Towards Blockchain for Suitable Efficiency and Data Integrity of IoT Ecosystem Transactions

C. Rodrigues, and V. Rocha

***Abstract*—This article analyzes the effectiveness of deploying the Blockchain technology in the implementation of the IoT ecosystem database. To this end, we assess the processing efficiency of transactions originated by smart devices and the stored-data integrity. The processing-efficiency evaluation is carried out through queue-theory-based analytical modeling, in which the average time for transaction confirmation is estimated. By its turn, the data-integrity is measured through simulations, where the probability of fraudsters altering already-stored data is estimated. Moreover, the experiments consider a set of scenarios related to different application domains. Final results show that the Blockchain technology may meet IoT efficiency requirements, besides providing adequate data integrity. Lastly, general conclusions and avenues for further research close this article.**

***Index Terms*—Blockchain, IoT, Efficiency, Data Integrity.**

I. Introdução

O ecossistema Internet das Coisas (do inglês, *Internet of Things* - IoT) resulta da integração de infraestruturas de redes de comunicação, protocolos de *software*, dados, aplicações e dispositivos inteligentes [1], [2]. Devido à concepção distribuída e descentralizada da base de dados, a garantia de eficiência e segurança constitui um desafio [3].

A eficiência se refere à celeridade de processamento dos dados, e a segurança contempla disponibilidade, integridade e confidencialidade de dados [4], [5]. Para atender a esses requisitos, a tecnologia de registros distribuídos *Blockchain* [6], [7] é vista como uma promissora abordagem [2], [8].

Neste contexto, o presente artigo tem o objetivo de analisar a efetividade do emprego da tecnologia *Blockchain* na implementação da base de dados do ecossistema IoT. Para tanto, são avaliadas a eficiência do processamento das transações provenientes de dispositivos inteligentes, bem como a integridade dos dados armazenados na base de dados.

A avaliação da eficiência é desenvolvida usando modelagem analítica com base na teoria das filas, onde é estimado o tempo médio de confirmação de transações. A avaliação da integridade é implementada por simulações, onde é computada a probabilidade de fraudadores alterarem a base de dados. A principal contribuição deste artigo é, portanto, prover novos subsídios para o desenvolvimento de aplicações IoT, considerando a base de dados implementada com *Blockchain*.

O restante deste artigo é organizado como segue. A Seção II revisa a tecnologia *Blockchain*. Na Seção III, apresenta-se o ecossistema IoT sob uma visão sintética de arquitetura em camadas. A Seção IV discorre sobre trabalhos relacionados. A Seção V traz os experimentos, resultados e análises. Por fim, conclusões gerais e trabalhos futuros constituem a Seção VI.

Carlo K. da S. Rodrigues e V. E. M. Rocha, Centro de Matemática, Computação e Cognição, Universidade Federal do ABC, Santo André - Brasil (e-mails: carlo.kleber@ufabc.edu.br e vladimir.rocha@ufabc.edu.br).

II. Bases da Tecnologia Blockchain

O objetivo da tecnologia *Blockchain* é a construção descentralizada de uma base de dados distribuída a partir de uma lista encadeada de blocos de transações de clientes, observando atributos de segurança providos por criptografia [6].

Para a submissão de transações, cada cliente tem duas chaves criptográficas: uma privada, que permite-lhe assinar a transação (i.e., autenticá-la), e uma pública, que permite ao sistema confirmar a autoria. As transações são submetidas pelos próprios clientes a uma rede de processadores, interligados sob arquitetura *peer-to-peer* (P2P). Os processadores trabalham coletando transações, construindo blocos, validando blocos e interligando-os à lista encadeada. Cada bloco é constituído por um conjunto de transações mais um cabeçalho. O *hash* do cabeçalho é usado para a identificação do bloco e possui os três grupos de metadados a seguir [2], [8].

O primeiro grupo tem a versão do protocolo de gerência da base de dados e o *hash* do cabeçalho do bloco anterior na lista encadeada (i.e., bloco *pai*). Cada bloco contém, portanto, o *hash* do cabeçalho de seu *pai* dentro de seu próprio cabeçalho. O algoritmo de *hash* é o SHA-256 [9]. O segundo grupo traz o instante de criação do bloco, t_0, o *alvo de dificuldade* do bloco, D, e o *golden nonce*. Estes dois últimos parâmetros são explicados mais adiante. Por fim, o terceiro grupo contém o resumo das transações armazenadas no bloco, computado pela raiz da árvore de Merkle dessas transações.

Sendo uma lista encadeada, cada bloco referencia então apenas o seu *pai*. Para tanto, usa-se o *hash* do cabeçalho do *pai* que, como explicado antes, está contido dentro do próprio cabeçalho do bloco. A sequência de *hashes* que liga cada bloco ao seu *pai* estabelece o caminho de volta até o primeiro bloco da lista, denominado de bloco *gênese*, como ilustrado na Fig. 1, onde i e $z > 3$ são números inteiros. A mudança de identificação de um bloco da lista gera o efeito cascata da mudança de identificação dos blocos subsequentes.

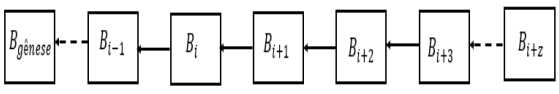

Fig. 1. Lista encadeada de blocos de transações.

A validação do bloco a adicionar à lista é implementada por um processo matemático denominado de *mineração*. O objetivo é encontrar a *prova de trabalho* (do inglês, *proof*

of work - PoW [10]) do bloco. Matematicamente, isso significa determinar por tentativas sucessivas um valor *nonce* que satisfaz a desigualdade expressa em (1), onde *head* é o cabeçalho do bloco a ser *minerado* e, como visto, D é o *alvo de dificuldade*, sendo referente à *mineração* do bloco [11].

$$\text{SHA-256}(\text{SHA-256}(head + nonce)) < D \qquad (1)$$

O primeiro valor de *nonce* na sequência de tentativas que satisfaz a desigualdade em (1) é a solução. Este valor é denominado de *golden nonce* e se torna, como visto, uma informação constituinte do cabeçalho do bloco [6], [11].

Quanto menor (maior) é D, maior (menor) é o número de tentativas para encontrar o *golden nonce*, ou seja, maior (menor) é o número de *hashes*, resultando em maior (menor) segurança sistêmica. Todavia, aumentar (diminuir) o número de *hashes* também implica aumentar (diminuir) o intervalo de tempo δt para adicionar um bloco à lista, ou seja, aumenta (diminui) o tempo de *mineração* e, consequentemente, o tempo para confirmação das transações submetidas [11].

Tendo sido *minerado*, o bloco é então adicionado à lista local do processador que realizou a *mineração* e, na sequência, é disseminado pela rede P2P para que os demais processadores possam atualizar as suas respectivas listas locais, possibilitando a convergência sistêmica da base de dados. Esta base atualizada é acessível para consulta pelos clientes do sistema por meio dos processadores da rede P2P.

A *mineração* descrita pode também ser realizada por um grupo de processadores em vez de apenas um único processador. Neste caso, o grupo de processadores constitui o que se chama *mining pool* [12]. Os processadores do grupo trabalham então cooperativamente para resolver o desafio matemático expresso em (1).

Para finalizar esta subseção, ressalta-se que o escopo deste trabalho está restrito à proposta original da *Blockchain* [6], considerando a base de dados na categoria *pública* [13], [14], e o algoritmo de consenso do tipo PoW. Discussão de variantes da *Blockchain* são deixadas como trabalhos futuros.

III. Arquitetura IoT

Sob uma visão *top-down*, a arquitetura do ecossistema IoT pode ser dividida nas quatro camadas [23], [3], [24] a seguir.

1) *Camada de aplicação*: utiliza as informações geradas pela camada de processamento. Esta camada abrange as aplicações IoT, considerando seus inúmeros domínios, e.g., transporte e mobilidade, logística, meio ambiente, cidades inteligentes, serviços de vigilância, e Indústria 4.0.

2) *Camada de processamento*: processa e armazena os dados provenientes da camada de rede, assim gerando informações que são utilizadas por camadas superiores ou inferiores. As técnicas utilizadas incluem, e.g., *Big Data*, computação em nuvem, computação de borda, e computação em névoa.

3) *Camada de rede*: transmite os dados da camada de percepção para a camada de processamento, podendo empregar variados tipos de infraestrutura de rede, e.g., cabeadas, sem fio, móveis, veiculares, e *mesh*. Algumas das tecnologias de enlace usadas incluem, e.g., 5G, Wi-Fi, Bluetooth, e ZigBee.

4) *Camada de percepção*: abrange os dispositivos eletroeletrônicos disponíveis no meio que, além de poderem se conectar à Internet, são capazes de monitorar eventos físicos e, quando necessário, tomar decisões sem intervenção humana.

Convém destacar que este artigo está delimitado à camada de processamento, na qual assume-se que estão localizados os processadores da rede P2P da *Blockchain*. Investigações sobre as demais camadas são deixadas como trabalhos futuros.

IV. Trabalhos Relacionados

Em [6], tem-se a proposta original da tecnologia *Blockchain*, apresentada em conjunto com o sistema de criptomoedas *Bitcoin*. O autor apresenta o formalismo conceitual da tecnologia e sua operação. Por meio de modelagem matemática, são realizados experimentos para estimar o nível de segurança (integridade) da tecnologia, cujos resultados revelam uma promissora resiliência a ataques de fraudes.

Em [17], os autores apresentam um método de análise do sistema *Bitcoin* para identificar a dinâmica das transações associadas a pagamentos devido a ataques cibernéticos do tipo *ransomware*. O objetivo é obter informações sobre as formas de dispersão e lavagem de dinheiro utilizadas pelos atacantes. O escopo do trabalho é, todavia, restrito ao aspecto operacional do *Bitcoin* como um sistema financeiro [25], ofertando subsídios direcionados para o projeto de estratégias de segurança em prol de regras de negócio.

Em [2], [3], [4], [8], [13], [15], [16], os respectivos autores trazem *surveys* sobre *Blockchain* em IoT. Além de aspectos conceituais, os trabalhos discutem uma gama de aplicações IoT (e/ou estudos de caso) baseadas em *Blockchain*, em sua maioria ressaltando as vantagens comparativas ao paradigma centrado na nuvem e os desafios de implementações reais. Embora não haja ineditismo absoluto, esses trabalhos são valiosos alicerces para trabalhos de pesquisa.

Em [18] é proposta uma arquitetura baseada em *Blockchain* para implementação de uma base de dados constituída por informações de saúde geradas pelos usuários e por atores do ecossistema de saúde (e.g., clínicas, hospitais, etc.), onde são contempladas questões técnicas sobre os dados armazenados (e.g., interoperabilidade, compartilhamento, segurança, e criptografia). Em que pese sua relevância, o trabalho todavia se restringe ao aspecto conceitual de arquitetura.

Em [19] é proposto o *framework* de compartilhamento de dados Sasha, cuja implementação considera o emprego da tecnologia *Blockchain* no ecossistema IoT. Em específico, a base de dados é usada para políticas de controle de acesso, permitindo que alterações e solicitações de acesso sejam corretamente aplicadas e publicamente auditáveis. Os experimentos são baseados em prototipagem usando FIWARE como plataforma de IoT e a estrutura Hyperledger Fabric para a base de dados. Os resultados experimentais mostram um desempenho adequado para operações de busca e inserção. Porém, inexistem investigações sobre segurança.

Em [20], tem-se uma proposta para a aplicação *casa inteligente*, admitindo o ecossistema IoT com emprego da tecnologia *Blockchain*. A implementação consiste de três camadas: *casa inteligente*, *sobreposição* e *armazenamento em*

TABELA I
SÍNTESE DOS TRABALHOS RELACIONADOS

Referências	Categoria	Escopo	Principal limitação comparativa ao nosso trabalho
[6]	Pesquisa	Proposta do *Bitcoin*/*Blockchain*.	Foco no *Bitcoin*, sem considerar IoT.
[2], [3], [4], [8], [13], [15], [16]	*Survey*	Conceitos, aplicações e estudos de caso.	Revisão conceitual e teórica.
[17]	Pesquisa	Análise de ataques cibernéticos.	Restrito ao *Bitcoin*.
[18]	Pesquisa	Arquitetura para sistema de saúde.	Abordagem conceitual e teórica.
[19]	Pesquisa	*Framework* para políticas de acesso.	Experimentos não avaliam segurança.
[20]	Pesquisa	Aplicação para *casa inteligente*.	Experimentos não avaliam segurança.
[21]	Pesquisa	Esquema descentralizado de base de dados.	Experimentos não avaliam segurança.
[22]	Pesquisa	Análise de segurança (integridade).	Experimentos não avaliam eficiência.

nuvem. Na primeira camada estão os dispositivos inteligentes, na segunda estão os processadores da rede, e na terceira está o armazenamento final dos dados. Simulações mostram que o *overhead* (em termos de tráfego, processamento e consumo de energia) é insignificante, levando-se em consideração os ganhos em segurança teoricamente advindos da solução.

Em [21], é apresentada uma plataforma IoT com *Blockchain*. O objetivo é fornecer ao usuário do dispositivo inteligente um esquema descentralizado no qual as informações podem ser armazenadas em uma base de dados abrangente e imutável, além de serem compartilhadas rapidamente. Ademais de análises comparativas com outras soluções, tem-se a validação da proposta apoiada por uma implementação de prova de conceito em cenários realistas de IoT, onde são utilizados dispositivos Raspberry Pi e a estrutura Hyperledger Fabric para a base de dados. Os experimentos constatam um adequado tempo de processamento de transações. Porém, como em [19], [20], não há experimentos sobre segurança.

Por fim, em [22], os autores realizam uma análise competitiva entre as tecnologias *Blockchain* e *Tangle* com foco na garantia de integridade das informações em aplicações IoT. Os resultados dos experimentos, baseados em modelagem analítica e simulações, sugerem uma maior robustez e uma maior escalabilidade da *Tangle*. Entretanto, os cenários examinados são restritos, além de não haver avaliações sobre a eficiência do processamento de transações e, tampouco, considerados ajustes de configuração da *Blockchain*.

Ante o exposto e em acordo com nosso conhecimento (vide síntese na Tabela I), os trabalhos de pesquisa usualmente admitem a segurança como intrinsecamente garantida, o que é uma lacuna de investigação. Mais especificamente, o foco dos trabalhos se direciona para avaliação do tempo de processamento das transações, mas sem considerar tolerâncias no tempo de resposta, o que também constitui uma lacuna de investigação. Essas duas lacunas são investigadas neste artigo.

V. AVALIAÇÃO DE PERFORMANCE

A Subseção V-A discorre sobre oito cenários de IoT examinados nos experimentos. A Subseção V-B trata sobre a eficiência do sistema e se desenvolve em torno da seguinte questão: Quanto tempo leva para uma transação de um dispositivo inteligente ser confirmada no sistema, i.e., ser processada pela camada de processamento? A Subseção V-C se dedica à integridade de dados e se desenvolve em torno da seguinte questão: Qual a probabilidade de um fraudador modificar as transações já registradas na base de dados do sistema?

A. Cenários de IoT

Os cenários de IoT podem ser analisados a partir de seus correspondentes domínios de aplicação. Esses domínios consistem de variados tipos de serviços com diferentes características associadas. Com base nos trabalhos de [26], [27], [28], a Tabela II traz uma síntese de oito populares domínios de aplicação e, por sua vez, a Tabela III traz a caracterização dos oito cenários examinados nos experimentos.

B. Eficiência do Sistema

Para avaliação da eficiência, assume-se inicialmente cada cenário S_j, $j \in \{1, 2, \ldots, 8\}$ (vide Tabela III), modelado por um sistema de fila [29] do tipo $M/M/1/\infty/FIFO$, como ilustrado na Fig. 2 e explicado a seguir.

O processo de chegada no cenário S_j se refere a blocos de transações, vindos da camada de rede e em direção à camada de processamento do ecossistema IoT. As transações são originadas coletivamente pelos dispositivos d_i da camada de percepção, para $1 \leq i \leq n_j$, onde n_j é o limite superior do tamanho da rede do cenário S_j, em número de dispositivos.

Esse processo de chegada de blocos é um processo de Poisson de parâmetro $\lambda_j = (\sum_{i=1}^{n_j} \lambda_i)/L_j$, onde λ_i é a taxa de transações do dispositivo d_i, e L_j é o tamanho do bloco de transações no cenário S_j, em número de transações. Para o processamento dos blocos, assume-se um servidor único, cujo tempo de serviço (ou atendimento) tem distribuição exponencial de parâmetro μ_j. Esse servidor único representa os processadores da rede P2P. Ademais, os blocos que chegam nunca são descartados, pois o sistema tem capacidade de armazenamento infinita (i.e., fila infinita), e são selecionados para processamento em acordo com a sua ordem de chegada, i.e., disciplina FIFO (*First-In, First-Out*).

O tempo de confirmação da transação no cenário S_j é estimado pelo tempo de resposta do sistema, W_j, sob a visão do bloco de transações, que é a unidade de informações da tecnologia *Blockchain*. Isso porque a transação somente se torna disponível para consulta pelas camadas superiores (ou

TABELA II
DOMÍNIO DE APLICAÇÃO DE CENÁRIOS DE IOT

Domínio	Aplicação considerada	Tam. da rede	Tx. de transações (individual)	Tolerância de retardo
Prédios e lares	Controle de iluminação	10-1.000 dispositivos	1/15 min	5 seg
Assistência de saúde	Monitoramento de pacientes	100-1.000 dispositivos	1/10 seg	3 seg
Meio ambiente	Rastreamento de animais	100-1.000 dispositivos	1/30 min	algumas horas
Cidades	Controle de tráfego	1.000-10.000 dispositivos	1/10 min	15 seg
Energia	Gerência de ativos	100-10.000 dispositivos	1/15 min	15 seg
Transporte e mobilidade	Localização de veículos	1.000-10.000 dispositivos	1/30 seg	10 seg
Produção e venda	Manutenção de máquinas	100-1.000 dispositivos	1/10 min	10 seg
Agricultura	Irrigação	10-1.000 dispositivos	1/h	1 min

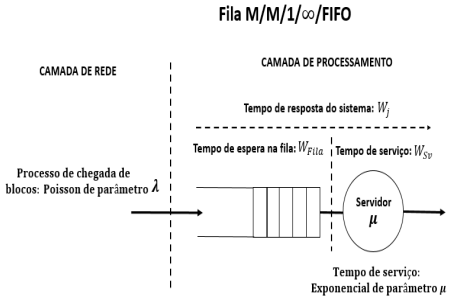

Fig. 2. Modelagem por fila $M/M/1/\infty/FIFO$.

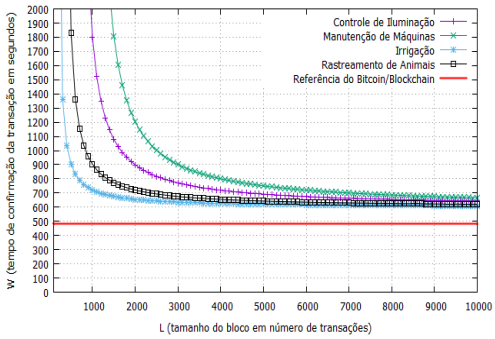

Fig. 3. Análise do tempo de confirmação da transação em diferentes domínios.

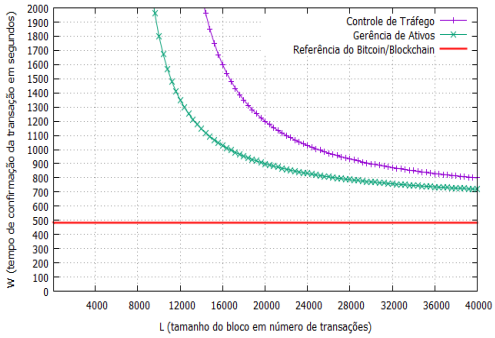

Fig. 4. Análise do tempo de confirmação da transação em diferentes domínios.

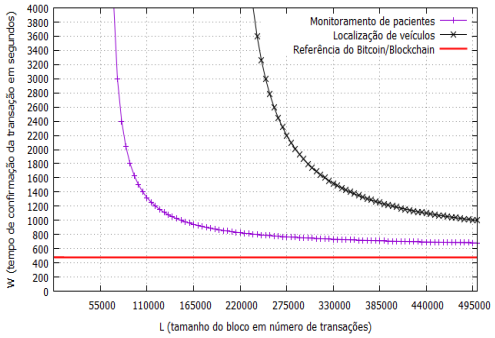

Fig. 5. Análise do tempo de confirmação da transação em diferentes domínios.

inferiores) quando o correspondente bloco que a contém tem seu processamento finalizado (i.e., o bloco é *minerado*).

Para o cenário S_j, tem-se em (2) o cálculo numérico de W_j [29], onde W_{Fila} é o tempo de espera na fila, W_{Sv} é o tempo de serviço (atendimento), e $\lambda_j/\mu_j < 1$ (i.e., sistema estável).

$$W_j = W_{Fila} + W_{Sv} = \frac{1}{\mu_j - \lambda_j} \tag{2}$$

Os valores de λ_j estão na Tabela III. O valor de μ_j é 1/10 min em todos os cenários S_j, que corresponde à configuração da *Blockchain* do sistema *Bitcoin* [6]. Isso é feito com o intuito de comparação. Também, para comparação, tem-se as seguintes outras estatísticas do sistema *Bitcoin* [30]: 1.825 (número médio de transações por bloco); 1,03 MB (tamanho médio do bloco); e 7,8 minutos ou 480,0 segundos (tempo médio de confirmação de uma transação).

As Figuras 3, 4 e 5 trazem os resultados calculados para W_j em função de L_j, onde a linha vermelha representa o valor de referência para o sistema *Bitcoin* [30]. Estes resultados mostram a inviabilidade da aplicação direta da *Blockchain* nos cenários examinados. Isso porque os correspondentes valores de convergência dos tempos de confirmação estão acima do valor tolerável $Max\,W_j$ (vide Tabela III).

Alternativamente, propõe-se então um cenário único S_U que agrega as capacidades de processamento individuais das redes dos cenários S_j, para $1 \leq j \leq 8$. A motivação para essa alternativa é o aproveitamento da eventual ociosidade das redes independentes durante a operação. Para avaliação, o cenário S_U é modelado por um sistema de fila [29] do tipo $M/M/c/\infty/FIFO$, ilustrado na Fig. 6. Sob o cenário S_U, o processo de chegada se refere a blocos de transações, que são provenientes da camada de rede e vão em direção à camada de processamento do ecossistema IoT. As transações são originadas coletivamente pelos dispositivos de todos os cenários S_j, para $1 \leq j \leq 8$.

Com base nas propriedades de superposição e decomposição do processo de Poisson [29], a chegada de blocos no cenário

TABELA III
VALORES DOS PARÂMETROS DE CARACTERIZAÇÃO DOS CENÁRIOS S_j.

S_j	λ_i	n_j	$\lambda_j = \sum_{i=1}^{n_j} \lambda_i$	$Max\,W_j$
Controle de iluminação	1/15 min	1.000	1.000/15 min	5 seg
Monitoramento de pacientes	1/10 seg	1.000	1.000/10 seg	3 seg
Rastreamento de animais silvestres	1/30 min	1.000	1.000/30 min	algumas horas
Controle de tráfego	1/10 min	10.000	10.000/10 min	15 seg
Gerência de ativos	1/15 min	10.000	10.000/15 min	15 seg
Localização de veículos	1/30 seg	10.000	10.000/30 seg	10 seg
Manutenção de máquinas	1/10 min	1.000	1.000/10 min	10 seg
Irrigação	1/h	1.000	1.000/h	1 min

S_j também ocorre em conformidade com um processo de Poisson, com parâmetro $\lambda_U = (\sum_{j=1}^{8} \lambda_j)/L_U$, onde λ_j é, como antes, a taxa de chegada de transações no cenário independente S_j, e L_U é o tamanho do bloco no sistema, em número de transações.

A camada de processamento do cenário S_U possui c servidores independentes, cujos tempos de serviço individuais são idênticos, possuindo individualmente distribuição exponencial de parâmetro μ. O servidor que estiver (ou ficar) disponível é automaticamente selecionado para servir o bloco da vez. O bloco recebe então um serviço de tempo exponencialmente distribuído de parâmetro μ. Se todos os c servidores estiverem ocupados, então o bloco que chega é colocado na fila.

Seja N_{Sist} o número de blocos no sistema (i.e., na fila e em serviço). A taxa de serviço do sistema é $N_{Sist}.\mu$ (para $0 \leq N_{Sist} < c$) ou $c.\mu$ (para $N_{Sist} \geq c$). Como antes, os blocos que chegam nunca são descartados, pois o sistema tem capacidade de armazenamento infinita (i.e., fila infinita), e são selecionados para serviço de acordo com a ordem de chegada, i.e., disciplina FIFO.

O tempo de confirmação da transação no cenário S_U é estimado pelo tempo de resposta do sistema, W_U, sob a visão do bloco de transações. Como antes, a transação somente se torna disponível para consulta pelas camadas superiores (ou inferiores) quando o correspondente bloco que a contém tem seu processamento finalizado (i.e, o bloco é *minerado*).

Em (3) tem-se então o cálculo de W_U [29] para o cenário S_U, onde W_{Fila} é o tempo de espera na fila, W_{Sv} é o tempo de serviço (atendimento), e N_{Fila} é o número de blocos na fila (i.e., aguardando serviço). Em seu turno, o valor de N_{Fila} é calculado em (4), onde: $r = \lambda_U/\mu$; $\rho = r/c < 1$ (i.e., sistema estável); e P_0 é a probabilidade de não haver blocos no sistema, dada em (5).

$$W_U = W_{Fila} + W_{Sv} = \frac{N_{Fila}}{\lambda_U} + \frac{1}{\mu} \tag{3}$$

$$N_{Fila} = \frac{P_0.r^c}{c!}.\frac{\rho}{(1-\rho)^2} \tag{4}$$

$$P_0 = \frac{1}{\sum_{n=0}^{c-1} \frac{r^n}{n!} + \frac{r^c}{c!(1-\rho)}} \tag{5}$$

Lembramos que os valores de λ_j para o cálculo de λ_U estão na Tabela III, o valor de c é igual a oito, e o valor de μ é novamente igualado a 1/10 min para fins de comparação com a *Blockchain* do sistema *Bitcoin* [6]. A Fig. 7 traz então

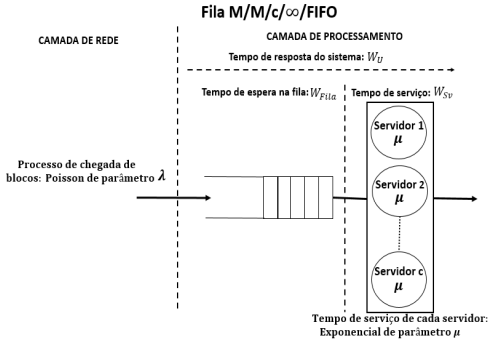

Fig. 6. Modelagem por fila $M/M/c/\infty/FIFO$.

os resultados calculados para W_U em função de L_U, onde a linha vermelha representa o valor de referência para o sistema *Bitcoin* [30]. Note que W_U converge para 600 s para $L_U \geq 60.000$. Comparativamente aos valores de W_j, para $1 \leq j \leq 8$, o valor de W_U é da mesma ordem de grandeza. Isso justifica, sob o critério da economicidade, a alternativa de um cenário único S_U em vez de cenários independentes S_j, para $1 \leq j \leq 8$. Isso porque a implantação de uma única infraestrutura de rede de maior capacidade de processamento tende a ser menos onerosa que a implantação de j infraestruturas de redes independentes de menor capacidade individual cada.

Porém, sob o critério da eficiência, o valor de W_U está acima da maioria dos valores toleráveis $Max\,W_j$ (vide Tabela III). Visualizam-se então três soluções para reduzir o valor de W_U a patamares toleráveis. A primeira é o aumento do número de servidores. Isso significa adicionar *hardware* ao sistema e, portanto, implica custo econômico. A segunda consiste no uso de uma política de admissão de transações para garantir o tempo de confirmação desejável. Isso implica, e.g., a necessidade do monitoramento da qualidade de serviço da rede, o que pode acarretar um *overhead* de processamento intolerável. A terceira é o ajuste do *alvo de dificuldade*, D, relativo à *mineração* do bloco (vide (1)). Esta última solução não traz custos econômicos, pois é baseada em *software*, tampouco implica a necessidade do monitoramento mencionado.

As três soluções supracitadas não são mutuamente exclusivas, podendo ser aplicadas em conjunto. Todavia, pelo seu menor custo de implantação e maior simplicidade de implementação, tem-se neste trabalho a investigação apenas da terceira solução, deixando as demais, bem como o uso combinado das mesmas, para trabalhos futuros.

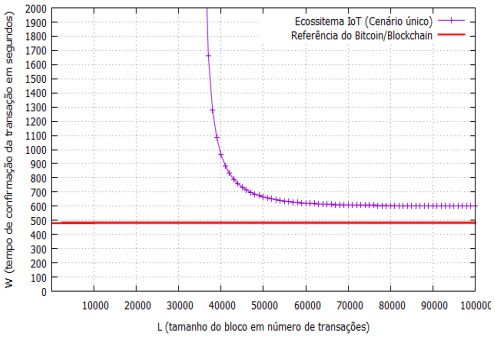

Fig. 7. Análise do tempo de confirmação da transação no ecossistema IoT.

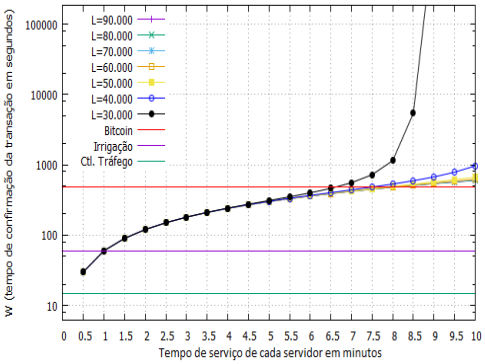

Fig. 8. Análise do tempo de confirmação da transação no ecossistema IoT.

Como visto na Seção II, o aumento de D implica que, com a mesma *hashrate*, consegue-se ter um menor tempo de *mineração* de bloco, δt, e consequentemente uma redução do tempo de resposta do sistema. Ante a modelagem realizada, isso significa reduzir o tempo de serviço de cada servidor, $1/\mu$. A análise a seguir então avalia indiretamente o impacto de D sobre W_U pela observação do impacto de $1/\mu$ sobre W_U.

Neste contexto, a Fig. 8 apresenta os resultados obtidos de W_U em função de $1/\mu$, para diferentes valores de L_U. Para referência, destacam-se o tempo de confirmação do sistema *Bitcoin* [30] e os valores de $Max\,W_j$ para as aplicações de irrigação e de controle de tráfego (vide Tabela III).

Para $1/\mu$ no intervalo de $0,5 - 6,0$ min, nota-se que o comportamento de W_U é o mesmo para todos os valores de L_U. Em específico, excluindo-se o valor $L_U = 30.000$, esse comportamento passa a abranger um intervalo ainda mais extenso: $0,5 - 7,5$ min. Além disso, o valor $L_U = 50.000$ estabelece um compromisso adequado entre W_U e $1/\mu$, pois maiores valores de L_U não acarretam otimizações relevantes em W_U. Por fim, nota-se ainda que o valor mínimo de W_U converge para o valor configurado para $1/\mu$ que, como mencionado, é aqui determinado pelo valor de D.

Todavia, o ajuste indiscriminado de D pode vir a aumentar a probabilidade de fraude, comprometendo a integridade dos dados armazenados (vide (1)). Por garantia de integridade de dados, entende-se a garantia de que as informações armazenadas e disponibilizadas pela camada de processamento do ecossistema IoT não possam ser adulteradas por processadores desonestos (i.e., fraudadores).

Em síntese, o ajuste da *Blockchain* no cenário S_U, para atender às tolerâncias estimadas por $Max\,W_j$ (vide Tabela III), não pode prescindir de um compromisso entre o tempo de confirmação de transação almejado e a garantia da integridade dos dados. Este compromisso é discutido na próxima seção.

C. Integridade de Dados

Esta seção avalia a integridade de dados a partir da modelagem de um ataque originado por um processador fraudador (ou conjunto de processadores fraudadores) da rede P2P. Este ataque ocorre durante a construção da cadeia de blocos pela camada de processamento no cenário S_U.

Embora as regras de negócio variem em função de cada aplicação específica constituinte do cenário S_U, a modelagem do ataque se restringe ao exame da cadeia de blocos da camada de processamento e, portanto, é idêntica e aplicável a quaisquer que sejam as aplicações específicas constituintes do cenário S_U. Essa modelagem é sinteticamente ilustrada na Fig. 9, sendo explicada a seguir.

Seja B_i o bloco adicionado à cadeia da camada de processamento no instante de tempo $t = t_0$. Assuma que B_i é um bloco legítimo, i.e., que contém dados verdadeiros, e que um fraudador deseja substituí-lo por um outro bloco B_i, maliciosamente manipulado. Em $t = t_0 + \Delta t$, a cadeia do bloco legítimo B_i já tem a adição de mais z blocos seguintes, constituindo a cadeia honesta: (B_i, B_{i+1}, B_{i+2}, ..., B_{i+z}). Neste instante o fraudador cria um $fork$ (i.e., bifurcação) a partir do bloco B_{i-1}, inserindo uma cadeia fraudada que contém o bloco B_i adulterado: (B_i, B_{i+1}, B_{i+2}, ..., B_{i+k}). Esta cadeia fraudada é construída no intervalo de tempo Δt, em acordo com o poder de processamento do fraudador e de maneira secreta (i.e., não é visível para o sistema até $t = t_0 + \Delta t$), e os seus blocos B_{i+1}, B_{i+2}, ..., B_{i+k} são legítimos (i.e., construídos a partir de transações verdadeiras).

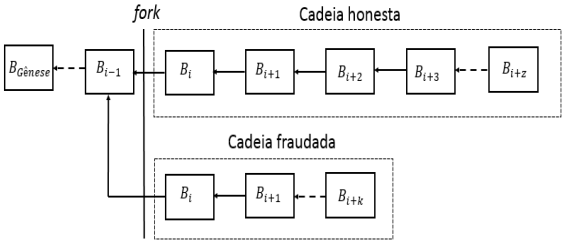

Fig. 9. Construção da *cadeia de blocos* na camada de processamento.

Quando da ocorrência de $forks$, a regra da cadeia mais longa [1], [6] estabelece que, dentre as cadeias concorrentes, aquela que se tornar a mais longa deve ser considerada, sendo as demais desprezadas. Mais especificamente, a cadeia mais longa é a que primeiro atinge uma certa vantagem, em número de blocos, conforme os blocos vão sendo adicionados. O valor desta vantagem é um parâmetro de configuração [1].

Assumimos que o ataque pode então ser bem sucedido em dois casos. Primeiro, quando o poder de processamento do fraudador é suficientemente maior que o do sistema para ter $(k+1) \geq (z+1)$ em $t = t_0 + \Delta t$. Neste caso, a probabilidade

de fraude é igual a 1,0, dispensando análises complementares. O segundo caso é o alvo de estudo desta seção e considera a situação onde: $(k+1) < (z+1)$ em $t = t_0 + \Delta t$. Admite-se então que, mesmo com poder de processamento inferior, o fraudador pode tornar sua cadeia mais longa, i.e., compensar a desvantagem de $z-k$ blocos em algum $t > t_0 + \Delta t$.

A partir de $t = t_0 + \Delta t$, há então uma disputa entre processadores honestos e desonestos (i.e., fraudadores), os quais adicionam blocos às cadeias honesta e fraudada, respectivamente. Quanto menor é Δt, maior é a chance de fraude, pois menor tende a ser a desvantagem $z-k$. O caso mais crítico é, portanto, quando o fraudador inicia a disputa imediatamente depois de $t = t_0$, ou seja: para $\Delta t \to 0$ e $z = 0$.

A disputa entre processadores honestos e fraudadores pode ser caracterizada como um passeio aleatório binomial. O evento de sucesso é quando a cadeia honesta é estendida por um bloco, aumentando a diferença $z-k$ inicial em $+1$, e o evento de insucesso é quando a cadeia fraudada é estendida por um bloco, diminuindo a diferença inicial $z-k$ em -1. Neste contexto, assuma que os processadores honestos e os fraudadores do cenário S_U têm respectivamente probabilidades p e $q = (1-p)$ de adicionar o próximo bloco. Esses valores de probabilidade refletem assim o percentual do poder computacional total do sistema pertencente aos processadores honestos, i.e., $(p \times 100)\%$, e aos fraudadores, i.e., $(q \times 100)\%$.

A modelagem é realizada e resolvida por simulação no ambiente Tangram-II [31]. Este ambiente foi concebido pela Universidade Federal do Rio de Janeiro (UFRJ), com a participação da Universidade da Califórnia em Los Angeles (UCLA), e se destina à modelagem e análise de sistemas computacionais e de comunicação. Os resultados de simulação a seguir têm intervalos de confiança de 95% que estão dentro do limite de 5% dos valores reportados, tendo sido realizadas 30 execuções (rodadas) com um tempo de simulação de 100.000 min cada. Informa-se, ainda, que a plataforma de *hardware* computacional utilizada é um Intel Core i5 (2,67 GHz), com 3,6 GB de RAM, de sistema operacional GNU/Linux.

A Fig. 10 apresenta então a probabilidade de sucesso de fraude em função do tempo de serviço de cada servidor. Neste experimento, assume-se $L = 50.000$, por estabelecer um compromisso adequado entre tempo de confirmação da transação e tempo de serviço de cada servidor (vide Subseção V-B), e $\Delta t \to 0$ e $z = 0$, por ser a situação mais crítica quanto à possibilidade de fraude, como discutido antes. São comparados cinco diferentes cenários, individualmente caracterizados pelos percentuais do poder de processamento de fraudadores em relação ao poder total do sistema.

A partir desses resultados, é possível notar o seguinte. Primeiro, a probabilidade de fraude somente excede $\approx$ 0,1 quando o poder dos fraudadores compromete mais que 30% do poder total do sistema, conforme ressaltado pela média calculada na Tabela IV. Segundo, a probabilidade de fraude pouco difere considerando o tempo de serviço de cada servidor no intervalo de 10 seg até 15 min, conforme indicado pela variância apresentada na mesma Tabela IV.

Tem-se então que D pode ser ajustado para obter o tempo de confirmação de transação desejado, conforme requisitos das aplicações. Isso pode ser feito sem prejudicar a integridade

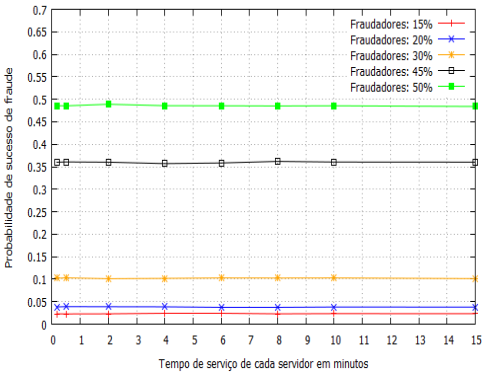

Fig. 10. Análise da probabilidade do sucesso de fraude.

TABELA IV
PROBABILIDADE DE SUCESSO DE FRAUDE.

Poder de fraudadores	Média	Variância
15%	2,31e-02	6,54-e08
20%	3,84e-02	1,39e-07
30%	1,02e-01	5,33e-07
45%	3,60e-01	2,36e-06
50%	4,86e-01	1,99e-06

dos dados (i.e., probabilidade de fraude menor que $\approx 0,1$), desde que o processamento do sistema não seja comprometido em mais que 30% (vide Tabela IV). Para maior garantia de integridade, os ataques devem ocorrer somente em $\Delta t > 0$. Para tanto, podem-se considerar as duas estratégias a seguir.

A primeira estratégia é que as regras de negócios imponham que as transações contidas no bloco B_i (vide Fig. 9), inserido em $t = t_0$, somente sejam reconhecidas pelo sistema quando $z > 0$ blocos subsequentes forem adicionados à cadeia. A segunda consiste na divulgação de subcadeias de blocos que vão sendo adicionadas à cadeia principal, não sendo mais permitida a adição de blocos individuais.

Nas duas estratégias supracitadas, os fraudadores são então compelidos a esperar por algum Δt antes de iniciar um ataque. Ressalta-se, contudo, que a modelagem aqui proposta permanece válida e ainda pode ser usada para essa análise, o que é deixado para trabalhos futuros.

VI. CONCLUSÕES E TRABALHOS FUTUROS

Este artigo analisou a *Blockchain* para implementação da base de dados do ecossistema IoT. Por meio de modelos de filas e simulações, os experimentos especialmente mostraram que *Blockchain* pode ser configurada para atender aos requisitos de tempo de confirmação de transações em diferentes aplicações sem impactar a integridade dos dados (i.e., probabilidade de fraude $\leq$ 0,1), desde que o processamento honesto do sistema seja ao menos 70% do total. Essa configuração consiste na alteração do tamanho do bloco de transações e na redução da complexidade de seu processamento. Como pesquisas futuras e contemplando limitações deste trabalho, apontam-se: análise de algoritmos de consenso diferentes do PoW; comparação de implementações de base de dados diferentes do tipo *pública*; estudo de tecnologias de registros distribuídos diferentes da *Blockchain*; e realização de *testbeds*.

REFERÊNCIAS

[1] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks," *IEEE Access*, vol. 7, pp. 22 328–22 370, 2019, doi: 10.1109/ACCESS.2019.2896108.

[2] M. Maroufi, R. Abdolee, and B. Tazekand, "On the Convergence of Blockchain and Internet of Things (IoT) Technologies," *Journal of Strategic Innovation and Sustainability*, vol. 14, no. 1, Mar. 2019, doi:10.33423/jsis.v14i1.990.

[3] H. Dai, Z. Zheng, and Y. Zhang, "Blockchain for Internet of Things: A Survey," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, 2019, doi: 10.1109/JIOT.2019.2920987.

[4] D. Minoli and B. Occhiogrosso, "Blockchain mechanisms for IoT security," *Internet of Things*, vol. 1-2, pp. 1 – 13, 2018, doi:10.1016/j.iot.2018.05.002.

[5] K. Zhang and H. Jacobsen, "Towards Dependable, Scalable, and Pervasive Distributed Ledgers with Blockchains," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, July 2018, pp. 1337–1346, doi: 10.1109/ICDCS.2018.00134.

[6] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008, Available at: https://bitcoin.org/bitcoin.pdf. Accessed on: June 28th, 2020.

[7] J. J. Hunhevicz and D. M. Hall, "Do you need a blockchain in construction? Use case categories and decision framework for DLT design options," *Advanced Engineering Informatics*, vol. 45, p. 101094, 2020, doi:10.1016/j.aei.2020.101094.

[8] Q. Wang, X. Zhu, Y. Ni, L. Gu, and H. Zhu, "Blockchain for the IoT and industrial IoT: A review," *Internet of Things*, p. 100081, 2019, doi:10.1016/j.iot.2019.100081.

[9] H. Gilbert and H. Handschuh, "Security Analysis of SHA-256 and Sisters," in *Selected Areas in Cryptography*, M. Matsui and R. J. Zuccherato, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 175–193, doi:10.1007/978-3-540-24654-1_13.

[10] M. Jakobsson and A. Juels, *Proofs of Work and Bread Pudding Protocols (Extended Abstract)*. Boston, MA: Springer US, 1999, pp. 258–272, doi: 10.1007/978-0-387-35568-9_18.

[11] R. Bowden, H. P. Keeler, A. E. Krzesinski, and P. G. Taylor, "Block arrivals in the bitcoin blockchain," 2018, [Online] Available at: https://arxiv.org/abs/1801.07447. Accessed on: June 28th, 2020.

[12] S. M. H. Bamakan, A. Motavali, and A. B. Bondarti, "A survey of blockchain consensus algorithms performance evaluation criteria," *Expert Systems with Applications*, vol. 154, p. 113385, 2020, doi:10.1016/j.eswa.2020.113385.

[13] T. M. Fernández-Caramés and P. Fraga-Lamas, "A Review on the Use of Blockchain for the Internet of Things," *IEEE Access*, vol. 6, pp. 32 979–33 001, 2018, doi: 10.1109/ACCESS.2018.2842685.

[14] C. K. da S. Rodrigues and P. C. da Silva, "Uma Análise de Algoritmos de Consenso para Blockchain visando à Implementação de Sistemas de Informação Distribuídos Transparentes," *Revista de Sistemas e Computação*, vol. 9, no. 1, pp. 163–188, 2019.

[15] M. Conoscenti, A. Vetrò, and J. C. De Martin, "Blockchain for the Internet of Things: A systematic literature review," in *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, 2016, pp. 1–6, doi: 10.1109/AICCSA.2016.7945805.

[16] A. A. Monrat, O. Schelén, and K. Andersson, "A Survey of Blockchain From the Perspectives of Applications, Challenges, and Opportunities," *IEEE Access*, vol. 7, pp. 117 134–117 151, 2019, doi: 10.1109/ACCESS.2019.2936094.

[17] V. G. Reyes-Macedo, M. Salinas-Rosales, and G. Gallegos Garcia, "A Method for Blockchain Transactions Analysis," *IEEE Latin America Transactions*, vol. 17, no. 07, pp. 1080–1087, 2019, doi: 10.1109/TLA.2019.8931194.

[18] P. S. R. Garcia and J. H. Kleinschmidt, "Sharing Health and Wellness Data with Blockchain and Smart Contracts," *IEEE Latin America Transactions*, vol. 18, no. 06, pp. 1026–1033, 2020, doi: 10.1109/TLA.2020.9099679.

[19] H. T. T. Truong, M. Almeida, G. Karame, and C. Soriente, "Towards Secure and Decentralized Sharing of IoT Data," in *2019 IEEE International Conference on Blockchain (Blockchain)*, July 2019, pp. 176–183, doi: 10.1109/Blockchain.2019.00031.

[20] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, March 2017, pp. 618–623, doi: 10.1109/PERCOMW.2017.7917634.

[21] L. Hang and D. Kim, "Design and Implementation of an Integrated IoT Blockchain Platform for Sensing Data Integrity," *Sensors*, vol. 10:2228, 2019, doi:10.3390/s19102228.

[22] G. G. Martinez and C. K. da S. Rodrigues, "Blockchain and Tangle: The Transaction Security in the IoT Ecosystem," *Revista de Sistemas e Computação*, vol. 10, no. 1, pp. 42–50, 2020.

[23] B. E. El-Shweky, K. El-Kholy, M. Abdelghany, M. Salah, M. Wael, O. Alsherbini, Y. Ismail, K. Salah, and M. AbdelSalam, "Internet of things: A comparative study," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan 2018, pp. 622–631, doi: 10.1109/CCWC.2018.8301678.

[24] S. Moin, A. Karim, Z. Safdar, K. Safdar, E. Ahmed, and M. Imran, "Securing IoTs in distributed blockchain: Analysis, requirements and open issues," *Future Generation Computer Systems*, vol. 100, pp. 325 – 343, 2019, doi:10.1016/j.future.2019.05.023.

[25] S. Singh and N. Singh, "Blockchain: Future of financial and cyber security," in *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, 2016, pp. 463–467, doi: 10.1109/IC3I.2016.7918009.

[26] J. Mocnej, A. Pekar, W. K. G. Seah, and I. Zolotova, "Network Traffic Characteristics of the IoT Application Use Cases," School of Engineering and Computer Science, Victoria University of Wellington, Technical Report ECSTR18-01, 6 Jan 2018. [Online]. Available: https://ecs.victoria.ac.nz/Main/TechnicalReportSeries. Accessed on: June 28th, 2020."

[27] J. Mocnej, W. K. Seah, A. Pekar, and I. Zolotova, "Decentralised IoT Architecture for Efficient Resources Utilisation," *IFAC-PapersOnLine*, vol. 51, no. 6, pp. 168 – 173, 2018, 15th IFAC Conference on Programmable Devices and Embedded Systems PDeS 2018, doi:10.1016/j.ifacol.2018.07.148.

[28] J. Mocnej, M. Miskuf, P. Papcun, and I. Zolotova, "Impact of Edge Computing Paradigm on Energy Consumption in IoT," *IFAC-PapersOnLine*, vol. 51, no. 6, pp. 162 – 167, 2018, 15th IFAC Conference on Programmable Devices and Embedded Systems PDeS 2018, doi: 10.1016/j.ifacol.2018.07.147.

[29] L. Kleinrock, *Queuing Systems. Volume I: Theory*. New York: Wiley, 1975.

[30] Blockchain.com, "Blockchain Charts," 2020, [Online]. Available at: https://www.blockchain.com/pt/charts. Accessed on: Jul. 1st, 2020.

[31] E. de Souza e Silva, R. Figueiredo, and R. Leão, "The TANGRAM-II Integrated Modeling Environment for Computer Systems and Networks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, no. 4, pp. 64–69, 2009, doi:10.1145/1530873.1530886.

Carlo K. da S. Rodrigues . é Doutor em Engenharia de Sistemas e Computação pela Universidade Federal do Rio de Janeiro (UFRJ, 2006) e Mestre em Sistemas e Computação pelo Instituto Militar de Engenharia (IME, 2000). É professor do Centro de Matemática, Computação e Cognição da Universidade Federal do ABC (UFABC), no curso de Ciência da Computação. Atua na subárea de Redes de Computadores. http://buscatextual.cnpq.br/buscatextual/visualizacv.do?id=K4721493T1.

Vladimir E. M. Rocha .é Doutor em Engenharia de Computação pela Escola Politécnica da Universidade de São Paulo (POLI-USP, 2017), e Mestre em Ciência da Computação pelo Instituto de Matemática e Estatística da mesma universidade (IME-USP, 2005). É professor do Centro de Matemática, Computação e Cognição (CMCC) da UFABC, no curso de Ciência da Computação. Atua em Sistemas Distribuídos. http://buscatextual.cnpq.br/buscatextual/visualizacv.do?id=K4753743J0.