12-19-1990

A Tight Upper Bound on the Benefits of Replication and Consistency Control Protocols

Donald B. Johnson
Dartmouth College

Larry Raab
Dartmouth College

A TIGHT UPPER BOUND ON THE BENEFITS OF REPLICATION AND CONSISTENCY CONTROL PROTOCOLS

Donald B. Johnson and Larry Raab

Technical Report PCS-TR90-157

A Tight Upper Bound on the Benefits of Replication and Consistency Control Protocols

Donald B. Johnson* Larry Raab†
Dartmouth College‡

December 19, 1990

Summary.

We present an upper bound on the performance provided by a protocol guaranteeing mutually exclusive access to a replicated resource in a network subject to component failure and subsequent partitioning. The bound is presented in terms of the performance of a single resource in the same network. The bound is tight and is the first such bound known to us. Since mutual exclusion is one of the requirements for maintaining the consistency of a database object, this bound provides an upper limit on the availability provided by any database consistency control protocol, including those employing dynamic data relocation and replication. We show that if a single copy provides availability A for $0 \leq A \leq 1$, then no scheme can achieve availability greater than $\sqrt{A}$ in the same network. We show this bound to be the best possible for any network with availability greater than .25. Although, as we prove, the problem of calculating A is $\#P$-complete, we describe a method for approximating the optimal location for a single copy which adjusts dynamically to current network characteristics. This bound is most useful for high availabilities, which tend to be obtainable with modern networks and their constituent components.

1. Introduction

A fundamental problem in a number of computer network applications is maximizing the availability of a resource while ensuring that at most one access request may be granted at a time. The question is complicated when there are multiple instances of the single resource scattered about a network. Since the network can partition into more than one connected component as a consequence of link and site failure, separate network components may be unaware of the access requests granted in other components, making mutual exclusion difficult to guarantee. Although numerous protocols have been invented, no tight upper bound on their performance has heretofore been presented in the literature.

In this paper we study networks in which components fail, are subsequently repaired, and are then again subject to failure. These failure and recovery events make possible the formation of connected components which are isolated from each other in what we call a *partition*. Since communication can not take place between components of a partition, each component is unaware of actions taken in other components. Thus, to assure mutually exclusive access to a resource which is present in more than one component, some set of rules, or protocol, governing the access to the resource must be enforced. The protocol must be agreed upon by all network components before system startup.

Determining when to allow updates to objects of a distributed database is one manifestation of this problem. In addition to mutual exclusion, database consistency constraints insist that each update be aware of all previous updates. The obvious method for guaranteeing these two constraints is to allow only one copy of each data item and require that updates are only allowed when this copy is accessible from the requesting site. We call this the *single copy* or *SC* protocol. Much attention has also been given to the development and evaluation of new consistency control protocols, protocols that attempt to maximize availabil-

*e-mail address:djohnson@dartmouth.edu
†e-mail address:raab@dartmouth.edu
‡Department of Mathematics and Computer Science. Hanover, N.H. 03755

ity while guaranteeing that these two constraints are fulfilled.

Many such protocols have been developed and have been found to perform better on the average than naive schemes. The coterie protocol[11] generalizes the voting protocols of [25] and [12], which we discuss in section 2, by allowing accesses in connected components that can not be described using votes. The majority of current protocol[15, 18, 22], an implementable version of the version vector protocol[7], and its enhancements[16, 17] improve upon the consensus protocols by liberalizing the restrictions placed upon accesses. The vote reassignment protocol[4] performs dynamic, autonomous reassignments of votes in an attempt to reduce the system's vulnerability to the effects of further failures. Other protocols have appeared in the literature[8, 9, 24]. We examine a number of these protocols in [14] and [20] and give an upper bound, which we obtain through simulation.

In this paper we develop an analytic upper bound on the behavior of any protocol, and discover that under certain circumstances the performance of the naive single copy protocol is nearly optimal. We also prove our bound is tight and in the process demonstrate that another naive protocol, majority consensus, performs optimally in certain networks.

We use terminology applicable to distributed databases. Thus we refer to the resource for which we guarantee mutual exclusion as a *data item*, and the instances of the resource are called *copies*. An *access request* is a request to update a data item[1]. The *access request distribution* is a probability distribution over the set of all sites. The value of the distribution for a particular site is the expected proportion of access requests that will be submitted at that site. Despite the use of database terminology, it should be remembered that these results apply to any resource for which replication is possible and mutual exclusion must be assured.

We define *availability* as the probability that an access request submitted to an arbitrary site will be allowed to succeed. We choose this definition, which we call *accessibility*, over another definition (frequently used in the literature) which we call *survivability*, which is the probability that at an arbitrary time there exists at least one site which may access the data object. We favor accessibility over survivability since it is our view that accessibility reports more nearly the availability as experienced by a user of the system, who typically cannot readily move from site to site or have knowledge a priori of which sites are functioning. In addition, the accessibility metric will always increase with an increase in the number of sites which can access the data item, whereas survivability may remain unchanged. For a further discussion of these two metrics, we direct the reader to [20].

The bound we give on the performance of protocols which guarantee mutual exclusion is the square root of the performance of a single copy. It is useful to express this bound in terms of the maximum improvement possible over the performance of a single copy. For example, our bound implies that if a database consisting of only one "well-placed"[2] copy of each data item is currently providing availability of .90, then no dynamic or replication scheme can improve the performance by more than $\frac{\sqrt{.90}-.90}{.90} = 5.4\%$ in the same network. If 5.4% does not justify the costs incurred in order to realize this gain, or if a protocol yielding $\sqrt{.90} = .949$ availability is found, then any search for further improvement is in vain. Conversely, if a replication protocol produces availability of .98 but proves expensive in terms of storage and

[1] We consider only update requests since read requests do not require mutual exclusion to guarantee consistency. Clearly, the potential benefit of replication increases as the number of read accesses increases. In the extreme case of all reads and no writes, full replication is undoubtably the best approach since no consistency control protocol is necessary. However, it is the updates which make this problem interesting, and therefore it is the success rate of update requests on which we concentrate. This assumption, that each access performs an update, is equivalent to allowing both read and write accesses and maximizing write availability[12]. This approach has been shown to produce optimal availability even in the presence of reads for a wide range of networks and read-write ratios[1, 19].

[2] We formalize the notion of "well-placed" in Lemma 2 of section 3.1. Although this placement problem can be solved quite simply in many networks, which are built incrementally around a single database, we devote section 4 to a discussion of efficient methods of computing or approximating the optimal copy placement in general networks.

communication costs, then we can substitute the simpler single copy protocol and sacrifice at most $\frac{.98-.98^2}{.98} = 2.0\%$. These examples illustrate the practical value of this bound. The main point is that the percentage possible improvement, $\frac{1}{\sqrt{A}} - 1$, decreases as A increases. This is precisely the trend which we are experiencing as the reliability of real-world components continues to increase from their current reliabilities near 95%[4, 5, 21]. Therefore, our bound will allow protocols to be compared, not only to each other, but also to a general bound that appears quite useful at high reliabilities.

In the next section we define the consistency control protocols which are necessary for the development of the bound, which is then proved in section 3 and generalized to include networks with changing access request distributions in section 3.2. In section 3.3, we show this bound to be tight. In section 4 we discuss efficient methods of determining the best location of a single data item. We conclude with a discussion of the practical consequences of this bound.

2. Protocols

By protocol we mean an algorithm for deciding at any point in time which sites are allowed to access the data object. A protocol can guarantee mutual exclusion within a single set of connected sites, or *component*, simply by locking the data item while it is undergoing an update. Ensuring mutual exclusion when a copy exists in two or more distinct network components is more difficult since it requires that only one component be allowed access at a time. We call this component the *distinguished* component. Thus for a protocol to ensure mutual exclusion, it must ensure that there exists at most one distinguished component at any point in time. As mentioned in the introduction, a database consistency protocol must also guarantee that successive distinguished components have at least one site in common thereby ensuring that each update is aware of all previous updates, but it is not necessary to invoke this additional constraint in order to prove the bound which follows. The mutual exclusion condition is sufficient.

A common protocol is one that selects, before system start-up, a special site called the *primary* site. A component is considered distinguished if and only if it contains the primary site as a member. This approach has been generalized to allow for numerous copies placed at different sites while requiring that one of them functions as the primary site[2]. Since both protocols provide the same availability, we will refer to both of them as the *single copy* protocol. This is the scheme against which all other protocols are judged.

We introduce a simple generalization of the single copy protocol which we call the *relocatable single copy* protocol, or *RSC*. As the name suggests, this protocol is identical to the single copy protocol except that the copy is allowed to be moved. If the access request distribution for some duration of time is known, then the data item can be placed at the most advantageous location. When this distribution changes, we can move the data item in an effort to maximize availability. In order to employ *RSC* as a consistency control protocol, it is necessary to ensure that the data item is not moved between components of a partition. That is, the data item can not be moved from one site to another until both sites are members of the same component. We discuss the consequences of this minor restriction in section 3.2.

The simplest protocol which guarantees mutual exclusion in the presence of multiple copies of a data item is the majority consensus protocol[25] which we designate by *MC*. In this protocol each copy is assigned a number of votes, and a component is distinguished if and only if the total votes of all the copies in the component sum to a majority of the votes in the network. Although protocols yielding higher availability exist[20], *MC* is simple and proves to be sufficient for proving the tightness of our bound.

The last protocol which we describe is neither implementable, since it requires complete knowledge of the system state, nor sufficient to guarantee consistency, since it does not guarantee that successive distinguished components overlap. The purpose of this protocol, which we call *best component* protocol or simply *BC*, is to provide an upper bound on the performance of any implementable and sufficient protocol. Like *RSC*, *BC* can be

- $N = (S, L)$, a network of sites S and links L with $n = |S|$.
- $E[X]$, the expected value of the random variable X.
- $Pr[S]$, the probability that the logical statement S is true.
- $\overline{S}$, the negation of the logical statement S; therefore, $Pr[\overline{S}] = 1 - Pr[S]$.
- A_P, the availability expected when employing the protocol P; and $A_{SC}(x)$, A_{SC} with the single copy placed at site x.
- T, the set of all network states.
- $C_{t,f} = \{c_1, c_2, ..., c_m\}$, the set of m sites in the connected component of state $t \in T$ for which the expected proportion of access requests is highest among all connected components in state t, given the fixed access request distribution f. We call this component the "best" component.
- $|C|$, the proportion of all access requests submitted to some site in the component C.
- B, a random variable which depends on both the network state and access request distribution and is the probability that an access request is submitted to some site in the best component.
- $BEST(s_1, s_2, \ldots, s_m)$, the statement "every site $s_1, s_2, \ldots, s_m$ is in the best component".
- $SAME(s_1, s_2, \ldots, s_m)$, the statement "every site $s_1, s_2, \ldots, s_m$ is in the same component".
- $STATE(t)$, the statement "the network is in state t".

Figure 1: Notation

called a protocol in the context of this paper since we only require that a protocol guarantee mutual exclusion. As mentioned earlier, database applications impose an additional constraint which BC does not guarantee. This fact, however, does not invalidate the use of BC in computing an upper bound. The rule used by BC for designating a distinguished component is very simple: the distinguished component is the component in which we expect the greatest proportion of access requests.

3. Bound

In this section we prove a bound on availability in terms of the performance of the single copy protocol. We first show that the best component protocol (BC) performs at least as well as any protocol which guarantees mutual exclusion, and analyze the performance of BC. In Lemma 1 we give a second expression for this performance, and in Lemma 2 we prove that there exists some site in the network with probability of membership in the "best component" no less than the availability of the system under BC. Using these lemmas, we prove that no protocol can achieve availability greater than the squareroot of that provided by a "well-placed" single copy. In section 3.2 we generalize the bound to systems in which the access request distribution changes over time, and in section 3.3 we show that this bound is the tightest possible.

Figure 1 shows the notation we use.

3.1. Fixed Access Distribution

In this section we state and prove four observations concerning the best component protocol, BC, and the expected value of the proportion of access requests submitted to the best component, $E[B]$. We then use these facts to prove, for a network with a fixed access request distribution, that the single copy protocol with the copy properly placed will always perform within a square of optimal.

Fact 1: *No protocol that guarantees mutual exclusion can provide availability greater than that*

provided by BC. That is, for any such protocol P, $A_P \leq A_{BC}$.

Proof:

For every state t and access request distribution f, A_{BC} allows accesses only in the component $C_{t,f}$ of t which yields the greatest expected availability. Since P insures mutual exclusion, it can do no better in any one state and therefore can do no better overall. □

Fact 2: *The availability provided by the BC protocol is equal to the probability that a transaction is submitted in the best component. That is, $A_{BC} = E[B]$, for fixed access distribution.*

Proof:

Let f be the access request distribution.

We have defined availability as the probability that an access request submitted to an arbitrary site will be allowed to succeed. Since BC grants requests submitted while the network is in state t if and only if they are submitted to a site in component $C_{t,f}$, $A_{BC} = \sum_{t \in T} |C_{t,f}| \; Pr[STATE(t)]$. This is precisely the definition of expectation applied to the random variable B. □

Lemma 1: If a network $N = (S, L)$ has a fixed submit distribution f, then

$$E[B] = \sum_{k \in S} f(k) \; Pr[BEST(k)]$$

Proof:

Let $u(k,t)$ be a function with value 1 if site k is in $C_{t,f}$ and zero otherwise.

Let $v(k,t)$ be a function with value $f(k)$ if site k is in $C_{t,f}$ and zero otherwise.

$$\begin{aligned}
E[B] &= \sum_{t \in T} |C_{t,f}| \; Pr[STATE(t)] \\
&= \sum_{t \in T} \Big(Pr[STATE(t)] \sum_{k \in C_{t,f}} f(k) \Big) \\
&= \sum_{t \in T} \sum_{k \in S} Pr[STATE(t)] \;\; v(k,t) \\
&= \sum_{k \in S} \Big(f(k) \sum_{t \in T} Pr[STATE(t)] \;\; u(k,t) \Big) \\
&= \sum_{k \in S} f(k) \; Pr[BEST(k)]
\end{aligned}$$

□

Lemma 2: *In any network with a fixed access request distribution, there exists some site x for which the probability that x is in the best component is greater than or equal to the probability that a transaction is submitted to a site in the best component. That is, $Pr[BEST(x)] \geq E[B]$.*

Proof:

Let x be the site which is most likely to be in the best component. That is $\forall k \in S$, $Pr[BEST(x)] \geq Pr[BEST(k)]$.

Let f be the access request distribution.

$$\begin{aligned}
Pr[BEST(x)] &= Pr[BEST(x)] \sum_{k \in S} f(k) \\
&\geq \sum_{k \in S} f(k) \; Pr[BEST(k)] \\
&= E[B] \qquad \text{by Lemma 1}
\end{aligned}$$

□

Theorem 1: *If a network with a fixed access request distribution achieves availability A_P using a protocol P which guarantees mutual exclusion, then there exists some site x in the network for which $A_{SC}(x) \geq (A_P)^2$.*

Proof:

We choose x, the location of the single copy, to be the site most likely to be in the best connected component of the network.

Let f be the access request distribution.

The following series of relations proves that $A_{SC} \geq (A_{BC})^2$. This is sufficient to prove the theorem, since we know from Fact 1 that $(A_{BC})^2 \geq (A_P)^2$. We conclude the proof with a justification for the numbered steps.

$$\begin{aligned}
A_{SC}(x) &= \sum_{k \in S} f(k) \; Pr[SAME(x,k)] && (1) \\
&\geq \sum_{k \in S} f(k) \; Pr[BEST(x,k)] \\
&= \sum_{k \in S} f(k) \; Pr[BEST(x)] \; Pr[BEST(k)] && (2) \\
&\geq E[B] \sum_{k \in S} f(k) \; Pr[BEST(k)] && (3) \\
&= \Big(A_{BC} \Big)^2 && (4)
\end{aligned}$$

(1) The single copy protocol guarantees mutual exclusion by allowing an access if and only if

both the site containing the data item and the site to which the request is submitted are in the same component.

(2) The site k to which the transaction is submitted is independent of x, the location of the copy.

(3) This substitution is justified by Lemma 2.

(4) This follows from Lemma 1 and Fact 2.

Therefore for any protocol P guaranteeing mutual exclusion, $A_{SC} \geq (A_{BC})^2 \geq (A_P)^2$ by Fact 1. □

3.2. Multiple Access Distributions

Thus far we have assumed a fixed access distribution. What happens if we allow the access distribution to change over time? Clearly, the advantage of a replication protocol over the single copy protocol will be substantial if we do not allow the single copy protocol to relocate the data item. This observation motives the generalization of *SC* to *RSC*.

Although *RSC* guarantees mutual exclusion, it does not, as consistency control requires, guarantee that successive distinguished components have a copy in common. If we were to add such a restriction to *RSC*, it would not be possible to move the data item from site x to site y until both sites were members of the same component. The practical effect of such a delay is inconsequential in real systems since failures and partitioning are infrequent and of short duration. In addition, the shift from one access distribution to another is not likely to be drastic, implying that neither the distance between x and y nor the delay in moving the data item from x to y is great.

Lemma 3: *If a_k and b_k are positive real numbers for $1 \leq k \leq r$, and $\sum_{k=1}^{r} a_k = 1$ then*

$$\sum_{k=1}^{r} a_k b_k^2 \geq \Big(\sum_{k=1}^{r} a_k b_k\Big)^2$$

Proof:

The proof of this well-know result, usually stated in terms of moments and standard deviations, is omitted. See [10]. □

Theorem 2: *If a network with multiple access request distributions achieves availability A_P using a protocol P which guarantees mutual exclusion, then there exists some sequence of sites at which a single copy can be placed such that $A_{RSC} \geq (A_P)^2$.*

Proof:

Let $\{f_k | 1 \leq k \leq r\}$ be a set of r access request distributions.

Let d_k be the duration of access request distribution f_k.

Let $D = \sum_{k=1}^{r} d_k$ be the total time under consideration.

We choose x_k, the location of the single copy, to be the site most likely to be in the best connected component of the network during the time at which the access request distribution is d_k.

Let $A_{SC_k}(x_k)$ be the expected availability using the single copy protocol with the copy at site x_k during the time in which the access request distribution f_k is in effect.

Let A_{BC_k} be the expected availability using the best component protocol during the time in which the access request distribution f_k is in effect.

$$\begin{aligned} A_{RSC} &= \frac{1}{D}\sum_{k=1}^{r} d_k A_{SC_i}(x_k) \\ &\geq \frac{1}{D}\sum_{k=1}^{r} d_k \Big(A_{BC_i}\Big)^2 && \text{by Theorem 1} \\ &\geq \Big(\sum_{k=1}^{r} \frac{d_k}{D} A_{BC_i}\Big)^2 && \text{by Lemma 3} \\ &= (A_{BC})^2 \end{aligned}$$

Therefore for any protocol P guaranteeing mutual exclusion, $A_{RSC} \geq (A_{BC})^2 \geq (A_P)^2$ by Fact 1. □

3.3. Tightness of the Bound

The natural question to address after exhibiting a bound is whether the bound can be improved. In this section we prove that the bound of Theorem 1 can not be improved, or tightened, for any network with single copy availability exceeding .25. We show this by constructing, for any $A > .25$, a network with availability A using the single copy protocol and availability $\sqrt{A}$ using the majority consensus protocol.

Theorem 3: *For any* $.25 < A_{SC} \leq 1$ *and* $\epsilon > 0$, *there exists a network and a protocol* P *for which* $|A_{SC} - (A_P)^2| < \epsilon$.

Proof:

The network that we construct is the fully-connected network of n sites and $\binom{n}{2}$ links. The links never fail. The sites are operational with a uniform probability p. Each site is given a unique number from 1 to n and is referred to by this number. The network has a fixed, uniform access request distribution f, that is $f(k) = \frac{1}{n}$ for all sites k.

First, we show that the single copy availability A_{SC} of this network approaches p^2 as n approaches infinity. This is intuitively clear since this protocol requires only that the primary site and the site to which the transaction is submitted be operational and that they are in the same component.

$$\begin{aligned}
A_{SC} &= \sum_{k=1}^{n}\Big(f(k) * Pr[SAME(x,k)]\Big) \\
&= \frac{1}{n}\sum_{k=1}^{n} Pr[SAME(x,k)] \\
&= \frac{1}{n}\sum_{k=1}^{n}\Big(Pr[x = k]\, Pr[x \text{ up}] + \\
&\qquad Pr[x \neq k]\, Pr[x \text{ up and } k \text{ up}]\Big) \\
&= \frac{1}{n}\sum_{k=1}^{n}\left(\frac{1}{n}p + \frac{n-1}{n}p^2\right) \\
&= \frac{1}{n}p + \frac{n-1}{n}p^2
\end{aligned}$$

Therefore $\lim_{n\to\infty} A_{SC} = p^2$, as required.

Now we show that the performance of the majority consensus protocol as defined in section 2 approaches p as n approaches infinity. We employ the *weak law of large numbers* which states that as the number of trials approaches infinity, the proportion of outcomes near the mean approaches one. Here "near the mean" means arbitrarily close to the mean. When applied to the binomial distribution $B[n,p,k]$ with n trials (sites) each with probability p of success (operational), this means that the probability that the fraction of successes (operational sites) is within any small fraction of the mean, $\lfloor (n+1)p \rfloor$, approaches 1 as n approaches infinity. For $p > .5$, this implies that the probability that at least half of the sites will be operational approaches 1 as n approaches infinity. Thus when $p > .5$ and the links are fully reliable, the access request can be granted whenever the site to which the access request is submitted is operational, which is true with probability p.

$$\begin{aligned}
&A_{MC} = \\
&\quad \sum_{k=1}^{n}\Big(Pr[k \text{ is up}] * f(k) * \\
&\qquad Pr[k \text{ can comm. with at least } \lfloor \tfrac{n}{2} \rfloor \text{ sites}]\Big) \\
&= \frac{1}{n}p\sum_{k=1}^{n} Pr[k \text{ can comm. with at least } \lfloor \tfrac{n}{2} \rfloor \text{ sites}] \\
&= \frac{1}{n}p\sum_{k=1}^{n}\sum_{j=\lfloor \frac{n}{2} \rfloor}^{n-1} Pr[k \text{ can comm. with exactly } j \text{ sites}] \\
&= \frac{1}{n}p\sum_{k=1}^{n}\sum_{j=\lfloor \frac{n}{2} \rfloor}^{n-1} B[n-1,p,j] \\
&= p\sum_{j=\lfloor \frac{n}{2} \rfloor}^{n-1} B[n-1,p,j]
\end{aligned}$$

Thus $\lim_{n\to\infty} A_{MC} = p$, since, by the weak law of large numbers, $\sum_{j=\lfloor \frac{n}{2} \rfloor}^{n-1} B[n-1,p,j]$ approaches 1 as n approaches infinity.

Therefore we have shown that, for this fully-connected network with fully-reliable links and availability $A_{SC} = p^2 \geq .5^2 = .25$, we can get arbitrarily close to $A_{SC} = (A_{MC})^2$. □

Theorem 3 proves that there exists no tighter bound than that of Theorem 1 for networks with single copy availability greater than .25.

4. Single Copy Placement

In Lemma 2 of section 3.1 we introduced the notion of a "well-placed" copy. In general we will refer to the location of this copy, which is the site most likely to be in the best component, as site x. Thus, in order to be assured that the performance of a single copy protocol is within a square of optimal, it would seem that we must be able to find the location x. But not only is finding x off-line $\#P$-complete (as can be shown in the same manner as is shown below for a better copy location),

it is not even possible on-line in a distributed system. Finding x on-line would require that a site determine the likelyhood of request submissions in other components, which in turn would require communication between components, a clear impossibility. Fortunately, we can find a site yielding single copy performance at least as good as x.

If we place the copy at site x, we know that the performance is within a square of optimal, but we may not have maximized A_{SC}, which is our ultimate goal. From the definition of availability we know that $A_{SC}(y) = \sum_{k \in S} f(k)\, Pr[SAME(y,k)]$. We will say that site y *witnesses* an access request if the request is presented at some site in the component containing y. Thus A_{SC} is maximized when the copy is located at that site which is expected to witness the greatest proportion of access requests.

In [26], Valiant proved that calculating $Pr[SAME(x,y)]$, the probability that sites x and y are in the same component, is $\#P$-complete. We use this fact in the following theorem to prove that calculating the expected proportion of witnesses for a given site x is also $\#P$-complete.

Theorem 4: *Determining the expected proportion of witnesses for a given site x and access request distribution f, $E[WIT(x,f)]$, is $\#P$-complete. The problem remains $\#P$-complete when f is the uniform distribution, that is, when $f(z) = \frac{1}{n}$ for all sites z.*

Proof:

As in the previous section, let $N = (S, L)$ be a network of sites S and links L with $n = |S|$. By the definition of expectations, $E[WIT(x,f)] = \sum_{t \in T} |C_x|\, Pr[STATE(t)]$, where C_x is the component containing the site x. Using arguments similar to those in the proof of Lemma 1, it can be shown that $E[WIT(x,f)] = \sum_{z \in S} f(z)\, Pr[SAME(x,z)]$.

Given a method for determining $E[WIT(z,f)]$ for an arbitrary site z and access request distribution f, we can determine $Pr[SAME(x,y)]$ for any pair (x,y) by finding $E[WIT(x,g)]$ with $g(y) = 1$ and $g(z) = 0$ for all $z \neq y$. Since $E[WIT(x,g)] = Pr[SAME(x,y)]$, and since determining $Pr[SAME(x,y)]$ is $\#P$-complete, determining $E[WIT(x,g)]$ must also be $\#P$-complete.

To show that this problem remains $\#P$-complete when f is the uniform access request distributions, we find $Pr[SAME(x,y)]$ by calculating the expected proportion of witnesses for each of the two networks described below.

Let $E_N[X]$ be the expected value of the random variable X in the network N.

Let $N' = (S', L')$ where $S' = S \cup \{y'\}$, $L' = L \cup \{(y,y')\}$, the reliability of the site y' is p, and the reliability of the edge (y,y') is r. Also let f' be the uniform access request distribution for the network N'. Therefore $f'(z) = \frac{1}{n+1}$ for all $z \in S'$.

$$
\begin{aligned}
E_{N'}[WIT(x,f')] &= \sum_{t \in T} |C_x|\, Pr[STATE(t)] \\
&= \sum_{z \in S'} f'(z)\, Pr[SAME(x,z)] \\
&= \frac{1}{n+1}\Big(Pr[SAME(x,y')] + \sum_{z \in S} Pr[SAME(x,z)]\Big) \\
&= \frac{1}{n+1}\Big(pr\, Pr[SAME(x,y)] + \sum_{z \in S} Pr[SAME(x,z)]\Big) \\
&= \frac{pr}{n+1}\, Pr[SAME(x,y)] + \frac{n}{n+1}\, E_N[WIT(x,f)]
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
Pr[SAME(x,y)] = {} & \frac{n+1}{pr}\, E_{N'}[WIT(x,f')] - \\
& \frac{n}{pr}\, E_N[WIT(x,f)]
\end{aligned}
$$

Thus, for the uniform access request distribution, calculating $E_N[WIT(x,f)]$ must be $\#P$-complete since calculating $Pr[SAME(x,y)]$ is $\#P$-complete. □

Although $\#P$-complete in general, the placement problem, as we refer to the determination of the optimal location for the data item, is solvable for some systems. Since often a network for an existing database is built incrementally around the database, the current location may be optimal. In addition, the single copy availability can be efficiently determined for regular network topologies[3, 13], such as ring, single-bus, fully-connected, and for series-parallel networks[6, 23]. Since, for these topologies, the single copy availability can be calculated in polynomial time, the placement problem can be solved in polynomial time simply by calculating $A_{SC}(k)$ for all sites k.

Although calculating the expected proportion of witnesses is feasible in some special cases, it is unnecessary and perhaps undesirable to do so in real systems. Instead, each site can record the actual number of access requests witnessed, and the site with the largest number can be made the location of the copy. If the past network performance and access distribution is, as one would expect, indicative of future behavior, then this technique leads to optimal copy placement. This method does not require a priori knowledge of the network topology, hardware reliability, or access distribution, and adjusts automatically to unanticipated changes in any of the these system parameters. These characteristics are precisely those necessary for an automated version of the *RSC* protocol. Our experience with simulation[20] indicates that this approach will be successful.

5. Conclusions

We have proven an upper bound on the increase in availability possible using replication and dynamic techniques including relocation of copies. We have shown that if a network achieves single copy availability A, $0 \leq A \leq 1$, for a well-located copy, then no technique which guarantees mutual exclusion can achieve availability greater than $\sqrt{A}$. We have also shown that no tighter bound in terms exclusively of single copy availability exists for $.25 < A \leq 1$. In order to make this bound useful, we have also addressed the optimal location problem.

References.

[1] Mustaque Ahamad and Mostafa H. Ammar. Performance Characterization of Quorum Consensus Algorithms for Replicated Data. In *Proceedings of the 6th Symposium on Reliability in Distributed Software and Database Systems*, pages 161–168. IEEE, 1987.

[2] P. A. Alsberg and J. D. Day. A Principle for Resilient Sharing of Distributed Resources. In *Proceedings of the 2nd Annual Conference on Software Engineering*, pages 627–644, October 1976.

[3] Daniel Barbara and Hector Garcia-Molina. The Reliability of Voting Mechanisms. *IEEE Transactions on Computers*, C-36(10):1197–1208, 1987.

[4] Daniel Barbara, Hector Garcia-Molina, and Annemarie Spauster. Increasing Availability Under Mutual Exclusion Constraints with Dynamic Vote Reassignment. *ACM Transactions on Computer Systems*, 7(4):394–426, 1989.

[5] John Carroll and Darrell D. E. Long. The Effect of Failure and Repair Distributions on Consistency Protocols for Replicated Data Objects. In *Proceedings of the 22nd Annual Simulation Symposium*, pages 47–60. IEEE, 1989.

[6] Charles J. Colbourn. *The Combinatorics of Network Reliability*. Oxford University Press, 1987.

[7] Dančo Davčev and Walter A. Burkhard. Consistency and Recovery Control for Replicated Files. In *Proceedings of the 10th ACM Symposium on Operating Systems Principles*, pages 87–96, December 1985.

[8] Derek L. Eager and Kenneth C. Sevcik. Achieving Robustness in distributed database systems. *ACM Transactions on Database Systems*, 8(3):354–381, September 1983.

[9] Amr El Abbadi and Sam Toueg. Availability in Partitioned Replicated Databases. In *Proceedings of the 5th ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*, pages 240–251. ACM, March 1986.

[10] William Feller. *An Introduction to Probability Theory and Its Applications*. John Wiley & Sons, 1968.

[11] Hector Garcia-Molina and Daniel Barbara. How to Assign Votes in a Distributed System. *Journal of the ACM*, 32(4):841–860, October 1985.

[12] D. K. Gifford. Weighted Voting for Replicated Data. In *Proceedings 7th ACM SIGOPS Symposium on Operating Systems Principles*, pages 150–159, Pacific Grove, CA, December 1979.

[13] E. N. Gilbert. Random Graphs. *Annals of Mathematical Statistics*, 30:1141–1144, 1959.

[14] Michael Goldweber, Donald B. Johnson, and Larry Raab. A Comparison of Consistency Control Protocols. Technical Report PCS-TR89-141, Dartmouth College, July 1989.

[15] Sushil Jajodia and David Mutchler. Dynamic Voting. In *Proceedings of the SIGMOD Annual Conference*, pages 227–237. ACM, May 1987.

[16] Sushil Jajodia and David Mutchler. Enhancements to the Voting Algorithm. In *Proceedings of the 13th International Conference on Very Large Data Bases*, pages 399–406, September 1987.

[17] Sushil Jajodia and David Mutchler. Integrating Static and Dynamic Voting Protocols to Enhance File Availability. In *Proceedings of the 4th International Conference on Data Engineering*, pages 144–153. IEEE, February 1988.

[18] Sushil Jajodia and David Mutchler. Dynamic Voting Algorithms for Maintaining the Consistency of a Replicated Database. *ACM Transactions on Database Systems*, 15(2):230–280, June 1990.

[19] Donald B. Johnson and Larry Raab. Finding Optimal Quorum Assignments. Technical Report PCS-TR90-158, Dartmouth College, November 1990.

[20] Donald B. Johnson and Larry Raab. Effects of Replication on Data Availability. *International Journal of Computer Simulation*, 1991. to appear.

[21] Darrell D. E. Long, Jehan-François Pâris, and C. J. Park. A Study of the Reliability of Internet Sites. Technical Report UCSC-CRL-90-46, University of California, Santa Cruz, September 1990.

[22] Jehan-François Pâris and Darrell D. E. Long. Efficient Dynamic Voting Algorithms. In *Proceedings of the 4th International Conference on Data Engineering*, pages 268–275. IEEE, February 1988.

[23] A. Satyanarayana and R. K. Wood. A Linear Time Algorithm for Computing k-Terminal Reliability in Series-parallel Networks. *SIAM Journal of Computing*, 14:818–832, 1985.

[24] D. Skeen and D. Wright. Increasing Availability in Partitioned Networks. In *Proceedings of the 3rd ACM SIGMOD Symposium on Principles of Database Systems*, pages 290–299, April 1984.

[25] R. Thomas. A Majority Consensus Approach to Concurrency Control. *ACM Transactions on Database Systems*, 4(2):180–209, June 1979.

[26] L. G. Valiant. The Complexity of Enumeration and Reliability Problems. *SIAM Journal on Computing*, 8(3):410–421, August 1979.