# Algorithmic Competition and Coordination Failure between Cryptocurrencies

October 2021

Kohei Kawaguchi (HKUST)

Shunya Noda (University of Tokyo)

# Emergence of Algorithm-Driven Markets

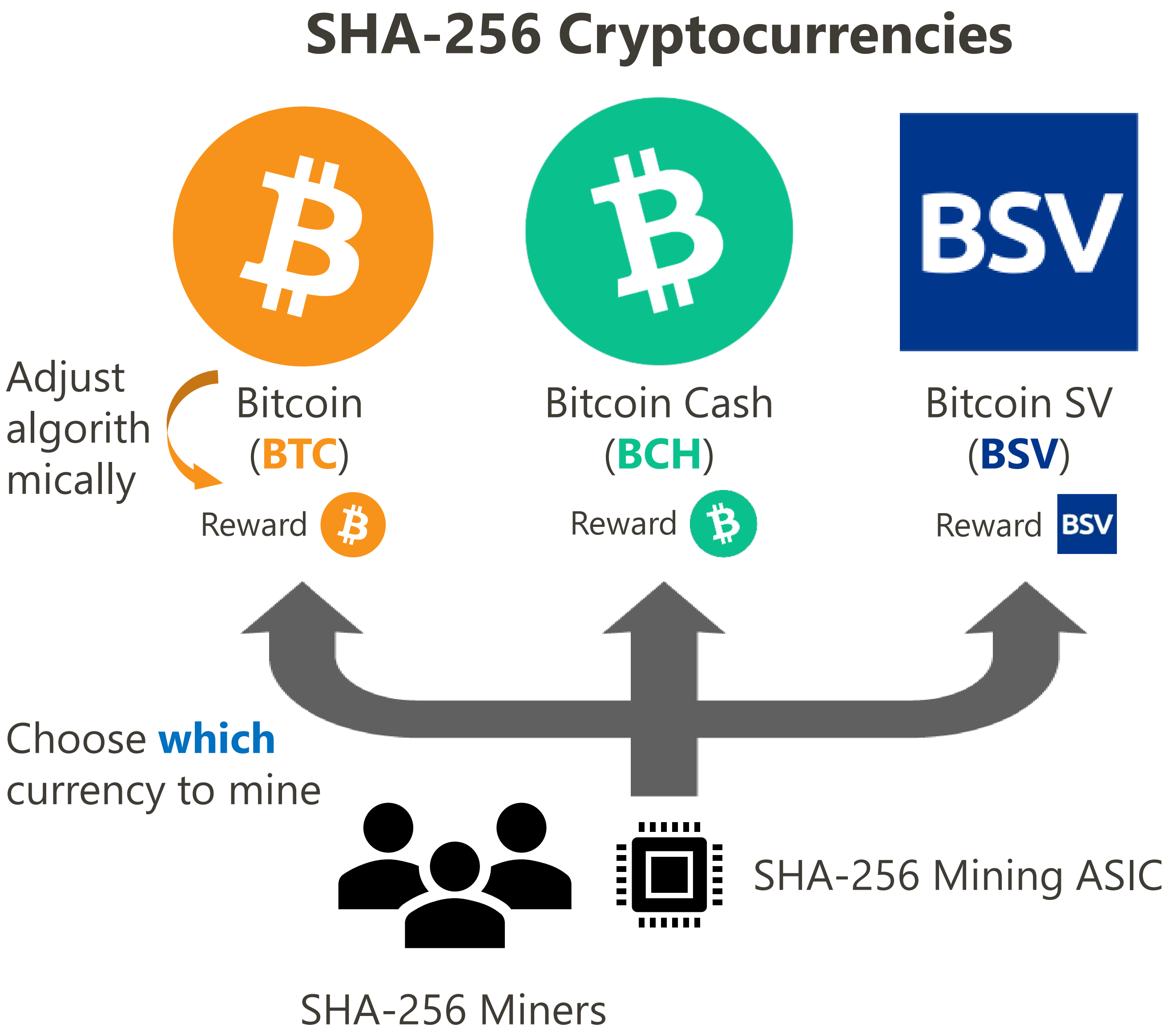
- **Algorithmic decision making** has increasingly been adopted.
  → Emergence of markets that involve algorithms as players.
  - Emergence of **Competition** and **coordination** between algorithms.

- In this paper, we study whether and how **cryptocurrencies** coordinate.

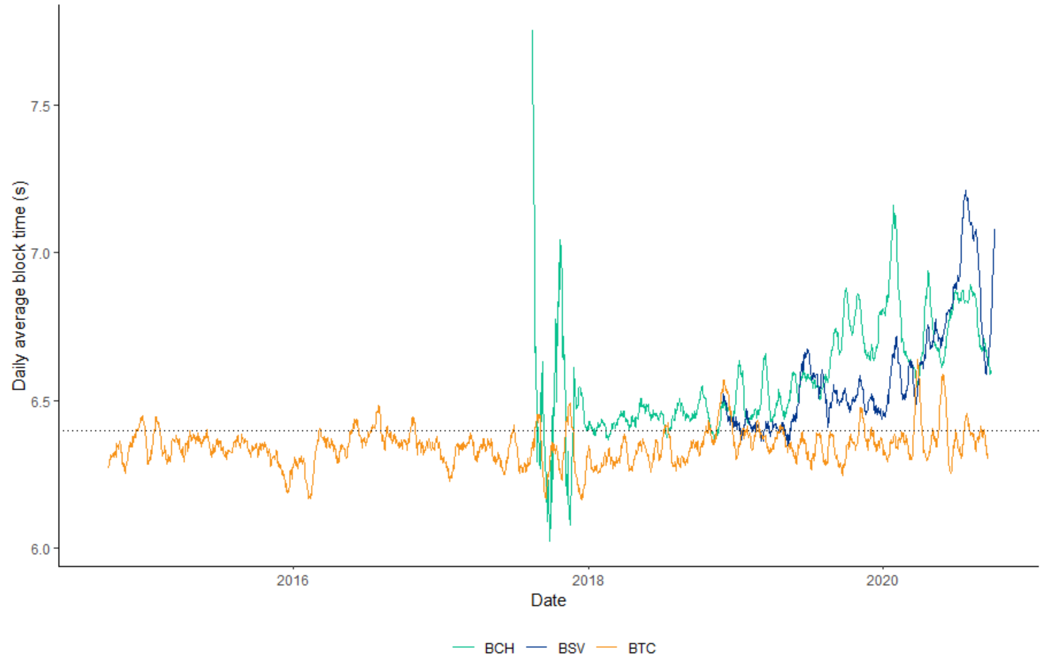
# Competition and Coordination b/w Cryptocurrencies

- Cryptocurrency is a decentralized electronic payment system.
  - Anyone can work as a record-keeper, called a **miner**.
- Each system attempts to **stabilize** the processing speed of transactions.
  - Each system adjusts a parameter called "**target**" algorithmically.
  - The target determines miners' **expected reward** (time wage).
    → **Algorithmic pricing** for hiring miners.
- We focus on cryptocurrencies that use **SHA-256** for their mining puzzle.

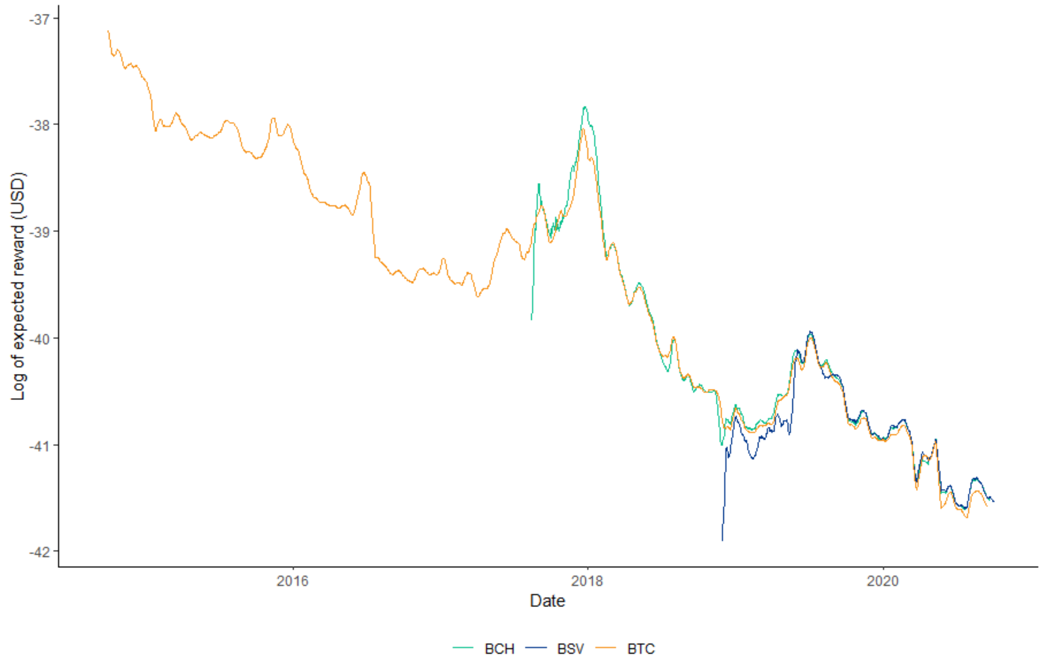
# SHA-256 Market Structure



# Tacit Coordination




**Historical log block time (daily average)**




**Historical log expected reward**

- Currencies **do not equate** the expected reward rate directly!
- Nevertheless, the three currencies had similar expected reward rate because of **algorithmic competition**.
- Occasional coordination failure? → Really? How are they mitigated?

# Research Question

**Question 1:**

Are cryptocurrencies successfully (tacitly) coordinating to stabilize the processing speed?

**Question 2:**

How tightly these currencies are connected in the mining market?

**Question 3:**

Can we resolve/mitigate the coordination failure by improving algorithms?

# Method

- We focus on the period where we observe a large variation in rewards.
  - **Halving**: the mining prize is halved every 210,000 blocks (4 years)
  - Agreed when Bitcoin was launched. → Foreseeable but exogenous.
  - A drastic change on the aggregate hash supply.
    → Can cryptocurrencies smoothly absorb the shock?

- We apply the RDD to detect the coordination failure.
- We estimate the aggregate hash supply (how miners respond to the change in the reward rate) to perform counterfactual simulations.

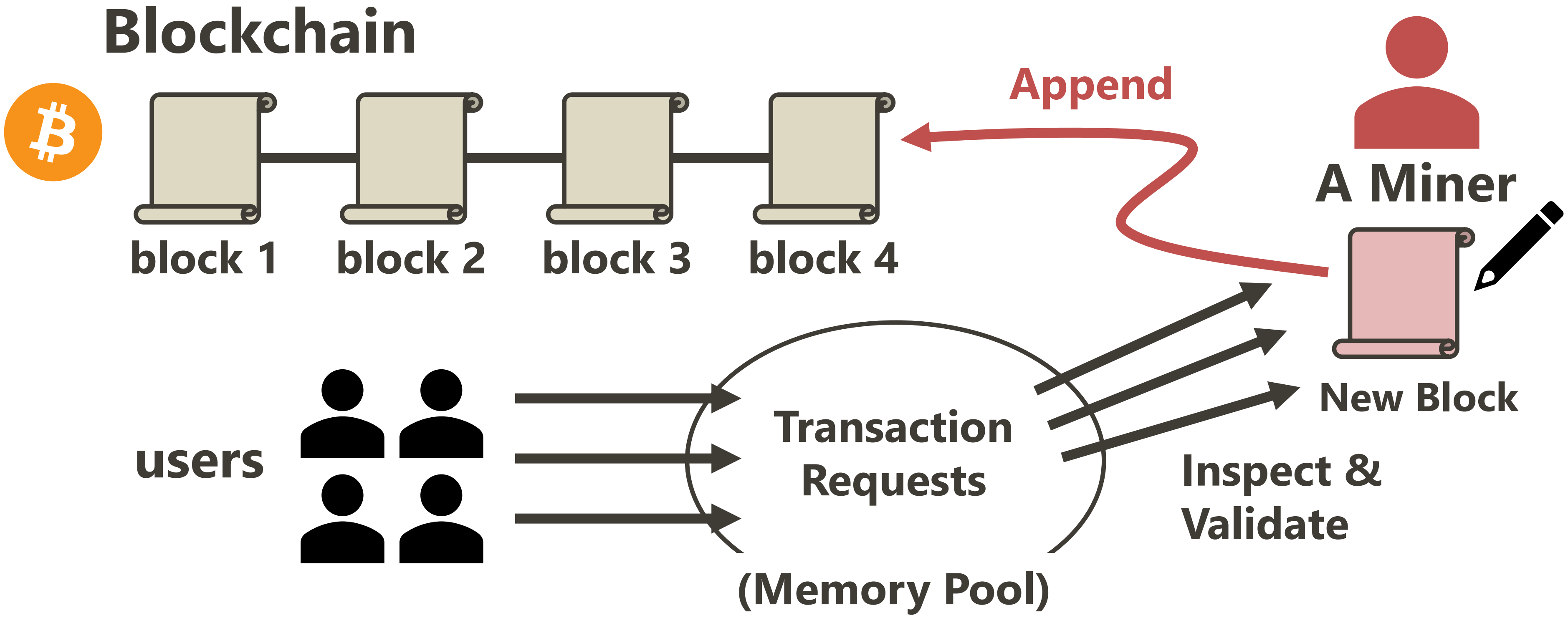
# Result Overview

- **BTC**'s halving significantly influenced **BCH** and **BSV**'s hash supply.
  - But, **BCH** and **BSV** made sophisticated algorithmic decisions.
    → Quickly absorbed the **BTC** halving shock.
- **BTC**'s algorithm is less efficient than **BCH** and **BSV**'s.
  - **BTC**'s hash supply was less elastic: **BTC** has many loyal miners.
  - This explains why **BTC** could survive with the obsolete algorithm.
- Counterfactual simulations show that **BCH** and **BSV** would have collapsed if they had adopted the **BTC**'s algorithm.

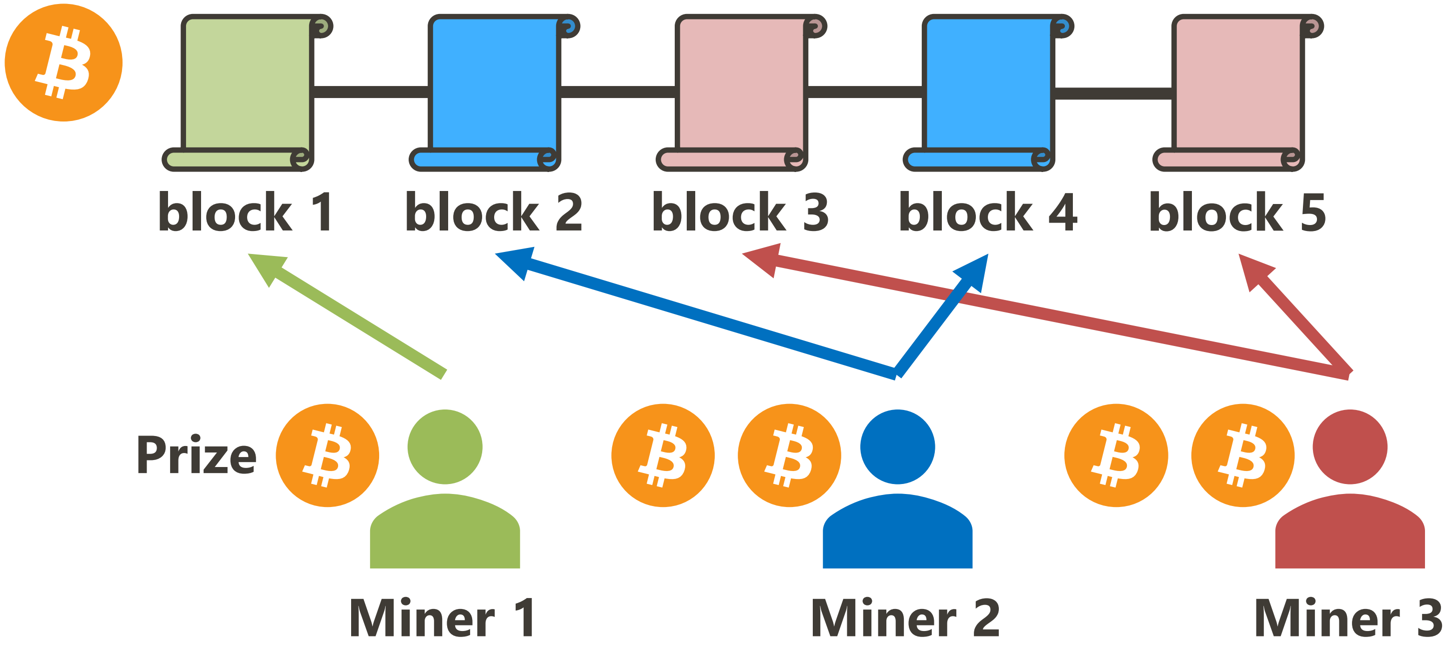
# Cryptocurrency Basics

# Blockchain

- Blockchain = A chain of blocks.
- Block = a group of newly validated transaction requests.
- A miner collects a set of (valid) pending transactions to produce a new block.



# Rewards




- **Multiple miners** work on this task, and different miners produce each block.
- Upon producing a block, the miner obtains a monetary (cryptocurrency) reward.
  - **Seigniorage** (dominant) + transaction fees

# Proof-of-Work (PoW)

- When a miner is allowed to append a new block?
- **Proof-of-Work (PoW)** (Dwork and Naor, 1992)
  - Miners are required to complete a **cumbersome task** to produce a block.
- The cryptographic **hash function** is a key part of PoW system.
- Any data ↦ a fixed-digit integer
  - **Block Header** (with **Nonce**) ↦ 256-bit number $(0 \sim 2^{256} - 1)$
- The return is (virtually) **ex-ante unpredictable**.
  - We cannot infer the returned value unless we actually calculate it.
- **"Cumbersome Task":** Find a "good nonce" by **try and error**.

# Hash Computation = Lottery Draw

- A miner is allowed to add a new block iff he finds a nonce that leads to a hash value smaller than the **difficulty target**.
  - The difficulty target is each currency's **policy variable**.
- The hash value is unpredictable → Computing a hash value with one nonce (counted as **1 hash**) is equivalent to drawing one lottery.
  - iid draw from a uniform distribution.
  - Probability of success (**winning rate**):

**The mass of the targeted range**

**Winning Rate** $$w = \frac{(\text{difficulty target})}{2^{256}}$$

**The mass of the whole range of SHA-256**

# Model

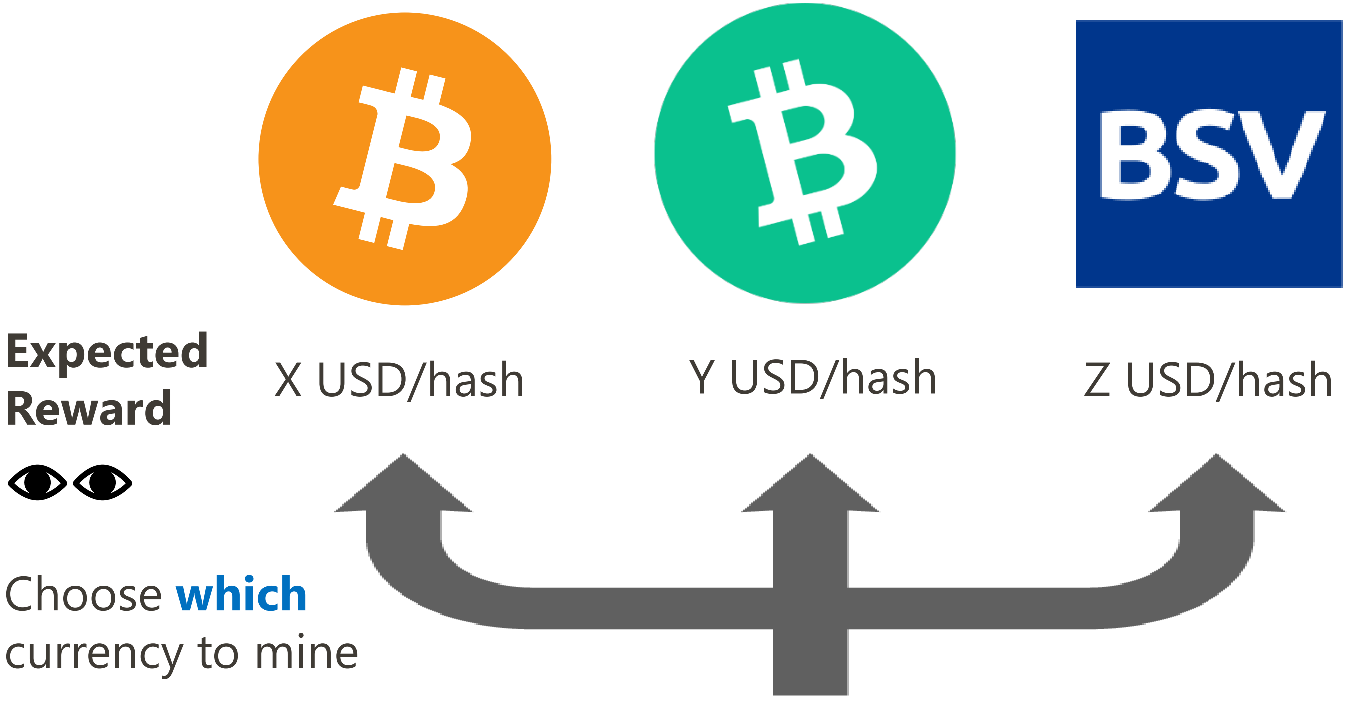
# Model

- **Continuous time** $t \in \mathbb{R}_+$
- The set of **cryptocurrencies** $[K] = \{1,2, \dots, K\}$. (BTC, BCH, BSV)

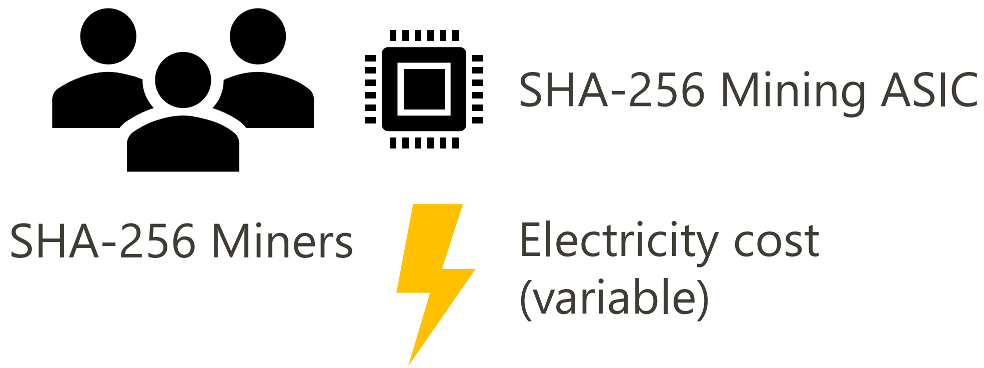
  At time $t$, Currency $k$'s...
  - **Winning rate**: $w(k,t)$ block/hash. (**Algorithmically** adjusted)
  - **Prize**: $m(k,t)$ TKN($k$)/block.
  - **Exchange rate** (against fiat money): $e(k,t)$ USD/TKN($k$).
  - **Expected reward rate**: $r(k,t) \coloneqq w(k,t) \cdot m(k,t) \cdot e(k,t)$ USD/hash.
  - **Hash rate**: $h(k,t)$ hash/second (unobservable).
- The hash rate is endogenously determined as a function of the reward rate.

# Miner's Problem & Hash Rate Determination

- Expected reward rate $(r)$
  $=$ winning rate $(w)$
  $\times$ prize $(m)$
  $\times$ exchange rate $(e)$
- Each miner **observes the expected reward rate** at each moment and decides which currency to mine




- Currency $k$'s hash rate $(h(k,\cdot))$
  = **Miners' total effort** spent on currency $k$.
  - The hash rate should be dependent on the profile of the expected reward rates.

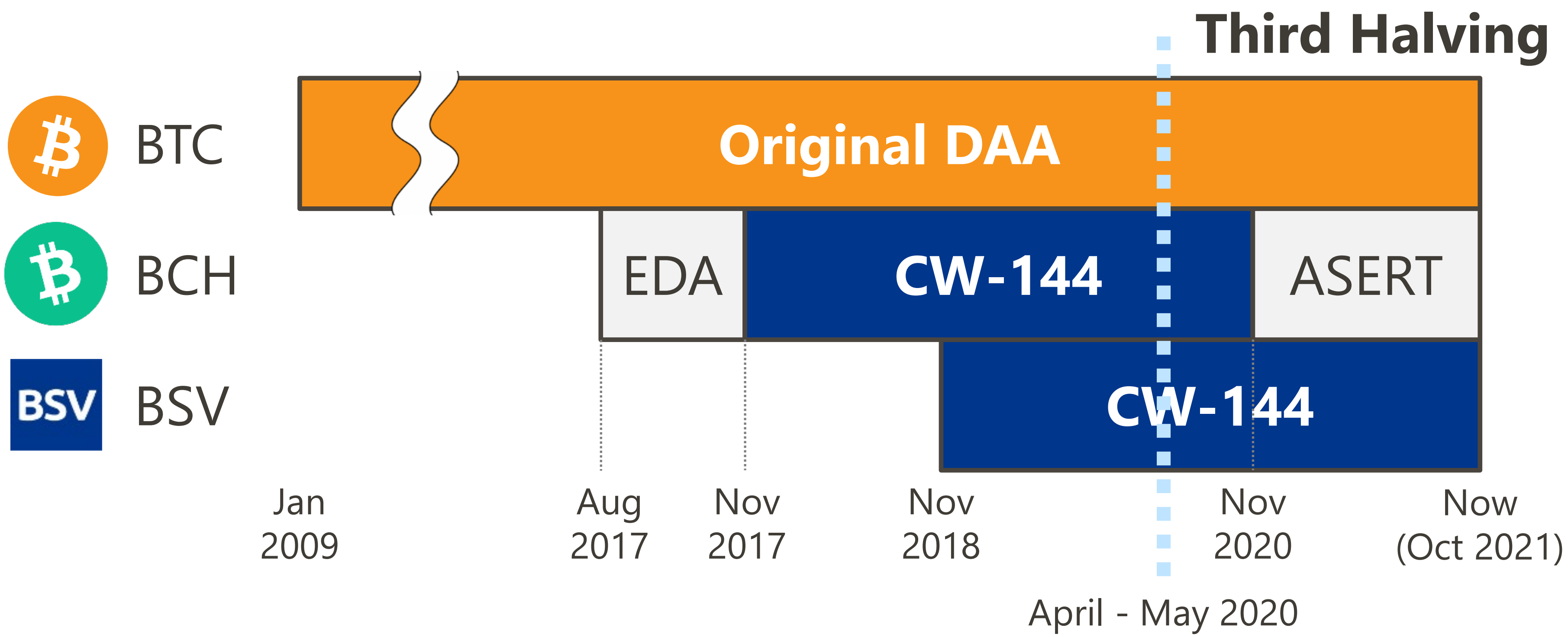


# Block Arrival Process

- The winning rate $w$ is extremely small ($< 10^{-20}$) and the hash rate $h$ is extremely large.
- Block arrival approximately follows a non-homogeneous **Poisson process** with **arrival rate** $wh$. → The average block time: $1/wh$.
- **BTC**, **BCH**, and **BSV** adjust the winning rate $w$ to produce blocks every 10 minutes ⇔ attempt to achieve $1/wh$ = 10 minutes.

  → Adjust the winning rate $w$ algorithmically (**difficulty adjustment algorithm**)

# Difficulty Adjustment Algorithms (DAA)




- Noda, Okumura, and Hashimoto (2020):
  - **Original DAA fails** to stabilize block time when the hash supply is elastic.
  - **CW-144 performs well** even when the hash supply is highly elastic.

# Original DAA

- Due to the time constraint, we skip the full detail of DAAs.

- **Original DAA (BTC)**
  - Periodic adjustment.
    Adjust the winning rate for **every 2,016 blocks** (= 2 weeks if blocks are produced for every 10 minutes).

$$\text{New winning rate} = \frac{\text{Average block time of past 2,016 blocks}}{\text{Targeted block time (10 minutes)}} \times \text{Old winning rate}$$

# CW-144

- CW stands for chainworks (the firm that developed this DAA).
- **CW-144 (BCH, BSV)**
  - Continuous adjustment.
    Adjust the winning rate for **every single block** (= 10 minutes).
  - Check the time needed for producing the **past 144 blocks** (= 1 day).

$$\text{New winning rate} = \frac{\text{Average block time of past 144 blocks}}{\text{Targeted block time (10 minutes)} \times \text{Average “Work” (= 1/winning rate) of past 144 blocks}}$$

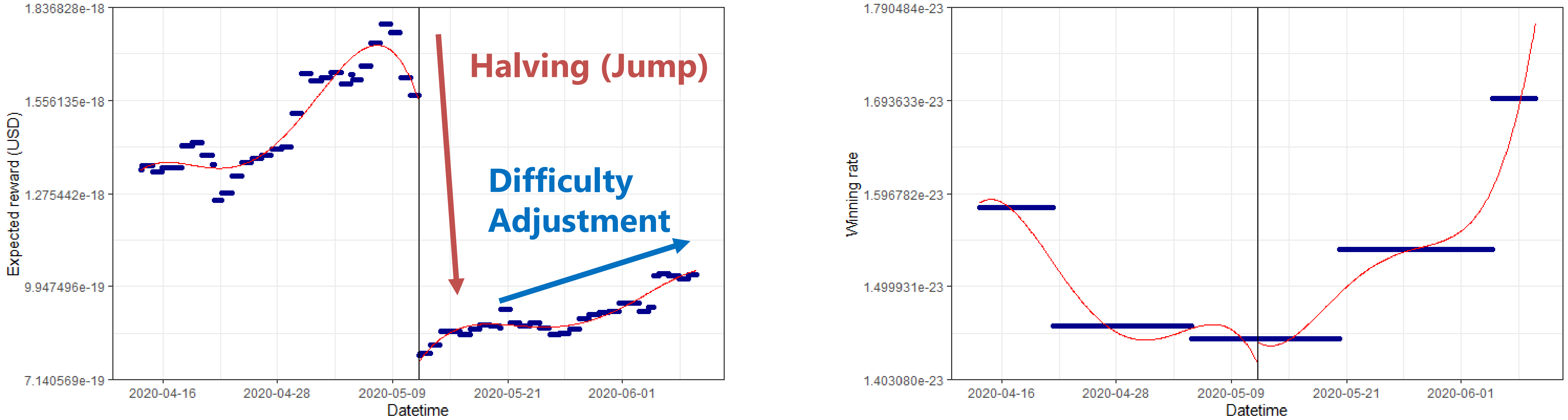
# The Impact of BTC Halving

# Halving

- We want to look at cryptocurrencies' behavior against a large "exogenous" shock. → Focus on **halving**.

- The prize ($m$) of **BTC**, **BCH**, and **BSV** are halved for every 210,000 blocks.
  - Block arrivals are independent across currencies → The halving period of these three currencies arrive sequentially.
  - **BCH** (April 8, 2020) → **BSV** (April 10, 2020) → **BTC** (May 11, 2020)
  - We look at the largest event – **BTC halving**.

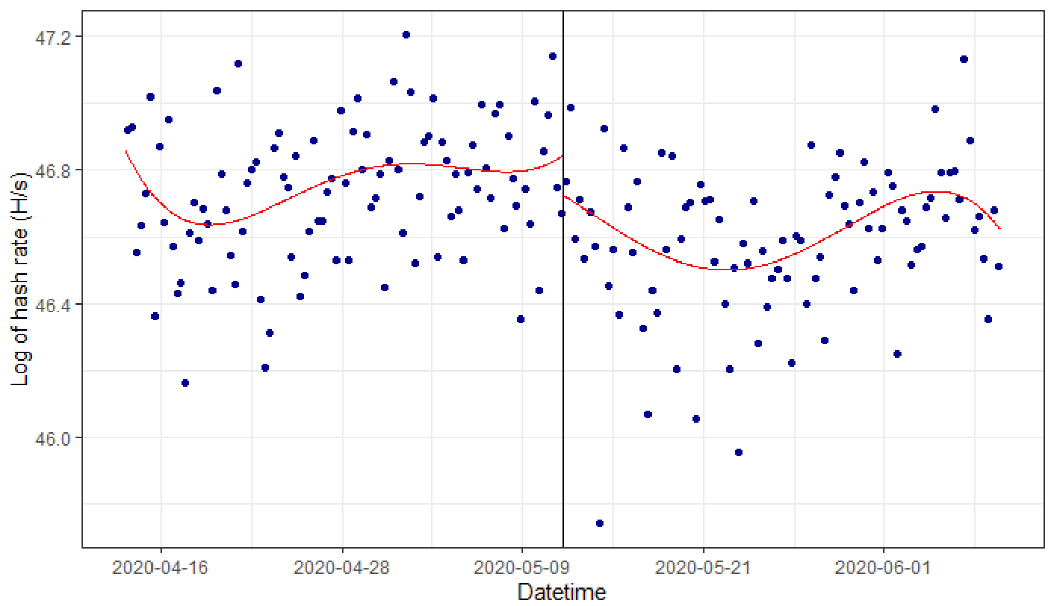
# The Impact of BTC's Halving on BTC (1)



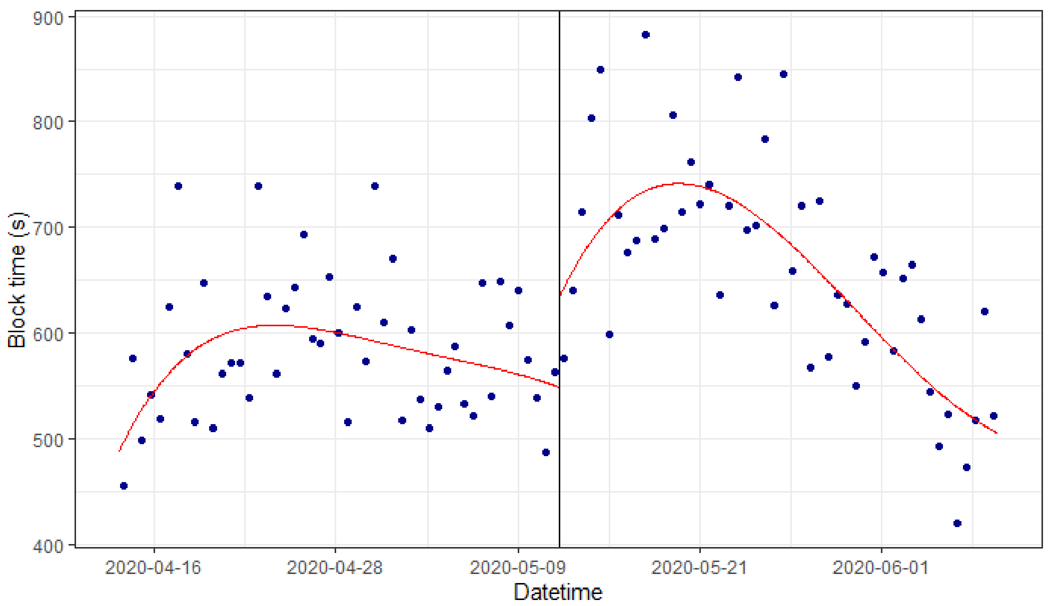

**Expected reward rate (USD/hash)** **Winning rate (block/hash)**

- Reward → Jumped down due to halving.
- Winning Rate → Adjusted for every 2,016 blocks (2 weeks).
  It is not reset to an ideal level instantly, reflecting the adjustment of the prize.

# The Impact of BTC's Halving on BTC (2)



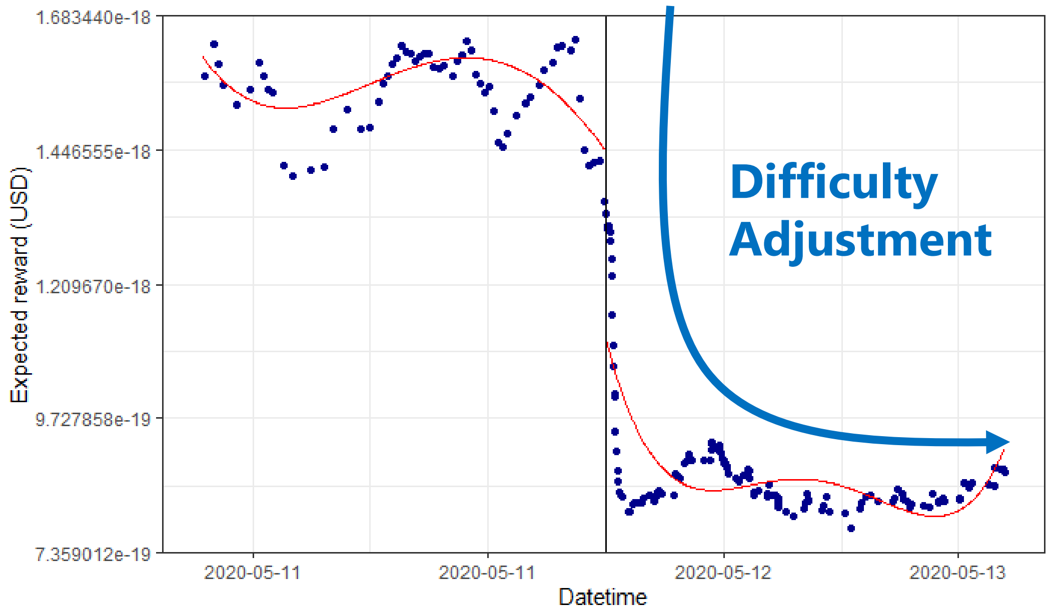

**Log estimated hash rate (hash/second)**



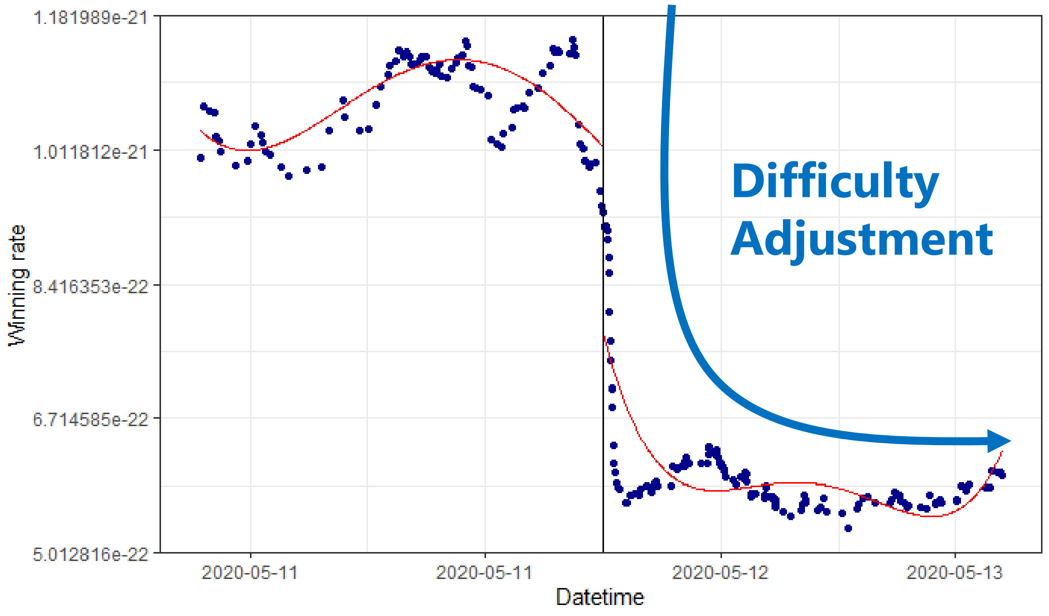

**Block time (second)**

- Hash rate decreased, and block time increased after halving, and it took time to go back to the stationary level.
- Not critical but a significant economic problem.

# The Impact of BTC's Halving on BCH




Expected reward rate (USD/hash)

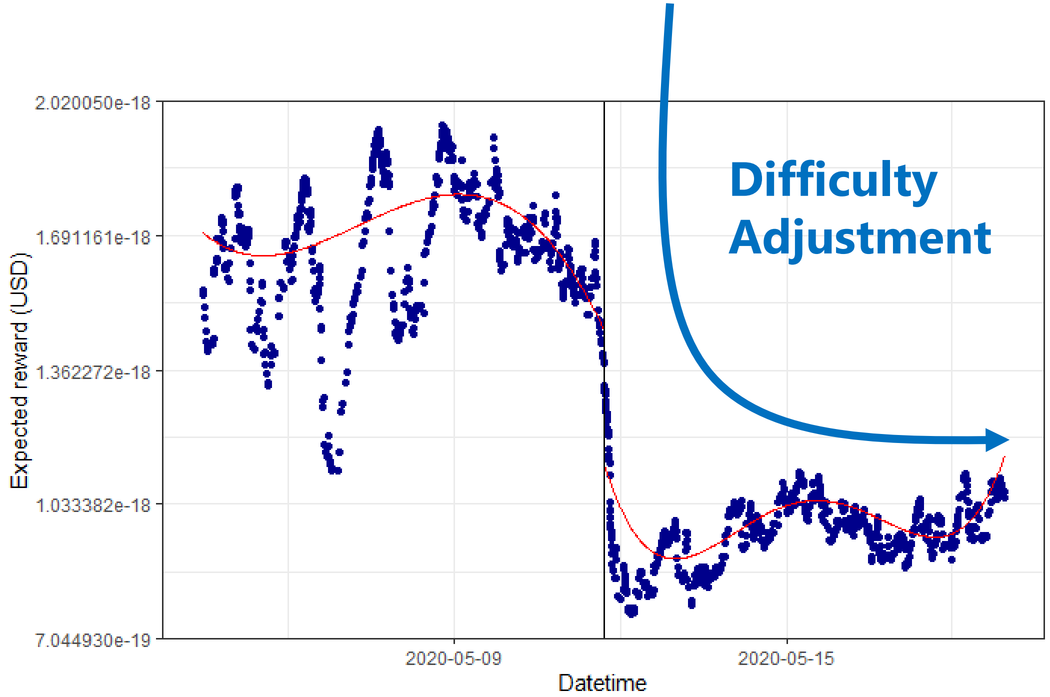



Winning rate (block/hash)

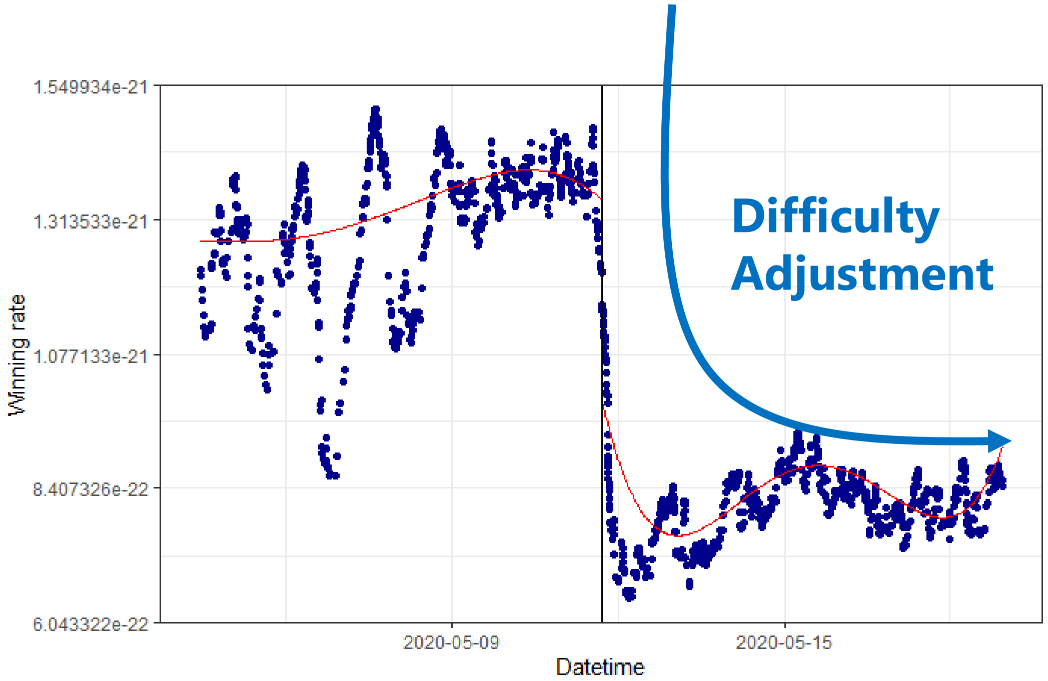
- Recall: **BCH's prize** is unchanged! (We are studying **BTC's halving**.)
- **BCH**'s DAA (CW-144) quickly adjusted the winning rate to absorb the BTC halving shock.

# The Impact of BTC's Halving on BSV

BSV




**Expected reward rate (USD/hash)**




**Winning rate (block/hash)**

- **BSV** also used CW-144.
- The observed patterns of the winning rate and reward rate of **BSV** were similar.

# Estimation of the Aggregate Hash Supply

# Reduced-Form of The Hash Supply

- We measure the **reward-elasticity** of the miners' **hash supply**.
  - The reward rate is miners' primary concern.
- We approximate the hash rate function by a log-log linear function.

$$h(k,t) = \bar{h}(t) \cdot \exp\left( \alpha_k + \sum_{k' \in [K]} \beta_{k',k} \log r(k',t) \right)$$

  - $\beta_{a,b}$ is currency $b$'s hash supply elasticity of currency $a$'s reward rate.
  - $\bar{h}(t)$ is the aggregate hash power (the total capacity of mining ASIC).

# Sketch of Estimation Strategy

- We combine the following two data sets to produce a combined data set about the expected reward rate, winning rate, and block arrivals.
  - The full history of blockchains (**BTC**, **BCH**, **BSV**)
  - The exchange rate against USD (downloaded from Yahoo Finance)

- **Short period**: from 28 days before BCH halving to 28 days after BTC halving ($\approx$ 3 months).
  → We can assume that miners' equipment (mining ASIC holding) was constant.
  (= We can assume that $\bar{h}$ **was a constant**.)

- We use the maximum likelihood method estimate the parameter $(\alpha, \beta)$.

# Estimation Result (1)

| | |
|---|---|
| **BTC** | 52.879***<br>(1.973) |
| **BCH** | 49.851***<br>(1.995) |
| **BSV** | 47.764***<br>(1.973) |

**Constants**

$*\ p < 0.05$, $**\ p < 0.01$, $***\ p < 0.001$

| | | Hash Supply (To) | | |
|---|---|---|---|---|
| | | **BTC** | **BCH** | **BSV** |
| Reward (From) | **BTC** | 0.626***<br>(0.103) | -3.981***<br>(0.113) | -3.186***<br>(0.106) |
| | **BCH** | -0.240*<br>(0.095) | 5.386***<br>(0.127) | -1.540***<br>(0.093) |
| | **BSV** | -0.223*<br>(0.098) | -1.219***<br>(0.076) | 4.869***<br>(0.118) |

**Table of row-reward-elasticity of column**

# Estimation Result (2)

| | |
|---|---|
| BTC | 52.879***<br>(1.973) |
| BCH | 49.851***<br>(1.995) |
| BSV | 47.764***<br>(1.973) |

**Constants**

$^{*} p < 0.05$, $^{**} p < 0.01$, $^{***} p < 0.001$

| Reward (From) | Hash Supply (To) | | |
|---|---|---|---|
| | BTC | BCH | BSV |
| BTC | 0.626***<br>(0.103) | -3.981***<br>(0.113) | -3.186***<br>(0.106) |
| BCH | -0.240*<br>(0.095) | 5.386***<br>(0.127) | -1.540***<br>(0.093) |
| BSV | -0.223*<br>(0.098) | -1.219***<br>(0.076) | 4.869***<br>(0.118) |

**Table of row-reward-elasticity of column**

- The hash supply is **increasing** in its own reward (diagonal elements) and **decreasing** in its rival's reward (off-diagonal elements).

# Estimation Result (3)

| | Constants |
|---|---|
| BTC | 52.879***<br>(1.973) |
| BCH | 49.851***<br>(1.995) |
| BSV | 47.764***<br>(1.973) |

**Constants**

* p < 0.05, ** p < 0.01, *** p < 0.001

| | | Hash Supply (To) | | |
|---|---|---|---|---|
| | | BTC | BCH | BSV |
| Reward (From) | BTC | 0.626***<br>(0.103) | -3.981***<br>(0.113) | -3.186***<br>(0.106) |
| | BCH | -0.240*<br>(0.095) | 5.386***<br>(0.127) | -1.540***<br>(0.093) |
| | BSV | -0.223*<br>(0.098) | -1.219***<br>(0.076) | 4.869***<br>(0.118) |

**Table of row-reward-elasticity of column**

- **BTC** is much larger than **BCH** and **BSV**
  → There are many loyal miners and its hash supply is less elastic.
  - (Could be rational. BTC-USD exchange market is much thicker.)
- Noda et al. (2020): **(BTC's) Original DAA** performs well iff elasticity < 1.

# Estimation Result (4)

* p < 0.05, ** p < 0.01, *** p < 0.001

| | |
|---|---|
| **BTC** | 52.879*** (1.973) |
| **BCH** | 49.851*** (1.995) |
| **BSV** | 47.764*** (1.973) |

**Constants**

| | | Hash Supply (To) | | |
|---|---|---|---|---|
| | | **BTC** | **BCH** | **BSV** |
| Reward (From) | **BTC** | 0.626*** (0.103) | -3.981*** (0.113) | -3.186*** (0.106) |
| | **BCH** | -0.240* (0.095) | 5.386*** (0.127) | -1.540*** (0.093) |
| | **BSV** | -0.223* (0.098) | -1.219*** (0.076) | 4.869*** (0.118) |

**Table of row-reward-elasticity of column**

- The hash supply of **BCH** and **BSV** was highly elastic.
  - Their difficulty adjustment problem is much more difficult than **BTC**'s.
- Noda et al. (2020): (**BCH** and **BSV's**) CW-144 is stable iff elasticity < 144.

# Counterfactual Simulations

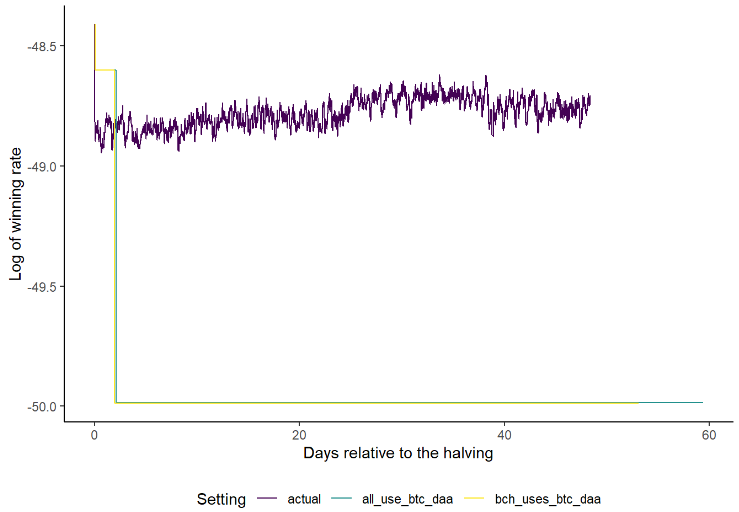
# Scenario

- What would have happened if **BCH** and **BSV** had used different DAAs in the period of third **BTC halving**?
- We use the estimated hash supply to run the counterfactual simulation.
- We consider the following scenarios.

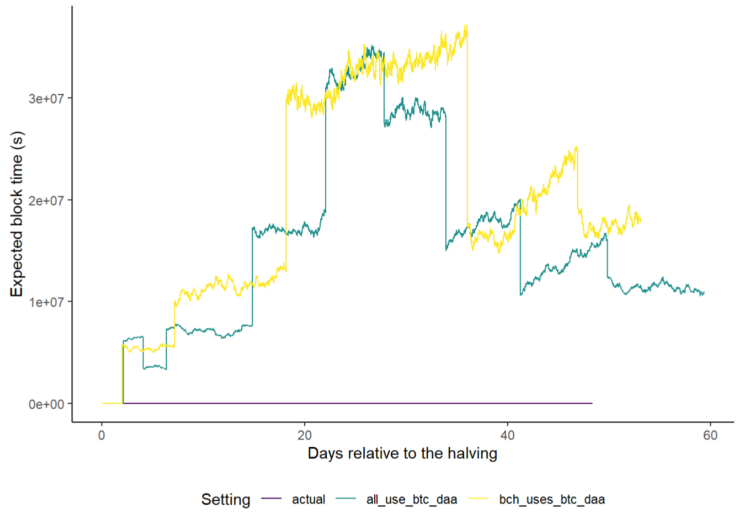
| Scenario | BTC | BCH | BSV |
|---|---|---|---|
| **actual** | Original DAA | CW-144 | CW-144 |
| bch_uses_btc_daa | Original DAA | Original DAA | CW-144 |
| all_use_btc_daa | Original DAA | Original DAA | Original DAA |

- We start the simulation right before the third **BTC halving**.
- The exchange rate is updated according to the geometric Brownian motion.

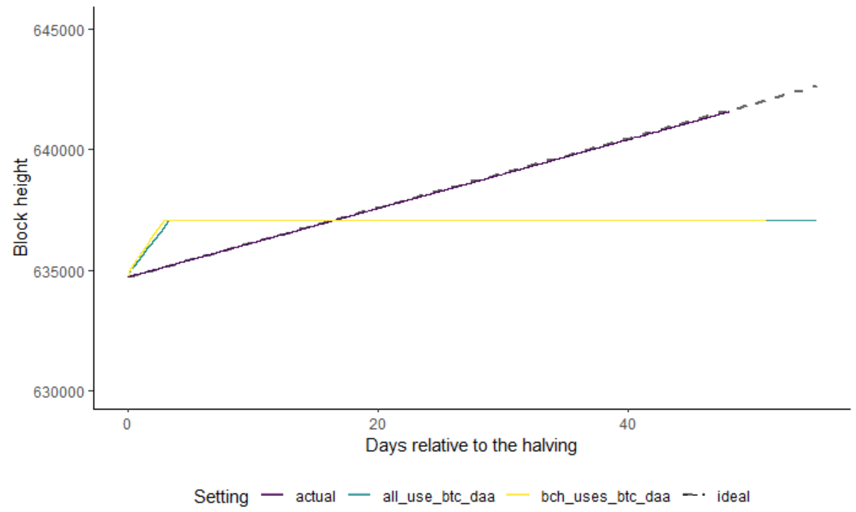
# BCH would have collapsed




Log of winning rate



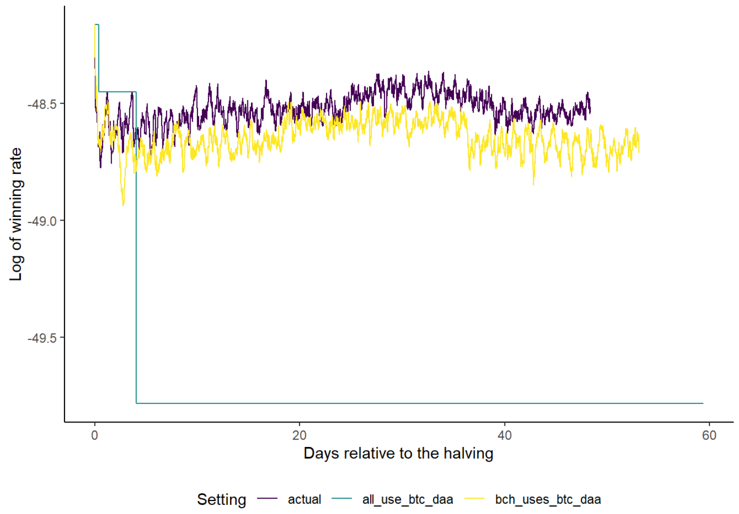

Expected block time



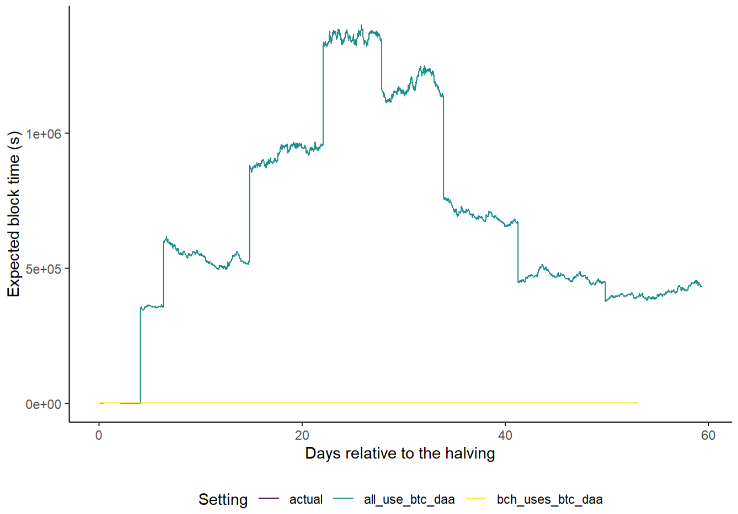

Block Height

- Only **actual** uses CW-144. CW-144 quickly stabilizes the block arrival rate.
- When the original DAA is used, mining is too easy after the halving (**BTC** is less profitable → many miners join **BCH** mining). Mining becomes too difficult in the next epoch, and **BCH** cannot produce 2,016 blocks for the next adjustment.

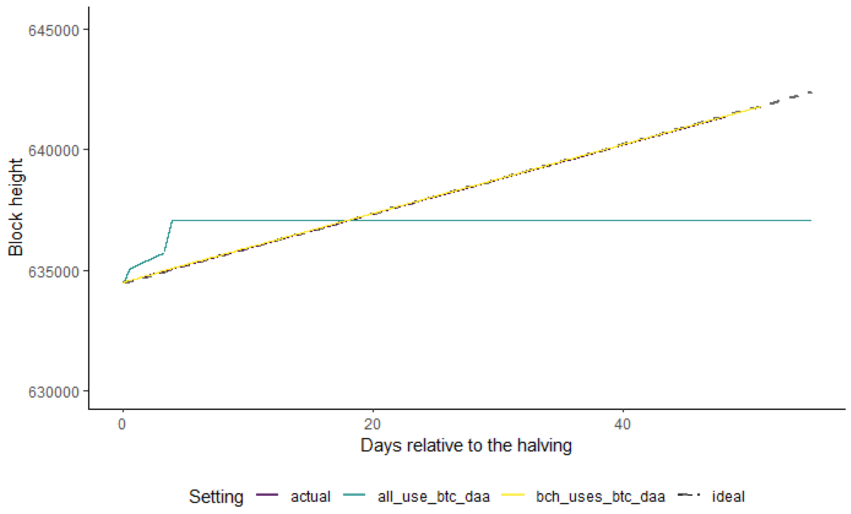
# **BSV** would also have collapsed




**Log of winning rate**



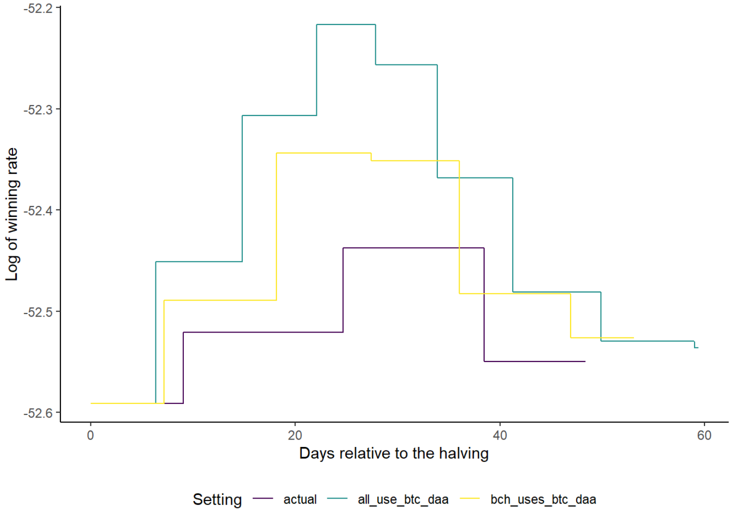

**Expected block time**



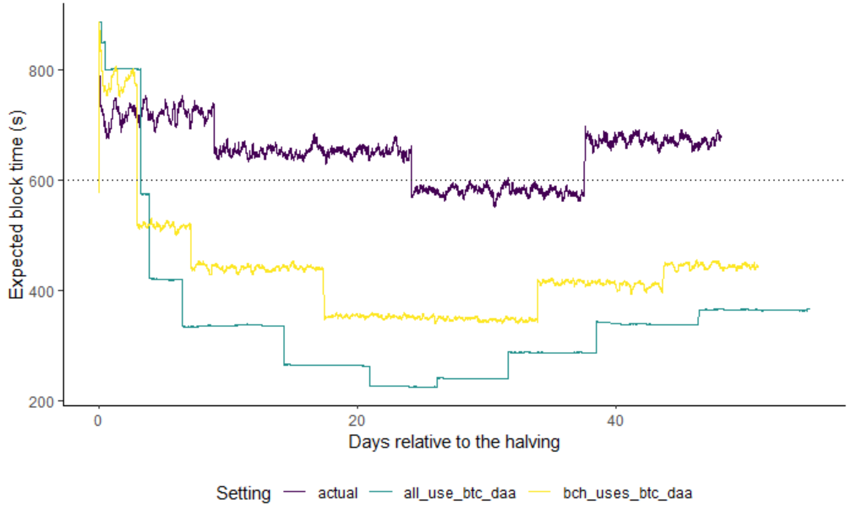

**Block Height**

- The same analysis applies to **BSV**.
- **BSV** survives only if CW-144 is adopted.
- **BSV** is attempting to restore the original Satoshi protocol, but restoring to the original DAA seems a bad idea.

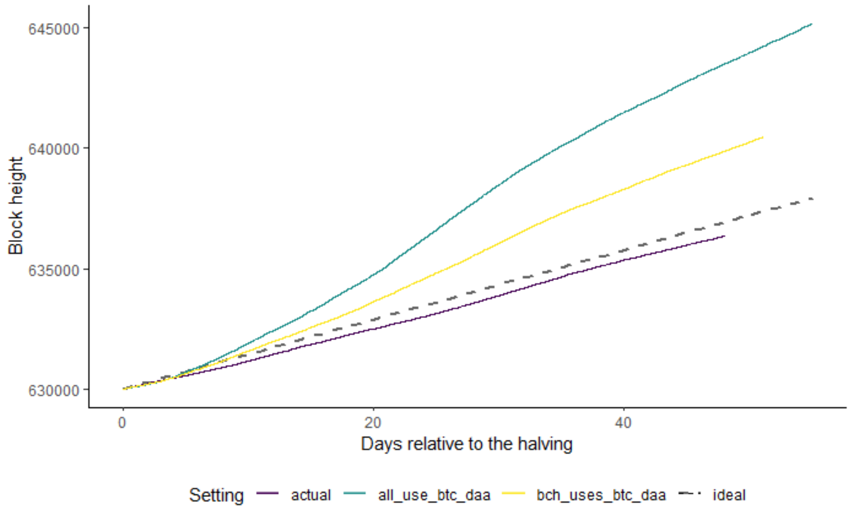
# Impact on BTC




**Log of winning rate**




**Expected block time**




**Block Height**

- **BCH** and **BSV** adopt CW-144 → **BTC** is benefitted.
- CW-144 quickly absorbs the shock and adjusts the hash rate of **BCH** and **BSV**.
  → It also stabilizes the hash power supplied to **BTC**.
- The influence of the halving on **BTC** would have been larger if **BCH** and **BSV** had not work as a “**shock absorber**”.

# Concluding Remarks

# Concluding Remarks

- We studied the structure of algorithmic competition and coordination between cryptocurrencies (**BTC**, **BCH**, and **BSV**).
- These currencies are involved in the same **miner-hiring market**, and their algorithmic pricing is influencing each other.
- When a large shock (such as halving) arrives, CW-144 (**BCH** and **BSV**) performs much better than the original DAA (**BTC**).
  - However, this problem has not been critical for **BTC** because **BTC** has many loyal miners and the hash supply to **BTC** is highly inelastic.
  - In contrast, **BCH** and **BSV** had collapsed if they had used the original DAA.
- Algorithmic adjustment also has an externality. **BCH** and **BSV**'s adoption of CW-144 is helping **BTC** to stabilize its block arrival rate.