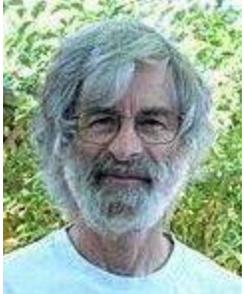


# 《拜占庭将军问题》

|      |   |
|------|---|
| 分类   | <a href="#">IT类 / 网络 / 分布式 / 容错性</a>  |
| 作者   | (美) <a href="#">Leslie Lamport</a> / <a href="#">莱斯利·兰波特</a><br><br>Robert Shostak / 罗伯特·肖斯塔克<br>Marshall Pease / 马歇尔·皮斯 |
| 英文书名 | 《Byzantine Generals Problem》  |
| 发表年份 | 1982(原著)  |
| 相关链接 | <a href="#">Byzantine fault tolerance @ Wikipedia</a><br><a href="#">拜占庭将军问题 @ 维基百科</a>   |

## 简介

第一作者 Lamport 是美国计算机科学家，2013年图灵奖得主，[ACM](#) 院士，[微软研究院](#) 的研究员。另外，他也是 [LaTeX](#) 里面的那个“La”。

他在分布式领域很有影响，提出了“[Paxos算法](#)”和“[拜占庭将军问题](#)”。

网盘上的英文 PDF 是“拜占庭将军问题”的原创论文，最早发表于1982年的 [ACM](#) 刊物上。

这是分布式领域关于“容错性”的经典难题，它还有如下几个不同叫法：

- Byzantine Fault Tolerance (BFT)
- Byzantine Agreement Problem
- Byzantine Generals Problem
- Byzantine Failure

该问题主要是指：当分布式系统中存在【**恶意节点**】(故意搞破坏的节点)时，如何让系统具备容错性。

作者在论文中描述了如下问题：

一组拜占庭将军分别各率领一支军队共同围困一座城市。为了简化问题，将每支军队的行动策略限定为“进攻”或“撤离”两种。如果部分军队进攻、部分军队撤离，可能会造成灾难性后果。因此各位将军必须通过投票来达成【一致策略】——要么所有军队一起进攻，要么所有军队一起撤离。由于各位将军分处城市不同位置，他们只能通过信使互相联系。在投票过程中，每位将军都投票给“进攻”或“撤退”，并通过信使把自己的决定通知其他所有将军。这样一来，每位将军根据自己的投票和其他所有将军送来的信息就可以知道共同的投票结果，然后决定自己的行动策略。

问题的【难点】在于：

将军中可能出现叛徒，叛徒不仅可以选择更坏的策略，还可以发送出【误导性】的投票信息。假设有9位将军投票，其中有1名叛徒。8名忠诚的将军中出现了4人投进攻，4人投撤离的情况。这时候叛徒可能故意给4名投进攻的将领送信表示投票进攻，而给4名投撤离的将领送信表示投撤离。这样一来，在4名倾向进攻的将领看来，投票结果是5人投进攻，从而发起进攻；而在4名倾向撤离的将军看来则是5人投撤离。于是，协同行动就被破坏啦。

由于将军之间需要通过信使通讯，即使在保证所有将军忠诚的情况下，也不能排除信使被敌人截杀，甚至被敌人间谍替换等情况。因此很难通过保证人员可靠性及通讯可靠性来解决问题。

假设那些忠诚的将军能够通过某种算法来达到一致行动（即便有少数叛徒也无法破坏这种一致性），那么这种算法就被称之为【拜占庭容错】。

[【编程随想】收藏的电子书清单](#)

[【编程随想】的博客](#)