Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Analytical model for performability evaluation of Practical Byzantine Fault-Tolerant systems

Marco Marcozzi [a,b,*], Leonardo Mostarda [a]

[a] Computer Science Division, University of Camerino, via Madonna delle Carceri 7, I-62032 Camerino (MC), Italy
[b] Institute of Data Science and Digital Technologies, Vilnius University, Akademijos str. 4, Vilnius, LT-08663, Lithuania

ARTICLE INFO

Keywords:
Byzantine consensus protocol
Fault-Tolerant systems
Performability model
Stochastic processes
Blockchain

ABSTRACT

Designing systems tolerant to faults is crucial to assure continuity of service for mission critical applications. However, their implementation may be costly and challenging. In this study, analytical models are presented for performance evaluation of systems equipped with Practical Byzantine Fault-Tolerant consensus protocols. Byzantine Fault Tolerance is particularly compelling, since it can provide a robust consensus mechanism to implement decentralized platforms, like Decentralized Ledger Technology and, notably, blockchains. The performability model is based on continuous-time Markov chains, in which the processes involved follow the exponential distribution. The numerical results presented report an inverse non-linear relation between number of nodes and performability. Performance decreases also as the ratio between break-down rate and repair rate increases.

1. Introduction

Distributed systems are structures in which their components, e.g. computers, smart devices, etc., are spatially separated, but interconnected by a network. Components in a distributed system interact with each other to achieve a common objective through the exchange of messages. For instance, distributed systems is an eminent topic in the field of computing and computer networks, especially in relation with its valuable and ubiquitous applications. Decentralized Ledger Technology (DLT) is a paradigmatic example of distributed system, in which servers are replicating a distributed ledger of transactions, agreed upon by passing messages on a network. Blockchain is indubitably the most known example of DLT, because of the massive attention given to Bitcoin (Nakamoto, 2008) and other platforms, in the context of cryptocurrencies (Belotti, Bozic, Pujolle, & Secci, 2019; Berdik, Otoum, Schmidt, Porter, & Jararweh, 2021; Kolb, AbdelBaky, Katz, & Culler, 2020; Paulavičius, Grigaitis, Igumenov, & Filatovas, 2019). However, finance is not the only field of application for DLTs and blockchain. Indeed, DLT solutions have been successfully applied in: distributed computing and smart contracts (Zheng et al., 2020), healthcare (Namasudra & Deka, 2021), identity verification (Dunphy & Petitcolas, 2018), Industry 4.0 and Internet of Things (IoT) (Bodkhe et al., 2020; Lao et al., 2020), supply chain management (Pournader, Shi, Seuring, & Koh, 2020), energy sector (Mollah et al., 2020), etc.

At the core of any DLT solution there is the consensus protocol, i.e. an algorithm/process in which the rules to agree on messages to be included into the ledger are defined. Notable examples include the consensus protocol family powering Bitcoin, called Proof of Work (PoW), and the family referred as Proof of Stake (PoS), which is now the backbone of the platform Ethereum (Wood, 2014). In this work, however, the focus is on a class of protocols that has contributed effectively to the development of distributed computing: Byzantine Fault-Tolerant (BFT) algorithms. Starting from the seminal work *The Byzantine Generals Problem* (Lamport, Shostak, & Pease, 1982), various important algorithms for distributed computing have emerged, including the Practical Byzantine Fault-Tolerant (PBFT) algorithm (Castro et al., 1999) and its modifications. The main advantage of BFT protocols is their resilience to arbitrary, malicious behaviors from a restricted (tolerated) amount of participants, also known as Byzantine nodes – in relation with the original analogy of the Byzantine generals problem. Differently from crash or stop-fail tolerant systems, BFT can indeed tolerate Byzantine nodes by requiring each participant to share with all the others the messages to be committed, and commit them only if a certain quorum is reached.

By any means, for what concerns blockchain technologies, in particular, there is an increasing interest in assessing performance evaluation for platforms and consensus protocols, both centered on empirical analysis and analytical modeling. Performance evaluation for consensus protocols gives access to a deeper understanding of the consensus process itself, through the formulation of the model. Additionally,

* Corresponding author at: Computer Science Division, University of Camerino, via Madonna delle Carceri 7, I-62032 Camerino (MC), Italy.
E-mail addresses: marco.marcozzi@unicam.it (M. Marcozzi), leonardo.mostarda@unicam.it (L. Mostarda).

https://doi.org/10.1016/j.eswa.2023.121838
Received 18 May 2023; Received in revised form 11 September 2023; Accepted 25 September 2023
Available online 28 September 2023

by applying the model, the parameters of a new system to be developed can be estimated, without actually implementing the system itself. Indeed, it is quite challenging and relatively expensive to develop a blockchain system from scratch, especially for testing purposes only (Rasolroveicy, Haouari, & Fokaefs, 2021). This challenge also compels companies and, in general, decision makers to evaluate the actual need for a blockchain architecture, since cloud-based services are actually cheaper from a business point of view (Rimba et al., 2017).

Analytical modeling is a widely used tool for evaluating the performability of complex systems, processes, and networks. These mathematical representations of real-world systems allow for the prediction of performability under various conditions, enabling the identification of bottlenecks and the development of strategies to improve efficiency. The field of analytical modeling encompasses a variety of techniques, including queuing models (Trivedi, 2008), simulation models (Law, Kelton, & Kelton, 2007), and optimization models (Rao, 2019), each with their own strengths and limitations.

Queuing models (Trivedi, 2008), for example, are particularly useful in the analysis of systems that involve waiting in line, such as call centers and computer networks. Simulation models (Law et al., 2007), on the other hand, are well-suited to the study of complex systems by creating virtual representations of the system and running experiments on these models. Optimization models (Rao, 2019), meanwhile, are particularly useful in identifying the best possible solution to a problem by finding the optimal combination of inputs.

The theoretical foundations of this work is based on continuous-time Markov chains (CTMC), therefore the rest of article focuses principally on the characterization and study of papers using queuing theory for analytical evaluation of performability metrics. In queuing theory, historically (Erlang, 1909), the main assumption was that arrival and service mechanisms are described as Poisson or exponential processes, so that the memoryless property of Markov chains is guaranteed. Although, there are other stochastic processes, characterized by different probability distributions, that may be used to study a queue (Bolch, Greiner, De Meer, & Trivedi, 2006). As shown in the following, Kendall's notation (Kendall, 1953) is a simple, widely used notation, useful to describe elementary queuing systems:

$$A/S/c/K/D, \tag{1}$$

where A is the distribution underlying the arrival process, S the distribution pertaining the service process, c the number of processors/nodes/servers, K the upper-limit for the queue length (e.g. buffer size, allowed customers in line, etc.), and D the serving discipline.

Kendall's notation is used in Section 2 to report concisely the assumptions and framework in which the proposed models were elaborated. In the following, briefly, there are the symbols for different probability distributions cited in the literature reported: M stands for "Markovian" (which is a Poisson process with exponential inter-arrival/service time), PH is a convolution of exponential distributions, while G (or GI) indicates that any general stochastic distribution is used, leading, still, to some analytical results. In some cases, the symbol describing the probability distribution is accompanied by an exponent, indicating that the involved process is a batch mechanism, e.g. $M/M^B/1$ describes a Markov arrival process with a batch Markov serving process, where B is the batch size.

In summary, the contributions of this research includes:

- An analytical approach to evaluate PBFT performability metrics, i.e. availability, blocking probability, throughput, mean queue length, and transaction latency.
- An iterative algorithm to calculate these metrics for different parameters, such as the number of nodes in the system or stochastic process rates, i.e. arrival, serving, breakdown, and repair processes.
- Results obtained applying the proposed methodology, both in an artificial setting, both replicating data from literature.

The rest of this article is structures as in the following: Section 2 presents a comprehensive review of prior researches on analytical models for performability evaluation. Section 3 introduces the research methodology employed in this work. Section 4 describes the performability model proposed and its underlying assumptions. The procedure and formulas used to obtain solutions are investigated in Section 5. Section 6 shows the results obtained in this study. Lastly, Section 7 summarizes the findings, discusses potential applications, and outlines future advancements for the presented model.

2. Related work

In recent years, there has been a growing interest in evaluating the performance of DLT and consensus protocols, both through empirical analysis and analytical modeling. The motivation behind this interest is twofold: to gain a deeper understanding of the process through modeling, and to estimate the parameters of new systems before actually developing them, saving time and resources (Rasolroveicy et al., 2021; Rimba et al., 2017).

To organize the existing knowledge, various efforts have been made to categorize methods and techniques for performance evaluation of, prevalently, blockchain systems (Fan, Ghaemi, Khazaei, & Musilek, 2020; Smetanin, Ometov, Komarov, Masek, & Koucheryavy, 2020). In particular, some studies have reviewed the use of analytical methods and simulations to study the performance of consensus protocols in blockchains (Ma, Fan, Zhang, & Liu, 2020; Smetanin, Ometov, Kannengieser, et al., 2020). There are different techniques that can be used in the task of analytic performance evaluation, including Stochastic Reward Net (SRN) (Sukhwani, Martínez, Chang, Trivedi, & Rindos, 2017), game theory (Qi, Yu, & Jin, 2020), and hierarchical model approach (Jiang, Chang, Liu, Mišić, & Mišić, 2020). In this study, the focus is on articles that apply Queuing Theory to the performance evaluation of DLT systems (Balsamo, Malakhov, Marin, & Mitrani, 2022; Balsamo, Marin, Mitrani, & Rebagliati, 2021; Fralix, 2020; Geissler, Prantl, Lange, Wamser, & Hossfeld, 2019; Huang, Ma, & Zhang, 2019; Kawase & Kasahara, 2017; Li, Ma, & Chang, 2018; Li, Ma, Chang, Ma, & Yu, 2019; Ma & Fan, 2022; Meng, Zhao, Wolter, & Xu, 2021; Ricci et al., 2019; Wilhelmi & Giupponi, 2021).

Of all the different DLT platforms available, Bitcoin (Nakamoto, 2008) has received the most attention, and the goal of many related works is to develop a queuing model of a general Proof of Work (PoW) consensus protocol, using Bitcoin as an example (Balsamo et al., 2022, 2021; Fralix, 2020; Frolkova & Mandjes, 2019; Geissler et al., 2019; Kawase & Kasahara, 2017; Li et al., 2018, 2019; Ricci et al., 2019; Wilhelmi & Giupponi, 2021). These articles focus their attention on different aspects of bitcoin-like blockchains, e.g. time delay and mining process, thus the development of specific models to describe those processes. Specifically, in Ricci et al. (2019), it is proposed a $M/G/1$ queueing theory model to characterize the delay experienced by Bitcoin transactions. Authors of Fralix (2020) and Frolkova and Mandjes (2019) make use of $G/M/\infty$ and $M/G/\infty$ queues to study the synchronization in Bitcoin network. Li et al. (2018, 2019) employ a $GI/M/1$ queue that can provide analysis both for the stationary performance measures and for the sojourn time of any transaction or block. By means of a batch Markov serving process $M/M^B/1$, in Balsamo et al. (2021) the consolidation time of transactions in Bitcoin network is studied, with regard to the relation between the fee offered by a transaction and its expected consolidation time. A similar endeavour is the focus in Kawase and Kasahara (2017), where $M/G^B/1$ queues are applied to study the transaction confirmation time for Bitcoin. In Wilhelmi and Giupponi (2021) it is presented a blockchain model based on a wireless infrastructure, where to determine its performance metrics a discrete-time $M/M^B/1/K$ queue is used. The work in Geissler et al. (2019) aims to investigate key performance indicators and general limits of blockchains with the aid of a discrete-time $GI/GI^B/1$ model.

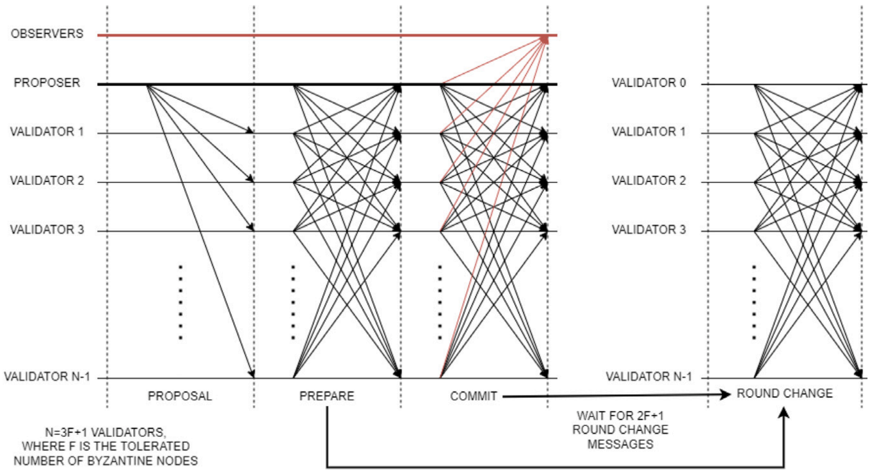


Fig. 1. A schematic representation of the PBFT consensus protocol's workflow.

These studies have opened the way to the study of other consensus protocols, such as Raft and Raft-based protocols for private (Huang et al., 2019) or consortium blockchains (Meng et al., 2021). For instance, Huang et al. (2019) model Raft using a simple, yet effective, $M/M/1$ queue. The proposed model can predict the network split time and the probability this may happen. Hyperledger Fabric has been modeled in Meng et al. (2021), where a $PH/PH/1$ queue is used to analyze the consistency properties of consortium blockchain protocols.

This article, however, focus on PBFT protocols and related variations applied to DLT and blockchain. These algorithms have been studied in the context of performance evaluation of blockchains using analytical models, as Nischwitz, Esche, and Tschorsch (2021) employing a formalism based on Bernoulli processes to model dynamic failures in PBFT systems. For what concerns, in particular, the use of queuing theory to the study of complex DLT networks, Ma and Fan (2022) proposed a $M/PH/1$ model to evaluate the performance of the Improved PBFT protocol and Chang, Li, Wang, and Song (2022) presented a $M \oplus M^b/M^b/1$ to describe dynamic PBFT systems. The works here mentioned are both applying the matrix-geometric solution to analyze the PBFT blockchain system. This is a standard approach to solve the problem of the increasing complexity (and dimensionality) of the problem, exploiting the repeating structure of the matrix representing the balance equations.

For what concerns our contribution, instead, the analytical model proposed in this article is based on CTMC to assess the performability of protocols related to PBFT (explained graphically in Fig. 1), including the Improved PBFT protocol and other variations. The main contribution of this work is the definition of a multi-dimensional state diagram, able to give performability metrics as exact solutions of the balance equations describing the system model.

3. Methodology

This methodology section outlines our approach for developing and proposing a novel analytical model for assessing the performability of a distributed system employing a PBFT consensus protocol.

Our research design follows a predominantly theoretical and analytical approach. It adheres to the established field of queueing theory, with assumptions and methods corroborated by the literature provided in Section 2. Indeed, uur analytical model is grounded in queueing theory principles, which provide a theoretical framework for understanding the queuing dynamics within distributed systems. We adapt and extend these principles to address the unique characteristics of systems employing PBFT consensus protocols. For what concerns the applicability of these methods, those can be applied to systems that can be viewed acting as a queue. For instance, there are several application fields for our proposed model, but prominently it can be applied to computer networks.

The proposed analytical model incorporates the following key components:

- Network Configuration: We define a simplified network topology based on common DLT architecture paradigms, i.e. the topology must allow communications to reach each member of the network without single-point of failure.
- Arrival Rate Modeling: We use a Poisson process to model the transaction arrival rates based on network activity patterns.
- Service Time Modeling: We use an exponentially distributed service time to model the transaction service rates.
- Availability Metrics: We derive analytical expressions for system uptime, fault tolerance, and resilience based on the theoretical model.

In the absence of empirical data, we rely on educated assumptions for model parameters. These assumptions are based on the available literature and expert consensus within the field of queueing theory.

We acknowledge that the limited availability of empirical data severely limits our ability to perform traditional model validation. Therefore, the validation of our model is primarily theoretical in nature. We assess the model's internal consistency and logical coherence to ensure it aligns with established queueing theory principles and PBFT system behaviors described in the literature.

The analysis and results deriving from the model are conducted using custom Python scripts.

Our methodology's primary limitation is the limited amount of empirical data for validation. This limitation restricts our ability to provide empirical evidence supporting the accuracy and applicability of the model to real-world systems. Additionally, our model relies on simplifications and assumptions that may not capture all complexities of distributed systems using PBFT consensus protocols.

In summary, our methodology for proposing this analytical performability model for PBFT-based systems is primarily theoretical due to the limited amount of empirical data. While the model is grounded in queueing theory principles and expert insights, we acknowledge the inherent limitations associated with the lack of a consistent empirical validation.

4. System model

In the following, a model to assess performability of systems based on PBFT protocols is presented. This analytical model is based on CTMC, and it aims to compute the performance of PBFT systems using queueing theory.

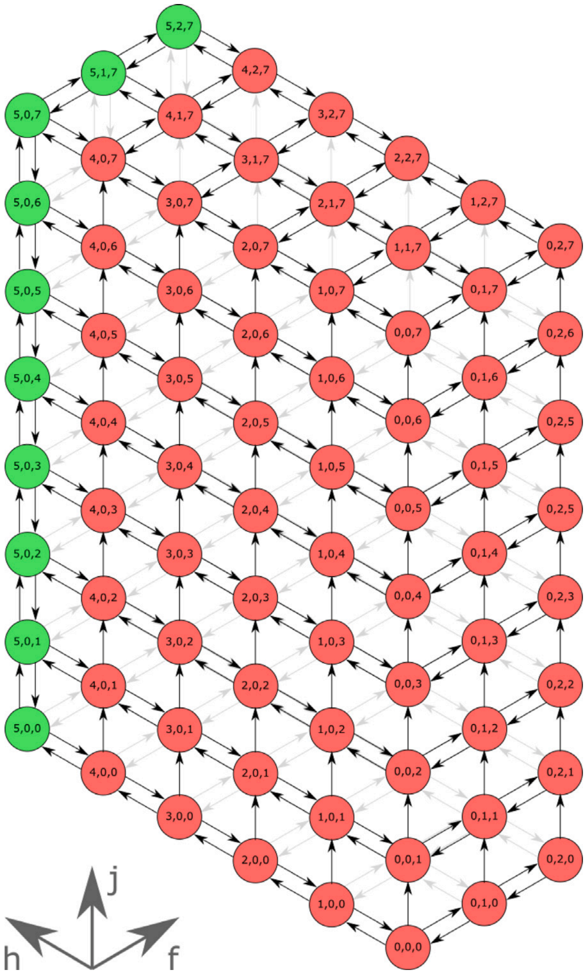


Fig. 2. Depiction of a PBFT state diagram with $H = 5, F = 2$ (hence, $N = 7$), and $J = 7$. The states in the diagram have indices (h, f, j), where $h \in [0, H]$, $f \in [0, F]$, and $j \in [0, J]$.

A system with $N \geq 4$ servers is considered – since, with unsigned messages, if $N < 4$ the problem has no solution –, in which servers can break-down and can be repaired at rate ξ and η, respectively. Jobs, in the form of messages/transactions, are handled by the system in order to agree on the validity of the submitted transactions. Transactions are bundled in blocks of transactions. Hence, once the validity of each transaction is confirmed, the block is committed by all the honest nodes, each in their own memory. Transactions in blocks that cannot be served immediately (because the system is busy) are stored in a finite memory buffer (with size J), until the memory buffer is not saturated, i.e. there are no memory slots free. At this point, arriving jobs would not be accepted and they will be lost. Transactions to be processed are modeled as a Poisson process with arrival rate λ, while service/processing time is exponentially distributed with rate μ.

Fig. 2 exemplifies the state diagram for the model elaborated. Nodes colored in red represent the area in which consensus is not reached, while nodes in green are those for which the system is available. The three indices, h, f, j, reported in each node are indicating states that the system may occupy.

Parameters of the model are: the total number of nodes $N \geq 4$, divided in honest $h \leq H$ and Byzantine nodes $f \leq F$, where $H \in [0, N]$ is the maximum number of honest nodes and $F \in [0, N]$ the maximum amount of Byzantine nodes, such that $H + F = N$; the buffer size $J > 0$ and the number of jobs $j \in [0, J]$ in the system; break-down rate $\xi > 0$, repair rate $\eta > 0$, arrival rate $\lambda > 0$, service rate $\mu > 0$, and timeout rate $\mu_t > 0$. In this model, it is assumed that N, H, h, F, f, J, j are positive integers. Therefore, for simplicity of exposition, when dealing with divisions, the *ceiling* $\lceil \cdot \rceil$ and *floor* $\lfloor \cdot \rfloor$ functions are implicitly applied, accordingly. Differently, all the rates - ξ, η, λ, μ, and μ_t - are positive real numbers. The latter, μ_t, is the rate indicating the process of losing a block of transactions, due to the internal time-out defined by the system. For instance, if a job cannot be served before the time-out occurs, it is not committed by the servers that received it and it is lost. This possibility may happen when the mean service time $1/\mu < 1/\mu_t$. There are several reasons why the system might be not able to serve jobs, notably the system can commit blocks only when it is available, i.e.

$$h > 2N/3. \tag{2}$$

For instance, in a BFT system, there may be present servers that are not acting accordingly to the rules set by the protocol – called Byzantine server – either maliciously, either because of malfunctioning. Hence, given a system with N servers, implementations of a BFT protocol tolerate up to $F < N/3$ Byzantine participants, that are acting deliberately in contrast with the network or are being unresponsive. In this formulation, however, unresponsive nodes are treated as broken-down servers, and not necessarily Byzantine.

In this mode model it is assumed that servers can break-down independently, but they are repaired sequentially, one at the time. Thus, the break-down rate ξ is multiplied by a number reflecting the current number of available nodes, i.e. if there are h nodes, the break-down rate is $h\xi$, while if there is only one node available ξ is the corresponding break-down rate. This is not the case for the repair process, since repairs occur only one at a time, with repair rate η.

From the state diagram in Fig. 2 the balance equations for the system are determined. In a compact way, the balance equations can be written as

$$\begin{aligned}
&\left[(2 - \delta_{hH} - \delta_{fF})\eta + (h+f)\xi\right] P_{h,f,j} + \\
&- \eta \left[P_{h-1,f,j}(1 - \delta_{h0}) + P_{h,f-1,j}(1 - \delta_{f0})\right] + \\
&- \xi \left[(h+1)P_{h+1,f,j}(1 - \delta_{hH}) - (f+1)P_{h,f+1,j}(1 - \delta_{fF})\right] + \\
&- \lambda\, P_{h,f,j-1}(1 - \delta_{j0}) - \mu\, P_{h,f,j+K}(1 - \delta_{jJ}) = 0,
\end{aligned} \tag{3}$$

where δ_{ij} indicates the Kronecker delta, i.e. $\delta_{ij} = 1$ if $i = j$ else $\delta_{ij} = 0$ if $i \neq j$. K indicates the possibility of bundling jobs in batches, hence serving all at once, up to a number of K transactions. Hence, compactly, Eq. (3) describes all the $(H+1)(F+1)(J+1)$ equations needed to describe transitions in the state diagram. Although, the condition that elements in $\vec{P}$ are probabilities imposes that

$$\sum_{j=0}^{J} \sum_{f=0}^{F} \sum_{h=0}^{H} P_{h,f,j} = 1. \tag{4}$$

Using the balance equations, for instance, the stationary probability distribution of the states in the system can be determined, hence the performability metrics. Therefore, because the stationary distribution of state probabilities $\vec{P}$ is to be found, the idea is to solve the matrix equation $\mathbf{Q}\vec{P} = 0$, where $\mathbf{Q}$ the coefficient matrix of the balance equations, i.e. the stochastic transition matrix associated with the CTMC. Balance equations in matrix form can be solved through many methods, but, since the null space of $\mathbf{Q}$ is to be determined, a simple method is to apply the Singular Value Decomposition (SVD).

However, in order to simplify the calculations and to avoid the state explosion (thus handling a tractable problem), it can be noted that the proposed model may be effectively divided into the product-form of – at least – two subsystems (Chandy & Martin, 1983). This means that the state diagram presented in Fig. 2 can be decomposed and represented as two independent processes. Figs. 3 and 4 are the state diagrams of the two subsystems. Here, Fig. 3 is a single layer on the plane (h, f) of Fig. 2, while Fig. 4 are chains parallel to the axis j of the same graphic. It is worthwhile to notice that the scheme reported in Fig. 3 has been

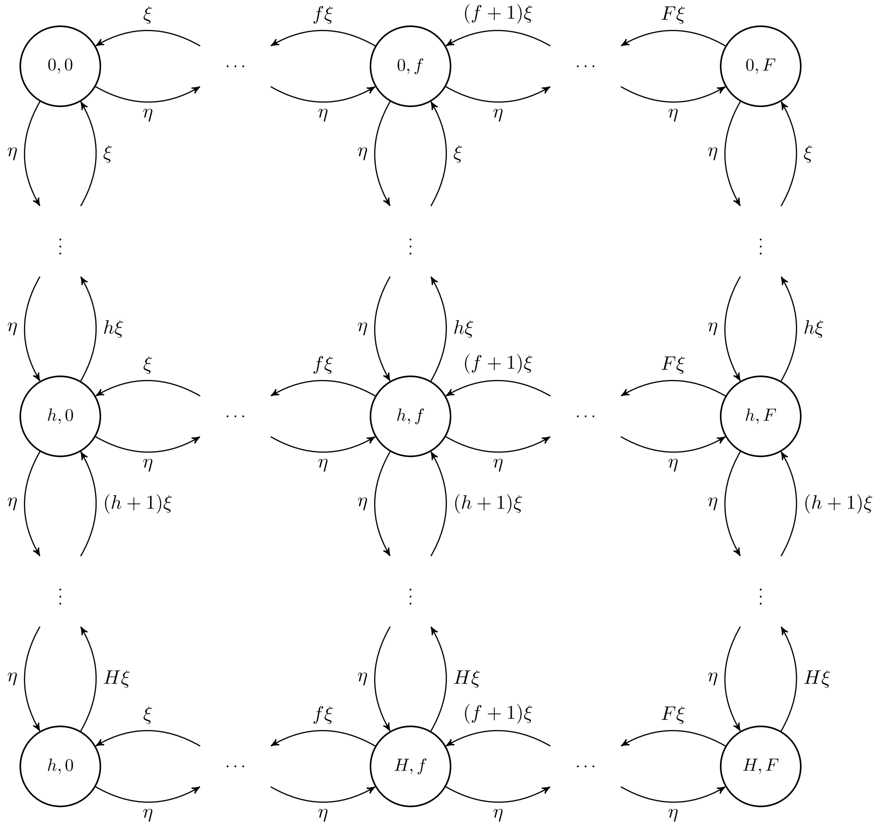
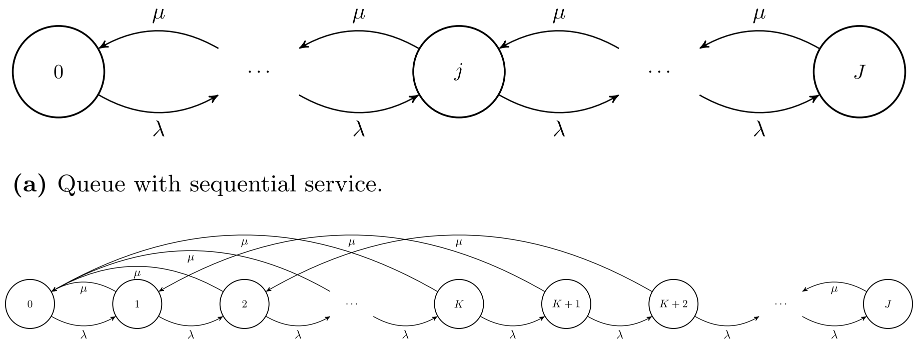


Fig. 3. One of the two components constituting the performability model for PBFT consensus protocol. This state diagram can be regarded as an availability model.



(a) Queue with sequential service.

(b) Queue with bulk/batch service.

Fig. 4. One of the two components constituting the performability model for PBFT consensus protocol. These two alternative state diagrams describe the processes of jobs arrival and service.

already presented and analyzed thoroughly in Marcozzi, Gemikonakli, Gemikonakli, Ever, and Mostarda (2023).

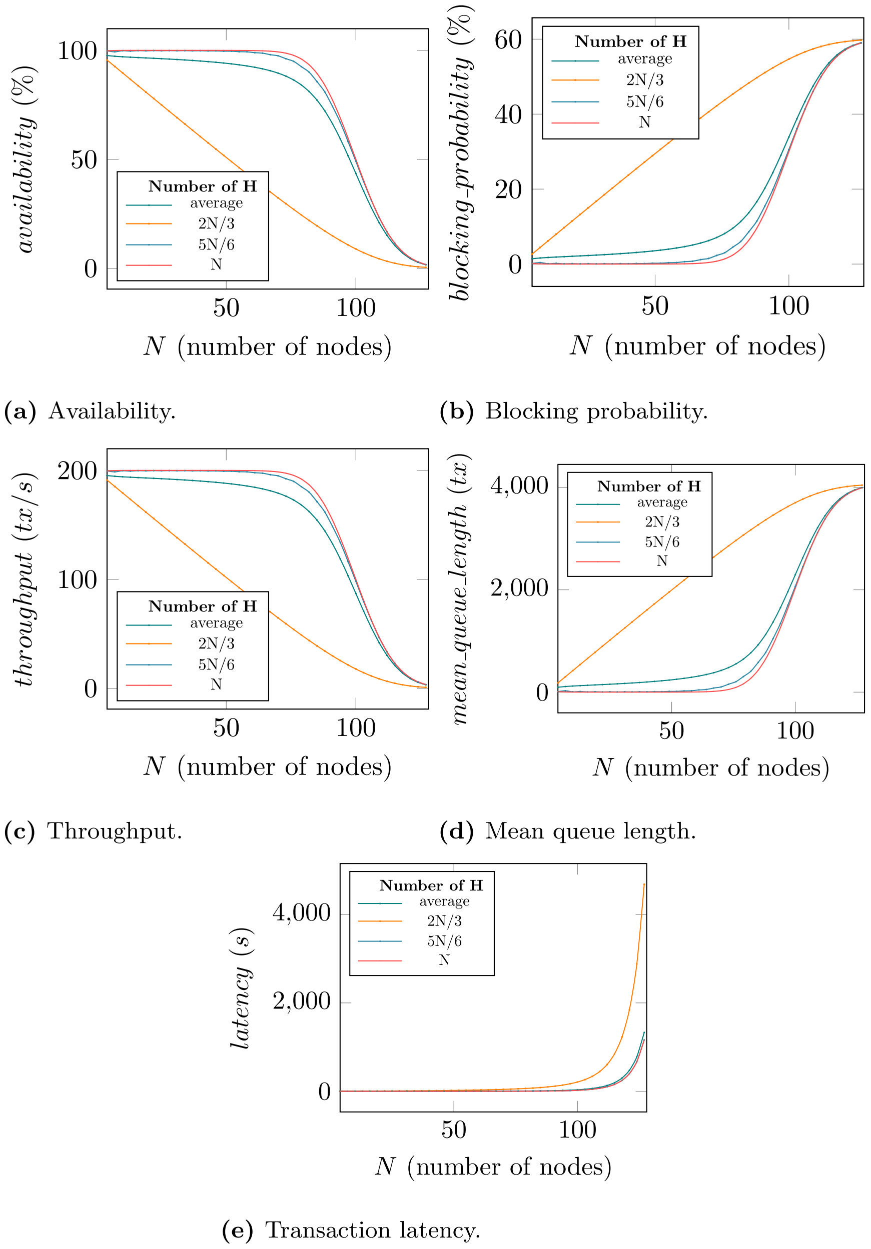
Diagrams in Fig. 4 present two distinct possibilities as serving policy: the systems may process single transactions (Fig. 4(a)), one at the time, or it can bundle them in blocks of transactions (Fig. 4(b)). While the former is a simple $M/M/1$ queue (Kendall, 1953), the latter is structured as a partial bulk/batch service queue, or $M/M^K/1$, where K is the maximum size of the batch (number of transactions in the block). Which model to use is up to the application under consideration and results will, in general, differ.

5. Performability analysis

The process to assess the performability metrics (see Algorithm 1 for an algorithmic presentation of the process) requires to define the model parameters $N \geq 4$, $J > 0$, and rates $\xi, \eta, \lambda, \mu, \mu_t > 0$. According to which one is considered as a free parameter, either H or F The matrix $\mathbf{Q}$ is determined using Eq. (3), and the solution $\vec{P}$ is computed through SVD. Elements in $\vec{P}$ are the stationary state probabilities of the system, and using these probabilities, important metrics associated with the system can be calculated. In this work, the following metrics can be computed: system availability, blocking probability, throughput, mean queue length, and latency (or latency).

This model allows also to compute the availability of the system, which is calculated as

$$A = \sum_{j=0}^{J} \sum_{f=0}^{F} \sum_{h>2N/3}^{H} P_{h,f,j}. \tag{5}$$



(a) Availability.

(b) Blocking probability.

(c) Throughput.

(d) Mean queue length.

(e) Transaction latency.

Fig. 5. Performability metrics as a function of the number of servers, where $J = 4096$, $K = 100$, $\xi = 5.02 \cdot 10^{-7}$, $\eta = 3.47 \cdot 10^{-5}$, $\lambda = 250$, $\mu = 1000$, and $\mu_t = 1$.

```
Algorithm 1 Pseudo-code to calculate performability metrics
Require: $N \geq 4$ and ($H \in [0, N]$ or $F \in [0, N]$) and $J > 0$ and
  $\xi, \eta, \lambda, \mu, \mu_t > 0$
  $F \leftarrow N - H$                                        ▷ or $H \leftarrow N - F$
  $\mathbf{Q} \leftarrow \mathbf{Q}(H, F, J, \xi, \eta, \lambda, \mu, \mu_t)$     ▷ generate the matrix of coefficients
  $\vec{P} \leftarrow SVD(\mathbf{Q}, 0)$             ▷ compute state probabilities through SVD
  $A \leftarrow \sum_{j=0}^{J} \sum_{f=0}^{F} \sum_{h>2N/3}^{H} P_{h,f,j}$                    ▷ availability
  $bp \leftarrow \sum_{f=0}^{F} \sum_{h=0}^{H} P_{h,f,J}$                          ▷ blocking probability
  $thr \leftarrow \mu \sum_{j=1}^{J} \sum_{f=0}^{F} \sum_{h>2N/3}^{H} P_{h,f,j}$                 ▷ thoughput
  $mql \leftarrow \sum_{j=0}^{J} \sum_{f=0}^{F} \sum_{h=0}^{H} j\, P_{h,f,j}$                ▷ mean queue length
  $lat \leftarrow mql/thr$                                              ▷ latency
```

Blocking probability

$$blocking\_probability = \sum_{f=0}^{F} \sum_{h=0}^{H} P_{h,f,J} \tag{6}$$

is the measurement that estimates the possibility of transactions being lost because of full memory buffer. Throughput is the amount of jobs being served by the system in the unit time:

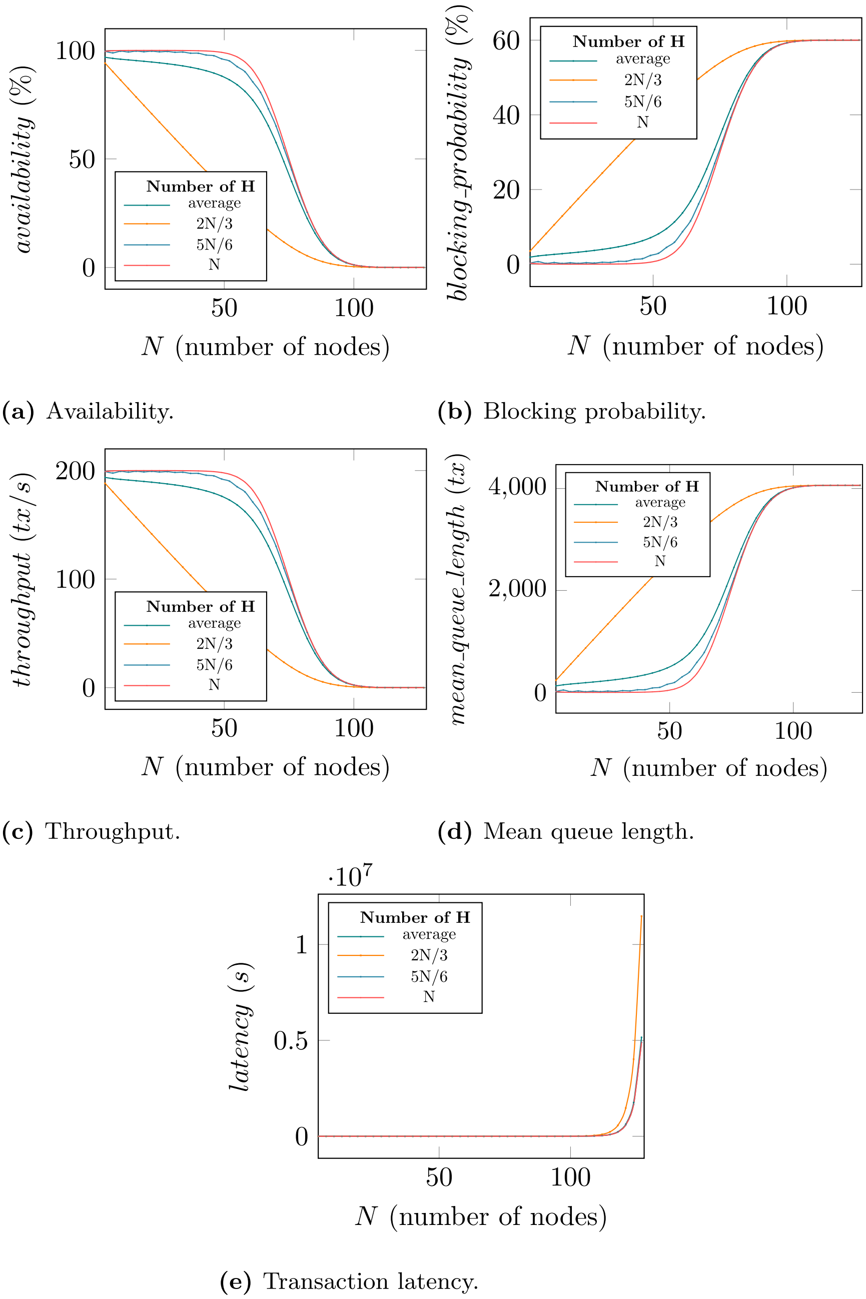
$$throughput = \mu \sum_{j=1}^{J} \sum_{f=0}^{F} \sum_{h>2N/3}^{H} P_{h,f,j}, \tag{7}$$

where $h > 2N/3$ indicates that the system can serve jobs only if there are enough available machines, i.e. their number h is greater or equal than the quorum. The mean queue length is the average number of jobs present in the system and it can be determined by enumerating

$$mean\_queue\_length = \sum_{j=0}^{J} \sum_{f=0}^{F} \sum_{h=0}^{H} j\, P_{h,f,j}, \tag{8}$$

The latency of the system, instead, is the amount of time required for a job to leave the system, hence it is

$$latency = \frac{mean\_queue\_length}{throughput}. \tag{9}$$



(a) Availability.

(b) Blocking probability.

(c) Throughput.

(d) Mean queue length.

(e) Transaction latency.

Fig. 6. Performability metrics as a function of the number of servers, where $J = 4096$, $K = 100$, $\xi = 6.94 \cdot 10^{-7}$, $\eta = 3.47 \cdot 10^{-5}$, $\lambda = 250$, $\mu = 1000$, and $\mu_t = 1$.

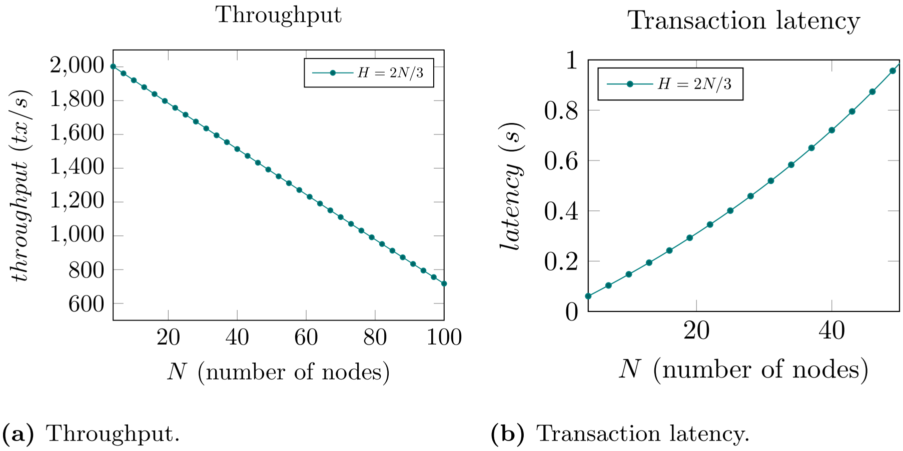
Finally, the procedure described above (and summarized in Algorithm 1) can be iterated over a range of Ns, such that the relation between performability metrics and number of nodes N is explored. Similarly, this methodology allows the investigators to study the connection between different aspects of the system under examination, simply by variating the parameters of interest.

6. Results and discussion

In this section, results are presented to evaluate the effects of distinct parameters, e.g. number of servers and Byzantine nodes, on the performability metrics of PBFT systems.

Graphs in Fig. 5 show how the number of servers, $N \in [4, 127]$, and the ratio of Byzantine nodes in the system are effecting the performances and availability of a PBFT system. In this context, Byzantine nodes are determined by counting the amount of honest nodes, $H = N, 5N/6, 2N/3$ and the average value for all N. Parameters used for the computation of these metrics are: $J = 4096$, $K = 100$, $\xi = 5.02 \cdot 10^{-7}$, $\eta = 3.47 \cdot 10^{-5}$, $\lambda = 250$, $\mu = 1000$, and $\mu_t = 1$. This collection of results reported in Fig. 5 displays a characteristic behavior in the performability of the system. There is a marked reduction in performance as the number of servers increases, with a seemingly threshold at $N \approx 60$, and performance worsen when the number of Byzantine nodes increment. As it might be expected, metrics are related and they show similar behavior in pairs: Figs. 5(a) and 5(c); Figs. 5(b) and 5(d); Fig. 5(e) is not coupled.

In addition to test the relation between number of servers and Byzantine nodes, the described methodology can be applied to study the behavior of the performability metrics due to an higher breakdown rate to repair rate ratio (ξ/η). For instance, the value of ξ can be increased, then the same procedure described in Algorithm 1 can be applied. The results obtained from this analysis are shown in Fig. 6.



(a) Throughput. **(b)** Transaction latency.

Fig. 7. Throughput and latency as a function of the number of servers, where $J = 4096$, $K = 100$, $\xi = 3.47 \cdot 10^{-7}$, $\eta = 3.47 \cdot 10^{-5}$, $\lambda = 2600$, $\mu = 10\,000$, and $\mu_t = 1$.

It can be noted that the number of servers at which the performances of the system are sharply degrading is shifted to the left, at $N \approx 40$. Besides for this shift in performance, the other considerations made for the results in Fig. 5 are, otherwise, applying also to Fig. 6. Likewise, the results pertaining the availability metric (Figs. 5(a) and 6(a)) are matching the ones obtained in Marcozzi et al. (2023).

In Fig. 7 there are results, which are replicating – at least in their outline and approximate values – two studies found in literature. In particular, Fig. 7(a) reproduce the values of throughput presented in Tang, Wang, Jiang, Ge, and Tan (2022). However, the analytical result (Fig. 7(a)) does not have an arched shape and a reduced decrease in the value of throughput at $N \approx 80$. Similarly, Fig. 7(b) is presenting results similar to the values of transaction latency obtained in the benchmark reported in Liu, Zhang, Feng, Huang, and Xu (2022). In this case, the shape of the two graphs is matching, except from irregularities in the plot found in literature. The parameters used to obtain these results are: $J = 4096$, $K = 100$, $\xi = 3.47 \cdot 10^{-7}$, $\eta = 3.47 \cdot 10^{-5}$, $\lambda = 2600$, $\mu = 10\,000$, and $\mu_t = 1$. Here, it is assumed that the number of Byzantine nodes is $F = N/3$, hence the honest nodes are $H = 2N/3$.

Fig. 8 presents a comparison between the benchmark of Tendermint (Buchman, Kwon, & Milosevic, 2018) reported in Fu et al. (2020) and the values obtained from the performability analysis. It can be noted that, while the trend of the data is reproduced by the analytical results, it fails to match consistently the values in the error interval given by the published data. The parameters used to obtain the analytical results are: $J = 4096$, $K = 3000$, $\xi = 6.59 \cdot 10^{-7}$, $\eta = 3.47 \cdot 10^{-5}$, $\mu = 0.5$, and $\mu_t = 0.2$. Here, it is assumed that the number of Byzantine nodes is $F = N/4$, hence the honest nodes are $H = 3N/4$.

In summary, from the results presented above, it can be concluded that all system's performability metrics are indeed non-linearly dependent on the number of the servers in the network. In particular, the performances are inversely proportional respect to the number of nodes. For instance, favorable metrics (availability and throughput) are decreasing at the increase of N, while the values of disadvantageous metrics (blocking probability, mean queue length, and transaction latency) are increasing. This tendency results strengthened when the rate of break-downs increases over the rate of repairs, i.e. performance decrease at the increase of ξ, or the decrease of η. Concerning the correspondence between the analytical results and data obtainable from the literature, there is indeed a certain degree of agreement in the general trend, but analytical results fail to match consistently benchmark data in the interval error provided.

7. Conclusion

The importance of reliable and effective distributed systems extends to many fields of science, engineering, and technology, with mission-critical applications in healthcare, productive processes, networked computers, etc. However, in recent time, a renewed interest in this topic is due to the development and increasing significance of distributed ledgers, blockchains, and cryptocurrencies.

Because of the relevance of these technologies and their applications, it is needed that solutions based on distributed systems are, indeed, reliable and they can sustain the workload necessary to achieve the intended goals. For instance, there are application in which downtime is not admissible, or others in which the number of participants is relatively high, but the performance cannot be compromised. Therefore, a deep understanding of distributed systems is necessary to properly plan, design, and develop platforms up to the expected requirements.

BFT protocols form an prominent class of fault-tolerant protocols. BFT systems are resilient even when malicious actors are partaking in the achievement of a common goal. However, because how messages are exchanged in BFT schemes, platforms based on a BFT consensus protocol have bottlenecks in terms of performance, as the number of participants increases. Therefore, it is of vital importance to know the limitations of the developed solution, given some parameters and expected performances. To obtain such information, there are different options, e.g., expensive benchmarks or challenging simulations of the systems. An other approach is to realize a mathematical model describing the system and get performance information using the model.

In this study, it is presented an analytical model to describe protocols based on PBFT consensus. This model – founded on queueing theory and CTMC – can assess important performance metrics, such as: availability, blocking probability, throughput, mean queue length, and latency. The model analyzes systems in the presence of break-downs and repairs, when malicious nodes are present. Service of jobs can happen both sequentially, message after message, or in batches. The total number of nodes N considered changes in a predefined range (maximum $N = 127$), while the proportion of malicious nodes is taken as a "high" ($N = 2N/3$), "medium" ($N = 5N/6$), none present ($N = H$), or the average value. Numerical results are presented reporting performance metrics as a function of the number of participants and a relative number of honest actors in the system, as well as in relation with the break-down and repair rates. The contribution of this work is to present a model the evaluate the performability of PBFT-based distributed systems. From the model, it can be concluded that there is a non-linear relationship between the number of servers and performability, with performance effects inversely proportional to the number of nodes in the system. This relationship is highlighted as the ratio of break-down rate to the repair rate increases. In addition, by mean of this model, it has been possible to replicate some results obtained through benchmarking, as found in literature, showing potential in prediction of actual metrics, even if limited in accuracy.

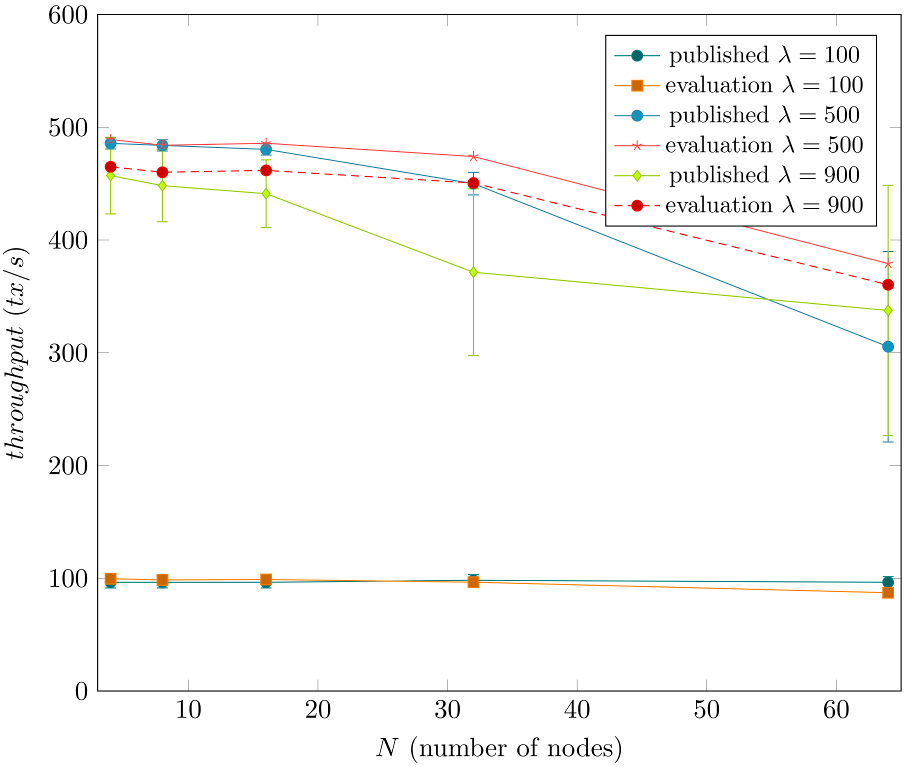


Fig. 8. Comparison between the throughout of Tendermint from literature (Fu et al., 2020) and values of throughput obtained from the performability analysis.

Potential extensions of this work include a deeper analysis of the relations between uncoupled parameters and performability in the form of threshold analysis for different scenarios. This may include a stochastic analysis on the occurrence of Byzantine nodes in distributed systems. Additionally, this model can be used as a predictive tool to characterize system's performance and availability metrics, then validate the results through benchmark or simulation. Particularly, such study would actually test the predictive capacities of this model, and it could be even possible to estimate the error range for the results obtained applying the presented model.

CRediT authorship contribution statement

Marco Marcozzi: Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. **Leonardo Mostarda:** Conceptualization, Resources, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request

References

Balsamo, S., Malakhov, I., Marin, A., & Mitrani, I. (2022). Transaction confirmation in proof-of-work blockchains: Auctions, delays and droppings. In *2022 20th Mediterranean Communication and Computer Networking Conference* (pp. 140–149). IEEE.

Balsamo, S., Marin, A., Mitrani, I., & Rebagliati, N. (2021). Prediction of the consolidation delay in blockchain-based applications. In *Proceedings of the ACM/SPEC International Conference on Performance Engineering* (pp. 81–92).

Belotti, M., Bozic, N., Pujolle, G., & Secci, S. (2019). A vademecum on blockchain technologies: When, which, and how. *IEEE Communications Surveys & Tutorials*, *21*(4), 3796–3838. http://dx.doi.org/10.1109/COMST.2019.2928178.

Berdik, D., Otoum, S., Schmidt, N., Porter, D., & Jararweh, Y. (2021). A survey on blockchain for information systems management and security. *Information Processing & Management*, *58*(1), Article 102397. http://dx.doi.org/10.1016/j.ipm.2020.102397.

Bodkhe, U., Tanwar, S., Parekh, K., Khanpara, P., Tyagi, S., Kumar, N., et al. (2020). Blockchain for Industry 4.0: A comprehensive review. *IEEE Access*, *8*, 79764–79800. http://dx.doi.org/10.1109/ACCESS.2020.2988579.

Bolch, G., Greiner, S., De Meer, H., & Trivedi, K. S. (2006). *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*. John Wiley & Sons.

Buchman, E., Kwon, J., & Milosevic, Z. (2018). The latest gossip on BFT consensus. arXiv preprint arXiv:1807.04938.

Castro, M., Liskov, B., et al. (1999). Practical Byzantine fault tolerance. In *OsDI. Vol. 99* (pp. 173–186).

Chandy, K. M., & Martin, A. J. (1983). A characterization of product-form queuing networks. *Journal of the ACM*, *30*(2), 286–299.

Chang, Y.-X., Li, Q.-L., Wang, Q., & Song, X.-S. (2022). Dynamic practical Byzantine fault tolerance and its blockchain system: A large-scale Markov modeling. arXiv preprint arXiv:2210.14003.

Dunphy, P., & Petitcolas, F. A. (2018). A first look at identity management schemes on the blockchain. *IEEE Security & Privacy*, *16*(4), 20–29.

Erlang, A. K. (1909). The theory of probabilities and telephone conversations. *Nyt. Tidsskr. Mat. Ser. B*, *20*, 33–39.

Fan, C., Ghaemi, S., Khazaei, H., & Musilek, P. (2020). Performance evaluation of blockchain systems: A systematic survey. *IEEE Access*, *8*, 126927–126950. http://dx.doi.org/10.1109/ACCESS.2020.3006078.

Fralix, B. (2020). On classes of Bitcoin-inspired infinite-server queueing systems. *Queueing Systems*, *95*, 29–52. http://dx.doi.org/10.1007/S11134-019-09643-W.

Frolkova, M., & Mandjes, M. (2019). A Bitcoin-inspired infinite-server model with a random fluid limit. *Stochastic Models*, *35*(1), 1–32.

Fu, W.-K., Lin, Y.-S., Campagna, G., Liu, C.-T., Tsai, D.-Y., Mei, C.-H., et al. (2020). Soteria: A provably compliant user right manager using a novel two-layer blockchain technology. In *2020 IEEE Infrastructure Conference* (pp. 1–10). IEEE.

Geissler, S., Prantl, T., Lange, S., Wamser, F., & Hossfeld, T. (2019). Discrete-time analysis of the blockchain distributed ledger technology. In *Proceedings of the 31st International Teletraffic Congress, ITC 2019* (pp. 130–137). Institute of Electrical and Electronics Engineers Inc, http://dx.doi.org/10.1109/ITC31.2019.00029.

Huang, D., Ma, X., & Zhang, S. (2019). Performance analysis of the Raft consensus algorithm for private blockchains. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *50*(1), 172–181.

Jiang, L., Chang, X., Liu, Y., Mišić, J., & Mišić, V. B. (2020). Performance analysis of Hyperledger Fabric platform: A hierarchical model approach. *Peer-to-Peer Networking and Applications*, *13*, 1014–1025. http://dx.doi.org/10.1007/S12083-019-00850-Z/FIGURES/10.

Kawase, Y., & Kasahara, S. (2017). Transaction-confirmation time for bitcoin: A Queueing analytical approach to blockchain mechanism. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*: *vol. 10591 LNCS*, (pp. 75–88). Springer Verlag, http://dx.doi.org/10.1007/978-3-319-68520-5_5.

Kendall, D. G. (1953). Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded Markov chain. *The Annals of Mathematical Statistics*, 338–354.

Kolb, J., AbdelBaky, M., Katz, R. H., & Culler, D. E. (2020). Core concepts, challenges, and future directions in blockchain. *ACM Computing Surveys*, *53*(1), 1–39. http://dx.doi.org/10.1145/3366370.

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

Lao, L., Li, Z., Hou, S., Xiao, B., Guo, S., & Yang, Y. (2020). A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling. *ACM Computing Surveys*, *53*(1), http://dx.doi.org/10.1145/3372136.

Law, A. M., Kelton, W. D., & Kelton, W. D. (2007). *Simulation Modeling and Analysis. Vol. 3*. New York: Mcgraw-hill.

Li, Q. L., Ma, J. Y., & Chang, Y. X. (2018). Blockchain queue theory. *Lecture Notes in Computer Science*, *11280 LNCS*, 25–40. http://dx.doi.org/10.1007/978-3-030-04648-4_3.

Li, Q. L., Ma, J. Y., Chang, Y. X., Ma, F. Q., & Yu, H. B. (2019). Markov processes in blockchain systems. *Computational Social Networks*, *6*, 1–28. http://dx.doi.org/10.1186/S40649-019-0066-1/FIGURES/4.

Liu, W., Zhang, X., Feng, W., Huang, M., & Xu, Y. (2022). Optimization of PBFT algorithm based on QoS-aware trust service evaluation. *Sensors*, *22*(12), 4590.

Ma, F. Q., & Fan, R. N. (2022). Queuing theory of improved practical Byzantine fault tolerant consensus. *Mathematics*, *10*, 182. http://dx.doi.org/10.3390/MATH10020182.

Ma, Z., Fan, J., Zhang, Y., & Liu, L. (2020). Performance analysis of blockchain consensus system with interference factors and sleep stage. *IEEE Access*, *8*, 119010–119019. http://dx.doi.org/10.1109/ACCESS.2020.3005919.

Marcozzi, M., Gemikonakli, O., Gemikonakli, E., Ever, E., & Mostarda, L. (2023). Availability evaluation of IoT systems with Byzantine fault-tolerance for mission-critical applications. *Internet of Things*, *23*, Article 100889. http://dx.doi.org/10.1016/j.iot.2023.100889.

Meng, T., Zhao, Y., Wolter, K., & Xu, C. Z. (2021). On consortium blockchain consistency: A queueing network model approach. *IEEE Transactions on Parallel and Distributed Systems*, *32*, 1369–1382. http://dx.doi.org/10.1109/TPDS.2021.3049915.

Mollah, M. B., Zhao, J., Niyato, D., Lam, K.-Y., Zhang, X., Ghias, A. M., et al. (2020). Blockchain for future smart grid: A comprehensive survey. *IEEE Internet of Things Journal*, *8*(1), 18–43.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, 21260.

Namasudra, S., & Deka, G. C. (2021). *Applications of Blockchain in Healthcare*. Springer.

Nischwitz, M., Esche, M., & Tschorsch, F. (2021). Bernoulli meets PBFT: Modeling BFT protocols in the presence of dynamic failures. In *2021 16th Conference on Computer Science and Intelligence Systems* (pp. 291–300). IEEE.

Paulavičius, R., Grigaitis, S., Igumenov, A., & Filatovas, E. (2019). A decade of blockchain: Review of the current status, challenges, and future directions. *Informatica*, *30*(4), 729–748. http://dx.doi.org/10.15388/Informatica.2019.227.

Pournader, M., Shi, Y., Seuring, S., & Koh, S. C. (2020). Blockchain applications in supply chains, transport and logistics: A systematic review of the literature. *International Journal of Production Research*, *58*(7), 2063–2081. http://dx.doi.org/10.1080/00207543.2019.1650976.

Qi, J., Yu, J., & Jin, S. (2020). Nash equilibrium and social optimization of transactions in blockchain system based on discrete-time queue. *IEEE Access*, *8*, 73614–73622. http://dx.doi.org/10.1109/ACCESS.2020.2986084.

Rao, S. S. (2019). *Engineering Optimization: Theory and Practice*. John Wiley & Sons.

Rasolroveicy, M., Haouari, W., & Fokaefs, M. (2021). Public or private? A techno-economic analysis of blockchain. In *Proceedings of the 31st Annual International Conference on Computer Science and Software Engineering* (pp. 83–92).

Ricci, S., Ziviani, A., Ferreira, E., Souza, J. E., Menasché, D. S., & Vieira, A. B. (2019). Learning blockchain delays. *ACM SIGMETRICS Performance Evaluation Review*, *46*, 122–125. http://dx.doi.org/10.1145/3308897.3308952.

Rimba, P., Tran, A. B., Weber, I., Staples, M., Ponomarev, A., & Xu, X. (2017). Comparing blockchain and cloud services for business process execution. In *2017 IEEE International Conference on Software Architecture* (pp. 257–260). IEEE.

Smetanin, S., Ometov, A., Kannengieser, N., Sturm, B., Komarov, M., & Sunyaev, A. (2020). Modeling of distributed ledgers: Challenges and future perspectives. In *Proceedings - 2020 IEEE 22nd Conference on Business Informatics, CBI 2020. Vol. 1* (pp. 162–171). Institute of Electrical and Electronics Engineers Inc, http://dx.doi.org/10.1109/CBI49978.2020.00025.

Smetanin, S., Ometov, A., Komarov, M., Masek, P., & Koucheryavy, Y. (2020). Blockchain evaluation approaches: State-of-the-art and future perspective. *Sensors*, *20*, 3358. http://dx.doi.org/10.3390/S20123358.

Sukhwani, H., Martínez, J. M., Chang, X., Trivedi, K. S., & Rindos, A. (2017). Performance modeling of PBFT consensus process for permissioned blockchain network (Hyperledger Fabric). In *2017 IEEE 36th Symposium on Reliable Distributed Systems* (pp. 253–255). IEEE.

Tang, S., Wang, Z., Jiang, J., Ge, S., & Tan, G. (2022). Improved PBFT algorithm for high-frequency trading scenarios of alliance blockchain. *Scientific Reports*, *12*(1), 4426.

Trivedi, K. S. (2008). *Probability & Statistics with Reliability, Queuing and Computer Science Applications*. John Wiley & Sons.

Wilhelmi, F., & Giupponi, L. (2021). Discrete-time analysis of wireless blockchain networks. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC. 2021-September* (pp. 1011–1017). Institute of Electrical and Electronics Engineers Inc, http://dx.doi.org/10.1109/PIMRC50174.2021.9569253.

Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, *151*(2014), 1–32.

Zheng, Z., Xie, S., Dai, H.-N., Chen, W., Chen, X., Weng, J., et al. (2020). An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems*, *105*, 475–491. http://dx.doi.org/10.1016/j.future.2019.12.019.