

Performance Analysis of Voting Algorithms with Non-zero Network Delay and Site Processing Time *

Yun Liu, Dongyan Chen, and Kishor S. Trivedi
Center for Advanced Computing and Communications
Department of Electrical and Computer Engineering
Duke University, Durham, NC 27708
Email: liu, dc, kst@ee.duke.edu

1 Introduction

Voting is the most popular replica control algorithm due to its simplicity in implementation and low communication and computing overhead. The basic idea of voting is to assign each site a vote. Majority consensus is required for each update request so that at any time only one update can be committed in the system.

In order to compare and evaluate different voting algorithms, analytic models have been developed to obtain the desired measures such as the site availability and the mean response time. Early studies related to availability/performance analysis of voting algorithms used Markov chain models [3]. The use of stochastic Petri nets (SPN) in this area has significantly increased the modeling capability and enabled comprehensive evaluations of voting algorithms [2]. All the previous analytic models ignored network delay and site processing time to remain tractable and easily solvable.

Although the no-delay assumption is justifiable in the situation where all the sites are contained in a local area network and the processing of each data entity is relatively simple and quick, it is not appropriate in an environment with rapidly developing heterogeneous networks, especially when geographically dispersed sites are connected by a wide area network (WAN) and used to provide web-based multimedia database services.

In this paper, we have developed a stochastic reward net (SRN) model which incorporates network delay and site processing time in a distributed database system based on a static majority voting algorithm. The model allows us to find the determining performance factors under different system conditions. The numerical results show that even a small network delay or site processing time leads to a significant degradation in performance. Our model uses the fixed point iteration technique to avoid the state space largeness problem that plagues most previous ana-

lytic models. Automated generation and solution of the underlying continuous-time Markov chain model is facilitated by our software package known as the Stochastic Petri Net Package (SPNP) [1].

2 Algorithm Description

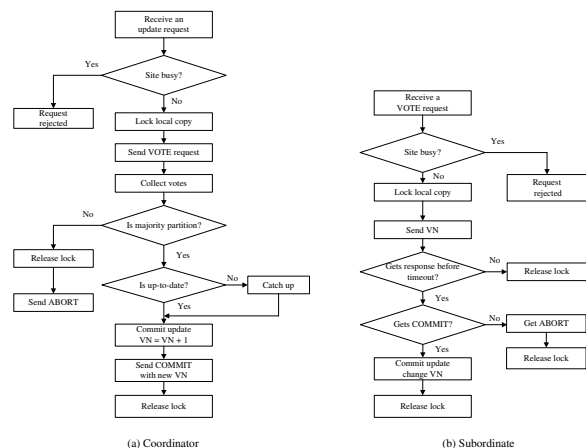


Figure 1: Flowchart of simple voting protocol

The flowchart of the static majority voting algorithm under normal operation is shown in Fig. 1.

Note that both the coordinator and the subordinates need to lock their local copies during the voting protocol. Such sites in the middle of a protocol are said to be blocked since they cannot respond to any other update or VOTE request during this period. All previous analytic models of voting algorithms ignored the occurrence of such blocking since they assumed no network delay and zero site processing time.

*Research supported by an AFOSR MURI grant no. F49620-1-0327

3 Model of Voting Protocol

The analysis of the whole system can be achieved by focusing on a single target site assuming that the sites are all statistically identical. The following is the list of other assumptions made in our model:

- Update and VOTE requests at each site form independent Poisson processes with arrival rates λ_u and λ_{vrq} , respectively.
- Network delay and site processing time are assumed to be exponentially distributed with mean λ_d^{-1} and μ_u^{-1} respectively.
- Both sites and the underlying network are assumed to be failure-free.
- No buffers are considered for update and VOTE requests in the system.

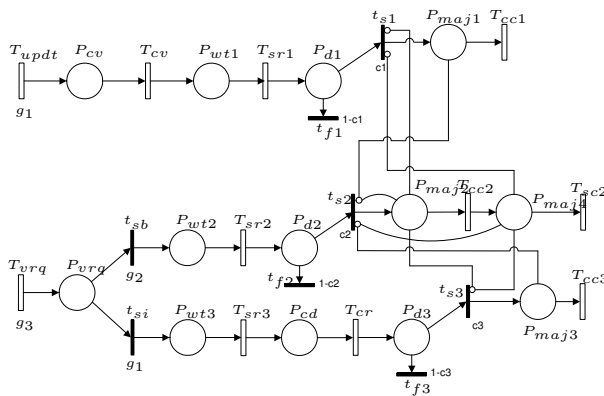


Figure 2: SRN model of simple voting algorithm

Fig. 2 shows the SRN model for the target site S_i in a distributed database system from which we can numerically compute the site available probability P_A and the mean response time T_{res} . Define reward rate R_a as: *if* (P_{cv} , P_{wt1} , P_{maj1} , P_{wt3} , P_{cd} , and P_{maj3} have no tokens) *then* $R_a = 1$, *else* $R_a = 0$. P_A is the expectation of R_a , which is computed by the built-in function in SPNP. By Little's formula, $T_{res} = L/\Lambda_{cc1}$, where L is the sum of the average number of tokens in P_{cv} , P_{wt1} , P_{maj1} , and P_{bk} and Λ_{cc1} is the throughput of transition T_{cc1} .

4 Numerical Results

In Fig. 3 and Fig. 4, we have plotted P_A and T_{res} as functions of site processing time and network delay respectively. We use $\lambda_{arv} = 1.0/N s^{-1}$ and $N = 5$. In Fig. 3, the P_A curve is almost flat when $\mu_u^{-1} < 0.1 s$. The impact becomes substantial when μ_u^{-1} exceeds $0.1 s$.

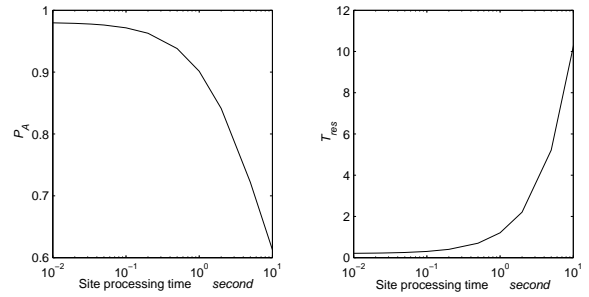


Figure 3: P_A and T_{res} vs. site processing time

For instance, P_A decreases to about 0.9 when the average site processing time is $1 s$. The corresponding T_{res} curve presents an opposite trend as P_A does. Similar site available probability and mean response time curves have been observed for the effect of network delay as shown in Fig. 4. This indicates that when network delay and site processing time are allowed in a voting system, they become a determining factor for both P_A and T_{res} .

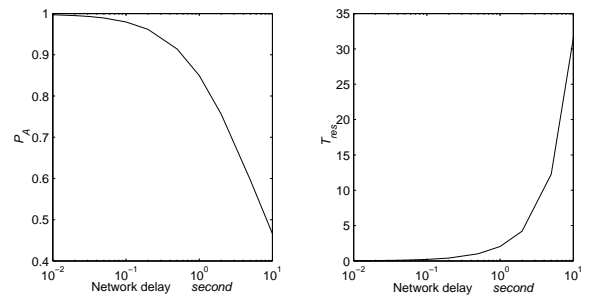


Figure 4: P_A and T_{res} vs. network delay

References

- [1] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi. *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. John Wiley, 1996.
- [2] I. R. Chen and D. C. Wang. Analyzing dynamic voting using Petri nets. In *Proc. SRDS*, Oct. 1996.
- [3] S. Jajodia and D. Mutchler. Dynamic voting algorithms for maintaining the consistency of a replicated database. *ACM Transactions on Database Systems*, 15(2):230–280, June 1990.