Optimal Frontrunning Attacks and How to Stop Them

Max Resnick

Rook

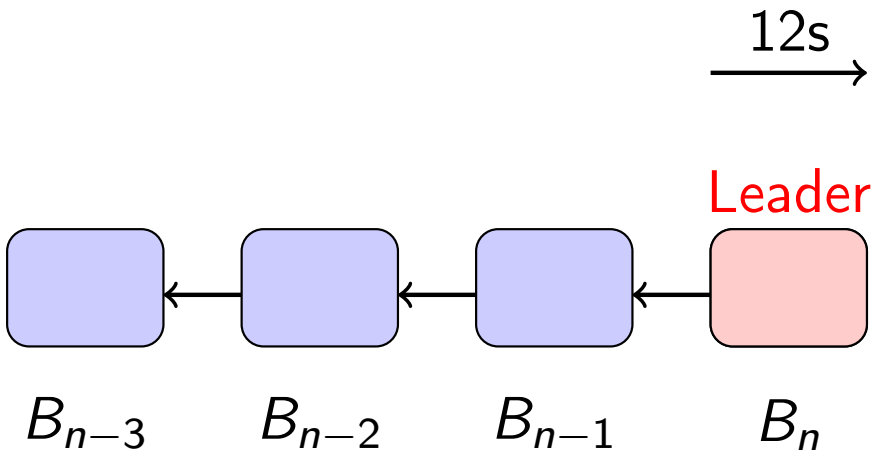
March 16, 2023

Introduction: MEV and Frontrunning

- **MEV (Miner Extractable Value)**: The total value that miners can capture from the *reordering*, *inclusion*, or *censoring* of transactions in a block.
- **Frontrunning**: Bots transact with the pool before your transaction to earn risk-free profits

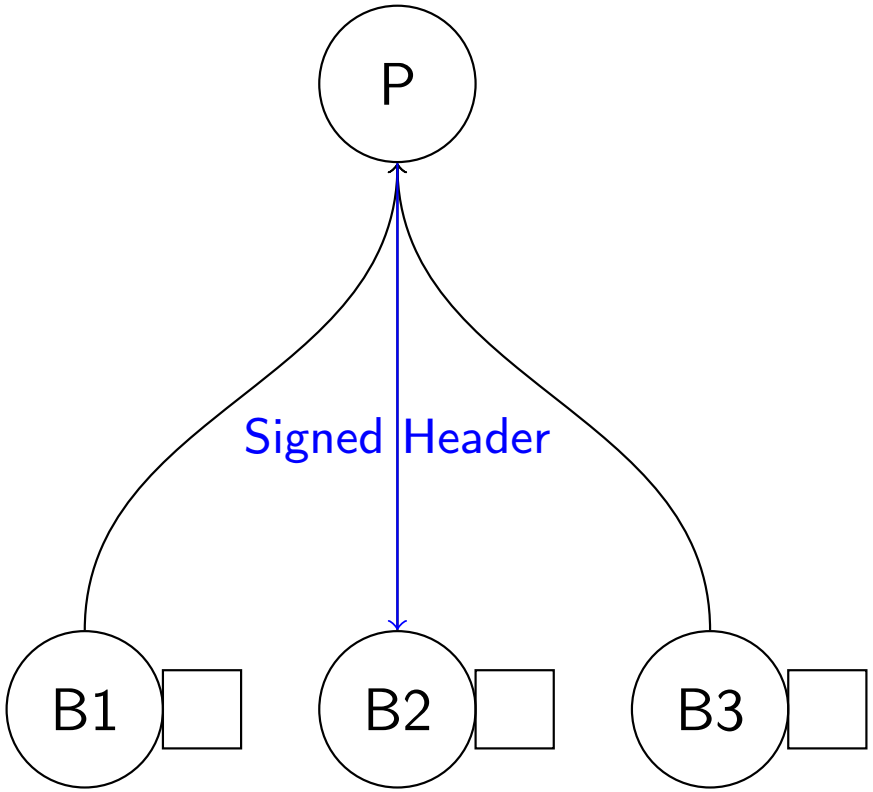
Role of the Leader in Proof of Stake Networks

- In Proof of Stake networks, a **leader** is responsible for proposing and validating new blocks.
- The leader has a **monopoly on write access** to the chain for a specific time slot, e.g., 12 seconds in Ethereum 2.0.
- This privilege allows the leader to:
  - Reorder, include, or censor transactions
  - Exploit MEV opportunities



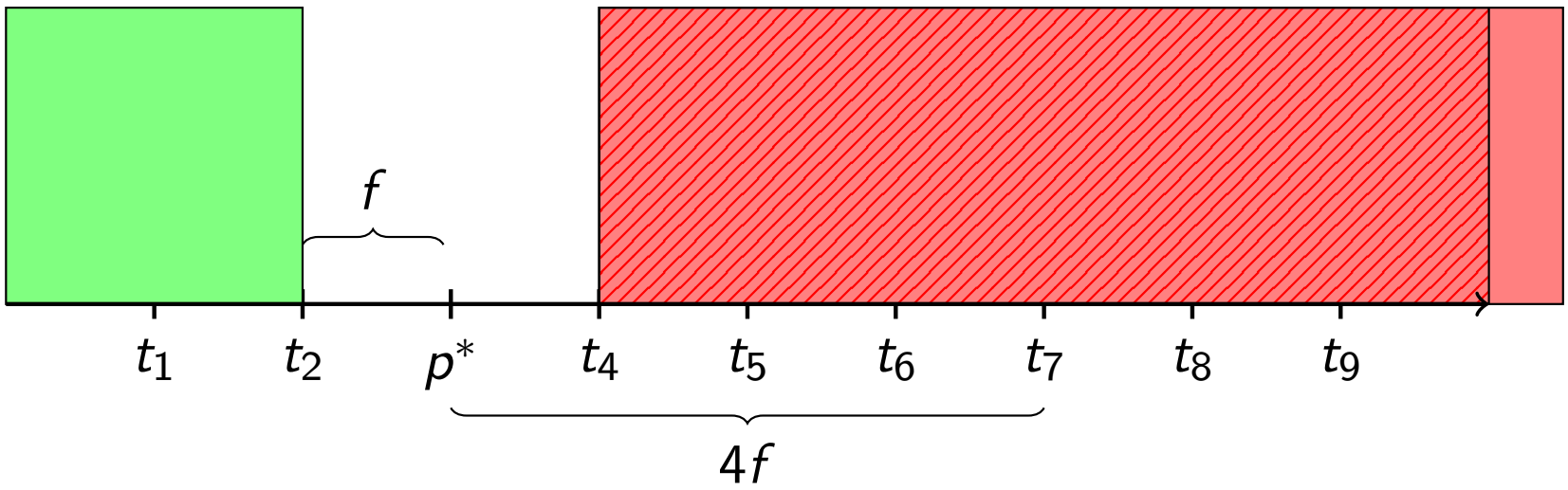
Proposer-Builder Separation

- Builders Create blocks and send them to the proposer. The proposer chooses the most valuable one and signs the header.
- They are auctioning off the right to be the proposer!



Model

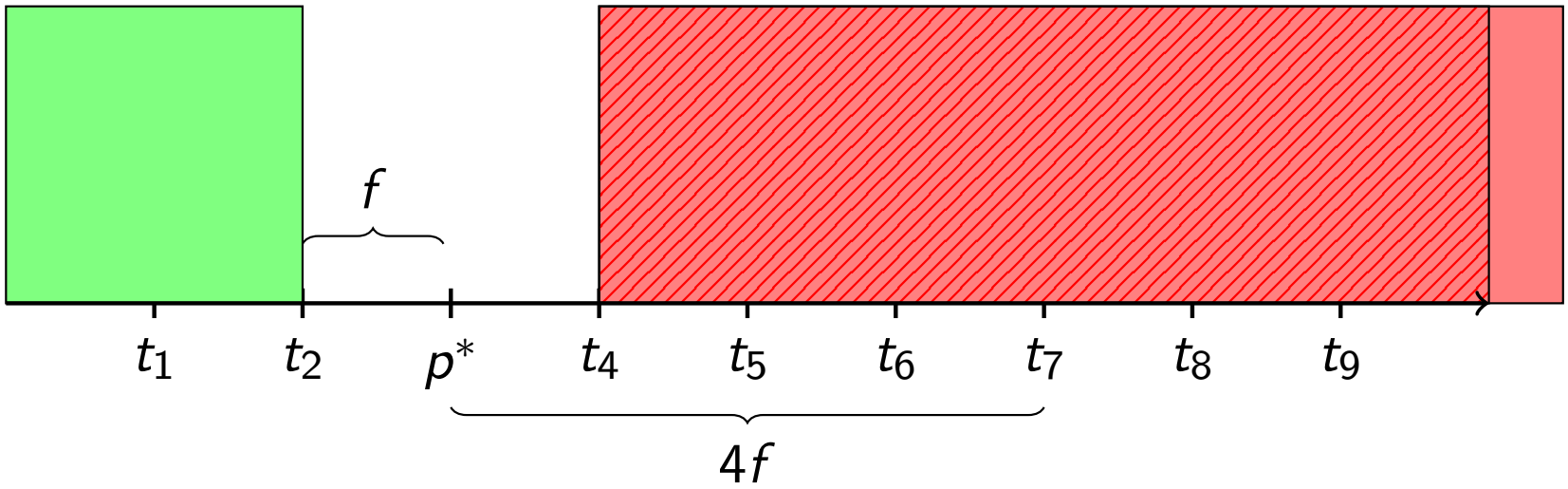
Suppose a trader wants to buy 6 units of liquidity. The average price he gets is $4f$. He sets his slippage tolerance to 0. He pays $6 \cdot 4f = 24f$



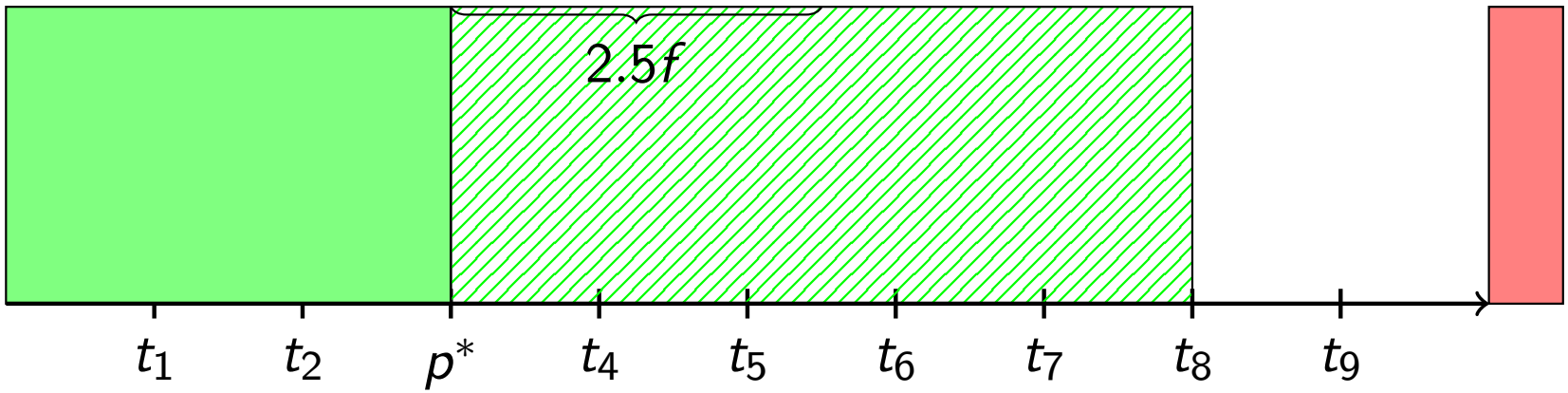
Throughout we assume that the attacker has unlimited liquidity on an external exchange at price p^*.
Based on a Twitter thread by Alex Nezlobin (@0x94305)

Simple Backrun (12.5f)

1. Execute Swap

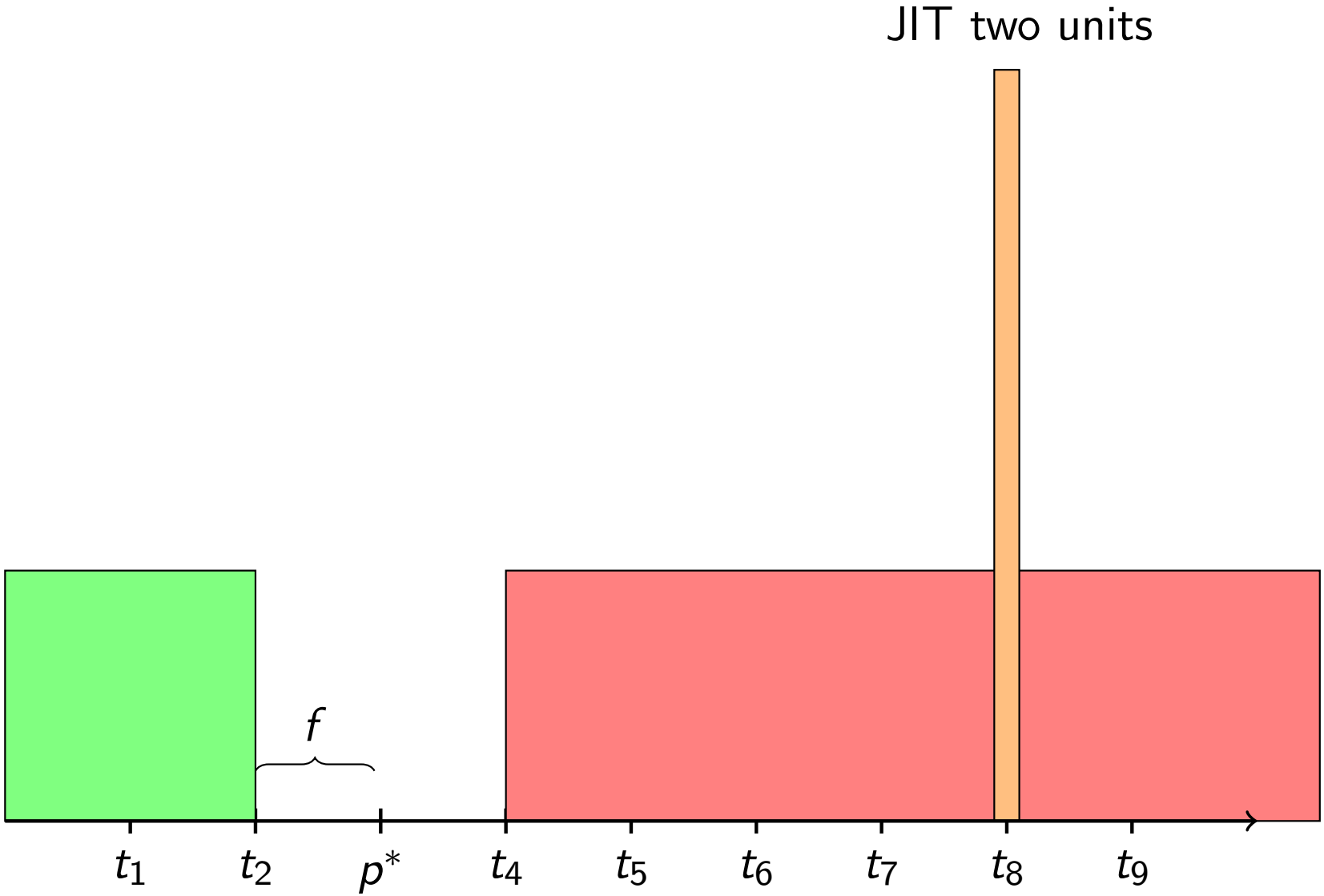


2. Backrun



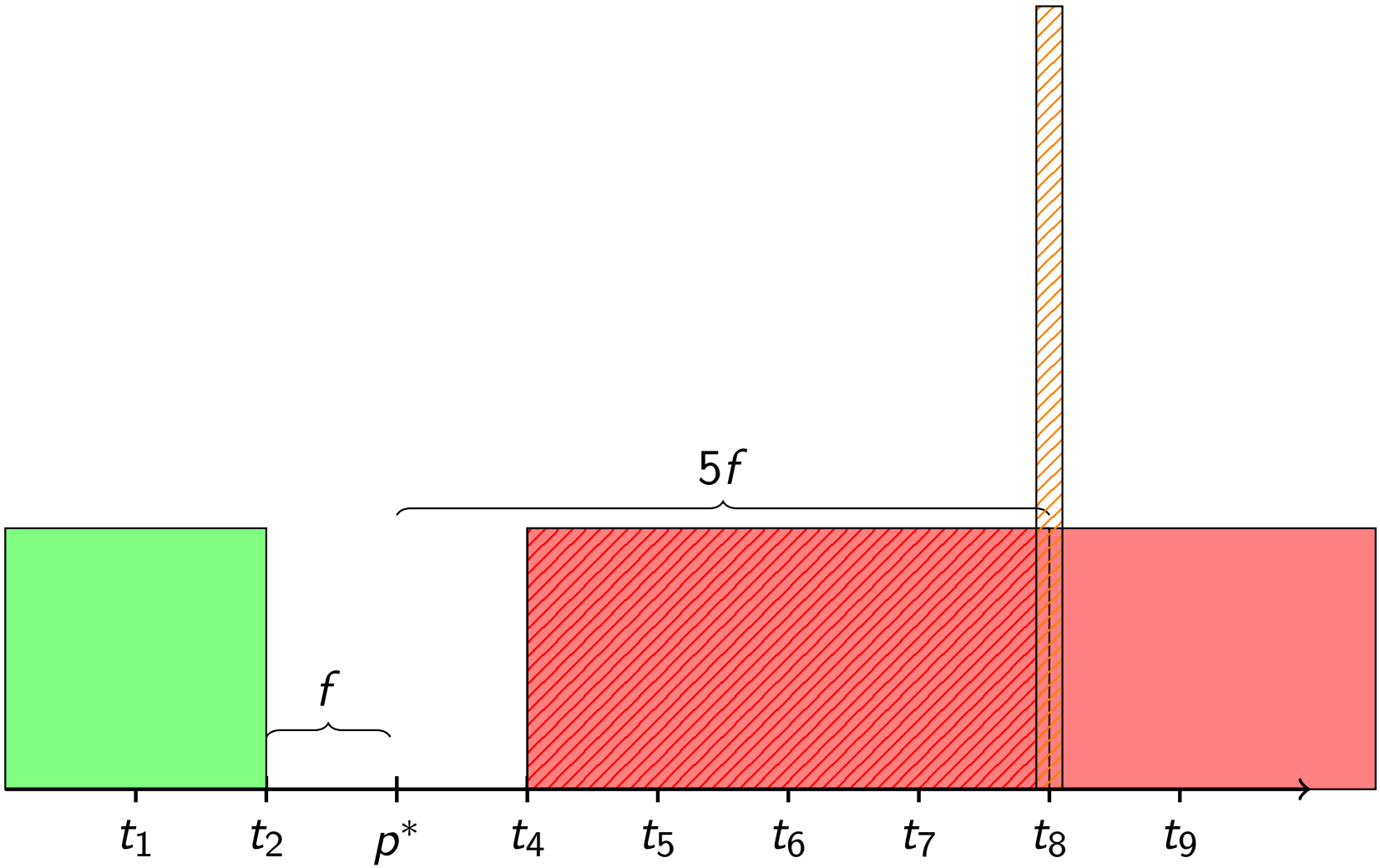
JIT + Backrun ($14.5f$)

1. JIT



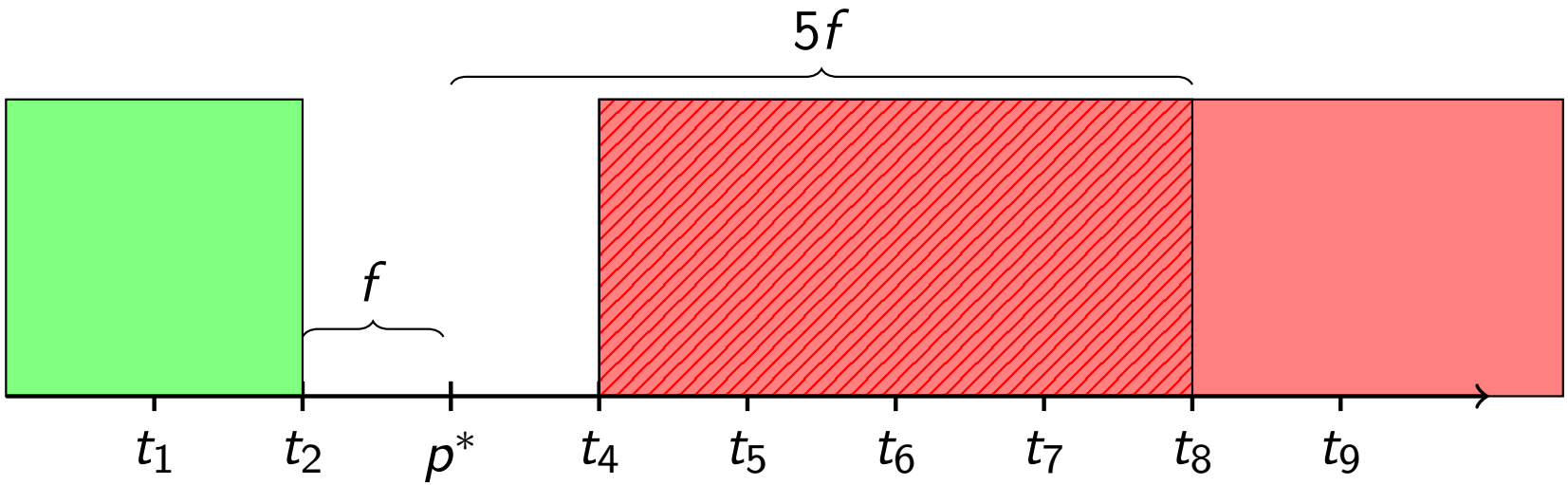
JIT + Backrun ($14.5f$)

2. Execute Trade

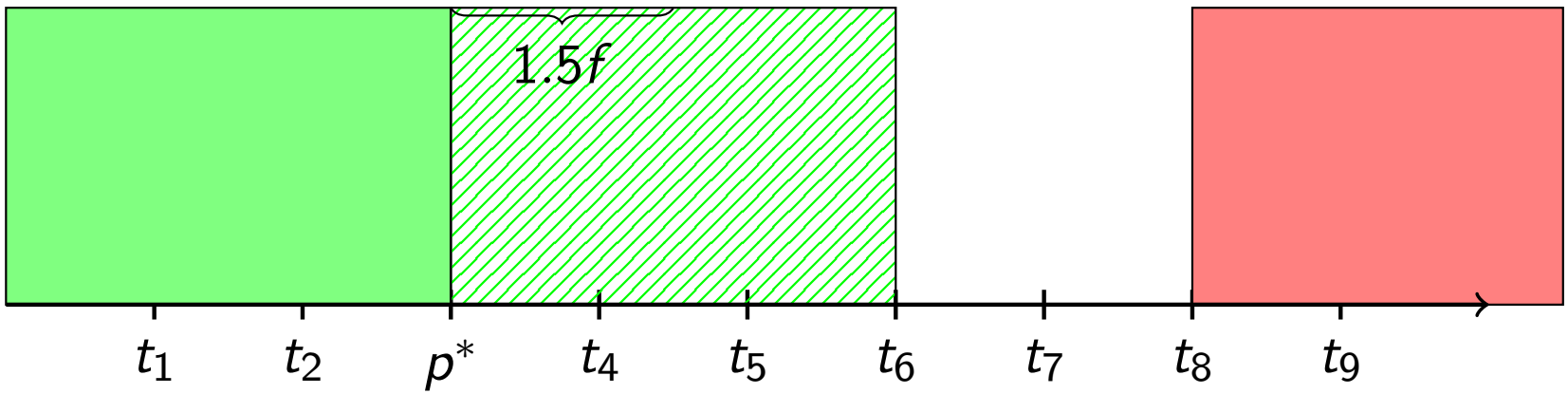


JIT + Backrun ($14.5f$)

3. Remove JIT

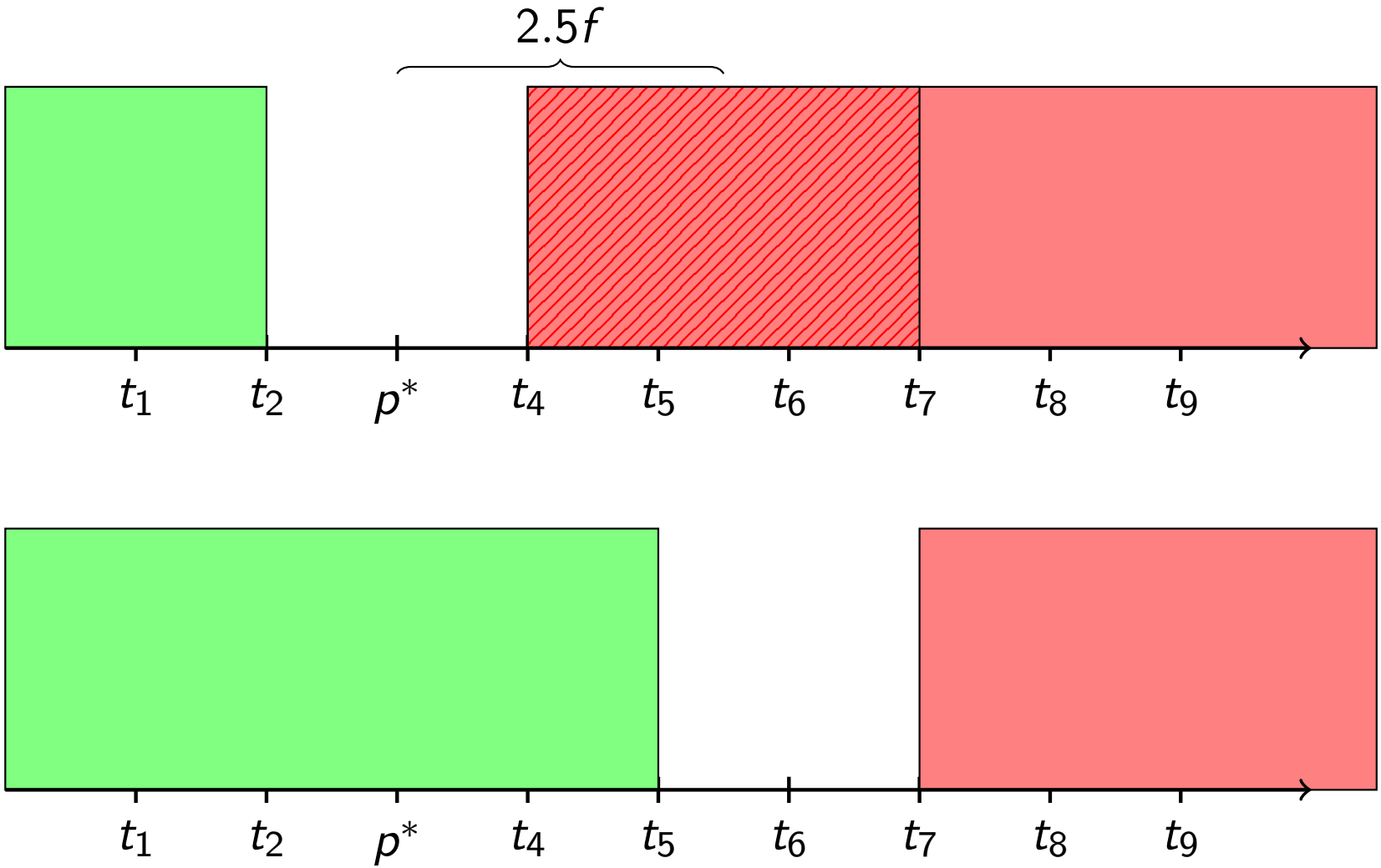


4. Backrun



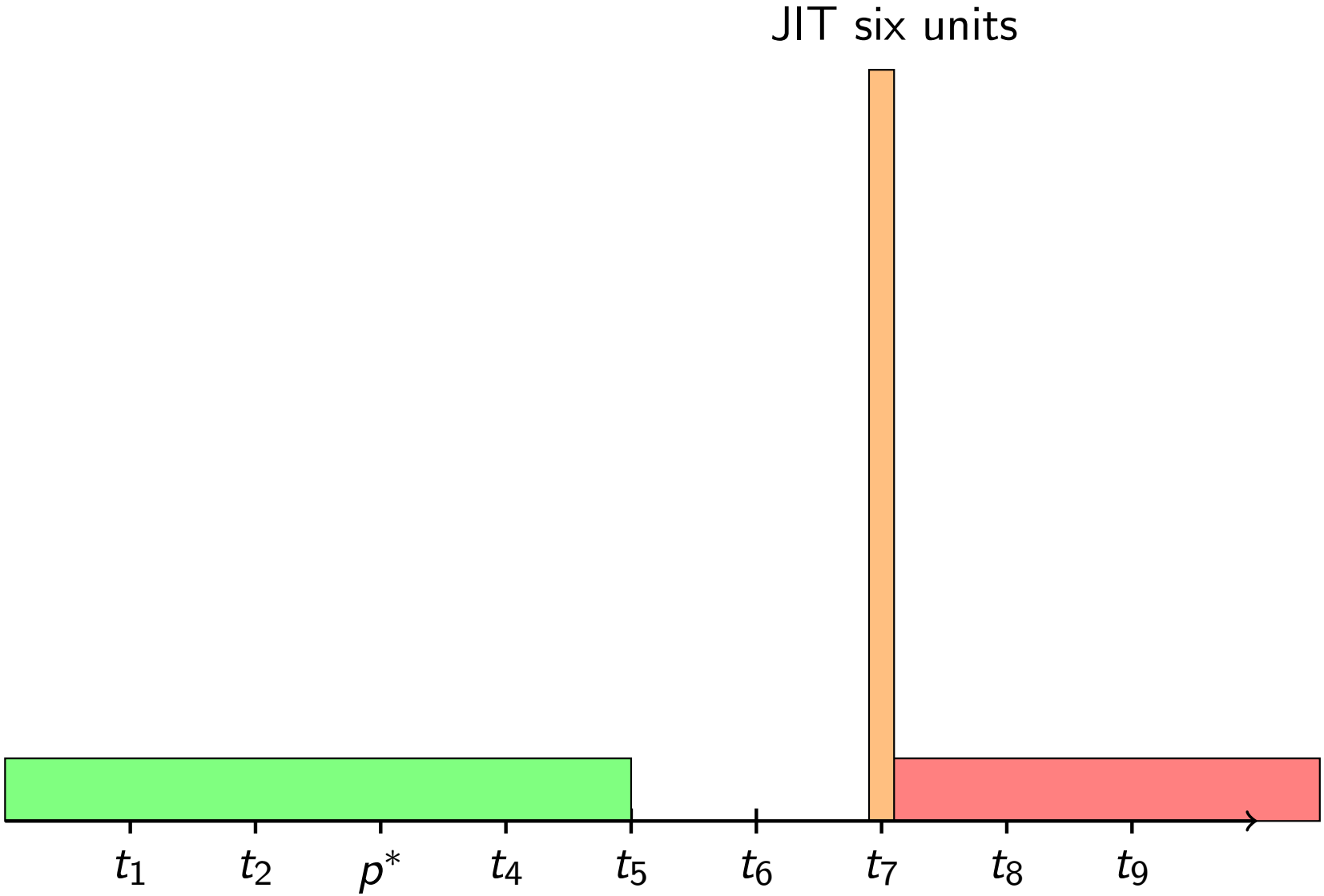
JIT + Sandwich (18.5f)

1. Frontrun three units



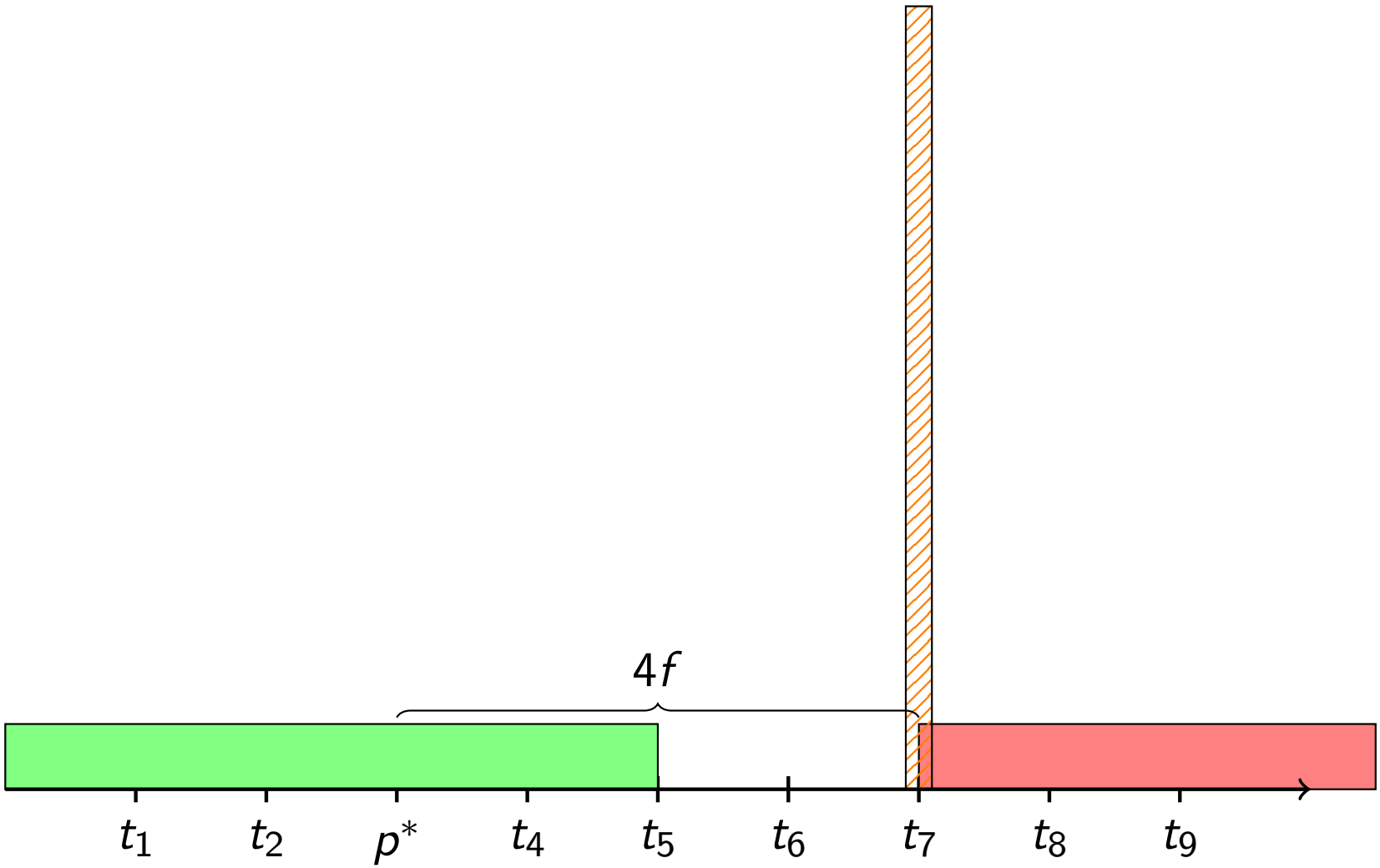
JIT + Sandwich (18.5f)

2. JIT



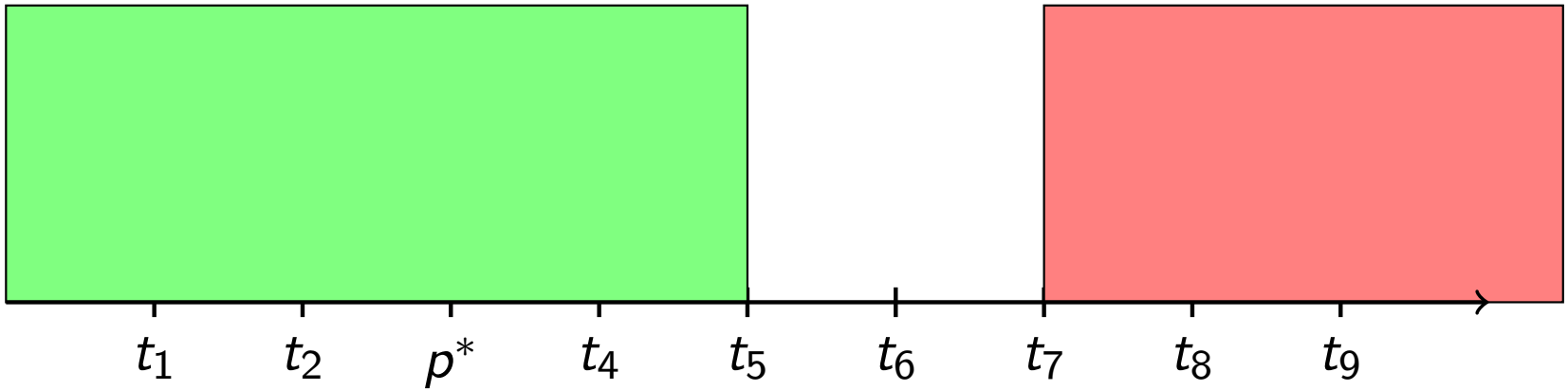
JIT + Sandwich ($18.5f$)

3. Execute

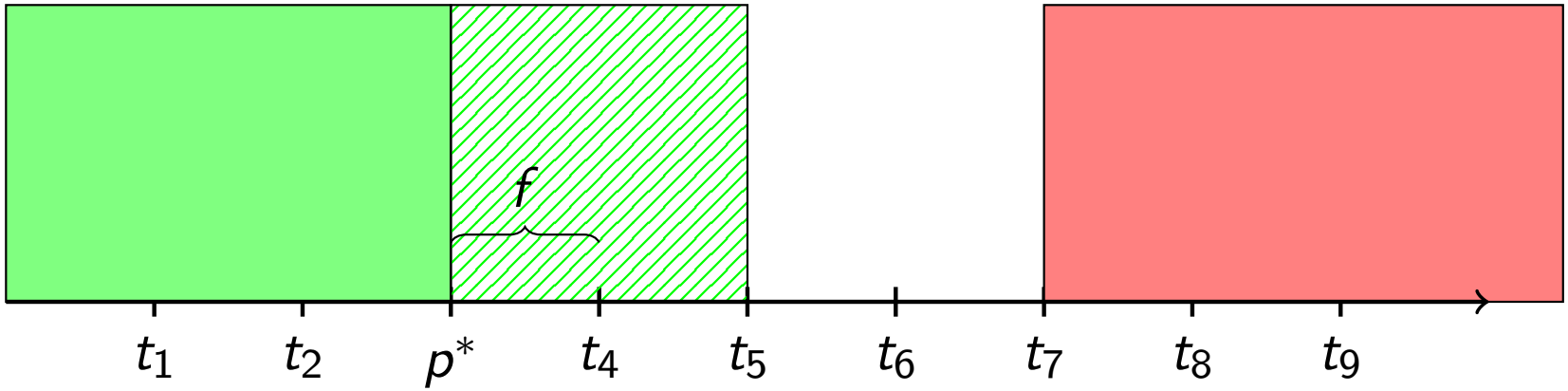


JIT + Sandwich (18.5f)

4. Remove JIT



5. Backrun two units



Results

	Swapper	LPs	Arb bot
Backrun	$-24f$	$11.5f$	$12.5f$
JIT + Backrun	$-20f$	$5.5f$	$14.5f$
Arb bot	$-24f$	$5.5f$	$18.5f$

Table: Payoffs for each of the participants.

Stopping Frontrunning Attacks

- Set tighter slippage tolerance (possibly negative)
- Break up the order into multiple orders
- Change JIT logic (speculation that this is Univ4)
- Order Flow Auctions

Strategy 1: Set Tighter Slippage Tolerance

- Slippage tolerance is the worst price you can get no matter what attacks happen
- You might even want a negative slippage tolerance!
- Tighter slippage tolerance = more failed transactions

Strategy 2: Break Up the Order

- Breaking up the order into multiple smaller orders increases the gas costs of frontrunning
- Spreading the orders over time allows you to take advantage of backrun liquidity
- This approach may increase gas costs for the trader but it doesn't have to (TWAMM)

Strategy 3: Change JIT Logic

- Would like to force JIT liquidity to be close to the current price so that you can't JIT + sandwich
- Would like this to look similar to RFQ system
- Can always get rid of JIT (Crocswap) but JIT might be helpful in some cases
- Maybe this is what they are cooking at Uniswap??? (rumor)

Strategy 4: OFA

- Can auction off the right to do whatever you want to the transactions (auction proceeds can compensate for these attacks)
- Auctioning off the exclusive right to execute an order gives whoever wins the auction more leverage when bidding in the PBS auction. Therefore, the proposer's rents are reduced.
- Auctions are difficult in decentralized environments due to collusion, Sybil attacks, and manipulation. See *Censorship-Resistance in On Chain Auctions*

Bibliography I

Alex Nezlobin

What is the optimal way to extract MEV from a large uninformed swap?.

Twitter Post.

Mallesh Pai, Max Resnick, Elijah Fox

Censorship Resistance in On Chain Auctions.

Preprint.

CrocSwap

CrocSwap Whitepaper.

whitepaper.