DOI 10.1007/s10878-012-9534-3

A simple Byzantine Generals protocol

Junxing Wang

© Springer Science+Business Media, LLC 2012

Abstract The *Byzantine Generals Problem* is a classical problem in distributed computing that models a system's resiliency against arbitrary adversarial faults. The existing solutions to this problem tend to be quite intricate and many of them employ some form of recursion. This paper gives a new algorithm that solves the problem in an exceptionally simple straight-line program.

Keywords Byzantine Generals problem · Distributed algorithm · Fault tolerant

1 Introduction

A reliable computer system must be able to cope with the failure of one or more of its components. A failed component may send conflicting information to different parts of the system. The problem of coping with this type of failure can be modeled abstractly as the Byzantine Generals Problem, first introduced and solved in the seminal papers by Lamport et al. (1982), Pease et al. (1980). Improved solutions to the problem as well as many variations have been studied in the ensuing decades (see for example Attiya and Welch 2004; Dolev and Strong 1983; Lynch 1996). The existing Byzantine Generals algorithms tend to be quite intricate and many of them employ some form of recursion. In this paper we give a new algorithm that solves the problem in an exceptionally simple straight-line program. It uses the same number of bits of messages as the original solution given by Lamport et al. (1982).

This research was supported in part by the Natural Science Foundation of China (NSFC) under grants 61033001 and 61061130540.

J. Wang (✉)
Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing 100084, China
e-mail: thuwjx@163.com

Published online: 01 August 2012

We consider the standard formulation of the Byzantine Generals Problem in a synchronous environment as follows. Let there be n generals $\{G_0, G_1, \ldots, G_{n-1}\}$, where G_0 is the commanding general (or *commander* for short). The generals can communicate with one another only by sending 1-to-1 messages. Assuming that the commander holds an information bit r_0 initially, the goal is to design a protocol so that all the generals will eventually share the same information bit. However, some of the generals including possibly the commander himself may be traitors, trying to prevent the loyal generals from reaching an agreement. (It is assumed that a loyal general will always execute a protocol faithfully, while a traitor may do anything as it wishes.) A correct protocol should satisfy the following conditions:

1. All loyal generals will reach an agreement (i.e., holding the same bit) at the end;
2. If the commander is loyal, every loyal general will hold r_0 at the end.

Let t be the number of traitors among the n generals. A protocol was given in Lamport et al. (1982) for the case $n > 3t$ (referred to as the Oral Message algorithm in Lamport et al. 1982); furthermore it was shown that $n > 3t$ is necessary for a solution to exist.

2 New solution

We consider the Byzantine Generals Problem with n generals including at most t traitors, where $n > 3t$.

Each general G_i will maintain a register r_i with a bit value, which will represent G_i's decision when the algorithm terminates. At the start, the commander G_0 sends his value r_0 to all the generals. Then, the algorithm will reset the generals' registers during $\binom{n-1}{n-t}$ successive rounds. In each round, a subset A consisting of $n-t$ generals chosen from $\{G_1, G_2, \ldots, G_{n-1}\}$ will synchronously send their values to all the generals, and each general will then reset his register to equal the majority value of the $n-t$ bits just received. (We shall refer to such a round as *round A*.) The algorithm is given in the following pseudo-code.

```
Commander $G_0$ sends $r_0$ to all.
for $A \subseteq \{1, 2, \ldots, n-1\}, |A| = n-t$ do
    for $i \in A$ do
        $G_i$ sends $r_i$ to all;
    end for
    for $j \in \{1, 2, \ldots, n-1\}$ do
        $r_j \leftarrow majority(R_{j \leftarrow A})$
    end for
end for
Each $G_i$ takes $r_i$ as his answer.
```

Algorithm 1: $BG(n, t)$

The notation $R_{j \leftarrow A}$ in Line 7 represents the set of all register values that G_j has received from A at that point. As remarked before, a traitor is not expected to follow the protocol exactly. In particular, any traitor G_i within subset A may send different values to different generals in Line 4 (and in the case that G_i sends no value to a general, the latter will receive a default value). Also, a traitor G_j may not always reset its register value in accordance with Line 7.

Theorem 1 *Let $n > 3t$. Algorithm 1 solves the Byzantine Generals Problem for n generals and at most t traitors.*

Proof First, we observe that Algorithm 1 satisfies the *monotone* property in the following sense: if at the end of some round A', all loyal generals G_i hold a single value in their r_i, then that single value will be maintained by all loyal generals in subsequent rounds after A'. This is because, in any round A the loyal generals within subset A always form the majority since $|A| = n - t > 2t$ (i.e., the traitors are the minority).

We now prove the theorem by considering two cases, depending on whether the commander G_0 is loyal. First assume that G_0 is loyal. In this case all the loyal general G_i will set $r_i = r_0$ at the start of the algorithm. The monotone property then guarantees that $r_i = r_0$ will be true for all the loyal generals G_i at the end of the algorithm. Next consider the case when the commander himself is not loyal. Then there are at least $(n-1)-(t-1) = n-t$ loyal generals among $\{G_1, G_2, \dots, G_{n-1}\}$. There must exist a round A^* when the chosen subset A^* will consist entirely of loyal generals. In this round, the same set of bits $R_{j \leftarrow A^*}$ is guaranteed to be sent to all the general G_j. This enables all loyal general G_j to update their r_j to the same majority value. Moreover, this value will be preserved until the end of the algorithm because of the monotone property. This completes the proof of the theorem. □

The number of rounds used by Algorithm 1 is $\binom{n-1}{n-t}$, or equivalently $\binom{n-1}{t-1}$, and in each round $(n-2) \times (n-t-1)$ bits are sent. Thus Algorithm 1 has total cost of $O(\binom{n-1}{t-1} n^2)$, which is the same complexity as the original solution by Lamport et al. (1982). However, Algorithm 1 stands out in its simplicity when compared with Lamport et al. (1982) and other existing Byzantine protocols.

3 Conclusion

The Byzantine Generals Problem is a fundamental problem in distributed algorithms and has been studied extensively in many variations. However, a simple and lucid solution to its basic formulation has heretofore been lacking. In this paper we give a new protocol which has the merit of being extremely simple. It uses the same number of bits of messages as the original solution given by Lamport et al. (1982).

Acknowledgements I would like to thank Prof. Andrew Yao for his encouragement and guidance on this research. I would also like to thank Prof. Danny Dolev for his helpful comments and Prof. Frances Yao for her careful reading of the manuscript.

References

Attiya H, Welch J (2004) Distributed computing: fundamentals, simulations and advanced topics. Wiley, New York

Dolev D, Strong HR (1983) Authenticated algorithms for Byzantine agreement. SIAM J Comput 12(4):656–666

Lamport L, Shostak RE, Pease MC (1982) The Byzantine Generals problem. ACM Trans Program Lang Syst 4(3):382–401

Lynch NA (1996) Distributed algorithms. Morgan Kaufmann, San Fransisco

Pease MC, Shostak RE, Lamport L (1980) Reaching agreement in the presence of faults. J ACM 27(2):228–234