HushRelay: A Privacy-Preserving, Efficient, and Scalable Routing Algorithm for Off-Chain Payments

Subhra Mazumdar∗, Sushmita Ruj†, Ram Govind Singh‡ and Arindam Pal†, ◦

∗ Indian Statistical Institute Kolkata, India,
◦Cyber Security CRC, Sydney, New South Wales, Australia,
† CSIRO, Data61, Australia,
‡ ICERT, Ministry of Electronics and Information Technology, India

Bitcoin Blockchain Scalability Problem

- Bitcoin - too slow and too expensive!
- Scalability limited by 2 main factors -
  - *Average Block Creation Time* - 10 minutes to create and secure a block containing around 2000+ Bitcoin transactions.
  - *Block Size Limits* - Every block has a limit of 1MB (1,000 KB). Limit imposed to prvenet DoS attack.

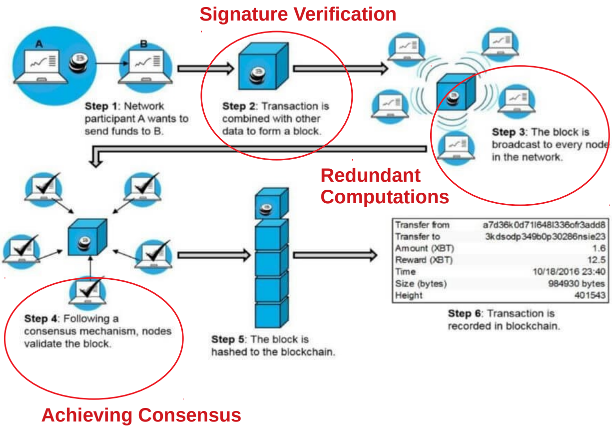


Photo Reference Link

Comparison Bitcoin Vs Conventional Payment Networks

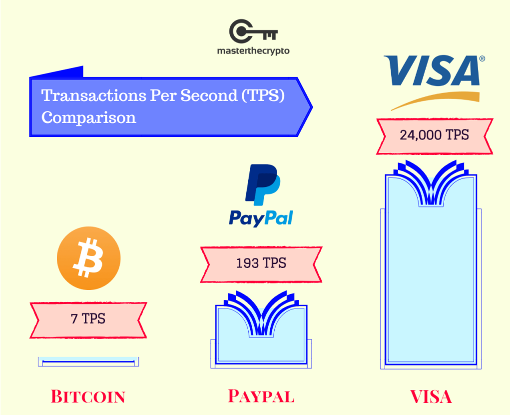


Figure: Processing speeds of Bitcoin compared to other centralized systems.

Photo Reference Link

Layer 2 Solution, Off-Chain Transactions

Transactions are 'off-loaded' from the main blockchain to save space and reduce network congestion.

- Payment Channels
- State Channels

Various State Channel Projects

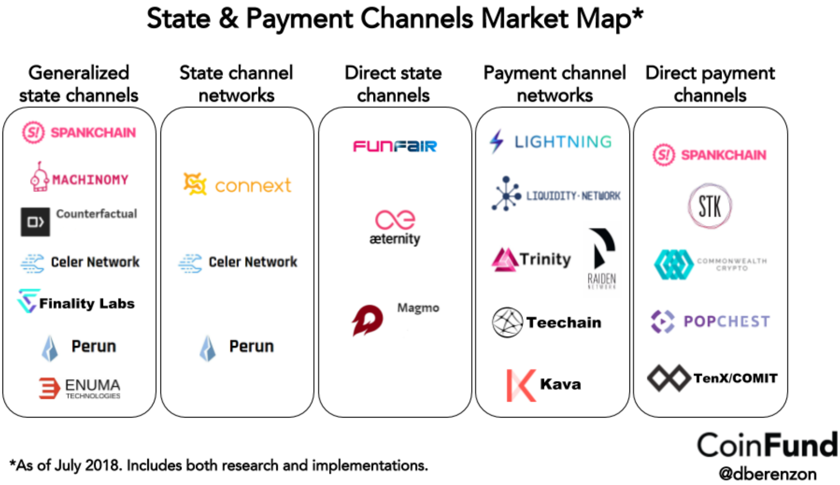
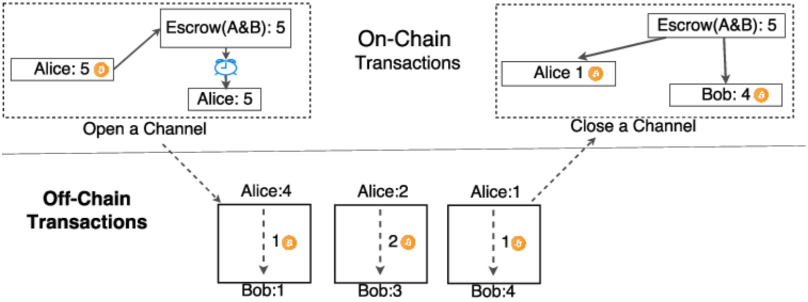


Photo Reference Link

Off-Chain Payments

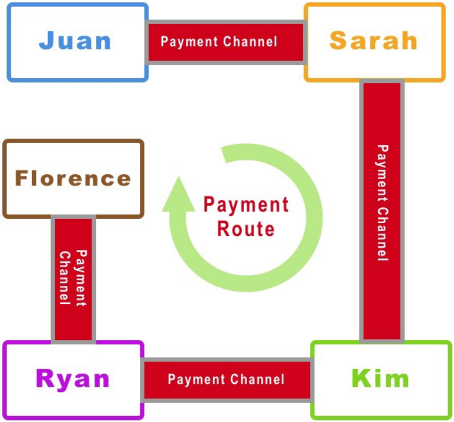
Photo Reference Link



Avoids recording all the transactions, except opening and closing of channel, in the Blockhain.

Payment Network Example

Photo Reference Link



Problems that needs to be addressed in such Payment Networks:

- Routing
- Payment

There are several Payment Algorithms: ensures privacy of payer and payee, hides the payment value. Eg. HTLC [11], Multihop HTLC [8], Anonymous Multihop Lock [9] etc.

Here we deal with just Routing in PCN

Routing in PCN

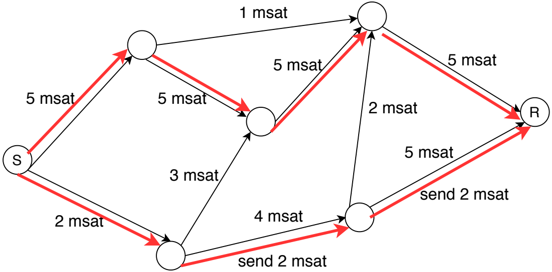


Figure: Transfer 7 msat from S to R

Hence splitting is inevitable.

Challenges faced

This is not like a conventional Routing Algorithm as S does not have sufficient information.

Since only opening of channel gets recorded on-chain, this is the information S has:

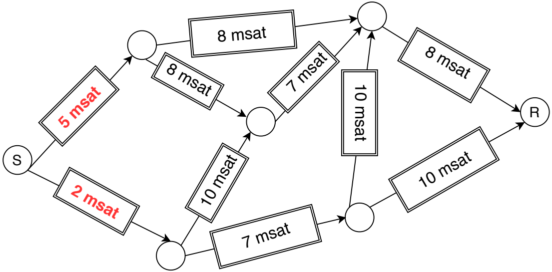
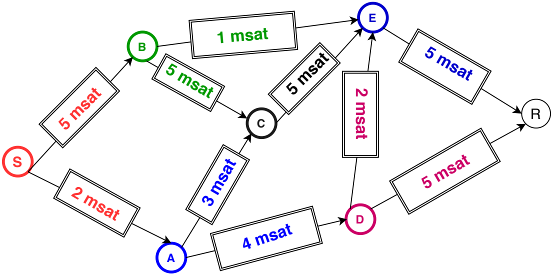


Figure: Except for its outgoing channel, S just knows the opening balance of rest of the channels

Each node has information of the residual capacity of their respective outgoing channel.



Inferences made

- Any routing algorithm designed for PCN must be *Decentralized*.
- Individual nodes take decision based on the information received from its neighbourhood.

Related Works

Landmark Based Routing: SpeedyMurmur [3]

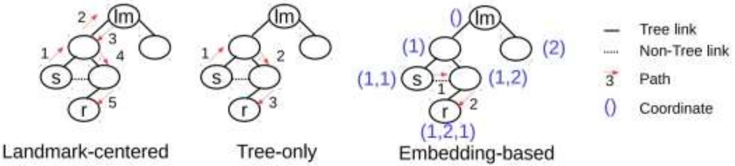


Fig. 1: Examples of different spanning tree routing schemes for landmark lm, sender s, receiver r.

Disadvantage

Wrong decision while deciding split!

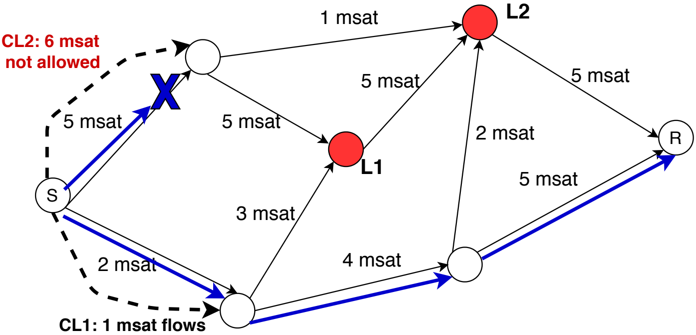


Figure: Transfer 7 msat from S to R: $c_{L1} = 1\ msat$ and $c_{L2} = 6\ msat$

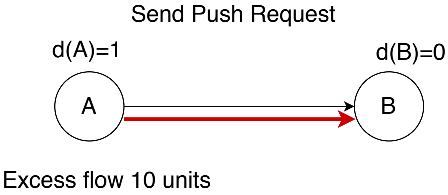
Our proposed solution: HushRelay

- Distributed in nature
- Allows optimal utilization of the available capacities present across multiple paths.
- Efficient and Privacy-Preserving

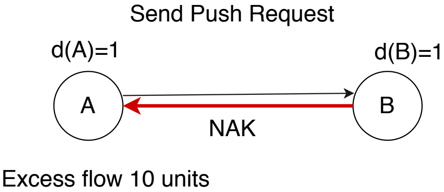
HushRelay: Basic Operations

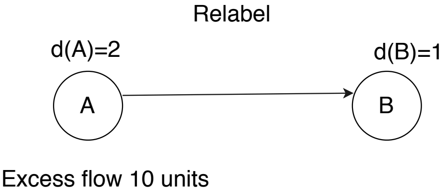
All the nodes acting as individual processing unit in parallel.

1) Try to push excess flow, Sending a Push Request



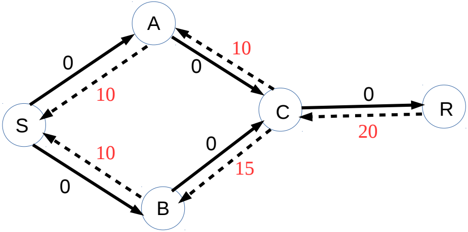
2) Incase of Negative Acknowledgement, Relabel





HushRelay: Example

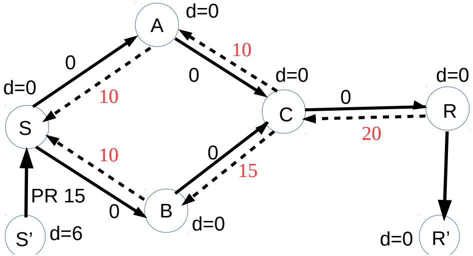
Given the PCN, transfer 15 units from S to R



(a) Initial state

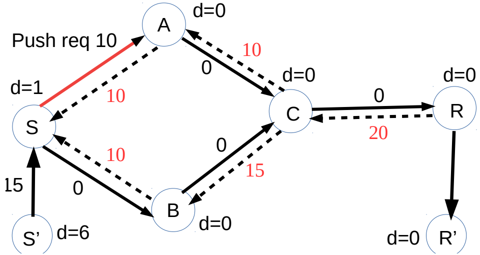
HushRelay: Initialization Phase

Dummy vertices S' and R' is added to the network with edges (S', S) and (R, R'). The edge capacities are as follows : $c(S', S) = 15$ and $c(R, R') = 15$. Each nodes is assigned a label of 0 except dummy vertex S'.



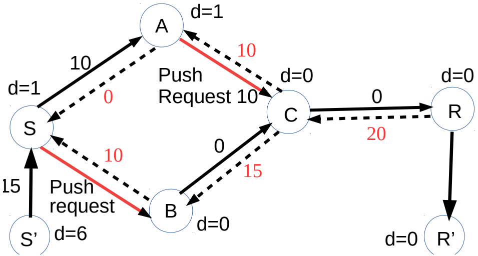
(b) Push request of 15 from S'

15 units pushed to S and d(S) set to 1. S sends push request to A.



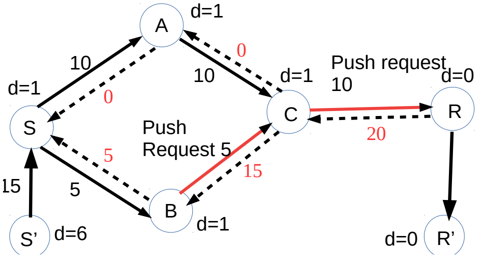
(c)

10 units pushed to A, set d(A)=1. Push request of 5 units to B by S and push request of 10 units to C by A



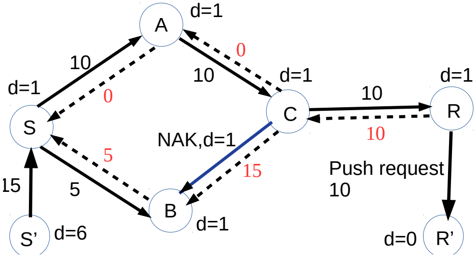
(d)

5 units pushed to B, 10 units pushed to C. Set d(C)=1 and d(B)=1. Push request of 5 units to C by B and Push request of 10 units to R by C.



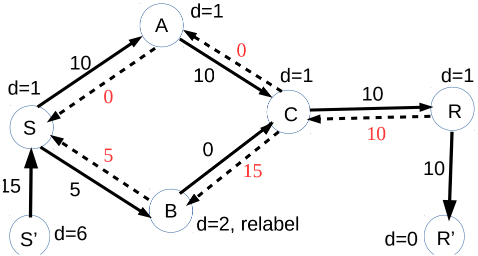
(e)

Negative Acknowledgement by C to B since both have same label, d(C)=d(B)=1. C cannot accept a request and generate a request at the same time. B pauses. C pushes 10 units to R. Set d(R)=1. R generates a push request for R'.



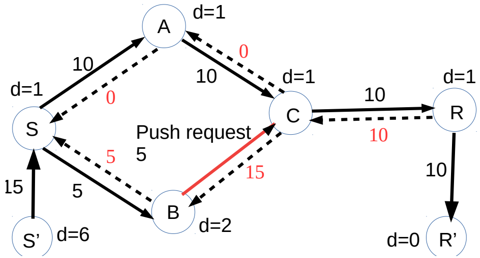
(f)

B performs relabel operation. Sets d=2 (max label of its neighbours + 1). R' accepts push request



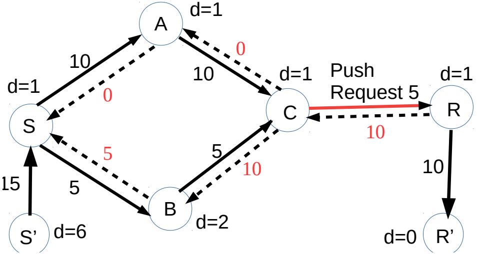
(g)

B sends a push request to C.



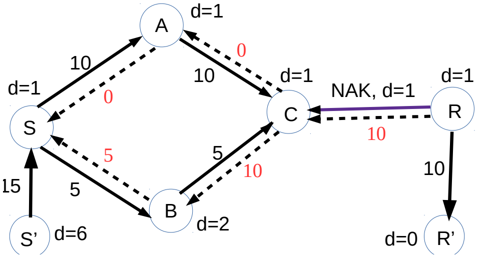
(h)

C accepts the push request. It won't be able to send push request to R as d(C)=d(R)=1.



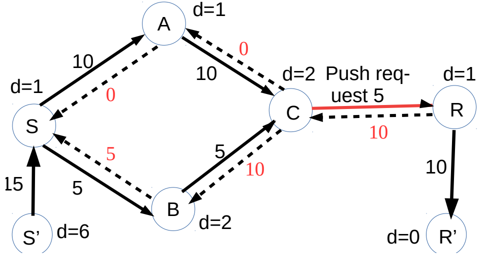
(i)

Negative Acknowledgement received from R. Hence C performs a relabel operation.



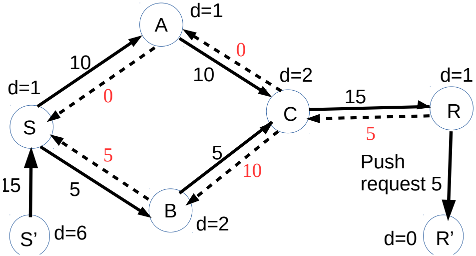
(j)

It sends push request of 5 units to R.



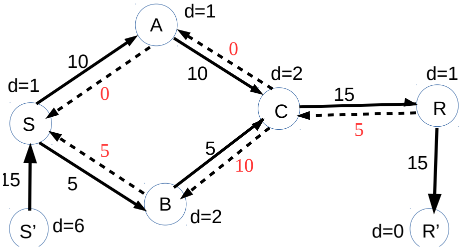
(k)

R accepts the push request, generate a push request of 5 units for R'.



(l)

Final State: Funds transferred to R



(m)

Complexity Analysis

n :Number of nodes in PCN, m: Number of payment channels

Under Asynchronous Implementation:
Message Complexity : $O(n^2 m)$
Runtime Complexity: $O(n^2)$

Analysis of *HushRelay* and *SpeedyMurmur*

HushRelay Source Code Link [1]

Table: SpeedyMurmur vs HushRelay - Performance Analysis on Real Instances

Network/Algorithm	SpeedyMurmur								HushRelay	
	Success Ratio Number of Landmarks				Time taken Number of Landmarks				Success Ratio	Time taken
	1	2	4	6	1	2	4	6		
Ripple Network	0.38	0.69	0.92	0.98	1.66s	2.2s	3.23s	4.74s	1	2.4s
Lightning Network	0.42	0.64	0.83	0.91	0.61s	0.69s	0.83s	1.94s	0.99	0.15s

On Simulated Instances

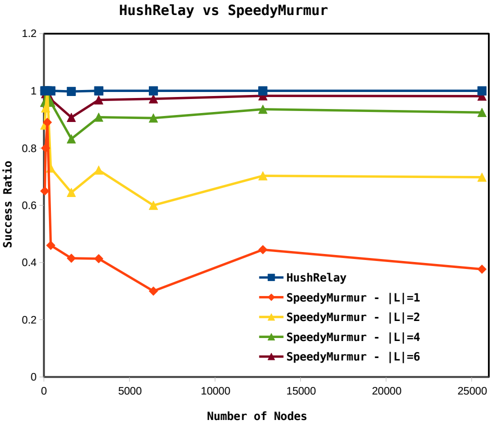


Figure: Success Ratio vs Number of Nodes

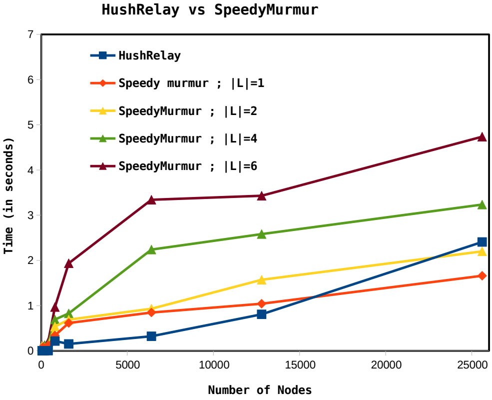


Figure: Time To Route vs Number of Nodes

References I

- "Hushrelay," https://www.dropbox.com/sh/x9pngj005dxh87b/AAAJNt-WquV0JZTspnijEXNVa?dl=0, 2019.
- "Source code : Speedymurmurs: Fast and private path-based transactions," https://crysp.uwaterloo.ca/software/speedymurmurs/, Nov 25, 2017.
- S. Roos, P. Moreno-Sanchez, A. Kate, and I. Goldberg, "Settling payments fast and private: Efficient decentralized routing for path-based transactions," in *Network and Distributed System Security Symposium*, 2018.
- C. Decker, R. Russell, and O. Osuntokun, "eltoo: A simple layer2 protocol for bitcoin," *White paper: https://blockstream. com/eltoo. pdf*, 2018.

References II

C. Decker and R. Wattenhofer, "A fast and scalable payment network with bitcoin duplex micropayment channels," in *Symposium on Self-Stabilizing Systems*. Springer, 2015, pp. 3–18.

L. Gudgeon, P. Moreno-Sanchez, S. Roos, P. McCorry, and A. Gervais, "Sok: Off the chain transactions," *IACR Cryptology ePrint Archive*, vol. 2019, p. 360, 2019.

G. Malavolta, P. Moreno-Sanchez, A. Kate, and M. Maffei, "Silentwhispers: Enforcing security and privacy in decentralized credit networks." in *Network and Distributed System Security Symposium*, 2017.

G. Malavolta, P. Moreno-Sanchez, A. Kate, M. Maffei, and S. Ravi, "Concurrency and privacy with payment-channel networks," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 455–471.

References III

- G. Malavolta, P. Moreno-Sanchez, C. Schneidewind, A. Kate, and M. Maffei, "Multi-hop locks for secure, privacy-preserving and interoperable payment-channel networks," in *Network and Distributed System Security Symposium*, 2019.
- T. L. Pham, I. Lavallee, M. Bui, and S. H. Do, "A distributed algorithm for the maximum flow problem," in *The 4th International Symposium on Parallel and Distributed Computing (ISPDC'05)*. IEEE, 2005, pp. 131–138.
- J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," *See https://lightning. network/lightning-network-paper. pdf*, 2016.
- S. Roos, M. Beck, and T. Strufe, "Voute-virtual overlays using tree embeddings," *arXiv preprint arXiv:1601.06119*, 2016.

Thank You

Any Questions?