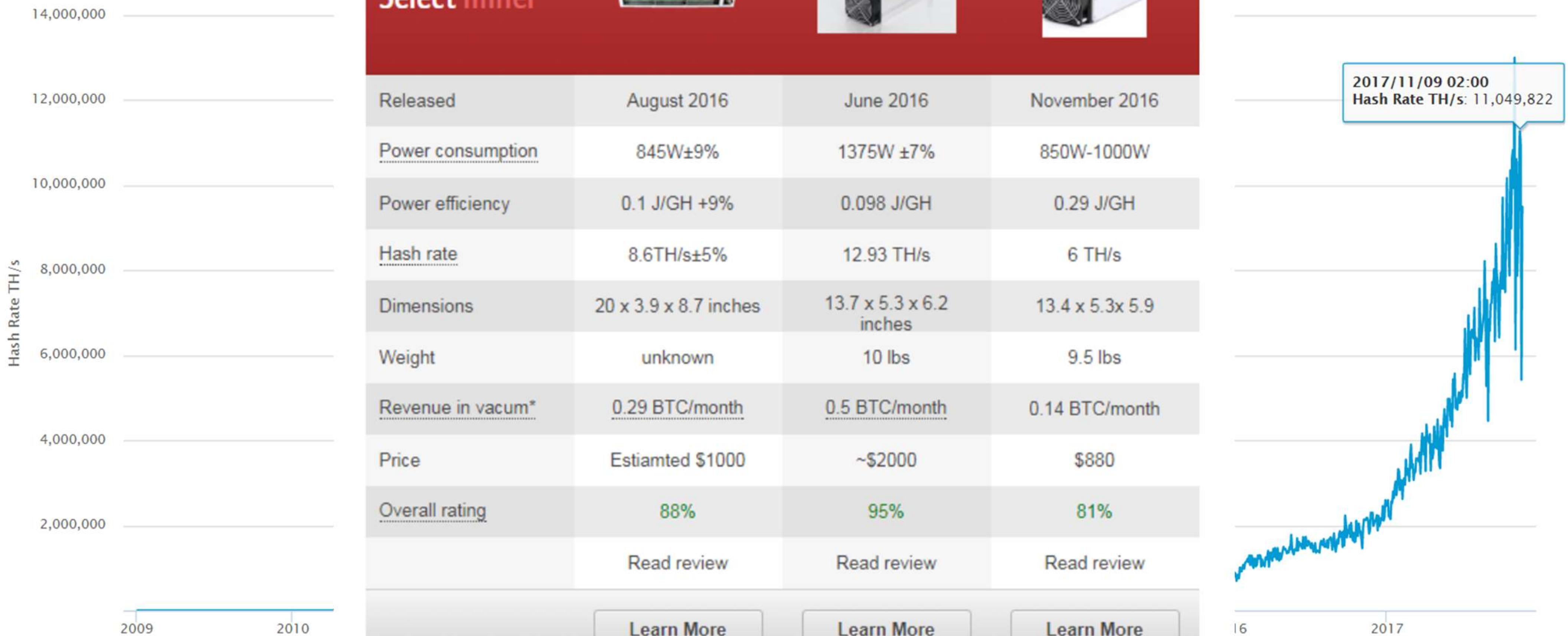
Incentive Compatibility of Bitcoin Mining Pool Reward Functions

By:
Okke Schrijvers
Joseph Bonneau
Dan Boneh
Tim Roughgarden

Yoni Asulin



Select miner	Antminer R4 NEW	AntMiner S9 Best!	Avalon 7
			
Released	August 2016	June 2016	November 2016
Power consumption	845W±9%	1375W ±7%	850W-1000W
Power efficiency	0.1 J/GH +9%	0.098 J/GH	0.29 J/GH
Hash rate	8.6TH/s±5%	12.93 TH/s	6 TH/s
Dimensions	20 x 3.9 x 8.7 inches	13.7 x 5.3 x 6.2 inches	13.4 x 5.3x 5.9
Weight	unknown	10 lbs	9.5 lbs
Revenue in vacum*	0.29 BTC/month	0.5 BTC/month	0.14 BTC/month
Price	Estiamted $1000	~$2000	$880
Overall rating	88%	95%	81%
	Read review	Read review	Read review
	Learn More	Learn More	Learn More

The model

- Single mining pool of n miners
  - Each miner contributes $\alpha_i \in [0,1]$ mining power to the pool
- No other pools or solo miners
- The pool having all mining power: $\alpha_P = \sum_{i=1}^{n} \alpha_i = 1$

- The pool manager
  - divides the reward among the n miners according to a reward function
  - Does not know the computational power each miner contributed to the pool
  - How to estimate this?

- Miners report full solutions and shares they found to the pool manager
- Once a full solution is reported, the reward is shared among miners, and the game restarts

Shares vs. full solution

- The SHA-256 hash function
  - Collision resistant
  - Uniform distribution - changing a single bit outputs a totally different hash value
  - Th
  - Ea

```
$ python hash_example.py

I am Satoshi Nakamoto0 => a80a81401765c8eddee25df36728d732...
I am Satoshi Nakamoto1 => f7bc9a6304a4647bb41241a677b5345f...
I am Satoshi Nakamoto2 => ea758a8134b115298a1583ffb80ae629...
I am Satoshi Nakamoto3 => bfa9779618ff072c903d773de30c99bd...
I am Satoshi Nakamoto4 => bce8564de9a83c18c31944a66bde992f...
I am Satoshi Nakamoto5 => eb362c3cf3479be0a97a20163589038e...
I am Satoshi Nakamoto6 => 4a2fd48e3be420d0d28e202360cfbaba...
I am Satoshi Nakamoto7 => 790b5a1349a5f2b909bf74d0d166b17a...
I am Satoshi Nakamoto8 => 702c45e5b15aa54b625d68dd947f1597...
I am Satoshi Nakamoto9 => 7007cf7dd40f5e933cd89fff5b791ff0...
I am Satoshi Nakamoto10 => c2f38c81992f4614206a21537bd634a...
I am Satoshi Nakamoto11 => 7045da6ed8a914690f087690e1e8d66...
I am Satoshi Nakamoto12 => 60f01db30c1a0d4cbce2b4b22e88b9b...
I am Satoshi Nakamoto13 => 0ebc56d59a34f5082aaef3d66b37a66...
I am Satoshi Nakamoto14 => 27ead1ca85da66981fd9da01a8c6816...
I am Satoshi Nakamoto15 => 394809fb809c5f83ce97ab554a2812c...
I am Satoshi Nakamoto16 => 8fa4992219df33f50834465d3047429...
I am Satoshi Nakamoto17 => dca9b8b4f8d8e1521fa4eaa46f4f0cd...
I am Satoshi Nakamoto18 => 9989a401b2a3a318b01e9ca9a22b0f3...
I am Satoshi Nakamoto19 => cda56022ecb5b67b2bc93a2d764e75f...
```

- Proof of wo … e α_i)
- A share is a

Source: Mastering Bitcoin, *Andreas M. Antonopoulos*

Exponential distribution

- the time it takes for a miner to find a share
  - Random variable with exponential distribution with parameter α_i

- The exponential distribution: let X be such a random variable with parameter $\lambda = \alpha_i$
  - PDF:

$$f(x) = \begin{cases} \alpha_i e^{-\alpha_i x}, & x \geq 0 \\ 0 \quad\quad\quad , & x < 0 \end{cases}$$

  - Expected value:

$$\mathbb{E}[X] = \frac{1}{\alpha_i}$$

- Each share is also a full solution with prob. $\frac{1}{D}$ (dice analogy)
- The pool manager sees on average D shares for every full solution

Miner's view

- Reward function - the only way in which miners get any payout
- Miner's dilemma (report immediately or wait)
- Leading question:

if individual miners are interested in maximizing their expected utility, is their behavior optimal for the pool as a group?

The reward function

$$R: \mathbb{N}^n \to [0,1]^n$$

- Convention: $\mathbb{N} \equiv \mathbb{N} \cup \{0\}$
- We call a vector $\boldsymbol{b} \in \mathbb{N}^n$ a history transcript (single round)
- In response to a reward function, miners choose strategy (<u>when</u> to report shares)

- What properties a good reward function should have?
    - Proportional payments
    - Report a full solution immediately (incentive compatibility)
    - Budget balanced

A word on notation

- history transcript: $\boldsymbol{b} = (b_1, \dots, b_n) \in \mathbb{N}^n$
- #share reports of miner $\boldsymbol{i}$ (per round): $\boldsymbol{b}_i \equiv [\boldsymbol{b}]_i \equiv b_i$
- L_1 norm: $\|\boldsymbol{b}\|_1 = \sum_{i=1}^{n} b_i$
- The $\boldsymbol{i}$-th component of the reward function:

$$R: \mathbb{N}^n \to [0,1]^n$$

$$R = (R_1, \dots, R_n)$$

Where $\forall 1 \leq i \leq n, \quad R_i: \mathbb{N}^n \to [0,1]$

- R_i is the reward function of miner $\boldsymbol{i}$

Proportional payments (def.)

A reward function R provides proportional payments, if for each miner $\boldsymbol{i}$:

$$\mathbb{E}_{\boldsymbol{b}}[R_i(\boldsymbol{b})] = \alpha_i$$

- Expectation over all $\boldsymbol{b} \in \mathbb{N}^n$

Incentive compatibility (def.)

A reward function R is incentive compatible if for every miner, the best strategy (given this particular R) is to report full solutions immediately.

budget-balanced (def.)

A reward function R is (γ, δ)-budget balanced if for all $\boldsymbol{b} \in \mathbb{N}^{\boldsymbol{n}}$:

$$\gamma \leq \sum_{i=1}^{n} R_i\,(\boldsymbol{b}) \leq \delta$$

- $(\gamma, 1)$ = ?
- Ideally we want $(1{,}1)$- budget balanced reward functions

Main goal: find reward functions that satisfy all three conditions

common reward functions

- Proportional reward $R^{(prop)}$:
  - Divide the reward based on %shares each miner reported

$$R_i^{(prop)}(\boldsymbol{b}) = \frac{\boldsymbol{b_i}}{\|\boldsymbol{b}\|_1} = \frac{\boldsymbol{b_i}}{\sum_{i=1}^{n} b_i}$$

- Proportional, budget balanced

Problems:

- Miners might prefer to leave the pool at some point (Miller et al.)
- Not incentive compatible
  - What if player I has been unlucky and reported a lower num. of shares than his comp. power α_i?

common reward functions

- Pay-per-share $R^{(pps)}$
    - Pays a fixed amount for every share that is reported

$$R_i^{(pps)}(\boldsymbol{b}) = \frac{\boldsymbol{b_i}}{D} = \boldsymbol{b_i} \cdot \frac{1}{D}$$

To delay or not to delay?

Or: when would we prefer to report immediately

- Assume that at time $\boldsymbol{t}$ miner $\boldsymbol{i}$ finds a **full solution.**
- At this point, $\boldsymbol{b}_t$ shares have been reported to the pool operator
- Assume that miner $\boldsymbol{i}$ waits for another $\boldsymbol{d} \in \mathbb{N}$ shares before reporting full solution. What would be his expected reward:

$$\mathbb{E}_{(\boldsymbol{b}\ s.t\ \|\boldsymbol{b}\|_1=d)}[R_i(\boldsymbol{b}_t + \boldsymbol{b})] = \sum_{(\boldsymbol{b}\ s.t\ \|\boldsymbol{b}\|_1=d)} \Pr(seeing\ \boldsymbol{b}) \cdot R_i(\boldsymbol{b}_t + \boldsymbol{b})$$

- Reminder: for a random variable $X\colon \Omega \to \mathbb{R}$, $\mathbb{E}[X] = \sum_{k \in Im(X)} k \cdot p_x(k)$
- Here we can define $X(\boldsymbol{b}) = R_i(\boldsymbol{b}_t + \boldsymbol{b})$

To delay or not to delay?

Or: when would we prefer to report immediately

- If miner i reports immediately (upon finding a full solution):

His reward will be:

$$R_i(\boldsymbol{b}_t) + d \cdot \frac{\mathbb{E}_{\boldsymbol{b}}[R_i(\boldsymbol{b})]}{\mathbb{E}_{\boldsymbol{b}}[\|\boldsymbol{b}\|_1]} = R_i(\boldsymbol{b}) + d \cdot \frac{\mathbb{E}_{\boldsymbol{b}}[R_i(\boldsymbol{b})]}{\sum_{k=1}^{\infty} k\left(1-\frac{1}{D}\right)^{k-1} \cdot \frac{1}{D}} =$$

$$= R_i(\boldsymbol{b}_t) + \frac{d}{D} \cdot \mathbb{E}_{\boldsymbol{b}}[R_i(\boldsymbol{b})]$$

To delay or not to delay?

Or: when would we prefer to report immediately

Overall, reporting a full solution immediately will be more profitable than delaying for d shares iff:

$$\sum_{(\boldsymbol{b}\ s.t\ \|\boldsymbol{b}\|_1 = d)} \Pr(seeing\ \boldsymbol{b}) \cdot (R_i(\boldsymbol{b}_t + \boldsymbol{b}) - (R_i(\boldsymbol{b}_t)) \le \frac{d}{D} \cdot \mathbb{E}_{\boldsymbol{b}}[R_i(\boldsymbol{b})]$$

equiv. condition for incentive comp.

Lemma:

A reward function R is incentive compatible if and only if for every $\boldsymbol{i}$, $\{\alpha_i\}_{i=1}^{n}$, $\boldsymbol{b}_t$, D:

$$\sum_{j=1}^{n} \alpha_j \cdot \left(R_i\left(\boldsymbol{b}_t + \boldsymbol{e}_j\right) - R_i(\boldsymbol{b}_t)\right) \leq \frac{\mathbb{E}_{\boldsymbol{b}}[R_i(\boldsymbol{b})]}{D}$$

- i.e, to determine the incentive compatibility of a reward function, we only need to see if it is profitable to delay reporting for a single additional share.

Pay-per-share is not incentive comp.

- reminder: $R_i^{(pps)}(\boldsymbol{b}) \ = \frac{\boldsymbol{b_i}}{D} = \boldsymbol{b_i} \cdot \frac{1}{D}$
- Eqiv. Condition: $\sum_{j=1}^{n} \alpha_j \cdot \left(R_i\left(\boldsymbol{b}_t + \boldsymbol{e}_j\right) - R_i(\boldsymbol{b}_t)\right) \leq \frac{\mathbb{E}_{\boldsymbol{b}}[R_i(\boldsymbol{b})]}{D}$

- $\sum_{j=1}^{n} \alpha_j \cdot \left(R_i^{(pps)}\left(\boldsymbol{b}_t + \boldsymbol{e}_j\right) - R_i^{(pps)}(\boldsymbol{b}_t)\right) = \alpha_i \cdot \frac{b_i + 1 - b_i}{D} = \frac{\alpha_i}{D}$
- $\frac{\mathbb{E}_{\boldsymbol{b}}[R_i^{(pps)}\boldsymbol{b})]}{D} = \frac{\mathbb{E}_{\boldsymbol{b}}[b_i/D]}{D} = \frac{\alpha_i}{D}$

Problems with $R_i^{(pps)}$

- Not incentive compatible
- Only $(\frac{1}{D}, \infty)$-budget balanced:
  - If a full solution is the first share that is reported:

$$\sum_{j=1}^{n} R_i^{(pps)}(\boldsymbol{b}) = \frac{1}{D}$$

  - The number of reported share is not bounded
  - The pool operator pays no more than it takes only in expectation
  - Needs large reserves to keep the prob. of bankruptcy low

The IC Reward Function

- in addition to a count of the shares per miner we also includes the identity of the discoverer of the full solution:

$$R_i^{(ic)}(\boldsymbol{b}, s) = \frac{b_i}{\max\{\|\boldsymbol{b}\|_1, D\}} + 1_{\{i=s\}} \cdot (1 - \frac{\|\boldsymbol{b}\|_1}{\max\{\|\boldsymbol{b}\|_1, D\}})$$

- if $\|\boldsymbol{b}\|_1 \geq D$: this is just the proportional function
- if $\|\boldsymbol{b}\|_1 < D$:
    - Each share gets a fixed reward of $1/D$ (like in pay-per-share)
    - The remainder of the reward goes to the discoverer of the full solution
    - No money is left on the table

The IC Reward Function

- $R^{(ic)}$ provides proportional payments
- $R^{(ic)}$ is incentive compatible
- $R^{(ic)}$ is $(1,1)$-budget balanced

$R^{(ic)}$ is (1,1)-budget balanced:

- Reminder: $R_i^{(ic)}(\boldsymbol{b}, s) = \frac{b_i}{\max\{\|\boldsymbol{b}\|_1, D\}} + 1_{\{i=s\}} \cdot (1 - \frac{\|\boldsymbol{b}\|_1}{\max\{\|\boldsymbol{b}\|_1, D\}})$
- If $\|\boldsymbol{b}\|_1 < D$ the total payout is:

$$\sum_{i=1}^{n} \frac{b_i}{D} + \left(1 - \sum_{i=1}^{n} \frac{b_i}{D}\right) = 1$$

- If $\|\boldsymbol{b}\|_1 \geq D$ the total payout is:

$$\sum_{i=1}^{n} \frac{b_i}{\|\boldsymbol{b}\|_1} = 1$$

$R^{(ic)}$ provides a Steady Payment Stream

- We look at the fraction of the reward given to the discoverer of the full solution:

$$\sum_{k=1}^{D-1} \Pr(\|\boldsymbol{b}\|_1 = k) \cdot \left(1 - \frac{k}{D}\right) = \sum_{k=1}^{D-1} \frac{1}{D} \cdot \left(1 - \frac{k}{D}\right)^{k-1} \cdot \left(1 - \frac{k}{D}\right) = \left(1 - \frac{1}{D}\right)^{D} \leq e^{-1}$$

$$\Rightarrow 1 - e^{-1} \approx 0.63$$

- The majority of the reward is paid out for shares and not full solutions
- hence the majority of the pool's rewards are redistributed in a steady stream

Pay-Per-Last-N-Shares (PPLNS)

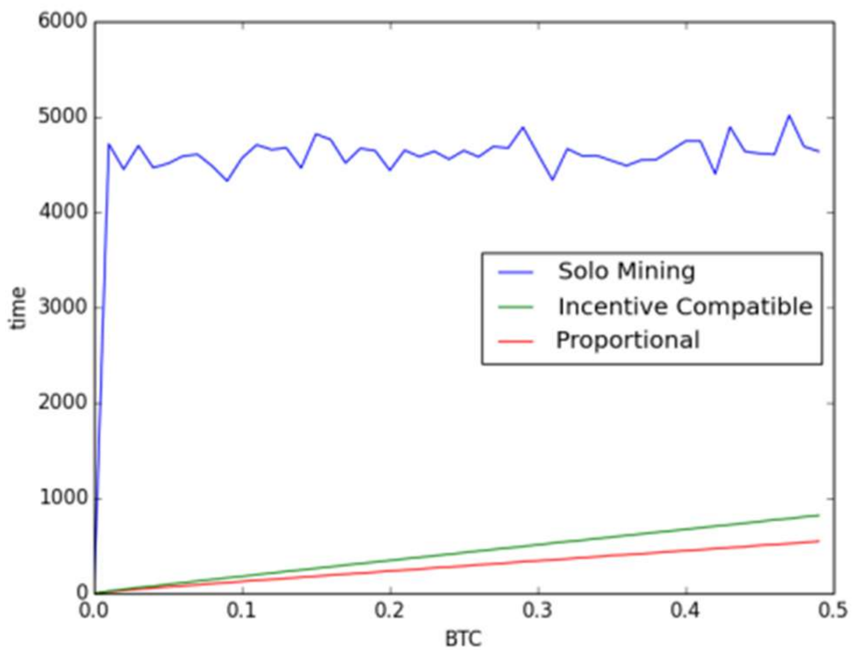
$$R_i^{(pplns)}(\boldsymbol{s}) = \frac{\#\{s_j \mid s_j \in \boldsymbol{s} \wedge s_j = i\}}{N}$$

- Widely used in practice
- maintains a history of reported shares that spans multiple rounds
  - So what happens in round T is no longer isolated from what happens in round $T + 1$
- takes the order of reported shares into account:
  - maintains a sliding window of length N and divides the reward proportionally over these N shares

Comparison by simulation

- the simulation goes as follows:
  - Assume 1000 miners, each with $\alpha_i = 0.001$
  - $D = 1{,}000{,}000$
  - Reward for a full solution is normalized to 1 BTC
  - A unit of time – a time to find a full solution (10 min)
  - We look at the most unluckiest miners (1% of miners) – how long they have to wait for a given amount

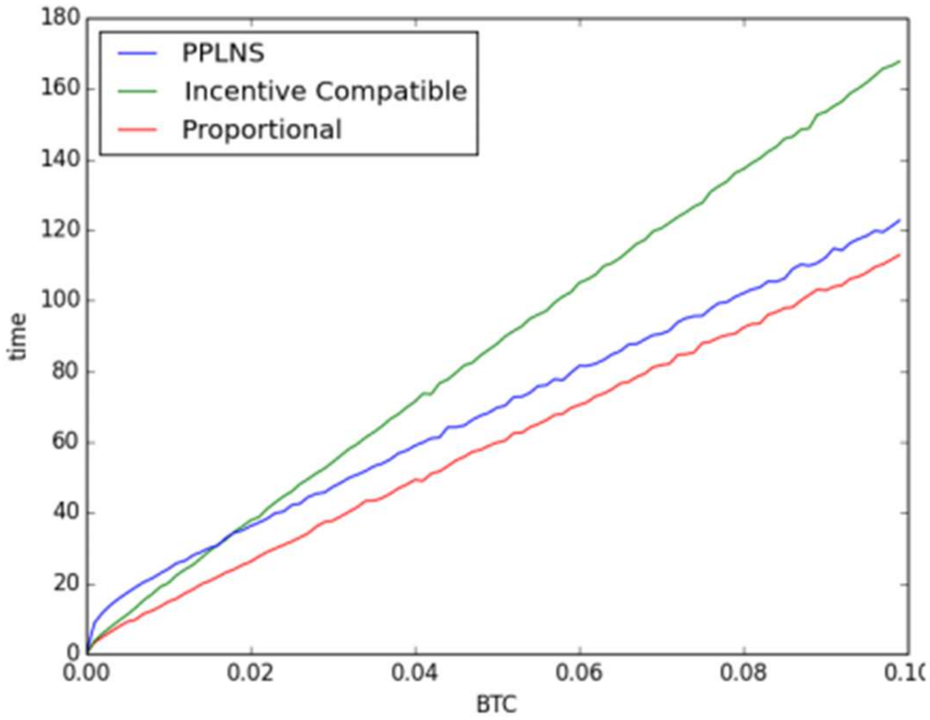
Comparison by simulation



(a) 99^{th} percentile time to earn rewards

- Even though in expectation a solo miner finds a solution once in 1000 rounds, in the worst 1% of cases he has to wait 4500 rounds
- incentive compatible scheme takes a bit longer to reach the same target than the proportional scheme – not all reward is shared according to reported shares
- The difference is up to a small factor

Comparison with PPLNS



- the incentive compatible (IC) scheme performs worse by a small multiplicative factor
- the PPLNS scheme performs worse by a small additive factor
- for small Bitcoin targets it would be faster to use the IC reward function, whereas for larger target the PPLNS reward function performs better

Conclusion

- Tradeoff of using PPLNS or IC to proportional reward:
- Using IC or PPLNS we pay with a modest delay in the time it would take miners to reach a minimal amount of bitcoin with high probability
- But in return we get a scheme in which it is obvious for miners what the most profitable strategy for them is

Thank you!

Source: http://coinalert.eu/maxthumb/20160922044332.jpg