

Article

User-Centric Internet of Things and Controlled Service Scheduling Scheme for a Software-Defined Network

Mohd Anjum ¹, Hong Min ^{2,*}  and Zubair Ahmed ³

¹ Department of Computer Engineering, Aligarh Muslim University, Aligarh 202002, India; mohdanjum@zhcet.ac.in

² School of Computing, Gachon University, Seongnam 13120, Republic of Korea

³ Department of Zoology, College of Science, King Saud University, Riyadh 11451, Saudi Arabia

* Correspondence: hmin@gachon.ac.kr

Abstract: Mobile users can access vital real-time services through wireless paradigms like software-defined network (SDN) topologies and the Internet of Things. Point-of-contact-based infrastructures and dynamic user densities increase resource access and service-sharing concurrency. Thus, controlling power consumption and network and device congestion becomes a major issue for SDN-based IoT applications. This paper uses the Controlled Service Scheduling Scheme (CS3) to address the challenge of simultaneous scheduling and power allocation. The suggested approach uses deep recurrent learning and probabilistic balancing for power allocation and service distribution during user-centric concurrent sharing intervals. The SDN control plane decides how much power to use for service delivery while forecasting user service demands directs the scheduling interval allocation. Power management is under the control plane of the SDN, whereas service distribution is under the data plane. Power-to-service requirements are evaluated probabilistically, and updates for both aircraft are obtained via the deep learning model. This allocation serves as the basis for training the learning model to alleviate power deficits across succeeding intervals. The simulation experiments are modeled using the Contiki Cooja simulator, where 200 mobile users are placed. The proposed plan delivers a 14.9% high-service distribution for various users, 18.29% less delay, 13.34% less failure, 5.54% less downtime, and 18.68% less power consumption.

Keywords: Internet of Things; software-defined networks; real-time services; concurrency; power allocation; deep learning; service scheduling



Citation: Anjum, M.; Min, H.; Ahmed, Z. User-Centric Internet of Things and Controlled Service Scheduling Scheme for a Software-Defined Network. *Appl. Sci.* **2024**, *14*, 4951. <https://doi.org/10.3390/app14114951>

Academic Editors: Carlos Enrique Palau Salvador and David Sarabia-Jácome

Received: 27 April 2024
Revised: 30 May 2024
Accepted: 3 June 2024
Published: 6 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The collaborative Internet of Things (IoT) technology venture enhances scalability and mobility support for compatible devices and users. IoT application services must correctly schedule network traffic and distribution intervals to prevent latency-less service distributions. Software-defined network (SDN) traffic classification capabilities and network virtualizations in IoT services provide more reliable scheduling. As a result, the distribution and widespread use of the IoT platform is managed without sacrificing service quality [1,2]. In addition, scheduling is performed for services customized to a given application to minimize resource stagnation and user waiting times. The SDN correctly distinguishes between user requirements, service availability, and distribution using software-based controls and control plane instructions [3]. Essential power requirements must be met for the IoT environment's gadgets to operate smoothly. As a result, power management and allocation became important study areas for IoT service deployments [4]. Power distribution, conservation, and harvesting are essential components of energy management. Based on IoT and SDN concepts, unnecessary power distribution and allocations are limited [5]. Power control and management extend their useful lives by preserving the devices' remaining energy. As a result, the device's responsiveness and widespread use are preserved,

making the smooth dissemination of IoT services more dependable [6]. Resource and user-centric power management and allocation techniques have been proposed for large-scale applications that require energy-efficient services. Consequently, IoT power management is integrated with resource distribution and allocation procedures, necessitating machine or SDN-based optimization [7].

The OpenFlow Protocol facilitates communication between the controller and the data plane in an SDN-based cloud environment. It decouples the network from individual switches and lets administrators remotely control routing tables and centralized packet-switching choices. This allows switches and data centers to be programmed independently [8]. Instead of a central server or cloud, predictions and data analysis are increasingly used at the edge to draw conclusions and take preventative measures [9]. The IoT is now heavily ingrained in our everyday lives and significantly impacts many aspects of our lifestyle, including how we drive, shop, and even decide what to eat and avoid, preserving our health. Flexible, agile, and adaptive IoT architecture is required due to the variety of applications associated with the IoT [10]. The design of an IoT architecture based on the SDN incorporates intelligent management functions by separating the control plane from the data plane to satisfy this requirement [11]. Nevertheless, most of the current studies in this sector focus on a particular area, which encourages researchers and engineers to provide comprehensive guidance on growth and direction for the future [12]. By switching from an open-loop to a closed-loop approach, the suggested strategy seeks to improve resource allocation efficiency and user experience in IoT networks by putting the user at the center of the network [13]. To improve traffic control and maintainability, the IoT application has recently included virtualization of resources and network control using software-defined networking policies. To enable the adjustments, though, the IoT and software-defined networking needs must coincide [14]. The IoT environment consists of heterogeneous devices, applications, and communication modes for aiding large-scale user-specific services. The computation, analytical, and visualization features are either built-in or acquired from different platforms [15]. Mobile customers can now receive real-time services on wireless networks thanks to the convergence of the IoT and SDN. The centralized control provided by the SDN optimizes resource allocation, guaranteeing reliable connectivity and effective communication between servers and IoT devices. Because of this integration, service delivery is enhanced by dynamically changing network characteristics in response to customer requests. The SDN also improves security via centralized management and permits control over power usage, which makes networks more adaptable, scalable, and effective. Social benefits of the proposed methods enable more personalized and efficient services for IoT users, leading to increased convenience and improved quality of life. This research can contribute to economic growth and efficiency by reducing network congestion and optimizing resource allocation, ultimately lowering operational costs for service providers and potentially stimulating innovation and investment in IoT technologies. The novel aspects of this research utilize state-of-the-art IoT and a software-defined network (SDN) that might include new algorithms, methodologies, or frameworks developed to address challenges in user-centric service scheduling within SDN environments.

The main contributions and novelty of this paper are:

1. To address the challenge of simultaneous scheduling and power allocation utilizing the Controlled Service Scheduling Scheme (CS3).
2. To use deep recurrent learning and probabilistic balancing for power allocation and service distribution during user-centric concurrent sharing intervals.
3. To train the learning model to alleviate power deficits across succeeding intervals.
4. To assess the efficacy of the proposed design by using matrices such as power consumption, service distribution ratio, delay, failures, and outages.

The remaining part of the study includes Section 2, covering the literature review; Section 3, covering the topics of the proposed study along with performance outcomes; and finally, the paper concludes in Section 4, along with future studies.

2. Literature Review

Sun et al. [16] efficiently implement resource allocation and compute offloading to reduce request time and energy usage. Reliable processing between IoT devices is approximated through computation offloading and resource allocation. The IoT-based fog-cloud environment reduces the cost and time required for this work. Consequently, the various sets of IoT-based devices experience higher energy consumption.

Iterative searching-based job offloading in multi-access edge computing for effective ultra-dense IoT devices was proposed by Guo et al. [17]. In this research, task offloading addresses the complexity of processing and transmission power. Offloading is performed on numerous edge servers to estimate power allocation efficiently. The computer task is identified for energy-conscious task offloading, and the power allocation is estimated.

The development of the delay-based allocation of workloads (DBWA) method addresses the problems of environmentally friendly and delay-guaranteed task allocation [18]. The Lyapunov drift-plus-penalty theory generates the latency guarantee in edge cloud systems based on the IoT. Enhancing green computing's energy consumption is the goal of this effort. In the IoT platform, the quality of service is preserved between the edge servers.

A dynamic microservices scheduling technique is presented for computational complexity in mobile edge computing. The network delay is reduced by increasing the number of IoT-based services utilized. Scheduling, in addition to queuing, Ref. [19] improves the pace of energy consumption. This paper aims to increase energy efficiency for the best available resources in mobile edge computing.

A multiple-input multiple-output non-orthogonal multiple access system was presented by Wang et al. [20] to convey the signal to the access control with high connectivity. This paper aims to enhance ultra-low latency and reliability. Zero-forcing beamforming, a beamforming optimization approach, is introduced for power allocation. The IoT network addresses the best local solution by optimizing the technique.

Ref. [21] describes a two-stage dimensional matching method for estimating the joint power control and time allocation platform. Here, the optimal allocation is determined to address the joint choice of channels and peer discovery. This method examines three problem-matching scenarios: linear programming, nonlinear fractional computer programming, and alternating optimization. By matching the priority-based data, this technique reduces its complexity.

A three-tier design is suggested, which raises the inquiry from the source and implements scheduling for the IoT devices. Mixed-integer programming is developed with the optimization model to shorten the deadline. The round-robin and scheduling of priorities are estimated using a genetic algorithm. The entire processing is created in [22] and assessed at the specified time.

Effective computation offloading and allocation of resources are introduced in [23] to reduce energy usage and request time. Computation offload and resource allocation are estimated for accurate processing between the IoT devices. This approach reduces the time and cost in an IoT-based fog-cloud scenario. As a result, the energy consumption of the various IoT-based devices is rising.

Singh et al. [24] proposed a systematic review examining how wireless sensor networks have evolved from simple sensory monitoring to sophisticated, service-oriented applications. By comparing various architectural styles and technological eras, it provides an understanding of how technology has advanced. It also outlines the directions for future research to direct additional creativity and investigation in this quickly developing sector.

Paulson et al. [25] suggested architecture prioritizes resource provisioning while ensuring the network's quality of service requirements are met. Jitter, latency, and throughput were compared between the SDN network strategy and the conventional method. The SDN-based IoT network increases network efficiency by lowering network overheads caused by frequent communication between the nodes and the controllers, according to the latency, delay, and throughput performance findings.

Huang et al. [26] proposed a digital twin-based user-centric resource management plan to lower resource use while increasing user satisfaction. First, the study creates a framework for resource reservations facilitated by user data transfer. Second, the study formulated a joint non-convex issue for reserving bandwidth and computing resources. To further increase user happiness, the study will look at the joint optimization of resource allocation and segment-level cache order based on distilled data from user data transfer.

Pervej et al. [27] presented a novel software-defined user-centric radio access technology solution that divides power-hungry access points into virtual cells and offers dependable, all-around connectivity to mobile-connected vehicles. Due to the heterogeneous preferences of the connected vehicles, the study introduced a preference–popularity trade-off into each content request, but also used content prefetching at the edge server with multiple classes in the content catalog to ensure fast decision making for mission-critical operations and uninterrupted onboard entertainment. The suggested method outperforms the current starting points regarding content delivery latency, deadline violation, and cache hit ratio.

Abir et al. [28] proposed a thorough analysis of software-defined unmanned aerial vehicle networks and architecture, emphasizing their needs, opportunities, and uses. In the context of 6G networking, it highlights how crucial it is to integrate the SDN and unmanned aerial vehicle-enabled networks since they provide superior flexibility, scalability, dependability, and efficient connectivity compared to traditional methods. The study also addresses practical issues with proposed networks, including security, flexibility, interference control, interoperability, and lack of standards.

Raeisi, M. and Sesay, A. B. [29] a novel mobility management feature that decreases the number of handovers in the vehicle-to-network service, hence improving the performance of 5G for high-speed road users, such as connected autonomous vehicles, in small cells. Additionally, the paper suggested a novel cell reselection process for high-speed users under our mobility management function's control using the vehicular frequency reuse scheme and in the RRC_Connected (Radio Resource Control) state. Compared to the conventional scheme, computer simulations demonstrate that the suggested vehicular frequency reuse approach can minimize the total amount of handovers (handover rate) for users by more than 99%.

Murshed, M. [30] proposed several methods for enhancing connection stability through effective decision-making during handover. First, M-FiVH, a modified probabilistic technique, was developed to improve network stability and minimize 5G handovers. Later, an adaptive learning strategy used connectivity-oriented SARSA reinforcement learning for user-centric virtual cell management to enable effective handover decisions. The study presented an analysis and comparison of several methods and showed that our suggested methods outperform the others regarding network connection.

Wenbo Wang et al. [31] suggested the Proactive Manufacturing Resources Assignment (PMRA) method for the Smart Factory to Production Performance Prediction. First, a smart factory's resources are equipped with distributed control capacity and made smart using modern IIoT and CPS technologies. In this scenario, resources at the cloud center and those at the edge may work together dynamically. A second technique is suggested for reliably predicting production status in the future by extracting real-time production information and using a real-time colored Petri net (RCPN) to analyze key production performance indicators (KPIs).

Based on the survey, there are several issues with existing models in attaining high power consumption, service distribution ratio, delay, failures, and outages. Hence, this study proposes the Controlled Service Scheduling Scheme (CS3) for controlling power consumption and network and device congestion for SDN-based IoT applications.

3. Proposed Scheduling Scheme

3.1. Problem Statement

There is a lack of an overall framework that provides all of the fundamental tools to manage end-to-end resources and traffic, and the existing solutions continue to be insufficient (they only take into account latency). Although the Internet of Things (IoT) platform is very complicated, heterogeneous, and large, there is a prominent lack of cognitive processes

that may reduce the amount of human involvement in the quality-of-service management process. A decrease in quality of service (QoS) levels in service provisioning is caused by the absence of mobility-aware latency-constrained service management at the edge. This is particularly relevant for an increase in communication latency. Because of the use of architectures that were developed for the Internet before the development of wireless technologies, service provisioning is affected negatively. At that time, most of the Internet was made up of static nodes, resulting in less fluctuation in the topology, and applications were not sensitive to delays. To reduce interruptions and latency for users of services at the edge, this research aims to examine problems and theories connected to service management. Similar to the previous mark, this research aims to reduce the need for service migrations and context transfers prompted by user mobility. This is because these events substantially influence the quality of service. Regarding addressing mobility-related difficulties at the network's edge, this research focuses primarily on using software-defined networking (SDN). Allowing for the fine-grained and active management of communication flows at the edge, the global view of network entities made possible by this networking paradigm may make it easier to implement innovative solutions.

Real-time service handling is provided to mobile users in the IoT environment by deploying the SDN. Here, concurrent service sharing is estimated for dynamic users on heterogeneous platforms. This work determines a point-of-contact-based infrastructure for concurrent processing in SDN-based IoT applications. The scope of this work is to avoid latency, failure, and outage by increasing the service distribution to the valid user. Thus, concurrent processing is estimated for the mobile devices in IoT applications concerning the SDN. A formal representation of the CS3 is presented in Figure 1. The simulation experiments are modeled using the Contiki Cooja simulator, where 200 mobile users are placed. An IoT cloud architecture with seven service providers, each with 1 TB of storage, is used for service distribution. The transmit power of the devices is set as 30 dBm, and it operates in 60 scheduling intervals.

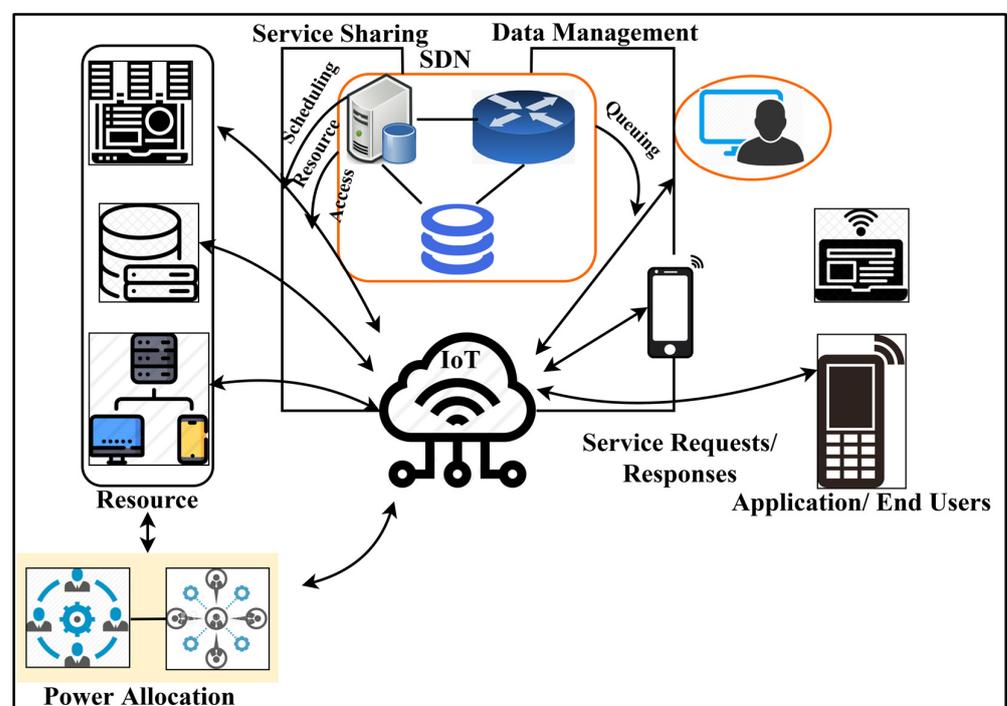


Figure 1. Representation of the proposed CS3.

Figure 1 presents the proposed scheme in the IoT-SDN platform. The SDN is responsible for service scheduling and power management using this scheme's data and control plane. It operates between the users and resources throughout scheduling, queuing, and

distribution processes. Moreover, the power allocation for the resources is governed by scheduling and prior distribution.

3.2. Problem Definition

This work addresses scheduling and power allocation in SDN-based IoT applications. This is resolved by introducing the CS3 in this article. In this case, the user estimates the cumulative congestion and power management. The power allocation and service distribution are processed to the network and analyzed by the scheduling approach. The point-of-contact-based infrastructure is used to derive the congestion-free service transmission.

Joint scheduling and power allocation are addressed here by proposing probabilistic balancing and deep recurrent learning. In this work, the SDN is derived for the service distribution, whereas the control plane indicates power management. In this evaluation step, the service is shared by deploying the concurrent processing. The following equation derives the service sharing that deploys power management and service handling.

$$\mu(h_a) = \left(\frac{r_0}{\sum_{v_0}^M s} \right) \times (s_u + i_d) \times \left(\frac{N + r_n/X}{a_0} \right) + \prod_{r_0} [(a_0 \times b') - e_t] \times c_k \quad (1a)$$

The determination is made for service sharing between the user and the devices in SDN-based IoT applications. Here, the problem definition is addressed by estimating the congestion in the data transfer to the other devices. In this computation step, the concurrent processing is derived that deploys service sharing and resource allocation. In this evaluation step, concurrent processing is carried out for the service distribution among the dynamic user densities and point-of-contact-based infrastructure. Here, the service sharing is performed for the valid user by deploying the reliable distribution to the device in an SDN network-based IoT application. The service sharing is performed for the other devices in the network, and it is represented as h_a ; the determination is denoted as μ . In this process, the distribution is estimated for scheduling.

Scheduling is performed for the devices in the IoT application termed as S . In this processing step, congestion is addressed and decreased, and it is represented as N ; the power allocation is estimated for the user and the device is termed as a_0 . The user and the number of users in the network are defined as r_0 and r_n , and the device and number of devices are represented as v_0 and v_n , respectively; service distribution is termed as i_d . The access is provided to a valid user, and it is denoted as c_k and V ; balancing is performed for the data transmission, and it is represented as b' . The time is calculated for a better examination of data analysis, referred to as e_t ; an examination is performed for the congestion in data forwarding, and it is denoted as X . The resources in the network are estimated for the sharing represented as s_u ; the power management is denoted as M . The following equation determines resource access for the requested user-centric method.

$$\mu(c_k, s_u) = \prod_{V}^{r_0} (N \times X) + v_n \times \left(\frac{S}{a_0 + r_0} \right) + (h_a(V) + r' - e_t) \quad (1b)$$

Resource access is provided to the valid end-user, who deploys concurrent service sharing in the IoT application. Here, the SDN is used to deploy service sharing, and power allocation is estimated for reliable data computation. The transmission is carried out for the network devices and estimates the scheduling for the user. In this computation step, power management is performed to derive the concurrent service handling from the end-user. The examination is conducted for the service distribution and power allocation method for the end-user, and it is denoted as r' . The valid user accesses the information by forwarding it from the power management system.

Scheduling is performed to estimate the better processing of devices in the SDN and provides the power allocation for further processing. Here, the resource is derived for the requested user and the balancing access is estimated. The power management decides whether the access is forwarded if the user requests a particular service. Based on this

process, the access is forwarded to the required user and the scheduling is evaluated as $v_n \times \left(\frac{S}{a_0+r_0}\right)$. Sharing is performed by posting to Equation (1a); resource access is shown in Equation (1b). Scheduling is performed based on the queuing technique discussed in the section below from these two equations.

3.3. Scheduling

Scheduling is based on the user and device requests for power in the SDN. In this derivation, if power is estimated for reliable processing, then the point-of-centric-based infrastructure is created for resource sharing. Resource sharing is conducted by deploying the scheduling with the existing and the current state of the process. The task and the power allocation are queued based on the usage and congestion in the network. This method performs scheduling by deploying the SDN-based IoT application, which is determined for better service distribution. The following equation estimates the scheduling process for the concurrent service handling in the SDN.

$$S = \left. \begin{aligned} &(h_a \times r_0/a_0) + \sum_{v_0} (M + i_d) + q_0 \times r' - e_t \\ &a_0 = (r_0 \times i_d) + (r' - e_t) \\ &i_d = r' - (h_a + N) \\ &h_a = r'(w_c) + M \end{aligned} \right\} \tag{2}$$

Scheduling is performed by deploying the queuing process, which deploys resource sharing to the end-user at the mentioned time interval. Here, the access is forwarded to the valid user in the IoT network, who estimates the reliable processing by handling the data. Power management is conducted to distribute the appropriate data to the appropriate resources in the SDN. The end-user is responsible for forwarding the service based on valid access generation. If the access generated is valid, the service is forwarded to the requested user at the mentioned time interval. In this computation step, the queueing is conducted by allocating the service to the devices denoted as q_0 ; forwarding the service is represented as w_c . The scheduling process is illustrated in Figure 2.

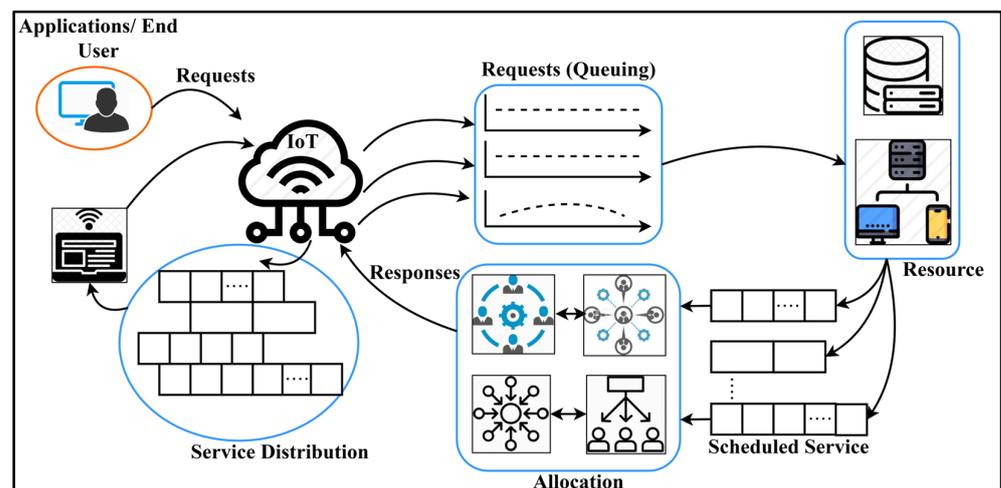


Figure 2. Scheduling process illustration.

Queueing is performed for the number of resources in the IoT application and deploys the power management system. The power allocation balances the service between the user and the end-user. From this evaluation step, scheduling is carried out by determining the sharing of services, and it is represented as $(h_a \times r_0/a_0)$. In this evaluation step, service sharing to the end-user is conducted by examining SDN access. The point-of-contact-based infrastructure distributes the number of end-user requests from the devices. Here,

scheduling is carried out by indicating the queuing process for the valid user in the SDN; this power allocation is performed in the equation below.

$$a_0 = \begin{cases} \prod_{s_u} w_c + X \times \left(q_0 + \frac{r_0(D)}{V} \right) \times S \\ = S \times r_0 + \left(\frac{v_0 \times r' / M}{X} \right) \times (q_0 - e_t) \end{cases} \quad (3)$$

In Equation (3), scheduling is performed by deploying appropriate service handling to the end-users. In this computation step, the user is used to define the congestion in the data transmission. Here, the queuing is conducted for the scheduling devices and determines the detection process, and it is represented as D . Queuing is performed for the power allocation process that deploys power management and estimates better processing. Here, scheduling is conducted for the number of devices in the SDN used to evaluate better detection. In this derivation, power allocation is examined for resource sharing to the end-user, and the detection of congestion in the SDN is termed as D .

ScheduleSDNPower manages power scheduling and allocation within an SDN using Algorithm 1. It queues service requests, verifies device access, and computes scheduling parameters based on several parameters. Using a sophisticated formula incorporating forwarding, congestion detection, and service sharing, the power distribution depends on the congestion level. Network performance is optimized by the algorithm's effective resource allocation management. Ultimately, it provides a formal SDN energy administration and scheduling method by returning the planned power allocation.

Algorithm 1: Scheduling

Function ScheduleSDNPower ($h_a, r_0, a_0, v_0, M, i_d, q_0, r', e_t, w_c, X, S, D$)

Input:

- h_a : Service sharing parameter
- r_0 : Initial service rate
- a_0 : Initial allocation parameter
- v_0 : Number of devices
- M : Power management parameter
- i_d : Device index
- q_0 : Service allocation queue
- r' : Optimized service rate
- e_t : Processing time
- w_c : Service forwarding parameter
- X : Power allocation parameter
- S : Scheduling parameter
- D : Congestion detection parameter

Output: Scheduled power allocation

Step 1: Calculate scheduling parameters

$$\begin{aligned} h_a &\leftarrow r'(w_c) + M; \\ a_0 &\leftarrow (r_0 * i_d) + (r' - e_t); \\ i_d &\leftarrow r' - (h_a + N); \\ S &\leftarrow (h_a * r_0 / a_0) + \sum_{v_0} (M + i_d) + q_0 * r' - e_t; \end{aligned}$$

Step 2: Validate and queue device requests

for each device in SDN, do

if device access is valid, then

Forward service to the requested user on the mentioned time interval;

$q_0 \leftarrow$ Queue service allocation;

$w_c \leftarrow$ Power allocation;

end if;

end

Algorithm 1 Cont.

```

Step 3: Calculate power allocation
    if there is congestion in end-user service, then
         $a_0 = \prod_{s_i} w_c + X * (q_0 + \frac{r_0(D)}{V}) * S$ 
    else if sharing resources when detecting congestion.
         $a_0 = S * r_0 + (\frac{v_0 * r' / M}{X}) * (q_0 - e_t)$ 
    end else if
    end if
return scheduled power allocation
end function

```

Queuing is performed to allocate power for the device to work in concurrent processing. Here, the resources (services) are distributed to the end-user. In this computation step, forwarding is conducted by deploying data that avoids congestion in the SDN. The congestion is addressed in this scheduling method, and it is associated with the queuing of resources. Queuing is examined for the end-user at the mentioned time, and it is represented as $(\frac{v_0 * r' / M}{X}) * (q_0 - e_t)$. Thus, power allocation is performed concerning the scheduling approach; after this process, the proposed work includes two methodologies: probabilistic balancing and recurrent learning. The following section discusses these two methodologies in detail.

3.4. Probabilistic Balancing

This method balances the service with the appropriate end-user by determining whether scheduling is carried out appropriately. This constraint is achieved by examining the tree-based computation that deploys either a probability factor or is used to estimate the detection. Balancing is conducted for the service distributed to the appropriate user, and the validity of the devices is monitored. The access forwarded to the appropriate user is defined in this category and examined in the equation below. Here, queuing is performed to balance the resources.

$$\rho = \frac{q_0[(S + r_0) \times (N - M)]}{r_n + v_n} \quad (4)$$

In this case, insertion and deletion are performed to balance the resource allocation method; here, it deploys better service detection in the SDN. Here, probabilistic balancing is examined for reliable computation in the tree-based process, and the root node represents the service from the power management. The left-side node indicates power allocation, whereas the right node represents sharing. Based on this, probabilistics is performed by evaluating the better detection of congestion in the SDN. Power management distinguishes service handling, and balancing is examined in this process. The examination process indicates cumulative congestion in the network and devices and estimates better power management. Figure 3 presents the queuing process of the CS3.

Power management is responsible for transmitting power to the appropriate end-user by deploying power allocation and service distribution. Power allocation is conducted for dynamic user densities and estimates congestion to avoid further failure. Here, scheduling is performed for the power allocation in this probabilistic balancing based on the tree structure, which indicates the point-of-contact-based infrastructure. Thus, probabilistic balancing is conducted for data forwarding in the network from the user to the devices, and it is termed as ρ . The following part discusses recurrent learning for power allocation in the management system.

The probabilistic balancing function uses a tree-based methodology to control service distribution and power allocation in the SDN, as shown in Algorithm 2. First, it creates a tree structure, with the root node representing power management services, the left-side representing power distribution, and the right-side branch representing service sharing. The function iterates through the tree for every service request in the queue, sharing or allocating

power according to the nodes it encounters. Throughout this process, it assesses congestion detection to modify sharing and allocations. SDN systems ensure balanced resource use and effective network functioning by appropriately allocating power and other services to devices.

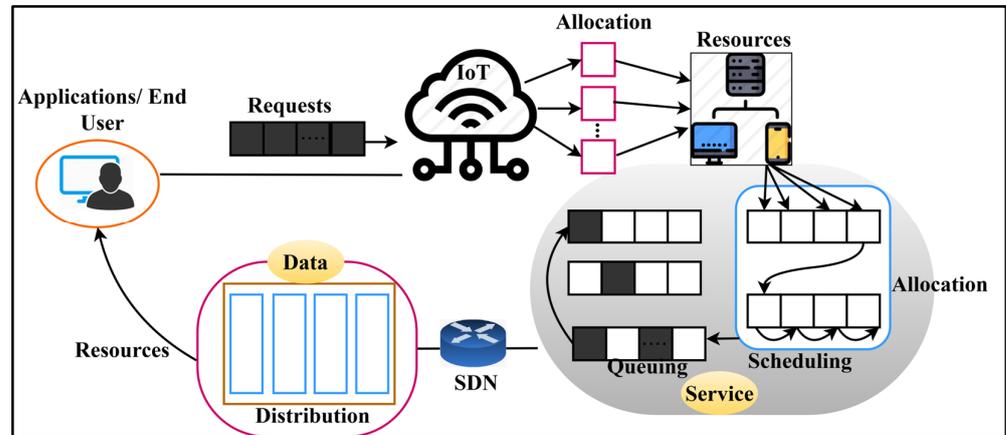


Figure 3. CS3 queuing process.

Algorithm 2: Probabilistic Balancing for SDN Data Transmission

Function ProbabilisticBalancing ($q_0, r_0, S, N, M, r_n, v_n$)

Input : $q_0, r_0, S, N, M, r_n, v_n$

Output: ρ (probability factor)

Step 1: Probabilistic Balancing

$$\rho \leftarrow \frac{q_0[(S+r_0)*(N-M)]}{r_n+v_n}$$

Step 2: Construct a probabilistic balancing tree

$$\text{tree} \leftarrow \text{ConstructProbabilisticTree}(q_0)$$

Step 3: Perform probabilistic balancing

for each flow in flow_demands do

 root \leftarrow tree.root

 while not leaf_node(root)

 end

Step 3: Evaluate congestion on the left and right child nodes

$$\text{left_congestion} \leftarrow \text{EvaluateCongestion}(\text{root.left_child}, \text{balanced_flows})$$

$$\text{right_congestion} \leftarrow \text{EvaluateCongestion}(\text{root.right_child}, \text{balanced_flows})$$

Step 4: Compute probabilities based on congestion levels

$$\text{left_probability} \leftarrow 1 - \text{right_congestion}/(\text{left_congestion} + \text{right_congestion})$$

$$\text{right_probability} \leftarrow 1 - \text{left_congestion}/(\text{left_congestion} + \text{right_congestion})$$

Step 5: Probabilistically choose the next node to traverse

if RandomChoice(left_probability) do

 root \leftarrow root.left_child

else

 root \leftarrow root.right_child

endif

Step 6: Update flow on the leaf node

$$\text{UpdateFlow}(\text{root}, \text{flow}, \text{balanced_flows})$$

return balanced_flows

end function

3.5. Recurrent Learning

Service distribution and power allocation are estimated by introducing recurrent learning to the user-centric concurrent sharing interval. Here, power management is used to deploy congestion-free transmission and evaluate sharing with the end-user on time. The prediction is achieved by mapping the existing process and deploying resource access for service shar-

ing. Training data determine the input neuron and perform an efficient distribution in this recurrency phase. In this processing, the input is the queued process, and it enters the initial neuron state, from the output of the first neuron forward as the input to the second neuron. Thus, the service delivery is estimated for the power allocation and addresses the deficiency. The equation below is used to analyze the current state of neuron processing.

$$k_0 = u(p_i) + r_a(S + q_0) \times o_0 \tag{5}$$

In Equation (5), the current state represents the service distribution by deploying resource access associated with power allocation. Here, the queued process transmits to the next neuron layer, followed by the number of devices and users. Sharing is performed on time for the valid user based on a queued process. Scheduling is performed by deploying network and device cumulative congestion and power management. The first state indicates resource sharing, and from this, it is forwarded to the next neuron layers in recurrent learning. Here, service delivery is conducted to allocate power and deploy distribution. The neuron layer is denoted as o_0 ; the function is used to define the current state, and it is represented as u and k_0 .

The current state defines the number of neuron layers in recurrent learning that deploys joint scheduling, and power allocation is addressed. Based on the scheduling process, the analysis is performed for the current neuron state; resource access is forwarded to the second neuron layers. The dynamic user densities are due to the concurrency in service sharing and resource access in SDN-based IoT applications. The hidden layer improves training data, concurrent service sharing, and resource access. The following Equation is used to derive the hidden layer, and here, two hidden layers are used to enhance service distribution efficiently.

$$\nabla_1 = \begin{cases} o_0 = (S + r_0) \times \left(\frac{M+i_d}{q_0+v_0} \right) + b' - e_t \\ o_1 = (S + r_1) \times \left(\frac{M+i_d}{q_0+v_1} \right) + b' - e_t \\ \vdots \\ o_n = (S + r_{n-1}) \times \left(\frac{M+i_d}{q_0+v_{n-1}} \right) + b' - e_t \end{cases} \tag{6a}$$

$$\nabla_2 = \begin{cases} o_0 = q_0 \times \left(\frac{a_0}{r_0} \right) + \sum_{k_0} v_0 \times c_k(r_0) \\ o_1 = q_0 \times \left(\frac{a_0}{r_1} \right) + \sum_{k_0} v_1 \times c_k(r_1) \\ \vdots \\ o_n = q_n \times \left(\frac{a_0}{r_{n-1}} \right) + \sum_{k_0} v_{n-1} \times c_k(r_{n-1}) \end{cases} \tag{6b}$$

In Equation (6a), the first hidden layers deploy the scheduling for the number of users in an SDN-based IoT application. Here, power management is used to determine the current layers in neuron layers, and balancing is achieved for resource access at the mentioned time. The neuron layer indicates scheduling for the number of users and deploys resource access based on power allocation. Concurrent service sharing and resource access are provided to the appropriate user. Here, the hidden layer enhances the training data for access to resources in the SDN. Scheduling and power allocation are conducted for resource access by performing service sharing. In Figure 4, the recurrent layer process is illustrated.

Figure 4 illustrates an RNN; it should typically show nodes connected in a way that indicates temporal or sequential processing. This might include feedback loops or connections where the output from one time step is used as the input for the next. The first neuron layer holds the power management, and balancing is examined at the mentioned time for the number of neurons, and it is denoted as o_n . Scheduling is conducted for the number of resources that determine service distribution on the network. This evaluation

step is used to deploy concurrent service sharing and resource access due to the dynamic user densities and point-of-contact-based infrastructure. Equation (6b) represents the second hidden layer that determines access to the end-user. The first layer indicates the queuing of the power allocation process, and balancing is conducted for resource access. The first hidden layer output is given as the input to the second neuron layer and processed to the number of neuron layers in the SDN.

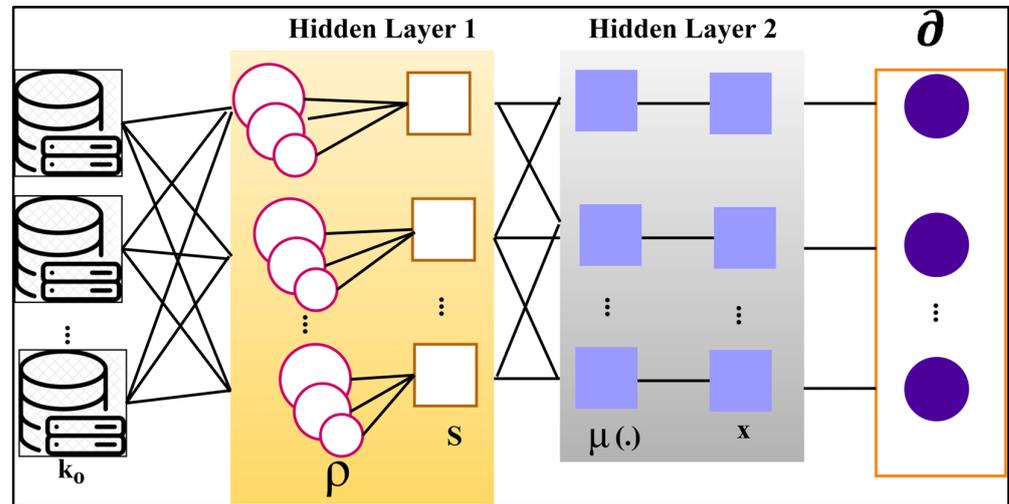


Figure 4. Recurrent layer process.

From the first hidden layer, the data are forwarded to the second hidden layer’s first neuron and deploy the current state of the neuron. Here, the neuron’s current and the input state determine resource access and evaluate the better processing. In this computation step, congestion is addressed and decreased concerning the scheduling process, which results in a better training phase. The neuron layers initiate the training in the recurrent learning method, which estimates the concurrent sharing and access provided. The first and second hidden layers are denoted as ∇_1 and ∇_2 , respectively. The training data are enhanced from this hidden layer by equating the following equation:

$$X = \left. \begin{aligned} & \prod_{r_0}^N (i_d \times b') + \left(\rho \times \frac{q_0 + r_n + v_n}{M+S} \right) - e_t \\ & b' = (r_0 \times w_c) + (c_k \times M) \\ & c_k = (r_0 + V) - (M - e_t) \\ & V = [c_k + (w_c * v_n)] + b' \end{aligned} \right\} \quad (7)$$

The above equation examines the training data that deploy the better detection of resources, and access is forwarded to the appropriate end-user on time. Here, the access is forwarded to the valid user in the network, which determines the concurrent processing and estimates scheduling and power allocation. Power allocation is performed for the different resources and deploys probabilistic balancing for the access and sharing of services. The derivation indicates the hidden layer process to enhance the training data for balancing resource sharing and estimating service distribution. Power management performs the allocation, which includes training for successive intervals.

The training data are determined for reliable sharing and service distribution to the end-user, and the power allocation is examined. The concurrency in service sharing and resource access is due to the dynamic user densities and point-of-contact based on the heterogeneous platform. Forwarding is performed to derive the balancing factor from the successive intervals of data processing. In this computation step, the prior state defines the current processing state. The evaluation is performed for the training data estimated from the hidden layers, and the queuing for valid access to resources is performed. The following equation distributes the service to the end-user, including power management:

$$\partial = \left(\frac{h_a + c_k}{s_u/v_0} \right) \times \sum \left[(k_0 + V) \times \left(\frac{\prod c_k (w_c + p_i)}{g_t} \right) \right] + o_0 \times (\nabla_1 + \nabla_2) \quad (8a)$$

The service distribution analyzes service distribution to the appropriate user better and deploys efficient processing. Here, congestion is addressed for power management, and scheduling for balancing data is estimated. Here, probabilistic balancing is performed for resource sharing, and the current state is used to define the prediction. Queuing is performed on the neuron layer, estimates the sharing of resources, and maintains power. The deficiency is maintained for service forwarding, and the hidden layers are estimated to improve the training data. The determination is achieved by examining the validation process and deploying the prior state.

The training state is represented as g_t and the prior state of processing in the neuron layer is denoted as p_i . Thus, the hidden layers are responsible for forwarding the data to the appropriate user by determining the resource sharing to estimate better sharing and resource access in IoT applications. The evaluation is represented as ∂ and avoids congestion for the scheduled resources in the SDN. The following equation is used to evaluate power management for the training data, and allocation is achieved for the training for the successive interval of resource processing.

$$M = \left. \begin{aligned} &(\nabla_1 + \nabla_2) \times \left[\left(\frac{c_k/g_t}{s} \right) + (v_0 \times i_d) \right] \times b' \\ &g' = (\nabla_1 + \nabla_2) \times p_i \\ &p_i = \gamma \times q_0(v_n) + k_0 \\ &\gamma = p_i + w_c * (h_a + r_n) \end{aligned} \right\} \quad (8b)$$

Power management is performed on the control plane by deploying the SDN in an IoT-based application. Here, power is provided to the required devices concerning the prediction state, γ , which determines the training data for reliable processing. Figure 5 illustrates the power management process of the proposed CS3.

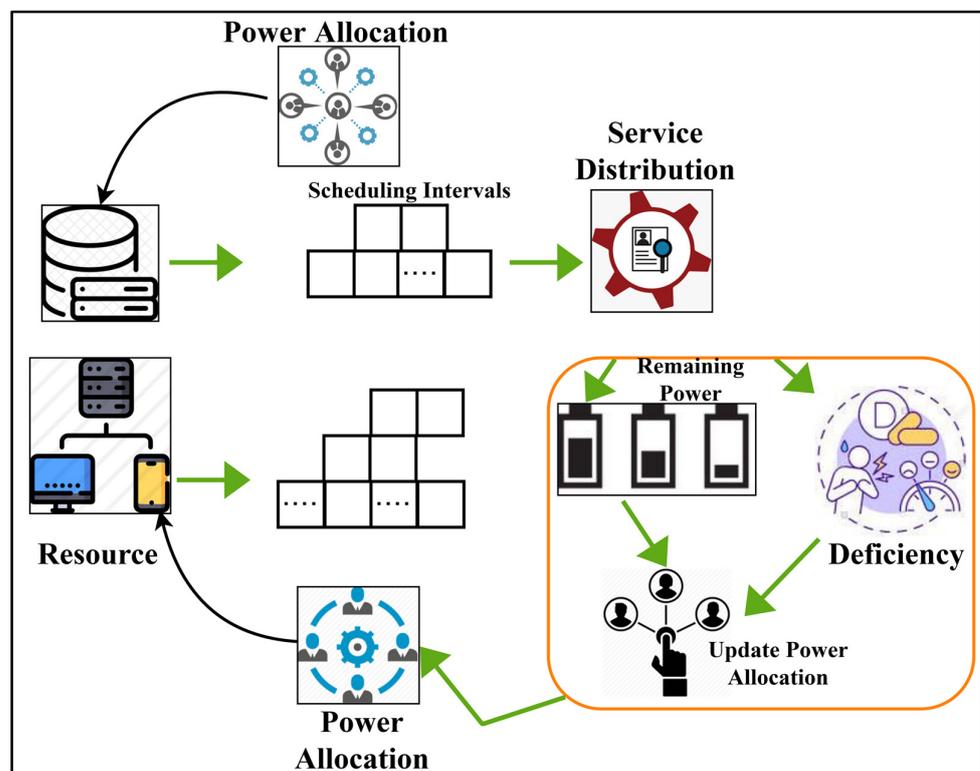


Figure 5. CS3 power management process.

This computation step examines queuing and scheduling to address cumulative congestion and power management. Service sharing is evaluated in the SDN for power allocation and service distribution. Balancing is achieved for the sharing and access provided to the appropriate device at the mentioned time. The power deficiency is addressed and decreased by performing training that deploys successive intervals in the network. Here, resource access is provided to the network and devices, and congestion is estimated. The following equation is used to predict the power usage of the prior device and prevent deficiency:

$$\delta = \left(\frac{1}{v_n + r_n} \right) \times \prod_{c_k + w_c}^S (a_0 + g_t) \times \left[\left(\frac{i_d/M}{\partial + s_u} \right) + (b' \times p_i) \right] - e_t \quad (9)$$

The analysis is carried out to predict and deploy access to the appropriate devices and users. Here, balancing is performed for the power management system, and power allocation for the trained data is deployed. Balancing is estimated for the prior and current states of the neuron layer and estimates the forwarding. The hidden layer is responsible for forwarding the data to the other neuron state, deriving the power allocation, and preventing deficiency. The analysis for the prediction is examined in the equation above, and it is denoted as δ . The equation below integrates joint scheduling and power allocation to improve power utilization and decrease latency:

$$\delta(z_i) = \int_{r_n}^{v_n} (S + a_0) \times k_0 + o_0 + \dots + o_n \times h_a + \left(V(r_0) \times \frac{X + s_u}{s_u} \right) + \gamma - e_t \quad (10)$$

The analysis for utilization is examined in Equation (10); here, scheduling-related queuing is performed by deploying power management and resource allocation. In this computation step, the prediction is performed by mapping with the existing resources and providing relevant information sharing. The service distribution is performed for the trained and valid resources posted to the hidden layer processing. The analysis defines the current and next states of the neuron and estimates the reliable process that integrates scheduling and allocation and addresses the deficiency.

From this processing, latency and outage are addressed and decreased by improving power utilization, whereas deficiency is addressed. Based on the allocation, training is performed on time for successive intervals. This objective is addressed by introducing the CS3, which includes joint scheduling and power allocation. Thus, the SDN-based IoT application resolves the cumulative congestion of networks and devices and manages power. Access is provided on time in this processing concurrency in service sharing and resources. In Table 1, the failed requests in different scheduled intervals are presented.

In the SDN, Algorithm 3 shows the recurrent learning function that improves routing choices and flow conditions. Routing decisions and flow demands are initialized. It performs data transmission simulation, assesses network performance, modifies routing choices depending on metrics for performance, and modifies flow matrices iteratively. This recurrent process iterates until the maximum number of provided iterations is reached. In an SDN setting, the function produces optimal flow and route matrices that facilitate effective data transfer and resource allocation.

Algorithm 3: Recurrent learning

Function to perform Recurrent Learning and Power Allocation

Function RecurrentLearningPowerAllocation

$(r_0, S, i_d, q_0, o_0, \nabla_1, \nabla_2, v_n, r_n, g_t, p_i, c_k, w_c, \gamma, h_a, X, e_t)$

Input: $r_0, S, i_d, q_0, o_0, \nabla_1, \nabla_2, v_n, r_n, g_t, p_i, c_k, w_c, \gamma, h_a, X, e_t$

Output: Power allocation, Service dissemination, Deficiency, Service distribution

Step 1: Recurrent Learning Phase

$$k_0 \leftarrow u(p_i + r_a(S + q_0) * o_0)$$

Algorithm 3 Cont.

Step 2: First Hidden Layer Calculation

Initialize gradient array
 for each service rate r_i in r do
 calculate gradient o_i

$$o_i \leftarrow (S + r_1) * \left(\frac{M + i_d}{q_0 + v_1} \right) + b' - e_t$$

 add o_i to the gradient array
 return gradient array

Step 3: Second Hidden Layer Calculation

Initialize gradient array
 for each index i from 0 to $n - 1$ do
 Initialize sum = 0
 for each coefficient c_k do
 sum \leftarrow sum + $v[i] * c_k * r[i]$
 calculate gradient o_i

$$o_i = q[i] * \left(\frac{a_0}{r[i]} \right) + \text{sum}$$

 add o_i to the gradient array
 return gradient array

Step 4: Power Management

Initialize $M = 0$
 Initialize $g' = 0$
 Initialize $\gamma = 0$
 for each index i from 0 to $n - 1$ do
 calculate g'

$$g' = (\nabla_1 + \nabla_2) * p_i$$

 calculate γ

$$\gamma = p_i + w_c * (h_a + r_n),$$

 calculate M

$$M \leftarrow (\nabla_1 + \nabla_2) * \left[\left(\frac{c_k / g_t}{S} \right) + (v_0 * i_d) \right] * b' + g' + \gamma * q_0(v_n) + k_0$$

 return M

Step 5: Prediction and Analysis

Initialize $\delta = 0$
 // Calculate the product of the terms inside the sigma

$$\text{sigma_product} = 1$$

 for each coefficient c_k in c do

$$\text{sigma_product} = \text{sigma_product} * ((a_0 + g_t) * ((i_d / M) / (\text{partial_derivative} + s_u)) + (b' * p_i))$$

 // Calculate the δ value

$$\delta = \left(\frac{1}{v_n + r_n} \right) * \text{sigma_product} - e_t$$

 return delta

Step 6: Service Dissemination Calculation

Initialize $\text{partial_derivative} = 0$
 for each coefficient c_k in c do
 // Calculate the inner sum

$$\text{inner_sum} = 0$$

 for each coefficient c_k in c do

$$\text{inner_sum} = \text{inner_sum} + (w_c + p_i)$$

 // Calculate the term inside the parenthesis

$$\text{term_inside_parenthesis} = \frac{h_a + c_k}{v_0} * ((k_0 + V) * (\text{inner_sum} / g_t))$$

 // Add the contribution of each term to the partial derivative

$$\text{partial_derivative} = \text{partial_derivative} + \text{term_inside_parenthesis}$$

 // Add the contribution of $o_0 * (\nabla_1 + \nabla_2)$ to the partial derivative

$$\text{partial_derivative} = \text{partial_derivative} + o_0 * (\nabla_1 + \nabla_2)$$

 return $\text{partial_derivative}$

Step 7: Output Results

Return ∂, δ

End Function

Table 1. Failed requests in scheduling intervals.

Scheduling Intervals	Sharing Intervals	Queued Requests	Service Dissemination	Failed Requests
10	11	23	0.96	0
20	19	25	0.92	19
30	15	20	0.94	12
40	25	75	0.93	20
50	38	98	0.91	25
60	49	103	0.87	36

In discussing the performance of the scheduling system, Table 1 provides crucial insights into the system's behavior across various scheduling intervals. The table presents data on queued requests, service dissemination success rates, and the number of failed requests, which are vital metrics for evaluating the effectiveness of the scheduling algorithm. The sharing interval is determined for the varying scheduling intervals for the queued request and increases if the shared interval increases. Suppose the queued request decreases and the service dissemination increases for varying intervals. In another case, if the queued request decreases, the failure request also decreases (Table 1). The iterates utilized at different sharing intervals are tabulated in Table 2.

Table 2. Training iterates for sharing intervals.

Sharing Intervals	Power Requirement (J)	Power Allocation (J)	Failed Requests	Training Iterates
10	80.37	30.1	39	107
20	80.54	40.58	34	98
30	78.14	46.34	32	75
40	98.36	54.25	29	68
50	102.87	60.48	22	38
60	107.98	75.37	20	41
70	111.25	80.69	18	39
80	118.8	91.33	12	23

The goal of the suggested method is to minimize latency while ensuring adequate power distribution to every user. Allocated power is computed based on an optimization model that distributes the total available power and total among users to reduce the overall latency. Required power signifies the theoretical power necessary to achieve a target latency for every user or sharing interval. The sharing interval is estimated for the power requirement and allocation and shows less processing than the power requirement. If the power allocation decreases, the failed request from the user is enhanced. The training iterates define the sharing intervals and increase if the failed request increases (Table 2). In Table 3, the service dissemination factor for different factors is presented.

Table 3. Service dissemination for different factors.

Users	Power Requirement (J)	Predicted (J)	Deficiency (J)	Service Dissemination
40	80.37	86.5	-6.13	0.87
80	82.36	93.58	-11.22	0.89
120	92.47	86.74	5.73	0.96
160	102.58	119.47	-16.89	0.91
200	118.8	126.3	-7.5	0.94

User density determines power requirement, and it is enhanced appropriately. The prediction is made by mapping the current and prior service and range increases. If the prediction increases, the deficiency decreases, and service dissemination is performed for the different types of users in the network. Service dissemination also increases if the deficiency is higher (Table 3). The article suggests the CS3, an advanced SDN resource management method. By integrating the SDN's adaptable control and data plane operations with deep learning's predictive capabilities, the CS3 efficiently handles high-user-request densities while minimizing latency. Network performance and customer satisfaction are enhanced by integrating these technologies, which provide efficient and resilient power allocation and service scheduling.

4. Performance Assessment

This sub-section assesses the proposed CS3's performance through comparative analysis. The simulation experiments are modeled using the Contiki Cooja simulator, where 200 mobile users are placed. An IoT cloud architecture with seven service providers, each with 1 TB of storage, is used for service distribution. The transmit power of the devices is set as 30 dBm, and it operates in 60 scheduling intervals. With these simulation details, the performance is assessed using service dissemination factor, latency, service failure ratio, outage, and power utilization metrics. In the comparative analysis, the Iterative Searching-based Algorithm for Task Offloading and Transmit Power Allocation (ISA-TOTPA) [17], Energy and Time-Efficient Computation Offloading and Resource Allocation (ETCORA) [16], and DBWA [18] methods are used. This study extended our simulation experiments using the Contiki Cooja simulator to include a broader range of scenarios. These scenarios now encompass different numbers of mobile users (50, 100, 200, and 300), varying mobility patterns, and diverse traffic conditions. This comprehensive evaluation aims to provide a more detailed understanding of the system's performance under varying circumstances. Our results indicate that our approach consistently outperforms others regarding latency and power allocation. Moreover, extended-duration tests reveal the long-term stability of our system.

4.1. Service Dissemination

The service dissemination for the proposed work is high for varying users and scheduling intervals. Here, power management determines appropriate user forwarding at the mentioned time. Power allocation is estimated for probabilistic balancing and examines the training data for the successive interval, and it is represented as $(s_u + i_d) \times \left(\frac{N + r_n/X}{a_0}\right)$. Service distribution is performed by evaluating IoT sharing and resource access. The computation is conducted on the SDN-based IoT application, ensuring valid access. Sharing is derived from the number of users in the network and better deploys power allocation. The current state estimates scheduling based on the queuing process and deployed probabilistic learning. Here, service utilization is performed for service sharing, and resource access is achieved for the training data. In this case, prediction is performed for the prior and current states and provides access to the required resources. Sharing is conducted by determining the point-of-contact-based infrastructure that performs concurrent service sharing and resource access. Scheduling is performed to initialize devices and power allocation is carried out during this process (Figure 6). The different fluctuations in the service dissemination comparison curve in Figure 6 likely result from a combination of factors related to network dynamics, resource management, algorithm efficiency, user behavior, and environmental conditions.

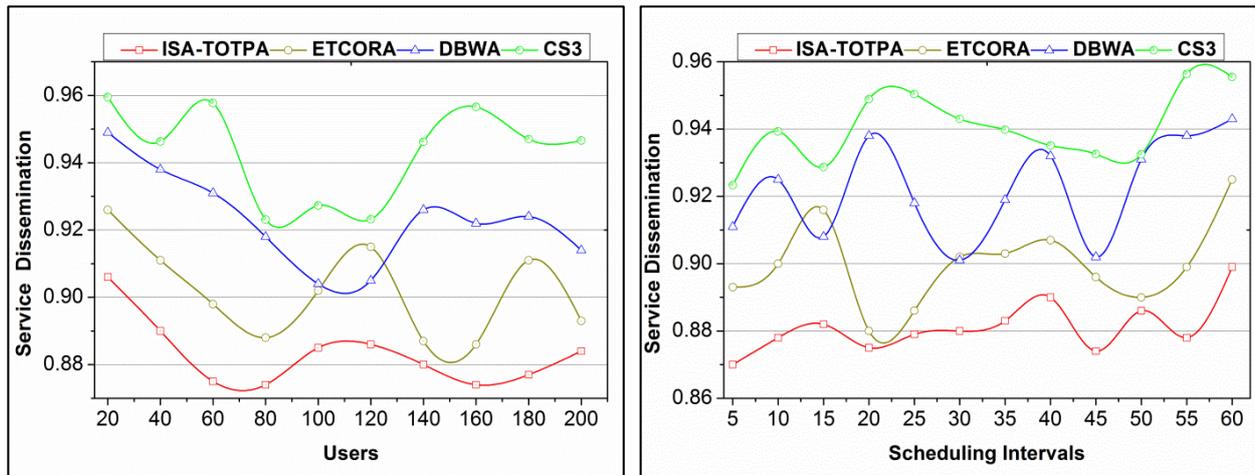


Figure 6. Service dissemination comparisons.

4.2. Latency

Latency decreases concerning user and scheduling intervals, which deploys the service distribution. Here, scheduling and queuing are derived for the control data plane in the network, indicating power management. Based on the scheduling process, queuing is performed for the resources and valid access is forwarded on time. The power allocation is performed for probabilistic balancing, and it is represented as $X \times \left(q_0 + \frac{r_0(D)}{V} \right)$. Concurrent service sharing and resource access determine the neuron state and deploy the hidden layers. The analysis is estimated for service distribution for the numbers of users and deploys the sharing efficiently. The training devices are used to deploy the scheduling for power management and determine the current state. The predictions are made for resource access, estimated power allocation, and derived scheduling. Joint scheduling is performed for service distribution to determine the SDN-based IoT application. Here, service sharing is conducted by determining the neuron’s current state and evaluating the probabilistic balancing. Integration is achieved for joint scheduling and power allocation and determines better processing. The hidden layer derives reliable service sharing and addresses cumulative congestion and power management (Figure 7). Figure 7 shows latency over different scheduling intervals, and a smaller curve for the CS3 indicates consistently lower latency across all intervals tested.

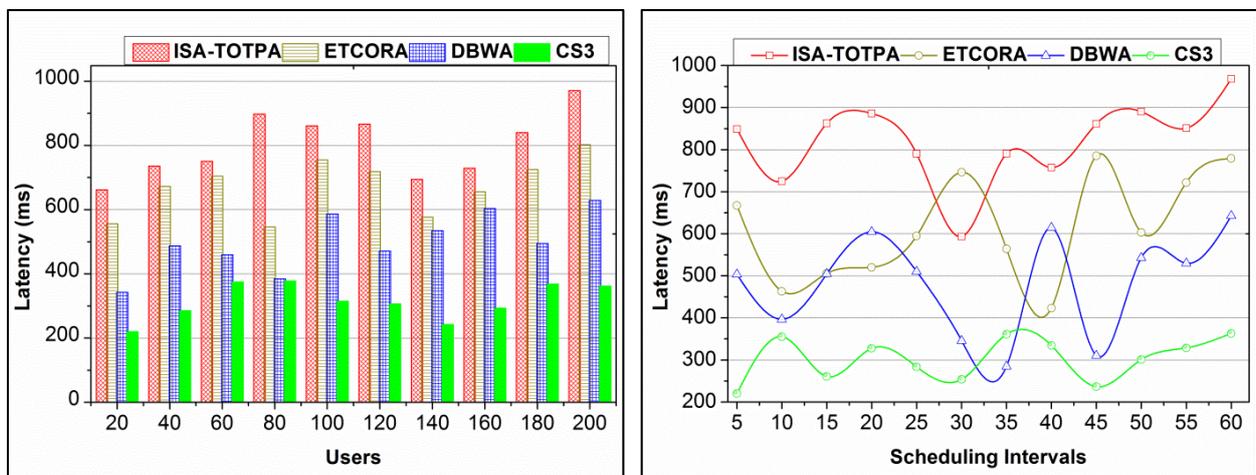


Figure 7. Latency comparisons.

4.3. Service Failure

Service failure is achieved by determining probabilistic balancing for the service distribution and deriving utilization. Utilization is improved for the valid user and examines the service forwarding, and it is represented as $\left(\rho \times \frac{q_0+r_n+v_n}{M+S}\right)$. The current state matches the prior state and estimates the service distribution for the number of users. Balancing is performed for resource access, and utilization is determined for efficient processing. Recurrent learning derives training data for efficient service sharing and determines congestion. Here, the analysis is performed for probabilistic learning, and access is derived. Here, concurrent processing is conducted to access the appropriate user network for service sharing and resource access. The analysis estimates power management by balancing the user’s service and deploying sharing. Here, power allocation is achieved for reliable processing and deploys better utilization. Utilization is improved for reliable processing between the user and the devices, and it is evaluated as $\left(\frac{\prod_{c_k}(w_c+p_i)}{g_t}\right)$. Latency is addressed by decreasing the failure transmission, which estimates the service distribution to the valid user. Access is provided to valid resources by determining the resource allocation for power management (Figure 8).

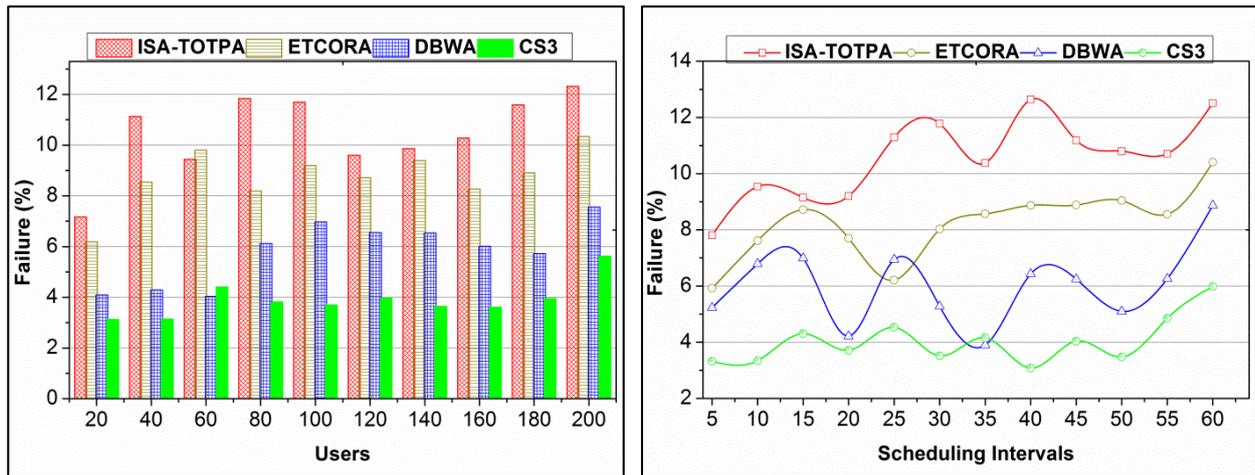


Figure 8. Failure ratio comparisons.

4.4. Outage

In Figure 9, the outage is decreased for the varying users and scheduling intervals, and the balancing of resources is determined. Sharing is derived from reliable resources and provides access to valid devices. The current state is used to define better processing and derive the mapping with the prior state, and it is represented as $\left(\frac{c_k/g_t}{S}\right) + (v_0 \times i_d)$. The prediction is performed for the prior state and estimates the joint scheduling and power allocation integration. Concurrent service sharing and resource access determine the probabilistic balancing. The evaluation is performed for power management and determines service distribution. The distribution is achieved to share the service and deploy the SDN. Balancing is performed to determine the current state and provides reliable processing. Training is conducted for the utilization process, and scheduling is estimated. Power management is used to derive reliable processing and determine the balancing. Power allocation is performed to address the outage and estimate congestion. Congestion is addressed by evaluating the prediction for the prior state with the current state. The service distribution is performed by deriving scheduling, and that is performed by performing queuing. Thus, the outage is reduced for the number of users and the service distribution.

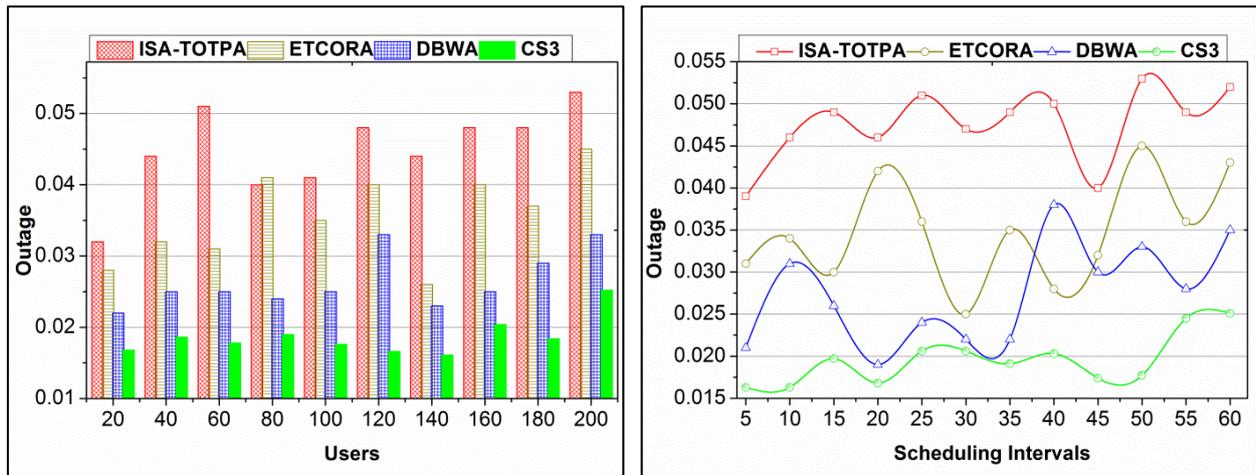


Figure 9. Outage comparisons.

4.5. Power Utilization

In Figure 10, power utilization is decreased by estimating the power allocation for the number of users and resources. Scheduling is performed for the queued devices, which deploy the current state and derive the neuron layers. The hidden layer is used to derive access to resources without congestion, and it is represented as $(\frac{i_d/M}{\partial+s_{ii}}) + (b' \times p_i)$. The analysis is performed to determine probabilistic balancing. Here, the network and device are used concurrently, and service sharing, and resource access are determined. Probabilistic balancing is conducted for service sharing and examines the power allocation, and it is computed as $\frac{X+s_{ii}}{s_{ii}}$. Here, the prior state matches the current state and provides the power management for the allocated resources. In this case, service sharing and resource access are provided concurrently for efficient processing. In this computation step, the current state determines the prediction with the prior state and provides reliable sharing. Sharing is achieved for the number of resources and determines the training service by performing hidden layers. If the utilization increases, the failure and outage of this proposed work will decrease. This utilization concerns users and scheduling intervals and shows less processing than the existing three methods.

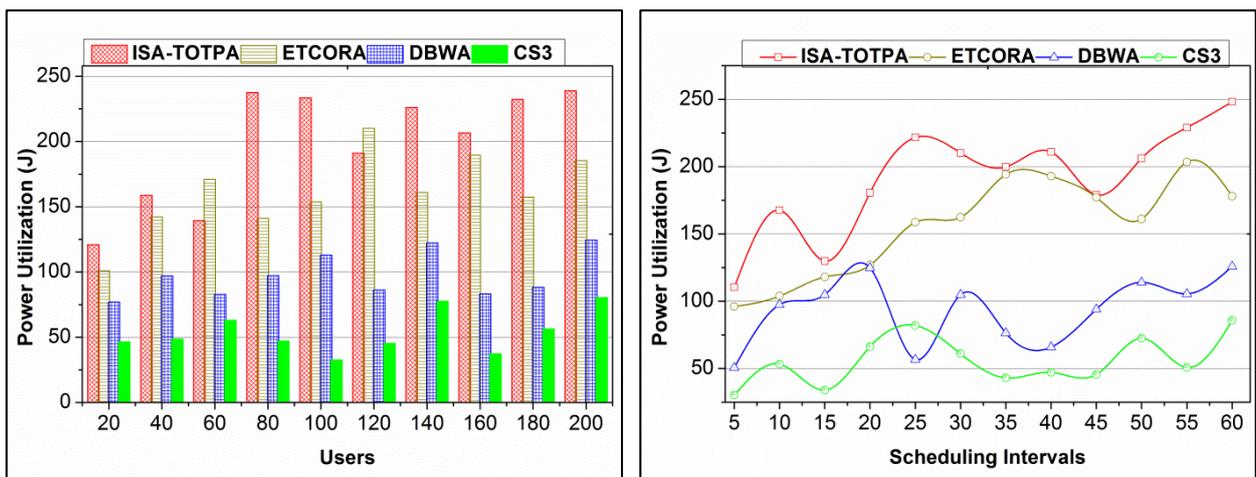


Figure 10. Power utilization comparisons.

The power allocation and latency observed for different users and sharing intervals are tabulated in Table 4.

Table 4. Power allocation and latency for different users and sharing intervals.

Sharing Intervals	Users = 50		Users = 100		Users = 150		Users = 200	
	Power Allocation	Latency (ms)						
10	33.47	219.7	38.25	360.21	40.12	420.12	70.15	541.89
20	35.48	220.36	42.36	450.23	45.14	480.36	82.36	596.37
30	42.15	360.23	51.25	521.36	51.36	512.36	102.54	698.47
40	45.69	390.36	58.69	584.12	55.47	520.98	135.14	745.21
50	49.69	412.01	63.25	612.36	69.36	674.14	145.21	847.26
60	52.14	436.14	71.25	630.12	85.21	695.48	181.36	921.47
70	63.14	470.23	73.69	624.12	121.14	714.00	196.32	963.47
80	65.47	490.23	78.69	710.25	151.32	894.12	192.36	971.05

In the article, Table 4 serves a crucial role in presenting empirical data related to the performance of the proposed system under varying conditions. Specifically, it displays the power allocation and latency metrics for different numbers of users and sharing intervals. This table is likely used to illustrate the efficiency and scalability of the system in terms of power distribution and response time as the number of users and the frequency of sharing intervals change. Service sharing is performed for power allocation and latency; latency decreases, and allocation is improved. Compared to user 50, power allocation for user 200 shows less service sharing. In another case, latency is estimated and decreases for user 50 compared to user 200. Both the power allocation and latency increase for the sharing of services (Table 4). Tables 5 and 6 present the comparative analysis summaries for users and scheduling intervals.

Table 5. Comparative analysis for users.

Metrics	ISA-TOTPA	ETCORA	DBWA	CS3	Findings
Service Dissemination	0.884	0.893	0.914	0.9466	14.9% High
Latency (ms)	971.04	802.36	629.13	361.388	18.29% Less
Failure (%)	12.32	10.34	7.56	5.627	13.34% Less
Outage	0.053	0.045	0.033	0.0252	5.54% Less
Power Utilization (J)	238.91	185.24	124.4	80.384	18.68% Less

Table 6. Comparative analysis for scheduling intervals.

Metrics	ISA-TOTPA	ETCORA	DBWA	CS3	Findings
Service Dissemination	0.899	0.925	0.943	0.9555	9.95% High
Latency (ms)	968.3	778.83	642.79	363.01	15.14% Less
Failure (%)	12.51	10.41	8.87	5.98	13.85% Less
Outage	0.052	0.043	0.035	0.0251	5.47% Less
Power Utilization (J)	248.19	178.02	125.83	85.726	17.8% Less

5. Conclusions

This article presents a CS3 for power and service management in an SDN-based IoT. This system uses resource scheduling to meet user request density and minimize latency. The SDN monitors user-resource queuing to detect response times or availability disruptions. Recurrent learning is used to probabilistically examine the service requirement and its queuing, which is the foundation for additional service distributions. The SDN's control and data plane operations handle scheduling and power management among various resources. The deep learning model in this method guaranteed the requisite power allocation and failure-free training of neurons for queuing. Schedules and subsequent service distributions are scheduled based on the residual and insufficient power. Consequently, the suggested technique controls response failures independent of scheduling intervals

and user density. The proposed plan delivers a 14.9% high-service distribution for various users, 18.29% less delay, 13.34% less failure, 5.54% less downtime, and 18.68% less power consumption. However, the limitation of this proposed method is that it is not appropriate for large environments and increases in computing time. Future studies will explore large-scale and resource-constraint IoT scenarios with real data to meet user requests.

Author Contributions: Conceptualization, M.A. and H.M.; methodology, M.A. and H.M.; software, M.A. and H.M.; validation, Z.A. and H.M.; formal analysis, M.A. and Z.A.; resources, Z.A. and H.M.; data curation, Z.A.; writing—original draft preparation, M.A. and H.M.; writing—review and editing, M.A. and Z.A.; visualization, Z.A. and H.M.; funding acquisition, Z.A. and H.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2021R1F1A1055408).

Data Availability Statement: Data is contained within the article.

Acknowledgments: The authors express their sincere appreciation to the Researcher Supporting Project Number (RSPD2024R1113) King Saud University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Rahman, A.; Islam, M.J.; Rahman, Z.; Reza, M.M.; Anwar, A.; Mahmud, M.P.; Noor, R.M. Distb-condo: Distributed blockchain-based IoT-SDN model for bright condominium. *IEEE Access* **2020**, *8*, 209594–209609. [\[CrossRef\]](#)
- Wu, T.; Zhou, P.; Wang, B.; Li, A.; Tang, X.; Xu, Z.; Chen, K.; Ding, X. Joint Traffic Control and Multi-Channel Reassignment for Core Backbone Network in SDN-IoT: A Multi-Agent Deep Reinforcement Learning Approach. *IEEE Trans. Netw. Sci. Eng.* **2020**, *8*, 231–245. [\[CrossRef\]](#)
- Kim, Y.; Song, C.; Han, H.; Jung, H.; Kang, S. Collaborative Task Scheduling for IoT-Assisted Edge Computing. *IEEE Access* **2020**, *8*, 216593–216606. [\[CrossRef\]](#)
- El-Hosseini, M.; ZainEldin, H.; Arafat, H.; Badawy, M. A fire detection model based on power-aware scheduling for IoT-sensors in smart cities with partial coverage. *J. Ambient Intell. Humaniz. Comput.* **2021**, *12*, 2629–2648. [\[CrossRef\]](#)
- Shabbir, G.; Akram, A.; Iqbal, M.M.; Jabbar, S.; Alfawair, M.; Chaudhry, J. Network performance enhancement of multi-sink enabled low power lossy networks in SDN based Internet of Things. *Int. J. Parallel Program.* **2020**, *48*, 367–398. [\[CrossRef\]](#)
- Kharwar, P.; Yadav, S.; Purohit, N. Optimal power allocation for the multiple-sensor single-DF relay-based IoT networks under transceiver hardware impairments. *Phys. Commun.* **2020**, *43*, 101183. [\[CrossRef\]](#)
- Shukla, M.K.; Nguyen, H.H.; Pandey, O.J. Multiuser Full-Duplex IoT Networks with Wireless-Powered Relaying: Performance Analysis and Energy Efficiency Optimization. *IEEE Trans. Green Commun. Netw.* **2020**, *4*, 982–997. [\[CrossRef\]](#)
- Singh, G. Research and implement solutions to tackle security threats in Mobile Cloud Gaming. Master's Thesis, University of Alberta, Edmonton, AB, Canada, 2023.
- Zia, K.; Chiumento, A.; Havinga, P.J. AI-enabled reliable QoS in multi-RAT wireless IoT networks: Prospects, challenges, and future directions. *IEEE Open J. Commun. Soc.* **2022**, *3*, 1906–1929. [\[CrossRef\]](#)
- Chakraborty, C.; Rajendran, S.R.; Rehman, M.H. *Security of Internet of Things Nodes*; Taylor & Francis Ltd: London, UK, 2021.
- Bekri, W.; Jmal, R.; Chaari Fourati, L. Internet of things management based on software defined networking: A survey. *Int. J. Wirel. Inf. Netw.* **2020**, *27*, 385–410. [\[CrossRef\]](#)
- Ai, Z.; Zhang, M.; Zhang, W.; Kang, J.; Tong, L.; Duan, Y. Survey on the scheme evaluation, opportunities and challenges of software defined-information centric network. *IET Commun.* **2023**, *17*, 2237–2274. [\[CrossRef\]](#)
- Mutruc, V. User Centric Quality of Service Improvement in IoT Networks: A Closed Loop Approach. Bachelor's Thesis, University of Twente, Enschede, The Netherlands, 2023.
- Imran, Ghaffar, Z.; Alshahrani, A.; Fayaz, M.; Alghamdi, A.M.; Gwak, J. A topical review on machine learning, software defined networking, internet of things applications: Research limitations and challenges. *Electronics* **2021**, *10*, 880. [\[CrossRef\]](#)
- Cui, X.; Gao, X.; Ma, Y.; Wang, W. A controller deployment scheme in 5G-IoT network based on SDN. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 248. [\[CrossRef\]](#)
- Sun, H.; Yu, H.; Fan, G.; Chen, L. Energy and time-efficient task offloading and resource allocation on the generic IoT-fog-cloud architecture. *Peer-Peer Netw. Appl.* **2020**, *13*, 548–563. [\[CrossRef\]](#)
- Guo, H.; Zhang, J.; Liu, J.; Zhang, H. Energy-aware computation offloading and transmit power allocation in ultradense IoT networks. *IEEE Internet Things J.* **2018**, *6*, 4317–4329. [\[CrossRef\]](#)
- Guo, M.; Li, L.; Guan, Q. Energy-efficient and delay-guaranteed workload allocation in IoT-edge-cloud computing systems. *IEEE Access* **2019**, *7*, 78685–78697. [\[CrossRef\]](#)

19. Samanta, A.; Tang, J. Dyme: Dynamic microservice scheduling in edge computing enabled IoT. *IEEE Internet Things J.* **2020**, *7*, 6164–6174. [[CrossRef](#)]
20. Wang, Q.; Wu, Z. Beamforming Optimization and Power Allocation for User-Centric MIMO-NOMA IoT Networks. *IEEE Access.* **2020**, *9*, 339–348. [[CrossRef](#)]
21. Wang, K. Energy-efficient resource allocation optimization algorithm in industrial IoTs scenarios based on energy harvesting. *Sustain. Energy Technol. Assess.* **2021**, *45*, 101201. [[CrossRef](#)]
22. Aburukba, R.O.; Landolsi, T.; Omer, D. A heuristic scheduling approach for fog-cloud computing environment with stationary IoT devices. *J. Netw. Comput. Appl.* **2021**, *180*, 102994. [[CrossRef](#)]
23. Umer, A.; Ali, M.; Jehangiri, A.I.; Bilal, M.; Shuja, J. Multi-Objective Task-Aware Offloading and Scheduling Framework for Internet of Things Logistics. *Sensors* **2024**, *24*, 2381. [[CrossRef](#)]
24. Singh, A.P.; Luhach, A.K.; Gao, X.Z.; Kumar, S.; Roy, D.S. Evolution of wireless sensor network design from technology centric to user centric: An architectural perspective. *Int. J. Distrib. Sens. Netw.* **2020**, *16*, 1550147720949138. [[CrossRef](#)]
25. Paulson, E.N.; Yusof, K.M.; Marsono MN, B.; Dauda, U.S.; Kofoworola, F. Quality of Service Evaluation of Software Defined Internet of Things Network. *ELEKTRIKA-J. Electr. Eng.* **2021**, *20*, 65–75. [[CrossRef](#)]
26. Huang, X.; Wu, W.; Hu, S.; Li, M.; Zhou, C.; Shen, X.S. Digital twin-based user-centric resource management for multicast short video streaming. *IEEE J. Sel. Top. Signal Process.* **2023**, *18*, 50–65. [[CrossRef](#)]
27. Pervej, M.F.; Jin, R.; Lin, S.C.; Dai, H. Efficient content delivery in user-centric and cache-enabled vehicular edge networks with deadline-constrained heterogeneous demands. *IEEE Trans. Veh. Technol.* **2023**, *73*, 1129–1145. [[CrossRef](#)]
28. Abir, M.A.B.S.; Chowdhury, M.Z.; Jang, Y.M. Software-Defined UAV Networks for 6G Systems: Requirements, Opportunities, Emerging Techniques, Challenges, and Research Directions. *IEEE Open J. Commun. Soc.* **2023**, *4*, 2487–2547. [[CrossRef](#)]
29. Raeisi, M.; Sesay, A.B. User-Centric Channel Allocation Scheme for 5G High-Speed Users by Utilizing Machine Learning Algorithm to Reduce Handover Rate. *IEEE Access* **2023**, *11*, 84113–84133. [[CrossRef](#)]
30. Murshed, M. Reinforcement Learning-Based User-Centric Handover Decision-Making in 5G Vehicular Networks. Master's Thesis, Brock University, St. Catharines, ON, Canada, 2024.
31. Wang, W.; Zhang, Y.; Gu, J.; Wang, J. A proactive manufacturing resources assignment method based on production performance prediction for the smart factory. *IEEE Trans. Ind. Inform.* **2021**, *18*, 46–55. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.