CPSC 3640/5400: Decentralized Payments, Contracts, and Finance for Humans and AI

Lecture 6: Bitcoin's Scalability & Quantum-Resistant Future

Instructor: Prof. Fan Zhang
Yale University
Fall 2026

Bitcoins I received so far

- Amount (BTC) Count
- 0.00010000 19
- 0.00009775 1
- 0.00001000 4
- 0.00200000 1
- 0.00254861 1

Pushing Bitcoin forward

- How to make Bitcoin faster?

- How to migrate Bitcoin to a quantum-resistant future?

Payment channels

Bitcoin is slow

~3000 transactions per block
~600 seconds interval
⇒ 5 tx per sec (TPS)

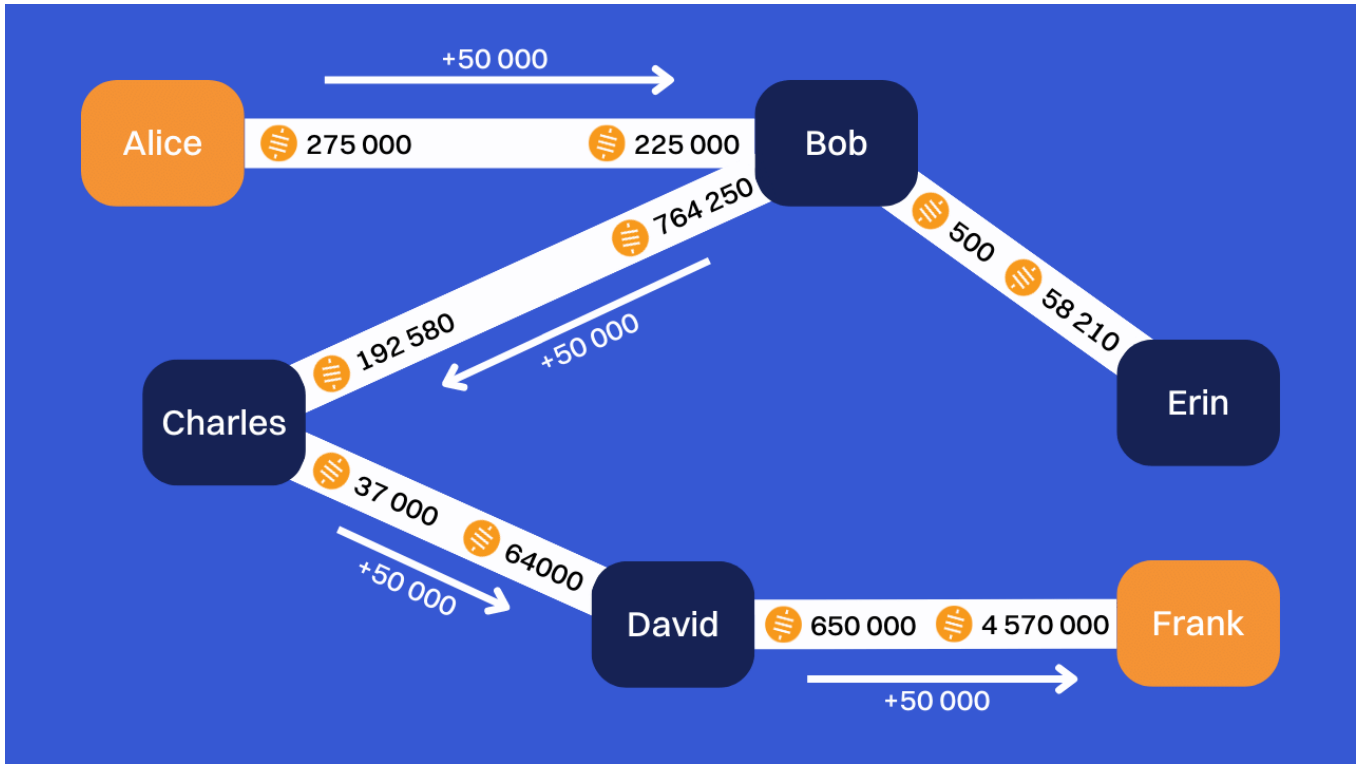
Also, 6 confirmations = 1 hour

- Can we scale up throughput & latency of Bitcoin?
- Unfortunately, scalability is bound by **security**
  - Reducing block interval or increasing block size will make Bitcoin less secure.
- Then, how to make Bitcoin faster?
- Answer: Payment channels

Bitcoin's Lightning Network

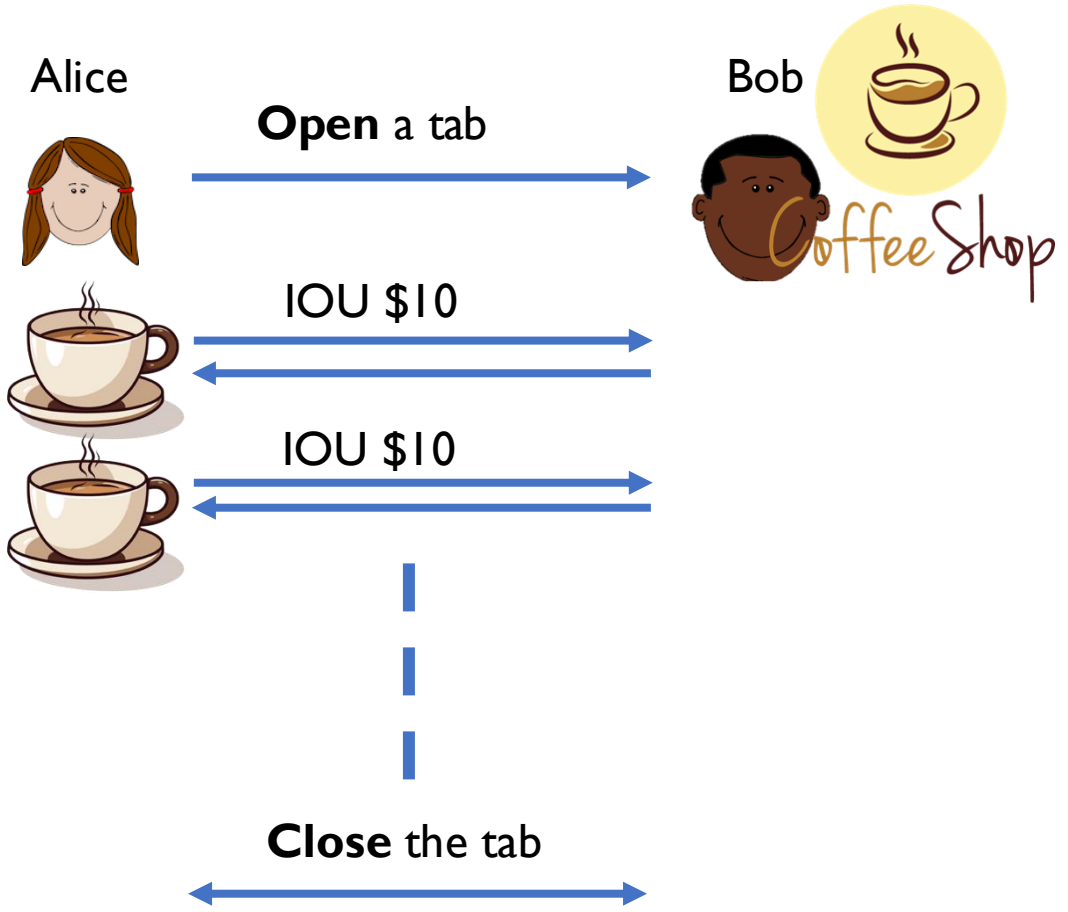
- We will study basic ideas behind Lightning Network
- Invented by researchers in 2016; launched by Lightning Labs in 2018.

- Used by **exchanges transfers** (Coinbase, Kraken), **retail payments** (e.g., Square, Cash App), and basically whenever Bitcoin is used.



Goals: fast & cheat off-chain payments

- You run a **coffee shop**, and Visa charges $1 fee for every transaction, how do you save?
- Have frequent client run a tab
- One Visa transaction fee for multiple *transactions*
- Payment channel is a similar idea.
  - Open, Payment, Close.
- Challenge: One or both parties might be malicious! (What can go wrong?)



Notation for UTXO

$TxId_3$	
INPUTS	OUTPUTS
$TxId_1:idx_1$ Alice's signature	**Charlie** 3 coins
$TxId_2:idx_2$ Bob's signature	**David** 5 coins

Think P2PKH

Attempt 1

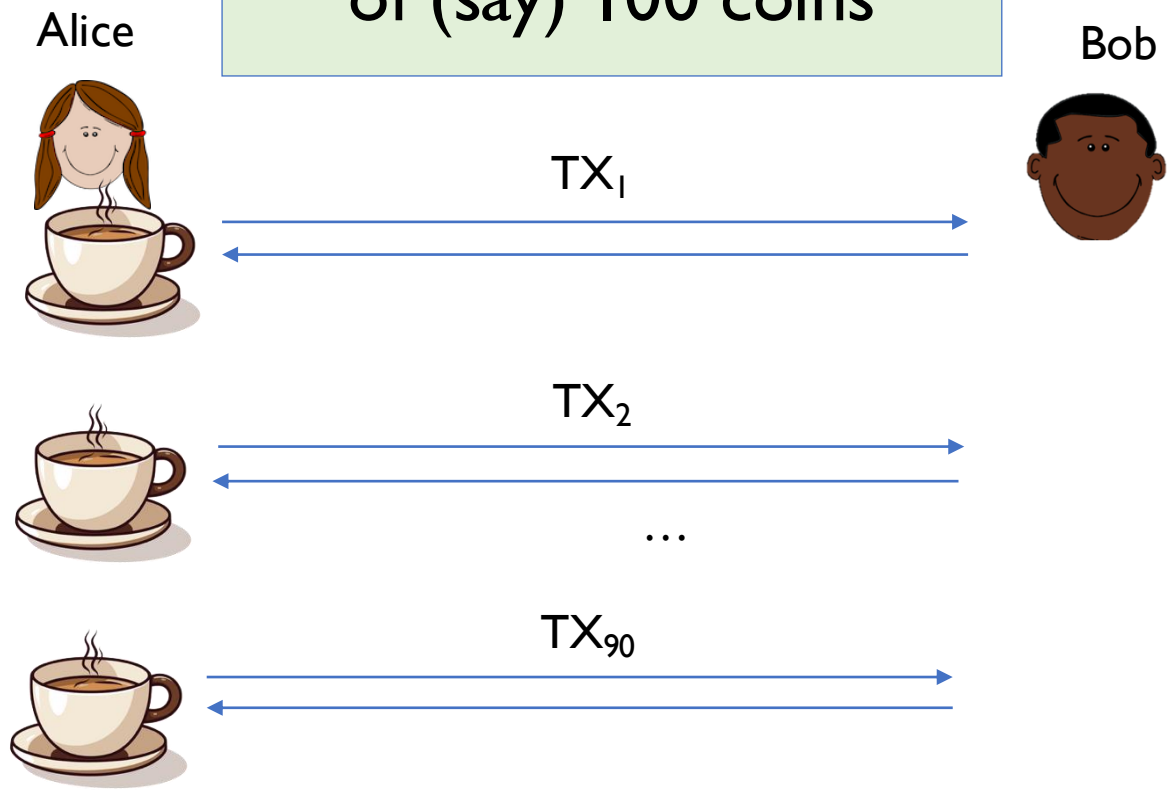
To **open** a channel, Alice creates an UTXO of (say) 100 coins

- Payment: Alice sends signed transactions to Bob (**not** to Bitcoin miners)

TX_i =

INPUT	OUTPUTS
UTXO A *Alice's Sig*	Alice: $(100 - i)$ coins
	Bob: i coins

Assuming each coffee costs 1 coin.



Can you see the problem?

Close: At at point, either party close the tab by sending the last TX to Bitcoin.

multisigs

- Example: Payment

$TxId_4$	
INPUTS	OUTPUTS
$TxId_3$:0 Charlie's signature	**Alice & Bob** 100 coins

2 out of 2 multisig

- Example: Spend

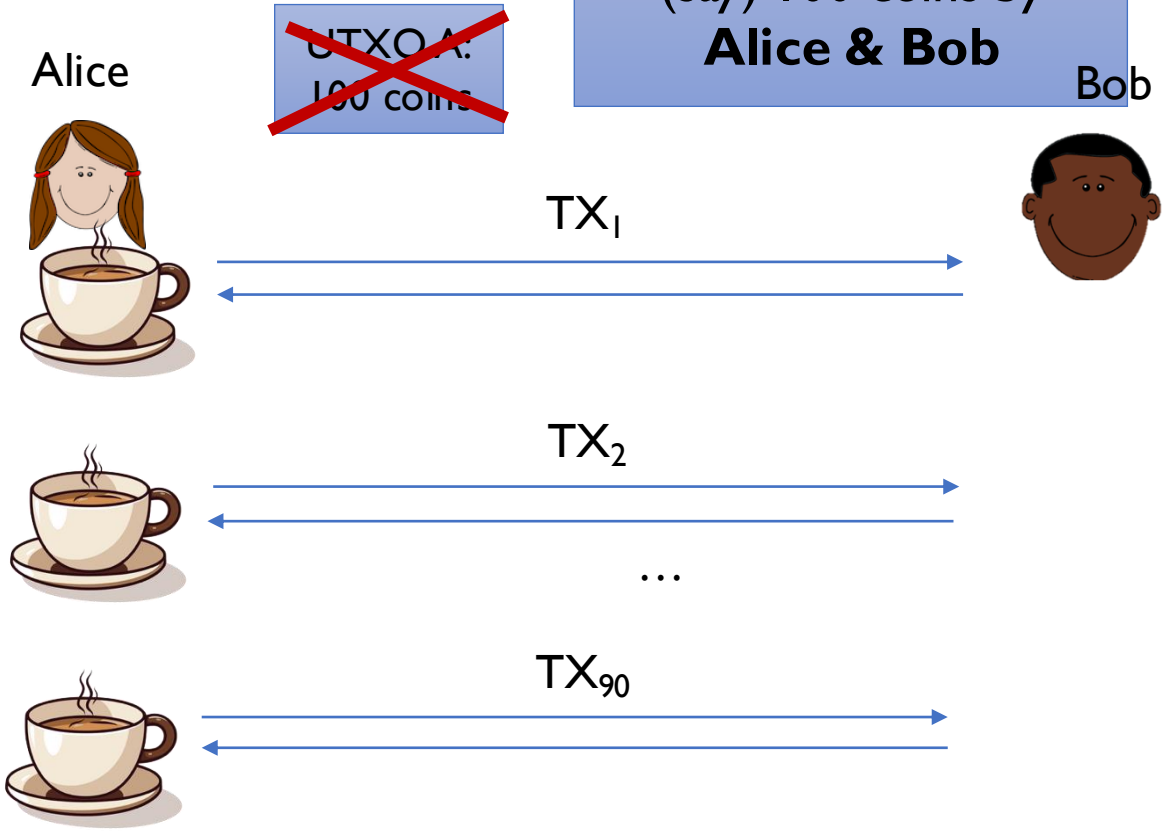
$TxId_5$	
INPUTS	OUTPUTS
$TxId_4$:0 **Alice's signature** **Bob's signature**	…

Attempt 2

- Payment: Alice signs the transactions

TX_i =

INPUT	OUTPUTS
$UTXO_{AB}$	Alice: $(100 - i)$ coins
Alice sig	Bob: i coins



Can you see the problem?

Bob can block Alice's refund!

Close: attach Bob's signature and send TX_{90} to Bitcoin.

Ingredient 3: Timelock

- Transactions have a "lock time" field
- Transaction is only valid after the lock time (block height) has passed

$Txld_4$, **LOCKTIME=Sep 28th,2026**	
INPUTS	OUTPUTS
$Txld_3$:0 Charlie's signature	**Alice & Bob** 8 coins

How to use Timelock to solve the problem?

RefundTx

FundTx	
INPUTS	OUTPUTs
Some Alice's UTXO	Alice & Bob 100 coins

Refund Tx **LOCKTIME: Sep 28th,2026**	
INPUTS	OUTPUTs
FundTx:0 *Bob's sig*	Alice 100 coins

- **Open:** *Only after* receiving RefundTx, does Alice open the channel by sending FundTx to Bitcoin.
- **Payment** & **Close** the same as before

- fully refunds Alice after a delay
- Has Bob's signature
- Alice keeps this in her pocket

This is a simple one-way channel

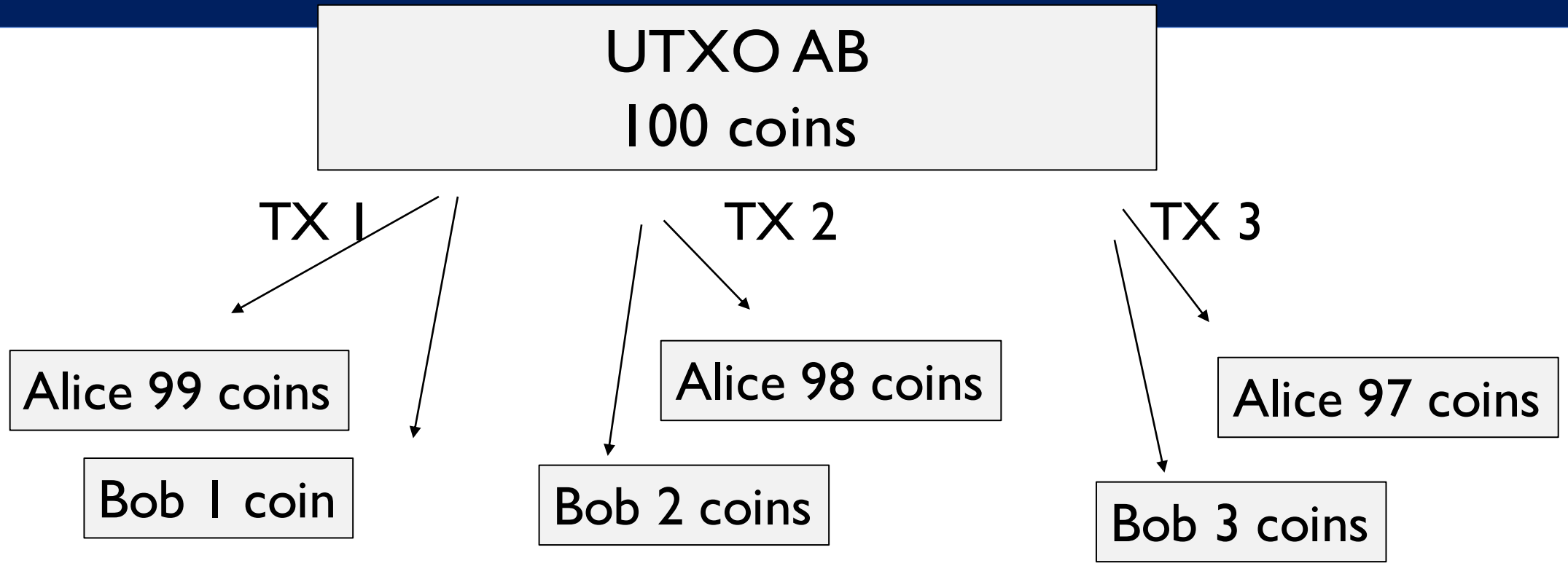
Pretty amazing! Except for opening & closing of channels:

- Payments don't use Bitcoin at al
- Instant confirmation
- No fees

Limitations:

1. Fixed life span: set by the refund tx
2. Payment is one-way: Bob cannot pay Alice back.

A closer look at limitations #2



- Suppose now Bob wants to pay Alice 1 coin
- Bob needs to *erase* TX_3
- No way for Alice to be sure that Bob indeed does.

Idea: Leverage economic incentives

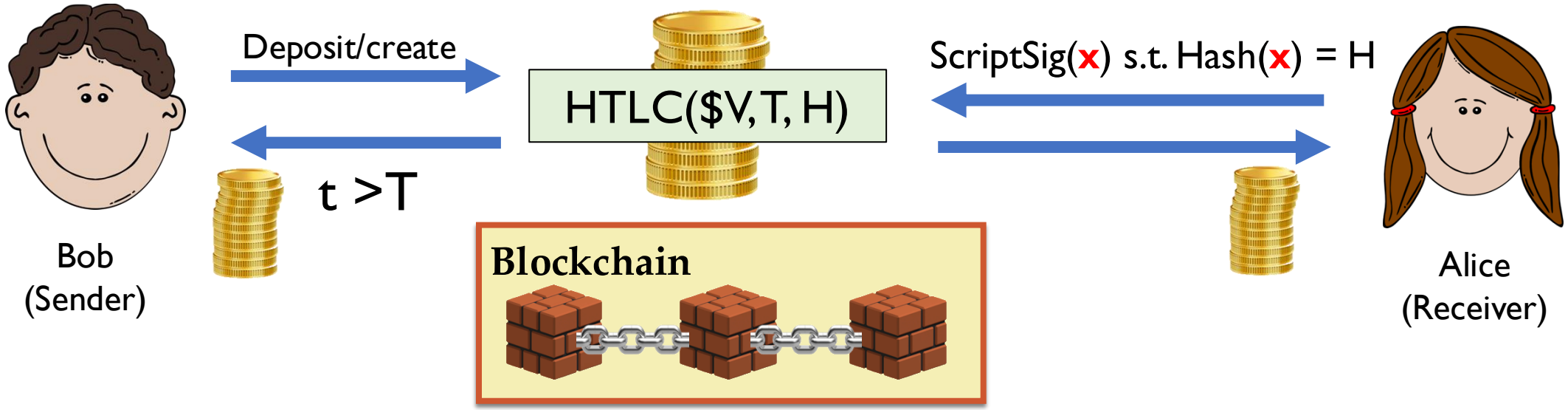
Economic penalty & incentives are a key enabler of many DeFi protocols, including payment channels.

Idea here:

- When closing the channel, a challenge period opens
- During which, anyone can present *evidence* if the channel is closed with a *stale* TX (a TX that *should have been erased*)
- Offender gets punished (e.g., if Bob broadcasts TX_3, then Alice can spend all of his coins)

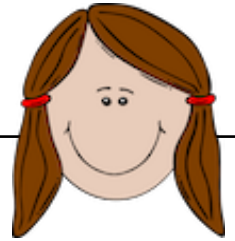
New Ingredient: Hash Time Locked Contract (HTLC)

- HTLC(v, T, H): a fancier version of retractable coins
- Before timeout T, receiver can spend if *she produces a preimage of H*
- After timeout T, sender can retract



Implementing *challenge period* w/ HTLC

- At any point, Alice and Bob hold symmetric transactions (e.g., channel at 97/3)



X=SHA256(x)

TX held by Alice	
INPUTS	OUTPUTs
FundTx *Alice sig* *Bob sig*	(Alice in 3 days) OR (HashLock(X) & Bob) 97 coins
	Bob 3 coins

Y=SHA256(y)

TX held by Bob	
INPUTS	OUTPUTs
FundTx	Alice 97 coins
Alice sig *Bob sig*	(Bob in 3 days) OR (HashLock(Y) & Alice) 3 coins

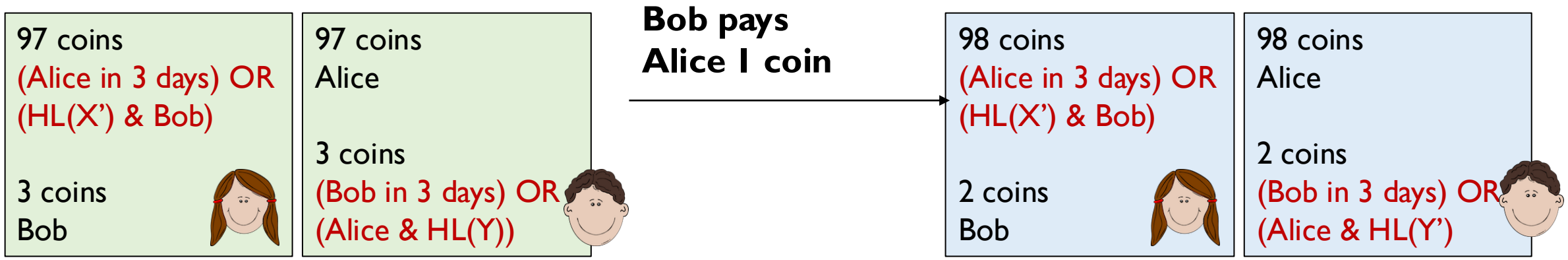
Implementing *challenge period* w/ HTLC

TX held by Bob	
INPUTS	OUTPUTs
FundTx	Alice 97 coins
Alice sig *Bob sig*	(Bob in 3 days) OR (HashLock(Y) & Alice) 3 coins

Bob closes the channel by sending this TX to Bitcoin network:

- Alice gets 97 coins immediately
- Bob's coins subject to *challenge period* of 3 days
- During which, if Alice reveals *preimage of Y*, she gets Bob's coins.
- If no one challenges, Bob gets his money back after 3 days.

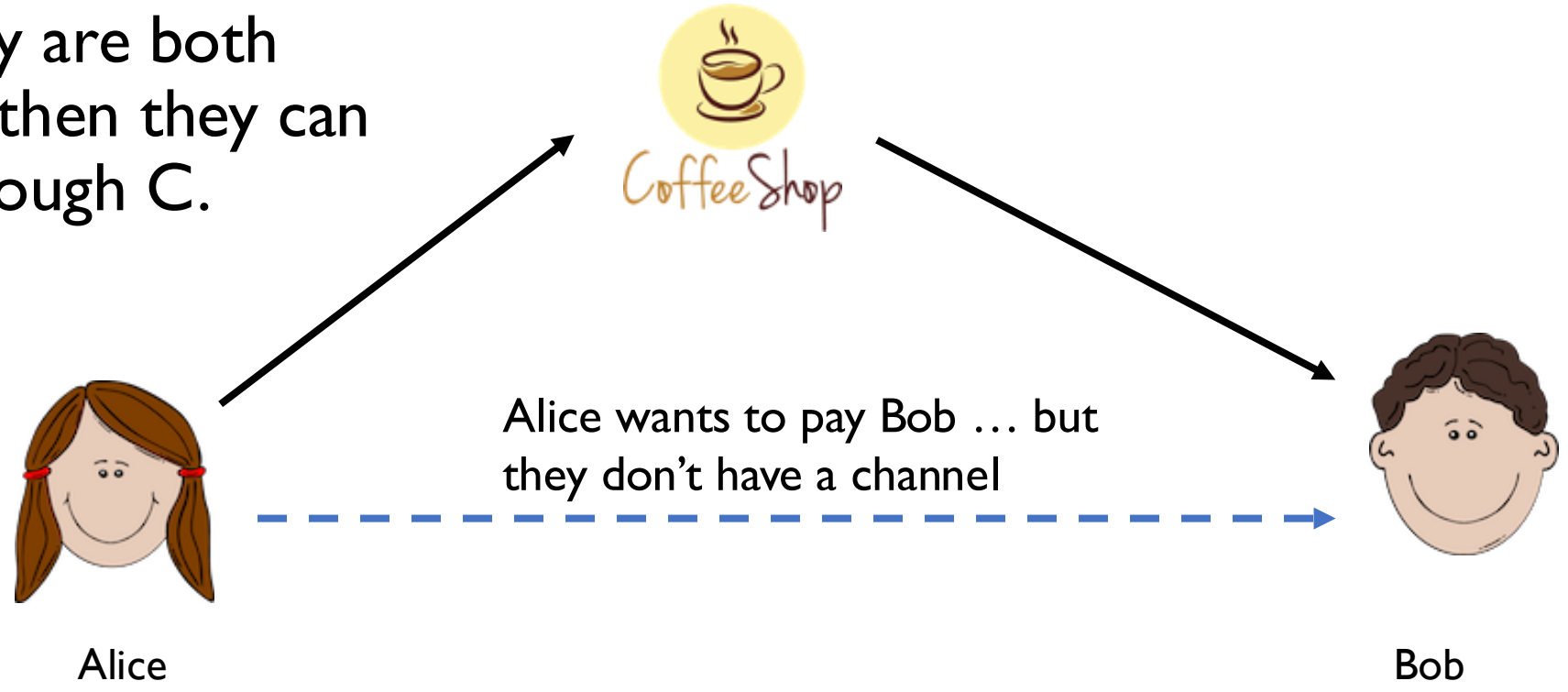
Now two-way payments are possible



- Along with new transactions, Bob reveals y to Alice
- If Bob later broadcasts revoked transaction, Alice can panelize Bob with y
- It's a delicate balance: Both parties need to actively monitor blockchain & be ready to challenge.

From this to Lightning channels

- Lightning supports **networks**
- If Alice & Bob don't have channels, but they are both connected to C, then they can route money through C.



We will stop here, the protocol gets very complicated very quickly…

Payment channels

- An off-chain payment protocol for faster & cheap Bitcoin payments.
- The protocol is incredibly complex (partly due to Bitcoin script limitations) but it's widely used.
- It uses **economic penalty** to enforce desirable actions by counterparties.
- This is one of the most powerful tools in DeFi protocols, e.g.,
  - **proof of stake** (Ethereum) relies on penalty to prevent forks
  - Penalty and rewards are used to encourage truth report in **Polymarket**

21)

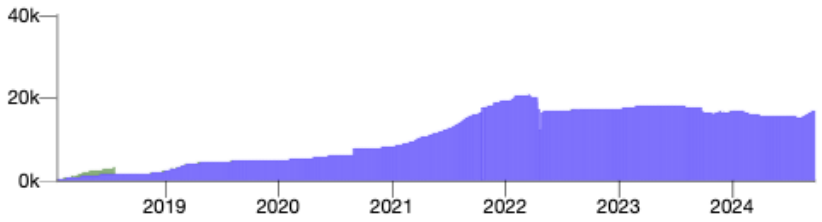
BITCOIN VISUALS

Buy Bitcoin Charts Resources

Want to sponsor BitcoinVisuals.com? Get in touch here.

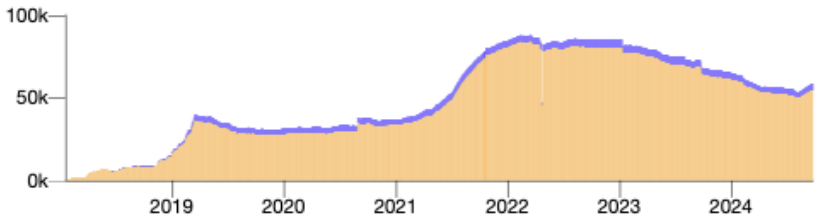
Nodes

Number of nodes with and without channels.



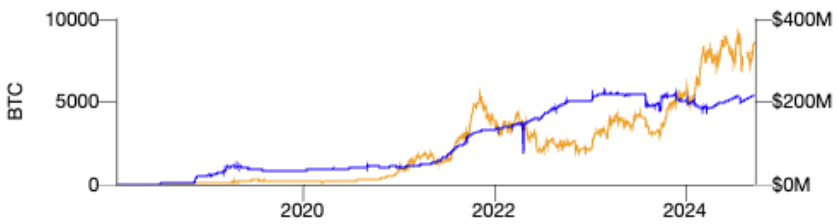
Channels

Unique = channels connecting nodes directly for the first time. Duplicate = channels between nodes that are already connected.



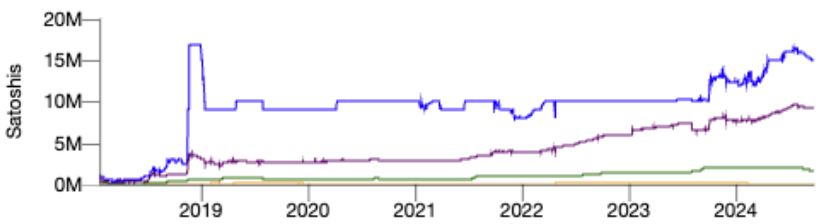
Network Capacity

Cumulative bitcoin capacity across all channels.



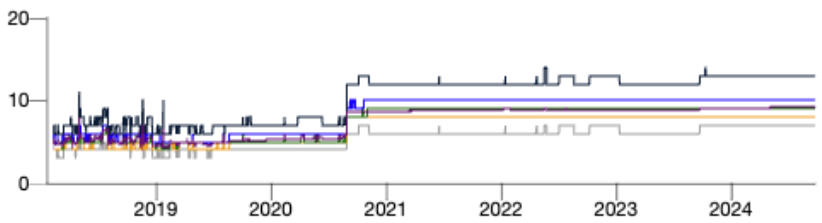
Capacity Per Channel

Channel capacity statistics.



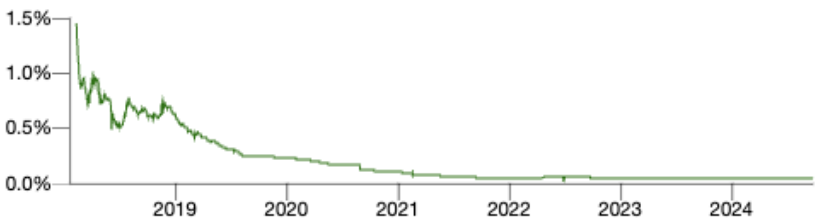
Distance Measures

The maximum number of hops required to reach another node (among shortest paths).



Completeness Measures

Density is a ratio of actual / potential channels.



Clustering Measures

Transitivity is the ratio of potential triangles present. A value of 1 means every path of length 2 loops back into a triangle. Clustering coefficient is the ratio of interconnections between a

Connectivity Measures

A cut channel (aka cut edge, or bridge) is a channel between two nodes that connects different components of the network. This channel's removal would prevent other nodes from

https://bitcoinvisuals.com/lightning

Questions…question…question…

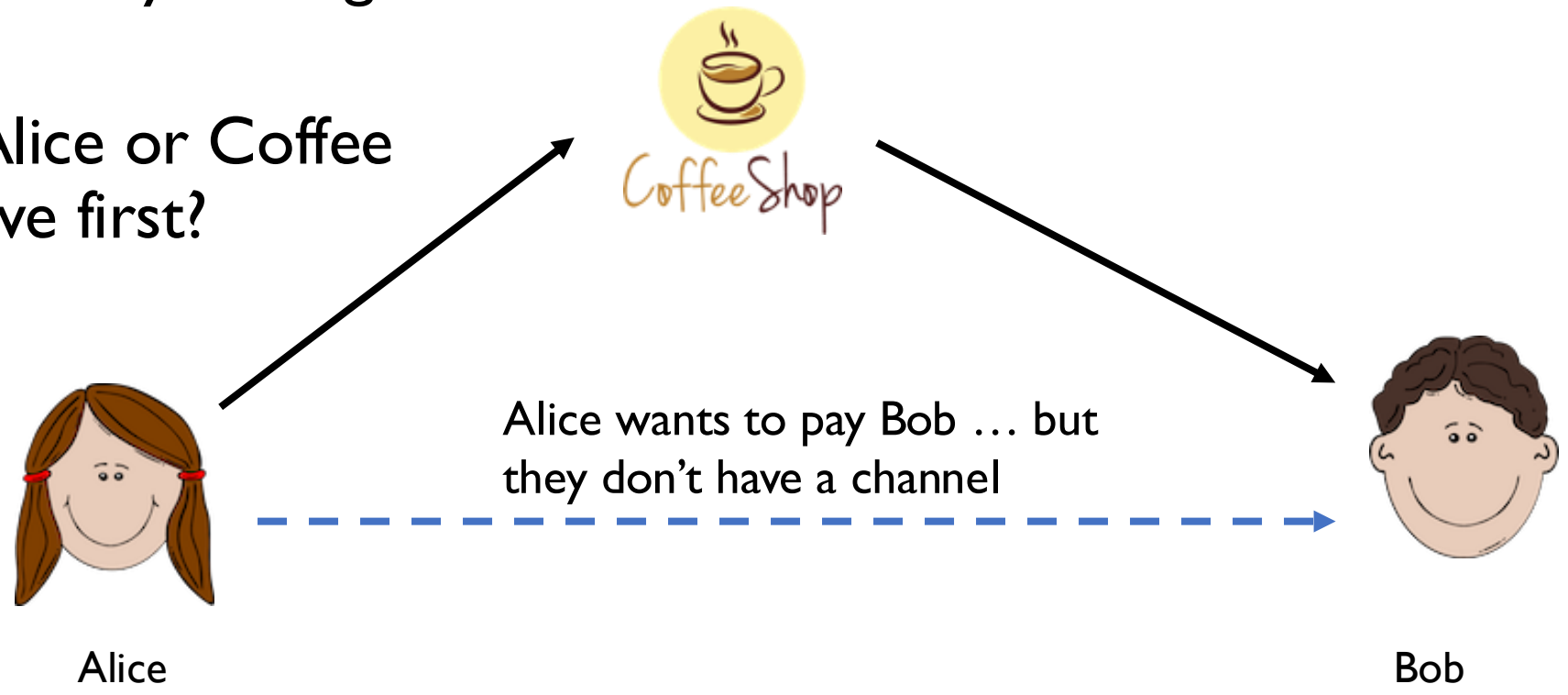
- How will Alice and Bob discover a path?
  - Note: Need to ensure monetary capacity along path
  - LN-proponent answer: Like Border-Gateway Protocol for Internet routing (which has its own problems…)
- Will system devolve into (quasi-centralized) hub-and-spoke?
  - LN-proponent answer: Perhaps, but hubs won't control system.
- Isn't the whole thing incredibly complicated?
  - LN-proponent answer: Maybe, but the system is evolving.

Optional: From channels to networks

Slides after this point is optional.

Multiple party channels

- Opening channels is expensive
- Can Alice route money through Coffee Shop?
- Problem: Who (Alice or Coffee shop) should move first?



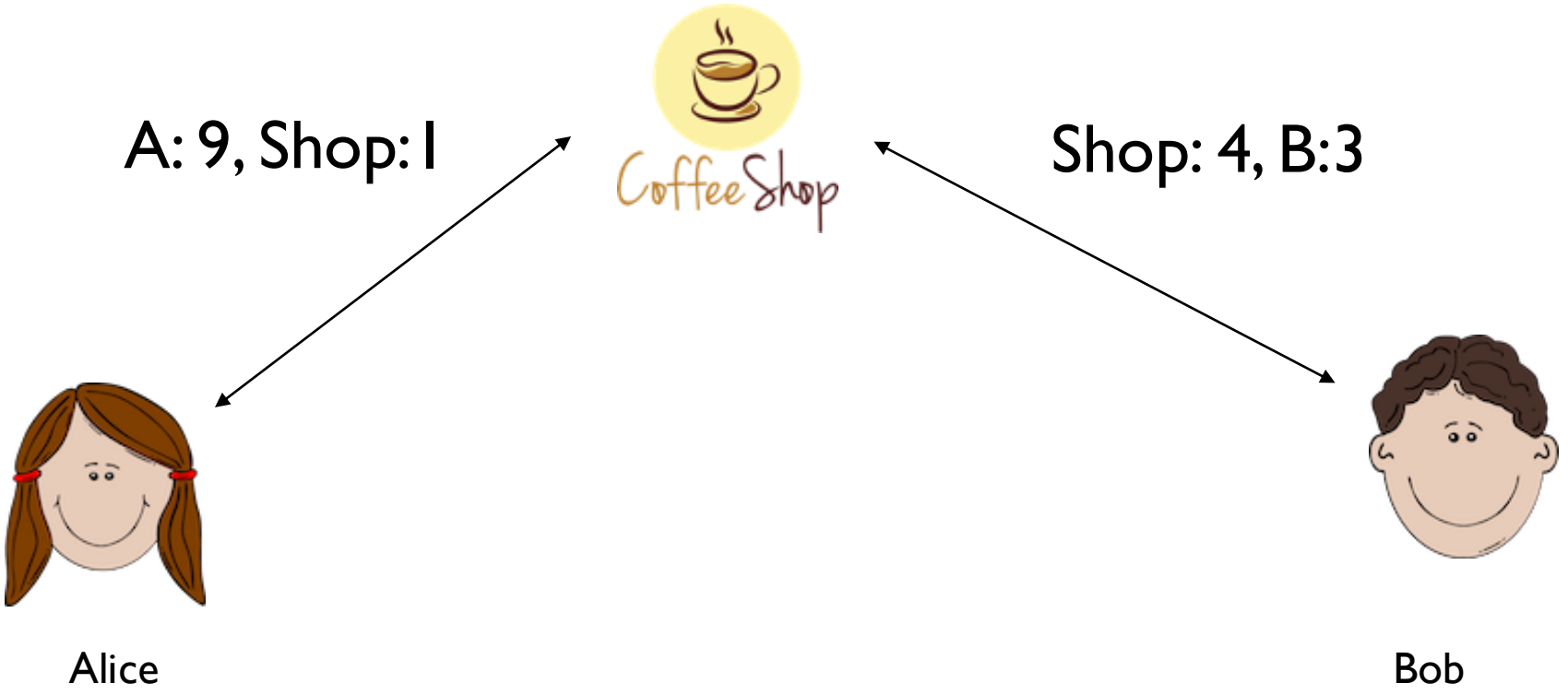
HTLC comes to the rescue again

- Alice makes a conditional payment to the coffee shop using HTLC
- Use absolute timeout (reasons shown later)

TxId	
INPUTS	OUTPUTS
Some UTXO	(Alice after 1pm Feb 19) OR (Shop & HL(X)) 8 coins

Routing through Mike

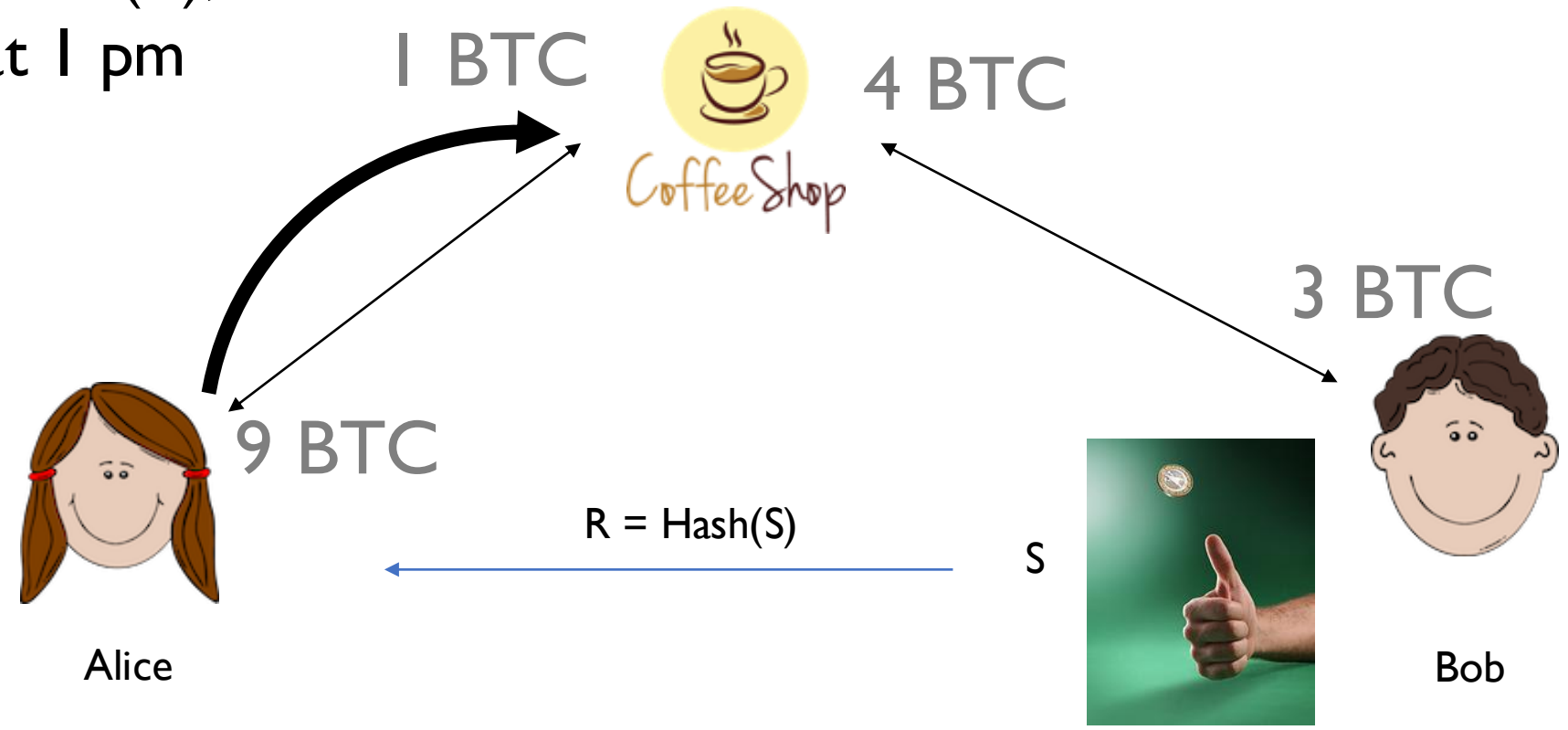
- Suppose Alice wants to send to Bob 1 coin via intermediary



Multiple party channels

Alice pays Shop 1 BTC with HTLC

- Shop & HL(R), or
- Alice at 1 pm



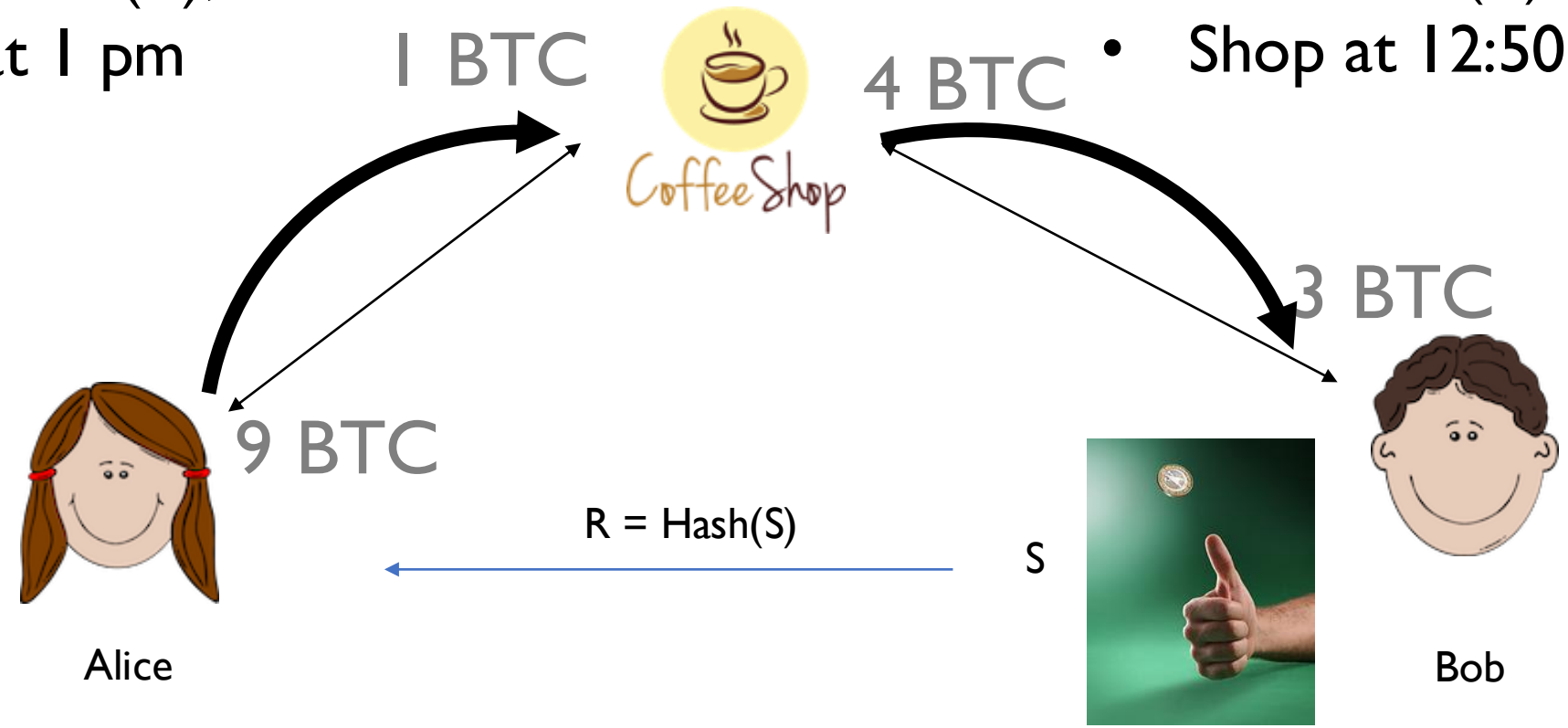
Multiple party channels

Alice pays Shop 1 BTC with HTLC

- Shop & HL(R), or
- Alice at 1 pm

Shop pays Bob 1 BTC with HTLC

- Bob & HL(R), or
- Shop at 12:50 pm



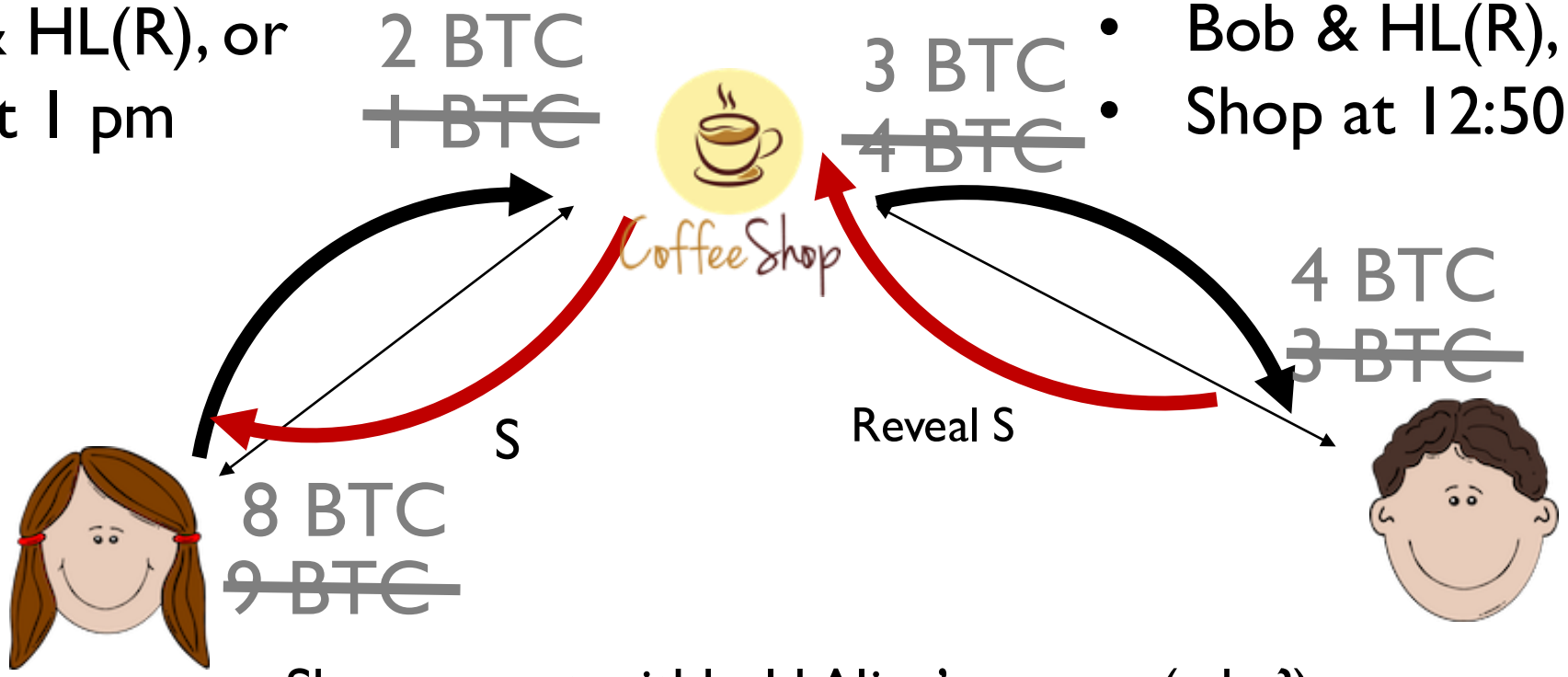
Multiple party channels

Alice pays Shop 1 BTC with HTLC

- Shop & HL(R), or
- Alice at 1 pm

Shop pays Bob 1 BTC with HTLC

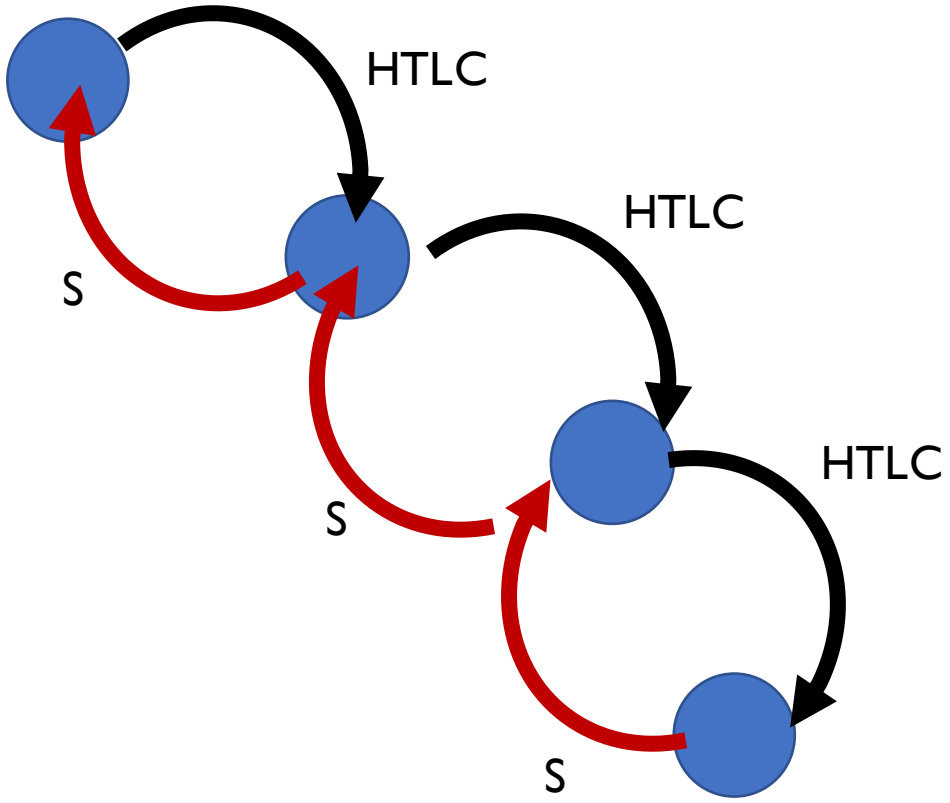
- Bob & HL(R), or
- Shop at 12:50 pm



Shop cannot withhold Alice's money (why?)

Shop won't lose money (why?)

Lightning network



"Those initial streamers are looking for a path of least resistance. When they find it; that stroke comes up from the ground." --- L.B. on https://earthscience.stackexchange.com/