The Bitcoin Cryptocurrency: Part 5

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

Determination of Transaction Fee

- Recall: max. size of a block is 4 MB
- Miners seek to maximize the total transaction fees earned from a block
- Miners prefer to include in candidate blocks transactions with:
  - ❑ high transaction fees and
  - ❑ small sizes
- In particular, let the "transaction fee per byte" or "fee rate" of a transaction be defined to be:
  - ❑ transaction fee/ size of transaction
- Miners prefer transactions with high fee rates
- So nodes that want to add their regular transaction to the blockchain need to pay a fee rate that is competitive with other transactions being broadcast on the network
- How can a node that wants to add its regular transaction to the blockchain find out what fee rate is competitive?

Determination of Transaction Fee (contd.)

- When a transaction t with fee rate r is first heard by a node, the height h_1 of the latest block in blockchain is recorded
- If t gets included in a block of height h_2:
  - $h_2 - h_1$ gives an estimate of the expected delay incurred by a transaction which pays fee rate r
- Using such estimates from several transactions, a competitive fee rate can be estimated as a function of:
  - the amount of delay in blocks that the node creating the transaction is willing to tolerate

Unspent Transaction Output (UTXO)

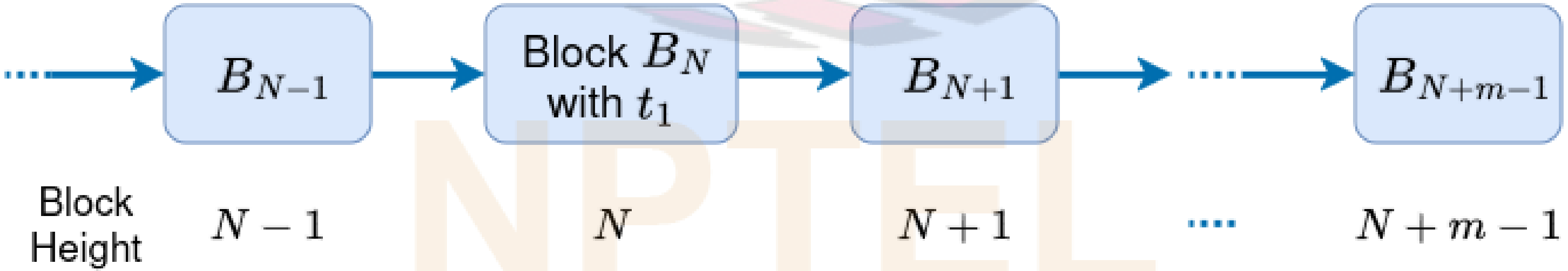
- Recall: when an output of a previous transaction is unlocked by an input of a later transaction:
  - all the bitcoins in the output are spent by the input
- Hence, a transaction output can be in one of only two states: “spent” and “unspent”
- UTXOs refer to outputs in transactions recorded on the blockchain which have not been unlocked by inputs of later transactions
- The total amount of bitcoins owned by an address is:
  - sum of the amounts in all the UTXOs it can unlock
- When a blockchain fork occurs, the sets of UTXOs seen by different nodes in the Bitcoin network differ
  - once the fork is resolved, the UTXO set seen by all the nodes in the Bitcoin network will be identical

Confirmations

- Suppose Alice wants to use bitcoins to pay for some goods that Bob is selling
- Alice creates a new regular transaction, say t_1, such that:
  - ❑ one or more of its inputs unlock UTXOs that Alice owns
  - ❑ one of its outputs contains the amount of bitcoins Alice wants to pay Bob and a challenge script such that:
    - o to create a valid response to it, a private key owned by Bob is required
- Alice broadcasts the transaction t_1 on the Bitcoin network
- Miners include it in the candidate blocks they are mining
- When should Bob hand over the goods to Alice?
  - ❑ Bob keeps scanning the new blocks being added to the blockchain for t_1
  - ❑ once Bob sees a new block with t_1 included in it, he will wait for the branch containing this block to grow further before handing over the goods to Alice

Confirmations (contd.)

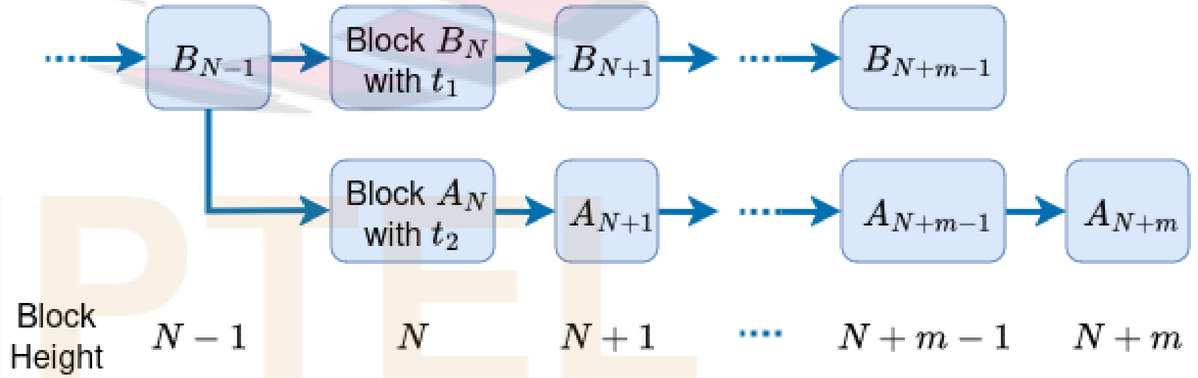
- In particular:
  - if the transaction t_1 is included in a block B_N at height N, then t_1 is said to have received one confirmation
  - when a valid block B_{N+1} at height $N+1$ is added to the blockchain with B_N as its previous block, the transaction t_1 is said to have received two confirmations
  - more generally, when $m-1$ valid blocks have been appended to the block B_N, the transaction t_1 is said to have received m confirmations
- Bob waits until t_1 has received m confirmations before handing over the goods to Alice
  - m is chosen depending on the value of the goods
- Why does Bob wait for m confirmations, instead of handing over the goods to Alice after one confirmation has been received?
  - to safeguard against a "*double spending attack*" by Alice in which the transaction t_1 can be cancelled



Double Spending Attack

- Suppose:
  - ❑ Bob asks Alice for x bitcoins as payment for the goods
  - ❑ Alice can unlock a UTXO O_A which has at least x bitcoins
- A double spending attack proceeds as follows:

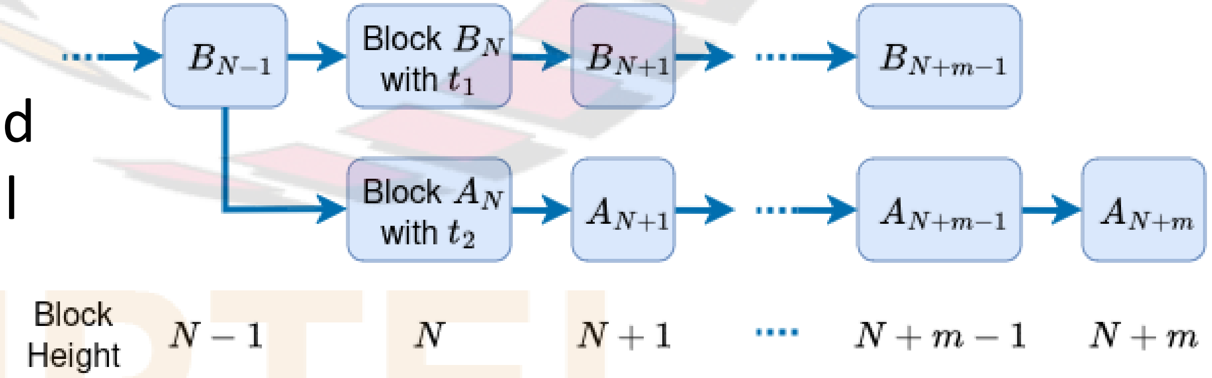
1) Alice creates two transactions t_1 and t_2:
   - ❑ in t_1, an input unlocks O_A and an output pays Bob x bitcoins
   - ❑ in t_2, an input unlocks O_A, but x bitcoins are paid back to Alice in an output
   - ❑ Note that the transactions t_1 and t_2 conflict with each other because they both spend the same UTXO O_A; only one of them can be included in blockchain
2) Alice broadcasts t_1 on the Bitcoin network for inclusion in the blockchain
   - ❑ she keeps t_2 a secret
3) Suppose a miner includes t_1 in a valid block B_N at height N
   - ❑ Bob waits until t_1 has received m confirmations before handing over the goods to Alice
4) Immediately after broadcasting t_1, Alice begins work on constructing a branch containing t_2
   - ❑ she does not announce the valid blocks found on this branch to the Bitcoin network before Bob transfers the goods to her



Double Spending Attack (contd.)

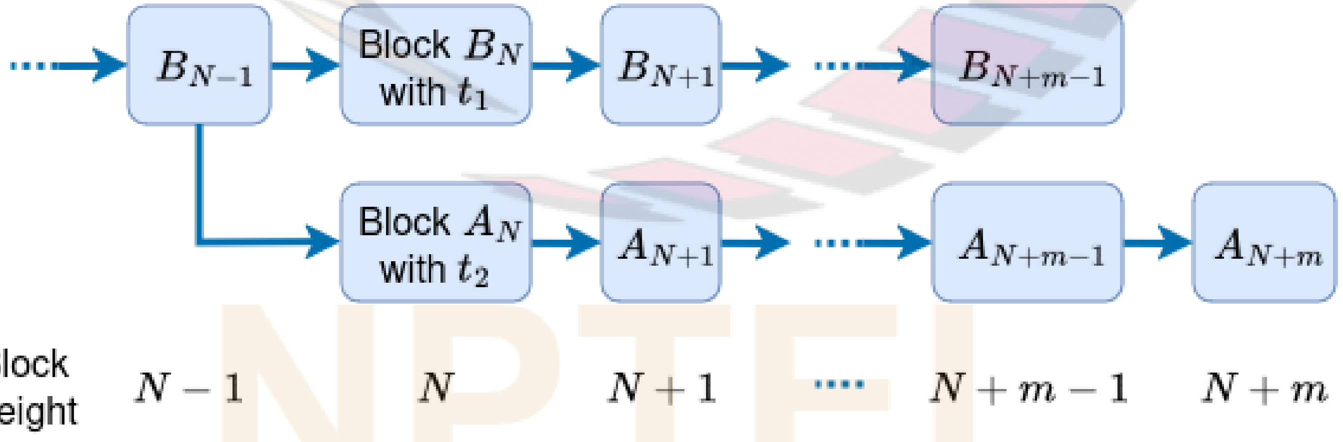
5) After receiving the goods, if Alice succeeds in creating a branch containing t_2 which is longer than the branch containing t_1, then:
   - ❑ she broadcasts all the blocks in the t_2 branch in the Bitcoin network
6) All the nodes in the Bitcoin network will eventually switch to the t_2 branch and the t_1 branch will be abandoned

- Note: usually, transactions present in blocks in abandoned branches are added back to the transaction pool (mempool) if they have not already appeared in the surviving branch
  - ❑ Miners use this transaction pool for constructing new candidate blocks
  - ❑ However, miners which have switched to the t_2 branch will not add t_1 to their mempools since:
    - ○ it conficts with t_2



Double Spending Attack (contd.)

- In summary:
  - ❑Bob has already transferred the goods to Alice
  - ❑but the x bitcoins he thought he received from Alice in t_1 are back in Alice's possession
  - ❑since Alice can now spend these bitcoins again, this attack is called a “double spending attack”



Number of Confirmations m

- Recall: in above example:
  - Bob waits until t_1 has received m confirmations before handing over the goods to Alice
- How many confirmations m should Bob wait for?
- Assuming that Alice controls less than 50% of the network hash rate:
  - the success probability of a double spending attack decreases as m increases
- But m cannot be very large as each confirmation takes approximately 10 minutes to appear
  - so a customer, irrespective of whether he/ she is honest or malicious, will experience a delay of about $10m$ minutes before the seller transfers the goods to him/ her
- Several merchants in the Bitcoin network wait for six confirmations ($m = 6$), which corresponds to a delay of about an hour before goods are transferred from merchant to customer
  - smaller or larger values of m may be used by merchants depending on the value of the goods being sold