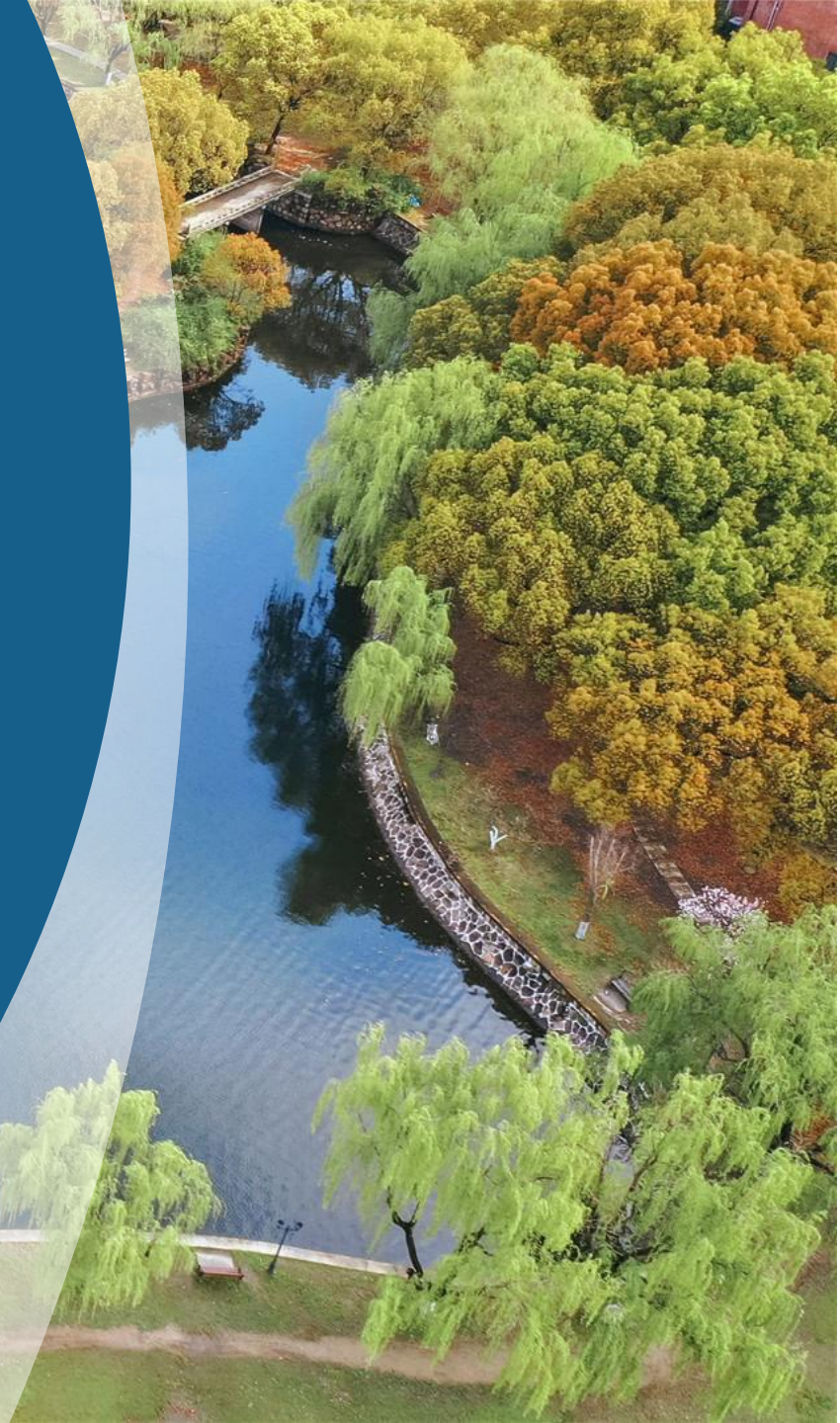
上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

异步共识协议

2022.04.08



饮水思源 · 爱国荣校

目 录

01

异步共识的特点和优势

共识协议的时间假设

- **同步（Synchrony）：** 网络中的所有消息能够在一个已知的时间间隔 Δ 内到达。
- **半同步（Partial Synchrony）：** 在一个未知的 GST（Global Stabilization Time）事件后，系统会趋于同步状态，消息在时间间隔 Δ 内到达。（PBFT、HotStuff、Raft）
- **异步（Asynchrony）：** 消息可以按照任意的顺序和任意的延迟传输，唯一需要保证的是正确节点发送的消息最终一定会被其他节点接收，但何时收到、以怎样的顺序收到没有任何预估。

FLP 不可能定理

在**纯异步**通信场景下，即使只有一个节点**发生错误**，也没有任何**确定性**的算法能保证协议的正确性和活性。

在保证容错共识协议活性和安全性的前提下，根据FLP定理的限制，可以放宽时间假设或者引入随机性共识。

robustness VS responsiveness

对网络延迟 Δ 设置过大

当系统真正出现故障时，需要等待过长的时间才能发现，导致系统大部分时间处于闲置状态，没有充分利用计算和带宽资源，影响系统的响应能力。

对网络延迟 Δ 设置过小

当网络状态出现小波动时就可能触发超时机制，导致系统一直处于排错状态而变得不可用，使系统过于依赖网络的稳定性。

PBFT 的失活情况

Replicas

Time	0 (faulty)	1	2	3
0Δ	●Req* ○PP_0 view:0	●Req* view:0	●Req* view:0	●Req* view:0
1Δ	○V_1 ●V_1* view:1	○V_1 ●PP_0X view:1	○V_1 ●PP_0X ●V_1* view:1	○V_1 ●PP_0X ●V_1* view:1
3Δ	○V_2 ●N_1,PP_1X ●V_2* view:2	●V_1 ○N_1,PP_1* ●V_2** ○V_2 view:1/2	○V_2 ●N_1,PP_1X view:2	○V_2 ●N_1,PP_1X ●V_2* view:2
7Δ	○V_3 ●N_2,PP_2X ●V_3* view:3	○V_3 ●N_2,PP_2X ●V_3* view:3	●V_2 ○N_2,PP_2* ●V_3** ○V_3 view:2/3	○V_3 ●N_2,PP_2X view:3

间歇性网络控制器

○ 表示节点发送的消息

● 表示节点收到的消息

× 表示该消息因为轮次滞后被忽略

* 表示该节点启动超时计时器

** 表示该节点增加轮次

粉红色区域 表示该节点的网络延迟大于 Δ

异步共识的优势

异步共识协议对网络传输延迟没有任何要求，非常适合用在网络通道不稳定或者缺乏时钟同步机制的环境中，尤其是在跨国甚至跨洲的广域网环境，网络波动非常大，如果使用同步或者半同步协议，往往会频繁地触发超时机制，让协议变得更复杂甚至失去可用性。

异步共识的随机性

根据FLP定理，异步共识没有时间假设，但为了保证活性需要引入随机源，即当协议无法达成共识时，随机选择一个结果作为最终的输出。

- ✓ 所有节点共同协商决定统一的状态，对输出值达成一致。
- ✓ 所有节点对输出正确值的概率达成一致。

异步共识的随机性

收到 2f 个 1

超时机制

大家一致进入下一状态
换主节点重新提案

收到 2f 个 0

随机抛硬币

大家一致按照硬币的结果
作为共识的结果

02

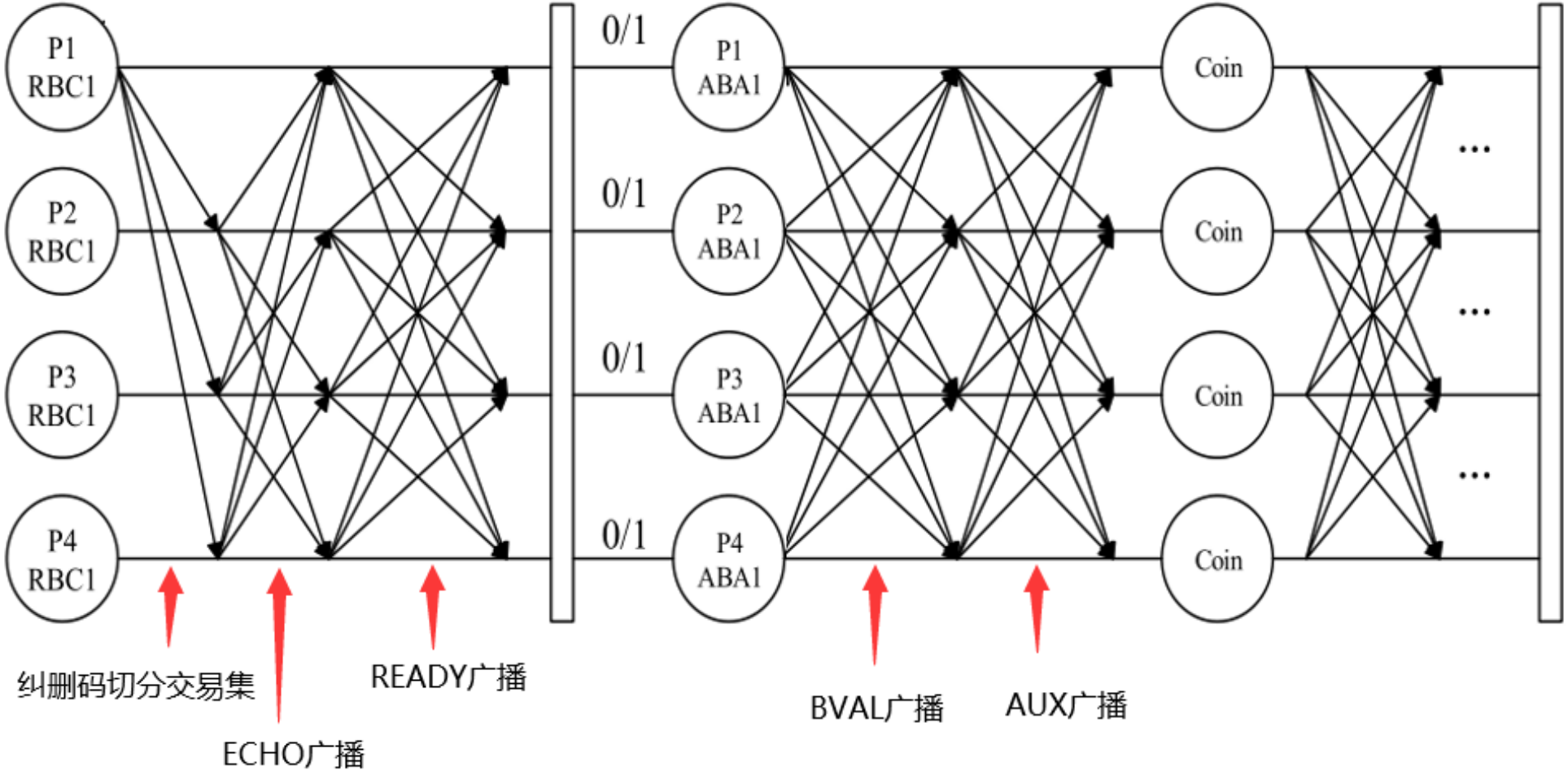
Honey Badger BFT

Honey Badger BFT

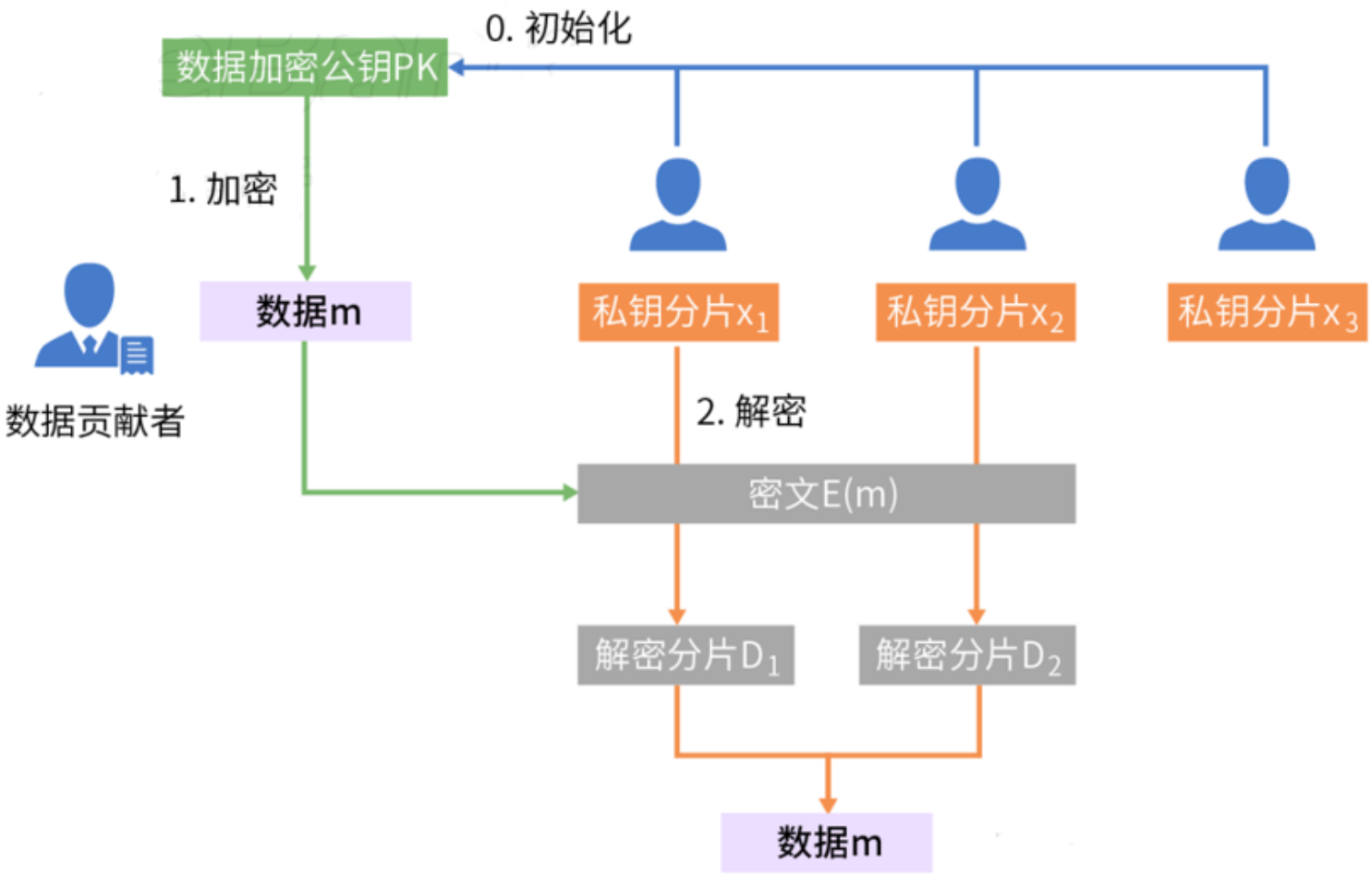
HB-BFT 是第一个实用的异步拜占庭容错共识协议，它不依赖于任何关于网络环境的时间假设。它同时也是一个多主协议，每个节点都能发起提案。

HB-BFT 的核心是异步公共子集协议，用来决定每个轮次中哪些节点的请求能被包含在最终的共识区块里。

HB-BFT 中异步公共子集（ACS）流程



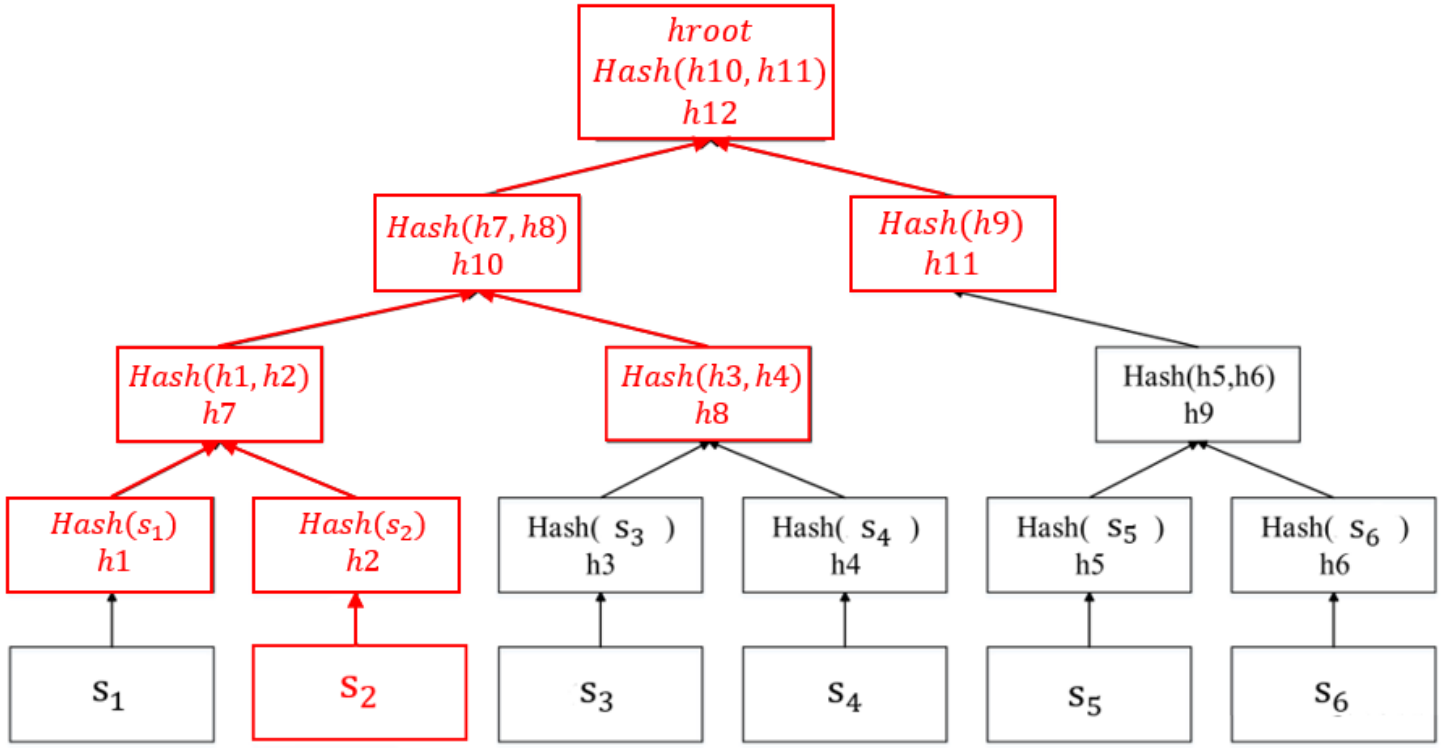
门限加密



门限加密本质上是一种非对称加密算法，它在初始化阶段生成一个公钥和 N 个私钥分片，对应 N 个参与共识的节点。数据用公钥加密后，在解密阶段，只要收集到一定门限数量的解密分片，该节点就可以恢复出密文。

门限加密在BFT类协议中使用非常广泛，它可以避免敌手提前知道关键信息从而阻挠共识进行。

默克尔树



默克尔树（哈希树）是一棵能够存储哈希值的树，树的叶子节点是数据块的哈希值，再知道它到根节点路径上其他子树的哈希值，即可得到根节点的哈希值。

默克尔树可以用来验证数据在经过传输后没有损坏，也没有改变。

可靠广播（Reliable Broadcast，RBC）

可靠广播用来把一个节点的提案广播给所有节点，针对信息发送者的情况，它保证：

- ✓ 如果发送者P是正确节点，并且广播 **v**，则所有正确节点都会收到 **v**。
- ✓ 如果发送者P是错误节点，则所有正确节点输出相同的值或者不接受任何来自P的提案。

- 节点P有自己的初始输入 **v**，广播 **VAL(v)** 消息给所有节点
- 其他节点收到来自P的 **VAL(v)** 消息，则广播 **ECHO(v)** 消息
- 当节点收到 **2f+1** 个 **ECHO(v)** 消息后，广播 **REDAY(v)** 消息
- 当节点收到 **2f+1** 个 **READY(v)** 消息后，接受 **v** 作为可靠广播的输出

异步二元共识（Asynchronous Binary Agreement，ABA）

异步二元共识用来让所有节点对0或1达成共识，每个节点的输入值任意（0或1），经过有限轮次的信息交互后，所有节点有得到相同的输出值。异步二元共识保证了以下三个特性：

- **一致性**：如果一个正确节点输出b，则所有正确节点都会输出b。
- **有效性**：如果一个正确节点输出b，则至少有一个正确节点输入b。
- **活性**：如果所有正确节点都有输入值，则所有正确节点都会有输出值。

异步二元共识（Asynchronous Binary Agreement，ABA）

- 节点P有自己的初始输入 **v**，并且维护一个可能的共识结果集合 **bin_value**
- 节点P广播 **BVAL(v)** 消息
- 当节点收到 **f+1** 个 **BVAL(b)** 消息后，如果自己还未广播过 **BVAL(b)** 则广播 **BVAL(b)** 消息
- 当节点收到 **2f+1** 个 **BVAL(b)** 消息后，**bin_value = bin_value∪{b}**
- 当 **bin_value** 不为空集合时广播 **AUX(w)** 消息，**w∈bin_value**
- 当节点收到 消息后，所有 **x** 组成集合 **vals**，如果 **vals** 是 **bin_value** 的子集，则**2f+1** 个 **AUX(x)**带着 **vals** 进入硬币比对环节，否则继续等待 **AUX** 消息或者 **BVAL** 消息来改变这两个集合中的值

二元共识的硬币比对

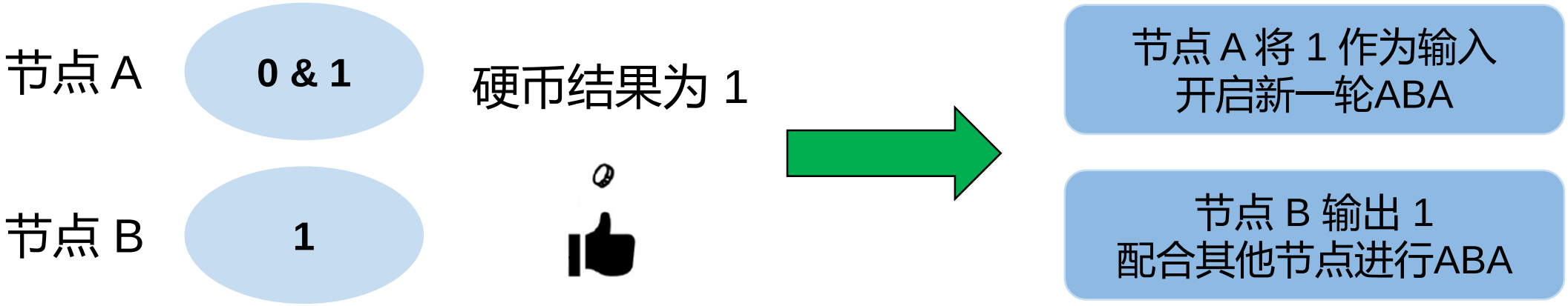
✓ **所有正确节点的 vals 集合不可能出现互为补集的情况。**

- 对于 **vals** 集合中只有一个值 **b** 的情况：
  1. 如果和硬币结果相同则输出 **b**。
  2. 如果和硬币结果不同则将 **b** 作为输入开启新一轮的 **ABA** 协议。
- 对于 **vals** 集合中有两个值的情况：
  1. 将硬币的结果作为输入开启新一轮的 **ABA** 协议。

✓ 经过常数轮次后所有正确节点都会输出相同的值。

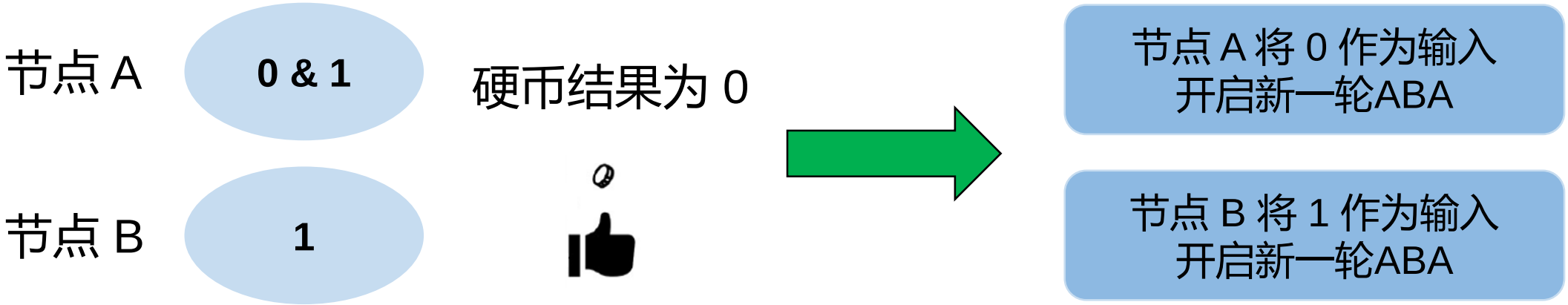
二元共识的硬币比对

- 对于 **vals** 集合中只有一个值 **b** 的情况：
  1. 如果和硬币结果相同则输出 **b**。
  2. 如果和硬币结果不同则将 **b** 作为输入开启新一轮的 **ABA** 协议。
- 对于 **vals** 集合中有两个值的情况：
  1. 将硬币的结果作为输入开启新一轮的 **ABA** 协议。

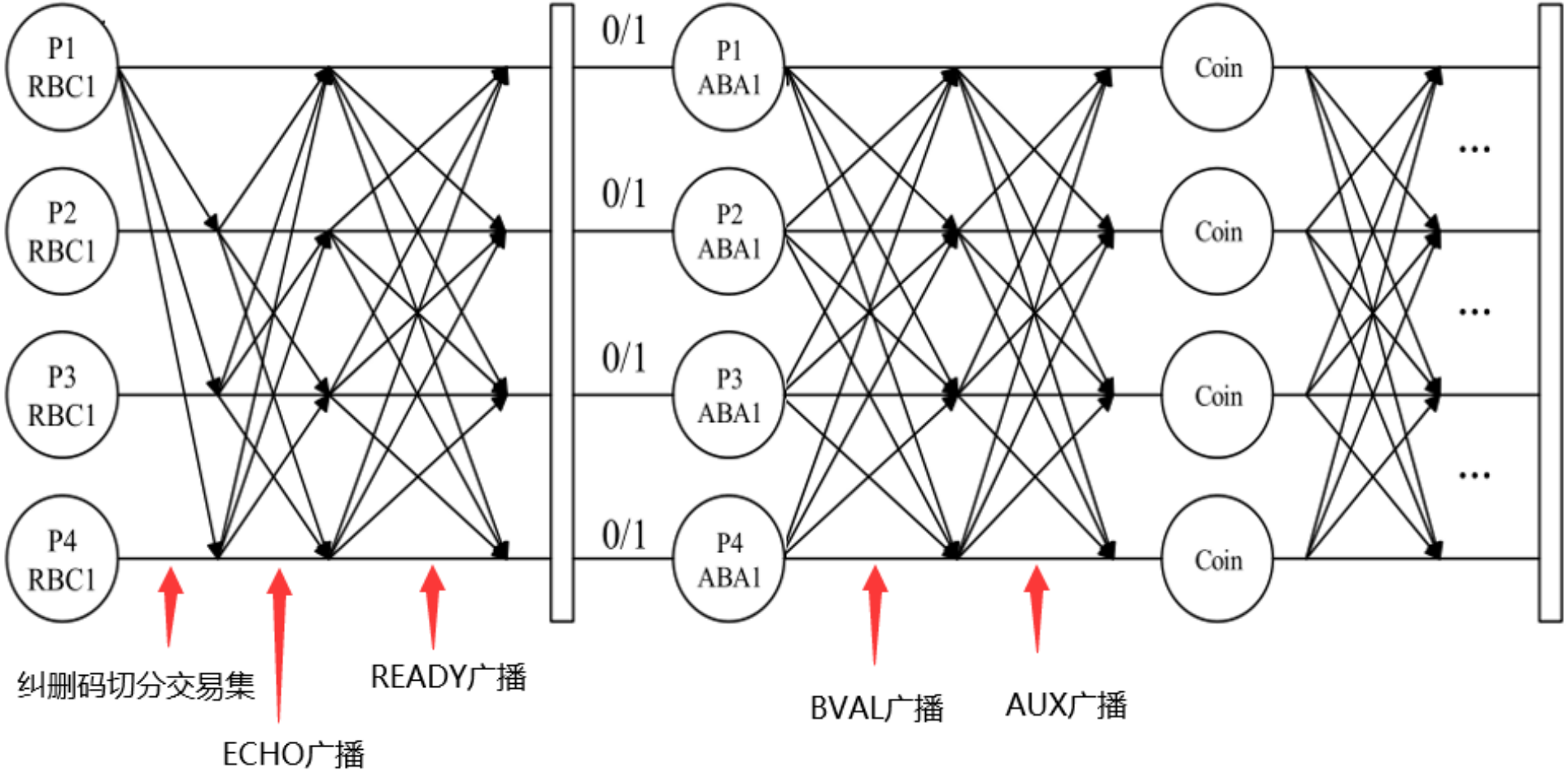


二元共识的硬币比对

- 对于 **vals** 集合中只有一个值 **b** 的情况：
  1. 如果和硬币结果相同则输出 **b**。
  2. 如果和硬币结果不同则将 **b** 作为输入开启新一轮的 **ABA** 协议。
- 对于 **vals** 集合中有两个值的情况：
  1. 将硬币的结果作为输入开启新一轮的 **ABA** 协议。



HB-BFT 中异步公共子集（ACS）流程



Honey Badger BFT 优点

HB-BFT 提出的 ACS 方法非常具有创新性和工程可用性，同时在设计上HB-BFT用纠删码技术降低了广播原始交易的通信复杂度，利用分割原始请求数据的方式来避免单提案节点的带宽瓶颈问题，突破性地提高了异步协议的可用性。

实验表明，当节点数量增大后 HB-BFT相比于 PBFT有更高的吞吐；并且 HB-BFT能轻易扩展到广域网百节点的大规模环境。在保证活性和安全性的前提下，HB-BFT已经可以支持大量的实际应用场景。

Honey Badger BFT 不足

每个节点在每轮ACS中都要运行N个RBC+ABA实例，每个RBC和ABA分别需要进行三轮多对多的信息交互，如果最后抛硬币的结果和投票结果不一致，协议将进入新一轮的ABA，再额外进行三轮通信，直到硬币结果理想。引入随机源虽然解决了异步共识的活性问题，但也带来了协议轮次的不确定性，可能让延迟显著增加，影响系统整体的性能。

03

异步共识协议的研究方向

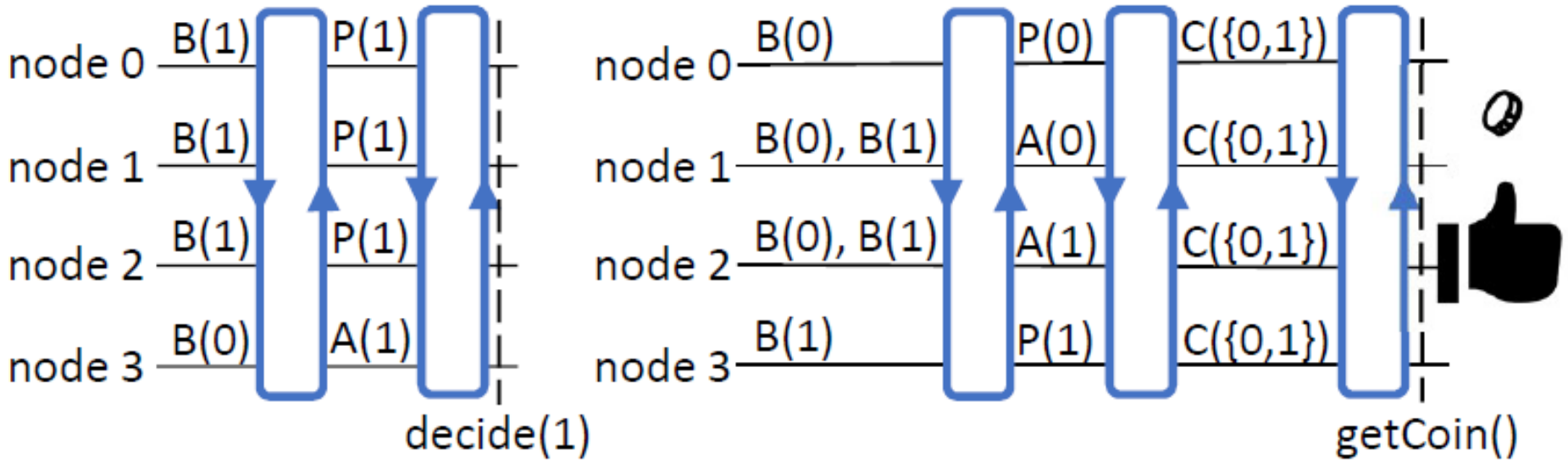
异步共识领域的挑战

- **设计快速异步共识：**在保证协议活性和安全性的前提下，当投票结果满足某些理想条件时跳过随机过程直接完成共识，只有当投票出现分歧时再通过随机源决定结果。
- **节点公平性问题：**在异步协议中由于没有超时的限制，网络较慢节点的提议和投票可能永远无法被采用。
- **节点扩展性问题：**BFT类协议一般需要相对封闭的初始设定，对节点身份和节点规模都有很高的要求，应用场景有限。

如何避免抛硬币

- 如果一个节点收到 N 个 **BVAL(b)** 消息。
- 确保其他节点不会收到 **2f+1** 个相反的 **BVAL** 投票。
- 在 **BVAL** 消息交互过程中不进行放大过程，即承诺不再投相反票。
- 如果正确节点收到 **2f+1** 个 **BVAL(b)** 消息，并且这些消息经过**数字签名**，则该节点直接将这些消息打包转发给其他节点，作为自己承诺不再投相反票的证明。
- 如果自己已经投过相反票，则说明在自己视角节点间投票存在巨大分歧，需要引入硬币来决定结果。

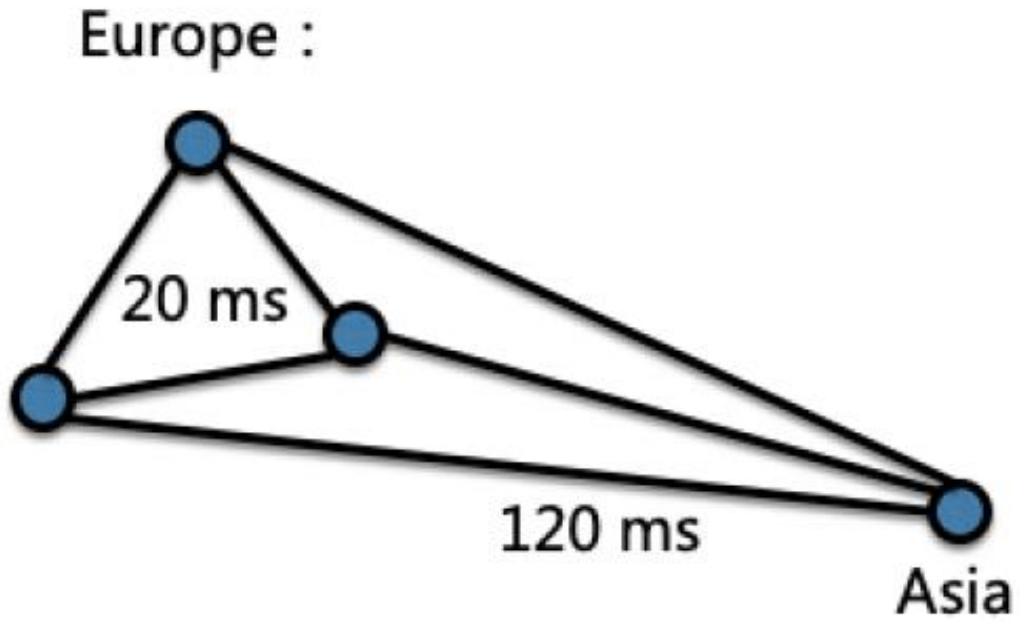
快速异步共识



(a) fast path

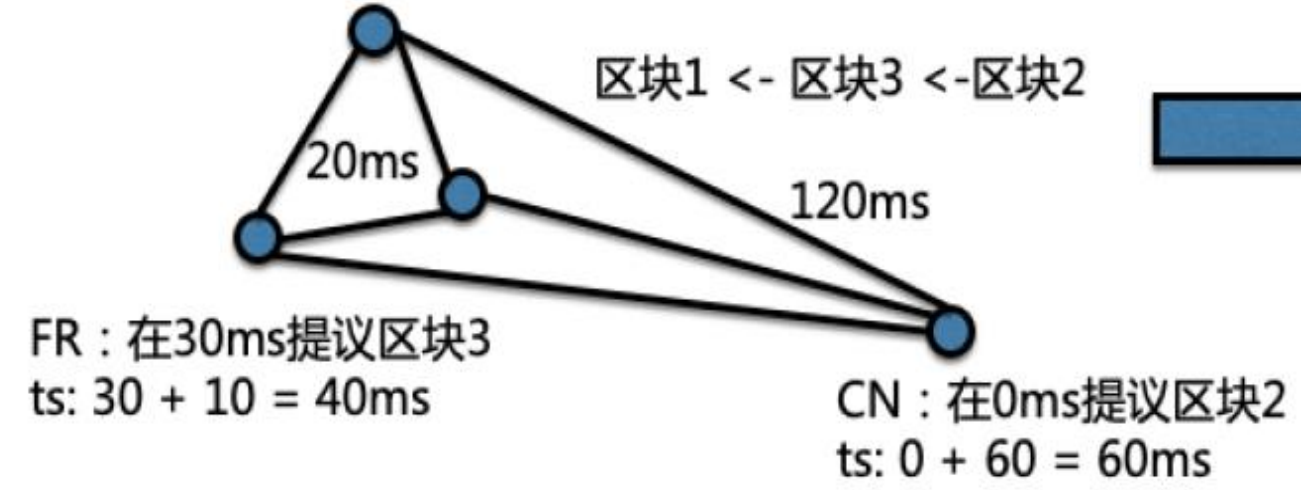
(b) normal path

异构环境下的节点公平性



当 ACS 开始后，图中三个在欧洲的节点通信延迟较小，并且相互投票达到 quorum，从而能快速完成 RBC+ABA；而亚洲节点到欧洲节点的延迟较大，会拖慢区块的共识过程，或者直接被剥夺区块的构造权（被一直共识为0）。

利用时间戳灵活定序



UK节点：
30ms区块1完成共识（ts: 10ms)
60ms区块3完成共识（ts: 40ms)
80ms区块2完成共识（ts: 60ms)

FR节点：
30ms区块1完成共识（ts: 10ms)
60ms区块3完成共识（ts: 40ms)
80ms区块2完成共识（ts: 60ms)

CN节点：
80ms区块1完成共识（ts: 10ms)
110ms区块3完成共识（ts: 40ms)
130ms区块2完成共识（ts: 60ms)

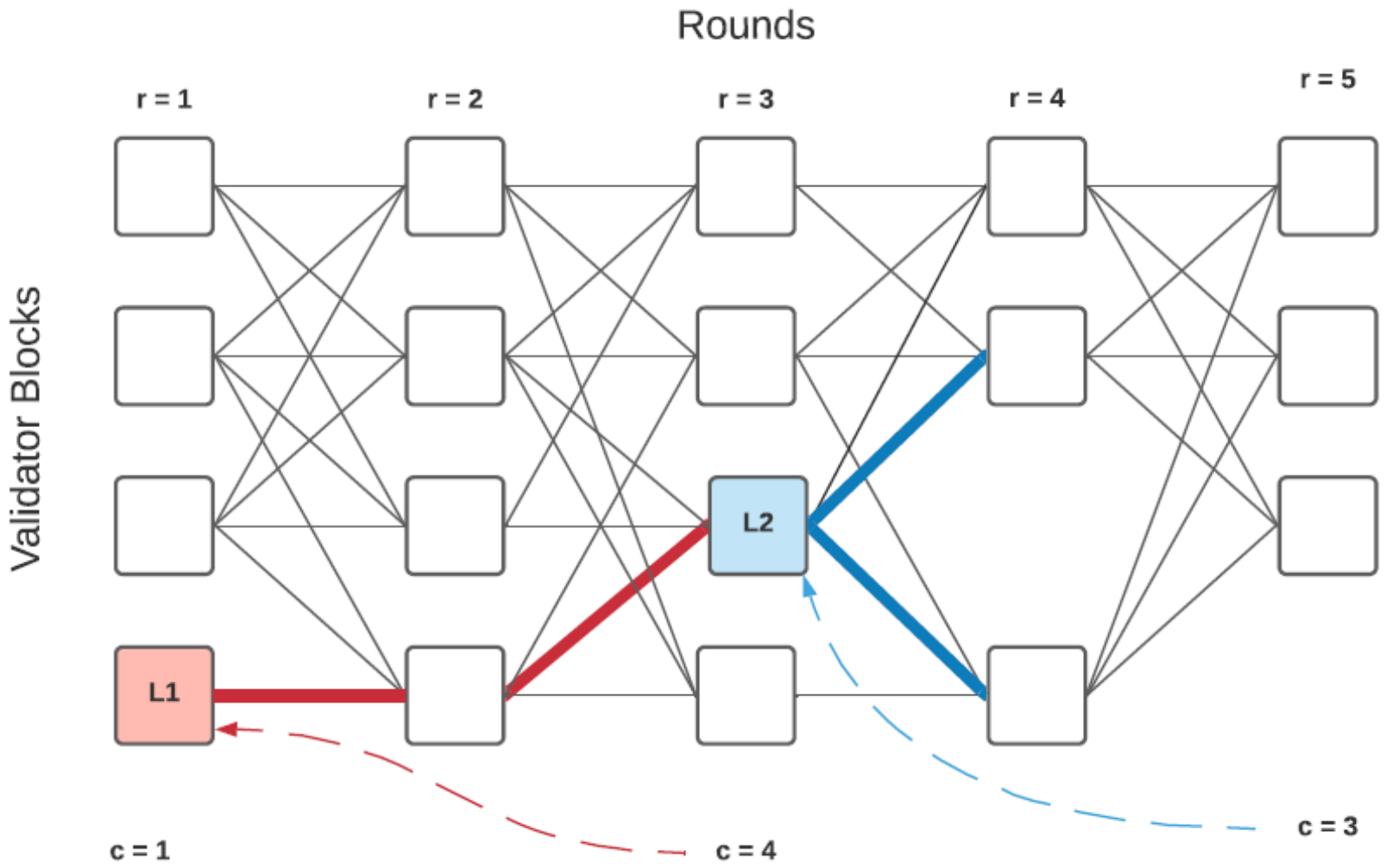
区块的时间戳顺序与三轮共识的完成顺序一致

有向无环图（DAG）在共识领域的应用

在传统的区块链中，数据结构是链式的，区块串行生成，每个区块链接单一的前序区块，任意一个被确定的区块到创世块的路径也是唯一固定的，这在一定程度上限制了系统的吞吐。

在DAG结构中，每个节点可以链接多个前序，到根节点路径上的节点集合是固定的，一个节点被确认代表着所有通向根节点路径上的节点被确认。这使得DAG结构可以支持异步并发地写入很多交易，相比单链结构提高了系统的吞吐量和可扩展性。

基于DAG的异步共识协议



Q & A