Journal of Systems Architecture 154 (2024) 103216

Contents lists available at ScienceDirect

Journal of Systems Architecture

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

Integrated analysis of reliability, power, and performance for IoT devices and servers

Keqin Li

Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

ARTICLE INFO

Keywords:
Average task response time
Continuous-time Markov chain (CTMC)
IoT servers
Power-performance tradeoff
Queueing system
Reliability
Server availability

ABSTRACT

The Internet of Things (IoT) is currently widely used in various sectors and spaces. IoT devices are becoming small yet powerful servers and perform server-like functions. Reliability is a critical aspect in both IoT devices and servers, as they work together to create a robust and dependable IoT ecosystem. Power and performance are two other major considerations of an IoT system. Modeling, analysis, evaluation, and optimization of reliability, power, and performance for IoT devices and servers are major components in IoT systems development and deployment. In this paper, we conduct an integrated study of reliability, power, and performance for IoT devices and servers by mathematically rigorous modeling and analysis. The contributions of the paper can be summarized as follows. We establish a continuous-time Markov chain (CTMC) model that incorporates server failure rate, server repair rate, task arrival rate, and task processing rate. Using such an analytical model, we can calculate the server availability, the average task response time, and the average power consumption. We point out that there is an optimal server speed that minimizes the power-time product and a combined cost-performance metric of power, performance, and reliability. We show the impact of server reliability on response time, power consumption, server utilization, and the power-performance tradeoff. To the best of the author's knowledge, this is the first paper that takes a combined approach to modeling and analysis of reliability, power, and performance for IoT devices and servers. It has been noticed that there has been little such theoretically solid investigation in the existing literature. Therefore, this paper has made tangible contributions and significant advances in the joint understanding of reliability, power, performance, and their interplay in IoT devices and servers quantitatively and mathematically.

1. Introduction

1.1. Background and motivation

The Internet of Things (IoT) is currently widely used in various sectors and spaces, including consumers (for home automation, smart home, intelligent appliances, and elder assistance), organizations (for medical and healthcare, smart healthcare, transportation systems, smart traffic control, and V2X communications), industries (for manufacturing equipment, digital control, and smart manufacturing), agriculture (for automated farming and fishing), infrastructure (for smart cities, smart buildings, smart grids, energy management, and environmental monitoring), and military (for the purposes of reconnaissance, surveillance, and other combat-related objectives) [1–6].

IoT devices and servers play a crucial role in the functioning of IoT ecosystems. IoT devices (1) are primarily designed to collect data from the physical environment through sensors or other data acquisition methods; (2) have some computing capabilities, e.g., processing data locally for immediate decision-making; (3) can communicate with servers or other devices to transmit data, receive commands, and access resources; (4) are typically built for specific purposes, such as monitoring temperature, controlling lights, or tracking assets. IoT servers (1) are dedicated to storing, processing, and managing data, with greater computational capabilities, memory capacities, storage resources, and communication bandwidth compared to IoT devices; (2) act as hubs for receiving data from multiple IoT devices and sending data to other servers, applications, or end-users; (3) run complex software applications and services that can perform data analytics, machine learning, and other advanced data-intensive computations; (4) are designed for high availability and scalability to handle large volumes of data and respond to requests from numerous devices simultaneously.

While IoT devices and servers have different primary functions, there are cases where IoT devices may perform server-like functions [7, 8]. IoT devices with local storage can temporarily store data before sending it to a server, acting as a local buffer and reducing transmission

E-mail address: lik@newpaltz.edu.

https://doi.org/10.1016/j.sysarc.2024.103216
Received 24 January 2024; Received in revised form 15 June 2024; Accepted 23 June 2024
Available online 28 June 2024

latency. Some IoT devices are equipped with edge computing capabilities, allowing them to perform advanced data processing on the devices themselves and enhancing performance by removing communication cost. In certain IoT architectures, IoT devices may communicate directly with each other (i.e., peer-to-peer communication), sharing computing power and storage capacity without routing all data to a central server. Some IoT gateway devices have more substantial processing power and memory resources and can aggregate data from multiple IoT devices before transmitting it to an edge or cloud server.

With increasing application demand and technology development, IoT devices (including sensors, actuators, gadgets, appliances, and machines) are able to collect, store, process, and transmit data over wireless and wired networks. They can be embedded into other mobile devices, industrial equipment, environmental monitors, medical devices, and so on. IoT devices are more and more powerful by integrating artificial intelligence and machine learning to bring stronger capabilities and autonomy to systems and processes, such as home automation, autonomous driving, medical equipment, and industrial manufacturing. Increasing data amount, network bandwidth, and consumer expectations for data privacy and user experience continue to demand more on-device processing, where data are stored and processed on IoT devices, rather than cloud servers. Therefore, IoT devices are becoming small yet powerful servers, not clients. (For convenience, we will use the terms "devices" and "servers" interchangeably in this paper.)

Reliability is a critical aspect in both IoT devices and servers, as they work together to create a robust and dependable IoT ecosystem [9–14]. IoT devices are typically small, power-constrained, and cost-constrained microprocessor-based and microcontroller-based systems. IoT devices are often deployed in harsh or remote environments, where they have to operate with minimal human intervention, low power consumption, high performance, and high availability under unpredictable temperature, humidity, rain, light, wind, noise, vibration, shock, dust, soil, pest, etc. IoT devices and servers must have (1) reliable hardware components that have a long lifespan and work consistently in various environmental conditions to maximize availability and to minimize downtime; (2) energy-efficient battery-powered devices with redundant power supply (batteries or other sources) to ensure continuous operation; (3) reliable and resilient data storage systems with redundancy, backups, and fault-tolerant storage solutions; (4) reliable communication channels to ensure continuous and consistent data transmission; (5) robust software free of critical bugs or severe defects with fault tolerance and error handling mechanisms; (6) strong security to protect data and services from unauthorized access, data breaches, malicious attacks, and cyberattacks, with robust security measures such as encryption and authentication; (7) regular monitoring and maintenance for prompt detecting and addressing hardware failures, software errors, and security vulnerabilities.

Power and performance are two other major considerations of an IoT system. First, IoT devices are typically battery-powered with a very limited battery lifetime. IoT devices should be energy-efficient to extend their operational life between battery replacements or recharges. Thus, energy efficiency is a critical and crucial concern for IoT devices and servers to provide sustainable services [15–18]. Second, big data computing requires fast task processing speed and short task response time in an IoT environment with a huge volume of data generation. Hence, high performance is an important and significant concern for IoT servers to handle both computation- and communication-intensive tasks [19–23]. However, faster computing and communicating speeds imply higher power consumption and energy supply. It is clear that the tradeoff between power and performance is a central topic for IoT systems, as in all distributed computing systems. Optimal server speed setting is usually an effective method to deal with the power-performance tradeoff. With the added factor of reliability, the interplay among reliability, power, and performance is even more complicated and challenging.

Modeling, analysis, evaluation, and optimization of reliability, power, and performance for IoT devices and servers are major components in IoT systems development and deployment. It is very important to understand the impact of server reliability on performance, power, server utilization, and the power-performance tradeoff of an IoT system. Unfortunately, there is a lack of serious and systematic study which takes a combined and comprehensive approach to addressing the above pressing issues. The motivation of this paper is to make efforts towards this direction.

1.2. New contributions

In this paper, we conduct an integrated study of reliability, power, and performance for IoT devices and servers by mathematically rigorous modeling and analysis. The contributions of the paper can be summarized as follows.

- We establish a continuous-time Markov chain (CTMC) model that incorporates server failure rate, server repair rate, task arrival rate, and task processing rate.
- Using such an analytical model, we can calculate the server availability, the average task response time, and the average power consumption.
- We point out that there is an optimal server speed that minimizes the power-time product and a combined cost-performance metric of power, performance, and reliability.
- We show the impact of server reliability on response time, power consumption, server utilization, and the power-performance tradeoff.

To the best of the author's knowledge, this is the first paper that takes a combined approach to modeling and analysis of reliability, power, and performance for IoT devices and servers. It has been noticed that there has been little such theoretically solid investigation in the existing literature. Therefore, this paper has made tangible contributions and significant advances in the joint understanding of reliability, power, performance, and their interplay in IoT devices and servers quantitatively and mathematically.

We would like to mention that our model and method only focus on those system properties that can be quantitatively defined and described. Other features such as security and privacy that cannot be quantitatively characterized using a queueing model are not included in our study.

The paper is organized as follows. In Section 2, we describe our IoT service system and a CTMC model. In Section 3, we calculate various reliability, performance, and power metrics. In Section 4, we present numerical data and examples. In Section 5, we review related work. In Section 6, we give a summary of the paper.

2. System and model

In this section, we describe our IoT service system and a CTMC model. Table 1 provides a summary of all notations and their definitions used in this paper.

2.1. An IoT service system

An IoT service system is specified by a variant of the M/M/m/K queueing system with server failure and repair and limited capacity ([24], pp. 103–105) due to the limited memory storage of IoT servers, i.e., there is a maximum number of tasks allowed to be in the system (see Fig. 1). (Notice that queueing systems and networks have been widely and extensively used in system modeling and performance evaluation of various computer systems [25]. *Notation*. M: exponential distribution of task inter-arrival times; M: exponential distribution of task execution times; m: multiple servers; K: finite waiting queue.)

Table 1
Notations and definitions.

Notation	Definition
M	the number of IoT servers
K	the capacity of an IoT service system
λ	the task arrival rate
μ	the service rate of an IoT server
α	the server failure rate
β	the server repairment rate
R	the reliability of an IoT device
(m, k)	a state of the CTMC model
$p(m, k)$	the probability of state (m, k)
A, A^*	the average number of available servers
F, F^*	the average number of failed servers
q_m	the probability of state m
N	the average number of tasks in the queueing system
Q	the average number of tasks in the waiting queue
D	the probability that a newly arriving task is discarded
λ_{eff}	the effective task arrival rate
T	the average task response time
W	the average waiting time
B	the average number of busy servers
U	the server utilization
P_d, P_s	dynamic and static components of power consumption
ξ, d	technology dependent constants of the power consumption model
P	the average power consumption

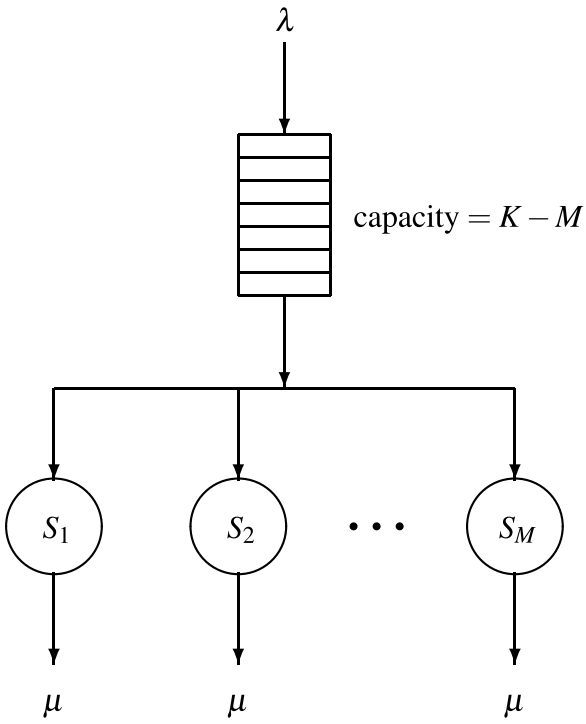

Fig. 1. An M/M/m/K queueing model for an IoT service system.

The queueing system has M identical IoT servers: $S_1, S_2, \ldots, S_M$. There is a stream of tasks generated according to a Poisson process with arrival rate λ (measured by the number of tasks per second). The service rate of an IoT server is μ (measured by the number of tasks per second). There is a task waiting queue adopting the first-come-first-served (FCFS) discipline. The total capacity of the IoT service system is K, i.e., there can be at most K tasks in the system and any further arriving tasks will be dropped out immediately without service.

It is worth mentioning that in reality, the above M/M/m/K queueing system can actually be implemented in a distributed fashion as follows. Each server S_i is an M/M/1/K_i queueing system with arrival rate λ_i and its own FCFS waiting queue of capacity K_i. When S_i reaches its capacity K_i, a newly arrived task is routed to another server. When S_i completes all its tasks (i.e., its waiting queue is empty), it can request a task from another server to process. Hence, the aggregation of the M distributed M/M/1/K_i queueing systems looks like a single M/M/m/K queueing system that has a combined task stream with arrival rate $\lambda = \lambda_1 + \lambda_2 + \cdots + \lambda_M$ and one unified and centralized waiting queue of capacity $K = K_1 + K_2 + \cdots + K_M$. Task transfer among servers can be implemented by appropriate coordination among the servers.

Of course, the above virtual waiting queue can also be implemented by a dedicated management device which collects all tasks and dispatches tasks to servers for processing. However, such a centralized mechanism significantly increases network traffic and consumes communication bandwidth.

IoT devices and servers have the potential to malfunction. The key reliability considerations for well-architected IoT solutions are how quickly you can detect failures and how quickly you can resume operations.

Assume that the lifetime of an IoT server is an exponential random variable with parameter α, i.e., the *mean time to failure* (MTTF) is $1/\alpha$. Assume that the time to repair/restore/replace a failed IoT server is an exponential random variable with parameter β, i.e, the *mean time to repair* (MTTR) is $1/\beta$.

The reliability of a server can be defined as the probability that the server can operate without failure for a certain amount of time [26]. Such a definition includes a time parameter. The definition of reliability for multiple servers with failure and repair is subtle since we do not expect all servers to operate simultaneously for a certain amount of time. Furthermore, we should take not only server failure but also server repairment into consideration. In this paper, the *reliability* of an IoT server is the probability that it is functioning at a random time, i.e.,

$$R = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}} = \frac{1/\alpha}{1/\alpha + 1/\beta} = \frac{\beta}{\alpha + \beta}, \tag{1}$$

which is the percentage of time when it is running, i.e., *availability*. While definitions may vary, in this paper, the reliability and the availability of an IoT server are both defined by Eq. (1). For multiple servers with failure and repair, we are interested in the percentage of functioning servers. Such a quantity needs careful analysis (which is the main topic of this paper) because it takes server repairment times into account and server repairment times can be independent or correlated (i.e., there are repair waiting times).

We consider two repairment models. The difference between the two models is whether multiple failed servers are repaired simultaneously or sequentially, depending on the system maintenance capabilities and capacities [27–29].

- Parallel repairment model – If there are multiple failed servers, they are repaired at the same time (simultaneously). Server repairment times are independent of each other and there is no repair waiting time.
- Sequential repairment model – If there are multiple failed servers, they are repaired one after another (sequentially). Server repairment times are related and correlated since there are repair waiting times.

To summarize, our IoT service system is characterized by a six-tuple: $(M, K, \lambda, \mu, \alpha, \beta)$. In a real IoT service system, these parameters can be easily collected by real measurement of task execution and system management.

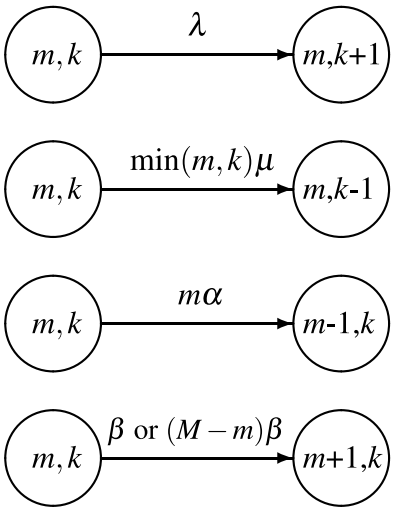

Fig. 2. Transition rates.

2.2. The CTMC model

The M/M/m/K queueing system with possible server failure can be analytically described by a continuous-time Markov chain (CTMC) model. The model includes $(M+1)(K+1)$ states. Each state is specified by (m,k), where m is the number of functioning IoT servers and k is the number of tasks in the queueing system, with $0 \le m \le M$ and $0 \le k \le K$. Notice that the CTMC approach has been successfully applied to other service systems, e.g., an elastic cloud computing system [30]. A CTMC model is able to specify the states of a server system and the transitions among the states. Based on the probabilities of the states, various characteristics and properties of a system can be obtained analytically or numerically.

Transitions among the states are given below (see Fig. 2). (Note: The notation $(m,k) \overset{r}{\rightarrow} (m',k')$ means a transition from state (m,k) to state (m',k') with transition rate r.)

- $(m,k) \overset{\lambda}{\rightarrow} (m,k+1)$, for all $0 \le m \le M$ and $0 \le k \le K-1$. This transition happens when a new task arrives to the queueing system.
- $(m,k) \overset{\min(m,k)\mu}{\rightarrow} (m,k-1)$, for all $1 \le m \le M$ and $1 \le k \le K$. This transition happens when a task is completed and departs from the queueing system.
- $(m,k) \overset{m\alpha}{\rightarrow} (m-1,k)$, for all $1 \le m \le M$ and $0 \le k \le K$. This transition happens when a server fails. (Note that servers fail independently.)
- $(m,k) \overset{\beta}{\rightarrow} (m+1,k)$ for the sequential repairment model, or $(m,k) \overset{(M-m)\beta}{\rightarrow} (m+1,k)$ for the parallel repairment model, for all $0 \le m \le M-1$ and $0 \le k \le K$. This transition happens when a failed server is repaired/restored/replaced.

(It is common sense that when several independent Poisson streams with rates $r_1, r_2, \ldots$ are merged into one Poisson stream, the rate of the combined stream is $r_1 + r_2 + \cdots$ [24].)

Fig. 3 gives an example of the above state-transition-rate diagram for the CTMC model with $M = 3$ and $K = 7$ and parallel repairment. For the sequential repairment model, all 2β and 3β are replaced by β.

Let $p(m,k)$ be the probability that our M/M/m/K queueing system is in state (m,k), where $0 \le m \le M$ and $0 \le k \le K$.

Such a two-dimensional Markov chain does not seem to accommodate an analytical and closed-form solution of $p(m,k)$. However, the $p(m,k)$'s can be easily found numerically by solving a linear system of equations with $(M+1)(K+1)$ unknowns, i.e., the $p(m,k)$'s. The standard Gaussian elimination and backward substitution method requires $O(M^2K^2)$ time ([31], Section 6.2).

3. The metrics

In this section, we calculate various reliability, performance, and power metrics. We show that the average number of available servers, the average task response time, and the average power consumption can all be represented in closed-form expressions. For all theorems and results in this section, the reader can skip the proofs and derivations without loss of continuity.

3.1. Average number of available servers

The *reliability* of an M/M/m/K service system with server failure and repair is A/M, where A is the average number of available and functioning servers, i.e.,

$$A = \sum_{m=0}^{M} \sum_{k=0}^{K} m p(m,k). \tag{2}$$

Note that server availability is independent of λ and μ. The average number of failed servers is

$$F = \sum_{m=0}^{M} \sum_{k=0}^{K} (M-m) p(m,k), \tag{3}$$

which is also independent of λ and μ.

As λ becomes large, the two-dimensional Markov chain in Fig. 3 converges to the last column, i.e., a one-dimensional Markov chain shown in Fig. 4, where $\alpha_m = m\alpha$, $1 \le m \le M$, and $\beta_m = \beta$ for the sequential repairment model, and $\beta_m = (M-m)\beta$ for the parallel repairment model, $0 \le m \le M-1$.

Let q_m be the probability of state m in this Markov chain, where $0 \le m \le M$. We define

$$A^* = \sum_{m=0}^{M} m q_m = \sum_{m=1}^{M} m q_m \tag{4}$$

to be the average number of available servers, and

$$F^* = \sum_{m=0}^{M} (M-m) q_m = \sum_{m=0}^{M-1} (M-m) q_m \tag{5}$$

to be the average number of failed servers. Since A and F are independent of λ, we have $A = A^*$ and $F = F^*$. Thus, we only need to evaluate A^*.

Theorem 1 gives the average number of available servers.

Theorem 1. *The average number of available servers is*

$$A^* = \left(\frac{\beta}{\alpha+\beta}\right) M, \tag{6}$$

for the parallel repairment model, and

$$A^* = (1-q_M)\frac{\beta}{\alpha} = \left(1 - \frac{1}{M!}\left(\frac{\beta}{\alpha}\right)^M \Big/ \sum_{m=0}^{M} \frac{1}{m!}\left(\frac{\beta}{\alpha}\right)^m\right)\frac{\beta}{\alpha}, \tag{7}$$

for the sequential repairment model.

Proof. It is straightforward to verify from Fig. 4 that

$$q_1 = \left(\frac{\beta_0}{\alpha_1}\right) q_0,$$

$$q_2 = \left(\frac{\beta_1}{\alpha_2}\right) q_1 = \left(\frac{\beta_0\beta_1}{\alpha_1\alpha_2}\right) q_0,$$

$$q_3 = \left(\frac{\beta_2}{\alpha_3}\right) q_2 = \left(\frac{\beta_0\beta_1\beta_2}{\alpha_1\alpha_2\alpha_3}\right) q_0,$$

$$\vdots$$

$$q_M = \left(\frac{\beta_{M-1}}{\alpha_M}\right) q_{M-1} = \left(\frac{\beta_0\beta_1\cdots\beta_{M-1}}{\alpha_1\alpha_2\cdots\alpha_M}\right) q_0,$$

where

$$q_0 = \left(1 + \frac{\beta_0}{\alpha_1} + \frac{\beta_0\beta_1}{\alpha_1\alpha_2} + \frac{\beta_0\beta_1\beta_2}{\alpha_1\alpha_2\alpha_3} + \cdots + \frac{\beta_0\beta_1\cdots\beta_{M-1}}{\alpha_1\alpha_2\cdots\alpha_M}\right)^{-1}.$$

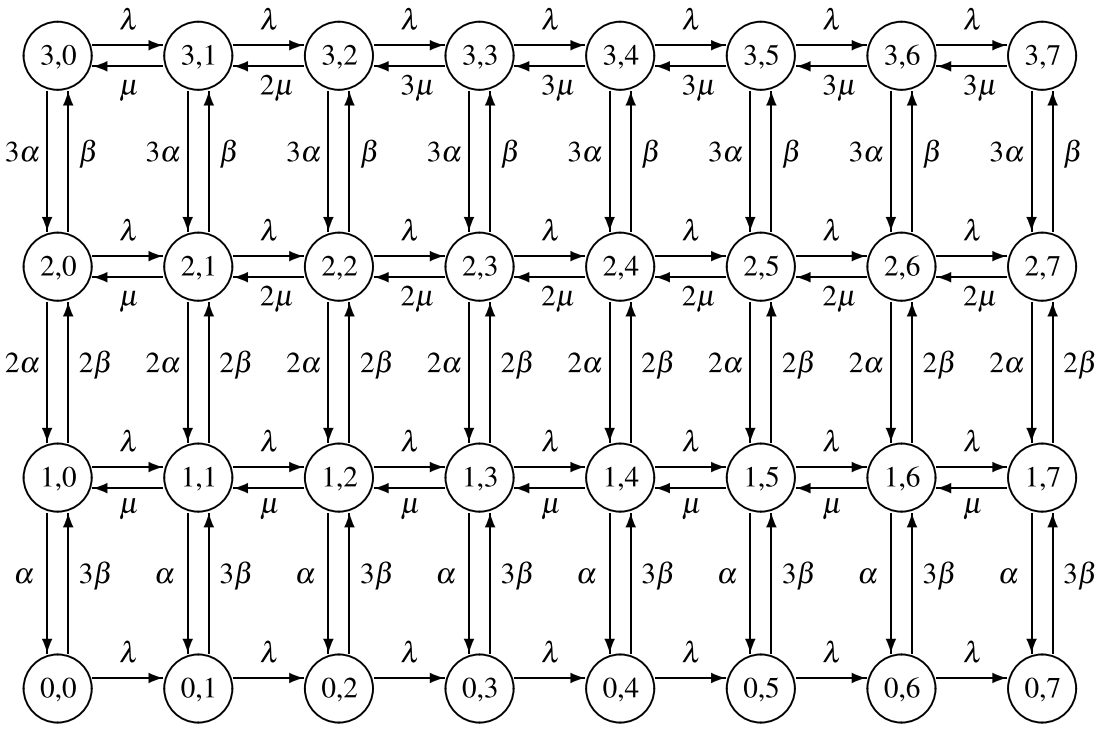

Fig. 3. A state-transition-rate diagram for the CTMC model ($M = 3$ and $K = 7$).

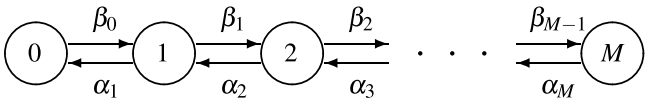

Fig. 4. A state-transition-rate diagram for the number of available servers.

Thus, we get

$$q_m = \left(\frac{\beta_0\beta_1\cdots\beta_{m-1}}{\alpha_1\alpha_2\cdots\alpha_m}\right)\left(1+\frac{\beta_0}{\alpha_1}+\frac{\beta_0\beta_1}{\alpha_1\alpha_2}+\frac{\beta_0\beta_1\beta_2}{\alpha_1\alpha_2\alpha_3}+\cdots+\frac{\beta_0\beta_1\cdots\beta_{M-1}}{\alpha_1\alpha_2\cdots\alpha_M}\right)^{-1},$$

for all $1 \le m \le M$.

For the parallel repairment model, $\alpha_m = m\alpha$ and $\beta_m = (M-m)\beta$, which yield

$$\begin{aligned}
q_1 &= \frac{M}{1!}\left(\frac{\beta}{\alpha}\right)q_0,\\
q_2 &= \frac{M(M-1)}{2!}\left(\frac{\beta}{\alpha}\right)^2 q_0,\\
q_3 &= \frac{M(M-1)(M-2)}{3!}\left(\frac{\beta}{\alpha}\right)^3 q_0,\\
&\vdots\\
q_M &= \frac{M(M-1)(M-2)\cdots 1}{M!}\left(\frac{\beta}{\alpha}\right)^M q_0.
\end{aligned}$$

Hence, we have

$$\begin{aligned}
A^* &= \sum_{m=1}^{M} m q_m\\
&= \sum_{m=1}^{M} m\times\frac{M(M-1)\cdots(M-m+1)}{m!}\left(\frac{\beta}{\alpha}\right)^m q_0\\
&= \sum_{m=1}^{M}\frac{M(M-1)\cdots(M-m+1)}{(m-1)!}\left(\frac{\beta}{\alpha}\right)^m q_0,
\end{aligned}$$

and

$$\begin{aligned}
A^*\left(\frac{\alpha}{\beta}\right) &= \sum_{m=1}^{M}\frac{M(M-1)\cdots(M-m+1)}{(m-1)!}\left(\frac{\beta}{\alpha}\right)^{m-1} q_0\\
&= \sum_{m=1}^{M}(M-m+1)\times\frac{M(M-1)\cdots(M-m+2)}{(m-1)!}\left(\frac{\beta}{\alpha}\right)^{m-1} q_0\\
&= \sum_{m=1}^{M}(M-(m-1))q_{m-1}\\
&= \sum_{m=0}^{M-1}(M-m)q_m\\
&= F^*.
\end{aligned}$$

Since $A^* + F^* = M$, that is,

$$A^* + A^*\left(\frac{\alpha}{\beta}\right) = M,$$

we obtain

$$A^* = \left(\frac{\beta}{\alpha+\beta}\right)M,$$

for the parallel repairment model.

For the sequential repairment model, $\alpha_m = m\alpha$ and $\beta_m = \beta$, which yield

$$\begin{aligned}
q_1 &= \frac{1}{1!}\left(\frac{\beta}{\alpha}\right)q_0,\\
q_2 &= \frac{1}{2!}\left(\frac{\beta}{\alpha}\right)^2 q_0,\\
q_3 &= \frac{1}{3!}\left(\frac{\beta}{\alpha}\right)^3 q_0,\\
&\vdots\\
q_M &= \frac{1}{M!}\left(\frac{\beta}{\alpha}\right)^M q_0.
\end{aligned}$$

Hence, we have

$$A^* = \sum_{m=1}^{M} m q_m$$

$$= \sum_{m=1}^{M} m \times \frac{1}{m!}\left(\frac{\beta}{\alpha}\right)^m q_0$$
$$= \sum_{m=1}^{M} \frac{1}{(m-1)!}\left(\frac{\beta}{\alpha}\right)^m q_0,$$

and

$$A^*\left(\frac{\alpha}{\beta}\right) = \sum_{m=1}^{M} \frac{1}{(m-1)!}\left(\frac{\beta}{\alpha}\right)^{m-1} q_0$$
$$= \sum_{m=1}^{M} q_{m-1}$$
$$= \sum_{m=0}^{M-1} q_m$$
$$= 1 - q_M,$$

which implies that

$$A^* = (1 - q_M)\frac{\beta}{\alpha},$$

where

$$q_M = \frac{1}{M!}\left(\frac{\beta}{\alpha}\right)^M \Big/ \sum_{m=0}^{M} \frac{1}{m!}\left(\frac{\beta}{\alpha}\right)^m,$$

for the sequential repairment model. ■

We would like to emphasize that Eqs. (6) and (7) imply that A^* only depends on α, β, and M, and is independent of λ, μ, and K.

For the parallel repairment model, the failure and repair of servers are independent of each other. Consequently, the average number of available servers is actually the sum of single server availability (or reliability), i.e.,

$$A = RM = \left(\frac{\beta}{\alpha+\beta}\right)M. \tag{8}$$

3.2. Average task response time

The average number of tasks in the queueing system is

$$N = \sum_{m=0}^{M}\sum_{k=0}^{K} kp(m,k). \tag{9}$$

The average number of tasks in the waiting queue is

$$Q = \sum_{m=0}^{M}\sum_{k=m+1}^{K} (k-m)p(m,k). \tag{10}$$

The probability that a newly arriving task is discarded due to limited capacity is

$$D = \sum_{m=0}^{M} p(m,K). \tag{11}$$

Theorem 2 gives the average task response time.

Theorem 2. *The average task response time is*

$$T = \frac{N}{\lambda(1-D)}, \tag{12}$$

and equivalently,

$$T = \frac{Q}{\lambda(1-D)} + \frac{1}{\mu}. \tag{13}$$

Proof. When the queueing system is in state (m,K), where $0 \le m \le M$, a newly arriving task is discarded without processing. Such an event occurs with probability D. Hence, the actual and effective task arrival rate is no longer λ, but

$$\lambda_{eff} = \lambda(1-D). \tag{14}$$

According to Little's result, the average task response time is

$$T = \frac{N}{\lambda_{eff}} = \frac{N}{\lambda(1-D)}.$$

Also according to Little's result, the average waiting time is

$$W = \frac{Q}{\lambda_{eff}} = \frac{Q}{\lambda(1-D)}. \tag{15}$$

Therefore, we get

$$T = W + \frac{1}{\mu} = \frac{Q}{\lambda(1-D)} + \frac{1}{\mu},$$

where $1/\mu$ is the average task execution time. ■

The average number of busy servers that are processing some tasks (i.e., the average number of tasks in processing) is

$$B = \sum_{m=0}^{M}\sum_{k=0}^{K} \min(m,k)p(m,k). \tag{16}$$

The server utilization is

$$U = \frac{B}{M}. \tag{17}$$

It is clear that $\lim_{\lambda\to\infty} B = A$ and $\lim_{\lambda\to\infty} U = A/M$.

Theorem 1 implies that

$$N = Q + \lambda_{eff}/\mu. \tag{18}$$

This can be seen as follows.

Notice that

$$N = \sum_{m=0}^{M}\sum_{k=0}^{K} kp(m,k)$$
$$= \sum_{m=0}^{M}\left(\sum_{k=0}^{m} kp(m,k) + \sum_{k=m+1}^{K} kp(m,k)\right)$$
$$= \sum_{m=0}^{M}\sum_{k=0}^{m} kp(m,k) + \sum_{m=0}^{M}\sum_{k=m+1}^{K} kp(m,k)$$
$$= \sum_{m=0}^{M}\sum_{k=0}^{m} kp(m,k) + \sum_{m=0}^{M}\left(\sum_{k=m+1}^{K} mp(m,k) + \sum_{k=m+1}^{K}(k-m)p(m,k)\right)$$
$$= \sum_{m=0}^{M}\sum_{k=0}^{m} kp(m,k) + \sum_{m=0}^{M}\sum_{k=m+1}^{K} mp(m,k) + \sum_{m=0}^{M}\sum_{k=m+1}^{K}(k-m)p(m,k)$$
$$= \sum_{m=0}^{M}\left(\sum_{k=0}^{m} kp(m,k) + \sum_{k=m+1}^{K} mp(m,k)\right) + \sum_{m=0}^{M}\sum_{k=m+1}^{K}(k-m)p(m,k)$$
$$= \sum_{m=0}^{M}\sum_{k=0}^{K} \min(m,k)p(m,k) + \sum_{m=0}^{M}\sum_{k=m+1}^{K}(k-m)p(m,k)$$
$$= B + Q,$$

where B is exactly λ_{eff}/μ.

We notice that λ_{eff}/μ is actually the average number of tasks in execution.

3.3. Average power consumption

An IoT server with service rate μ consumes power

$$P_d + P_s = \xi\mu^d + P_s,$$

where

$$P_d = \xi\mu^d$$

is the dynamic component, P_s is the static component, and ξ and d are technology dependent constants. The above power consumption model has been widely used in the literature [32].

We assume that a failed server does not consume power. In reality, a failed server can be easily detected and shut down immediately.

Theorem 3 gives the average power consumption.

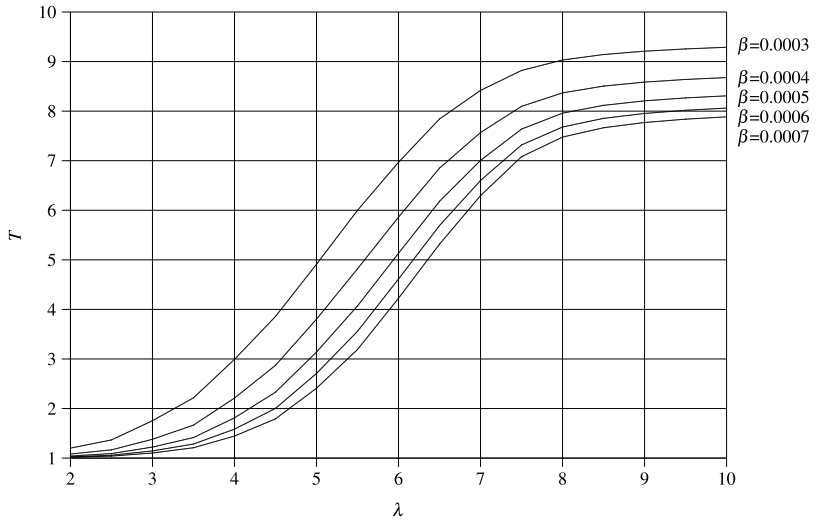

Fig. 5. The average task response time T vs. λ (parallel, varying β).

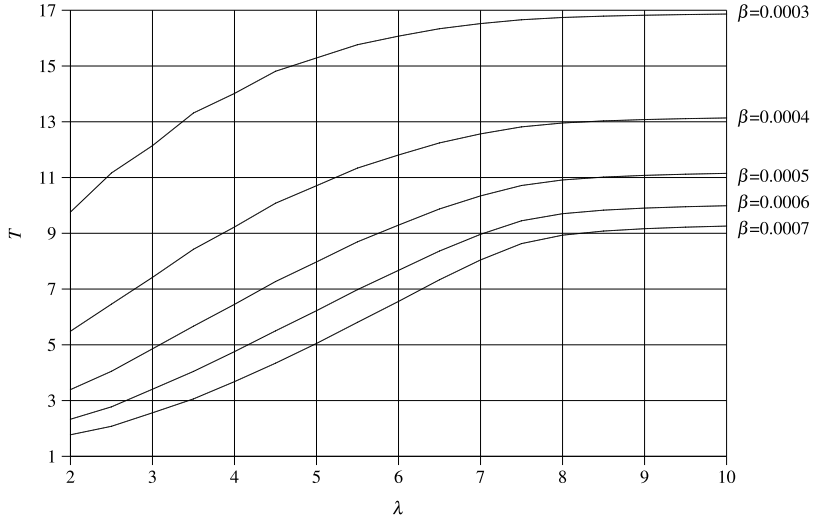

Fig. 6. The average task response time T vs. λ (sequential, varying β).

Theorem 3. *The average power consumption is*

$$P = (\xi\mu^d + P_s)\left(\frac{\beta}{\alpha+\beta}\right)M, \tag{19}$$

for the parallel repairment model, and

$$P = (\xi\mu^d + P_s)(1-q_M)\frac{\beta}{\alpha} = (\xi\mu^d + P_s)\left(1-\frac{1}{M!}\left(\frac{\beta}{\alpha}\right)^M \Big/ \sum_{m=0}^{M}\frac{1}{m!}\left(\frac{\beta}{\alpha}\right)^m\right)\frac{\beta}{\alpha}, \tag{20}$$

for the sequential repairment model.

Proof. Note that

$$P = (\xi\mu^d + P_s)\sum_{m=0}^{M}\sum_{k=0}^{K} mp(m,k).$$

We can rewrite P as

$$P = (\xi\mu^d + P_s)A = (\xi\mu^d + P_s)A^*.$$

The rest of the proof is straightforward based on Theorem 2. ■

4. Numerical data

In this section, we present numerical data and examples.

4.1. Parameter settings

We consider an IoT system with $M = 7$ servers and total capacity $K = 50$ tasks.

We have the following parameter settings for Figs. 5–10. The task service rate is $\mu = 1.0$ task/second. The server failure rate is $\alpha = 0.0001$ server/second. Since the server reliability $R = \gamma/(1+\gamma)$ is essentially determined by the ratio $\gamma = \beta/\alpha$, we will fix α and change β.

We have the following parameter settings for Figs. 11–14. The task arrival rate is $\lambda = 5.0$ tasks/second. The server failure rate is $\alpha = 0.0001$ server/second. The parameters of the power consumption model are $\xi = 5$ Watts/(task/second)^d, $d = 2$, and $P_s = 3$ Watts.

It is well known that server failure rate α depends on the computing speed, which is the service rate μ in our context. In the following, we consider such μ-dependent α, which is characterized by (adapted from [26])

$$\alpha = \alpha_0 10^{s(\mu_{\max}-\mu)/(\mu_{\max}-\mu_{\min})},$$

where α_0 is a base failure rate, $\mu_{\min}$ is the minimum service rate, $\mu_{\max}$ is the maximum service rate, and s is some scaling constant. It is clear that (1) α is a decreasing function of μ; (2) α is in the range $[\alpha_0, 10^s\alpha_0]$; (3) $\alpha = \alpha_0$ when $\mu = \mu_{\max}$, and $\alpha = 10^s\alpha_0$ when $\mu = \mu_{\min}$.

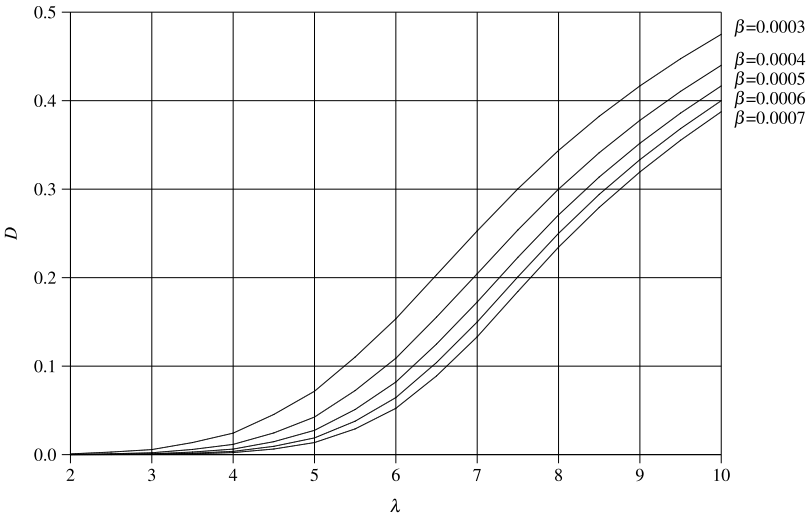

Fig. 7. The probability of task rejection D vs. λ (parallel, varying β).

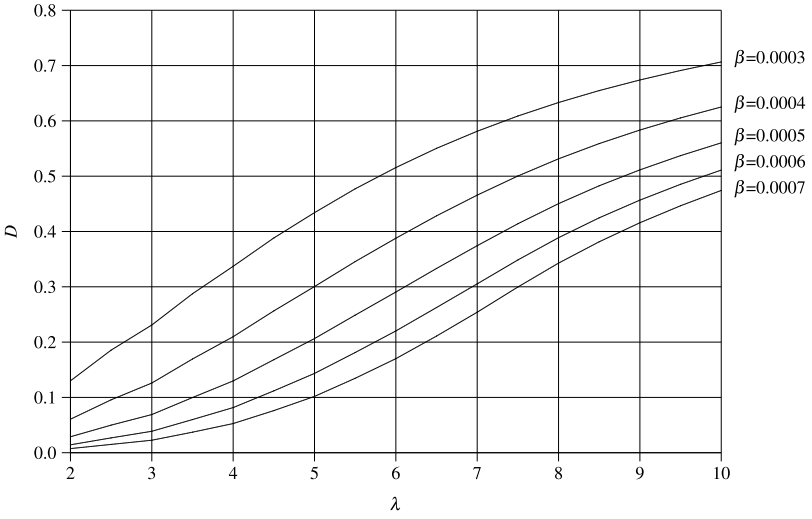

Fig. 8. The probability of task rejection D vs. λ (sequential, varying β).

In Figs. 15–22, we keep the same parameter settings as Figs. 11–14, with the following new parameters: $\alpha_0 = 0.00001$ server/second, $\mu_{\min} = 1.0$ task/second, $\mu_{\max} = 4.0$ tasks/second, and $s = 1.0$.

4.2. Average task response time

Figs. 5 and 6 show the average task response time T vs. the task arrival rate λ for the parallel and sequential repairment models respectively, with the server repairment rate $\beta = 0.0003, 0.0004, 0.0005, 0.0006, 0.0007$ server/second.

We have the following observations. First, T increases as λ increases; however, as λ further increases, more tasks are discarded due to the limited capacity of an IoT system, and T increases very slowly. Second, the average response time of the parallel repairment model is significantly lower than that of the sequential repairment model due to more available servers. To be more specific, for $\beta = 0.0003, 0.0004, 0.0005, 0.0006, 0.0007$, we have $A = 5.25000, 5.60000, 5.83333, 6.00000, 6.12500$ for the parallel repairment model, and $A = 2.93441, 3.74900, 4.39741, 4.88967, 5.25790$ for the sequential repairment model. Third, as β (and γ and R as well) increases, the average number A of available servers increases, and T decreases noticeably for the parallel repairment model and significantly for the sequential repairment model. This implies that server reliability does have a strong impact on system performance.

4.3. Probability of task rejection

Figs. 7 and 8 show the probability of task rejection D vs. the task arrival rate λ for the parallel and sequential repairment models respectively, with the server repairment rate $\beta = 0.0003, 0.0004, 0.0005, 0.0006, 0.0007$ server/second.

We have the following observations. First, as λ increases, D increases noticeably, i.e., more tasks are discarded due to the limited capacity of an IoT system. Second, the task rejection probability of the parallel repairment model is significantly lower than that of the sequential repairment model due to more available servers. Third, as β (and γ and R as well) increases, the average number A of available servers increases, and D decreases noticeably for the parallel repairment model and significantly for the sequential repairment model. This implies that server reliability does have a strong impact on task rejection probability.

4.4. Server utilization

Figs. 9 and 10 show the server utilization U vs. the task arrival rate λ for the parallel and sequential repairment models respectively, with the server repairment rate $\beta = 0.0003, 0.0004, 0.0005, 0.0006, 0.0007$ server/second.

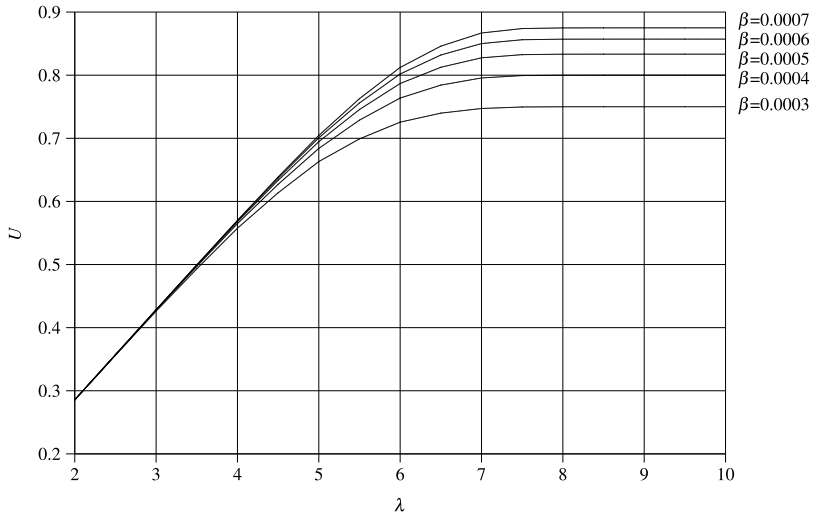

Fig. 9. The server utilization U vs. λ (parallel, varying β).

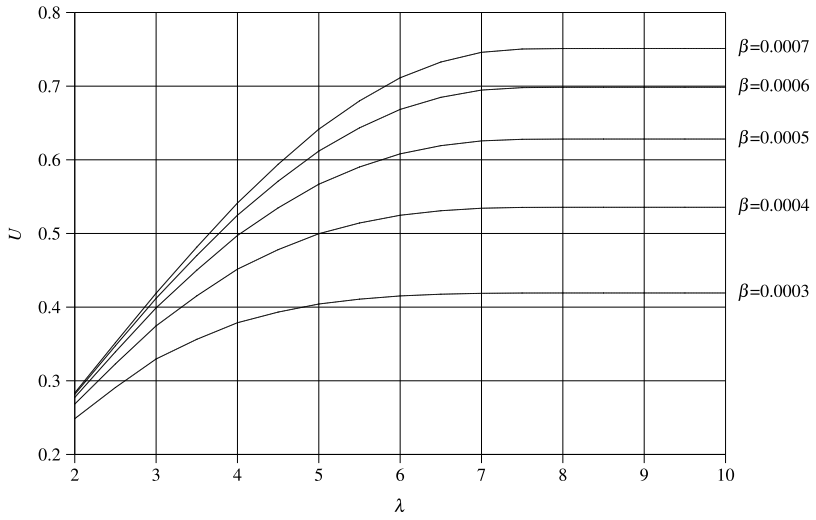

Fig. 10. The server utilization U vs. λ (sequential, varying β).

We have the following observations. First, U increases as λ increases and eventually, U approaches A/M. To be more specific, for $\beta = 0.0003, 0.0004, 0.0005, 0.0006, 0.0007$, we have $A/M = 0.75000, 0.80000, 0.83333, 0.85714, 0.87500$ for the parallel repairment model, and $A/M = 0.41920, 0.53557, 0.62820, 0.69852, 0.75113$ for the sequential repairment model. Second, server utilization of the parallel repairment model is significantly higher than that of the sequential repairment model due to more available servers. Third, as β (and γ and R as well) increases, the average number A of available servers increases, and U increases noticeably for the parallel repairment model and significantly for the sequential repairment model. This implies that server reliability does have a strong impact on server utilization.

4.5. Average power consumption

Figs. 11 and 12 show the average power consumption P vs. the task service rate $\mu = 1.00, 1.05, 1.10, \ldots, 4.00$ tasks/second for the parallel and sequential repairment models respectively, with the server repairment rate $\beta = 0.0003, 0.0004, 0.0005, 0.0006, 0.0007$ server/second.

We have the following observations. First, P increases as μ increases, since power consumption is a quadratic function of μ. Second, the average power consumption of the parallel repairment model is significantly higher than that of the sequential repairment model due to more available servers. Third, as β (and γ and R as well) increases, the average number A of available servers increases, and P increases noticeably for the parallel repairment model and significantly for the sequential repairment model. This implies that server reliability does have a strong impact on power consumption.

4.6. Power-time product

Figs. 13 and 14 show the power-time product PT vs. the task service rate $\mu = 1.00, 1.05, 1.10, \ldots, 4.00$ tasks/second for the parallel and sequential repairment models respectively, with the server repairment rate $\beta = 0.0003, 0.0004, 0.0005, 0.0006, 0.0007$ server/second.

We have the following observations. First, for the parallel repairment model, PT is approximately a convex function of μ and there is μ^* that makes PT reaching its minimum. Specifically, for $\beta = 0.0003, 0.0004, 0.0005, 0.0006, 0.0007$, we have $\mu^* = 1.95, 1.85, 1.80, 1.70, 1.50$. For the sequential repairment model, PT fluctuates and has several local minimums. Of course, there is μ^* that makes PT reaching its global minimum. Specifically, for $\beta = 0.0003, 0.0004, 0.0005, 0.0006, 0.0007$, we have $\mu^* = 1.80, 2.80, 2.80, 2.80, 2.75$. Unfortunately, the optimal service rate (i.e., the optimal server speed) μ^* is analytically not available but only observed numerically. Second, the power-time product of the parallel repairment model is significantly lower than

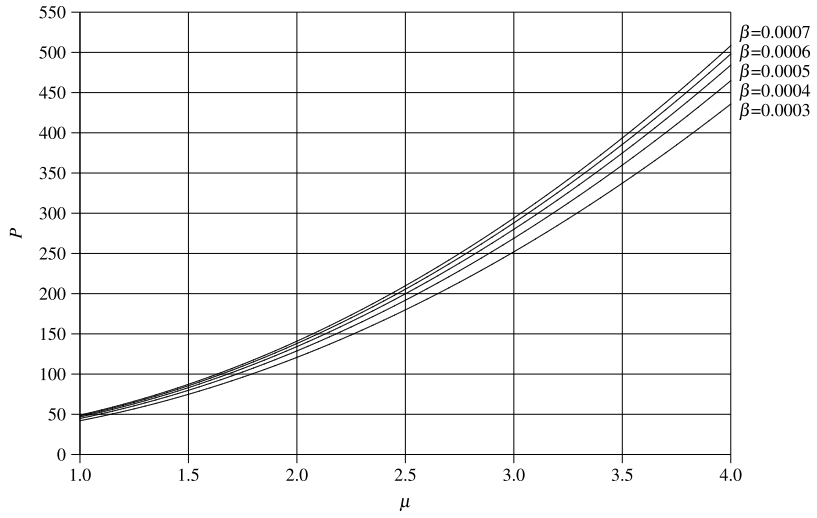

Fig. 11. The average power consumption P vs. μ (parallel, varying β).

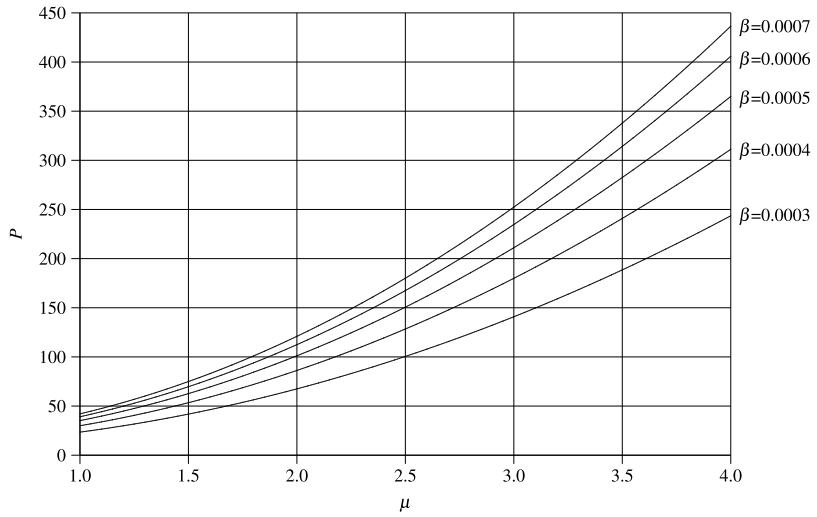

Fig. 12. The average power consumption P vs. μ (sequential, varying β).

that of the sequential repairment model due to more available servers and shorter average response time. Third, as β (and γ and R as well) increases, PT decreases noticeably (but increases slightly beyond a certain point) for the parallel repairment model and decreases significantly for the sequential repairment model. This implies that server reliability does have a strong impact on the power-time product.

4.7. Server failure rate

Fig. 15 displays the server failure rate α (actually, $\alpha \times 10^5$) as a function of μ. It is observed that α is a decreasing function of μ and α is in the range $[0.00001, 0.0001]$.

4.8. Server reliability

Fig. 16 displays the server reliability R as a function of μ, with the server repairment rate $\beta = 0.0003, 0.0004, 0.0005, 0.0006, 0.0007$ server/second.

We have the following observations. First, R increases as μ increases and α decreases. Second, as β increases, the server reliability R increases significantly when μ is relatively small and α is relatively large, and noticeably when μ is relatively large and α is relatively small.

4.9. Average number of available servers

Figs. 17 and 18 display the average number of available servers A as a function of the task service rate $\mu = 1.00, 1.05, 1.10, \ldots, 4.00$ tasks/second for the parallel and sequential repairment models respectively, with the server repairment rate $\beta = 0.0003, 0.0004, 0.0005, 0.0006, 0.0007$ server/second.

We have the following observations. First, A increases as μ increases and α decreases. Second, the server availability A of the parallel repairment model is significantly higher than that of the sequential repairment model (especially when μ is relatively small and α is relatively large) due to faster repairment and fewer failed servers. Third, as β increases, the server availability A increases significantly when μ is relatively small and α is relatively large, and noticeably when μ is relatively large and α is relatively small.

4.10. Power-time product with μ-dependent α

Figs. 19 and 20 demonstrate the power-time product PT vs. the task service rate $\mu = 1.00, 1.05, 1.10, \ldots, 4.00$ tasks/second for the parallel and sequential repairment models respectively, with the server repairment rate $\beta = 0.0003, 0.0004, 0.0005, 0.0006, 0.0007$ server/second.

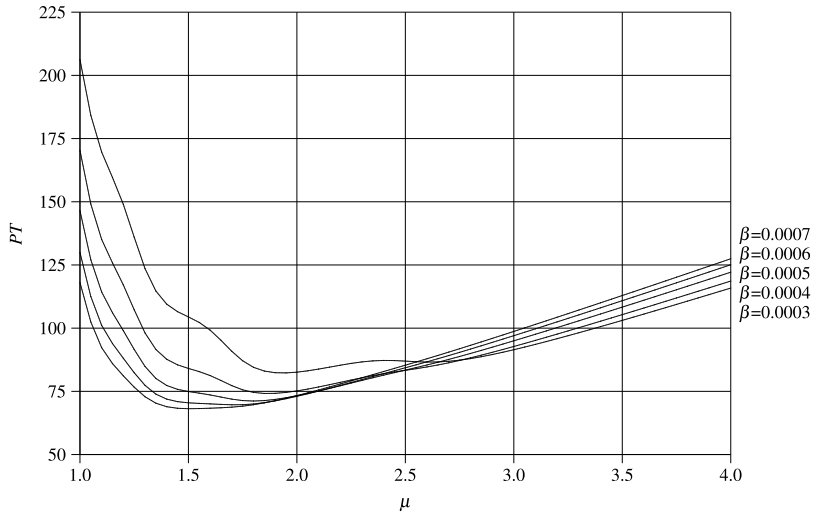

Fig. 13. The power-time product PT vs. μ (parallel, varying β).

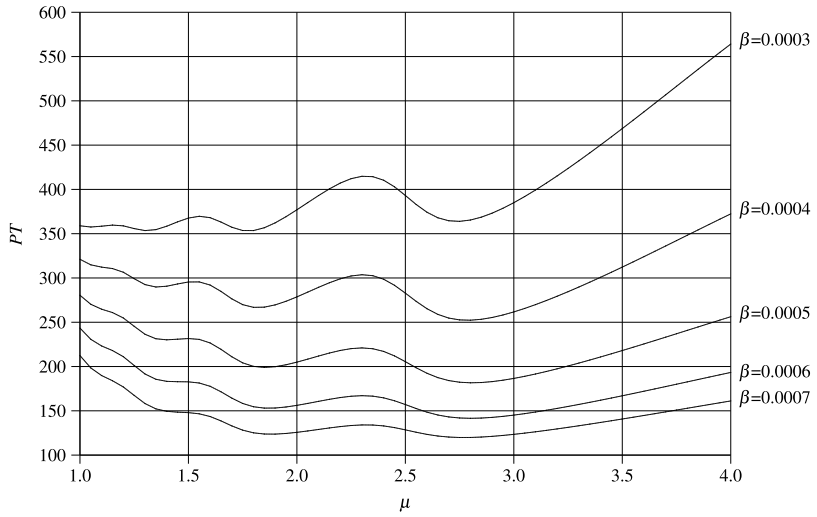

Fig. 14. The power-time product PT vs. μ (sequential, varying β).

We have the following observations. First, PT is approximately a convex function of μ and there is μ^* that makes PT reaching its minimum. For the parallel repairment model, for $\beta = 0.0003, 0.0004, 0.0005, 0.0006, 0.0007$, we have $\mu^* = 1.80, 1.70, 1.55, 1.50, 1.45$. For the sequential repairment model, for $\beta = 0.0003, 0.0004, 0.0005, 0.0006, 0.0007$, we have $\mu^* = 2.65, 2.45, 2.10, 1.95, 1.85$. Unfortunately, the optimal service rate (i.e., the optimal server speed) μ^* is analytically not available but only observed numerically. Second, the power-time product of the parallel repairment model is significantly lower than that of the sequential repairment model due to more available servers and a shorter average response time. Third, as β (and γ and R as well) increases, PT decreases noticeably (but increases slightly beyond a certain point) for the parallel repairment model and decreases significantly (but increases slightly beyond a certain point) for the sequential repairment model.

4.11. Joint cost-performance metric

Figs. 21 and 22 demonstrate a joint cost-performance metric that combines power, time, and reliability together (i.e., PT/R) vs. the task service rate $\mu = 1.00, 1.05, 1.10, \ldots, 4.00$ tasks/second for the parallel and sequential repairment models respectively, with the server repairment rate $\beta = 0.0003, 0.0004, 0.0005, 0.0006, 0.0007$ server/second.

We have the following observations. First, PT/R is approximately a convex function of μ and there is μ^* that makes PT/R reaching its minimum. For the parallel repairment model, for $\beta = 0.0003, 0.0004, 0.0005, 0.0006, 0.0007$, we have $\mu^* = 1.85, 1.75, 1.65, 1.55, 1.50$. For the sequential repairment model, for $\beta = 0.0003, 0.0004, 0.0005, 0.0006, 0.0007$, we have $\mu^* = 2.70, 2.50, 2.15, 1.95, 1.90$. Unfortunately, the optimal service rate (i.e., the optimal server speed) μ^* is analytically not available but only observed numerically. Second, PT/R of the parallel repairment model is significantly lower than that of the sequential repairment model due to more available servers and a shorter average response time. Third, as β (and γ and R as well) increases, PT/R decreases noticeably for the parallel repairment model and decreases significantly (but increases slightly beyond a certain point) for the sequential repairment model.

5. Related work

In this section, we review related work.

There are multiple reviews and surveys covering various aspects of IoT research [33–38].

Researchers have adopted IoT devices as computational resources to construct distributed and pervasive computing environments. Hasan et al. presented a highly localized IoT-based cloud computing model that allows mobile users to create an ad hoc and flexible cloud by using

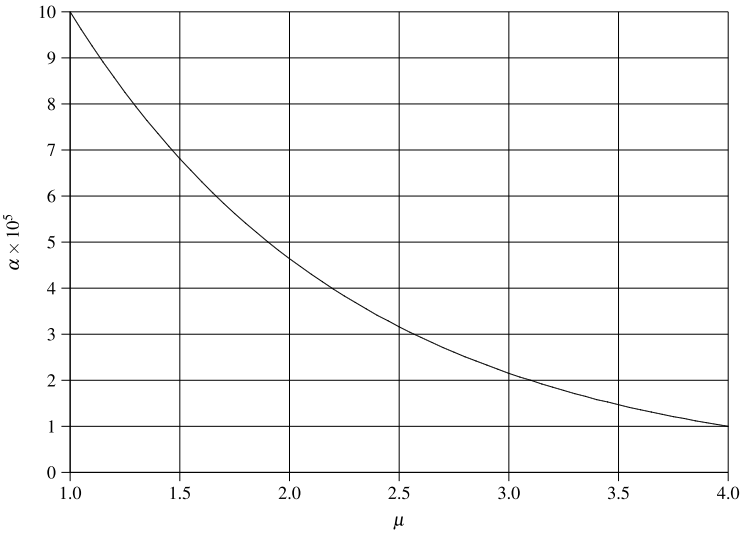

Fig. 15. The server failure rate $\alpha \times 10^5$ vs. μ (μ-dependent α).

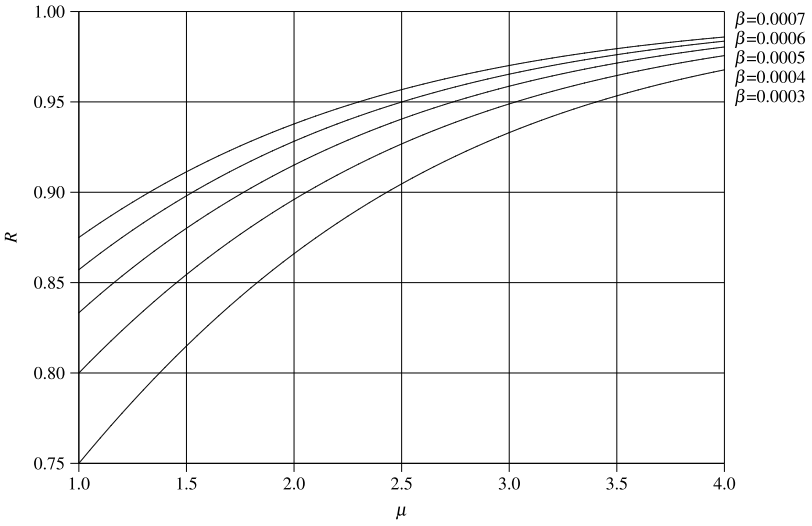

Fig. 16. The server reliability R vs. μ (μ-dependent α).

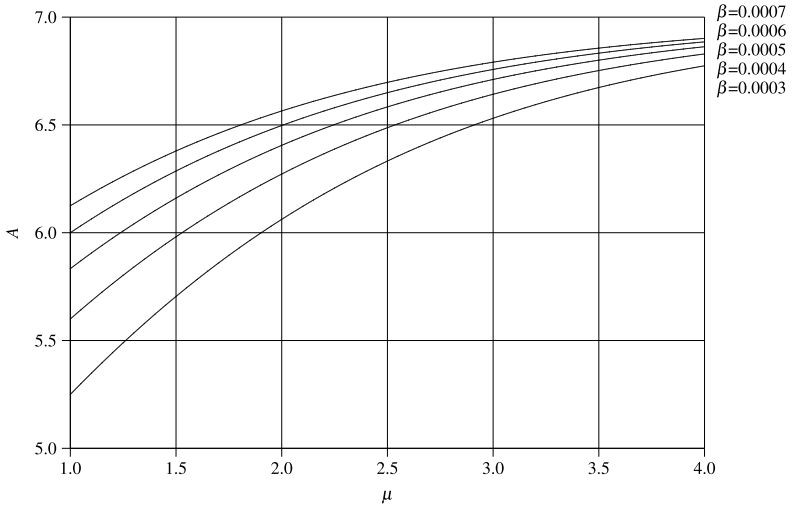

Fig. 17. The average number of available servers A vs. μ (μ-dependent α, parallel, varying β).

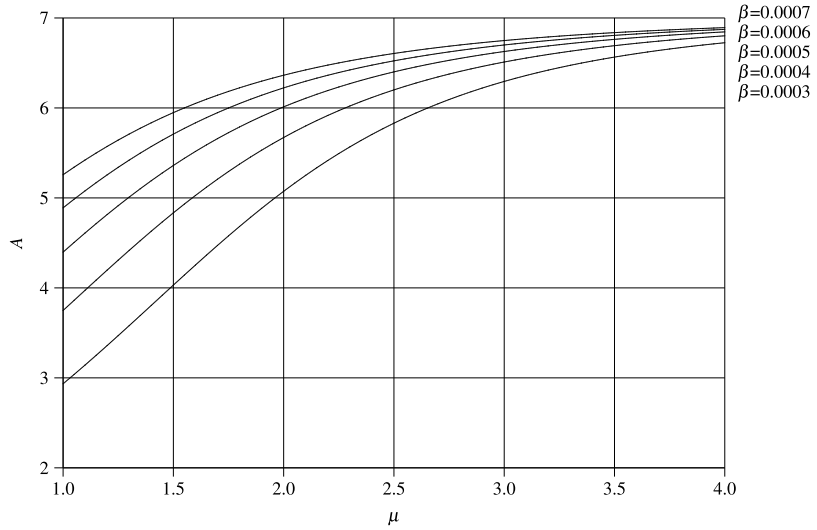

Fig. 18. The average number of available servers A vs. μ (μ-dependent α, sequential, varying β).

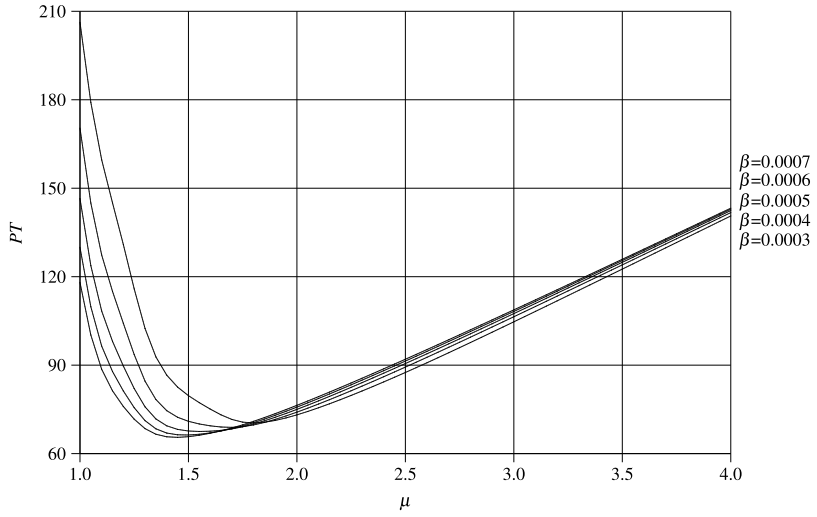

Fig. 19. The power-time product PT vs. μ (μ-dependent α, parallel, varying β).

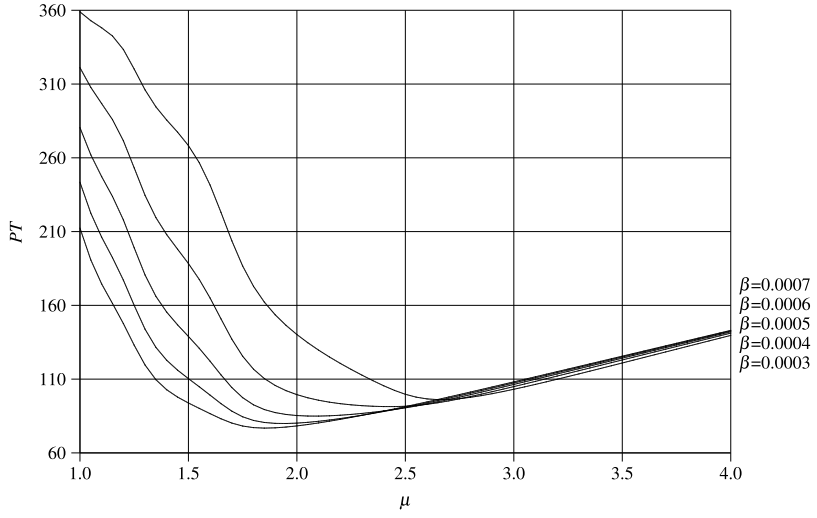

Fig. 20. The power-time product PT vs. μ (μ-dependent α, sequential, varying β).

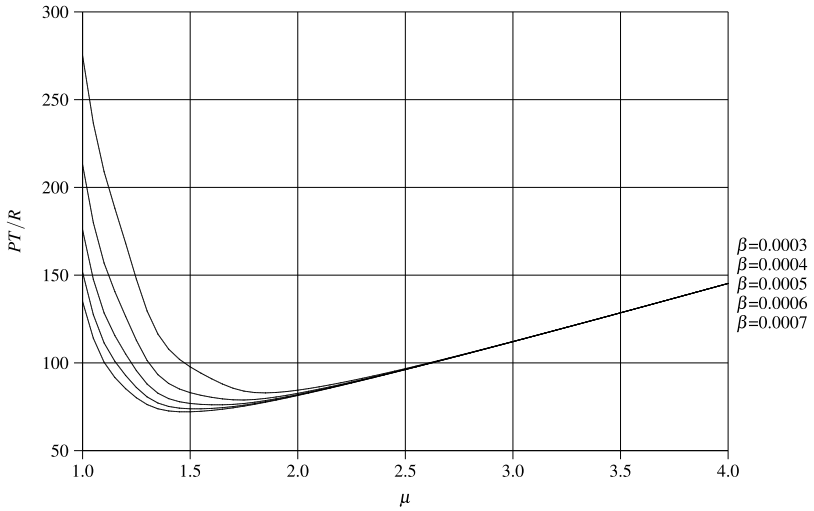

Fig. 21. Joint cost-performance metric PT/R vs. μ (μ-dependent α, parallel, varying β).

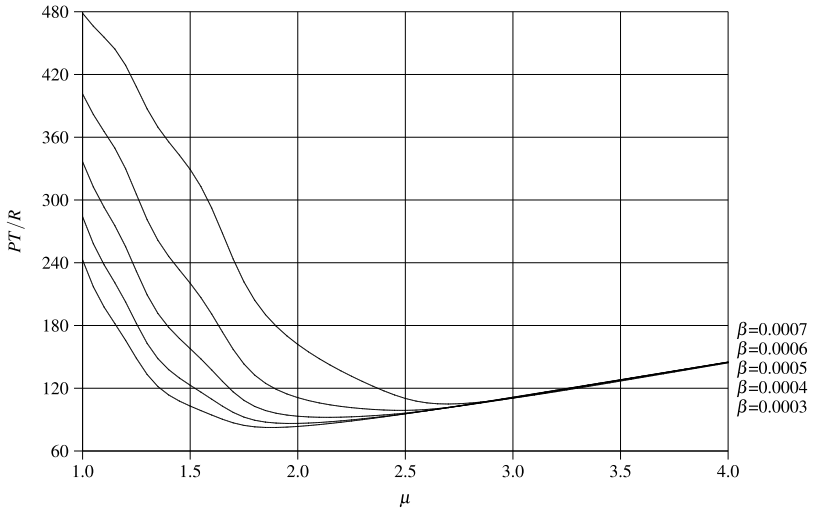

Fig. 22. Joint cost-performance metric PT/R vs. μ (μ-dependent α, sequential, varying β).

IoT devices in the nearby physical environment with full control to start, stop, migrate, and restart computations in localized IoT devices as a mobile user moves between different physical locations [7]. Laroui et al. described a virtual edge servers-based framework, where smart connected devices act as virtual edge servers that work together to provide computation services close to end-users and to accomplish tasks for end-users [8].

There is a large body of literature on IoT reliability. Several authors have produced comprehensive surveys and reviews of IoT reliability. Khan et al. provided an overview of reliable data transmission in IoT networks with a focus on resource allocation, latency management, security, and reliability metrics [10]. Moore et al. conducted a survey of IoT reliability based on a four-layer IoT architecture, which includes device, fog, service management, and cloud and application layer [12]. Based on another four-layer IoT architecture, Xing gave a systematic review of IoT perception technologies reliability, IoT communication and transport reliability, IoT support technologies reliability, and IoT applications and services reliability [14].

Many researchers have considered quantitative reliability analysis of IoT systems. By applying the reliability block diagram paradigm, Azghiou et al. established a framework for end-to-end IoT system reliability modeling and analysis based on a layered IoT architecture (i.e., perception layer, access network layer, core network layer, middleware layer, and application layer) [9]. Nguyen et al. proposed a hierarchical modeling framework for the availability and security quantification of an IoT infrastructure, including a reliability block diagram at the top level to capture the overall architecture, a fault tree at the middle level to elaborate member systems, and a continuous time Markov chain at the bottom level to capture detailed states and transitions [13].

A number of researchers have investigated performance evaluation in IoT environments. Aslanpour et al. presented a taxonomy of real-world performance metrics for evaluating IoT, edge, fog, and cloud computing [19]. Ejaz et al. compared the traditional cloud-IoT model, a MEC-based edge–cloud-IoT model, and a local edge–cloud-IoT model with respect to their execution time, end-to-end latency, services completed, energy consumption, and operational cost [20]. Based on transparent computing, Guo et al. proposed an IoT architecture which consists of five layers (i.e., end user layer, edge network layer, core network layer, service and storage layer, and management layer) to support scalable and manageable IoT applications, and evaluated the performance of the proposed architecture in terms of service delay and energy consumption [21].

Some authors have attempted to study performance optimization in IoT computing. Teng established an open Jackson network with feedback for a three-layer (i.e., IoT devices, fog nodes, and a cloud server)

fog-based IoT platform and solved an optimal fog node service capability allocation problem to minimize the mean service request sojourn time [22]. Zhang et al. proposed a joint optimization framework for fog nodes, data service operators (DSOs), and data service subscribers (DSSs) in a three-tier IoT fog network to achieve an optimal resource allocation scheme in a distributed fashion with the formulation of a Stackelberg game to analyze the pricing problem for the DSOs as well as the resource allocation problem for the DSSs [23].

Albreem et al. provided insights on green IoT applications, practices, awareness, and challenges, including energy-efficient hardware and software design and implementation, together with new enabling computing such as artificial intelligence and machine learning [15]. Alsharif et al. presented a thorough examination of energy-efficient practices and strategies for eco-friendly and eco-sustainable IoT with consideration of M2M, WSN, RFID, MCU, and IC [16]. Raval et al. developed an energy management system for IoT devices by using a genetic algorithm to optimize the parameters of a multi-agent system with consideration of both hardware and software aspects [17]. Reddy et al. adopted an AI-based approach to fairly distribute power levels to small portable devices in a massive IoT by using a new improved random energy optimization algorithm [18].

As mentioned earlier, despite the above research, there has been little study that considers reliability, power, and performance in a combined way. The difference and deviation of the present paper from all previous studies is that we take an integrated approach to analytical modeling, evaluation, and optimization of reliability, power, performance, and power-performance tradeoff.

6. Concluding remarks

Although IoT reliability, power, and performance have been studied extensively, there has been little joint analytical investigation on these important aspects together with their interaction and interplay. This paper has made efforts and contributions towards this direction.

We are able to construct an analytical model of IoT devices and servers such that server reliability, server utilization, response time, power consumption, and cost-performance tradeoff can all be analytically defined and available, and numerically evaluated and compared. Our study provides deep insights into the impact of server failure rate and repairment rate on server reliability, server availability, and further on response time, power consumption, and cost-performance tradeoff.

An important advantage and a unique strength of our model and method is that based on several key parameters that can be routinely obtained in any real-world scenario, we are able to capture and calculate important system attributes and properties. This makes it possible and feasible to apply our approach to a wide range of diversified IoT service systems and application environments.

Furthermore, the methodology developed in this paper can also be applied to other distributed computing systems and service-oriented systems such as mobile edge computing, fog computing, device–edge–cloud collaborative computing, and UAV-enabled systems.

CRediT authorship contribution statement

Keqin Li: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

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.

Acknowledgments

Sincere and special thanks are due to four anonymous reviewers and the editor for their timely review and their criticism and comments on the manuscript.

References

[1] S. Abdulmalek, A. Nasir, W.A. Jabbar, A.K. Bairagi, A.-M. Khan, S.-H. Kee, IoT-based healthcare-monitoring system towards improving quality of life: A review, Healthcare 10 (10) (1993) 2022.
[2] Z. Alavikia, M. Shabro, A comprehensive layered approach for implementing Internet of Things-enabled smart grid: A survey, Digit. Commun. Netw. 8 (3) (2022) 388–410.
[3] P. Bellini, P. Nesi, G. Pantaleo, IoT-enabled smart cities: A review of concepts, frameworks and key technologies, Appl. Sci. 12 (3) (2022) 1607.
[4] A. Kaur, G. Singh, V. Kukreja, S. Sharma, S. Singh, B. Yoon, Adaptation of IoT with blockchain in food supply chain management: An analysis-based review in development, benefits and potential applications, Sensors 22 (21) (2022) 8174.
[5] M. Mishra, P.B. Lourenço, G.V. Ramana, Structural health monitoring of civil engineering structures by using the Internet of Things: A review, J. Build. Eng. 48 (2022) 103954.
[6] B.B. Sinha, R. Dhanalakshmi, Recent advancements and challenges of Internet of Things in smart agriculture: A survey, Future Gener. Comput. Syst. 126 (2022) 169–184.
[7] R. Hasan, M. Hossain, R. Khan, Aura: An incentive-driven ad-hoc IoT cloud framework for proximal mobile computation offloading, Future Gener. Comput. Syst. 86 (2018) 821–835.
[8] M. Laroui, H.I. Khedher, H. Moungla, H. Afifi, A.E. Kamal, Virtual mobile edge computing based on IoT devices resources in smart cities, in: IEEE International Conference on Communications, Dublin, Ireland, 2020, pp. 7–11.
[9] K. Azghiou, M.E. Mouhib, M.-A. Koulali, A. Benali, An end-to-end reliability framework of the Internet of Things, Sensors 20 (9) (2020) 2439.
[10] M.Z. Khan, O.H. Alhazmi, M.A. Javed, H. Ghandorh, K.S. Aloufi, Reliable Internet of Things: Challenges and future trends, Electronics 10 (19) (2021) 2377.
[11] K. Liu, L. Guo, Y. Wang, X. Chen, Timely reliability analysis of virtual machines considering migration and recovery in an edge server, Sensors 21 (1) (2021) 93.
[12] S.J. Moore, C.D. Nugent, S. Zhang, I. Cleland, IoT reliability: A review leading to 5 key research directions, CCF Trans. Pervasive Comput. Interact. 2 (2020) 147–163.
[13] T.A. Nguyen, D. Min, E. Choi, A hierarchical modeling and analysis framework for availability and security quantification of IoT infrastructures, Electronics 9 (1) (2020) 155.
[14] L. Xing, Reliability in Internet of Things: Current status and future perspectives, IEEE Internet Things J. 7 (8) (2020) 6704–6721.
[15] M.A. Albreem, A.M. Sheikh, M.H. Alsharif, M. Jusoh, M.N.M. Yasin, Green Internet of Things (GIoT): applications, practices, awareness, and challenges, IEEE Access 9 (2021) 38833–38858.
[16] M.H. Alsharif, A. Jahid, A.H. Kelechi, R. Kannadasan, Green IoT: A review and future research directions, Symmetry 15 (3) (2023) 757.
[17] M. Raval, S. Bhardwaj, A. Aravelli, J. Dofe, H. Gohel, Smart energy optimization for massive IoT using artificial intelligence, Internet Things 13 (2021) 100354.
[18] K.S.S. Reddy, M. Manohara, K. Shailaja, P. Revathy, T.M. Kumar, G. Premalatha, Power management using AI-based IOT systems, Meas.: Sens. 24 (2022) 100551.
[19] M.S. Aslanpour, S.S. Gill, A.N. Toosi, Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research, Internet Things 12 (2020) 100273.
[20] M. Ejaz, T. Kumar, M. Ylianttila, E. Harjula, Performance and efficiency optimization of multi-layer IoT edge architecture, in: 2nd 6G Wireless Summit, Levi, Finland, 2020, pp. 17–20.
[21] H. Guo, J. Ren, D. Zhang, Y. Zhang, J. Hu, A scalable and manageable IoT architecture based on transparent computing, J. Parallel Distrib. Comput. 118 (part 1) (2018) 5–13.
[22] S. Tang, Performance modeling and optimization for a fog-based IoT platform, IoT 4 (2) (2023) 183–201.
[23] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F.R. Yu, Z. Han, Computing resource allocation in three-tier IoT fog networks: A joint optimization approach combining stackelberg game and matching, IEEE Internet Things J. 4 (5) (2017) 1204–1215.
[24] L. Kleinrock, Queueing Systems – Volume 1: Theory, John Wiley & Sons Inc, New York, 1975.
[25] G. Bolch, S. Greiner, H. de Meer, K.S. Trivedi, Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications, Second Ed., John Wiley & Sons Inc, New York, 2006.

[26] L. Zhang, K. Li, Y. Xu, J. Mei, F. Zhang, K. Li, Maximizing reliability with energy conservation for parallel task scheduling in a heterogeneous cluster, Inform. Sci. 319 (2015) 113–131.
[27] C.-H. Cai, J. Sun, G. Dobbie, Z. Hóu, H. Bride, J.S. Dong, Fast automated abstract machine repair using simultaneous modifications and refactoring, Form. Asp. Comput. 4 (2) (2022) 1–31, Article no. 8.
[28] F.I. Dehayem Nodem, J.P. Kenné, A. Gharbi, Simultaneous control of production, repair/replacement and preventive maintenance of deteriorating manufacturing systems, Int. J. Prod. Econ. 134 (1) (2011) 271–282.
[29] G. Obradović, A.B. Strömberg, K. Lundberg, Simultaneous scheduling of replacement and repair of common components in operating systems, Ann. Oper. Res. 322 (2023) 147–165.
[30] K. Li, Quantitative modeling and analytical calculation of elasticity in cloud computing, IEEE Trans. Cloud Comput. 8 (4) (2020) 1135–1148.
[31] R.L. Burden, J.D. Faires, A.C. Reynolds, Numerical Analysis, second ed., Prindle, Weber & Schmidt, Boston, Massachusetts, 1981.
[32] K. Li, Scheduling independent tasks on multiple cloud-assisted edge servers with energy constraint, J. Parallel Distrib. Comput. 184 (2024) 104781.
[33] M. Aboubakar, M. Kellil, P. Roux, A review of IoT network management: Current status and perspectives, J. King Saud Univ. – Comput. Inform. Sci. 34 (7) (2022) 4163–4176.
[34] A. Chatterjee, B.S. Ahmed, IoT anomaly detection methods and applications: A survey, Internet Things 19 (2022) 100568.
[35] K. Gulati, R.S.K. Boddu, D. Kapila, S.L. Bangare, N. Chandnani, G. Saravanan, A review paper on wireless sensor network techniques in Internet of Things (IoT), Mater. Today: Proc. 51 (part 1) (2022) 161–165.
[36] P.K.R. Maddikunta, Q.-V. Pham, D.C. Nguyen, T. Huynh-The, O. Aouedi, G. Yenduri, S. Bhattacharya, T.R. Gadekallu, Incentive techniques for the Internet of Things: A survey, J. Netw. Comput. Appl. 206 (2022) 103464.
[37] P.K. Sadhu, V.P. Yanambaka, A. Abdelgawad, Internet of Things: Security and solutions survey, Sensors 22 (19) (2022) 7433.
[38] A. Souri, A. Hussien, M. Hoseyninezhad, M. Norouzi, A systematic review of IoT communication strategies for an efficient smart environment, Trans. Emerg. Telecommun. Technol. 33 (3) (2022) e3736.