Module 2: Blockchain Transaction Ordering

Duration: 60 minutes | **Level:** Beginner | **Author:** Obelisk Core

Learning Objectives

By the end of this module, you will:
- Understand how transactions are processed and ordered in blockchain blocks
- Learn about mempool dynamics and transaction prioritization
- Explore block building and validation processes
- Understand the technical mechanisms behind MEV opportunities

---

Transaction Lifecycle Overview

From Submission to Confirmation

```
User submits transaction → Enters Mempool → Miner/Validator selects →
Included in Block → Confirmed
```

Transaction Components

1. **Nonce:** Unique identifier for transaction ordering
2. **Gas Price:** Fee per unit of computation
3. **Gas Limit:** Maximum gas consumption allowed
4. **To Address:** Destination contract or account
5. **Value:** Amount of native token to transfer
6. **Data:** Function call parameters or message
7. **Signature:** Cryptographic proof of ownership

The Mempool

What is the Mempool?

Definition: A waiting area for pending transactions before they're included in blocks.

Ethereum Mempool Characteristics:
- **Public by default:** All transactions visible before inclusion
- **Gas price ordering:** Higher gas transactions prioritized
- **Competitive environment:** Multiple transactions competing for block space
- **Time-sensitive:** Transaction ordering affects MEV opportunities

Mempool Segments

1. High Priority Pool

- Gas Price: 50+ gwei
- Typical Size: 100-500 transactions
- Inclusion Time: 0-2 blocks (0-30 seconds)

2. Medium Priority Pool

- Gas Price: 20-50 gwei
- Typical Size: 200-1000 transactions
- Inclusion Time: 2-10 blocks (30 seconds - 2.5 minutes)

3. Low Priority Pool

- Gas Price: 10-20 gwei
- Typical Size: 500-2000 transactions
- Inclusion Time: 10-50 blocks (2.5-12.5 minutes)

4. Dropped Transactions

- Gas Price: <10 gwei
- May never be included
- Eventually dropped from mempool

Block Building Process

Ethereum Block Structure

```
Block Header
├── Parent Hash
├── State Root
├── Receipts Root
├── Logs Bloom
├── Difficulty
├── Number
├── Gas Limit
├── Gas Used
├── Timestamp
├── Extra Data
└── Mix Hash / Nonce

Transactions
├── Transaction 1 (Gas: 21,000)
├── Transaction 2 (Gas: 45,000)
├── Transaction 3 (Gas: 156,000)
└── ...
```

Block Constraints

- **Gas Limit:** Maximum total gas per block (currently ~15-30 million)
- **Block Time:** ~12-15 seconds on Ethereum
- **Transaction Ordering:** Gas price descending order (by default)

Typical Block Composition

- **Simple transfers:** 80-90% of transactions
- **Smart contract calls:** 10-15% of transactions
- **Complex DeFi transactions:** 1-5% of transactions
- **MEV bundles:** <1% but highest gas prices

Transaction Ordering Strategies

1. Gas Price Ordering (Default)

Method: Sort by gas price (highest first)
Rationale: Maximize validator/miner revenue
Limitations: Predictable, enables front-running

2. FIFO (First-In-First-Out)

Method: Process based on arrival time
Used by: Some alternative chains
Challenges: Requires precise timing synchronization

3. Custom Ordering

Method: Validator chooses specific order
Benefits: Can implement MEV protection
Examples: Random ordering, gas price caps

Transaction Prioritization Factors

Primary Factors

1. **Gas Price:** Highest impact on ordering
2. **Gas Limit:** Affects block gas consumption
3. **Nonce:** Prevents transaction replay
4. **Timestamp:** Transaction validity window

Secondary Factors

1. **Transaction Size:** Larger transactions slower to propagate
2. **Network Congestion:** During high activity periods
3. **Validator Preferences:** Some prefer specific transaction types
4. **MEV Optimization:** Validators may prioritize certain patterns

---

Transaction Propagation

P2P Network Structure

```
Transaction Submitted
     ↓
Local Node
     ↓
Peer-to-Peer Network
     ↓
Global Mempool
     ↓
Mining/Validation Pool
     ↓
Block Inclusion
```

Propagation Speed Factors

1. **Network Topology:** Geographic distribution of nodes
2. **Bandwidth:** Node connection quality
3. **Node Load:** Computational resources available
4. **Geographic Location:** Distance from transaction source

MEV Impact

- **Fastest propagation:** Better chance of inclusion
- **Coordinated bundles:** Submit simultaneously to multiple validators
- **Private mempools:** Avoid public propagation (e.g., Flashbots)

Gas Auction Mechanics

Gas Price Dynamics

```
Normal Conditions:
- Base fee: 20 gwei
- Priority fee: 2 gwei
- Total: 22 gwei

High Demand:
- Base fee: 50 gwei
- Priority fee: 15 gwei
- Total: 65 gwei

MEV Competition:
- Base fee: 30 gwei
- Priority fee: 100+ gwei
- Total: 130+ gwei
```

Priority Fee Calculation

Formula: Miner's extra reward above base fee
Market Forces: Competition determines priority fee levels
MEV Impact: High-priority transactions drive priority fees up

---

Block Building Strategies

Traditional Mining

Process:
1. Collect transactions from mempool
2. Sort by gas price
3. Fill block up to gas limit
4. Solve proof-of-work puzzle
5. Broadcast completed block

Challenges:
- Predictable ordering enables front-running
- No ability to optimize transaction selection
- Limited control over final block content

MEV-Boost Architecture

Components:
- **Builders:** Construct optimal blocks
- **Validators:** Choose best block from builders
- **Relays:** Facilitate block submission

Benefits:
- Validators can select best block without running builder infrastructure
- Competition between builders improves block optimization
- MEV extraction can be shared more efficiently

Front-Running Protection Mechanisms

Commit-Reveal Schemes

Mechanism:
1. **Commit:** Submit transaction hash (not content)
2. **Reveal:** Later submit actual transaction
3. **Benefit:** Transaction content protected during commit phase

Implementation:

```
Step 1: commit_hash = keccak256(transaction_data + salt)
Step 2: Submit commit_hash to contract
Step 3: After time delay, submit transaction_data + salt
Step 4: Contract verifies commit matches reveal
```

Batch Auctions

Concept: All transactions in same time window get same price
Benefits: Eliminates gas price competition
Challenges: Implementation complexity, timing coordination

Randomized Ordering

Method: Randomize transaction order within blocks
Benefits: Reduces predictability of front-running
Challenges: May reduce validator revenue, complex implementation

---

Real-World Block Analysis

Sample Block #18,000,000

Block Statistics:
- **Total Transactions:** 153
- **Gas Used:** 29,847,891 (99.2% of limit)
- **Average Gas Price:** 67.2 gwei
- **Block Time:** 12.3 seconds
- **Total Fees:** 2.01 ETH

Transaction Breakdown:
- Simple transfers: 134 transactions (87.6%)
- ERC-20 transfers: 12 transactions (7.8%)
- DeFi interactions: 5 transactions (3.3%)
- MEV bundles: 2 transactions (1.3%)

MEV Transaction Patterns

Typical MEV Bundle Structure:

```
Bundle Contents:
1. Front-run transaction (sandwich victim's trade)
2. Victim transaction (arbitrage opportunity)
3. Back-run transaction (complete the arbitrage)
4. Liquidation transaction (if applicable)

Gas Pricing:
- All transactions: 150 gwei (2.2x average)
- Bundle submitted via Flashbots relay
- Included in same block (atomic execution)
```

Technical Deep Dive: Transaction Simulation

Why Simulate?

1. **Profit Calculation:** Estimate net profit before execution
2. **Competition Assessment:** Understand competing transactions
3. **Gas Optimization:** Minimize costs while ensuring inclusion
4. **Failure Prevention:** Test before submitting real funds

Simulation Process

```
1. Fork current blockchain state
2. Execute hypothetical transaction
3. Record state changes and gas usage
4. Calculate profitability
5. Compare to competing scenarios
```

Tools and Libraries

- **Tenderly:** Transaction simulation platform
- **Brownie:** Python framework for testing
- **Hardhat:** JavaScript development environment
- **Foundry:** Rust-based smart contract suite

MEV and Network Congestion

High-Activity Periods

Characteristics:
- 2-5x normal transaction volume
- 3-10x higher gas prices
- Increased MEV competition
- Higher failure rates for marginal transactions

MEV Impact During Congestion:
- More arbitrage opportunities due to price volatility
- Increased sandwich attack profits
- Higher gas costs reduce net MEV
- Faster execution critical for success

Famous MEV Incidents

1. **Uniswap V3 LP Incident (June 2023):**
   - $2.4M in sandwich attacks during one hour
   - Triggered by large LP position update
   - Demonstrated MEV on concentrated liquidity pools
2. **LUNA/UST Collapse (May 2022):**
   - Massive arbitrage opportunities during depeg
   - $50M+ in MEV extraction during 48 hours
   - Led to development of L2 MEV protection

3. **Merge Weekend (September 2022):**
   - Reduced MEV due to validator changes
   - 60% drop in sandwich attacks
   - Demonstrated impact of consensus changes

Future Developments

Ethereum Roadmap

1. **Danksharding (2024-2025):**
   - 1,000x increase in data availability
   - New types of MEV opportunities
   - Reduced transaction costs
2. **Proposer-Builder Separation:**
   - Validators focus on consensus
   - Specialized block builders
   - More sophisticated MEV strategies

Alternative Approaches

1. **Pre-trade Privacy:**
   - Transaction contents encrypted until execution
   - Prevents front-running at network level
   - Technical challenges remain
2. **Formal Verification:**
   - Mathematical proofs of MEV protection
   - Protocol-level guarantees
   - Integration with existing systems

Practical Exercise

Transaction Analysis

Scenario: You detect a large DEX trade in the mempool

```
Trade Details:
- Token: UNI
- Size: 10,000 UNI
- Current Price: $8.50
- Trade Impact: Estimated 2% price increase
- Your Capital: 100 ETH
- Gas Budget: 0.5 ETH
```

Tasks:
1. Calculate potential sandwich profit
2. Determine optimal gas price strategy
3. Plan transaction bundle structure
4. Assess competition risks

Solution Framework

```
Step 1: Price Impact Analysis
- Current price: $8.50
- Expected new price: $8.67 (2% increase)
- Profit per token: $0.17

Step 2: Trade Size Optimization
- Max tokens you can trade: 100 ETH / $8.50 = 11.76 UNI
- Realistic trade size: 10 UNI (avoids detection)
- Total profit: 10 × <span class="math-inline" style="display:
inline;"><math xmlns="http://www.w3.org/1998/Math/MathML"
display="inline"><mrow><mn>0.17</mn><mo>&#x0003D;</mo></mrow></math></
span>1.70

Step 3: Gas Cost Consideration
- Estimated gas cost: 0.3 ETH (<span class="math-inline"
style="display: inline;"><math xmlns="http://www.w3.org/1998/Math/
MathML" display="inline"><mrow><mn>450</mn><mi>a</mi><mi>t</mi></
mrow></math></span>1500/ETH)
- Net profit: <span class="math-inline" style="display: inline;"><math
xmlns="http://www.w3.org/1998/Math/MathML"
display="inline"><mrow><mn>1.70</mn><mo>&#x02212;</mo></mrow></math></
span>450 = negative

Conclusion: Opportunity not profitable given gas costs
```

Module Summary

Key Concepts

- **Mempool:** Waiting area for pending transactions
- **Gas Auctions:** Competition for block space
- **Transaction Ordering:** Determines MEV opportunities
- **Block Building:** Process of creating blockchain blocks
- **Front-Running:** Extracting value from transaction knowledge

Technical Skills Acquired

- Understanding blockchain transaction flow
- Knowledge of mempool mechanics
- Familiarity with block building process
- Insights into MEV extraction timing

Next Steps

- **Module 3:** Learn about MEV ecosystem participants
- **Module 4:** Explore specific MEV strategies
- **Module 5:** Understand market impact and ethics
- **Module 6:** Analyze real-world MEV transactions

Quick Check: Test Your Understanding

1. **What primarily determines transaction ordering in Ethereum blocks?**
   - [ ] Transaction timestamp
   - [ ] Gas price (highest first)
   - [ ] Transaction size
2. **What is the typical time between Ethereum blocks?**
   - [ ] 5-8 seconds
   - [ ] 12-15 seconds
   - [ ] 30-45 seconds
3. **How does MEV-Boost improve on traditional mining?**
   - [ ] Faster block times
   - [ ] Separates block building from validation
   - [ ] Reduces gas costs

4. **What is a key limitation of the public mempool?**
    - [ ] Limited transaction capacity
    - [ ] Enables front-running
    - [ ] Slow transaction propagation

This module is part of the MEV Fundamentals course by ObeliskCore Education. For questions or feedback, contact our support team.