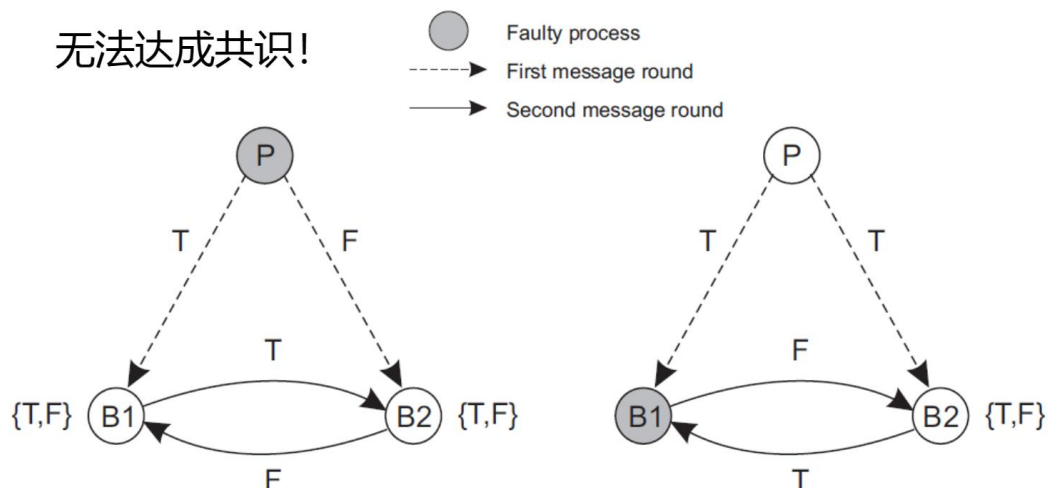


1. Can the model of triple modular redundancy (三倍模块冗余) handle Byzantine failures?

Why?

## 拜占庭容错：3k进程无法k容错



- Round 1: the faulty primary P is sending two different values to the backups B1 and B2, respectively.
- Round 2: in order to reach consensus, both backup processes forward the received value to the other.
- B1 and B2 each have received the set of values {T, F}, impossible to draw a conclusion.

三倍模块冗余无法解决拜占庭容错

如上图中的两种情况：

①有故障的主节点 P 将两个不同的值分别发送给备份节点 B1 和 B2。为了达成共识，两个备份进程将接收到的值转发给对方。 B1 和 B2 各自收到了值集 {T, F}

②有故障的节点 B1 错误的值分别发送给备份节点 B2。 B2 各自收到了值集 {T, F}

以上两种情况都显示了三倍模块冗余无法解决拜占庭容错

2. In reliable multicasting, is it always necessary that the communication layer keeps a copy of a message for retransmission purposes?

通信层不一定需要保存消息副本

①当发送方发送消息时，如果接收方未收到信息则反馈一个消息，发送方重传消息。在这种情况下，通信层需要存储消息的副本，以便在需要进行重传。

②使用定时可靠多播协议，其中消息会按照一定的时间间隔重复发送，而无需等待接收方的确认。这种方法不需要在通信层保留消息的副本，因为它使用固定的时间间隔进行重传，而不依赖于接收方的确认。

3. In the two-phase commit protocol, why can blocking never be completely eliminated, even when the participants elect a new coordinator?

两阶段提交协议：

①准备阶段（Prepare Phase）：

协调者向所有参与者发送事务请求，并询问是否可以进行事务提交。

每个参与者执行事务操作，并将执行结果和是否可以提交的信息反馈给协调者。

②提交阶段（Commit Phase）：

协调者根据所有参与者的反馈情况，决定事务是否可以提交。

如果所有参与者都同意提交，协调者向所有参与者发送提交请求；否则，协调者发送中止请求。

但两阶段提交协议无法判断协调者的最后决定是否到达所有参与者，导致 2PC 可能被阻塞。

4. Does a stateless server need to take checkpoints?

检查点通常与具有持久状态的系统相关。在这种情况下，检查点用于定期保存系统的状态，以便在系统发生故障或需要进行恢复时能够从先前的状态中恢复。

一个无状态的服务器通常不需要进行检查点。"无状态"意味着服务器在处理请求时不依赖于先前的请求状态。每个请求被视为独立的，服务器不会保留关于请求处理状态的信息。