# Corrigendum for “Scalable Leader Election”

Valerie King and Jared Saia

August 8, 2022

## 1 The problem

This corrigendum fixes a problem in the 2006 SODA paper [1], which was pointed out to the authors by Kartik Nayak and Sravya Yandmuri in May 2022.

In [1], at each level, starting with the lowest, small sets of processors run elections at election nodes to reduce the number of candidates for the leadership. Monitoring sets are used to communicate election results between these sets. Processors which do not win or which win too many elections drop out of the competition in higher levels. A monitoring set and election nodes are called *bad* if the fraction of processors assigned to the node which are bad exceeds the fraction of bad processors in the network by a parameter which depends on the level of the set.

The problem is that bad monitoring sets may cause good processors to think they have won a large number of elections and thereby drop out. Note that every good election node by definition has a good monitoring set; the problem is that bad election nodes with potentially bad monitoring sets may convince too many good participants in bad elections that they are winners of these elections, and hence drop out.

## 2 Solution

The algorithm can be modified as follows. First, monitoring sets can be increased in size so that at every level, the fraction of bad monitoring sets is reduced by a $O(1/\log n)$ factors by expanding the size increase by only a polylogarithmic amount. Second, a new communication protocol can be introduced to enable good monitoring sets to prevent equivocation of the election outcome even when their election node is bad.

However, rather than give a detailed explanation of the changes here, we note that our paper “Towards Secure and Scalable Communication in Peer-To-Peer Networks” published later the same year [2] correctly addresses this problem in a way somewhat similar to the solution described above. Additionally, the algorithm in [2] is a stronger result in that it requires only a sparse communication graph, as opposed to the algorithm in [1], which requires a complete communication graph.

Briefly, in [2], for each election node, there is a *s-node*, a set of processors for each election which communicates winners between levels. The s-nodes are analogous to monitoring sets. But, their size starts and grows more quickly, so that the fraction of s-nodes that are bad at each layer of the election graph are kept to a $1/\log^{10} n$ fraction.

Additionally, there are communication protocols between s-nodes (Section 5 of [2]) that prevent a bad election node from passing on more than 7 different sets of winners to the next layer. In particular, see Claim 7.3 of [2].

# References


[1] Valerie King, Jared Saia, Vishal Sanwalani, and Erik Vee. Scalable leader election. In *Proceedings of ACM-SIAM symposium on Discrete algorithm*, pages 990–999, 2006.

[2] Valerie King, Jared Saia, Vishal Sanwalani, and Erik Vee. Towards secure and scalable computation in peer-to-peer networks. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 87–98. IEEE, 2006.