CS 251: Scaling I
Payment & State Channels

Instructor: Ben Fisch

Blockchain scalability

- Two types of scaling problems
  - ❖ Transaction throughput (txs/sec)
  - ❖ Blockchain size (state storage required to validate txs)

Transaction throughput

Two possible bottlenecks

- Consensus: fixed rate of blocks/sec
  - **Solutions:** increase block size (tx/block), block DAGs, faster consensus
- Verification time (both rate & latency)
  - **Solutions:** “off-chain” txs (payment/state channels), sharding, verifiable computation (SNARKs)

Blockchain size

How to reduce state storage of validators/miners?

- “off-chain” txs (reduce transactions stored in state)
- “State commitments” using authenticated data structures, like Merkle trees and other Accumulators

See references:
Utreexo – by T. Dryja
Batching Techniques for Accumulators w/ Applications to Stateless Blockchains
https://eprint.iacr.org/2018/1188.pdf by D. Boneh, B. Bünz, B. Fisch

Focus of next two lectures

- Payment & state-channels (off-chain txs)
- Sharding (distributing the verification work)

Payment channel

- Concept: use blockchain for net settlement
  - Alice buys coffee from Bob every day, only wants to settle on blockchain once/month, Bob doesn't want to take any risk
- Strawman: Deposit X coins in 2-of-2 multisig escrow E

  Day 1: A signs Tx1 -- "E pays 1 coin to B, X-1 coins to A"

  Day 2: A signs Tx2 – "E pays 2 coins to B, X-2 coins to A"

  End of month: B cosigns last transaction, e.g. from Day 30

Problem: If B doesn't co-sign, A loses all X coins

Payment channel

- If B doesn't co-sign, A loses all X coins
  - **Solution:** Timelocks!
  - B signs refund Tx3 – "E pays X coins to A" with a lock-time of 30 days
  - A waits to receive B's signed Tx3 before signing the deposit of X coins to E
- Remaining problems:
  - Two-way channel (B also wants to pay A)?
  - Non-expiring channel?

Bidirectional

Strawman Bidirectional

- B signs TX1 – “E pays X to A” time-lock Day 30
- A submits Deposit transaction of X to E 2/2 multisig
- A signs TX2 – “E pays X-5 to A, 5 to B” time-lock Day 29
- B signs TX3 – “E pays X-1 to A, 1 to B” time-lock Day 28
- …

Problem: Channel still expires in 30 days from Deposit

Non-expiring channel

- **Relative time-lock**: output can be claimed t timesteps (i.e., blocks) from the time the TX is accepted to the blockchain
- **Hash lock**: Claiming output is pre-conditioned on providing the preimage of a cryptographic hash

Intuition: Both A and B hold TXs they can submit to settle the current split balance. Balance is updated by exchanging new TXs and "invalidating" old. Unilateral settlement is time-locked for one party, allows the other to challenge by providing hash-lock preimage. TXs invalidated by exchanging hash-lock preimages.

Non-expiring channel

Establish channel

- A creates funding transaction “Deposit Z coins in address E, spendable w/ 2-of-2 multisig A,B” **DOES NOT YET SIGN**
- A receives H(y), y known to B. A chooses x and sends H(x).
- **A signs** TX1: “E pays Z-1 coins to A, E pays 1 coin to EITHER (B time-lock 7 days) OR (A given preimage of H(y))
- **B signs** TX2: “E pays 1 coin to B, E pays Z-1 coins to EITHER (A time-lock 7 days) OR (B given preimage of H(x))
- Now A signs the Deposit Tx and **POSTS**. B could settle by signing and posting TX1. A could settle by signing and posting TX2.

Non-expiring channel

Update channel off-chain (A -> B)

- A sends new H(x').
- A signs TX3: "E pays Z-2 coins to A, E pays 2 coins to EITHER (B time-lock 7 days) OR (A given preimage of H(y))
- B signs TX4: "E pays 2 coins to B, E pays Z-2 coins to EITHER (A time-lock 7 days) or (B given preimage of H(x))
- A sends x for H(x)

Why secure? B can settle with TX3. A can settle with TX4. If A attempts to settle with TX2, B can use x to claim the Z-1 coins first.

Bidirectional non-expiring channel

Update channel off-chain (B -> A)

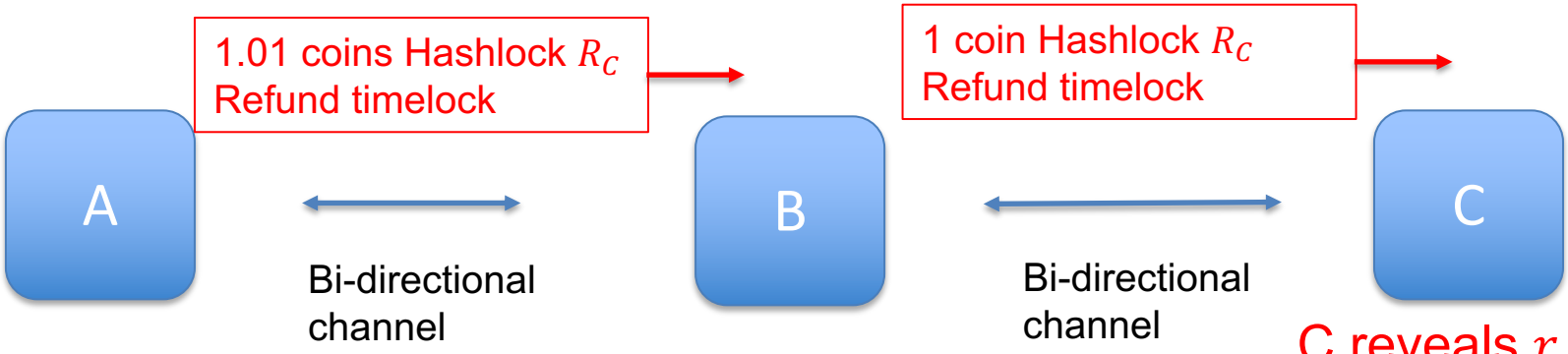
- B sends new H(y').
- A signs TX5: "E pays Z-1 coins to A, E pays 1 coins to EITHER (B time-lock 7 days) OR (A given preimage of H(y'))
- B signs TX6: "E pays 1 coins to B, E pays Z-1 coins to EITHER (A time-lock 7 days) or (B given preimage of H(x'))
- B sends y for H(y)

Why secure? B can settle with TX5. If B attempts to settle with TX3, A can use y to claim the 2 coins first.

Multi-hop channel (Lightning)

Idea: route payments through intermediary

C sends $R_c \leftarrow H(r)$ to A for secret r



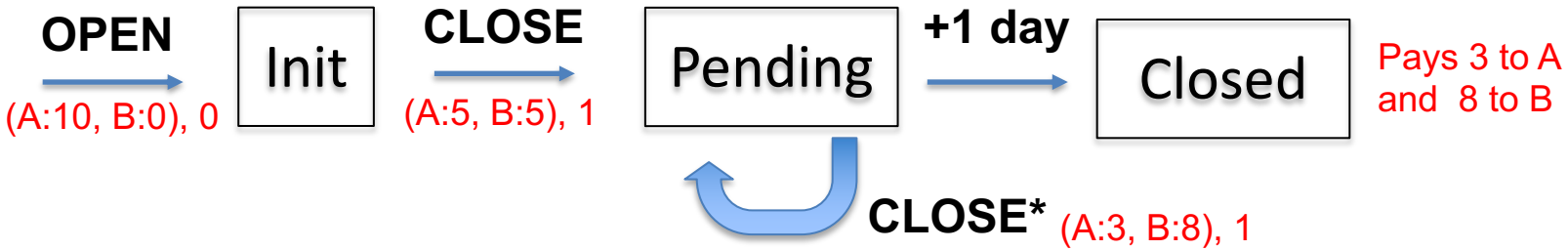
Multi-hop channel (Lightning)

HW exercise: Modify the non-expiring Bi-directional channel to achieve the hashlock/refund functionality required for Lightning…

State channel (Ethereum)

Much simpler to design payment channel with stateful contract!

- Contract state variables: Mode, BalSplit, Counter
- Initialize contract with balance split of (A:10, B:0)
- Valid CLOSE tx: "(A:x, B:y), Sequence #" signed by A & B, enters Pending state mode, +1 day wait
- In Pending, new CLOSE tx is accepted for a higher Sequence #, triggers +1 day wait



Blockchain size

How to reduce state storage of validators/miners?

- “off-chain” txs (reduce transactions stored in state)
- Merkle trees
  - Replace state (e.g. utxo-set, accounts) with Merkle root on blockchain
  - Txs include Merkle proofs for each UTXO/account inputs
  - Miners verify proofs, and update Merkle root
- RSA accumulators
  - Smaller network communication that w/ Merkle trees

End