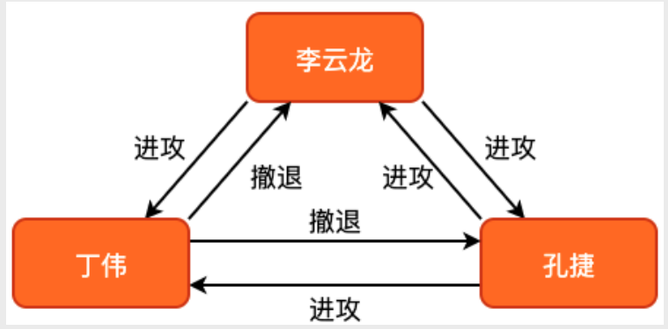
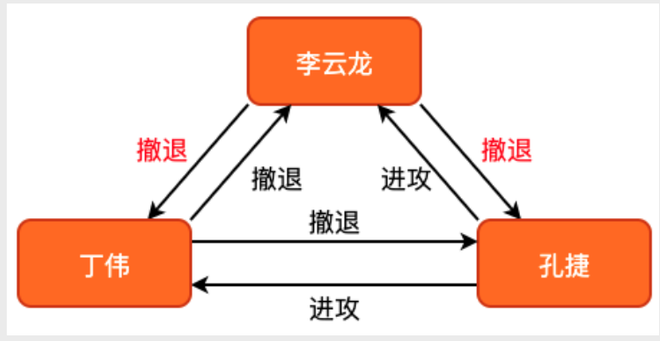
拜占庭将军问题

问题

- 拜占庭是一个古希腊城市，拜占庭帝国其实就是东罗马帝国，是中古欧洲历史最长久的专制君主制国家。至于为什么 Lamport 喜欢使用古希腊的城市、岛屿名来作为论文的一部分，就不得而知了，大佬的世界就是这么的跳脱
- 拜占庭将军问题（The Byzantine Generals Problem）其实是借助拜占庭将军的故事来展现分布式共识的问题，并且还讨论了如何进行解决这个共识问题
- 什么是共识？
  - 我们在新闻上能经常听到这样的报道，我国与 XXX 就 XXX 方面达成了共识，共识其实就是对某一个事情的看法达成一致。比如我和小明相约周末一起去钓鱼，小明同意了，那么我和小明就在钓鱼这件事情上达成了共识
- 我们以李云龙、孔捷和丁伟攻打平安县城为例，将拜占庭将军问题"本土"化，并且还有一丝丝恶搞的趣味.....

描述

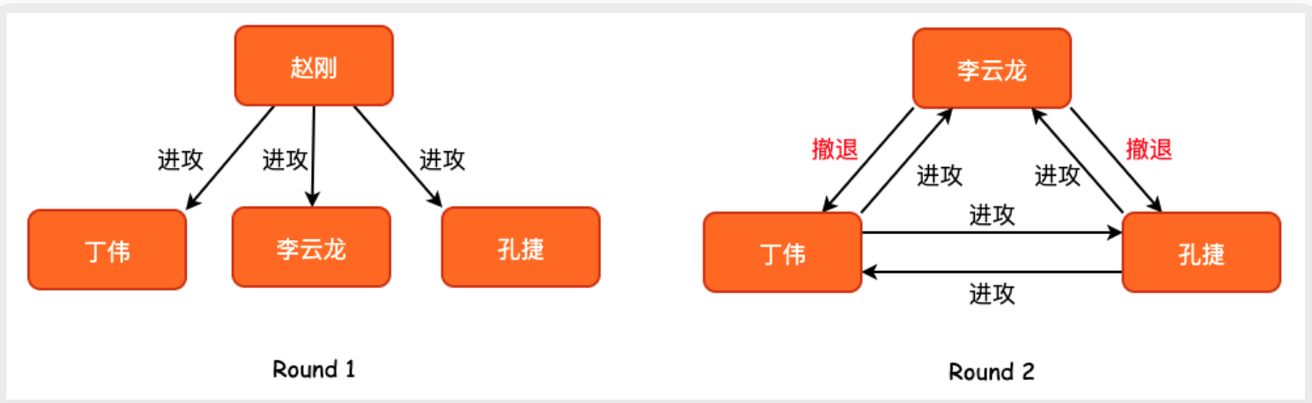
- 背景
  - 李云龙、孔捷和丁伟一起攻打平安县城，并且李云龙没有意大利炮。同时，必须有两只部队一起攻打平安县城，才能成功
  - 李云龙、孔捷和丁伟的部队分别驻扎在平安县城的 3 个方向，所以部队之间只能通过信使相互联系。并且，信使在送信的路上可能被杀，也可能被敌人的间谍渗透，传递假消息扰乱正常的作战计划
  - 3 位指挥官分别向其它两名指挥官传递进攻或者是撤退的消息，指挥官收到消息后，按照 "少数服从多数" 的方式来进行决策
  - 

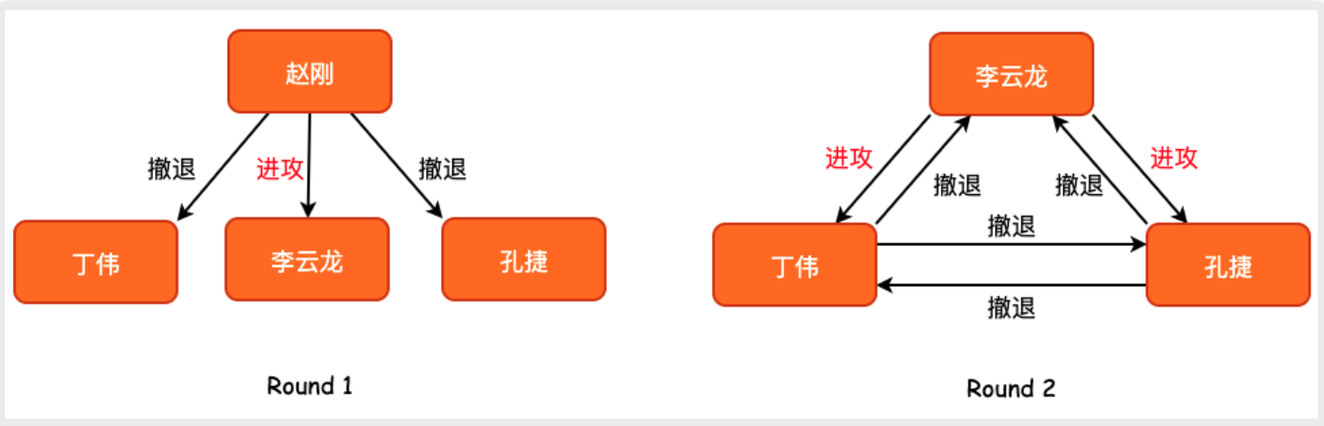
  - 如左图所示，每一位指挥官最终得到的是进攻、进攻和撤退，所以会同时出兵攻打平安县城，最终取得胜利
- 在正常情况下，上面的方式能够有效的组织进攻/撤退计划，3 只部队要么一起进攻，要么一起撤退。但是，假如说李云龙的信使被敌军渗透了，向丁伟和孔捷传递了错误的消息，又会出现什么情况呢？
  - 

  - 被敌军渗透的信使向丁伟和孔捷传递了错误的撤退信息，导致李云龙认为有 2 票进攻、 1 票撤退，而丁伟和孔捷均有 2 票撤退 和 1 票进攻
  - 这就会导致只有李云龙的一只部队前去攻打平安县城，导致伤亡惨重
- 实际上这就是"两忠一叛"问题，在 3 位指挥官中如果出现了 1 个叛徒，那么这 3 个人是无法达成共识的，叛变的那个人总是能够想办法干扰到最终的决定

解决

口信消息型解决方案

- 在前面的讨论中我们已经明确了"两忠一叛"问题无解，必须增加忠诚将领的人数才能达成一致。因此，我们再增加一只部队，指挥官为赵刚
- 在口信消息型解决方案中，首先发送消息的指挥官称为施令官，其余指挥官为副官。对于3忠1叛的场景需要进行两轮作战信息协商，如果没有收到作战信息那么默认撤退
- 流程
  1. 在第一轮讨论中，我们随机地选取一个施令官，不管这是施令官是忠诚的还是叛徒，然后这个施令官向其它指挥官下达命令
  2. 除了第一轮的施令官以外，剩余的 3 位指挥官分别向另外两名指挥官发送作战信息，而这个信息其实就是施令官在第一轮告诉当前指挥官的。最后，3 位指挥官按照 "少数服从多数"，执行相应的作战计划
- 信使被渗透
  - 

  - 此时，即使是李云龙的信使向其它指挥官发送了错误的信息，最终的票数还是决定了丁伟和孔捷将会发起攻击，3 位指挥官达成共识
- 施令官被渗透
  - 假设赵刚被敌军渗透了，向李云龙下达了错误的进攻信息，但是在第二轮信息共享时，由于丁伟和孔捷收到的都是撤退信息，最终李云龙部队也不会向平安县城发动总攻，3 名指挥官仍然能达成一致
  - 

- Lamport 在论文中论证了假设有 m 名叛军，那么将军总人数不能少于 3m + 1，否则无法使用口信消息型解决方案来解决拜占庭将军问题

签名消息型解决方案

- 这种解决方式就比较简单了，采用对消息加密且篡改后会被发现的方式来进行消息传递。那么一旦指挥官接收到了被篡改的信息，将会直接忽略掉此信息。若进攻和撤退的票数相同的话，只需要约定一下在此情况下是进攻还是撤退即可

总结

- 拜占庭将军问题提供了对分布式共识问题的一种情景化描述，并且描述的是存在恶意篡改节点的情况下满足分布式共识，现有的分布式协议和算法基于拜占庭将军问题可以分为两类: 拜占庭容错算法和非拜占庭容错算法
- 1. 拜占庭容错算法
  - 可以解决分布式系统中既存在故障，又存在恶意攻击场景下的共识问题，最为经典的应用就是区块链了，常用的算法有 PoW 算法
- 2. 非拜占庭容错算法
  - 又称之为故障容错算法，解决的是分布式系统中存在故障，但不存在恶意攻击的场景下的共识问题。也就是说，在该场景下可能存在消息丢失，消息重复，但不存在消息被篡改或伪造的场景
  - 故障容错算法多用于分布式数据库中，比如常见的 Paxos、Raft 以及 ZAB 协议等