# Message Complexity of the Tree Quorum Algorithm for Distributed Mutual Exclusion

Her-Kun Chang and Shyan-Ming Yuan
Department of Computer and Information Science
National Chiao Tung University
1001 Ta Hsueh Road, Hsinchu 30050, Taiwan

## Abstract

The tree quorum algorithm (TQA), which logically organizes the sites in a system into a tree, is an efficient and fault-tolerant solution for distributed mutual exclusion. Quorum size can be reduced to $\log N$ in the best case of TQA. In this paper, message complexity of TQA is analyzed. Moreover, it is shown that the ratio of message complexity to quorum size converges to $1/p$, where $p$ is the probability that a site is operational.

*Keywords*: distributed system, mutual exclusion, tree quorum algorithm, quorum size, message complexity.

## 1 Introduction

A distributed system provides the capability that the sites in the system can share a common resource. However, the accesses to a common resource should be controlled such that at most one site is allowed to access the resource in any time instant. This is known as distributed mutual exclusion. A survey of various algorithms for mutual exclusion can be found in [4] and a simple taxonomy for distributed mutual exclusion algorithms was reported in [5].

*Quorum-consensus* is a popular technique for solving distributed mutual exclusion. A quorum is a set of sites with the property that any two quorums has a nonempty intersection [2]. Mutual exclusion is ensured by requiring each access to get permissions from any quorum of sites. The quorums can be defined implicitly by voting mechanisms or explicitly by coteries [2]. Moreover, in order to reduce quorum size, the quorums can be defined by imposing a logical structure on the system [1,3].

The tree quorum algorithm (TQA) [1] logically organizes the sites in a system into a tree structure. Quorum size can be reduced to $\log N$ in the best case of TQA. In [1], quorum size was used to estimate message complexity for evaluating the performance of the algorithm. It was assumed that message complexity is proportional to quorum size. However, the assumption was not proved in the work. In this paper, message complexity of TQA is analyzed. Moreover, an asymptotic analysis on the ratio of message complexity to quorum size is presented. It is shown that the ratio converges to $1/p$, where $p$ is the probability that a site is operational.

The remainder of the paper is organized as follows. In Section 2, we briefly review TQA. In Section 3, message complexity of TQA is analyzed. An asymptotic analysis on the ratio of message complexity to quorum size is presented in Section 4. Some concluding remarks are given in the final section.

## 2 Tree quorum algorithm

In the paper, we discuss only complete binary trees, which is a representative form of TQA. In this section, we briefly review TQA (see [1] for details).

TQA logically organizes the sites in a system into a tree structure. For a binary tree, a tree quorum (recursively) consists of

1. the root and a tree quorum of the left subtree, or
2. the root and a tree quorum of the right subtree, or
3. a tree quorum of the left subtree and a tree quorum of the right subtree.

It was shown in [1] that any two tree quorums intersect with each other. Thus mutual exclusion is ensured by requiring that each access to get permissions from any (tree) quorum of sites.



1063-6927/94 $03.00 © 1994 IEEE

Consider the tree structure in Figure 1. The quorums (corresponding to above definition) are:

1. $\{1,2,4\}$, $\{1,2,5\}$, $\{1,4,5\}$,
2. $\{1,3,6\}$, $\{1,3,7\}$, $\{1,6,7\}$,
3. $\{2,4,3,6\}$, $\{2,4,3,7\}$, $\{2,4,6,7\}$, $\{2,5,3,6\}$, $\{2,5,3,7\}$, $\{2,5,6,7\}$, $\{4,5,3,6\}$, $\{4,5,3,7\}$, $\{4,5,6,7\}$.

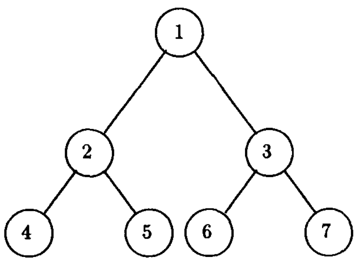


Figure 1: A 7-node binary tree.

The sites are either *operational* or *failed*. The state (operational or failed) of a site is independent to the others. The probability that a site is operational is referred to as the *availability* of the site. A (tree) quorum is *available* if all sites in the quorum are operational. The *availability* of a tree is defined to be the probability that there is an available tree quorum.

*Message complexity* is defined to be the expected number of messages required per invocation of the algorithm. *Quorum size* is defined to be the average number of sites that form a quorum.

In the paper, we use the following notation:

- $p$ : availability of a single site, $1/2 < p < 1$,
- $A_i$ : availability of an $i$ level tree, $i \geq 0$,
- $C_i$ : quorum size of an $i$ level tree, $i \geq 0$,
- $M_i$ : message complexity of an $i$ level tree, $i \geq 0$,
- $R_i$ : $M_i/C_i$, $i \geq 0$.

The availability of a tree is the probability that there is an available tree quorum. Thus the availability of a binary tree is the probability that

1. the root is operational and a tree quorum of the left or right subtree is available, or
2. a tree quorum of the left subtree and a tree quorum of the right subtree are available.

Formally, for a $k$ level complete binary tree with root at level $k$ and leaves at level 0, $A_i$, $1 \leq i \leq k$, is given as

$$\begin{aligned} A_i &= p(1-(1-A_{i-1})^2)+(1-p)A_{i-1}^2 \\ &= A_{i-1}^2 + 2pA_{i-1}(1-A_{i-1}) \end{aligned} \quad (1)$$

where $A_0 = p$ is the availability of a subtree with a single site and the availability of the whole tree is $A_k$.

Note that if $p \leq \frac{1}{2}$, $A_i - A_{i-1} = (2p-1)\, A_{i-1}(1-A_{i-1}) \leq 0$, i.e. $A_i \leq A_{i-1}$, for all $i \geq 1$. So we consider only the case that $\frac{1}{2} < p < 1$.

Consider a tree quorum at level $i$, $i \geq 1$. If the root is operational, the quorum size is $1 + C_{i-1}$; otherwise, the quorum size is $2C_{i-1}$. Thus, for $i \geq 1$, $C_i$ can be computed by the following recurrence:

$$\begin{aligned} C_i &= p(1+C_{i-1}) + (1-p)2C_{i-1} \\ &= (2-p)C_{i-1} + p \end{aligned} \quad (2)$$

## 3 Message complexity

The recursive definition of the tree quorums in the previous section also define the quorum construction algorithm. That is, if the root is operational, then the construction algorithm tries to construct a (tree) quorum form left or right subtree; otherwise, it must construct quorums from both left and right subtrees. In other words, the quorum construction algorithm first visits the root and then traverses the left and/or right subtrees (in some specified order or randomly). The major difference between message complexity and quorum size is that no matter whether a visited site is operational or not, a message is spent.

For a $k$ level complete binary tree, with root at level $k$ and leaves at level 0, $M_i$ is the expected number of messages required to construct a tree quorum at level $i$, $0 \leq i \leq k$. Consider the construction of a tree quorum at level $i$. Without loss of generality, we assume that the quorum construction algorithm tries to form a quorum (recursively) from left subtree to right subtree. Regardless the root is operational or not, one message must be spent. The additional messages required are described as below.

1. if the root is operational and a quorum of the left subtree is available – only messages for traversing the left subtree are spent, i.e. $M_{i-1}$ messages are needed;

2. if the root is failed and no quorum of the left subtree is available – since it is impossible to form a quorum, only messages for traversing the left subtree are spent, i.e. $M_{i-1}$ messages are needed;

3. otherwise – messages are spent for traversing both left and right subtrees, i.e. $2M_{i-1}$ messages are required.

Formally, for $i \geq 1$,

$$\begin{aligned} M_i &= 1 + (pA_{i-1} + (1-p)(1-A_{i-1}))M_{i-1} \\ &\quad + (p(1-A_{i-1}) + (1-p)A_{i-1})2M_{i-1} \\ &= (1 + p + A_{i-1} - 2pA_{i-1})M_{i-1} + 1 \qquad (3) \end{aligned}$$

Note that $M_0 = 1$ and $M_k$ is the expected number of messages required for every invocation of the constructing algorithm.

## 4 Asymptotic analysis of $R_i$

**Lemma 1** For TQA, $\frac{1}{2} < p < 1$, $|R_{i+m} - R_i| < ax^i + y^i$, $i \geq 0, m \geq 1$, where

$$\begin{aligned} a &= \frac{1-p}{p(2-p)} e^{\frac{1+p-p^2}{p(2-p)}} \\ x &= 1 + p - 2p^2 \\ y &= \frac{1}{2-p} \end{aligned}$$

PROOF. The proof is shown in the appendix.

**Theorem 1** If $\frac{1}{2} < p < 1$,

$$\lim_{i \to \infty} R_i = \frac{1}{p}$$

PROOF.

Let $a$, $x$ and $y$ be defined as in Lemma 1. Since $p > \frac{1}{2}$, $x = (1 + p - 2p^2) < 1$ and $y = \frac{1}{2-p} < 1$.

Thus, for any $\epsilon > 0$, we can find a positive integer N such that

if $i > N$ and $m \geq 1$, then

$$|R_{i+m} - R_i| < ax^i + y^i < ax^N + y^N < \epsilon$$

That is, the sequence $\{R_i\}$ is convergent.

Let $\lim_{i \to \infty} R_i = \gamma$. As $i$ goes to infinity, $A_i = 1$ and

$$R_i = R_{i-1} = \gamma$$

that is,

$$\frac{M_i}{C_i} = \frac{M_{i-1}}{C_{i-1}} = \gamma$$

Thus,

$$\frac{(2-p)\gamma C_{i-1} + 1}{(2-p)C_{i-1} + p} = \gamma$$

$$\gamma = \frac{1}{p}$$

Q.E.D.

## 5 Conclusion

The tree quorum algorithm (TQA) is an efficient solution for distributed mutual exclusion. TQA can reduce quorum size to $\log N$ in the best case. Quorum size was used to appraise the performance of TQA based on the assumption that message complexity is proportional to quorum size. However, the assumption was not proved in the work [1]. In this paper, we analyze message complexity of TQA. Moreover, an asymptotic analysis on the ratio of message complexity to quorum size is presented. It is shown that the ratio converges to $1/p$, where $p$ is the probability that a site is operational.

## Acknowledgment

This work was supported in part by NSC under grant NSC82-0408-E009-289.

## References

[1] D. Agrawal and A. El Abbadi, "An Efficient and Fault-Tolerant Solution for Distributed Mutual Exclusion," ACM Trans. Computer Systems, Vol. 9, pp. 1–20, 1991.

[2] H. Garcia-Molina and D. Barbara, "How to Assign Votes in a Distributed System," J. ACM, Vol. 32, pp. 841–860, 1985.

[3] M. Maekawa, "A $\sqrt{N}$ Algorithm for Mutual Exclusion in Decentralized Systems," ACM Trans. Computer Systems, Vol. 3, pp. 145–159, 1985.

[4] M. Raynal, *Algorithms for Mutual Exclusion*, The MIT Press, 1986.

[5] M. Raynal, "A Simple Taxonomy for Distributed Mutual Exclusion Algorithms," Operating Systems Review, Vol. 25, pp. 47–50, 1991.

## Appendix

**Lemma 2** If $0 < ax_i < 1$, for all $i$, then

$$(1+ax_1)(1+ax_2)\cdots(1+ax_n) < e^{a\sum x_i}$$

PROOF.
It is known that

$$\ln(1+z) = z - \frac{z^2}{2} + \frac{z^3}{3} - \frac{z^4}{4} + - \cdots$$

Hence, for any $0 < z < 1$,

$$\ln(1+z) < z$$

Then, if $0 < ax_i < 1$, for all $i$, we have

$$\ln((1+ax_1)(1+ax_2)\cdots(1+ax_n)) < a\sum x_i$$

That is,

$$(1+ax_1)(1+ax_2)\cdots(1+ax_n) < e^{a\sum x_i}$$

Q.E.D.

**Lemma 3** For TQA, $\frac{1}{2} < p < 1$, $A_i$ has the following properties:

1. $1 - A_i \leq (1-p)(1+p-2p^2)^i$, for all $i \geq 0$.
2. $(1-A_i) + \cdots + (1-A_{i+m}) < \frac{1-p}{p(2p-1)}(1+p-2p^2)^i$, for all $i, m \geq 0$.

PROOF.
$1 - A_0 = 1 - p$ and for $i \geq 1$,

$$\begin{aligned} 1 - A_i &= 1 - A_{i-1}^2 - 2pA_{i-1}(1-A_{i-1}) \\ &= (1 + (1-2p)A_{i-1})(1-A_{i-1}) \\ &\leq (1+(1-2p)p)(1-A_{i-1}) \\ &= (1+p-2p^2)A_{i-1} \end{aligned}$$

Let $x = 1 + p - 2p^2$, by iteration,

$$1 - A_i \leq (1-p)x^i$$

Then,

$$\begin{aligned} & (1-A_i) + (1-A_{i+1}) + \cdots + (1-A_{i+m}) \\ \leq\ & (1-p)x^i\{1 + x + \cdots + x^m\} \\ <\ & (1-p)x^i \frac{1}{1-x} \\ =\ & \frac{1-p}{p(2p-1)}(1+p-2p^2)^i \end{aligned}$$

Q.E.D.

**Lemma 4** For TQA, $C_i$ has the following properties:

1. $C_i = \frac{(2-p)^i - p}{1-p} > (2-p)^i$, for all $i \geq 0$.
2. $\frac{1}{C_i} + \cdots + \frac{1}{C_{i+m}} < \frac{2-p}{1-p}\frac{1}{(2-p)^i}$, for all $i, m \geq 0$.

PROOF.
First, $C_0 = 1$ and for $i \geq 1$, by equation (2),

$$C_i = (2-p)C_{i-1} + p \qquad (4)$$

Solving recurrence (4) by iteration, we have

$$C_i = \frac{(2-p)^i - p}{1-p} > (2-p)^i$$

Then,

$$\begin{aligned} & \frac{1}{C_i} + \cdots + \frac{1}{C_{i+m}} \\ <\ & \frac{1}{(2-p)^i}\{1 + \cdots + \frac{1}{(2-p)^m}\} \\ <\ & \frac{1}{(2-p)^i}\{\frac{1}{1-\frac{1}{(2-p)}}\} \\ =\ & \frac{2-p}{1-p}\frac{1}{(2-p)^i} \end{aligned}$$

Q.E.D.

**Lemma 5** For TQA, $\frac{1}{2} < p < 1$,

$$R_i < e^{\frac{1+p-p^2}{p(2-p)}}$$

PROOF. For $i \geq 1$,

$$\begin{aligned} R_i &= \frac{M_i}{C_i} \\ &= \frac{(1+p+A_{i-1}-2pA_{i-1})M_{i-1}+1}{(2-p)C_{i-1}+p} \\ &= (\frac{(1+p+A_{i-1}-2pA_{i-1})C_{i-1}+\frac{1}{R_{i-1}}}{(2-p)C_{i-1}+p})R_{i-1} \\ &= (1 + \frac{(2p-1)(1-A_{i-1})C_{i-1}}{(2-p)C_{i-1}+p} \\ &\quad + \frac{\frac{1}{R_{i-1}} - p}{(2-p)C_{i-1}+p})R_{i-1} \\ &< (1 + \frac{(2p-1)(1-A_{i-1})}{2-p} + \frac{1-p}{C_i})R_{i-1} \end{aligned}$$

By iteration,

$$\begin{aligned} R_i &< (1 + \frac{(2p-1)(1-A_0)}{2-p} + \frac{1-p}{C_1}) \\ &\quad \cdots (1 + \frac{(2p-1)(1-A_{i-1})}{2-p} + \frac{1-p}{C_i}) \end{aligned}$$

According to Lemmas 2 3 and 4,

$$
\begin{aligned}
R_i &< e^{(\frac{2p-1}{2-p}((1-A_0)+\cdots+(1-A_{i-1}))+(1-p)(\frac{1}{C_1}+\cdots+\frac{1}{C_i}))} \\
&< e^{(\frac{2p-1}{2-p}\frac{1-p}{p(2p-1)}+(1-p)(\frac{2-p}{1-p})\frac{1}{2-p})} \\
&= e^{\frac{1+p-p^2}{p(2-p)}}
\end{aligned}
$$

Q.E.D.

**Proof of Lemma 1:**

Let $\delta_{i+1} = R_{i+1} - R_i$, then

$$
\begin{aligned}
\delta_{i+1} &= \frac{(1+p+A_i-2pA_i)M_i+1}{(2-p)C_i+p} - \frac{M_i}{C_i} \\
&= \frac{(2p-1)(1-A_i)M_i+1-p\frac{M_i}{C_i}}{(2-p)C_i+p} \\
&< \frac{(2p-1)(1-A_i)M_i}{(2-p)C_i} + \frac{1-p}{C_{i+1}} \\
&= \frac{(2p-1)(1-A_i)}{(2-p)}R_i + \frac{1-p}{C_{i+1}}
\end{aligned}
$$

According to Lemma 5

$$
\delta_{i+1} < \frac{(2p-1)(1-A_i)}{(2-p)}e^{\frac{1+p-p^2}{p(2-p)}} + \frac{1-p}{C_{i+1}}
$$

$$
\begin{aligned}
|R_{i+m} - R_i| &= |\delta_{i+m} + \delta_{i+m-1} + \cdots + \delta_{i+1}| \\
&< \frac{2p-1}{(2-p)}e^{\frac{1+p-p^2}{p(2-p)}}((1-A_i)+\cdots+(1-A_{i+m-1})) \\
&\quad +(1-p)(\frac{1}{C_{i+1}}+\cdots+\frac{1}{C_{i+m}})
\end{aligned}
$$

According to Lemma 3 and Lemma 4

$$
\begin{aligned}
|R_{i+m} - R_i| &< \frac{2p-1}{(2-p)}e^{\frac{1+p-p^2}{p(2-p)}}\frac{1-p}{p(2p-1)}(1+p-2p^2)^i \\
&\quad +(1-p)\frac{2-p}{1-p}\frac{1}{(2-p)^{i+1}} \\
&= \frac{1-p}{p(2-p)}e^{\frac{1+p-p^2}{p(2-p)}}(1+p-2p^2)^i + (\frac{1}{2-p})^i \\
&= ax^i + y^i
\end{aligned}
$$

where

$$
\begin{aligned}
a &= \frac{1-p}{p(2-p)}e^{\frac{1+p-p^2}{p(2-p)}} \\
x &= 1+p-2p^2 \\
y &= \frac{1}{2-p}
\end{aligned}
$$

Q.E.D.