网络安全技术

刘振

上海交通大学 计算机科学与工程系

电信群楼3-509

liuzhen@sjtu.edu.cn

比特币的共识机制

- 共识基本概念、共识技术发展路线及分类
- 比特币的共识机制

共识基本概念、共识技术发展路线及分类

共识的基本定义

- 在一个分布式系统中，为了使得整个系统正常工作，一个**久远而又核心的问题**就是**如何保证系统中所有节点的数据完全相同并且能够对发起的提案达成一致**
- 共识算法就是用来解决上述问题，从而保证分布式系统一致性的方法
- **共识定义**:

-- **终止性(Termination)**：所有正常运作的进程（节点）最终会在有限步数中结束并做出决定, 算法不会无尽执行下去

-- **一致性**: 所有进程必须做出相同的决定(**意见一致 Agreement**)；如果所有进程都提议相同的初始决定值，那么所有正确进程都应选择该值(**行为统一Integrity**)

-- **有效性(Validity)**: 最终达成一致的决定必须是其他进程提交值中的某一个

FLP impossibility定理

- **FLP Impossibility**（FLP不可能性）是分布式领域中一个非常著名的定理，该定理由Fischer, Lynch, Patterson于1985年提出，相关论文获得了Dijkstra奖。

- FLP不可能定理表明：在异步通信场景（没有时钟、不能时间同步、不能使用超时、不能探测失败、消息可任意延迟、消息可乱序），即使只有一个进程失败，没有任何算法能保证非失败进程达到一致性！

CAP定理

- 分布式系统的CAP理论把分布式系统中的三个特性进行了如下归纳：

-- **一致性（Consistency）**：在分布式系统中的所有数据备份，在同一时刻是否同样的值（等同于所有节点访问同一份最新的数据副本）

-- **可用性（Availability）**：在数据库集群中一部分节点故障后，集群整体是否还能响应客户端的读写请求（对数据更新具备高可用性）

-- **分区容错性（Partition tolerance）**：以实际效果而言，分区相当于对通信的时限要求。系统如果不能在时限内达成数据一致性，就意味着发生了分区的情况，必须就当前操作在C和A之间做出选择

- CAP理论表明：**在分布式存储系统中，最多只能实现上面三个特性中的两点**。

- 由于当前的网络硬件肯定会出现延迟丢包等问题，所以分区容错性是我们必须需要实现的

分布式系统的“熵值”

- “熵值” --- 不确定性，可视为一致性的对立面
- 一般地，去中心化的大型分布式系统具有较大的“熵值”
- 如何大幅降低去中心化的分布式系统的“熵值”：

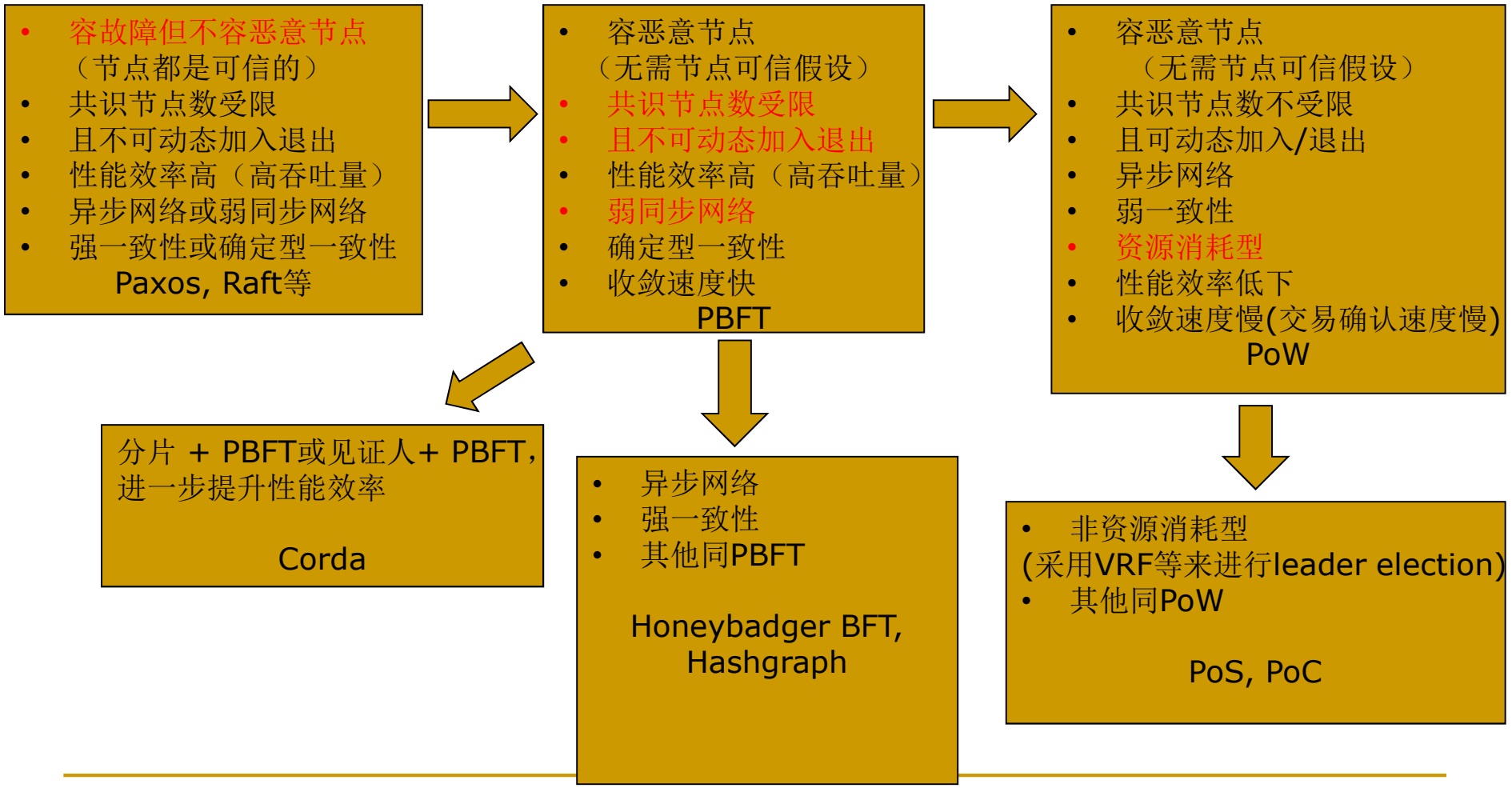
-- **通过求解数学（或密码学）困难问题来随机选取leader候选人（由其来提出proposal）**可以大大降低系统的“熵值”

-- **经济机制同样有可能作为减少熵值的工具**，因为经济主体可以被激励引导 — 也就是说，被引导着更有可能做出特定的行为

共识算法系统模型需考虑的因素

- 一致性/弱一致性，或是说确定型一致性/概率型一致性
- 可用性（活性）
- 异步网络/同步或弱同步网络
- 分区容忍（容故障）
- 拜占庭容错（容恶意行为）

共识技术的发展与分类


• 容故障但不容恶意节点
（节点都是可信的）
• 共识节点数受限
• 且不可动态加入退出
• 性能效率高（高吞吐量）
• 异步网络或弱同步网络
• 强一致性或确定型一致性
Paxos, Raft等
• 容恶意节点
（无需节点可信假设）
• 共识节点数受限
• 且不可动态加入退出
• 性能效率高（高吞吐量）
• 弱同步网络
• 确定型一致性
• 收敛速度快
PBFT
• 容恶意节点
（无需节点可信假设）
• 共识节点数不受限
• 且可动态加入/退出
• 异步网络
• 弱一致性
• 资源消耗型
• 性能效率低下
• 收敛速度慢(交易确认速度慢)
PoW
分片 + PBFT或见证人+ PBFT，
进一步提升性能效率
Corda
• 异步网络
• 强一致性
• 其他同PBFT
Honeybadger BFT,
Hashgraph
• 非资源消耗型
(采用VRF等来进行leader election)
• 其他同PoW
PoS, PoC

共识技术的发展与分类

- 容恶意节点
  （无需节点可信假设）
- 共识节点数不受限
- 且可动态加入/退出
- 异步网络
- 弱一致性
- 资源消耗型/非资源消耗型
- 性能效率低下
- 收敛速度慢(交易确认速度慢)

PoW, PoS, PoC

- 弱一致性共识机制与强一致性共识机制的结合，用以改善收敛速度慢的问题，如：PoW+PBFT
- PoW与PoS的结合，用以增强共识协议的安全性（抗51%算力攻击、抗自私挖矿攻击）

- PoW + PBFT + 分片机制，用以提升性能效率
- 解耦PoW共识中的Leader election和Transaction serialization，加快Transaction serialization处理，提升性能效率，如Bitcoin-NG
- DPoS + PBFT：社群投票（社区治理）+ PBFT，大大提升性能效率

- 单一链结构 → inclusive blockchain结构，增强公平性的同时提升了些许吞吐量，如：GOHST协议
- 单一链结构 → DAG结构，将交易处理方式由顺序化处理转变为可并行化处理，极大提升性能效率，如：TANGLE, Byteball的共识机制等
- PoW/PoS + DAG：继承PoW/PoS安全性的同时，兼有交易并行处理特性

比特币的共识机制

比特币的共识机制 – Nakamoto共识或PoW共识协议

- 比特币网络中每个节点本地存储哪些数据？做哪些事情？
  - **完全有效节点：（全节点）**
    - 完整的（共识）区块链
      - » 整条区块（头）链
      - » 区块链上的交易
    - 还未被消费的“交易输出”列表
      - » 这个列表最好是放在内存里而非硬盘上，要用来快速判断待处理交易的有效性
    - 交易池
      - » 收到的、待打包成块处理的交易
    - ◆ 会从交易池收集数据、挖矿
  - **轻量节点（轻客户端，简单付款验证，Simplified Payment Verification，SPV）：**
    - 完整的区块（头）链
    - 只存储所关心的、需要进行核验的部分交易
    - ◆ 不挖矿

比特币的共识机制 – Nakamoto共识或PoW共识协议

- 每个比特币网络节点如何确保自己本地的区块链数据是与系统内（大多数）节点是一致的?
  - 分布式共识

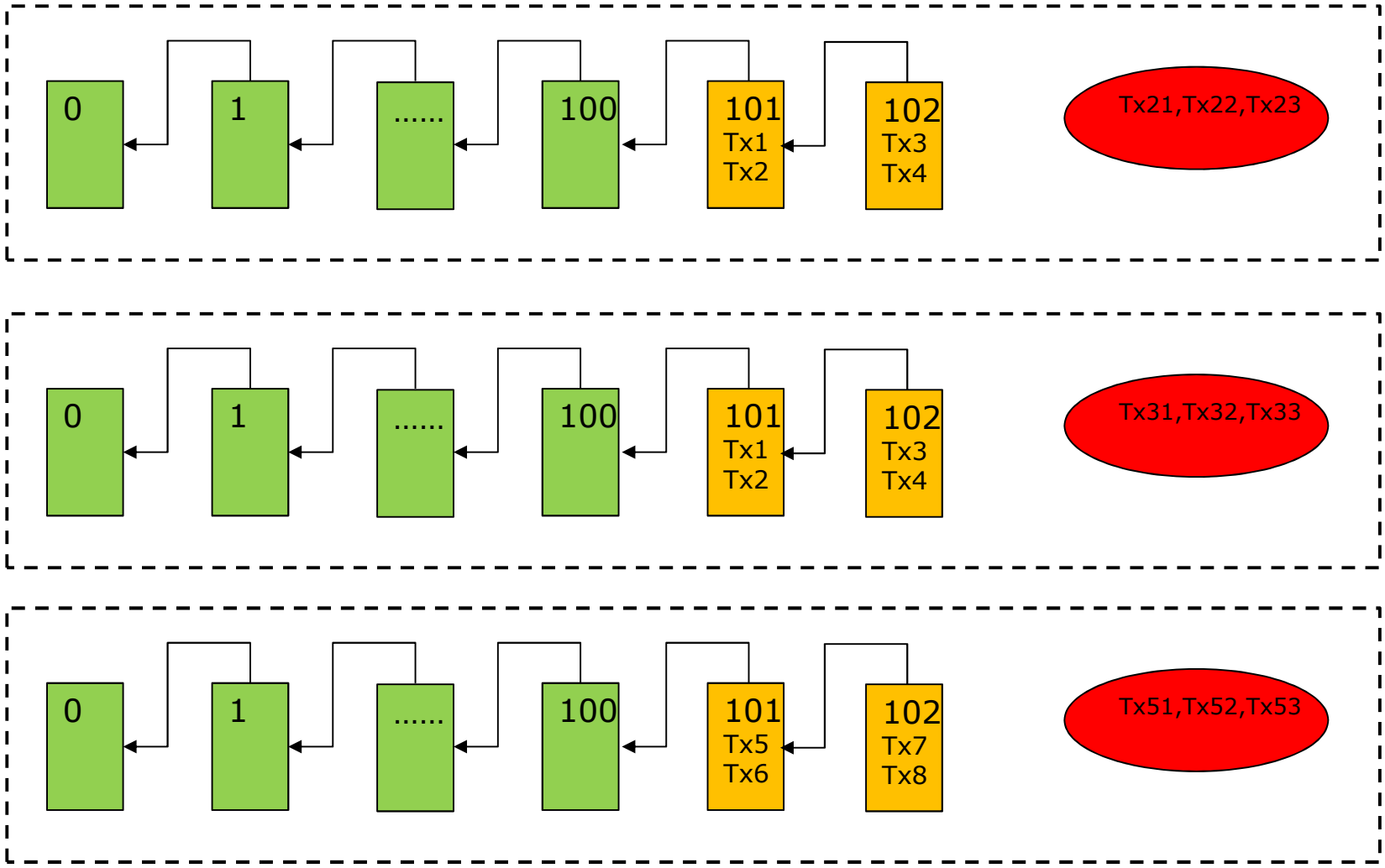
- 分布式共识（Distributed Consensus）
  - 协议能够终止，且此时所有诚实节点对数据达成一致
  - 最终达成一致的数据必须是由诚实节点生成的

比特币的共识机制 – Nakamoto共识或PoW共识协议

- 比特币的共识目标： 在任何时间点，
  - 所有在比特币网络上的节点都有包含一系列区块的总账本，每个区块中包含了所有节点达成共识的交易清单
  - 每个节点都在其交易池里有已经被通知、但还没有进入区块的交易，每个节点的交易池可能是与其他节点的交易池有差别的

- 问题：
  - 一个刚刚被挖出来的区块，被广播出去，有的节点收到并接受了，有的节点还没有收到，处于什么状态？
  - 如果不只一个处于这样状态的区块，它们处于什么状态？
    - 一定会有这样的（多个）区块的，因为是在一个分布式网络里，网络是异步的

比特币的共识机制 – Nakamoto共识或PoW共识协议



比特币的共识机制 – Nakamoto共识或PoW共识协议

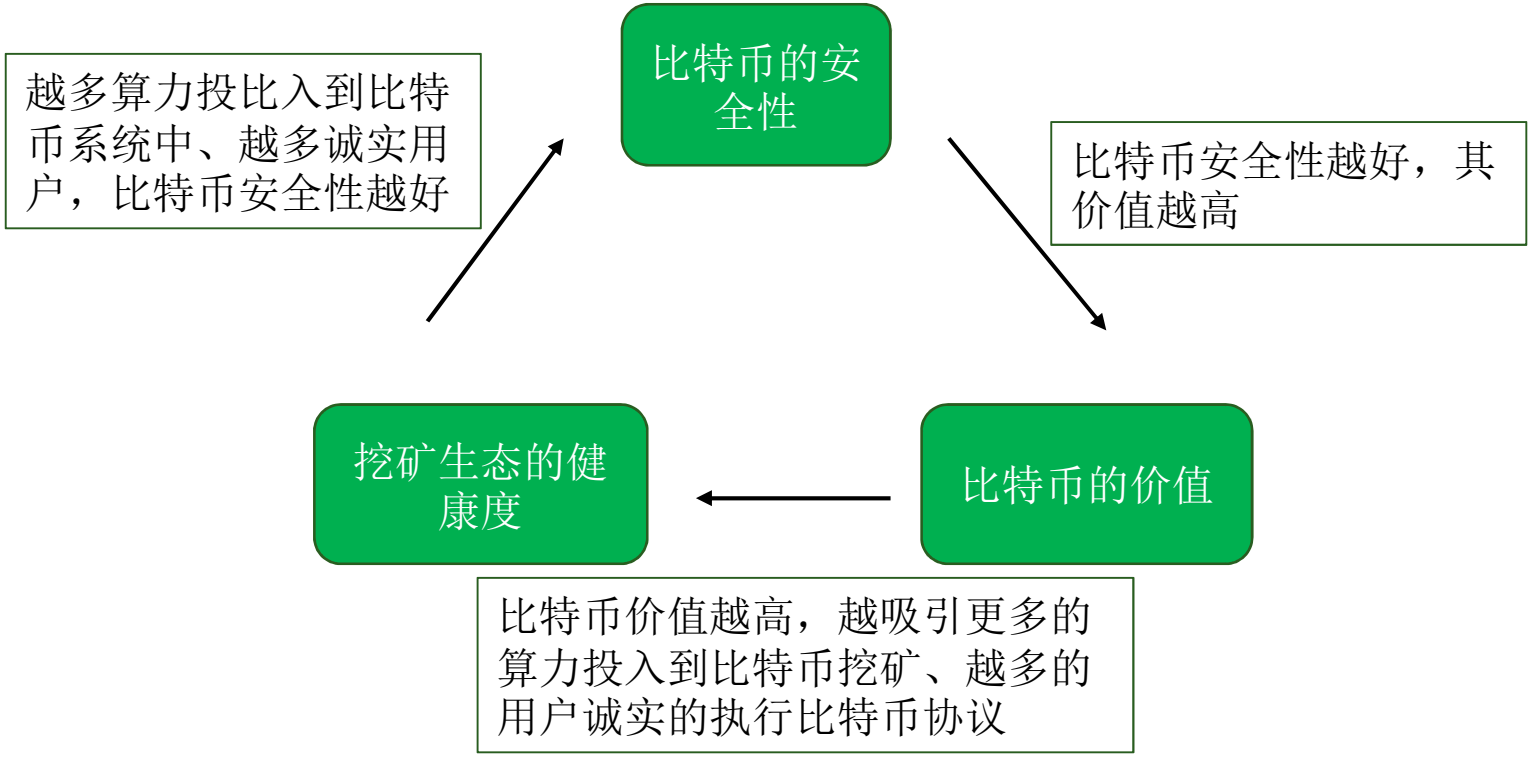
- 比特币共识机制
  - 谁提出最新的块（的内容）？
    - 节点挖矿竞争，挖得合格的块后广播到比特币网络上去 -- **本质上是Leader election & transaction serialization**
  - 谁决定被提出的块是否被整个网络接受？（对区块链的内容达成共识）
    - 没有协商过程：各节点是互不信任的，不会互相协商
    - 采用的是**隐性共识**：每个节点自己决定是否接受收到的块
      - » 检验块的有效性
      - » **如果接受，就把收到的该块作为最新块加到自己本地的区块链尾部来更新本地区块链，并在更新后的链的基础上再挖矿**
      - » **如果不接受，则仍然以本地当前的区块链为基础继续挖矿**
    - 共识的结果：
      - » 系统中诚实节点占“大多数”的话，则所有节点本地的共识区块链是一致的。
        - **诚实节点：**遵循比特币协议的节点
        - **大多数：**并不是节点大多数、身份大多数等，而是“算力”大多数
        - **所有节点达成共识：**数据不与系统共识区块链一致的节点，其后继处理的交易、块等并不会被其他节点接受、纳入共识区块链。

比特币的共识机制 – Nakamoto共识或PoW共识协议

- 比特币共识机制
  - 比特币网络节点为什么要诚实?
    - 区块被整个网络接受后能获得区块奖励和交易费：经济刺激
    - 如果把无效的交易放到自己提出的块中，这个块虽然可以被自己以及合谋的节点纳入各自的本地链中，但其他的诚实的节点是不会接受这个含有无效的交易的块的，诚实节点挖矿时不会在这样的“无效块”的基础上挖，其中的区块奖励自然也就花不出去了
    - 只要合谋的节点后面继续在这个“无效块”的基础上挖就行了?
      - » 在一次竞争“提出块”的过程中，赢得竞争的概率等于算力占全系统算力的百分比。
      - » 所以，“大多数”是算力大多数，而不是节点个数
      - » 如果算力占到系统总算力的大多数，以当前情况来看，在计算资源的投入上已经很大了。如果去做恶意的、不诚实的行为，会导致比特币价格下跌甚至崩溃，那么这些投入的资源、以及已经取得的比特币的价值也就随之下跌了。会这样去做吗?

比特币的共识机制 – Nakamoto共识或PoW共识协议

- 比特币共识机制
  - 比特币共识机制的成功的根本原因：经济刺激

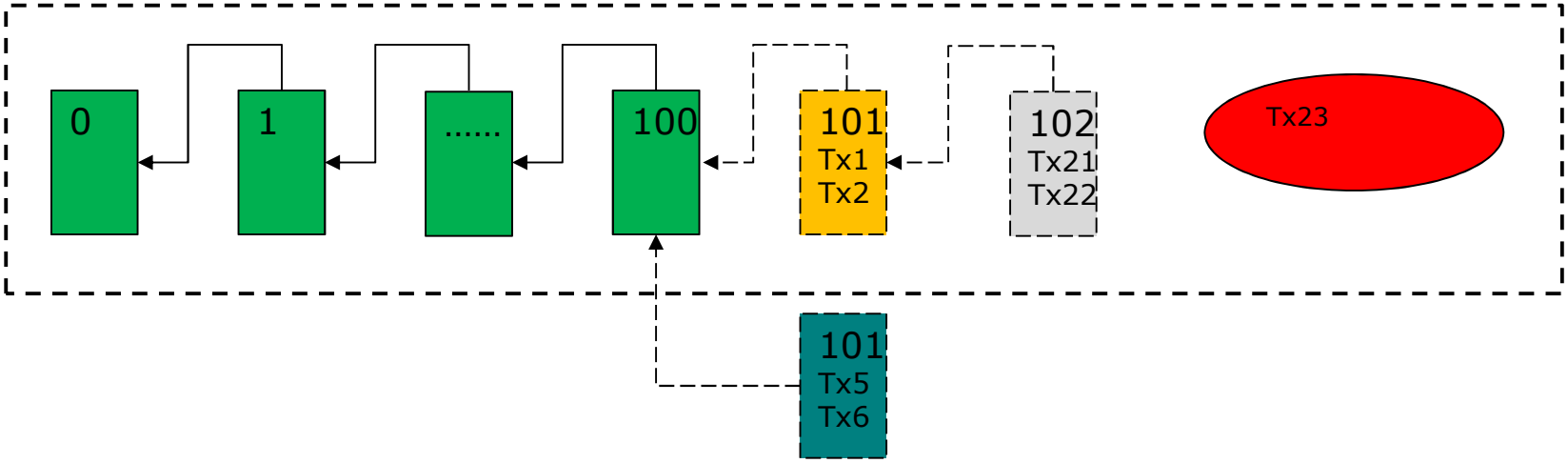


比特币的共识机制 – Nakamoto共识或PoW共识协议

- 比特币共识机制
  - 比特币共识机制的成功的根本原因：经济刺激
  - 三者互相依赖
    - 正向：正向促进
    - 反向：任何一个环节出问题，会导致整个系统崩溃
  - 比特币最初并不具备任何一个条件
    - 比特币是幸运的，安全地度过了其脆弱的“婴儿期”
    - 后来的AltCoin怎么办？

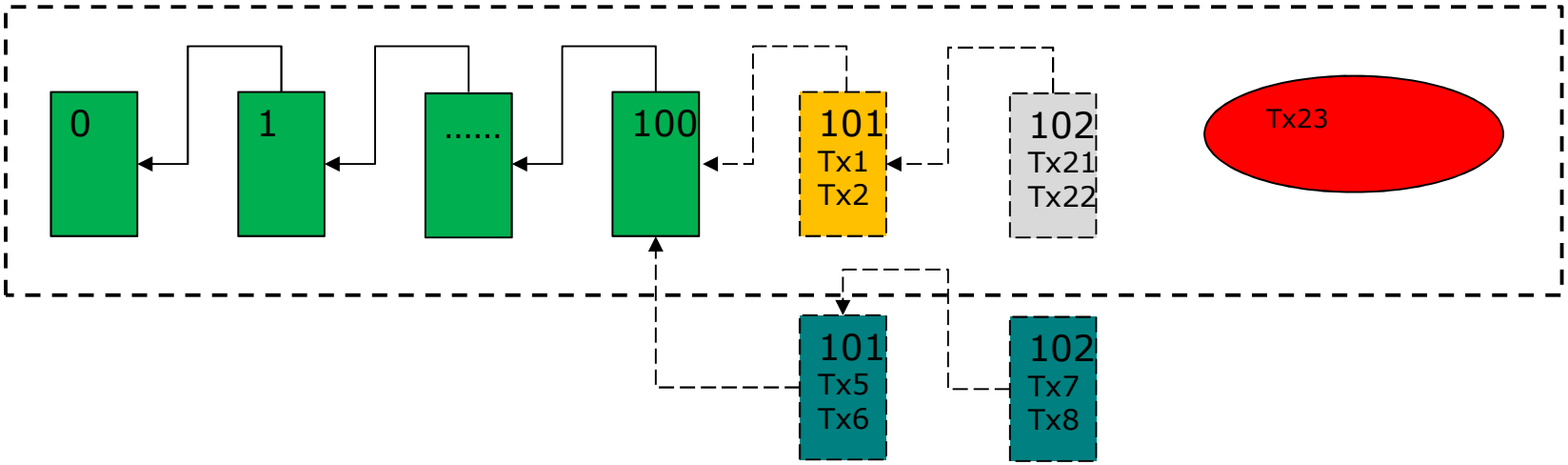
比特币的共识机制 – Nakamoto共识或PoW共识协议

- 比特币共识机制
  - PoW （工作量证明）
  - 隐性共识
  - **当接收到的新块可能导致链“分叉”时怎么办?**
    - 在自己挖到一个新块之前，收到了两个相同高度的最新块
    - “跟随先收到的”



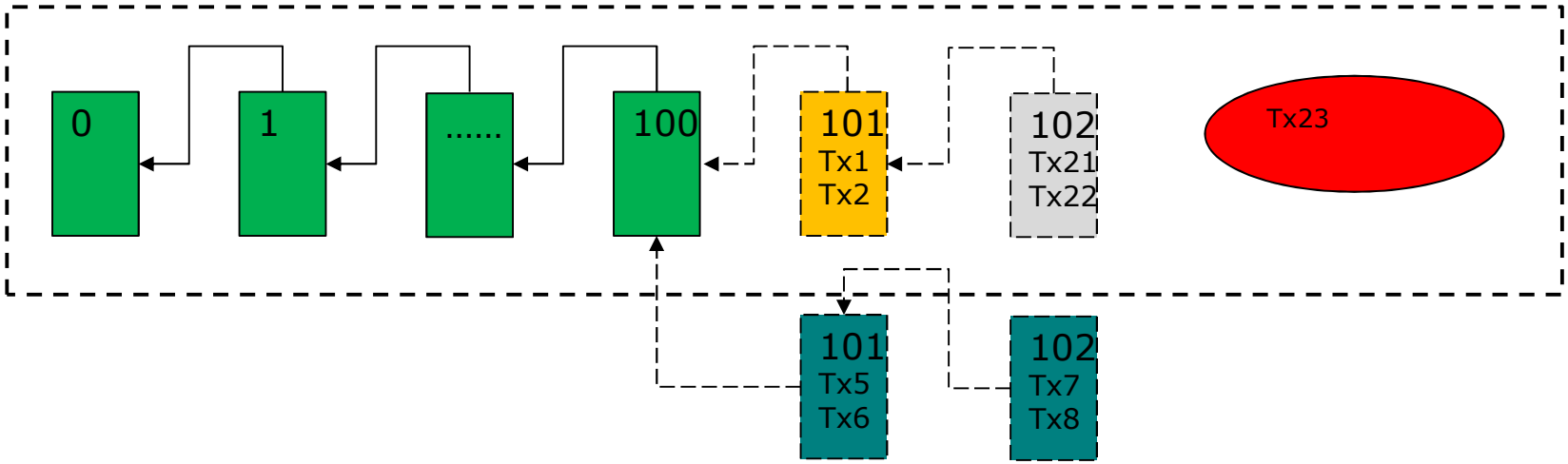
比特币的共识机制 – Nakamoto共识或PoW共识协议

- 比特币共识机制
  - PoW （工作量证明）
  - 隐性共识
  - **当接收到的新块可能导致链“分叉”时怎么办?**
    - 当收到不同高度的新块时



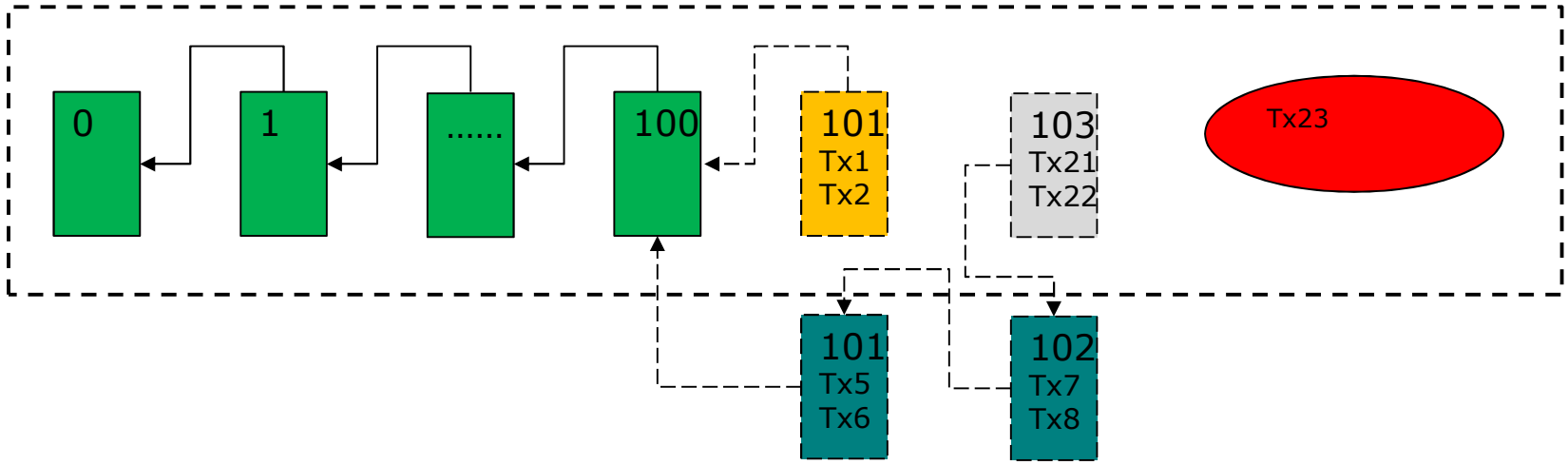
比特币的共识机制 – Nakamoto共识或PoW共识协议

- 比特币共识机制
  - PoW （工作量证明）
  - 隐性共识
  - **当接收到的新块可能导致链“分叉”时怎么办?**
    - 当收到不同高度的新块时
    - 选择跟随“最长链”：**遵从最长链原则**



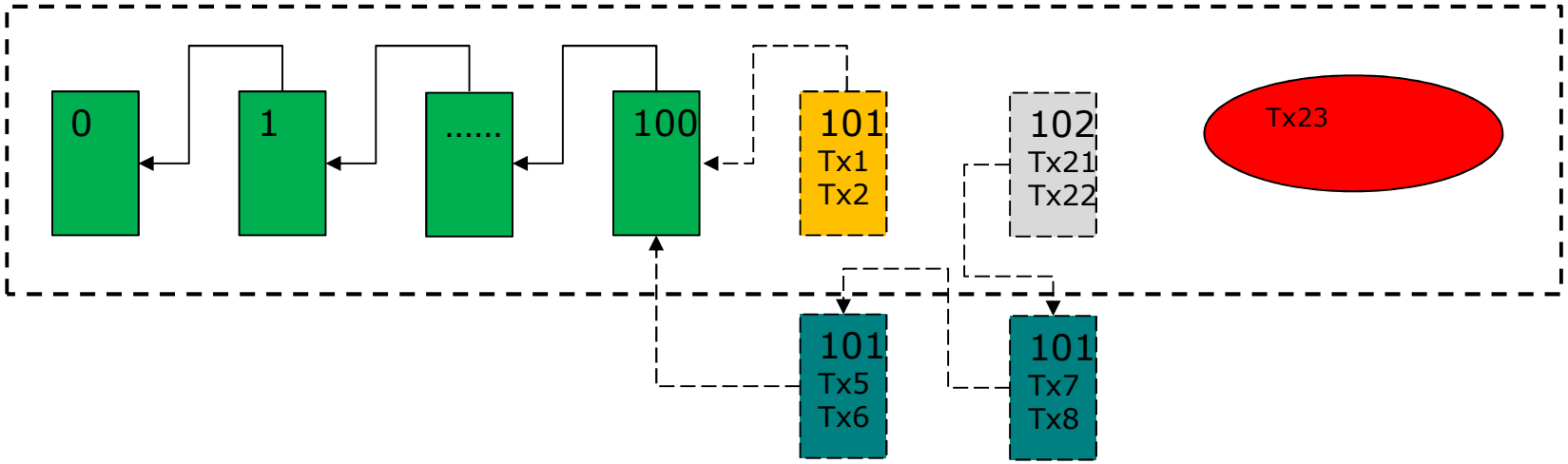
比特币的共识机制 – Nakamoto共识或PoW共识协议

- 比特币共识机制
  - PoW （工作量证明）
  - 隐性共识
  - **当接收到的新块可能导致链“分叉”时怎么办?**
    - 当收到不同高度的新块时
    - 选择跟随“最长链”：**遵从最长链原则**



比特币的共识机制 – Nakamoto共识或PoW共识协议

- 比特币共识机制
  - PoW （工作量证明）
  - 隐性共识
  - **当接收到的新块可能导致链“分叉”时怎么办?**
    - 当收到不同高度的新块时
    - 选择跟随“最长链”：**遵从最长链原则**
    - 挖到一个块是相当困难的，每一个块都证明了有相应的算力已经“投资”在该块上，“长链”意味着有更多的算力（节点）认可了该链，最终这些块被纳入最终链的可能性更大；而不与最长链在一起的，则会被“抛弃”（网络内节点不认可）

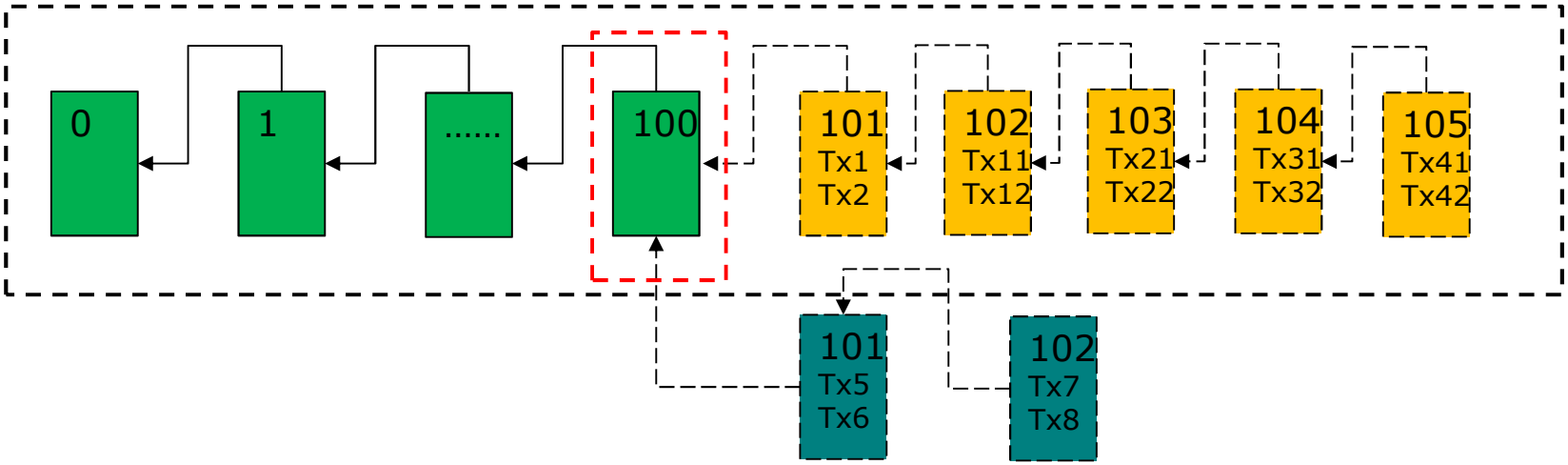


比特币的共识机制 – Nakamoto共识或PoW共识协议

- 比特币共识机制
  - PoW （工作量证明）
  - 隐性共识
  - **当接收到的新块可能导致链“分叉”时怎么办?**
    - 先到原则
    - **最长链原则**
  - **已经上链的也可能因为会被丢弃，什么时候才真正的被纳入最终共识链、不会再被丢弃?**
    - Alice向Bob买书、并通过比特币支付，什么时候Bob才放心给Alice发货?
    - 交易的多次确认：包含交易的块上链是第1次确认，在这个块后每增加一个新块，确认次数加1（每多一次确认，意味着更多的算力（节点）认可了该交易）
    - 比特币认为6次确认后，交易就很难被篡改了。（除非攻击者达到算力大多数，否则，攻击者使得相应的块被丢弃的概率可忽略不计，攻击者没有能力连追6个块制造出一条更长链）

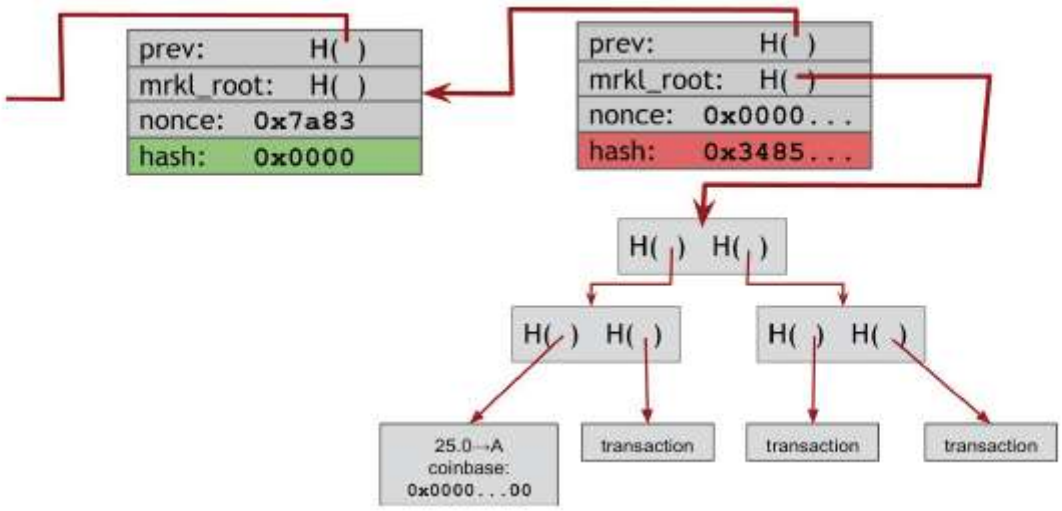
比特币的共识机制 – Nakamoto共识或PoW共识协议

- 比特币共识机制
  - PoW （工作量证明）
  - 隐性共识
  - **当接收到的新块可能导致链“分叉”时怎么办?**
    - 先到原则
    - **最长链原则**
  - **已经上链的也可能因为会被丢弃，什么时候才真正的被纳入最终共识链、不会再被丢弃?**
    - **比特币的6块确认原则。为何是6块确认原则?**



PoW共识协议的基本数学模型

- PoW挖矿就是不断进行*Hash运算*寻找合适的nonce值使区块头的Hash值小于给定的目标值



PoW共识协议的基本数学模型

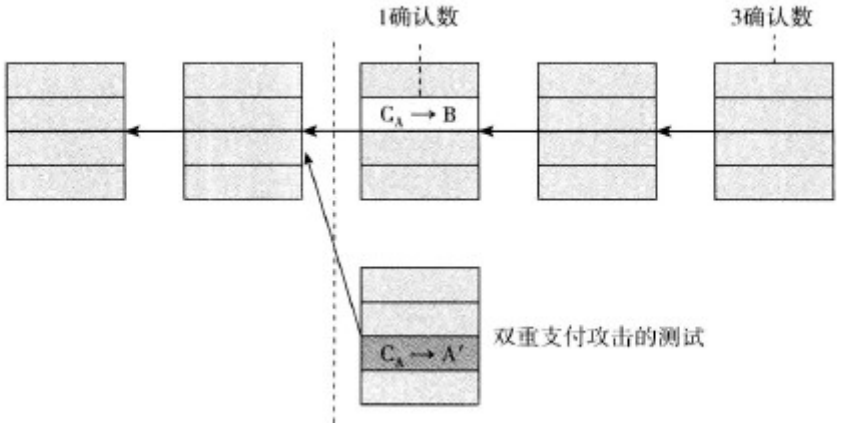
- 一般地，可以认为密码学hash函数在输出空间上均匀分布。不妨设Hash值小于目标值的概率为p。
- PoW挖矿过程（即区块创建过程）可以看成伯努利过程（iid的伯努利试验），为此，在时间t内挖矿成功的次数服从 B（vt , p），其中v为单位时间hash运算的次数（即算力）
- 可以进一步连续化为泊松过程：即在时间t内挖矿成功的次数服从Poisson（vpt）
- 可以证明在忽略网络时延的情况下，先挖到块的概率与矿工算力成正比。

PoW共识协议的基本数学模型

- 交易场景：Alice向Bob购买商品，Alice先向Bob转一笔比特币，Bob何时发货比较稳妥？

- 双花攻击：上述交易被打包至区块B_i，即使B_i被后面的多个区块确认后，Alice仍然可能制造一条分叉链超过B_i所在的链，根据最长链原则，B_i所在的链被超过后，B_i中的交易将成为无效交易。

- 为此，需要计算：B_i被后面多少个区块确认后，才能使得上述双花攻击成功的概率足够小。

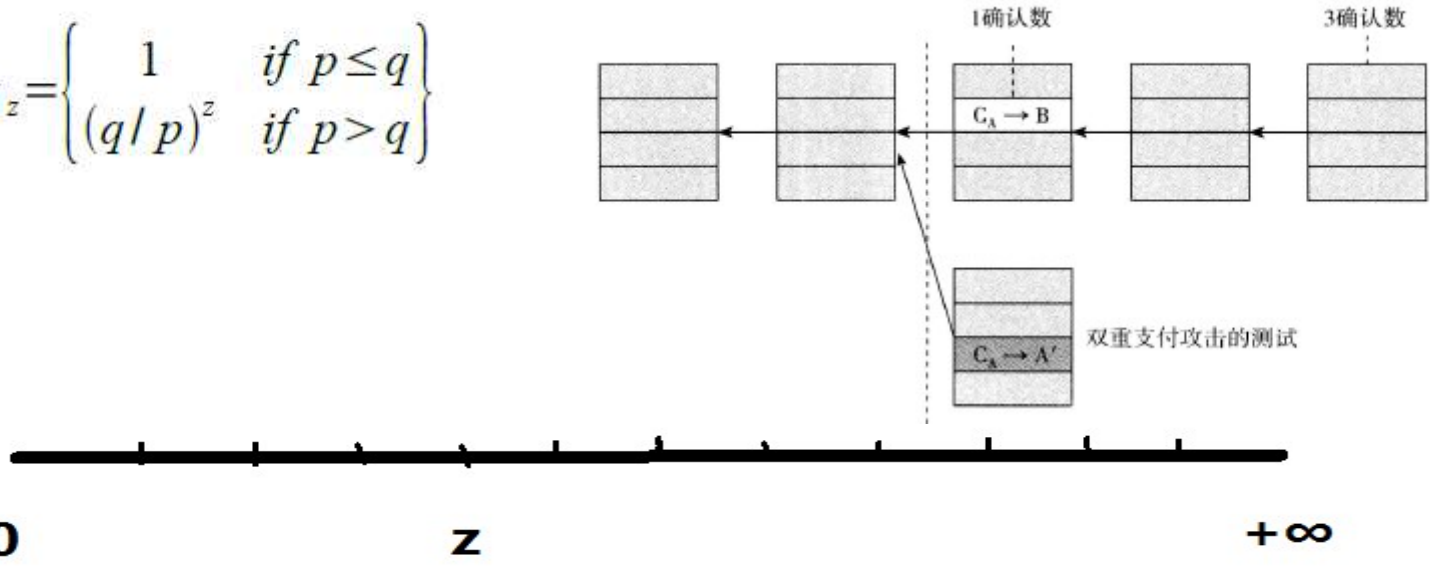


PoW共识协议的基本数学模型

模型类比：将Gambler's Ruin模型引入到（诚实节点产生的）原有最长链与（攻击者产生的）分叉链的竞争模型中，特别地，将Gambler's Ruin中的金额 i 对应于（诚实节点产生的）原有最长链与（攻击者产生的）分叉链之间的长度差。

p = probability an honest node finds the next block
q = probability the attacker finds the next block
q_z = probability the attacker will ever catch up from z blocks behind

$$q_z = \begin{cases} 1 & \text{if } p \le q \\ (q/p)^z & \text{if } p > q \end{cases}$$



PoW共识协议的基本数学模型

- 回顾我们的问题：等待几个确认才能使双花攻击成功的概率足够小呢？
- 假设Bob的交易包含在区块B_i中，且后面连了z个区块，在产生B_{i+1}至B_{i+z}的时间段内，攻击者在B_{i-1}后产生的分叉链上的区块数 k 服从参数为 $\lambda = z\frac{q}{p}$ 的泊松分布。
- 由全概率公式可以得到分叉链追上原有最长链的概率为：

$$\sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \begin{cases} (q/p)^{(z-k)} & if\ k \le z \\ 1 & if\ k > z \end{cases}$$

PoW共识协议的基本数学模型

- **实验结果：** 双花攻击成功概率随着区块确认个数 z 的增加而大幅降低。

```
q=0.1
z=0     P=1.0000000
z=1     P=0.2045873
z=2     P=0.0509779
z=3     P=0.0131722
z=4     P=0.0034552
z=5     P=0.0009137
z=6     P=0.0002428
z=7     P=0.0000647
z=8     P=0.0000173
z=9     P=0.0000046
z=10    P=0.0000012
```

```
q=0.3
z=0     P=1.0000000
z=5     P=0.1773523
z=10    P=0.0416605
z=15    P=0.0101008
z=20    P=0.0024804
z=25    P=0.0006132
z=30    P=0.0001522
z=35    P=0.0000379
z=40    P=0.0000095
z=45    P=0.0000024
z=50    P=0.0000006
```

PoW共识协议的基本数学模型

- 在攻击者不同算力条件下，使得双花攻击成功的概率小于0.001所需要的区块确认数：

```
P < 0.001
q=0.10   z=5
q=0.15   z=8
q=0.20   z=11
q=0.25   z=15
q=0.30   z=24
q=0.35   z=41
q=0.40   z=89
q=0.45   z=340
```

- 增加区块确认数可以更好地防止双花攻击
- 现实中，比特币系统采用6个区块确认，约为一小时