Uma Avaliação da Tecnologia *Blockchain* considerando Eficiência e Segurança de Aplicações do Ecossistema IoT

Carlo K. da S. Rodrigues[1], **Vladimir E. M. Rocha**[1]

[1]Centro de Matemática, Computação e Cognição (CMCC)
Universidade Federal do ABC (UFABC) – Santo André – SP – Brasil

{carlo.kleber,vladimir.rocha}@ufabc.edu.br

Abstract. *This article assesses the use of the Blockchain technology in the implementation of the IoT-ecosystem database. Through queueing models and simulations, we carry out experiments involving different IoT-application domains. Average transaction-confirmation time, fraud probability, block-loss rate, and average number of queue blocks are chosen as performance metrics, which together provide us with inferences about efficiency, availability, and integrity requirements. The final results reveal that Blockchain may satisfactorily meet the previous requirements. In this sense, the main contribution of this article is thus to yield subsidies for IoT-application real projects. At last, general conclusions and future work conclude this article.*

Resumo. *Este artigo avalia a tecnologia Blockchain para implementação da base de dados do ecossistema IoT. Por meio de modelos analíticos de filas e simulações, são realizados experimentos envolvendo diferentes domínios de aplicações IoT. Tempo de confirmação, probabilidade de fraude, taxa de perda de blocos e número médio de blocos na fila são as métricas de desempenho consideradas, as quais propiciam inferências sobre o atendimento dos requisitos de eficiência, disponibilidade e integridade. Os resultados finais revelam que a Blockchain pode atender satisfatoriamente os requisitos anteriores. A principal contribuição deste artigo é, portanto, fornecer subsídios para projetos reais de aplicações IoT. Conclusões gerais e trabalhos futuros finalizam este artigo.*

1. Introdução

O ecossistema Internet das Coisas (do inglês, *Internet of Things* - IoT) deve permitir que atividades em uma sociedade moderna, tanto relacionadas ao indivíduo como aquelas de nível de sistema, sejam mais automatizadas e efetivamente executadas [Wang et al. 2019, Fernández-Caramés and Fraga-Lamas 2018, Farouk et al. 2020].

A maioria das aplicações desenvolvidas para o ecossistema IoT são baseadas no paradigma cliente-servidor, admitindo dispositivos inteligentes conectados a servidores na nuvem por meio da Internet [Fernández-Caramés and Fraga-Lamas 2018, Wang et al. 2019]. Neste paradigma, os dados coletados pelos dispositivos inteligentes são enviados para a nuvem por meio da rede de comunicação subjacente. A maior parte do processamento computacional é assim realizado na nuvem. Todavia, a centralização na nuvem é um gargalo devido ao número de dispositivos continuamente adicionados ao ecossistema [Fernández-Caramés and Fraga-Lamas 2018, Farouk et al. 2020].

Neste contexto, é previsto que mais de 50 bilhões de dispositivos estejam conectados até 2030 [Statista Research Department 2020b]. Ademais, alinhado com essa previsão, o mercado global de soluções IoT para usuários finais alcançou 100 bilhões de dólares em receita em 2017, devendo atingir 1,6 trilhão em 2025 [Statista Research Department 2020a].

Um paradigma alternativo ao cliente-servidor, sob a visão de arquitetura em camadas, considera a transferência de parte da função da nuvem, localizada em camada superior, para camadas intermediárias mais próximas da camada dos dispositivos inteligentes. Para tanto, são utilizados os conceitos de computação de borda e em névoa. Neste caso, tem-se *gateways* em camadas intermediárias que são capazes de processar os dados provenientes dos dispositivos inteligentes. Se necessário, esses *gateways* podem também interagir entre si e/ou com a própria nuvem [Fernández-Caramés and Fraga-Lamas 2018].

Ante a organização descentralizada e distribuída desse paradigma alternativo, a garantia de dois requisitos é um desafio: eficiência e segurança. A eficiência se traduz pela rapidez do processamento de transações provenientes dos dispositivos, e a segurança engloba os conceitos de confidencialidade, integridade e disponibilidade dos dados [Zhang and Jacobsen 2018]. Para atender esses requisitos, a academia e a indústria [Wang et al. 2019, Farouk et al. 2020] têm vislumbrado o uso da tecnologia de registros distribuídos *Blockchain* [S. Nakamoto 2008, Hunhevicz and Hall 2020].

Esta conjuntura é a motivação para este artigo, cujo objetivo é então avaliar a tecnologia *Blockchain* na implementação da base de dados do ecossistema IoT. Por meio de modelos de filas e simulações, são realizados experimentos envolvendo diferentes domínios de aplicações. Tempo de confirmação, probabilidade de fraude, taxa de perda de blocos e número médio de blocos na fila são as métricas de desempenho consideradas, as quais propiciam inferências sobre os requisitos de eficiência, disponibilidade e integridade. O requisito de confidencialidade não é alvo de estudo nesta pesquisa. Diante do objetivo anunciado, a principal contribuição deste artigo é, assim, prover novos subsídios teóricos para o desenvolvimento de projetos reais de aplicações IoT.

O restante deste artigo é organizado como segue. A Seção 2 revisa fundamentos teóricos da tecnologia *Blockchain* e da arquitetura IoT. A Seção 3 trata sobre trabalhos relacionados. Na Seção 4 são elencados alguns domínios de aplicações IoT, bem como os cenários considerados nos experimentos. A Seção 5 explica a modelagem realizada e discute os experimentos. Por fim, conclusões gerais e trabalhos futuros estão na Seção 6.

2. Fundamentos Teóricos

2.1. Tecnologia *Blockchain*

A tecnologia *Blockchain* surgiu em 2008, conjuntamente com o sistema de criptomoedas Bitcoin [S. Nakamoto 2008]. Seu objetivo é a construção descentralizada de uma lista encadeada de blocos de transações de clientes para a implementação de uma base de dados distribuída, considerando atributos de segurança fornecidos por criptografia [S. Nakamoto 2008, Wang et al. 2019, Farouk et al. 2020].

As transações são submetidas pelos clientes a uma rede de nós processadores, organizados sob arquitetura *peer-to-peer* (P2P). Cada nó processador trabalha coletando transações válidas [Decker and Wattenhofer 2013], construindo blocos, validando blocos

e interligando-os segundo uma lista encadeada [Wang et al. 2019]. Sendo uma lista encadeada, cada bloco referencia apenas o bloco anterior, denominado de bloco *pai*. Para tanto, usa-se o *hash* do cabeçalho do *pai*, o qual está contido dentro do cabeçalho do próprio bloco. A sequência de *hashes* que liga cada bloco ao seu *pai* constitui o caminho de volta até o primeiro bloco da lista, denominado de bloco *gênese*. A mudança da identificação de um bloco qualquer traz a necessidade de mudança da identificação de todos os blocos subsequentes.

A validação de cada bloco é executada por meio de um processo denominado de *mineração*. Este processo se baseia em um algoritmo de consenso do tipo *Proof of Work* (PoW) [Jakobsson and Juels 1999], o qual implica a necessidade de determinar um valor *nonce* que é solução de um desafio criptográfico expresso em termos da função *hash* criptográfica SHA-256. Este desafio possui um parâmetro chamado de *alvo de dificuldade*, D. Quanto maior (menor) é D, menor (maior) é o número de *hashes* a computar, fazendo com que o tempo de *mineração*, δt, seja menor (maior) [Bowden et al. 2018].

Após a *mineração*, o bloco é adicionado à lista local do nó processador que executou a *mineração* e, na sequência, é disseminado pela rede P2P para que os demais nós processadores possam atualizar as suas respectivas listas locais, permitindo a convergência da base de dados distribuída. A *mineração* pode também ser executada por um grupo de nós processadores em vez de um único nó. Neste caso, o grupo de nós processadores constitui o chamado *mining pool*, onde nós processadores trabalham cooperativamente.

2.2. Visão em Camadas da Arquitetura IoT

As quatro camadas hierarquizadas, sinteticamente descritas a seguir, formam a arquitetura do ecossistema IoT [Dai et al. 2019, Moin et al. 2019, Farouk et al. 2020].

1. *Camada de aplicação*: usa as informações provenientes da camada de processamento. Esta camada engloba as aplicações dos inúmeros domínios possíveis, e.g., transporte e mobilidade, logística, meio ambiente, cidades inteligentes, vigilância, e Indústria 4.0. Sua implementação é baseada no conceito de computação em nuvem. É virtualmente a camada mais próxima dos usuários humanos finais.
2. *Camada de processamento*: processa e armazena os dados vindos da camada de rede, assim obtendo informações que podem ser utilizadas por camadas superiores ou inferiores. As principais técnicas de implementação incluem, e.g., *Big Data*, processamento inteligente, computação de borda, e computação em névoa.
3. *Camada de rede*: transfere os dados da camada de percepção para a camada de processamento, podendo empregar diversos tipos de infraestrutura de rede, e.g., cabeadas, sem fio, móveis, veiculares, *mesh*, etc. Algumas das tecnologias empregadas na implementação incluem, e.g., 4G/5G, Wi-Fi, Bluetooth e ZigBee.
4. *Camada de percepção*: contempla os dispositivos eletroeletrônicos que podem monitorar eventos físicos e, se desejável, tomar decisões.

3. Trabalhos Relacionados

Esta seção discorre sobre alguns trabalhos relacionados, direta ou indiretamente, ao tema de pesquisa deste artigo. O objetivo é prover uma visão essencial da literatura disponível.

Em [S. Nakamoto 2008], tem-se a proposta original da tecnologia *Blockchain*. 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 da

tecnologia, cujos resultados revelam uma promissora resiliência a ataques de fraudes. O foco do trabalho se limita, contudo, ao sistema de criptomoedas Bitcoin.

Como *surveys* sobre o emprego de *Blockchain* para a base de dados do ecossistema IoT, os trabalhos de [Fernández-Caramés and Fraga-Lamas 2018, Wang et al. 2019, Dai et al. 2019] são bem valiosos. Além de aspectos conceituais, é discutida uma diversidade de aplicações IoT baseadas em *Blockchain*, ressaltando vantagens comparativas ao paradigma centrado na nuvem e desafios de implementações reais, incluindo segurança.

Considerando propostas de *frameworks*, cita-se o trabalho de [Truong et al. 2019]. Seu diferencial está na apresentação de experimentos baseados em prototipagem usando FIWARE como plataforma IoT, e a estrutura Hyperledger Fabric para a base de dados. Os resultados experimentais mostram um desempenho adequado para operações de busca e inserção de dados, mas não há discussão explícita sobre segurança.

Também em contexto mais especifico, tem-se o trabalho de [Hang and Kim 2019]. É apresentada uma plataforma IoT integrada com a tecnologia *Blockchain*. O objetivo é oferecer ao proprietário do dispositivo inteligente um esquema descentralizado no qual as informações são armazenadas em uma base de dados abrangente e imutável, além de serem compartilhadas rapidamente. Seu diferencial é a validação da proposta por meio de prova de conceito em cenários IoT, onde são utilizados dispositivos Raspberry Pi e a estrutura Hyperledger Fabric para a base de dados. Os experimentos constatam adequado tempo de processamento de transações, mas não trata-se de segurança.

Por fim, em [Akbari et al. 2020], os autores analisam a tecnologia *Blockchain*, avaliando o impacto que o tamanho do bloco de transações e o intervalo de tempo para adição de blocos têm sobre o *throughput* (em termos de transações por segundo), bem como sobre a segurança (em termos de blocos processados mas não adicionados) do sistema. Embora os resultados das simulações realizadas permitam obter estimativas de valores para uma configuração sistêmica de desempenho satisfatório, os cenários modelados são restritos e não abordam explicitamente o ecossistema IoT e as suas aplicações.

Ante os trabalhos mencionados, o diferencial desta pesquisa se revela então pelo emprego conjunto de modelos de filas e simulações para avaliar a eficiência do processamento de transações no ecossistema IoT, considerando limites de tempo de resposta, bem como a segurança associada, sob os requisitos de integridade e disponibilidade.

4. Domínios de Aplicações IoT e Cenários

Os domínios de aplicações IoT consistem de variados tipos de serviços e apresentam diferentes características associadas. Com base nos trabalhos de [Mocnej et al. 2018a, Mocnej et al. 2018b, Mocnej et al. 2018c], a Tabela 1 resume oito populares domínios, e a Tabela 2 caracteriza oito cenários S_j, para $1 \leq j \leq 8$, cada um representando uma aplicação IoT específica. Os parâmetros da Tabela 2 são explicados na Seção 5.

Ademais, o ecossistema considerado nos experimentos deste trabalho é formado por um cenário único S_U, o qual resulta da integração dos oito cenários S_j mencionados. Sob uma visão de arquitetura em camadas, a Figura 1 ilustra o ecossistema em questão. Investigações de cenários específicos independentes são deixadas como trabalhos futuros.

Tabela 1. Domínios de aplicações IoT

Domínio	Aplicação considerada	Tamanho da rede	Tx. de transação (ind.)	Retardo
Assistência de saúde	Monitoramento de pacientes	100-1.000 dispositivos	1/10 seg	3 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
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
Prédios e lares	Controle de iluminação	10-1.000 dispositivos	1/15 min	5 seg
Meio ambiente	Rastreamento de animais	100-1.000 dispositivos	1/30 min	algumas horas
Agricultura	Irrigação	10-1.000 dispositivos	1/h	1 min

Tabela 2. Cenários de aplicações IoT

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

5. Avaliação de Desempenho: Modelagem e Experimentos

A avaliação realizada a seguir está restrita à proposta original da *Blockchain*, considerando a base de dados na categoria *pública* [Fernández-Caramés and Fraga-Lamas 2018] e o algoritmo de consenso do tipo PoW [Jakobsson and Juels 1999]. Além disso, o foco de atenção é direcionado para a camada de processamento, na qual assumem-se localizados os nós processadores da rede P2P referentes à tecnologia *Blockchain*.

Para adequada organização, a discussão é feita em duas subseções. A Subseção 5.1 trata dos requisitos de eficiência e disponibilidade, abordando as seguintes questões: Quanto tempo leva para uma transação de um dispositivo inteligente ser confirmada, i.e., processada na camada de processamento? Qual a disponibilidade do sistema quando servidores (i.e., nós processadores) ficam inoperantes, i.e., como o tempo de confirmação da transação é afetado devido à inoperância de servidores? Por sua vez, a Subseção 5.2 se dedica à integridade de dados e se desenvolve em torno da seguinte questão: Qual a probabilidade de um ataque modificar as transações registradas na lista encadeada?

5.1. Avaliação de Eficiência e Disponibilidade

Assuma que cada cenário S_j, $j \in \{1, 2, \ldots, 8\}$ (vide Tabela 2), é modelado por uma fila do tipo $M/M/1/\infty/FIFO$ [Kleinrock 1975], como ilustrado na Figura 2(a) e explicado a seguir. O processo de chegada de cada cenário S_j se refere a blocos de transações, vindos da camada de rede e em direção à camada de processamento. As transações são originadas pelos dispositivos inteligentes d_i da camada de percepção, para $1 \leq i \leq n_j$,

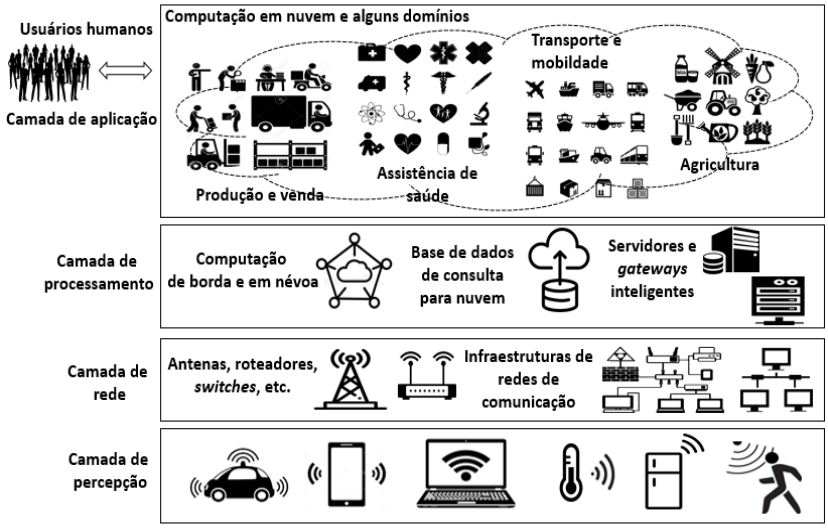


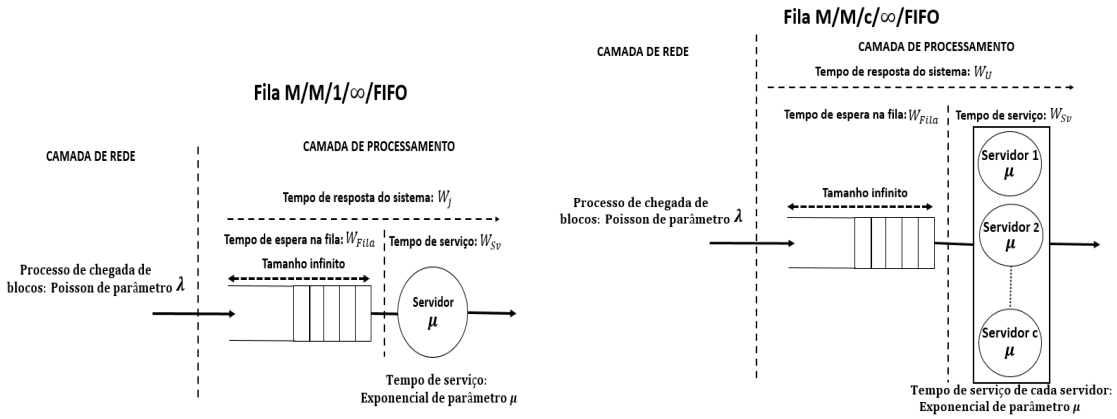
Figura 1. Exemplificação do ecossistema IoT em análise.

onde n_j é o limite superior do tamanho da rede do cenário S_j, medido em número de dispositivos. Esse processo é do tipo Poisson [Trivedi 2002], com taxa $\lambda_j = (\sum_{i=1}^{n_j} \lambda_i)/L_j$, onde λ_i é a taxa de transações do dispositivo d_i, e L_j é o tamanho do bloco de transações de cada cenário S_j, medido em número de transações.

Para o processamento dos blocos de cada cenário S_j, assume-se existência de um servidor único na camada de processamento, cujo tempo de serviço tem distribuição exponencial de parâmetro μ_j. Esse servidor representa o conjunto dos nós processadores da rede P2P da *Blockchain*. Ademais, os blocos que chegam nunca são descartados, pois a capacidade de armazenamento é infinita (i.e., fila infinita), e são selecionados para serviço conforme ordem de chegada, i.e., disciplina FIFO (*First-In, First-Out*). O tempo médio de confirmação da transação de cada cenário S_j é estimado pelo tempo médio 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 inferiores) quando o correspondente bloco que a contém tem seu processamento finalizado (i.e., o bloco é *minerado*). Em específico, o valor de retardo tolerável no cenário S_j é indicado pelo parâmetro $Max\,W_j$ na Tabela 2.

Para avaliar eficiência e disponibilidade, assuma então um cenário único S_U, o qual agrega as capacidades de processamento individuais de todos os cenários independentes S_j, para $1 \leq j \leq 8$. A consideração de um cenário único S_U, em vez da análise dos cenários independentes S_j, se justifica pelo entendimento de que a implantação de uma única infraestrutura de rede de capacidade de processamento nominal P tende a ser uma solução mais efetiva (em termos de custo, eficiência e segurança) que a implantação de j infraestruturas independentes de menor capacidade individual cada, que juntas totalizam a mesma capacidade nominal P.

O cenário S_U é modelado por um sistema de fila do tipo $M/M/c/\infty/FIFO$ [Kleinrock 1975], como ilustrado na Figura 2(b). O processo de chegada novamente se refere a blocos de transações, que são provenientes da camada de rede e vão em direção à camada de processamento. As transações são originadas coletivamente pelos dispositivos



(a) Fila $M/M/1/\infty/FIFO$. (b) Fila $M/M/c/\infty/FIFO$.

Figura 2. Modelagem da camada de processamento do ecossistema.

de todos os cenários S_j, para $1 \leq j \leq 8$. Com base nas propriedades de superposição e decomposição do processo de Poisson [Trivedi 2002], a chegada de blocos no cenário S_U segue também um processo de Poisson, com taxa $\lambda_U = (\sum_{j=1}^{8} \lambda_j)/L_U$, onde λ_j é, como antes, a taxa de chegada de transações de cada cenário S_j, e L_U é o tamanho do bloco de cada cenário S_U, medido em número de transações.

A camada de processamento do cenário S_U possui c servidores independentes (i.e, um servidor para cada cenário S_j), cujos tempos de serviço individuais são idênticos, possuindo cada um 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. Ademais, seja n o número de blocos no sistema (i.e., na fila e em serviço). A taxa de serviço do sistema é $n.\mu$ (para $0 \leq n < c$) ou $c.\mu$ (para $n \geq c$). 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 na ordem de chegada, i.e., disciplina FIFO.

Para a avaliação específica de eficiência, o tempo médio de confirmação da transação no cenário S_U é estimado pelo tempo médio de resposta do sistema, W_U, sob a visão do bloco de transações. Isso porque, como antes, a transação somente se torna disponível para consulta pelas camadas superiores (ou inferiores) quando o correspondente bloco que a possui tem seu processamento finalizado (i.e, o bloco é *minerado*). Neste contexto, a Equação 1 [Kleinrock 1975] calcula W_U para o cenário S_U, onde W_{Fila} é o tempo de espera na fila, W_{Sv} é o tempo de serviço, e N_{Fila} é o número de blocos na fila (i.e., aguardando serviço). Em seu turno, o valor de N_{Fila} é calculado pela Equação 2, 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 pela Equação 3. Lembramos que os valores de λ_j para o cálculo de λ_U estão na Tabela 2, o valor de c é igual a oito, e o valor de μ é igualado a 1/10 min para simples comparação com a *Blockchain* do sistema Bitcoin [Bowden et al. 2018].

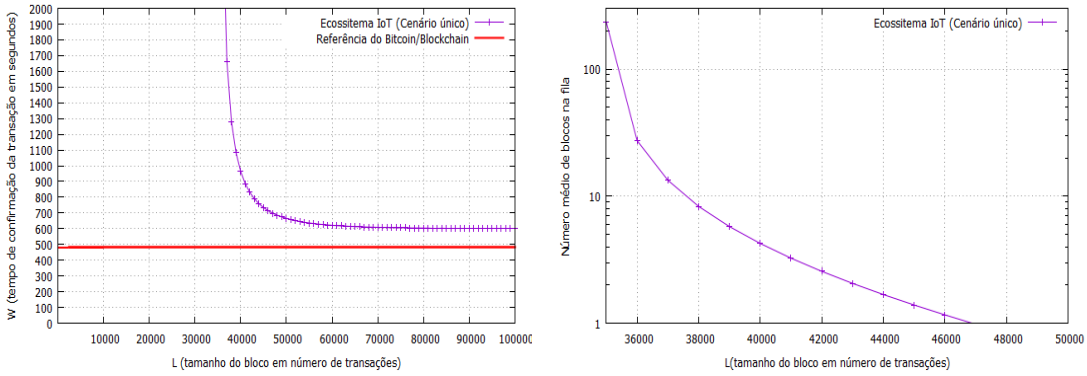
$$W_U = W_{Fila} + W_{Sv} = \frac{N_{Fila}}{\lambda_U} + \frac{1}{\mu} \tag{1}$$

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

$$P_0 = \frac{1}{\sum_{n=0}^{c-1}\frac{r^n}{n!} + \frac{r^c}{c!(1-\rho)}} \tag{3}$$

As Figuras 3(a) e 3(b) trazem, respectivamente, os resultados de W_U e N_{Fila} em função de L_U. Em particular, o valor referente ao sistema Bitcoin está destacado na primeira das figuras [Blockchain.com 2020]. Desses resultados, tem-se que W_U converge para 600 seg quando $L_U \geq 60.000$, e que $N_{Fila} \leq 1$ para $L_U \geq 47.000$. Todavia, sob o aspecto da eficiência, o valor de W_U está acima da maioria dos valores toleráveis indicados por $Max\, W_j$, para $1 \leq j \leq 8$ (vide Tabela 2). Assim, o cenário S_U não tem a eficiência necessária para justificar seu emprego.

Vislumbram-se então três alternativas para diminuir W_U. Como primeira alternativa, tem-se o aumento da capacidade de processamento. Isso pode ser conseguido pelo emprego de um maior número de servidores ou de servidores de maior capacidade de processamento. Existe neste caso um custo financeiro. Uma segunda alternativa consiste na implantação de uma política de admissão de transações. Para tanto, pode-se limitar a capacidade máxima de armazenamento de blocos que chegam para processamento. Neste caso, há uma taxa de perda de blocos que precisa ser analisada. Finalmente, a terceira alternativa é o ajuste do *alvo de dificuldade*, D (vide Subseção 2.1). Esta alternativa não implica custo financeiro, tampouco ocasiona perda de blocos.

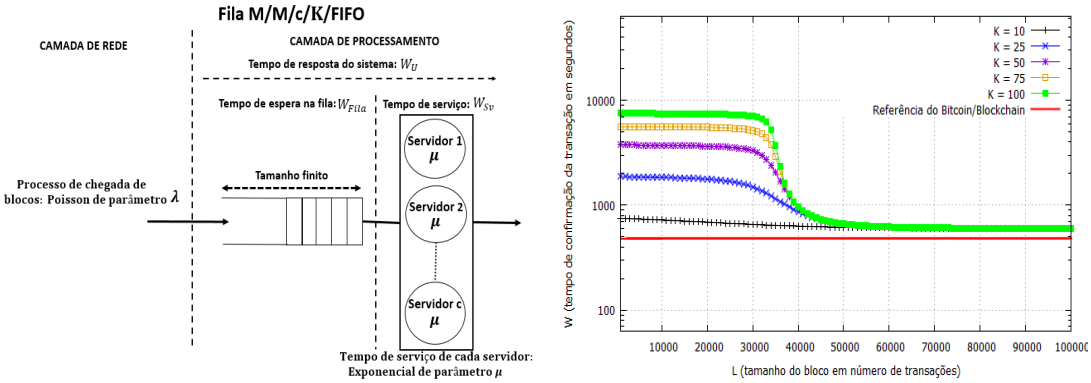


(a) Tempo de confirmação da transação. (b) Número médio de blocos em espera.

Figura 3. Camada de processamento sem limite de armazenamento.

As três alternativas supracitadas não são mutuamente exclusivas, podendo ser aplicadas em conjunto. Todavia, para evitar discussão sobre custo financeiro, discute-se na sequência a combinação apenas da segunda e da terceira alternativas, deixando-se a primeira como trabalhos futuros. Para essa discussão, assuma o cenário S_U agora modelado por um sistema de fila do tipo $M/M/c/K/FIFO$ [Kleinrock 1975], como ilustrado na Figura 4(a). Essa nova modelagem tem uma descrição similar à anterior, com a exceção principal de que a fila tem tamanho finito. O parâmetro associado a essa restrição é K, o qual define um limite superior para o número total de blocos no sistema, n (i.e., blocos na fila mais blocos em serviço). Na chegada de um bloco, é feita a seguinte verificação:

se $n < K$, então o bloco é aceito; caso contrário, é rejeitado (i.e., perdido). A taxa de serviço é, portanto, $n.\mu$, para $0 \leq n < c-1$, e $c.\mu$, para $c \leq n \leq K$. Os blocos aceitos são então servidos na ordem de chegada, i.e., disciplina FIFO.



(a) Modelagem por fila. (b) Tempo de confirmação da transação.

Figura 4. Camada de processamento com armazenamento limitado.

Para a avaliação pretendida, tem-se o valor de W_U calculado pela Equação 4 [Kleinrock 1975], onde, como antes, W_{Fila} é o tempo de espera na fila, W_{Sv} é o tempo de serviço, e N_{Fila} é o número de blocos na fila. Em seu turno, o valor de N_{Fila} é calculado pela Equação 5, onde: $r = \lambda_U/\mu$; $\rho = r/c$; P_K é a probabilidade de haver K blocos no sistema (vide Equação 6); e P_0 é a probabilidade de não haver blocos no sistema (vide Equação 7). Lembramos que os valores de λ_j para o cálculo de λ_U estão na Tabela 2, o valor de c é igual a oito, e o valor de μ é igualado a 1/10 min para fins de comparação com a *Blockchain* do sistema Bitcoin [Bowden et al. 2018].

$$W_U = W_{Fila} + W_{Sv} = \frac{N_{Fila}}{\lambda_U(1-P_K)} + \frac{1}{\mu} \tag{4}$$

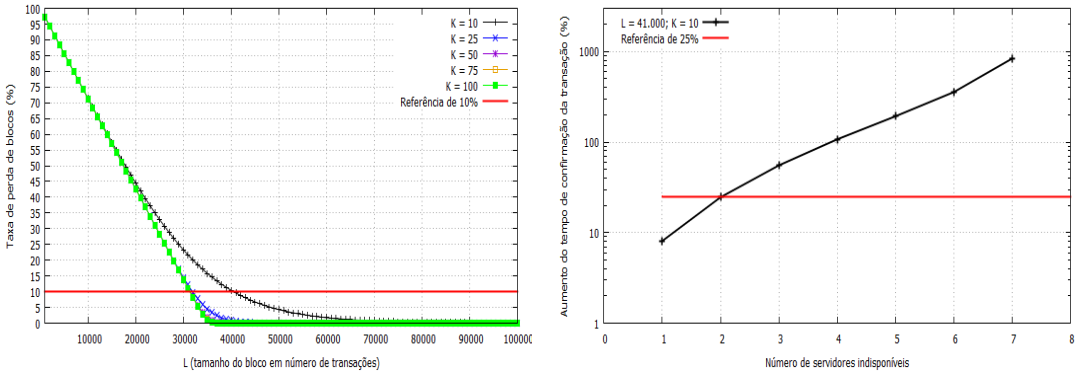
$$N_{Fila} = \begin{cases} \frac{P_0\, r^c}{c!} . \frac{\rho\,(K-c+1)(K-c)}{2} & \text{, se } \rho = 1 \\ \frac{P_0\, r^c}{c!} . \frac{\rho\,(1-\rho^{K-c+1}) - (K-c+1)\,\rho^{(K-c)}\,(1-\rho)}{(1-\rho)^2} & \text{, se } \rho \neq 1 \end{cases} \tag{5}$$

$$P_K = \frac{P_0(\lambda_U)^K}{c!\,\mu^K\,c^{(K-c)}} \tag{6}$$

$$P_0 = \begin{cases} \frac{1}{\sum_{n=0}^{c-1} \frac{r^n}{n!} + \frac{r^c\,(K-c+1)\,\rho}{c!}} & \text{, se } \rho = 1 \\ \frac{1}{\sum_{n=0}^{c-1} \frac{r^n}{n!} + \frac{r^c\,(1-\rho^{(K-c+1)})}{c!\,(1-\rho)}} & \text{, se } \rho \neq 1 \end{cases} \tag{7}$$

Para avaliar a eficiência, tem-se as Figuras 4(b) e 5(a). A primeira traz W_U em função de L_U, para diferentes valores de K. Os resultados mostram que $W_U \rightarrow 600$ seg para $L_U > 50.000$. Além disso, tem-se que $K = 10$ é o valor que mais rapidamente propicia essa convergência. A segunda figura tem a taxa de perda de blocos, calculada

como $(P_K \times 100)\%$. Essa taxa é inferior a 10% para $L_U < 41.000$, independentemente de K. Uma adequada relação de compromisso entre tempo de confirmação e taxa de perda pode então ser obtida pela tupla $(L_U = 41.000, K = 10)$, fazendo este sistema mais eficiente que o de armazenamento ilimitado: menor W_U para um mesmo L_U, desde que admita-se aceitável uma taxa de perda de até 10%.



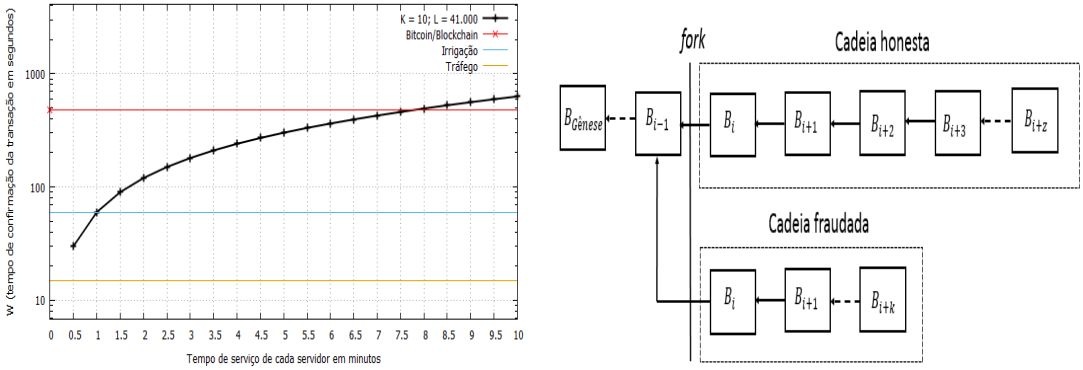
(a) Taxa de perda de blocos. (b) Aumento do tempo de confirmação (%).

Figura 5. Camada de processamento com armazenamento limitado.

Para avaliar a disponibilidade, tem-se a Figura 5(b). Nesta figura são apresentados os aumentos percentuais sobre o valor de W_U em função do número de servidores inoperantes, considerando a tupla $(L_U = 41.000, K = 10)$. Desses resultados, é possível notar aumentos percentuais não desprezíveis, revelando a pouca robustez. Por exemplo, de um total de oito servidores, a inoperância de dois servidores produz um aumento no tempo de confirmação de 25%, e de quatro servidores produz um aumento de 100%. Em síntese, esses resultados permitem constatar que a disponibilidade do sistema pode, e.g., ser seriamente afetada por ataques de negação de serviço [Heinrich and Obelheiro 2019]. Porém, a mitigação dessa vulnerabilidade pode ser conseguida, e.g., pela adição de servidores de contingência, cuja análise é deixada como trabalhos futuros.

Para terminar esta subseção, discute-se novamente a eficiência, considerando agora o ajuste de D. Como visto na Subseção 2.1, o aumento de D implica um menor tempo de *mineração*, δt, e portanto um menor tempo de confirmação de transações. Ante a modelagem realizada, isso significa reduzir o tempo de serviço de cada servidor, $1/\mu$. Com isso em mente e assumindo a tupla $(L_U = 41.000, K = 10)$, a análise a seguir avalia indiretamente o impacto de D sobre W_U pelo impacto de $1/\mu$ sobre W_U.

A Figura 6(a) traz os resultados de W_U em função de $1/\mu$. Para referência, destacam-se o tempo de confirmação do sistema Bitcoin e os valores de $Max\,W_j$ para as aplicações de irrigação e de controle de tráfego (vide Tabela 2). Os resultados mostram que o ajuste de D produz uma otimização em W_U, viabilizando o atendimento dos limites de tempo de confirmação. Mas, o aumento de D não pode prescindir de uma avaliação de integridade, pois também aumenta-se a chance de haver blocos *minerados* simultaneamente, ocasionado bifurcações na lista encadeada (i.e., na cadeia de blocos) que podem impactar a segurança [Akbari et al. 2020]. A garantia de integridade se refere à certeza de que as informações armazenadas na camada de processamento não serão adulteradas por fraudadores. Essa discussão está na próxima subseção.



(a) Tempo de confirmação em função de $1/\mu$. (b) Cadeia de blocos na camada de processamento.

Figura 6. Tempo de confirmação e construção da cadeia de blocos.

5.2. Avaliação de Integridade

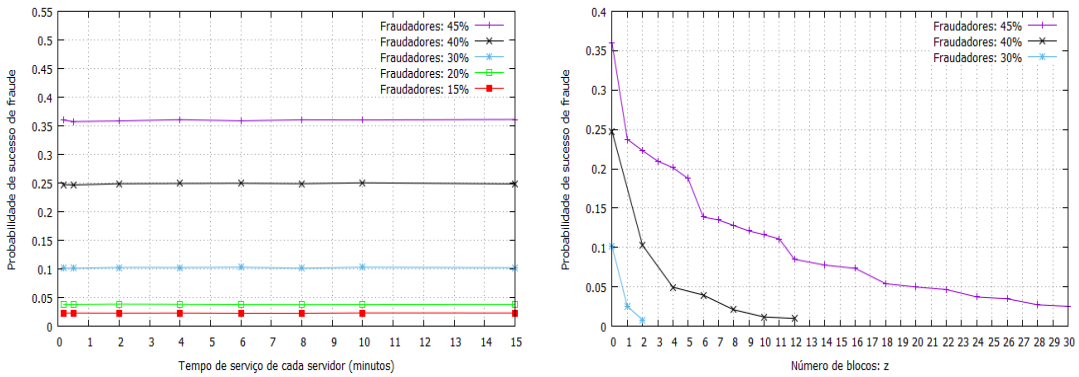
A avaliação a seguir admite o ataque de um processador fraudador da rede (ou conjunto de processadores fraudadores) durante a construção da cadeia de blocos na camada de processamento do cenário S_U, com fila de tamanho finito (vide seção anterior). A modelagem desse ataque é ilustrada na Figura 6(b) e explicada a seguir.

Seja B_i o bloco adicionado à cadeia da camada de processamento em $t = t_0$. Assuma que B_i é um bloco legítimo, i.e., que contém transações verdadeiras, 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 (i.e., legítima): (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 com o bloco B_i adulterado: (B_i, B_{i+1}, B_{i+2}, ..., B_{i+k}). A cadeia fraudada é construída no intervalo 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). Quando da ocorrência de *forks*, a regra da cadeia mais longa [Wang et al. 2019] estabelece que, dentre as cadeias concorrentes, aquela que se tornar a mais longa deve ser a considerada.

A partir de $t = t_0 + \Delta t$, há então uma disputa entre processadores honestos e processadores fraudadores, os quais adicionam blocos às cadeias honesta e fraudada, respectivamente, ao longo do tempo. Essa disputa 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. Doravante, assuma que o poder de processamento dos processadores honestos é sempre maior que aquele dos fraudadores. Isso posto, quanto mais cedo for realizado o ataque, menor é a diferença $z-k$ inicial e, portanto, maior é a probabilidade de sucesso de fraude. Assuma, ainda, que os processadores honestos e os fraudadores têm respectivamente probabilidades p e $q = (1 - p)$ de adicionar o próximo bloco. Essas probabilidade refletem o percentual do poder computacional total do sistema pertencente aos processadores honestos, i.e., $(p \times 100)\%$, e aos fraudadores, i.e., $(q \times 100)\%$. Para análise, considere que a fraude tem sucesso quando as cadeias honesta e fraudada se igualam em tamanho.

A modelagem explicada é resolvida por simulação no ambiente Tangram-II [de Souza e Silva et al. 2009]. Este ambiente foi concebido pela Universidade Federal do Rio de Janeiro (UFRJ), com a participação da Universidade da Califórnia em Los Angeles (UCLA), e serve para modelagem e análise de sistemas computacionais e de comunicação. Os resultados da simulação apresentados 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 Figura 7(a) traz a probabilidade de fraude (i.e., probabilidade de sucesso de fraude) em função de $1/\mu$. Consideram-se a tupla $(L_U = 41.000, K = 10)$, conforme discutido na subseção anterior, e $\Delta t \to 0$ (portanto, $z = 0$), por ser a situação mais favorável à ocorrência de fraude, como já explicado. São comparados cinco diferentes cenários, individualmente caracterizados pelos percentuais do poder de processamento de fraudadores em relação ao poder total do sistema. Desses resultados, tem-se o seguinte. Primeiro, a probabilidade de fraude pouco difere em função de $1/\mu$ no intervalo de 10 seg até 15 min. Segundo, a probabilidade de fraude somente excede $\approx$ 0,1 quando o poder dos fraudadores compromete mais que 30% do processamento do sistema. Em síntese, D pode então ser ajustado para se ter o tempo de confirmação exigido, sem prejudicar a integridade (i.e., probabilidade de fraude menor que $\approx 0,1$), desde que o poder de processamento do sistema não seja controlado por fraudadores em mais que 30%.



(a) Probabilidade de sucesso de fraude. (b) Mitigação de fraude.

Figura 7. Fraude na camada de processamento com armazenamento limitado.

Para maior garantia de integridade, ataques devem ocorrer somente em $\Delta t > 0$ (e, portanto, $z > 0$). Para essa avaliação, apresenta-se então a Figura 7(b). Por exemplo, para uma probabilidade de fraude abaixo de 0,05, tem-se: $z = 1$ (fraudadores: 30%); $z = 4$ (fraudadores: 40%); $z = 20$ (fraudadores: 45%). Daí, veja que é possível garantir integridade mesmo em situações bem desfavoráveis. Por outro lado, fazer $z > 0$ implica impactar o tempo de confirmação da transação: o correspondente bloco da transação somente é aceito no sistema após z blocos subsequentes serem adicionados à cadeia. Em síntese, existe uma relação de compromisso entre tempo de confirmação (i.e., eficiência) e garantia de integridade, cuja discussão é deixada como trabalhos futuros.

6. Conclusões e Trabalhos Futuros

Este artigo avaliou a eficiência e a segurança de aplicações IoT, considerando o uso da tecnologia *Blockchain* na implementação da base de dados do ecossistema. Usando modelos de filas e simulações, diferentes cenários de IoT foram examinados.

Os experimentos mostraram que a *Blockchain* pode atender os requisitos de tempo de confirmação de transações, com satisfatórios níveis de integridade e disponibilidade. Para tanto, relações de compromisso são estabelecidas. Neste contexto, destacam-se: (i) o tamanho do bloco deve ser de 60.000 transações (ou 41.000 transações para taxa de perda de até 10%); (ii) a complexidade da *mineração* dos blocos pode ser reduzida sem prejudicar a integridade (i.e., prob. de fraude menor que $\approx 0,1$), desde que o poder de processamento de fraudadores não exceda 30% do total; (iii) maior garantia de integridade (i.e., prob. de fraude $< 0,05$) pode ser alcançada para requisitos de tempo mais flexíveis; e (iv) servidores de contingência são recomendáveis para adequada disponibilidade.

Como trabalhos futuros e abarcando limitações desta pesquisa, sugerem-se a análise de algoritmos de consenso diferentes do PoW, uma modelagem mais ampla considerando toda a arquitetura IoT, e a comparação de tecnologias diferentes da *Blockchain*.

Referências

Akbari, E., Zhao, W., Yang, S., and Luo, X. (2020). The Impact of Block Parameters on the Throughput and Security of Blockchains. In *2nd International Conference on Blockchain Technology*, Hilo, HI, USA.

Blockchain.com (2020). Blockchain Charts. [Online]. Available at: https://www.blockchain.com/pt/charts. Accessed on: June 2nd, 2020.

Bowden, R., Keeler, H. P., Krzesinski, A. E., and Taylor, P. G. (2018). Block arrivals in the bitcoin blockchain. [Online] Available at: https://arxiv.org/abs/1801.07447. Accessed on: May. 17th, 2020.

Dai, H., Zheng, Z., and Zhang, Y. (2019). Blockchain for Internet of Things: A Survey. *IEEE Internet of Things Journal*, 6(5):8076–8094.

de Souza e Silva, E., Figueiredo, R., and Leão, R. (2009). The TANGRAM-II Integrated Modeling Environment for Computer Systems and Networks. *ACM SIGMETRICS Performance Evaluation Review*, 36(4):64–69.

Decker, C. and Wattenhofer, R. (2013). Information propagation in the Bitcoin network. In *IEEE International Conference on Peer-to-Peer Computing*, Trento, Italy.

Farouk, A., Alahmadi, A., Ghose, S., and Mashatan, A. (2020). Blockchain platform for industrial healthcare: Vision and future opportunities. *Computer Communications*, 154:223 – 235.

Fernández-Caramés, T. M. and Fraga-Lamas, P. (2018). A Review on the Use of Blockchain for the Internet of Things. *IEEE Access*, 6:32979–33001.

Hang, L. and Kim, D. (2019). Design and Implementation of an Integrated IoT Blockchain Platform for Sensing Data Integrity. *Sensors*, 10:2228.

Heinrich, T. and Obelheiro, R. R. (2019). Brasil vs Mundo: Uma Análise Comparativa de Ataques DDoS por Reflexão. In *Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg)*, São Paulo, SP, Brasil.

Hunhevicz, J. J. and Hall, D. M. (2020). Do you need a blockchain in construction? Use case categories and decision framework for DLT design options. *Advanced Engineering Informatics*, 45:101094.

Jakobsson, M. and Juels, A. (1999). *Proofs of Work and Bread Pudding Protocols (Extended Abstract)*, pages 258–272. Springer US, Boston, MA.

Kleinrock, L. (1975). *Queuing Systems. Volume I: Theory*. Wiley, New York.

Mocnej, J., Miskuf, M., Papcun, P., and Zolotová, I. (2018a). Impact of Edge Computing Paradigm on Energy Consumption in IoT. *IFAC-PapersOnLine*, 51(6):162 – 167. 15th IFAC Conference on Programmable Devices and Embedded Systems PDeS 2018.

Mocnej, J., Pekar, A., Seah, W. K. G., and Zolotová, I. (2018b). Network Traffic Characteristics of the IoT Application Use Cases. Technical Report ECSTR18-01, School of Engineering and Computer Science, Victoria University of Wellington.

Mocnej, J., Seah, W. K., Pekar, A., and Zolotová, I. (2018c). Decentralised IoT Architecture for Efficient Resources Utilisation. *IFAC-PapersOnLine*, 51(6):168 – 173. 15th IFAC Conference on Programmable Devices and Embedded Systems PDeS 2018.

Moin, S., Karim, A., Safdar, Z., Safdar, K., Ahmed, E., and Imran, M. (2019). Securing IoTs in distributed blockchain: Analysis, requirements and open issues. *Future Generation Computer Systems*, 100:325 – 343.

S. Nakamoto (2008). Bitcoin: A peer-to-peer electronic cash system. Available at: https://bitcoin.org/bitcoin.pdf. Accessed on: Apr. 2nd, 2020.

Statista Research Department (2020a). Global IoT end-user spending worldwide 2017-2025. [Online]. Available at: https://www.statista.com/statistics/976313/global-iot-market-size/statisticContainer. Accessed on: Apr. 4th, 2020.

Statista Research Department (2020b). Number of Internet of Things (IoT) connected devices worldwide in 2018, 2025 and 2030. [Online]. Available at: https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology/. Accessed on Apr. 4th, 2020.

Trivedi, K. S. (2002). *Probability and Statistics with Reliability, Queuing and Computer Science Applications*. John Wiley & Sons, New York, second edition.

Truong, H. T. T., Almeida, M., Karame, G., and Soriente, C. (2019). Towards Secure and Decentralized Sharing of IoT Data. In *IEEE International Conference on Blockchain*, Atlanta, GA, USA.

Wang, Q., Zhu, X., Ni, Y., Gu, L., and Zhu, H. (2019). Blockchain for the IoT and industrial IoT: A review. *Internet of Things*, page 100081.

Wang, W., Hoang, D. T., Hu, P., Xiong, Z., Niyato, D., Wang, P., Wen, Y., and Kim, D. I. (2019). A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks. *IEEE Access*, 7:22328–22370.

Zhang, K. and Jacobsen, H. (2018). Towards Dependable, Scalable, and Pervasive Distributed Ledgers with Blockchains. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, Vienna, Austria.