EVALUATING THE IMPACT OF NETWORK PARTITIONS ON REPLICATED DATA AVAILABILITY

Jehan-François Pâris

Department of Computer Science
University of Houston
Houston, TX 77204-3475.

ABSTRACT

Many distributed systems maintain multiple replicas of their critical data to protect these data against equipment failures. When this is the case, a *replication control protocol* must be chosen to insure that a consistent view of the data is always presented.

In this paper, we present a simple aggregation technique leading to closed form estimates of the availability of replicated objects whose replicas reside on networks subject to communication failures. We illustrate our technique by comparing the availabilities of replicated objects with three replicas managed by majority consensus voting (MCV), and dynamic-linear voting (DLV), under three different network configurations.

Keywords: fault-tolerance, replicated systems, redundancy, voting.

1. Introduction

Many applications depend on critical data that must remain available in the presence of equipment malfunctions. Recent advances in networking technology have made the replication of these data on several sites of a local area network a cost-effective proposition. First, having multiple replicas of the same data virtually eliminates the risk of permanent data loss. Second, distributing the replicas among distinct sites of a network increases the probability that the data will remain available in the presence of hardware faults. Managing replicated data presents however a special challenge as site failures and network malfunctions are likely to result in inconsistent replica updates. Special *replication control protocols* have been devised to avoid this occurrence and insure that a consistent view of the replicated data is always presented.

Various replication control protocols have been presented in the literature. These protocols vary greatly in their complexity, their communication overhead, the protection they provide or do not provide against communication failures, and the number of replicas they require to guarantee full access to the replicated data in the presence of a given number of site failures. As a result, the evaluation of the performance of replication control protocols has become an area of great practical interest. An important measure of this performance is the *availability* of the replicated data object managed by the protocol. By definition the availability of a replicated data object represents the steady-state probability that the object is available at any given moment.

Several techniques have been used to evaluate the availability of replicated data. Combinatorial models are very simple to use [1, 2] but cannot represent complex recovery modes as these found in available copies and dynamic voting protocols. Simulation models can be very accurate if all the parameters of the modeled system are known. They have two major disadvantages; the first is that they are computationally intensive and the second is that they provide only numerical results. As a result, stochastic models have become the method of choice for evaluating the availability of replicated data managed by protocols with complex recovery modes [3-6]. These however suffer from two important limitations: First, stochastic models become quickly intractable unless all failure and repair processes have exponential distributions. Second, stochastic processes do not handle well communication failures as the number of distinct states in a model increases exponentially with the number of failure modes being considered. As a result, all recent studies of the availability of replicated data have either relied on simulation models or have totally neglected communication failures. This neglect has resulted in over-optimistic evaluations of the availability of the replicated data objects under study.

We present in this paper a simple aggregation technique leading to closed form estimates of the availability of replicated objects whose replicas reside on networks subject to communication failures. We illustrate our technique by comparing the availabilities of replicated objects with three replicas managed by majority consensus voting (MCV) and dynamic-linear voting (DLV) under three different network configurations. We show that communication failures have a very different impact on the availability of three replicas managed by MCV and DLV with DLV being the least affected.

The remainder of this paper is organized as follows: Section two reviews voting protocols; Section three introduces our aggregation technique; Section four illustrates our method on an example. Our conclusions appear in Section five.

2. Voting Protocols

Voting protocols [7] probably constitute the best known class of replication protocols. Voting protocols ensure the consistency of replicated data objects by disallowing all read and write requests that cannot collect an appropriate quorum of replicas. Different quorums for read and write operations can be defined, and different weights, including none, assigned to every replica [8]. Consistency is guaranteed as long as the write quorum W is high enough to disallow parallel writes on two disjoint subsets of replicas, and the read quorum R is high enough to ensure that read and write quorums always intersect.

These conditions are simple to verify, which accounts for the conceptual simplicity and the robustness of voting schemes. Voting has however some disadvantages. It requires a minimum number of three replicas to be of any practical use. Even then, quorum requirements tend to disallow a relatively high number of access requests.

Several solutions have been proposed to overcome these limitations. *Dynamic voting* (DV) [9] and *dynamic-linear voting* (DLV) [4] adjust quorums to reflect changes in replica availability and network topology. Both protocols greatly improve the availability and reliability of replicated objects with more than three replicas. *Voting with witnesses* [3], *voting with ghosts* (VWG) [2] and *voting with bystanders* [10] share the common thread of introducing auxiliary entities that are used by the protocol to increase the availability of replicated data objects.

3. The State Aggregation Technique

The inability of stochastic models to model replicated data objects with multiple failure modes and complex recovery procedures is probably their important limitation as it severely restricts our ability to evaluate the availability of replicated data in the presence of network partitions. This inability is a direct consequence of the fact that the number of distinct states in the model increases exponentially with the number of failure modes being considered. Dugan and Ciardo have proposed to use Petri nets to generate stochastic models of replicated data object with witnesses managed by the MCV protocol [11].

Another solution consists of reducing the complexity of the model itself by identifying parts of the system that can be studied in isolation and replaced by simpler equivalent components [12, 13]. This technique has been widely used in computer systems performance evaluation to solve Markov models too complex to be directly tractable. We will show how it can be applied to the evaluation of the availability of replicated data objects.

Many local-area networks consist of several carrier-sense segments or token rings linked by selective repeaters or gateway hosts. Figure 1 shows one example of such networks: it contains three CSMA segments *AB, ACDE* and *EF*. *A* is the gateway between *AB* and *ACDE* while *E* is the gateway between *ACDE* and *EF*. Since repeaters and gateways can fail without causing a total network failure, such networks can be partitioned. The key difference with conventional point-to-point networks is that sites that are on the same carrier-sense network or token ring will never be separated by a partition. We will refer to these entities as *LAN segments* [2].

Consider now the replicated object *X* represented on figure 2. It consists of two replicas *A* and *B* located on the same LAN segment and a third replica *C* on a second segment. Let us assume that the two LAN segments are linked by a gateway *G*. Under MCV, replica *C* will only be able to participate in elections when the gateway *G* is operational. For all practical purposes, a failure of *G* will have the same effect as a failure of *C*. We propose therefore to replace the subsystem consisting of site *C* and its gateway *G* by an *aggregate site C′* that will remain operational as long as *both C* and *G* are operational. The replicated object consisting of sites *A*, *B* and the aggregate site *C′* will have the same availability as the replicated object *X* but will be much easier to investigate since we do not have to consider gateway failures.

This aggregation technique can be trivially extended to replicated objects consisting of an arbitrary number of replicas located on a network consisting of LAN segments linked by gateways provided that the following conditions are met:

(a) There is at most one LAN segment that contains more than one replica. (We will refer to that segment as the *backbone segment*.)

(b) If there is a backbone segment, all sites that are not on the backbone segment communicate with the sites on the backbone segment through their own gateways or sequences of gateways.

(c) If there is no backbone segment, there is at least one LAN segment such that all sites that are not on the segment communicate with the sites on the segment through their own gateways or sequences of gateways.

Condition (a) is necessary to ensure that the replicated object can be reduced to an equivalent object with all its aggregate sites on the same LAN segment. Conditions (b) and (c) are necessary to ensure that the aggregate sites do not include common gateways as common gateways would introduce non-independent failures of aggregate sites.

These conditions clearly restrict the number of replicated objects that can be analyzed through our aggregation method. Fortunately they are generally met by replicated objects with two or three replicas and these replicated objects are the most likely to be encountered in practice.

Another limitation of the method is its implicit assumption that sites that become part of an aggregate site can never become a single site majority. While this assumption is correct for all voting protocols that never allow single site majorities, it is not true for weighted voting and dynamic-linear voting protocols. Consider for instance the replicated data object represented in figure 2 and assume that its three replicas are managed by a weighted voting protocol assigning one vote to replica *A*, one vote to replica *B*, and three votes to replica *C*. Since replica *C* holds a majority of the votes, the replicated data object will remain available as long as *C* remains available. Failures of the gateway *G* will affect the accessibility of the object from sites *A* and *B* but not its overall availability. This is not true for the simplified ‘‘equivalent’’ model obtained by merging sites *C* and *G* into a single aggregate state *C′* since any failure of *G* results in a failure of *C′*.

This situation could be dismissed as an oddity since assigning a majority of the votes to a single site negates most of the benefits of replication. Single site majorities are however a feature of dynamic-linear voting protocols. Let us go back to the replicated object represented in figure 2 and assume that it is now managed by a dynamic-linear voting protocol with $C > B > A$. As long as *A*, *B* and *C* are operational, the majority partition will consist of these three sites: *{A, B, C}*. A failure of site *B* would result in the *exclusion* of *B* from the majority partition, which would now become *{A, C}*. Should site *A* fail while *B* is still unavailable, site *C* would become a single site majority partition *{C}*. Here again there would be a discrepancy between the original model and the equivalent model obtained by aggregating *C* and *G* into *C′*. The problem can be avoided by reordering the sites in such a way that *C* becomes the lowest site. This is not possible for a replicated object consisting of three replicas located on three distinct LAN segments as the one represented on figure 3.

4. An Example

In this section we illustrate our aggregation technique by comparing the availabilities of replicated objects with three replicas managed by majority consensus voting (MCV) and dynamic-linear voting (DLV) under three different network configurations. Our model consists of a set of

sites with independent failure modes that are connected via a network composed of LAN segments linked by gateways. When a site fails, a repair process is immediately initiated at that site. Should several sites fail, the repair process will be performed in parallel on those failed sites. We assume that failures are exponentially distributed with mean failure rate λ, and that repairs are exponentially distributed with mean repair rate μ. The system is assumed to exist in statistical equilibrium and to be characterized by a discrete-state Markov process.

The three configurations investigated are: (a) three replicas on the same LAN segment (1LS), (b) three replicas on two LAN segments linked by one gateway (2LS), and (c) three replicas on three LAN segments linked by two gateways (3LS). Configuration (a) is the only configuration that is immune to network partitions as the three replicas are on the same LAN segment. 3. Configuration (b) is represented on figure 2: it has one backbone segment containing replicas *A* and *B* and one LAN segment containing a single replica *C*. Its only aggregate site is site *C′*, which results from the merge of gateway *G* with site *C*. Configuration (c) is represented on figure 3. Since replicas *B* and *C* communicate with replica *A* through distinct gateways, we will have the two aggregate sites *B′* and *C′* respectively consisting of *G* and *B* and *H* and *C*.

Observing that all aggregate sites consist of *one* gateway and *one* site holding a replica, we now derive the failure and repair rates of an aggregate site. Figure 4 (a) contains the state transition diagram for a subsystem consisting of a gateway and a site holding a replica. The diagram has four states. State $\langle 11 \rangle$ represents the state of the subsystem when the site and its gateway are both operational. States $\langle 01 \rangle$ and $\langle 10 \rangle$ represent states when either the site or its gateway have failed while site $\langle 00 \rangle$ corresponds to a failure of both entities.

As seen on figure 4 (b), the state transition diagram for the aggregate site has only two states. State 1 represents the state of the subsystem when the aggregate site is operational and can participate in elections. It corresponds to the state $\langle 11 \rangle$ of the subsystem and has one outbound transition whose rate 2λ is the sum of the rates of the two transitions leaving state $\langle 11 \rangle$. State 0 is a failure state that corresponds to the three other states of the subsystem. Its outbound transition has a rate μ' given by

$$\mu' p_0 = \mu \, (p_{01} + p_{10})$$

where p_0 is the probability of the aggregate site being in state 0 while p_{01} and p_{10} respectively represent the probabilities that the subsystem is in state $\langle 01 \rangle$ or $\langle 10 \rangle$.

Observing that

$$p_0 = 1 - p_1 = 1 - \frac{\mu^2}{(\mu+\lambda)^2}$$

and

$$p_{01} = p_{10} = \frac{\lambda\mu}{(\mu+\lambda)^2},$$

we have

$$\mu' = \frac{\mu}{(1+\lambda/2\mu)}.$$

In the absence of network failures, the availability of a replicated data object with three replicas managed by MCV A_{MCV} is equal to the probability that at least two of the three sites holding replicas are operational. We have therefore

$$A_{MCV} = A_1A_2A_3 + (1-A_1)A_2A_3 + A_1(1-A_2)A_3 + A_1A_2(1-A_3)$$

where A_1, A_2 and A_3 are the respective availabilities of the three replicas. Since the availability of a single replica is given by

$$A = \frac{\mu}{(\lambda+\mu)}$$

and the availability of an aggregate site is given by

$$A' = p_1 = \frac{\mu^2}{(\mu+\lambda)^2},$$

we have

$$A_{MCV}(1LS) = A^3 + 3A^2(1-A) = \frac{\mu^3+3\lambda\mu}{(\mu+\lambda)^3},$$

$$A_{MCV}(2LS) = A^2A' + 2AA'(1-A) + A^2(1-A')$$

$$= \frac{\mu^4+4\lambda\mu^3+\lambda^2\mu^2}{(\mu+\lambda)^4},$$

and

$$A_{MCV}(3LS) = AA'^2 + 2AA'(1-A') + A'^2(1-A)$$

$$= \frac{\mu^5+5\lambda\mu^4+2\lambda^2\mu^3}{(\mu+\lambda)^5}.$$

The graph in figure 5 represents the compared availabilities of the three configurations under study for values of $\rho = \lambda / \mu$ between 0 and 0.2. The first value corresponds to perfectly reliable sites and the latter to sites that are repaired five times faster than they fail and have an individual availability of 0.833. The dotted curve at the bottom represents the availability of an unreplicated data object and was added to provide an element of comparison. As one can see, the availability of replicated objects managed by MCV is strongly affected by the possibility of network partitions. When $\rho > 0.1$, the availability of a replicated object with three replicas on three distinct LAN segments barely exceeds that of an unreplicated object and is even worse for $\rho > 0.2$.

The same approach can be followed for deriving expressions for the availabilities of the three configurations under dynamic linear voting (DLV). The derivations are somewhat more complex because DLV is a more sophisticated protocol.

Figure 6 contains the state transition rate diagram for three identical replicas managed by DLV in the absence of communication failures. Note that left-to-right and top-to-bottom transitions represent site failures while right-to-left and bottom-to-top transitions indicate site repairs. State 3 represents the state of the replicated object when all its three replicas are operational. The majority partition then comprises these three replicas. A failure of one of these three replicas brings the replicated object in state 2. The failure does not affect the availability of the replicated object since a majority of the replicas in the previous majority partition remain operational. The DLV protocol does however update the majority partition which loses the replica that failed and is now comprised of the two replicas that remain operational.

A failure of one of the two replicas that are available when the replicated object is in state 2 would result in a tie because the remaining replica would constitute exactly one half of the current majority partition. The DLV protocol breaks such ties by using the linear ordering of the sites holding the replicas. Two cases need therefore to be considered:

(1) If the site holding the replica that failed is lower ranked than the site holding the replica that remains operational, that last operational replica becomes the new majority partition and the replicated object remains available. This corresponds to a transition from state 2 to state 1 on the diagram.

(2) If the site holding the replica that failed is higher ranked than the site holding the replica that remains operational, the replicated object becomes unavailable and the majority partition is not updated. This corresponds to a transition from state 2 to state 1′ on the diagram.

State 0′ represents the state of the replicated object after its three replicas have failed. Recovering from state 0 would bring the replicated object into state 1 if the site that recovers is the higher ranked of the two sites in the last majority partition or into state 1′ if this is not the case. Finally state 2′ represents the state of the replicated object when one of its two operational replicas does not belong to the current majority partition and the other one resides on the lower ranked of the two sites in the current majority partition. It is therefore an unavailable state.

Let p_i represent the probability that the system is in an available state i and q_j the probability of being in an unavailable state j'. The state transition diagram, along with the normalization condition

$$\sum_{i=1}^{3} p_i + \sum_{j=0}^{2} q_j = 1,$$

yield a system of linear equations that can be solved using standard techniques. Symbolic manipulation software is essential because, although the process is simple, it is tedious and error-prone.

The availability is given by the sum of probabilities of being in one of the three available states:

$$A_{DLV}(1LS) = \sum_{i=1}^{3} p_i = \frac{\rho^3 + 3\rho^2 + 4\rho + 1}{(\rho + 1)^4}$$

with $\rho = \lambda / \mu$.

Figure 7 shows the state transition diagram for DLV when the three replicas are on two LAN segments. States are now identified by pairs of numbers $\langle mp \rangle$ where m is the number of operational replicas on the first LAN segment ($0 \le m \le 2$) and n is the state of the aggregate site formed by the third replica and the gateway to its LAN segment. Hence state $\langle 21 \rangle$ represents the state of the replicated objects when its three replicas and the gateway linking them are all operational. States where the replicated object is unavailable are identified by a prime mark (').

Transitions between states are similar to those observed on the diagram of figure 6 with one major exception: while the three replicas had previously the same failure and repair rates, the aggregate site has now a failure rate $\lambda' = 2\lambda$ and a repair rate

$$\mu' = \frac{\mu}{(1+\lambda / 2\mu)}.$$

As before, failure transition reduce the number of operational sites while recovery transitions

have the opposite effect. Some recovery transitions leave the replicated object in an unavailable state because the replica that failed last is still unavailable. For instance, state $\langle 11' \rangle$ is an unavailable state although two of the three replicas are operational because the replica that failed last has not yet recovered.

The availability of the replicated object is then given by the sum of the probabilities of being in one of the four available states:

$$A_{DLV}(2LS) = p_{21} + p_{20} + p_{11} + p_{10}$$
$$= \frac{\rho^4 + 4\rho^3 + 6\rho^2 + 5\rho + 1}{(\rho + 1)^5}$$

where p_{ij} is the probability of being in state $\langle ij \rangle$.

The case where the three replicas are on three distinct LAN segments was easy to solve: As figure 3 indicates, the replicated object is then represented by one replica on the backbone segment and two aggregate sites. The state transition diagram for DLV with three replicas can therefore be derived from that for three replicas on two LAN segments by replacing all instances of λ and μ by λ' and μ' and vice versa. The availability of the replicated object is then given by:

$$A_{DLV}(3LS) = p_{21} + p_{20} + p_{11} + p_{10}$$
$$= \frac{3\rho^6 + 19\rho^5 + 49\rho^4 + 67\rho^3 + 54\rho^2 + 27\rho + 4}{(\rho + 1)^6(3\rho + 4)}$$

The graph in figure 8 represents the compared availabilities of the three configurations under study for the same values of ρ as in figure 5. The availabilities afforded by DLV and MCV for three replicas in the absence of network partitions were known to be practically equal [14]. We were therefore very surprised to observe than DLV with three replicas on three distinct LAN segments performed almost as well as MCV with the same number of replicas on two segments. This result is even more impressive when one recalls that our aggregation technique tends to underestimate the availability afforded by protocols allowing single site majorities and that DLV belongs to that class of protocols.

Previous studies of the DLV protocol had concluded that it needed at least four replicas to outperform MCV in any significant fashion [14]. We have shown that this conclusion does not hold when communication failures are taken into account as MCV with three replicas tends to behave poorly when the failure-rate-to-repair-rate ratio exceeds 0.1 while DLV continue to provide a good availability.

5. Conclusions

We have presented in this paper a simple aggregation technique leading to closed form estimates of the availability of replicated objects whose replicas reside on networks subject to communication failures. To illustrate our technique, we have compared the availabilities of replicated objects with three replicas managed by majority consensus voting (MCV) and dynamic-linear voting (DLV) under three different network configurations.

Two conclusions can be reached from our study. First, communication failures can severely reduce the availability of replicated data. Second, the effect of communication failures on data availability is not equally distributed among all protocols. Hence some replication control protocols (among which DLV and MCV) may appear equivalent when reliable communication is assumed and behave quite differently when communication failures are considered.

Acknowledgements

We wish to thank Elizabeth Pâris for her editorial comments.

The Markov analysis of the availability of the protocols under study has been done with the aid of MACSYMA, a large symbolic manipulation program developed at the Massachusetts Institute of Technology Laboratory for Computer Science. MACSYMA is a trademark of Symbolics, Inc.

References

[1] C. Pu, J. D. Noe and A. Proudfoot, ‘‘Regeneration of Replicated Objects: A Technique and its Eden Implementation,’’ *IEEE Transactions on Software Engineering*, SE-14, 7 (1988), pp. 936-945.

[2] R. van Renesse and A. Tanenbaum, ‘‘Voting with Ghosts,’’ *Proc. 8th International Conference on Distributed Computing Systems,* (1988), pp. 456-462.

[3] J.-F. Pâris, ‘‘Voting with Witnesses: A Consistency Scheme for Replicated Files,’’ *Proc. 6th International Conference on Distributed Computing Systems,* (1986), pp. 606-612.

[4] S. Jajodia and D. Mutchler, ‘‘Enhancements to the Voting Algorithm,’’ *Proc. 13th VLDB Conference* (1987), pp. 399-405.

[5] M. Ahamad and M. H. Ammar, ‘‘Performance Characterization of Quorum-Consensus Algorithms for Replicated Data,’’ *IEEE Transactions on Software Engineering,* SE-15, 4 (1989), pp. 492-496.

[6] J.-F. Paris and D.D.E. Long ‘‘On the Performance of Available Copy Protocols,’’ *Performance Evaluation, 11 (1990), pp 9-30.*

[7] C. A. Ellis, ‘‘Consistency and Correctness of Duplicate Database Systems,’’ *Operating Systems Review*, 11 (1977).

[8] D. K. Gifford, ‘‘Weighted Voting for Replicated Data,’’ *Proc. 7th ACM Symposium on Operating System Principles* (1979), pp. 150-161.

[9] D. Davcev and W. A. Burkhard, ‘‘Consistency and Recovery Control for Replicated Files,’’ *Proc. 10th ACM Symposium on Operating System Principles* (1985) pp. 87-96.

[10] J.-F. Paris ‘‘Voting with Bystanders,’’ *Proc. 9th International Conference on Distributed Computing Systems,* (1989), pp. 394-401.

[11] J. B. Dugan and G. Ciardo, ‘‘Stochastic Petri Net Analysis of a Replicated File System,’’ *IEEE Transactions on Software Engineering,* SE-15, 4 (1989), pp. 394-401.

[12] P. J. Courtois *Decomposability: Queuing and Computer System Applications,* Academic Press, New York (1977).

[13] D. Ferrari, G. Serazzi, A. Zeigner, *Measurement and Tuning of Computer Systems,* Prentice-Hall, Englewood Cliffs, NJ (1983).

[14] D. D. E. Long and J.-F. Pâris, ‘‘A Realistic Evaluation of Optimistic Dynamic Voting,’’ *Proc. 7th Symposium on Reliable Distributed Systems,* (1988), pp. 129-137.

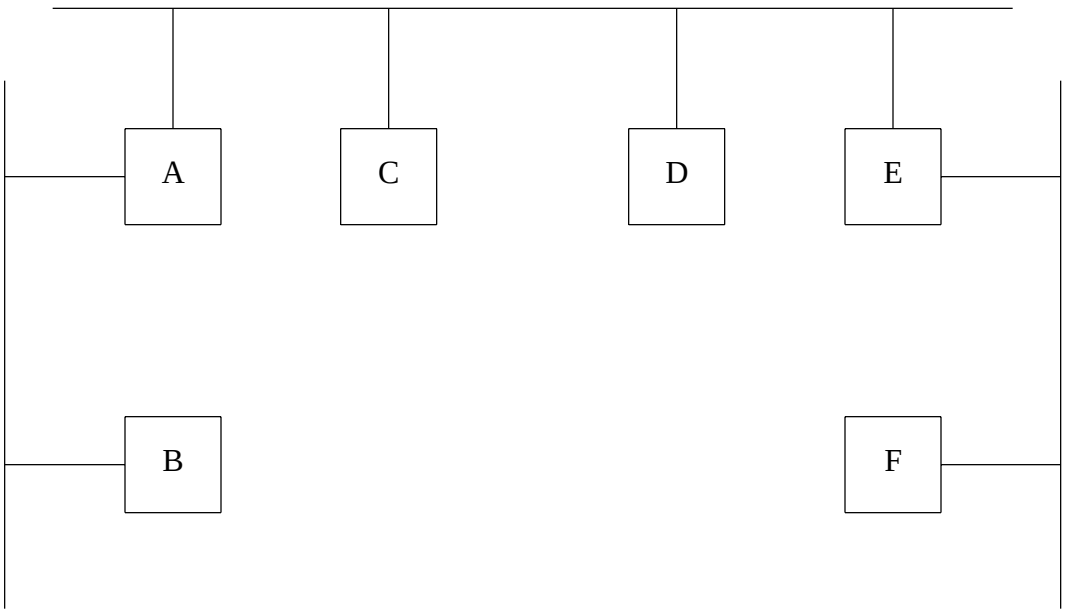


Figure 1: A LAN with Six Sites on Three Segments

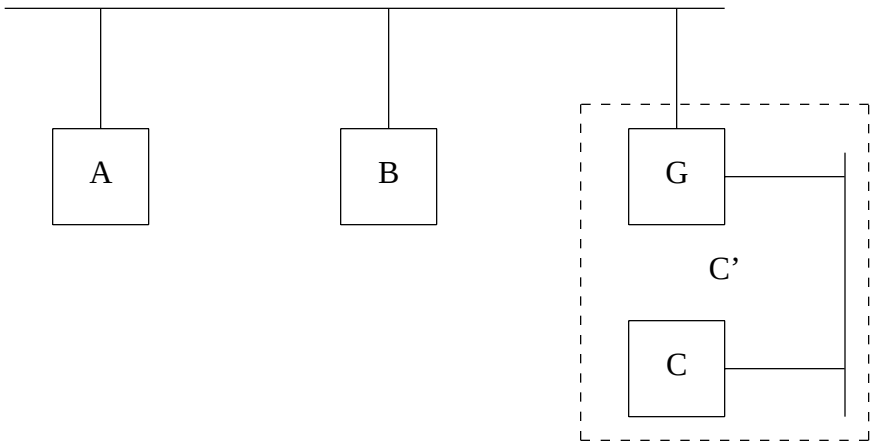


Figure 2: Three Replicas on Two LAN Segments

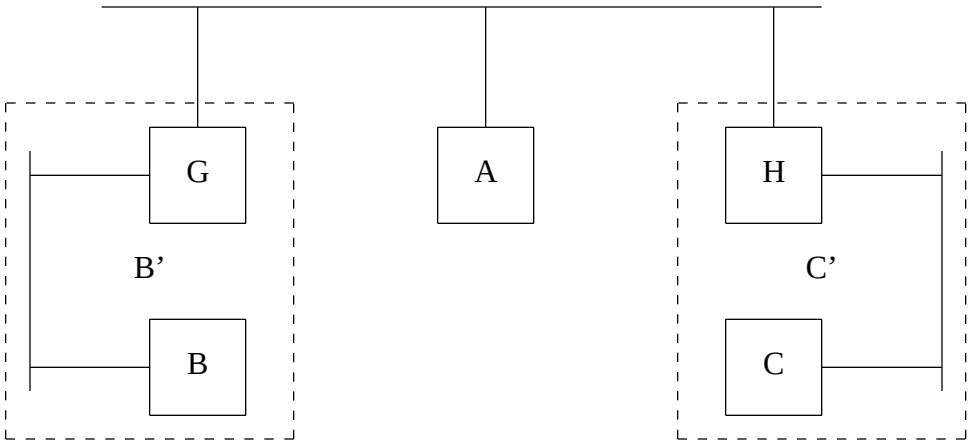
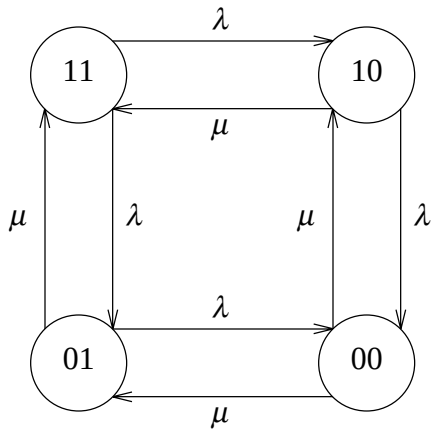
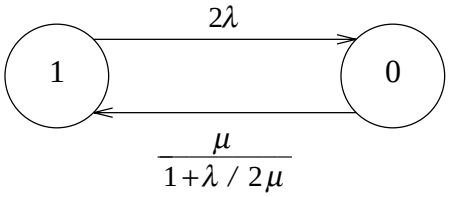


Figure 3: Three Replicas on Three LAN Segments



(a) State Transition Diagram for a Site and its Gateway



(b) State Transition Diagram for the Equivalent Aggregate Site

Figure 4: Aggregating a Site with its Gateway

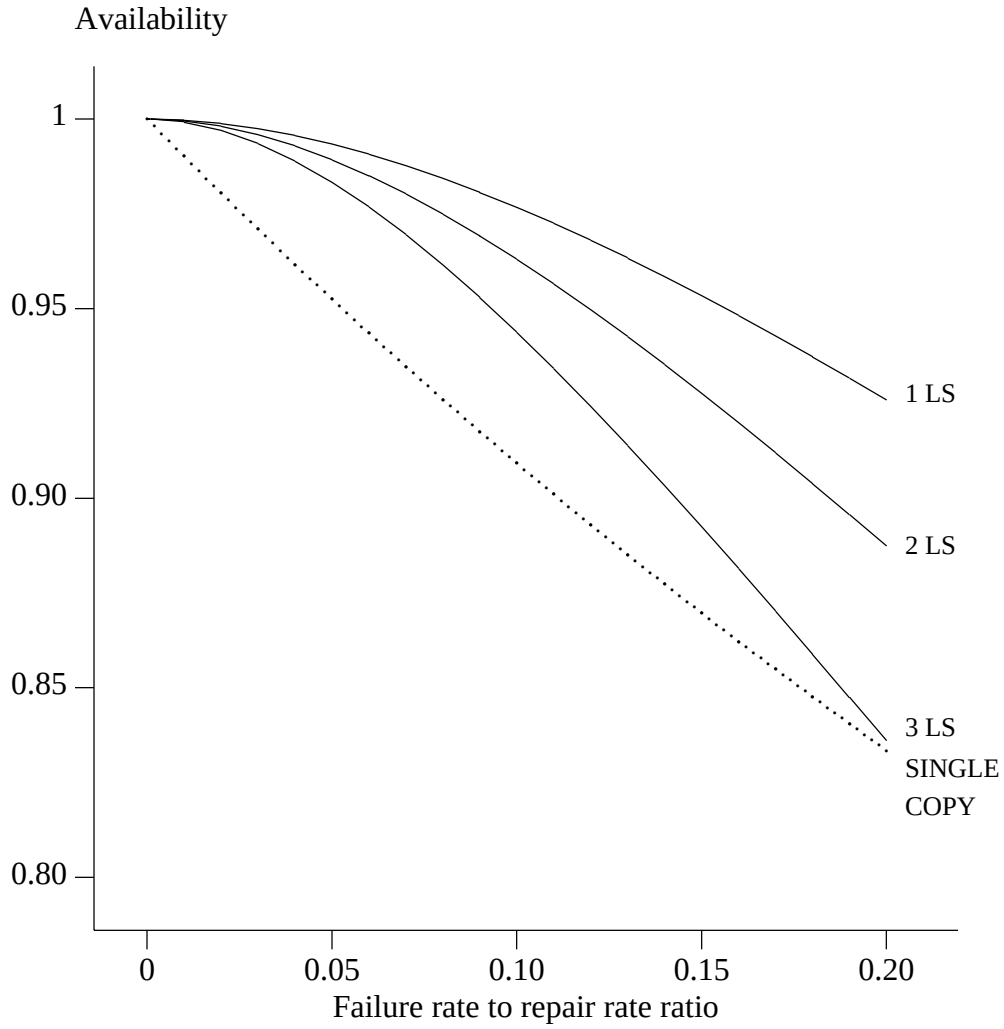


Figure 5: Compared Availabilities for Three Replicas (Majority Consensus Voting)

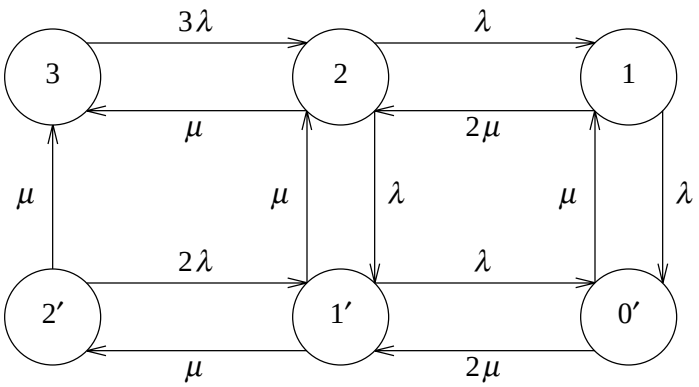


Figure 6: State Transition Diagram for DLV (Three Replicas on the Same LAN Segment)

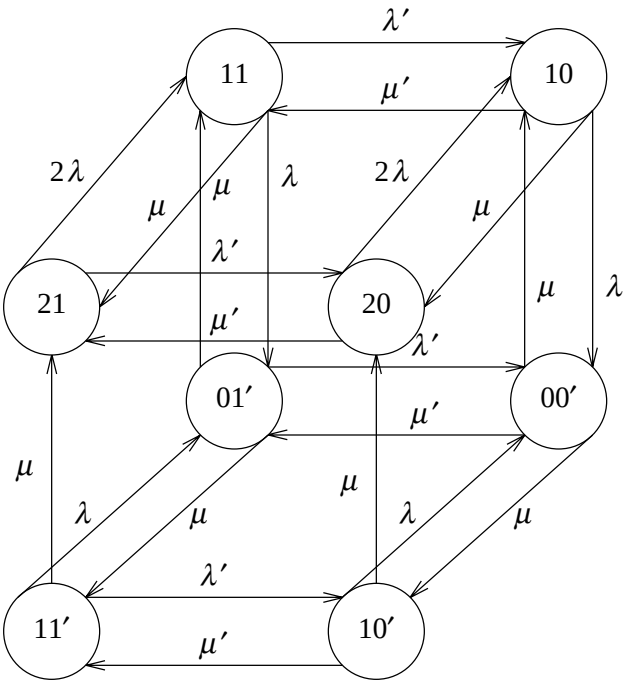


Figure 7: State Transition Diagram for DLV (Three Replicas on Two LAN Segments Separated by a Gateway)

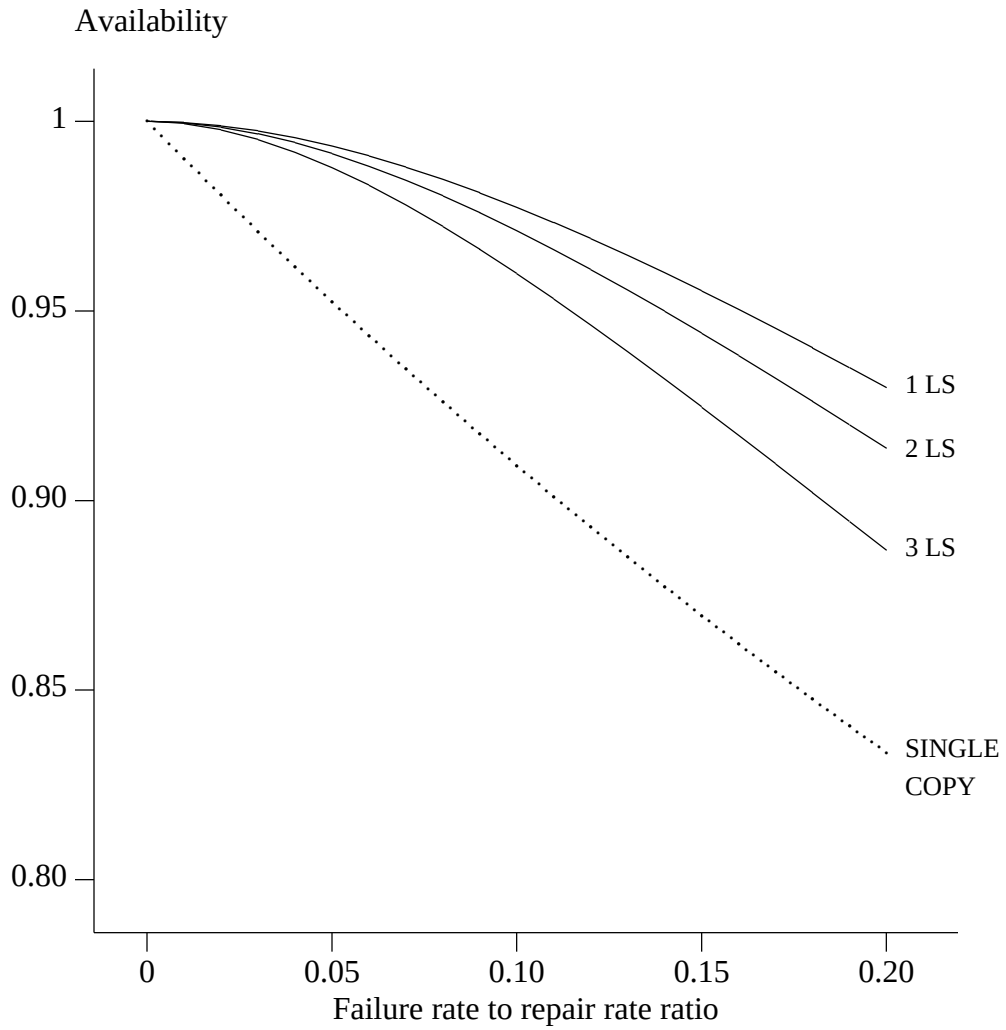


Figure 8: Compared Availabilities for Three Replicas (Dynamic-Linear Voting)