The Effect of Network Topology on Credit Network Throughput

Vibhaalakshmi Sivaraman[*], Weizhao Tang[†], Shaileshh Bojja Venkatakrishnan[**], Giulia Fanti[†], Mohammad Alizadeh[*]

[*]MIT CSAIL [†] Carnegie Mellon University [**] The Ohio State University

1. INTRODUCTION

The global economy relies on digital transactions between entities who do not trust one another. Today, such transactions are handled by intermediaries who extract fees (e.g., credit card providers). A natural question is how to build financial systems that limit the need for such middlemen.

Credit networks (similarly, *debit networks*) are systems in which parties can bootstrap pairwise, distributed trust relations to enable transactions between parties who do not trust each other. The core idea is that even if Alice does not trust Charlie directly, if they both share a pairwise trust relationship with Bob, then Alice and Charlie can execute a credit- (or debit-) based transaction through Bob. The trust relationships that comprise such a network can be based on prior experience or observations (e.g., credit scores in a credit network), or they can be based on escrowed funds that are managed either by a third party or an algorithm (debit networks). Recent debit/credit networks from the blockchain community establish pairwise trust relationships through cryptographically secured data structures stored on a blockchain. Prominent examples include *payment channel networks (PCNs)* such as Bitcoin's Lightning Network.

Fig. 1 depicts the operation of a pairwise trust channel (or a *payment channel*) in PCNs. Alice and Bob first cryptographically escrow some number of tokens into a contract stored on the blockchain that ensures the money can only be used to transact between them for a predefined time period. While the channel is active, Alice and Bob can exchange funds without committing to the public ledger. However, if the time period expires or either participant closes the channel, the final state of the channel is committed to the blockchain. PCNs are a network of these pairwise payment channels; if Alice wants to transact with Carol, she can use Bob as a relay and leverage his channel with Carol.

Like traditional communication networks, a central performance metric in credit and debit networks is **throughput**: the total number of transactions a credit network can process per unit time. However, reasoning about credit network throughput is more difficult than in traditional communication networks because of *imbalanced channels*. That is, because channels impose upper limits on credit in either direction, transactions cannot flow indefinitely in one direction over a channel. For instance, in Fig. 1, once Alice sends 3 tokens to Bob, she cannot send any more tokens to Bob (or Carol) until Bob sends her some money back.

Imbalanced channels can affect the throughput of credit networks in unusual ways that depend on topology, user transaction patterns, and transaction routes. An imbalanced channel can harm throughput in other parts of the network due to dependencies between paths (e.g., an imbalanced channel blocks a route, which prevents that route of payments from balancing other channels, blocking more routes and so on). Certain configurations of imbalanced channels can even lead to *deadlocks* where no transactions can flow over certain edges or even the whole network. Recovering from degraded throughput caused by imbalanced channels requires settlement mechanisms outside of the credit network, such as performing "on-chain" transactions on the blockchain to add funds to a channel. These mechanisms incur higher cost and overhead compared to transactions within the credit network and should be avoided as much as possible.

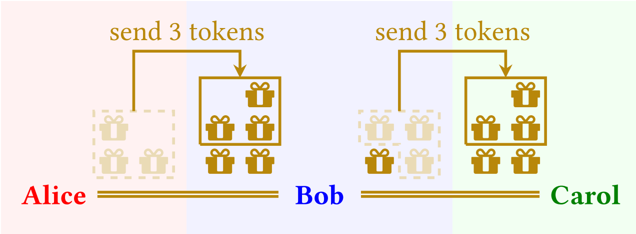

Figure 1: Example payment channel network that allows Alice to send 3 tokens to Carol via Bob.

In this work, we study the role of network topology and channel imbalance on credit network throughput. While system designers cannot directly control the topology of a decentralized network, existing PCNs (e.g., Lightning Network) indirectly influence network topology, for example, with "autopilot" systems that recommend new channels to participants based on a peer's channel degree, size, and longevity. These systems currently lack an understanding of how network topology and channel imbalance impacts the throughput in credit networks. Our goal is to bridge this gap, paving the way for autopilot systems that encourage high-throughput topologies with minimal deadlocks.

2. KEY RESULTS

2.1 Characterizing Throughput Sensitivity

The first part of our results focuses on the relationship between throughput sensitivity and channel imbalance. Informally, the *state* $\boldsymbol{b}$ of a credit network refers to the (instantaneous) balance allocation on every channel in the network. In Fig. 1, for example, the state is: $(0, 5)$ on the Alice-Bob channel and $(1, 5)$ on the Bob-Carol channel (after Alice sends her payment). The *steady-state throughput* $\phi(\boldsymbol{b})$ refers to the highest possible long-term average throughput that

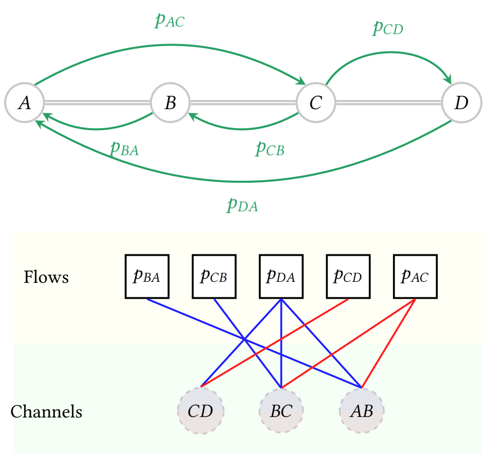

Figure 2: Example bipartite graph that is peelable and identifies the lack of deadlock in the above topology.

can be sustained starting from initial state $\boldsymbol{b}$ for a given usage pattern and path choices. Throughput sensitivity refers to whether the steady-state throughput depends on the state $\boldsymbol{b}$. This property depends on the topology and demand pattern. For some topologies and demands, the state has little to no effect on the achieved throughput; in other topologies, even a slight perturbation of the channel balances can cause an irrevocable loss in steady-state throughput.

To capture this spectrum, we analyze the best- and worst-case throughput ($\Phi_{\min}$ and $\Phi_{\max}$) of a credit network across all possible balance states. A *fully-imbalanced channel* has all of its balance moved to one side of the channel, and we define a fully-imbalanced channel to be *deadlocked* at a credit network state if no transaction can be sent on the channel. Our first result draws a connection between deadlocked channels and throughput sensitivity.

Theorem 1. *If a credit network is deadlock-free, i.e., any state with one or more fully imbalanced channels can be moved to a state with no fully-imbalanced channels, the steady-state throughput* $\phi(\boldsymbol{b})$ *is the same for all initial states* $\boldsymbol{b}$.

Having established that a deadlock-free topology has constant steady-state throughput, we further show that in topologies with deadlocks, the credit network state with the largest deadlock achieves the minimum throughput ($\Phi_{\min}$) of the credit network, while the state with perfect balance on all channels has the largest throughput ($\Phi_{\max}$).

Theorem 2. *A state* $\boldsymbol{b}$ *has the worst steady-state throughput* $\phi(\boldsymbol{b}) = \Phi_{\min}$ *(across all states) if it has the largest number of deadlocked channels across all credit network states.*

We formalize and prove the above theorems in our paper [2].

2.2 Peeling Algorithm

The above results suggest that deadlocks completely characterize the throughput sensitivity of a topology. However, we show that finding a deadlocked state (or showing that no such state exists) for a topology is NP-hard [2]. Instead, we propose a "peeling algorithm" inspired by decoding algorithms for erasure codes [1] that can be used to bound the number of deadlock-free channels in the network.

The peeling algorithm takes as input a credit network topology and a set of flows (paths) in use. The process iteratively identifies channels that cannot be deadlocked in a particular direction, owing to flows that traverse them in that direction and are not blocked elsewhere. The algorithm assigns such channels to a fully-imbalanced state in the opposite direction in search of a deadlock. Each assignment removes a channel from consideration as a possible blocker for flows traversing it in one direction, which may allow more channel states to be eliminated as a possible deadlock.

We visualize the process using a bipartite graph (Fig. 2) with flow and channel partitions. Edges connect each channel to the flows that use it. The connecting edge is colored red if the flow uses the channel from left to right and blue otherwise. The peeling process colors each channel node red or blue (corresponding to one of two fully-imbalanced states) and determines if either color results in a deadlock. The key observation is that every flow traversing a single channel (called flows of *length 1*) constrains the channel's possible deadlocked state to one color. For instance, AB can never be deadlocked in the red direction (with all tokens on B's end) because the blue direction flow P_{BA} can independently move tokens from B to A. We look for such channels with flows of length 1, assign their remaining possible deadlock color, and "peel" those channels from the flows using them in the un-deadlocked color's direction. For example, since AB cannot be deadlocked in the red color direction, flow P_{AC} (with a red edge to AB) is not constrained by AB. Therefore, we can remove AB from consideration for P_{AC}, reducing its length to 1 . The peeling process then repeats with the new length 1 flows. If a channel sees conflicting constraints on its deadlocked states (i.e., flows of length 1 traversing it in both directions), it is declared deadlock-free. This either happens for all channels in the network (the credit network is deadlock-free), or the procedure gets stuck providing a lower bound for the number of deadlock-free channels.

2.3 Empirical Evaluation

We empirically verify that the bounds provided by the peeling algorithm on the number of deadlocked channels are accurate for a variety of topologies. We then use it to compare the best- and worst-case throughput behaviors of standard random graph topologies and a subset of the Lightning Network for randomly-sampled transaction demands. We find that different topologies have different benefits. For example, scale-free graphs have fewer deadlocks and achieve better worst-case throughput than random regular and Erdos-Renyi graphs when the network contains fewer demand pairs, but achieve lower throughput when the network is heavily utilized. We use the evolution of the peeling process to explain these differences. We also take initial steps towards synthesizing deadlock-resilient topologies with optimized path length distributions by building on prior approaches for designing efficient LT codes. We leave details to the evaluation presented in the extended version [2].

3. REFERENCES

[1] M. Luby. LT Codes. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 271–280. IEEE, 2002.

[2] V. Sivaraman, W. Tang, S. B. Venkatakrishnan, G. Fanti, and M. Alizadeh. The effect of network topology on credit network throughput. *Performance Evaluation*, page 102235, 2021.