CORRIGENDUM

AN IMPROVEMENT OF THE MAEKAWA'S MUTUAL EXCLUSION ALGORITHM TO MAKE IT FAULT-TOLERANT

[*Parallel Processing Letters* Vol. 2 No. 2 & 3 (1992) 283-290]

Y.-C. KUO and S.-T. HUANG

*Department of Computer Science, National Tsing Hua University
 HsinChu, Taiwan, 30043 R. O. C.
 Email: sthuang@cs.nthu.edu.tw*

In a recent paper, Bouabdallah and König [2] presented a distributed mutual exclusion algorithm to improve the fault-tolerance of Maekawa's result [1]. However, the algorithm does not guarantee mutual exclusion. The paper concludes, "To enter its critical section, a process i is required to receive at least $|S_i| - t + 1$ LOCKED messages. From Proposition 1, S_i 's are chosen in such a way that for all i and j , $|S_i \cap S_j| \geq t$. Thus, in the presence of $t - 1$ or less arbitrary failures, the intersection is not empty. Since each process gives permission to only one process, mutual exclusion is guaranteed in the presence of $t - 1$ or less failures." [2, p. 289, lines 10-14] It is incorrect. The statement " $S_i \cap S_j$ is not empty" does not imply that the intersection between a subset of S_i that contains $|S_i| - t + 1$ elements and a subset of S_j that contains $|S_j| - t + 1$ elements is also not empty. That is, the algorithm does not meet the mutual exclusion requirement when the system contains less than $t - 1$ node failures.

As a concrete counterexample to the algorithm, consider a system with $n = 5$, and $t = 3$. According to the algorithm, it can tolerate one or two arbitrary node failures. By Proposition 1 (for more details see [2], p. 284, line 24), the following sets can be derived which satisfy ($\forall i, j : 0 \leq i, j \leq 4 : |S_i \cap S_j| \geq 3$).

$$\begin{aligned} S_0 &= \{0, 1, 2, 3\}, & S_1 &= \{4, 0, 1, 2\}, & S_2 &= \{3, 4, 0, 1\}, & S_3 &= \{2, 3, 4, 0\} \\ S_4 &= \{1, 2, 3, 4\}, & |S_0| &= |S_1| = |S_2| = |S_3| = |S_4| = 4. \end{aligned}$$

Assume that both processes 0 and 4 request to enter their critical section at the same time. Process 0 needs to wait for LOCKED messages from processes in S_0 ; and process 4 needs to wait for LOCKED messages from processes in S_4 . If process 0 receives LOCKED messages from processes 0 and 1, and process 4 receives LOCKED messages from processes 3 and 4. According to the algorithm, both processes 0 and 4 are allowed to enter their critical section because $|S_i| - t + 1 = 4 - 3 + 1 = 2$.

Although the conclusion is incorrect, it can be revised as, "To enter its critical section, a process i is required to receive at least $|S_i| - \lceil t/2 \rceil + 1$ LOCKED messages. From Proposition 1, S_i 's are chosen in such a way that for all i and j , $|S_i \cap S_j| \geq t$. Thus, in the presence of $\lceil t/2 \rceil - 1$ or less arbitrary failures, the intersection is not empty." This revision guarantees mutual exclusion because $|S_i| - \lceil t/2 \rceil + 1$ LOCKED messages contain a majority of LOCKED messages from processes in $S_i \cap S_j$.

References

- [1] M. Maekawa, A $\sqrt{n}$ algorithm for mutual exclusion in decentralized system, *ACM Trans. Comput. Syst.* **3**, 2 (1985) 145–159.
- [2] A. Bouabdallah and J.-C. König, An improvement of Maekawa's mutual exclusion algorithm to make it fault-tolerant, *Parallel Process. Lett.* **2**, 2&3 (1992) 283–290.