Uma Estratégia Leve para a Confiabilidade de Mecanismos de Consenso baseada em Redes Definidas por Software

Gabriel R. Carrara, Lúcio H. A. Reis, Célio V. N. Albuquerque, Diogo M. F. Mattos
Laboratório MídiaCom – Universidade Federal Fluminense (UFF)
Niterói/RJ – Brasil

***Resumo*— Aplicações de cadeias de blocos para redes privadas permitem o uso de mecanismos de consenso baseados em votação, pois possuem menor custo computacional quando comparados à prova de trabalho. Contudo, mecanismos de consenso baseados em votação exigem uma elevada troca de mensagens na rede, já que o número de mensagens cresce com o aumento do número de participantes. Este artigo propõe a estratégia de uso de técnicas de Qualidade de Serviço (QoS) para garantir o funcionamento confiável dos mecanismos de consenso baseados em votação. A estratégia proposta é baseada em Redes definidas por *Software* (SDN) e visa reduzir o tempo para a terminação dos protocolos de consenso. A avaliação da proposta foi realizada através de emulação em Mininet, considerando os mecanismos de consenso Raft e BFT-SMaRt. Os resultados mostram que a proposta garante a terminação dos protocolos em tempo reduzido e a diminuição da carga de controle para eleição de líderes, já que assegura taxa mínima no encaminhamento dos fluxos de consenso.**

***Palavras-Chave*— Redes definidas por software, consenso, Qualidade de Serviço.**

***Abstract*— Blockchain usage in private networks allows the deployment of consensus mechanisms based on voting. Such mechanisms present lower computational cost when compared to proof-of-work. Voting-based consensus mechanisms require a high level of message exchange on the network, and the number of messages is proportional to the number of participants. This paper proposes a strategy for applying Quality of Service (QoS) techniques to ensure the reliable operation of voting mechanisms based on consensus. The proposed strategy is based on Software Defined Networking (SDN) and aims to reduce the time for the termination of the consensus protocols. The evaluation of the proposal was done through emulation using Mininet, considering the Raft and BFT-SMaRt consensus mechanisms. Results show that the proposal guarantees the termination of the protocols in a reduced time and the reduction of the control load for the election of leaders since it assures a minimum rate in the forwarding of the consensus flows.**

***Keywords*— Software-defined networking, Consensus, Quality of Service.**

I. Introdução

Aplicações baseadas em cadeias de blocos utilizam a natureza distribuída das redes par-a-par para atingir um grande número de usuários e expandir seu alcance. Aplicações comuns de cadeias de blocos, como a Bitcoin [1] e a Ethereum [2], são redes abertas ao público e amplamente utilizadas para transações financeiras. Outras aplicações de destaque executam em redes privadas, em que empresas podem criar e administrar seus próprios ativos e redes para trocar informações ou realizar transações e, assim, interagir de maneira segura, controlada e com confiança distribuída [3]. Alguns exemplos de plataformas que permitem a criação de redes privadas são a Hyperledger Fabric[1] e a R3 Corda[2]. Devido à natureza distribuída, essas plataformas necessitam de mecanismos de consenso capazes de manter as informações atualizadas em todos os participantes da rede. Contudo, o aumento do número de participantes, assim como o aumento do alcance da rede podem causar um aumento no atraso da divulgação das mensagens da rede prejudicando a terminação dos mecanismos de consenso. Em redes privadas, o uso de mecanismos de consenso baseados em votação as torna mais sensíveis a seu próprio crescimento, já que exige a troca de um número de mensagens elevado entre os participantes.

Os mecanismos de consenso podem ser classificados em baseados em competição e baseados em votação. A primeira categoria é normalmente aplicada em redes públicas em que não se conhece o número total de participantes, ou o número de participantes é altamente variável, sendo impossível se obter confiança [4]. Os mecanismos baseados em votação são aplicados em ambientes de redes privadas e colaborativos. Nestes ambientes os participantes da rede são conhecidos e a rede é administrada pelas partes interessadas em sua utilização. Dessa maneira, os nós na rede têm sua identidade garantida e a entrada de novos participantes é controlada.

O uso de mecanismos de consenso baseados em votação impõe uso adicional da comunicação entre os nós da rede. Nesse modelo de mecanismos é necessária troca frequente de mensagens para a realização de tarefas como a eleição de um líder ou a atualização do estado dos dados da rede. A necessidade frequente de troca de mensagens aumenta a sensibilidade desses mecanismos de consenso ao estado da rede. Problemas como congestionamento ou gargalos em certos pontos da rede podem resultar na degradação do funcionamento desses mecanismos prejudicando sua terminação. A terminação de um mecanismo de consenso baseado em votação pode ser definida como o momento em que um participante do consenso obtém a visão atual do estado da rede.

Este artigo propõe uma estratégia de aplicação de técnicas de Qualidade de Serviço (QoS) para garantir o funcionamento de maneira confiável dos mecanismos de consenso baseados

[1]Disponível em http://www.hyperledger.org.
[2]Disponível em http://www.corda.net/.

em votação. Essa estratégia faz uso de redes definidas por *software* (*Software Defined Networking* - SDN) aplicadas como uma maneira eficiente e flexível com o intuito de reduzir o tempo de terminação e garantir o funcionamento dos mecanismos de consenso através da criação de filas e fornecimento de banda mínimo. O paradigma de redes definidas por *software* desacopla o plano de dados distribuído do plano de controle logicamente centralizado. Nesse paradigma, a definição de políticas restringe-se ao plano de controle e, posteriormente, são replicadas no plano de dados de acordo com os recursos disponíveis, como filas e limitação de banda [5].

O restante deste artigo está disposto da seguinte maneira, a Seção 2 apresenta trabalhos relacionados. Na Seção 3 serão discutidos os requisitos de rede para mecanismos de consenso. A Seção 4 apresenta a estratégia que foi utilizada para prover confiabilidade aos mecanismos de consenso. O procedimento de avaliação e seus resultados são apresentados na Seção 5. Por fim conclusão e trabalhos futuros serão discutidos na Seção 6.

II. Trabalhos Relacionados

As redes definidas por *software* permitem a implantação de regras que em redes legadas exigem alterações em seu núcleo, tais como regras para a reserva de recursos. Dessa forma, diversas aplicações são propostas visando fornecer qualidade de serviço, como a criação de regras específicas para alocação de banda em vídeo conferencia [6]. Mattos e Duarte propõem o QFlow [7], um mecanismo capaz de prover isolamento de recurso e qualidade de serviço em redes virtuais, no ambiente de SDN, com controle eficiente dos recursos disponíveis. O QFlow ainda permite o mapeamento dos parâmetros de Qualidade de Serviço, definidos para cada rede virtual, em recursos do plano de dados.

Para facilitar e automatizar a configuração e o provimento de qualidade de serviço utilizando SDN, Bari *et al.* propõem o PolicyCop [8], um arcabouço com uma estrutura de aplicação de política de QoS autônoma. O PolicyCop fornece uma interface para especificar políticas de qualidade de serviço além de explorar a API *northbound* do controlador SDN. O PolicyCop também se aproveita de aplicativos de controle para monitorar a conformidade das políticas e adaptar autonomicamente as regras do plano de controle às condições de tráfego. Tomovic *et al.* propõem outro arcabouço que apresenta um ambiente de controle SDN/OpenFlow capaz de fornecer garantias de largura de banda para fluxos prioritários [9]. Esses fluxos e seus requisitos precisam ser especificados para que o controlador possa calcular rotas e reservar recursos. Para proteger o tráfego de melhor esforço, ao invés de aplicar o roteamento de caminho mais curto, um novo algoritmo que leva em conta a utilização de recursos é proposto.

O paradigma de redes definidas por *software* possui a capacidade de escalar o plano de controle logicamente centralizado de acordo com a demanda da rede, ao passo que implementa os controladores fisicamente distribuídos. Diversos trabalhos propõem a aplicação de um plano de controle distribuído nas redes SDN [10]. Tal abordagem exige que informações necessárias para a tomada de decisão nos controladores sejam distribuídas e sincronizadas de maneira consistente entre os controladores. Mattos *et al.* propõem uma arquitetura para plano de dados distribuídos capaz de fornecer um modelo de comunicação entre múltiplos controladores distribuídos fisicamente, permitindo o compartilhamento de suas visões locais da rede e criando assim uma visão global compartilhada entre todos os controladores [11]. Entretanto essa abordagem ainda possui um certo grau de centralização, pois um controlador deve ser designado para armazenar e gerenciar as informações globais sobre os recursos da rede.

Por fim Sharma *et al.* propõem o *DistBlockNet* [12] que consiste na utilização das redes SDN para permitir a criação de regras de controle flexíveis e facilitar o monitoramento das redes de dispositivos de internet das coisas. Para facilitar a instalação e disseminação de novas regras através dos controladores distribuídos pela rede é utilizada uma estrutura de cadeia de blocos em que as informações sobre regras de fluxos são mantidas atualizadas. Dessa maneira, sempre que um usuário necessita verificar a integridade ou buscar atualizações de regras, elas estão disponíveis na cadeia, permitindo que essas regras sejam acessadas de maneira rápida e segura, reduzindo o custo de manutenção das regras da rede.

III. Requisitos de rede para mecanismos de consenso

Os mecanismos de consenso são a responsáveis pela replicação e pela manutenção da visão do estado da rede. No ambiente das cadeias de blocos os mecanismos de consenso exercem papel fundamental para garantir que o estado atual da cadeia esteja acessível para todos os participantes. As cadeias de blocos podem ser classificadas em não permissionadas e permissionadas, cada um desses tipos de cadeias possui requisitos diferentes para alcançarem o consenso [4].

Nas cadeias de blocos públicas não há controle sobre o acesso dos participantes à rede, logo não há confiança entre os participantes. Por esse motivo é necessário a implementação de mecanismos de consenso baseados em provas de empenho de recursos computacionais, em que os participantes do consenso empenham grandes quantidades de recurso computacionais de maneira a desestimular a aplicação de fraudes, evitar a criação de identidades falsas e permitir que a resposta seja facilmente validada pelo restante da rede.

As cadeias de blocos privadas normalmente são formadas por participantes de algum tipo de negócio ou consórcio. Nesses casos existe controle de acesso dos participantes da rede, sendo todos conhecidos. O funcionamento da rede também é administrado pelas partes interessadas na manutenção da correção das transações realizadas na cadeia, facilitando a resolução de problemas em caso de falhas. Nessas cadeias normalmente são utilizados mecanismos de consenso baseados em votação que não exigem empenho de recursos computacionais, ao custo de exigir maior número de troca de mensagens.

Os mecanismos de consenso baseados em prova normalmente fazem uso de recursos computacionais para resolver algum tipo de prova ou quebra-cabeça. Como exemplos desses mecanismos podem ser citados a Prova de Trabalho (*Proof-of-Work*, PoW) [1] e a prova de participação (*Proof-of-Stake*, PoS)[13]. Nesses mecanismos os participantes devem dispor

de grandes quantidades de recursos computacionais para se tornarem os responsáveis pela realização do consenso e, em troca, recebem algum tipo de recompensa. Esse tipo de mecanismo possui a característica de que os participantes apenas se comunicam entre si para divulgar seus resultados, gerando assim um número reduzido de troca de mensagens, o que as torna mais resilientes a problemas na rede.

Para os mecanismos baseados em votação não é necessária a disposição de recursos computacionais, porém necessitam de comunicação frequente. Sempre que há uma proposta de mudança na cadeia, é necessário que haja um líder encarregado de validá-la e enviá-la para os demais participantes, tornando-se necessário que todos os participantes se comuniquem com esse líder. Assim, no caso de haver um gargalo de encaminhamento próximo ao líder, muitas mensagens são perdidas e o mecanismo não será capaz de exercer sua função. Por esses motivos os mecanismos baseados em votação são mais sensíveis a variações da carga da rede.

As plataformas de cadeia de blocos dependem do uso de mecanismos de consenso para garantirem que os dados são corretamente replicados em todos os participantes. Os mecanismos de consenso garantem terminação e segurança. A terminação consiste na garantia de que o mecanismo de consenso não bloqueia. A segurança refere-se ao mecanismo impedir que dois nós convirjam para valores de consenso distintos. Em caso de falha do mecanismo, as informações na cadeia de blocos têm a integridade comprometida. Plataformas de cadeias de blocos privadas aplicam mecanismos de consenso baseados em votação.

Neste trabalho serão avaliados o desempenho dos mecanismos de consenso *Raft* e *BFT-SMaRt* e o impacto causado pela aplicação da estratégia proposta em seu funcionamento. O *Raft* [14] é um mecanismo de consenso derivado do Paxos [15], baseado em votação que fornece um protocolo tolerante a falhas. Seu funcionamento depende é baseado na eleição de um líder responsável pelo recebimento e ordenação das requisições de alteração da rede. Nós que não exercem o papel de líder são denominados seguidores. O líder deve frequentemente informar seus seguidores da sua presença. Caso isso não ocorra, um dos seguidores inicia um processo de eleição de um novo líder. Na presença de um líder ativo os seguidores podem enviar as requisições de alteração de estado da rede que são armazenados pelo líder. Quando um certo número de requisições é atingido o líder decide quais e em qual ordem elas serão executadas. Após a decisão o líder envia aos seguidores essa informação. Cada seguidor por sua vez executa as operações segundo a ordem definida atualizando assim sua visão do estado da rede. Para que todos os seguidores possuam a mesma visão da rede ao final desta operação é necessário que apenas operações deterministas sejam executadas.

O BFT-SMaRt [16] é uma implementação robusta e leve de um mecanismo de replicação de estado tolerante a falhas bizantinas. Seu funcionamento é similar ao do *Raft* porém a escolha da liderança é feita segundo a lista de nós que cada réplica possui e não de maneira aleatória como o *Raft*. Outra diferença está na maneira como os seguidores realizam e as propostas de mudança da rede. Nesse caso cliente devem enviar requisições para cada réplica. Quando o líder comunica

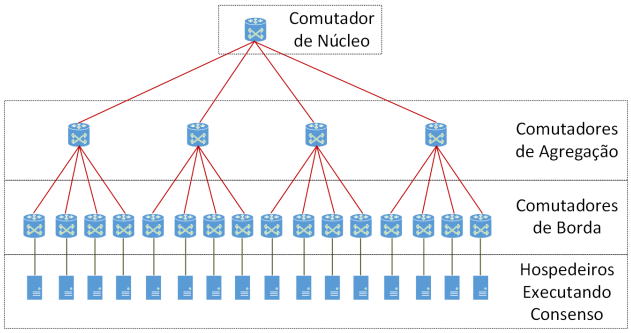


Fig. 1. Topologia de árvore típica de *Datacenters*. As folhas da árvore contêm os nós hospedeiros responsáveis por executar os protocolos de consenso.

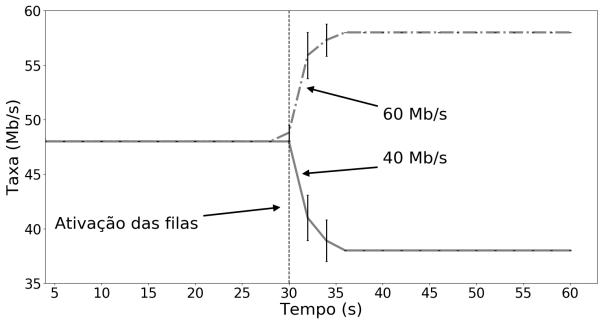


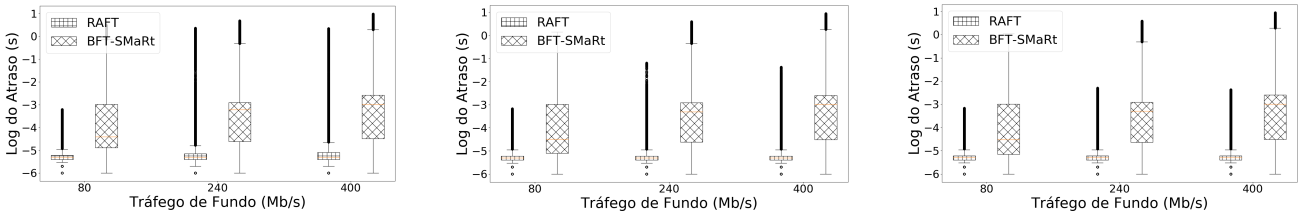
Fig. 2. Teste de validação das regras criadas pelo controlador. Fluxos distintos são transmitidos em uma rede composta por dois hospedeiros e um comutador. Nos primeiros 30 segundos os fluxos disputam os recursos da rede. Aos 30s políticas são instaladas no controlador criando filas distintas para os fluxos.

quais requisições devem ser executadas, os seguidores somente executam as que foram recebidas do cliente. Caso uma requisição não seja executada por um determinado período de tempo, aquele seguidor se considera desatualizado e requisita a versão mais atual dos dados da rede para seus vizinhos.

IV. Estratégia para prover confiabilidade ao consenso

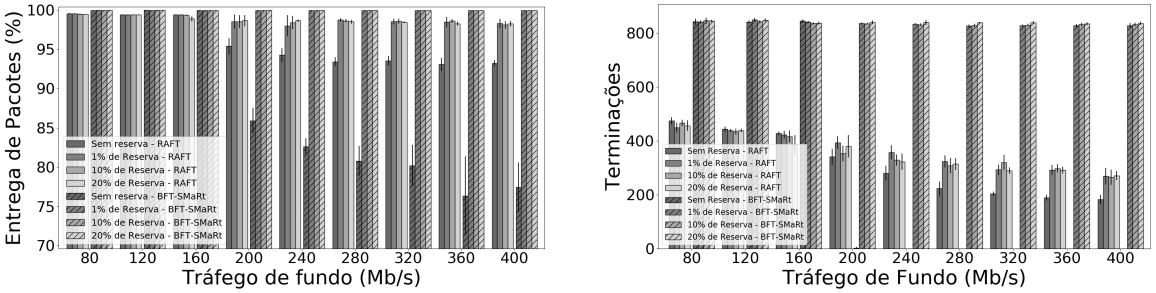
A estratégia proposta neste artigo visa a divisão de recursos entre os diferentes fluxos da rede definidas através de políticas de reserva. Essas políticas foram definidas de maneira empírica e traduzidas na criação de filas com reservas de recursos em cada dispositivo de encaminhamento. Assim quando o controlador cria um novo fluxo ele decide baseado na política definida para qual fila destiná-lo. Portanto para fornecer qualidade de serviço aos mecanismos de consenso, fluxos gerados por eles são alocados em filas com maior prioridade.

Esta estratégia utiliza o mecanismo de criação de filas disponibilizado pelo protocolo *OpenFlow* a partir da versão 1.3. Esse mecanismo possibilita que filas sejam criadas nas portas de saída de cada comutador da rede. Para cada fila criada podem ser definidas restrições de recursos, como por exemplo alocação de banda, na forma de limites máximos e mínimos. Para garantir o funcionamento dos mecanismos de consenso são criadas em cada comutador da rede duas filas. Na primeira fila são alocados fluxos que não pertençam ao mecanismo de consenso e ela é configurada com o limite máximo de uso de banda. A segunda fila destina-se para a



(a) Sem reserva de banda (b) Reserva de 1 Mb/s (c) Reserva de 20 Mb/s

Fig. 3. Resultados dos testes para medir o tempo de fila dos pacotes de cada mecanismo.



(a) Percentual de entrega de pacotes em cada cenário de teste com diferentes níveis de tráfego de fundo e reserva de recursos.

(b) Número de terminações dos mecanismos de consenso com diferentes níveis de tráfego de fundo e reserva de recurso.

Fig. 4. Resultados dos mecanismos para percentual de entrega de pacotes e número de terminações do protocolo.

alocação de fluxos pertencentes ao mecanismo de consenso e para essa fila é definido um limite mínimo de uso de banda. Dessa maneira, caso haja congestionamento ou gargalos na rede, os comutadores serão capazes de garantir banda mínima para os fluxos do mecanismo de consenso.

Este trabalho foca em uma topologia de rede típica de um centro de dados (*datacenter*), em que estações trocam constantemente informações entre si e existem aplicações que possuem o requisito de manter seus dados sincronizados e consistentes. Para tanto, empregam mecanismos de consenso. A topologia considerada é uma árvore. Essa topologia pode ser vista na Figura 1 e é dividida em três níveis hierárquicos. No primeiro nível se encontram os hospedeiros responsáveis por executar as aplicações do *Datacenter*, no segundo nível se encontram os comutadores de borda responsáveis por realizar a conexão dos hospedeiros com o restante da rede, no nível acima estão os comutadores de agregação que reúnem os dados da borda e definem para onde devem ser enviados e por fim o comutador de núcleo é responsável por conectar a topologia e agir como um *gateway* para a rede. Neste trabalho a função de *gateway* do comutador de núcleo foi desconsiderada pois o modelo de comunicação utilizado é somente entre os nós da rede.

V. Avaliação e Resultados

Para realizar a avaliação da estratégia proposta foi criada em um ambiente emulado com o auxílio do *Mininet* a topologia discutida na Figura 1. Para isso foi utilizado um servidor de virtualização com processador Intel i7 de 4ª geração com 3.4GHz de processamento e 16GB de memória *RAM*. Nele foi criada uma máquina virtual com 4 núcleos de processamento e 8GB de memória *RAM* com o sistema operacional Linux Ubuntu 16.04 LTS. Para a captura dos pacotes da rede foi utilizada a ferramenta *tcpdump*. O controlador *Ryu* foi utilizado para traduzir as políticas de reservas de recursos em regras para o plano de dados e permitir a criação das filas. Para isso, foi utilizada a *API REST* disponibilizada pelo controlador que possibilita a criação de filas gerenciadas pelo algoritmo *Hierarchical Token Bucket* nas portas de saída dos comutadores da rede. O controlador também é responsável por associar fluxos a ações de saída para cada fila correta através do protocolo *OpenFlow*. O critério utilizado para a classificar os fluxos se baseou no valor da porta utilizada pelo mecanismo de consenso, uma vez que esta é definida estaticamente. Dessa maneira, em cada cenário foram criadas duas filas, uma destinada para os pacotes enviados pelos mecanismos com garantia de taxa de transferência mínima e outra para os demais pacotes sem garantias de taxa.

Para validar a capacidade de aplicação das políticas através das filas foi realizado um teste preliminar onde, com o auxílio da ferramenta *Iperf*, dois fluxos de 100Mb/s são enviados por uma rota com 100Mb/s de vazão. A Figura 2 apresenta os resultados desta avaliação. No instante 30s são instaladas as políticas no controlador, que são imediatamente traduzidas em filas com diferentes limites de vazão, dos instantes 30s a 36s os fluxos se ajustam aos limites impostos e até o instante 60s os fluxos respeitam os limites. Esse resultado mostra a capacidade do controlador de criar e dos comutador em respeitar as regras de qualidade de serviço.

Para avaliar o impacto da estratégia sobre os mecanismos de consenso foram escolhidos cenários onde a rede apresenta diferentes níveis de congestionamento. Levando em conta enlaces com capacidade de 100Mb/s, em cada cenário cada estação executou o mecanismo de consenso e um tráfego de fundo, criado com o auxílio da ferramenta *Iperf* com taxas de transferência variando de 10Mb/s a 50Mb/s. Cada taxa foi testada com e sem a utilização das políticas. Além disso, foram definidos diferentes valores de reserva de recursos com o

intuito de definir a melhor configuração de reserva de recursos para cada mecanismo. Cada rodada de teste seguiu os seguintes passos, primeiro a topologia emulada é criada com o auxílio do *Mininet*, em seguida as políticas de qualidade de serviço são instaladas no controlador e imediatamente traduzidas para os comutadores da rede. Após a instalação das regras cada estação conectada a uma folha da árvore inicia o envio de seu tráfego de fundo, esse tráfego é baseado no modelo de tráfego interno *Scatter-Gather* encontrado em *Datacenters* onde cada estação realiza o envio e/ou recebimento de grandes quantidades de dados de outras estações em pontos distantes da rede [17]. Simultaneamente ao envio do tráfego de fundo são iniciadas as instâncias do mecanismo de consenso em cada estação. Ao ser iniciada, cada instância do mecanismo de consenso busca realizar uma conexão com cada um de seus vizinhos para formar uma rede par-a-par. Quando o número mínimo de participantes da rede conectados é alcançado os mecanismos começam a enviar suas requisições de mudança de valores.

A frequência de envio de novas requisições segue o modelo usado em [4] que se baseia na frequência de chegada de novas transações da rede Bitcoin. Cada cenário de testes foi repetido 10 vezes. O objetivo dessa avaliação foi comparar o funcionamento dos mecanismos com o cenário sem reservas de recursos. Nas Figura 3 pode-se observar os tempos de fila de ambos mecanismos para os cenários sem reserva de recursos (Figura 3(a)), com 1Mb/s (Figura 3(b)) e com 20Mb/s de reserva (Figura 3(c)). Para o *Raft* todos os cenários apresentam concentrações semelhantes de tempos de fila dos pacotes. Esse resultado demonstra que as reservas de recursos não causaram nenhum impacto negativos no enfileiramento dos pacotes e que os recursos reservados foram suficientes para que as mensagens fossem entregues sem atrasos adicionais.

Em seguida foram avaliados (i) o impacto da estratégia sobre a quantidade de pacotes entregues e (ii) o número de terminações do protocolo. A Figura 4(a) apresenta os valores percentuais dos pacotes entregues por cada mecanismo. Pode-se observar que tanto o *Raft* quanto o *BFT-SMaRt* demonstraram maiores taxas de entrega de pacotes nos cenários com reservas de recursos, essa ganho também pode ser observada no número de terminações bem sucedidas em ambos os casos (Figura 4(b)). Esses resultados demonstram que a estratégia utilizada foi efetiva mesmo nos cenários com apenas 1% de reserva de recursos. A diferença de comportamento entre os mecanismos ao serem aplicadas as estratégias demonstra uma necessidade de análise mais profunda do funcionamento de ambos.

VI. Conclusão

As plataformas de cadeias de blocos dependem dos protocolos de consenso para funcionar de maneira correta. Os protocolos baseados em votação são especialmente sensíveis a problemas na rede pois necessitam trocar um grande número de mensagens para realizar suas funções. Este trabalho teve como objetivo desenvolver uma estratégia para prover qualidade de serviço para protocolos de consenso baseados em votação através da alocação de recursos da rede. Para aplicar a estratégia foi usada a capacidade das redes definidas por *software* de traduzir políticas em regras aplicadas diretamente no plano de dados através da criação de filas exclusivas para os mecanismos de consenso. Para avaliar a proposta os mecanismos *Raft* e *BFT-SMaRt* foram testados em cenários com diferentes graus de congestionamento e reservas de recursos.

Os resultados mostram que ambos os protocolos se beneficiaram da aplicação da estratégia demonstrando assim sua efetividade. Como trabalho futuro resta analisar de maneira mais profunda o funcionamento dos protocolos para entender seu comportamento e permitir a criação de regras específicas para cada um e assim obter maior impacto sobre seu funcionamento.

Agradecimentos

Este trabalho foi realizado com recursos do CNPq, CAPES, CGI/FAPESP, FAPERJ e TAESA.

Referências

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[3] M. T. Oliveira, G. R. Carrara, N. C. Fernandes, C. V. N. Albuquerque, R. C. Carrano, D. S. V. Medeiros, and D. M. F. Mattos, "Towards a performance evaluation of private blockchain frameworks using a realistic workload," in *22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, Feb 2019, pp. 180–187.

[4] M. T. Oliveira, G. R. Carrara, N. C. Fernandes, C. Albuquerque, R. Carrano, D. S. Medeiros, and D. Mattos, "Uma avaliação de desempenho de cadeias de blocos privadas permissionadas através de cargas de trabalho realísticas," in *XVIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, 2018, pp. 309–322.

[5] S. Sezer *et al.*, "Are we ready for sdn? implementation challenges for software-defined networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 36–43, 2013.

[6] S.-N. Yang, S.-W. Ho, Y.-B. Lin, and C.-H. Gan, "A multi-rat bandwidth aggregation mechanism with software-defined networking," *Journal of Network and Computer Applications*, vol. 61, pp. 189–198, 2016.

[7] D. M. F. Mattos and O. Duarte, "Qflow: Um sistema com garantia de isolamento e oferta de qualidade de serviço para redes virtualizadas," *XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuıdos*, 2012.

[8] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, "Policycop: An autonomic qos policy enforcement framework for software defined networks," in *Future Networks and Services (SDN4FNS), 2013 IEEE SDN For*. IEEE, 2013, pp. 1–7.

[9] S. Tomovic, N. Prasad, and I. Radusinovic, "Sdn control framework for qos provisioning," in *22nd Telecommunications Forum (TELFOR)*. IEEE, 2014, pp. 111–114.

[10] Y. Zhang, L. Cui, W. Wang, and Y. Zhang, "A survey on software defined networking with multiple controllers," *Journal of Network and Computer Applications*, vol. 103, pp. 101–118, 2018.

[11] D. M. Mattos, O. C. M. Duarte, and G. Pujolle, "A resilient distributed controller for software defined networking," in *IEEE International Conference on Communications (ICC)*. IEEE, 2016.

[12] P. K. Sharma, S. Singh, Y.-S. Jeong, and J. H. Park, "Distblocknet: A distributed blockchains-based secure sdn architecture for iot networks," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 78–85, 2017.

[13] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," *self-published paper, August*, vol. 19, 2012.

[14] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm." in *USENIX Technical Conference*, 2014, pp. 305–319.

[15] L. Lamport, "The part-time parliament," *ACM Transactions on Computer Systems (TOCS)*, vol. 16, no. 2, pp. 133–169, 1998.

[16] J. Sousa, E. Alchieri, and A. Bessani, "State machine replication for the masses with bft-smart," 2013.

[17] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of data center traffic: measurements & analysis," in *9th ACM SIGCOMM Conference on Internet Measurement*, 2009, pp. 202–208.