i \in B \text{ and } P_i \notin T] = Prob[P_i \notin T \,/\, P_i \in B]Prob[P_i \in B].$$

With f differing pages,

$$Prob[P_i \in B] = \frac{f}{n}$$

and,

$$Prob[P_i \in G] = \frac{(n-f)}{n}$$

thus, the expected number of pages incorrectly diagnosed can be expressed as:

$$E_F = E_G + E_B \qquad (2)$$

where

$$E_G = (n-f)Prob[P_j \in T \,/\, P_j \in G]$$

and

$$E_B = f\ Prob[P_j \notin T \,/\, P_j \in B].$$

Now, we can start by bounding the probabilities, for the different strategies. We begin with the following lemma.

Lemma 2.2 In the strategies **FU** and **BA**, for a given δ, the term E_B is less than or equal to δ if

$$b \geq log(s) + log(\frac{1}{\delta}) + log(f)$$

where s is the number of rows sent by those mechanisms, and $f = 1$ for the mechanism **FU** and $f = 2$ for the one in **BA**.

Proof: In both mechanisms, the only way a ''bad'' page can be left out of the solution is if that page is in a signature that fails to point out the difference between the pages. This happens with probability 2^{-b}, thus the term is bounded by $fm2^{-b}$. So, if we want the term to be less than or equal to δ we have to chose b as shown. ○

The term $log\,(\frac{1}{\delta})$ comes from the level of confidence that one wants to inject to the strategy in order to avoid a ''bad'' page passing unnoticed.

In **FU** and **BA**, the signatures are chosen carefully to make sure that no good page gets into the solution when there are up to 1 or 2 differing pages respectively. Therefore, the $E_G = 0$ in both strategies. Now, we can state the following theorems:

Theorem 2.1 The strategy **FU** diagnoses correctly 1 differing page in an n page size file with $E_F < \delta$, using

$$\beta(n, 1, \delta) = O\,(log\,(n)(log\,(log\,(n)) + \log(\frac{1}{\delta})))$$

bits.

Proof: By using lemma 2.2 and the fact that the number of rows sent by **FU** is $s = O\,(log\,(n))$ we arrive at the result. ○

Theorem 2.2 The strategy **BA** diagnoses correctly up to 2 differing pages in an n page size file with $E_F < \delta$, using

$$\beta(n, 2, \delta) = O\,(log^2(n)(log\,(log\,(n)) + \log(\frac{1}{\delta})))$$

bits.

Proof: By using lemma 2.2 and the fact that the number of rows sent by **FU** is $s = O\,(log^2(n))$. ○

In [Fu86] and [Ba88] the claim was that the number of signatures sent were $O\,(log\,(n))$ and $O\,(log^2(n))$ respectively. The reader should recall that the extra terms in β that we found in Theorems 2.1 and 2.2 are due to the fact that we are taking into consideration the actual number of bits sent. As we saw in Lemma 2.2, the number of bits per signature has to be a function of the number of signatures sent, the number of differing pages one wants to catch and the bound in the expected number of falsely diagnosed pages.

The problem of comparing file copies can be reduced to that of correcting bit patterns in algebraic coding theory, as has been suggested by Madej [Ma88]. We end this section by an analysis of this technique.

If $sig\,(P_i)$ has b bits, we can construct b words of m parity check symbols of the form

$$\gamma_1(j),...,\gamma_m(j)$$

with $j = 1,..,b,$ using a BCH code that can correct up to f errors [Pe61].

Now, instead of sending the signatures, site A sends $\gamma_1,...,\gamma_m$, where

$$\gamma_k = \gamma_k(1),...,\gamma_k(b).$$

Site B treats

$$w_j = (sig(P'_1),...,sig(P'_n),\gamma_1(j),...,\gamma_m(j))$$

as a BCH codeword, with the possible errors occurring among $(sig(P'_1),...,sig(P'_n))$. Decoding the words $w_1,...,w_m$, B can find the differences between the signatures $sig(P_i)$ and $sig(P'_i)$.

This mechanism has two drawbacks, however. The first is that the decoding of the codewords is considerably complex. The second, and most important, is that BCH codes are designed to correct up to a specific number of errors in the code words. If more errors occur, the decoding algorithm makes no guarantees. Thus, if there are more than f differing pages in the copies, some of them may pass undetected by this method. In any of the other methods presented in this paper, the solution is always a superset of the actual set of differing pages.

In computing the expected number of pages that are falsely diagnosed in this method, we should take into account that the only way a page P_i can be falsely diagnosed is if $sig(P_i) = sig(P'_i)$ and the pages P_i and P'_i are different. This happens with probability 2^{-b}.Thus, the term E_B is bounded by $f\,2^{-b}$. That establishes the following lemma.

Lemma 2.3. In the strategy **BCH**, $E_F < \delta$ if

$$b \geq log(f) + log(\frac{1}{\delta})$$

Proof: From the argument above. ○

To compute the number of parity bits needed, we can use the following theorem from Peterson [Pe61].

Theorem 2.3. [Pe61] For any positive integers t and $f < \frac{n}{2}$, there is a BCH code of length $n = 2^t - 1$ which corrects all combinations of f or fewer errors and has no more than tf parity check symbols. ○

Theorem 2.3 establishes that the number of parity symbols needed by the BCH code is $m = O(f\,log(n))$. The following theorem establishes the bound for the **BCH** strategy.

Theorem 2.4. The strategy **BCH** diagnoses correctly up to f differing pages in an n page size file with the $E_F < \delta$ using,

$$\beta(n,f,\delta) = O(f\,log(n)(log(f) + log(\frac{1}{\delta})))$$

bits.

Proof: The strategy sends m words γ_k of b bits each. Using Theorem 2.3 and Lemma 2.3, the bound is established. ○

As we shall see, a bound very close to that of Theorem 2.4 can be achieved by a randomized algorithm that, contrary to the **BCH** strategy, always finds a superset of the actual differing pages.

3. A FIRST BOUND

In this section we develop an algorithm that achieves the bound of $\beta(n,f,\delta) = O(4^f(log(n) + \log(\frac{1}{\delta})))$ where n is the number of pages of the file, f is the number of differing pages that the algorithm is set up to diagnose with a level of confidence of δ. This algorithm uses time $O(nlog(n))$ to diagnose the differing pages.

As usual, the file is divided into n pages $P_1, \ldots, P_n$. There are two copies of the file and presumably some of the pages may differ. The sites need to agree on m random sets $\tilde{S}_1, \tilde{S}_2, \ldots, \tilde{S}_m$, where each set $\tilde{S}_i \subseteq \{P_1, P_2, ..., P_n\}$, before any comparison can be performed. To do this, one of the sites computes the sets and sends them to the other. (The sets need only to be computed and sent **once**, not every time that the file comparison is to be performed.) A page P_j is in the set $\tilde{S}_i$ with probability $\frac{1}{2}$.

When a file comparison is needed, both sites compute m signatures over the sets $\tilde{S}_i$. The $i-th$ signature is constructed as the exclusive or of the signature functions of the pages in $\tilde{S}_i$. We call the $i-th$ signature computed by sites A and B, c_i^1 and c_i^2 respectively. One of the sites sends its signatures to the other and this site performs the comparison by creating the *syndrome matrix* with elements

$$\alpha_i = \begin{cases} 0 & \text{if } c_i^1 = c_i^2 \\ 1 & \text{if } c_i^1 \neq c_i^2 \end{cases}$$

Having computed the syndrome matrix, the site proceeds to build a set T of pages that are diagnosed as differing. To do so, the site includes every page that appears in at least $m\delta_f$ sets $\tilde{S}_i$ for which $\alpha_i = 1$. We shall show how to choose δ_f, given δ and f later. Building T takes $O(m\log(n))$ time. Intuitively, the algorithm proceeds by watching pages that are present when a "faulty" signature occurs. If a page is present in too many "faulty" signatures, then it becomes "suspicious" and is included in the set of culprits. (The same way that the police might arrest a citizen that happens to be at the scene of many crimes, on the grounds of circumstantial evidence.) Figure 3.1 shows the diagnosing algorithm, which we have called **FIND_SUSPECTS** **(FS)** for obvious reasons.

Algorithm **FIND_SUSPECTS**

```
$T := \emptyset$;
for $j := 1$ to $m$ do
  if $\alpha_j = 1$ then
    for $i := 1$ to $n$ do
      if $P_i \in \tilde{S}_j$ then
        $count[i] = count[i] + 1$
for $i := 1$ to $n$ do
  if $count[i] \geq m\delta_f$ then
    $T := T \cup \{P_i\}$ ;
```

Figure 3.1

We are now interested in showing that with m large enough we can be confident that every page that appears in T is also in B and vice versa. Therefore, we would like to bound E_F as expressed in equation (2). We will show that by sending enough bits, we can make (2) arbitrarily small. To do so, we need to bound the terms E_G and E_B. Both of them can be computed using the tail of a binomial distribution as we will show later. To bound them, we need the following pair of lemmas.

Lemma 3.1 Let $\tilde{X}_i$ be a Bernoulli random variable as follows:

$$\tilde{X}_i = \begin{cases} 1 & \textit{with probability } p \\ 0 & \textit{with probability } 1-p \end{cases}$$

then

$$Prob\,[\sum_{i=1}^{m} \tilde{X}_i \geq (p+\varepsilon_1)m\,] \leq e^{-c\varepsilon_1^2 m}$$

where $c > 0$ is a constant (independent of m and ε_1) and $\varepsilon_1 > 0$ is small.

Proof: Using Chernoff [Ch52] inequality, we have:

$$Prob\,[\sum_{i=1}^{m} \tilde{X}_i \geq (p+\varepsilon_1)m\,] \leq e^{mk_p}$$

where

$$k_p = (1-(p+\varepsilon_1))log\,(\frac{1-p}{1-(p+\varepsilon_1)}) + (p+\varepsilon_1)log\,(\frac{p}{p+\varepsilon_1}) \qquad (3)$$

now,

$$log\,(1+x) = x - \frac{x^2}{2} + O\,(x^3)$$

for x small. Thus, equation (3) becomes:

$$k_p = -\frac{\varepsilon_1^2}{2}\left(\frac{1}{1-(p+\varepsilon_1)} + \frac{1}{p+\varepsilon_1}\right) + O\,(\varepsilon_1^3) \qquad (4)$$

Now clearly, $\frac{1}{p} \geq \frac{1}{p+\varepsilon_1}$ and $\frac{2}{1-p} \geq \frac{1}{1-p-\varepsilon_1}$, provided that $\varepsilon_1 \leq \frac{1-p}{2}$. Thus,

$$Prob\,[\sum_{i=1}^{m} \tilde{X}_i \geq (p+\varepsilon_1)m\,] \leq e^{-\frac{\varepsilon_1^2}{2}(\frac{1}{p} + \frac{2}{1-p})}$$

○

Lemma 3.2 Let $\tilde{X}_i$ be a Bernoulli random variable as follows:

$$\tilde{X}_i = \begin{cases} 1 \ \textit{with probability}\ p \\ 0 \ \textit{with probability}\ 1-p \end{cases}$$

then

$$Prob\,[\sum_{i=1}^{m} \tilde{X}_i \leq (p-\varepsilon_2)m\,] \leq e^{-c\varepsilon_2^2 m}$$

where $c > 0$ is a constant (independent of m and ε_2) and $\varepsilon_2 > 0$ is small.
Proof: Again by using Chernoff [Ch52] inequality, we have:

$$Prob\,[\sum_{i=1}^{m} \tilde{X}_i \leq (p-\varepsilon_2)m\,] \leq e^{mk_p}$$

where,

$$k_p = (1-(p-\varepsilon_2))log\,(\frac{1-p}{1-(p-\varepsilon_2)}) + (p-\varepsilon_2)log\,(\frac{p}{p-\varepsilon_2})$$

and by the same approximation used in lemma 3.1,

$$Prob\,[\sum_{i=1}^{m} \tilde{X}_i \leq (p-\varepsilon_2)m\,] \leq e^{-\frac{\varepsilon_2^2}{2}(\frac{2}{p} + \frac{1}{1-p})}$$

provided that $\varepsilon_2 \leq \frac{p}{2}$. ○

Now we can use lemmas 3.1 and 3.2 to bound the conditional probabilities in (3) by noticing that for a page in B, the probability of being in a ''faulty'' signature, i.e., one with $\alpha_i = 1$ is

$$p = \frac{1}{2}(1 - 2^{-b}) \tag{5}$$

where b is the number of bits used per symbol. For a page in G, the same probability becomes:

$$p = \frac{1}{2}(1 - 2^{-f})(1 - 2^{-b}) \tag{6}$$

Now, choosing $\delta_f = \frac{1}{2}(1 - 2^{-b})(1 - 2^{-f}) + 2^{-f}$, we can state the following lemmas.

Lemma 3.3

$$E_B \leq fe^{-c_1 \frac{m}{4^f}} \tag{7}$$

where $c_1 = \frac{1}{2}(\frac{1}{p} + \frac{2}{1-p})$ and p as in equation (5).

Proof: By using lemma 3.2 and $p{-}\varepsilon_2 = \delta_f$, we arrive at equation (7). ○

Lemma 3.4

$$P_G \leq (n-f)e^{-c_2 \frac{m}{4^f}} \tag{8}$$

where $c_2 = \frac{1}{2}(\frac{1}{p} + \frac{2}{1-p})$ and p as in equation (6).

Proof: By using lemma 3.2 and $p{-}\varepsilon_1 = \delta_f$, we arrive at equation (8). ○

Using Lemmas 3.3 and 3.4 we can rewrite equation (2) as follows:

$$E_F \leq fe^{-c_1 4^{-f} m} + (n-f)e^{-c_2 4^{-f} m}) \tag{9}$$

making $c \leq min(c_1, c_2)$, (9) becomes:

$$E_F \leq ne^{-c 4^{-f} m} \tag{10}$$

In fact, we can make $c = 1.5$, since both c_1 and c_2 can be proven to be greater than 1.5. If we want this probability to be less than δ, then m should be chosen to satisfy:

$$m \geq \frac{4^f(log(\frac{1}{\delta}) + \log(n))}{1.5} \tag{11}$$

We can now state the following theorem:

Theorem 3.1 With the technique **FS**, we can diagnose up to f differing pages in an n page file with a probability of false diagnosis less than or equal to δ, sending $\beta(n,f,\delta) = O(4^f(log(n) + log(\frac{1}{\delta})))$ bits.

Proof: By equation (11). ○

The number of bits per signature b plays a very small role here. Having bounded the constant c as 1.5, we are better off by making $b=1$, so the number of bits sent is equal to m.

4. FINDING INNOCENT PAGES

In this section we shall improve the bound found in Theorem 3.1 by developing a technique that uses the signatures that agree on both copies.

In this technique, the sites agree in m randomly chosen sets of pages $\tilde{S}_1, \tilde{S}_2, \ldots, \tilde{S}_m$. Each set is chosen so that a page P_i is in set $\tilde{S}_j$ with probability $\frac{1}{f}$. To perform a comparison, site A sends the actual signatures c_j^1 of the sets to site B and site B compares them with its own signatures, (c_j^2), building again the syndrome matrix as

$$\alpha_j = \begin{cases} 0 & \text{if } c_j^1 = c_j^2 \\ 1 & \text{if } c_j^1 \neq c_j^2 \end{cases}.$$

Intuitively, in this algorithm we discard pages that are in signatures that agree and form the set T with the rest of the pages. (The same way the police would discard somebody from the list of suspects if this person has a strong alibi.) The algorithm to diagnose the pages is shown in Figure 4.1.

The set S in the algorithm of Figure 4.1 is the whole set of pages. The T_g contains the pages that are diagnosed as ''good'' by the algorithm, and finally, the set T contains the pages diagnosed as ''bad'' pages.

Again, we have to bound E_F in equation (2). We begin with the second term of the equation, by stating a lemma similar to Lemma 2.2.

Lemma 4.1 In the strategy **FIND_INNOCENTS (FI)**, the term E_B is bound by $\frac{\delta}{2}$ if

$$b \geq log(m) + log(f) + log(\frac{2}{\delta})$$

Algorithm **FIND_INNOCENTS**

```
$T_g := \emptyset$;
**for** $j := 1$ **to** $m$ **do**
    **if** $\alpha_j = 0$ **then**
        **for** $i := 1$ **to** $n$ **do**
            **if** $P_i \in \tilde{S}_j$ **then**
                $T_g = T_g \cup \{P_i\}$
$T = S - T_g$;
```

Figure 4.1

is the number of bits sent in each signature.

Proof: Similar to Lemma 2.2. ○

We turn our attention to the other term in equation (2). It is easy to see that a non-differing page is in T_g if there is a signature to which the page belongs and has no "bad" page in it. Since a page gets into a signature with probability $\frac{1}{f}$, the probability for a non-differing page to be in T_g is $\frac{1}{f}(1 - \frac{1}{f})^f$. Therefore,

$$Prob\,[P_i \in T \,/\, P_i \in G] = (1 - \frac{1}{f}(1 - \frac{1}{f})^f)^m$$

this equation can be approximated by

$$Prob\,[P_i \in T \,/\, P_i \in G] \sim e^{-\frac{m}{ef}} \tag{12}$$

Using the results, equation (2) can be rewritten for this case as:

$$E_F \leq \frac{\delta}{2} + (n - f)e^{-\frac{m}{ef}} \tag{13}$$

and if we want to make this probability less than or equal δ, we should choose

$$m \geq ef(log\,(n{-}f) + log\,(\frac{2}{\delta})) \tag{14}$$

With equation (14) and Lemma 4.1, we can prove the following Theorem.

Theorem 4.1 The technique **FI**, achieves a bound

$$\beta(n,f,\delta) = O\,(f\,(log\,(n) + log\,(\frac{1}{\delta}))(log\,(log\,(n)) + log\,(f) + log\,(\frac{1}{\delta})))$$

Proof: By equation (14) and Lemma 4.1. ○

5. SCREENING OUT PAGES

In this section we present a class of strategies that exhibit a good practical behavior, although their asymptotic bound is not as good as the one found for the strategy in section 4. We begin with the simplest technique in the family and then generalize the results to the whole class of algorithms.

In this strategy each of the two sites compute m pairs of signatures, each pair consisting of the signature of a set $\tilde{S}_j$ and its complement $\tilde{S}_j^c$ (i.e., any page not in $\tilde{S}_j$ is in $\tilde{S}_j^c$). As before, the pages in $\tilde{S}_j$ are selected at random with every page having a probability $\frac{1}{2}$ of being in the set. A page that is not in $\tilde{S}_j$ belongs to $\tilde{S}_j^c$. This idea is similar to the one used in [Fu86], except that the two signatures in the same row do not have to contain the same number of pages. Again, the sites agree on the sets once before doing any comparison. For a file comparison, both sites compute the signatures of the sets and their complements c_{1j} and c_{2j}, for $j = 1,...,m$ and one of the sites sends the signatures to the other. This site computes an m by 2 syndrome matrix as follows:

$$\alpha_{1j} = \begin{cases} 0 & \text{if } c_{1j}^1 = c_{1j}^2 \\ 1 & \text{if } c_{1j}^1 \neq c_{1j}^2 \end{cases}$$

$$\alpha_{2j} = \begin{cases} 0 & \text{if } c_{2j}^1 = c_{2j}^2 \\ 1 & \text{if} c_{2j}^1 \neq c_{2j}^2 \end{cases}$$

Having built the matrix, we can run the following algorithm that outputs the set T as the set of pages that differ in the two copies. Intuitively, the algorithm looks at the instances in which only one of the signatures differ, and discards the pages that are in the non-differing signature.

We call the algorithm **SCREEN(2)**, because the pages are ''screened out'' as the algorithm proceeds. The intuition here is to discard pages that appear in signatures that agree. Every time a row is found where $\alpha_{1l} \neq \alpha_{2l}$, a subset of pages is left out of the solution. We call these rows *distinguished* rows.

Again, we are interested in showing that with this algorithm, E_F (equation (1)) can be made arbitrarily small by choosing m and b properly. As before, $B = \{$ all pages for which the copies are different $\}$ and $G = \{$ all pages that are the same in both copies $\}$.

Algorithm **SCREEN(2)**

```
$T := \{P_0,....,P_{n-1}\}$;
for $j := 1$ to $m$ do
    if $\alpha_{1j} \neq \alpha_{2j}$ then
        if $\alpha_{1j} = 1$ then
            $T := T \cap \tilde{S}_j$
        else
            $T := T \cap \tilde{S}_j^c$;
```

Figure 5.1

We begin by bounding E_B by noticing that a differing page will not be in the T_j of a distinguished row j only if the signature fails to pinpoint the difference, which happens with probability smaller than 2^{-b}. Now,

$$E_B \leq fm2^{-b} \tag{15}$$

By making

$$b \geq \log_2 m + \log_2 f + \log_2(\frac{2}{\delta}) \tag{16}$$

we can make E_b to be less than $\frac{\delta}{2}$.

We turn our attention to E_G. We can compute it as follows:

$$E_G = (n-f)Prob[P_i \in T / P_i \in G] = (n-f)(1-2^{-f})^m \tag{17}$$

this comes from the fact that a non-differing page can only be in T if it is not screened out in any of the rows, i.e., if it happens to be together with differing pages in every single row.

Now,

$$E_F \leq \frac{\delta}{2} + ((n-f)(1-2^{-f})^m) \tag{18}$$

Again, since we want E_F to be smaller than δ, we should choose m to satisfy:

$$m \geq \frac{\log_2(n-f) + \log_2(\frac{2}{\delta})}{\log_2(\frac{1}{1-2^{-f}})} \tag{19}$$

Using the equations above, we can now prove the following Theorem.

Theorem 5.1 The mechanism **SCREEN(2)** achieves

$$\beta(n,f,\delta) = O\,(2^f(log\,(n) + log\,(\frac{1}{\delta}))(f + log\,(log\,(n)) + log\,(\frac{1}{\delta}))).$$

Proof: Using the approximation:

$$-log\,(1-x) = x + \frac{x^2}{2} + O\,(x^3)$$

we can prove that

$$\frac{1}{log\,(\frac{1}{1-2^{-f}})} = \frac{2^f}{1 + \frac{2^{-f}}{2} + O\,(2^{-2f})} = O\,(2^f)$$

thus, using equation (19)

$$m = O\,(2^f(log\,(n) + log\,(\frac{2}{\delta})))$$

now,

$$b = O\,(log\,(m) + log\,(\frac{2}{\delta}) + log\,(f)) = O\,(f + log\,(log\,(n)) + log\,(\frac{2}{\delta}) + log\,(f))$$

and the theorem follows. ○

From a practical point of view, as suggested in [Fu86], we do not need to send $2mb$ bits for this technique. In each row, the second signature can be recovered by using the first signature and the signature for the whole set of pages. Thus, only $(m + 1)b$ bits are necessary.

We now generalize the previous analysis to define a class of randomized mechanisms and we show that we can improve the bound on Theorem 5.1. We will do so by allowing each row to be composed of k signatures. Thus, for each row j, there are k sets $S_{1j}, S_{2j}, \ldots, S_{kj}$. A page is equally likely to be in any of the sets with probability $\frac{1}{k}$. Again, the sets are agreed upon by the sites before any comparison is performed. When a comparison is to be performed, the sites compute the actual signatures over the sets and, as before one of the sites computes a syndrome matrix of size m by k with elements

$$\alpha_j^i = \begin{cases} 0 & \text{if } c_{ij}^1 = c_{ij}^2 \\ 1 & \text{if } c_{ij}^1 \neq c_{ij}^2 \end{cases}$$

where $1 \le j \le m$ and $1 \le i \le k$.

The solution is found by intersecting sets for which the signatures differ. If we send k signatures and $0 < k_1 \leq k$ of them differ, we will intersect the solution so far with the union of the subsets that compose those k_1 signatures. Again, pages that are in sets where the signatures agree are screened out of the solution. The modified algorithm to diagnose page differences is presented in Figure 5.2.

Algorithm **SCREEN***(k)*

```
T := $\{P_0,....,P_{n-1}\}$;
for $j := 1$ to $m$ do
   $T_j = \varnothing$
   if$\exists\ \alpha_j^i \neq 1$ then
      for every $i$ such that $\alpha_j^i = 1$
         $T_j = T_j \cup \tilde{S}_{ij}$
   $T := T \cap T_j$
```

Figure 5.2

By an analysis identical to the one presented above, we can show that

$$E_F \leq \frac{\delta}{2} + ((n-f)(1-r^{-f})^m \tag{20}$$

where $r = \dfrac{k}{k-1}$. Equation (20) follows from the fact that now the probability of a non-differing page not being screened out in a row is 1 minus the probability that no differing page is in the set where the non-differing page is located. The probability of a differing page being in any of the other $k-1$ signatures is $\dfrac{k-1}{k}$, thus

$$E_G = (n-f)(1 - \sum_{i=1}^{k} \frac{1}{k}(\frac{k-1}{k})^f)$$

and thus,

$$E_G = (n-f)(1-(\frac{k}{k-1})^{-f})^m$$

By using equation (20), we can state that m should satisfy

$$m \geq \frac{\log_2(n-f) + \log_2(\frac{2}{\delta})}{\log_2(\frac{1}{1-r^{-f}})} \tag{21}$$

Equation (21) serve as the basis to establish the bound for these techniques, as proven in the following theorem.

Theorem 5.2 The mechanism **SCREEN(k)** sends $\beta(n,f,\delta) = O(r^f(log(n) + log(\frac{1}{\delta}))(f + log(log(n)) + log(\frac{1}{\delta})))$ bits to diagnose f differing pages in a file of n pages with a level of confidence δ

Proof: Using again the approximation:

$$-log(1-x) = x + \frac{x^2}{2} + O(x^3)$$

we can prove that

$$\frac{1}{log(\frac{1}{1-r^{-f}})} = \frac{r^f}{1 + \frac{r^{-f}}{2} + O(r^{-2f})} = O(r^f)$$

thus, using equation (21)

$$m = O(r^f(log(n) + \log(\frac{2}{\delta})))$$

and the theorem follows. ○

An important corollary follows

Corollary 5.1 By increasing k, the bound can be made almost linear in f

Proof: Since $r = \frac{k}{k-1}$, as k grows, r tends to 1. ○

As before, only $(m(k-1)+1)b$ bits are necessary to perform the comparison, since the signature of the last set of a row can be recovered from the previous $k-1$ signatures and the one for the entire set of pages.

6. COMPARING THE TECHNIQUES

In this section, we compare all the techniques developed in sections 3, 4 and 5. To do so, we plot β as a function of f for the different mechanisms.

In table 6.1, we show the bounds for the different techniques presented and those of strategies **FU** and **BA**.

Notice again that the strategies **FU** and **BA** are aimed at diagnosing up to 1 and 2 differing pages respectively. For a fixed f, the best bound achieved is $O(log(n))$ by all three of the new strategies presented. For an arbitrary f, the strategy **FI** achieves the best bound.

Strategy	β
FS	$O(4^f(log(n) + log(\frac{1}{\delta})))$
FI	$O(f(log(n) + log(\frac{1}{\delta}))(log(log(n) + log(f) + log(\frac{1}{\delta})))$
SCREEN*(k)*	$O(r^f(log(n) + log(\frac{1}{\delta}))(f + log(log(n)) + log(\frac{1}{\delta})))$
FU	$O(log(n)(log(log(n)) + log(\frac{1}{\delta})))$
BA	$O(log^2(n)(log(log(n)) + \log(\frac{1}{\delta})))$
BCH	$O(f\ log(n)(log(f) + log(\frac{1}{\delta})))$

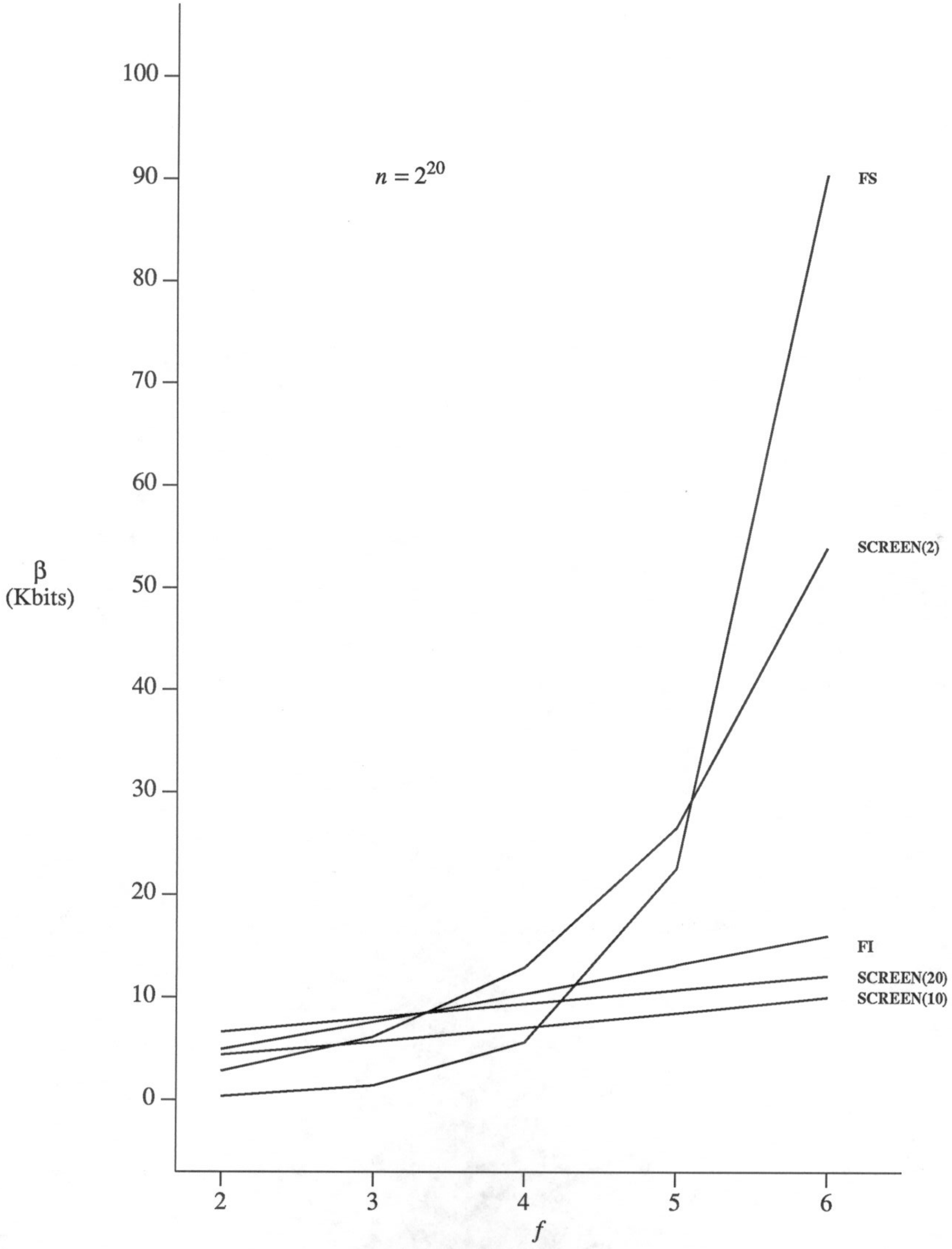
Table 6.1

In Figure 6.1, we show $\beta(n,f,\delta)$ as a function of f for the techniques **FS**, **FI**, and **SCREEN(2), SCREEN(10), SCREEN(20)**, using a file size of 2^{20} pages and for a level of confidence $\delta = 2^{-20}$. We see that in the case of **FS**, the number of bits that we have to send grows exponentially fast with the number of differing pages that we want to diagnose. For $f = 2$, however, the algorithm sends only 350 bits to perform the diagnosis, a remarkable figure when compared with previous approaches. For the same level of confidence and file size, [Ba88] needs to send 221 signatures of 32 bits each, i.e., 7072 bits. To diagnose a single differing page the strategy on [Fu86] sends 21 signatures of 32 bits, i.e., 672 bits.

For the case of the **SCREEN** mechanisms, we can see that as k grows, the behavior becomes more and more linear with f. However, after a point, the gains obtained are offset by the number of extra signatures per row sent. This is the case of $k = 20$ which is outperformed by $k = 10$. It is also worth noticing that the algorithm **FS** outperforms all the **SCREEN** mechanisms for $f \leq 4$. From there on, those mechanisms send less number of bits.

The mechanism **FI** exhibits a performance close to the one for **SCREEN(10)** and **SCREEN(20)**, but is outperformed by them almost in the entire region of values of f plotted. In figure 6.2, we show a different region of values of f (10 to 20) and in there we see that eventually **FI** gets better than **SCREEN(10)**. The strategy **FI** finally outperforms **SCREEN(20)** for values of f larger than 38.

As a conclusion, we can say that for small values of f, it is advisable to use the algorithm **FS**. As f grows, the other techniques become the choice. For large values of f, we either have to choose **FI** or a technique **SCREEN***(k)* with a value of k sufficiently large.



100
90
80
70
60
50
40
30
20
10
0
β
(Kbits)
$n = 2^{20}$
FS
SCREEN(2)
FI
SCREEN(20)
SCREEN(10)
2
3
4
5
6
f

Figure 6.1

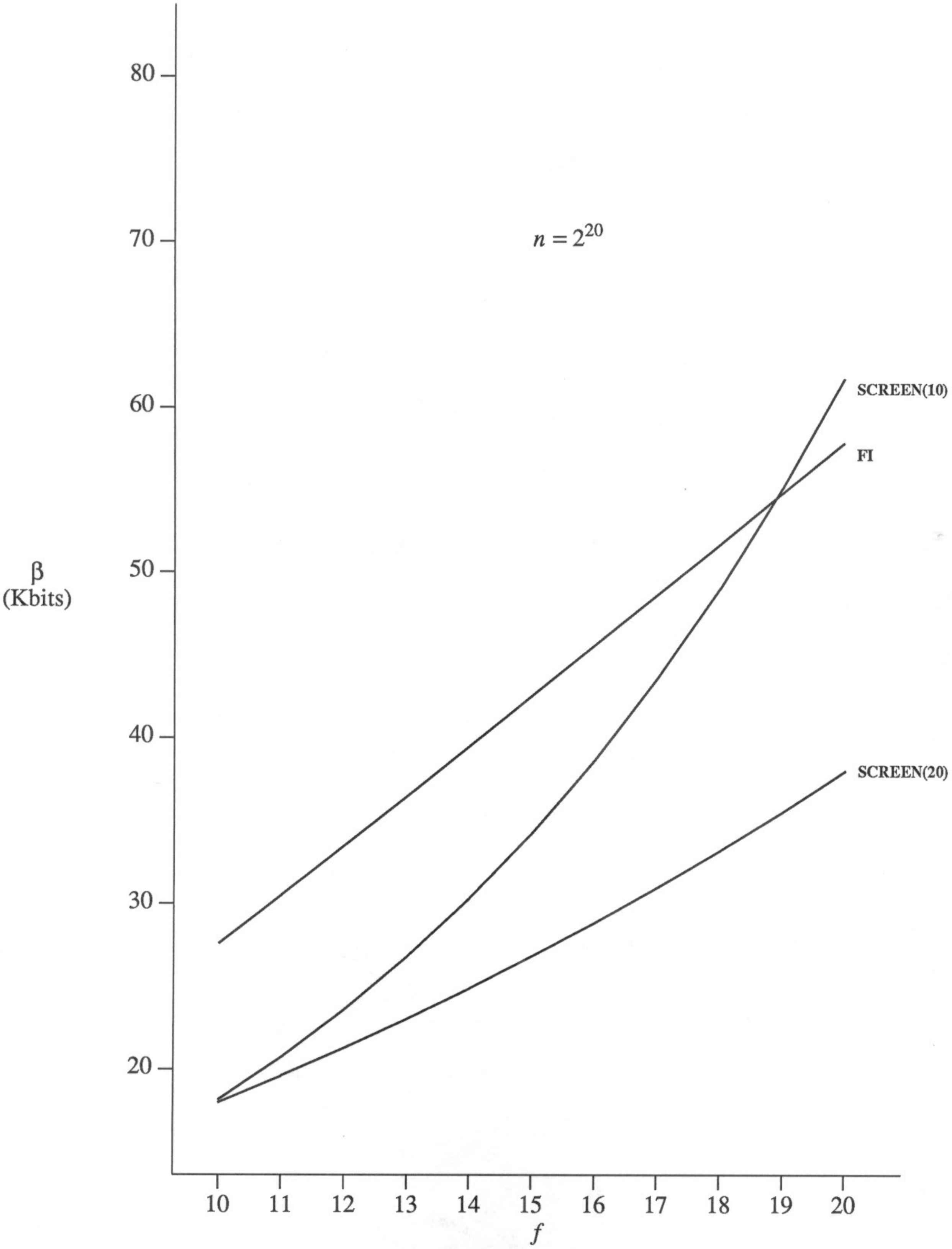

80
70
60
50
40
30
20
β
(Kbits)
$n = 2^{20}$
SCREEN(10)
FI
SCREEN(20)
10
11
12
13
14
15
16
17
18
19
20
f

Figure 6.2

7. AN EXAMPLE

In this section we present one example of how one of the proposed mechanisms work. We will call the pages by their numbers. In Figure 7.1, we show a 2 by m array for the case $n = 16$, with $m = 8$ for the **SCREEN***(4)* mechanism. The signatures were generated by running a random number generator for each page and placing the page in the one of the signatures with the same probability $(\frac{1}{4})$. (The fact that there are almost as many rows as pages in this example is irrelevant, since we are only interested in showing how the algorithm works.)

Assume that only pages 1 and 7 differ. Then, the T_js are as follows:

$$T_1 = \{1,7,9,11\}$$

$$T_2 = \{1,4,5,7,12,14,15\}$$

$$T_3 = \{0,1,4,5,7,11,12,13,15\}$$

$$T_4 = \{1,3,4,5,7,8,9,10,12,15\}$$

$$T_5 = \{0,1,6,7,9,11,12,13,14\}$$

$$T_6 = \{1,7,9,14\}$$

$$T_7 = \{0,1,5,7,9,10,11,12,13,14,15\}$$

$$T_8 = \{1,2,4,7,8,9,11,12,15\}$$

$$T_9 = \{0,1,2,3,5,4,6,7,8,11,13,15\}$$

$$T_{10} = \{0,1,2,3,4,6,7,8,11,13,15\}$$

And finally the intersection of all T_js gives us:

$$T = \{1,7\}$$

and the two pages are diagnosed without any non-differing page getting into the set T. In this example this is always the case for every pair of pages i,j, although this is not true in general.

Row	S_1	S_2	S_3	S_4
(1)	1 7 9 11	2 4 12 14 15	6 8	0 3 5 10 13
(2)	1 12 14 15	0 8 9 10 13	2 3 6 11	4 5 7
(3)	7 11 15	10 14	0 1 4 5 12 13	2 3 6 8 9
(4)	1 4 10 12	5 6 13	3 7 8 9 15	0 2 11 14
(5)	0 7 9 11 14	4 8 10	2 3 5 15	1 6 12 13
(6)	0 2 3 4 8 11 12 15	5	6 10 13	1 7 9 14
(7)	4 6	0 7 9 12 14 15	1 5 10 11 13	2 3 8
(8)	5 10 13 14	1 2 9 15	0 3 6	4 7 8 11 12
(9)	1 2 6 11 15	5 10 12 14	0 3 4 7 8 13	9
(10)	9 12 14	5	10	0 1 2 3 4 6 7 8 11 13 15

Figure 7.1
A random array of signatures for $n = 16$ and the mechanism SCREEN(4)

8. THE ROLE OF RANDOMNESS

Throughout this paper, the techniques presented have all been randomized. This fact raises some questions about the bounds that we have found. First, what is the role that randomness play in these bounds? Secondly, is it fair to compare those bounds with the ones achieved by techniques that are not random?

To answer those questions we first have to take a close look at the way randomness have been used in the techniques exposed. There are two places in which one uses randomness. The first one is in the signature functions. These functions are hashing functions and therefore are subject to the probability that two different pages produce a collision, i.e., give the same value for their signature. This is true in every published mechanism for file comparison that is based on check sums. The second place in which we use randomness is in the structure of the signatures themselves. Previous approaches selected the structure of the signature carefully so for a given number of differing pages (1 or 2) there was no chance of finding an instance for which the algorithm did not diagnose correctly those pages. Our approach differs in the sense that if one is allowed to look at the signatures and select the ''bad'' pages, it is conceivable that one could find an instance that defeats the algorithm, i.e., for which the algorithm diagnoses a larger set of pages than the actual set of differing ones. However, the number of instances for which this happens, is bound to be very small.

However, if one would like to make sure that no such instance can ever occur, there is a way of transforming our techniques to insure that, by making them deterministic in the structure of the signatures sent. To show that this statement is true, we use the algorithm **FI** presented in section 4 and show that by paying the right price, the algorithm can be made deterministic.

Theorem 8.1 Algorithm **FI** can be made deterministic by sending $O(f^2 log(n))$ signatures.

Proof: To make the algorithm work in all cases, we need to insure the following property

> For all tuples $x_0,x_1,...,x_f$ distinct, there exists $\tilde{S}_i$ such that $x_0 \in \tilde{S}_i$, $x_j \notin \tilde{S}_i$, for $j = 1,...,f.$ and $x_l \in \{P_1,...,P_n\}$, for $0 \le l \le f$.

This property says that in order for a ''good'' page to get in T_g (and subsequently not be in T), it is enough that there exists a signature to which the page belongs and no ''bad'' page is located in that signature. It is easy to see that the expected number of $f+1$-tuples of pages that fail to satisfy the property is given by: $n^{f+1}(1-\frac{1}{f}(1-\frac{1}{f})^f)^m$, which is $n^{f+1}e^{-\frac{m}{ef}}$. In order to find an arrangement of signatures that satisfies the property, this expected value has to be less than 1. Thus, we have to send on the order of $m = O(f^2 log(n))$ signatures to avoid any instance from defeating the algorithm. ○

As we can see, we pay the price by affecting the behavior of the bound in *f*. A similar analysis can be made for the other techniques.

9. CONCLUSIONS

We have presented a class of randomized mechanisms for low cost comparison of remotely located files. Several conclusions can be drawn from the results.

First, the mechanisms perform very well without having to resort to more elaborate signatures. In some cases, we can guarantee a good performance over a large range of the number of differing pages, while keeping the total number of signatures sent very low.

Also, we have presented a technique, **FI** that achieves a bound β, linear on the number of differing pages that one wants to diagnose and logarithmic in the size of the file. It remains to be proven whether this bound is optimal or not. In section 8, we showed that if one wants to do away with the randomness in choosing the signatures, the price to be paid is to increase the complexity of the algorithm to f^2. A deterministic technique that achieves this bound for an arbitrary *f*, is yet to be found. (The strategy **FU** actually achieves the bound, but only for $f = 1$.)

The results shown in section 6 prove that if one is expecting only a few differing pages, a mechanism like **FS** offers a very good performance. On the other hand, if the expected number of differing pages is larger, as it may happen after the crash of one of the machines, one is better off by using a **SCREEN***(k)* mechanism or the **FI** mechanism.

ACKNOWLEDGEMENTS

We would like to thank R.Alonso, H. Garcia-Molina, and J.L. Palacios for helpful technical discussions and comments.

10. REFERENCES.

[Ba88] D. Barbará, B. Feijoo and H. Garcia-Molina Exploiting Symmetries for Low-Cost Comparison of File Copies, *Proc. International Conference on Distributed Computing Systems*, San Jose, June 1988.

[Ch52] H. Chernoff. A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the Sum of Observations. *Annals of Math. Stat.*, 23,493-509.

[Fu86] W.K. Fuchs, K. Wu and Abraham J. Low-Cost Comparison and Diagnosis of Large Remotely Located Files, *Proc. Fifth Symposium on Reliability in Distributed Software and Database Systems*, January 1986, pp. 67-73

[Ma88] T. Madej. Private communication.

[Me83] J. Metzner A Parity Structure for Large Remotely Located Replicated Data Files *IEEE Transactions on Computers*, Vol. C-32, No. 8, August 1983.

[Pe61] W.W. Peterson, *Error-Correcting Codes*, MIT Press, Cambridge, MA, 1961.

[Pi86] F. Pittelli and H. Garcia-Molina Database Processing with Triple Modular Redundancy, *Proc. Fifth Symposium on Reliability in Distributed Software and Database Systems*, January 1986, pp. 95-103

[Pi87] F. Pittelli and H. Garcia-Molina Recovery in a Triple Modular Redundancy Database System, *Proc. Seventh International Conference on Distributed Computing Systems*, Berlin, September 1987.