Building with Lightning

DEV DECAL APR 2018

Max Fang

BLOCKCHAIN
AT BERKELEY

ABOUT MAX

DEV DECAL APR 2018

- Developer Advocate at **Lightning Network!**
- **UC Berkeley** Student: CS + Econ
- Upcoming Adjunct Professor **Berkeley Law**
  - LL.M program
- President of **Blockchain at Berkeley** since 2015
  - (Previously Bitcoin Association of Berkeley)
- **ChangeTip** Jan 2015 - Jan 2016
- ~4 years cryptocurrency experience since founding GPU-based cryptocurrency mining startup Feb 2014
- Designed & founded the **Blockchain Fundamentals** course at UC Berkeley: world's only undergraduate survey course (~120 students)
  - (Previously Cryptocurrency Decal)
- **Research interests:** Bitcoin privacy and PoW game theory
- Misc teaching and consulting
  - **Executive Education**

Me and the ChangeTip team

BLOCKCHAIN
AT BERKELEY

LIGHTNING

OUTLINE

DEV DECAL APR 2018

1 ▶ **Limitations of Bitcoin**

2 ▶ **Intro to Lightning**

3 ▶ **Lightning Use Cases**

4 ▶ **Lightning Development**

5 ▶ **The State of Lightning**

1

LIMITATIONS OF BITCOIN

LIMITATIONS OF BITCOIN

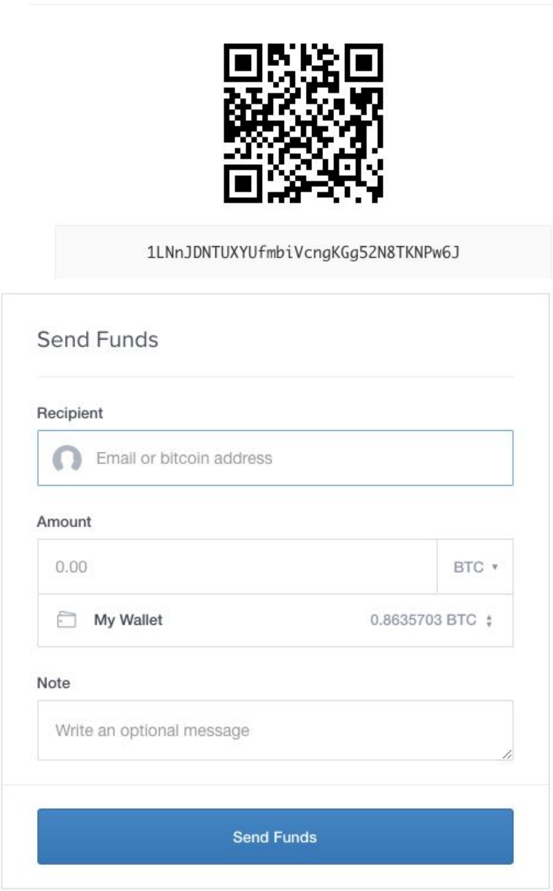
DEV DECAL APR 2018

Bitcoin exists as software. Transactions are conducted through wallet software that makes our lives easy.

When we click "Send Funds", what is going on under the hood?

The transaction is

- **broadcast** to the network,
- where other nodes **verify it,**
- **add it to the transaction history**,
- and eventually, the recipient **accepts** the tx



1LNnJDNTUXYUfmbiVcngKGg52N8TKNPw6J

Send Funds

Recipient

Email or bitcoin address

Amount

0.00 BTC ▾

My Wallet 0.8635703 BTC

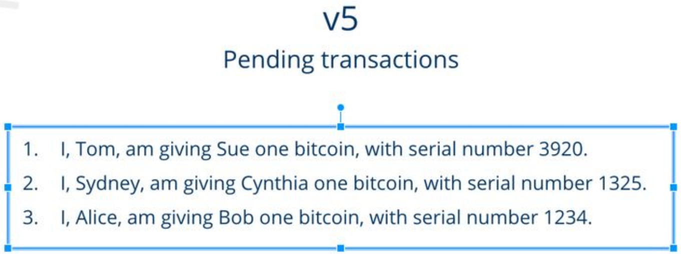
Note

Write an optional message

Coinbase interface

TX: BROADCAST & VERIFY

Slide by Viget



v5

Pending transactions

1. I, Tom, am giving Sue one bitcoin, with serial number 3920.
2. I, Sydney, am giving Cynthia one bitcoin, with serial number 1325.
3. I, Alice, am giving Bob one bitcoin, with serial number 1234.

- I want to send money to @roasbeef 🍖
  - Sign transaction
  - **Broadcast** to network
    - A **flooding algorithm** or **gossip protocol** gets the entire network to hear about a transaction
    - Miners are listening for transactions
- **Verify** transaction with four checks*:
  - Does the script for each previous output being redeemed return true?
  - Have all redeemed outputs not been spent?
  - Have I already seen this transaction? Do not relay it if so.
  - Only accept and relay "standard scripts": based off a small whitelist of scripts
  - *The actual software is a lot more complicated than this

BLOCK: BROADCAST & VERIFY

DEV DECAL APR 2018

- After a while, miner finds PoW, **broadcasts** valid block
    - Block propagates across the network
- Miners **verify the block**
    - **Verify** the block header
    - **Verify** each of the transactions
        - Signature correct, script correct, output correct, etc
    - Miners start work on the next block

TX: ADD TO HISTORY

DEV DECAL APR 2018

Transactions

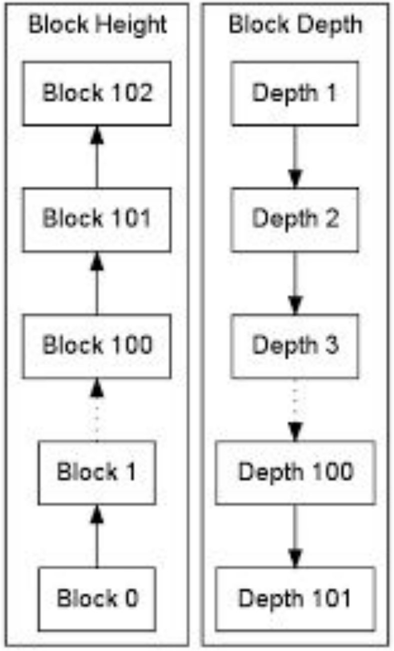
- Maps input addresses to output addresses
  - Typical tx: one input, two outputs
- Contains signature of owner of funds

Blocks

- Contains an ordered bunch of transactions
- Timestamps the transactions, are **immutable**
- Each block references a previous block

Blockchain

- The entire series of blocks 'chained' together by its hash



Block Height Compared To Block Depth

Source: Bitcoin Developer Guide

AUTHOR: MAX FANG

TX: FINALIZE

DEV DECAL APR 2018

Need to protect against a **double spend attack**. The attack:

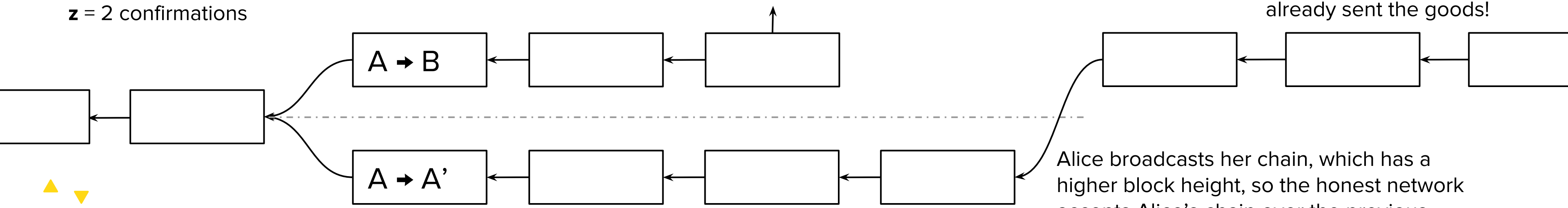
- Max makes a transaction to @roasbeef 🍖. @roasbeef 🍖 waits **z** confirmations sending goods.
- Max creates a double spend transaction in his private chain
- Honest network finds **z** blocks, @roasbeef 🍖 sends the goods
- Max broadcasts a chain longer than **z** blocks *after* @roasbeef 🍖 sends the goods.
- Max gets the good for free!

Suppose Bob waits for
z = 2 confirmations

Honest Network

z = 2 confs, Bob sends goods

Now, Alice has her money back, even though Bob already sent the goods!



Alice mining blocks, withholding from honest network

Alice broadcasts her chain, which has a higher block height, so the honest network accepts Alice’s chain over the previous chain.

LIMITATIONS OF BITCOIN

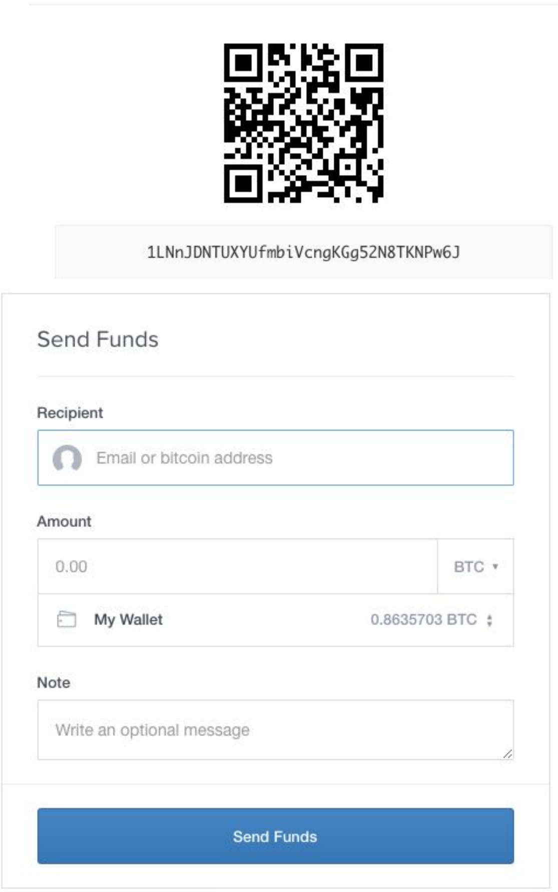
DEV DECAL APR 2018

What just happened?

A transaction from Max to @roasbeef 🍖 :

- Was broadcast to the whole world, and **no one cares**
- Made every Bitcoin node **waste computation** by verifying that transaction and the block that contained it
- Was added into the blockchain history, and now **everyone has to store it forever** 🍖
- Required @roasbeef 🍖 to **wait an hour** before he can safely accept the payment.
- Needs to fit into the **~3 tx/s** allowed by the Bitcoin network

Can we do better?



1LNnJDNTUXYUfmbiVcngKGg52N8TKNPw6J

Send Funds

Recipient: Email or bitcoin address

Amount: 0.00 BTC

My Wallet 0.8635703 BTC

Note: Write an optional message

Coinbase interface

2

INTRO TO LIGHTNING

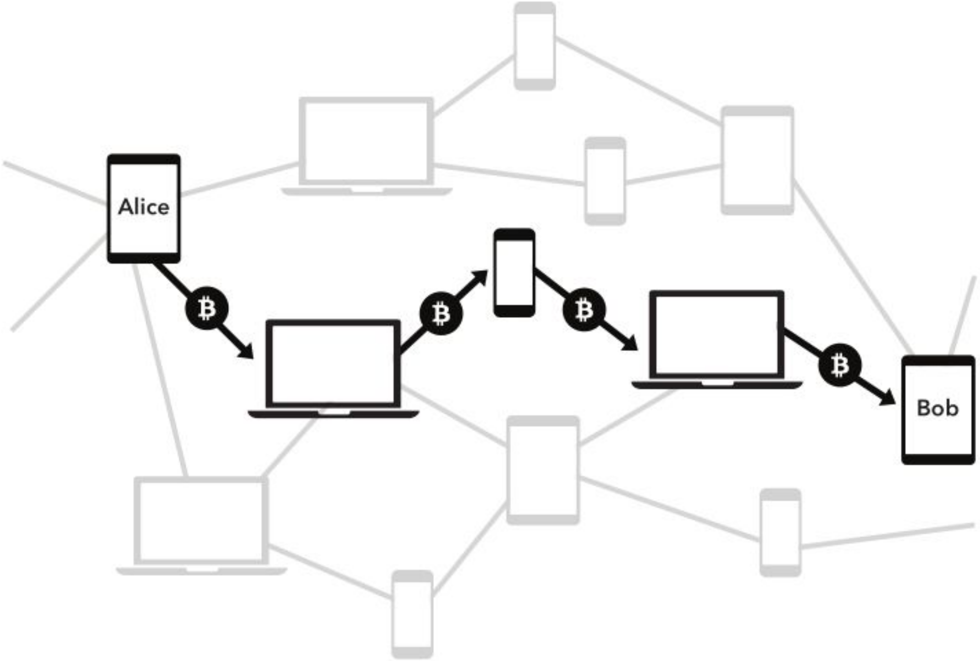
LIGHTNING NETWORK

DEV DECAL APR 2018

Lightning enables scalable blockchains through high-volume instant transactions without custodial delegation.

Transactions are secured in a “local consensus” that rely on the security of the underlying blockchain.

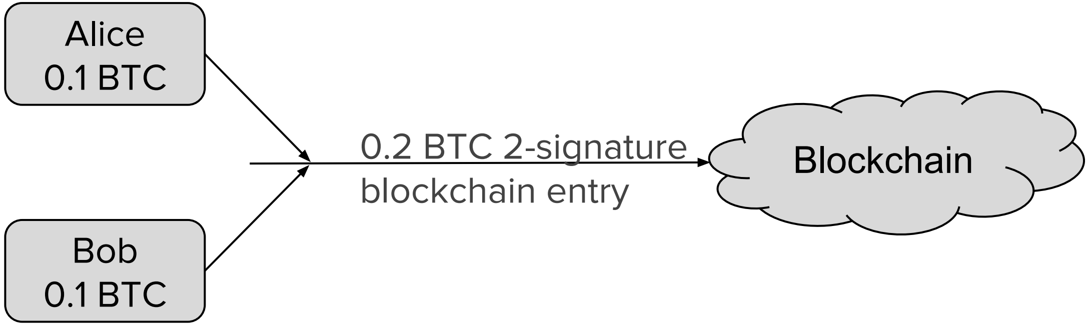
Lightning Network is infrastructure for speeding up digital ledger technology.



AUTHOR: JOSEPH POON

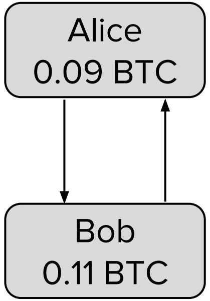
HOW THE LIGHTNING NETWORK WORKS

DEV DECAL APR 2018



0.2 BTC 2-signature ledger entry

Two participants assign funds on the blockchain into an entry which **requires both parties to sign** off to spend from the blockchain entry.



Alice and Bob **exchange digital signatures** directly with each other every time they want to **update their balance** in this local off-blockchain "channel."

They can **close out and settle the transaction at any time** on the blockchain claiming their off-chain balance (since they've signed off but not broadcasted the old balances). They have also exchanged directly with each other a **cryptographic bonded proof revoking the old balances**, so only the newest one can be used.

CRYPTOGRAPHIC HASH FUNCTIONS

A **cryptographic hash function**

$$H : \{0,1\}^* \longmapsto \{0,1\}^k$$

Maps some **arbitrarily-sized bit string** to some **fixed-size bit string**.

The function only **evaluates in one direction**; you can't find the input given just the output.

The function is **"deterministic"**; the same input always yields the same output.

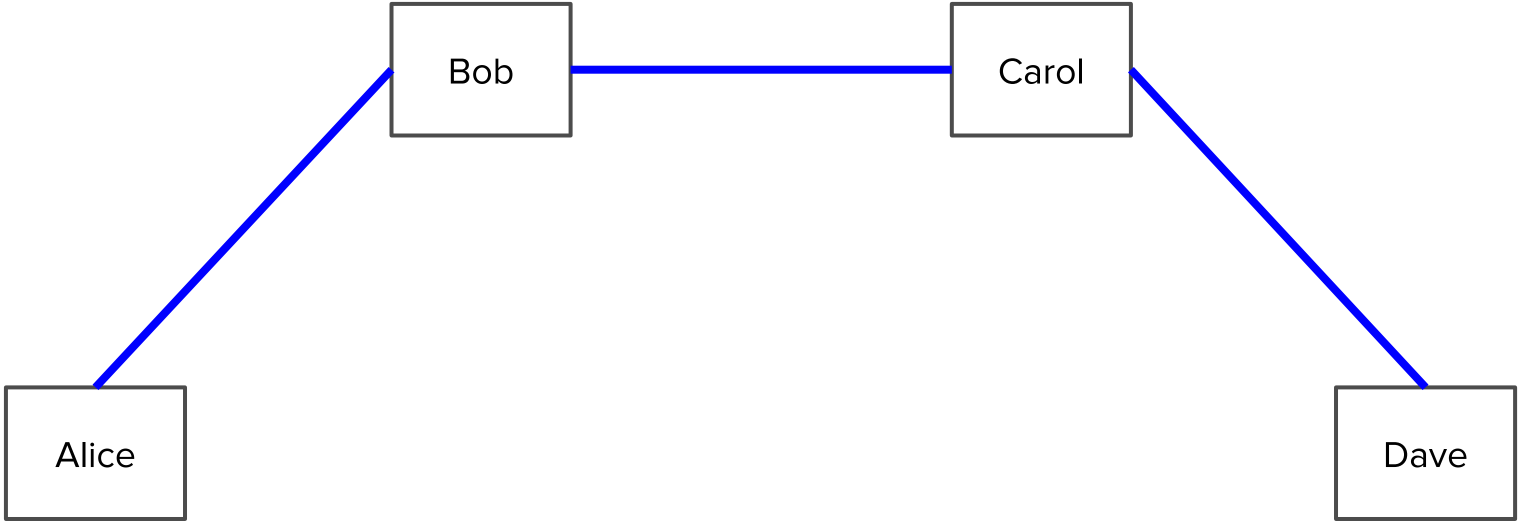
“The workhorses of modern cryptography”

- Bruce Schneier

3+ PARTY NON-CUSTODIAL CLEARING

DEV DECAL APR 2018

Alice wants to pay Dave without opening a new channel

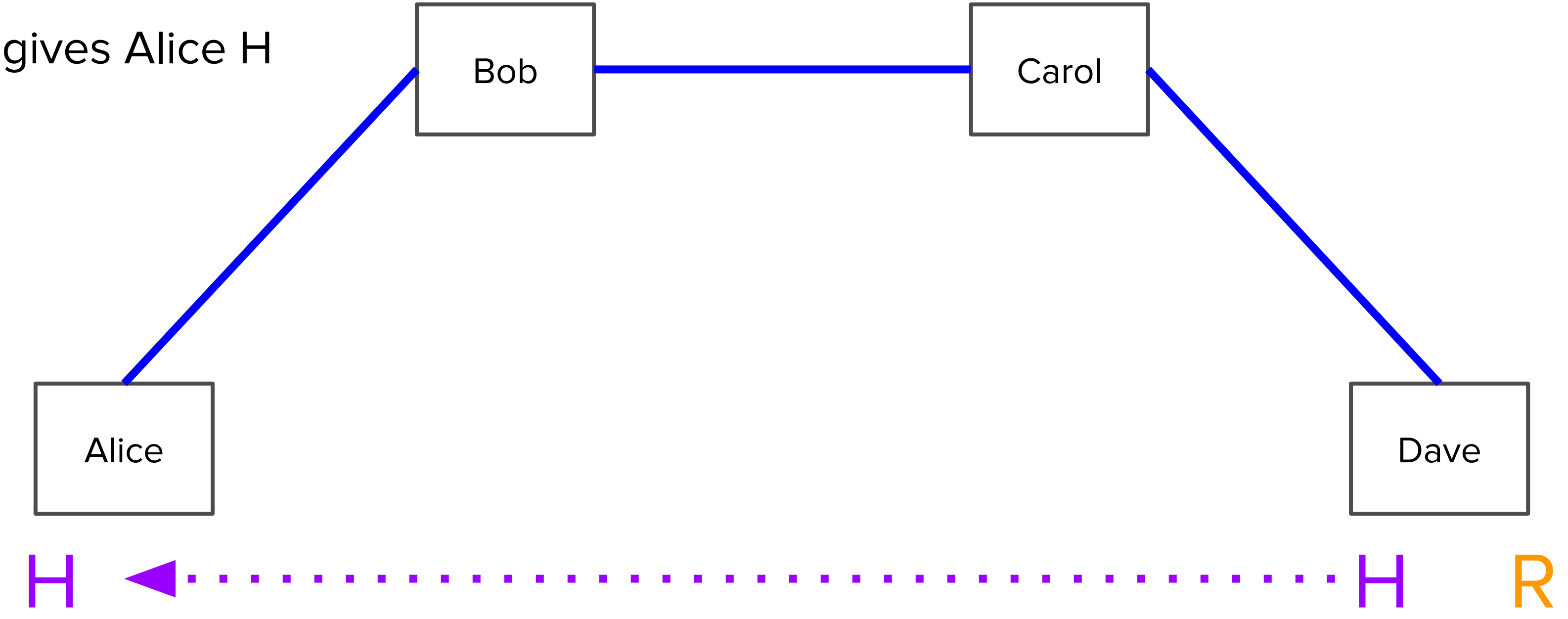


3+ PARTY NON-CUSTODIAL CLEARING

DEV DECAL APR 2018

Dave makes a random number R and hashes it to H.

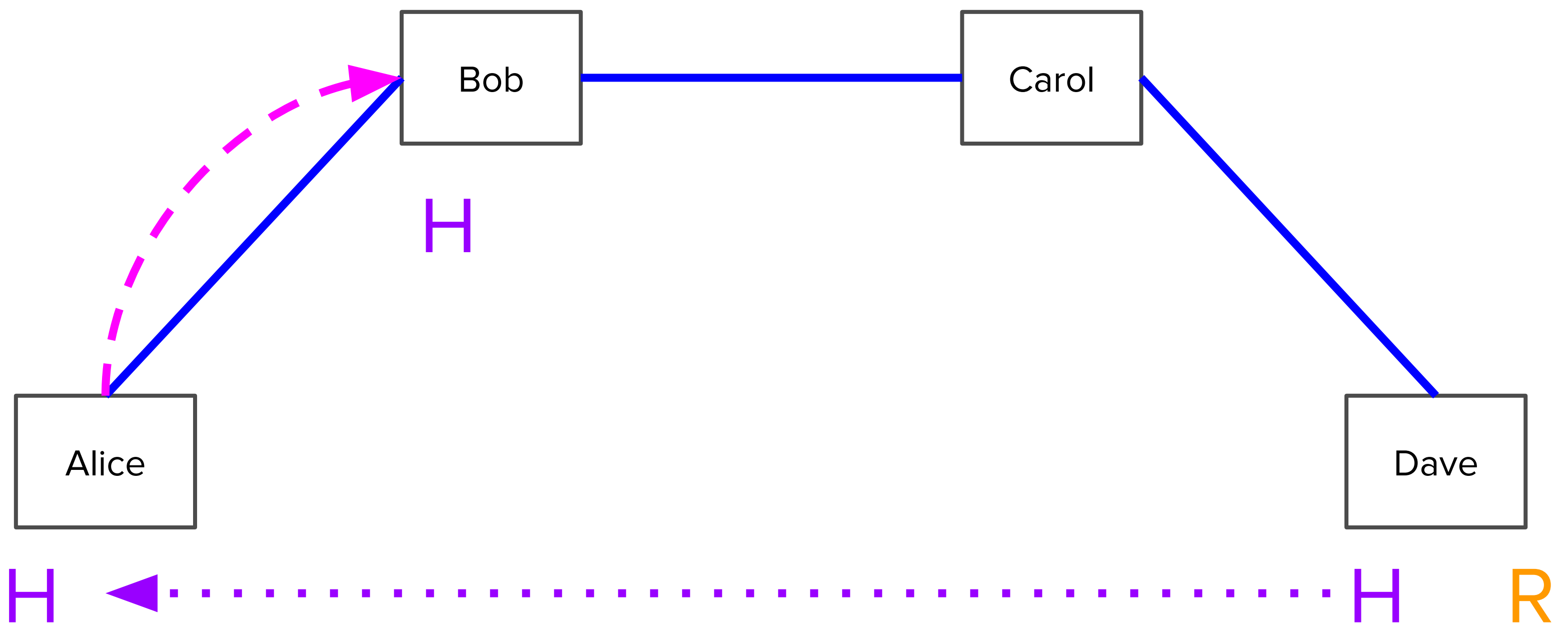
Dave gives Alice H



3+ PARTY NON-CUSTODIAL CLEARING

DEV DECAL APR 2018

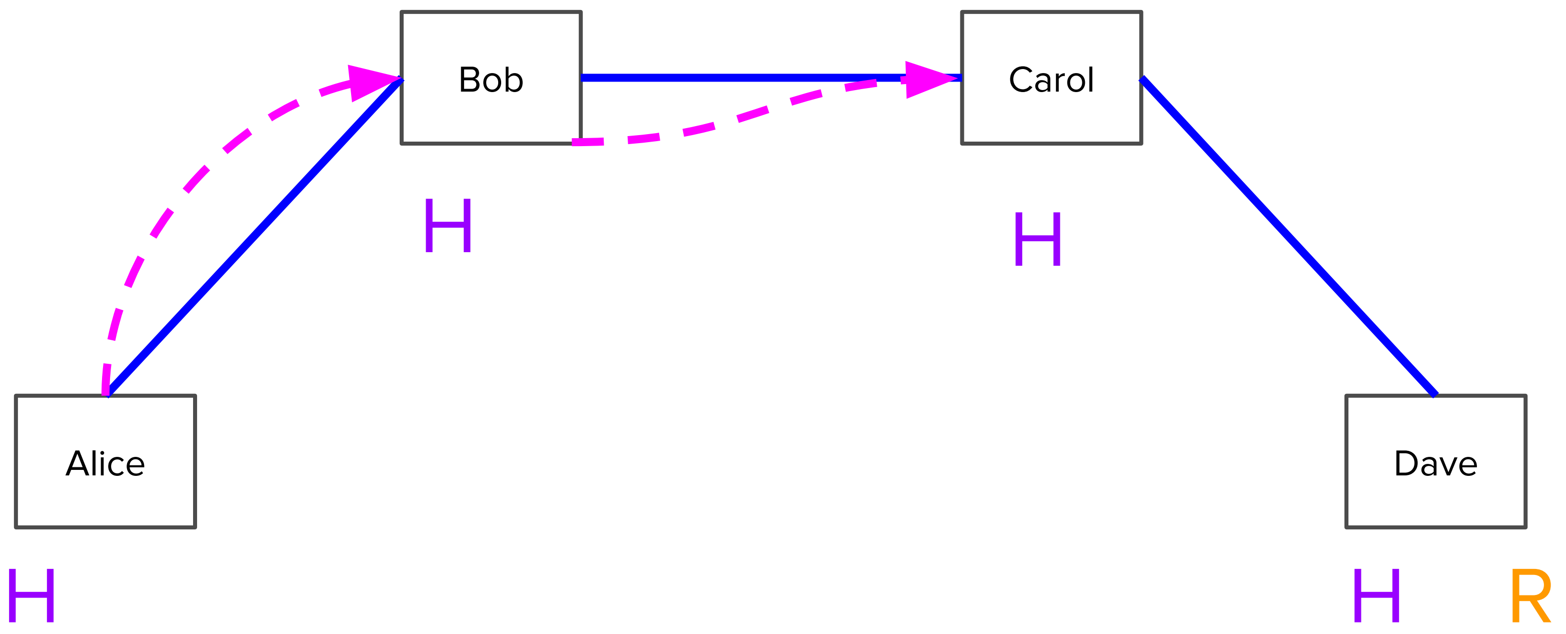
Alice pays Bob, but only if he knows R, the pre-image of H



3+ PARTY NON-CUSTODIAL CLEARING

DEV DECAL APR 2018

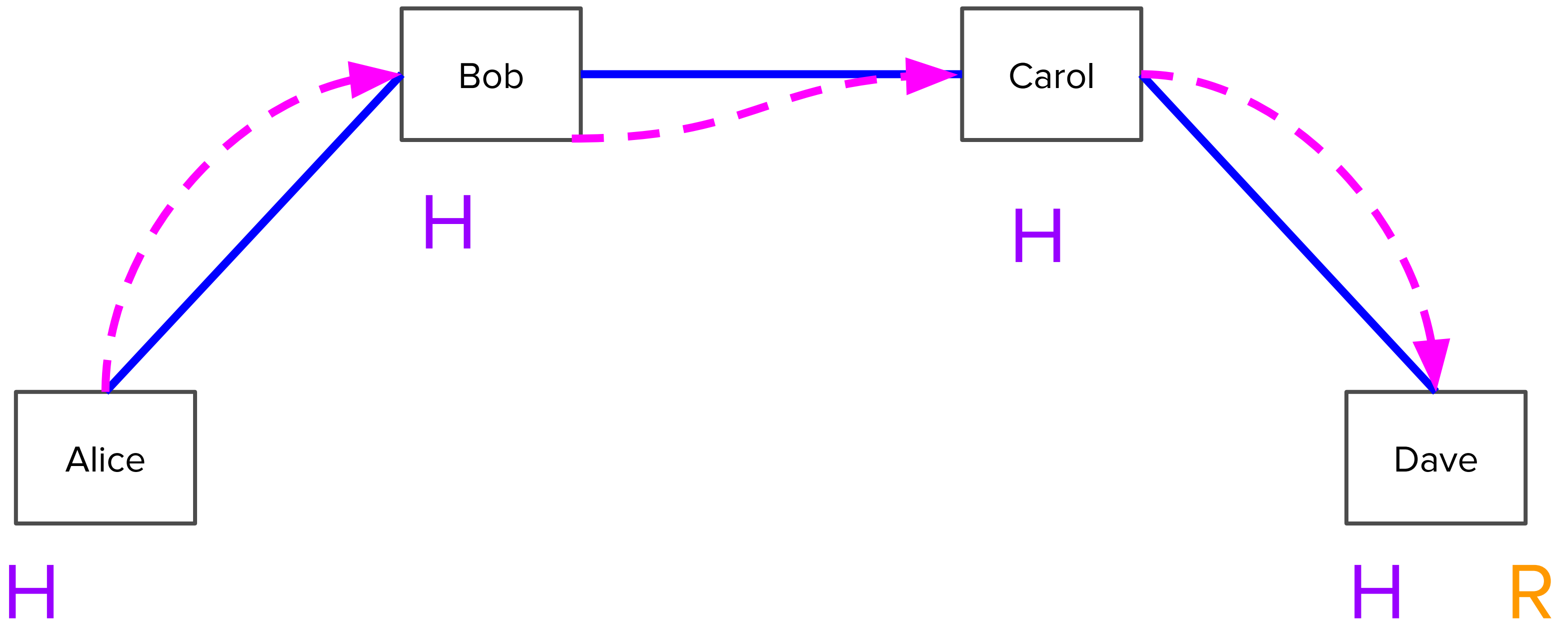
Bob pays Carol, but only if she knows R, the pre-image of H



3+ PARTY NON-CUSTODIAL CLEARING

DEV DECAL APR 2018

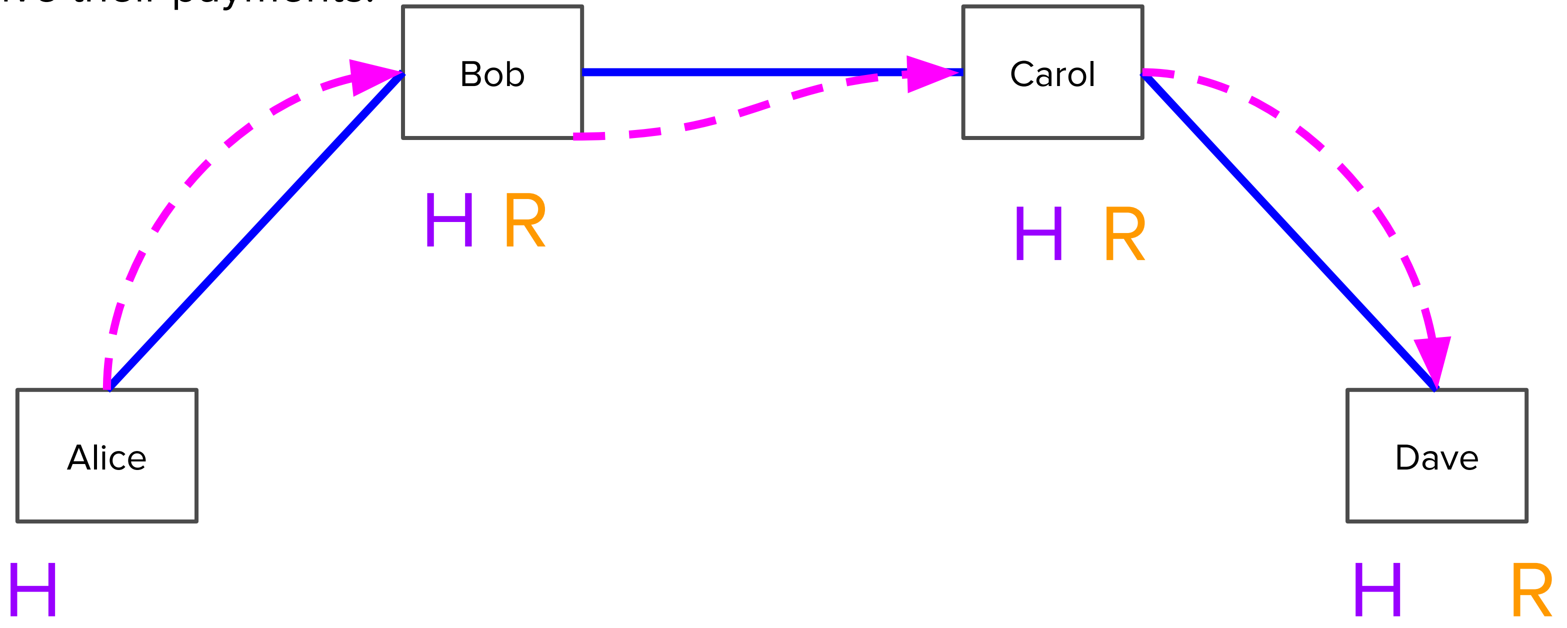
Carol pays Dave, but only if he knows R... and he does!



3+ PARTY NON-CUSTODIAL CLEARING

DEV DECAL APR 2018

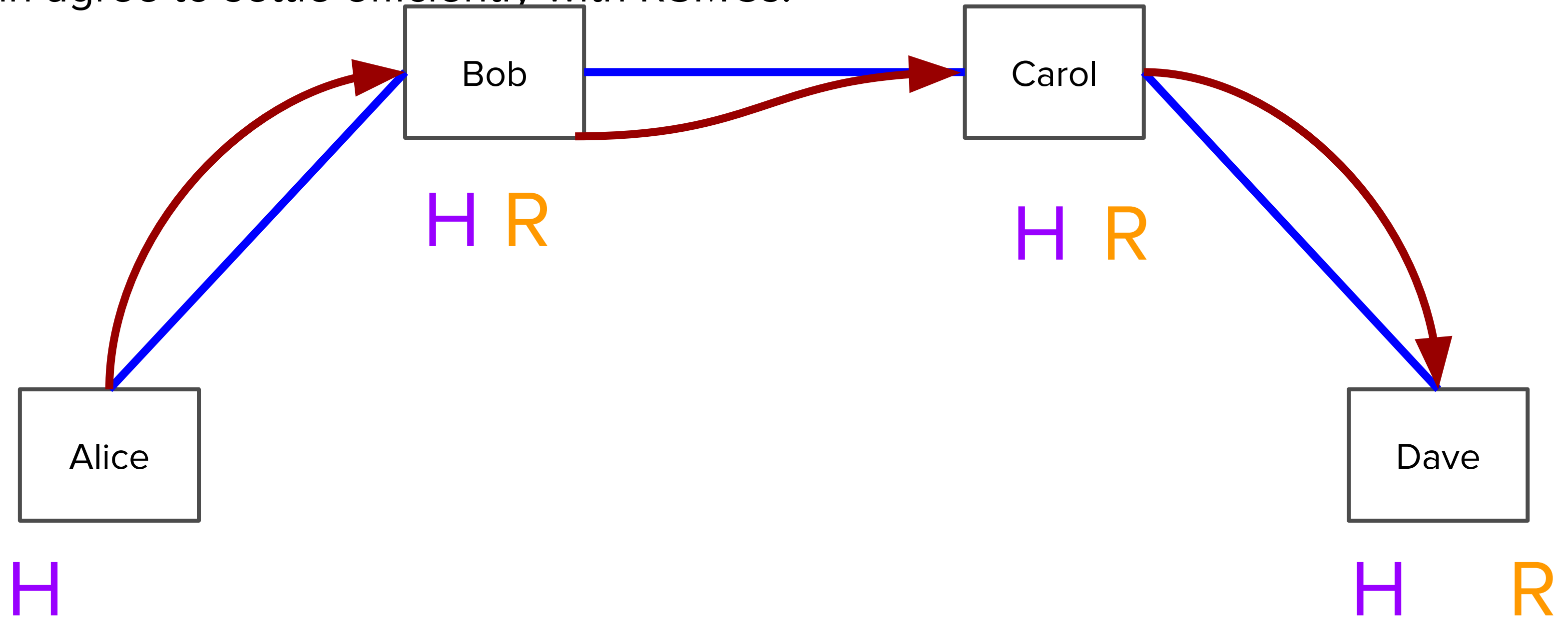
When Dave receives the payment, he must reveal R. Revealing R allows Carol and Bob to receive their payments.



3+ PARTY NON-CUSTODIAL CLEARING

DEV DECAL APR 2018

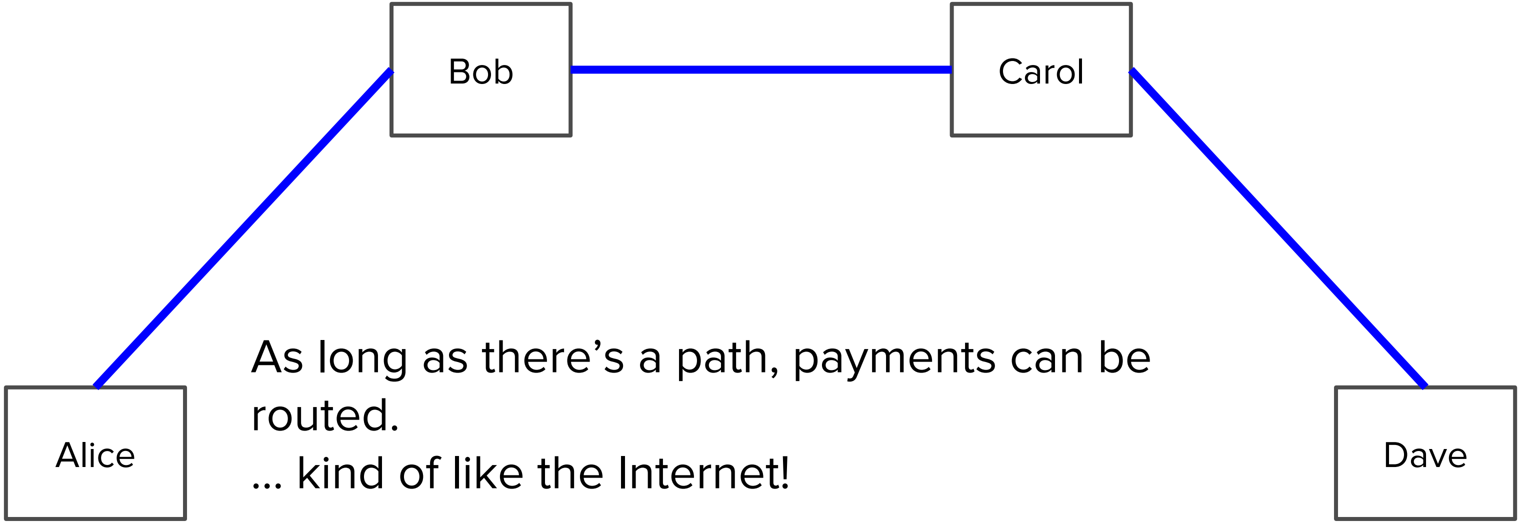
Alice<->Bob, Bob<->Carol, and Carol<->Dave now have a safeguard against stealing, so they can agree to settle efficiently with RSMCs.



3+ PARTY NON-CUSTODIAL CLEARING

DEV DECAL APR 2018

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



As long as there’s a path, payments can be routed.
… kind of like the Internet!

ADVANTAGES OF LIGHTNING

DEV DECAL APR 2018

- Micropayments:
  - Possible to send **1 satoshi** ($0.00001) **instantly** in open+decentralized network
- Scalability via amortization:
  - Channel setup: 1 **on-chain** transaction
  - Channel operation:
    - Can send an **unbounded** number of payments **off-chain**
    - With a **few channels** can reach **entire network** via multi-hop payments
    - Fees are **predictable** and **known a priori**
    - Channels can stay open **indefinitely**
  - Channel closure: 1 **on-chain** transactions, can be closed **unilaterally**
- Higher level abstraction for developers:
  - Payment invoices/requests
  - Instant success/error
- Tradeoffs: not suitable for **very** large payments, liveness assumptions

SECOND LAYER BITCOIN PROTOCOLS

DEV DECAL APR 2018

- Layer 1:
  - Write **directly** to the Blockchain
  - Highest security (each write backed by PoW)
  - Best for **high-value** payments
  - Constrained to **3-7** transactions per second

- Layer 2:
  - Uses **security** of Layer 1
  - Greater flexibility via Bitcoin’s scripting capabilities
  - Low cost
  - Point-to-point, greater scalability
  - Suitable for **real-time** use cases
  - Day-to-day payments (thousands of transactions per second)

USING LIGHTNING INSTEAD

DEV DECAL APR 2018

A transaction from Max to @roasbeef 🍖:

- Was **known and computed only by participating parties**
- Did not add to the forever-growing blockchain history stored by all nodes
- Allowed @roasbeef 🍖 to instantly and safely accept the payment.
- Cost Max **satoshi amounts** in fees
- Has no limit on the number of transactions per second.

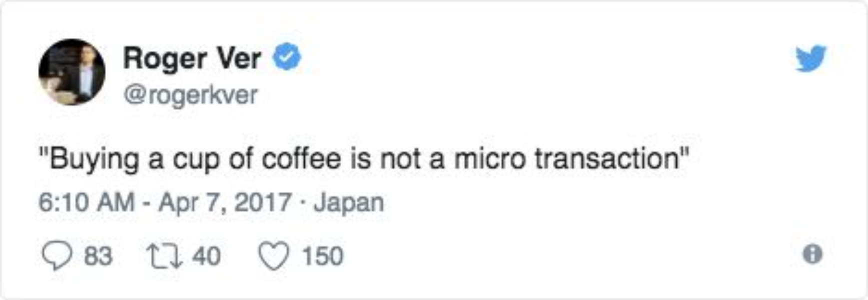
Much better.

Coinbase interface

LIGHTNING NETWORK

DEV DECAL APR 2018

With Lightning, you can finally pay for a coffee with Bitcoin

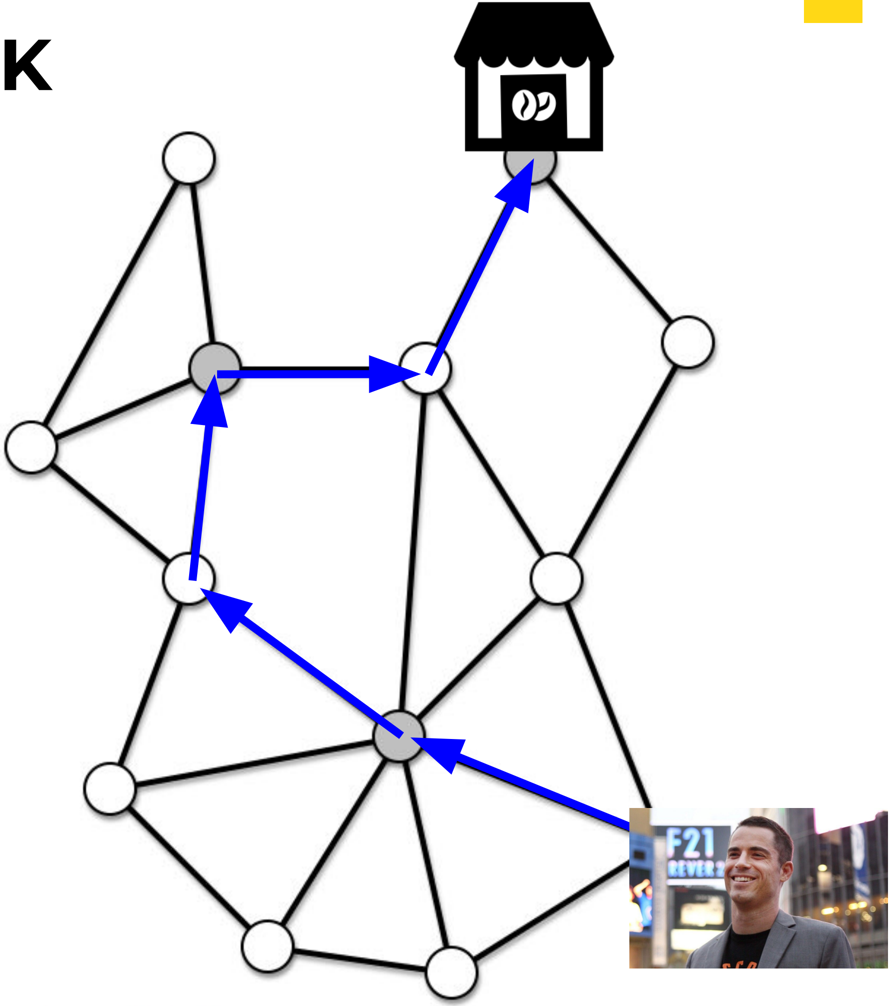


Roger Ver @rogerkver

"Buying a cup of coffee is not a micro transaction"

6:10 AM - Apr 7, 2017 · Japan

💬 83 🔁 40 ♡ 150



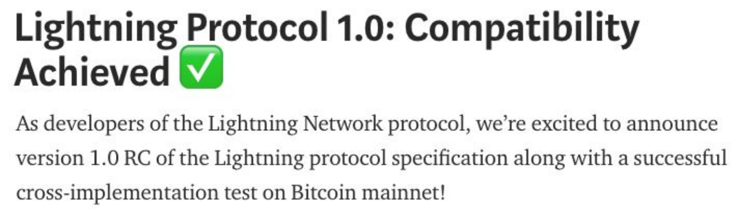
4

THE STATE OF LIGHTNING

STATE OF THE NETWORK

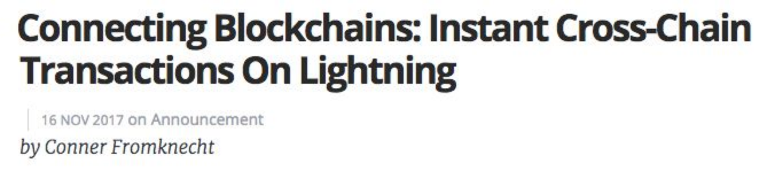
DEV DECAL APR 2018

- Lightning Network In-Progress specifications (lightning-rfc)
  - Basis of Lightning Technology (BOLT)
  - Specs cover: funding process, key derivation, p2p interaction, messages, etc.
- **4+** implementations being **actively developed** e.g:
  - lnd, eclair, lightningd (c-lightning), lit
  - Implementations are interoperable! (Lightning Protocol 1.0)
- Development around cross-chain swaps!
  - Nov 2017 Lightning Labs demonstrated the first cross-chain atomic swap between Bitcoin and Litecoin testnets!
- Growing list of LApps at dev.lightning.community/lapps/
  - ~33 Lapps as of March 2018
  - Developer tools, games, streaming, visualizations, network stats



Lightning Protocol 1.0: Compatibility Achieved ✅

As developers of the Lightning Network protocol, we're excited to announce version 1.0 RC of the Lightning protocol specification along with a successful cross-implementation test on Bitcoin mainnet!



Connecting Blockchains: Instant Cross-Chain Transactions On Lightning

16 NOV 2017 on Announcement

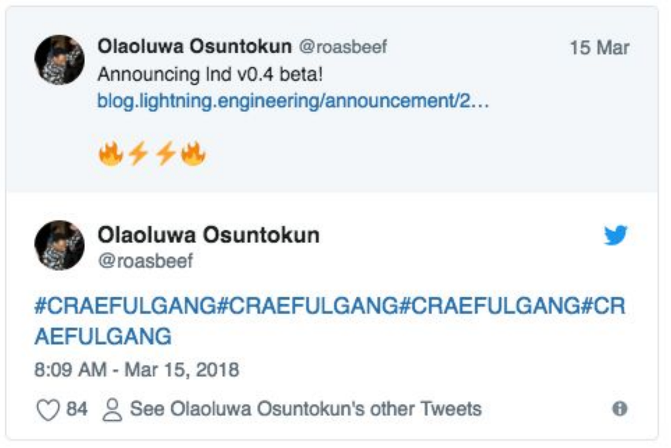
by Conner Fromknecht

AUTHOR: LAOLU OSUNTOKUN

LND - THE LIGHTNING NETWORK DAEMON

DEV DECAL APR 2018

- Code: https://github.com/lightningnetwork/lnd/
  - Uses the **btcsuite (a.k.a btcd)** set of Bitcoin libraries
- Developed by **Lightning Labs.** Lead developer: **roasbeef**
- Recently released **v0.4-beta: MAINNET RELEASE!**
  - Developer-oriented; desktop/mobile app support soon
  - Recommended experimentation with only small amounts
  - New Capabilities:
    - **bitcoind** support in addition to **btcd**
    - Easier backup, and recovery from data loss
    - Vastly improved **fault-tolerance** (persistence)
    - Smarter Path-Finding
    - Automated contract resolution
    - Segwit only
    - Routing node metrics



Olaoluwa Osuntokun @roasbeef 15 Mar

Announcing lnd v0.4 beta!

blog.lightning.engineering/announcement/2...

🔥⚡⚡🔥

Olaoluwa Osuntokun

@roasbeef

#CRAEFULGANG#CRAEFULGANG#CRAEFULGANG#CRAEFULGANG

8:09 AM - Mar 15, 2018

♡ 84 See Olaoluwa Osuntokun's other Tweets

Lightning Labs $2.5 million round

- Investors: Jack Dorsey, Charlie Lee, Ben Davenport, Digital Currency Group and others
- Upcoming Lightning ICO! Ticker OST (Osuntokuns)
  - (no, not really)

4

LIGHTNING USE CASES

MICROPAYMENTS

DEV DECAL APR 2018

LIGHTNING

The **Lightning Network** enables a high volume of instant, low-fee payments while retaining a fully trustless, decentralized nature

- Utilizes "payment channels," allowing for the amortization of transaction fees
- Relies on the security of the underlying blockchain

Benefits

- Enables decentralized global scale payment network **competitive with credit cards**
- In Bitcoin, **possible to send 1 satoshi** ($0.00001) per payment
- **Real-time payments** with **instant confirmations**
  - No more waiting an hour to safeguard against double spends
- **Enhanced privacy** - transactions by default only shared between interested parties
  - Can also add onion routing on top

MICROPAYMENTS

DEV DECAL APR 2018

Media Use Cases

- Payroll by the minute
- Micropayment paywalls
  - Replacing ads or expensive subscriptions with "freemium" models
    - Save media companies!
  - Pay per article, pay per video
- Replace virtual game currency with real currency

IoT Use Cases

- Buzzwords like "Machine-to-machine payments"
- IBM ADEPT
- Pay for metered energy

AUTHOR: MAX FANG

MICROPAYMENTS

LIGHTNING

Tech Use Cases

- DDOS-resistant internet
  - Embed a micropayment in each internet packet
- Pay-per-API call
  - Anonymous API tokens
- Pay for Internet by the megabyte

Financial use cases

- Alternative payment method: low-free, irreversible payments
- HFT trading between cryptocurrency exchanges
- Decentralized exchanges
- Reduce risk of theft at centralized exchanges
  - Store money in channels instead of accounts
  - Let users be custodians of keys

MICROPAYMENTS

DEV DECAL APR 2018

LIGHTNING

Blockchain Use Cases

- Pay for blockchain services!
  - Running full nodes and other key network infrastructure
  - Storage
  - Computation
  - Oracles
  - Arbitration
  - Almost any other blockchain product that claims to need its own token

Other Use Cases

- Trustless, efficient off-chain betting
- Experimental game-theoretic protocols e.g. to surface high quality content
  - Create profit opportunities for hipsters

CRYPTOCURRENCIES & MICROPAYMENTS

DEV DECAL APR 2018

Conclusions

- Cryptocurrencies are neutral; can serve as common ground between competing entities
  - No primary beneficiary
  - Well-suited for horizontal integration
- Cryptocurrency payments don't require personal information
  - Potentially send money to pseudonyms across the internet
- Cryptocurrencies are **irreversible**
  - May or may not be desired
  - Elizabeth Stark: "Reversible systems can be built on top of irreversible ones, but not vice versa"
- Lightning Network is like the 'checking account' of Bitcoin
  - Lightning adds benefits of microtransactions and instant finality to the existing benefits of cryptocurrencies

5

LIGHTNING DEVELOPMENT

LIGHTNING DEVELOPMENT

Developer site: dev.lightning.community

- Resources
- Tutorial
- Install guide
- Python / Javascript gRPC guide
- Slack

CONCLUSION

Thank you!

Lightning Twitter: @lightning

Max: @maxfangx

- max@lightning.engineering
- Website: maxfa.ng

LIGHTNING

QUESTIONS?

X ARCHIVED SLIDES

TRANSACTION THROUGHPUT LIMITS

DEV DECAL APR 2018

Rough network stats:

- Average of ~340 bytes per transaction (0.68 MB blocks, 2000 tx/block)
- Current blocksize is 1 MiB.
- Expected time to next block is 10 min.

Therefore we can compute the sustained maximum transaction volume in tps:

TPS COMPARISONS

DEV DECAL APR 2018

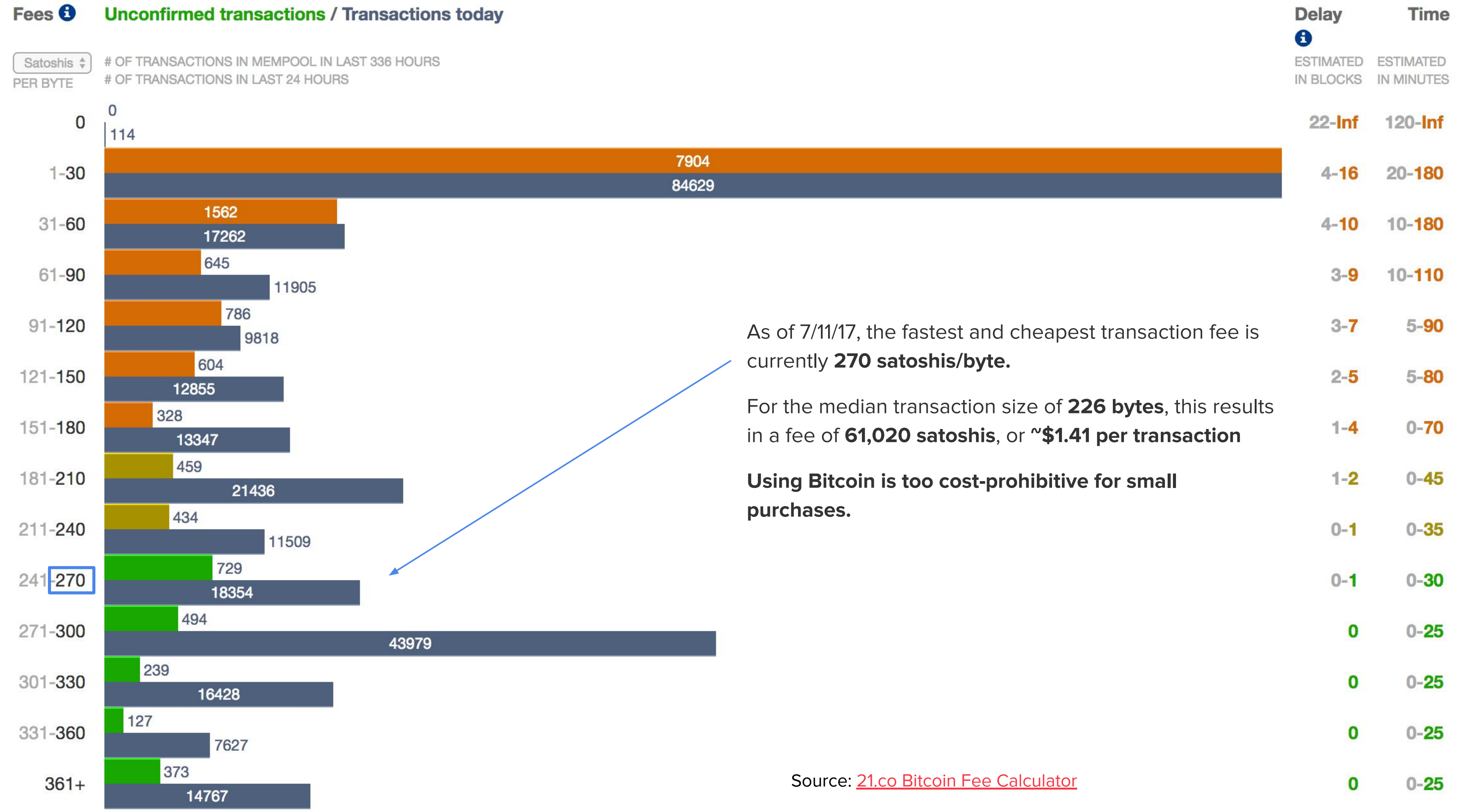
How does Bitcoin compare with other traditional payment systems?

	Average	High Load / Maximum
Bitcoin	3 tps	3.2 tps
PayPal*,**	150 tps	450 tps
VISA***	2,000 tps	56,000 tps

* https://investor.paypal-corp.com/secfiling.cfm?filingID=1206774-16-5430&CIK=1633917

** http://www.fool.com/investing/general/2016/02/04/5-things-paypal-holdings-inc-wants-you-to-know.aspx

*** https://usa.visa.com/dam/VCOM/download/corporate/media/visa-fact-sheet-Jun2015.pdf



Fees (Satoshis per byte)	# of transactions in mempool in last 336 hours	# of transactions in last 24 hours	Delay (estimated in blocks)	Time (estimated in minutes)
0	0	114	22-Inf	120-Inf
1-30	7904	84629	4-16	20-180
31-60	1562	17262	4-10	10-180
61-90	645	11905	3-9	10-110
91-120	786	9818	3-7	5-90
121-150	604	12855	2-5	5-80
151-180	328	13347	1-4	0-70
181-210	459	21436	1-2	0-45
211-240	434	11509	0-1	0-35
241-270	729	18354	0-1	0-30
271-300	494	43979	0	0-25
301-330	239	16428	0	0-25
331-360	127	7627	0	0-25
361+	373	14767	0	0-25

As of 7/11/17, the fastest and cheapest transaction fee is currently **270 satoshis/byte.**

For the median transaction size of **226 bytes**, this results in a fee of **61,020 satoshis**, or **~$1.41 per transaction**

Using Bitcoin is too cost-prohibitive for small purchases.

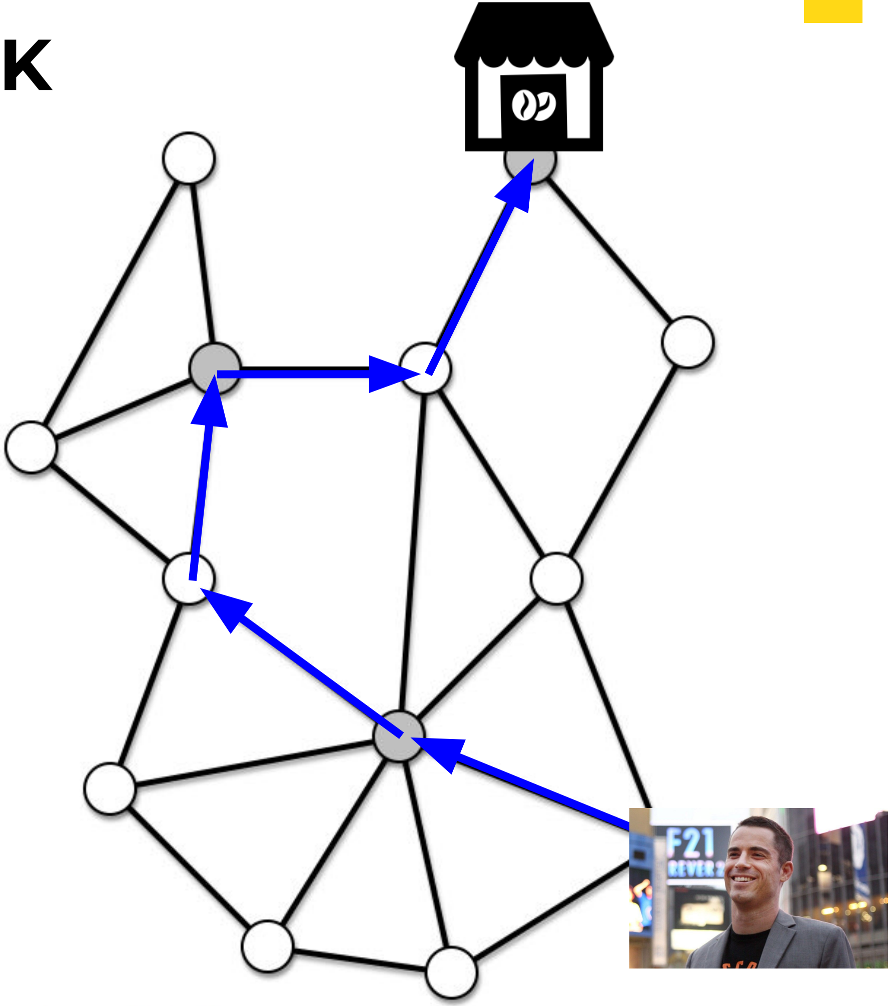
Source: 21.co Bitcoin Fee Calculator

5 ADVANCED SLIDES

LIGHTNING NETWORK

DEV DECAL APR 2018

With Lightning, you can finally pay for a coffee with Bitcoin



BITCOIN CONTRACTS

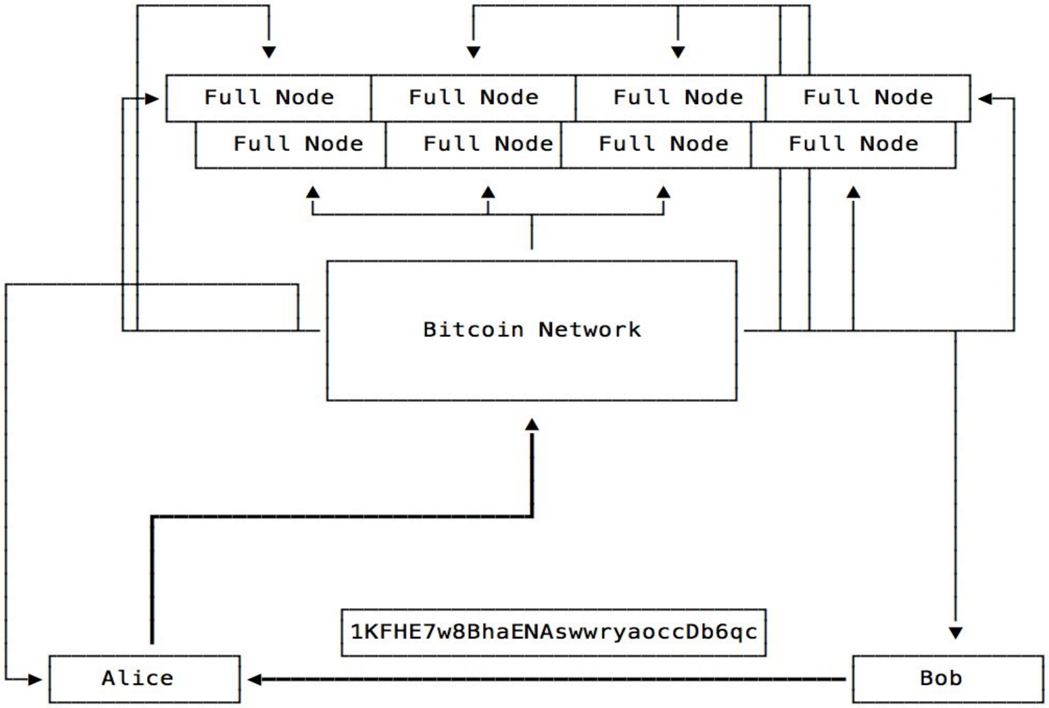
DEV DECAL APR 2018

- Bitcoin’s core is a **distributed**, highly replicated **deterministic VM**:
  - We call this VM: “Script”
    - Uses a Forth-like stack-based programming language
  - Nodes **globally** process inputs to this VM in **lock-step**
  - Building block in **advanced** cryptographic protocols
- Programs in Script (predicates):
  - Public Key Script: program which encodes **redemption conditions**
  - Witness: input to program, if returns **True** then the spend is **permitted**
- Example payment script:
  - pkScript: `OP_DUP OP_HASH160 <pubKeyHash> OP_CHECKSIG`
  - sigScript: `<signature> <pubKey>`
  - Stack eval: sigScript on stack, **eval** with pkScript

BITCOIN PAYMENTS TODAY

DEV DECAL APR 2018

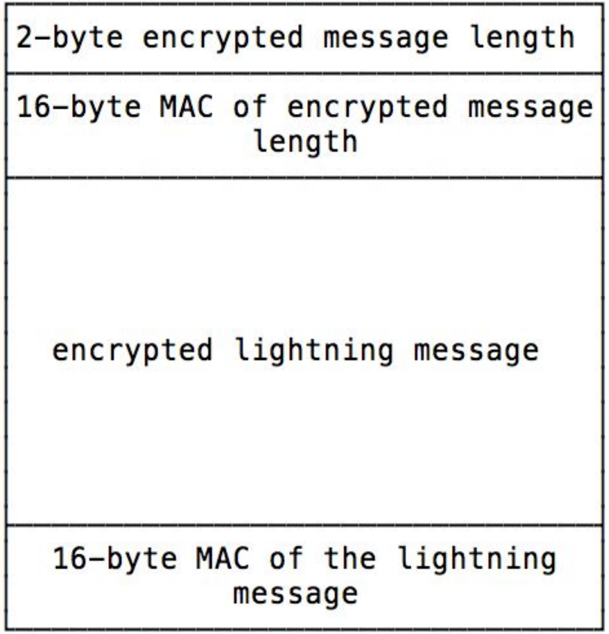
- All participants connected to **global** network
- **All** payments broadcast to **all other** participants
- **Each** payment **must** be fully verified
- Drawbacks:
  - Scalability limitations of **global broadcast** network
  - Each node does work even if not involved in payment
  - **Public** record of **each** payment kept for **ALL TIME**



PEER-TO-PEER NETWORKING LAYER

DEV DECAL APR 2018

- **All** communications between nodes **encrypted+authenticated:**
  - No protocol messages sent until **brontide** session initiated
- Brontide (BOLT #8):
  - Variant of the **Noise Protocol Framework** (brontide):
    - Framework for **Authenticated Key Agreement**
    - Init: series of **handshake** messages (ECDH+hashing)
    - Transport: **AEAD** cipher mode used for encryption
  -
    ```
    Noise_XK_secp256k1_ChaChaPoly_SHA256
    <- s
    …
    -> e, es
    <- e, ee
    -> s, se
    ```
  - Hash rachet for **key rotation**



2-byte encrypted message length

16-byte MAC of encrypted message length

encrypted lightning message

16-byte MAC of the lightning message

PEER-TO-PEER NETWORKING LAYER

DEV DECAL APR 2018

- Nodes identified on the network by **public key**
- **Bitcoin keys** and **node keys** used to authenticate information
    - Node Announcement:
        - Announces node existence: PubKey+sig, reachability
        - **Global features**
    - Channel Announcement (channel proof):
        - Channel ID: **location** of funding output in chain (8-bytes)
        - 4 keys: two multi-sig keys, two node keys
            - Verify: `2 <key1> <key2> 2 OP_CHECKMULTISIG`
        - 4 sigs:
            - Can be compressed to single key w/ **signature aggregation**
        - Verification can be sped up via **batch signature** verification
    - Channel Update Announcement:
        - Advertises **routing policy** for a **directed** channel edge
        - Signed by node advertising

ONION-ENCODED PAYMENT ROUTES

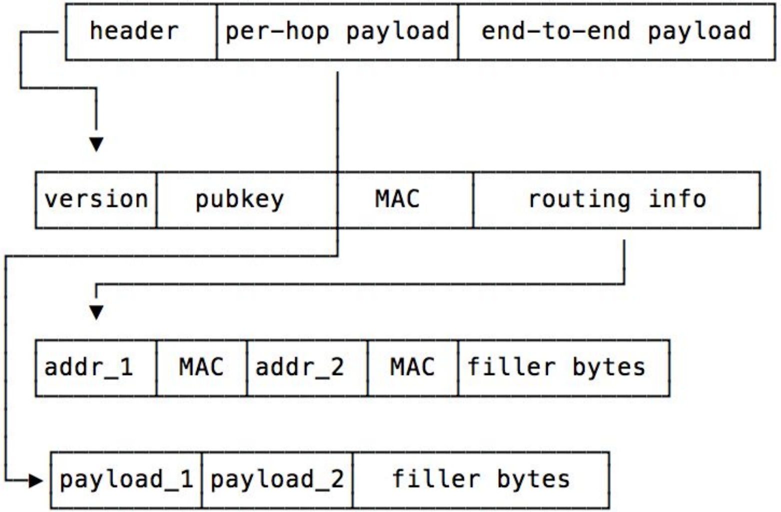
DEV DECAL APR 2018

- Sphinx: compact, provably secure **mix-net** packet format
  - Used within lightning as basis for **onion routing**
  - **Fixed-sized** payload
  - Modified version in BOLT #4
- Security Features
  - Nodes don’t know their **location** in the route
    - Packet remains **fix sized** during processing
  - Nodes don’t how **how long** the route really was
    - All packets encode the **max hop** limit
  - Nodes only know their predecessor and successor
    - Received from downstream node, contains instructions to forward
  - All packets **indistinguishable** from all others
    - Packet is **re-randomized** at each hop
- Shared secret re-used to back-propagate **error messages**

ONION-ENCODED PAYMENT ROUTES

DEV DECAL APR 2018

- Payments routed through network using **source routing:**
  - Gives sender **total control** over payment path
  - Authenticated per-hop payload:
    - Outgoing time-lock (#blocks or time)
    - Satoshis to forward (ensure proper fees)
    - Outgoing "realm" (Bitcoin, Litecoin, etc)
- Replay attack prevention:
  - Each Sphinx packet commits to the **payment hash**
  - HTLC's past **absolute** are rejected
- Routes still subject to **traffic+timing** analysis
  - Poor path diversity weakens security



AUTHOR: LAOLU OSUNTOKUN

BIDIRECTIONAL PAYMENT CHANNELS

- Lightning uses two Bitcoin contracts to ensure proper execution
  - Some call this “**Cryptoeconomics**”
- Contract #1: The Hash-Time-Locked-Contract (HTLC)
  - Set up: **receiver** gives **sender** `H = Hash(R), where R <-$ {0, 1}^n`
  - Conditions: “I will pay you N BTC, iff you present R s.t `Hash(R) == H`”
  - Escape hatches: “If you don’t within T days, I get my money back”
  - Result: **end-to-end** secure **multi-hop** payments through **untrusted intermediaries**
- Contract #2: Commitment State Revocations
  - Set up: each **commitment state** references secret from other side
  - Conditions: each **update** requires **revealing** that secret
  - Escape hatches: if **prior state** is broadcast, knowledge of secret entitles other to **all funds**
  - Result: both sides **incentivized** to *always* only save/broadcast the **latest state**
- **Optimistically,** the escape hatches or punishments are **never needed**

BLINDED CHANNEL OUTSOURCING

DEV DECAL APR 2018

- Lightning requires parties to occasionally **monitor** the blockchain
  - **TEE** based schemes (e.g. Teechan) can help
  - Each “pay-to-self” output has a **relative time-lock** (CSV)
  - Delay acts as **adjudication** period
- Responsibility for watching the chain can be outsourced to a third-party
  - With `SIGHASH_NOINPUT` or MAST, can reduce storage to `O(log(N))`
    - `N = number_of_commitment_updates`
  - For now, we can at least make the process more **private**
- Blinded Channel Monitoring
  - Outsourcer shouldn’t be able to distinguish **which** channel they’re watching
  - Achieved by **randomizing** commitment keys on each update
  - Able to **collapse** the revocation state, saving disk-space
    - “Reverse” merkle-tree -- once you have parent, can **discard** children

ON-CHAIN LIVELINESS

DEV DECAL APR 2018

- Security model of Lightning:
    - Relies on Bitcoin for **ordering of transactions**
    - Dependent on **time-based** windows of action `(T)`
        - Longer `T` (CSV delay) provides more security during **channel breaches**
        - Longer `T` also results in unavailability of funds for **unilateral closes**
        - In the optimistic case: higher `T`, as closures assumed to be **cooperative**
- Thundering herd failure mode
    - Massive **network-wide channel closure** clogs up chain
    - Depending on `T` (unique to each channel), and duration of backlog, adversaries may profit
- Possible solutions:
    - Time-stop
        - “Relative time-lock” stops ticking above “higher-water” mark
    - Consensus enforced transaction **dependency**
        - Covenant or op-code
    - Fee-based dependency: CPFP