数字货币和区块链

- 共识与区块链

山东大学网络空间安全学院

钱宸 2023年11月29日

区块链与共识协议

- 简介与历史
- 拜占庭广播与Dolev-Strong协议
- 拜占庭广播（无签名）
- 区块链
- 区块链协议

区块链

- 目前为止，我们所遇到的共识协议都是一次性的。
- 定义可重复滚动使用的共识协议。
  - 区块链
  - 分布式系统中，可复制状态机

扩展定义

- 目前为止，所有算法都是强同时性：
  - 下一轮开始之前，所有信息都会传递给目标用户
  - Δ同时性：第n轮发出的信息，至少在第$n+\Delta$结束之前能够被接收到。
  - Dolve-Strong，无签名的拜占庭共识协议：Δ=0的特例

区块链

- 目标：一系列分布式节点计划对一个永远增长的，线性排序的交易记录产生共识
- 区块链：
  - 块：通常会采用分块的方式，来增加输入输出率。
  - 链：线性增长的交易记录
- 所需要满足的性质：
  - 一致性
  - 活跃性

区块链

- 一致性：
  - 所有诚实用户的账本要保持一致
  - 这里因为存在延迟，所以要求每个诚实用户的记录都互为前缀
- 活跃性
  - 如果某个诚实的用户在第t轮收到了某个记录，则所有用户会在第$t + T_{conf}$轮的账本上收到这个记录

区块链 - 严格定义

- 我们用Log_i^r来表示用户i在第r轮的记录，Log_i^r中的记录不能被更改，只能增加
- 一致性：
  - 对于所有的诚实用户i, j，以及任意轮数r, t满足$\mathsf{Log}_i^r \preceq \mathsf{Log}_j^t$或者$\mathsf{Log}_j^t \preceq \mathsf{Log}_i^r$。其中$\mathsf{Log} \preceq \mathsf{Log}'$表示$\mathsf{Log}$是$\mathsf{Log}'$的前缀。
- T_{conf}-活跃性
  - 如果某个诚实的用户在第t轮收到了某个记录tx，则所有用户会在第$t + T_{conf}$轮的账本上收到这个tx。

从共识协议到区块链

- 假设一共有n个不同的节点，假设每一个拜占庭协议会用R轮完成
- 区块链构造：
  - 在每个kR轮的时候，产生一个拜占庭共识协议，其中使$L_k := (k \mod n)$作为该拜占庭共识协议的消息发送者。
    - 为了方便，我们将在第kR轮产生的拜占庭共识标记为BB_k
    - L_k收集到目前为止已接收到但是没有计入账本的记录，即$tx \notin \mathsf{Log}_{L_k}^{kR}$
    - 将所有这类输入放入拜占庭共识协议当中
  - 在$BB_0, BB_1, \ldots, BB_k$结束的时候，账本记录为$m_0 \| m_1 \| \ldots \| m_k$

从共识协议到区块链

- 假设拜占庭协议满足n个用户中有f个叛徒的情况
- 且拜占庭协议的轮复杂度为R
- 则上述的区块链满足
  - 一致性
  - 活跃性
    - 确认时间是多少?

区块链 - 一些简单的讨论

- 优点：
  - 从拜占庭协议我们可以直接构造区块链系统
- 缺点：
  - 直接堆叠通常不是最高效的做法

简单的区块链示例 - Streamlet

- Streamlet
  - 2020年由Chan和Shi提出 [AFT’ 20]
  - 将已有的Paxos，PBFT，RAFT等进行整合，并给出了一个较为简单，教学性质的区块链系统
  - 可以容忍<n/3的攻击者存在

Streamlet

- 提议投票
  - 该轮的指定信息发布者提出一个从当前已记录的账本中增长信息块的请求
  - 每一个节点，检查请求是否是对自己看到的最长的账本的一个增长，对提出的请求进行投票
  - 当一个增长请求获得至少(2/3)n个投票时，则承认该节点
- 结算
  - 将结果计入账本中

Streamlet - 严格定义

- 轮数：
  - 协议按照每一轮来进行，每一轮持续1秒
  - 每个节点从同样的时间点开始第一轮，然后每隔一秒进入下一轮
- 轮领导：
  - 假设n个不同的计算节点标记为1,2,…,n，在第e轮选出H(e)作为这轮的轮领导，其中H是一个哈希函数。在证明的时候被模拟为随机寓言机

Streamlet - 严格定义

- 块与区块链：
  - 块：每一块都是(h,e,txs)的形式，其中
    - H：父哈希，通常是链前缀的哈希值
    - e：块轮数，记录这一区块是何时被记录的
    - tx：负载字符，记录一串交易记录
    - 此外：有一个特殊的块为$(\bot,0,\bot)$为每个区块链的初始区块

Streamlet - 严格定义

- 区块链：
  - 一个区块链chain，是由一串块组成的顺序集合，并且第一块为初始块 chain$[0] := (\perp, 0, \perp)$。
  - 区块链是成立的，对于$l > 0$有：
    - chain$[l] = (h, e, tx)$，其中
    - e大于任意chain$[l']$中的记录轮数，这里$l' < l$
    - 另外，h= H(chain[0], …, chain[l-1])
    - （思考：这里可以优化么？）

Streamlet - 严格定义

- 投票与公证
  - 一个投票就是对某一个区块进行签名
  - 至少（2/3）n个对于同一个区块的签名被称作一个区块的公证
  - 一个区块链被称为被公证的，如果其中的每一个区块都被公证了

Streamlet

- 假设前提：每一个诚实节点在接收到一个交易或者信息以后会发给其余所有节点。
- 仍然使用$\langle m \rangle_i$来表示(m, σ)，其中σ是用户i对信息m的签名

Streamlet 协议

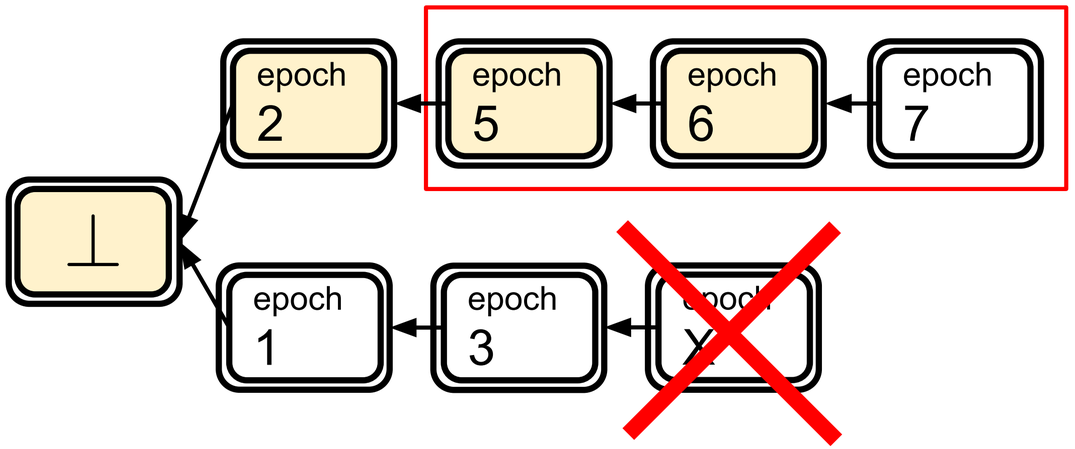
- 对于第e轮，e = 1,2,…
- 提议：
  - 在第e轮开始的时候，轮领导L选择自己见到公证过的最长的链chain
  - 计算h = H(chain)
  - 收集所有未经记录的交易tx
  - 将$\langle (h, e, tx) \rangle_L$进行广播

Streamlet 协议

- 投票：
  - 每一个节点i收到$\langle (h, e, tx) \rangle_L$以后，检查收否是一个正确的区块
  - 如果$\langle (h, e, tx) \rangle_L$是目前见过公证的区块链的最长的一个区块链的延伸，则投票。即将$\langle (h, e, tx) \rangle_i$发送给所有节点。
- 结算：
  - 如果看到三个连续的被公证过的节点，则将第二个节点以及之前的所有都进行记账。

一些简单的讨论

- 整个协议遵循提议-投票的模式，结算方式比较奇特
- 给出下列的例子
  - $1 \rightarrow 3 \rightarrow ..$
  - $2 \rightarrow 5 \rightarrow 6 \rightarrow 7$



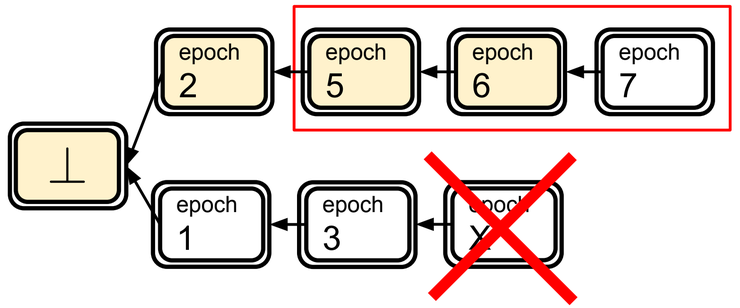
Streamlet - 一致性

- 证明：
  - streamlet满足一致性要求：
    - 无论网络延迟多严重

Streamlet - 一致性

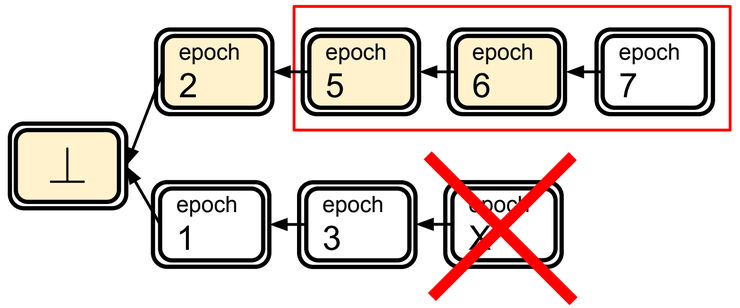
- 引理1：每轮只公证一个区块
- 证明：
  - 利用反证法进行证明

Streamlet - 一致性



- 根据引理1，不存在被公证的区块与区块6在同一高度
- 反证法，考虑如下两种情况：
- X<5:
  - 因为X被公证了，说明至少有n/3个诚实用户（记作S）公证了X，并且他们在公证X之前也公证了区块3。所以S是不会公证区块5的，因为它不是最长的一条区块链。
  - 因此，攻击者数量小于n/3使得区块5永远不会被诚实用户公证

Streamlet - 一致性



- 考虑第二种情况X>7:
  - 如果区块7被公证了，说明至少n/3个诚实的用户同时公证了区块6
  - 因此在时间X，这n/3个诚实的用户肯定公证了区块6，不会再公证区块X。因为此时，3并不是最长的链。
- 一致性：
  - 如果一个诚实节点看见一个公证过的链，其中包含三个相邻的区块B_1, B_2, B_3且他们的轮数相邻，则不可能存在一个矛盾的区块$B \neq B_2$被公证了且与B_2在同样的高度。