



Alexandria University  
**Alexandria Engineering Journal**

[www.elsevier.com/locate/aej](http://www.elsevier.com/locate/aej)  
[www.sciencedirect.com](http://www.sciencedirect.com)



# A UCB-based dynamic CoAP mode selection algorithm in distribution IoT



Shidong Zhang, Xinhong You<sup>\*</sup>, Pengping Zhang, Min Huang, Shuai Li

*Electric Power Research Institute of State Grid Shandong Electric Power Company, Jinan, China*

Received 24 February 2021; revised 11 April 2021; accepted 29 April 2021  
Available online 05 June 2021

## KEYWORDS

Constrained application protocol (CoAP);  
Upper confidence bound (UCB);  
Distribution internet of things (IoT);  
Dynamic mode selection;  
Differentiated QoS requirements

**Abstract** Lightweight constrained application protocol (CoAP) has emerged as a common communication protocol for resource-constrained equipment in distribution internet of things (IoT). CoAP introduces two modes for data transmission, i.e., non-confirmed mode for reducing transmission delay and confirmed mode for reducing packet-loss ratio, which can be dynamically selected to satisfy the service requirements. However, there are still some challenges in dynamic CoAP mode selection, including incomplete information and differentiated quality of service (QoS) requirements of distributed IoT services. In this paper, we propose a upper confidence bound (UCB)-based dynamic CoAP mode selection algorithm for data transmission to address these challenges. The simulation results validate that, compared with the fixed mode selection algorithm, the proposed algorithm can flexibly balance the tradeoff between packet-loss ratio and transmission delay as well as satisfy the differentiated QoS in distribution IoT.

© 2021 THE AUTHORS. Published by Elsevier BV on behalf of Faculty of Engineering, Alexandria University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## 1. Introduction

With the rapid development of internet of things (IoT) technology in the distribution network, the effective information interaction among multifarious equipment is imperative to realize the comprehensive perception, lightweight communication, and reliable operation of distribution services, where communication protocol plays a critical role [1,2]. From the perspective of the application layer, the protocols for distribution IoT include message queue telemetry transport (MQTT)

[3], extensive messaging and presence protocol (XMPP) [4], advanced message queuing protocol (AMQP) [5], and constrained application protocol (CoAP) [6,7]. Particularly, MQTT, XMPP, and AMQP rely on transport control protocol (TCP), which has some inherent limitations that make it inefficient in certain restricted environments, such as large delay caused by triple handshakes, slow start, and header congestion [8,9]. Compared with the aforementioned protocols, CoAP is based on user datagram protocol (UDP), which is designed for constrained nodes and networks [10]. In addition, CoAP provides two modes for data transmission, which can be flexibly selected to balance the packet-loss ratio and transmission delay for satisfying differentiated quality of service (QoS) requirements of distribution IoT services [11,12].

To be specific, depending on whether there is a retransmission mechanism, CoAP introduces two modes for data

<sup>\*</sup> Corresponding author at: Electric Power Research Institute of State Grid Shandong Electric Power Company, Jinan 250003, China.  
E-mail address: [sddky\\_coap@163.com](mailto:sddky_coap@163.com) (X. You).

Peer review under responsibility of Faculty of Engineering, Alexandria University.

transmission, i.e., non-confirmed mode and confirmed mode. For the non-confirmed mode without data retransmission mechanism, the server transmits data unidirectionally to the client at fixed intervals, regardless of whether the client receives the packet or not [13–15]. As a result, the non-confirmed mode reduces the transmission delay at a cost of increasing the packet-loss ratio, which may degrade reliability performance. For the confirmed mode adopting retransmission mechanism, the server will transmit the next packet only if it receives the acknowledgment (ACK) packet sent by the client. If the ACK packet is not received within the specified time, the retransmission starts. Moreover, the retransmission terminates until it reaches the maximum number of retransmission or the packet is received by the client successfully. The retransmission mechanism may lead to a larger transmission delay. Nevertheless, it can efficiently reduce the packet-loss ratio to provide a high communication reliability [5,16]. Therefore, a dynamic mode selection method is necessary for data transmission in CoAP to balance the tradeoff between transmission delay and packet-loss ratio.

However, there are still some critical technical challenges in dynamic CoAP mode selection. First, the mode selection optimization is based on time-varying context information, including channel state information (CSI) and bandwidth, which is unavailable due to the prohibitive signaling overhead [17,18]. Therefore, how to realize the optimal mode selection decision under incomplete information remains a challenge. Second, various distribution IoT services have differentiated QoS requirements for packet-loss ratio and transmission delay. For example, services that have the characteristics of large data flow, such as state monitoring and emergency communication, which impose a more stringent requirement on transmission delay [19]. On the other hand, services like meter reading management and load forecasting impose a more stringent requirement on packet-loss ratio than transmission delay, where the high packet-loss ratio may cause frame loss and communication interruption. Therefore, how to balance transmission delay and packet-loss ratio to satisfy differentiated QoS requirements of distribution IoT services through dynamic CoAP mode selection remains an open issue.

Numerous works have investigated CoAP mode selection. In [19], Rolando Herrero proposed a dynamic mode selection mechanism that takes packet-loss ratio as an input parameter to extend battery life while minimizing transmission delay and packet-loss ratio to satisfy QoS requirements. However, the authors have assumed that the complete information is available in this paper, which is impractical in practical CoAP mode selection. Multi-armed bandit (MAB) is usually utilized to cope with the sequential decision problems under incomplete information. Among various MAB algorithms, UCB distinguishes itself from others with its rapid convergence speed and well-balanced tradeoff performance between exploration and exploitation. In [20], Wang et al. proposed an improved high frequency cognitive multi-channel selection algorithm based on UCB to improve the success rate of data transmission. In [21], Liao et al. proposed a UCB-based channel selection algorithm to dynamically balance throughput, energy consumption, and service reliability. However, it cannot comprehensively consider the data transmission delay and packet-loss ratio.

In this paper, we consider the performances in both packet-loss ratio and transmission delay. The mode selection problem

is suitable to be modeled as an MAB problem. Among various MAB algorithms, UCB distinguishes itself from others with its rapid convergence speed and well-balanced tradeoff performance between exploration and exploitation.

The main contributions of this work are summarized as follows:

- *Dynamic Mode Selection under Incomplete Information:* We leverage UCB to observe and interact with environment under incomplete channel information. The proposed algorithm can learn the optimal dynamic CoAP mode selection strategy based on only local information.
- *Low-Complexity and Dynamic CoAP Mode Selection:* A low-complexity and dynamic mode selection mechanism is developed to minimize the weighted sum of transmission delay and packet-loss ratio based on UCB algorithm.
- *Extensive Performance Evaluation:* A lot of simulations have been carried out to demonstrate the advantages of the proposed algorithm compared with fixed mode operation. Specifically, the effects of different parameters in terms of packet number, packet-loss threshold, etc, have been illustrated to evaluate the superior performances of the proposed algorithm.

The remainder of this paper is organized as follows. The system model and problem formulation are presented in Section 2. The proposed algorithm is introduced in Section 3. In Section 4, simulation results are given. Section 5 concludes the paper.

## 2. System model and problem formulation

### 2.1. System model

As shown in Fig. 1, the CoAP is adopted between the distribution IoT equipment and the gateway or IoT Hub. The distribution IoT equipment sends a communication request to the gateway through the CoAP protocol, and the gateway forwards the relevant configuration information to the distribution IoT equipment. The real-time data collected by the IoT equipment is reported to IoT Hub through CoAP to realize the data interaction and standardized access. The significance of parameters is shown in Table 1.

CoAP presents two main modes for data transmission, i.e., non-confirmed mode and confirmed mode. The non-confirmed mode does not have a retransmission mechanism, where the gateway unidirectionally sends a packet to client without any guarantee of successful delivery. Therefore, this mode has lower delay and higher packet-loss ratio [22]. The confirmed mode has a retransmission mechanism, where the gateway decides whether to retransmit the packet according to the result feedback by the client within the maximum retransmission number. Therefore, this mode has a lower packet-loss ratio, while increasing transmission delay and energy consumption [23]. Consequently, it is crucial to choose the appropriate CoAP mode according to different application scenarios and QoS requirements.

In this paper, we assume that there are a total of  $I$  large data packets, and each large packet contains  $J$  smaller packets of equal size. Denote the mode indicator as  $m$ , where  $m = 1$  indicates non-confirmed mode and  $m = 2$  indicates confirmed

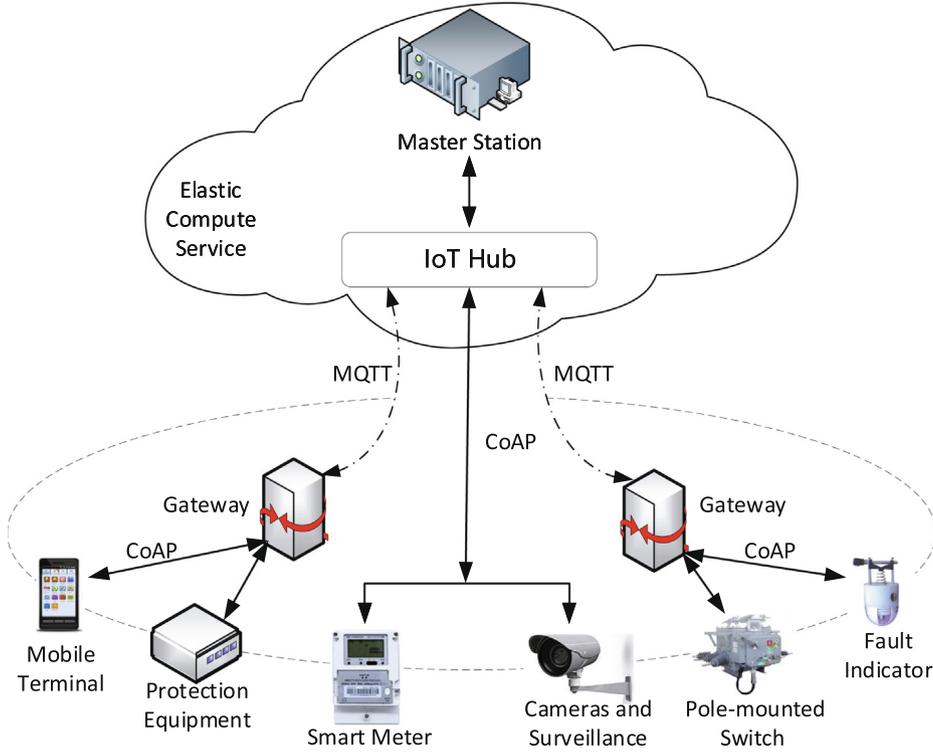


Fig. 1 System model.

mode. The transmission delays of the  $i$ -th large packet and the  $j$ -th small packet of the  $i$ -th large packet in mode  $m$  are denoted as  $T_i^m (1 \leq i \leq I)$  and  $t_{ij}^m (1 \leq j \leq J)$ , respectively. For the same large packet, the selected transmission mode remains unchanged, and the transmission mode can dynamically switch between any two large packets. Then, we define a mode selection indicator  $x_{i,m}$ , where  $x_{i,m} = 1$  if the  $i$ -th large packet selects mode  $m$ , and  $x_{i,m} = 0$  otherwise. Furthermore, the channel state remains unchanged during the transmission of a small packet, but randomly changes across the transmission of different small packets [24]. Specifically, when the small packet adopts mode 2 for data transmission, the retransmission process is also recorded as a new small packet transmission process, i.e., the channel state varies randomly compared with the previous one. Taking the  $i$ -th large packet as an example, Fig. 2(a) and Fig. 2(b) describe the data transmission processes of  $j$ -th small packets of the  $i$ -th large packet in the non-confirmed mode and confirmed mode, respectively. For the non-confirmed mode without data retransmission mechanism, the gateway transmits data unidirectionally to the client at fixed intervals, regardless of whether the client receives the packet or not. For the confirmed mode with the data retransmission mechanism, the client will return the ACK packet to the gateway after receiving the small packet, and if the ACK packet is not received within  $t_0$ , i.e., the maximum waiting time for the ACK packet, the gateway would retransmit the same small packet. In addition, the gateway will transmit the next small packet when it receives the ACK packet sent by the client or when it reaches the maximum retransmission number.

### 2.1.1. Channel model

According to [25], the channel gain for the  $n$ -th retransmission of the  $j$ -th small packet of the  $i$ -th large packet is given by

$$g_{i,j,n} = \frac{|H_{i,j,n}|^2}{N_{i,j,n}}, \quad (1)$$

where  $H_{i,j,n}$  represents the channel frequency response of the  $n$ -th retransmission of the  $j$ -th small packet of the  $i$ -th large packet in the mode  $m$ ,  $N_{i,j,n}$  represents the total noise power of the  $n$ -th retransmission of the  $j$ -th small packet of the  $i$ -th large packet.

### 2.1.2. Packet-loss ratio

In the confirmed mode, the vector set of retransmission is denoted as  $\mathcal{A} = \{a_{i,j,1}^m, \dots, a_{i,j,n}^m, \dots, a_{i,j,N}^m\}$ ,  $a_{i,j,n}^m \in \{0, 1\}$ , where  $N$  represents the maximum number of retransmission [25]. The initial value is set as  $a_{i,j,n}^m = 0$ , which indicates that the  $n$ -th retransmission of the  $j$ -th small packet of the  $i$ -th large packet is successful, and  $a_{i,j,n}^m = 1$  otherwise.

If the current channel quality  $p_{i,j,n}g_{i,j,n}$  is larger than the threshold  $G_{th}$ , it means that the packet successfully arrives at the client. Therefore, the retransmission result of the non-confirmed mode is expressed as

$$a_{i,j,n}^1 = \begin{cases} 0, & p_{i,j,n}g_{i,j,n} \geq G_{th}, \\ 1, & p_{i,j,n}g_{i,j,n} < G_{th}. \end{cases} \quad (2)$$

where  $p_{i,j,n}$  denotes the transmission power of the  $n$ -th retransmission of the  $j$ -th small packet of the  $i$ -th large packet.

The retransmission indicator variable of the confirmed mode is expressed as

$$a_{i,j,n}^2 = \begin{cases} 0, & p_{i,j,n}g_{i,j,n} \geq G_{th}, \\ 1, & p_{i,j,n}g_{i,j,n} < G_{th}. \end{cases} \quad (3)$$

**Table 1** Significance of parameters.

Parameters	Value
$I$	The total number of large data packets
$J$	The total number of small packages in each large packet
$\mathcal{A}$	The vector set of retransmission, used to mark whether the packet is retransmitted.
$\mathcal{B}$	The vector set of ACK transmission, used to mark whether the ACK packet is transmitted.
$G_{th}$	Channel gain threshold
$S$	The small packet size (bit)
$S_{back}$	The ACK packet size (bit)
$m$	The mode indicator
$n$	The $n$ -th retransmission
$N$	The maximum number of retransmission
$B$	Channel bandwidth
$g_{i,j,n}$	The channel gain of the $n$ -th retransmission of the $j$ -th small packet of the $i$ -th large packet
$g_{i,j,n,back}$	The channel gain of the $n$ -th retransmission of the $j$ -th ACK packet of the $i$ -th large packet
$H_{i,j,n}$	The channel frequency response of the $n$ -th retransmission of the $j$ -th small packet of the $i$ -th large packet in the mode $m$
$N_{i,j,n}$	The total noise power of the $n$ -th retransmission of the $j$ -th small packet of the $i$ -th large packet
$p_{i,j,n}$	The transmission power of the $n$ -th retransmission of the $j$ -th small packet of the $i$ -th large packet
$p_{i,j,n,back}$	The transmission power of the $n$ -th transmission of the $j$ -th ACK packet of the $i$ -th large packet
$a_{i,j,n}^m$	Indicator function, indicating the retransmission result
$b_{i,j,n}^2$	Indicator function, indicating the ACK transmission result
$x_{i,m}$	Indicator function, indicating whether the $i$ -th large packet selects the mode $m$
$a_{i,j}^m$	Indicator function, indicating the packet loss of the $j$ -th small packet of the $i$ -th large packet
$Q_i^m$	The packet-loss ratio of the $i$ -th large packet in the mode $m$
$t_{i,j}^m$	The transmission delay of the $j$ -th small packet of the $i$ -th large packet in the mode $m$
$t_0$	The maximum waiting time for the ACK packet
$T_i^m$	The transmission delay of the $i$ -th large packet
$\alpha_i^n$	The total number of small packet loss of the $i$ -th large packet
$V$	The weights of the packet-loss ratio in the optimization goal
$\psi_i$	The mode that maximizes the $i$ -th large packet preference value
$k_{i,m}$	The number of times that mode $m$ is selected when the $i$ -th large packet is transmitted

where  $p_{back}$  denotes the transmission power of transmitting ACK packet of the  $n$ -th retransmission of the  $j$ -th small packet of the  $i$ -th large packet. And  $g_{i,j,n,back}$  is the channel gain of ACK packet transmission. There are two conditions which can result the  $j$ -th small packet of the  $i$ -th large packet retransmission, i.e., the channel quality  $p_{i,j,n}g_{i,j,n}$  is inferior, or the ACK packet transmission is failed.

In the confirmed mode, the retransmission terminates when the ACK packet sent by the client successfully transmits or it

reaches the maximum number of retransmission. The vector set of ACK packet transmission is denoted as  $\mathcal{B} = \{b_{i,j,1}^2, \dots, b_{i,j,n}^2, \dots, b_{i,j,N}^2\}$ ,  $b_{i,j,n}^2 \in \{0, 1\}$ . The initial value is set as  $b_{i,j,n}^2 = 0$ , which indicates that the ACK packet of the  $n$ -th retransmission of the  $j$ -th small packet of the  $i$ -th large packet is transmitted successful, and  $b_{i,j,n}^2 = 1$  otherwise. The transmission result of the ACK packet in the confirmed mode is expressed as

$$b_{i,j,n}^2 = \begin{cases} 0, & p_{i,j,n}g_{i,j,n} \geq G_{th} \text{ and } p_{back}g_{i,j,n,back} \geq G_{th}, \\ 1, & p_{i,j,n}g_{i,j,n} \geq G_{th} \text{ and } p_{back}g_{i,j,n,back} < G_{th}. \end{cases} \quad (4)$$

Hence, the packet loss of the  $j$ -th small packet of the  $i$ -th large packet is expressed as

$$a_{i,j}^m = \begin{cases} 0, & m = 1, p_{i,j,1}g_{i,j,1} \geq G_{th}, \\ 1, & m = 1, p_{i,j,1}g_{i,j,1} < G_{th}, \\ \prod_{n=1}^N a_{i,j,n}^m, & m = 2. \end{cases} \quad (5)$$

The total number of small packet loss of the  $i$ -th large packet is

$$\alpha_i^m = \sum_{j=1}^J a_{i,j}^m. \quad (6)$$

The packet-loss ratio of the  $i$ -th large packet is

$$Q_i^m = \frac{\alpha_i^m}{J}. \quad (7)$$

### 2.1.3. Transmission delay

The transmission delay of the  $j$ -th small packet of the  $i$ -th large packet is given by

$$t_{i,j}^m = \begin{cases} \frac{S}{B \log_2(1 + p_{i,j,1}g_{i,j,1})}, & \text{if } m = 1, \\ t_{i,j}^2, & \text{if } m = 2, \end{cases} \quad (8)$$

where  $B$  denotes the channel bandwidth, and  $S$  represents the small packet size (bit). Here,

$$t_{i,j}^2 = \sum_{n=1}^N a_{i,j,n}^2 \left( \frac{S}{B \log_2(1 + p_{i,j,n}g_{i,j,n})} + t_0 \right) + \sum_{n=1}^N b_{i,j,n}^2 \left( \frac{S}{B \log_2(1 + p_{i,j,n}g_{i,j,n})} + t_0 \right) + \frac{S}{B \log_2(1 + p_{i,j,n}g_{i,j,n}^*)} + \frac{S_{back}}{B \log_2(1 + p_{back}g_{i,j,n}^*,_{back})}, \quad (9)$$

where  $S_{back}$  represents the ACK packet size (bit).  $N$  is the maximum number of retransmission.  $n^*$  is the number of retransmission times. Specifically, the first term of the formula represents the retransmission delay when the small packet retransmission is failed. The second term of the formula represents the transmission delay caused by the ACK packet transmission failure. The third and the fourth terms of the formula denote the transmission delay of the last transmission of the small packet and the ACK packet, respectively.

The transmission delay of the  $i$ -th large packet is given by

$$T_i^m = \sum_{j=1}^J t_{i,j}^m. \quad (10)$$

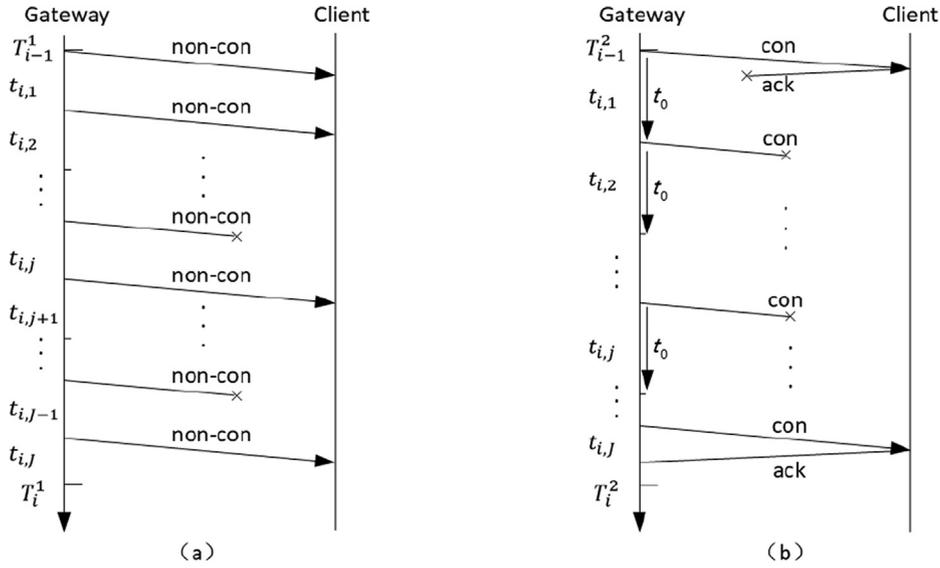


Fig. 2 (a) non-confirmed mode. (b) confirmed mode.

## 2.2. Problem formulation

The purpose of dynamic mode selection is to select the optimal transmission mode and balance the tradeoff between differentiated QoS requirements. Therefore, we set the optimization objective as minimizing the weighted sum of transmission delay and packet-loss ratio. Accordingly, the dynamic mode selection problem is formulated as

$$\begin{aligned}
 \text{P1} : & \min_{\{x_{i,m}\}} \sum_{i=1}^I \sum_{m=1}^2 x_{i,m} (T_i^m + VQ_i^m) \\
 \text{s.t. } & C_1 : x_{i,m} \in \{0, 1\}, \forall i \in \{1, 2, \dots, I\}, \forall m \in \{1, 2\}, \\
 & C_2 : \sum_{m=1}^2 x_{i,m} = 1, \forall i \in \{1, 2, \dots, I\},
 \end{aligned} \quad (11)$$

where  $V$  represents the weight of the packet-loss ratio in the optimization goal. Here,  $C_1$  and  $C_2$  guarantee that each large packet must select a mode, and can only select one mode.

## 3. UCB-based dynamic CoAP mode selection algorithm

The key concept of UCB-based dynamic CoAP mode selection algorithm is to make estimates based on historical observations while simultaneously considering the uncertainty of those observations, i.e., the confidence bound. UCB does not need a large amount of prepared training data, which enables equipment to observe and make mode selection based on the number of selections for modes and its corresponding empirical performance [26]. The proposed algorithm is summarized in Algorithm 1, which consists of three phases, i.e., initialization, estimation and decision making, and learning.

In the first phase of initialization, initialize the variable, and then it is assumed that the equipment transmits the first large packet in mode 1 and the second large packet in mode 2.

In the second phase of estimation and decision making, when transmitting the  $i$ -th large packet, the equipment estimates its preference towards mode  $m$  as [25]

$$\hat{V}_i^m = \frac{1}{T_{i-1}^m + VQ_{i-1}^m} + \sqrt{\frac{2 \ln i}{k_{i-1,m}}},$$

where the first term represents the empirical performance of the mode  $m$ , the denominator of which is the weighted sum of packet-loss ratio and transmission delay to measure the quality of transmission mode. The second term represents the confidence bound, which is designed to balance the tradeoff between exploration and exploitation.  $k_{i,m}$  represents the number of times that mode  $m$  is selected when the  $i$ -th large packet is transmitted.

The first term of the preference value reflects the empirical performances of packet-loss ratio and transmission delay, which is negatively proportional to them, i.e., the mode with lower packet-loss ratio and transmission delay is more preferred. The second term is the estimation uncertainty, which is defined as the square-root of a fraction, where the denominator is the number of times that mode  $m$  has been selected up to the  $i$ -th large packet. Therefore, the mode with a lower weighted sum of packet-loss ratio and transmission delay or a smaller number of selections will have a larger UCB value, thereby resulting in more chances of being selected.

Then, the mode with the maximum estimation value will be selected, which is determined as

$$\psi_i = \arg \max_{m=1,2} \hat{V}_i^m. \quad (13)$$

In the third phase of learning, the equipment updates  $\frac{T_i^m}{T_i^m + VQ_i^m}$  and  $k_{i,m}$  as

$$\frac{T_i^m}{T_i^m + VQ_i^m} = \frac{T_{i-1}^m + VQ_{i-1}^m k_{i-1,m} + (T_i^{\psi_i} + VQ_i^{\psi_i}) x_{i,m}}{k_{i-1,m} + x_{i,m}}, \quad (14)$$

and

$$k_{i,m} = k_{i-1,m} + x_{i,m}. \quad (15)$$

Finally, increase  $i$  to  $i + 1$ , and repeat lines 6~13 until  $i > I$ .

**Algorithm 1.** UCB-Based Dynamic CoAP Mode Selection Algorithm

---

```

1: Input:  $V, G_{th}, B, S$ .
2: Phase 1: Initialization
3: Set  $a_{i,j}^m = 0, x_{i,m} = 0, k_{i,m} = 0$ .
4: Traverse to select two transmission modes.
5: Repeat
6: Phase 2: Estimation and decision making
7: Calculate the estimation value  $\hat{V}_i^m$  as (12).
8: Select the optimal option  $\psi_i$  based on (13).
9: Phase 3: Learning
10: Observe  $g_{i,j,n}$  and compare  $p_{i,j,n}g_{i,j,n}$  with  $G_{th}$ .
11: Calculate  $T_i^m$  and  $Q_i^m$  of the selected transmission mode based on (7) and (10).
12: Update  $T_i^m + VQ_i^m$  based on (14).
13: Update  $k_{i,m}$  based on (15).
14: Until  $i > I$ .

```

---

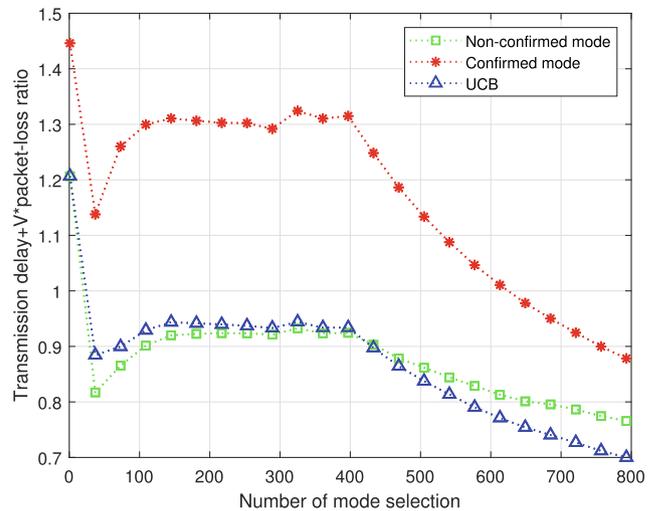
#### 4. Simulation results and discussions

In this section, the performance of the proposed algorithm is evaluated through simulations. We assume that the channel gain is randomly distributed within [4, 11] in the first 400 mode selections, and randomly distributed within [7, 11] in the next 400 mode selections. Two fixed CoAP data transmission modes are used for comparison, i.e., confirmed mode and non-confirmed mode [16]. The simulation parameters are summarized in Table 2 [19].

Fig. 3 shows the weighted sum of transmission delay and packet-loss ratio versus the number of mode selection. It can be seen that the non-confirmed mode performs best in the first 400 mode selections. The reason is that the poor channel quality causes the substantial increase of transmission delay in the confirmed mode. After 400 mode selections, the weighted sums achieved by three mode selection algorithms decrease since the channel quality is improved after the 400-th mode selection which results in the reduction of the transmission delay and the number of retransmissions. Simulation results demonstrate that the proposed algorithm outperforms non-confirmed mode and confirmed mode in weighted sum by 8.58% and 20.30%, respectively. The reason is that the proposed algorithm can

**Table 2** Simulation parameters.

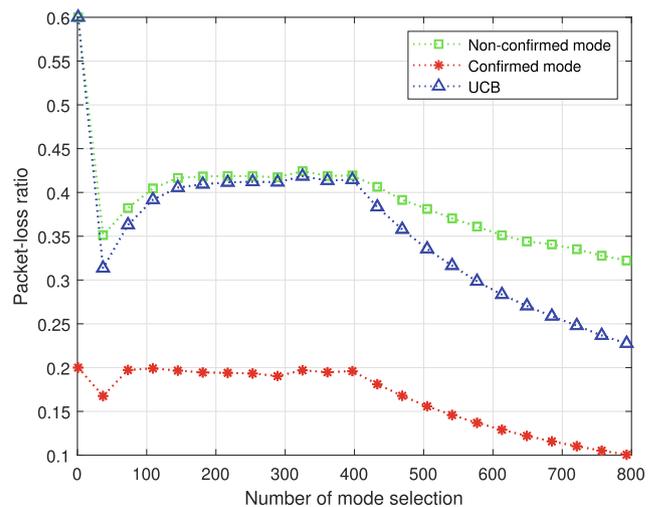
Parameters	Value
Total number of large data packets $I$	800
Total number of small data packets $J$	10
Maximum number of retransmission $N$	5
The weights of the packet-loss ratio $V$	1.5
The timeout $t_0$	0.005 s
Channel bandwidth $B$	0.1 MHz
Channel gain threshold $G_{th}$	0.26
The data of a small packet $S$	1 kbit
The data of a ACK packet $S_{back}$	0.032 bit
Transmission power $p_{i,j,n}$	35 mW

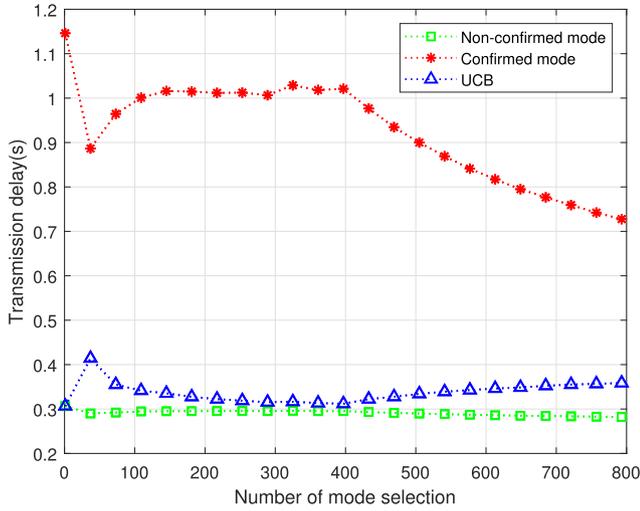

**Fig. 3** The weighted sum of transmission delay and packet-loss ratio versus the number of mode selection.

dynamically balance the tradeoff between transmission delay and packet-loss ratio, and make the optimal mode selection decision.

Figs. 4 and 5 show the packet-loss ratio and the transmission delay versus the number of mode selection, respectively. Since the channel quality has improved after 400 mode selections, it can be seen that the packet-loss ratio and transmission delay have a decreasing trend in all the three mode selection algorithms, while the transmission delay of the confirmed mode has a faster downward trend. The reason is that after 400 mode selections, the channel quality improves, and the number of retransmissions in the confirmed mode is significantly reduced. Simulation results demonstrate that the proposed algorithm can balance the tradeoff between packet-loss ratio and transmission delay when the channel quality is changing.

Fig. 6 shows the optimal mode selection probability versus the number of mode selection. When the number of mode

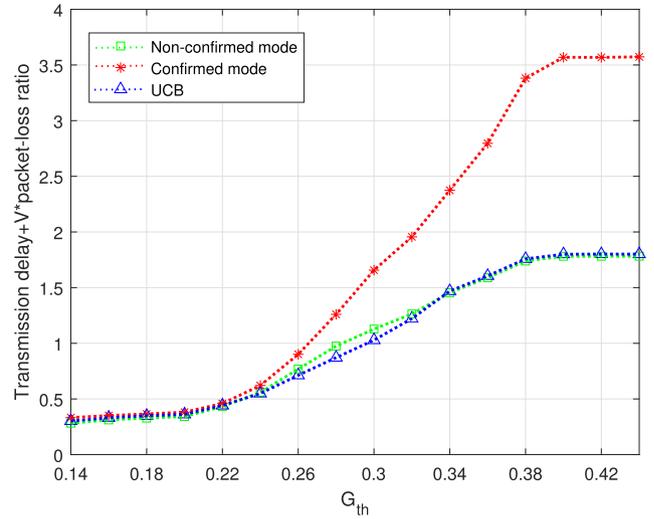

**Fig. 4** Packet-loss ratio versus the number of mode selection.



**Fig. 5** Transmission delay versus the number of mode selection.

selection reaches 400, the optimal mode selection probability will converge to 90%. After the 400 mode selections, due to the dynamic change of the channel state, the proposed algorithm needs to relearn the optimal mode selection strategy. The optimal mode selection probability firstly decreases and then reconverges to 80%. Therefore, the proposed algorithm can adapt to channel changes and learn the optimal transmission mode under incomplete information.

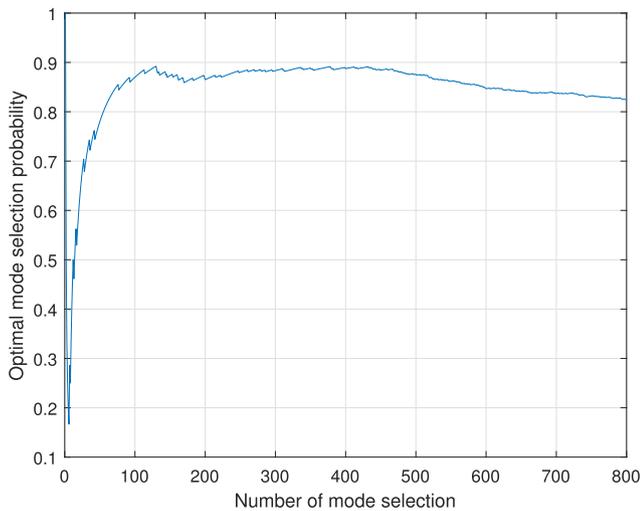
Fig. 7 shows the weighted sum of transmission delay and packet-loss ratio versus threshold  $G_{th}$ . When  $G_{th}$  is low, the confirmed mode performs little difference with the non-confirmed mode since the retransmission occurs infrequently. As the threshold increases, the weighted sums achieved by two modes gradually increase. The increment performed by the confirmed mode is significantly larger than the non-confirmed mode since the transmission delay is the dominant part of the weighted sum when  $V = 1.5$ . After  $G_{th}$  is larger than 0.4, the weighted sums of two modes reach the performance floor since the threshold is too stringent to satisfy regardless



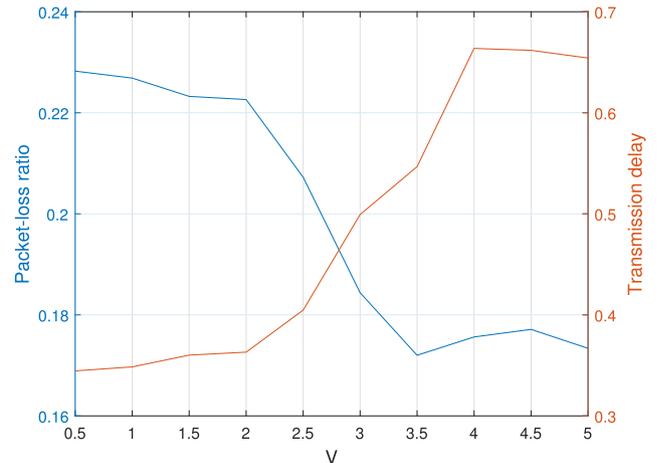
**Fig. 7** The weighted sum of transmission delay and packet-loss ratio versus threshold  $G_{th}$ .

of the number of retransmission. Simulation results demonstrate that the proposed algorithm performs the best when the packet-loss ratio threshold ranges from 0.22 to 0.33 since it can dynamically select the mode with better performance. When  $G_{th}$  is lower than 0.22 or higher than 0.4, the slight performance difference between the proposed algorithm and the non-confirmed stems from the exploration cost.

Fig. 8 shows the impact of  $V$  on packet-loss ratio and transmission delay. As  $V$  increases, the packet-loss ratio shows a downward trend, while the delay shows an upward trend. The reason is that as  $V$  increases, the proposed algorithm puts more emphasis on packet-loss ratio minimization rather than transmission delay reduction. For instance, when  $V$  increases from 2 to 3, the packet-loss ratio is reduced by 22.73%, while the delay is increased by 50.56%. Therefore, the proposed algorithm can dynamically balance the tradeoff between packet-loss ratio and transmission delay by adjusting the value of  $V$ , so as to meet the differentiated QoS requirements of various distribution IoT services.



**Fig. 6** The optimal mode selection probability versus the number of mode selection.



**Fig. 8** The impact of parameter  $V$ .

## 5. Conclusions

In this paper, we minimize the weighted sum of transmission delay and packet-loss ratio in the distribution IoT and propose a UCB-based dynamic CoAP mode selection algorithm for data transmission. The simulation results show that the proposed UCB-based dynamic CoAP mode selection algorithm can dynamically select the best mode under incomplete information and outperforms non-confirmed mode and confirmed mode in weighted sum by 8.58% and 20.30%, respectively. Furthermore, when the threshold of packet-loss ratio is adjusted, the proposed algorithm can always choose the mode with better performance. Our future work will focus on further extend the model derived in this paper and improve learning performance.

## Author Contributions

Conceptualization, Shidong Zhang and Xinhong You; investigation, Shidong Zhang, Xinhong You, Pengping Zhang, Min Huang, and Shuai Li; writing-original draft, Shidong Zhang, Xinhong You, Pengping Zhang, Min Huang, and Shuai Li. All authors have read and agreed to the published version of the manuscript.

## Funding

This research was supported by Science and Technology Project of State Grid Shandong Electric Power Company under Grant No. 2020A-010.

## Declaration of Competing Interest

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

## References

- [1] M. Yu, A. Liu, N.N. Xiong, T. Wang, An intelligent game based offloading scheme for maximizing benefits of iot-edge-cloud ecosystems, *IEEE Internet Things J.* (2020) 1, <https://doi.org/10.1109/JIOT.2020.3039828>.
- [2] C. Xie, G. Li, H. Cai, L. Jiang, N.N. Xiong, Dynamic weight-based individual similarity calculation for information searching in social computing, *IEEE Syst. J.* 11 (1) (2017) 333–344, <https://doi.org/10.1109/JSYST.2015.2443806>.
- [3] D. Eridani, K.T. Martono, A.A. Hanifah, MQTT performance as a message protocol in an IoT based chili crops greenhouse prototyping pp (99) (2019) 184–189. doi:10.1109/ICITISEE48480.2019.9003975.
- [4] N. Nikolov, Research of MQTT, CoAP, HTTP and XMPP IoT communication protocols for embedded systems pp (99) (2020) 1–4. doi:10.1109/ET50336.2020.9238208.
- [5] N.Q. Uy, V.H. Nam, A comparison of amqp and mqtt protocols for internet of things, in: 2019 6th NAFOSTED Conference on Information and Computer Science (NICS), 2019, pp. 292–297. doi:10.1109/NICS48868.2019.9023812.
- [6] H.W. Chen, F.J. Lin, Converging mqtt resources in etsi standards based m2m platform, in: 2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom), 2014, pp. 292–295. doi:10.1109/iThings.2014.52.
- [7] J.E. Luzuriaga, M. Perez, P. Boronat, J.C. Cano, C. Calafate, P. Manzoni, A comparative evaluation of amqp and mqtt protocols over unstable and mobile networks, in: 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), 2015, pp. 931–936. doi:10.1109/CCNC.2015.7158101.
- [8] S. Bendel, T. Springer, D. Schuster, A. Schill, R. Ackermann, M. Ameling, A service infrastructure for the internet of things based on xmpp, in: 2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013, pp. 385–388, <https://doi.org/10.1109/PerComW.2013.6529522>.
- [9] D. Eridani, K.T. Martono, A.A. Hanifah, Mqtt performance as a message protocol in an iot based chili crops greenhouse prototyping, in: 2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), 2019, pp. 184–189, <https://doi.org/10.1109/ICITISEE48480.2019.9003975>.
- [10] D. Garcia-Carrillo, R. Marin-Lopez, Multihop bootstrapping with eap through coap intermediaries for iot, *IEEE Internet Things J.* 5 (5) (2018) 4003–4017, <https://doi.org/10.1109/JIOT.2018.2870984>.
- [11] H. Wang, D. Xiong, P. Wang, Y. Liu, A lightweight xmpp publish/subscribe scheme for resource-constrained iot devices, *IEEE Access* 5 (2017) 16393–16405, <https://doi.org/10.1109/ACCESS.2017.2742020>.
- [12] M. Huang, A. Liu, N.N. Xiong, T. Wang, A.V. Vasilakos, A low-latency communication scheme for mobile wireless sensor control systems, *IEEE Trans. Sys., Man, Cybernet.: Syst.* 49 (2) (2019) 317–332, <https://doi.org/10.1109/TSMC.2018.2833204>.
- [13] N. Naik, Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http, in: 2017 IEEE International Systems Engineering Symposium (ISSE), 2017, pp. 1–7, <https://doi.org/10.1109/SysEng.2017.8088251>.
- [14] J. Esquiagola, L. Costa, P. Calcina, M. Zuffo, Enabling coap into the swarm: A transparent interception coap-http proxy for the internet of things, in: 2017 Global Internet of Things Summit (GIoTS), 2017, pp. 1–6, <https://doi.org/10.1109/GIOTS.2017.8016220>.
- [15] M.L. Kome, F. Cuppens, N. Cuppens-Boulahia, V. Frey, Coap enhancement for a better iot centric protocol: Coap 2.0, in: 2018 Fifth International Conference on Internet of Things: Systems, Management and Security, 2018, pp. 139–146. doi:10.1109/IoTSMS.2018.8554494.
- [16] A. Betzler, C. Gomez, I. Demirkol, J. Paradells, Coap congestion control for the internet of things, *IEEE Commun. Mag.* 54 (7) (2016) 154–160, <https://doi.org/10.1109/MCOM.2016.7509394>.
- [17] F. Haider, C. Wang, B. Ai, H. Haas, E. Hepsaydir, Spectral/energy efficiency tradeoff of cellular systems with mobile femtocell deployment, *IEEE Trans. Veh. Technol.* 65 (5) (2016) 3389–3400, <https://doi.org/10.1109/TVT.2015.2443046>.
- [18] T. Li, N.N. Xiong, J. Gao, H. Song, A. Liu, C. Zhiping, Reliable code disseminations through opportunistic communication in vehicular wireless networks, *IEEE Access* 6 (2018) 55509–55527, <https://doi.org/10.1109/ACCESS.2018.2870928>.
- [19] R. Herrero, Dynamic coap mode control in real time wireless iot networks, *IEEE Internet Things J.* 6 (1) (2019) 801–807, <https://doi.org/10.1109/JIOT.2018.2857701>.
- [20] D. Wang, Y. Wang, Q. Sun, Improved hf cognitive multi-channel selection algorithm based on ucb, *J. Southeast Univ.* 49 (5) (2019) 897–903, <https://doi.org/10.3969/j.issn.1001-0505.2019.05.012>.
- [21] H. Liao, Z. Zhou, X. Zhao, L. Zhang, S. Mumtaz, A. Jolfaei, S. H. Ahmed, A.K. Bashir, Learning-based context-aware resource allocation for edge-computing-empowered industrial iot, *IEEE*

- Internet Things J. 7 (5) (2020) 4260–4277, <https://doi.org/10.1109/JIOT.2019.2963371>.
- [22] J. Feng, Z. Liu, C. Wu, Y. Ji, Mobile edge computing for the internet of vehicles: Offloading framework and job scheduling, *IEEE Veh. Technol. Mag.* 14 (1) (2019) 28–36, <https://doi.org/10.1109/MVT.2018.2879647>.
- [23] L. Ni, J. Zhang, C. Jiang, C. Yan, K. Yu, Resource allocation strategy in fog computing based on priced timed petri nets, *IEEE Internet Things J.* 4 (5) (2017) 1216–1228, <https://doi.org/10.1109/JIOT.2017.2709814>.
- [24] Y. Chen, S. He, F. Hou, Z. Shi, J. Chen, An efficient incentive mechanism for device-to-device multicast communication in cellular networks, *IEEE Trans. Wireless Commun.* 17 (12) (2018) 7922–7935, <https://doi.org/10.1109/TWC.2018.2872981>.
- [25] Z. Zhou, H. Liao, X. Zhao, B. Ai, M. Guizani, Reliable task offloading for vehicular fog computing under information asymmetry and information uncertainty, *IEEE Trans. Veh. Technol.* 68 (9) (2019) 8322–8335, <https://doi.org/10.1109/TVT.2019.2926732>.
- [26] H. Liao, Y. Mu, Z. Zhou, M. Sun, Z. Wang, C. Pan, Blockchain and learning-based secure and intelligent task offloading for vehicular fog computing, *IEEE Trans. Intell. Transp. Syst.* (2020) 1–13, <https://doi.org/10.1109/TITS.2020.3007770>.