Bitcoin Scalability

static const unsigned int MAX_BLOCK_SIZE = 1000000;

uhoh...

- Each node creates $\sim 1$ transaction ($1 \times n$)
- Each node stores all transactions ($n \times n$)
- Total transactions stored = $O(n^2)$
Different Solutions

- The SQL Database Model
  - Very scalable, very fast
  - Off chain transactions implemented today with ChangeTip, Coinbase, others

- Altcoins
  - Many blockchains with inter-chain transfers

- Larger Blocks

- Payment Channels
  - Many payments between two pre-determined parties
SQL

- 100 users send their coins to 1 address
- The 1 node maintains balances in an SQL database -- User : Balance
- Users can transfer internally, deposit and withdraw
- Very fast, can support millions of transactions per second
SQL problems

- Likely to happen if no other actions taken
- Already very popular
- \( \lim_{t \to \infty} \): Good delivery model
Alts

- Sell your bitcoin, buy some NeatoCoin™
- Transact *fast* with NeatoCoin™ and it’s HydroFlex Negative BlockTimes™ (Block N+1 comes out before Block N. It’s non-causal!)
- When done transacting with NeatoCoin™, buy back your bitcoin.
Alt Problems

- The Altcoin Exchange is the same as the SQL server. (Atomic cross chain txs could work, but not being used)
- Does NeatoCoin™ really work? Is it going to fall apart?
- If it doesn’t work, you shouldn’t use it
- If it works… why not just stick with NeatoCoin™? It’s going to the moon.
- Not a good solution for Bitcoin, because it’s not Bitcoin.
Larger Blocks

- Computers are great. Moore’s Law works.
- Storage: 100MB block, always full, is 5TB /yr. 5 TB HD costs <1 BTC.
- If you were actually filling 100MB blocks every 10 min, a 5 TB HD would be way less than 1 BTC.
- CPU, RAM: Have you tried v0.10? So fast!!
- v0.11, pruning? Blockchain down to 1GB!!!
Larger Blocks - Big O

- $n^2$ is not that bad! it’s polynomial! If it were $2^n$, then it wouldn’t scale
- While the total network cost is $O(n^2)$, for each users it’s $O(n)$
- If the value of the network obeys Metcalf’s law, then the value is $O(n^2)$, and value per user is $O(n)$
- Cost $\approx$ value, no problem!
Larger Blocks - problems?

- Miners are centralized anyways
- 20 MB still only gets you ~80 tx/sec
- Would need *much* larger blocks for billions of people
- What about the IoT? What if your fridge pays your drone to go pick up some eggs?
- Larger blocks can help. Necessary but not sufficient.
Payment channels

- Transactions can be delayed and aggregated before being cleared on the blockchain.
- Confirmed transactions are now only needed to open and close channels.
Payment Channels - Free lunch?

- Opt-in
- Many transactions
- Instant confirmation
- How to scale to many users
1:1 Payment Channels

First Alice gets a refund signed by Bob, then sends to the multisig address.

Even if Bob disappears, she can get the coins back tomorrow.
Alice signs 0.1 to Bob, and gives Bob the signature.
Bob doesn’t sign or broadcast.
The signature itself is the payment.
1:1 Payment Channels

Alice and Bob Multisig Channel Address

Signed by Bob
Valid Tomorrow

Signed by Alice

Alice signs 0.2 to Bob, overwriting the previous spend.
Alice can increment many times without transaction fees.
3 party - optimistic (iterative)

Alice wants to pay Carol. They both have a channel open with Bob.
3 party - optimistic (iterative)

Alice

0.01 BTC to Bob

Bob

Carol
3 party - optimistic (iterative)

0.01 BTC to Bob

0.01 BTC to Carol
3 party - optimistic (iterative)

- Alice sends 0.02 BTC to Bob
- Bob sends 0.01 BTC to Carol
3 party - optimistic (iterative)

Alice

Bob

0.02 BTC to Bob

Bob

0.02 BTC to Carol

Carol
3 party - optimistic (iterative)

Alice

0.03 BTC to Bob

Bob

0.02 BTC to Carol

Carol
3 party - optimistic (iterative)

Alice

0.03 BTC to Bob

Bob

0.03 BTC to Carol

Carol
3 party - Trust Issues

Problem: Bob can simply keep the 0.01 BTC
Problem: Carol can claim she never got the coins!

"0.01 BTC to... I think I'll keep this."
3+ party - trustless

Alice wants to pay Dave without opening a new channel
Dave makes a random number $R$ and hashes it to $H$.
Dave gives Alice $H$
3+ party - trustless

Alice pays Bob, but only if he knows R, the pre-image of H
3+ party - trustless

Bob pays Carol, but only if she knows R, the pre-image of H.
3+ party - trustless

Carol pays Dave, but only if he knows R... and he does!
When Dave receives the payment, he must reveal R. Revealing R allows Carol and Bob to receive their payments.
3+ party - trustless

Lots of payments to anyone within the networks, without the need to make new channels.

Bob — Carol

Alice — Bob — Dave

As long as there’s a path, payments can be routed.

… kind of like the Internet!
Using Time for Atomicity

- Historical norm for using time as the primary method for atomicity in financial markets with multiple parties
  - T+3 in equities
  - Correspondent Banking
  - “Overnight” anything
Systemic Coin Theft

- Isolated attacks don’t work
  - They’ll lose all their money, too!
- Systemic attacks unlikely but disastrous
  - Millions of channels with lots of coins in channels
  - Simultaneously broadcast previous channel states where the attacker gets more coins
  - Pay very high miners fees
    - Child pays for parent
Mitigating Systemic Risks

- Blocks should be mostly full, a fee market is good!
- Possible solution(s):
  - Soft-cap block size
  - Some sidechain thing (put soft-cap in this)
- Blocks full most of the time, credible threat that the block size can be increased quickly.
Economic Implications

- Coins locked up in channels
  - Reserved in case counterparty receives funds
  - Immediately available to spend, but some time-value of money allocated in relationship
  - Intermediary nodes have funds locked up

- Reduction in money supply may increase the price per bitcoin to accommodate necessary amount of economic transactions
Fee Market

- Fee market will exist with Lightning paid to liquidity providers
  - Separate from on-blockchain fees
- Fees can be positive or negative
  - Maybe a lot of coins are moving across a channel, if you have a relationship between both, you can keep that channel open and receive some fee
(Speculative) Economic Implications

- Economic incentives are aligned with keeping channel paths open and available
  - “Network Liquidity”
- Ratio of funds locked up to funds available to one’s channel counterparty
  - “Channel Liquidity”
Providing Liquidity

Erin, an end user with a smartphone, helps with liquidity (and earns coins) on frequently used channels.
(Speculative) Economic Implications

- Channel liquidity is what is really being locked up
- Fees will also exist if you want high amount of funds available in the channel
  - Fees will be very very cheap
  - Long-term demand liquidity reflects in higher exchange rates to accommodate
Applications

- Micropayments
  - Pay for publishing. Newspapers get paid per view, donation for per song played on your MP3 player, etc.
- Pay for Bandwidth (Cell phones)
- Instant Payments: Paying for coffee actually works
  - Arbitrage
What Lightning Network Needs

- Malleability fix which allows spends from unconfirmed transactions
- Relative Maturity
  - (a.k.a. OP_RELATIVECHECKLOCKTIMEEVERIFY)
- Accounting for bursts in block sizes
- Coding the wallet
  - Network communication layer
  - Will take some time
Bitcoin Scalability Solutions

Questions?

Thanks for listening!