Performance Modeling of the Consensus Mechanism in a Permissioned Blockchain

Udo R. Krieger[1(✉)], Michael H. Ziegler[1], and Hendrik L. Cech[2]

[1] Fakultät WIAI, Otto-Friedrich-Universität,
An der Weberei 5, 96047 Bamberg, Germany
udo.krieger@ieee.org
[2] Fakultät Informatik, Technische Universität München,
85748 Garching, Germany

Abstract. We consider a permissioned blockchain and analyze the dissemination and commitment processes of blocks among its corresponding miner nodes in the underlying peer-to-peer network. We propose a Markovian non-purging (n, k) fork-join queueing model to analyze the delay performance of the synchronization process among these miner nodes that apply a vote-based consensus procedure. We determine the impact of the most influential design and load parameters on the resulting commitment delay of new blocks that are appended to the blockchain after successful commitment decisions and the approval by the fully distributed consensus procedure. The proposed analysis of a permissioned blockchain is illustrated by means of a simple example of a fully interconnected P2P graph applying mean-value analysis techniques.

Keywords: Blockchain · Consensus mechanism · Performance modeling · Fork-join queueing network · Mean-value analysis

1 Introduction

The *Internet-of-Things* (IoT) describes the fundamental paradigm shift of enhancing previously analog devices and their associated gateways to the Internet with effective computing, storage and networking capabilities (cf. [1,2,15]). Combined with the paradigm of fog and edge computing, the Internet-of-Things will provide the fast growing basis for new, rapidly evolving application scenarios of the blockchain technology (cf. [5,11,12,15,22]). A fog computing architecture integrating the basic functionality of a scalable blockchain framework like Plasma or Multichain [27] constitutes the major motivation of our research regarding the performance analysis of the blockchain technology (cf. [9,25]).

In recent years, major research efforts have been devoted to public-permissionless blockchains such as Bitcoin [17] or Ethereum [6] and the basic functionality of the involved consensus protocols (cf. [20]). The latter process is

governed by a block validation. It is influenced by the properties of the underlying public-key cryptographic system that is controlling the pseudonymous interactions among the clients based on their emitted transactions. It depends also on the executed functionality of the communicating partners that are associated with these cryptographically signed transactions and smart contracts in the related peer-to-peer (P2P) overlay network of a blockchain. This distributed dissemination and synchronization processes regarding the aggregation of transactions into so-called blocks and the associated validation of transactions and their blocks by the subset of all mining nodes, called the miners or validators, constitute core elements of the distributed database functionality. It is applied to store the non-immutable transaction history in the blockchain.

Considering the application of blockchain technology in advanced Internet-of-Things scenarios, a permissioned blockchain such as Hyperledger Fabric [3] or Multichain [27] that use different variants of a lightweight vote-based consensus scheme constitutes a more adequate alternative to a permissionless blockchain with its heavyweight protocol (cf. [12,23]). Regarding the performance of the blockchain dynamics and thereby induced vulnerabilities due to long-tailed communication delays, only a few studies have applied a queueing-theoretical framework to analyze the underlying consensus mechanism and its implications on the blockchain (see [13,18,21]). The study of Göbel et al. [14] represents the most prominent realization of such a theoretical approach. We intend to investigate this response time issue studied in [14] in a more detailed manner. For this purpose we analyze the dissemination and commitment processes of blocks among the authenticated miner nodes in the peer-to-peer network associated with the fundamental transaction layer of a permissioned blockchain. It is assumed that the latter applies a vote-based consensus procedure which is inspired by the Practical Byzantine Fault Tolerance (PBFT) scheme [8].

We propose a Markovian non-purging (n, k) fork-join queueing model to analyze the response time of the block validation and synchronization processes among all mining nodes that apply such a vote-based procedure. We determine some relevant design and load parameters influencing the resulting commitment delay of new blocks that are appended to the blockchain view of the transaction history after a successful approval by the distributed consensus procedure. The proposed analysis is illustrated by means of a simple example of a fully interconnected P2P graph in a permissioned blockchain applying mean-value analysis techniques.

The paper is organized as follows. In Sect. 2 the fundamental properties of public-permissionless and permissioned blockchains applying a Proof-of-Work or vote-based procedure for issued transactions are briefly discussed. Section 3 presents the performance modeling and analysis of the consensus mechanism in a vote-based blockchain. In Subsects. 3.1 and 3.2 we derive a fork-join queueing network to describe the distributed decision processes associated with blocks and analyze its performance. In Subsect. 3.3 a simple example of a P2P network of miners is used to illustrate our analysis approach. In Sect. 4 the discussion is finalized by some conclusions and a brief outlook on future work.

2 Fundamental Properties of a Blockchain

A blockchain is a decentralized database which is distributed via replication among its contributing peers. Data are not directly updated, but a collection of change records represented by transactions are appended as aggregated entities called blocks (see Fig. 1). All participating peers can validate the state changes and must agree on the state of those data stored in the associated data plane. Thus, a blockchain represents the immutable history of state changes triggered by the transactions of the peers up to the most recent block. The blockchain technology has been introduced a decade ago to realize a truly decentralized digital currency called *Bitcoin* in a peer-to-peer interaction mode among its users without the need of a centralized authority (cf. [17]). In recent years the blockchain concept has been extended enormously and applied to diverse areas such as logistics, manufactoring, or the aforementioned, rapidly evolving Internet-of-Things (IoT) applications (cf. [1,2,6,7,11]).

From a technical point of view, the distributed control model of a blockchain like Bitcoin [17] or Ethereum [6] organizes the interworking of a set of identifiable, interconnected peers that belong to the underlying peer-to-peer (P2P) network of this transaction-oriented system architecture according to three basic features:

- *Decentralized architecture:* The blockchain functionality is executed by a set of independent peers that can dynamically join and leave the underlying P2P network. These peers can be maintained by different entities that do not even need to know their identities or intentions.
- *Fundamentally anonymous interaction:* The functionality of a permissionless blockchain allows peers to participate in the P2P network without identification, while a permissioned blockchain requires an authentication. Read access is not recorded and writing data to the storage plane of a blockchain is supported by a pseudonymity concept to protect the privacy of the interacting peers.
- *Stability guarantees:* Peers participating in the interactions can store data in the storage plane of this P2P network and can be assured that these data will not be manipulated, even if they do not trust other peers.

Peers of a blockchain share new cryptographically signed transactions with the whole P2P overlay. For this purpose an aggregation of a finite set of transactions into a block is applied using hashed links and a Merkle root tree technique, see Fig. 1 (cf. [6,17]). After a verification step to validate the information of a generated block each peer stores all transactions recently created by that block and then restarts to gather transactions for the next block, see Fig. 2. The goal of negotiating a common history of blocks among the peers realizes a complex *consensus* issue in the associated distributed agreement model.

The choice of an efficient consensus mechanism coincides with the access model of a peer in a blockchain. Bitcoin [17] is the prime example of an open, *unpermissioned*, i.e. *permissionless*, P2P network. Access to the data in the network or participation in the consensus process are not restricted. In the alternative *permissioned* blockchain, e.g. Hyperledger Fabric [3], the access to the

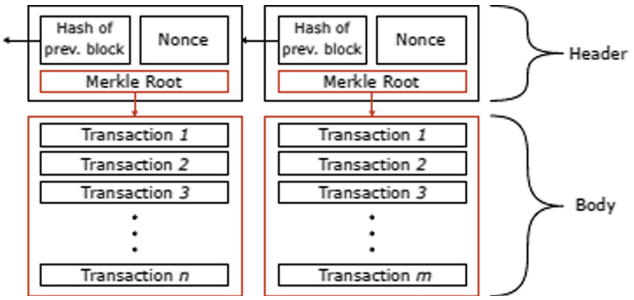

Fig. 1. A block header is indirectly dependent on the transactions by a Merkle root, and including the hash of each previous block in the current block header, a linking among the blocks is achieved.

P2P network requires authentication and authorization. The latter can be realized by well-known techniques, e.g., public-key cryptography, shared secrets, or white listed IP addresses. Regarding many IoT application scenarios, e.g. certain health-care applications, this approach is more appropriate to reduce the complexity of the transaction management in the blockchain.

The consensus mechanisms of a blockchain can be divided into *proof-based* and *vote-based* schemes (cf. [7,20]). Bitcoin [17], for instance, applies an expensive proof-based consensus mechanism called Proof-of-Work (PoW), whereas the Byzantine Fault Tolerant (BFT) replication, i.e., a solution to the *Byzantine*

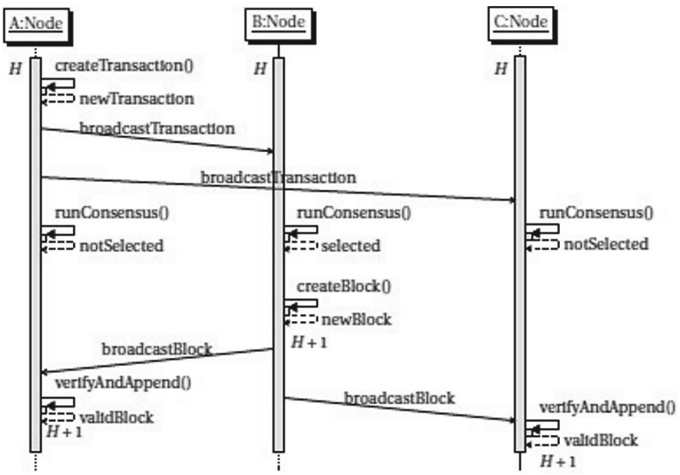

Fig. 2. Sequence diagram of an interaction between peers A, B, C where new transactions are created. A new block is created by a selected mining peer $B \equiv j \in V$ that sends it to all connected peers, which in turn verify the block and append it to their blockchain replica such that the block height H is increased, respectively.

generals problem subject to malicious nodes, constitutes a vote-based scheme (cf. [16,23]). The algorithms Practical Byzantine Fault Tolerance (PBFT) [8], BFT-SMaRt [4], and Delegated Byzantine Fault Tolerance (dBFT) are widely applied representatives of this latter class. Crash-tolerant protocols that protect the distributed system of a blockchain only against crashed but honest nodes constitute a related weaker class of consensus protocols. An extensive overview of these different types of the consensus mechanisms can be found in [20, Table 3].

Here we restrict our attention to the consensus scheme of a permissioned, vote-based blockchain in an IoT scenario. We focus on the transaction processing by those authenticated voting peers of the overlay, called mining peers, miner nodes or validators, that are selected and authorized to propose the adherence of a new block to the data plane of the blockchain, see Fig. 3.

3 Performance Analysis of the Consensus Mechanism

3.1 A Fork-Join Queueing Model of the Consensus Protocol

We consider a permissioned blockchain that applies a vote-based consensus process for the transactions emitted by the peers of the blockchain. The associated P2P network of miners comprises authenticated nodes of the blockchain that process the incoming transactions and execute the validation function to decide on a block index for the next block based on cryptographic functions. The latter include a set of validated transactions and their hashes aggregated into a Merkle tree. Finally, a validated new block is generated and appended to the blockchain, see Figs. 1 and 2. In the following we assume that the associated P2P network of authenticated miners comprises n nodes governing fog cells, see Fig. 3.

We model the associated P2P network of these n miners by an undirected, weighted graph $G = (V, E, c)$ with nodes $V = \{0, 1, \ldots, n-1\}$ and describe

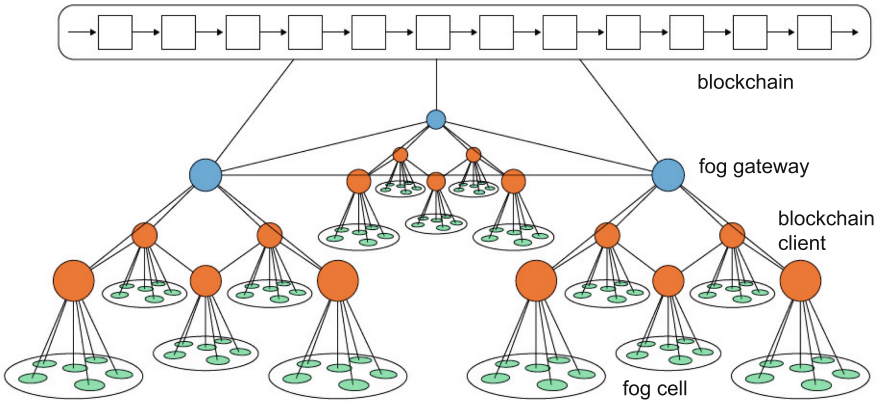

Fig. 3. P2P structure of a permissioned blockchain in a fog computing architecture with miner nodes (blue) running at fog gateways and blockchain clients (red) managing IoT devices (green) in fog cells. (Color figure online)

the edges $e = (i, j) \in E$ in terms of the corresponding symmetric adjacency matrix $A = A(G) \in \mathbb{R}^{n \times n}$ of the miner graph G. We assume that the graph is strongly connected and A is an irreducible matrix. The entry $A_{(i,j)} = c(i, j) \geq 0$, $i, j \in V$, indicates the length of the underlying direct path between two miners $i \in V, j \in V$ according to the number of the logical links at the IP-network layer (or physical links) in the P2P network.

Considering a starting node $i \in V$, e.g. $i = 0$, we construct the corresponding shortest paths $p(i, j)$ of lengths

$$d(i, j) = (A^{k(i,j)})_{(i,j)}, \quad k(i, j) = \text{argmin}_{k \geq 1}\{k \in \mathbb{N} : (A^k)_{(i,j)} > 0\}$$

to all its reachable neighbors $j \neq i, j \in V$, in the transitive closure of the reachability relation E among these nodes generated by A. We assume that $0 < \varepsilon(i) \leq d$ is the eccentricity of node $i \in V$, i.e. the greatest distance between i and any other miner $j \neq i$, and $0 < d = \max_{i \in V} \varepsilon(i)$ is the diameter of the P2P network.

Without loss of generality, we arrange the resulting associated reachability tree $\tau(i)$ of a considered miner $i = 0$ in such a way that we order all paths $p(i, j)$ from i to all nodes $j \neq i$ in ascending order of the length $d(i, j)$ starting with an initial path (i, i) of length zero as leftmost entry. It means that we construct the partition of V according to the level structure of its mining graph G subject to the selected initial miner node $i = 0$ which can be determined by an enhanced breadth-first search procedure in G. We assume that equal length paths to a certain peer j are randomly resolved and the resulting tree $\tau(i)$ has $n - 1$ paths of monotone increasing orders $0, 1, \ldots, \varepsilon(i) \leq d$ to all $j \in V \setminus \{i\}$. It means we assume a geodetic graph with a selection of a unique shortest path from an initial miner $i = 0$ to all other mining peers $j \neq i$ in the blockchain.

We describe the transactions generated by node $i = 0$ that are sent to a considered miner $j \neq i$ and aggregated by it into a cluster structure of blocks in terms of a point process $\{B_i^{(j)} : j \geq 0\}$ of an associated class $C_i = \{i\} \subset V$. We assume that the resulting block traffic is a Poisson process with rate $\lambda_i = \lambda$ with class-dependent mean interarrival time $1/\lambda_i$. The latter traffic of these blocks $B_i^{(j)}$ and their inherent transactions $\{T_i^{(k(j),l)} : 1 \leq l \leq k(j)\}$ is validated and approved by a vote-based consensus process in each miner after a broadcast of the related transactions $T_i^{(k(j),l)}$ to all corresponding miner nodes $j \in V$, see Fig. 2.

These disseminations and the associated validation processes from an initial node $i = 0$ according to the level structured mining tree $\tau(i)$ are described by a fork-join queueing network with $n = \varepsilon(i) \leq d$ branches of single-server stations $P_1, \ldots P_n$ with station-dependent service rates $\mu_k, 1 \leq k \leq n$, see Fig. 4. We assume that the latter are arising from independent, exponentially distributed service times. This model captures the aggregated transmission, propagation and transaction validation times of blocks along the tree $\tau(i)$.

We describe the processes of the cryptographically signed transfer and verification from the initial miner i to a node $j \neq i$ on the forward path and on the backward path by an infinite server queue with exponentially distributed service times $S_{(i,j)}, S_{(j,i)}$ with common mean $\mathbb{S}_{(i,j)} = d(i, j)/\nu = \mathbb{S}_{(j,i)} = d(j, i)/\nu$,

respectively. The latter are accumulated to the exponentially distributed service time for blocks ${B_i}^{(j)}$ of the transaction traffic from i with mean $\mathbb{S}_{(j,j)} = \chi_{ji}/\eta_j$ and the related waiting time for processing, i.e. the resulting sojourn time $R_{(j,j)}$ in the queue P_j of miner j. This mean $\mathbb{S}_{(j,j)}$ of the block traffic of class C_i generated by a mining peer i is arising from the local validation and block creation procedures applied by the addressed miner node j. The sharing coefficient of block traffic class C_i at node j is given by $\chi_{ji} = \chi_j = 1/n$ and reflects the uniform proportion $\chi_j = 1/n$ of the service rate η_j of node j that is assigned to class C_i among all n traffic classes. We can model the time difference between competing consensus strategies by a real scaling term $s \in (0, 1)$ and then replace η_j in terms of a scaling dependent service rate $\eta_j(s) = \widehat{\eta}_j/s$.

Using Norton's theorem (cf. [10]) or related mean-value approximations, we can approximate the resulting Erlang distribution $\widehat{R}_{(i,j)} = S_{(i,j)} + R_{(j,j)} + S_{(j,i)}$ including the transfer-verification delays $S_{(i,j)}, S_{(j,i)}$ of the forward and backward paths from i to j and the response time $R_{(j,j)}$ of the miner j by an exponential distribution with the same mean $1/\mu_j = 1/\mu$. We approximate this three station network of two infinite server queues of the combined transfer-verification delays and the single-server queue of the block stream processing at miner j by a (λ, μ)-equivalent single server with identical mean response time $\widehat{\mathbb{R}}_{(i,j)} = \mathbb{E}(\widehat{R}_{(i,j)})$, i.e.

$$\widehat{\mathbb{R}}_{(i,j)} = 2\frac{d(i,j)}{\nu} + \frac{\chi_j \cdot s/\widehat{\eta}_j}{1 - n \cdot \lambda \cdot s/\widehat{\eta}_j} = \frac{1}{\mu - \lambda}, \quad \mu = 1/\widehat{\mathbb{R}}_{(i,j)} + \lambda \tag{1}$$

subject to a scaling of the block service rate $\eta_j = \widehat{\eta}_j/s, s \in (0, 1)$.

In the following we consider a vote-based consensus process of the blocks by the involved P2P network G with $n = |V|$ miner nodes. We assume that $k = n - \lfloor (n-1)/3 \rfloor \approx 2/3 \cdot n < n$ nodes must agree before a block is approved by the miners. This approval is modelled by a non-purging fork-join process, see Fig. 5 regarding $n = 3, k = 2$ (cf. [24]).

3.2 Analysis of the Fork-Join Model of a Vote-Based Consensus Protocol

We intend to analyze the impact of broadcasting transactions within the P2P network of authenticated miners and the effect of a vote-based consensus protocol on the approval of blocks in the blockchain. To study the effect of the design and load parameters on the performance metrics of the consensus protocol, such as the scalability in terms of the number of mining nodes n and the delay spreading during the dissemination of transactions and block approval messages within the P2P network due to the distance structure, we assume here that all miners $j \in V$ have to handle the same total load of block arrivals as result of the aggregated shared information on the transactions, see Fig. 2. The latter is given in terms of a uniform arrival rate $\widehat{\lambda}_j = \lambda_S(n) = \sum_{i=1}^{n} \lambda_i = |V| \cdot \lambda$ due to broadcasting. We conclude that the block arrival rate on each node j is given by $\lambda_S(n) = n \cdot \lambda$. Moreover, we assume a common service rate $\eta_j = \widehat{\eta}$ for the block generation and validation processes of each miner node j. Then the proportion dedicated

at miner j to each service class C_i determined by the n block streams of the different miners is given by $\chi_{ji} = \chi_j = 1/n$ and yields an individual service rate $\eta_{ji} = 1/\chi_j \cdot \eta_j = \widehat{\eta}/n > 0$ per block stream $\{B_i{}^{(j)} : j \geq 1\}$ of a selected miner node i.

We further assume that the P2P mining network is designed in such a way that a homogeneous connectivity structure of the underlying graph G is established which yields a uniform level structure of all associated trees $\tau(i)$ with $n-1$ branches of equal length $\varepsilon(i) = d(i,j) = d \geq 1$ to all nodes $j \neq i$.

We follow Wang et al. [24] to analyze the delay performance of the derived non-purging (n,k) fork-join model. Therein pending jobs are not flushed when the consensus is reached by a completion of k out of n jobs which are traveling along the n different queues P_i of the model, see Figs. 4 and 5. As discussed in the previous Subsect. 3.1, this model is derived from the vote-based consensus mechanism which is applied to the block traffic of a considered miner $i = 0$ along its mining tree $\tau(i)$.

The special case of a classical (n,n) fork-join queue is symmetrical in the sense that all n queues P_i in its branches are interchangeable, see Fig. 4 (cf. [24]). Therefore, any $k < n$ arbitrarily chosen queues $\{P_{i_1}, \ldots, P_{i_k}\}$ generate the same joint probability distribution and their resulting stable sojourn times are jointly-identical random variables (cf. [24]). The basic fork-join queue and the non-purging (n,k) fork-join queue under the same job arrival process of rate λ and with the same P_i-queue's service time distribution with uniform rate μ are called (λ,μ)-equivalent queues, see Fig. 5 (cf. [24]). We use these analytic means to compute the delay metrics of the (n,k) fork-join queue of the consensus mechanism.

We compute the first passage time Σ_i until $k = \lfloor 2/3 \cdot n \rfloor \in \mathbb{N}$ miners $j \in V \backslash \{i\}$ have returned their affirmative approval result to the initiating node $i = 0$ in the fork-join queueing network with its n single-server queues P_i of common service rate μ. Due to Wang et al. [24], this time Σ_i coincides with the sojourn time $t_{(n,k)}$ of a general non-purging (n,k) fork-join queue. Applying results on a

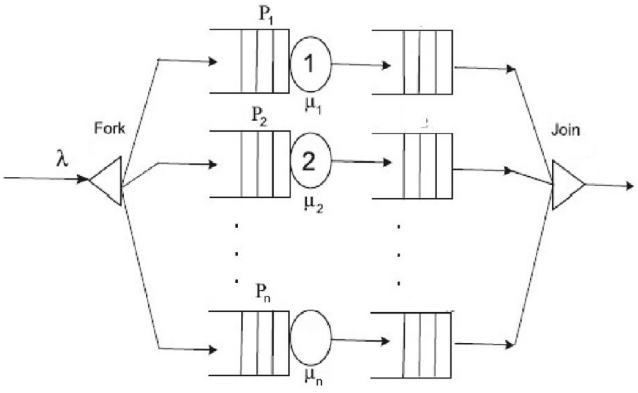

Fig. 4. Fork-join queueing model.

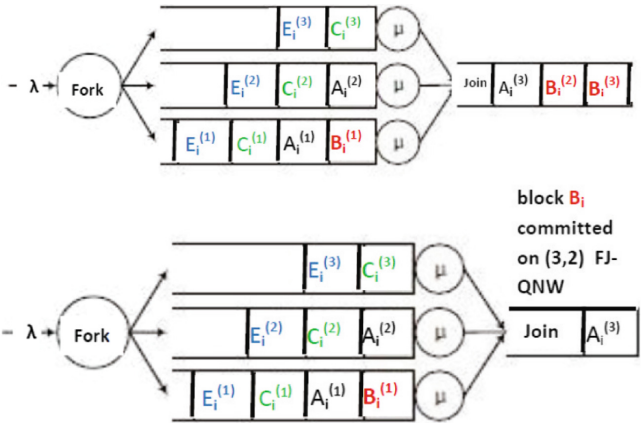

Fig. 5. Non-purging fork-join queueing model with a uniform service rate μ and an approval by $k = 2$ out of $n = 3$ nodes for block sequences $(A_i), (B_i), (C_i), (E_i)$.

linear transformation of the kth order statistics $X_{(n,k)}$ of independent, identically distributed random variables $\mathcal{X} = \{X_1, \ldots, X_n\}$ and the representation of the distribution function of the maximum $M_k = \max\{X_1, \ldots, X_k\}$ by means of coefficients $\{A_i^{(n,k)} : i = 1, \ldots, n\}$, it can be represented by a linear combination of the sojourn times $t_{(j,j)}$ of the underlying (λ, μ)-equivalent basic fork-join queues in the following way (cf. [24]):

$$t_{(n,k)} = \sum_{j=k}^{n} W_j^{(n,k)} \cdot t_{(j,j)} \quad 1 \le k \le n \tag{2}$$

$$W_i^{(n,k)} = \sum_{j=k}^{i} \binom{n}{j} \cdot A_i^{(n,j)} \quad 1 \le k \le n, \quad 1 \le i \le n \tag{3}$$

$$A_i^{(n,k)} = \begin{cases} 1, & i = k \\ -\sum_{j=1}^{i-k} \binom{n-i+j}{j} \cdot A_{i-j}^{(n,k)} & k+1 \le i \le n \end{cases} \tag{4}$$

The recursion (4) of the coefficients $A_i^{(n,k)}, 1 \le k < n$, can be simplified in the following manner:

$$A_{k+l}^{(n,k)} = \begin{cases} 1, & l = 0 \\ -\sum_{j=0}^{l-1} \binom{n-k-j}{n-k-l} \cdot A_{k+j}^{(n,k)} & 1 \le l \le n-k \end{cases} \tag{5}$$

Using the $W^{(n,\cdot)}$- and $A^{(n,\cdot)}$-coefficients in (3), (5), we can define upper-triangular matrices W, A, B in the following way:

$$W = \left((W^{(n,\cdot)})_{k,i}\right) = \left(W_i^{(n,k)}\right) = \begin{pmatrix} W_1^{(n,1)} & W_2^{(n,1)} & W_3^{(n,1)} & \dots & \dots & W_n^{(n,1)} \\ 0 & W_2^{(n,2)} & W_3^{(n,2)} & \dots & \dots & W_n^{(n,2)} \\ 0 & 0 & W_3^{(n,3)} & \dots & \dots & W_n^{(n,3)} \\ \vdots & \dots & \ddots & \ddots & \dots & \vdots \\ \vdots & \dots & \dots & 0 & W_{n-1}^{(n,n-1)} & W_n^{(n,n-1)} \\ 0 & \dots & \dots & \dots & 0 & W_n^{(n,n)} \end{pmatrix} \tag{6}$$

$$A = \left((A^{(n,\cdot)})_{k,i}\right) = \left(A_i^{(n,k)}\right) = \begin{pmatrix} A_1^{(n,1)} & A_2^{(n,1)} & A_3^{(n,1)} & \dots & \dots & A_n^{(n,1)} \\ 0 & A_2^{(n,2)} & A_3^{(n,2)} & \dots & \dots & A_n^{(n,2)} \\ 0 & 0 & A_3^{(n,3)} & \dots & \dots & A_n^{(n,3)} \\ \vdots & \dots & \ddots & \ddots & \dots & \vdots \\ \vdots & \dots & \dots & 0 & A_{n-1}^{(n,n-1)} & A_n^{(n,n-1)} \\ 0 & \dots & \dots & \dots & 0 & A_n^{(n,n)} \end{pmatrix} \tag{7}$$

$$B = \left((B^{(n,\cdot)})_{k,i}\right) = ((B_i)_{i\geq k}) = \begin{pmatrix} B_1 & B_2 & B_3 & \dots & \dots & B_n \\ 0 & B_2 & B_3 & \dots & \dots & B_n \\ 0 & 0 & B_3 & \dots & \dots & B_n \\ \vdots & \dots & \ddots & \ddots & \dots & \vdots \\ \vdots & \dots & \dots & 0 & B_{n-1} & B_n \\ 0 & \dots & \dots & \dots & 0 & B_n \end{pmatrix} \tag{8}$$

$$B_i = \binom{n}{i} \tag{9}$$

Then the $W^{(n,\cdot)}$-coefficients can be computed in terms of the B_i- and $A^{(n,\cdot)}$-coefficients by a simple matrix multiplication of two upper-triangular matrices:

$$W = B \cdot A \tag{10}$$

The expected sojourn time $\mathbb{T}_{(n,k)} = \mathbb{E}(t_{(n,k)})$ of a general non-purging (n,k) fork-join queue can be represented by a linear combination of the expected sojourn times of the (λ,μ)-equivalent basic fork-join queues $P_i, 1 \leq i \leq n$, in the following way (cf. [24, Theorem 4]):

$$\mathbb{T}_{(n,k)} = \sum_{i=k}^{n} W_i^{(n,k)} \mathbb{T}_i \tag{11}$$

Here $\mathbb{T}_i = \mathbb{E}(t_{(i,i)})$ is the expected sojourn time of the (λ, μ)-equivalent basic (i, i) fork-join queue and $W_i^{(n,k)}$ are the corresponding W-coefficients given by (3).

In the special case of an exponentially distributed service time distribution with a uniform service rate μ in each branch of the fork-join network, we can apply Nelson's approximation $\mathbb{T}^*_{(n,k)}$ of the expected mean sojourn time $\mathbb{E}(t_{(n,k)})$ (cf. [24, Theorem 5], [19]).

$$\mathbb{T}^*_{(n,k)} = \max\{\widehat{\mathbb{T}}_{(n,k)}, 0\} \tag{12}$$

$$W_S{}^{(n,k)} = \sum_{i=\max\{2,k\}}^{n} W_i^{(n,k)} \left[\frac{11H_i + 4\rho(H_2 - H_i)}{H_2}\right] k \in \{1, \ldots, n\} \tag{13}$$

$$\widehat{\mathbb{T}}_{(n,k)} = \begin{cases} n \cdot \frac{1/\mu}{1-\rho} + \frac{12-\rho}{88} \cdot \frac{1/\mu}{1-\rho} \cdot W_S{}^{(n,1)} & k = 1 \\ \frac{12-\rho}{88} \cdot \frac{1/\mu}{1-\rho} \cdot W_S{}^{(n,k)} & k \geq 2 \end{cases} \tag{14}$$

to the (n, k) fork-join queue. Here $H_i = \sum_{k=1}^{i} \frac{1}{k}$ are the harmonic numbers, λ is the single arrival rate of all queues P_i and $\rho = \lambda/\mu$ is the corresponding load. This approximation is based on the mean response time $\mathbb{E}(R_{M/M/1}) = \frac{1/\mu}{1-\rho} = (\mu - \lambda)^{-1}$ of an M/M/1 queue.

3.3 Performance Evaluation of an Illustrative P2P Network

To investigate the scalability and commitment delay of the vote-based consensus protocol applied in a permissioned blockchain, we consider a fully interconnected P2P network of n miner nodes. They are integrated into a fog computing architecture as fog gateways controlling corresponding fog cells and their blockchain clients (cf. Fig. 3). To illustrate the mean-value analysis approach, we consider here only a small, simple configuration scenario of two interconnected routers R_1, R_2 that couple several fog cells arranged in two clusters with a population of n miners where $\lfloor n/2 \rfloor$ peers in a fog cell cluster are attached to each router R_i. It is assumed that there is a negligible link delay between the latter routers in the P2P network compared to the access delay by the peers.

Given a simple example with $n = 6$ peers, three attached nodes Φ_1, Φ_2, Φ_3 and Φ_4, Φ_5, Φ_6, respectively, interact as mining peers of this blockchain in each cell cluster. The related miner graph $G = (V, E)$, $V = \{0, \ldots, 5\}$, has an associated irreducible, block-structured adjacency matrix:

$$A(G) = \begin{pmatrix} 0&2&2&2&2&2 \\ 2&0&2&2&2&2 \\ 2&2&0&2&2&2 \\ 2&2&2&0&2&2 \\ 2&2&2&2&0&2 \\ 2&2&2&2&2&0 \end{pmatrix} = \begin{pmatrix} A_1 & 2E_3 \\ 2E_3 & A_1 \end{pmatrix}, \; A_1 = \begin{pmatrix} 0&2&2 \\ 2&0&2 \\ 2&2&0 \end{pmatrix}, \; E_3 = \begin{pmatrix} 1&1&1 \\ 1&1&1 \\ 1&1&1 \end{pmatrix}$$

We consider the representative traffic of blocks generated by the peer $\Phi_0 \equiv 0$ to all other peers $\{\Phi_1, \ldots, \Phi_5\} \equiv \{1, \ldots, 5\} = V \setminus \{0\}$. We assume that the rate of

this generated block traffic is given uniformly by $\lambda_i = \lambda > 0$ for each peer $i \in V$. Due to the broadcasting of the blocks of class i from the considered peer $i = 0$ to all others $j \neq i$, a representative miner, e.g. $j = 2$, will get additionally block traffic with a rate $\lambda_S(n-1) = (n-1)\lambda$, i.e. in total it has to process $\lambda_S(n) = n\lambda$ blocks per time unit t_u.

We assume a block service rate $\mu_j = \mu$ at each node $j \in V$ related to $\widehat{\mathbb{R}}_{(i,j)} = 2d(i,j)/\nu + \frac{\chi_j/\eta_j}{1-\lambda_S(n)/\eta_j}, \chi_j = 1/n$, which captures the mean link delay $\mathbb{S}_{(i,j)} = \mathbb{E}(S_{(i,j)}) = d(i,j)/\nu$ and link distance $d(i,j) = d(0,2) = 2$ between the sender $i = 0$ of the transaction set and the receiving miner $j = 2$ processing these items in terms of related blocks (see Fig. 2).

Inspired by previous measurements (cf. [13]), we assume a block traffic rate $\lambda < 1/500$ per time unit $t_u = 1\,\mathrm{s}$ and a mean block service time of $\mathbb{E}(S_{(j,j)}) = \chi_j/\eta_j(s) = \chi_j \cdot s/\widehat{\eta}_j = s/n \cdot 20\ t_u$'s. It includes a scaling term $s \in (0,1) \subset \mathbb{R}$ that can model, for instance, the time difference between competing consensus strategies. The mean delay on a link including transmission, propagation, and transaction validation times is supposed to be $\mathbb{S}_{(i,j)} = d(i,j)/\nu = 500\,\mathrm{s}$.

We consider an example with $n \in \{6, 12, 18, 24, 27\}$ peers in the miner network and the initial peer $i = 0$ where $n/2$ miners are attached to each router R_1, R_2 with a negligible transfer delay between them. Then a uniform arrival rate $\lambda_0 = 1/634$ blocks/sec, the mean block service time $1/\eta_j(s) = 20\,\mathrm{s}$ for $s = 1$, as well as the link delay value $1/\nu = 250\,\mathrm{s}$ between the miners and the unidirectional distance $d(0,j) = 2$ are applied as uniform load model to all peers $j \neq i$ in (1).

We can calculate the corresponding mean block synchronization delay $\mathbb{T}_{(n,k)}$ of the consensus mechanism by Taylor's approximation $\mathbb{T}^*_{(n,k)}$ in (12) to (14) for a required feedback by $k = 2/3{\cdot}n \in \{4, 8, 12, 16, 18\}$ peers. The outcome is depicted in Fig. 6 where a parametrization in terms of the scaling factor $s \in (0,1)$ of the block processing time and a variable load $\rho = \lambda/\mu$ in the case $n = 12, k = 8, s = 1$ is used. It reveals the robustness w.r.t. scaling and a linear behavior of the mean consenus delay, as expected, due to the low utilization levels of the (λ, μ)-equivalent queues in the fork-join network of the considered examples.

An indispensable validation of the sketched performance model in a fully virtualized IoT scenario requires an implementation of a vote-based consensus protocol in a permissioned blockchain such as the Multichain [27] framework. Such a blockchain system that is integrated into a virtualized fog computing environment is currently under development in a LINUX test bed. It is based on a cluster of Raspberry Pi's with their 64-bit ARM processor architecture and Hypriot Cluster Lab (HCL) realizing a master-slave clustering by Docker Swarm as software environment (cf. [9,15,26]). However, only a few preliminary performance results regarding the basic Docker Swarm operation mode of the blockchain clients that are calling a developed Multichain Docker image running on a master node are available so far (cf. [9]). An extended measurement study regarding the realized commitment delay of the consensus protocol and its comparison with the developed performance model is a subject of our future research.

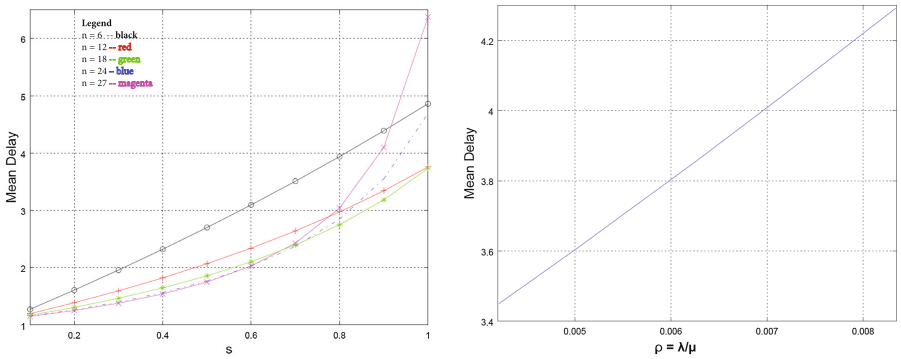

Fig. 6. Approximation $\mathbb{T}^*_{(n,k)}$ of the mean consensus delay by a (n,k)-fork-join model for $n \in \{6, 12, 18, 24, 27\}$ parametrized by the scaling factor $s \in (0,1)$ (left-hand side) and for $n = 12, k = 8, s = 1$, parametrized by the load $\rho = \lambda/\mu$ (right-hand side).

4 Conclusions

A blockchain is a decentralized database which is distributed via replication among its contributing clients. It is governed by a dissemination of signed transaction sets and a consensus-based validation of the derived aggregated objects called blocks by the mining peers. In recent years public-permissionless blockchains such as Bitcoin [17] or Ethereum [6] and the basic functionality of the involved Proof-of-Work based consensus protocol have been intensively studied (cf. [20]). Regarding the application of blockchain technology in advanced Internet-of-Things scenarios, permissioned blockchains such as Hyperledger Fabric [3] or Multichain [27] that use different variants of a lightweight vote-based consensus protocol constitute a more adequate alternative to public-permissionless blockchains.

Considering such a permissioned blockchain with a vote-based consensus algorithm embedded in a fog computing environment, we have analyzed the dissemination and commitment processes of blocks among the corresponding mining nodes in the underlying peer-to-peer network. Inspired by Wang et al. [24], we have proposed a Markovian, non-purging (n,k) fork-join queueing model to analyze the response time performance of the block synchronization process among these mining nodes that apply a vote-oriented consensus procedure. We have determined the influencing parameters of the commitment delay of new blocks that are appended to the blockchain after a successful approval by the distributed consensus procedure.

It has been a major goal of our performance study to investigate the scalability issues of the consensus strategy and its inherent limitations by means of the derived analytic fork-join queueing model and to gain insights on the impact of all basic load parameters of the model. The proposed mean-value analysis of the fundamental delay performance metric has been illustrated by means of a simple example of a fully interconnected P2P graph arising from the interconnected clients in several fog cells.

Regarding the accuracy of the proposed approach, an indispensable validation of our performance model by extended simulations and adequate measurements in an IoT setting that is fully virtualized in terms of Docker [26] containers and operating in Docker Swarm mode such as our Raspberry Pi test bed HCL-BaFog based on Hypriot Cluster Lab (HCL) (cf. [9,15,25]) will constitute important items of our future research.

References

1. Al-Fuqaha, A., et al.: Internet of Things: a survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutorials **17**(4), 2347–2376 (2015). Fourth Quarter
2. Atzori, L., et al.: Internet of Things: a survey. Comput. Netw. **54**, 2787–2805 (2010)
3. Androulaki, E., et al.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Oliveira, R., Felber, P., Hu, Y.C. (eds.) Proceedings of the Thirteenth EuroSys Conference, EuroSys 2018, Porto, Portugal, 23–26 April 2018, pp. 30:1–30:15. ACM (2018)
4. Bessani, A., Sousa, J., Alchieri, E.: State machine replication for the masses with BFT-SMART. In: 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 355–362. IEEE (2014)
5. Brody, P., Pureswaran, V.: Device democracy: saving the future of the Internet of Things. IBM, September 2014
6. Buterin, V.: Ethereum White Paper: A Next-Generation Smart Contract and Decentralized Application Platform (2013). https://www.blockchainresearchnetwork.org/research/whitepapers/. Accessed 22 Nov 2017
7. Cachin, C., Vukolić, M.: Blockchains consensus protocols in the wild. arXiv preprint arXiv:1707.01873 (2017)
8. Castro, M., Liskov, B.: Practical Byzantine fault tolerance. In: Seltzer, M.I., Leach, P.J. (eds.) Proceedings of the Third USENIX Symposium on Operating Systems Design and Implementation (OSDI), New Orleans, Louisiana, USA, 22–25 February 1999, pp. 173–186. USENIX Association (1999)
9. Cech, H.L., Großmann, M., Krieger, U. R.: A fog computing architecture to share sensor data by means of blockchain functionality. In: 2019 IEEE International Conference on Fog Computing (ICFC 2019) (2019, accepted paper)
10. Chandy, K.M., Herzog, U., Woo, L.: Parametric analysis of queuing networks. IBM J. Res. Dev. **19**(1), 36–42 (1975)
11. Christidis, K., Devetsikiotis, M.: Blockchains and smart contracts for the Internet of Things. IEEE Access **4**, 2292–2303 (2016)
12. Conoscenti, M., Vetrò, A., De Martin, J.C.: Blockchain for the Internet of Things: a systematic literature review. In: IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA) (2016)
13. Decker, C., Wattenhofer, R.: Information propagation in the Bitcoin network. In: 13th IEEE Conference on Peer-to-Peer Computing, pp. 1–10 (2013)
14. Göbel, J., Keeler, H.P., Krzesinski, A.E., Taylor, P.G.: Bitcoin blockchain dynamics: the selfish-mine strategy in the presence of propagation delay. Perform. Eval. **104**, 23–41 (2016)
15. Großmann, M., Eiermann, A., Renner, M.: Hypriot Cluster Lab: An ARM-powered cloud solution utilizing Docker. In: 23rd International Conference on Telecommunications (ICT 2016), Thessaloniki, Greece, 16–18 May 2016 (2016)

16. Lamport, L., Shostak, R., Pease, M.: The Byzantine generals problem. ACM Trans. Program. Lang. Syst. **4**(3), 382–401 (1982)
17. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2008). https://bitcoin.org/bitcoin.pdf. Accessed 30 Nov 2018
18. Natoli, C., Gramoli, V.: The Balance Attack Against Proof-Of-Work Blockchains: The R3 Testbed as an Example, 30 December 2016. arXiv:1612.09426v1
19. Nelson, R.D., Tantawi, A.N.: Approximate analysis of fork/join synchronization in parallel queues. IEEE Trans. Comput. **37**(6), 739–743 (1988)
20. Nguyen, G.-T., Kim, K.: A survey about consensus algorithms used in blockchain. J. Inf. Process. Syst. **14**(1), 101–128 (2018)
21. Ron, D., Shamir, A.: Quantitative analysis of the full bitcoin transaction graph. In: Sadeghi, A.-R. (ed.) FC 2013. LNCS, vol. 7859, pp. 6–24. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39884-1_2
22. Shafagh, H., Hithnawi, A., Burkhalter, L., Duquennoy, S.: Towards blockchain-based auditable storage and sharing of IoT Data. arXiv Preprint arXiv:1705.08230 (2017)
23. Vukolić, M.: The quest for scalable blockchain fabric: proof-of-work vs. BFT replication. In: Camenisch, J., Kesdoğan, D. (eds.) iNetSec 2015. LNCS, vol. 9591, pp. 112–125. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39028-4_9
24. Wang, H., et al.: Approximations and bounds for (n, k) fork-join queues: a linear transformation approach. In: 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID) (2018)
25. Ziegler, M.H., Großmann, M., Krieger, U.R.: Integration of Fog Computing and Blockchain Technology Using the Plasma Framework. Technical report, University of Bamberg (2019, submitted)
26. Docker Inc.: Docker Overview (2018). https://docs.docker.com/engine/docker-overview/. Accessed 28 Aug 2018
27. MultiChain: Multichain 1.0 Beta 2 and 2.0 Roadmap (2017). https://www.multichain.com/blog/2017/06/multichain-1-beta-2-roadmap/. Accessed 2 Sept 2018