Bitcoin Mechanics

Thierry Sans

Properties of Bitcoin

- Uses Elliptic Curve Public Keys (secp256k1) and ECDSA signature algorithm
- UTXO Blockchain (Unspent Transaction Output)
- Block size limit: 1 MB (~2000 transactions/block)
- Block time : ~10 minutes
- Consensus: Proof of Work (coming later)

Propagation Time

According to the paper "*Information propagation in the bitcoin network*" by Decker and Wattenhofer (2013):

> *The **median time** until a node receives a block is **6.5 seconds** whereas **the mean** is at **12.6 seconds**.*
>
> *The long tail of the distribution means that even **after 40 seconds there still are 5% of nodes that have not yet received the block***

- It is hard to maintain data consistency and avoid double spending attack (rf lecture 1)

The Bitcoin solution : Mining

Confirming transaction into blocks

- Miners validate every transaction broadcasted on the network and add them to a mempool of unconfirmed transactions
- Approximately every 10 minutes, *one node is selected* (see consensus later) to create a block containing all unconfirmed transactions and broadcast that block to the network to be added to the blockchain
- All blocks validate the new node before adding it to their own copy of the blockchain

How is this solving the problems

➡ New transactions cannot use input UTXO of transactions that have not been confirmed yet

✓ Data is always consistent and not double spending attack unless two different blocks are mined at the same time (see consensus problem)

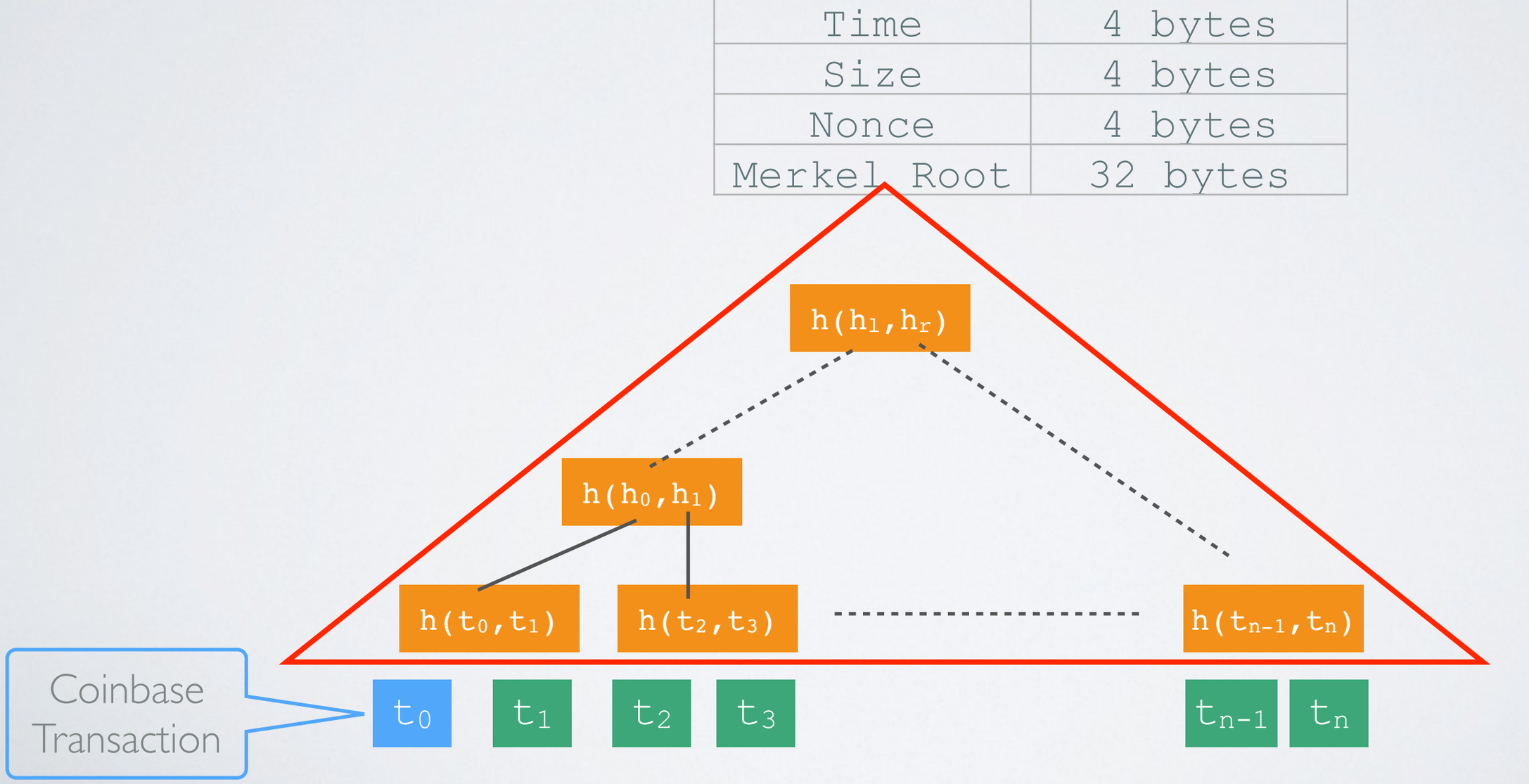
Mining awards

➡ Miners verify/broadcast blocks transactions and broadcast and are rewarded for that work

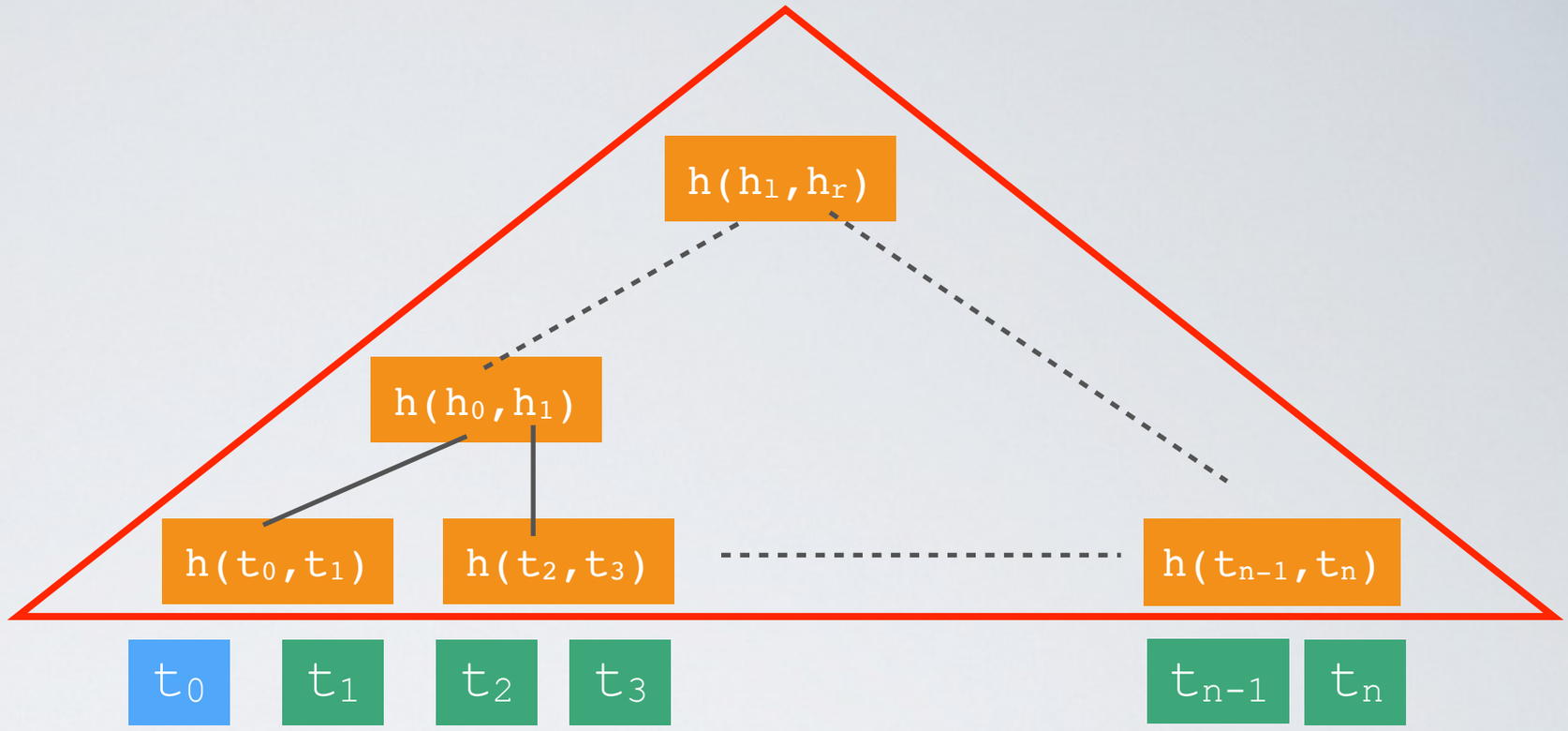
- **Coinbase transaction** (first transaction in the block)
  Currently 6.25 BTC - Block reward halves every four years
  The only way BTC is created (max 21M BTC in total)
- and/or **Transaction Fees** (chosen by the issuer)

Anatomy of the Bitcoin blockchain

Block Hash	32 bytes
Block Id	4 bytes
Old Hash	32 bytes
Time	4 bytes
Size	4 bytes
Nonce	4 bytes
Merkel Root	32 bytes



Properties of Merkle Trees



Why using a Merkle-Tree to record all transactions in the block rather than a "flat list" such as $H(T_0 + T_1 + ... + T_n)$?

✓ Because it is easier to prove than a given transaction belongs to a block

- Using flat list, need all n transactions to build the proof
- Using "Merkle Tree, just $\log_2(n)$ intermediate hashes to build the proof

Let's look at some blocks