

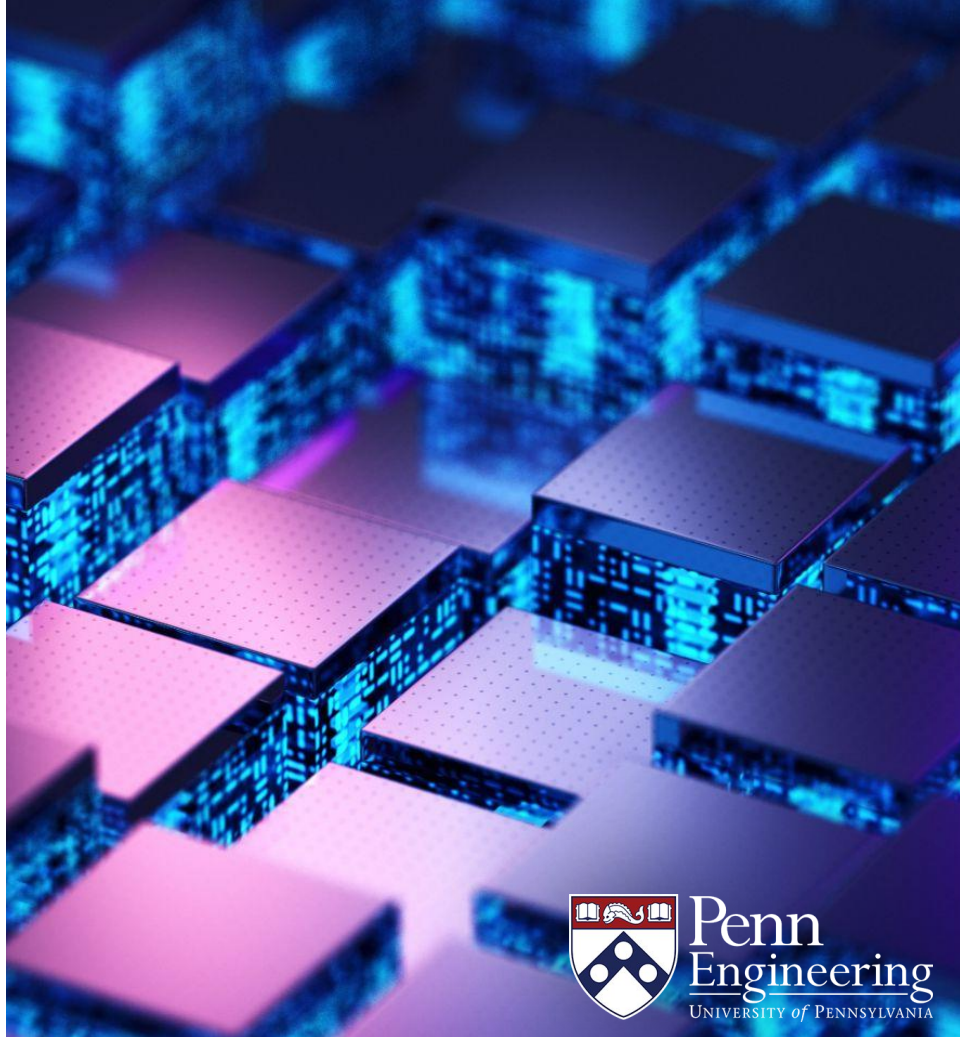
EAS 5830: BLOCKCHAINS

Gas Fees

Professor Brett Hemenway Falk



Penn
Engineering
UNIVERSITY of PENNSYLVANIA



Priority Fees

- Users submit transactions

Priority Fees

- Users submit transactions
- There is a limited amount of block space

Priority Fees

- Users submit transactions
- There is a limited amount of block space
 - Bitcoin produces a block every 10 minutes

Priority Fees

- Users submit transactions
- There is a limited amount of block space
 - Bitcoin produces a block every 10 minutes
 - Ethereum produces a block every 12 seconds

Priority Fees

- Users submit transactions
- There is a limited amount of block space
 - Bitcoin produces a block every 10 minutes
 - Ethereum produces a block every 12 seconds
- How do block producers choose which transactions to include?

Priority Fees

- Users submit transactions
- There is a limited amount of block space
 - Bitcoin produces a block every 10 minutes
 - Ethereum produces a block every 12 seconds
- How do block producers choose which transactions to include?
 - Every transaction includes a fee that is paid to block producer

Priority Fees

- Users submit transactions
- There is a limited amount of block space
 - Bitcoin produces a block every 10 minutes
 - Ethereum produces a block every 12 seconds
- How do block producers choose which transactions to include?
 - Every transaction includes a fee that is paid to block producer
 - Transactions with higher fees usually get included in a block sooner

Priority Fees

- Users submit transactions
- There is a limited amount of block space
 - Bitcoin produces a block every 10 minutes
 - Ethereum produces a block every 12 seconds
- How do block producers choose which transactions to include?
 - Every transaction includes a fee that is paid to block producer
 - Transactions with higher fees usually get included in a block sooner
 - Transactions with low fees may have to wait many blocks before being included into the chain

Gas

- Transaction size is measured in “gas”

Gas

- Transaction size is measured in “gas”
- Every EVM operation has a gas cost

Gas

- Transaction size is measured in “gas”
- Every EVM operation has a gas cost
 - ADD → 3 Gas

Gas

- Transaction size is measured in “gas”
- Every EVM operation has a gas cost
 - ADD → 3 Gas
 - MUL → 5 Gas

Gas

- Transaction size is measured in “gas”
- Every EVM operation has a gas cost
 - ADD → 3 Gas
 - MUL → 5 Gas
- Every transaction uses a certain amount of gas

Gas

- Transaction size is measured in “gas”
- Every EVM operation has a gas cost
 - ADD → 3 Gas
 - MUL → 5 Gas
- Every transaction uses a certain amount of gas
 - Gas is an inherent property of the transaction

Gas

- Transaction size is measured in “gas”
- Every EVM operation has a gas cost
 - ADD → 3 Gas
 - MUL → 5 Gas
- Every transaction uses a certain amount of gas
 - Gas is an inherent property of the transaction
 - “Basic” transfer requires 21,000 gas

Gas

- Transaction size is measured in “gas”
- Every EVM operation has a gas cost
 - ADD → 3 Gas
 - MUL → 5 Gas
- Every transaction uses a certain amount of gas
 - Gas is an inherent property of the transaction
 - “Basic” transfer requires 21,000 gas
 - Complex contract calls can use much more

Gas

- Transaction size is measured in “gas”
- Every EVM operation has a [gas cost](#)
 - ADD → 3 Gas
 - MUL → 5 Gas
- Every transaction uses a certain amount of gas
 - Gas is an inherent property of the transaction
 - [“Basic” transfer requires 21,000 gas](#)
 - Complex contract calls can use much more
 - You can use the [Python Web3 Library](#) or [Foundry](#) to estimate gas cost before deploying a transaction on chain

Gas

- Transaction size is measured in “gas”
- Every EVM operation has a [gas cost](#)
 - ADD → 3 Gas
 - MUL → 5 Gas
- Every transaction uses a certain amount of gas
 - Gas is an inherent property of the transaction
 - [“Basic” transfer requires 21,000 gas](#)
 - Complex contract calls can use much more
 - You can use the [Python Web3 Library](#) or [Foundry](#) to estimate gas cost before deploying a transaction on chain
- Ethereum blocks can hold [30M gas worth of transactions](#)

Gas

- Transaction size is measured in “gas”
- Every EVM operation has a [gas cost](#)
 - ADD → 3 Gas
 - MUL → 5 Gas
- Every transaction uses a certain amount of gas
 - Gas is an inherent property of the transaction
 - [“Basic” transfer requires 21,000 gas](#)
 - Complex contract calls can use much more
 - You can use the [Python Web3 Library](#) or [Foundry](#) to estimate gas cost before deploying a transaction on chain
- Ethereum blocks can hold [30M gas worth of transactions](#)
 - Target size is 15M

Gas

- Transaction size is measured in “gas”
- Every EVM operation has a [gas cost](#)
 - ADD → 3 Gas
 - MUL → 5 Gas
- Every transaction uses a certain amount of gas
 - Gas is an inherent property of the transaction
 - [“Basic” transfer requires 21,000 gas](#)
 - Complex contract calls can use much more
 - You can use the [Python Web3 Library](#) or [Foundry](#) to estimate gas cost before deploying a transaction on chain
- Ethereum blocks can hold [30M gas worth of transactions](#)
 - Target size is 15M
- Fees are paid **per unit of gas**

Priority fees (Pre-London)

- Before EIP1559

Priority fees (Pre-London)

- Before [EIP1559](#)
 - Type 0 or “Legacy” transactions

Priority fees (Pre-London)

- Before [EIP1559](#)
 - Type 0 or “Legacy” transactions
 - Sender specifies “gasPrice”

Priority fees (Pre-London)

- Before [EIP1559](#)
 - Type 0 or “Legacy” transactions
 - Sender specifies “gasPrice”
 - Fee is (Amount of gas)*(gasPrice)

Priority fees (Pre-London)

- Before [EIP1559](#)
 - Type 0 or “Legacy” transactions
 - Sender specifies “gasPrice”
 - Fee is (Amount of gas)*(gasPrice)
 - Fee goes to block producer (miner)

Priority fees (Post-London)

- After [EIP1559](#)

Priority fees (Post-London)

- After [EIP1559](#)
 - Type 2 transaction

Priority fees (Post-London)

- After [EIP1559](#)
 - Type 2 transaction
 - Block specifies baseFeePerGas

Priority fees (Post-London)

- After [EIP1559](#)
 - Type 2 transaction
 - Block specifies baseFeePerGas
 - All transactions pay a baseFee (set by protocol)

Priority fees (Post-London)

- After [EIP1559](#)
 - Type 2 transaction
 - Block specifies baseFeePerGas
 - All transactions pay a baseFee (set by protocol)
 - Sender specifies two values

Priority fees (Post-London)

- After [EIP1559](#)
 - Type 2 transaction
 - Block specifies baseFeePerGas
 - All transactions pay a baseFee (set by protocol)
 - Sender specifies two values
 - MaxPriorityFeePerGas

Priority fees (Post-London)

- After [EIP1559](#)
 - Type 2 transaction
 - Block specifies baseFeePerGas
 - All transactions pay a baseFee (set by protocol)
 - Sender specifies two values
 - MaxPriorityFeePerGas
 - MaxFeePerGas

Priority fees (Post-London)

- After [EIP1559](#)
 - Type 2 transaction
 - Block specifies baseFeePerGas
 - All transactions pay a baseFee (set by protocol)
 - Sender specifies two values
 - MaxPriorityFeePerGas
 - MaxFeePerGas
 - Fee is minimum of
 - $(\text{Amount of gas}) * (\text{baseFeePerGas} + \text{MaxPriorityFeePerGas})$
 - $(\text{Amount of gas}) * (\text{MaxFeePerGas})$

Priority fees (Post-London)

- After [EIP1559](#)
 - Type 2 transaction
 - Block specifies baseFeePerGas
 - All transactions pay a baseFee (set by protocol)
 - Sender specifies two values
 - MaxPriorityFeePerGas
 - MaxFeePerGas
 - Fee is minimum of
 - $(\text{Amount of gas}) * (\text{baseFeePerGas} + \text{MaxPriorityFeePerGas})$
 - $(\text{Amount of gas}) * (\text{MaxFeePerGas})$
 - $(\text{Amount of gas}) * (\text{baseFeePerGas})$ is burned

Priority fees (Post-London)

- After [EIP1559](#)
 - Type 2 transaction
 - Block specifies baseFeePerGas
 - All transactions pay a baseFee (set by protocol)
 - Sender specifies two values
 - MaxPriorityFeePerGas
 - MaxFeePerGas
 - Fee is minimum of
 - $(\text{Amount of gas}) * (\text{baseFeePerGas} + \text{MaxPriorityFeePerGas})$
 - $(\text{Amount of gas}) * (\text{MaxFeePerGas})$
 - $(\text{Amount of gas}) * (\text{baseFeePerGas})$ is burned
 - Remainder goes to block producer



\$4.6 Billion in Ethereum Up in Flames Since EIP-1559

NFT and DeFi on the Ethereum network, in particular via OpenSea and Uniswap, are fueling the fire of ETH burning.



By Liam J. Kelly

📅 Jan 21, 2023

🕒 3 min read

Markets

Ether Turns Deflationary Again, Led by Spike in NFT Sales

Nearly one-fourth of ether burned stems from NFT trades over the past seven days, according to data from ultrasound.money.

By Jocelyn Yang ⌚ Jan 23, 2023 at 8:16 p.m. EST Updated Jan 24, 2023 at 11:35 a.m. EST

Prioritizing transactions

- Post 1559 transactions have “Priority Fee”

Prioritizing transactions

- Post 1559 transactions have “Priority Fee”
 - [Default node behavior](#) is to include transactions based on priority fee

Prioritizing transactions

- Post 1559 transactions have “Priority Fee”
 - [Default node behavior](#) is to include transactions based on priority fee
 - This is the “greedy” approach to packing blocks

Prioritizing transactions

- Post 1559 transactions have “Priority Fee”
 - [Default node behavior](#) is to include transactions based on priority fee
 - This is the “greedy” approach to packing blocks
 - This works well for “simple” transactions (e.g. transferring ETH)

Prioritizing transactions

- Priority Fees can't express nuanced priority goals

Prioritizing transactions

- Priority Fees can't express nuanced priority goals
- In DeFi people have complex preferences

Prioritizing transactions

- Priority Fees can't express nuanced priority goals
- In DeFi people have complex preferences
 - Backrunning (I want my transaction **after** a target transaction)

Prioritizing transactions

- Priority Fees can't express nuanced priority goals
- In DeFi people have complex preferences
 - Backrunning (I want my transaction **after** a target transaction)
 - Sandwiching (I want to trade **before** and **after** a target transaction)

Prioritizing transactions

- Priority Fees can't express nuanced priority goals
- In DeFi people have complex preferences
 - Backrunning (I want my transaction **after** a target transaction)
 - Sandwiching (I want to trade **before** and **after** a target transaction)
- Flashbots pioneered the notion of “bundles”

Prioritizing transactions

- Priority Fees can't express nuanced priority goals
- In DeFi people have complex preferences
 - Backrunning (I want my transaction **after** a target transaction)
 - Sandwiching (I want to trade **before** and **after** a target transaction)
- Flashbots pioneered the notion of “bundles”
 - “Searchers” submit a bundle of transactions

Prioritizing transactions

- Priority Fees can't express nuanced priority goals
- In DeFi people have complex preferences
 - Backrunning (I want my transaction **after** a target transaction)
 - Sandwiching (I want to trade **before** and **after** a target transaction)
- Flashbots pioneered the notion of "bundles"
 - "Searchers" submit a bundle of transactions
 - Offer a fee for including bundle (at the top of the block)

Prioritizing transactions

- Priority Fees can't express nuanced priority goals
- In DeFi people have complex preferences
 - Backrunning (I want my transaction **after** a target transaction)
 - Sandwiching (I want to trade **before** and **after** a target transaction)
- Flashbots pioneered the notion of “bundles”
 - “Searchers” submit a bundle of transactions
 - Offer a fee for including bundle (at the top of the block)
 - Block builders use a [first-price sealed bid auction to place bundles](#)



Copyright 2020 University of Pennsylvania
No reproduction or distribution without permission.