Does a Blockchain Need Altruism?

Roger Wattenhofer

ETH Zurich – Distributed Computing Group

Do You Trust the Miners?


IL BUONO
IL BRUTTO
IL CATTIVO
SERGIO LEONE
CLINT EASTWOOD • ELI WALLACH • LEE VAN CLEEF

Modeling Distributed Systems

Altruistic

Rational

Crash

Byzantine

Modeling Distributed Systems

Crash **R**ational **A**ltruistic **B**yzantine

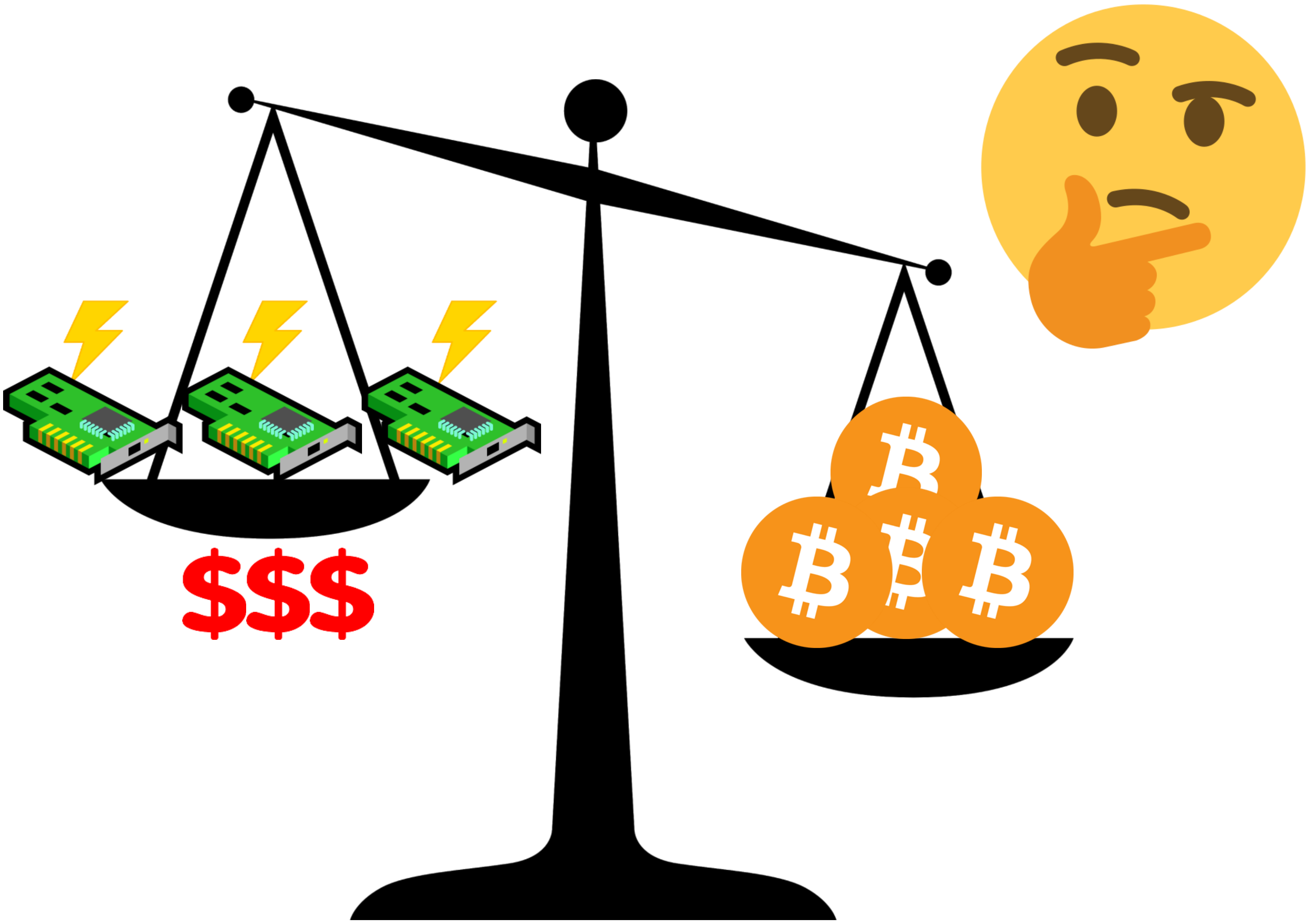
Who are the Miners?

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

“The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.”

Mining is a Rational Business



Mining is a Rational Business

ALTCOIN MINING MAY 21, 2018 21:50 CET

Japanese Cryptocurrency Monacoin Hit by Selfish Mining Attack

Selfish Mining Timeline

Majority is not Enough:
Bitcoin Mining is Vulnerable

Ittay Eyal and Emin Gün Sirer

Department of Computer Science, Cornell University
ittay.eyal@cornell.edu, egs@systems.cs.cornell.edu

2009 2010 2013 2018

Topic: Mining cartel attack (Read 31693 times)

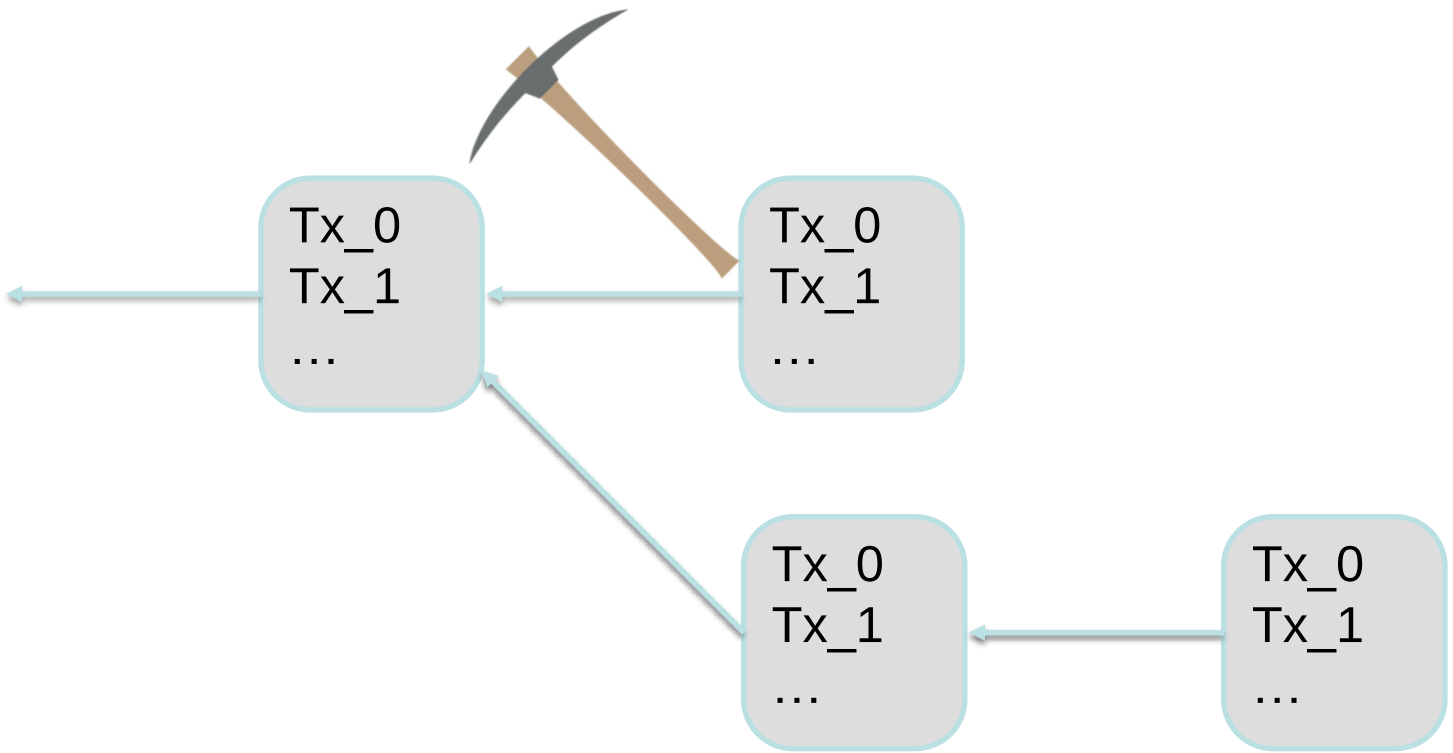
Mining cartel attack
December 12, 2010, 06:09:12 PM

I came across an idea that I think is worth dis
calling this a "mining cartel attack". I have nc
of describing it as I'm sure the thought has co
essential element of Bitcoin here, but I think t
are in place to stop this.

ALTCOIN MINING MAY 21, 2018 21:50 CET

Japanese Cryptocurrency Monacoin Hit by Selfish Mining Attack

What is Selfish Mining



Simpler

Analysis


Majority is not Enough: Bitcoin Mining is Vulnerable

Ittay Eyal and Emin Gün Sirer

Department of Computer Science, Cornell University
ittay.eyal@cornell.edu, egs@systems.cs.cornell.edu

Original Algorithm

Algorithm 1: Selfish-Mine

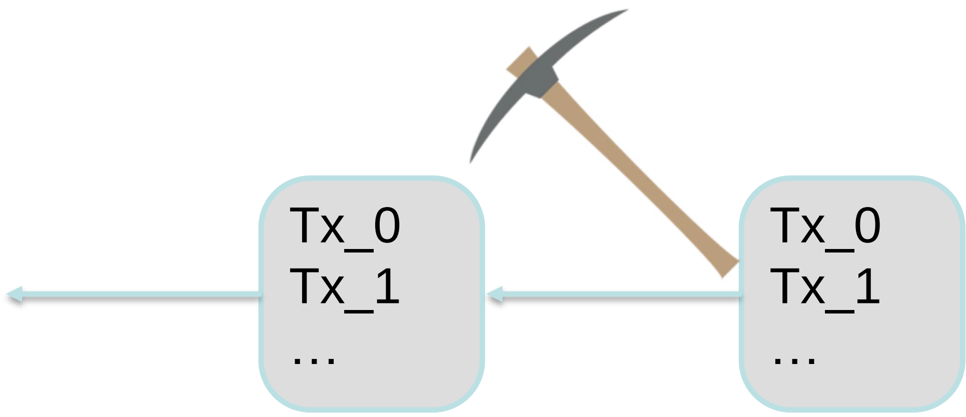
```
on Init
    public chain ← publicly known blocks
    private chain ← publicly known blocks
    privateBranchLen ← 0
    Mine at the head of the private chain.

on My pool found a block
    Δ_prev ← length(private chain) − length(public chain)
    append new block to private chain
    privateBranchLen ← privateBranchLen + 1
    if Δ_prev = 0 and privateBranchLen = 2 then          (Was tie with branch of 1)
        publish all of the private chain                (Pool wins due to the lead of 1)
        privateBranchLen ← 0
    Mine at the new head of the private chain.

on Others found a block
    Δ_prev ← length(private chain) − length(public chain)
    append new block to public chain
    if Δ_prev = 0 then
        private chain ← public chain                    (they win)
        privateBranchLen ← 0
    else if Δ_prev = 1 then
        publish last block of the private chain         (Now same length. Try our luck)
    else if Δ_prev = 2 then
        publish all of the private chain                (Pool wins due to the lead of 1)
        privateBranchLen ← 0
    else                                                (Δ_prev > 2)
        publish first unpublished block in private block.
    Mine at the head of the private chain.
```

Somewhat Simpler Algorithm

Algorithm 26.2 Selfish Mining

Idea: Mine secretly, without immediately publishing newly found blocks
Let d_p be the depth of the public blockchain
Let d_s be the depth of the secretly mined blockchain
if a new block b_p is published, i.e., d_p has increased by 1 **then**
    if $d_p > d_s$ **then**
        Start mining on that newly published block b_p
    else if $d_p = d_s$ **then**
        Publish secretly mined block b_s
        Mine on b_s and publish newly found block immediately
    else if $d_p = d_s - 1$ **then**
        Publish both secretly mined blocks
    end if
end if

$$d_p > d_s$$

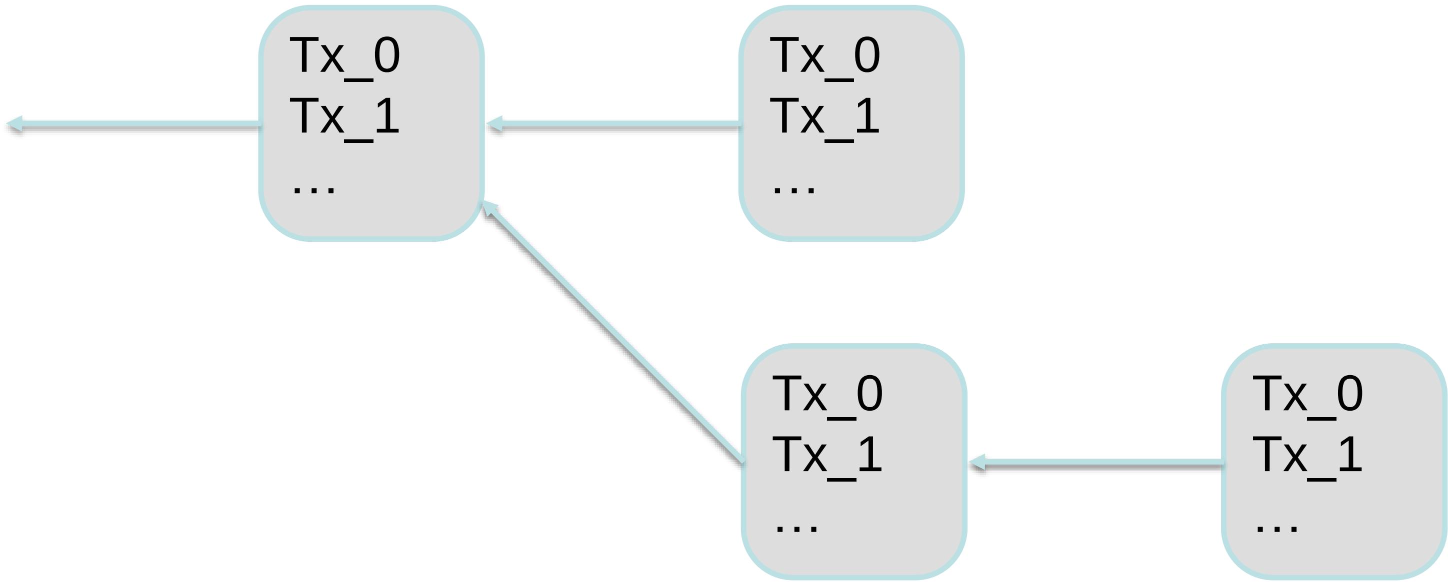


Somewhat Simpler Algorithm

Algorithm 26.2 Selfish Mining

```
Idea: Mine secretly, without immediately publishing newly found blocks
Let d_p be the depth of the public blockchain
Let d_s be the depth of the secretly mined blockchain
if a new block b_p is published, i.e., d_p has increased by 1 then
    if d_p > d_s then
        Start mining on that newly published block b_p
    else if d_p = d_s then
        Publish secretly mined block b_s
        Mine on b_s and publish newly found block immediately
    else if d_p = d_s − 1 then
        Publish both secretly mined blocks
    end if
end if
```

$$d_p = d_s - 1$$

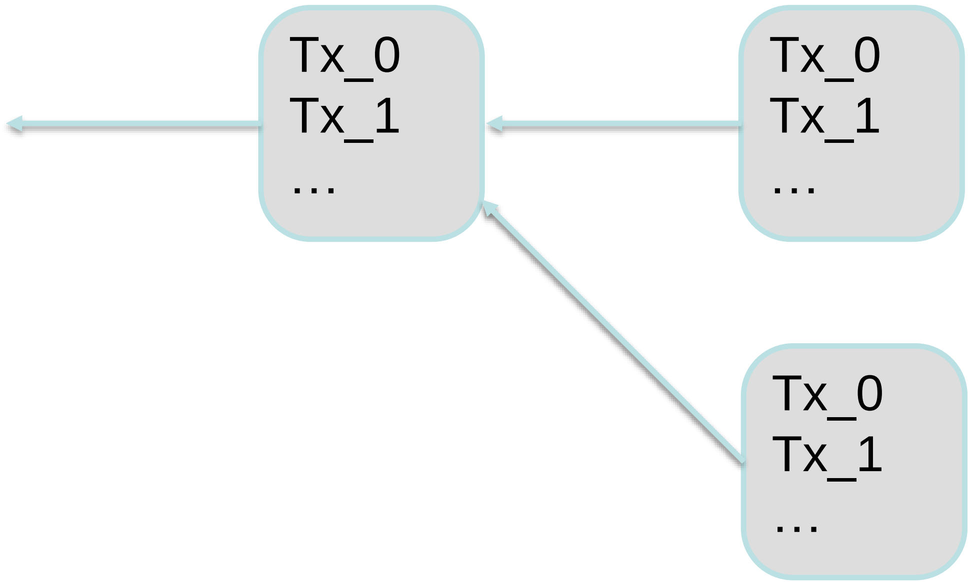


Somewhat Simpler Algorithm

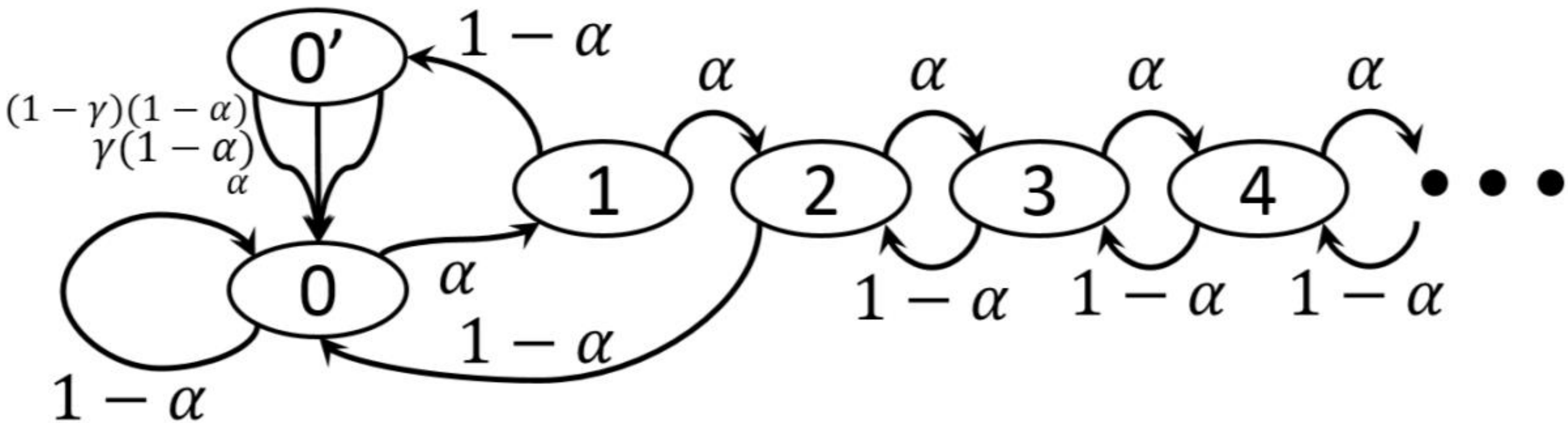
Algorithm 26.2 Selfish Mining

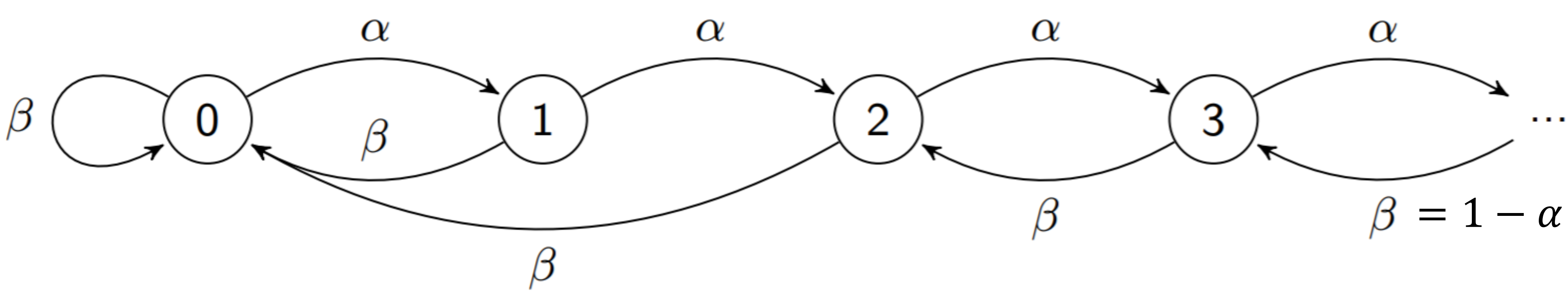
Idea: Mine secretly, without immediately publishing newly found blocks
Let d_p be the depth of the public blockchain
Let d_s be the depth of the secretly mined blockchain
if a new block b_p is published, i.e., d_p has increased by 1 **then**
    if $d_p > d_s$ **then**
        Start mining on that newly published block b_p
    else if $d_p = d_s$ **then**
        Publish secretly mined block b_s
        Mine on b_s and publish newly found block immediately
    else if $d_p = d_s - 1$ **then**
        Publish both secretly mined blocks
    end if
end if

$$d_p = d_s$$



State Machine (Original & Simpler)





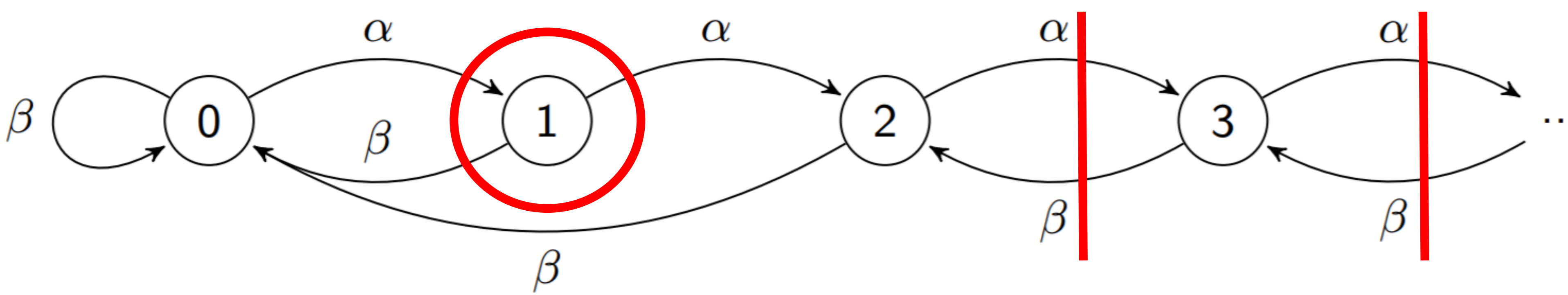
α: probability that selfish miner finds a block

Stationary Distribution

$$p_1 = \alpha p_0$$

$$\beta p_{i+1} = \alpha p_i, \text{ for all } i > 1$$

$$\text{and } 1 = \sum_i p_i.$$



Computation…

$$p_1 = \alpha p_0$$
$$\beta p_{i+1} = \alpha p_i, \text{ for all } i > 1$$
$$\text{and } 1 = \sum_i p_i.$$

Using $\rho = \alpha/\beta$, we express all terms of above sum with p_1:

$$1 = \frac{p_1}{\alpha} + p_1 \sum_{i \geq 0} \rho^i = \frac{p_1}{\alpha} + \frac{p_1}{1-\rho}, \text{ hence } p_1 = \frac{2\alpha^2 - \alpha}{\alpha^2 + \alpha - 1}$$

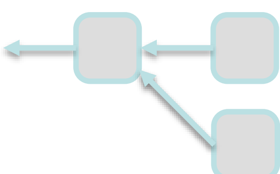
All β Transitions

$0 \rightarrow 0$: Block for honest miners

$i+1 \rightarrow i$: Block for selfish miner (for $i > 2$)

$2 \rightarrow 0$: **Two** blocks for selfish miner

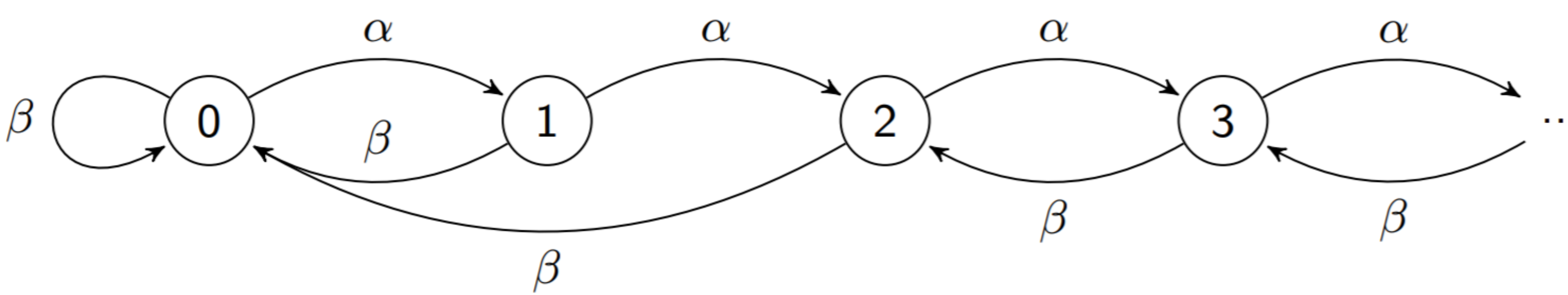
$1 \rightarrow 0$: Race who wins next block



with probability α **two** blocks for selfish miner

with probability $\beta(1-\gamma)$ **two** blocks for honest miners

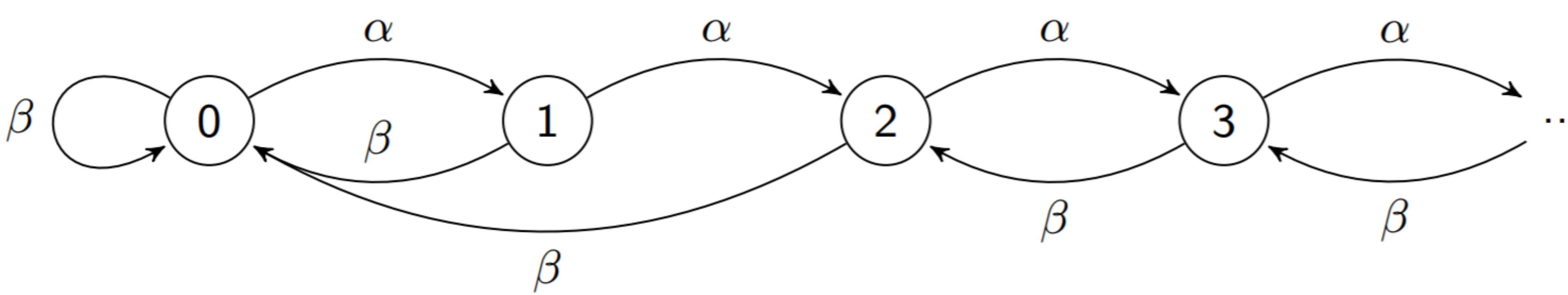
with probability $\beta\gamma$ **one block each**



γ: probability that honest miners append block to selfish miner's block (in race)

Ratio of Selfish Blocks in Chain

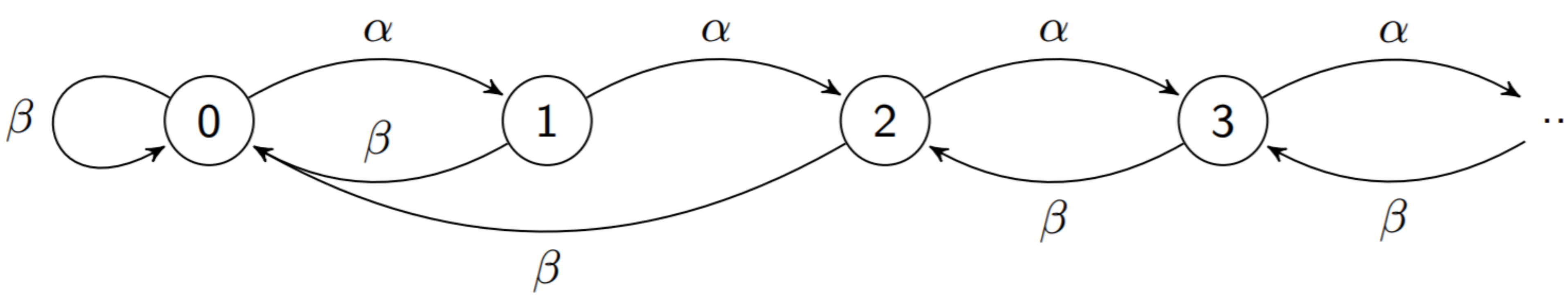
$$\frac{1 - p_0 + p_2 + \alpha p_1 - \beta(1-\gamma)p_1}{1 + p_1 + p_2}$$



γ: probability that honest miners append block to selfish miner's block (in race)

Selfish Miner Share

$$\frac{\alpha(1-\alpha)^2(4\alpha+\gamma(1-2\alpha))-\alpha^3}{1-\alpha(1+(2-\alpha)\alpha)}$$



Selfish Miner Share

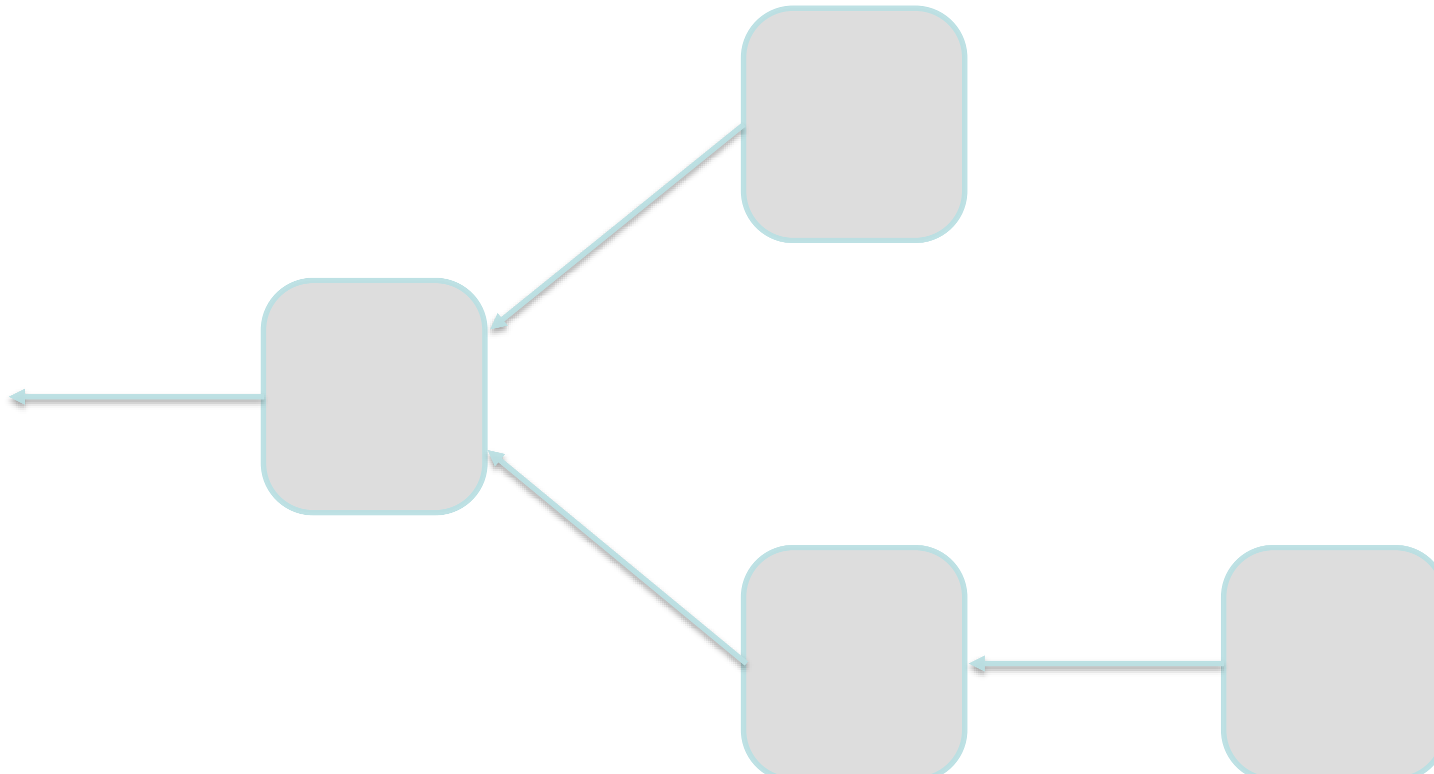
$$\frac{\alpha(1-\alpha)^2(4\alpha+\gamma(1-2\alpha))-\alpha^3}{1-\alpha(1+(2-\alpha)\alpha)}$$

$\gamma = 0$: break even at $\alpha = 1/3$
$\gamma = 0.5$: break even at $\alpha = 1/4$
$\gamma = 1$: break even at $\alpha > 0$

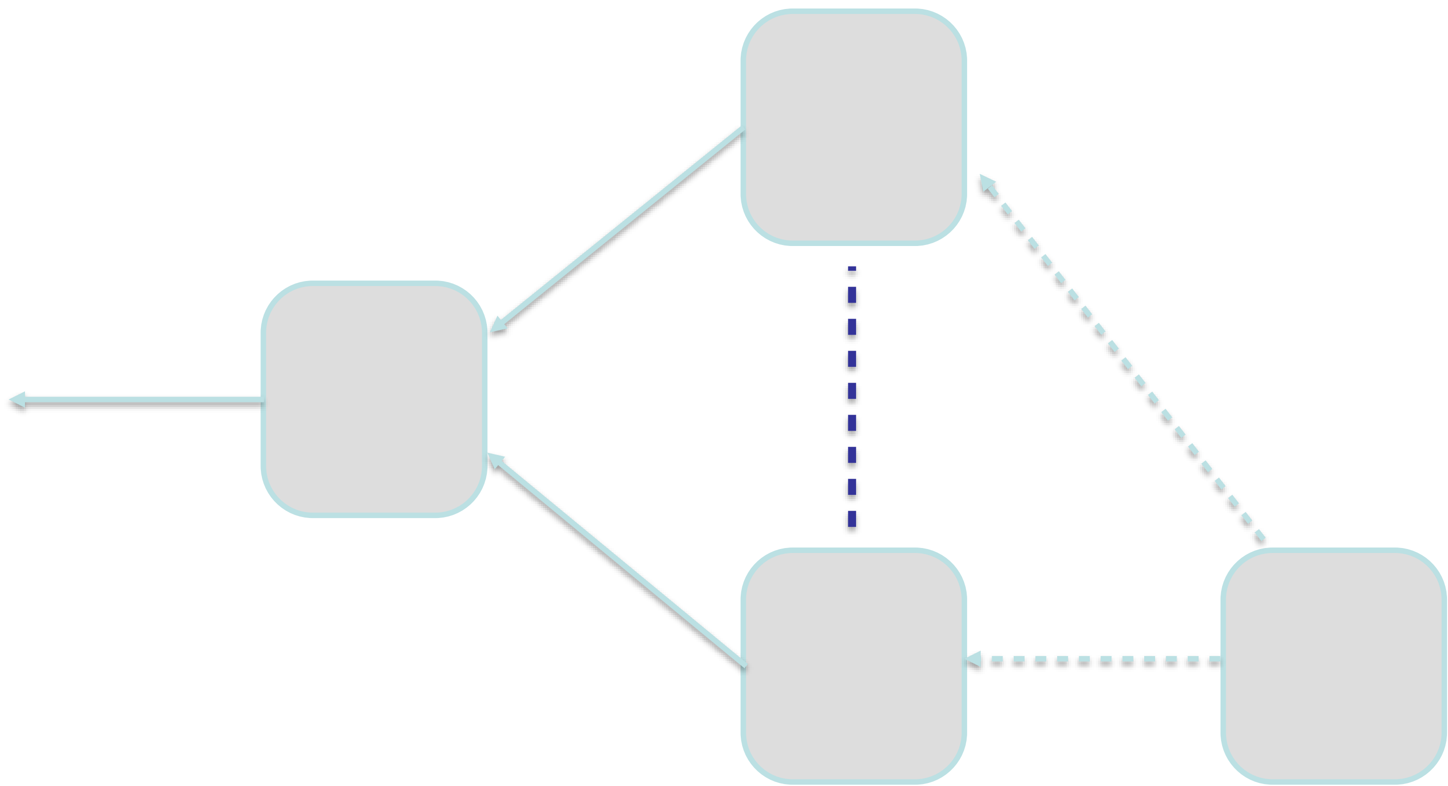
A Blockchain Without Altruism?

[Joint Work with Jakub Sliwinski]

Simple Chains Are Too Simple

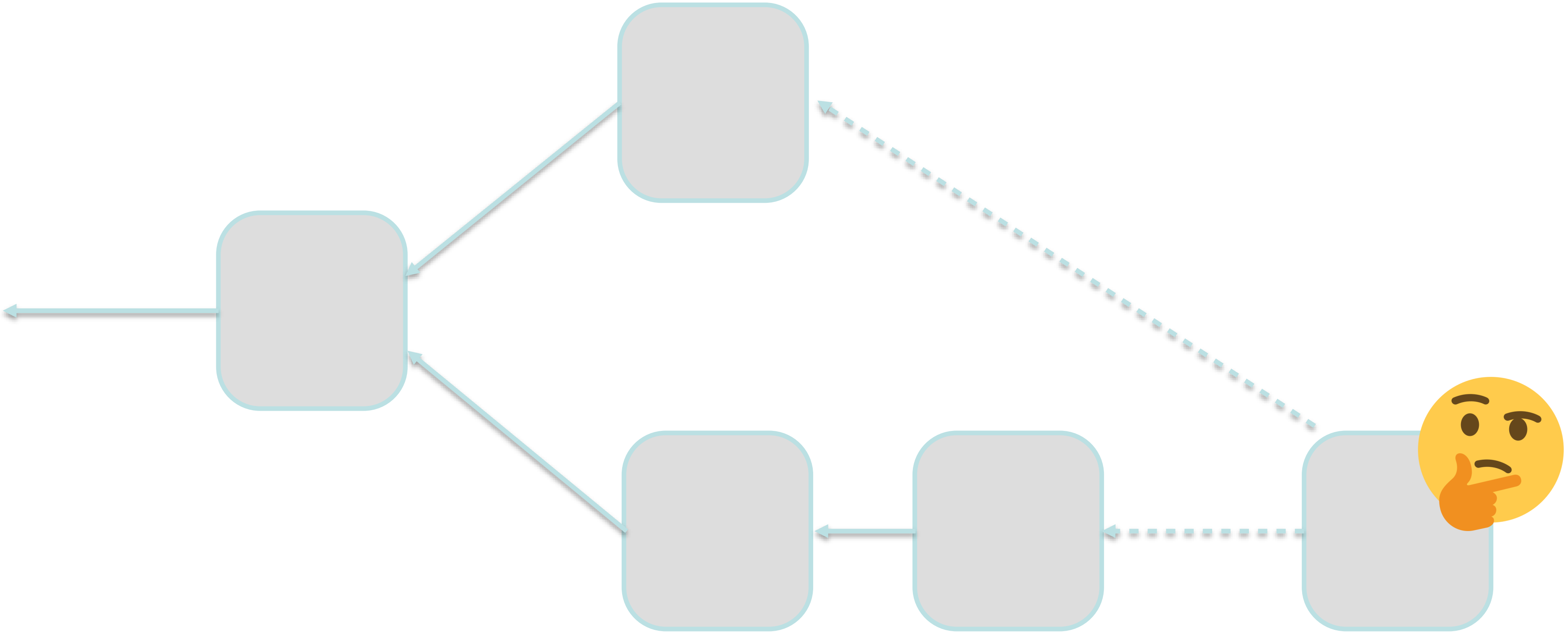


Better: Expose Competition

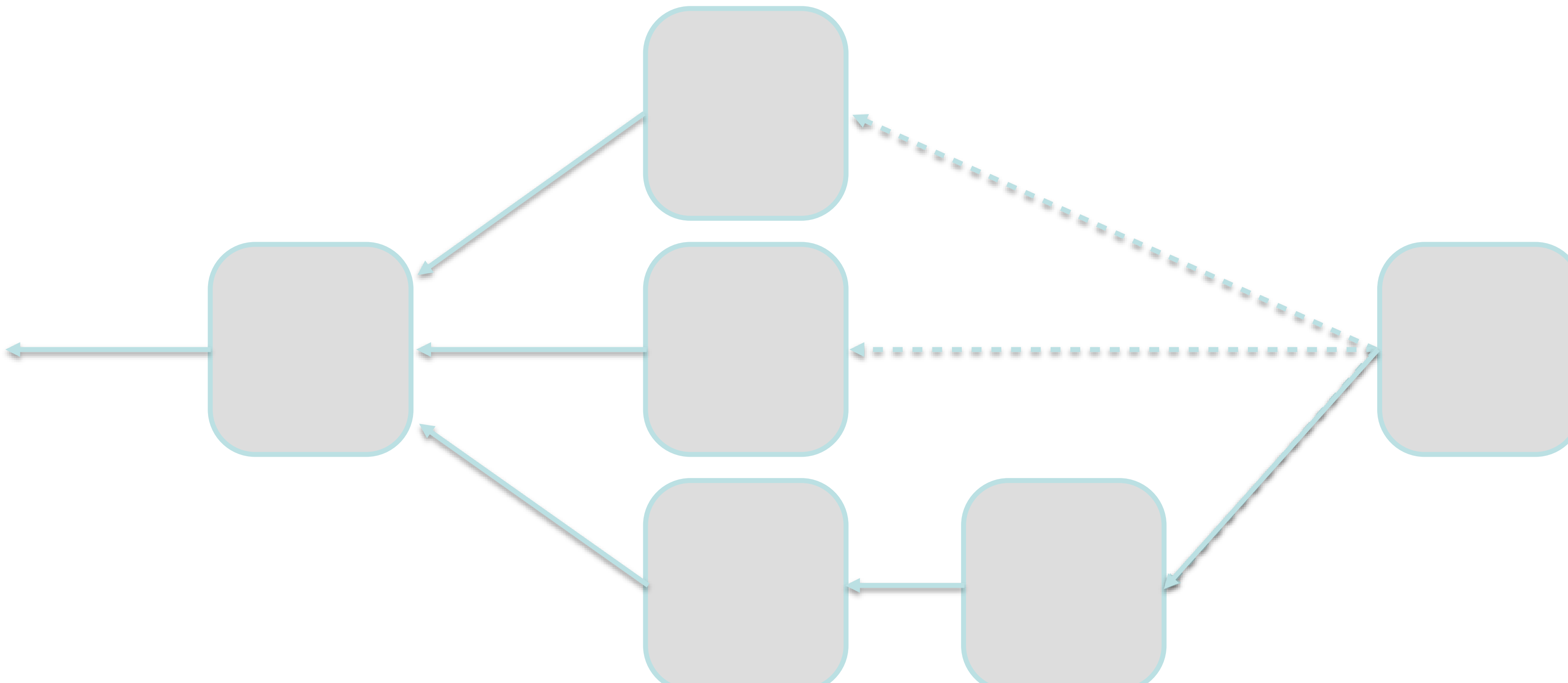


Our Rational Blockchain

Always Refer to All Childless Blocks

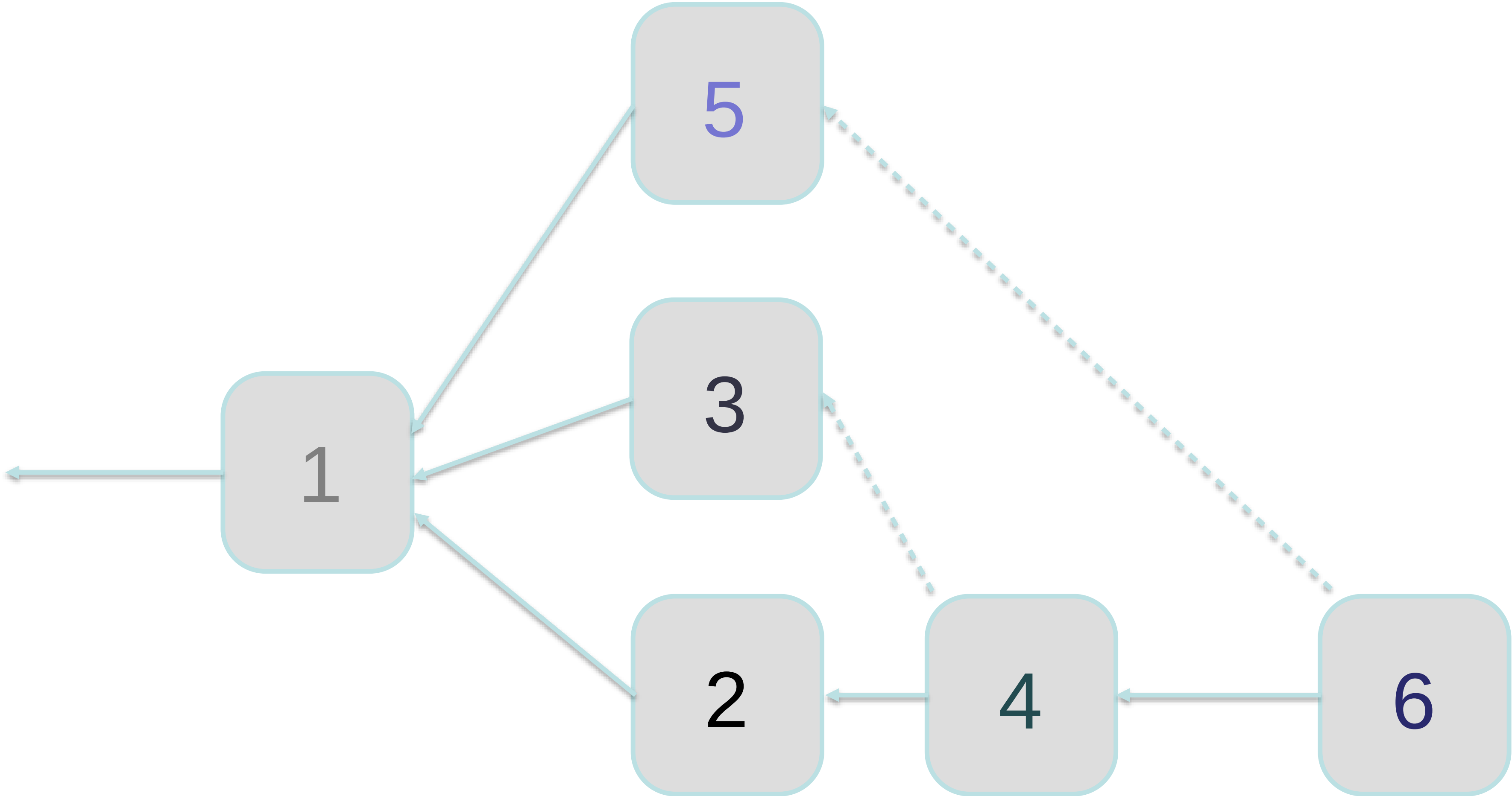


Only One Type of Reference



(Heaviest Reference is Your "Parent")

Block Ordering is Recursive



Inclusive Block Chain Protocols

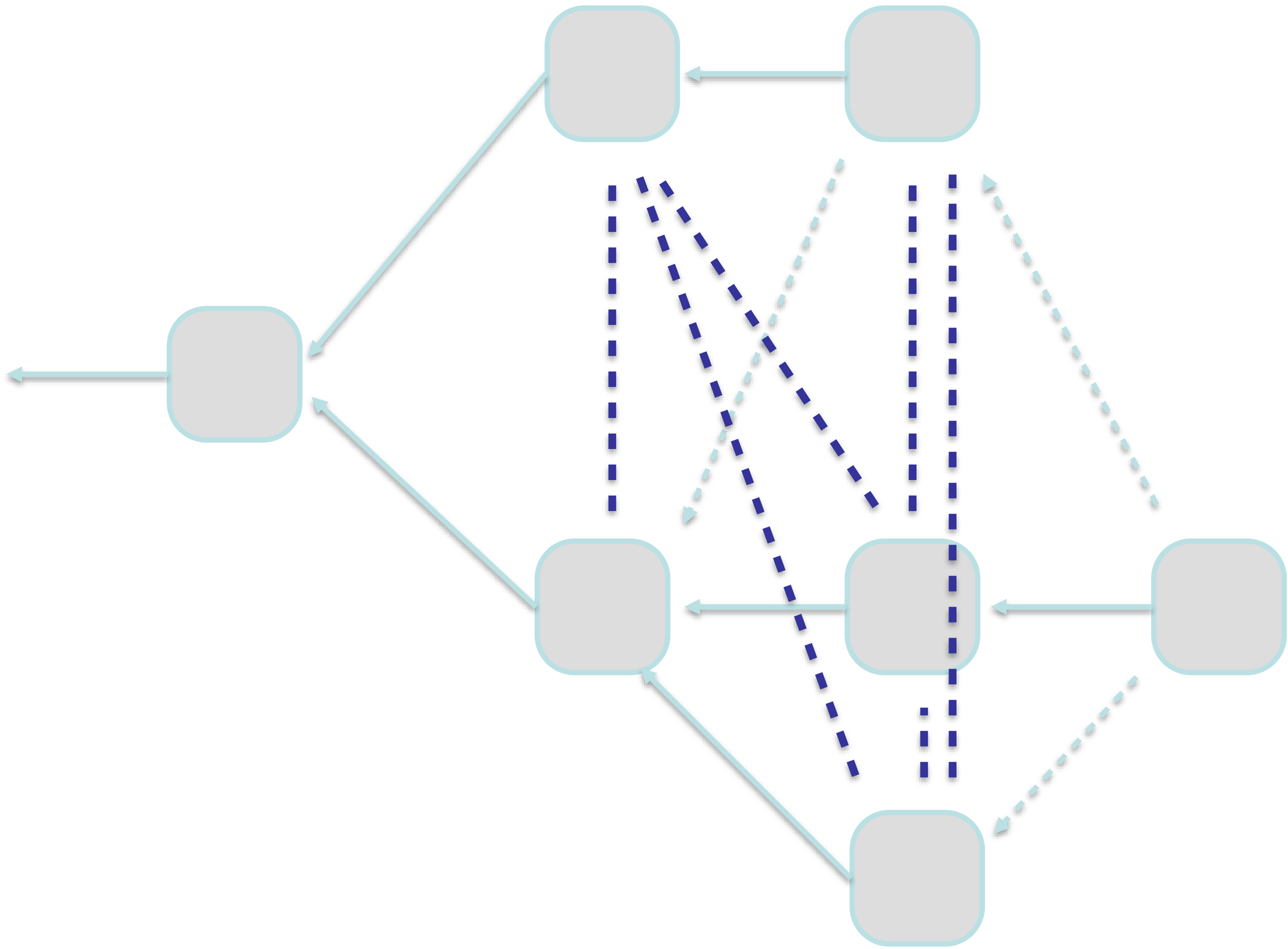
Yoad Lewenberg[1], Yonatan Sompolinsky[1], and Aviv Zohar[1,2]

Incentives

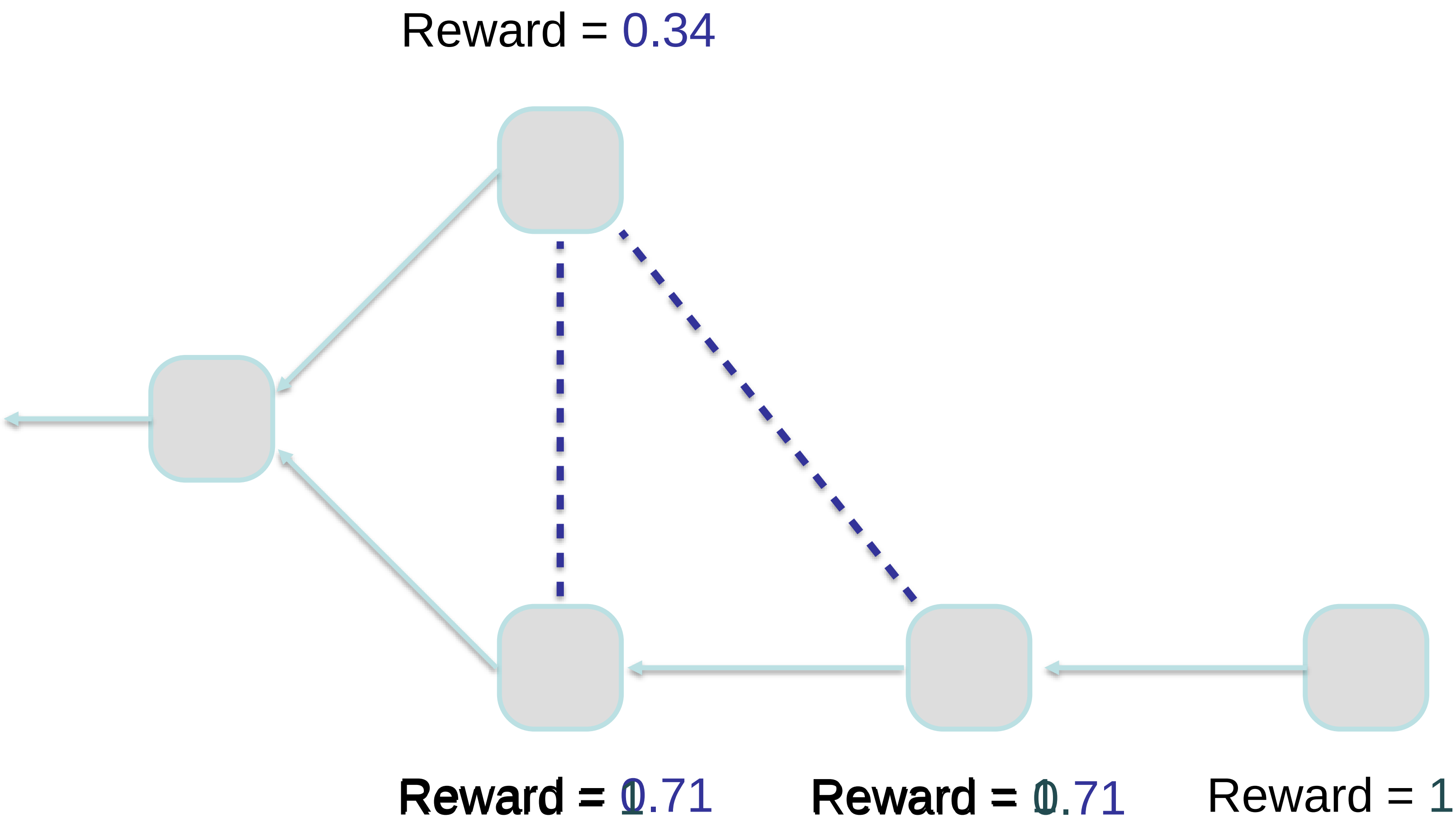
Why Miners Should Always Refer to All Childless Blocks?

Because of our Block Rewards!

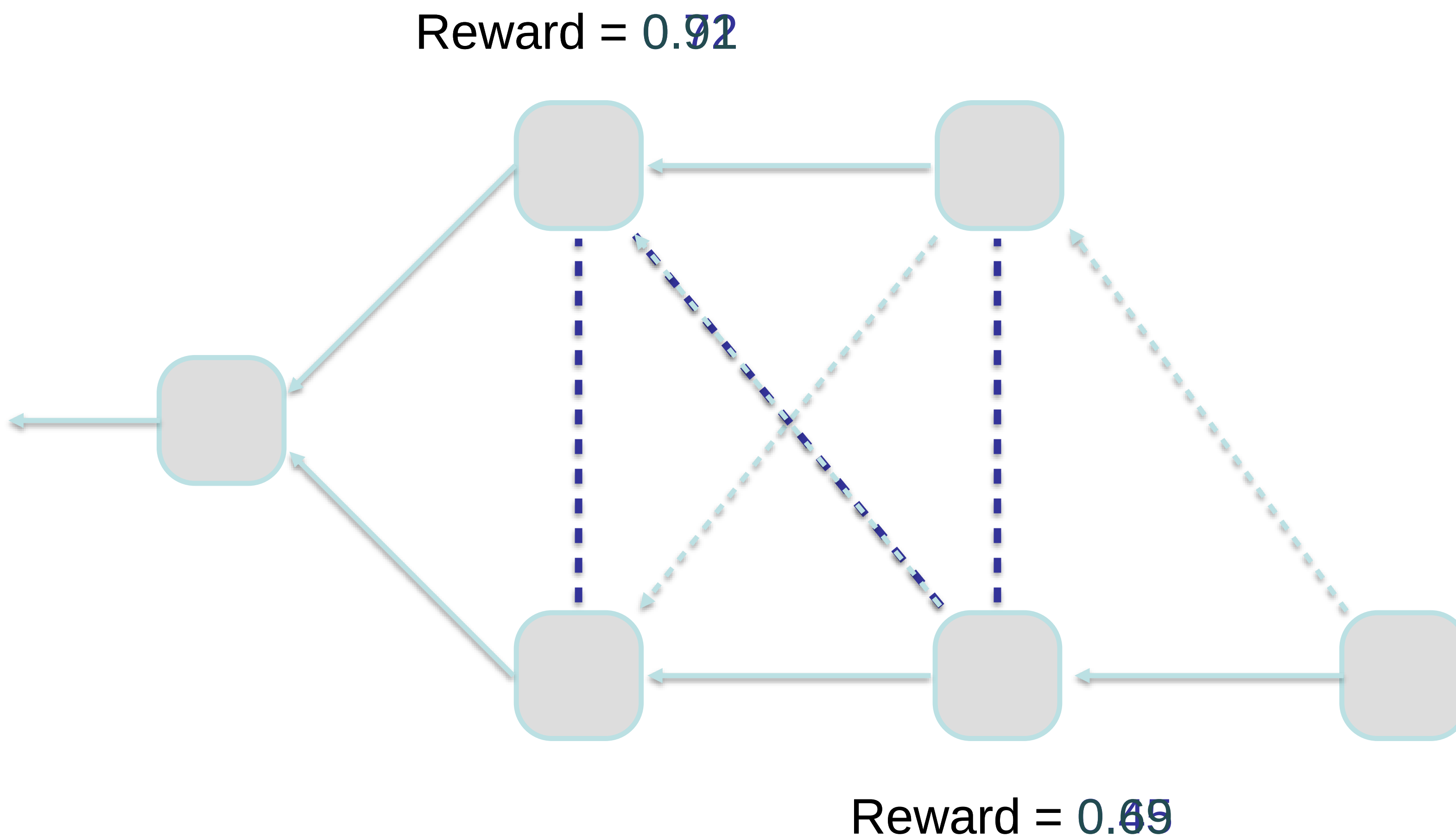
It’s Somewhat Complicated…



Motivating Block Rewards I



Motivating Block Rewards II



Our Solution

Definition 3 (Penalty Function). *Given are a pair of competing branches $\mathcal{B}_X$ and $\mathcal{B}_Y$ where $|\mathcal{B}_X| \geq |\mathcal{B}_Y|$, and a set E of edges between them, such that every block in $\mathcal{B}_Y$ has an incident edge. Then f is defined as follows:*

1. *f assigns a maximum penalty to all blocks in the smaller branch:*

$$\forall_{B \in \mathcal{B}_Y} : f(B) = 1.$$

2. *Each block's penalty is divided among incident edges:*

$$\left(\forall_{(A,B) \in E} : f((A,B)) \geq 0\right) \wedge \left(\forall_{B \in \mathcal{B}_X \cup \mathcal{B}_Y} : f(B) = \sum_{A \in E(B)} f((A,B))\right).$$

3. *Differences in penalties between blocks in the bigger branch are minimised:*

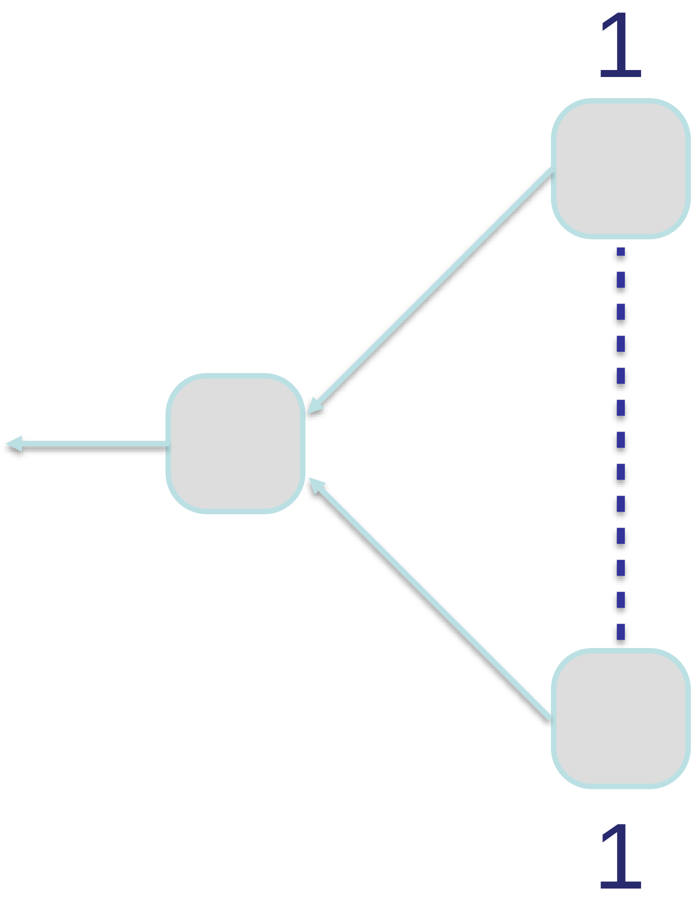
$$\forall_{B \in \mathcal{B}_Y} : \left(\left((A_1,B),(A_2,B) \in E \wedge f((A_1,B)) > 0\right) \implies f(A_1) \leq f(A_2)\right).$$

Definition 4 (Reward Scheme). *Creator of any block B receives an amount $r(B)$ of cryptocurrency to the address c_B. Any spending transaction from this address is valid only if included in a block C such that $LCA(B,C) > 2p$.*

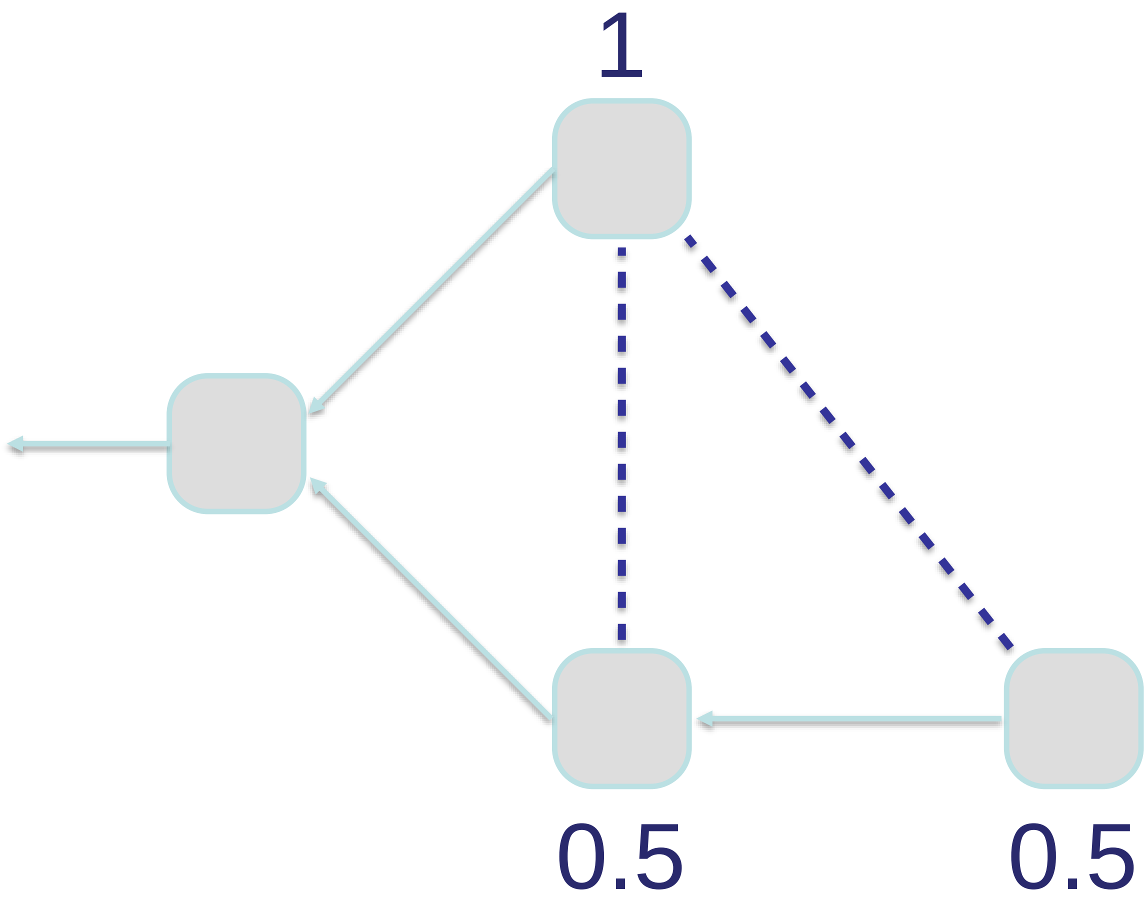
$$r(B) = R(1 - \max_{\mathcal{B}_X, \mathcal{B}_Y} f_{\mathcal{B}_X, \mathcal{B}_Y, E}(B)) + \sum_{tx \in \mathcal{T}_B} fee_B(tx)$$

Here, R is the base block reward, and E consists of edges from the conflict graph of G. $fee_B(tx)$ is discussed in section 3.1.

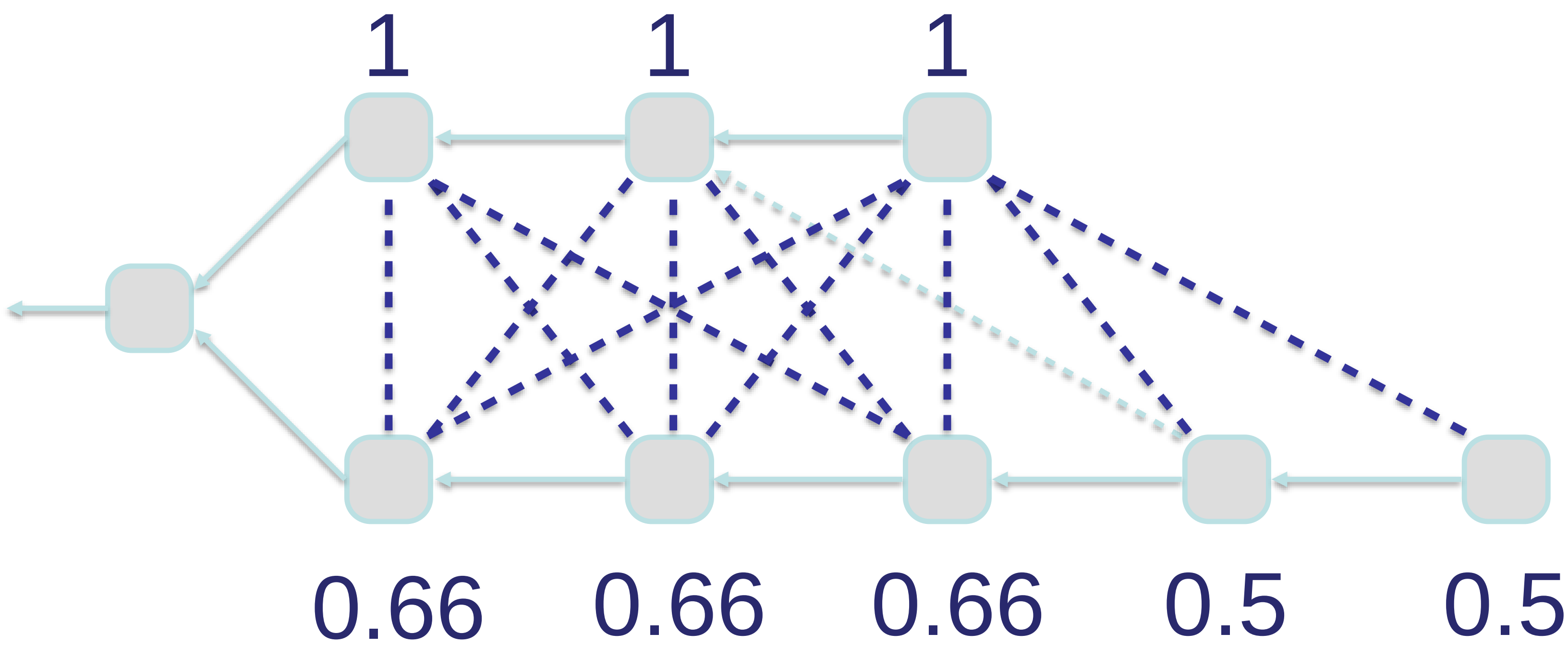
Block Penalty Example



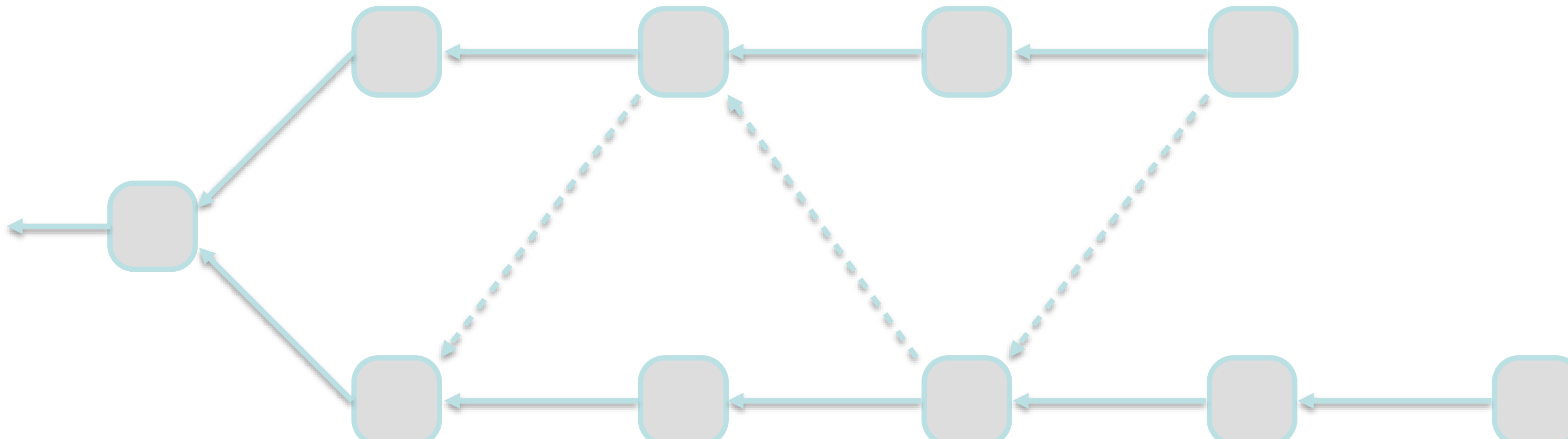
Block Penalty Example



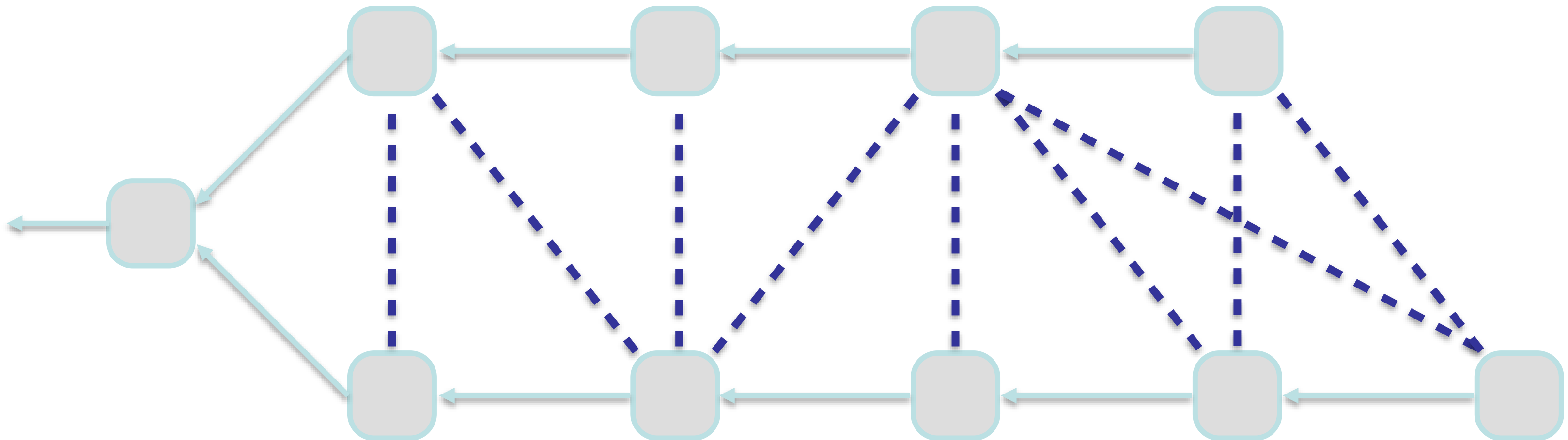
Block Penalty Example



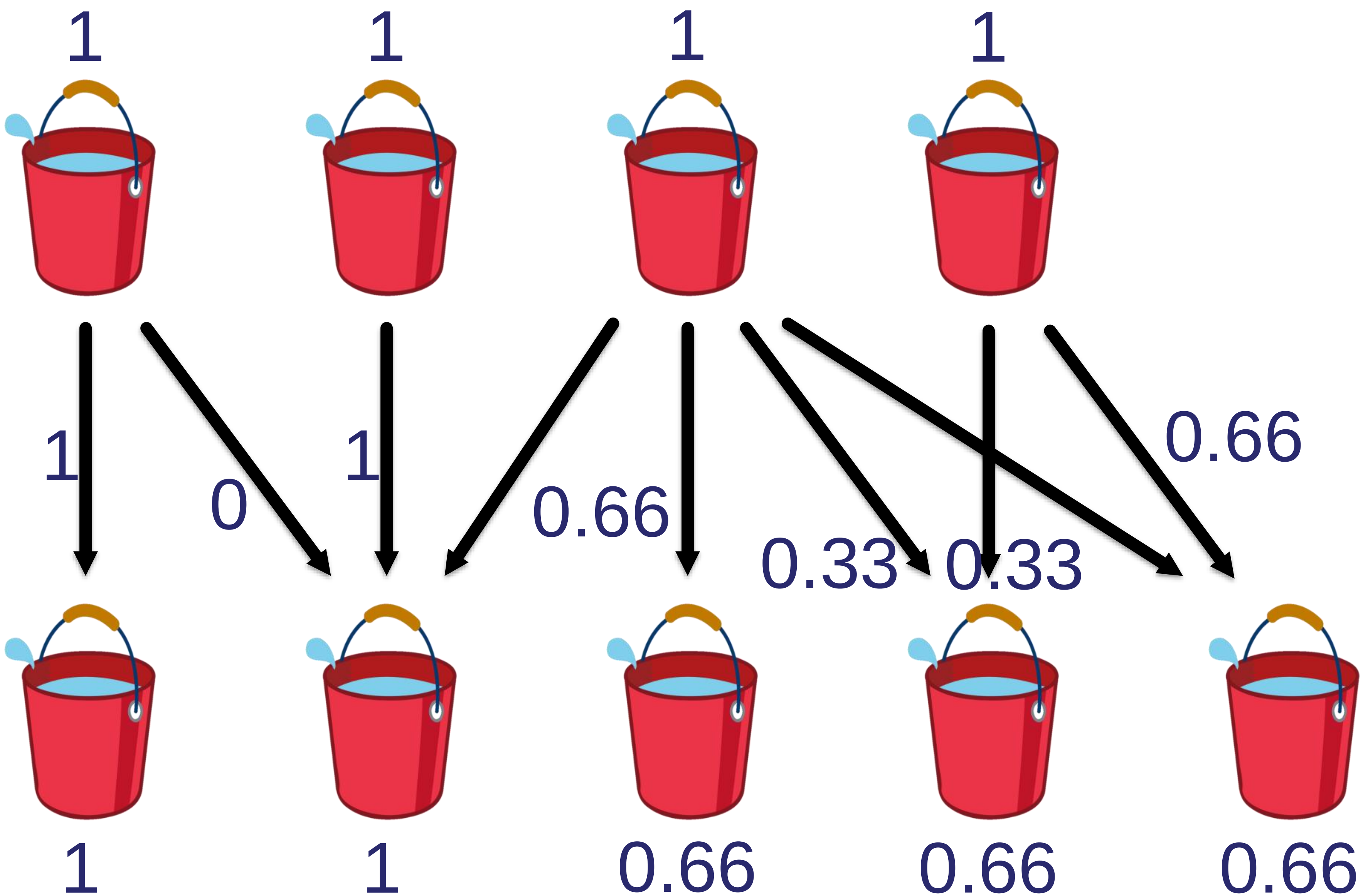
The Penalty Algorithm



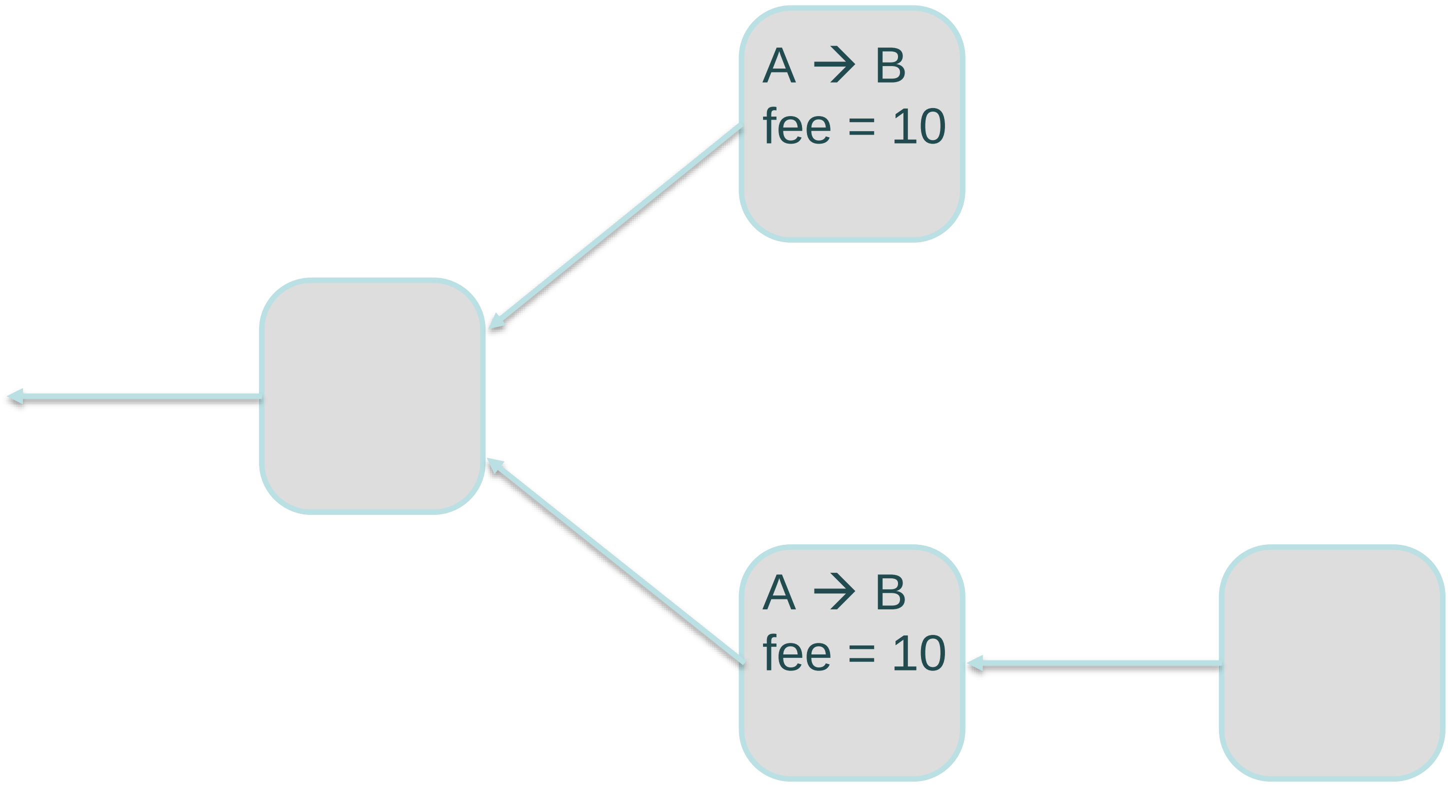
The Penalty Algorithm

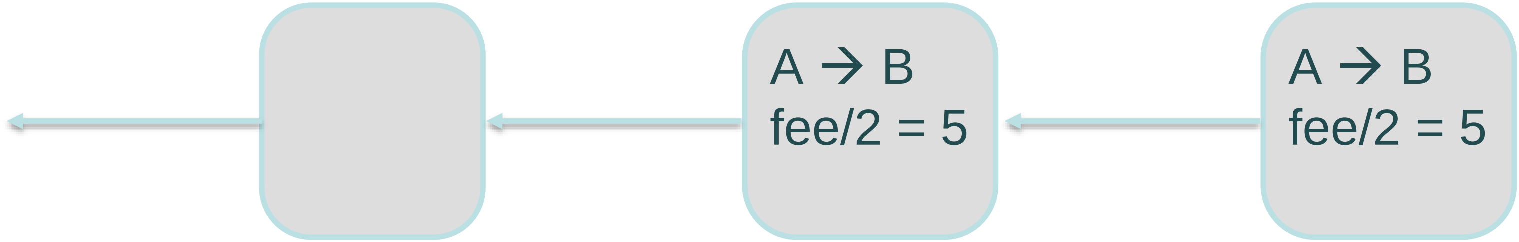


The Penalty Algorithm



Transaction Fees


A → B
fee = 10
A → B
fee = 10


A → B
fee/2 = 5
A → B
fee/2 = 5

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

“The system is secure as long as
rational ~~honest~~ nodes collectively control more
CPU power than any cooperating
group of attacker nodes.”

Thank You!

Questions & Comments?

Thanks to Jakub Sliwinski