CSE 3550/5000: Blockchain Technology

Lecture 6

Ethereum - Part I

Ghada Almashaqbeh

UConn - Spring 2026

Outline

- Ethereum.
  - Work model.
  - Account types.
  - Transaction processing.
  - Mining and consensus.
  - State machine (or blockchain).

Motivation

- Bitcoin has several limitations, among them:
  - No user accounts, tracking a party's balance is done by tracking a chain of transactions.
    - I.e., the UTXO model.
  - Limited scripting language, non-Turing complete that supports a small set of instructions.
- This motivated creating more flexible systems that allow users to instruct the miners to implement any program they wish.
  - Such programs are known as dApps or smart contracts.
- Led to smart contract-enabled blockchains.
- We will explore the first and most widely used smart contract-enabled system; ***Ethereum***.

Ethereum’s History

- Proposed by Vitalik Buterin in 2013 and went live in 2015.
- Supports a permissionless blockchain—started as a PoW one then switched to proof of stake (PoS).
- Its native token called Ether (ETH).
    - Intended to be a utility token to pay for computation.
- Its ability to execute user programs, aka smart contracts, made it a very attractive option for numerous applications.
    - No need to spin a new system, use Ethereum’s infrastructure for building application and even creating new cryptocurrencies (i.e., tokens on top of Ethereum)!
- Because of the DAO incident, Ethereum experienced a hard fork in 2016.
    - Resulted in two separate systems; Ethereum and Ethereum Classic.

The Big Picture

- Build a global computer called Ethereum virtual machine (EVM).
- Allow users to instruct the miners to run user-defined functionalities and computations.
    - Supports a Turing-complete scripting language to write smart contracts.
- View the miners as a global virtual computer to execute smart contracts.
    - Users can deploy smart contracts on the blockchain.
    - Users can invoke functions in a smart contract, and the miners execute such function calls—a computing on demand model.
    - Function calls are packaged as transactions.
    - Code execution cost money, paid in gas units to prevent denial of service attacks (think of infinite loops).

EVM

- Used to implement smart contracts, or distributed applications (dApp), deployed by clients.
- The code of these smart contracts is recorded publicly on the blockchain.
    - Anyone can submit transactions encapsulating fundion calls.
- Each instruction in any contract is executed by every miner in the network.
    - Those who receive the function calls submitted by clients.
- Changes in the smart contract state (i.e., memory/variable values/etc.) after these function calls are also recorded on the blockchain.
    - Can be verified by anyone.

Computation Costs Money I

ETH GAS STATION

- How about DoS attacks?
  - E.g., deploy programs with infinite loops that will stay active in the EVM forever.
- Miners charge a fee for each instruction they execute.
  - These include arithmetic computations, data access, flow control, data storage, etc.
- This fee is called gas.
  - Each instruction type has a designated price in gas units.
  - A transaction issuer has to provide the suitable fee (total gas cost) in order for the miners to implement the requested operations.

Computation Costs Money II

ETH GAS STATION

- Gas is purchased using Ether.
    - So a transaction issuer sends Ether that is used to buy gas based on the gas price it is willing to pay. This price is specified in the transaction.
- Gas price is not fixed, it is miner dependent.
    - Miners announce their gas prices, check https://etherscan.io/gastracker
    - The higher the gas price a tracation issuer is willing to pay, the larger the number of miners willing to process that transaction.
- A miner computes the number of gas units, then charges the issuer in Ether.
- Any extra fees are refunded back to the issuer.

ETH GAS STATION

Computation Costs Money III

- The total cost of a transaction is computed as follows:

$$\text{Total cost} = \text{gas used} \times (\text{base fee} + \text{tips})$$

- **Gas used** is the actual cost of a transaction, e.g., currency transfer cost 21000 gas units.
    - For function calls, this cost depends on the exact computation to be performed. See Appendix G in Ethereum’s yellow paper https://ethereum.github.io/yellowpaper/paper.pdf, e.g. ADD cost 3 gas units.
- **Base fee** is computed by the network protocol and may change per block. This is burned to manage inflation.
- **Tips** are any additional gas units the sender is willing to pay to boost the priority of their transaction. This is collected by the miner who mines the block that will include this transaction.

Main Components

- Similar to other permissioned, public cryptocurrencies, Ethereum’s main components are:
    - P2P network.
    - miners/clients.
    - Transactions.
    - Mining and consensus rules.
    - Blockchain.
    - Economic security.
- Different from UTXO-based cryptocurrencies, Ethereum implements an **account-based model**.
    - A more natural option that mirrors conventional bank accounts in practice.

Account Types

- **(1) Externally Owned Accounts (EOAs):**
  - Associated with a private/public key pair (contains the balance owned by the account owner and a nonce tracking the number of transactions issued so far).
  - The user who owns this key has full control of the currency in this account.
- **(2) Contract account:**
  - Associated with a smart contract code (it contains the code, nonce, program state including coin balance).
  - No private key, it is owned and controlled by the contract code.
- Both account types have addresses.
  - EOA: derived from its public key (hash then take the least 20 bytes of the hash as the address).
  - Contract account: derived from the creator's address (the user who deployed the smart contract) and thier account nonce.
- One needs an EOA to deploy a contract.

Transactions I

- Transactions can be initiated by EOAs.
- To prevent replay attacks, each EAO has a counter that increments after each issued transaction (usually called a nonce).
- The notion of accounts and contracts make the structure of a transaction much different than Bitcoin's one.
  - No need to reference other transactions as input.
  - Just reference the account that the sender owns (this will be used to deduct gas fees and the transferred currency).
- Transactions can be:
  - Standard currency transfer.
  - Contract deployment.
  - Function calls.
- A contract account can issue transactions if triggered by an EOA and based on the contract logic (usually called internal transactions).

Transactions II

- The destination of a transaction can be:
  - EOA:
    - Usually used for currency transfer.
    - Leads to updating the balance of the sender and receiver.
  - Contract:
    - Causes a code in the contract to be executed using the data in the transaction as input.
    - Updates the contract state on the blockchain (and EOAs if any).
  - The address zero.
    - Used when deploying a contract (known as registering the contract on the blockchain).
    - The payload is the compiled code of the contract.

Transactions III

- The fields of a transaction are:
  - Nonce (or a sequence number).
  - Gas price (gas unit price the issuer is willing to pay).
  - Gas limit (total gas the issuer is willing to pay for the transaction).
  - Recipient.
  - Value (amount of currency to be transferred including the fees).
  - Data (function inputs, etc.).
  - Signature (Ethereum also uses ECDSA for signatures and Keccak-256 for hashing)).
- Processing a transaction means validating its format, fees, updating account status (if any), execute a function call and update a contract state (if any), and refund of extra fees (if any).
- Each transaction is recorded on the blockchain.

Ethereum’s Blockchain Explorer

- Visit https://etherscan.io/
  - How long does it take miners to generate a new block on average?
  - Does the shorter block time increase the transactions per second (check the TPS value under resources -> charts and stats)? Why is that?
  - Are 6 blocks enough to confirm a transaction in Ethereum as in Bitcoin? Why?

Mining – Old PoW I

- Up until late Sep, 2022, mining was proof-of-work based, with a slightly different version than the one used in Bitcoin.
    - Called Ethash.
- Ethash is a memory-bound algorithm instead of computation-bound.
    - To be an ASIC-resistant algorithm that is controlled by memory access cost instead of computation cost.
- At a basic level, each miner generates a pseudorandom dataset, called a DAG (Direct Acyclic Graph), that is expanded every 30K blocks.
    - The seed is derived from the current length of the blockchain.
    - All miners, who have the same blockchain view, will generate the same DAG.

Mining - Old PoW II

- The candidate block header and the nonce (a guess for the hash puzzle solution) are used to select a random subset of the DAG.
- The DAG subset, the header, and the nonce are all hashed together.
- If the output meets the network difficulty, then a valid solution has been found.
- Other miners can verify the work by retrieving only the relevant parts of the DAG and perform one hash operation.
- The number of transactions in a block is specified by the block gas limit, i.e., the max total gas amount spent by all transactions in a block.
    - Used to be 8 million gas units per block.
    - Recently the target gas limit has been raised to 15 million with a maximum of 60 million gas units.

Mining - Current PoS

- Recently, Ethereum moved to proof-of-stake (PoS)
    - The new protocol is called Gasper–a combination of Casper the Friendly Finality Gadget (Casper-FFG) and the LMD-GHOST fork choice algorithm.
    - The timeline has been pushed many times for many years.
    - This is part of an upgrade known as Ethereum 2.0
    - We will study proof-of-stake later (including Gasper showing how miners are selected to mine blocks and how to resolve forks, i.e., consensus).

Ethereum’s Blockchain I

- In its yellow paper, Ethereum's blockchain is defined as “cryptographically secure transactional singleton machine with shared-state.”
    - The blockchain operates as a single machine responsible of tracking all transactions, i.e., a single truth of the system’s state.
- Ethereum’s state machine changes state based on the transactions processed so far.
    - A state machine is a machine that reads an input and changes to a new state based on the output according to some transition function.
    - Internal execution of the smart contract code/transaction is also stack-based. However, this is abstracted from the user through the use of wallets and high level smart contract languages, e.g., Solidity.

Ethereum’s Blockchain II

- This state machine starts with the genesis state (aka genesis block).
- Similar to Bitcoin, transactions are grouped into blocks, and these blocks are chained using their hash.
- Based on the transactions included, the newly mined block defines a new state for the system.
  - An account state can contain the account balance, contract code associated with the account, or any digital information about the system.
  - A state is a mapping between addresses and account states.
- In addition, a block contains an identifier of the new system state.
  - This ID is simply the root of the Merkle tree over all mappings in the state.

Ethereum’s Blockchain III

- The mapping between addresses and accounts is stored in a state tree called Merkle Patricia Trie.
    - A combination of radix trees (or prefix trees/tries) and Merkle trees.
        - (For more information see Chapter 14 in the Mastering Ethereum book).
    - The hash of the tree root node is stored in a block’s header to reflect the new state of the system.
        - The full tree is stored off-chain.
- Each block header also contains hashes of the root nodes of the transactions tree, contract storage tree, and transaction receipt tree for all transactions included in the block.

Miner’s Rewards

- Similar to Bitcoin, miners have two sources of income:
  - Staking rewards (newly minted currency in each newly mined block), a percentage of the stake needed to become a validator, i.e., 32 ETH
    - Decrease over time to reduce inflation.
    - Called staking rewards due to the use of PoS.
    - Miners collect these for proposing new blocks, attesting (voting) on these blocks, and syncing the network.
  - Transactions fees in the form of computation cost in gas units and tips.