Consenso por Localidade: Um Mecanismo de Consenso Leve com Convergência por Vizinhanças para Cadeia de Blocos

Gabriel R. Carrara, Diogo M. F. Mattos, Célio V. N. Albuquerque

[1]Laboratório MídiaCom - IC/TET/PPGEET
Universidade Federal Fluminense (UFF)
Niterói/RJ – Brasil

Abstract. *Private blockchains tend to apply deterministic consensus mechanisms as a more efficient alternative to proof-based mechanisms. Deterministic mechanisms tolerate two types of failures, byzantine, and crash-fault. Byzantine fault-tolerant consensus assumes restrictive assumptions of time and number of failures to guarantee validity, while termination depends on message broadcasting between the nodes. Crash-Fault tolerant consensus is less stringent to ensure termination and higher throughput while sacrificing agreement. This paper proposes a lightweight consensus mechanism based on locality voting with confirmed message broadcasting. Formation rules in the neighborhoods of the peer-to-peer network relax the trade-off between agreement and termination. Experimental results show that agreement and termination are guaranteed in the case of using more permissive formation rules. At the same time, the cost of achieving consensus is reduced by up to* 46% *in more stringent formation rules with little impact on termination and agreement.*

Resumo. *Cadeias de blocos privadas tendem a aplicar mecanismos de consenso determinísticos como alternativa aos baseados em prova. Mecanismos determinísticos toleram dois tipos de falhas, bizantinas e de parada. O consenso tolerante a falhas bizantinas assume hipóteses restritivas de tempo e número de falhas para garantir a validade, enquanto a terminação depende da difusão de mensagens entre o nós. O consenso tolerante a falhas é menos restritivo para garantir a terminação e maior vazão, sacrificando o acordo. Este artigo propõe um mecanismo de consenso leve baseado na votação por localidade com difusão confirmada de mensagens. Regras de formação nas vizinhanças da rede par-a-par flexibilizam a relação de compromisso entre acordo e o custo de terminação. Resultados experimentais mostram que o acordo e terminação são garantidos no caso de uso de regras mais permissivas, enquanto o custo para alcançar o consenso é reduzido em até* 46% *em regras mais restritivas com baixo impacto sobre a terminação e o acordo.*

1. Introdução

Mecanismos de consenso buscam resolver o problema de consenso definido em um conjunto de n de processos conhecidos, onde pode haver um número máximo de f de processos defeituosos. As falhas de processo são do tipo falha de parada, na qual um processo deixa de responder ou responde em tempo infinito, ou falha bizantina, na qual os processos se comportam em desacordo com o protocolo definido [Correia et al. 2011]. O problema do consenso é formalmente definido em termos de três propriedades, validade,

acordo e terminação. A validade garante que se todos os processos corretos propuserem o mesmo valor v, qualquer processo correto converge para v. O acordo define que não há dois processos corretos que decidem de forma diferente. A terminação assegura que cada processo correto eventualmente decida.

Cadeias de blocos privadas tendem a aplicar mecanismos de consenso determinísticos baseados em votação para garantir a alta taxa de efetivação de transações com baixo custo computacional. O consenso determinístico garante a validade e o acordo forte, ao custo do sacrifício da terminação. A terminação de mecanismos de consenso por votação é dependente da realização da comunicação na rede par-a-par adjacente à cadeia de blocos e da consequente difusão de mensagens de votação [Carrara et al. 2020]. O processo de votação utilizado por mecanismos de consenso determinísticos possui elevado grau de troca de mensagens entre os nós da rede. Esses mecanismos se baseiam em múltiplas rodadas de troca de mensagens para garantir o acordo mesmo na presença de falhas bizantinas. Por outro lado, mecanismos de consenso como o Raft [Ongaro and Ousterhout 2014] apresentam baixo custo de troca de mensagens, porém não são capazes de tolerar tipos de falhas bizantinas, somente falhas de parada.

Este artigo propõe um mecanismo de consenso leve baseado na votação por localidade com difusão confirmada de mensagens. São aplicadas regras de formação para as vizinhanças da rede par-a-par para flexibilizar a relação de compromisso entre acordo e o custo para alcançar a terminação. Uma vizinhança é o conjunto de nós do qual um nó requisita votos para realizar o consenso. As regras de formação são critérios aplicados ao selecionar os vizinhos de um nó com o intuito de modificar parâmetros do consenso. A abordagem baseada em vizinhanças permite que processos de votação sejam resolvidos localmente antes de convergir para um resultado global. Dessa maneira, ao fim de uma rodada de votação bem sucedida, é garantido que o consenso foi atingido em todas as vizinhanças que possuem a maioria de nós corretos.

Mecanismos de consenso atuais que toleram falhas bizantinas, elevam o custo de comunicação para alcançar o consenso [Castro et al. 1999, Bessani et al. 2014, Kwon 2014], ou tentam reduzir esse custo sacrificando sua capacidade de tolerar falhas mais complexas [Ongaro and Ousterhout 2014, Lamport 2006]. O principal diferencial do consenso por localidade é a capacidade de flexibilizar o acordo, terminação e custo ainda sendo capaz de tolerar alguns modelos de falhas mais complexos que a falha de parada (*crash-fault*). Resultados experimentais mostram que o consenso por localidade é capaz de reduzir o número de mensagens para alcançar o consenso em até 46% com uma redução de apenas $3,68\%$ no número de nós alcançados pelo consenso.

O restante deste artigo está organizado da seguinte maneira. A Seção 2, discute os trabalhos relacionados. A Seção 3 apresenta as principais propriedades dos mecanismos de consenso. A Seção 4 discute o funcionamento do mecanismo proposto. A Seção 5 avalia a proposta e apresenta os resultados e a Seção 6 conclui o trabalho.

2. Trabalhos Relacionados

Mecanismos de consenso probabilísticos são modelos de mecanismo que sacrificam o acordo, assumindo que uma rodada de consenso pode obter um valor com uma certa probabilidade. Tais mecanismos, normalmente, se baseiam no uso de recursos computacionais dos participantes para obter o consenso, assumindo um modelo de prova e permitindo

que um nó vencedor decida que bloco acrescentar a cadeia. O mais famoso desses mecanismos é a Prova de Trabalho[Nakamoto 2008]. Esse mecanismo se baseia em desafios criptográficos para garantir a integridade dos blocos propostos. Esses desafios são resolvidos utilizando uma abordagem baseada em força bruta, o que, consequentemente, exige um grande investimento de recursos computacionais e de energia, tornando a Prova de Trabalho um mecanismo com alto custo de execução.

Para reduzir o investimento de energia sem abrir mão da necessidade de investimento de recursos computacionais, são propostos outros modelos de mecanismos baseados em prova. A ideia central dos mecanismos baseados em prova é apenas um nó na rede seja capaz de gerar evidências de que detém o direito de inserir um bloco correto na rede. A evidência é verificável pelos outros nós, assim como a corretude do bloco. A prova de participação (Proof-of-stake) [Rebello et al. 2020] troca o desperdício de energia necessário para resolver o desafio criptográfico por um investimento na forma de ativos da rede. A prova de tempo decorrido (Proof-of-Elapsed-Time) [Chen et al. 2017] utiliza um hardware proprietário presente nos processadores Intel para exigir que os nós aguardem um período de tempo aleatório de maneira certificada. Nós que terminam esse período primeiro propõem um novo bloco.

Os mecanismos de consenso determinísticos tendem a aplicar técnicas de sincronização, como por exemplo votações, para garantir que o consenso seja alcançado. O Paxos [Lamport et al. 2001] e suas variantes o Fast Paxos [Lamport 2006] e o Cheap Paxos [Lamport and Massa 2004] são tolerantes a falhas de parada. Essa família de mecanismos busca alcançar o consenso através de uma complexa troca de mensagens e sincronização entre nós exercendo diferentes papéis na rede. A grande quantidade de troca de mensagens garante o acordo do mecanismo ao custo de sua terminação. O Fast Paxos propõe uma melhoria na terminação do Paxos através da redução do número de mensagens necessárias para alcançar o consenso. O Cheap Paxos assume um modelo de rede com maior sincronização temporal entre nós reduzindo o número de confirmações necessárias para alcançar o consenso. O consenso por localidade permite a criação de abordagens que toleram falhas mais complexas que a falhas de parada através da aplicação de modelos de rede sobreposta diferentes.

Mecanismos de consenso tolerantes a falhas bizantinas aplicam técnicas para resolver o problema dos generais bizantinos definido por Lamport [Lamport et al. 1982]. Esses mecanismos tendem a aplicar processos de votação complexos para garantir a segurança e acordo do consenso ao custo de terminação com maior sobrecarga de mensagens e mais demorada. O PBFT (*Practical Byzantine Fault Tolerance*) [Castro et al. 1999] prevê a aplicação de três etapas de troca de mensagens entre todos os nós da rede. O PBFT assume que as falhas são independentes e que os nós dependem parcialmente um do outro. Algumas das vantagens do PBFT estão relacionadas ao fato de ter um baixo custo energético, e também a um tempo de execução abaixo da média para sistemas assíncronos com um pequeno aumento na latência. No entanto, em uma cadeia de blocos com um grande número de pares, um grande número de mensagens trocadas para obter consenso implica perda significativa de desempenho.

Kwon propõe o Tendermint [Kwon 2014], um mecanismo de consenso capaz de punir os participantes que tentam criar ramificações da cadeia de blocos. O Tendermint propõe resolver os problemas provenientes de mecanismos de Prova de Trabalho adicio-

nando uma forma de punição aos nós maliciosos e prevenindo o ataque de gasto duplo. Bessani *et al* propõem o BFT-SMaRt [Bessani et al. 2014], um mecanismo baseado em máquinas de estado replicadas tolerante a falhas bizantinas. Seu funcionamento é baseado na eleição de um líder responsável pela validação dos pedidos de transação recebidos de outros nós. No entanto, o mecanismo apresenta alto custo de terminação, pois as requisições feitas por clientes devem ser enviadas para todos os nós da rede e os nós devem enviar votos para todos os demais antes de alcançar consenso.

O Ripple [Schwartz et al. 2014] é um projeto de código aberto que funciona como uma criptomoeda e como uma rede de pagamento para transações financeiras. Ambas as aplicações têm subjacente o Algoritmo de Consenso do Protocolo Ripple, que pode suportar $(n-1)/5$ falhas em que n é o número de nós na rede. As principais vantagens do Protocolo Ripple são as melhorias fornecidas na utilidade como a conveniência dos usuários e à baixa latência do sistema. O princípio das votações por vizinhanças utilizado no Consenso por localidade se assemelha ao conceito de UNL proposto pelo Ripple. Uma UNL(*Unique Node List*) é uma lista de nós na qual um determinado nó confia. Em ambos os mecanismos, regras para criação de grupos de votação são utilizadas para assegurar o acordo e a terminação do consenso.

3. Cadeia de Blocos e Mecanismos de Consenso

A arquitetura da cadeia de blocos se divide em três camadas: transação, geração de blocos e distribuição [Oliveira et al. 2019]. A camada de transação define os critérios usados para gerar transações. Os usuários devem assinar as transações antes de divulgá-las para garantir o não repúdio e para permitir o controle de acesso e autenticação de seu conteúdo. O processo de validação das transações e de mineração de blocos reside na camada de geração de blocos. Todas as transações emitidas aguardam execução em uma base de transações não validadas. Os nós responsáveis pela validação das transações e pela mineração de blocos selecionam conjuntos de transações e as inserem em um bloco candidato. Antes de adicionar a transação ao bloco, o nó de validação deve verificar a transação em relação às regras de validação da rede. Uma transação é adicionada ao bloco somente se for válida de acordo com as regras da rede. Caso contrário, essa transação é descartada.

Os nós de validação organizam a ordem na qual as transações são inseridas no bloco. O conteúdo restante do bloco depende do mecanismo de consenso usado na rede. O mecanismo de consenso rege o processo de criação de novos blocos. Quando um validador gera um novo bloco, dissemina o bloco por toda a rede. O processo de distribuição é parte da camada de distribuição, assim como a inserção do bloco gerado na cadeia. Tal inserção é bem sucedida se um nó gerou o bloco corretamente. Caso contrário, o bloco gerado é descartado.

Os mecanismos de consenso são algoritmos que buscam chegar a um acordo sobre um dado ou estado operando sobre um sistema distribuído. Lamport define duas propriedades fundamentais para o desenvolvimento de mecanismos de consenso, segurança e vivacidade [Lamport et al. 2001]. A propriedade de segurança define que o consenso é responsável por garantir que um sistema distribuído escolha apenas um único valor proposto e um processo correto escolhe apenas entre os valores propostos. A propriedade de vivacidade subdivide os mecanismos de consenso em dois grandes grupos: mecanismos

determinísticos e mecanismos probabilísticos. Mecanismos determinísticos de consenso asseguram o acordo sobre os dados após sua aplicação. Os mecanismos probabilísticos de consenso assumem que o acordo tende a acontecer, mas sua convergência não é garantida.

Diferentes propostas se concentram na tolerância de falhas de parada (*crash-failure*) [Ongaro and Ousterhout 2014, Lamport et al. 2001, Schwartz et al. 2014] ou falhas bizantinas [Castro et al. 1999, Bessani et al. 2014, Kwon 2014]. O modelo tolerante a falhas de parada permite a criação de mecanismos leves, geralmente focados em votação. Falhas bizantinas exigem protocolos, tais como PBFT [Castro et al. 1999] e bft-SMaRt [Bessani et al. 2014], que aplicam processos de votação complexos, difundindo as propostas de votação entre todos os nós da rede. Esta abordagem, juntamente com o uso de um limiar de consenso mais estrito, normalmente $3f + 1$, sendo f o número de falhas toleradas, permite que cada rodada de consenso termine com um número significativo de nós alcançados, garantindo uma forte propriedade de acordo. No entanto, o grande número de mensagens trocadas entre nós traz altos custos de comunicação e demora para a convergência dos mecanismos, enfraquecendo a propriedade de terminação [Carrara et al. 2019].

O compromisso entre os parâmetros de acordo e terminação permite uma melhor adaptação aos diferentes cenários de aplicação, não se limitando apenas aos extremos das propriedades de consenso. O consenso por localidade permite a criação de diferentes abordagens através da aplicação de modelos de vizinhança.

4. Consenso com Convergência por Vizinhanças

O mecanismo proposto consiste de um mecanismo de consenso probabilístico baseado em votação para redes de cadeia de blocos privadas não permissionadas. O funcionamento do mecanismo se baseia na difusão confirmada de propostas de votação através de vizinhanças previamente formadas. Cada nó participante do consenso possui sua própria vizinhança e propaga as propostas de votação nela. Na visão de cada nó, o consenso é alcançado localmente, dentro de sua vizinhança. Do ponto de vista global, o consenso ocorre em toda a rede garantindo a propriedade de segurança do mecanismo.

4.1. Premissas da Proposta

A proposta assume premissas quanto aos participantes, o modelo de comunicação e modelos de falhas.

Participantes do Protocolo: Assume-se que os nós participantes do protocolo pertencem a uma rede de cadeia de blocos privada não permissionada. A rede pode ser composta tanto por um consórcio de instituições interessadas no seu funcionamento, mas também pode ser criada em um ambiente restrito, como o centro de dados de uma empresa. A discussão sobre as possíveis aplicações da tecnologia de cadeia de blocos nesses cenários está fora do escopo deste trabalho. Nestes ambientes, os participantes da rede são conhecidos e possuem identidades pre-estabelecidas através de um par de chaves pública e privada resistente a ataques Sybil [Douceur 2002]. Essas identidades são atribuídas aos participantes no momento de configuração da rede. Uma vez que a rede é estabelecida, assume-se que não haverá saída e entrada de nós na rede. Caso um nó seja desconectado da rede, ele será considerado em falha pelo protocolo. A inserção e remoção de nós na rede pode ser feita a partir de sua reconfiguração. Nesse caso, o funcionamento da rede

deve ser interrompido e uma nova configuração de vizinhanças deve ser gerado com o novos nós. A discussão sobre a reconfiguração eficiente da rede para realizar a entrada de novos nós não é escopo deste trabalho.

Comunicação: Assume-se canais de comunicação com as garantias fornecidas pelo protocolo TCP. Considera-se que a camada de distribuição fornece um serviço de entrega confiável capaz de entregar mensagens dentro de um limite de tempo finito. Essas hipóteses não são restritivas, já que em aplicações reais a comunicação baseada em protocolo de transporte TCP fornece a entrega confiável e ordenada de dados. As mensagens trocadas pelos nós são assinadas com suas privadas. Dessa forma, é possível identificar de forma inequívoca o remetente de uma mensagem.

Modelo de Falhas: No modelo de falhas aplicado visa representar dois tipos comportamentos. No primeiro, nós maliciosos escolhem não propagar as mensagens pela rede na tentativa de impedir que o limiar de consenso seja alcançado. No segundo modelo, os nós maliciosos enviam as mensagens recebidas, porém com o conteúdo das mensagens alterados de maneira que o processo de validação dos nós corretos falhe e não sejam iniciadas novas votações locais. Em ambos os casos o comportamento por parte da rede é o mesmo, não são iniciadas novas votações locais. Nós que se desconectarem da rede são considerados em falha, pois também não serão capazes de trocar mensagens e nem iniciar votações em suas vizinhanças.

4.2. Propriedades do Consenso

A partir das premissas anteriores, define-se o comportamento da proposta perante as propriedades de validade, terminação e acordo.

Validade: O processo de validação de blocos e transações é feito através de uma função externa. De maneira semelhante ao Protocolo Elastico [Luu et al. 2016], assume-se uma função externa $C : \mathbb{Z} \rightarrow \{0, 1\}$ responsável por validar blocos e suas transações. Essa função atua como um oráculo para os nós da rede, garantindo que propostas inválidas não sejam aceitas. Essa premissa pode ser assumida, pois no caso das cadeias de blocos, cada bloco é verificável localmente, através da réplica da cadeia possuída pelo nó.

Terminação: A terminação de uma rodada de consenso ocorre de duas maneiras distintas. Na primeira, o líder da rodada obtém votos suficientes em sua vizinhança, atestando que pelo menos metade de todas as vizinhanças da rede estão em acordo com a proposta feita. Para isso acontecer, também é necessário que cada vizinhança possua pelo menos metade de seus nós em acordo com a proposta. Na segunda maneira ocorre caso o tempo máximo de espera para a votação seja atingido no líder, nesse caso, o líder considera que a votação falhou e avança para a próxima rodada. A falha da votação na vizinhança de um nó que não seja o líder, faz com que esse nó também descarte o bloco proposto e aguarde uma nova proposta. Nesse caso, o nó também deixa de responder às requisições referentes a votação que falhou. Quando isso acontece, o estado da rede pode ser tornar temporariamente inconsistente. Porém essa inconsistência será revertida na próxima vez em que o nó receber uma proposta válida.

Acordo: Como o consenso por localidade é um protocolo com acordo probabilístico, é esperado que o acordo seja alcançado com uma certa probabilidade em cada rodada. No modelo proposto, são definidos diferentes graus de acordo com base no modelo de

vizinhança aplicado na rede. Esses graus de acordo podem ser verificados através do número de nós em acordo no momento em que o limiar de votação é alcançado na vizinhança do líder. Ressalta-se que, após esse limiar ser atingido, o consenso continua sendo propagado na rede, de maneira que, eventualmente, todos os nós da rede sejam alcançados.

4.3. Formação de Vizinhanças

Um componente chave da proposta é a possibilidade de formar vizinhanças que favoreçam as propriedades do consenso em diferentes cenários. Estas regras devem garantir que cada nó pertença a múltiplas vizinhanças, criando assim uma rede sobreposta conectada e que as propostas de votação cheguem a toda a rede. Para garantir a conectividade da rede sobreposta, é definido um critério de tamanho mínimo de vizinhança. O critério é adicionar pelo menos metade dos nós da rede a cada vizinhança. Desta forma, é trivial provar que a rede sobreposta está conectada.

São propostas 4 abordagens para escolhas de vizinhos considerando a conectividade da rede sobreposta: vizinhança com malha completa, vizinhança com valor mínimo (metade da rede), vizinhança com valor mínimo com escolha baseada na coloração do grafo e vizinhança baseada em uma árvore balanceada. A última abordagem, embora não respeite o valor mínimo de vizinhos, forma uma rede sobreposta conexa como consequência da formação de uma árvore de cobertura.

Vizinhança com malha completa: Nesta abordagem cada nó da rede é vizinho de todos os demais nós. Essa abordagem forma vizinhanças robustas, capazes de propagar as votações de maneira mais eficiente, uma vez que cada nó da rede envia a propostas para todos os demais. Por outro lado, essa proposta também é a que apresenta o maior custo em número de mensagens trocadas. A formação de vizinhanças com malhas completas é semelhante a abordagens utilizadas por mecanismos do tipo tolerantes a falhas bizantinas (BFT) como o BFT-SMaRt [Bessani et al. 2014] e o PBFT [Castro et al. 1999], caracterizados pelo alto custo imposto por seus processos de votação [Carrara et al. 2020].

Vizinhança com valor mínimo: Nesta abordagem cada nó da rede possui metade dos demais nós da rede como vizinhos. A escolha dos vizinhos é feita de maneira aleatória para cada nó. Essa abordagem visa reduzir o número de mensagens necessárias para alcançar o consenso em troca da criação de uma rede menos robusta. Essa abordagem possui um grau de acordo reduzido em relação a abordagem de malha completa. A menor conectividade causa o aumento no tempo necessário para difundir as propostas devido ao número de saltos adicionais necessários para alcançar todos os nós.

Vizinhança baseada na coloração do grafo: Esta abordagem é semelhante à anterior, porém a escolha de vizinhos não é feita de maneira aleatória. Para este cenário, é utilizado uma abordagem baseada na coloração do grafo formado pela rede física à qual os nós pertencem. A abordagem é possível no ambiente das redes privadas, como em centro de dados, uma vez que os participantes da rede são conhecidos assim como a topologia da rede. A abordagem utiliza o método de coloração de grafos por conjuntos independentes [Kosowski and Manuszewski 2004]. Nesse método, conjuntos independentes são identificados e removidos do grafo. Para cada conjunto independente identificado é atribuída uma cor. A escolha dos vizinhos de um nó é feita baseada em sua própria cor. Para cada nó, primeiro são escolhidos o máximo possível de vizinhos que possuam

a mesma cor. Caso não haja vizinhos suficientes com a mesma cor do nó, são escolhidos vizinhos das demais cores de maneira alternada. O objetivo desta abordagem é formar vizinhanças de maneira justa. Ao priorizar vizinhos com a mesma cor do nó, a vizinhança resultante prioriza o envio de requisições para os nós mais distantes do nó, uma vez que, pelas propriedades dos algoritmos de coloração de grafos, nós de mesma cor não são vizinhos de primeiro salto na rede física. A principal vantagem dessa abordagem é permitir que novas propostas de votação alcancem primeiro os nós mais distantes em relação ao nó que fez a proposta. Assim, o consenso é disseminado de maneira mais distribuída dando oportunidade para nós mais lentos, ou sujeitos a maior latência, responderem em tempo hábil.

Vizinhança com árvore balanceada: Nesta abordagem, cada nó da rede se conecta a f outros nós. Preenche-se cada vizinhança, começando pelo nó líder, utilizando uma abordagem em largura, formando uma árvore balanceada. A abordagem por árvore busca garantir o acordo do mecanismo de consenso. A Prova 1 demonstra o acordo dentro de uma estrutura de árvore balanceada. Prova-se que é possível assegurar a convergência da votação quando mais de 50% dos nós de cada vizinhança não apresenta falhas.

Teorema 1. *Dada uma vizinhança com estrutura de árvore balanceada, é possível assegurar a convergência da votação quando mais da metade dos nós não apresenta falhas.*

Assume-se uma topologia de árvore com f filhos e as seguintes propriedades:

1. *A árvore é balanceada com fator de balanceamento $B = 1$;*
2. *Cada nó possui $|V_i| <= f$ filhos. V_i é a vizinhança do nó;*
3. *O limiar de consenso local é $L = \frac{|V_i|}{2} + 1$.*

Hipótese: *O consenso é alcançado pela raiz em uma árvore de nível N.*

Base Indutiva: *O consenso é alcançado em uma árvore de altura $N = 0$.*
Em $N = 0$, o nó é uma folha, portanto não tem filhos, e a única aresta de conexão do nó com o grafo de difusão é através da qual ele recebeu a proposta de consenso. Assim, $L = 0$, e o consenso é decidido localmente através da verificação da validade da proposta. O consenso é alcançado nas folhas de uma forma trivial.

Hipótese Indutiva: *O consenso é alcançado em uma árvore com altura $\phi - 1$.*

Passo Indutivo: *Convergência do consenso com uma árvore de altura N = ϕ.*

Cada filho do nó raiz é também a raiz de uma sub-árvore de altura $\phi - 1$. Assim, para provar o acordo no nível ϕ, tem-se dois casos:

Os nós no nível $\phi - 1$ são folhas: *Assim, o consenso é alcançado pela verificação local (caso base).*

Cada nó i no nível $\phi - 1$ é uma sub-árvore: *Cada nó recebeu $\frac{|Vi|}{2}+1$ respostas. Assim, V_i tem no máximo f^{n-1} e pelo menos f^{n-2} votos a favor do consenso.*

Se cada vizinho tem uma sub-árvore de altura $\phi - 1$ e com π_f e π_c sendo o conjunto de nós a favor e contra o consenso, respectivamente, pode-se assumir que cada sub-árvore possui:

$$\sum_{i=0}^{\frac{f}{2}+1} |\pi_f| = \frac{f}{2} + 1 * f^{N-1} = \frac{f^N}{2} + f^{N-1}$$

$$\sum_{i=0}^{\frac{f}{2}} |\pi_c| = \frac{f}{2} * f^{N-1} = \frac{f^N}{2}$$

Assim, tem-se o número de votos a favor e contra o consenso:

$$Q = \sum_{i=0}^{\frac{f}{2}+1} |\pi_f| - \sum_{i=0}^{\frac{f}{2}} |\pi_c|$$

$$Q = (\frac{f^N}{2} + f^{N-1}) - (\frac{f^N}{2}) = f^{N-1}$$

Como $f^{N-1} > 0$, tem-se uma maioria de votos a favor, e o consenso converge na raiz da árvore. ■

4.4. Votação de Propostas

Após a formação das vizinhanças, a rede pode iniciar o processo de validação de blocos. Para isso, o nó líder da rodada cria um bloco candidato contendo as transações que deseja validar. Esse bloco é enviado para cada nó na vizinhança do líder. Ao receber a proposta de votação, cada nó realiza o processo de verificação local através da função externa C. Caso o bloco candidato seja considerado correto pela função, o nó também propaga o bloco candidato para sua própria vizinhança, iniciando a sua votação local. Caso contrário, o bloco é descartado e o nó volta a esperar por novas propostas de votação. O processo de propagação de propostas pode ser visualizado através da formação de uma árvore geradora (Figura 1(a)). Nessa árvore, as arestas representam as propagação de novas propostas através da rede e a raiz representa o líder da rodada de consenso.

Após iniciar uma votação local, um nó aguarda, pelo tempo limite da votação, as respostas de seus vizinhos. Um nó envia confirmações de votação em três momentos. Primeiro, quando recebe confirmações suficientes de sua vizinhança, o nó responde ao nó do qual recebeu originalmente a proposta. Segundo, caso ele receba uma proposta do mesmo bloco candidato enquanto aguarda a convergência de sua votação local, ele responde imediatamente ao nó que enviou essa requisição. Terceiro, caso receba uma requisição após ter sua confirmação enviada, o nó também responde imediatamente.

A convergência do processo de votação ocorre quando todos os nós da rede já receberam pelo menos uma proposta de votação. Nesse momento, todas as propostas propagadas passam a gerar respostas imediatas. Quando isso ocorre, os nós em estado de espera passam a receber respostas e alcançar os limiares de votação locais. Na Figura 1(a), as arestas pontilhadas representam as respostas imediatas formando elos na árvore geradora formada pelo processo de votação. Conforme as votações convergem nos níveis mais baixos da árvore, confirmações são enviadas para os níveis superiores e assim, quando esses níveis convergirem, a votação poderá convergir na raiz da árvore. Quando a votação converge na vizinhança do líder, o consenso é considerado alcançado. Nesse momento, a maioria das vizinhanças da rede já convergiu em sua votação, com isso o acordo da maior parte da rede é garantido. Após isso, outras vizinhanças eventualmente irão convergir em suas votações, entrando em acordo com o restante da rede.

O processo de votação pode falhar caso não haja votos suficientes para atingir o limiar de consenso na vizinhança do líder. Nesse caso, o líder descarta o bloco proposto e avança para a próxima rodada de consenso. Em caso de falha do consenso, algumas vizinhanças já podem ter convergido suas votações e assim ficar em desacordo com o restante da rede. Nesse caso, ao receber a proposta da próxima rodada, esses nós descartam ao bloco atual e substituem pela nova proposta.

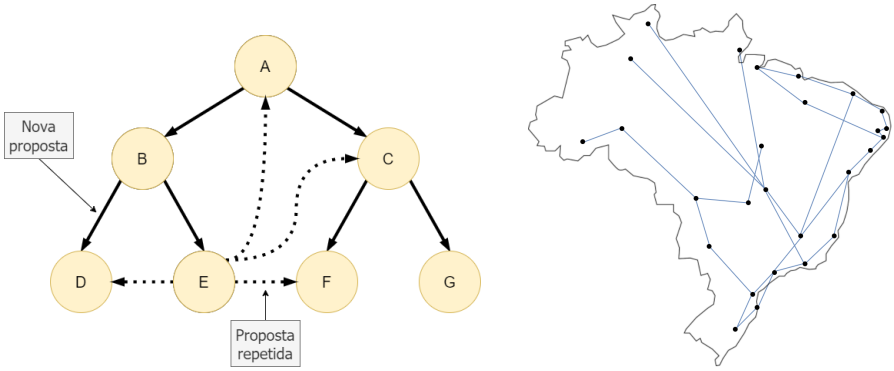
5. Avaliação e Resultados

A avaliação da proposta foi realizada em um simulador orientado a eventos discretos [Oliveira et al. 2020]. O simulador de eventos discretos para a avaliação de plataformas de cadeia de blocos é validado em trabalhos anteriores [Oliveira et al. 2019, Oliveira et al. 2020]. A utilização de uma abordagem orientada a eventos discretos permite simular o comportamento de diferentes mecanismos de consenso de maneira justa e independente do sistema em que o teste é executado. Além das abordagens de malha completa, vizinhança mínima, vizinhança baseada na coloração do grafo e árvore balanceada, foram implementados os seguintes modelos de vizinhança: (i) Consenso na vizinhança da rede física, sem considerar uma rede sobreposta, (ii) o protocolo Raft e (iii) uma árvore balanceada com base na coloração do grafo. Foi estabelecido um limiar de $(n/2) + 1$ nós nas simulações, com n igual ao número de nós em cada vizinhança. Ambas as abordagens de árvore balanceada e de árvore colorida, consideram $f = 6$ filhos por nó. Esse valor foi escolhido empiricamente para as árvores balanceadas após os testes preliminares.

Para avaliar cada abordagem, um grafo representando uma topologia de rede física foi criado utilizando como modelo a topologia da Rede Nacional de Ensino e Pesquisa (RNP)[1], representado na Figura 1(b). Este grafo é composto por 28 nós e os pesos das arestas representam o valor do atraso de propagação entre dois nós da rede. Os valores de atraso foram calculados a partir da distância geográfica entre os nós adjacentes na rede. O cálculo dos menores caminhos entre cada par de nós não adjacentes foi utilizado para definir uma matriz de custos de comunicação. Esses valores foram utilizados para estimar os tempos em cada simulação.

A abordagem de consenso sem considerar uma rede sobreposta consiste na execução do consenso por localidade utilizando somente as vizinhanças contidas no grafo da rede física. As vizinhanças formadas por essa abordagem possuem poucos vizinhos, permitindo assim, que cada vizinhança alcance o limiar de consenso mais facilmente. A segunda abordagem se baseia na execução do mecanismo Raft [Ongaro and Ousterhout 2014]. O Raft é um mecanismo de consenso baseado em votação no qual um nó eleito como líder inicia o processo de votação enviando uma proposta para os demais nós. Cada nó, em seguida, valida a proposta recebida e responde ao líder confirmando o valor proposto. O líder, ao receber respostas suficientes, tipicamente metade dos nós da rede, considera que o consenso foi alcançado confirmando o novo estado. No entanto, uma nova votação só é iniciada quando todos os nós respondem. O Raft é um mecanismo de consenso de baixo custo, porém só é capaz de tolerar falhas de parada (*crash-fault*). A última abordagem visa explorar as vantagens da árvore balanceada e a abordagem com grafo colorido, formando uma árvore balanceada onde cada nó prioriza seus vizinhos mais distantes. Uma árvore balanceada é construída, em que cada

[1]Disponível em `http://www.topology-zoo.org/`



(a) Árvore geradora resultante da propagação de uma nova proposta a partir do nó A através da rede.

(b) Rede Nacional de Ensino e Pesquisa (RNP).

Figura 1. a) Os ramos da árvore representam a primeira vez que a proposta é recebida, iniciando uma votação local. Os elos representam o envio de propostas repetidas pelo nó. b) Os pesos das arestas representam a latência de propagação dos enlaces calculada com base na distância entre nós.

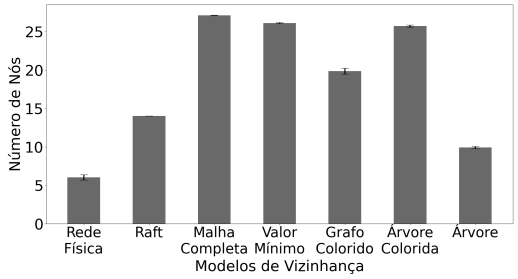
nó prioriza o nó com a mesma cor até atingir 50% da rede como vizinhos.

A avaliação experimental foi executada em duas etapas, primeiro cada abordagem foi testada em um ambiente sem falhas, no qual todos os nós respondem normalmente a requisições de votação. Nesse cenário, mensagens são entregues respeitando o atraso de propagação definido pela rede sobreposta. As seguintes métricas foram avaliadas no experimento: número de nós alcançados pelo consenso no momento em que o limiar de consenso é alcançado, Figura 2(a); número de mensagens necessárias para alcançar o consenso, Figura 2(b); e os tempos para alcançar o limiar de consenso e para o consenso alcançar toda a rede, Figura 2(c). Os valores apresentados compreendem a médias de 1.000 rodadas de teste para cada cenário com intervalos de confiança de 95%.

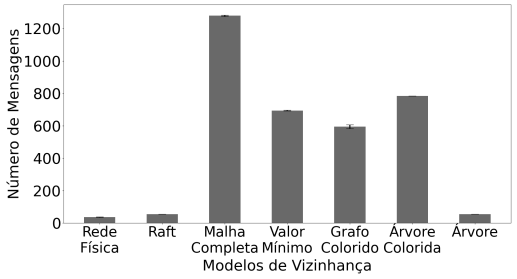
O segundo experimento consiste em avaliar o funcionamento de cada abordagem na presença de falhas segundo o modelo apresentado na Seção 4. A Figura 2(d) apresenta os resultados do experimento. Os valores apresentados são a probabilidade de falha de uma tentativa de alcançar o consenso do ponto de vista do nó que iniciou a votação, em função do número de nós em falha.

Nas Figuras 2(a) e 2(b), verifica-se o compromisso entre o acordo e o custo de terminação dos modelos avaliados, visto que, modelos com maior número de nós alcançados possuem custo de terminação mais elevados. Essa relação é uma consequência do número adicional de mensagens necessárias para alcançar mais nós. O modelo baseado na rede física apresenta o menor tempo de terminação, aproximadamente $12\ ms$, enquanto o modelo de grafo colorido apresenta o maior valor, aproximadamente $50\ ms$. No Raft, o número de nós alcançados é constante, 14 nós, esse valor representa o modelo de votação direta entre o líder e os nós sem propagações adicionais de mensagens. Para a árvore balanceada, o resultado representa a estrutura de árvore formada sem ligações adicionais.

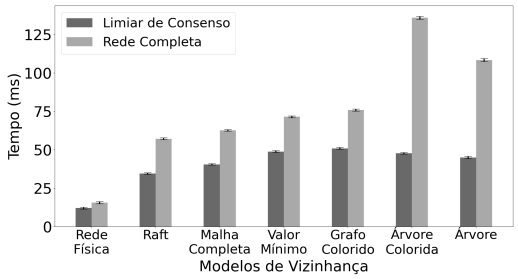
Ao se comparar os modelos propostos para o consenso por localidade pode-se verificar que, entre as abordagens de malha completa e valor mínimo, houve uma queda de aproximadamente 46% no número de mensagens trocadas para alcançar o consenso,



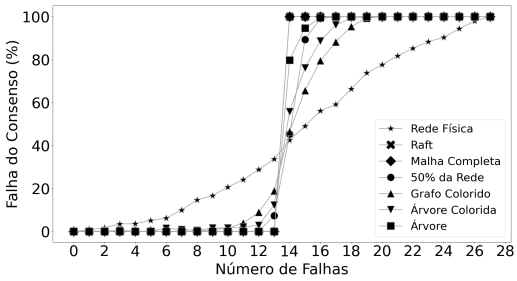
(a) Quantidade de nós da rede alcançados pelo consenso no momento em que uma rodada de consenso é considerada terminada com sucesso.



(b) Número de mensagens trocadas para finalizar uma rodada de consenso.



(c) Medição de tempo para atingir o consenso e para obter a resposta do ultimo nó alcançado.



(d) Probabilidade de falha de uma votação em função do número de nós defeituosos na rede.

Figura 2. Resultados dos testes executados em ambiente simulado.

enquanto a média de nós alcançado foi reduzida de $27,09$ para $26,10$, uma redução de $3,68\%$. Nesse cenário, a maior variação ocorre ao comparar os tempos de terminação. A abordagem de malha completa apresenta a média de $40,46\ ms$, enquanto a abordagem com valor mínimo apresenta uma média de $48,82\ ms$. Nesse caso, também não houve grande impacto na terminação em relação a redução do custo de mensagens.

A abordagem baseada na coloração do grafo apresenta uma redução nos nós alcançados e um aumento do tempo de terminação, $19,85$ nós e $50,88\ ms$. Essa redução relaciona-se ao fato de que cada nó prioriza os nós mais distantes de si, aumentando a latência de comunicação entre cada vizinhança. A redução no custo se mantém proporcional ao número de nós alcançados. Como menos nós foram alcançados, menos mensagens também foram trocadas no processo. A principal vantagem dessa abordagem é observada nos resultados dos testes de tolerância a falhas, Figura 2(d). Nesse teste, a abordagem baseada na coloração do grafo, apresenta a maior probabilidade de sucesso na votação mesmo em cenários em que mais da metade da rede está comprometida. A maior tolerância deve-se à distribuição mais justa de nós entre as vizinhanças, permitindo que nós funcionando corretamente sejam alcançados mesmo na presença de nós defeituosos.

A Figura 2(d) apresenta as probabilidades de falha do consenso em função do número de nós em falha na rede. Observa-se a incapacidade das abordagens baseadas no Raft e em malha completa de alcançar o consenso com o número de nós em falhas acima de 14 nós. A incapacidade é uma consequência do modelo de votação do Raft que sempre espera que metade dos nós da rede respondam. Semelhantemente, a abordagem em malha completa forma vizinhanças contendo todos os nós. Nesse cenário, a convergência da votação exige a resposta de metade dos nós da rede em cada vizinhança, levando cada processo de votação local a se comportar como o Raft. A abordagem em árvore balance-

ada apresenta o mesmo custo de mensagem que a abordagem Raft, 54 mensagens, porém apresenta maior tolerância a falhas. Esta abordagem atinge uma melhor tolerância a falhas porque, em alguns casos, a maioria das falhas ocorre em um ramo da árvore, enquanto os outros ramos ainda mantêm nós suficientes para concluir o processo de votação.

Ao observar os resultados para a abordagem baseada na rede física, observa-se falsa tolerância a falhas. Essa tolerância é alcançada devido ao baixo número de nós que esse consenso alcança (Figura 2(a)). Esse resultado é agravado ainda mais pelo fato de falhas dos nós dessa abordagem facilmente desconectarem a rede, impedindo tanto nós corretos como maliciosos de participar da votação. Como consequência, essa abordagem possui probabilidade falha maior que zero a partir da presença de 2 nós em falha na rede.

6. Conclusão

O projeto de mecanismos de consenso exige que as propriedades de validade, acordo e terminação sejam asseguradas. Enquanto a validade é assegurada, as propriedades de acordo e terminação estão sujeitas a uma relação de compromisso entre si, forçando mecanismos a favorecerem uma em detrimento da outra. Assim, mecanismos de consenso tendem a favorecer o acordo ao custo da terminação e aumentar sua tolerância a falhas ou favorecer a terminação ao custo do acordo e tolerância a falhas. Este artigo apresentou um mecanismo de consenso leve baseado em convergência por localidade. O modelo de convergência por localidade permite que diferentes abordagens de vizinhanças sejam criadas, conferindo ao mecanismo diferentes graus de compromisso entre acordo, terminação e custo de comunicação. Os resultados experimentais mostram que é possível reduzir o custo de comunicação em aproximadamente 46% ao custo de $3,68\%$ de redução no número de nós alcançados ainda sendo capaz de tolerar tipos de falhas mais complexos que falhas de parada. Como trabalhos futuros, pretende-se expandir o modelo de votação do mecanismo através do desenvolvimento de um processo de eleição distribuído e determinístico, reduzindo a necessidade de troca de mensagens para eleger um líder.

7. Agradecimentos

Este trabalho foi realizado com recursos do CNPq, CAPES, FAPERJ, RNP e CGI/FAPESP (Projeto FAPESP 2018/23062-5).

Referências

Bessani, A., Sousa, J., and Alchieri, E. E. (2014). State machine replication for the masses with bft-smart. In *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 355–362. IEEE.

Carrara, G. R., Burle, L. M., Medeiros, D. S., de Albuquerque, C. V. N., and Mattos, D. M. (2020). Consistency, availability, and partition tolerance in blockchain: a survey on the consensus mechanism over peer-to-peer networking. *Annals of Telecommunications*, 75(3):163–174.

Carrara, G. R., Reis, L. H., Albuquerque, C. V., and Mattos, D. M. (2019). A lightweight strategy for reliability of consensus mechanisms based on software defined networks. In *2019 Global Information Infrastructure and Networking Symposium (GIIS)*, pages 1–6. IEEE.

Castro, M., Liskov, B., et al. (1999). Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186.

Chen, L., Xu, L., Shah, N., Gao, Z., Lu, Y., and Shi, W. (2017). On security analysis of proof-of-elapsed-time (poet). In *International Symposium on Stabilization, Safety, and Security of Distributed Systems*, pages 282–297. Springer.

Correia, M., Veronese, G. S., Neves, N. F., and Verissimo, P. (2011). Byzantine consensus in asynchronous message-passing systems: a survey. *International Journal of Critical Computer-Based Systems*, 2(2):141–161.

Douceur, J. R. (2002). The sybil attack. In *International workshop on peer-to-peer systems*, pages 251–260. Springer.

Kosowski, A. and Manuszewski, K. (2004). Classical coloring of graphs. *Contemporary Mathematics*, 352:1–20.

Kwon, J. (2014). Tendermint: Consensus without mining. *Draft v. 0.6, fall*, 1(11).

Lamport, L. (2006). Fast paxos. *Distributed Computing*, 19(2):79–103.

Lamport, L. et al. (2001). Paxos made simple. *ACM Sigact News*, 32(4):18–25.

Lamport, L. and Massa, M. (2004). Cheap paxos. In *International Conference on Dependable Systems and Networks, 2004*, pages 307–314. IEEE.

Lamport, L., Shostak, R., and Pease, M. (1982). The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401.

Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., and Saxena, P. (2016). A secure sharding protocol for open blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 17–30.

Nakamoto, S. (2008). Bitcoin whitepaper. *URL: https://bitcoin. org/bitcoin. pdf*.

Oliveira, M. T., Carrara, G. R., Fernandes, N. C., Albuquerque, C. V., Carrano, R. C., Medeiros, D. S., and Mattos, D. M. (2019). Towards a performance evaluation of private blockchain frameworks using a realistic workload. In *2019 22nd conference on innovation in clouds, internet and networks and workshops (ICIN)*, pages 180–187. IEEE.

Oliveira, M. T., Reis, L. H. A., Medeiros, D. S. V., Carrano, R. C., Olabarriaga, S. D., and Mattos, D. M. F. (2020). Blockchain reputation-based consensus: A scalable and resilient mechanism for distributed mistrusting applications. *Comput. Networks*, 179:107367.

Ongaro, D. and Ousterhout, J. (2014). In search of an understandable consensus algorithm. In *2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14)*, pages 305–319.

Rebello, G. A. F., Camilo, G. F., Guimarães, L. C., de Souza, L. A. C., and Duarte, O. C. M. (2020). On the security and performance of proof-based consensus protocols. In *2020 4th Conference on Cloud and Internet of Things (CIoT)*, pages 67–74. IEEE.

Schwartz, D., Youngs, N., Britto, A., et al. (2014). The ripple protocol consensus algorithm. *Ripple Labs Inc White Paper*, 5(8):151.