MIDDLEWARE SYSTEMS RESEARCH GROUP
MSRG.ORG

Deconstructing Blockchains: Concepts, Systems and Applications

PROOF OF WORK AND MINING

Blockchain “Puzzles”

verify(***nonce***, data) meets some “requirements”

Use of “trapdoor functions” (hash functions)

- Cannot reverse the function to find the input
- Therefore, keep trying random values (called nonce) until you find a solution
- Like trying random combinations to a lock…
- *The more computing power* you have, *faster* you can solve the puzzle.
- “*Magic blocks*” are blocks with puzzles, where everyone has the same power.

Proof-of-Work Example

E.g., the challenge is:

- sha256sum("data:**nonce**") starts with a "0"
- Normally more complicated than that! (e.g., 18 zeroes)

➢P1 wants to send "1:v" to P2

```
arno@grey:~$ echo "1:v:118" | sha256sum
9479038ca7543ece09f48e8c77fcea147d7561cac14058199afea18c2f323b8b
arno@grey:~$ echo "1:v:119" | sha256sum
79ae2bbac929112a349c2fe7f50210355f4a24683b2dd1ea8f059c9beeed7fd6
arno@grey:~$ echo "1:v:120" | sha256sum
002ce3a3b7092d960abf1795a89f70eb0f9ef960036e7d4620cbd3d26d34ffc8
```

➢Send "1:v:120" to P2

Proof-of-Work Example

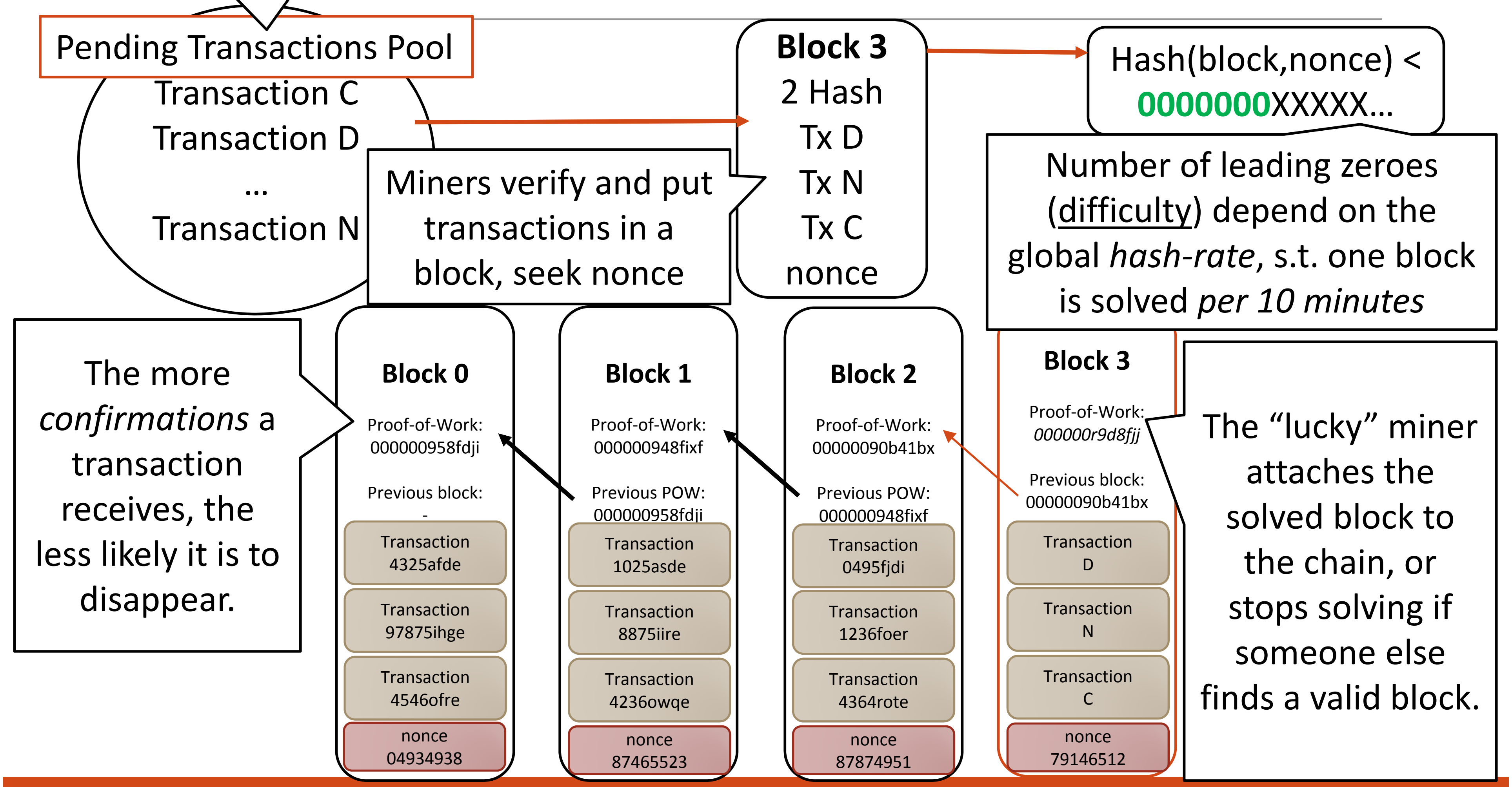
- P2 verifies “1:v:120” is correct (very quick!) (sha256sum(“1:v:120”) starts with a “0”)
- P2 wants to send “2:1:v:120” to P3

```
arno@grey:~$  echo "2:1:v:120:119" | sha256sum
911ab1edf1f331ff423a45fe4c382db30a3f1cf802bb2211df53c80d5798c7baa
arno@grey:~$  echo "2:1:v:120:120" | sha256sum
5344a3561673b1481b9cf69493368ca408b1edef67e3f96819c5d1b36cea53ce
arno@grey:~$  echo "2:1:v:120:121" | sha256sum
0a908c651e9ec5374976dc8f49a3342a4a789660011551da8871a6cc123c5b57
```

- P2 sends “2:1:v:120:121”
- P3 verifies “1:v:120” *AND* “2:1:v:120:121” are correct
- If P2 wants to send “2:1:w” and fool P3, it needs to find n_1 for “1:w: n_1” and n_2 for “2:1:w: n_1 : n_2”
- If P3 has a way to ***detect*** that P2 is ***doing too much work***, it can detect fraud.

f-of-Work in Bitcoin

Pending transactions are propagated to the peers (miners)



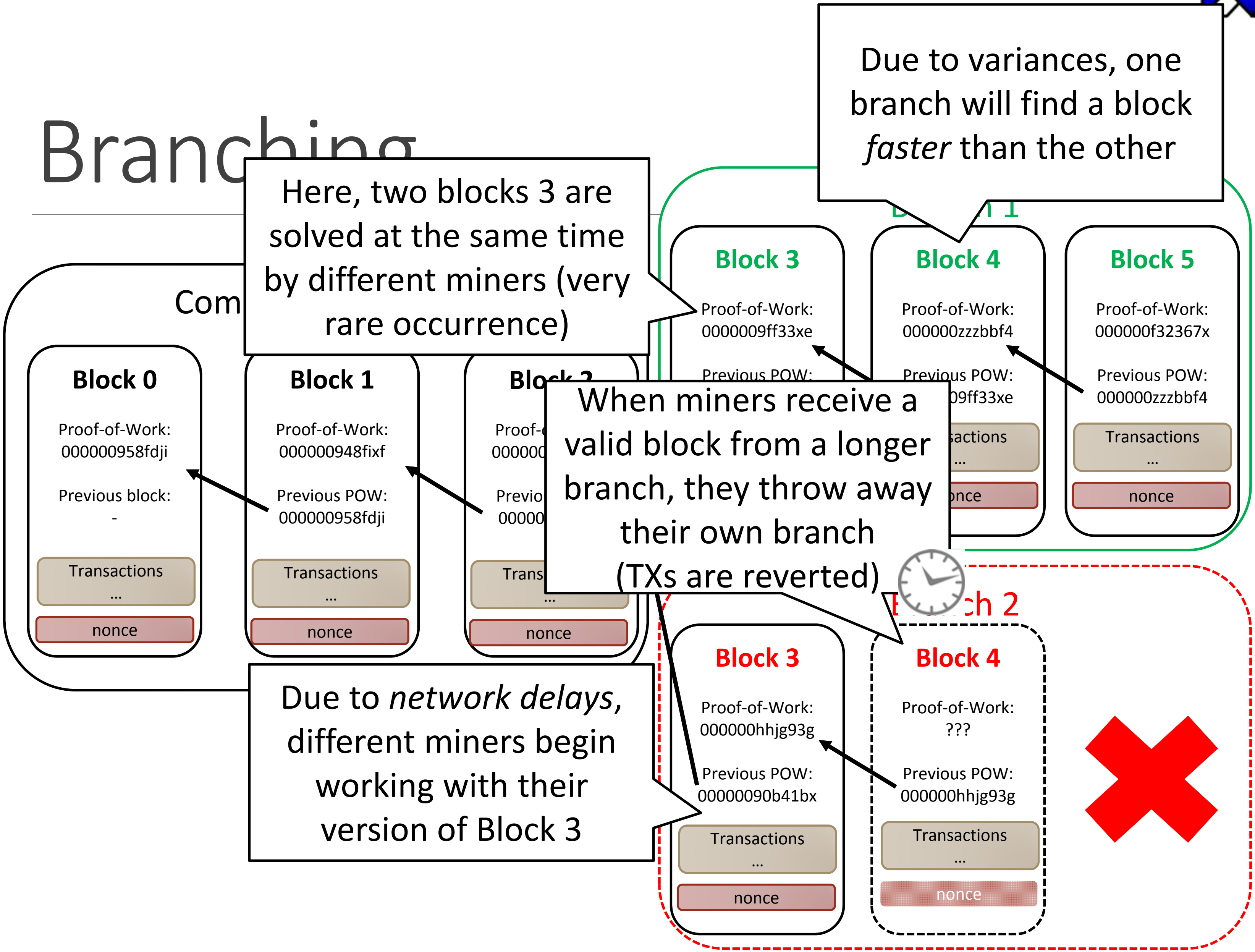
Miners verify and put transactions in a block, seek nonce

Number of leading zeroes (difficulty) depend on the global *hash-rate*, s.t. one block is solved *per 10 minutes*

The more *confirmations* a transaction receives, the less likely it is to disappear.

The “lucky” miner attaches the solved block to the chain, or stops solving if someone else finds a valid block.

Branching



Due to variances, one branch will find a block *faster* than the other

Here, two blocks 3 are solved at the same time by different miners (very rare occurrence)

When miners receive a valid block from a longer branch, they throw away their own branch (TXs are reverted)

Due to *network delays*, different miners begin working with their version of Block 3

Incentives

$$$$$$

Block reward, started with 50 BTC, 25 BTC, 12.5 BTC. …

- Creating of new coins (the only means to create coins)
- Reward reaped by miner whose block ultimately makes it into the chain
- Block reward will converge toward zero

Transaction fee

- Small amount that is paid by transaction issuer to miner
- Not a fixed amount, amount declared by issuer
- Ultimately, market forces may set this value