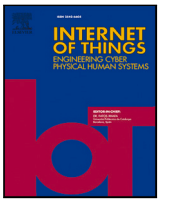
Internet of Things 23 (2023) 100889

Contents lists available at ScienceDirect

Internet of Things

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



Research article

Availability evaluation of IoT systems with Byzantine fault-tolerance for mission-critical applications

Marco Marcozzi [a,*], Orhan Gemikonakli [b], Eser Gemikonakli [c], Enver Ever [d], Leonardo Mostarda [a]

[a] Computer Science Division, University of Camerino, Via Madonna delle Carceri, 7, Camerino (MC), 62032, Italy
[b] Faculty of Engineering, Final International University, Kyrenia, Cyprus
[c] Department of Computer Engineering, Faculty of Engineering, University of Kyrenia, Mersin 10, Girne, Turkey
[d] Computer Engineering, Middle East Technical University Northern Cyprus Campus, Mersin 10, Guzelyurt, 99738, Turkey

ARTICLE INFO

Keywords:
Byzantine consensus protocol
Fault tolerant systems
Availability
Stochastic processes
Blockchain

ABSTRACT

Byzantine fault-tolerant (BFT) systems are able to maintain the availability and integrity of IoT systems, in the presence of failure of individual components, random data corruption or malicious attacks. Fault-tolerant systems in general are essential in assuring continuity of service for mission-critical applications. However, their implementation may be challenging and expensive. In this study, IoT Systems with BFT are considered. Analytical models and solutions are presented as well as a detailed analysis for the evaluation of the availability. BFT is particularly important for blockchain mechanisms, and in turn for IoT, since it can provide a secure, reliable and decentralized infrastructure for IoT devices to communicate and transact with each other. A continuous-time Markov chain is used to model the IoT systems with BFT where the breakdown and repair times follow exponential distributions, and the number of the Byzantine nodes in the network follows various distributions. The presented numerical findings demonstrate the relationship between the number of nodes in the system, the proportion of honest users, and the overall availability. Based on the model, it can be inferred that the correlation between the scale of the system (nodes) and network availability is non-linear. Additionally, results show that even for relatively small-size systems with 40 nodes, an average availability greater than 0.999 and an estimated downtime per year that is less than 9 h is possible.

1. Introduction

IoT (Internet of Things) devices are often deployed in distributed and decentralized environments, where a large number of devices need to communicate and coordinate with each other to perform various tasks. BFT (Byzantine Fault Tolerance) systems can benefit IoT, particularly in terms of improved reliability and fault tolerance. BFT systems are designed to ensure the system's reliability and fault tolerance in a distributed environment. They can tolerate a certain number of faulty or malicious nodes in the network without compromising the system's overall performance. This feature can be particularly beneficial in IoT systems where a large number of devices need to communicate and coordinate with each other, and the failure of a single node can have a significant impact on the entire system's performance [1].

* Corresponding author.
E-mail address: marco.marcozzi@unicam.it (M. Marcozzi).

https://doi.org/10.1016/j.iot.2023.100889
Received 30 June 2023; Received in revised form 21 July 2023; Accepted 1 August 2023
Available online 7 August 2023

Fault-tolerance in general is a critical concept when designing and implementing high-availability systems. High-availability systems are those that are designed to operate continuously without interruption and are essential for applications where downtime can have serious consequences, as in healthcare, finance, transportation, industry, information technology, and communication systems [2,3]. Assuredly, fault-tolerance has many applications in various engineering fields such as automotive, aerospace, and avionics [4–6], as well as distributed computing such as cloud computing [7–10], and other distributed systems [11,12].

Fault-tolerant distributed systems are attracting increasing interest due to the possibilities connected with the applications of blockchains particularly in cyber–physical systems such as IoT and Industry 4.0 applications. In addition, blockchain technologies are also used in finance, and other distributed computing applications such as Smart Contracts. Blockchain is considered a subgroup of distributed ledger technology (DLT). DLT requires a consensus protocol to commit transactions (e.g. read/write operations on the local storage of its members, or to perform some actuation scheme). Indeed, blockchains are popular because they record information in a way that makes it difficult or impossible to change, hack, or cheat the system. Transactions are duplicated and distributed across the entire network of computer systems on the blockchain. However, data theft has seen an increasing threat, especially when financial transactions are concerned.

Blockchain technology is being integrated into IoT systems to improve their availability [13]. A key benefit of blockchain technology is its ability to provide a decentralized infrastructure that can be used to manage and distribute resources across a network of devices. In IoT systems, this can be particularly useful for managing and distributing computational resources, such as processing power and storage capacity, to ensure that the system remains available and responsive even when individual devices fail [14]. For example, blockchain-based smart contracts can be used to automatically allocate computational resources to IoT devices based on their needs and availability [15]. Additionally, blockchain can be used to create a distributed ledger of device availability and resource usage, making it easier to manage and monitor the health of an IoT system.

Various techniques have been developed and implemented to enhance the fault tolerance of systems, aiming to ensure different levels of resilience against faults. This study specifically focuses on distributed computer systems that are capable of tolerating Byzantine faults [16]. A Byzantine fault refers to a scenario where a node within the network behaves maliciously, such as sending conflicting messages to different nodes or becoming unresponsive. It is important to note that this definition is distinct from a crash, which occurs when a node is not malicious but becomes unresponsive due to technical failures like power or connectivity outages. In both cases, whether it is a Byzantine fault or a crash, the system may encounter difficulties in achieving a consensus.

The initial research work that introduced the BFT scheme for achieving consensus in distributed computing systems [16] was structured using an analogy centered on the "Byzantine generals problem". Consequently, BFT became synonymous with this type of systems. In a general sense, the "Byzantine generals problem" asserts that in a scenario where successful coordination is required for victory, the presence of uncertain loyalty among any of the generals (such as an unanticipated retreat by one or more previously committed generals) can fail the attack. A solution to this problem is asking each general to send to any other attacker a message containing the plan of action and execute it if a predefined quorum is reached. In a similar fashion BFT protocols employ consensus algorithms that allow a network of nodes to reach an agreement on the order and validity of transactions or operations. By tolerating a certain number of faulty or malicious nodes, BFT systems provide robustness and reliability. Unlike traditional fault-tolerant systems that assume fail-stop failures, BFT systems handle Byzantine failures, where components may exhibit arbitrary and contradictory behaviors. This makes them suitable for environments where malicious actors can deliberately deviate from the expected behavior and attempt to disrupt the system's operation. The development of BFT systems continues to be an active area of research, aiming to improve their performance, scalability, and resilience against sophisticated attacks.

To implement a BFT consensus protocol, the minimum number of nodes required is $N \geq 4$ when they exchange unsigned messages. Indeed, with unsigned messages, if $N < 4$, the problem does not have a solution [16]. In BFT systems, the term "quorum" refers to the minimum number of commit messages required to achieve consensus. A quorum is obtained when the count of responsive nodes that are honest, denoted as h, reaches a certain threshold defined as:

$$h > 2N/3. \tag{1}$$

Hence, a BFT distributed system, in which unsigned messages are exchanged, can handle up to f Byzantine faults, where

$$f < N/3. \tag{2}$$

Prior to the implementation of a BFT protocol, conducting an availability evaluation is an essential analysis that must be performed. This evaluation plays a vital role in ensuring the successful application of the system. Indeed, the implementation of a computer network can be both costly and technically demanding, often leading to potential underperformance in terms of expected availability levels. In light of this consideration, analytical techniques to study a desired system have been developed. The advantage of the analytical approach resides in the possibility of tuning the parameters characterizing the modeled system in a straightforward and inexpensive way. This is particularly true for what concerns the development of DLTs, where analytical approaches applied to the study of network availability can address security bottlenecks caused by the malevolent nodes or crashes in the system. This is critical particularly in a decentralized environment, because there is no central authority enforcing network policies or scheduling repairs, therefore a thoughtful knowledge of critical scenarios is necessary to overcome possible service downtime. An evident downside is the difficulty of the development of an analytical model suitable to describe the network. Nevertheless, over the past few decades, certain analytical methodologies such as continuous-time Markov chains (CTMC) have been extensively and effectively utilized to assess the availability of intricate systems. [17,18].

In this article, we present an analytical model (based on CTMCs), that is able to evaluate BFT systems availability. The IoT systems are considered as collections of computing nodes [19–21]. In [19] a dependability study of IoT devices is considered.

The study presents closed-form solutions to represent system models such as Markov chains, and Reliability Block Diagrams. Similarly, [20] explores IoT devices and edge-level applications in Fog computing environments, employing a generic CTMC-based resource availability model. Additionally, [21] develops a CTMC model and utilizes sensitivity analysis to enhance system availability. Our approach aligns with these studies by considering IoT applications at a generic level, making it applicable to similar scenarios. The significant additional contribution of our approach lies in incorporating BFT. Markov analysis is performed in [22] as well for a more specific setup to evaluate the protection against data loss and availability for two main improvements on Raft consensus algorithm. The approach presented in this study can be adopted for similar consensus algorithms. However, our current focus is on more generic BFT consensus approaches. The model presented is employed to investigate the relationship between different parameters, in order to identify the best configurations to maximize the availability.

Since the occurrence of Byzantine nodes is not treated as a dynamic process, i.e. a malicious node does not change its stance over time, it is vital to understand what scenarios are to be expected in this framework. A critical downside for this kind of assumption is the complex estimation of how many Byzantine nodes may be present in the network for a given configuration. In [23], the assumption of a degenerate distribution for the number of Byzantine nodes in the system led to the definition of three levels of Byzantine threats; low, medium, and high level. This work, instead, presents a novel approach to estimate the number of Byzantine nodes in a given network. Considering the number of Byzantine nodes to be the result of a stochastic process, the probabilistic distribution used for the number of malicious nodes depends on very specific features characterizing the system under investigation. Indeed, this study proposes a methodology to describe the system availability in the presence of f Byzantine actors, where the distribution of f is a choice of the decision-maker/investigator or a characteristic of the system itself. The methodology is useful in case the distribution of the values of f is known because the model should accurately predict and reproduce the behavior of the system under investigation. Conversely, if the statistical properties of the process describing the number of Byzantine nodes are not known, this methodology allows testing different distributions, in order to outline possible scenarios and design the implementation of the system according to the retrieved information. The advantage of this approach is two-fold: it provides a light-weight framework to assess system availability, indicating the best and worst cases without the need to actually develop and implement the system. The contributions of this study can be summarized as follows:

- In this study, we propose an analytical approach to model Byzantine nodes in the presence of malicious nodes and failures.
- An iterative algorithm is also presented to calculate availability for various distributions and parameters (distribution dependent) of a number of Byzantine faults.
- The method presented is tested for Uniform, Poisson, Binomial and Degenerate distributions.

The rest of this study is organized as follows. In Section 2, a comprehensive review of prior research on the analytical availability models is presented. Section 3 describes the availability model proposed, along with its underlying assumptions. The mathematical prerequisites to derive a solution for the model are investigated in Section 4. Section 5 presents the obtained results from the study. Lastly, Section 6 summarizes the findings, discusses potential applications, and outlines future advancements for the presented model.

2. Related work

The definition of availability can vary depending on the specific context, but it is generally characterized as the system's ability to carry out its intended operation at any given moment. In particular, availability might be viewed as a failure-free operation at any time t, in which case it takes the name of point or instantaneous availability. Point availability, denoted as $A(t)$, is formally defined as the probability that the analyzed component functions correctly at a given time t.

$$A(t) = W(t) + \int_0^t W(t-x)\, r(x)\, dx$$

here $W(t)$ is the probability of not experiencing any failures for the component in the interval $[0, t]$ whereas $r(x)$ is the repair frequency. Clearly, the equation demonstrates that the system achieves availability either when there are no failures within the interval $[0, t]$, or if failures do occur, they are promptly repaired prior to time t [24]. However, availability can be also studied in a non-transient scenario, where the average probability to have the component functioning is considered. Therefore, by employing the concepts of mean time to failure ($MTTF$) and mean time to repair ($MTTR$), it is possible to express the limiting availability A as follows:

$$A = \lim_{t \longrightarrow \infty} A(t) = \frac{MTTF}{MTTF + MTTR} \tag{3}$$

It is important to note that the limiting availability is solely dependent to the values of $MTTF$ and $MTTR$, irrespective of the specific probability distributions governing the failure as well as repair times.

The assessment of availability in multi-server systems has gained significant importance in the relevant literature. Much information on the subject can be found in research articles, books, and reviews [25]. Markov chain-based availability models are commonly employed in numerous studies, alongside other examples, to analyze and assess system availability.

The work in [26] presents an analytical approach to the availability of healthcare IoT infrastructures. The authors employ two-dimensional CTMCs to depict the availability of the considered IoT systems for healthcare infrastructure and its end-nodes. Additionally, they present an approach that uses Markov models to examine attacks targeting the potentially vulnerable aspects of healthcare IoT systems. The model includes a state diagram representing attacks on the IoT infrastructures in healthcare. The

authors analyze the system's availability with respect to the flow intensities of service requests, emphasizing safety concerns and security-related issues in the healthcare IoT context. Similarly in [27], an investigation is conducted on the availability of healthcare IoT systems. The authors outline two categories of structures comprising the IoT system, utilizing separate two-dimensional Markov state-space models for the systems considered. They subsequently solve the equilibrium equations of the system employing a similar approach to the one described in the preceding article. The authors present various performance metrics related to availability, such as the probability of being able to provide service at full capacity, reduced performance service, and the probability of having the system in a state which is not providing any services at all. In addition, the work presented in [28] showcases the performability evaluation of a smart hospital architecture to guarantee the quality of service in healthcare. The model employs two Stochastic Petri Nets, allowing the tuning of several parameters to adjust different scenarios and identify the most critical components of the architecture. The authors also present some results based on three possible scenarios, where a best-case result is obtained in the scenario where redundancy is implemented.

Several studies have focused on analyzing the availability of IoT systems, including [29,30]. Additionally, some studies such as [31] have modeled the facilitating infrastructures in presence of failures. In [29], the authors present analytical models to evaluate the availability considering various physical edge as well as fog nodes used in various applications. They compute $MTTF$ and $MTTR$ values for the systems considered and present a two-dimensional model that includes both failures and repairs. Similarly, in [30], in addition to availability and performance, the authors also evaluate the energy consumption-related measures of clustered IoT systems. They solve two-dimensional models for steady-state probabilities, which they use to compute crucial measures related to availability (e.g. the probability of being in a state where the system is fully operational) and assess other measures of performance such as the mean value of energy consumption.

Considering the modeling of cloud infrastructures, especially those based on Infrastructure as a Service (IaaS), scalability becomes a significant limiting factor. In the publication by Ataie et al. [32], scalability challenges are addressed through the utilization of approximate Stochastic Reward Net (SRN) models, combined with folding and fixed-point iteration techniques. They use various failure and repair rates in their approach accurately capturing the characteristics of failures as well as repairs for physical machines in order to enable the analysis of availability-related functionalities.

The article by Longo et al. [33] concentrates on attaining high availability in IaaS cloud systems. To expedite the analysis and resolution process, the authors employ an approach based on interacting Markov chains. They employ SRNs to compute the metrics of the Markov chains. The study includes a trade-off analysis between longer $MTTF$ and faster $MTTR$ in terms of system availability. Additionally, the impact of incorporating multiple concurrent facilities for repair facilities is examined.

In [34], a novel approach with an approximate solution is introduced to address the potentially large numbers of servers in cloud based systems. The analytical models and solutions proposed in the study are comprehensive, yet they are capable of handling substantial numbers of nodes, ranging from hundreds to thousands. The research focuses on the quality of service provided by cloud centers, taking into account both server availability and performability metrics considering server failures as well as repairs. Notably, this study distinguishes itself from other reviewed works by emphasizing the ability to analyze and model large-scale cloud systems while incorporating considerations for server availability and quality of service.

In [35], the focus is on blockchain-based systems that can provide service over cloud infrastructures. The research introduces models to assess the availability and capacity-oriented availability of cloud computing infrastructures that host distributed applications using the Ethereum blockchain platform. The conventional approach is employed to represent the system's availability by considering the ratio of $MTTF$ to $MTTR$. The availability outcomes are depicted as functions of $MTTF$ and $MTTR$ for servers as well as miner and bootnodes.

The aforementioned studies employ analytical models to evaluate the availability of various distributed systems. Similar to these studies, in this study as well the approach presented assumes that the time between failures and repair times adheres to an exponential distribution. The main focus of this article is to analyze the availability of a system through the utilization of a Markov chain-based model. When the existing studies in the field are investigated, we see that many studies have employed Markov processes as a common formalism and terminology for modeling various systems. However, based on the authors' knowledge, they are the first to utilize a Markov formalism specifically for modeling the availability of BFT systems, as previously introduced in a previous work [23]. By introducing a Markov-based approach, the authors aim to provide a novel perspective and contribution to the analysis of BFT system availability. This approach allows for a systematic examination of the system's behavior and performance in terms of availability, considering the unique characteristics and challenges associated with BFT systems. The utilization of a Markov formalism in the context of BFT system availability modeling distinguishes this work from previous studies and emphasizes its novelty and potential impact in understanding and evaluating the availability of such systems.

3. System model

The system is designed as a network comprising N nodes, with the responsibility of collaborating to get an agreement on specific tasks. These nodes aim to reach a consensus on those tasks through message exchange among themselves. The communication can occur in any way suitable for the applications, such as end-to-end, as shown in Fig. 1. In the diagram, the white pawns symbolize non-Byzantine or honest (h) nodes while the black pawns represent Byzantine, or faulty (f) nodes.

In this study, the IoT system analyzed utilizing the proposed analytical model is assumed to have N nodes composing the network of devices and servers deputed to exchange messages to determine whether these messages have to be committed or not. Since, the single point of failure is avoided, it is not essential to assume a hierarchical structure in the network. In this context, the model presented in generic levels does not distinguish between IoT devices and servers (i.e., edge or cloud gateways). All the participants in

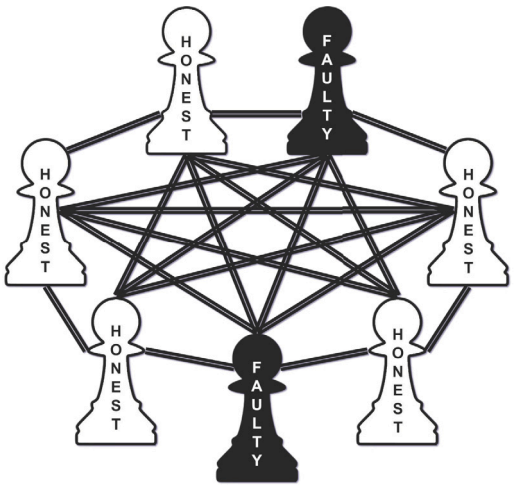


Fig. 1. BFT system in which $N = 7$, $h = 5$, and $f = 2$.

the consensus process are interchangeably called nodes, servers, or devices. Since the described layout is common to various systems, we used a generic multi-server structure focusing on establishing the quorum in the presence of failures and potentially malicious users. For instance, this scenario is generally seen in distributed databases and blockchain networks. However, as a practical example, we present and discuss, in Section 5, a probable implementation of an IoT system in the context of IoT-Blockchain applications [36].

The availability model being used is a quasi-birth–death process based on a CTMC. In this type of stochastic process, the random variables have an exponential distribution, and the system can switch between states at rates specified by the stochastic transition matrix. The Markov property is satisfied, meaning that the distribution of the state probabilities for future states depends solely on the present state and not on past states.

In this model, the parameters ξ and η represent the breakdown and repair rates respectively, as illustrated in Fig. 2. In this model, node breakdowns are independent. In an event of a failure, the broken nodes are considered for repair one at a time. Consequently, the breakdown rate ξ is scaled considering the number of available nodes f. In other words, $f\xi$ is the break-down rate for the state where there are f nodes, and ξ is the corresponding break-down rate when only one node is available. However, the repair process can only occur for one node assuming a single repair facility with a repair rate of η.

The model Fig. 2 is proposed with the following assumptions[1]: there are N nodes in the system. $h \leq N$ of these nodes are honest nodes taking part in the network operations as expected. $f \leq N$ of the nodes are malevolent nodes. In this context, $H : \{h \in \mathbb{N}_0 \mid h \leq N]\} \to \{h \in \mathbb{N}_0 \mid h \leq N]\}$ and $F : \{f \in \mathbb{N}_0 \mid f \leq N]\} \to \{f \in \mathbb{N}_0 \mid f \leq N]\}$ can be treated as random variables following an arbitrary distribution. H and F are dependent on each other, such that their realizations sum up to N, i.e. $H(h) + F(f) = h + f = N$. Therefore, since N is considered to be a constant, H and F are dependent random variables and their outcomes can be written as $f = N - h$ or $h = N - f$, considering either H or F to be the independent random variable. Without loss of generality, F is the independent discrete random variable. Hence, $h = N - f$ is a realization dependent on the value of the discrete random variable F. The reasons behind this abstraction are presented in Section 1.

The state diagram shown in Fig. 2 consists of $(h + 1)(f + 1)$ states. It is worth noting that every state in the chain can be reached from any initial state, demonstrating the chain's property of being both irreducible and ergodic. These two conditions are adequate for the chain to possess a stationary distribution. This means that, given a system with specific parameters, it is possible to compute the system's limiting availability using the stationary probability distribution associated with its CTMC. To calculate the stationary distribution of the chain, it is necessary to formulate the generating equations as a system of linear equations where the state probabilities for given transition rates can be determined. Such a system of simultaneous equations is known as Kolmogorov equations. The following equation represents the whole system of linear equations:

$$\begin{aligned}&\left[(2 - \delta_{ih} - \delta_{jf})\eta + (i + j)\xi\right] P_{i,j} + \\ &\quad - \eta \left[P_{i,j-1}(1 - \delta_{j0}) + P_{i-1,j}(1 - \delta_{i0})\right] + \\ &\quad - (i + 1)\xi P_{i+1,j}(1 - \delta_{ih}) - (j + 1)\xi P_{i,j+1}(1 - \delta_{jf}) = 0,\end{aligned} \tag{4}$$

where δ_{ij} indicates the Kronecker delta, i.e. $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$. In a compact form, Eq. (4) describes all the possible equations in the system by varying the indices i and j, where $i \in [0, h]$ and $j \in [0, f]$. Thus, consistently with number of possible

[1] To facilitate clarity in presentation, we assume that $N, h, f \in \mathbb{N}_0$. Consequently, when performing divisions, the *ceiling* $\lceil\cdot\rceil$ and *floor* $\lfloor\cdot\rfloor$ functions are implicitly applied as appropriate.

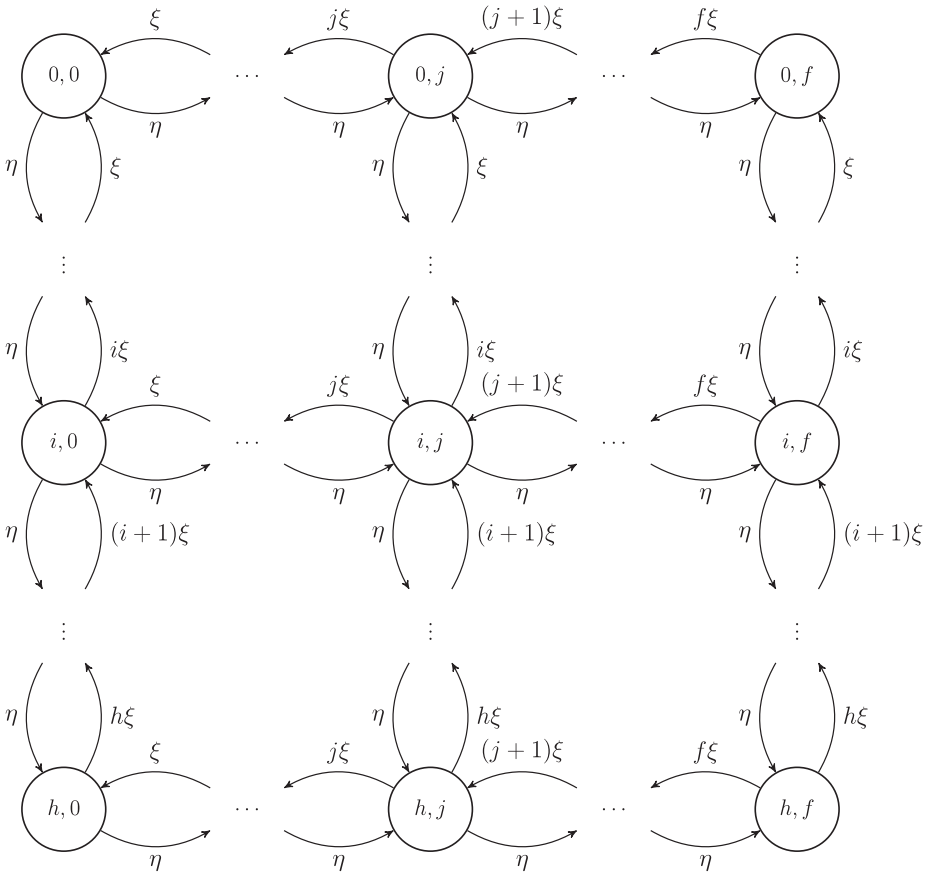


Fig. 2. A representation of the availability model for a BFT consensus protocol.

states, there are $(h+1)(f+1)$ equations to be solved simultaneously. However, because the elements of $\vec{P}$, $P_{i,j}$, are probabilities, the additional condition $\sum_i \sum_j P_{i,j} = 1$ is imposed.

In Eq. (4), $P_{i,j}$ is the probability that the system is in state (i, j), while the coefficients of $P_{i,j}$s are the entries in a coefficient matrix, $\mathbf{Q}$. Indeed, $\mathbf{Q}$ represents the stochastic transition matrix for CTMC under consideration, encompassing the transition rates from one state to another.

Note that, mirroring the lattice structure of the model, it seems natural to write the elements of $\vec{P}$ using the indices i and j, although $\dim \mathbf{Q} = (h+1)(f+1) \times (h+1)(f+1)$, because there are $(h+1)(f+1)$ states in the system. This means that it is not proper to use i and j while computing the elements of $\vec{P}$. Indeed, $\vec{P}$ is having a matrix structure when expressed as $P_{i,j}$, therefore it has to be flattened into a vector with elements P_i, where $i \in [0, (h+1)(f+1)]$. This last remark is important, because it ensures that the dimensions of $\vec{P}$ and $\mathbf{Q}$ are matching, since the matrix of coefficients $\mathbf{Q}$ has indices $i, j \in [0, (h+1)(f+1)]$.

The system of simultaneous equations derived from Eq. (4) can be expressed straightforwardly in the form $\mathbf{Q}\vec{P} = 0$, where $\mathbf{Q}$ represents the coefficient matrix, $\vec{P}$ denotes the vector of unknowns, and 0 corresponds to the vector of constants (zero). To solve homogeneous matrix equations there are a plethora of techniques, even though some of them might not be applied in this context. For instance, since $\mathbf{Q}$ is, indeed, a singular matrix, commonly used methods of linear algebra to solve matrix equation, such as LU decomposition or Gauss elimination, cannot be applied for singular matrices. It is assumed that a non-trivial solution to the system of simultaneous linear equations exists. Notably, matrix $\mathbf{Q}$ possesses at least one singular value equal to zero, indicating the presence of a non-trivial solution to the linear simultaneous equations. To address this, the Singular Value Decomposition (SVD) method is employed.

4. Availability analysis

In [23], a stringent premise regarding the occurrence of Byzantine nodes is adopted. Specifically, it is assumed that the threat level due to Byzantine nodes is either low, medium, or high, with a different number f for each of the three levels. While this is a pragmatic assumption, it may not, in principle, reflect the statistics of a real implementation. From this observation, it is clear that the analysis of the availability may be influenced by the method used to describe the occurrences of Byzantine nodes. Therefore, a new paradigm is presented, in which the number of Byzantine nodes f is determined by the random variable F.

Note that the study of the stochastic properties associated with the distribution of the number of Byzantine nodes in the network is not affecting the analytical model used to describe the system. In other words, a separate layer of abstraction is added on top of the availability model, in order to analyze the system in a more general way. This additional abstraction can be considered as an experimental framework, in which the experimenter/decision-maker is testing the system. Given the system parameters (N, η, ξ) and a probability distribution $Pr(F = f) = p(f)$, the methodology to observe is composed by the steps reported in Algorithm 1.

Algorithm 1 Pseudo-code to calculate availability with f as random variable

Require: $N, N_{max} \geq 4$ and $\xi, \eta > 0$ and $\xi/\eta \ll 1$
  for $N \leq N_{max}$ **do**
    $f \leftarrow 0$
    while $f < N/3$ **do**
      $h \leftarrow N - f$
      $\mathbf{Q} \leftarrow \mathbf{Q}(N, f, h, \xi, \eta)$
      $\vec{P} \leftarrow SVD(\mathbf{Q}, 0)$ ▷ compute state probabilities through SVD
      $A_{h,f} \leftarrow \sum_{i>2N/3}^{h} \sum_{j=0}^{f} P_{i,j}$ ▷ availability
      $f \leftarrow f + 1$
    end while
    $\bar{A} \leftarrow \sum_{h,f} p(f)\, A_{h,f}$ ▷ mean availability
  end for

The process described in Algorithm 1 requires a maximum number of nodes to be considered $N_{max} \geq 4$ and rates $\xi, \eta > 0$ such that $\xi/\eta \ll 1$. The process starts setting the value $f = 0$, hence imposing $h = N - f$. The matrix $\mathbf{Q}$ is determined using Eq. (4), thus $\vec{P}$ can be computed through Singular Value Decomposition. For the pair (h, f), the probabilities $P_{i,j}$ are computed and in turn arranged in matrix $\mathbf{P}$ (vector $\vec{P}$ is transformed into matrix $\mathbf{P}$ to align with the two-dimensional structure depicted in Fig. 2.). Finally, for the resulting set (N, f, h, η, ξ), availability can be calculated. Availability refers to the cumulative probability that the system is operational and capable of committing messages. As prescribed in Eq. (1), the system is available for all the states with $i > 2N/3$, consequently, the corresponding state probabilities are aggregated to calculate the availability.:

$$A_{h,f} = \sum_{i>\frac{2N}{3}}^{h} \sum_{j=0}^{f} P_{i,j}. \tag{5}$$

At each iteration, f is increased by 1. The algorithm iterates until a predefined value of N_{max} is reached. After the iterative part, there is a resulting collection of $A_{h,f}$s, one for each (N, f, h, η, ξ). Therefore, the mean value of the availability is

$$\bar{A} = \sum_{h,f} p(f)\, A_{h,f}. \tag{6}$$

Essentially, the procedure described above compute the mean availability for a system with N nodes (subjected to break-down and repair processes at rate ξ and η), where the number of Byzantine actors in the system, f, is deriving from the realizations of a random variable F distributed according to an arbitrary probability distribution (see Table 1 for a concise description of the probability distributions used to determine f). This means that the procedure in Algorithm 1 can be iterated over a range of several N and different probability distributions for F. In this way, the behavior of the system's availability, for different distributions, can be studied as a function of the number of nodes. Similarly, to study the relationship between rates ξ, η and availability, the probability distribution for $Pr(F = f)$ can be fixed and then it can be computed the availability of the system at the variation of ξ and η, for different N.

A special attention should be reserved to the analysis of the Poisson distribution. The pmf of Poisson distribution is defined on the positive integers, therefore a truncated version of the pmf is needed to match the domain of definition $[0, N]$ for the occurrence of Byzantine nodes. The right-truncated Poisson distribution [37] is defined as

$$p(x; \lambda, N) = \begin{cases} \frac{\lambda^x}{x!} \left(\sum_{y=0}^{N} \frac{\lambda^y}{y!} \right)^{-1}, & x = 0, 1, \ldots, N \\ 0, & \text{otherwise} \end{cases}$$

that is derived from the definition of Poisson distribution, in which the series representation of the exponential is truncated to N. The mean can be computed from the definition of expected value $\mu = E[X] = \sum_{x=0}^{N} x\, p(x) = \lambda \frac{N\, \Gamma(N,\lambda)}{\Gamma(N+1,\lambda)}$, where $\Gamma(N, \lambda)$ is the incomplete gamma function.

Regarding the proposed methodology, note that it is vital to choose appropriately the parameters of the arbitrary probability function generating the random values f. While it is out of the scope for this study to determine whether there is an *a priori* restriction on which probability function to use in characterizing the occurrences of Byzantine nodes, it is advisable to properly select the first two moments, i.e. mean and variance, of any chosen distribution. To better explain this, consider the impact that the parameters of the probability distribution have: if the mean μ is outside the interval $[0, N/3)$ and the probability function is narrow (low variance), several zero-valued availability numbers will be sampled; same situation would occur if $\mu \in [0, N/3)$, but the variance is high; an optimal choice, instead, is represented by the distribution not spreading excessively and $\mu \in [0, N/3)$.

Table 1
A summary of the probability distributions used in this work to characterize the occurrences of Byzantine nodes, with $N \in [4, 128]$. pmf indicates probability mass function, μ the mean of each distribution, and σ^2 the variance of the distribution. Parameters for each distributions are specified in the next section, in correspondence of the two comparative results: Figs. 3 and 4.

Distribution	pmf	Mean μ	Variance σ^2
Uniform	$p(x; a, b) = \frac{1}{b-a+1}$	$\frac{b+a}{2}$	$\frac{(b-a+1)^2-1}{12}$
Right-truncated Poisson	$p(x; \lambda, n) = \frac{\lambda^x}{x!}\left(\sum_{y=0}^{n} \frac{\lambda^y}{y!}\right)^{-1}$	$\lambda \frac{n\,\Gamma(n,\lambda)}{\Gamma(n+1,\lambda)}$	–
Binomial	$p(x; n, q) = \binom{n}{x} q^x (1-q)^{n-x}$	nq	$nq(1-q)$
Degenerate	$p(x; x_0) = \delta_{x\,x_0}$	x_0	0

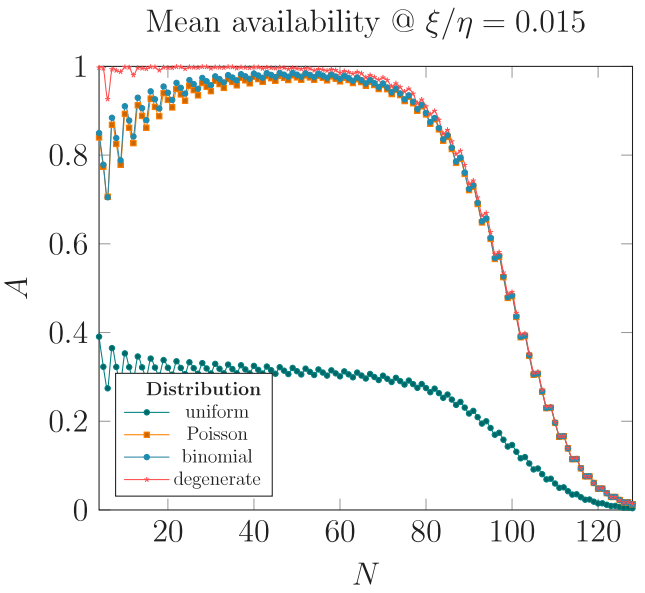


Fig. 3. Availability as a function of the number of nodes and fixed ratio ξ/η.

Lastly, this methodology recreates the results presented in our previous study [23], where a constant number f is selected to reflect a threat level due to the ratio of Byzantine nodes in the system. This validates the observation that, when defining some possible threat levels of the system, the decision-maker is, indeed, assuming a degenerate distribution for F, i.e. a constant value f representing the number of Byzantine nodes in a system of N nodes.

5. Results and discussions

In this section, results are provided to show the effects of various distributions of the Byzantine faults on availability.

Fig. 3, shows the effects of four different probability distributions for the random variable F (as from Table 1), on mean system availability for $N \in [4, 128]$ and $\xi/\eta = 0.015$. Different lines represent a separate choice of a probability distribution for the value of f. Parameters of the uniform distribution are $a = 0$ and $b = N$. $\lambda = N/6$ is used for the right-truncated Poisson distribution, in which $\Gamma(n, \lambda)$ is the incomplete gamma function. For the binomial distribution $n = N$ and $q = 1/6$, where $\binom{n}{x}$ is the binomial coefficient. Lastly, the degenerate distribution uses the Kronecker delta $\delta_{x\,x_0}$, with $x_0 = N/6$. In the figure, the uniform distribution has the mean $\mu = N/2$, while all the other distributions have the mean $\mu = N/6$, the center of the interval $[0, N/3)$. The figure shows that, for different choices of the probability distribution of F, there is a distinctive behavior of the mean availability. This behavior varies between the worst-case scenario, which can be observed when the random variable F is drawn from a uniform distribution to the best case, where a degenerate distribution with constant value $f = N/6$ is used. However, this configuration for the uniform distribution is expected to give the worst-case scenario, since the mean of the distribution is centered around the middle of the interval for the values of N, while the other distributions are centered around $N/6$.

Fig. 4 presents the behavior of the mean system availability for $N \in [4, 128]$ and $\xi/\eta = 0.015$, when the mean of each probability distribution is $\mu = N/2$. In this figure, different lines represent a separate choice of probability distribution for the value of f. Parameters of the uniform distribution are $a = 0$ and $b = N$. For the right-truncated Poisson distribution, in which $\Gamma(n, \lambda)$ is the incomplete gamma function, $\lambda = N/2$ is used. The binomial distribution has $n = N$ and $q = 1/2$, where $\binom{n}{x}$ is the binomial coefficient. As expected, the degenerate distribution, when $f = N/2$, gives availability that is constantly zero, therefore it is not reported. In

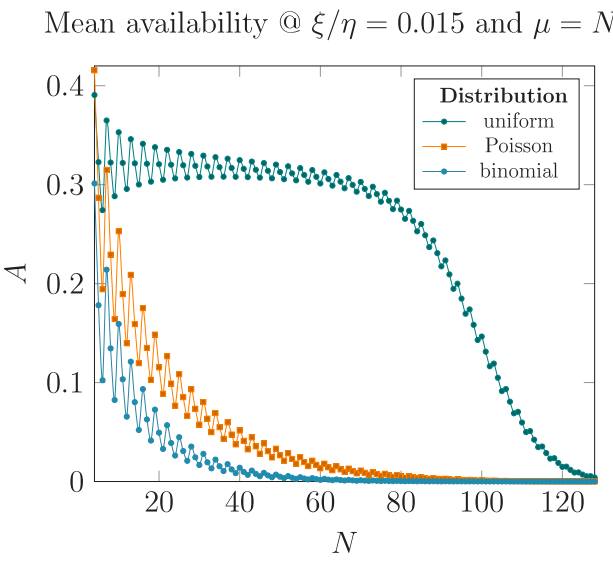


Fig. 4. The variation in system availability as a function of the number of nodes and fixed ratio ξ/η.

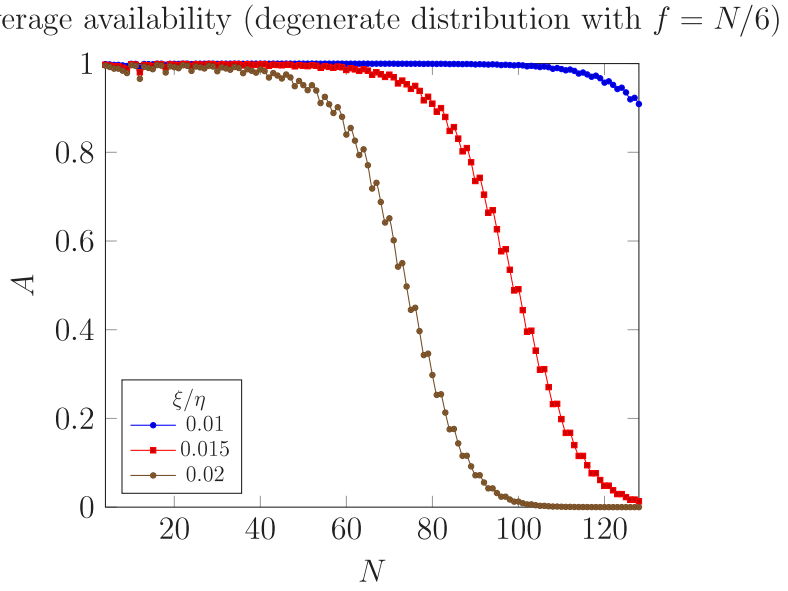


Fig. 5. System availability as a function of the number of nodes with ratio $\xi/\eta = 0.01, 0.015, 0.02$.

this graph, the best case is the one in which the uniform distribution is employed, while the worst case occurs when the binomial distribution describes the occurrence of Byzantine nodes in the system. Differently from Fig. 3, with this configuration, the uniform distribution is clearly the distribution giving the best result in Fig. 4. This is because the probability to get a value $f < N/3$, such that the quorum is reached, is higher for the uniform distribution than for the other distributions. This is simply because, while the mean is the same for the selected distributions, the variance of the possible values of f is larger for the uniform distribution, hence there is a higher probability to select a value f satisfying the quorum.

Fig. 5 presents an example of system availability trend for different ratios of ξ/η. Here F is distributed according to the degenerate distribution centered around the value $f = N/6$. The plot shows how the availability of the system is degrading when the ratio ξ/η is increasing.

Please note that the values of availability are not represented by a smooth line because some numbers for N correspond to optimal configurations of BFT systems. For instance, any N satisfying the equation $(N \mod 3) = 1$, $N \geq 4$, produces a system with better availability than the ones generated by $N - 1$ and $N - 2$, e.g., the value of availability when $N = 16$ is higher than when $N = 15$ or $N = 14$.

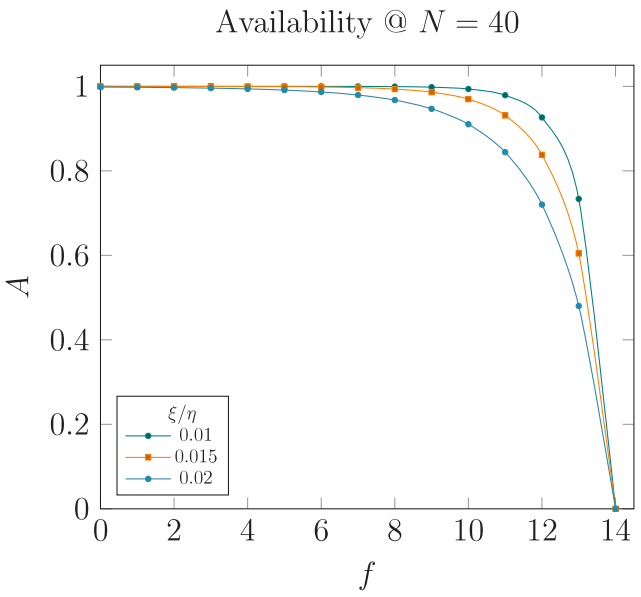


Fig. 6. Availability depending on the number of Byzantine nodes at different ratios ξ/η.

Fig. 6 shows the availability of an IoT system as a function of the number of Byzantine nodes (f) in the network. The total number of nodes in the system is constant. $N = 40$ is assumed similar to the study presented in [36]. Indeed, in [36] a study is presented where there is a comparison between an implementation of the Practical BFT (PBFT) protocol [38] and a novel BFT-based protocol, G-PBFT. The G-PBFT protocol works on the assumption that most IoT-blockchain applications rely on geographically fixed IoT devices (called endorsers) for data collection and processing, unlike mobile devices such as smartphones and sensors. This way, it is easier to safely select a subset of the available devices present in the IoT environment. The authors consider $N = 40$ as the maximum value of endorsers.

Fig. 6 shows that, if the number of trustworthy and reliable nodes decreases, the availability of the IoT system deteriorates non-linearly, until it reaches abruptly zero when $f > N/3$ (system has non-zero availability for $0 \leq f \leq 13$). This behavior is even more accentuated for increasing values of ξ/η. The maximum availability, when $\xi/\eta = 0.1$, is greater than 0.999999 (i.e. 31.56 s of downtime per year), while it becomes less than 0.999 (i.e. 8.77 h of downtime per year) when $\xi/\eta = 0.2$.

In summary, from this study, it can be concluded that system availability is indeed non-linearly dependent on the number of the nodes in the network. This relation is inversely proportional to the number of the nodes. Moreover, results show that the occurrence of Byzantine nodes in the system effects the overall availability, especially with regards to the probability distribution describing this phenomenon and its parameters. Finally, the ratio between break-down rate and repair rate, likewise, regulates the value of availability for the system, with lower values at the increase of the ratio ξ/η.

6. Conclusion and future work

Distributed systems are widely used in various engineering sectors, industrial production, data analysis and management, cryptocurrencies, and more. Ensuring fault tolerance and security against malicious attacks has become increasingly important, particularly in scenarios where high availability is crucial. The uninterrupted operation of vital applications requires a well-designed distributed system capable of handling various potential scenarios.

BFT protocols play a significant role in achieving fault tolerance. These protocols were developed to model consistent distributed computer networks and parallel computing. BFT systems can maintain resilience even when malicious actors are involved in pursuing a common goal. Computer networks, including DLTs, like blockchains, often employ BFT algorithms to ensure continuous system operation.

We present an analytical availability model designed to assess fault-tolerant multi-node systems. The model leverages CTMCs to analyze the availability of BFT systems, taking into account breakdowns, repairs, and the presence of malicious nodes. The analysis considers a range of total nodes, N, from 4 to 128, and incorporates various probability distributions to model the proportion of malicious nodes. The numerical results exhibit the relationship between availability and the number of participants, as well as the relative number of honest actors, utilizing various probability distributions that represent the number of malicious nodes.

This work makes a significant contribution by expanding the availability modeling to incorporate the existence of malicious nodes with arbitrary non-deterministic probabilistic distributions. The model unveils a non-linear association between the number of nodes and availability, where availability is inversely related to the number of nodes in the system, regardless of the distribution tested. This relationship becomes stronger as the ratio of breakdown rate to repair rate increases.

The model serves as an initial step in the modeling of distributed systems based on BFT consensus protocols. As part of future work, it is desirable to further expand the proposed models, particularly for performability analysis of BFT systems. In other words, analytical models are required to analyze performance metrics such as response time, delay, and throughput in the presence of Byzantine faults.

In the context of BFT systems, performability analysis can help assess the system's ability to handle both normal operation and fault scenarios. BFT systems should aim to provide not only fault tolerance but also desirable performance characteristics. Performability analysis can help system designers and evaluators to understand and quantify the trade-offs between fault tolerance and performance in BFT systems.

The main challenge for these models would be the combination of pure performance models and availability models without causing some well-known problems such as state space explosion.

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

[1] T.M. Fernández-Caramés, P. Fraga-Lamas, A review on the use of blockchain for the Internet of Things, IEEE Access 6 (2018) 32979–33001.
[2] Z. Gao, C. Cecati, S.X. Ding, A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches, IEEE Trans. Ind. Electron. 62 (6) (2015) 3757–3767.
[3] I. Koren, C.M. Krishna, Fault-Tolerant Systems, Morgan Kaufmann, 2020.
[4] M. Baleani, A. Ferrari, L. Mangeruca, A. Sangiovanni-Vincentelli, M. Peri, S. Pezzini, Fault-tolerant platforms for automotive safety-critical applications, in: Proceedings of the 2003 International Conference on Compilers, Architecture and Synthesis for Embedded Systems, 2003, pp. 170–177.
[5] S. Yin, B. Xiao, S.X. Ding, D. Zhou, A review on recent development of spacecraft attitude fault tolerant control system, IEEE Trans. Ind. Electron. 63 (5) (2016) 3311–3320.
[6] C. Edwards, T. Lombaerts, H. Smaili, et al., Fault tolerant flight control, Lecture Notes in Control and Inform. Sci. 399 (2010) 1–560.
[7] A. Bala, I. Chana, Fault tolerance-challenges, techniques and implementation in cloud computing, Int. J. Comput. Sci. Issues (IJCSI) 9 (1) (2012) 288.
[8] Z. Amin, H. Singh, N. Sethi, Review on fault tolerance techniques in cloud computing, Int. J. Comput. Appl. 116 (18) (2015).
[9] R. Jhawar, V. Piuri, Fault tolerance and resilience in cloud computing environments, in: Computer and Information Security Handbook, Elsevier, 2017, pp. 165–181.
[10] P. Kumari, P. Kaur, A survey of fault tolerance in cloud computing, J. King Saud Univ.-Comput. Inf. Sci. 33 (10) (2021) 1159–1176.
[11] F. Cristian, Understanding fault-tolerant distributed systems, Commun. ACM 34 (2) (1991) 56–78.
[12] A. Avizienis, J.-C. Laprie, B. Randell, C. Landwehr, Basic concepts and taxonomy of dependable and secure computing, IEEE Trans. Dependable Secur. Comput. 1 (1) (2004) 11–33.
[13] A. Panarello, N. Tapas, G. Merlino, F. Longo, A. Puliafito, Blockchain and IoT integration: A systematic survey, Sensors 18 (8) (2018) 2575.
[14] A. Boudguiga, N. Bouzerna, L. Granboulan, A. Olivereau, F. Quesnel, A. Roger, R. Sirdey, Towards better availability and accountability for IoT updates by means of a blockchain, in: 2017 IEEE European Symposium on Security and Privacy Workshops, EuroS&PW, IEEE, 2017, pp. 50–58.
[15] K.R. Ozyilmaz, A. Yurdakul, Designing a blockchain-based IoT with Ethereum, swarm, and LoRa: the software solution to create high availability with minimal security risks, IEEE Consum. Electron. Mag. 8 (2) (2019) 28–34.
[16] L. Lamport, R. Shostak, M. Pease, The Byzantine generals problem, ACM Trans. Program. Lang. Syst. 4 (3) (1982) 382–401.
[17] A. Goyal, S.S. Lavenberg, Modeling and analysis of computer system availability, IBM J. Res. Dev. 31 (6) (1987) 651–664.
[18] G. Bolch, S. Greiner, H. De Meer, K.S. Trivedi, Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications, John Wiley & Sons, 2006.
[19] F. Oliveira, P. Pereira, J. Dantas, J. Araujo, P. Maciel, Dependability evaluation of a smart poultry house: Addressing availability issues through the edge, fog, and cloud computing, IEEE Trans. Ind. Inform. (2023).
[20] S.K. Battula, M.M. O'Reilly, S. Garg, J. Montgomery, A generic stochastic model for resource availability in fog computing environments, IEEE Trans. Parallel Distrib. Syst. 32 (4) (2020) 960–974.
[21] F.A. Silva, I. Fé, C. Brito, G. Araújo, L. Feitosa, E. Choi, D. Min, T.A. Nguyen, Supporting availability evaluation of a smart building monitoring system aided by fog computing, Electron. Lett. 58 (12) (2022) 471–473.
[22] J.-F. Pâris, D.D. Long, Reducing the energy footprint of a distributed consensus algorithm, in: 2015 11th European Dependable Computing Conference, EDCC, IEEE, 2015, pp. 198–204.
[23] M. Marcozzi, O. Gemikonakli, E. Gemikonakli, E. Ever, L. Mostarda, Availability model for Byzantine fault-tolerant systems, in: Advanced Information Networking and Applications: Proceedings of the 37th International Conference on Advanced Information Networking and Applications, Vol. 1, (AINA-2023), Springer, 2023, pp. 31–43.
[24] K.S. Trivedi, Probability & Statistics with Reliability, Queuing and Computer Science Applications, John Wiley & Sons, 2008.
[25] K.S. Trivedi, A. Bobbio, Reliability and Availability Engineering: Modeling, Analysis, and Applications, Cambridge University Press, 2017.
[26] A. Strielkina, V. Kharchenko, D. Uzun, Availability models for healthcare IoT systems: Classification and research considering attacks on vulnerabilities, in: 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies, DESSERT, IEEE, 2018, pp. 58–62.
[27] S. Tang, Y. Xie, Availability modeling and performance improving of a healthcare Internet of Things (IoT) system, IoT 2 (2) (2021) 310–325.
[28] L. Rodrigues, I. Gonçalves, I. Fé, P.T. Endo, F.A. Silva, Performance and availability evaluation of an smart hospital architecture, Computing 103 (2021) 2401–2435.
[29] P. Pereira, J. Araujo, C. Melo, V. Santos, P. Maciel, Analytical models for availability evaluation of edge and fog computing nodes, J. Supercomput. 77 (9) (2021) 9905–9933.
[30] E. Ever, P. Shah, L. Mostarda, F. Omondi, O. Gemikonakli, On the performance, availability and energy consumption modelling of clustered IoT systems, Computing 101 (12) (2019) 1935–1970.

[31] Y. Kirsal, E. Ever, A. Kocyigit, O. Gemikonakli, G. Mapp, Modelling and analysis of vertical handover in highly mobile environments, J. Supercomput. 71 (12) (2015) 4352–4380.
[32] E. Ataie, R. Entezari-Maleki, L. Rashidi, K.S. Trivedi, D. Ardagna, A. Movaghar, Hierarchical stochastic models for performance, availability, and power consumption analysis of IaaS clouds, IEEE Trans. Cloud Comput. 7 (4) (2017) 1039–1056.
[33] F. Longo, R. Ghosh, V.K. Naik, K.S. Trivedi, A scalable availability model for infrastructure-as-a-service cloud, in: 2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks, DSN, IEEE, 2011, pp. 335–346.
[34] E. Ever, Performability analysis of cloud computing centers with large numbers of servers, J. Supercomput. 73 (5) (2017) 2130–2156.
[35] C. Melo, J. Dantas, P. Pereira, P. Maciel, Distributed application provisioning over Ethereum-based private and permissioned blockchain: availability modeling, capacity, and costs planning, J. Supercomput. 77 (9) (2021) 9615–9641.
[36] L. Lao, X. Dai, B. Xiao, S. Guo, G-PBFT: A location-based and scalable consensus protocol for IoT-blockchain applications, in: 2020 IEEE International Parallel and Distributed Processing Symposium, IPDPS, IEEE, 2020, pp. 664–673.
[37] N.L. Johnson, A.W. Kemp, S. Kotz, Univariate Discrete Distributions, Vol. 444, John Wiley & Sons, 2005.
[38] M. Castro, B. Liskov, Practical Byzantine fault tolerance, in: OsDI, Vol. 99, 1999, pp. 173–186.