Topological Analysis of Bitcoin's Lightning Network

István András Seres[1][0000−0003−0143−4057], László Gulyás[1], Dániel A. Nagy[1], and Péter Burcsi[1][0000−0003−3306−6500]

Eötvös Loránd University, Budapest 1053, HUN
istvanseres@caesar.elte.hu,gulyahps.elte.hu,nagy.da@gmail.com,bupe@inf.elte.hu

Abstract. Bitcoin's Lightning Network (LN) is a scalability solution for Bitcoin allowing transactions to be issued with negligible fees and settled instantly at scale. In order to use LN, funds need to be locked in payment channels on the Bitcoin blockchain (Layer-1) for subsequent use in LN (Layer-2). LN is comprised of many payment channels forming a payment channel network. LN's promise is that relatively few payment channels already enable anyone to efficiently, securely and privately route payments across the whole network. In this paper, we quantify the structural properties of LN and argue that LN's current topological properties can be ameliorated in order to improve the security of LN, enabling it to reach its true potential.

Keywords: Bitcoin · Lightning Network · Network Security · Network Topology · Payment Channel Network · Network Robustness.

Since its launch, Bitcoin [7] gained a huge popularity due to its publicly verifiable, decentralized, permissionless and censorship-resistant nature. This tremendous popularity and increasing interest in Bitcoin pushed its network's throughput to its limits. Without further advancements, the Bitcoin network can only settle 7 transactions per second (tps), while mainstream centralized payment providers such as Visa and Mastercard can process approximately 40,000 tps in peak hours. Moreover one might need to pay large transaction fees on the Bitcoin network, while also need to wait 6 new blocks (∼ 1 hour) to be published in order to be certain enough that the transaction is included in the blockchain.

To alleviate these scalability issues the Lightning Network (LN) was designed in 2016 [8], and launched in 2018, January. The main insight of LN is that transactions can be issued off-blockchain in a trust-minimized manner achieving instant transaction confirmation times with negligible fees, whilst retaining the security of the underlying blockchain. Bidirectional payment channels can be formed on-chain using a construction called Hashed Timelock Contracts (HTLC). Later several payments can take place in a payment channel. The main advantage of payment channels is that one can send and receive thousands of payments with essentially only 2 on-chain transations: the opening and closing channel transations.

Using these payment channels as building blocks one might establish a payment channel network, where it is not necessary to have direct payment channels between nodes to transact with each other, but they could simply route their payments through other nodes' payment channels. Such a network can be built, because LN achieves payments to be made without any counterparty risk, however efficient and privacy-preserving payment routing remains a challenging algorithmic task [9].

Our contributions. We empirically measure[1] and describe LN's topology and show how robust it is against both random failures and targeted attacks. These findings suggest that LN's topology can be ameliorated in order to achieve its true potential.

1   Background on Lightning Network

In this section we provide a short recap on how LN works. In the following we will use the terms Layer-2 and off-chain interchangeably. LN is a so-called Layer-2 technology, which allows participants issuing transactions without sending a Layer-1 transaction on the Bitcoin parent chain. All parties cooperatively open a channel by locking collateral on the blockchain. The funds can only be released by unanimous agreement or through a pre-defined refund condition [3]. One of the greatest challenge of Layer-2 technologies is to solve how participants can agree on new state updates in a trustless or trust-minimized manner.

Let's take the following toy example: Alice and Bob creates by a single Layer-1 transaction a payment channel with initial balances 10₿ and 0₿ respectively. Straightaway Alice can issue off-chain transactions to Bob up to 10₿. Let's assume Alice issued 3 off-chain transactions to Bob each worth of 1₿. Afterwards Alice's and Bob's balance should be 7₿ and 3₿ and neither Alice, nor Bob should be able to redeem previous balances on the parent chain. LN achieves this by a replace by revocation mechanism, namely both parties collectively authorize a new state before revoking the previous state. Upon dispute, the blockchain provides a time period for parties to prove that the published state is a revoked state [3]. Revoking old channel states is achieved by the exchange of revocation secrets. These secrets, hash preimages, are needed to be retained during the channel's lifetime. A penalty mechanism discourages parties from broadcasting older states. If one party broadcasts a revoked state, the blockchain accepts within a time-window proofs of maleficence from the other party. A successful dispute allots the winning party *all* coins of the channel. In our example if Alice broadcasts a revoked channel state with balances 8₿ and 2₿, Bob can prove, that Alice maliciously broadcasted a revoked state. The penalty mechanism grants all the 10₿ to Bob.

The great insight of LN, is that if now Alice would like to issue a payment to Cecily, who eventually has already established a payment channel to Bob, then Alice does not need to open a payment channel and create a costly on-chain

---

[1] https://github.com/seresistvanandras/LNTopology

transaction, rather she can route her payment through her payment channel with Bob to Cecily. However the maximum amount of bitcoins Alice can send to Cecily is the minimum of all the individual balances on the payment route from Alice to Cecily. Hashed time-locked contracts (HTLC) enable routed payments to be atomic. For a technical description of HTLCs and multi-hop payments the astute reader is referred to [8].

In the following we will model LN as an undirected, weighted graph, where nodes are entities who can issue payments using payment channels which are the edges of the LN graph. The weight on the edges, capacities, are the sum of individual balances. Note, that in most cases individual balances are not known to outsiders. Only the capacity of a payment channel is public information, however one can effectively assess individual balances with handy algorithms [4].

2   Lightning Network's Topology

LN can be described as a weighted graph $G = (V, E)$, where V is the set of LN nodes and E is the set of bidirectional payment channels between these nodes. We took a snapshot[2] of LN's topology on the 10th birthday of Bitcoin, 2019 January 3rd. In the following we are going to analyze this dataset. Although the number of nodes and payment channels are permanently changing in LN, we observed through several snapshots that the main topological characteristics (density, average degree, transitivity, nature of degree distribution) remained unchanged. We leave it for future work to analyze the dynamic properties of LN.

LN gradually increased adoption and attraction throughout

Number of nodes	2344
Number of payment channels	16617
Average degree	7.0891
Connected components	2
Density	0.00605
Total BTC held in LN	543.61855₿
s-metric	0.6878
Maximal independent set	1564
Bridges	530
Diameter	6
Radius	3
Mean shortest path	2.80623
Transitivity	0.1046
Average clustering coefficient	0.304
Degree assortativity	−0.2690

Fig. 1: LN at a glance: basic properties of the LN graph.

2018, which resulted in 3 independent client implementations (c-lightning[3], eclair[4] and lnd[5]) and 2344 nodes joining LN as of 2019, January 3rd. The density of a graph is defined as $D = \frac{2|E|}{|V||V-1|}$ which is the ratio of present and potential edges. As it is shown in Figure 1. LN is quite a sparse graph. This is further justified by the fact that LN has 530 bridges, edges which deletion increases the

---

[2] https://graph.lndexplorer.com
[3] https://github.com/ElementsProject/lightning/
[4] https://github.com/ACINQ/eclair
[5] https://github.com/lightningnetwork/lnd

number of connected components. Although LN consists of 2 components, the second component has only 3 nodes. The low transitivity, fraction of present and possible triangles in the graph, highlights the sparseness of LN as well.

Negative degree assortativity of the graph indicates that on average low degree nodes tend to connect to high degree nodes rather than low degree nodes. Such a dissortative property hints a hub and spoke network structure, which is also reflected in the degree distribution, see Figure 2.
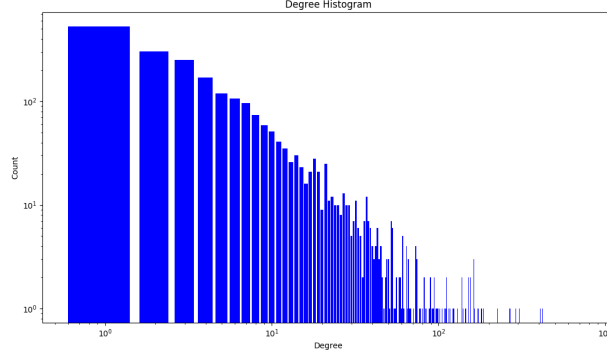
Average shortest path length is 2.80623, without taking into account capacities of edges, which signals

Fig. 2: LN's degree distribution

that payments can easily be routed with a few hops throughout the whole network. Although this is far from being a straightforward task, since one also needs to take into consideration the capacity of individual payment channels along a candidate path.

When a new node joins LN, it needs to select which other nodes it is trying to connect to. In the *lnd* LN implementation key goals for a node is to optimize its centrality by connecting to central nodes. This phenomena sets up a preferential attachment pattern. Other LN implementations rely on their users to create channels manually, which also most likely leads to users connecting to high-degree nodes. Betweenness centrality of a node v is given by the expression $g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$, where σ_{st} is the total number of shortest paths between node s and t, whilst $\sigma_{st}(v)$ is the number of those paths, that pass through v. Closeness centrality of a node v is defined as $CC(u) = \frac{N}{\sum_{u \neq v} d(u,v)}$, where N is the number of nodes in the graph and $d(u,v)$ is the distance between node u and v. Closeness centrality measures how close a node is to all other nodes.

Small-world architectures, like LN, exhibit high clustering with short path lenghts. The appropriate graph theoretic tool to asses clustering is the clustering coefficient [11]. Local clustering coefficient measures how well a node's neighbors are connected to each other, namely how close they are to being a clique. If a node u has $deg(u)$ neighbors, then between these $deg(u)$ neighbors could be at maximum $\frac{1}{2} deg(u)(deg(u) - 1)$ edges. If $N(u)$ denotes the set of u's neighbors, then the local clustering coefficient is defined as $C(u) = \frac{2|(v,w):v,w \in N(u) \wedge (v,w) \in E|}{deg(u)(deg(u)-1)}$.

LN's local clustering coefficient distribution suggestively captures that LN is essentially comprised of a small central clique and a loosely connected periphery.

2.1   Analysis of LN's degree distribution

LN might exhibit scale-free properties as the s-metric suggests. S-metric was first introduced by Lun Li et al. in [5] and defined as $s(G) = \sum_{(u,v) \in E} deg(u)deg(v)$. The closer to 1 s-metric of G is, the more scale-free the network. Diameter and radius of LN suggest that LN is a small world. Somewhat scale-freeness is also exhibited in the degree distribution of
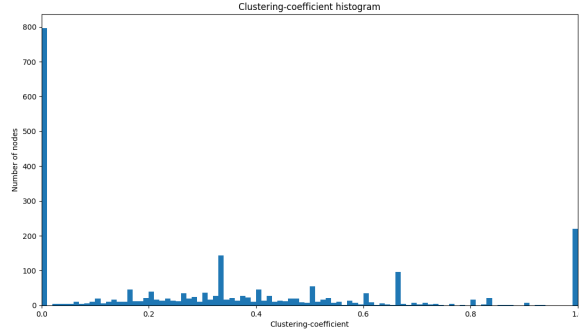

Fig. 3: Local clustering coefficient of LN

LN. Majority of nodes have very few payment channels, although there are a few hubs who have significantly more connections as it can be seen in Figure 2. The scale-freeness of LN is further justified also by applying the method introduced in [2]. The maximum-likelihood fitting asserted that the best fit for the empirical degree distribution is a power law function with exponent $\gamma = -2.1387$. The goodness-of-fit of the scale-free model was ascertained by the Kolmogorov-Smirnov statistic. We found that the p-value of the scale-free hypothesis is $p = 0.8172$, which is accurate within 0.01. Therefore the scale-free hypothesis is plausible for the empirical degree distribution of LN.

3   Robustness of LN

It is a major question in network science how robust a given network is. LN, just like Bitcoin, is a permissionless network, where nodes can join and leave arbitrarily at any point in time. Nodes can also create new payment channels or close them any time. Furthermore as new payments are made, capacities of payment channels are changing steadily. Despite the dynamic nature of LN, its topology's characteristics remain constant after all. In this section we investigate how resilient LN is, whether it can effectively withhold random node failures or deliberate attacks.

Measuring robustness means that one gradually removes nodes and/or edges from the graph, until the giant component is broken into several isolated components. The fraction of nodes that need to be removed from a network to break it

into multiple isolated connected components is called the percolation threshold and is denoted as f_c. In real networks percolation threshold can be well estimated by the point where the giant component first drops below 1% of its original size [1].

3.1 Random Failures

Random failures are a realistic attack vector for LN. If nodes happen to be off-line due to bad connections or other reasons, they can not participate in routing payments anymore. Such a failure can be modeled as if a node and its edges are removed from the graph.

For scale-free networks with degree distribution $P_k = k^{-\gamma}$, where $2 < \gamma < 3$ the percolation threshold can be calculated by applying the Molloy-Reed criteria, ie. $f_c = 1 - \frac{1}{\frac{\gamma-2}{3-\gamma}k_{min}^{\gamma-2}k_{max}^{3-\gamma}-1}$, where k_{min} and k_{max} denote the lowest and highest degree nodes respectively. This formula yields $f_c = 0.9797$ for LN in case of random failures. This value is indeed close to the percolation threshold measured by network simulation as shown in

Network	f_c
Internet	0.92
WWW	0.88
US Power Grid	0.61
Mobil Phone Call	0.78
Science collaboration	0.92
E. Coli Metabolism	0.96
Yeast Protein Interactions	0.88
LN	0.96

Fig. 4: Random failures in networks. Values of critical thresholds for other real networks are taken from [1].

Figure 4, that is, LN provides an evidence of topological stability under random failures. In particular this is due to the fact that in LN a randomly selected node is unlikely to affect the network as a whole, since an average node is not significantly contributing to the network's topological connectivity, see also degree distribution at Figure 2.

3.2 Targeted attacks

Targeted attacks on LN nodes are also a major concern as the short history of LN has already shown it. On 2018 March 21st [6], 20% of nodes were down due to a Distributed Denial of Service (DDoS) attack against LN nodes. Denial of Service (DoS) attacks are also quite probable by flooding HTLCs. These attack vectors are extremely harmful, especially if they are coordinated well. One might expect that not only state-sponsored attackers will have the resources to attack a small network like LN. In the first attack scenario we removed 30 highest-degree nodes one by one starting with the most well-connected one and gradually withdraw the subsequent high-degree nodes. We recorded the number of connected components. As it is shown in Figure 5. even just removing the highest-degree node[7] fragments the LN graph into 37 connected components! Altogether the removal of the 30 largest hubs incurs LN to collapse into 424 components, although most

---

[6] https://www.trustnodes.com/2018/03/21/lightning-network-ddos-sends-20-nodes
[7] http://rompert.com/

of these are isolated vertices. This symptom can be explained by the experienced dissortativity, namely hubs tend to be at the periphery.
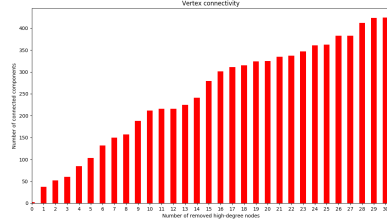

Fig. 5: LN's vertex connectivity, when all the 30 largest hubs are removed one by one
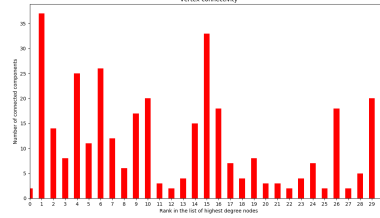

Fig. 6: LN's vertex connectivity if only one high-degree node is removed from the graph.

We reasserted the targeted attack scenario, but for the second time we only removed one of the 30 largest hubs and recorded the number of connected components. As it can be seen in Figure 6 most of the hubs, 25, would leave behind several disconnected components.

Such network fragmentations are unwanted in case of LN, because they would make payment routing substantially more challenging (one needs to split the payment over several routes) or even impossible (there would be no routes at all).

Furthermore we estimated the percolation threshold by simulating two attacking strategies. In the first scenario we removed high degree nodes one by one (high degree removal attack, HDR) and in the second we removed nodes with the highest betweenness centrality (high betweenness removal attack, HBR). Note that in both cases after each node removal we recalculated which node has the highest degree or betweenness centrality in order to have a more powerful attack. We found out that $f_c = 0.1627$ for removing high degree nodes, while for removing high

Network	f_c
Internet	0.16
WWW	0.12
Facebook	0.28
Euroroad	0.59
US Power Grid	0.20
Mobil Phone Call	0.20
Science collaboration	0.27
E. Coli Metabolism	0.49
Yeast Protein Interactions	0.16
LN	0.14

Fig. 7: Real networks under targeted attacks. Values of critical thresholds for other real networks are taken from [1] and [6].

betweenness centrality nodes $f_c = 0.1409$, therefore choosing to remove high betweenness centrality nodes is a better strategy as it can also be seen in Figure 10.

Node outage not only affects robustness and connectivity properties, but also affects average shortest path lengths and available liquidity. Although the outage of random nodes does not significantly increase the average shortest path lengths
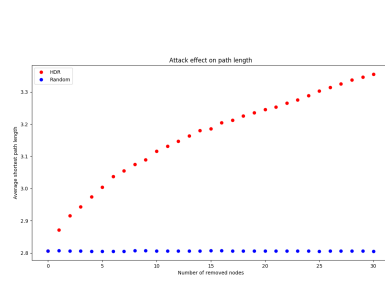
Fig. 8: High degree removal (HDR) attack
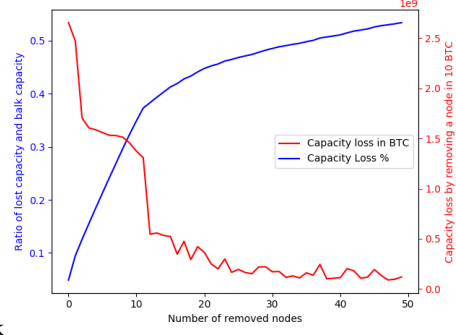effects average shortest path lengths

Fig. 9: Lost capacity as removing high-
degree nodes

in LN, targeted attacks against hubs increase distances between remaining nodes. The spillage of high-degree nodes not only decreases the amount of available liquidity but also rapidly increases the necessary hops along payment routes as Figure 8 and 9 suggest. This could cause increased ratio of failed payments due to larger payment routes and sparser liquidity. Figure 9 demonstrates how capital allocation is centred upon a few high degree nodes, namely already the removal of as few nodes as 37 decreases the available liquidity by more than 50%. Unfortunately most of these nodes are run by a handful of companies.
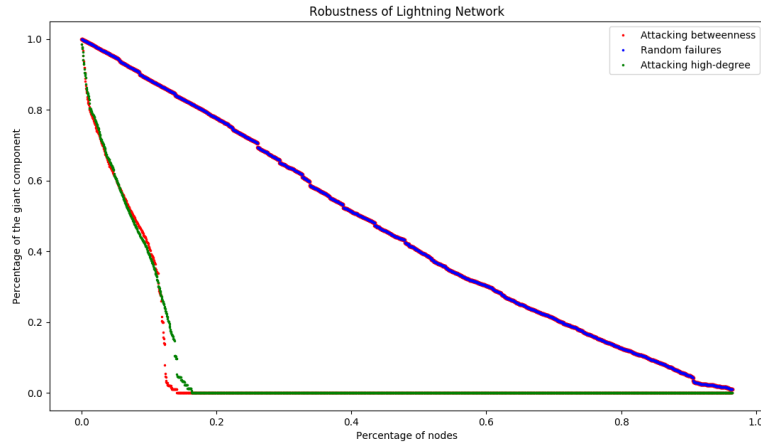

Fig. 10: Percolation thresholds for various attack scenarios: $f_c^{HDR} = 0.1627$, $f_c^{HBR} = 0.1409$, $f_c^{RND} = 0.9645$

3.3   Improving LN's resilience against random failures and attacks

Designing networks which are robust to random failures and targeted attacks appear to be a conflicting desire [1]. For instance a star graph, the simplest hub and spoke network, is resilient to random failures. The removal of any set of spokes does not hurt the connectedness of the main component. However it can not withstand a targeted attack against its central node, since it would leave behind isolated spokes. Furthermore when one attempts increasing robustness of a network, they do desire not to decrease the connectivity of nodes.

A similar optimization strategy of robustness and connectivity to that of [10] could be applied to LN as well. We leave it for future work to empirically assess the robustness and connectivity gains if the strategy of [10] would be implemented in LN client implementations.

Nonetheless, we can still enhance the network's attack tolerance by connecting its peripheral nodes [1]. This could be achieved by LN client implementations by implicitly mandating newcomers to connect to not only hubs, as current implementations do, but also to at least a few random nodes.

4   Conclusion

In summary, a better understanding of the network topology is essential for improving the robustness of complex systems, like LN. Networks' resilience depends on their topology. LN is well approximated by the scale-free model and also its attack tolerance properties are similar to those of scale-free networks; in particular, while LN is robust against random failures, it is quite vulnerable in the face of targeted attacks. High-level depictions of LN's topology convey a false image of security and robustness. As we have demonstrated, LN is structurally weak against rational adversaries. Thus, to provide robust Layer-2 solutions for blockchains, such as LN and Raiden, the community needs to aim at building resilient network topologies.

---

[8] https://www.blocktap.io/

References

1. Barabási, A.L., et al.: Network science. Cambridge university press (2016)
2. Clauset, A., Shalizi, C.R., Newman, M.E.: Power-law distributions in empirical data. SIAM review **51**(4), 661–703 (2009)
3. Gudgeon, L., Moreno-Sanchez, P., Roos, S., McCorry, P., Gervais, A.: Sok: Off the chain transactions. Cryptology ePrint Archive, Report 2019/360 (2019), https://eprint.iacr.org/2019/360
4. Herrera-Joancomarti, J., Navarro-Arribas, G., Ranchal-Pedrosa, A., Pérez-Sola, C., Garcia-Alfaro, J.: On the difficulty of hiding the balance of lightning network channels. Cryptology ePrint Archive, Report 2019/328 (2019), https://eprint.iacr.org/2019/328
5. Li, L., Alderson, D., Doyle, J.C., Willinger, W.: Towards a theory of scale-free graphs: Definition, properties, and implications. Internet Mathematics **2**(4), 431–523 (2005)
6. Lin, Y., Chen, W., Zhang, Z.: Assessing percolation threshold based on high-order non-backtracking matrices. In: Proceedings of the 26th International Conference on World Wide Web. pp. 223–232. International World Wide Web Conferences Steering Committee (2017)
7. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)
8. Poon, J., Dryja, T.: The bitcoin lightning network: Scalable off-chain instant payments. See https://lightning. network/lightning-network-paper. pdf (2016)
9. Roos, S., Moreno-Sanchez, P., Kate, A., Goldberg, I.: Settling payments fast and private: Efficient decentralized routing for path-based transactions. arXiv preprint arXiv:1709.05748 (2017)
10. Shargel, B., Sayama, H., Epstein, I.R., Bar-Yam, Y.: Optimization of robustness and connectivity in complex networks. Physical review letters **90**(6), 068701 (2003)
11. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world'networks. nature **393**(6684), 440 (1998)