



Application of Polling Scheduling in Mobile Edge Computing

Xiong Wang ¹ , Zhijun Yang ^{1,2,3,*}  and Hongwei Ding ¹

¹ School of Information Science and Engineering, Yunnan University, Kunming 650000, China; wxfire1998@163.com (X.W.); dhw1964@163.com (H.D.)

² Educational Instruments and Facilities Service Center, Educational Department of Yunnan Province, Kunming 650000, China

³ Key Laboratory of Education Informatization for Nationalities of Ministry of Education, Yunnan Normal University, Kunming 650000, China

* Correspondence: yzj207@aliyun.com

Abstract: With the Internet of Things (IoT) development, there is an increasing demand for multi-service scheduling for Mobile Edge Computing (MEC). We propose using polling for scheduling in edge computing to accommodate multi-service scheduling methods better. Given the complexity of asymmetric polling systems, we have used an information-theoretic approach to analyse the model. Firstly, we propose an asymmetric two-level scheduling approach with priority based on a polling scheduling approach. Secondly, the mathematical model of the system in the continuous time state is established by using the embedded Markov chain theory and the probability-generating function. By solving for the probability-generating function's first-order partial and second-order partial derivatives, we calculate the exact expressions of the average queue length, the average polling period, and the average delay with an approximate analysis of periodic query way. Finally, we design a simulation experiment to verify that our derived parameters are correct. Our proposed model can better differentiate priorities in MEC scheduling and meet the needs of IoT multi-service scheduling.

Keywords: asymmetric polling system; MAC; MEC; high-performance computing; complex network applications; markov chains; probability-generating function



Citation: Wang, X.; Yang, Z.; Ding, H. Application of Polling Scheduling in Mobile Edge Computing. *Axioms* **2023**, *12*, 709. <https://doi.org/10.3390/axioms12070709>

Academic Editor: Jin Sun

Received: 15 June 2023

Revised: 12 July 2023

Accepted: 19 July 2023

Published: 21 July 2023



Copyright: © 2023 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 rapid rise of the mobile Internet in recent years, coupled with the accelerating pace of communication technology iterations and updates, has made the Internet of Everything a hot topic [1]. The experience of the Internet of Things (IoT) is directly related to the speed of transmission [2]. Although the Central Processing Unit (CPU) computing power of wireless devices is becoming increasingly powerful, it is still impossible to complete the data processing of these complex applications in a short time [3,4]. In addition, data processing locally also faces the problem of high energy consumption [5,6]. To overcome these problems, some scholars have tried to establish Mobile Edge Computing (MEC) techniques, and they have received much attention [7,8].

The Medium Access Control (MAC) layer protocol is responsible for time and channel access in the network. It is essential in allocating network resources and affects performance [9]. Currently, MEC is mainly focused on server deployment and compute offload, with less research on MAC protocols for MEC. It is crucial to design an efficient MAC protocol in MEC. Su, H. et al. [10] apply the MEC network architecture to an e-healthcare system and design a wireless body area network multi-channel MAC protocol based on a Markov decision process, which reduces MEC system conflicts and improves efficiency. L. Hu et al. [11] introduce AI alerting traffic to improve the scheduling of MEC, reducing server scheduling conflict and improving system efficiency. B. Li et al. [12] use NVF technology to propose a random scheduling method that reduces communication costs and improves communication efficiency. J. Tian and Hou. P. et al. [13–17] all use

reinforcement learning to improve the scheduling algorithm to enhance the scheduling efficiency of MEC and reduce delay.

However, when edge servers are deployed, there is the problem of maximizing the efficiency of their utilization. As shown in Figure 1, an edge server is deployed for the hospital in an edge zone for the doctors to get the patient's status as soon as possible. The capacity of the hospital information center to process the data (X-rays, Computed Tomography, blood tests) needs to be enhanced. The edge server provides services to the hospital information center. When the hospital information center does not need the services, the edge server will be idle—wasting computing resources. In order to maximize the utilization of the edge servers, the idle edge server can be made to provide services to other users within the service area. While the edge service is serving other users, the hospital information center will generate new data that needs to be served by the edge server. If we wait for the edge server to finish serving all the users in the service range and then provide service to the hospital information center, it will generate high latency for the hospital. The patient's condition will not be treated in time, causing further deterioration. So we need to update the way MEC of scheduling. A MEC with priority cases was proposed by Z. Qin et al. [18]. Task Scheduling: Scheduling reconnaissance tasks' execution sequence. Task selection and scheduling in UAV-enabled MEC for reconnaissance are exceptionally challenging due to many factors. A user may have a group of tasks to be processed in parallel, with different tasks having different delay limits and computation-workload requirements. D. Lin and P. Shukla et al. [19–24] mentioned the need for MEC in prioritizing. None of the proposed MAC scheduling methods with priority in these papers calculate an accurate expression for the parameter metrics and cannot be evaluated for reasonableness. Therefore, we must propose a suitable scheduling method and calculate an accurate expression for the parameter metrics.

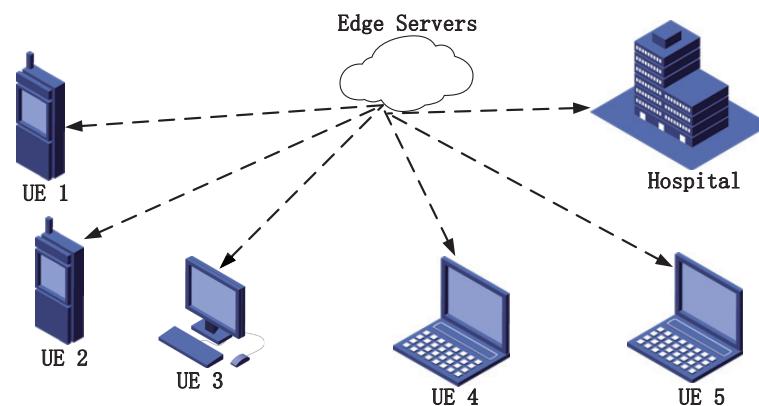


Figure 1. The multiple services for edge servers.

Polling is a typical scheduling method. The main metrics of a polling system are average queue length, average polling period, and average delay. Two-level priority asymmetric polling system is a complex network system. How to model the system? How to solve the system parameters? How to optimize the scheduling method? These issues have been a challenge in the study of polling systems.

We know that the transmission of information packets is a Poisson process, and the probability-generating function can solve the first-order moments and the second-order moments of the distribution function when only the distribution function is known. How to establish the probability-production function for asymmetric polling systems? How to solve for the first-order and second-order moments of an asymmetric polling system? These problems have been the difficulties and hotspots in studying asymmetric polling systems using probability-production functions.

To meet the need to differentiate MEC multi-services, we propose a two-level polling Medium Access Control (MAC) protocol with differentiated priority for MEC and different

unloading times and calculation processing times for User Equipment (UE). The differences between this paper and other literature are shown in Table 1.

Compared to other works of literature, our contribution is to propose a two-level priority model for asymmetric polling systems. Moreover, we accurately calculate the system's average queue length, average polling period, and average delay. The asymmetric polling system with two-level priority is more relevant than the symmetric polling system, and our proposed model can derive user-specific parameter metrics, which reduces the edge computing scheduling task latency and improves the rate at which users can transmit data.

Table 1. Comparison of relevant literature.

Literature	Asymmetrical	Machine Learning	Priority	Indicators of Measurement	Accurate Calculation
[10]	No	No	No	channel utilization, the system throughput	No
[11]	No	Yes	No	Packet blockage rate and average delay	No
[12]	No	No	No	Lyapunov function of queue backlog of operation functions and overall cost of communication and computing in the MEC	No
[13]	No	Yes	Yes	Server utilization and number of users	No
[14]	No	Yes	No	Server utilization	No
[15]	No	Yes	No	Average delay and	No
[16]	No	Yes	No	Average task queuing delay per time slot	No
[17]	No	Yes	No	Total execution time and Average task failure	No
Our work	Yes	No	Yes	Average queue length and average polling period, and average delay	Yes

The innovations of this paper are as follows:

- We propose a two-level priority exhaustive service model, and the parameters of normal nodes are asymmetrical, corresponding to the 3rd section of this paper.
- We computed specific expressions for the model's average queue length and polling period. We calculated the average delay with an approximate analysis of periodic query way, corresponding to this paper's fourth and fifth sections.
- We designed a Monte Carlo experiment with a small error between the theoretical and experimental values for many repetitions, corresponding to the 7th section of this paper.

The paper is organized as follows. The first section focuses on some of the hotspots and difficulties of current research in MEC clarifies why it is essential to improve the MAC protocol for MEC, and introduces polling techniques to mobile edge computing. The second part is a review of the literature. We present the polling technique's features and the current research's difficulties, and compare them with the relevant literature. In the third section, a model of a polling system with priority is proposed, and random variables for the system state are defined. The 4th section pushes the specific equations for the average queueing length, polling period, and system delay. In Section 5, a Monte Carlo experiment was designed to verify the error rate between the theoretical and simulated values when the experiment was repeated 300,000 times. In Section 6, we summarise the advantages of polling techniques applied to MEC and reveal the direction of future work.

2. Literature Review

Polling originates in the so-called polling data link control scheme, in which the server asks each terminal on the communication line to determine if it has any information to transmit. The addressed terminal transmits the information, and the computer then switches to the next terminal to check whether that terminal has any information to transmit. From a broader perspective, the polling model is suitable for several users competing to access a shared resource only available to one user at a time. The prevalence of polling systems can be observed in many applications, for example, in computer communications,

production, transport, and maintenance systems. This paper discusses the main application areas of polling systems and an extensive list of references, examines how these different applications can be represented and analyzed using a polling model, and outlines several topics for future research both within and outside the mentioned area [25].

In the past, various survey papers dedicated to voting systems have appeared in the open literature. Detailed and comprehensive descriptions of the mathematical analysis of voting systems are given in [25–28]. While these papers have many mathematical models to derive parameter metrics for polling systems, they all have limitations. Furthermore, since the publication of the survey papers 10–15 years ago, many papers on voting models have been published. This survey is aimed at academics and practitioners. The list of references is relatively recent but has yet to be totaled. Readers interested in delving into the specific mathematical details of the polling system can do so from the references in the survey papers mentioned above.

Polling is an effective transmission technique, and it is widely used in various industries, such as computers, communications, and industrial control. D. Miculescu and Li, S. et al. [29,30] use a polling mechanism in autonomous driving, reducing delays and improving system stability. Mohamed S. and Zakir, R. et al. [31,32] apply a polling mechanism to robots, enabling low-latency robot swarm control to cooperate in completing tasks. The swarm of robots can perform more complex tasks than a single robot. In order to increase the computational efficiency of UAVs and to perform more complex tasks, Nidal Nasser and Sudesh Kumar et al. [33–35] used polling techniques. Polling techniques are now widely used in information technology.

According to the service method, polling service systems are classified as gated, exhausted, and limited-k [36]. The exhaustive service rule is that the server continues working until the queue becomes empty; the gated service rule, under which precisely those clients that appear in the queue at the start of the access are served; and the global gated service, which states that only those clients that appear in the epoch when the server reaches some predetermined "parent queue" are served. With the limited-k service policy, the server works in a queue until a predefined client is served or the queue becomes empty. Average queue length and average delay are key metrics of how good a polling system is, and few methods are solving for polling system metrics. Some scholars have used machine learning methods to make predictions [37,38]. However, machine learning is subject to errors between predicted and actual values. In order to meet practical business needs, a two-level priority polling is proposed based on three basic polling service models [39]. The concept of cycle delay and the exact Laplace-Stieltjes transform (LST) formula and representation of the average delay is given by Chu Y Q et al. [40]. A two-level priority polling system based on a gated multi-level threshold service is proposed by WH.M. et al. [41]. This polling system meets the development needs of service diversity and resilient services while allocating service resources in a cyclical system. A new two-level priority polling system is proposed by Z.G. et al. [42], ensuring queue priority requirements, avoiding time delays caused by idle queries, improving system utilization, and reducing latency effects. ZJ. Y. et al. [43] proposes a two-level priority exhaustive service polling system model, but symmetrically. In this paper, we propose a two-level priority exhaustive service model, and the parameters of normal nodes are asymmetrical.

3. System Model

As shown in Figure 2, the MEC network consists of UE i ($i = 1, 2, \dots, N$) nodes and a central node (Hospital). The process of information packets arriving at each node obeys independent Poisson distribution. λ_i ($i = 1, 2, \dots, N$) is the arrival rate of normal nodes. λ_h is the arrival rate of the central node. Normal nodes are asymmetric and use exhaustive services to transmit data. Central nodes use exhaustive services to transmit data. β_i ($i = 1, 2, \dots, N, h$) is the service time of each node. γ_i ($i = 1, 2, \dots, N$) is the time to switch from i to $i + 1$.

As shown in Figure 3, the polling system process [44] is:

1. Server initialization
2. Edge servers use parallel scheduling to query the central node and node i
3. Edge server service center node if the central node needs to process data
4. Server servicing of node i
5. Let $i = i + 1$, and return to step (2) if $i \neq N$, else step (1)

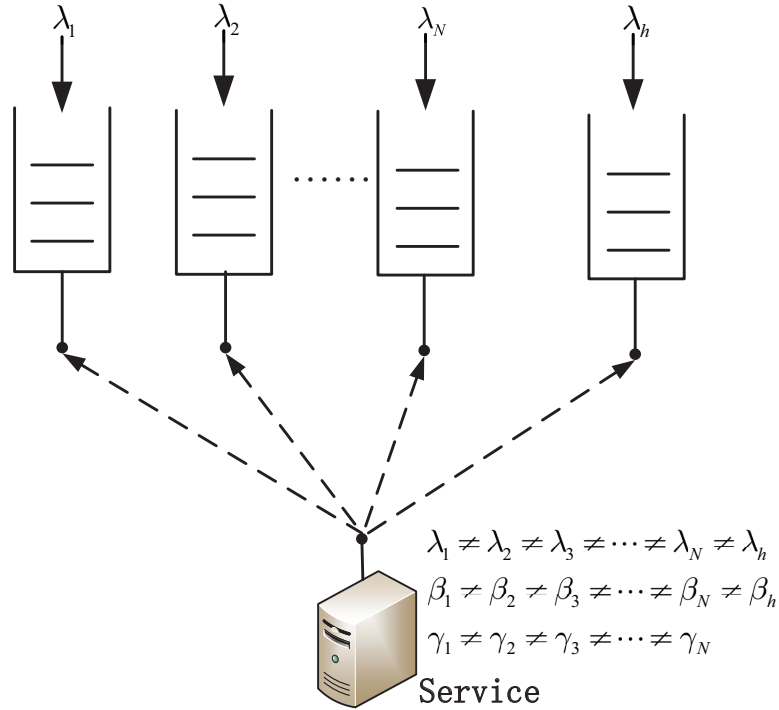


Figure 2. System model.

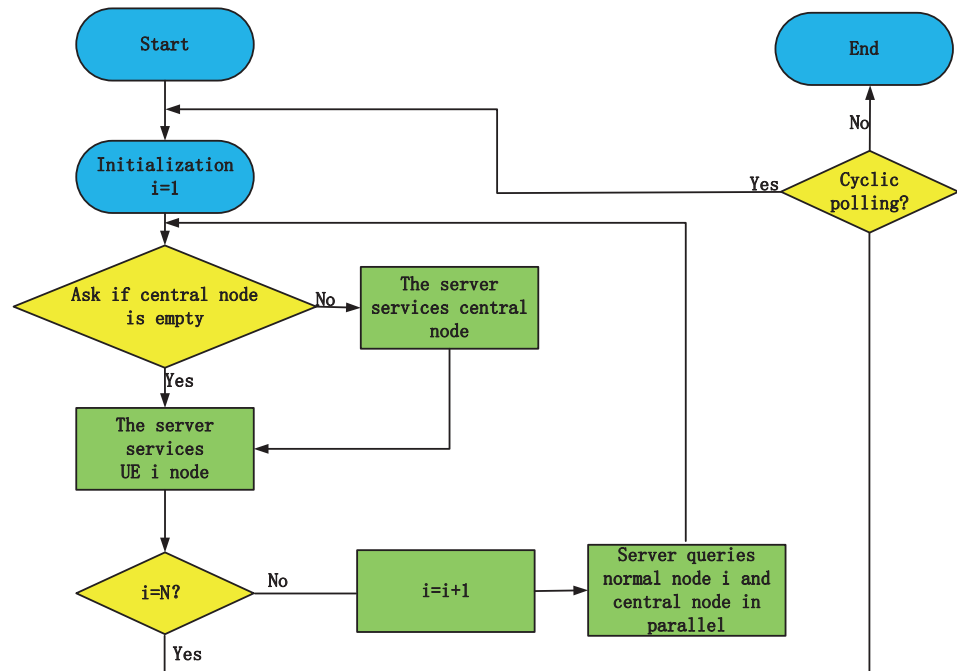


Figure 3. The flow chart of the MEC networks.

4. System Model Analysis

4.1. Variable Definitions

We list the defined variables in tabular form. List the variables in the form of Table 2.

Table 2. Definition of system variables.

Variables	Definition
i or j or k	Normal nodes
h	Central node
$\mu_i(n)$	The server switching time from node i to other nodes
$v_i(n)$	The time of service provided by server to node i
$v_i(n^*)$	The time of service provided by the server to the central node
$\mu_j(\mu_i)$	The amount of data entering the node j within time $\mu_i(n)$
$\mu_h(\mu_i)$	The amount of data entering the central node within time $\mu_i(n)$
$\eta_j(v_i)$	The amount of data entering the node j within time $v_i(n)$
$\eta_h(v_i)$	The amount of data entering the node within time $v_i(n)$
$\eta_j(v_h)$	The amount of data entering the node j within time $v_i(n^*)$
$\eta_h(v_h)$	The amount of data entering the central node within time $v_i(n^*)$

4.2. System Conditions

There are three moments of system working status. One: the node $i(i = 1, 2, \dots, N)$ receiving server services within time t_n until the node i is empty. The servers use parallel scheduling to query the central node and node $i + 1$ within time γ_i . Two: the central node receiving server services within time t_n^* if the central node needs to process data. Three: the node $i + 1(i = 1, 2, \dots, N - 1)$ receiving server services within time t_{n+1} until the node $i + 1$ is empty. This state process is shown in Figure 4.

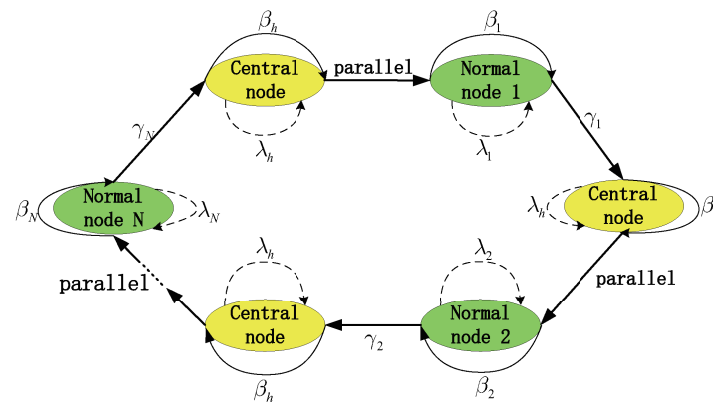


Figure 4. State transition diagram.

The time sequence is $t_n < t_n^* < t_{n+1}$. Moreover, the random variable $\xi_i(n)$ represents the amount of data stored in the buffer of node i at time t_n , and the random variable $\xi_h(n^*)$ represents the amount of data stored in the buffer of central node at time t_n^* . At time t_n and t_n^* , and t_{n+1} , the system conditions variables are $\{\xi_1(n), \xi_2(n), \dots, \xi_i(n), \dots, \xi_N(n), \xi_h(n)\}$ and $\{\xi_1(n^*), \xi_2(n^*), \dots, \xi_i(n^*), \dots, \xi_N(n^*), \xi_h(n^*)\}$, and $\{\xi_1(n + 1), \xi_2(n + 1), \dots, \xi_i(n + 1), \dots, \xi_N(n + 1), \xi_h(n + 1)\}$. The system's state variables are described by Markov chains, which are acyclic and ephemeral in each state.

At time $t_n \rightarrow t_n^*$, we get:

$$\begin{cases} \xi_h(n^*) = \xi_h(n) + \mu_h(\mu_i) + \eta_h(v_i) \\ \xi_i(n^*) = \mu_i(\mu_i) \\ \xi_j(n^*) = \xi_j(n) + \mu_j(\mu_i) + \eta_j(v_i), j = 1, 2, \dots, N, j \neq i \end{cases} \tag{1}$$

And at time $t_n^* \rightarrow t_{n+1}$:

$$\begin{cases} \zeta_k(n+1) = \zeta_k(n^*) + \eta_k(v_h), k = 1, 2, \dots, N \\ \eta_h(n+1) = 0 \end{cases} \tag{2}$$

4.3. Mathematical Models

According to the working mechanism of the polling system, the arrival process, and the transmission characteristics of data in the communication process, the following working conditions are defined:

- Each information packet arrives at the nodes independently and Poisson distribution. The probability-generating function of normal node is $A(z)$, and the mean and variances are $\lambda_i (i = 1, 2, \dots, N) = \lambda = A'(1)$ and $\sigma_\lambda^2 = A''(1) + \lambda - \lambda^2$. And the central node, the probability-generating function is $A_h(z)$, and the mean and variances are $\lambda_h = A'_h(1)$ and $\sigma_{\lambda_h}^2 = A''_h(1) + \lambda_h - \lambda_h^2$
- The time for the server to serve any normal nodes is independent and Poisson distribution, and its probability-generating function is $B(z)$, the mean is $\beta = B'(1)$, and the variance is $\sigma_\beta^2 = B''(1) + \beta - \beta^2$. And the central node, the probability-generating function is $B_h(z)$, the mean is $\beta_h = B'_h(1)$, and the variance is $\sigma_{\beta_h}^2 = B''_h(1) + \beta_h - \beta_h^2$
- The switch times of the server from node i to another node are independent and Poisson distribution. The probability-generating function is $R(z)$, the mean is $\gamma = R'(1)$, and its variance is $\sigma_\gamma^2 = R''(1) + \gamma - \gamma^2$
- Each data is First Input First Output (FIFO)
- Sufficient node capacity, no data overflow [45]

The system remains stable under the condition of $\sum_{i=1}^N \lambda_i \beta_i + \lambda_h \beta_h = \sum_{i=1}^N \rho_i + \rho_h < 1$. In steady state [46], the probability distribution function of the system state variables is:

$$\lim_{n \rightarrow \infty} P[\zeta_i(n) = x_i; i = 1, 2, \dots, N, h] = \pi_i(x_1, x_2, \dots, x_i, \dots, x_N, x_h) \tag{3}$$

According to the definition of the probability-generating function, the generating function is:

$$\begin{aligned} & \pi_i(x_1, x_2, \dots, x_i, \dots, x_N, x_h) \\ &= G_i(z_1, z_2, \dots, z_i, \dots, z_N, z_h) \\ &= \sum_{x_1=1}^{\infty} \sum_{x_2=2}^{\infty} \dots \sum_{x_i=i}^{\infty} \dots \sum_{x_N=N}^{\infty} \sum_{x_h=h}^{\infty} z_1^{x_1} z_2^{x_2} \dots z_i^{x_i} \dots z_N^{x_N} z_h^{x_h} \end{aligned} \tag{4}$$

At the time t_n^* , the server starts to transmit the data of h , and the probability-generating function of the system state variable is:

$$\begin{aligned} & G_{ih}(z_1, z_2, \dots, z_i, \dots, z_N, z_h) \\ &= \lim_{t \rightarrow \infty} E\left[\prod_{j=1}^N z_j^{\zeta_j(n^*)} \cdot z_h^{\zeta_h(n^*)}\right] \\ &= \lim_{t \rightarrow \infty} E\left[\prod_{j=1 \neq i}^N z_j^{\zeta_j(n) + \mu_j(u_i) + \eta_j(v_i)} \cdot z_i^{\mu_i(u_i)} \cdot z_h^{\zeta_h(n^*)} \cdot z_h^{\zeta_h(n) + \mu_h(u_i) + \eta_h(v_i)}\right] \\ &= \lim_{t \rightarrow \infty} E\left[\prod_{j=1 \neq i}^N z_j^{\zeta_j(n)} \cdot z_h^{\zeta_h(n)}\right] \cdot E\left[\prod_{j=1 \neq i}^N z_j^{\eta_j(v_i)} \cdot z_h^{\eta_h(v_i)}\right] \cdot E\left[\prod_{j=1}^N z_j^{\mu_j(u_i)} \cdot z_h^{\mu_h(u_i)}\right] \\ &= R_i\left[\prod_{j=1}^N A_j(z_j) A_h(z_h)\right] \cdot G_i(z_1, z_2, \dots, z_{i-1}, B_i\left(\prod_{j=1}^N A_j(z_j) A_h(z_h)\right), z_{i+1}, \dots, z_N, z_h) \end{aligned} \tag{5}$$

At the time t_{n+1} , the server starts to transmit the data of $i + 1$ node, and the probability-generating function of the system state variable is:

$$\begin{aligned}
 &G_{i+1}(z_1, z_2, \dots, z_i, \dots, z_N, z_h) \\
 &= \lim_{t \rightarrow \infty} E\left[\prod_{j=1}^N z_j^{\xi_j^{(n+1)}} \cdot z_h^{\xi_{ih}^{(n+1)}}\right] \\
 &= \lim_{t \rightarrow \infty} E\left[\prod_{j=1}^N z_j^{\xi_j^{(n^*)} + \eta_j(v_h)}\right] \\
 &= \lim_{t \rightarrow \infty} E\left[\prod_{j=1}^N z_j^{\xi_j^{(n)} + \mu_j(u_i) + \eta_j(v_i) + \eta_j(v_h)}\right] \\
 &= G_{ih}[z_1, z_2, \dots, z_i, \dots, z_N, B_h(\prod_{j=1}^N A_j(z_j) F_i(\prod_{j=1}^N A_j(z_j)))]
 \end{aligned} \tag{6}$$

In the Equation (6), $F_i(z_i) = A_i(B_i(z_i F_i(z_i)))$ is the probability-generating function of the data of entering in node i [47].

5. Analysis of System Variables

5.1. The Average Queue Length

The average queue length measures how fast packet information is transmitted. To get a more intuitive view of the polling system, we need to find the average queue length of the polling system. Definition: The average queue length $g_i(j)$ of the system is the average information packet stored in the node j when the node i starts to transmit data at time t_n . We get:

$$g_i(j) = \frac{\partial G_i(z_1, z_2, \dots, z_i, \dots, z_N, z_h)}{\partial z_j} \tag{7}$$

According to Equations (3)–(6), we get:

$$g_i(i) = \frac{\lambda_i(1 - \rho_i) \sum_{j=1}^N \gamma_j}{1 - \rho_h - \sum_{j=1}^N \rho_j} \tag{8}$$

Similarly, we get the h average queue length as

$$g_{ih}(i) = \frac{\lambda_h \gamma_i (1 - \rho_h)}{1 - \rho_h - \sum_{j=1}^N \rho_j} \tag{9}$$

5.2. The Average Cycle

The average cycle time is an essential indicator of a polling system. Smaller average cycles mean more system throughput and more information is processed in the same amount of time. Definition: The average cycle of the system is the time interval for the server to query the same site twice. The specific expression is the time it takes for the server to complete one service for N nodes in the system according to the service rules [48]. We get:

$$E(\theta) = \frac{\sum_{i=1}^N \gamma_i}{1 - \rho_h - \sum_{i=1}^N \rho_i} \tag{10}$$

5.3. The Average Delay

The average delay is a measure of information being sent from the server to the end. When the average delay is lower, the node receives information from the server faster, and the system is smoother. In order to calculate the average delay of the polling system, we need to calculate the second-order partial derivatives of the probability-generating function.

Definition: second-order partial derivative variables:

$$g_i(j, k) = \lim_{z_1, z_2, \dots, z_j, \dots, z_k, \dots, z_N, z_h \rightarrow 1} \frac{\partial^2 G_i(z_1, z_2, \dots, z_N, z_h)}{\partial z_j \partial z_k} \tag{11}$$

And $i = 1, 2, \dots, N, h; j = 1, 2, \dots, N, h; k = 1, 2, \dots, N, h$.

According to Equations (3)–(5), we get:

$$\begin{aligned} g_{i+1}(k, l) &= g_{ih}(k, l) + \beta_h g_{ih}(k, h) [\lambda_l + F'_h(1)\lambda_l] + g_{ih}(h, l) \beta_h [\lambda_k + F'_h(1)\lambda_k] \\ &+ g_{ih}(h, h) \beta_h [\lambda_l + F'_h(1)\lambda_l] + g_{ih}(h) B''_h(1) [\lambda_l + F'_h(1)\lambda_l] \cdot [\lambda_k + F'_h(1)\lambda_k] \\ &+ g_{ih}(h) \beta_h [\lambda_l \lambda_k + F_h(1)\lambda_l \lambda_k + \lambda_l F'_h(1)\lambda_k + F''_h(1)\lambda_l \lambda_k + F'_h(1)\lambda_l \lambda_k] \end{aligned} \tag{12}$$

$$\begin{aligned} g_{i+1}(k, i) &= \lambda_i \lambda_k R''_i(1) + \lambda_i \lambda_k \gamma_i + \lambda_i \lambda_k g_i(k) + [2\lambda_i \lambda_k \beta_i \gamma_i + \lambda_i \lambda_k B''_i(1) + \lambda_i \lambda_k \beta_i] g_i(i) \\ &+ \lambda_i \beta_i g_i(k, i) + \lambda_i \lambda_k \beta_i^2 g_i(i, i) + \lambda_i \beta_h [1 + F'_h(1)] g_{ih}(k, h) + \lambda_k \beta_h [1 + F'_h(1)] g_{ih}(h, i) \\ &+ \lambda_i \lambda_k \beta_h^2 [1 + F'_h(1)]^2 g_{ih}(h, h) + \lambda_i \lambda_k B''_h(1) [1 + F'_h(1)]^2 g_{ih}(h) \\ &+ \lambda_i \lambda_k \beta_h [1 + 3F'_h(1) + F''_h(1)] g_{ih}(h) \end{aligned} \tag{13}$$

$$\begin{aligned} g_{i+1}(i, l) &= \lambda_i \lambda_l R''_i(1) + \lambda_i \lambda_l \gamma_i + \lambda_i \lambda_l g_i(l) + [2\lambda_i \lambda_l \beta_i \gamma_i + \lambda_i \lambda_l B''_i(1) + \lambda_i \lambda_l \beta_i] g_i(i) \\ &+ \lambda_i \beta_i g_i(i, l) + \lambda_i \lambda_l \beta_i^2 g_i(i, i) + \lambda_l \beta_h [1 + F'_h(1)] g_{ih}(i, h) + \lambda_i \beta_h [1 + F'_h(1)] g_{ih}(h, l) \\ &+ \lambda_i \lambda_l \beta_h^2 [1 + F'_h(1)]^2 g_{ih}(h, h) + \lambda_i \lambda_l B''_h(1) [1 + F'_h(1)]^2 g_{ih}(h) \\ &+ \lambda_i \lambda_l \beta_h [1 + 3F'_h(1) + F''_h(1)] g_{ih}(h) \end{aligned} \tag{14}$$

$$\begin{aligned} g_{ih}(k, h) &= \lambda_k \lambda_h R''_i(1) + \lambda_k \lambda_h \gamma_i + \lambda_k \gamma_i [g_i(h) + \lambda_h \beta_i g_i(i)] + \lambda_h \gamma_i [g_i(k) + \lambda_k \beta_i g_i(i)] \\ &+ g_i(k, h) + \lambda_h \beta_i g_i(k, i) + \lambda_k \beta_i g_i(i, h) + \lambda_k \lambda_h B''_i(1) g_i(i) + \lambda_k \lambda_h \beta_i g_i(i) \end{aligned} \tag{15}$$

$$\begin{aligned} g_{ih}(h, l) &= \lambda_l \lambda_h R''_i(1) + \lambda_l \lambda_h \gamma_i + \lambda_h \gamma_i g_i(l) + \lambda_l \gamma_i g_i(h) + [2\lambda_l \lambda_h \beta_i \gamma_i + \lambda_l \lambda_h B''_i(1) \\ &+ \lambda_l \lambda_h \beta_i] g_i(i) g_i(h, l) + \lambda_l \beta_i g_i(h, i) + \lambda_h \beta_i g_i(i, l) + \lambda_l \lambda_h \beta_i^2 g_i(i, i) \end{aligned} \tag{16}$$

$$g_{ih}(h, h) = \lambda_h^2 R''_i(1) + \gamma_i A''_h(1) + [2\lambda_h^2 \beta_i \gamma_i + \lambda_h^2 B''_h(1) + \beta_i A''_h(1)] g_i(i) + \lambda_h^2 \beta_i^2 g_i(i, i) \tag{17}$$

Summing Equation (15) gets:

$$\begin{aligned} &\sum_{i=1}^N g_{ih}(k, h) \\ &= \lambda_k \lambda_h \sum_{i=1}^N R''_i(1) + \lambda_k \lambda_h \sum_{i=1}^N \gamma_i + \lambda_h \sum_{i=1 \neq k}^N \gamma_i g_i(k) + \lambda_k \lambda_h \sum_{i=1}^N [2\beta_i \gamma_i + B''_i(1) + \beta_i] g_i(i) \\ &+ \lambda_h \sum_{i=1}^N \beta_i g_i(k, i) + \lambda_k \lambda_h \sum_{i=1}^N \beta_i^2 g_i(i, i) \end{aligned} \tag{18}$$

Summing Equation (16) gets:

$$\begin{aligned} & \sum_{i=1}^N g_{ih}(h, l) \\ &= \lambda_l \lambda_h \sum_{i=1}^N R_i''(1) + \lambda_l \lambda_h \sum_{i=1}^N \gamma_i + \lambda_h \sum_{i=1 \neq k}^N \gamma_i g_i(k) + \lambda_l \lambda_h \sum_{i=1}^N [2\beta_i \gamma_i + B_i''(1) + \beta_i] g_i(i) \quad (19) \\ &+ \lambda_h \sum_{i=1}^N \beta_i g_i(i, l) + \lambda_l \lambda_h \sum_{i=1}^N \beta_i^2 g_i(i, i) \end{aligned}$$

According to Equations (11)–(13), $\sum_{i=1}^N g_{ih}(h, l)$ and $\sum_{i=1}^N g_{ih}(k, h)$, and $g_{ih}(h, h)$, we get:

$$\sum_{i=1}^N g_{i+1}(k, l)$$

According to Equation (18) we get:

$$\begin{aligned} & \sum_{i=1}^N g_{ih}(h, k) \\ &= \lambda_h \lambda_k \sum_{i=1}^N R_i''(1) + \lambda_h \lambda_k \sum_{i=1}^N \gamma_i + \lambda_h \sum_{i=1}^N \gamma_i g_i(k) \quad (20) \\ &+ \sum_{i=1}^N [2\lambda_h \lambda_k \beta_i \gamma_i + \lambda_h \lambda_k B_i''(1) + \lambda_h \lambda_k \beta_i] g_i(i) + \lambda_h \sum_{i=1}^N \beta_i g_i(i, k) + \lambda_h \lambda_k \sum_{i=1}^N \beta_i^2 g_i(i, i) \end{aligned}$$

According to Equation (20) and use Equation $\sum_{i=1}^N g_{i+1}(k, l)$ we get:

$$\begin{aligned} & g_k(k, k) \\ &= \lambda_k^2 \sum_{i=1}^N R_i''(1) + A_k''(1) \sum_{i=1}^N \gamma_i + \sum_{i=1}^N [2\lambda_k^2 \beta_i \gamma_i + \lambda_k^2 B_i''(1) + \gamma_i A_k''(1)] g_i(i) \\ &+ 2\lambda_k \sum_{i=1 \neq k}^N \gamma_i g_i(k) + \lambda_k \sum_{i=1 \neq k}^N \beta_i g_i(k, i) + \lambda_k \sum_{i=1 \neq k}^N \beta_i g_i(i, k) + \lambda_k^2 \sum_{i=1 \neq k}^N \beta_i^2 g_i(i, i) \quad (21) \\ &+ \lambda_k \beta_h [1 + F_h'(1)] \sum_{i=1}^N g_{ih}(h) + \lambda_k \beta_h [1 + F_h'(1)] \sum_{i=1}^N g_{ih}(h, k) + \lambda_k^2 \beta_h^2 [1 \\ &+ F_h'(1)]^2 \sum_{i=1}^N g_{ih}(h) + \beta_h [A_k''(1) + 2\lambda_k F_h'(1) + \lambda_k^2 F_h''(1) + A_k''(1) F_h'(1)] \sum_{i=1}^N g_{ih}(h) \end{aligned}$$

Based on the approximate analysis of the cyclic query cycle, the probability-generating function of the cyclic cycle is defined as:

$$\theta_i(A_i(z_i)) = G_i(1, \dots, z_i, \dots, 1), i = 1, \dots, N, h \quad (22)$$

We sum up Equations (17) and (18), and according to second order partial derivative Equation (21), we get:

$$\lambda_i^2 \theta_i''(1) + \theta A_i''(1) = g_i(i, i) \quad (23)$$

According to second order partial derivative of the cyclic cycle, we get:

$$\theta_i''(1) \approx \theta_j''(1) \quad (24)$$

According to Equations (21)–(23), we get:

$$g_i(i, i) = \frac{\lambda_i^2}{\sum_{k=1}^N \rho_k(1 + \rho_k)} \left[\sum_{k=1}^N \frac{\beta_k}{\lambda_k} (1 + \rho_k) g_k(k, k) - \theta \sum_{k=1}^N \frac{\beta_k}{\lambda_k} (1 + \rho_k) A_k''(1) \right] \tag{25}$$

According to Equations (16) and (24), we get:

$$\begin{aligned} &g_{ih}(h, h) \\ &= \lambda_h^2 R_i''(1) + \gamma_i A_h''(1) + [2\lambda_h^2 \beta_i \gamma_i + \lambda_h^2 B_h''(1) + \beta_i A_h''(1)] g_i(i) + \\ &\lambda_h^2 \beta_i^2 \left\{ \frac{\lambda_i^2}{\sum_{k=1}^N \rho_k(1 + \rho_k)} \left[\sum_{k=1}^N \frac{\beta_k}{\lambda_k} (1 + \rho_k) g_k(k, k) - \theta \sum_{k=1}^N \frac{\beta_k}{\lambda_k} (1 + \rho_k) A_k''(1) \right] \right\} \end{aligned} \tag{26}$$

Definition 1. The average delay of the system is the time interval between the information packet arriving at the node and the information packet being sent out. The average delay of entering node is $E(w_i)$, and the average delay of h is $E(w_{ih})$ [49], we get:

$$E(w_i) = \frac{g_i(i, i)}{2\lambda_i g_i(i)} - \frac{\lambda_i^2}{2\lambda_i^2(1 + \rho_i)} + \frac{\lambda_i \beta_i^2}{2(1 - \rho_i)} \tag{27}$$

Similarly, we get:

$$E(w_{ih}) = \frac{g_{ih}(h, h)}{2\lambda_h g_{ih}(i)} - \frac{\lambda_h^2}{2\lambda_h^2(1 + \rho_h)} + \frac{\lambda_h \beta_h^2}{2(1 - \rho_h)} \tag{28}$$

6. Simulation

We have simulated the experimental and theoretical values of the system using MATLAB R2022a, and the entire simulation was carried out under Windows 11 operating system. The simulation conditions are as follows.

- Each information packet of the nodes is asymmetric
- Each information packet arrives at the nodes independently and Poisson process
- The experiment was repeated 300,000 times, and the mean was taken as the statistic.
- The system remains stable under the condition of $\sum_{i=1}^N \lambda_i \beta_i + \lambda_h \beta_h = \sum_{i=1}^N \rho_i + \rho_h < 1$

Based on the initial parameters set in Table 3, the variation of the performance characteristics of the service model is obtained in the following.

Table 3. Initial parameters.

<i>i</i> Node	λ_i Arrival Rate	β_i Service Time	γ_i Switch Time
1	0.01	4	2
2	0.006	5	3
3	0.001	3	1.5
4	0.005	2	1.7
5	0.002	1	1.9
<i>h</i>	0.008	1	0

Figures 5 and 6 are the average queue length of nodes i , and h varies with the arrival rate. The theoretical and experimental values are consistent, indicating the rationality of the theoretical analysis. According to Figure 5, we can conclude that: The higher the arrival rate, the larger the average queue length. As the arrival rate increases, the average queue length for all normal nodes increases. According to Figure 6, we know the average queue length of i

to h proportional to the switch time of node i . The primary variable that affects the average queue length is the service time of the node. Moreover, we get: The average queue length of h is lower than i . Better meet the multiple service priority requirements of the MEC network.

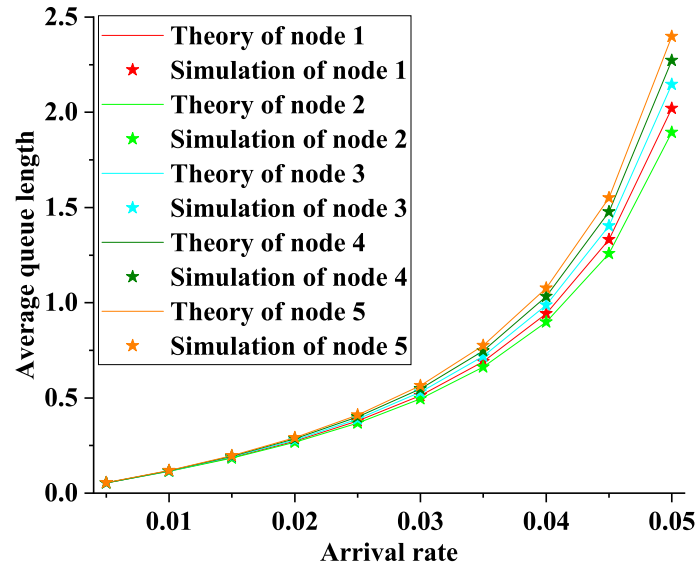


Figure 5. The average queue length of node i varies with the arrival rate ($N = 5$).

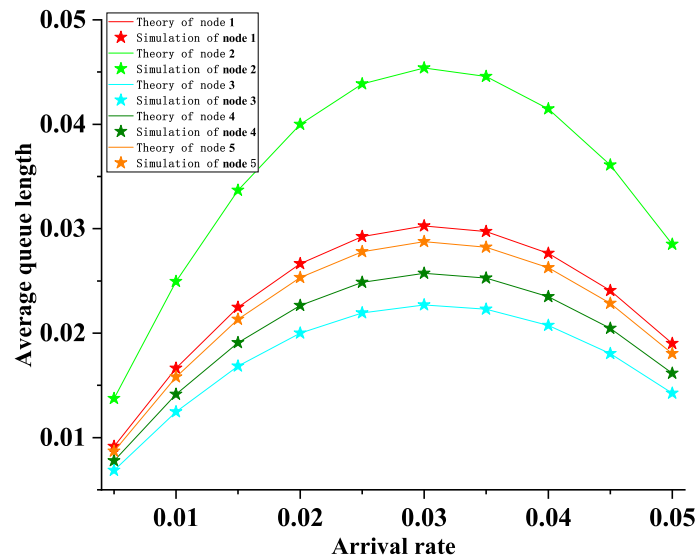


Figure 6. The average queue length of h varies with the arrival rate ($N = 5$).

Figures 7 and 8 are the average queue length of nodes i , and h varies with service time. The experimental values fit the theoretical values. According to Figure 7, we can conclude that: The higher the service time, the larger the average queue length. Moreover, the service time of normal nodes has a more significant impact on the average queue length of nodes than the arrival rate of information packets from normal nodes. According to Figure 8, we know the average queue length of h is proportional to the switch time of node i . This conclusion is consistent with the above because the polling system throughput is $T = \sum_{i=1}^N \lambda_i \beta_i + \lambda_h \beta_h = \sum_{i=1}^N \rho_i + \rho_h$. Moreover, when the service time and switch time of normal nodes are short, the smaller the average queue length of the central node, the faster the transmission rate.

Comparing Figures 5–8, we can see that the impact of the system arrival rate on the average queue length is more significant than the service time of a single user, provided that other parameter characteristics remain constant, indicating that the system has a

finite throughput and cannot transmit an unlimited number of packets of information. In addition, we conclude that the average queue length of the central node is much lower than the normal node, satisfying our need for priority.

We compare Figures 5–8 to know that the average queue length in both Figures 5 and 7 increases with increasing system parameters (Figure 5 is arrival rate, Figure 7 is service time). Furthermore, both Figures 6 and 8 decreases with increasing system parameters (Figure 6 is arrival rate, Figure 8 is service time), indicating that when both central and normal nodes use exhausted services, the increase in system throughput is instead beneficial to the priority requirement (central node transmits faster).

Figure 9 shows the image of the average queue length as a function of the arrival rate for two-level priority and symmetric polling system. Comparing Figures 5 and 6 with Figure 9, we know that although the average queue length of the central node is lower than that of the average node, the symmetric system can only derive the parameters of one node. The average queue length derived by each node is the same, and the system cannot cope with multi-task scheduling requirements. The initial parameters of Figure 9 are taken as the average of Figures 5 and 6.

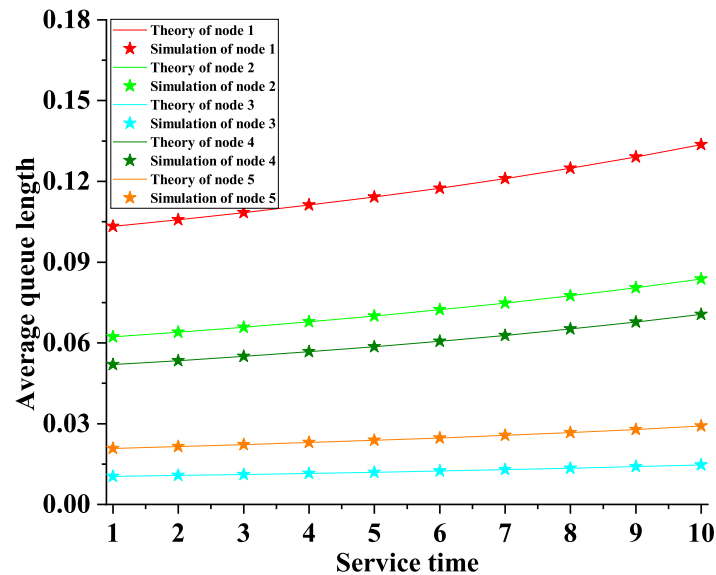


Figure 7. The average queue length of node i with the service time ($N = 5$).

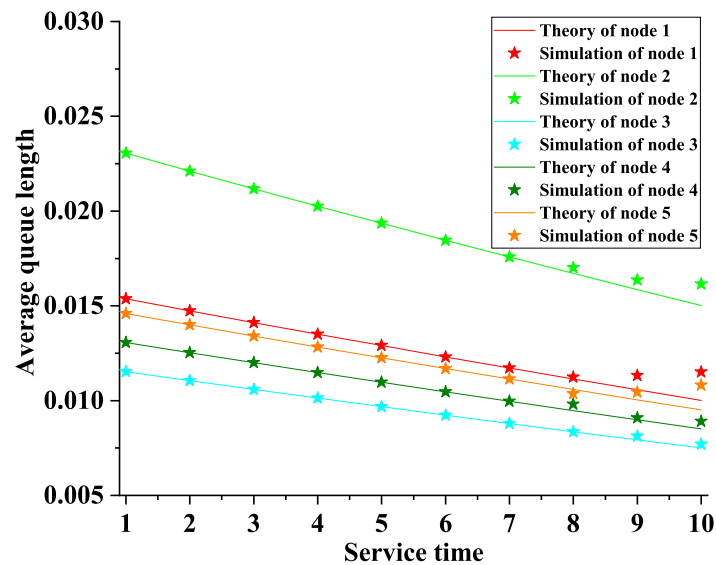


Figure 8. The average queue length of h varies with the service time ($N = 5$).

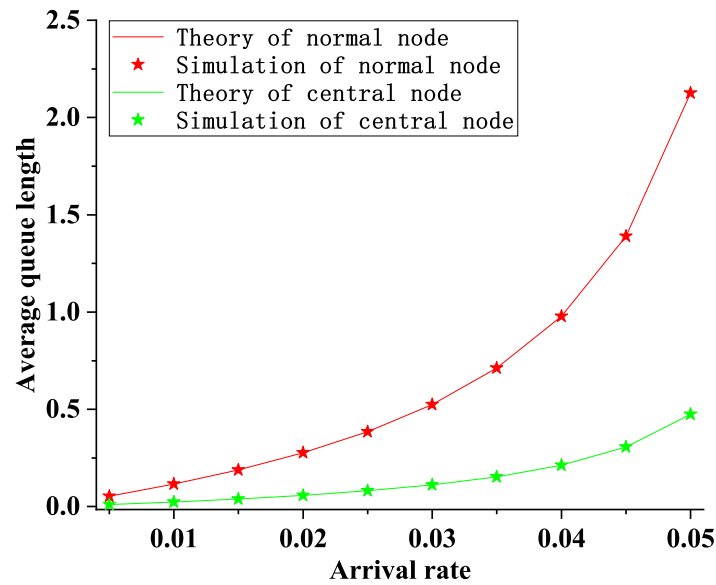


Figure 9. The average queue length of normal node varies with the arrival rate (N = 5).

Figure 10 shows that the average cyclic period varies with the arrival rate. It can be seen that the simulated curve and the theoretical value curve show the same trend. Figure 10 shows the average cyclic period proportional to the arrival rate (or polling system throughput). The load is the total load of the normal nodes and does not include the load of the central node. The purpose of this is to facilitate the construction of the simulation graph and to contrast it with other asymmetric models, in addition to the fact that changes in the load of the center node will also affect the system’s characteristics.

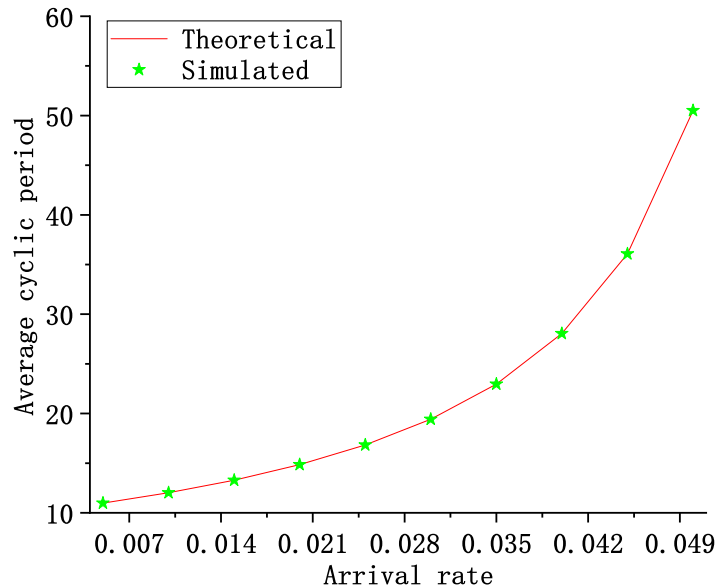


Figure 10. The average cyclic period varies with the arrival rate (N = 5).

Figures 11 and 12 are nodes of average delay affected by load changes. The theoretical value of the average delay is consistent with the experimental value, and the error is negligible if the number of cycles is large enough. Comparing node 3 with the other nodes shows that the average delay of the node is longer when both the service time and the switching time of the node are more significant. When only one parameter is larger, the average delay of the node is insignificant. For the same system, as the load on the system increases, the average delay also increases, and this relationship continues as the number of nodes in the system increases. The average delay of the central node is smaller than the

average delay of the normal nodes for the same load, which is a good indication of the effectiveness of using a mixed polling service for prioritization.

The above analysis shows that the system’s throughput is limited. Comparing Figures 11 and 12, we can see that when the system throughput increases, the average delay of the system increases rapidly. When the throughput increases to the threshold value, the system will not work correctly (the delay is much higher than the actual requirement). However, the impact of increasing system throughput on the central node’s average latency is less than the average latency of the normal nodes, indicating that the system meets the priority requirements.

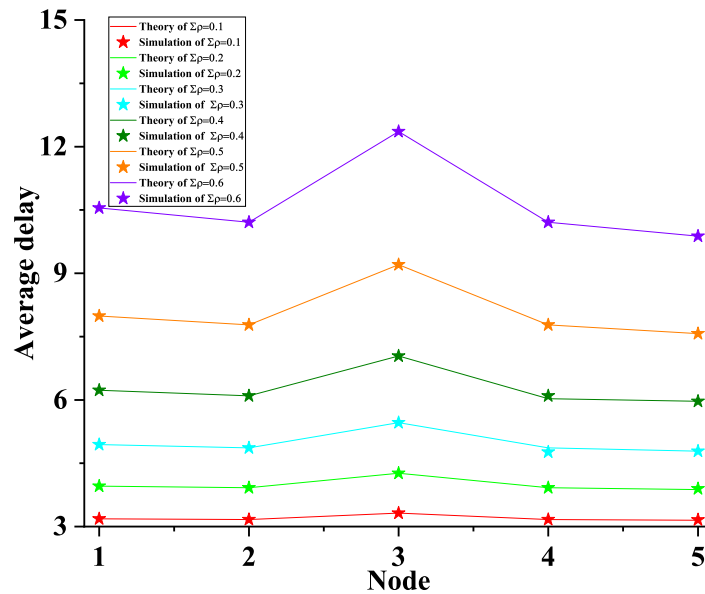


Figure 11. The average waiting time of node i queues is affected by load changes ($N = 5$).

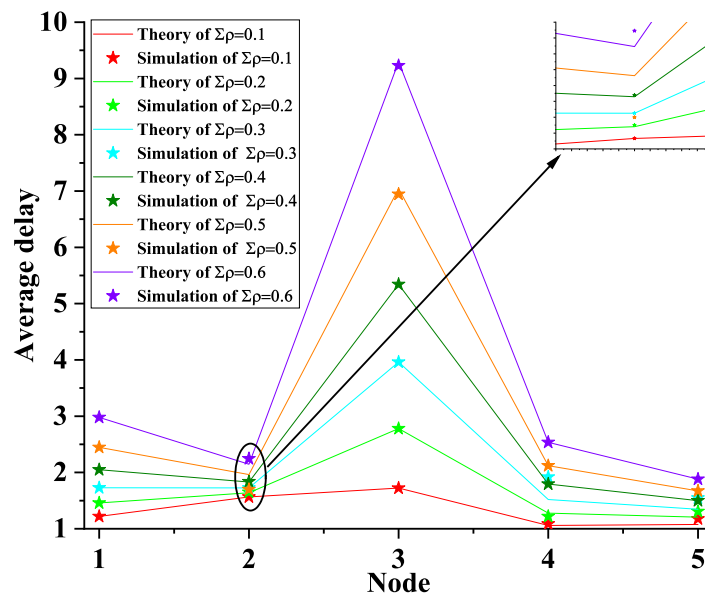


Figure 12. The average waiting time of h varies is affected by load changes ($N = 5$).

Figure 13 shows a plot of the average delay as a function of the arrival rate for the two-level priority and symmetric polling system. Although the average delay of the central node is much lower than that of the average node, the delay derived by any node is the same, which cannot be adapted to multi-task scheduling, and the system could be more resilient.

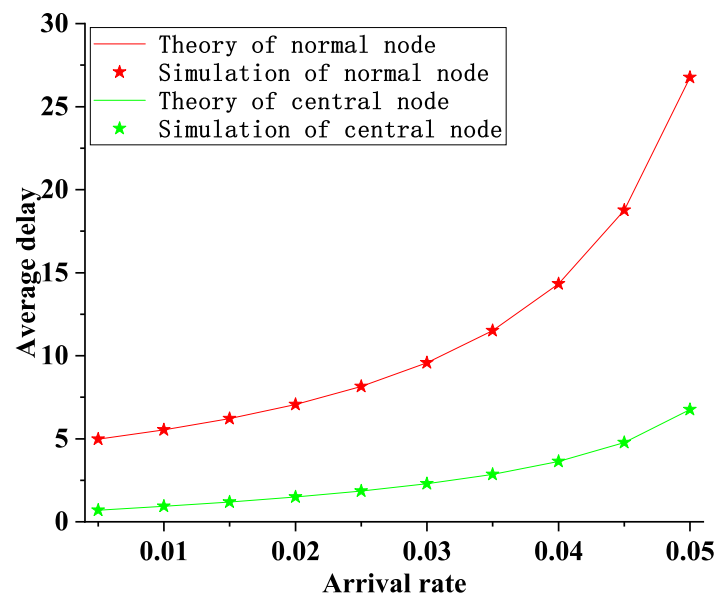


Figure 13. The average delay of normal node varies with the arrival rate ($N = 5$).

7. Conclusions

With the entry into the 5G/6G era, IoT services have diversified. Taking the MEC network service as an example, to reduce edge server energy consumption and improve utilization, we propose a two-level polling MAC protocol with a differentiated priority that accurately calculates the average length and average period using the probability-generating functions and the average delay using the period cycle approximation. Moreover, we used MATLAB R2022a to simulate the experimental and theoretical values and conformity with reality. Two-level priority and asymmetric polling systems are more relevant than symmetric polling systems. Experiments have shown that our proposed two-level priority asymmetric polling system yields node-specific latency, which increases the system's availability, reduces the latency of the central node, and improves the total system throughput. Asymmetric polling systems with multiple assignments are closer to practical applications, computers, communications, the Internet of Things, industrial control, etc. This paper investigates a polling system with priority for a central node exhaustive service and an exhaustive asymmetric service for normal nodes. Moreover, the second-order partial derivative of the Probability-generating function is solved using a periodic query approximation. In future work, finding a more accurate method for solving the second-order partial derivatives of the Probability-generating function is the focus of the work.

Author Contributions: Conceptualization, Z.Y. and H.D.; methodology, H.D.; software, X.W.; validation, X.W., Z.Y. and H.D.; formal analysis, X.W.; investigation, Z.Y.; resources, H.D.; data curation, X.W.; writing—original draft preparation, X.W.; writing—review and editing, Z.Y.; visualization, Z.Y.; supervision, X.W.; project administration, X.W.; funding acquisition, Z.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was supported by the Yunnan Provincial Talent Support Program for Industrial Innovation Talents and the Expert Workstation Project of Wu Zhonghai, Yunnan Province (Project No. 202305AF150045), and the National Natural Science Foundation of China (NSFC) 61461054 and NSFC 61461053.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset and code is available from the corresponding author on reasonable request.

Acknowledgments: The authors would like to thank the editor and the anonymous reviewers for their helpful and constructive comments, which have helped us to improve the manuscript significantly.

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

References

1. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376. [\[CrossRef\]](#)
2. Zhang, J.; Letaief, K.B. Mobile edge intelligence and computing for the internet of vehicles. *Proc. IEEE* **2020**, *108*, 246–261. [\[CrossRef\]](#)
3. Chen, Y.; Zhao, F.; Chen, X.; Wu, Y. Efficient Multi-Vehicle Task Offloading for Mobile Edge Computing in 6G Networks. *IEEE Trans. Veh. Technol.* **2022**, *71*, 4584–4595. [\[CrossRef\]](#)
4. Zhao, H.; Deng, S.; Liu, Z.; Yin, J.; Dustdar, S. Distributed Redundant Placement for Microservice-based Applications at the Edge. *IEEE Trans. Serv. Comput.* **2022**, *15*, 1732–1745. [\[CrossRef\]](#)
5. Mao, S.; Wu, J.; Liu, L.; Lan, D.; Taherkordi, A. Energy-Efficient Cooperative Communication and Computation for Wireless Powered Mobile-Edge Computing. *IEEE Syst. J.* **2022**, *16*, 287–298. [\[CrossRef\]](#)
6. Chen, X.; Zhang, J.; Lin, B.; Chen, Z.; Wolter, K.; Min, G. Energy-Efficient Offloading for DNN-Based Smart IoT Systems in Cloud-Edge Environments. *IEEE Trans. Parallel Distrib. Syst.* **2022**, *33*, 683–697. [\[CrossRef\]](#)
7. Wang, X.; Li, J.; Ning, Z.; Song, Q.; Guo, L.; Guo, S.; Obaidat, M.S. Wireless powered mobile edge computing networks: A survey. *Acm Comput. Surv.* **2023**, *55*, 263. [\[CrossRef\]](#)
8. Hou, P.; Li, B.; Wang, Z.; Ding, H. Joint hierarchical placement and configuration of edge servers in C-V2X. *Hoc Netw.* **2022**, *131*, 102842.
9. Ghafoor, S.; Boujnah, N.; Rehmani, M.H.; Davy, A. MAC Protocols for Terahertz Communication: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2236–2282. [\[CrossRef\]](#)
10. Su, H.; Pan, M.-S.; Chen, H.; Liu, X. MDP-Based MAC Protocol for WBANs in Edge-Enabled eHealth Systems. *Electronics* **2023**, *12*, 947. [\[CrossRef\]](#)
11. Hu, L.; Wang, W.; Zhong, S.; Guo, H.; Li, J.; Pan, Y. APP-Aware MAC Scheduling for MEC-Backed TDM-PON Mobile Fronthaul. *IEEE Commun. Lett.* **2023**, *27*, 1175–1179. [\[CrossRef\]](#)
12. Li, B.; Hou, F.; Yang, G.; Zhao, H.; Chen, S. Data Analysis-Oriented Stochastic Scheduling for Cost Efficient Resource Allocation in NFV Based MEC Network. *IEEE Trans. Veh. Technol.* **2023**, *72*, 6695–6708. [\[CrossRef\]](#)
13. Tian, J.; Wang, D.; Zhang, H.; Wu, D. Service Satisfaction-Oriented Task Offloading and UAV Scheduling in UAV-Enabled MEC Networks. *IEEE Trans. Wirel. Commun.* **2023**. [\[CrossRef\]](#)
14. Jiang, X.; Hou, P.; Zhu, H.; Li, B.; Wang, Z.; Ding, H. Dynamic and intelligent edge server placement based on deep reinforcement learning in mobile edge computing. *Hoc Netw.* **2023**, *145*, 103172. [\[CrossRef\]](#)
15. Lin, L.; Zhou, W.A.; Yang, Z.; Liu, J. Deep reinforcement learning-based task scheduling and resource allocation for NOMA-MEC in Industrial Internet of Things. *Peer-Peer Netw. Appl.* **2023**, *16*, 170–188. [\[CrossRef\]](#)
16. Niu, L.; Chen, X.; Zhang, N.; Zhu, Y.; Yin, R.; Wu, C.; Cao, Y. Multi-Agent Meta-Reinforcement Learning for Optimized Task Scheduling in Heterogeneous Edge Computing Systems. *IEEE Internet Things J.* **2023**, *10*, 10519–10531. [\[CrossRef\]](#)
17. Al-Hammadi, I.; Li, M.; Islam, S.M.N. Independent tasks scheduling of collaborative computation offloading for SDN-powered MEC on 6G networks. *Soft Comput.* **2023**, *27*, 9593–9617. [\[CrossRef\]](#)
18. Qin, Z.; Wang, H.; Wei, Z.; Qu, Y.; Xiong, F.; Dai, H.; Wu, T. Task Selection and Scheduling in UAV-Enabled MEC for Reconnaissance With Time-Varying Priorities. *IEEE Internet Things J.* **2021**, *8*, 17290–17307. [\[CrossRef\]](#)
19. Lin, D.; Hu, S.; Gao, Y.; Tang, Y. Optimizing MEC Networks for Healthcare Applications in 5G Communications With the Authenticity of Users' Priorities. *IEEE Access* **2019**, *7*, 88592–88600. [\[CrossRef\]](#)
20. Jiang, L.; Chang, X.; Mišić, J.; Mišić, V.B.; Bai, J. Understanding MEC empowered vehicle task offloading performance in 6G networks. *Peer-to-Peer Netw. Appl.* **2022**, *15*, 1090–1104. [\[CrossRef\]](#)
21. Zhou, W.; Fan, L.; Zhou, F.; Li, F.; Lei, X.; Xu, W.; Nallanathan, A. Priority-Aware Resource Scheduling for UAV-Mounted Mobile Edge Computing Networks. *IEEE Trans. Veh. Technol.* **2023**, *72*, 9682–9687. [\[CrossRef\]](#)
22. Paymard, P.; Mokari, N. Resource allocation in PD-NOMA-based mobile edge computing system: Multiuser and multitask priority. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e3631. [\[CrossRef\]](#)
23. Shukla, P.; Pandey, S.; Agarwal, D. An Efficient Offloading Technique using DQN for MEC-IoT Networks. In Proceedings of the 2023 6th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 3–4 March 2023; pp. 1–7. [\[CrossRef\]](#)
24. Sharif, Z.; Jung, L.T.; Ayaz, M.; Yahya, M.; Pitafi, S. Priority-based task scheduling and resource allocation in edge computing for health monitoring system. *J. King Saud Univ.-Comput. Inf. Sci.* **2023**, *35*, 544–559. [\[CrossRef\]](#)
25. Boon, M.A.A.; van der Mei, R.D.; Winands, E.M.M. Applications of polling systems. *Appl. Poll. Syst.* **2011**, *16*, 67–82. [\[CrossRef\]](#)
26. Suman, R.; Krishnamurthy, A. Analysis of tandem polling queues with finite buffers. *Ann. Oper. Res.* **2020**, *293*, 343–369. [\[CrossRef\]](#)

27. Suman, R.; Krishnamurthy, A. Analysis of two-station polling queues with setups using continuous time markov chain. *arXiv* **2022**, arXiv:2202.10045. [[CrossRef](#)]
28. Lê, A.T. A Tandem Queueing Model to Optimize the Efficiency of Public Administrative Services. Master's Thesis, Vietnam National University, Hanoi, Vietnam, 2021.
29. Miculescu, D.; Karaman, S. Polling-Systems-Based Autonomous Vehicle Coordination in Traffic Intersections With No Traffic Signals. *IEEE Trans. Autom. Control* **2020**, *65*, 680–694. [[CrossRef](#)]
30. Li, S.; Shu, K.; Chen, C. Planning and Decision-making for Connected Autonomous Vehicles at Road Intersections: A Review. *Chin. J. Mech. Eng.* **2021**, *34*, 133. [[CrossRef](#)]
31. Talamali, M.S.; Saha, A.; Marshall, J.A.; Reina, A. When less is more: Robot swarms adapt better to changes with constrained communication. *Sci. Robot.* **2021**, *6*, eabf1416. [[CrossRef](#)]
32. Zakir, R.; Dorigo, M.; Reina, A. Robot Swarms Break Decision Deadlocks in Collective Perception Through Cross-Inhibition. In *Swarm Intelligence; ANTS 2022; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2022; Volume 13491*. [[CrossRef](#)]
33. Nasser, N.; Fadlullah, Z.M.; Fouda, M.M.; Ali, A.; Imran, M. A lightweight federated learning based privacy preserving B5G pandemic response network using unmanned aerial vehicles: A proof-of-concept. *Comput. Netw.* **2022**, *205*, 108672. [[CrossRef](#)]
34. Chour, K.; Reddinger, J.P.; Dotterweich, J.; Childers, M.; Humann, J.; Rathinam, S.; Darbha, S. An agent-based modeling framework for the multi-UAV rendezvous recharging problem. *Robot. Auton. Syst.* **2023**, *166*, 104442. [[CrossRef](#)]
35. Kumar, S.; Raw, R.S.; Bansal, A. LoCaL: Link-optimized cone-assisted location routing in flying ad hoc networks. *Int. J. Commun. Syst.* **2023**, *36*, e5375. [[CrossRef](#)]
36. Yang, Z.; Zheng, H.; Ding, H. Research on polling MAC protocol for multi-robot system in WLAN. *Appl. Res. Comput.* **2022**, *39*, 1178–1182. [[CrossRef](#)]
37. Wang, Z.; Ding, H.; Li, B.; Bao, L.; Yang, Z. An energy efficient routing protocol based on improved artificial bee colony algorithm for wireless sensor networks. *IEEE Access* **2020**, *8*, 133577–133596. [[CrossRef](#)]
38. Zhou, W.; Xia, J.; Zhou, F.; Fan, L.; Lei, X.; Nallanathan, A.; Karagiannidis, G.K. Profit maximization for cache-enabled vehicular mobile edge computing networks. *IEEE Trans. Veh. Technol.* **2023**. [[CrossRef](#)]
39. Boxma, O.J.; Kella, O.; Kosinski, K.M. Queue lengths and workloads in polling systems. *Oper. Res. Lett.* **2011**, *39*, 401–405. [[CrossRef](#)]
40. Chu, Y.Q.; Liu, Z.M. The impact of priority policy in a two queue markovian polling system with multi-class priorities. In *Proceedings of the 12th International Conference on Queueing Theory and Network Applications, Qinhuaungdao, China, 21–23 August 2017; Springer: China, Switzerland, 2017; pp. 282–296*.
41. Mu, W.-H.; Bao, L.-Y.; Ding, H.-W.; Zhao, Y.-F. An exact analysis of discrete time two-level priority polling system based on multi-times gated service policy. *Acta Electron. Sin.* **2018**, *46*, 276–280.
42. Guan, Z.; Yang, Z.-J.; He, M.; Qian, W.-H. Study on the delay performance of station dependent two-level polling systems. *Acta Autom. Sin.* **2016**, *42*, 1207–1214.
43. Yang, Z.; Mao, L.; Yan, B.; Wang, J.; Gao, W. Performance analysis and prediction of asymmetric two-level priority polling system based on BP neural network. *Appl. Soft Comput.* **2021**, *99*, 106880. [[CrossRef](#)]
44. Yang, Z.J.; Zhao, D.F.; Ding, H.W.; Zhao, Y.F. Research on two-class priority based polling system. *Acta Electron. Sin.* **2009**, *37*, 1452–1456.
45. Siddiqui, S.; Ghani, S. Towards dynamic polling: Survey and analysis of channel polling mechanisms for wireless sensor networks. In *Proceedings of the 2016 International Conference on Intelligent Systems Engineering, Islamabad, Pakistan, 15–17 January 2016; IEEE: Piscataway Township, NJ, USA, 2016; pp. 356–363*.
46. Rehman, M.U.; Drieberg, M.; Badruddin, N. Probabilistic polling MAC protocol with unslotted CSMA for wireless sensor networks (WSNs). In *Proceedings of the 2014 5th International Conference on Intelligent and Advanced Systems, Kuala Lumpur, Malaysia, 3–5 June 2014; IEEE: Piscataway Township, NJ, USA, 2014; pp. 1–5*.
47. Siddiqui, S.; Ghani, S.; Khan, A.A. ADP-MAC: An adaptive and dynamic polling based mac protocol for wireless sensor networks. *IEEE Sens. J.* **2018**, *18*, 860–874. [[CrossRef](#)]
48. Zhao, D.; Zheng, S. Analysis of a polling model with exhaustive service. *Acta Electron. Sin.* **1994**, *22*, 102–107.
49. He, M.; Guan, Z.; Bao, L.; Ge, J. Mean cyclic period analysis of polling access control for wireless sensor networks. *Chin. J. Sci. Instrum.* **2016**, *37*, 2637–2644.

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.