University of Basel

Center for Innovative Finance

Bitcoin, Blockchain and Cryptoassets

Blockchain Forks

Prof. Dr. Fabian Schär
University of Basel

Release Ver.: (Local Release)
Version Hash: (None)
Version Date: (None)

Greatest Accumulated Difficulty

There is a simple rule to identify the most recent state of the ledger:

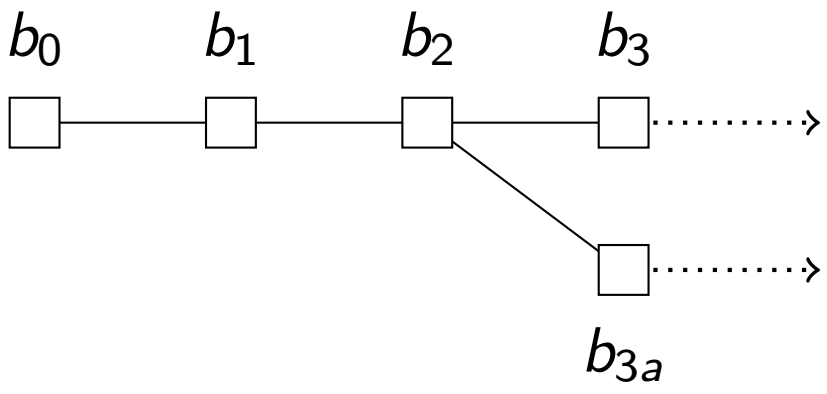
Intuition: The longest chain, i.e., the chain with the longest sequence of valid blocks is seen as the most recent version.

Rule: The chain with the greatest accumulated difficulty is seen as the most recent version.

Under normal circumstance this rule leads to a clear status quo. Forks are the exception...

What is a Fork?

Disagreement on the current state of the ledger that leads to two or more competing versions of the blockchain.



Forks may arise for two distinct reasons:

1. Same rules ($A = B$): Process-based, i.e., agents have not the same information set or choose to compete.
2. Different rules ($A \neq B$): Protocol-based, i.e., agents have a different understanding of consensus rules.

Classification of Forks

	Process-based ($A = B = S$)	**Protocol-based** ($A \neq B$)
Unintentional	Probabilistic Block Race	Client Incompatibility • Soft Fork • Hard Fork • Forced Fork
Deliberate	Block Withholding & Forced Block Race	Rule Change • Soft Fork • Hard Fork • Forced Fork

Table: The four fork types [2]

Process-based Forks

Probabilistic block race: Block creation is probabilistic. Two or more blocks may be created at approx. the same time.

Forced block race: Deliberate mining of own chain with the goal to overtake consensus version.

Block withholding: Purposeful delay of propagation of own valid candidate block to gain head start on next block.[1]

$\Rightarrow$ All temporary and resolved through accumulated difficulty (longest chain rule).

Protocol-based forks

Client incompatibility: Delta in consensus rule implementations by different network client software, causing some nodes to accept certain blocks rejected by others. Root causes:

- Loosely defined consensus rules
- Software bugs

Example: Upgrade to Bitcoin client 0.8 in 2013

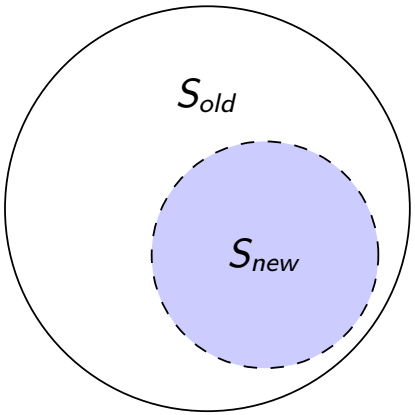
Rule change: Part of the network decides to alter the consensus rule set S and proceed with adapted protocol.

Example: Split of Bitcoin ABC over Blocksize increase.

$\Rightarrow$ Not resolved automatically and may cause permanent splits. Let us denote the old rules and the new rules by S_{old} and S_{new} respectively and analyze various situations.

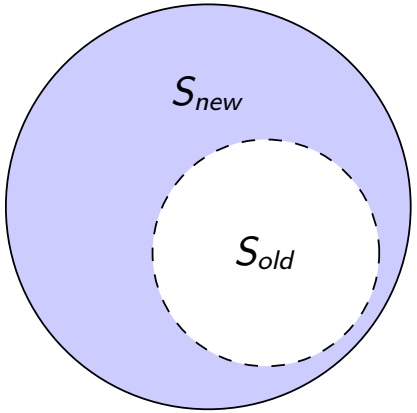
Types of Protocol-based Forks

Soft Fork



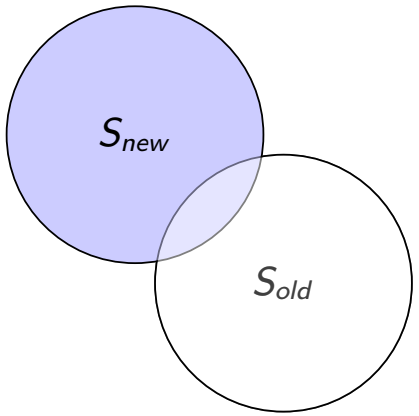
$$S_{new} \subset S_{old}$$

Hard Fork



$$S_{new} \supset S_{old}$$

Forced Fork



$$(S_{new} \setminus S_{old} \neq \emptyset) \wedge (S_{old} \setminus S_{new} \neq \emptyset)$$

Figure: Types of protocol-based forks [2]

Fork Persistency by Type and Dominance Scenario

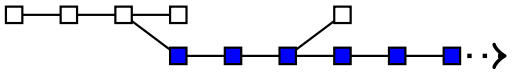
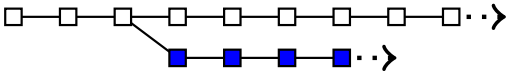
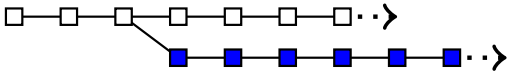
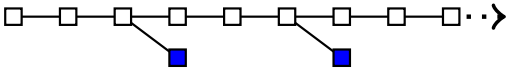
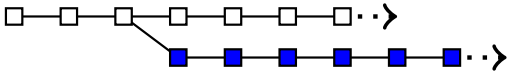
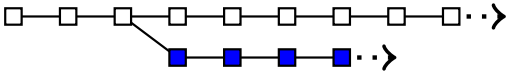
	S_{new} dominant ($r_{new} > r_{old}$)	S_{old} dominant ($r_{new} < r_{old}$)
Soft fork		
Hard fork		
Forced fork		

Table: Persistency by fork type and scenario [2]

Why Should We Care about Forks?

1. **Uncertainty:** Confirmation status of transactions.
2. **Confusion:** Various competing versions of the asset.
3. **Security Tokens:** Competing promises delivery of 1 good.
4. **Cost driver:** Tax / legal questions, maintaining compatibility.

But: Risk of fork may increase stability and strengthen status quo.

References and Recommended Reading

[1] Ittay Eyal and Emin Gün Sirer, *Majority is not enough: Bitcoin mining is vulnerable*, International conference on financial cryptography and data security, Springer, 2014, pp. 436–454.

[2] Fabian Schär, *Blockchain forks: A formal classification framework and persistency analysis*, Singapore Economic Review (forthcoming).