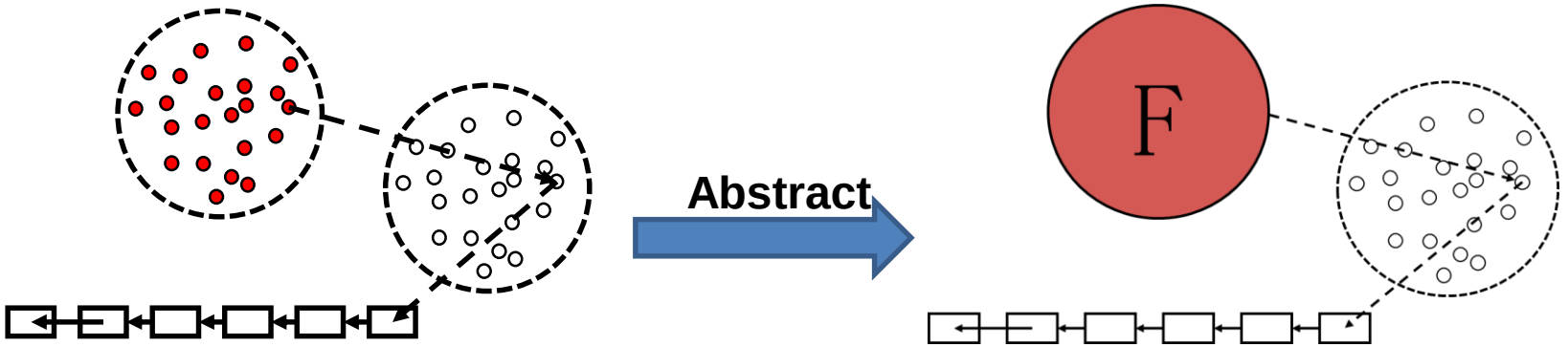
权益证明共识协议

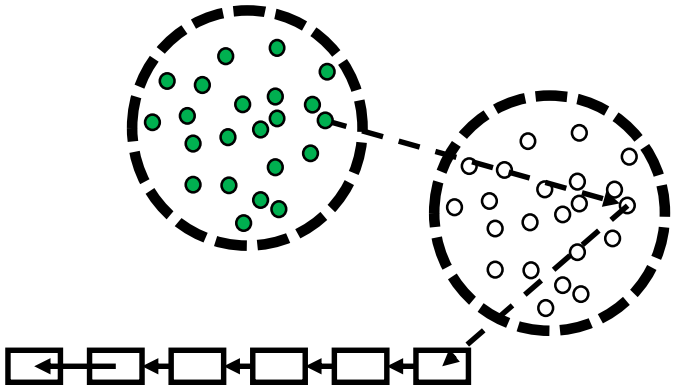
范磊
上海交通大学

重新审视区块链的基本原理



- 区块链可以看做是由哈希值或者数字签名算法连接起来的区块链条，区块是由参与者分布式生成的
- 可以将PoW区块链的生成过程抽象为一个随机函数指定参与者轮流生成区块

PoS是不是可以模拟PoW



- 如果有个理想的函数代替PoW产生随机数，随机指定用户生成区块则可以模拟PoW过程
- 持续不断的理想随机数产生是非常困难的

随机灯塔 Random Beacon

持续输出随机数的随机源称为Random Beacon，如何构造Beacon?

- 来自外部
  - 由可信第三方通过可靠的方法产生并输出随机数（NIST？可信吗）
  - 由外部可不受控物理源产生随机数（太阳黑子？不够纯）
  - 弱随机数提纯（Random Extractor？需要输入纯随机种子）
- 来自内部
  - 通过安全多方计算产生（问题回到了PoS共识协议）

可验证随机函数VRF

Verifiable Random Functions

- VRF算法组成：Keygen、Evaluate、Verify
  - Keygen(r) → (VK, SK)：Keygen 产生一对非对称密钥对：验证密钥 VK（公钥） 和私钥 SK
  - Evaluate(SK, X) → (Y, ρ)：求值函数 Evaluate 输入私钥 SK、消息 X，输出伪随机字符串 Y 和证明 ρ。
  - Verify(VK, X, Y, ρ)→ 0/1： 验证函数 Verify 输入验证密钥 VK、消息 X 以及求值函数中输出的伪随机字符串 Y 和证明 ρ。只有该函数验证了证明 ρ 是根据 X 生成的，且根据证明 ρ 可以推导出 Y，输出 1。

可验证随机函数VRF

Verifiable Random Functions

- VRF算法特性
  - Uniqueness： 给定输入，有且仅有一组<Y, ρ> 满足Verify(VK, X, Y, ρ)→ 0/1
  - Provability： 如果是Y, ρ是合法生成的，Verify(VK, X, Y, ρ)=1
  - Pseudo-randomness： 如果没有计算过Evaluate(SK, X) → (Y, ρ)，任何多项式时间算法无法预测Y的信息

VRF构造

Unique Signature

- 满足数字签名的基本特性
  - 可验证性：$s = sig_i(m)$, then $V(pk_i, m, s) = YES$
  - 不可伪造性：$V(pk_i, m, s) \neq YES$
- 额外满足
  - 唯一性：$s \neq s'$ and $V(pk', m, s) = V(pk', m, s') \neq 1$

BLS签名

Unique Signature

两个阶为r的群：G, G_T

满足双线性映射 e: $G \times G \to G_T$

g是G上的生成元，则有：$e(g^x, g^y) = e(g, g)^{xy}$

设x是私钥，对应的公钥是：g^x

给定一个消息m，则对应的签名是：$h = H(m)$ $\sigma = h^x$

签名验证：$e(\sigma, g) = e(H(m), g^x)$

VRF构造

Unique Signature + Hash Function ⇒ VRF

$$m \rightarrow H(sig_i(m))$$

VRF输入来源

- 前面若干区块的状态
  - 保证前后状态连接，形成链式结构
- 生成者的身份信息
  - 确定块的生成者，实现对出块者的奖励
- 生成者的数字签名
  - 确认块是由token的拥有者生成的
- 某种随机性
  - 确保能够出块，避免陷入到死结状态

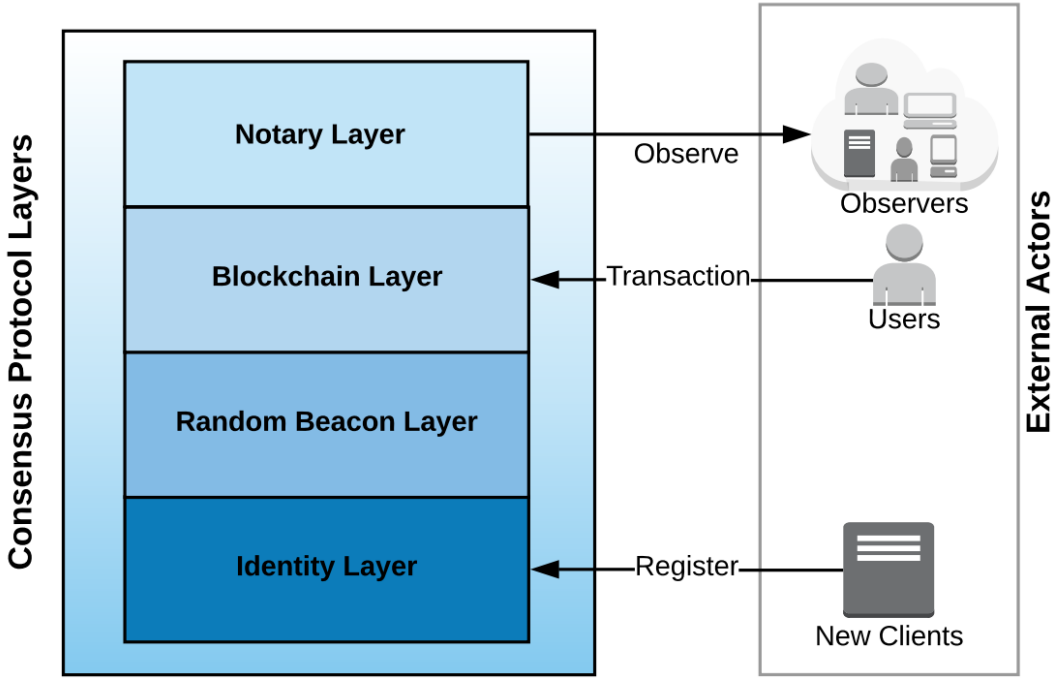
避免随机数的不确定性

- 降低输入的信息量
  - 去除nonce
  - 去除Transaction
- 与Account绑定
  - 避免Sybil attack

VRF仍有不确定性

- 攻击者可能有限地改变已有状态，选择最有利的输入
- 不同账号生成不同的输出，并不完全一致
- Nothing At Stake 攻击
- 解决方案
  - 投票表决
    - 表决出一个Committee（Difinity）
    - 表决出一个block(Algorand)
  - 依赖一个整体筛选概率
    - Ourpboros Praos
  - 自然竞争
    - iChing

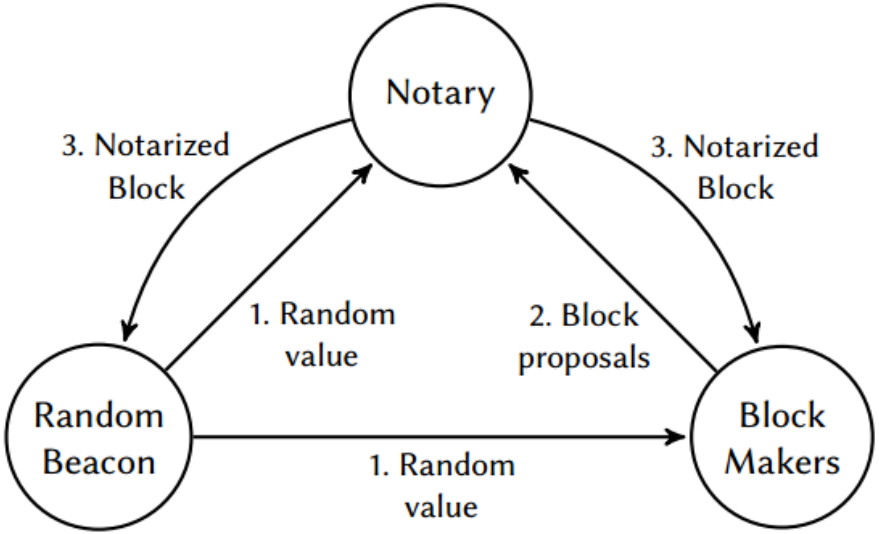
Difinity基本架构



- Difinity采用层次化的抽象模型设计
- 其核心思想来源于存在随机灯塔
- 随机灯塔之后被分布式算法代替

- Unique Threshold Signature

Difinity共识流程

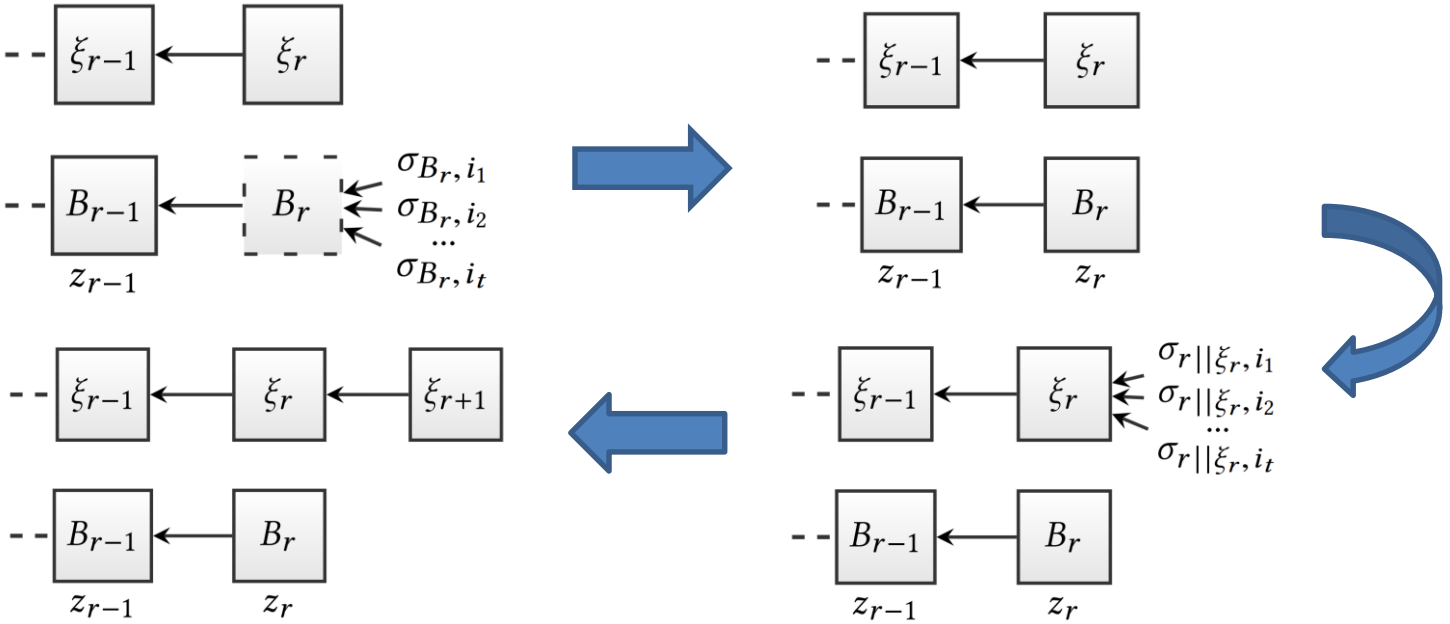


随机灯塔

产生区块

背书区块

Difinity区块与Beacon的伴生



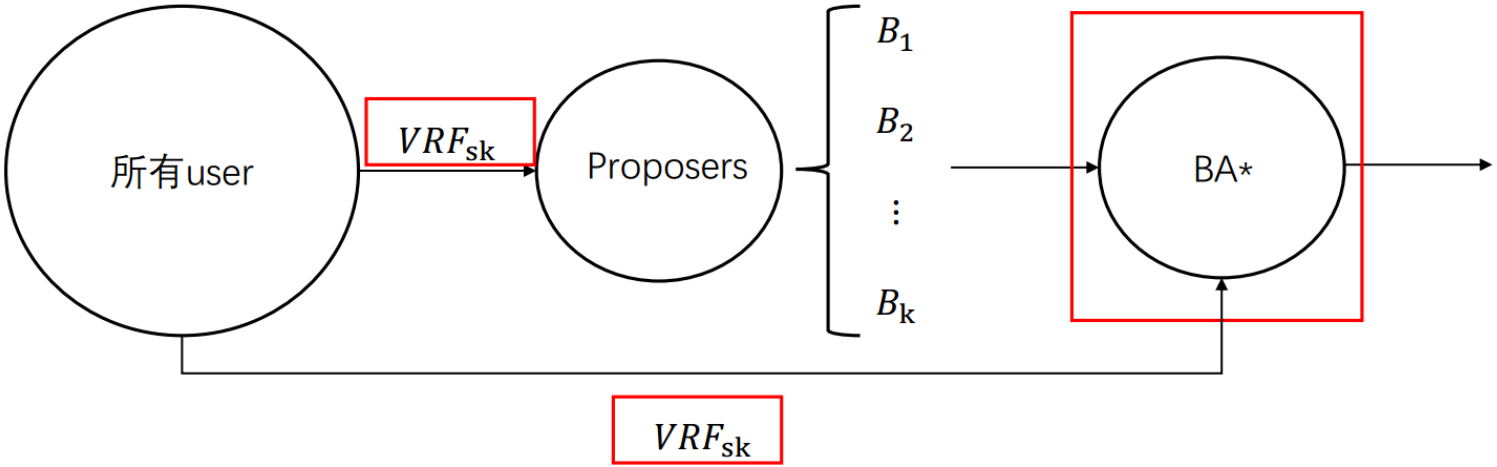
Difinity如何落地

- Threshold Relay 实现委员会的更替
  - 将用户分割为若干组
  - 每组执行Distributed Key Generation （DKG）协议
  - 每个时间片由所生成的随机数选择相应的组
  - 组内用户分布式生成新的随机数，从而选择下一个委员为
- Open Participation 实现新用户的加入
  - 新用户注册为系统用户
  - 新注册用户加入组需要组成员投票，并更新DKG过程

Difinity分析

- 安全性
  - 假设每一组诚实用户占优势，并且超过门限签名所需上限
  - 每组用户数量不能太少
- 网络依赖
  - 执行过程假设半同步网络，类似PBFT类协议
- 执行效率
  - DKG效率较低
  - 做门限签名O(n)的复杂度

Algorand基本构造



将时间分片，在每一个round中运行如下过程，生成这一轮的块

Algorand具体构造

$$Q^r \triangleq H(SIG_{\ell^r}(Q^{r-1}), r)$$

- 作为第r round VRF输入的参数，保持对上一个区块的链接性.
- Sig 是unique digital signing

Algorand: Construction

$$.H\left(SIG_i\left(r, 1, Q^{r-1}\right)\right) \leq p$$

- Leader (candidates) selection

$$.H\left(SIG_i\left(r, s, Q^{r-1}\right)\right) \leq p'$$

- Verifiers selection

Algorand具体构造

$$\sigma_i^r \triangleq SIG_i\left(r, 1, Q^{r-1}\right)$$

$$H(\sigma_i^{r,1}) \leq H(\sigma_j^{r,1})$$

- Leader竞争

Algorand具体构造

$$B^r = (r, PAY^r, SIG_{\ell^r}(Q^{r-1}), H(B^{r-1}))$$

- r Round: round number
- Pay^r: payments (transactions)
- SIG: signature
- l^r: id of leader
- Q^r: the random seed for r-th round
- H: ideal hash function

BA*协议

- 当不止一个用户满足Leader条件时，通过BA*协议确定唯一的胜利者

procedure $BA\star(ctx, round, block)$:

$hblock \leftarrow \text{Reduction}(ctx, round, H(block))$
$hblock_\star \leftarrow \text{Binary}BA\star(ctx, round, hblock)$
// Check if we reached "final" or "tentative" consensus
$r \leftarrow \text{CountVotes}(ctx, round, \text{FINAL}, T_{\text{FINAL}}, \tau_{\text{FINAL}}, \lambda_{\text{STEP}})$
if $hblock_\star = r$ **then**
  return $\langle \text{FINAL}, \text{BlockOfHash}(hblock_\star) \rangle$
else
  return $\langle \text{TENTATIVE}, \text{BlockOfHash}(hblock_\star) \rangle$

Algorithm 3: Running $BA\star$ for the next *round*, with a proposed *block*. H is a cryptographic hash function.

调用Reduction初步消除非空块冲突

调用BinaryBA*消除空块冲突

BA*协议

Step1 第一轮Reduction，为最优块第一次投票并计数

Step2 第二轮Reduction，如果超时没有出块，则产生一个占位的空块，否则为投票最多的块重新投票

procedure Reduction($ctx, round, hblock$):

// step 1: gossip the block hash

CommitteeVote(ctx, $round$, REDUCTION_ONE, τ_{STEP}, $hblock$)

// other users might still be waiting for block proposals,

// so set timeout for $\lambda_{\text{BLOCK}} + \lambda_{\text{STEP}}$

$hblock_1 \leftarrow$ CountVotes(ctx, $round$, REDUCTION_ONE, $T_{\text{STEP}}, \tau_{\text{STEP}}, \lambda_{\text{BLOCK}} + \lambda_{\text{STEP}}$)

// step 2: re-gossip the popular block hash

$empty\_hash \leftarrow H(\text{Empty}(round, H(ctx.last\_block)))$

if $hblock_1 = TIMEOUT$ **then**

  CommitteeVote(ctx, $round$, REDUCTION_TWO, τ_{STEP}, $empty\_hash$)

else

  CommitteeVote(ctx, $round$, REDUCTION_TWO, τ_{STEP}, $hblock_1$)

$hblock_2 \leftarrow$ CountVotes(ctx, $round$, REDUCTION_TWO, $T_{\text{STEP}}, \tau_{\text{STEP}}, \lambda_{\text{STEP}}$)

if $hblock_2 = TIMEOUT$ **then return** $empty\_hash$;

else return $hblock_2$;

Algorithm 7: The two-step reduction.

BA*协议

循环多轮，直到达成一致

如果对某个非空块达成一致直接返回

如果对某个空块达成一致直接返回

如果不能一致，通过CommonCoin协调

如果最终不能达成一致，需要额外机制

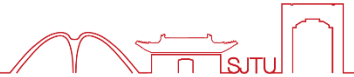
```
procedure BinaryBA⋆(ctx, round, block_hash):
step ← 1
r ← block_hash
empty_hash ← H(Empty(round, H(ctx.last_block)))
while step < MaxSteps do
    CommitteeVote(ctx, round, step, τ_STEP, r)
    r ← CountVotes(ctx, round, step, T_STEP, τ_STEP, λ_STEP)
    if r = TIMEOUT then
        r ← block_hash
    else if r ≠ empty_hash then
        for step < s′ ≤ step+3 do
            CommitteeVote(ctx, round, s′, τ_STEP, r)
        if step = 1 then
            CommitteeVote(ctx, round, FINAL, τ_FINAL, r)
        return r
    step++

    CommitteeVote(ctx, round, step, τ_STEP, r)
    r ← CountVotes(ctx, round, step, T_STEP, τ_STEP, λ_STEP)
    if r = TIMEOUT then
        r ← empty_hash
    else if r = empty_hash then
        for step < s′ ≤ step+3 do
            CommitteeVote(ctx, round, s′, τ_STEP, r)
        return r
    step++

    CommitteeVote(ctx, round, step, τ_STEP, r)
    r ← CountVotes(ctx, round, step, T_STEP, τ_STEP, λ_STEP)
    if r = TIMEOUT then
        if CommonCoin(ctx, round, step, τ_STEP) = 0 then
            r ← block_hash
        else
            r ← empty_hash
    step++
// No consensus after MaxSteps; assume network
// problem, and rely on §8.2 to recover liveness.
HangForever()
```

Algorithm 8: Binary*BA⋆* executes until consensus is reached on either *block_hash* or *empty_hash*.

Algorand正确性讨论

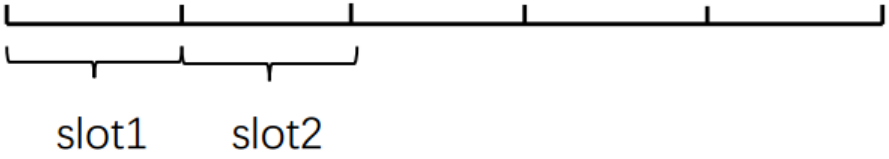
- BA* 在网络同步的假设下可以达成一致（Common Prefix）
- 诚实用户按照所拥有的stake数量按照相应的概率被选为Leader（Chain Quality）
- 如果恶意用户不出块，或者不按照规则出块，最终仍然能够产生下一个块（Chain Growth）

Algorand的问题



- 没有真正解决中心化问题
- Committee不能过小，否则可能被坏人掌握
- 多轮协议执行时间度复杂，尤其当leader是恶意节点时

Ouroboros Praos



- Semi-synchronous假设
- 节点之间时间同步的偏差对于一个slot的长度而言是可忽略的
- Slot的编号是单调递增的，而且参与的节点均可知。
- 坏人的能力
- 坏人可以重排消息的发送顺序，给某些人发送某些消息，拒绝某些消息，可以将honest发送的消息延迟最多Δ slots。
- Corrupt没有延迟

出块条件

B_0

S_0
$\eta \leftarrow \{0,1\}^{\lambda}$

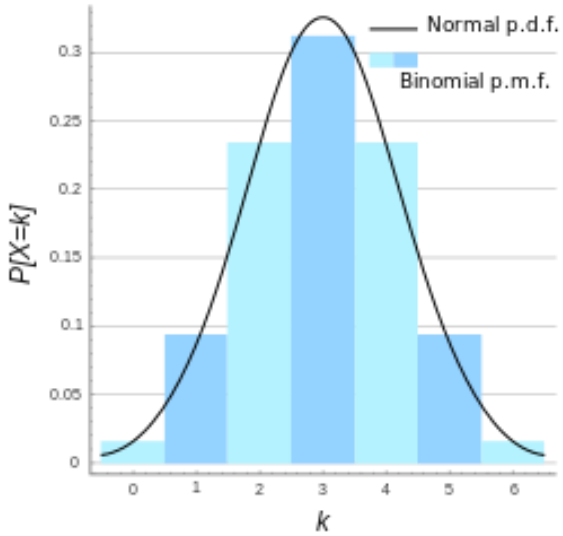
$$\mathbb{S}_0 = \left((U_1, v_1^{\text{vrf}}, v_1^{\text{kes}}, v_1^{\text{dsig}}, s_1), \ldots, (U_n, v_n^{\text{vrf}}, v_n^{\text{kes}}, v_n^{\text{dsig}}, s_n)\right)$$

- Ui 成为slot$j$的leader的条件是：$VRF_{sk_i^{vrf}}(\eta||sl_j) < T_i$
  - $vvrf$:用来验证结果确实是由对应的私钥与$\eta||slj$生成的。
  - $vkes$: 用来对生成的块进行签名，该签名是前向安全的
  - $v$dsig: 用来对交易签名，使用一般的签名机制即可。

安全性分析（informally）

- 用户筛选符合二项式分布

$$\mu \pm 3\sigma = np \pm 3\sqrt{np(1-p)} \in (0, n).$$



iChing

- 设计原则
  - 避免交互式协议设计
  - 避免引入可信第三方
  - 避免使用工作量证明的竞争机制
  - 避免使用其他损耗性物理资源

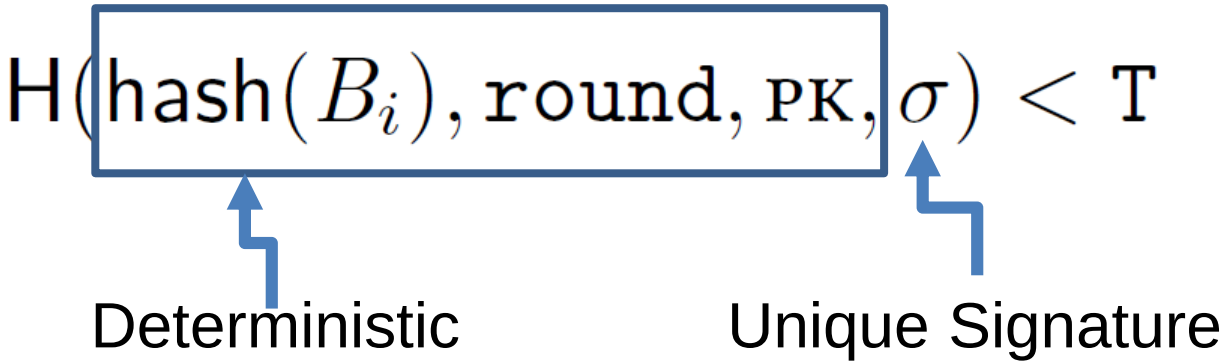
iChing-类似PoW的区块链构造

$$\mathrm{H}(\mathrm{hash}(B_i), \mathtt{round}, \mathrm{PK}, \sigma) < \mathrm{T}$$

关键点:

- 将PoW中的Nonce随机数去掉，并去除其他随机输入，使用round代替。
- 使用数字签名σ 确定数据来源，绑定用户token

Randomness Input of Hash Inequality



假设条件

- 所有用户使用网络连接，不假设网络延迟
- 诚实用户占大多数
- 出块率足够低，保证区块同步时间

等同于PoW 区块链假设条件

共识策略

- 选取最长链原则，最长链即为最优链
- 区块链生长过程中可能产生分叉
- 最长链分叉将会合并在一起

等同于PoW 区块链共识策略

如何扩展（生长）

- 选择最长链作为最优链，并尝试扩展
- 生成速率设为 p ，p 等效于在一个时间片内生成一个有效区块的概率

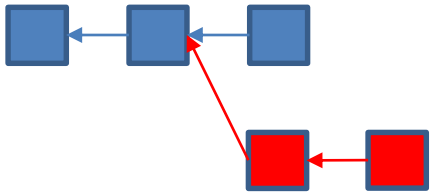
等同于PoW 区块链扩展算法

iChing安全性分析(Informally)

- 应为假设出块速率远小于1，所以大多数Round所有的诚实节点会集中扩展一条链
- 每一个stake在单位时间内生成区块的概率相同
- 假设节点用户持有的stake数量占多数，结合上一条，诚实节点所扩展的区块链生长速度占优

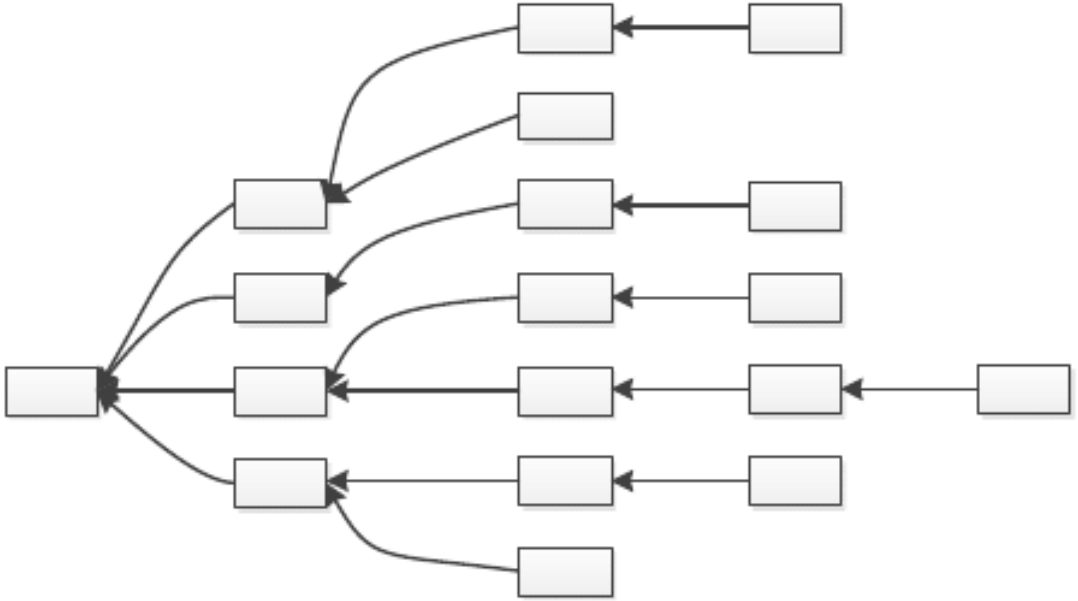
进一步分析

- 如果恶意节点实施Nothing at Stake 攻击，也就是尝试多个位置出块的可能性，将会带来增长速度的增加



根据仿真结果，如果对所有位置尝试出块，区块链的增长速度将会翻倍.

恶意区块增长速度分析



恶意区块增长速度分析

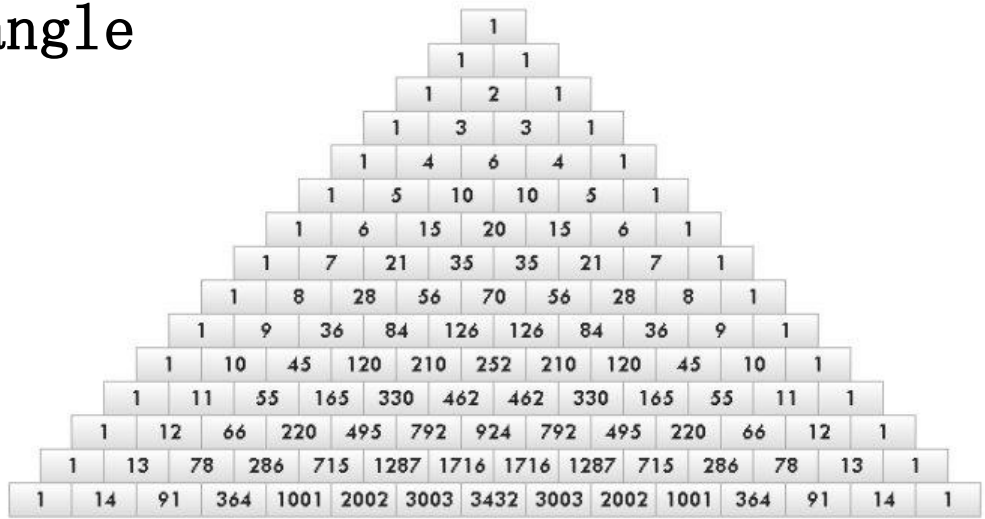
- p: 一个区块在一个round内被扩展的概率
- n: round数量
- l: 区块的高度
- $f(n,l)$: 在n round时，高度为l的区块的数量

- We have $f(n,l) = f(n-1,l) + f(n-1,l-1)p$

恶意区块增长速度分析

Pascal's triangle

杨辉三角



恶意区块增长速度分析

- *p*：一个区块在一个round内被扩展的概率
- *n*：round数量
- *l*：区块的高度
- *f(n,l)*：在n round时，高度为l的区块的数量

- We have *f(n,l) = f(n-1,l) + f(n-1,l-1)p*

$$f(n,l) = \binom{n}{l} p^l$$

恶意区块增长速度分析

- We have *f(n,l) = f(n-1,l) + f(n-1,l-1)p*

$$f(n,l) = \binom{n}{l} p^l$$

With Stirling's approximation

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

If $f(n,l) > \varepsilon$, we have

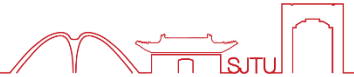
$$\frac{l}{n} \leq ep$$

如何防范贪心攻击

- 鼓励诚实用户采用适量的贪心策略.
- 诚实用户同时扩展最长链以及头部的若干区块.

Honest Players	Malicious Players	Honest Ratio Requirement
Normal	Normal	51%
Normal	Greedy	73%
Greedy Limited	Greedy	57%

如何允许新用户注册



- 新用户注册可能带来的攻击行为

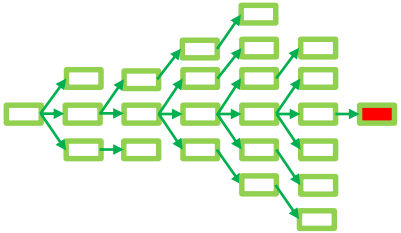
- 防范新用户注册攻击

Balancing 攻击

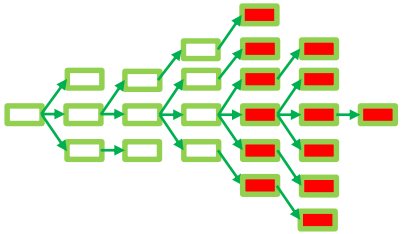
- 什么是balancing攻击

- 防范balancing攻击

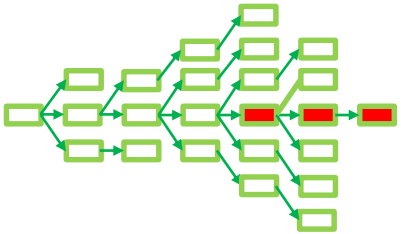
如何抵抗Balancing攻击



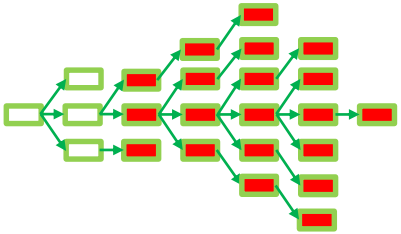
0 greedy



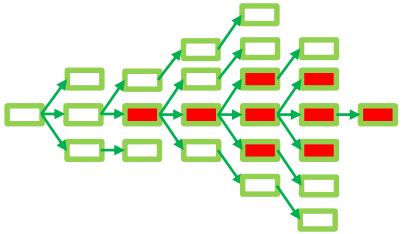
2 greedy-by-height



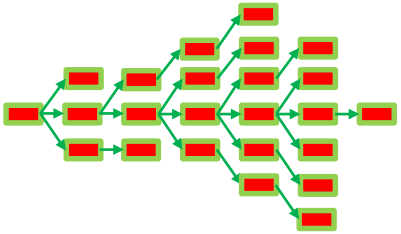
2 greedy-by-path



4 greedy-by-height



4 greedy-by-path



full greedy

安全性证明

Chain Growth

- 几个观察：
  - 恶意节点无法缩短诚实节点最长链的长度
  - 任何诚实节点最长链的长度是随着时间单调递增的
  - 大多数轮次所有诚实节点具有相同的最长链
- 推论：
  - 大多数诚实节点成功轮次将带来诚实节点最长链的增长

安全性证明

Chain Quality

- 几个观察：
  - 恶意节点无法阻止诚实节点产生区块并传播给其他节点
  - 恶意节点可以并仅可以通过竞争使得诚实节点产生的区块失效
  - 链的增长速度满足前述Chain Growth特性
- 推论：
  - 如果不是因为竞争失败，诚实节点所产生的区块将留存在最长链上

安全性证明

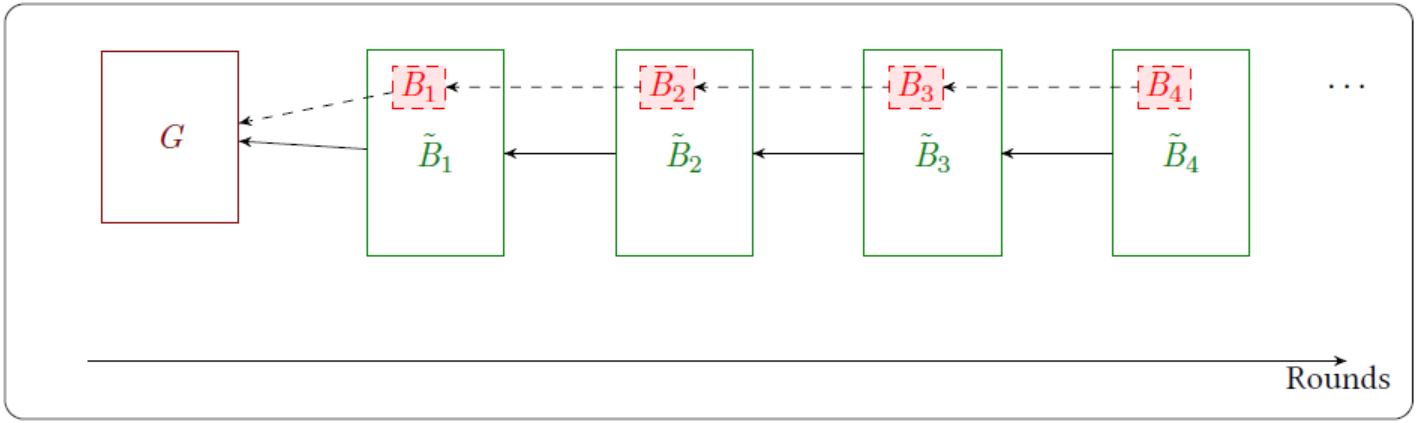
Common Prefix

- 几个观察：
  - 所有的诚实节点都会选择自己观察到的最长链
  - 链的增长速度满足前述Chain Growth特性
  - 诚实节点Token占优的假设下，恶意节点难以产生另外一条增长速度满足Chain Growth的链
- 推论：
  - 经过足够长的时间，仅有一条链增长速度达到Chain Growth

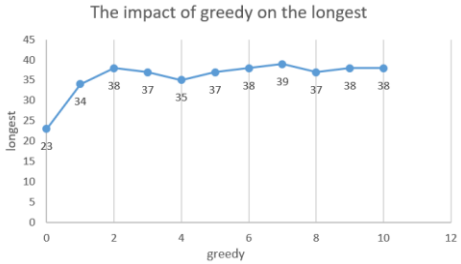
Extension to Main Blockchain

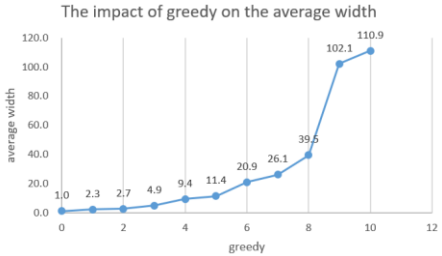
- Why we decouple core chain and main chain

$$\tilde{B}_{i+1} = \langle \mathsf{hash}(\tilde{B}_i), B_{i+1}, \tilde{X}_{i+1}, \tilde{\mathrm{PK}}, \tilde{\sigma} \rangle$$

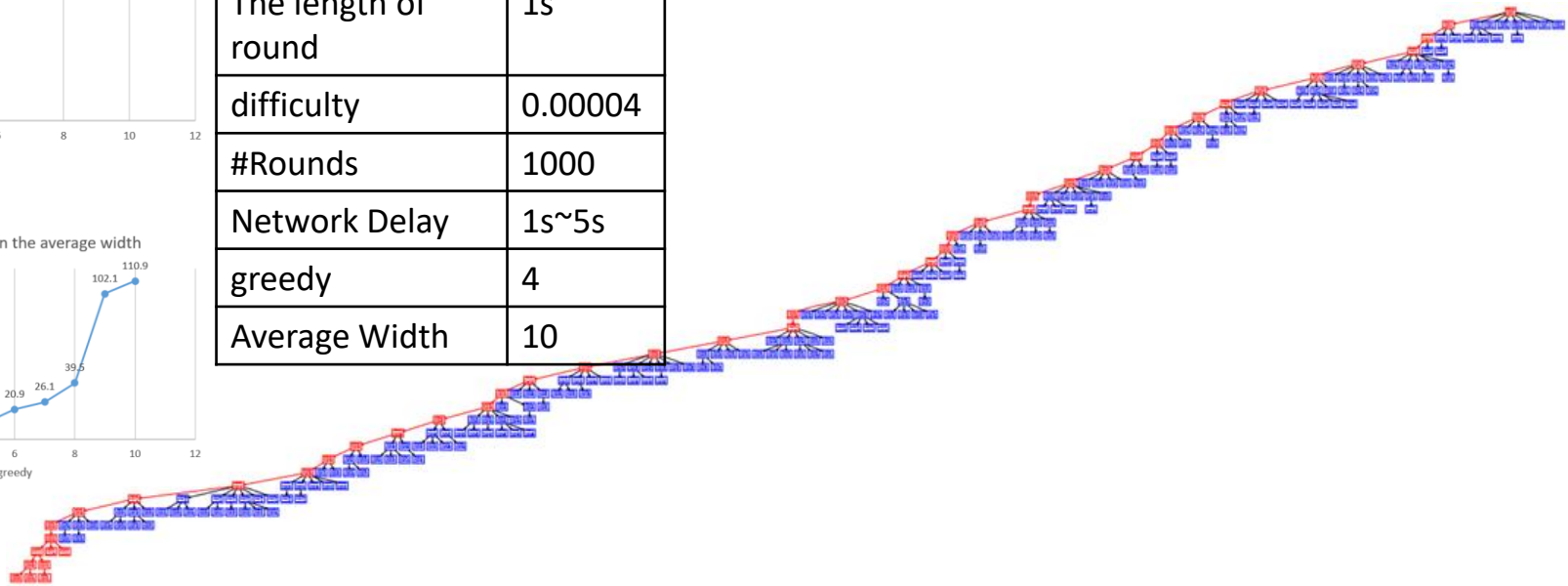


仿真运行结果





#node	1000
The length of round	1s
difficulty	0.00004
#Rounds	1000
Network Delay	1s~5s
greedy	4
Average Width	10



谢谢

期末作业

- 撰写一篇区块链相关的论文，内容可以是：
  - 某项技术的研究与分析
  - 某个应用场景的设计与创新
  - 某个区块链项目的分析
  - 某个论文的学习与理解
  - 某个相关的区块链开发工作
  - ……
- 必须原创，不能抄袭或者直接翻译