Posicionamento Lucrativo de Nós e Criação de Rotas de Baixo Custo na Rede Relâmpago

Gustavo F. Camilo[1], Gabriel Antonio F. Rebello[1,2], Lucas Airam C. de Souza[1], Miguel Elias M. Campista[1], Luís Henrique M. K. Costa[1]

[1]Grupo de Teleinformática e Automação
Universidade Federal do Rio de Janeiro (UFRJ)
[2]Sorbonne Université, CNRS, LIP6, F-75005 Paris, França

Resumo. *As redes de canais de pagamento (Payment Channel Networks - PCN) têm atingido sucesso ao substituir os lentos mecanismos de consenso global por acordos criptográficos locais entre nós participantes. Apesar do sucesso, as PCNs sofrem com os modelos atuais de posicionamento de novos nós participantes, que ignoram possíveis ganhos financeiros dos usuários e incentivam a centralização da rede. Este artigo apresenta um modelo de adição de novos nós à rede que cria conexões de alto retorno financeiro ao usuário e baixo custo de emissão de transações. O trabalho formula o problema da adição do nó à rede matematicamente e demonstra que o problema é NP-difícil. O modelo proposto permite ainda que usuários criem canais de longa duração. O artigo desenvolve uma heurística baseada em um algoritmo guloso para resolução do problema. A análise da heurística implementada mostra que a solução oferece recompensa até 3 vezes maior e custo 2 vezes menor que métodos tradicionais que priorizam nós de maior grau, maior centralidade ou PageRank.*

1. Introdução

As redes de canais de pagamento (*Payment Channel Networks* - PCN) oferecem uma solução para o problema de escalabilidade das criptomoedas públicas [Poon e Dryja 2016]. Nesta tecnologia, participantes interessados em transacionar estabelecem uma rede de comunicação "fora-da-corrente" (*off-chain*) em que as transações são diretamente encaminhadas para usuários, sem necessidade de passarem pela corrente de blocos. Ao trocar o requisito de consenso global sobre transações por acordos criptográficos entre participantes fora-da-corrente, as redes de canais de pagamentos permitem a transferência de valores de maneira rápida e segura. Enquanto as principais criptomoedas, como Bitcoin e Ethereum, levam, respectivamente, em torno de uma hora e cinco minutos para confirmar uma transação em média, as redes de canais de pagamentos possibilitam a transferência em poucos segundos [BitcoinWiki 2019]. Devido a essas vantagens, as PCNs têm atingido sucesso e atraído a atenção do público em geral e da academia. A Rede Relâmpago (*Lightning Network* - LN), rede de canais de pagamentos da criptomoeda Bitcoin e principal implementação de PCN, apresenta mais de 75 mil nós e mais de R$ 400 milhões investidos [1ML.com 2022]. A rapidez da Rede Relâmpago é mencionada como um dos motivos para a adoção do Bitcoin como forma de pagamento oficial em El Salvador [CloudTweaks 2021].

Apesar do sucesso, o problema de posicionamento de nós nas redes de canais de pagamento é particularmente importante [Ersoy et al. 2020, Avarikioti et al. 2020, Lange et al. 2021]. Um nó, ao entrar na rede, bloqueia uma quantidade de moedas, que ficam impedidas de serem utilizadas na corrente de blocos, para serem utilizadas no canal por

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, CNPq, FAPERJ (E-26/203.211/2017, E-26/202.932/2017, E-26/202.689/2018 e E-26/200.892/2021) e FAPESP (15/24494-8, 18/23292-0, 15/24485-9 e 14/50937-1).

tempo indeterminado. Apesar desta desvantagem, os usuários recebem como incentivo a possibilidade de emitir transações de forma rápida e obter ganhos financeiros através de tarifas pelo encaminhamento de pagamentos para outros usuários. Os modelos atuais de posicionamento de nós, no entanto, desconsideram completamente os possíveis ganhos financeiros que podem aumentar as receitas dos usuários [LND 2022, Pickhardt 2019]. Em geral, os modelos focam em conexões aos nós de maior grau, buscando obter menores caminhos aos demais participantes, o que aumenta a centralização da rede [Seres et al. 2020, Camilo et al. 2022]. Assim, faz-se necessário mudar os paradigmas dos modelos atuais para elevar os ganhos financeiros que os canais de pagamento podem gerar aos usuários ao longo da vida útil dos canais.

Além dos ganhos financeiros, o posicionamento dos nós influencia também a vida útil dos canais. É de interesse do usuário manter o canal ativo pelo maior tempo possível para receber tarifas de encaminhamento e compensar os custos de abertura de canal. Nas redes de canais de pagamento, o problema de extensão de vida útil dos canais está diretamente ligado ao balanceamento de canais. Isso ocorre porque um canal que encaminha pagamentos em direções opostas a taxas diferentes torna-se desbalanceado e, caso este desbalanceamento não seja corrigido, o canal perde a capacidade de realizar novos pagamentos [Sivaraman et al. 2020]. Neste caso, o usuário possui duas opções para *rebalancear* o canal: fechar e reabrir o canal na corrente de blocos ou efetuar auto-pagamentos em ciclos, utilizando a topologia da rede para realocação de moedas [Decker 2017, Khalil e Gervais 2017]. A primeira opção, de recorrer à corrente de blocos, é lenta e custosa, além de encurtar a vida útil do canal ao fechá-lo. A segunda opção é promissora devido aos baixos custos e rapidez no rebalanceamento. Entretanto, o rebalanceamento através de auto-pagamentos em ciclos depende fortemente das características topológicas da rede. Este artigo mostra que a topologia atual da Rede Relâmpago restringe o rebalanceamento através de ciclos a poucos nós da rede. Os modelos de posicionamento de nós adotados reforçam esta restrição, ignorando a necessidade de formação de ciclos e forçando grande parte dos usuários a recorrerem à corrente de blocos, encurtando a vida útil do canal. Assim, é necessário um novo modelo de conexão que incentive usuários a estabelecerem ciclos, permitindo a extensão da vida útil do canal e a obtenção de altos ganhos financeiros pelo usuário.

Este artigo propõe um método para entrada de nós em redes de canais de pagamentos que seja financeiramente vantajoso ao usuário. Primeiro, propõe-se um modelo que cria canais lucrativos através do posicionamento estratégico do nó, maximizando o recebimento de tarifas de encaminhamento. Além disso, diferentemente de outras propostas [Ersoy et al. 2020, LND 2022], o modelo proposto busca reduzir os custos do usuário em tarifas de encaminhamento pagas por transação, apresentando mais um incentivo para sua adesão. Segundo, o trabalho demonstra que o modelo proposto é um problema NP-difícil e propõe uma heurística baseada em um algoritmo guloso (*greedy*) como solução ao problema formulado[1]. O algoritmo guloso desenvolvido permite ao usuário a criação de ciclos para habilitar operações de rebalanceamento menos custosas e mais rápidas. Por fim, o artigo apresenta uma avaliação de desempenho do algoritmo desenvolvido, comparando-o com outros métodos existentes na literatura [LND 2022, Bianchini et al. 2005]. Os resultados mostram que o algoritmo proposto apresenta incentivos até 3 vezes maiores que métodos tradicionais. Além disso, a adoção do algoritmo proposto pode aumentar a probabilidade do usuário receber tarifas de encaminhamento em até 20 vezes e que a abordagem que cria ciclos não afeta a recompensa significativamente.

O restante do artigo é organizado da seguinte forma. A Seção 2 introduz o funcionamento e fundamentos das redes de canais de pagamentos e da importância de manter canais balanceados. A Seção 3 estuda o estado da arte de redes de canais de pagamentos e áreas re-

[1]Implementação disponível em `https://www.github.com/gFrancoCamilo/ln-looprebalance`.

lacionadas. A Seção 4 formula a proposta do artigo, mostra que o problema considerado é NP-difícil e apresenta um algoritmo guloso de tempo polinomial para o problema. A Seção 5 exibe os resultados de avaliação da proposta do artigo. Por fim, a Seção 6 conclui o artigo e discute trabalhos futuros.

2. Canais de Pagamento

Os canais de pagamento são uma tecnologia baseada em corrente de blocos que permite a dois usuários realizarem pagamentos entre si de forma privada e em tempo real. A premissa principal dessa tecnologia é que o protocolo de consenso, apesar de ser o principal mecanismo que garante a segurança da corrente de blocos, também é o principal causador de atrasos e tarifas excessivas. Usuários interessados em transacionar criam um canal de pagamento ao emitir uma transação de financiamento na corrente de blocos, alocando parte dos seus recursos para um canal "fora-da-corrente". Neste canal, os usuários podem transacionar livremente de maneira segura através de mensagens criptográficas assinadas que, em caso de disputa, podem ser publicadas na corrente de blocos para resolução. Assim, os canais de pagamento permitem a realização do máximo possível de transações diretamente entre as partes envolvidas, sem a publicação em um bloco, e utilizar o protocolo de consenso apenas quando for necessário. As transações "fora-da-corrente" necessitam apenas de uma assinatura de cada parte envolvida [Poon e Dryja 2016].

Nas redes de canais de pagamento, usuários utilizam canais já estabelecidos para rotear pagamentos através de intermediários. O conjunto dos canais de pagamento estabelecidos entre usuários do sistema forma uma rede de canais de pagamento, como mostra a Figura 1. Cada enlace possui uma capacidade, ilustrada dentro do retângulo na figura, que indica o total de *tokens* reservados pelas partes naquele canal. Essa informação é acessível e fica armazenada na transação de financiamento, que é emitida na corrente de blocos para criar o canal de pagamento. Os saldos de cada parte do canal, representados pelos números em cada extremo da aresta, indicam o estado atual do enlace, i.e., quanto cada participante possui para transacionar.

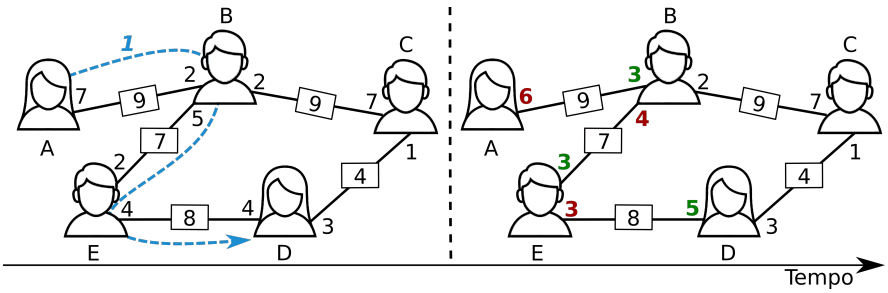


Figura 1: Um exemplo de pagamento de 1 *token* ocorrendo de um usuário A para um usuário D em uma rede de canais de pagamento. O pagamento percorre o trajeto indicado pelo remetente, modificando os saldos em cada canal no caminho. As capacidades dos canais, indicadas nos retângulos, representam o total reservado pelas duas partes e são constantes durante todo o tempo de vida do canal.

O saldo de um canal influencia diretamente a sua capacidade de encaminhar pagamentos. Ao efetuar múltiplos pagamentos em uma única direção, um participante concentra o saldo na direção contrária, diminuindo a capacidade de encaminhar futuros pagamentos nesta direção. Além disso, tendo em vista que a maior parte das aplicações possui uma tendência bem definida para o fluxo de pagamentos, p. ex., de compradores para vendedores, os canais de pagamento devem ser constantemente rebalanceados para se manterem ativos. O método tradicional de rebalancear um canal passa por fechar o canal desbalanceado e abrir novamente através da corrente de blocos [Poon e Dryja 2016]. Ao reabrir o canal, o usuário pode escolher seu novo

saldo reativando-o para novos pagamentos. Esse método, no entanto, requer duas transações na corrente de blocos, sendo lento e caro para o usuário.

3. Trabalhos Relacionados

Diversos trabalhos estudam diferentes estratégias de adição de novos nós [Pickhardt 2019, Ersoy et al. 2020, Lange et al. 2021]. Pickhardt apresenta um *software* que automatiza a criação de canais na rede, chamado de piloto automático (*autopilot*), alternativo ao padrão que permite que o usuário selecione heurísticas para o estabelecimento da conexão, tais como aleatório, central e diâmetro reduzido [Pickhardt 2019]. Lange *et al.* estudam modelos de preferências de conexão presentes na literatura aplicados ao contexto de redes de canais de pagamentos [Lange et al. 2021]. Os autores verificam um compromisso entre segurança e eficiência nos modelos avaliados. Enquanto métodos de conexão aos nós de maior grau na rede apresentam maior eficiência no roteamento, também resultam em diversas vulnerabilidades de segurança para a rede. Ersoy *et al.* focam na criação de canais lucrativos para o usuário [Ersoy et al. 2020]. Os autores formalizam o problema de máxima recompensa, mostram que o problema é NP-difícil e propõem um algoritmo para criação de canais que retorna os canais de máxima recompensa. Os autores, no entanto, consideram um cenário em que os nós buscam somente agir como roteadores, sem emitir pagamentos. Essa consideração impede o desenvolvimento de uma solução próxima ao cenário real das redes de canais de pagamento, em que participantes assumem papéis tanto de vendedores quanto de compradores [Poon e Dryja 2016]. Além disso, todas as soluções citadas ignoram a questão do rebalanceamento dos canais, o que pode implicar em baixo tempo de vida útil para o canal e altos custos na reabertura.

O problema de tornar o rebalanceamento menos custoso e mais rápido é abordado por múltiplas propostas na literatura [Sivaraman et al. 2020, Khalil e Gervais 2017, Otto 2022, Awathare et al. 2021]. Sivaraman *et al.* propõem o Spider, um algoritmo de roteamento para redes de canais de pagamento que utiliza controle de fluxo para manter canais balanceados [Sivaraman et al. 2020]. Os autores analisam que a desigualdade no fluxo de transações da rede causa o desbalanceamento dos canais, o que diminui a vida útil deles. Para corrigir essa assimetria de fluxos, o Spider define que emissores de pagamentos devem manter uma janela de fluxo ajustada para emitir transações na mesma taxa em que recebe transações, mantendo o canal balanceado de acordo com a demanda. No Spider, os nós intermediários estabelecem filas que podem ser usadas para manter as transações em caso de falta de fundos para transferência e aguardar pagamentos na direção contrária. Caso o fluxo de pagamentos em uma direção esteja maior do que o fluxo na direção contrária, a transação aguarda na fila uma transação de mesmo valor na direção contrária para igualar fluxos nas duas direções. Essa solução, apesar de melhorar o balanceamento de canais, apresenta algumas desvantagens. Primeiro, o Spider depende da existência de demandas de pagamentos inconstantes e imprevisíveis na direção contrária para igualar o fluxo nas duas direções. Caso as demandas não existam, o usuário pode ter seu pagamento preso por tempo indeterminado. Segundo, o Spider desconsidera as tarifas em seu algoritmo de roteamento, o que pode resultar em alto custo ao usuário. Por fim, a implementação de filas e controle de fluxo requer mudanças estruturais na Rede Relâmpago, que não apresenta estes recursos atualmente.

Uma das formas mais utilizadas de rebalanceamento nas redes de canais de pagamento envolve a emissão de auto-pagamentos através de rotas circulares, como mostra a Figura 2. Esse método permite que usuários movimentem fundos de um canal de baixa demanda para um de alta demanda, reativando o canal sem recorrer à corrente de blocos e aos lentos mecanismos de consenso. Khalil e Gervais propõem o REVIVE, um método seguro para pagamentos

em rotas circulares [Khalil e Gervais 2017]. No REVIVE, um líder eleito recebe requisições de rebalanceamento de múltiplos usuários e calcula um conjunto de transações que devem ser efetuadas. Esse conjunto de transações busca atender aos requisitos dos usuários e deve garantir que usuários não percam dinheiro no processo. O algoritmo, no entanto, fere a privacidade dos usuários, uma vez que, para calcular o conjunto de transações de rebalanceamento, o líder deve conhecer o saldo dos canais envolvidos. Awathare *et al.* propõem o REBAL, um método de rebalanceamento circular que utiliza o histórico de fluxo de transações no canal para definir a quantidade de fundos que devem ser movimentados [Awathare et al. 2021]. No REBAL, os participantes executam o protocolo de rebalanceamento localmente, o que remove a necessidade de compartilhar informações privadas com terceiros. Otto apresenta uma ferramenta para automatizar o rebalanceamento de canais através de auto-pagamentos na implementação da Rede Relâmpago em linguagem Go [Otto 2022]. A ferramenta permite que o usuário configure a quantidade de fundos que deseja em cada canal e calcula, a partir de um problema de otimização, o melhor conjunto de transações de rebalanceamento localmente.

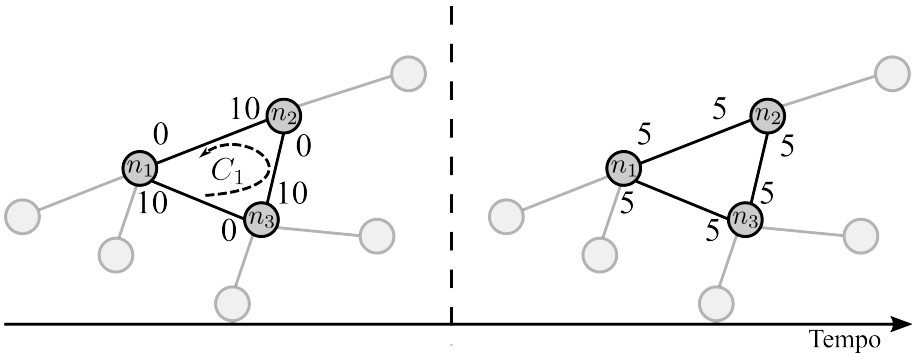


Figura 2: Rebalanceamento através de auto-pagamentos em redes de canais de pagamentos. O nó n_1 encontra uma rota circular e efetua um auto-pagamento, rebalanceando o canal com n_2.

Apesar de se apresentar como uma solução rápida e de baixo custo para rebalancear canais, a técnica de auto-pagamentos por rotas circulares depende fortemente das características topológicas da rede. Seres *et al.* e Camilo *et al.* mostram que a Rede Relâmpago apresenta altas tendências de centralização e baixa transitividade, que têm diminuído [Seres et al. 2020, Camilo et al. 2022]. A baixa transitividade implica a existência de poucos ciclos de três nós na rede que, combinado com a alta centralização de conectividade na rede, restringe as operações de rebalanceamento a poucos participantes da rede. A centralização da conectividade e renda na rede é reforçado pelos pilotos automáticos [LND 2022, Pickhardt 2019], que geralmente conectam os novos nós aos participantes mais centrais.

A baixa transitividade e a crescente tendência de centralização da rede restringem as operações de rebalanceamento por rotas circulares a nós centrais com alto poder econômico. A partir de uma imagem da Rede Relâmpago de agosto de 2021 [Decker 2021], este trabalho verifica que 57% dos nós participam de zero ciclos de três nós (triângulos) na rede. A Figura 3 ilustra a relação do número de triângulos na rede com o grau dos participantes destes triângulos. O número de triângulos na rede é particularmente importante porque, em geral, esses ciclos representam rotas menos custosas para efetuar as operações de rebalanceamento. A concentração do número de ciclos nos nós mais centrais, que geralmente são os mais ricos, contribui ainda mais para a concentração de renda nestes poucos nós. Esses nós podem evitar as altas tarifas e a alta latência de utilizar a corrente de blocos para rebalancear os canais. Ao invés disso, utilizam o rebalanceamento por ciclos, apresentado na Figura 4c, que apresenta baixo custo e alta vazão. Assim, torna-se necessário formar ciclos de três nós na rede para habilitar operações de rebalanceamento de maneira menos custosa e mais rápida para os usuários.

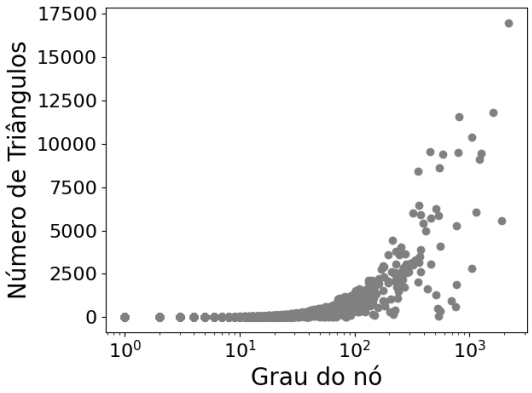


Figura 3: Relação entre participação do nó em um triângulo na rede e grau do nó avaliado. O resultado avaliado considera uma imagem da Rede Relâmpago de agosto de 2021.

Diferentemente dos trabalhos anteriores, este trabalho apresenta uma estratégia de abertura de canais que prioriza o aumento dos ganhos financeiros e diminui possíveis custos aos usuários pela criação de canais. Além disso, o artigo desenvolve um algoritmo que permite que o usuário estabeleça ciclos na topologia da rede, habilitando operações de rebalanceamento e estendendo a vida útil do canal. O modelo proposto, que prioriza os ganhos financeiros do usuário, compensa o custo de abertura de múltiplos canais para estabelecimento de ciclos.

4. Criação de Canais Lucrativos na Rede Relâmpago

Este artigo modela uma rede de canais de pagamentos como um grafo não-direcionado $G = (V, E)$, em que V é o conjunto de vértices representando os nós da rede e E é o conjunto de arestas que representa os canais de pagamentos entre os nós da rede. Um nó u deseja se conectar à rede criando canais duradouros e que sejam lucrativos. O objetivo do nó u, além de coletar tarifas de pagamentos, é utilizar os canais criados para efetuar pagamentos a nós da rede, sem saber *a priori* a quem esses pagamentos serão direcionados. Considera-se que o nó u pode abrir canais com quaisquer nós $v \in V$, mas que, para isso, u deve arcar com os custos de abertura de canal, denotado por C_c. Além disso, assume-se que o nó u possua visão global da rede. Esta hipótese é atendida no cenário real da Rede Relâmpago, em que todos os participantes possuem visão global da topologia da rede e também encontra-se disponível em sítios [Poon e Dryja 2016, 1ML.com 2022]. Para esta abertura de canais, o usuário u possui um orçamento $b(u)$ que define o limite máximo que u pode gastar na criação de canais. Ao abrir um novo canal e_{uv}, o usuário u pode determinar as tarifas $f_{e_{uv}}(tx)$ cobradas por encaminhamento de transação de valor tx neste novo canal. No cenário da Rede Relâmpago, as tarifas são determinadas por uma tarifa base, com valor constante e paga a cada salto, e uma tarifa proporcional ao valor do pagamento, determinada em cada canal. Define-se, então, o evento de uma transação com origem em s e destino t de valor tx passar pelo novo canal estabelecido como $X(tx, s, t)$. O usuário u busca maximizar os ganhos por encaminhar transações nos novos canais criados. Modela-se, então, o ganho esperado do usuário u ao criar um canal com o usuário v em tarifas de uma transação de valor tx neste novo canal como [Ersoy et al. 2020]:

$$\mathbb{E}[G_u] = \sum_{\substack{\forall s,t \in V \\ s \neq t \neq u}} Pr[P = (s,t)] \sum_{tx=1}^{T_{max}} Pr[T = tx] \sum_{v \in C} f_{e_{ui}}(tx) Pr[X(j,s,t)], \tag{1}$$

em que $Pr[P = (s,t)]$ define a probabilidade do pagamento de valor tx ocorrer entre os nós s e t, modelado por uma distribuição P, $Pr[T = tx]$ indica a probabilidade do pagamento, mode-

lado pela variável estatística T, assumir o valor tx, $Pr[X(tx, s, t)]$ representa a probabilidade do pagamento de s para t de valor tx passar pelo canal e_{uv} criado e $C = V - \{u\}$.

Além de maximizar os ganhos em tarifas, expressos na Equação 1, u espera também reduzir o custo de emissão de transações. Ao contrário de artigos anteriores [Ersoy et al. 2020], que consideram o novo usuário somente como um roteador de pagamentos, este artigo considera que o usuário possa efetuar pagamentos para outros participantes da rede. Assim, enquanto outras propostas focam somente no ganho esperado do canal em tarifas, este artigo busca também reduzir o custo médio do usuário u ao efetuar transações. Para isso, assume-se que u inicialmente desconhece os destinos de seus pagamentos. O usuário u, então, busca reduzir sua distância aos outros usuários da rede para minimizar possíveis tarifas, uma vez que as tarifas acumulam a cada salto. O usuário u paga as tarifas a cada salto e a cada pagamento que efetua. O custo esperado $\mathbb{E}[C_u]$ de uma transação de valor tx é dado por

$$\mathbb{E}[C_u] = \sum_{\forall t \in V t \neq u} Pr[P = (u, t)] \sum_{tx=1}^{T_{max}} Pr[T = tx] \sum_{\substack{\forall t \in V \\ t \neq u}} f_{e_{uv}}(tx) d(u, t), \tag{2}$$

em que $d(u, t)$ apresenta a distância em saltos de s até t. Nas redes de canais de pagamento, as distâncias de um nó até outro consideram as tarifas pagas pelo usuário no caminho, i.e., o algoritmo de roteamento utiliza a tarifa cobrada em cada canal como peso das arestas para determinar o menor caminho da origem ao destino.

As Equações 1 e 2 apresentam o ganho esperado de uma transação encaminhada e o custo esperado de uma transação emitida, desconsiderando o custo C_c de abertura do canal. Considerando que u deseja abrir $|C|$ canais e que, ao longo da vida destes canais, u encaminhou K transações de outros participantes e efetuou N transações, o objetivo de u pode ser modelado por $\max\left(K \cdot \mathbb{E}[G_u] - (|C| \cdot C_C + N \cdot \mathbb{E}[C_u])\right)$, em que $\max(.)$ representa o máximo ganho. O problema de posicionamento do nó possui a restrição de orçamento do usuário, definida por $\sum_{i \in C} C_c \leq b(u)$. O modelo matemático desenvolvido, apesar de não estabelecer ciclos, incentiva fortemente os usuários a criá-los. Argumenta-se que, ao acrescentar caminhos para ciclos de baixo custo, habilitando operações de rebalanceamento, a restrição adicionada permite a extensão da vida útil do canal. Com essa extensão, espera-se que os ganhos de $K \cdot \mathbb{E}[G_u]$ diminuam o impacto do custo de abertura de canais $|C| \cdot C_c$, tornando-os insignificantes [Ersoy et al. 2020].

Para simplificar o problema, assume-se a probabilidade de emissão de transação entre dois nós na Equação 1 como uniforme. Assim, a primeira parcela de soma pode ser considerada constante e com valor igual a $Pr[P] = \frac{1}{(|V|-1)(|V|-2)}$, em que $|V|$ é o número de vértices. O mesmo pode ser feito ao fixar um valor médio de transação tx e a tarifa média cobrada $\bar{f}_{e_{uv}}$, eliminando também a segunda parcela da soma como variável. Assim, a Equação 1 pode ser reescrita como $\mathbb{E}[G_u] = \sum_{v \in C} Pr[X(tx, s, t)]$. Por fim, este artigo utiliza a centralidade de intermediação (*Betweenness Centrality* - BC) para modelar a probabilidade $Pr[X(tx, s, t)]$, ou seja, a probabilidade de um nó encaminhar um pagamento de valor tx entre s e t.

Definição 4.1. ***Centralidade de intermediação da aresta.*** *A centralidade de intermediação de uma aresta $e_{uv} = (u, v)$ é proporcional ao número de caminhos mais curtos que passam por e_{uv}, sendo definida por*

$$bc(e_{uv}) = \sum_{\substack{s \neq t, \\ \sigma_{st} \neq 0}} \frac{\sigma_{st[e_{uv}]}}{\sigma_{st}},$$

em que σ_{st} representa o número de caminhos mais curtos entre os vértices s e t e $\sigma_{st[e_{uv}]}$ representa o número de caminhos mais curtos entre s e t que passam pela aresta e_{uv}.

A Definição 4.1 define a centralidade de intermediação de uma única aresta de u. A soma das centralidades de intermediação de todas as arestas de u fornece a centralidade de intermediação $bc(u)$ do nó u. É possível associar a Definição 4.1 de centralidade de intermediação da aresta à probabilidade de um pagamento passar por um canal [Ersoy et al. 2020, Avarikioti et al. 2020]. Essa associação pode ser feita porque a Rede Relâmpago utiliza como modelo de roteamento padrão o algoritmo de Dijkstra para encontrar o caminho mais curto. Como as tarifas representam o peso das arestas, o caminho mais curto é dado pelo caminho com as menores tarifas. Considerando apenas as tarifas fixas por encaminhamento, a centralidade de intermediação pode, sem perda de generalidade, estimar a probabilidade de um pagamento passar por uma aresta. Reescreve-se a Equação 1 como $\mathbb{E}[G_u] = bc(e_{uv})$. O mesmo raciocínio pode ser aplicado com a Equação 2, relacionando-a com o conceito de centralidade de proximidade (*Closeness Centrality* - CC).

Definição 4.2. ***Centralidade de proximidade.*** *A centralidade de proximidade de um vértice v é inversamente proporcional à distância de u a outros nós, sendo definida por*

$$cc(u) = \sum_{\substack{v \in V, \\ v < \infty}} \frac{1}{d_{ut}},$$

em que d_{ut} é a distância entre u e t determinada em número de saltos.

Fixando as tarifas encontradas pelos participantes, a distância para um participante t reflete corretamente os custos encontrados pelos usuário u ao enviar uma transação. A soma das distâncias age, então, como uma média natural para as tarifas encontradas por u assumindo uma distribuição uniforme de valor de transações [Avarikioti et al. 2020]. A minimização do custo descrito na Equação 2 pode ser equivalentemente substituída pela maximização de $1/\mathbb{E}[C_u] = cc(u)$, em que $cc(u)$ representa a centralidade de proximidade do nó u. A combinação da centralidade de intermediação e da centralidade de proximidade representa corretamente os incentivos do usuário para abertura de canais [Avarikioti et al. 2020]. O objetivo do usuário u passa a ser a maximização da centralidade de intermediação, para maiores ganhos, e da centralidade de proximidade para menores distâncias e custos. Introduz-se, então, o incentivo esperado $\text{EI}(e_{uv})$ do usuário u ao abrir um canal e_{uv} com v como sendo dado por e o incentivo esperado $\text{EI}(u)$ pelo nó u ao criar múltiplos canais na rede

$$\text{EI}(e_{uv}) = bc(e_{u,v}) + \frac{1}{d_{uv}} \tag{3}$$

$$\text{EI}(u) = \sum_{v \in C} \text{EI}(e_{uv}). \tag{4}$$

4.1. Incentivo Máximo como Problema NP-difícil

A Equação 3, que determina o incentivo esperado para o usuário u pode ser reduzida em dois problemas independentes: o problema de melhoria máxima da centralidade de intermediação (*Maximum Betweenness Improvement* - MBI) e de melhoria máxima da centralidade de proximidade (*Maximum Closeness Improvement* - MCI) [Bergamini et al. 2018, Crescenzi et al. 2016]. A resolução dos dois problemas fornece a solução para o posicionamento dos nós lucrativo e menos custoso. Estes problemas, apresentados nas Definições 4.3 e 4.4, equivalem a encontrar a melhor resposta de posicionamento de um nó para se maximizar os ganhos com tarifas e reduzir os custos de emissão de transações, respectivamente.

Definição 4.3. ***Problema de melhoria máxima de intermediação (MBI).*** *Dado um grafo não-direcionado G e um vértice v, encontrar k arestas incidentes a v tal que $bc(v)$ é máximo.*

Definição 4.4. ***Problema de melhoria máxima de proximidade (MCI).*** *Dado um grafo não-direcionado G e um vértice v, encontrar k arestas incidentes a v tal que $cc(v)$ é máximo.*

No entanto, o problema de MBI e MCI não são possíveis de serem resolvidos por um algoritmo em tempo polinomial a não ser que $P = NP$, como mostram Bergamini *et al.* [Bergamini et al. 2018] e Crescenzi *et al.* [Crescenzi et al. 2016].

Teorema 4.1. *[Bergamini et al. 2018] O problema de MBI não pode ser aproximado em tempo polinomial com fator maior que $1 - \frac{1}{2\epsilon}$, a não ser que $P = NP$.*

Teorema 4.2. *[Crescenzi et al. 2016] O problema de MCI não pode ser aproximado em tempo polinomial com fator maior que $1 - \frac{1}{15\epsilon}$, a não ser que $P = NP$.*

Os Teoremas 4.1 e 4.2 permitem provar que o problema do incentivo máximo, descrito pela Equação 3 é NP-difícil.

Teorema 4.3. *O problema de incentivo máximo não pode ser aproximado em tempo polinomial, a não ser que $P = NP$.*

Demonstração. A demonstração do Teorema 4.3 passa por reduzir o problema de incentivo máximo definido pela Equação 3 nos problemas de MBI e MCI apresentados na Definição 4.3 e 4.4. Para isso, define-se o algoritmo $\text{MII}(G, n, k) \rightarrow \mathcal{N}$, que recebe o nó n a ser adicionado, o grafo G e a quantidade de canais k a serem criados e retorna o conjunto de nós $\mathcal{N}$ para estabelecer conexão. Assim:

$$
\begin{aligned}
\text{MII}(G, k, n) \rightarrow \mathcal{N} &= \max_{\substack{|\mathcal{N}| \leq k \\ u = n}} \Big(\text{EI}(u) = \sum_{v \in C} \text{EI}(e_{uv}) \Big) \\
\text{MII}(G, k, n) \rightarrow \mathcal{N} &= \max_{\substack{|\mathcal{N}| \leq k \\ u = n}} \Big(\sum_{v \in C} (bc(e_{uv}) + \frac{1}{d_{uv}}) \Big) \\
\text{MII}(G, k, n) \rightarrow \mathcal{N} &= \max_{\substack{|\mathcal{N}| \leq k \\ u = n}} (bc(u)) + \max_{\substack{|\mathcal{N}| \leq k \\ u = n}} cc(u) \\
\text{MII}(G, k, n) \rightarrow \mathcal{N} &= \text{MBI}(G, k, n) + \text{MCI}(G, k, n).
\end{aligned}
$$

No terceiro passo, os somatórios são eliminados, uma vez que a soma da centralidade de intermediação de todas as arestas é igual a centralidade de intermediação do nó e a soma das distâncias do nó a outros pode ser substituída pela centralidade de proximidade. A resolução do problema de MII passa pela resolução de dois problemas independentes NP-difíceis, sendo eles o MBI e o MCI. Como os dois problemas, MBI e MCI, assumem valores iguais ou maiores que zero, a solução do problema de MII passa por calcular os dois problemas separadamente, tornando-se também NP-difícil.

□

4.2. Algoritmo Guloso para Aproximação

Este artigo propõe um algoritmo guloso para uma aproximação em tempo polinomial ao problema descrito. A solução proposta é detalhada no Algoritmo 1. O algoritmo recebe a rede de canais de pagamentos, o nó a ser adicionado e a quantidade de canais a serem criados como entrada e retorna nas conexões indicadas para este nó. A solução proposta passa pelo conjunto de nós criando conexões com cada participante e verificando a recompensa pela criação

da conexão. O algoritmo seleciona, então, o nó que apresenta a melhor recompensa para se criar uma conexão e estabelecer o canal. O cálculo desta recompensa, apresentado no Algoritmo 2, é dado pela verificação da centralidades de intermediação e proximidade. O usuário pode ajustar conforme sua preferência utilizando o parâmetro α. Caso o usuário não pretenda efetuar muitas transações, o parâmetro α pode ser ajustado para $\alpha > 0,5$, privilegiando os ganhos em encaminhamento de transações. Caso pretenda efetuar múltiplas transações, o usuário pode utilizar $\alpha < 0,5$ para obter caminhos mais curtos. Além disso, o algoritmo, por padrão, prioriza o estabelecimento de ciclos de três nós para facilitar e reduzir os custos das operações de rebalanceamento. Entretanto, o algoritmo permite ao usuário optar pela criação de canais sem a priorização de ciclos.

Algoritmo 1: Algoritmo guloso para criação de canais.

Entradas: $G = (V, E) \rightarrow$ representação da rede de canais de pagamentos como um grafo não-direcionado
$n \rightarrow$ nó a ser adicionado
$K \rightarrow$ número de canais que o usuário deseja abrir
$C_y \rightarrow$ variável Booleana indicando se o algoritmo deve priorizar ciclos ou não.
Saída : $\mathcal{N} \rightarrow$ conjunto de nós indicados pelo algoritmo para se criar um canal.

```
Inicializar o conjunto de nós indicados N ← ∅;
enquanto |N| < K faça
    Inicializar a recompensa máxima R_M ← 0;
    Inicializar o nó selecionado N_+ ← None;
    para cada n_i ∈ V faça
        Abrir canal (n, n_i);
        Calcular a recompensa R_{n_i} ← CalculoRecompensa(G, n, n_i, α);
        se R_{n_i} ≥ R_M então
            se C_y = Verdadeiro e |N| > 0 então
                se n_i é vizinho de n_s ∈ N então
                    R_M ← R_{n_i};
                    N_+ ← n_i;
                fim
            fim
            senão
                R_M ← R_{n_i};
                N_+ ← n_i;
            fim
        fim
    fim
    N ← N ∪ N_+;
    Adicionar canal (n, N_+) à rede G;
fim
retorna N
```

5. Avaliação da Proposta

O algoritmo proposto foi implementado utilizando a linguagem de programação Python v3.8 com a biblioteca NetworkX para análise de grafos. Este artigo apresenta uma análise de desempenho do algoritmo proposto, comparando-o com métodos tradicionais de preferências

Algoritmo 2: Algoritmo para cálculo de recompensa.

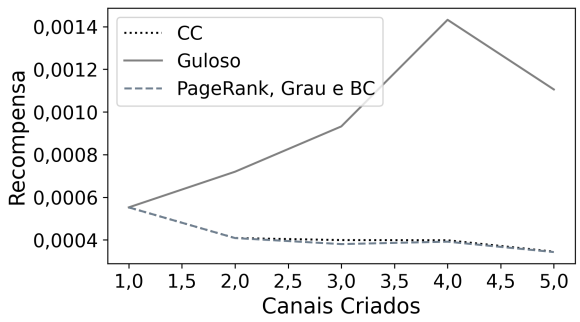
Entradas: $G = (V, E) \rightarrow$ representação da rede de canais de pagamentos como um grafo não-direcionado
$n \rightarrow$ nó a ser adicionado
$n_i \rightarrow$ número de canais que o usuário deseja abrir
$\alpha \rightarrow$ variável Booleana indicando se o algoritmo deve priorizar ciclos ou não.

Saída : $R_{n_i} \rightarrow$ recompensa da criação do canal.

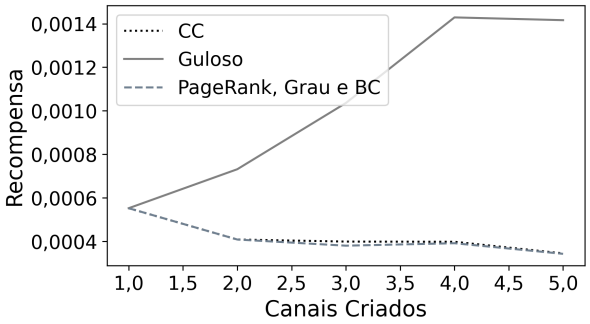
Calcular a centralidade de intermediação $B_C \leftarrow bc(e_{n,n_i})$;
Calcular a centralidade de proximidade $C_C \leftarrow cc(n)$;
Calcular a recompensa $R_{n_i} \leftarrow \alpha \cdot B_C + (1 - \alpha) \cdot C_C$;
retorna R_{n_i}

de conexão adotados nas redes de canais de pagamentos atuais [LND 2022, Pickhardt 2019]. O objetivo do experimento é verificar se o algoritmo proposto atinge o objetivo de recompensas mais altas para o usuário e de reduzir os custos com possíveis transações. Os experimentos utilizam uma imagem (*snapshot*) real da Rede Relâmpago, principal rede de canais de pagamentos atualmente, de janeiro de 2020 com 3.863 nós e 21.628 arestas [Decker 2021]. A imagem da rede foi reconstruída a partir de um conjunto de dados de mensagens de anúncios de nós e canais (*node announcements*), utilizadas na rede por nós que possuam interesse em tornar seus canais públicos para roteamento. As mensagens foram coletadas utilizando a implementação da Rede Relâmpago em linguagem C, a plataforma `c-lightning`.

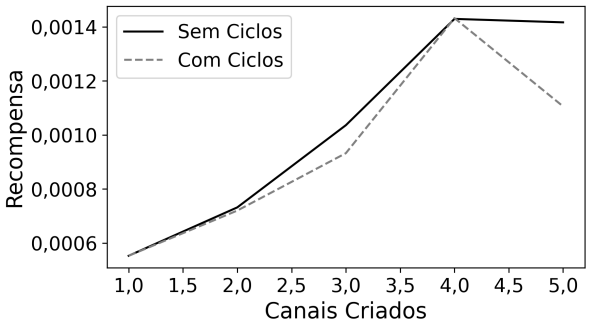
O primeiro experimento compara o algoritmo proposto a métodos que priorizam: centralidade de intermediação, centralidade de proximidade, grau e PageRank [Bianchini et al. 2005]. Nestes métodos, o nó a ser adicionado a rede somente adiciona conexão aos participantes de maior métrica considerada, sem verificar máxima recompensa. Vale ressaltar que essas escolhas refletem opções reais, uma vez que a implementação padrão de abertura de canais na Rede Relâmpago se conecta aos nós de maior grau [LND 2022]. No cenário do experimento, o parâmetro α é ajustado para $\alpha = 0,5$ para priorizar igualmente as recompensas e a redução do custo de transações e o nó adicionado cria 5 canais. A cada canal criado, a recompensa é medida de acordo com a Equação 3. A Figura 4a mostra o resultado da recompensa utilizando o algoritmo guloso proposto comparado a estratégias de PageRank, grau, centralidade de intermediação (BC) e centralidade de proximidade (CC). No cenário do experimento, os nós que apresentam maiores métricas de PageRank, grau e BC coincidem e estão representados por uma única curva. O algoritmo guloso apresenta desempenho mais de 3 vezes melhor que o métodos tradicionais de adição de nós à rede. Isso acontece porque o algoritmo guloso sempre apresenta a melhor recompensa entre todos os nós da rede, enquanto os métodos tradicionais se restringem a adicionar o nó sem verificação de recompensa. No caso do experimento, a adição do quinto canal levou a uma queda na recompensa. O usuário pode decidir, então, criar canais com somente os quatro primeiros nós para obter a recompensa máxima. Neste caso, o algoritmo pode ser facilmente modificado para além de considerar a recompensa máxima testando todos os nós, considerar também criar conexões somente se a recompensa for maior que a anterior. Além disso, como as tarifas determinam o peso das arestas, é possível ajustar as tarifas para aumentar a probabilidade de receber pagamentos e aumentar a recompensa. Esse ajuste de tarifas, como o proposto por Ersoy *et al*, garantiria um crescimento monótono da recompensa [Ersoy et al. 2020]. O mesmo comportamento aparece no caso em que os ciclos não são priorizados, como apresentado na Figura 4b.



(a) Comparação da recompensa do algoritmo guloso priorizando a criação de ciclos proposto com métodos tradicionais de adição de um nó à rede.



(b) Comparação da recompensa do algoritmo guloso não priorizando a criação de ciclos proposto com métodos tradicionais de adição de um nó à rede.



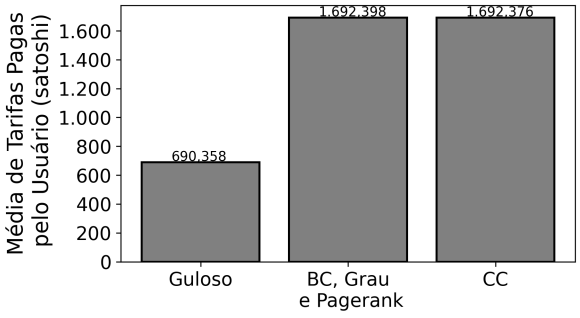
(c) Comparação da recompensa em relação à abordagem que prioriza a criação de ciclos de 3 nós com a abordagem que não prioriza a criação de ciclos.

Figura 4: Resultados do incentivo de um nó ao ser adicionado à rede.

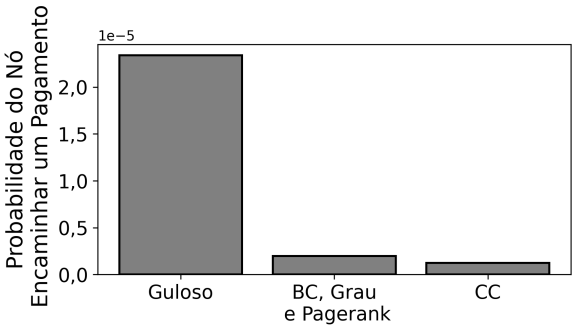
O segundo experimento compara as recompensas com o algoritmo guloso sem priorizar ciclos e com priorização de ciclos. A Figura 4c ilustra o resultado do experimento. É possível perceber que a solução sem ciclos apresenta recompensa um pouco melhor comparado com a solução que prioriza a criação de ciclo. Esse resultado é esperado, uma vez que a solução que prioriza ciclos apresenta um conjunto de nós mais restrito que a solução sem ciclos. Enquanto a opção sem ciclos pode se conectar a qualquer nó da rede e obter a recompensa máxima, a solução com ciclos deve verificar os vizinhos dos nós já conectados para garantir a criação de um ciclo, não necessariamente atingindo a recompensa máxima. Apesar dessa restrição, é possível verificar que a solução que prioriza a formação de ciclos se aproxima da solução sem priorização. Ao entregar as duas opções ao usuário, o algoritmo permite que o próprio participante decida pela garantia de opções para operações de rebalanceamento.

O terceiro experimento verifica a média de tarifas pagas pelo novo usuário ao efetuar pagamentos. O experimento verifica as tarifas cobradas por cada nó ao encaminhar um pagamento por seu canal e calcula a média de todas as tarifas do novo usuário até outros participantes. Neste cenário, o novo nó cria cinco canais sem priorizar ciclos e o parâmetro α é ajustado para $\alpha = 0,5$. A Figura 5a mostra que o algoritmo proposto de adição de nó é efetivo em posicionar o nó de maneira estratégica para efetuar pagamentos menos custosos. Ao utilizar o algoritmo proposto, o usuário consegue pagar até duas vezes menos tarifas quando comparado aos métodos tradicionais. Ao reduzir as distâncias aos outros usuários, o algoritmo garante que pagamentos efetuem poucos saltos, diminuindo as tarifas ao usuário.

O último experimento mede a probabilidade do nó adicionado encaminhar pagamentos e, assim, receber tarifas. O novo nós ajusta a taxa cobrada nos novos canais à média da rede e verifica sua centralidade de intermediação. O cenário do experimento é o mesmo dos anteriores, em que o novo nó cria cinco canais sem priorizar ciclos e o parâmetro α é ajustado para $\alpha = 0,5$.



(a) Comparação da média de tarifas pagas pelo usuário ao efetuar pagamentos utilizando o algoritmo guloso e métodos tradicionais.



(b) Comparação da probabilidade do usuário encaminhar pagamentos utilizando o algoritmo guloso e métodos tradicionais.

Figura 5: Resultados das vantagens financeiras do usuário ao ser adicionado à rede.

A Figura 5b aponta que o participante, ao adotar o algoritmo proposto, pode aumentar sua probabilidade de receber pagamentos em aproximadamente 20 vezes. A alta probabilidade de encaminhar pagamentos apresenta um incentivo financeiro real ao usuário para a adoção do algoritmo, uma vez que o usuário passa a receber tarifas por estes pagamentos.

6. Conclusão

Este artigo apresentou um modelo de preferência de conexão de novos nós nas redes de canais de pagamento. O modelo formulado matematicamente apresenta vantagens financeiras ao usuário, como o posicionamento estratégico do nó de modo a receber tarifas de encaminhamento e economizar na emissão de transações. Espera-se que estas características facilitem a adesão do modelo proposto por novos usuários. O artigo desenvolveu uma heurística baseada em um algoritmo guloso para resolução do problema formulado. Os resultados do algoritmo proposto mostram que a solução apresentada recompensa o usuário em até três vezes mais que os métodos tradicionais de adição de nós na rede. Além disso, o resultado priorizando a criação de ciclos comparado ao resultado sem priorizar não apresenta diferenças significativas financeiramente. Por fim, verificou-se que o algoritmo desenvolvido apresenta incentivos financeiros consideráveis para o usuário, que pode aumentar sua probabilidade de receber tarifas em até 20 vezes e reduzir seus custos com pagamento em até duas vezes. Como trabalhos futuros, espera-se integrar a solução desenvolvida como um piloto automático para utilização em cenários reais da Rede Relâmpago e aplicar o modelo aos nós já existentes na rede.

Referências

1ML.com (2022). 1ML - Lightning Network Search and Analysis Engine. `https://1ml.com/`. Acessado em 30 de dezembro de 2022.

Avarikioti, Z., Heimbach, L., Wang, Y. e Wattenhofer, R. (2020). Ride the Lightning: The Game Theory of Payment Channels. Em Bonneau, J. e Heninger, N., editors, *Financial Cryptography and Data Security*, páginas 264–283, Cham. Springer International Publishing.

Awathare, N., Suraj, Akash, Ribeiro, V. J. e Bellur, U. (2021). REBAL: Channel Balancing for Payment Channel Networks. Em *2021 29th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, páginas 1–8.

Bergamini, E., Crescenzi, P., D'angelo, G., Meyerhenke, H., Severini, L. e Velaj, Y. (2018). Improving the betweenness centrality of a node by adding links. *ACM J. Exp. Algorithmics*.

Bianchini, M., Gori, M. e Scarselli, F. (2005). Inside pagerank. *ACM Transactions on Internet Technology (TOIT)*, 5(1):92–128.

BitcoinWiki (2019). Bitcoin Scalability. https://en.bitcoin.it/wiki/Scalability. Acessado em 30 de dezembro de 2022.

Camilo, G. F., Rebello, G. A. F., de Souza, L. A. C., Potop-Butucaru, M., Amorim, M. D., Campista, M. E. M. e Costa, L. H. M. K. (2022). Topological evolution analysis of payment channels in the lightning network. Em *2022 IEEE Latin-American Conference on Communications (LATINCOM)*, páginas 1–6.

CloudTweaks (2021). How Bitcoin Brought The Lightning Network To El Salvador. https://cloudtweaks.com/2021/07/how-bitcoin-brought-lightning-network-el-salvador/. Acessado em 31 de dezembro de 2022.

Crescenzi, P., D'angelo, G., Severini, L. e Velaj, Y. (2016). Greedily Improving Our Own Closeness Centrality in a Network. *ACM Transactions on Knowledge Discovery from Data*, 11(1):9:1–9:32.

Decker, C. (2017). Splicing. [Lightning-dev] Channel top-up. Disponível em: https://lists.linuxfoundation.org/pipermail/lightning-dev/2017-May/000696.html. Acessado em 30 de dezembro de 2022.

Decker, C. (2021). Lightning network research; topology datasets. https://github.com/lnresearch/topology. Acessado em 30 de dezembro de 2022.

Ersoy, O., Roos, S. e Erkin, Z. (2020). How to Profit from Payments Channels. Em Bonneau, J. e Heninger, N., editors, *Financial Cryptography and Data Security*, páginas 284–303, Cham. Springer International Publishing.

Khalil, R. e Gervais, A. (2017). Revive: Rebalancing Off-Blockchain Payment Networks. Em *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, páginas 439–453, New York, NY, USA. ACM.

Lange, K., Rohrer, E. e Tschorsch, F. (2021). On the impact of attachment strategies for payment channel networks. Em *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, páginas 1–9.

LND (2022). lnd-autopilot. Disponível em: https://github.com/lightningnetwork/lnd/tree/master/autopilot. Acessado em 30 de dezembro de 2022.

Otto, C. (2022). Rebalance-LND. Disponível em: https://github.com/C-Otto/rebalance-lnd. Acessado em 30 de dezembro de 2022.

Pickhardt, R. (2019). lightning-network-autopilot. Disponível em: https://github.com/renepickhardt/lightning-network-autopilot. Acessado em 30 de dezembro de 2022.

Poon, J. e Dryja, T. (2016). The bitcoin lightning network: Scalable off-chain instant payments.

Seres, I. A., Gulyás, L., Nagy, D. A. e Burcsi, P. (2020). Topological analysis of bitcoin's lightning network. Em Pardalos, P., Kotsireas, I., Guo, Y. e Knottenbelt, W., editors, *Mathematical Research for Blockchain Economy*, páginas 1–12, Cham.

Sivaraman, V., Venkatakrishnan, S. B., Ruan, K., Negi, P., Yang, L., Mittal, R., Fanti, G. e Alizadeh, M. (2020). High throughput cryptocurrency routing in payment channel networks. Em *Proceedings of the 17th Usenix Conference on Networked Systems Design and Implementation*, NSDI'20, páginas 777–796, USA. USENIX Association.