The Bitcoin Cryptocurrency: Part 4

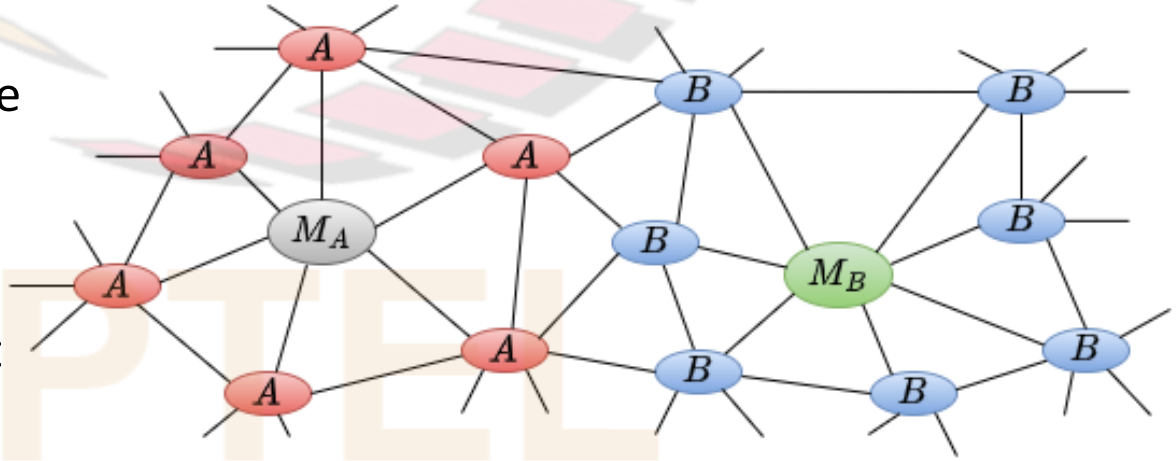
Gaurav S. Kasbekar
Dept. of Electrical Engineering
IIT Bombay

References

- Saravanan Vijayakumaran, "*An Introduction to Bitcoin*", Lecture notes, IIT Bombay, Oct. 4, 2017. Available at: https://www.ee.iitb.ac.in/~sarva/bitcoin.html
- A. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder, "*Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*", Princeton University Press, 2016

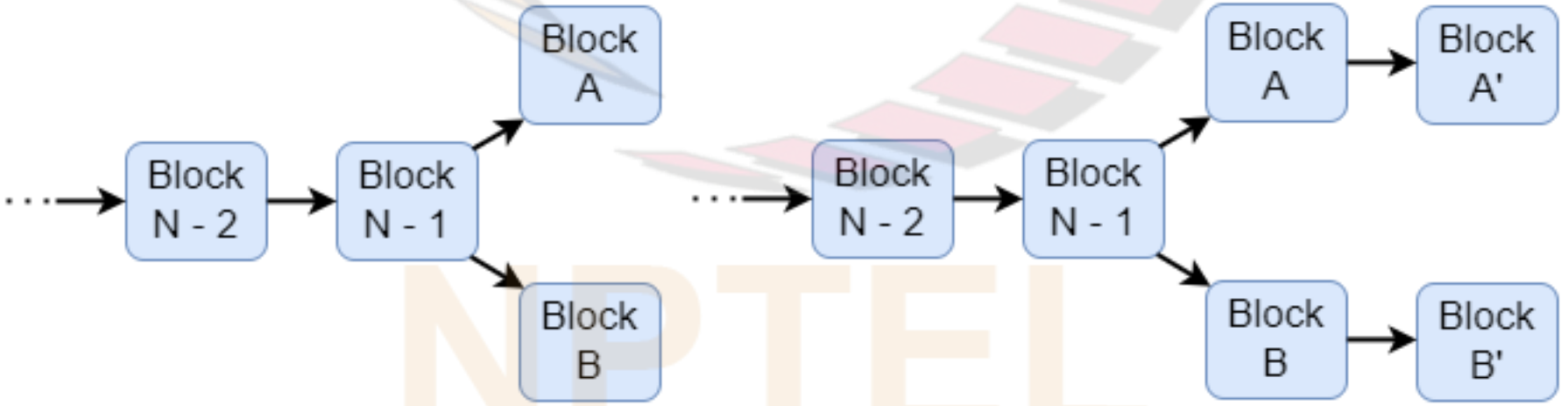
Blockchain Forks

- What happens if two miners find valid blocks at around the same time and broadcast the blocks?
- For concreteness, assume that:
  - ❑ initially, all the network nodes have the same copy of the blockchain, which ends in a block at height $N-1$
  - ❑ two valid blocks, A and B, both at height N, are found by two different miners, M_A and M_B, respectively, and are broadcasted before one of M_A and M_B received the other miner's block
- The other nodes either receive block A first or block B first
- Each node accepts the first block at height N that it receives and rejects the second block, e.g.,
  - ❑ if a miner receives block A first, it appends it to its local copy of the blockchain and starts mining for a block at height $N+1$ with block A as the previous block
  - ❑ if the miner later receives block B, it will reject it
- Eventually each node would have received either block A or B and appended it to its local copy of blockchain at height N
- Such a situation called a *blockchain fork* since the state of the network as seen by network as a whole consists of two branches both originating from same parent block at height $N-1$



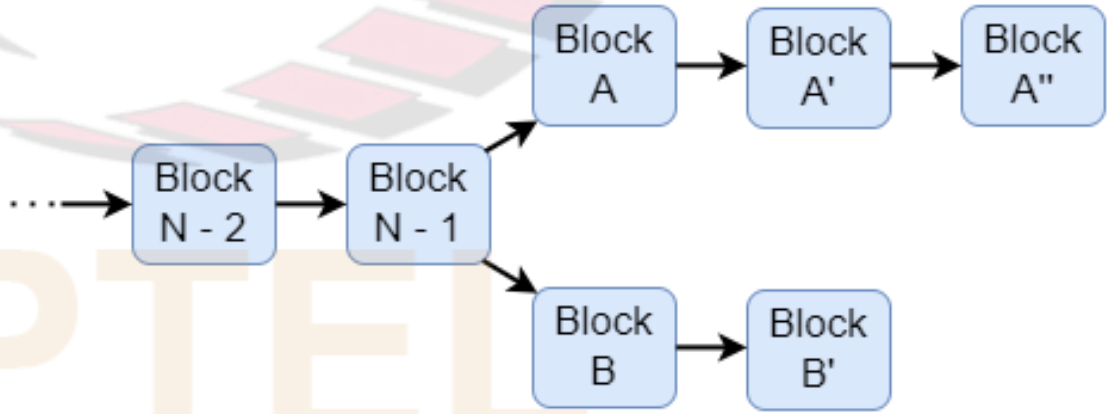
Blockchain Forks (contd.)

- A blockchain fork is shown in fig. on left
- Both branches will have some proportion of the miners in Bitcoin network working to extend them
- It is possible that valid blocks are found once again around the same time on both branches and broadcast on network
- This results in situation shown in fig. on right
  - blocks A' and B' were found by miners trying to extend the branches containing blocks A and B, respectively
- How is a blockchain fork resolved?



Resolution of Blockchain Forks

- Due to randomness inherent in mining process:
  - ❑ it is unlikely that both branches in a blockchain fork get extended to equal heights indefinitely
  - ❑ eventually one branch will become longer than the other
- An example of such a situation is shown in fig.
  - ❑ branch starting from block A has been extended to height $N + 2$, while branch starting from block B has been extended to height $N + 1$
- The Bitcoin protocol requires the network nodes to *switch to the longest branch they become aware of*
- E.g., when the block A'' is received by the miner nodes which are working on extending the branch starting at block B:
  - ❑ they will switch to the branch starting at block A and
  - ❑ begin mining candidate blocks which have block A'' as their previous block
  - ❑ note that they will request the intermediate blocks A and A' from the peer who communicated block A'' to them
- Eventually, the branch consisting of blocks B and B' will no longer be extended
  - ❑blocks belonging to such abandoned branches are called "*stale blocks*"
  - ❑they are eventually deleted

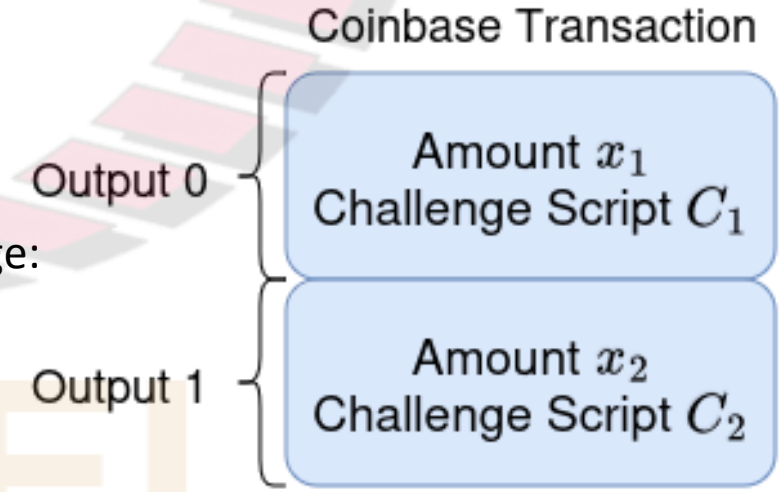


Resolution of Blockchain Forks (contd.)

- In summary:
  - by having all nodes switch to the longest branch, the Bitcoin protocol ensures that only a single linear list of blocks survives after the resolution of blockchain forks
  - the network is said to have achieved *"consensus"* about which linear list of blocks constitute the blockchain
- What happens to the transactions in the stale blocks?
  - A transaction is valid only if it belongs to a block which survives after any blockchain forks have been resolved
  - The coinbase transactions in stale blocks become invalid
  - A regular transaction in a stale block may already be present in one of the blocks which survived after fork resolution
    - If not, it is added back to the mempool of transactions which nodes use to construct new candidate blocks

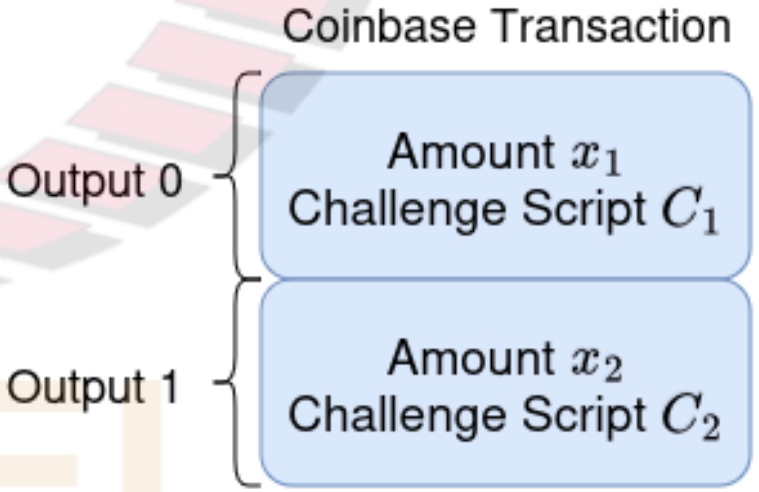
Bitcoin Transactions

- Recall: each block contains one coinbase transaction and 0 or more regular transactions
- A Bitcoin transaction encodes a transfer of bitcoins between entities
- A destination of the transfer in a transaction is called an "*output*"
- A single transaction can have several outputs
  - Each output can serve as a source of bitcoins in a subsequent transaction
- When previous transaction outputs are specified as sources of bitcoins in a transaction, they are called "*inputs*"
- A coinbase transaction:
  - has no input and at least one output
- Why is there no input?
  - since the source of bitcoins is not a previous transaction output, but the block reward, i.e., sum of block subsidy and transaction fees from the transactions in the block
- Each output in the coinbase transaction specifies the following two items (see fig.):
  - Amount of bitcoins from the block reward which are associated with this output
  - A script which specifies the conditions under which the bitcoins associated with this output can be spent
- A script in an output can be considered as a challenge:
  - Any entity which provides a satisfactory response can transfer the bitcoins associated with the output



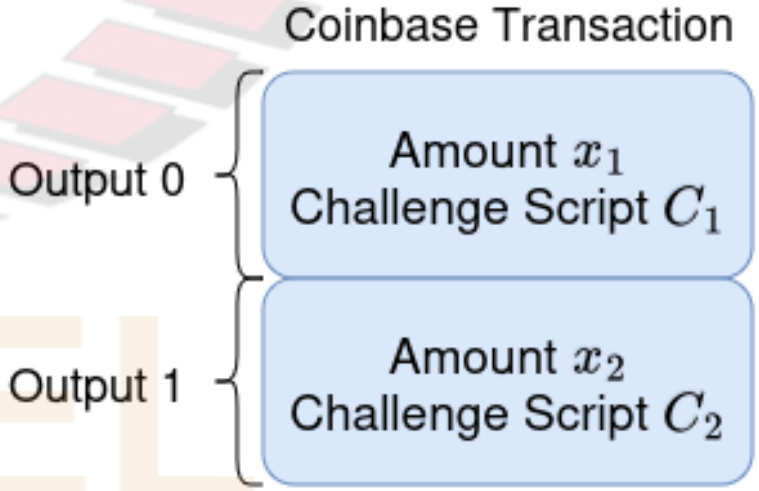
Blockchain Transactions (contd.)

- Sum of amounts in all the outputs of the coinbase transaction should not exceed:
  - ❑ the block reward
- Consider e.g. in fig.; suppose block reward is R
- Then $x_1 + x_2 \leq R$
- Typically equality holds
- If $x_1 + x_2 < R$, then:
  - ❑ the $R - x_1 - x_2$ bitcoins from the block reward become unspendable
- Note that a single output in the coinbase transaction is sufficient for a miner to gain control of the block reward
- Why are multiple outputs used?
  - ❑ Each input in a regular transaction unlocks *all* the bitcoins associated with a previous transaction output
  - ❑ Hence, multiple outputs give the miner flexibility to distribute the block reward to multiple addresses



Example

- We now study an example of a challenge script and a satisfactory response to it
- Consider a miner who creates a block:
  - ❑ wants the block reward to be transferred to addresses it owns
  - ❑ an address is a public key of the miner and is called a “Pay-to-Public-Key (P2PK)” address
  - ❑ “ownership” of a P2PK address means knowing the corresponding private key
- Then challenge script in an output of the coinbase transaction will contain a P2PK address
- This challenge script will require anyone who wants to spend the bitcoins to provide a response script, which contains:
  - ❑ a digital signature created using the private key corresponding to the P2PK address
- Note that this digital signature can be verified using the public key (i.e., P2PK address)



Regular Transactions

- To spend the bitcoins earned in a coinbase transaction, miner needs to create a regular transaction
- Regular transactions have:
  - ❑ at least one input and
  - ❑ at least one output
- Each input specifies three items:
  1) The transaction identifier (TXID) of a previous transaction on the blockchain
     - ❑ the TXID of a transaction is its double SHA-256 hash
  2) The index of an output in the previous transaction
     - ❑ the first output in a transaction has index 0, the second output has index 1 and so on
  3) A response script which satisfies the conditions required to spend the bitcoins in the output
- Each input unlocks *all* the bitcoins associated with the output of a previous transaction (coinbase or regular)
- Outputs of a regular transaction have same format as the outputs of a coinbase transaction
  - ❑ each specifies amount of bitcoins being associated with that output and a challenge script

Regular Transactions (contd.)

- Amounts in regular transaction outputs can take any values such that:
  - sum of amounts does not exceed total amount of bitcoins unlocked by the inputs of the transaction
- Suppose a transaction has N inputs and M outputs
  - let i'th input unlock x_i bitcoins from a previous transaction output
  - let j'th output specify an amount of y_j bitcoins
- The transaction is valid if:
  - $\sum_{j=1}^{M} y_j \leq \sum_{i=1}^{N} x_i$
- The difference $\sum_{i=1}^{N} x_i - \sum_{j=1}^{M} y_j$ is:
  - the transaction fees paid to the miner who includes this transaction in a block

Example

- The amounts must satisfy:
  - $y_1 + y_2 \leq x_1 + x_2 + x_3$
- Transaction fees:
  - $x_1 + x_2 + x_3 - y_1 - y_2$