Payment Channel Networks for Blockchain-based Cryptocurrencies: Why, What and How?

Guoliang Xue

Arizona State University

ARIZONA STATE UNIVERSITY

Outlines

- The “Why”s for cryptocurrency and PCN
- The “What” for PCN
- The “How”s for PCN
- From “W” to “C”: Conclusions

A Little Bit of History of Money

Bartering — 9000 BC

2000 BC — Cowry Shells

Metallic Money — 1000 BC

1000 AC — Paper Money

Cryptocurrencies — 2008

Fig 1: https://urbantips.wordpress.com/2012/04/03/im-bringing-back-the-barter-system/
Fig 2: https://www.moneymuseum.com/en/coins/lead-currencies?&id=884
Others are noncommercially reusable based on Google Images

Why is money evolving?

Bartering | 9000 BC

Convenient and Durable
(storage, transport)

2000 BC | Cowry Shells

Durable, Rare,
Unforgeable

Metallic Money | 1000 BC

Convenient and Flexible
Cost-efficient

1000 AC | Paper Money

Convenient, Inflation-resistant,
Secure, Unforgeable, …

Cryptocurrencies | 2008

Arizona State University

Fig 1: https://urbantips.wordpress.com/2012/04/03/im-bringing-back-the-barter-system/
Fig 2: https://www.moneymuseum.com/en/coins/lead-currencies?&id=884
Others are noncommercially reusable based on Google Images

Why digital cash / cryptocurrencies?

- Modern assets have already been digitized
  - Online accounts, credit cards, online stocks / futures / options, …
- Need fast & convenient & inexpensive way for global payment
  - Traditional bank settlement: typically 1-3 days, transaction fees
- Universal accessibility / 7/24 finance
- Fear of inflation
- Fear of loss due to market crash / government manipulation / freezing / human error / forged paper bills / identity theft / …
- Anonymity / untraceability

Cryptocurrency = Crypto + Currency

A digital asset designed to work as a medium of exchange that uses cryptography (blockchain) to secure its transactions. [Wikipedia]

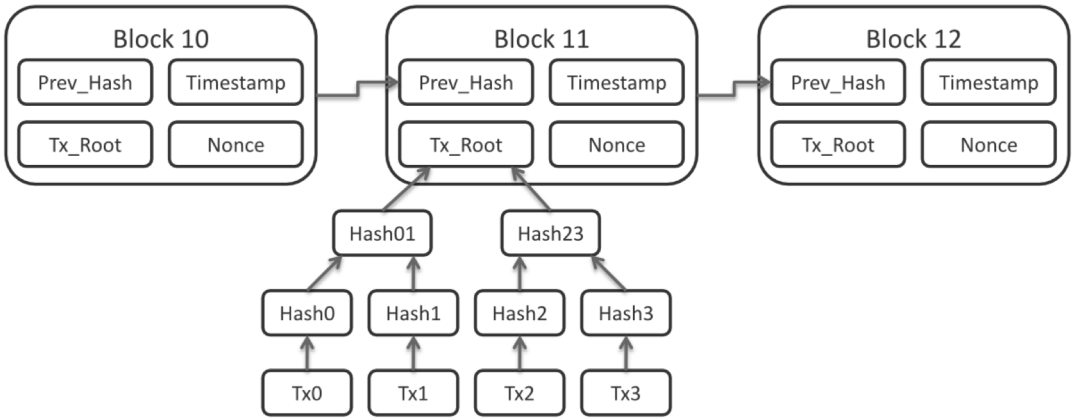
Components:

- Transaction / scripting protocol
  - How transactions are broadcast and stored.
  - How scripts / smart contracts are programmed.
- Consensus algorithm
  - Achieve global consensus on the set of accepted transactions.
- Incentive mechanism
  - How to (economically) encourage active and honest validation.

Example: Bitcoin

A chain of *blocks*, each has a set of transactions and a header with:

- **Hash of the previous block**, a timestamp,
- **Merkle root** of all associated (validated) transactions, and
- A **Proof-of-Work**, i.e., the nonce.



- Proof-of-Work (Consensus): Hash(block_hdr) <= 0x0000xxxxxxxxxxx
  - Cannot be solved efficiently.
  - The only way is exhaustive search, in other words, **mining!**
  - Difficulty (RHS) can be tuned based on history generation rate, s.t., **~10 min per block**.

- Incentive: each block grants miner **block reward (bitcoins)**, and each associated transaction gives (optional) **tips (transaction fees)**.

ARIZONA STATE UNIVERSITY

Limitations of Cryptocurrencies

- However, why are we still not using cryptocurrencies today?
  - **Complaint 1**: Bitcoin transfer is too **slow**!
    - ~10 min per block × 6 confirmations (blocks) = ~ 1 hour settlement.
  - **Complaint 2**: Bitcoin has a **high transaction fee**!
    - Peak fee at around $55 per transaction (to confirm in 6 blocks)[1].
  - **Complaint 3**: Bitcoin **does not scale**!
    - Block size: max 1MB
    - Tx size: ~ 250 Byte
    - 4000 tx / 10 min => 7 tx per sec (tps), **globally**!
    - *Comparison*: VISA supports 45,000 peak tps.

1. https://bitinfocharts.com/comparison/bitcoin-transactionfees.html

Existing Scalability Solutions

- **On-chain solutions:**
  - Increase block size
    - Directly increasing scalability
    - Centralization, less incentive, limited improvement, hard fork
  - Sharding: horizontal partitioning
    - Scalability improvement
    - Expensive cross-shard comm., protocol complexity, lower per-shard security, hard fork
  - Proof-of-Stake (or other lightweight consensus)
    - Low energy footprint/cost, highly scalable, fast txs, negates 51% attacks
    - Monopoly problem (centralization), poor stay poor, hard fork

- **Off-chain solutions:**
  - Segwit: moving bulky signature data to parallel chain
    - Scalability improvement
    - Sidechain security (lack of incentive), protocol complexity, hard fork
  - Pegged sidechains / parallel chains / Plasma (tree of chains)
    - Great scalability improvement, bridging different chains
    - Lower per-chain security, need inter-chain comms.
  - Payment Channel Network (PCN)

The Blockchain Scalability Trilemma

[1]A blockchain system can satisfy at most two of the following three properties:

- **Decentralization**: each participant only has access to O(c) resources.
- **Scalability**: system is able to process Ω(n) > O(c) transactions.
- **Security**: secure against attackers with up to O(n) resources.

Not proved yet!

1. https://github.com/ethereum/wiki/wiki/Sharding-FAQ

Why PCN will prevail?

- Reason 1: PCN is almost totally off-chain.
  - Can circumvent the scalability trilemma to some extent.
  - Eliminates most on-chain operations by taking transactions off-chain.
  - Does not require hard-fork (thus leaving the whole community as a whole).
- Reason 2: PCN has almost the same security as the main chain.
  - Follows the same security assumptions from the main chain.
  - Blockchain used as arbitration to prevent dishonest behaviors.
  - Does not reduce main chain security.
- Reason 3: PCN drastically reduces settlement time and transaction fee.
  - Local settlement, no costly global consensus required.
- Reason 4: PCN can support cross-chain atomic swaps[1].
- Some potential problems:
  - Fund locking, possible centralization (not known yet), always-on requirement.

1. https://lightning.network/

PCN is (Almost) Production-Ready

- Two leading forerunners in the industry
  - Bitcoin Lightning Network[1]:
    - Alpha release in Jan, 2017; currently in Beta.
    - Jan 20, 2018: first known purchase through the Lightning Network
    - Development efforts from multiple different groups
    - Mar 20, 2018: first DDoS attack, taking ~200 nodes offline.
    - Current status[3]: 2111 nodes, 7351 channels, network capacity 18.569 BTC ($178k)

quick, easy, painless, and most importantly: instantaneous and fee-free![2]

  - Ethereum Raiden Network / uRaiden:
    - uRaiden launched on Ethereum mainnet in Nov, 2017.
    - Currently only supports unidirectional channels and single-hop payments.

- Yet it gives rise to new challenges that shall be tackled!
  - Payment Routing
  - Privacy and Security / DoS-resistance
  - Economics

More on these later…

ARIZONA STATE UNIVERSITY

1. https://en.wikipedia.org/wiki/Lightning_Network
2. https://www.cointelligence.com/content/first-purchase-via-bitcoins-lightning-network-just-happened/
3. https://1ml.com/ as of May 3, 2018

Outlines

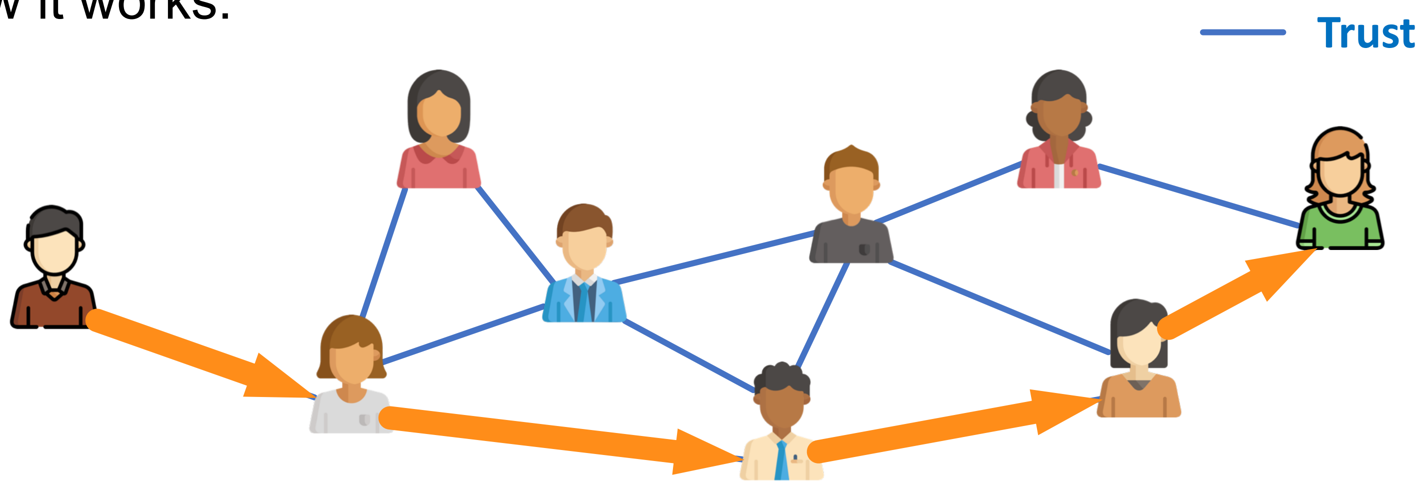
- The “Why”s for cryptocurrency and PCN
- The “What” for PCN
- The “How”s for PCN
- From “W” to “C”: Conclusions

Precursor: Credit Network

- Built upon credit channels among banks and corporations.
  - Originates in economics, extended to make payments w/ blockchain.

ripple stellar

  - How it works:



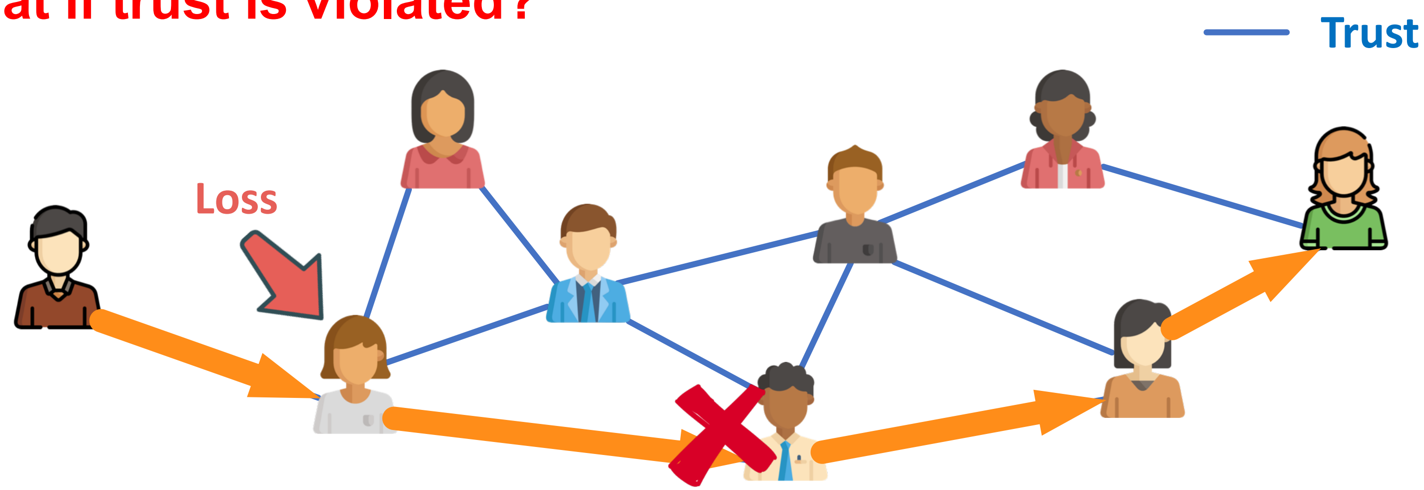
    - Users specify trusted peers and amounts
    - A payment is a path of trust from sender to recipient

Precursor: Credit Network

- Built upon credit channels among banks and corporations.
  - Originates in economics, extended to make payments w/ blockchain.

ripple stellar

  - **What if trust is violated?**



Local loss: one link's default will not spread loss to other nodes.

Removing Trust from CN

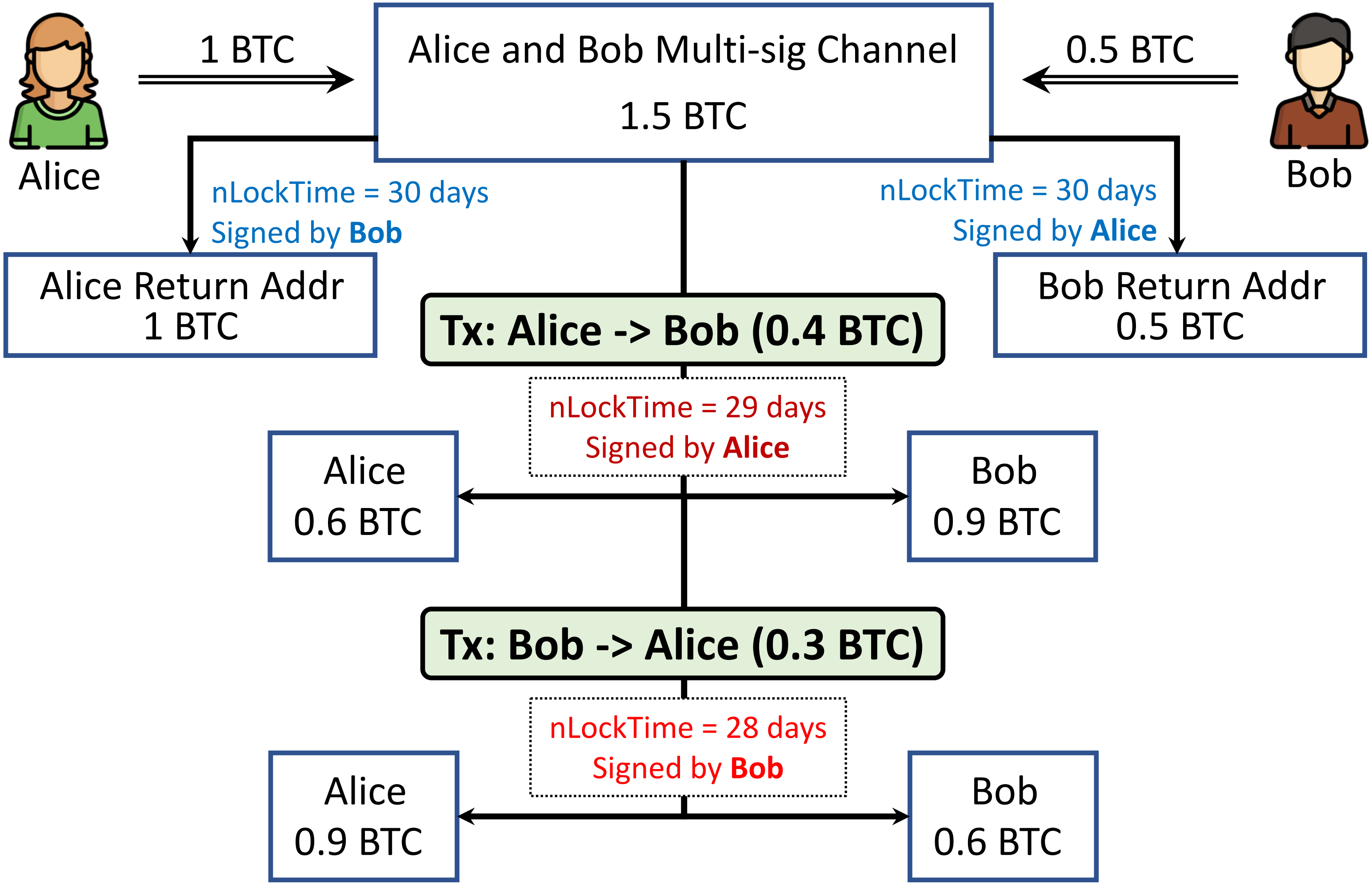
- CN is most suitable for bank-bank or bank-user scenarios.
  - **Low fees, fast settlements**
  - Need of **trust** and resolution of **local losses** **(nothing-at-stake)**
  - **Cannot scale** to global P2P payment scenario

Credit Channel $\xrightarrow{\text{Remove Trust}}$ Payment Channel

- *Locked fund (stake)*
- *Multi-signature smart contracts*
- *Blockchain*

} **Decreasing Time-Locks or Revocable Sequence Maturity Contract (RSMC)**

Payment Channel via Decreasing Time-Lock



Payment Game with Decreasing Time-Lock

- If both Alice and Bob play honestly:
  - Initial funds distributed via on-chain transaction (Channel Opening).
  - Each time of a payment, both parties sign to update balance (generate new Commitment transaction pairs).
  - At/Near time of expiration (smallest nLockTime), both parties publish newest transactions to blockchain (Channel Closing).

- If Bob wants to hack (steal Alice's fund):
  - Bob publishes an old transaction where he has higher fund.
  - Alice sees Bob's misbehavior, and immediately publishes the newest transaction signed by Bob.
  - Since Alice's transaction has earlier nLockTime, it will become valid before Bob's transaction, hence invalidating Bob's transaction.

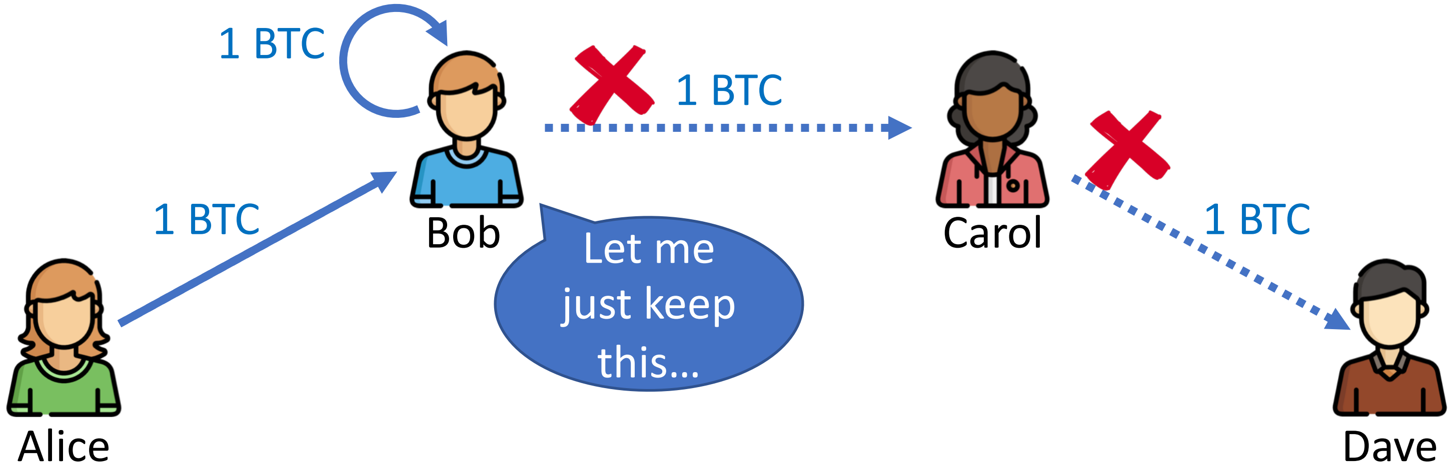
RSMC

- Issue with Decreasing Time-Lock:
  - Each payment decreases channel expiration time.
  - No punishment of misbehavior.

- Revocable Sequence Maturity Contract (RSMC):
  - Each **Commitment** transaction comes with an unsigned **Remedy** transaction that grants all funds to counterparty.
    - Commitment has a sequence requirement of 1000; Remedy has 0.
    - Remedy needs signature of both parties to work.
  - Each new Commitment invalidates previous Commitments by both parties handing signing keys for previous Remedys to the other.
  - When old Commitment is published by one party, it will be invalidated by the other party publishing the corresponding Remedy.
  - Does not reduce channel expiration.
  - Punishment of misbehavior by granting all funds to counterparty.

The Multi-hop Problem & HTLC

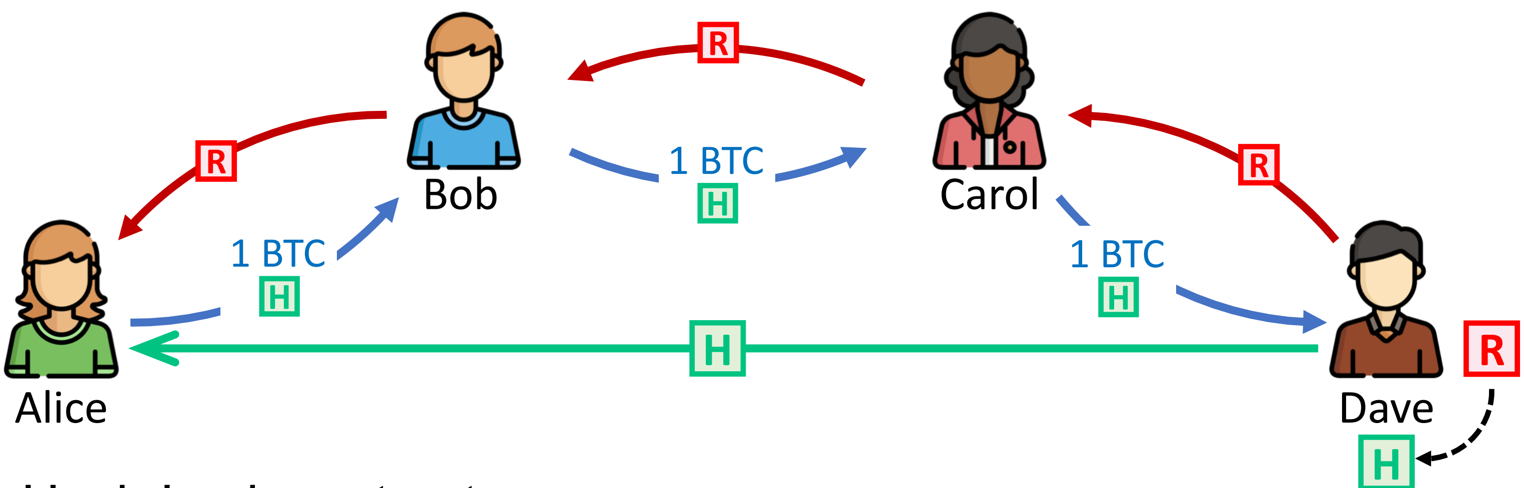
- Trust issue in multi-hop scenario



- Solution: Hash Timelock Contract (HTLC)
  - Hash Lock
  - Time Lock

The Multi-hop Problem & HTLC

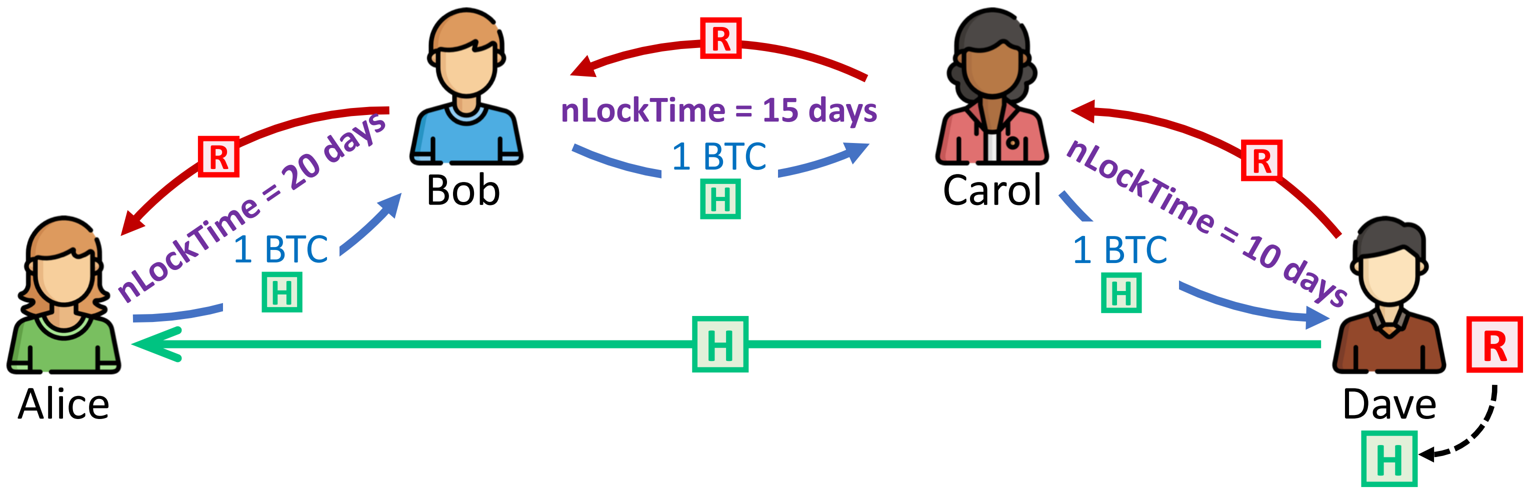
- Trust issue in multi-hop scenario



- Hash Lock contract:
  - ∗ *Each node cannot spend payment without giving R that generates H.*
  1. Dave generates random R and hash H = H(R), and send H to Alice.
  2. Alice sends payment and H, requesting for R; each node forwards.
  3. Dave replies R upon receiving payment; each node forwards.

The Multi-hop Problem & HTLC

- Trust issue in multi-hop scenario



  - **Issue:** Dave can wait until some previous channel to expire.
  - Time Lock contract:
    - Refund w/ decreasing nLockTime per hop, ensuring no defaulters.
    - Not providing R within nLockTime refunds to transferor
  - **HTLC** (Hashed Timelock Contract) **= Hash Lock + Time Lock**

Payment Channel Network

- A network of users and RSMC+HTLC-guaranteed channels.

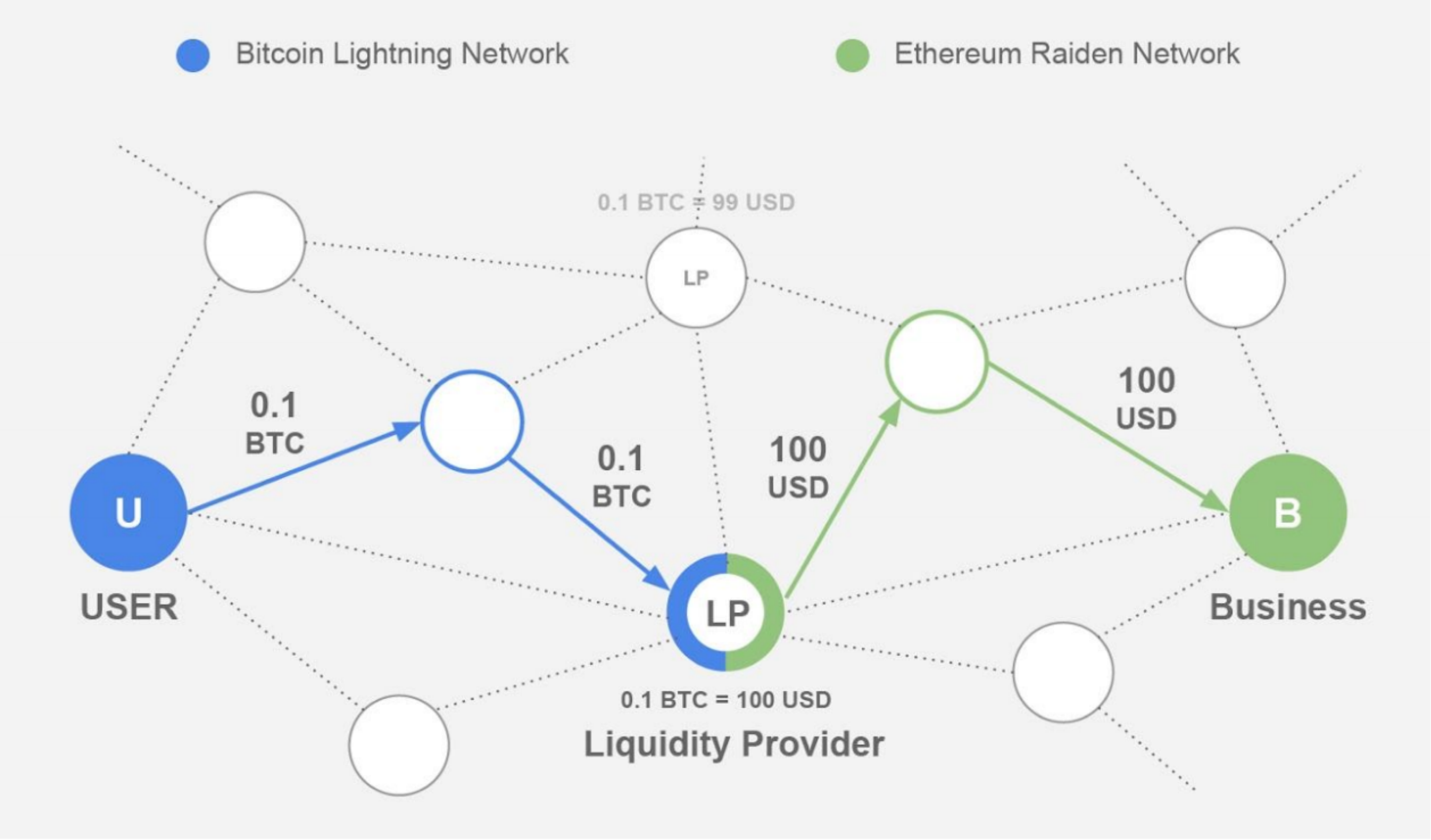


Fig: Hosp, Julian, "Three Technical Requirements to Connect Blockchains Without a Token," https://blog.tenx.tech/three-technical-requirements-to-connect-blockchains-without-a-token-98d693084849

Benefits of PCN

- Risk-free
  - Fund security ensured by crypto protocols / smart contracts.
  - No trust placed on anyone (except for performance issues).
  - (Almost) have the same security as the blockchain itself.
    - No coin loss unless blockchain 51% attacks; DoS.
- Off-chain transactions (blockchain scalability)
  - The only operations involving blockchain are Open, Close and Dispute.
- Fast settlement
  - Local settlement without global confirmations; support for real-time apps.
- Low fees
  - Low cost of transactions; support for *micropayments*.
- Cross-chain/currency compatibility
  - Intermediate nodes play as exchanges; P2P exchanging.

Outlines

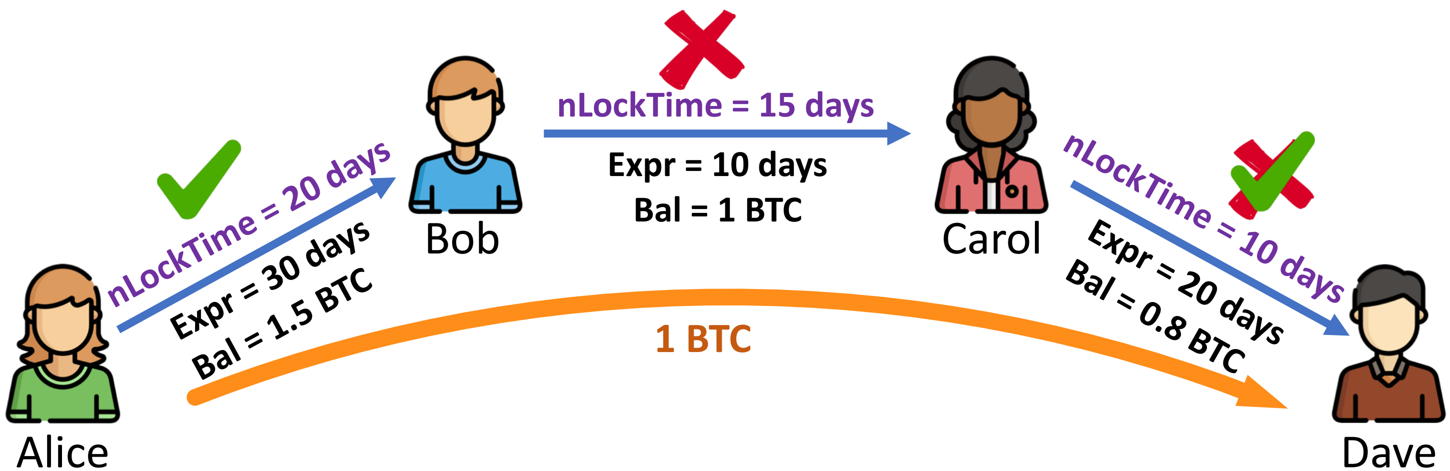
- The "Why"s for cryptocurrency and PCN
- The "What" for PCN
- The "How"s for PCN
- From "W" to "C": Conclusions

PCN Challenges Overview

- PCN is still in its infancy
  - Payment Routing
    - Finding paths for payments
  - Privacy and Security (other than risk-freeness)
    - Privacy-preservation can be harder than blockchain
    - DDoS or routing blockage attacks
  - Economics
    - Incentivization: PCN as an investment vehicle

Problem 1: Routing

- Finding a path/multiple paths from sender to recipient, s.t.:
  - A successful **HTLC** can be established on any path.
    - Meaning the expiration time of each channel needs to be satisfied.
  - Sufficient **balance** presents in the joint of all paths.



- Other requirements:
  - **Real-time**: user-specified payment deadline
  - **Exchange**: go through specific exchange nodes

Formulating the Routing Problem

- For payment (s, t, val, st, dl) in PCN $G = (V, E)$.

$$
\begin{aligned}
\text{find} \quad & (s,t)\text{-path set } P \text{ and balances } v_p \\
\text{s.t.} \quad & \sum\nolimits_{p \in P} v_p \geq val; \\
& \sum\nolimits_{p \in P : e \in p} v_p \leq b_e, \quad \forall e \in E; \\
& \sum\nolimits_{e \in p} d_e \leq dl - st, \quad \forall p \in P; \\
& \sum\nolimits_{e \in p} d_e^1 + \sum\nolimits_{e \in p_\varepsilon^+} d_e^2 \leq expr(\varepsilon) - st, \quad \forall p \in P, \forall \varepsilon \in p.
\end{aligned}
$$

- b_e: channel balance (directional).
- d_e, d_e^1, d_e^2: total, forward and backward delay of a channel.
- p_e^+: downstream segment of path p from edge e.
- $expr(e)$: channel expiration time.

Is Routing Hard?

- **Theory**: the problem is NP-hard if multiple paths allowed.
  - Multi-Path routing with Bandwidth and Delay constraints (MPBD) -- Proved to be NP-hard [Misra2009b]

- **Practice**:
  - **Fully-distributed** algorithm needed
    - No cryptocurrency user would trust any central authority, even for routing!
  - **Dynamic** network environment
    - Each transaction changes channel balances!
    - Unpredictable load across the network!
    - Nodes may join/leave, or go offline/online at any time!
  - **Concurrency** issue
    - Non-blockingness required for simultaneous payments!
  - Goodput, efficiency, reliability, privacy, DoS-resiliency, …

States-of-the-Art Routing

- In practice:
  - Bitcoin Lightning network: BGP-like protocol[1]
    - Non-adaptive, no privacy, best-effort and no concurrency
  - Ethereum Raiden network: best-effort guessing[2]
    - Not exactly routing…

- In development:
  - Max-flow / Push-Relabel [Rohrer2017]
    - High goodput, concurrent
    - High overhead, does not scale, HTLC-agnostic
  - Prefix routing + landmark routing [Moreno-sanchez2015, Malavolta2017a, Roos2018]
    - Privacy-preserving, concurrent
    - Semi-distributed, non-adaptive, limited paths, HTLC-agnostic
  - Hybrid proactive + reactive routing with beacons [Prihodko2016]
    - Best-effort, privacy-agnostic

1. https://bitcoin.stackexchange.com/questions/43687/how-are-paths-found-in-lightning-network
2. http://raiden-network.readthedocs.io/en/stable/spec.html#transfer-routing

A Search-based Routing Algorithm /1

- Ford-Fulkerson augmenting path algorithm

Algorithm 1: Ford-Fulkerson max-flow algorithm [9]

Input: network $G = (V, E)$, source s, destination t
Initialize: start with an empty flow f and $G^f = G$

repeat
  Find (s, t)-path p in G^f with positive balance f_p;
  Add p to f, and update G^f;
until *no augmenting* (s, t)*-path can be found*;
return f.

- **Issue**:
  - Not distributed.
  - Augmenting path is delay-agnostic.
  - Does not support multiple simultaneous routing requests.

A Search-based Routing Algorithm /2

- Ford-Fulkerson augmenting path algorithm
- **Issues**:
  - Not distributed.
  - Augmenting path is delay-agnostic.
  - Does not support multiple simultaneous routing requests.
- **Solutions**:
  - Distributed BFS for augmenting path finding.
  - Delay-feasible augmenting path only.
  - Probe-and-Reservation: balance reservation and locking at the time of routing.

A Search-based Routing Algorithm /3

Algorithm 1: CoinExpress: Algorithm Overview

```
1 Initialize empty flow f and residual graph G^f = G;
2 while b(f) < a do
3     Sender: for each neighbor channel e, send probe (R, β, δ, p) where
              β = min{val, b_e^f}, δ = d_e^1, p = (e);
4     Intm.: upon probe, update and send to each neighbor e where
              β = min{β, b_e^f}, δ = δ + d_e^1, p = p + (e);
5     Recip.: select probe with max β and send back conf (R, β, δ, p);
6     Intm.: upon conf, find next hop e and last hop e_last in p, first let
       δ = δ + d_e^2, then check: 1) b_e^f ≥ β, and 2) δ ≤ min{expr(e), dl} − st;
          if both checks pass then reserve β on e, and send conf to e_last;
          else reply cancel along p to cancel all reservations on p;
7     Sender: upon conf, record path p and β and update f and G_e^f;
8     Recip.: upon cancel, select a new probe and repeat from Line 5;
```

Forward balance probing phase (lines 3–4)

Backward checking and balance reservation phase (lines 5–7)

Cancel and retry (line 8)

A Search-based Routing Algorithm /4

- Residual flow update
  - If there is a single flow:

$$b^f_{u,v} = b_{u,v} - f(u,v) + f(v,u)$$

  - **Concurrency issue**: another flow may *steal* the reserved flow.
    - If f(v,u) > 0, another flow along (u,v) may use it, which is not guaranteed if later on the current flow cancels f(v,u) via another augmenting path.

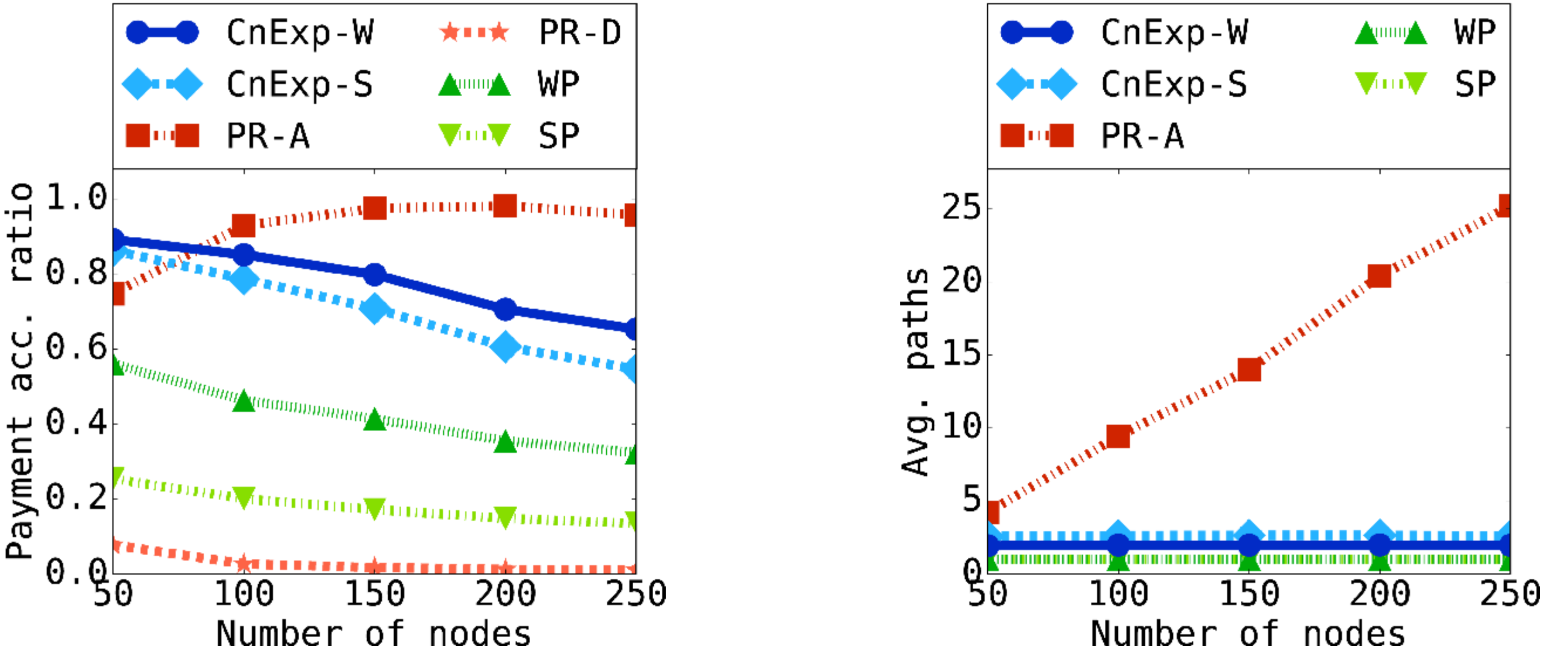
  - **Balance locking**: each node keeps per-flow state $f_R(u,v)$.

$$b^f_{u,v}(R) = b_{u,v} - \sum_{R'} f_{R'}(u,v) + f_R(v,u)$$

    - Each node can only use its own residual flow on the reverse direction.

A Search-based Routing Algorithm /5

- Some simulation results



CnExp-W: CoinExpress with widest path selection

CnExp-S: CoinExpress with shortest path selection

PR-D: Push-Relabel with delay-based path pruning [Rohrer2017]

PR-A: Push-Relabel *without delay* (infeasible paths) [Rohrer2017]

WP: Single widest path | **SP**: Single shortest path

Some Other Good Directions on Routing

- QoS routing
  - **Similarities**: time & bandwidth constraints
  - *Existing work*: approximation [Xue2008, Misra2009b], distributed [Chen1999]
  - Challenges: adaptivity, concurrency, QoS privacy

- Routing in WSN/MANET, P2P routing
  - **Similarities**: distributed & dynamic
  - *Existing work*: reactive [Johnson1996, Perkins2003], proactive [Rowstron2001], opportunistic [Biswas2005]
  - Challenges: balance adaptivity, QoS, concurrency, privacy

- Bandwidth provisioning / traffic steering
  - **Similarities**: bandwidth sharing and guarantee
  - *Existing work*: centralized algorithms [Duan2003]
  - Challenges: distributed and adaptive algorithm design, QoS, privacy

Problem 2: Privacy and Security

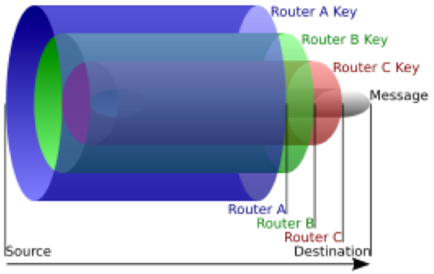
- Sensitive information:
  - **Identities**: sender, recipient
  - **Locations**: sender, recipient, intermediate (path)
  - **Relations**: sender-recipient, sender/recipient-transaction,
  - **Content**: value, start / deadline
  - **States / Side-channels**: balance, load / queuing delay, path

- Is protecting privacy hard?
  - Much of the information is needed in the payment process
    - Value, balance, path (next-hops)
  - Compared to on-chain solutions:
    - On-chain: protects source/target/amount, but not time [Ben-Sasson2014]; incurs global overhead (discouraging verification, lowers overall security)
    - PCN: network structural exposes more information; local overhead

ARIZONA STATE UNIVERSITY

Possible Approaches: Routing

- Onion Routing [Osuntokun2017]
  - Layered encryption that reveals only next hop at each node.
  - Long studied, well adopted, but vulnerable to certain attacks.
    - GPA: global passive adversary
    - Byzantine: arbitrary subset of malicious nodes



- Mix-Nets
  - Mixing nodes permute groups of messages before forwarding.
  - Protects against GPA and Byzantine;
  - Large overhead, long latency.
    - Due to the need for waiting or generation for mix messages.
    - Verifiable permutation.

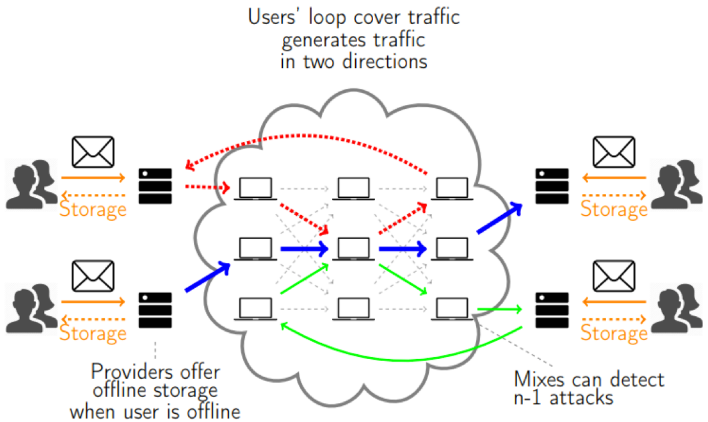


Fig 1: Wikipedia, https://en.wikipedia.org/wiki/Onion_routing
Fig 2: A. M. Piotrowska, J. Hayes, T. Elahi, K. U. Leuven, S. Meiser, G. Danezis, A. M. Piotrowska, J. Hayes, S. Meiser, and G. Danezis, "The Loopix Anonymity System," in *Proc. USENIX Security*, 2017.

Possible Approaches: Payment

- Multi-hop HTLC [Malavolta2017]
  - Sender-receiver anonymity, (off-path) value privacy
  - **Negative result**: trade-off between concurrency & privacy
    - Not really, if we can solve concurrency through routing!
    - Similar to Onion Routing and Sphinx [Danezis2009]: once we obtain a circuit, anonymous communications become easy…

- More efforts needed to provide better privacy:
  - GPA / Byzantine
  - Sender, recipient
  - On-path value
  - Time
  - …

PCN Security

- PCN security assumptions:
  - Blockchain is secure and accessible (for dispute)
  - Local node is securely functional (secure storage and computation)

- Possible security breaches:
  - Any attack that applies to the blockchain itself
    - 51% attacks, large-scale routing attacks (network partition), DoS, …
  - Network attacks
    - Blockchain accessibility: blocks disputing
    - Blocking communication between users / DoS: cause loss to honest users
    - Breaching network traffic security

- Possible solutions:
  - Secure & anonymous communications between nodes
  - Reliable network traffic routing
  - Group paying: multi-party channels
    - As long as one node is live, the payment goes on
    - Requires intensive work on multi-party smart contracts and overhead

Problem 3: The Economics Perspective

- Why do people use PCN?
  - I want fast payment from/to someone in the network…
  - I want to invest and expand my retirement account…
- In Bitcoin/Ethereum/…, if you want to invest:
  - Coin speculation… you may be leek-cut (割韭菜)
  - Run a miner node and collect tips/gas/…
- In PCN:
  - Open up a channel with some congested node and put your money.
    - Or you can open up multiple to bridge multiple congested nodes.
  - Wait until channel expires, then collect your fees.
  - A light client is sufficient.

More on PCN Investment

- A fairly **risk-free** investment
  - Fully self-involved.
  - Your fund is safely protected by crypto (and your network)!
  - You need minimal resources other than your investment
    - An all-time running PC, a reliable network, and a few megabytes
    - *Bitcoin miner node: expensive GPU/ASIC, 167 GB space (growing)*
  - No risk of market manipulation and/or bank bankruptcy.

- A few notes for possible investors
  - Secure your wallet ☺
  - Keep the machine and network running at all times

How PCN Economics Work?

- Perspective 1: strategic investment
  - Based on loads, node decides investment strategy
    - Select channel peers that yield the best gains
    - Best allocation of investment among channels
    - Normal node / Exchange node
    - Investment based on empirical data / past returns
    - Group investment

- Perspective 2: incentive mechanism
  - **User strategy**: decide values and select routes with minimum fees.
  - **Node strategy**: decide fees and select requests with maximum gains.
  - Possible models:
    - *Stochastic game*: user demands are unknown
    - *Stackelberg game*: network decides mechanism, user follows
    - *Auction*: single/double auction, user selection and payment decision

Outlines

- The “Why”s for cryptocurrency and PCN
- The “What” for PCN
- The “How”s for PCN
- From “W” to “C”: Conclusions

Will cryptocurrencies/PCN survive?

- We've heard a lot of buzzes.
  - Bitcoin is a hype.
  - Too much bubble.
  - Mining wastes energy.
  - There is no value in Bitcoin.
  - They won't work when quantum computer comes.
  - …
  - Lightning network will not work.
  - It will become centralized.
  - No one would open a channel.
  - Routing is hard.
  - Why not use XRP/RSK/sidechain…
  - When the bubble comes down.
  - …
- But, they solve real-world problems!
  - Centralization / manipulability.
  - Inflation.
  - Traceability.
  - Insecurity.
  - Fast and cheap micropayments.
  - Blockchain scalability.
  - Inter-currency exchange.
- Blooming research and development efforts.
  - Blockchain on Google Scholar:

2015	2016	2017
1000+	3000+	8000+

Conclusions

- Why we need PCN?
  - Blockchain scalability, high fee, high settlement latency.
  - Existing solutions compromises security for scalability.

- What is PCN?
  - Network of smart contract-based trustless payment channels.
  - Security ensured by cryptographic methods.
    - Almost the same level of security as blockchain itself.

- How PCN could evolve?
  - Distributed adaptive routing.
  - Privacy preserving routing and payment.
  - Economics to encourage participation / increase performance.

- A lot of interesting and challenges problems ahead!

Thank you very much!

Q&A?

References /1

- [Ben-Sasson2014] E. Ben Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, “Zerocash: Decentralized Anonymous Payments from Bitcoin,” in *Proc. IEEE S&P*, 2014, pp. 459–474.
- [Biswas2005] S. Z. Biswas, “Opportunistic Routing in Multi-Hop Wireless Networks,” *MSc Thesis*, 2005.
- [Chen1999] S. Chen and K. Nahrstedt, “Distributed quality-of-service routing in ad hoc networks,” *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1488–1505, 1999.
- [Danezis2009] G. Danezis and I. Goldberg, “Sphinx: A Compact and Provably Secure Mix Format,” in *Proc. IEEE S&P*, 2009, pp. 269–282.
- [Duan2003] Z. Duan, Z.-L. Zhang, and Y. T. Hou, “Service overlay networks: slas, qos, and bandwidth provisioning,” *IEEE/ACM Trans. Netw.*, vol. 11, no. 6, pp. 870–883, Dec. 2003.
- [Johnson1996] D. B. Johnson and D. A. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Networks,” *Mob. Comput.*, vol. 353, pp. 153–181, 1996.
- [Malavolta2017] G. Malavolta, P. Moreno-Sanchez, A. Kate, M. Maffei, and S. Ravi, “Concurrency and Privacy with Payment-Channel Networks,” in *Proc. ACM CCS*, 2017, pp. 455–471.
- [Malavolta2017a] G. Malavolta, P. Moreno-Sanchez, A. Kate, and M. Maffei, “SilentWhispers: Enforcing Security and Privacy in Decentralized Credit Networks,” in *Proc. ISOC NDSS*, 2017.
- [Misra2009b] S. Misra, G. Xue, and D. Yang, “Polynomial Time Approximations for Multi-Path Routing with Bandwidth and Delay Constraints,” in *Proc. IEEE INFOCOM*, 2009, pp. 558–566.
- [Moreno-sanchez2015] P. Moreno-Sanchez, A. Kate, M. Maffei, and K. Pecina, “Privacy Preserving Payments in Credit Networks: Enabling trust with privacy in online marketplaces,” in *Proc. ISOC NDSS*, 2015, pp. 8–11.

References /2

- [Osuntokun2017] L. Osuntokun, "Security Analysis of the Lightning Network." 2017.
- [Perkins2003] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," *IETF RFC 3561*, 2003.
- [Prihodko2016] P. Prihodko, S. Zhigulin, M. Sahno, and A. Ostrovskiy, "Flare: An Approach to Routing in Lightning Network (White Paper)," 2016.
- [Rohrer2017] E. Rohrer, J.-F. Laß, and F. Tschorsch, "Towards a Concurrent and Distributed Route Selection for Payment Channel Networks," in *Proc. of Cryptocurrencies and Blockchain Technology (CBT)*, 2017, pp. 411–419.
- [Roos2018] S. Roos, P. Moreno-Sanchez, A. Kate, and I. Goldberg, "Settling Payments Fast and Private: Efficient Decentralized Routing for Path-Based Transactions," in *Proc. ISOC NDSS*, 2018.
- [Rowstron2001] A. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," in *Proc. IFIP/ACM Middleware*, 2001, pp. 329–350.
- [Xue2007] G. Xue, A. Sen, W. Zhang, J. Tang, and K. Thulasiraman, "Finding a Path Subject to Many Additive QoS Constraints," *IEEE/ACM Trans. Netw.*, vol. 15, no. 1, pp. 201–211, Feb. 2007.
- [Xue2008] G. Xue, W. Zhang, J. Tang, and K. Thulasiraman, "Polynomial Time Approximation Algorithms for Multi-Constrained QoS Routing," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 656–669, Jun. 2008.
- [Yu2018] R. Yu, G. Xue, V. T. Kilari, D. Yang, and J. Tang, "CoinExpress: A Fast Payment Routing Mechanism in Blockchain-based Payment Channel Networks," to appear in *IEEE ICCCN*, 2018.