Parameter analysis and optimization of polling-based medium access control protocol for multi-sensor communication

Jialin Guo1, Fufang Li2, Tian Wang3,4, Shaobo Zhang5 and Yuqing Zhao1

Abstract
In this article, a comprehensive parameter analysis for the polling-based medium access control protocol is executed. The theoretical expressions of the relationship between network parameters and performance including delay and energy consumption are given for the first time. The specific conclusions in this article are as follows: (1) awake duration is the parameter that has the greatest impact on delay and energy consumption. Increasing the duty cycle (awake duration) will effectively reduce the delay, but will also increase the energy consumption within a certain range; (2) increasing polling duration can reduce the delay, but it will also increase the energy consumption; and (3) more forwarding nodes cause a smaller delay, and it can save the energy with modest increase of delay by reducing the polling duration. An adaptive parameter optimization polling-based medium access control protocol is proposed to optimize network performance. In this protocol, the residual energy gets fully used to increase awake duration and polling duration, which makes the delay smaller, and the network maintains a long lifetime meanwhile. Based on the results of the analysis, the adaptive parameter optimization polling-based medium access control protocol proposed in this article reduces the delay by 22.40% and increases the energy efficiency by 23.25%.

Keywords
Multi-sensor communication system, cooperative technology, energy-efficient, delay, lifetime

Date received: 24 February 2021; accepted: 3 March 2021
Handling Editor: Yanjiao Chen

Introduction
A report from Cisco shows that the number of devices connected to the Internet of things (IoT) is unprecedentedly large.1–3 The number of devices connected to the network is expected to reach 24 billion by 2020.1,3,4 One of the huge driving forces behind the development of IoT is the expansion of various sensor devices.5–7 With the development of microprocessor technology, the size of current sensor devices is getting smaller,8,9 and its functions are becoming much more powerful. Sensor devices can be deployed in various applications,9–11 such as monitoring scenarios including

---

1School of Computer Science and Engineering, Central South University, Changsha, China
2School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, China
3College of Computer Science and Technology, Huaqiao University, Xiamen, China
4Institute of Artificial Intelligence and Future Networks, Beijing Normal University – Hong Kong Baptist University United International College (UIC), Zhuhai, China
5School of Computer Science and Technology, Hunan University of Science and Technology, Xiangtan, China

Corresponding author:
Fufang Li, School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China.
Email: liff@gzhu.edu.cn

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
industrial monitoring,12–14 traffic information collection and temporary testing,15,16 and monitoring of wild animals and plants.17,18 Generally, these sensor nodes are huge in number and self-organized into a network,19–21 forming a multi-sensor system. Some are embedded in mobile phones, mobile vehicles, and other carriers.15,22,23 In addition to completing the sensing tasks, such sensing devices can also communicate with neighboring sensing devices to form a multi-sensor system in a collaborative way. The current application of the multi-sensor system has gradually expanded from initial military investigation to many civilian areas, including environmental detection, traffic monitoring, and emergency rescue.24,25

The multi-sensor system collaborates to accomplish diverse tasks, the prerequisite of which is the effective communication between sensors. The medium access control (MAC) protocol is the basis of the communication protocol,4,7,14,26 which dominates important performances in the network, including energy consumption, data transmission delay, throughput rate, and so on.4,7 These performances have an important impact on the multi-sensor base system. Since sensor nodes are mostly powered by batteries, the energy consumption directly determines their lifetime.11,13,19,27 According to the research, the energy consumed by sensor nodes for communication accounts for 70% of the total energy. Therefore, an efficient communication protocol plays a key role for multi-sensor–based systems. And it is the most critical issue for the multi-sensor system that how to design an effective communication protocol to save energy.4,7,14,26 The energy consumption of sensor nodes in awake state is more than 1000 times that in sleep state.4,7 In most applications, the frequency of sensing data is relatively slow, often in seconds, minutes, or even hours. However, the duty cycle of sensor nodes is much less than 1 s. Therefore, it does not affect the application and can save a lot of energy using the duty cycle as working mode.7,28 In such a working mode, nodes are periodically in the awake/sleep rotation. Obviously, shorter awake duration causes more energy saved. But this may increase the transmission delay and reduce the network throughput rate.

Some MAC protocols using duty cycle working mode have been proposed.7,28 There is one called polling-based MAC protocol used widely.29,30 The main principle of this protocol is that sensor nodes use duty cycle working mode.29,30 When the sender has data to send, it will change to awake state and transmit preamble signal. The transmitting time is called polling duration. If there is a node in the awake state, it responds to the sender with an early ACK as the receiver after reaching the preamble signal. Then, the sender and receiver can build a connection for data transmission. If there is no receiver waking up when the sender sends the preamble signal, the receiver will not receive the ACK message. The sender will pause for a short period called polling pause and then sends the preamble signal again. The process above continues until the data are successfully sent or overtime. Then, the sender transfers to the normal awake/sleep rotation cycle.29,30

From the above analysis, there are many parameters that affect the performance of polling-based MAC protocol, mainly including awake duration, polling duration, and polling pause. The qualitative impact of these parameters on communication performance is analyzed as follows: the larger awake duration, which means that the sensor nodes are in the awake state longer. Therefore, when the sender has data to send, the probability its receiver awake is higher. It is helpful to reduce the delay.29,30 However, the nodes are in awake state for a long time, so their energy consumption is large, which lead to a short lifetime. The polling duration indicates the duration of the preamble signal sent by the sender when looking for the receiver. Obviously, the receiver can receive the preamble signal once it wakes up if the polling duration lasts for a long time, which helps to establish a communication connection in time and reduce the delay. The same as awake duration, however, the long polling duration makes the energy consumption larger. In addition, the receiver needs to return the ACK after the polling duration time. An excessive polling duration will make the receiver waiting too long, which will increase the delay. Polling pause represents the interval between two preamble signals.29,30 Obviously, there is a positive correlation between polling pause and delay. The receiver that wakes up during the polling pause cannot establish a communication connection because the sender is in a silent state, which makes delay increasing. However, a large polling pause will reduce the energy consumption of nodes. Not only is the performance of the polling-based MAC protocol related to its own parameter values, but also the number of neighbor nodes in the network. In the network, sensor nodes usually send data to the sink by multi-hop routing. Then, the nodes within the transmitting radius and closer to the sink are the forwarding nodes of the sender, the set of which is called forwarding nodes set (FNS).7 In this way, the sender only needs to select one awake node from its FNS to send the data. Obviously, if there are more elements in the FNS, it is easier to find the node in the awake state to communicate, so the transmission delay is smaller.7 However, excessive forwarding nodes result in more serious communication interference. When multiple nodes have data to send, due to the communication interference, the delay of the nodes that do not pre-empt the channel will increase.7 From the above analysis, there are many parameters affecting the performance of polling-based MAC protocol which interact with each other, so the multiple parameters’
comprehensive optimization is very challenging. These major challenge issues are as follows:

1. Although we have analyzed the impact of these parameters on the MAC protocol above, it is only a qualitative analysis giving the trend of performance, which cannot optimize the MAC protocol accurately. Due to the various parameters of the MAC protocol and the interaction between them, the optimization work is very difficult. To the best of our knowledge, no research has been able to reveal the relationship between parameters and the network performance such as delay and energy consumption theoretically. Therefore, it is very challenging to analyze the theoretical relationship between these parameters and performance.

2. There are some studies on the communication between a pair of nodes, but lacking in theoretical analysis of the communication between multi-sensors. Since multiple sensors interfere with each other in the communication and various parameters affect each other. As a result, the optimization of multi-sensors communication faces a larger challenge.

To conquer these challenge issues, we first theoretically analyze the relationship between MAC protocol parameters and the performance. Then, we propose a better protocol called adaptive parameter optimization polling-based medium access control (APOPM) protocol. The main contributions of this article are as follows:

1. In this article, a comprehensive parameter analysis for polling-based MAC protocol is executed. The relationship expressions between network parameters and the performance are given theoretically for the first time. The theoretical analysis of this article helps guide the setting of parameters to optimize the network performance, such as reducing the energy consumption, reducing the delay, and increasing the network lifetime.

2. Based on the comprehensive parameters analysis for polling-based MAC protocol, an APOPM protocol is proposed to optimize network performance. APOPM protocol first uses the theoretical analysis results in this article to select the parameters to optimize the network lifetime for the near sink area. Then, the APOPM protocol makes full use of the residual energy of the outer nodes to increase its awake duration and polling duration to reduce the delay. Overall, the APOPM protocol reduces the delay while maintaining long lifetime. Based on the analysis results, the APOPM protocol proposed in this article reduces the delay by 22.40% and increases energy efficiency by 23.25%.

The rest of this article is organized as follows. Section “Related work” reviews related works. System models and problem statement are illustrated in section “System model and problem statements.” Section “The parameter analysis of polling-based MAC protocol” theoretically gives the results of the parameter analysis of polling-based MAC protocol. In section “APOPM protocol,” the APOPM protocol is proposed. The conclusion of this article is given in section “Conclusion.”

**Related work**

With the development of microprocessor technology, the volume of sensor devices has become smaller while their functions are more powerful.\(^3\)\(^1\),\(^3\)\(^2\) The computing ability of current mobile phones has surpassed that of personal computers 10 years ago. Massive devices connected to the IoT have caused huge changes in the current network architecture.\(^3\)\(^1\) Due to abundant devices connected to the edge network, the total amount of computing and storage is very huge, which makes the focus of the network shift from the cloud to the edge of the network, forming the edge network and edge computing model gradually.\(^3\)\(^3\)–\(^3\)\(^5\) With the development of artificial intelligence, it is possible to dig up rich knowledge from the big data perceived by the collaborative work of sensor devices, thereby bringing human life great changes.\(^3\)\(^6\)

However, the basis of collaborative work of multi-sensor systems is the effective communication mechanism between the nodes. Through the communication between multiple nodes the data can be transmitted successfully, which is the most important task for multi-sensor systems. Therefore, designing an effective communication mechanism is one of the significant contents of sensor-based systems.

There has been some research on the design of communication mechanism or the improvement on the basic MAC protocol for different application scenarios. For instance, Zhang et al.\(^3\)\(^7\) proposed a Q-learning-based MAC switching scheme for the fly ad hoc networks (FANETs). Applying this scheme, each node in the type of network can switch the MAC protocol between carrier-sense multiple access with collision avoidance (CSMA/CA) and time-division multiple access (TDMA) according to the current condition of network to improve the communication performance.\(^3\)\(^7\) Moreover, for the vehicular ad hoc networks (VANETs), Kumar and Kim\(^3\)\(^8\) proposed a new MAC protocol (bitmap-based hybrid medium access control (BH-MAC)) based on TDMA. BH-MAC uses bitmaps
with small size to represent the slot status, and the CSMA mechanism for the slot reservations is applied. Therefore, the data amount in the transmission can be reduced, and the transmission delay and energy consumption are both lower compared to the existing MAC protocol. In addition, also for the VANETs, a traffic-aware time-division multiple access–based medium access control (TA-MAC) protocol is proposed by Deng et al. They considered the varying data traffic due to the high mobility of vehicles, and the basic slot allocation approach of TDMA cannot fully use the channel resource, and even leads to high data collision probability. Therefore, Deng et al. established a traffic-aware mechanism for the real-time data traffic acquisition, and the TA-MAC dynamically changes the slot assignment based on the varying data traffic. The channel efficiency and low transmission delay are both guaranteed.

The protocols introduced above are mostly designed for the characteristic of different scenarios, and the theoretical analysis and universal optimization considering the essential features of MAC protocol is lacking. Currently, the most working mode of MAC protocol in wireless sensor network (WSN) is the periodic awake/sleep rotation duty cycle, because the energy of sensor nodes is generally limited by batteries and they need to save by periodic sleeping. And there is much research on the duty cycle working mode, which mainly studies the impact of proportion of awake duration in one cycle on the network performance. Generally, if the awake time of the node is longer, the energy consumption is larger, so the lifetime is shorter. However, if the sender has data to send, its FNS can easily get the receiver in the awake state, which leads to a small delay. Otherwise, the energy consumption is low and the lifetime is long, while its delay is large. As can be seen from the above, the awake time affects the life and delay of nodes. Therefore, some studies try to optimize these two performances. Byun et al. proposed a strategy of duty cycle adjustment based on feedback control, the basic idea of which is to reduce the awake time of nodes as much as possible to extend their lifetime under the premise that the data transmission delay meets the application requirements. The thought of this strategy is: if the time of the data flow reaching the sink is larger than the demand of the application, which means the awake time adopted by the current nodes is too small, so the sink node sends control information along the data propagation path. The nodes receive the information and increase their awake time to reduce the delay. If the delay is too small, the same method is used to notify the data source nodes to reduce the awake time to prolong the lifetime. Even in the case of the same duty cycle, the performances differ when the sender chooses distinct forwarding nodes. Liu et al. found that the first forwarding node waking up is not necessarily superior if the sender has data to send, for the sender has multiple forwarding nodes whose distances from the sink are different. The forwarding node that first wakes up may be far from the sink. If the sender chooses it as the relay node, more hops are required to reach the sink. Each hop will introduce a delay, so the more hops the greater the total delay. If the sender waits for the node closest to the sink waking up and selects it as the relay node, the data can forward a long distance in one hop. In this way, although the delay at this hop is large, it can reduce the number of hops to the sink, which perhaps causes a smaller overall delay. Accordingly, Liu et al. proposed an optimized forwarding node selection strategy.

In the specific organization of MAC time slots, the classic protocols are as follows. Figure 1 shows the principle of the Berkeley medium access control (B-MAC) protocol, which is one of the classic MAC protocols. In B-MAC protocol, the nodes periodically enter the awake and sleeping state. When the node has data to send, it will wake up and issue a long preamble, waiting until a receiver wakes up. The receiver in B-MAC protocol only listens to the channel awake. If the long preamble from the sender is received by the receiver, the connection between them is established, and then the data can be transmitted.

The disadvantage of B-MAC protocol is that the sender must deliver the long preamble constantly until the receiver wakes up, causing much more energy consumption. Aiming at this shortcoming, another protocol named the X-MAC protocol is proposed. In the X-MAC protocol, the sender issues multiple short preambles instead of one long preamble when has data to send. After sending each short preamble, the sender pauses for a period and then sends the next one. In this way, the sender can reduce the time of sending preambles, which saves the energy. However, the delay will increase compared to the B-MAC protocol, because in one case when the sender sends a short preamble but the receiver is in the sleeping state, it cannot receive the short preamble. The sender needs to send more preambles until the receiver wakes up and obtains one of the

![Figure 1. The B-MAC protocol.](image)
preambles, and then the connection can be established. The polling-based MAC in this article is the X-MAC protocol. The polling duration is equivalent to the short preamble in X-MAC protocol, and the polling pause corresponds to the listening time of the sender is shown in Figure 2. The X-MAC protocol is widely used in sensor systems. Therefore, analyzing and optimizing parameters on the theoretical level make great sense in the performance improvement of network.

The protocols discussed above have one thing in common that the sender is initiative to establish a data connection. This type of protocol is called sender-initiated (SI) duty-cycled MAC protocol. In fact, there is another type called receiver initialization (RI) MAC protocol. In the RI protocol, the node with data does not send a data transmission request. The sender enters listening state instead. Once the sender obtains a beacon sent from the receiver, which is equivalent to the preamble in the SI protocol, the data transmission connection will be established.

In addition to the significant impact of the awake duration, the polling duration and the polling pause in the polling-based MAC protocol also affect the network performance. Besides the internal parameters, there are other factors impacting on the performance of network, the most important of which is the delay caused by communication conflicts and interference. Since the nodes in wireless networks share the channel, only one node can be allowed to send the data at the same time. Conflicts will occur if multiplesenders initiate data transmission simultaneously, and only one sender will get the channel and successfully send it. Obviously, in a network with a high density of nodes, excessive awake time will increase the probability of collision between the nodes. Communication collision makes all the energy and time consumed in building connections wasted, which leads to a low system revenue. Therefore, simply increasing the awake time of nodes will not reduce the delay effectively in the network with abundant nodes. Another factor affecting the network performance is the load of nodes. Apparently, the node with heavy load will be allocated more awake time, which also probably causes communication collision. From above, the factors that affect network communication performance are diverse and complex, so the relative research is very challenging.

There are also some studies to dynamically adjust the parameters of polling-based MAC protocol to optimize network performance based on the load of nodes. For instance, the variable traffic-adaptive duty cycled sensor medium access control (VTA-SMAC) proposed by Masood Ur et al. effectively improves the network performance by adapting duty cycle based on the varying data traffic or node load. Al-Janabi and Al-Raweshidy proposed a hybrid MAC protocol incorporating TDMA and CSMA/CA, which can adapt the sleep/wake-up period according to the variance of network load. Moreover, Siddiqui et al. proposed the adaptive and dynamic polling-medium access control (ADP-MAC) protocol. The thought of ADP-MAC protocol is to dynamically adjust the polling distributions and polling intervals in the MAC protocol according to the data volume of nodes. When the load is light, the polling interval is increased to save energy. Otherwise, reduce the polling interval to increase the processing power of nodes. The polling distributions are also adjusted according to the arrival distribution of the data, which makes the two parameters adapted to improve network performance, although it is difficult in practice.

System model and problem statements

Network model

We consider a sensor network with several nodes, the radius of which is \( R \) and the node density is \( \rho \). These nodes no longer change their positions after being deployed. Each node sends the data packets received from the outer nodes together with self-generated data packets to the next layer until they reach the sink nodes. The probability of data generating is \( a \), and the radius of data transmission range is \( r \).

Energy consumption model

The energy of nodes in the network is consumed in receiving packets from outer nodes, sending data to the next layer and sleeping, which respectively expressed as \( E_r \), \( E_s \), and \( E_{\text{sleep}} \). Nodes at the fringe of network only send self-generated data packets while sink nodes receive and process data in the whole network. Thus, the energy consumption is as follows

\[
E = E_r + E_s + E_{\text{sleep}}
\]
Problem statement
In multi-sensor systems, sensors cooperate to complete complex calculating tasks, which depend on the communication between the nodes. According to related research, the energy consumed in communication accounts for more than 70% of its total energy consumption for sensor nodes, and the MAC protocol is a key affecting the performance of nodes. The polling-based MAC protocol is one of them widely used and researched. In polling-based MAC protocol, there are many parameters such as awake duration, polling duration, and polling pause duration, impacting on the performance of network including the delay, energy consumption, throughput rate, channel conflict, and so on. In addition, the network environment, for instance, the number of nodes in the communication range called forwarding nodes will also affect the network performance. There are some relative studies; however, due to the interaction between parameters, the research is very complicated, which leads to the previous research mainly on the analysis of communication between two single nodes. There is no research on MAC parameter optimization of multi-sensor system, also the lack of optimized polling-based MAC protocol.

The research objectives of this article are mainly as follows: (1) theoretically analyze the impact of different MAC parameters such as awake duration, polling duration, and polling pause on the performance of communication protocols consisting of energy consumption and delay in multi-sensor systems, providing the guidance for MAC protocol parameter optimization and (2) propose an optimized polling-based MAC protocol, which can effectively use the energy and significantly improve the performance of the entire network. The performance indexes to be optimized are as follows:

Reduce the transmission delay. Transmission delay refers to the time it takes for the outer nodes to propagate data packets to the inner ones until the sink nodes successfully receive the data, which is expressed as $D$. The smaller the transmission delay, the better the timeliness of the network. Considering that there are altogether $k$ hops from the edge of network to sink nodes, the time it takes for the data packet to transmit at the $i$th hop is $D_i$. The minimum transmission delay can be expressed as

$$\min(D) = \min\left(\sum_{i=1}^{k} D_i\right)$$

(2)

Extend the lifetime of network. Network lifetime refers to the duration from it starting to work to the first node dead, expressed as $L$. Since the sink nodes exhaust energy first, maximizing the lifetime of the network is equivalent to maximizing that of the sink nodes. Assuming the initial energy of the sink nodes $E_{\text{init}}$ and the average power at work $P_{\text{sink}}$, the maximum network lifetime can be expressed as

$$\max(L) = \max\left(\frac{E_{\text{init}}}{P_{\text{sink}}}\right)$$

(3)

Increase the energy efficiency. There is residual energy in the outer nodes when the network dies. In the local optimization work, we adjust the parameters of the outer nodes to use the remaining energy better and maximize the energy efficiency of the network. Assume that there are $n$ nodes in the network and the initial energy of each node is $E_i$. When the network dies, the energy consumed of the $i$th node is $E_i$. The maximum energy efficiency $\theta$ can be expressed as

$$\max(\theta) = \max\left(\frac{\sum_{i=1}^{n} E_i}{n \times E_{\text{init}}}\right)$$

(4)

The parameter analysis of polling-based MAC protocol

The polling-based MAC protocol
Polling-based MAC protocol is a common communication protocol used for nodes to transmit data in WSN, which is shown in Figure 3.

As is shown in Figure 3, in this protocol, sender and receiver both have two states awake and sleeping. In addition, there are two states of polling and polling-off appearing alternately in the awake duration of receiver. Receiver can only receive packets in polling state. Before the formal transmission, the sender sends preamble packets constantly to ensure that the receiver is ready to receive the data packets. Receiver will successfully receive the preamble and respond an early acknowledge when polling, indicating that it is ready to receive data. And then, the data transmission starts. However, if the receiver is in polling-off state, the preamble cannot be received. Therefore, the sender can only continue to transmit them during its awake duration until receives the early acknowledge from the other side, or enter the next sleeping period. The symbols and their meanings used in the following network analysis are listed in Table 1.

Transmission delay analysis
This section discusses how MAC parameters including awake duration and polling duration affect the transmission delay in networks. We will start with the single-
to-single model, where there is only one sender and one receiver in the network, and then we promote to discuss the other three models.

**Single-to-single model.** In this section, a single-to-single model for calculating delay is established.

**Theorem 1.** The transmission delay of single-to-single model $D_{sts}$ is calculated as equation (5), the parameters in which are listed in Table 1

$$D_{sts} = \frac{T_a}{T_a + T_o} \cdot d_a + \frac{T_o}{T_a + T_o} \cdot d_o + \frac{sd}{\gamma}$$  \hspace{1cm} (5)

where

$$d_a = \frac{T_{po}}{T_{po} + T_{pp}} \cdot d_{po} + \frac{T_{pp}}{T_{po} + T_{pp}} \cdot d_{pp}.$$  
$$d_o = \frac{T_o}{T_o} \cdot d_{of} + \frac{T_a}{T_a} \cdot d_{ol}.$$  
$$d_{po} = \frac{1}{\mu} \cdot \left(1 + \frac{T_{pp}}{T_{po} + T_{pp}} \right) \cdot \tau_p + \frac{s_p + s_e}{\gamma},$$  
$$d_{pp} = d_{po} + \frac{T_{pp}}{2},$$  
$$d_{of} = T_a + T_o + \frac{T_o}{T_a + T_o} \cdot d_a + \frac{T_o}{T_o + T_a} \cdot d_{ol},$$  
$$d_{ol} = d_{po} + \frac{T_o}{2},$$  
$$\gamma = W \cdot \log_2 \left(1 + \frac{P_s}{N} \right).$$

**Proof.** There is only one sender and one receiver communicating in the single-to-single model. When the sender has data packets to send, it constantly propagates preambles to the receiver until the receiver receives one and returns the early acknowledge. The receiver can only attain the preamble during polling. However, the sender probably starts sending preambles at the time receiver polling-off or sleeping, which causes extra transmission delay. Two situations of transmitting preambles are discussed: (1) sender starts sending preambles during the awake period of the receiver and (2) sender starts sending preambles when the receiver is in sleep. The probabilities of these two situations are $T_a/T_a + T_o$ and $T_o/T_a + T_o$, and the transmission delay of two situations is $d_a$ and $d_o$, respectively.

For the first situation, it can be divided into the following two cases. First, when the receiver is in the polling state, the preamble sent by the sender can be received immediately, probability of which is $T_{po}/T_{po} + T_{pp}$. Considering the transmitting success rate $\mu$, the transmission delay in this case is $d_{po}$. When the first preamble is successfully transmitted, there is no extra delay and only the transmitting time of the preamble and the returning time of early acknowledge. $s_p + s_e/\gamma$, where $s_e$ is the data size of early acknowledge, and $\gamma$ is the transmitting rate in this channel, which is related to the bandwidth $W$ and the signal-to-noise ratio (SNR) $P_s/N$. However, if the first preamble fails, the sender will continue sending

![Figure 3. The sequence diagram of the data transmission in the polling-based MAC protocol.](image)

**Table 1.** Parameter description.

| Parameter | Meaning |
|-----------|---------|
| $T_a$     | The duration when the sender/receiver is awake |
| $T_o$     | The duration when the sender/receiver is off |
| $T_{cycle}$ | The duration of each cycle (the sum of $T_a$ and $T_o$) |
| $T_{po}$  | The duration of one polling |
| $T_{pp}$  | The duration of the pause between two polling |
| $T_{ea}$  | The duration of sending/receiving the early acknowledgment |
| $T_{ack}$ | The duration of sending/receiving the acknowledgment |
| $T_{ad}$  | The additional time of the awake duration |
| $T_{cs}$  | The duration of carrier sense |
| $T_{bo}$  | The duration of back off |
| $T_{pp}$  | The duration of one preamble |
| $T_{po}$  | The duration of the pause between two preambles |
| $T_d$     | The duration of sending/receiving data |
preambles after the duration \( \tau_p \), and the probability of this situation is \( 1 - \mu \). The expected transmitting times of preambles are \( 1/\mu \), but the sender may re-transmit preambles in the polling pause of receiver. In this case, the receiver never receives the preambles, and the probability is \( T_{pp}/T_{pp} + T_{po} \). Therefore, considering the case that the sender sends the preambles at polling pause, the total transmitting times are \( 1/\mu \cdot (1 + T_{pp}/T_{pp} + T_{po}) \). Therefore, the extra delay is \( 1/\mu \cdot (1 + T_{pp}/T_{pp} + T_{po}) \cdot \tau_p \), where \( \tau_p \) is the interval between two preambles. Second, the other case is that the sender starts to establish communication when the receiver is in the polling pause, the probability of which is \( T_{pp}/T_{po} + T_{pp} \). In this case, preambles cannot immediately be received, so the sender needs to wait till the receiver go into the next polling period, and the expected value of waiting time is \( T_{pp}/2 \). Therefore, compared to the situation that the sender starts sending preambles when the receiver is polling, the extra delay of this situation is \( T_{pp}/2 \). Above all, when the sender sends preambles in the awake duration of the receiver, the expected delay is

\[
d_a = \frac{T_{po}}{T_{po} + T_{pp}} \cdot d_{po} + \frac{T_{pp}}{T_{po} + T_{pp}} \cdot d_{pp}
\]

The second situation is that the sender establishes communication when the receiver is sleeping, which also includes two cases. Assuming that the sleeping duration is always larger than the awake duration, we cut the sleeping duration into \( T_a - T_o \) and \( T_a \). When the sender begins transmission in the later section \( T_a \), it can wait until the receiver enters polling state in this cycle. The calculating method of the expected transmission delay is similar to \( d_{pp} \). In this case, the delay can be expressed as \( d_{ol} = d_{po} + (T_a/2) \).

However, the sender never waits for the receiver to wake up in this cycle when sending preambles in the former segment \( T_o - T_a \), and it needs to enter sleeping until the next cycle to continue to send packets, which causes a cycle of delay \( T_o + T_o \). When the next awake period arrives, the reception of the preamble is transformed into the first situation, which calculation method is the same as above. To simplify the problem, we default that the tolerance of communication establishing is two cycles. The delay in the second cycle plus the duration of the first cycle is the total one in this case

\[
d_{of} = T_o + T_o \cdot \frac{T_o}{T_o + T_o} \cdot d_a + \frac{T_o}{T_o + T_o} \cdot d_{ol}
\]

The probabilities of the two cases above are \( T_o/T_o \) and \( T_o - T_a/T_o \), respectively. Therefore, the transmission delay in the second situation is expressed as

\[
d_o = \frac{T_o - T_a}{T_o} \cdot d_{of} + \frac{T_a}{T_o} \cdot d_{ol}
\]

In summary, the expected transmission delay in the single-to-single model is

\[
D_{sts} = \frac{T_a}{T_a + T_o} \cdot d_o + \frac{T_o}{T_a + T_o} \cdot d_o + \frac{s_d}{\gamma}
\]

Figure 4 illustrates that the transmission delay changes with awake duration and polling duration. First, it can be seen that the delay reduces slightly with the increase in polling duration when other parameters are fixed. Second, the awake duration of nodes has great impact on the transmission delay that larger \( T_a \) brings smaller delay, shown as the black, red, and blue lines in Figure 4. Third, from the black line and green line, the delay increases with the decrease in transmitting success rate \( \mu \), which is consistent with the theoretical analysis above. Moreover, the transmitting rate \( \gamma \) has similar impact. When the bandwidth or the SRN is unsatisfactory, the transmitting rate is smaller, and the delay of transmitting preambles and data packets is larger. In summary, to reduce transmission delay, the awake duration and polling duration of nodes should be increased.

Figure 5 illustrates the impact of sleep duration and polling pause duration on the delay. As the sleep duration increases, the probability of sender sending the preambles when the receiver is awake is lower, so the transmission delay increases. Similarly, the increase of polling pause duration has the same effect on the transmission delay.

**Single-to-m model.** In this section, the single-to-m model is established based on Theorem 1 and the relationship between the transmission delay and the number of receivers is analyzed.

**Theorem 2.** The transmission delay of single-to-m model \( D_{stm} \) is calculated as equation (6)
\[ D_{stm} = P_a \cdot d_{am} + P_o \cdot d_{om} + \frac{s_d}{\gamma} \]  

where

\[ P_o = 1 - \left( \frac{T_o}{T_a + T_o} \right)^m \]
\[ P_a = \left( \frac{T_o}{T_a + T_o} \right)^m \]
\[ d_{am} = d_a, d_{om} = \frac{T_o - T_a}{T_o} \cdot d_{ofm} + \frac{T_a}{T_o} \cdot d_{olm} \]
\[ d_{olm} = d_o, d_{ofm} = T_a + T_o + P_a \cdot d_{am} + P_o \cdot d_{olm} \]

**Proof.** In the single-to-\( m \) model, the transmission delay is defined as the duration from the sender starting sending preambles to any receiver obtaining the preamble and responding the early acknowledge to the sender. The calculation of delay in this model is similar as the single-to-single model, but the probability is different. First, the worst situation is that all receivers are sleeping when the sender begins transmission, where the sender needs to wait till one receiver enters the polling state. In single-to-\( m \) model, the probability of \( m \) receivers all sleeping is \( P_o = (T_o/T_a + T_o)^m \). Thus, the delay in this situation can be expressed as

\[ d_{om} = \frac{T_o - T_a}{T_o} \cdot d_{ofm} + \frac{T_a}{T_o} \cdot d_{olm} \]

where the probabilities in \( d_{ofm} \) should be changed.

The other situation is that as long as there is one receiver awake when the sender sends the preamble, the data transmission can start immediately, whose probability is \( P_a = 1 - (T_o/T_a + T_o)^m \). In summary, the delay in the single-to-\( m \) model is

\[ D_{stm} = P_a \cdot d_{am} + P_o \cdot d_{om} \]

**Proof.** In the single-to-\( m \) model, the transmission delay is defined as the duration from the sender starting sending preambles to any receiver obtaining the preamble and responding the early acknowledge to the sender. The calculation of delay in this model is similar as the single-to-single model, but the probability is different. First, the worst situation is that all receivers are sleeping when the sender begins transmission, where the sender needs to wait till one receiver enters the polling state. In single-to-\( m \) model, the probability of \( m \) receivers all sleeping is \( P_o = (T_o/T_a + T_o)^m \). Thus, the delay in this situation can be expressed as

\[ d_{om} = \frac{T_o - T_a}{T_o} \cdot d_{ofm} + \frac{T_a}{T_o} \cdot d_{olm} \]

where the probabilities in \( d_{ofm} \) should be changed.

The other situation is that as long as there is one receiver awake when the sender sends the preamble, the data transmission can start immediately, whose probability is \( P_a = 1 - (T_o/T_a + T_o)^m \). In summary, the delay in the single-to-\( m \) model is

\[ D_{stm} = P_a \cdot d_{am} + P_o \cdot d_{om} \]

**Proof.** In the single-to-\( m \) model, the transmission delay is defined as the duration from the sender starting sending preambles to any receiver obtaining the preamble and responding the early acknowledge to the sender. The calculation of delay in this model is similar as the single-to-single model, but the probability is different. First, the worst situation is that all receivers are sleeping when the sender begins transmission, where the sender needs to wait till one receiver enters the polling state. In single-to-\( m \) model, the probability of \( m \) receivers all sleeping is \( P_o = (T_o/T_a + T_o)^m \). Thus, the delay in this situation can be expressed as

\[ d_{om} = \frac{T_o - T_a}{T_o} \cdot d_{ofm} + \frac{T_a}{T_o} \cdot d_{olm} \]

where the probabilities in \( d_{ofm} \) should be changed.

The other situation is that as long as there is one receiver awake when the sender sends the preamble, the data transmission can start immediately, whose probability is \( P_a = 1 - (T_o/T_a + T_o)^m \). In summary, the delay in the single-to-\( m \) model is

\[ D_{stm} = P_a \cdot d_{am} + P_o \cdot d_{om} \]
Theorem 3. The transmission delay of $n$-to-single model $D_{nts}$ is calculated as equation (7)

$$D_{nts} = \frac{T_o}{T_o + T_a} \times d_{an} + \frac{T_a}{T_o + T_o} \times d_{on} + \frac{s_d}{\gamma}$$

(7)

where

$$d_{an} = \frac{T_{po}}{T_{po} + T_{pp}} \cdot d_{pon} + \frac{T_{pp}}{T_{po} + T_{pp}} \cdot d_{ppn}$$

$$d_{on} = \frac{T_o - T_a}{T_o} \cdot d_{ofn} + \frac{T_a}{T_o + T_o} \cdot d_{oln}$$

$$d_{pon} = \frac{1}{\mu} \left( \frac{\theta}{1 - \frac{T_{po}}{T_{pp} + T_{po}} \cdot \frac{T_a}{T_o + T_a}} \right)^{\alpha n - 1} + \frac{T_{pp}}{T_{pp} + T_{po}}$$

$$d_{ppn} = d_{pon} + \frac{T_{pp}}{2}$$

$$d_{ofn} = T_{a} + T_{a} + \frac{T_o}{T_o + T_o} \cdot d_{on} + \frac{T_o}{T_o + T_o} \cdot d_{oln}$$

$$d_{oln} = d_{on} + \frac{T_a}{2}$$

Proof. Similar to the single-to-single model, we can also divide the transmission into four cases. The expected delay of these four cases is $d_{pon}$, $d_{ppn}$, $d_{ofn}$, and $d_{oln}$ respectively. In addition, the data collision should be considered in the $n$-to-single model because several senders may send preambles at the same time slot. When the preambles from different senders collide, the receiver obtains neither of them and all the senders should wait for a random interval and re-transmit preambles. Therefore, data collision will cause larger delay. In the single-to-single model, we have calculated the delay $d_{po}$. Due to the channel noise, the expected transmitting times are $1/\mu \cdot (1 + T_{pp}/T_{pp} + T_{po})$. However, in the $n$-to-single model, although the preamble is sent successfully at the probability of $\mu$, there may be other data collides with it. Therefore, the transmitting times should be modified to

$$\frac{1}{\mu} \left( \frac{\theta}{1 - \frac{T_{po}}{T_{pp} + T_{po}} \cdot \frac{T_a}{T_o + T_a}} \right)^{\alpha n - 1} + \frac{T_{pp}}{T_{pp} + T_{po}}$$

when there are $n$ senders. In this formula

$$\left(1 - \frac{T_{po}}{T_{pp} + T_{po}} \cdot \frac{T_a}{T_o + T_a}\right)^{\alpha n - 1}$$

is the probability of no collision, where $\alpha$ is the probability of data generation. From the probability, there will be no collision if the other $\alpha \cdot n - 1$ senders with data do not enter the polling state and send preambles at this time slot. $\theta$ is the rate of the average back-off time $t_{bo}$ due to the collision and the interval of preambles $t_p$, that is, $\theta \cdot t_p = t_{bo}$. The calculation of expected delay in the other three cases is similar with the single-to-single model, so there is no further elaboration.

Figure 8 shows the transmission delay changes with MAC parameters in the $n$-to-single model. It can be seen that when the awake duration and polling duration increase, the delay can be reduced. However, with the increase in the number of senders, the performance of delay reduction is worse, and even the delay increases compared to that when the value of parameters is not increased. This is because longer polling duration of senders leads to larger probability of data collision. Therefore, in this delay model, the impact of MAC parameters becomes complex when considering the data collision.

Figure 9 illustrates the transmission delay changes with other parameters in equation (7). First, larger transmitting success rate $\mu$ brings smaller delay, corresponding to Figure 7. Second, when the data generating probability is larger, more senders join in the communication establishment with the receiver. Therefore, the collision probability is larger and the transmission delay increases.

N-to-m model. In this section, the $n$-to-$m$ model for transmission delay is established based on the models above.

Theorem 4. The transmission delay of $n$-to-$m$ model $D_{nm}$ is calculated as equation (8)

$$D_{nm} = P_a \times d_{an}^m + P_o \times d_{on}^m + \frac{s_d}{\gamma}$$

(8)
where

\[
\begin{align*}
    d_{on}^m &= \frac{T_a - T_o}{T_o} \cdot d_{on}^m + \frac{T_a}{T_o} \cdot d_{on}^m \\
    d_{on}^m &= \frac{T_{po}}{T_{po} + T_{pp}} \cdot d_{on}^m + \frac{T_{pp}}{T_{po} + T_{pp}} \cdot d_{ppn} \\
    d_{pon}^m &= d_{pon}^m, d_{ppn}^m = d_{ppn} \\
    d_{on}^m &= d_{on} + \frac{T_a}{2}, d_{on}^m = T_a + T_o + P_a \cdot d_{on}^m + P_o \cdot d_{on}^m
\end{align*}
\]

**Proof.** In the \(n\)-to-\(m\) model, there are \(n\) senders and \(m\) receivers. The senders with data need to communicate with any receiver and transmit data packets to it. \(D_{nm}\) is the average expected delay. In the calculation of \(D_{nm}\), we consider the characteristic of single-to-\(m\) model and \(n\)-to-single model simultaneously: (1) multiple receivers increase the probability of preamble received and (2) the preambles from multiple senders may collide. The calculation method of components in equation (8) has been illustrated in the models above, so there is no further elaboration.

Figure 10 illustrates the transmission delay changes with the number of nodes in the \(n\)-to-\(m\) model. When the number of senders \(n\) increases or the number of receivers decreases, the delay is higher. This is because the increase of senders leads to a higher collision rate, while the increase of receivers causes a higher probability of preambles received. Moreover, Figure 11 shows the impact of MAC parameters on the delay in the \(n\)-to-\(m\) model, which is consistent with Figure 9. When the awake duration and polling duration are excessively large, the collision rate is higher and the transmission delay will increase.

**Energy consumption analysis**

This section mainly discusses the impact of MAC parameters on the energy consumption in the network. The model of network energy consumption is established on the basis of the delay calculated in section “Transmission delay analysis.” Similar to the delay model, we first start with single-to-single model, and then promote it to discuss the single-to-\(m\) model, \(n\)-to-single model, and \(n\)-to-\(m\) model.

**Single-to-single model.** In this section, the single-to-single model for transmission delay is established based on section “Transmission delay analysis.”

**Theorem 5.** The energy consumption in the single-to-single model \(E_{sts}\) can be calculated as

\[
E_{sts} = E_{sts}^d + E_{sts}^r + \frac{S_d}{r} (P_s + P_r) \tag{9}
\]
where

\[
E_{sts}^s = \frac{T_a}{T_a + T_o} \cdot e_p^a + \frac{T_o}{T_a + T_o} \cdot e_o^a
\]
\[
E_{sts}^r = \frac{T_a}{T_a + T_o} \cdot e_p^r + \frac{T_o}{T_a + T_o} \cdot e_o^r
\]
\[
e_p^a = \frac{T_{pp}}{T_{pp} + T_{po}} \cdot e_p^a + \frac{T_{pp}}{T_{po} + T_{pp}} \cdot e_p^p
\]
\[
e_p^r = \frac{1}{\mu} \left( 1 + \frac{T_{pp}}{T_{pp} + T_{po}} \right) \cdot \tau_p \cdot \frac{P_{po}}{P_{pr}} + \frac{s_e}{r} \cdot P_r
\]
\[
e_p^p = \frac{1}{\mu} \left( 1 + \frac{T_{pp}}{T_{pp} + T_{po}} \right) \cdot \tau_p \cdot \frac{P_{po}}{P_{pr}} + \frac{s_e}{r} \cdot P_s
\]
\[
e_p^a + \frac{T_{pp}}{T_{pp} + T_{po}} \cdot P_{sleep}
\]
\[
e_o^a = \frac{T_a}{T_o + T_a} \cdot e_o^a + \frac{T_o}{T_o + T_a} \cdot e_o^r
\]
\[
e_o^r = \frac{T_a}{T_o + T_a} \cdot e_o^a + \frac{T_o}{T_o + T_a} \cdot e_o^r
\]
\[
e_{of} = T_a \cdot \left( \frac{T_{pp}}{T_{pp} + T_{po}} \cdot \frac{P_{po}}{P_{pr}} + \frac{T_{pp}}{T_{po} + T_{pp}} \cdot P_{pp} \right)
\]
\[
+ T_o \cdot P_{sleep} + \frac{T_a}{T_o + T_a} \cdot e_o^a + \frac{T_o}{T_o + T_a} \cdot e_o^r
\]
\[
e_{of} = T_a \cdot \left( \frac{T_{pp}}{T_{pp} + T_{po}} \cdot \frac{P_{po}}{P_{pr}} + \frac{T_{pp}}{T_{po} + T_{pp}} \cdot P_{pp} \right)
\]
\[
+ T_o \cdot P_{sleep} + \frac{T_a}{T_o + T_a} \cdot e_o^a + \frac{T_o}{T_o + T_a} \cdot e_o^r
\]

\[
E_{of} = \frac{T_{pp}}{T_{pp} + T_{po}} \cdot \frac{P_{po}}{P_{pr}} + \frac{T_{pp}}{T_{po} + T_{pp}} \cdot P_{pp}
\]
\[
+ T_o \cdot P_{sleep} + \frac{T_a}{T_o + T_a} \cdot e_o^a + \frac{T_o}{T_o + T_a} \cdot e_o^r
\]

The energy consumption for polling is as 
\[
e_{o} = T_a \cdot \left( \frac{T_{pp}}{T_{pp} + T_{po}} \cdot \frac{P_{po}}{P_{pr}} + \frac{T_{pp}}{T_{po} + T_{pp}} \cdot P_{pp} \right)
\]
\[
+ T_o \cdot P_{sleep} + \frac{T_a}{T_o + T_a} \cdot e_o^a + \frac{T_o}{T_o + T_a} \cdot e_o^r
\]

Proof. In the single-to-single model, the energy consumption of communication establishment consists of two parts: the energy consumed by the sender and the receiver, expressed as \(E_{sts}^s\) and \(E_{sts}^r\), respectively. In addition, \(s_d/r(P_s + P_r)\) is the energy consumption for transmitting data.

Similar to the delay calculation, we also divide the communication establishment into four cases. The first case is the sender starting sending preambles when the receiver is polling. The energy consumption of this case for the sender and receiver is \(e_p^a\) and \(e_p^r\), respectively. \(e_{po}\) consists of the energy for sending preambles \(1/\mu \cdot (1 + T_{pp}/T_{pp} + T_{po}) \cdot \tau_p \cdot P_{po}\) and receiving early acknowledge \(s_e/r \cdot P_r\), where \(P_{po}\) is the average power of sending preambles with a certain interval. Similarly, \(e_{po}\) consists of the energy for polling before receiving the preamble \(1/\mu \cdot (1 + T_{pp}/T_{pp} + T_{po}) \cdot \tau_p \cdot P_{po}\) and the energy for sending back early acknowledge \(s_e/r \cdot P_r\), where \(P_{po}\) is defined as the average power of polling state and polling pause state. The second case is that the sender starts sending the preamble when the receiver is in polling pause. According to the delay calculation, we know that the extra expected delay is \(T_{pp}/2\). In this duration, the sender keeps sending preambles, so its power is \(P_{po}\), while the power of receiver is \(P_{pp}\).

The two cases above constitute the situation of the sender sending preambles in the awake duration of receiver; the energy consumption for both is \(e_p^a\) and \(e_p^r\), respectively.

Moreover, for the situation of the sender sending preambles when the receiver is sleeping, the sleeping state of receiver is also divided into two sections \(T_o - T_a\) and \(T_a\). When the sender starts sending preambles in the latter section, the sender can wait for the receiver waking up in the current cycle. From section “Transmission delay analysis,” the extra delay of this case is \(T_a/2\). In the extra delay, the sender continues sending preambles, while the receiver is sleeping. The energy is calculated as \(e_{ol}^a\) and \(e_{ol}^r\). On the contrary, when the sender starts sending preambles in the former section, the sender cannot wait for the receiver waking up in the current cycle. The extra delay of this situation is \(T_{cycle}\), and in the next cycle, the sender still sends the preambles like the three cases above, so the energy consumption is

\[
e_{of} = T_a \cdot \left( \frac{T_{pp}}{T_{pp} + T_{po}} \cdot \frac{P_{po}}{P_{pr}} + \frac{T_{pp}}{T_{po} + T_{pp}} \cdot P_{pp} \right)
\]
\[
+ T_o \cdot P_{sleep} + \frac{T_a}{T_o + T_a} \cdot e_p^a + \frac{T_o}{T_o + T_a} \cdot e_p^r
\]

In this formula

\[
T_a \cdot \left( \frac{T_{pp}}{T_{pp} + T_{po}} \cdot \frac{P_{po}}{P_{pr}} + \frac{T_{pp}}{T_{po} + T_{pp}} \cdot P_{pp} \right)
\]
\[
+ T_o \cdot P_{sleep} + \frac{T_a}{T_o + T_a} \cdot e_p^a + \frac{T_o}{T_o + T_a} \cdot e_p^r
\]

is the energy consumption in the first cycle, and \(T_a/(T_o + T_a) \cdot e_{ol}^a + (T_o/(T_o + T_a)) \cdot e_{ol}^r\) is the energy consumption of the next cycle. Similarly, the energy consumption of the receiver is as \(e_{of}^r\).

In summary, the energy consumption of four cases is illustrated above. Combined with the corresponding probability, the total energy consumption can be obtained as equation (9).

Figure 12 shows that under different polling duration, the energy consumption of nodes increases monotonically with the awake duration. Moreover, the transmission success rate \(\mu\) and the data rate \(\gamma\) also impact the energy consumption that larger \(\mu\) or \(\gamma\) brings smaller energy consumption.

Single-to-m model. In this section, we extend from the single-to-single model to discuss condition where there is one sender and several receivers.
Theorem 6. Based on Theorem 2, the energy consumption of single sender to \( m \) receivers \( E_{stm} \) is calculated as

\[
E_{stm} = E_{stm}^s + m \cdot E_{stm}^r
+ \frac{s_d}{r} \left( P_a + \left( \frac{T_{po}}{T_{pp} + T_{po}} \cdot \frac{T_a}{T_a + T_o} \right) \cdot m \cdot P_r \right)
\]

where

\[
E_{stm}^s = P_a \cdot e_{am}^s + P_o \cdot e_{om}^s
\]

\[
E_{stm}^r = P_a \cdot e_{am}^r + P_o \cdot e_{om}^r
\]

\[
e_{am}^s = e_{am}^r = e_{am}^s
\]

\[
e_{om}^s = \frac{T_a - T_o}{T_o} \cdot e_{ofm}^s + \frac{T_o}{T_o} \cdot e_{olm}^s
\]

\[
e_{om}^r = \frac{T_a - T_o}{T_o} \cdot e_{ofm}^r + \frac{T_o}{T_o} \cdot e_{olm}^r
\]

\[
e_{olm}^s = e_{ol}^s, e_{olm}^r = e_{ol}^r
\]

\[
e_{ofm}^s = T_a \cdot \left( \frac{T_{po}}{T_{po} + T_{pp}} \cdot \frac{T_{pp}}{T_{pp} + T_{po}} + \frac{T_{pp}}{T_{pp} + T_{po}} \cdot P_{pp} \right) + T_o \cdot P_{sleep} + P_a \cdot e_{am}^r + P_o \cdot e_{olm}^r
\]

\[
e_{ofm}^r = T_a \cdot \left( \frac{T_{po}}{T_{po} + T_{pp}} \cdot \frac{T_{pp}}{T_{pp} + T_{po}} + \frac{T_{pp}}{T_{pp} + T_{po}} \cdot P_{pp} \right) + T_o \cdot P_{sleep} + P_a \cdot e_{am}^r + P_o \cdot e_{olm}^r
\]

Proof. In the single-to-\( m \) model, it is allowed that the sender only send data to any receiver successfully. Similar to the calculation of delay, the probability of cases in the single-to-single model should be modified to \( P_a \) and \( P_o \), which represent the probability of any receiver awake and that of no receiver awake, respectively. In addition, when the sender successfully sends the preamble to one receiver, the preamble may be also obtained by other receivers in awake duration, and the expected number of receivers is \( T_{po}/(T_{pp} + T_{po}) \cdot T_a/(T_a + T_o) \cdot m \). Therefore, the energy for receiving data packets should be modified as \( (T_{po}/(T_{pp} + T_{po}) \cdot T_a/(T_a + T_o)) \cdot m \cdot P_r \cdot s_d/r \). The rest of the calculation method is the same as the single-to-single model, which is not elaborated in detail.

Figures 13 and 14 illustrate the change of energy consumption with channel parameters in the single-to-\( m \) model. First, from Figure 13, we see that the total energy consumption increases with the number of receivers, because more receivers may be awake and receive the data packets simultaneously. In addition, the increase of awake duration and polling duration brings more energy consumption in the single-to-m model. Second, from Figure 14, the reduction of channel parameters such \( \mu \) and \( \gamma \) also increases the energy consumption.
N-to-single model. In this section, we continue promoting to discuss the n-to-single model. There are several senders and one receiver in the network. The formula of energy consumption and its proof are as follows.

**Theorem 7.** Based on Theorem 3, the energy consumption of n sender to single receiver $E_{nts}$ is calculated as

$$E_{nts} = n \cdot E^s_{nts} + E^r_{nts} + \alpha \cdot n \cdot \frac{s_d}{r} \cdot (P_s + P_r) \quad (11)$$

where

$$E^s_{nts} = \frac{T_o}{T_o + T_o} \cdot e^s_{on} + \frac{T_o}{T_o + T_o} \cdot e^s_{on}$$

$$E^r_{nts} = \frac{T_o}{T_o + T_o} \cdot e^r_{on} + \frac{T_o}{T_o + T_o} \cdot e^r_{on}$$

$$e^s_{on} = \frac{T_p}{T_p + T_p} \cdot e^s_{pon} + \frac{T_p}{T_p + T_p} \cdot e^s_{pon}$$

$$e^r_{on} = \frac{T_p}{T_p + T_p} \cdot e^r_{pon} + \frac{T_p}{T_p + T_p} \cdot e^r_{pon}$$

$$e^s_{pon} = \frac{1}{\mu} \left( \frac{\theta}{1 - \frac{T_p}{T_p + T_p} \cdot \frac{T_o}{T_o + T_o}} \right)$$

$$\tau_p \cdot \frac{P_{pp}}{T_o} + \frac{s_d}{r} \cdot P_r$$

$$e^r_{pon} = \frac{1}{\mu} \left( \frac{\theta}{1 - \frac{T_p}{T_p + T_p} \cdot \frac{T_o}{T_o + T_o}} \right)$$

$$\tau_p \cdot \frac{P_{pp}}{T_o} + \frac{s_d}{r} \cdot P_s$$

$$e^s_{ppn} = e^s_{pon} + \frac{T_p}{T_o} \cdot \frac{T_p}{T_p}$$

$$e^r_{ppn} = e^r_{pon} + \frac{T_p}{T_o} \cdot \frac{T_p}{T_p}$$

$$e^s_{on} = \frac{T_o}{T_o} \cdot e^s_{on} + \frac{T_o}{T_o} \cdot e^s_{on}$$

$$e^r_{on} = \frac{T_o}{T_o} \cdot e^r_{on} + \frac{T_o}{T_o} \cdot e^r_{on}$$

$$e^s_{ola} = e^s_{ona} + \frac{T_o}{T_o} \cdot \frac{T_o}{T_o}$$

$$e^r_{ola} = e^r_{ona} + \frac{T_o}{T_o} \cdot \frac{T_o}{T_o}$$

$$e^s_{off} = T_o \left( \frac{T_p}{T_p + T_p} \cdot \frac{T_p}{T_p + T_p} \cdot \frac{P_p}{P_p} \right)$$

$$+ T_o \cdot P_{sleep} + \frac{T_o}{T_o + T_o} \cdot e^r_{on} + \frac{T_o}{T_o + T_o} \cdot e^r_{on}$$

$$e^r_{off} = T_o \left( \frac{T_p}{T_p + T_p} \cdot \frac{T_p}{T_p + T_p} \cdot \frac{P_p}{P_p} \right)$$

$$+ T_o \cdot P_{sleep} + \frac{T_o}{T_o + T_o} \cdot e^r_{on} + \frac{T_o}{T_o + T_o} \cdot e^r_{on}$$

**Proof.** The calculation of energy consumption in the n-to-single model is similar to the single-to-single model.

However, the difference is that we should consider the data collision when there are multiple senders. According to the modification of section “N-to-single model” in section “Transmission delay analysis,” we should change the expected transmitting times in $e^s_{pon}$ and $e^r_{pon}$ from $1/\mu \cdot (1 + T_{pp}/T_{pp} + T_{po})$ to

$$\frac{1}{\mu} \left( \frac{\theta}{1 - \frac{T_p}{T_p + T_p} \cdot \frac{T_o}{T_o + T_o}} \right)$$

when considering the data collision. The other parts of the calculation are demonstrated in detail in section “N-to-single model” in section “Transmission delay analysis,” and the basic thought is that the duration of the different states is multiplied by the corresponding power.

Figures 15 and 16 illustrate the change of energy consumption with MAC parameters and channel parameters. First, the total energy consumption increases with the number of senders, and larger awake duration or polling duration leads to more energy consumption in the n-to-single model. Moreover, from Figure 16, the reduction of channel parameters such $\mu$ and $\gamma$ also increases the energy consumption, which is consistent with Figure 14.

**Data amount analysis.** Before the analysis of n-to-m model, we first focus on the calculation of node number and data amount. In the network, the data amount undertaken by nodes changes with the distance of the node from the sink area. In this section, we expand the scenario to the whole network, and establish a model for calculating the data amount.
According to Huang et al., assume that the radius of network is $R$, the node density is $\rho$, and the probability of each node generating a packet is $\alpha$. The network area is divided into $L$ layers, so the width of each layer is $R/L$. The sink nodes are deployed in the first layer, also the center of network. Nodes in two adjacent layers can transmit data packets, so the data amount in the $l$th layer will be added into the $(l+1)$th layer. Based on the assumption above, the data amount of the $l$th layer can be calculated as equation (12). In this formula

$$E_l = E'_l + E''_l$$

where

$$E'_l = \rho \cdot \pi \left[ R^2 - \left( \frac{R}{L} \cdot (l - 1) \right)^2 \right] \cdot E_{ntm} + A_{l+1} \cdot \frac{S_d}{\gamma} \cdot P_r$$

$$E''_l = \rho \cdot \pi \left[ R^2 - \left( \frac{R}{L} \cdot (l - 1) \right)^2 \right] \cdot E_{ntm} + A_{l} \cdot \frac{S_d}{\gamma} \cdot P_s$$

Proof: According to section “Data amount analysis,” we know that the nodes at the $l$th layer should receive the data packets from the $(l+1)$th layer and forward the data to the next layer. In addition, the data generated by the nodes at the $l$th layer also should be sent to the next layer. Therefore, the energy consumption consists two parts $E'_l$ and $E''_l$. In the calculation of $E'_l$

$$\rho \cdot \pi \left[ R^2 - \left( \frac{R}{L} \cdot (l - 1) \right)^2 \right] \cdot E_{ntm}$$

is the energy consumption in the stage of communication establishment, where

$$\rho \cdot \pi \left[ R^2 - \left( \frac{R}{L} \cdot (l - 1) \right)^2 \right]$$

is the node number in the $l$th layer. The calculation of $E_{ntm}$ is similar to $E_{ntm}$ in the $n$-to-single model, but the probabilities $T_o/T_a + T_o$ and $T_o/T_a + T_a$ should be modified to $P_o$ and $P_o$ according to the single-to-$m$ model. And $A_{l+1} \cdot S_d/\gamma \cdot P_r$ is the energy for receiving the data packets from the $(l+1)$th layer, where $A_{l+1}$ is the packet number in the $(l+1)$th layer. Moreover, the energy consumption of sending data packets to the next layer $E''_l$ is also calculated as the way above.
Figure 18 shows that the energy consumption of the nodes in different layers. As the distance from sink increases, the data amount undertaken by nodes is smaller and the energy consumption reduces. And the increase of MAC parameters brings more consumed energy. Moreover, from Figure 19, we see there is more residual energy in nodes of outer layer when the initial energy of nodes is fixed. And the data generation rate has great impact on the residual energy. Therefore, the residual energy of outer nodes can be used to reduce more delay.

APOPM protocol

The proposal of the APOPM protocol

According to the calculation models of delay and energy consumption established in section “The parameter analysis of polling-based MAC protocol,” the adjustment of various network parameters, such as awake duration and polling duration, will impact on the performance of network. Therefore, a new APOPM MAC protocol is proposed to find suitable network parameters to ensure that the energy efficiency of network while the delay reduces. The APOPM MAC protocol uses an adaptive MAC parameter strategy that different network parameters are applied to different network areas to maintain the high performance. Specifically, the lifetime of nodes near the sink area determines that of the entire network. Therefore, the APOPM MAC protocol is used in the area close to sink to low the energy consumption of nodes and maintain the network lifetime. Otherwise, the nodes far from sink area have energy surplus, so the MAC parameters will be adjusted to use the residual energy to continue reducing the delay.

Therefore, the optimization of the APOPM protocol mainly consists of two organic components. The first is to find the optimal MAC parameters for sink nodes, which improve the performance on energy consumption, transmission delay, and data transmission success rate in a tradeoff way. Then, the lifetime of sink nodes can be prolonged and the delay is controlled to be not excessively large. The second step is to optimize the performance of the outer nodes. N-to-m model for energy consumption reveals that the outer nodes still have residual energy when the network dies, which causes energy waste. And we find the farther the node is away from the sink, the more energy is left. Therefore, the MAC parameters are dynamically optimized according to the different physical positions of the nodes, which appropriately use the residual energy and optimize other performance such as transmission delay and transmission success rate.

The performance optimization for sink nodes

According to the characteristic that sink nodes only receive data from outer nodes and do not send packets, we choose the n-to-m model to calculate the transmission delay and energy consumption. Therefore, the objective function $O(T_a, T_{po})$ is shown in equation (14), where $\lambda_D$ and $\lambda_E$ are the weights of two performance indicators, respectively. Our goal is to find the optimal parameters $T_a^*$ and $T_{po}^*$ which make the value of objective function reaching minimum

$$O(T_a, T_{po}) = \lambda_D \cdot D_{sink} + \lambda_E \cdot E_{sink}$$

s.t. $T_a \in [0, T_{cycle}]$, $T_{po} \in [0, T_{cycle}]$  \hspace{1cm} (14)

Aiming at the above optimization problem, the common method is to obtain the partial derivative of the objective function for several parameters, and apply the gradient descent method to get the optimal solution.
However, the gradient descent method only directly applies if the objective function is continuous and convex in the definition domain. If the objective function is not strictly convex, there may be multiple local minima. Gradient descent may fall into a local minimum and fail to reach the global optimum. Therefore, we first judge that the objective function is not convex. Then, according to the characteristic of the problem, an alternate optimization method for multiple MAC parameters is proposed.\textsuperscript{43}

**Theorem 9.** The MAC parameter optimization for sink nodes is not a convex problem.

**Proof.** The objective function of MAC parameter optimization problem for the sink nodes is a multivariate function. The principle of determining whether a function with several variables is convex is that the Hessian matrix of the function is semi-positive definite. Since the theoretical calculation of $D_{\text{sink}}$ and $E_{\text{sink}}$ is excessively complicated, the scientific computing modules SymPy in Python is applied to solve the second-order partial derivatives of the objective function separately, and the Hessian matrix $H$ is obtained, as shown in equation (15)

$$H = \begin{bmatrix} \frac{\partial^2 O}{\partial T_{po}^2} & \frac{\partial^2 O}{\partial T_{po} \partial T_a} \\ \frac{\partial^2 O}{\partial T_{po} \partial T_a} & \frac{\partial^2 O}{\partial T_a^2} \end{bmatrix} \quad (15)$$

According to the calculation result of SymPy, $\frac{\partial^2 O}{\partial T_a \partial T_{po}} = \frac{\partial^2 O}{\partial T_{po} \partial T_a}$, which shows that $H$ is a symmetric matrix and also proves that the objective function is continuously derivative of the second order in the definition domain, which meets the first condition of using gradient descent method. Since the second-order derivative expression in $H$ is complicated and it is hard to find the eigenvalues of $H$, we transform the problem to determine whether the principal minors of $H$ are all greater than 0. The first-order principal minor of $H$ is $\frac{\partial^2 O}{\partial T_{po}^2}$, which is constantly greater than or equal to 0 in the definition domain, and its trend is shown in Figure 20. The second-order principal minor is $(\frac{\partial^2 O}{\partial T_{po} \partial T_a}) \cdot (\frac{\partial^2 O}{\partial T_a^2}) - (\frac{\partial^2 O}{\partial T_{po} \partial T_a})^2$, whose function image is shown in Figure 21, does not satisfy the condition that the second-order principal minor is constantly greater than or equal to 0. Therefore, $H$ is not a semi-positive definite matrix, and the objective function is not a convex function.

Theorem 9 proves that the objective function $O(T_a, T_{po})$ is not convex, so there is no guarantee that the final result of applying gradient descent optimization directly is globally optimal. However, although our optimization problem is not convex for multiple MAC parameters, it is convex for the unitary functions $O(T_{po})$ and $O(T_a)$, that is, fixing $T_a$ and $T_{po}$ as constant value, respectively. Therefore, an alternate optimization method for multiple MAC parameters is proposed.\textsuperscript{43}

**Theorem 10.** Fixing the value of $T_a$, $O(T_{po})$ is a convex function in the definition domain. Similarly, fixing the value of $T_{po}$, $O(T_a)$ is a convex function in the definition domain.

**Proof.** Fixing $T_a$ to $T_{a0}$, the objective function becomes a function with only one variable. The trend of its second-
order derivative $\frac{\partial^2 O}{\partial T^2_p}$ in the definition domain is already shown in Figure 20. $\frac{\partial^2 O}{\partial T^2_p}$ is constantly greater than 0 in the definition domain, and $O(T_p)$ is also a convex function. Therefore, an alternate optimization method for multiple MAC parameters is proposed to find the optimal value of $T_a$ and $T_{po}$.

Algorithm 1 shows the process of the alternate optimization method, where $R$ is the number of iterations, and $\sigma_a$ and $\sigma_{po}$ are the learning rates for updating $T_a^r$ and $T_{po}^r$, respectively. The algorithm first fixes $T_{po}$ as $T_{po}^r$ and updates the value of $T_a^r$ using the gradient descent method. The termination condition of the iterative search for the lowest point of $O(T_a, T_{po})$ is $|T_a^{r+1} - T_a^r| \leq \Delta$, where $\Delta$ is a preset small threshold. After determining $T_a^{r+1}$, $T_a$ is fixed and $T_{po}^{r+1}$ is updated in the same way for the purpose of alternate optimization parameters.

The performance optimization of outer nodes

In section “The performance optimization for sink nodes,” Algorithm 1 is applied to find the optimal parameter value for sink nodes and balance the delay and energy consumption. However, based on the previous analysis, sink nodes will first run out of energy, while the outer nodes have residual energy. Therefore, we next propose the APOPM algorithm to optimize the parameter value adaptively according to the physical location of nodes. Then, the simulations also illustrate the performance of optimizing method.

Algorithm 1. Alternate optimization algorithm for sink nodes.

**Input:** $O, R, \sigma_a, \sigma_{po}$

**Output:** $T_a, T_{po}$

1. Initialize $T_a^0$ and $T_{po}^0$
2. $T_a, T_{po} \leftarrow \max$
3. For $r = 0, 1, 2 \ldots R$
   4. While $|T_{po}^{r+1} - T_{po}^r| > \Delta$
   5. $G_{a}(T_a) = O(T_a, T_{po})$
   6. $T_a^{r+1} \leftarrow T_a^{r} - \sigma_a G_{a}(T_a)$
   7. $T_{po}^{r+1} \leftarrow T_{po}^{r+1} - \sigma_{po} G_{po}(T_{po})$
   8. End While
   9. $T_{po}^{r+1} \leftarrow T_{po}^{r}$
10. End For
11. Return $\left(T_a^R, T_{po}^R\right)$

Algorithm 2. APOPM algorithm.

**Input:** Nodes, $O, L, \epsilon_a, \epsilon_{po}$

**Output:** MAC parameters of each layer

1. Initialize the MAC parameters of nodes in Nodes
2. Initialize the energy of nodes $E_{init}$
3. Apply Algorithm 1 to find $T_a$ and $T_{po}$
4. $T_a^L \leftarrow T_a, T_{po}^L \leftarrow T_{po}$
5. For $l = 2, 3 \ldots L$
   6. $T_a^L \leftarrow T_a^L + \epsilon_a, T_{po}^L \leftarrow T_{po}^L + \epsilon_{po}$
   7. $E_{res} \leftarrow \max$
   8. While $E_{res} > 0$
   9. $T_a^l \leftarrow T_a^l + \epsilon_a, T_{po}^l \leftarrow T_{po}^l + \epsilon_{po}$
10. Calculate $E_l$ according to equation (13)
11. $E_{res} \leftarrow E_{res} - E_l$
12. End While
13. End For
14. Return $T_a^1, T_a^2, T_{po}^L$

Algorithm 2 shows the process of APOPM algorithm, where Nodes is the node list in the network. $L$ is the total number of layers in the network, and sink nodes are located at the first layer. $\epsilon_a$ and $\epsilon_{po}$ are the update strength of $T_a$ and $T_{po}$, respectively. In the algorithm, the residual energy of nodes in each layer of the network is calculated. And the nodes still have residual energy when sink nodes die, which means that their MAC parameter value can continue to increase to reduce the data transmission delay. After the layer-by-layer optimization of Algorithm 2, the nodes in the network find suitable parameter value: the outer nodes make full use of the remaining energy to reduce the delay, and the inner nodes try to save energy and extend the lifetime within the tolerance range of delay.
Next, we analyze the cost of APOPM algorithm. The iteration number for parameter alternate optimization in Algorithm 1 is $R$, and $\omega$ parameters (two in this research, $T_a$ and $T_{po}$) need to be optimized in each iteration. For each parameter, it is necessary to find the minimum value using gradient descent. Assume that the average number of computing gradient is $t$, so the time complexity of Algorithm 1 is $O(R \cdot \omega \cdot t)$. In addition, there is the optimization for outer nodes at $(L - 1)$ layers. However, since the algorithm is sequential optimization, the parameters are always incremental. Therefore, the maximum iteration number is only related to the nodes at the $L$th layer. Suppose the final parameter value of nodes at the $L$th layer is $T_a^L$ and $T_{po}^L$, then $(T_a^L - T_a^1)/\epsilon_a$ or $(T_{po}^L - T_{po}^1)/\epsilon_{po}$ is the total number of iterations, so the updating number is $(T_a^L - T_a^1)/\epsilon_a \cdot \omega$. In summary, the algorithm complexity of APOPM is

$$O\left(\omega \left( Rt + \frac{T_a^L - T_a^1}{\epsilon_a} \right) \right).$$

Analysis of the simulation experiment

In this section, the simulation experiment is designed to prove the performance of the algorithms proposed above. We apply Python to implement the APOPM and simulate the communication establishment and data transmission between nodes. Table 2 lists the initial value of the parameters when using the APOPM protocol to optimize the network. Figure 23 presents the change of transmission delay of nodes at different layers after optimization using Algorithm 1 and APOPM. First, comparing the black curve and red curve, we see that better MAC parameters are obtained by Algorithm 1, and the transmission delay of nodes is reduced. In addition, due to more data undertaken by the sink nodes, the delay is larger compared to the outer nodes. Second, when the APOPM is applied, the performance of outer nodes on delay is better, shown as the blue curve in Figure 23. It is proved that the residual energy of outer nodes can be used to further reduce the transmission delay. The other three curves illustrate the situation when the data generation probability is larger, so the transmission delay is larger.

Based on the theoretical analysis, the change of MAC parameters also impacts on the energy consumption of nodes. Figure 24 illustrates the change of energy efficiency of nodes at different layers after applying Algorithm 1 and APOPM. The energy efficiency is the rate of energy used and the initial energy. From the black curve, we see that the energy efficiency of outer nodes is lower because the data amount at outer layers is smaller. Although the energy efficiency raises generally after applying Algorithm 1, the condition of energy consumption of outer nodes is still unsatisfactory.
However, after using APOPM, the MAC parameters of outer nodes are modified adaptively, so the large residual energy in the outer nodes can be further used, and the energy efficiency is higher.

In summary, the simulation experiment further proves the feasibility of the APOPM proposed. After using APOPM protocol, the average transmission delay of the network is reduced by 22.40%, and the average energy efficiency of the nodes increases by 23.25%.

Conclusion

In multi-sensor systems, the communication between multiple sensors affects and interferes with each other, so it is a challenge issue to analyze the MAC communication protocol between multiple sensors. In this article, we establish the data transmission model of multiple nodes, and theoretically analyze the effect of network parameters such as awake duration and polling duration on the transmission delay and energy consumption of nodes. Based on this, we propose a new protocol called the APOPM protocol to optimize the existing network MAC protocol. Some performance of the network, including transmission delay, network lifetime, and energy efficiency all get improved. According to the relationship between parameters and performance, we propose the optimization algorithm to modify the parameters of the sink nodes and improve the node performance in a compromise strategy. The algorithm can adjust the weights of the two aspects of performance according to the timeliness and energy-saving requirements of the actual network application, and use the parameter alternate optimization method to find the best parameter values according to the characteristics of the objective function that it is continuously derivable and has a minimum value in the range of parameters. Second, we dynamically adjust the parameters of the outer nodes according to their positions, using the residual energy in the network to further reduce the delay. In the experiments, we calculate the theoretical results through the model in section “The parameter analysis of polling-based MAC protocol.” The changes of delay and energy efficiency before and after optimization are given in the form of graphs. Then, we use Python to simulate the data transmitting in the network and present the comparison of performance between the APOPM protocol and the previous protocols, which further proves the effectiveness of the APOPM protocol that achieves the optimization goal of improving network performance.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported, in part, by the National Natural Science Foundation of China (62072475, 61772554, and 61472092).

ORCID iD

Fufang Li https://orcid.org/0000-0002-1448-5665

References

1. Sarkar S, Chatterjee S and Misra S. Assessment of the suitability of fog computing in the context of Internet of things. IEEE T Cloud Comput 2018; 6(1): 46–59.
2. Yu M, Liu A, Xiong N, et al. An intelligent game based offloading scheme for maximizing benefits of IoT-edge-cloud ecosystems. IEEE Internet Things. Epub ahead of print 23 November 2020. DOI: 10.1109/JIOT.2020.3039828.
3. Zhu X, Luo Y, Liu A, et al. Multi-agent deep reinforcement learning for vehicular computation offloading in IoT. IEEE Internet Things. Epub ahead of print 26 November 2020. DOI: 10.1109/JIOT.2020.3040768.
4. Xu X, Zhang N, Song H, et al. Adaptive beaconing based MAC protocol for sensor based wearable system. IEEE Access 2018; 6: 29700–29714.
5. Huang S, Zeng Z, Ota K, et al. An intelligent collaboration trust interconnections system for mobile information control in ubiquitous 5G networks. IEEE T Netw Sci Eng 2021; 8: 347–365.
6. Xu Q, Su Z, Dai M, et al. APIS: privacy-preserving incentive for sensing task allocation in cloud and edge-cooperation mobile Internet of things with SDN. IEEE Internet Things 2020; 7: 5892–5905.
7. Liu Y, Liu A, Zhang N, et al. DDC: dynamic duty cycle for improving delay and energy efficiency in wireless sensor networks. J Netw Comput Appl 2019; 131: 16–27.
8. Ouyang Y, Liu A, Xiong N, et al. An effective early message ahead joint adaptive data aggregation scheme for sustainable IoT. IEEE T Netw Sci Eng 2021; 8: 201–219.
9. Liu Y, Liu X, Liu A, et al. A trust computing-based security routing scheme for cyber physical systems. ACM T Intel Syst Tec 2019; 10(6): 61.
10. Zhu X, Luo Y, Liu A, et al. A deep learning-based mobile crowdsensing scheme by predicting vehicle mobility. IEEE T Intell Transp. Epub ahead of print 7 October 2020. DOI: 10.1109/TITS.2020.3023446.
11. Liu X, Song H and Liu A. Unmanned aerial vehicle trajectory optimization for improved data collection in social networks. IEEE T Netw Sci Eng. Epub ahead of print 9 September 2020. DOI: 10.1109/TNSE.2020.3017556.
12. Huang S, Liu A, Zhang S, et al. BD-VTE: a novel baseline data based verifiable trust evaluation scheme for smart network systems. IEEE T Netw Sci Eng. Epub ahead of print 7 August 2020. DOI: 10.1109/TNSE.2020.3014455.
13. Li T, Liu A, Xiong N, et al. A trustworthiness-based vehicular recruitment scheme for information collections in Distributed Networked Systems. *Inform Sciences* 2021; 545: 65–81.

14. Huang C, Huang G, Liu W, Xie M, et al. A parallel joint optimized relay selection protocol for wake-up radio enabled WSNs. *Phys Commun* 2021; 47: 101320.

15. Guo J, Liu A, Ota K, et al. ITCN: an intelligent trust collaboration network system in IoT. *IEEE T Netw Sci Eng*. Epub ahead of print 11 February 2021. DOI: 10.1109/TNSE.2021.3057881.

16. Li F, Huang G, Yang Q, et al. Adaptive contention window MAC protocol in a global view for emerging trends networks. *IEEE Access* 2021; 9: 18402–18423.

17. Li A, Liu Q, Zeng L, et al. An efficient data aggregation scheme based on differentiated threshold configuring joint optimal relay selection in WSNs. *IEEE Access* 2021; 9: 19254–19269.

18. Hafeez KA, Zhao L, Ma B, et al. Performance analysis and enhancement of the DSRC for VANET's safety applications. *IEEE T Veh Technol* 2013; 62(7): 3069–3083.

19. Liu X, Liu A, Wang T, et al. Adaptive data and verified message disjoint routing for gathering big data in energy harvesting networks. *J Parallel Distri Com* 2020; 135: 140–155.

20. Li X, Tan J, Liu A, et al. A novel UAV-enabled data collection scheme for intelligent transportation system through UAV speed control. *IEEE T Intell Transp*. Epub ahead of print 17 December 2020. DOI: 10.1109/ITITS.2020.3040557.

21. Liu X, Lin P, Liu T, et al. Objective-variable tour planning for mobile data collection in partitioned sensor networks. *IEEE T Mobile Comput*. Epub ahead of print 17 June 2020. DOI: 10.1109/TMC.2020.3003004.

22. Shen M, Liu A, Huang G, et al. ATTDCC: an active and traceable trust data collection scheme for industrial security in smart cities. *IEEE Internet Things* Epub ahead of print 5 January 2021. DOI: 10.1109/JIOT.2021.3049173.

23. Wang T, Luo H, Zeng X, et al. Mobility based trust evaluation for heterogeneous electric vehicles network in smart cities. *IEEE T Intell Transp* 2021; 22: 1797–1806.

24. Zhou C, Gu Y, Fan X, et al. Direction-of-arrival estimation for coprime array via virtual array interpolation. *IEEE T Signal Proces* 2018; 66(22): 5956–5971.

25. Teng H, Dong M, Liu Y, et al. A low-cost physical location discovery scheme for large-scale Internet of things in smart city through joint use of vehicles and UAVs. *Future Gener Comp Sys* 2021; 118: 310–326.

26. Polastre J, Hill JL and Culler DE. Versatile low power media access for wireless sensor networks. In: *Proceedings of the 2nd international conference on embedded networked sensor systems*, Baltimore, MD, 3–5 November 2004, pp.95–107. New York: ACM.

27. Zhou C, Gu Y, He S, et al. A robust and efficient algorithm for coprime array adaptive beamforming. *IEEE T Veh Technol* 2018; 67(2): 1099–1112.

28. Sun Y, Gurewitz O and Johnson DB. RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In: *Proceedings of the 6th ACM conference on embedded network sensor systems*, Raleigh, NC, 5–7 November 2008, pp.1–14. New York: ACM.

29. Siddiqui S, Ghani S and Khan AA. ADP-MAC: an adaptive and dynamic polling-based MAC protocol for wireless sensor networks. *IEEE Sens J* 2018; 18(2): 860–874.

30. Buettner M, Yee GV, Anderson E, et al. X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In: *Proceedings of the 4th international conference on embedded networked sensor systems*, Boulder, CO, 31 October–3 November 2006, pp.307–320. New York: ACM.

31. Huang W, Ota K, Dong M, et al. Result return aware offloading scheme in vehicular edge networks for IoT. *Comput Commun* 2020; 164: 201–214.

32. Li G, Li F, Wang T, et al. Bi-adjusting duty cycle for green communications in wireless sensor networks. *EURASIP J Wirel Comm*. Epub ahead of print 31 August 2020. DOI: 10.1186/s13638-020-01767-5.

33. Ouyang Y, Zeng Z, Li X, et al. A verifiable trust evaluation mechanism for ultra-reliable applications in 5G and beyond networks. *Comp Stand Inter* 2021; 77: 103519.

34. Liu S, Huang G, Gui J, et al. Energy-aware MAC protocol for data differentiated services in sensor-cloud computing. *J Cloud Comput* 2020; 9: 48.

35. Mo W, Wang T, Zhang S, et al. An active and verifiable trust evaluation approach for edge computing. *J Cloud Comput* 2020; 9: 51.

36. Liu Y, Wang T, Zhang S, et al. Artificial intelligence aware and security-enhanced traceback technique in mobile edge computing. *Comput Commun* 2020; 161: 375–386.

37. Zhang M, Dong C and Huang Y. FS-MAC: an adaptive MAC protocol with fault-tolerant synchronous switching for FNANETs. *IEEE Access* 2019; 7: 80602–80613.

38. Kumar S and Kim H. BH-MAC: an efficient hybrid MAC protocol for vehicular communication. In: *Proceedings of the 2020 international conference on communication systems & networks (COMSNETS)*, Bengaluru, India, 7–11 January 2020, pp.362–367. New York: IEEE.

39. Deng D, Rao W, Liu B, et al. TA-MAC: a traffic-aware TDMA MAC protocol for safety message dissemination in MEC-assisted VANETs. In: *Proceedings of the 2020 29th international conference on computer communications and networks (ICCCN)*, Honolulu, HI, 3–6 August 2020, pp.1–9. New York: IEEE.

40. Byun H, Son S and So J. Queue management based duty cycle control for end-to-end delay guarantees in wireless sensor networks. *Wirel Netw* 2013; 19(6): 1349–1360.

41. Masood Ur R, Irfan ud D, Adnan M, et al. VTA-SMAC: variable traffic-adaptive duty cycled sensor MAC protocol to enhance overall QoS of S-MAC protocol. *IEEE Access* 2021; 9: 33030–33040.

42. Al-Janabi TA and Al-Raweshidy HS. An energy efficient hybrid MAC protocol with dynamic sleep-based scheduling for high density IoT networks. *IEEE Internet Things* 2019; 6(2): 2273–2287.

43. Gur E, Sabach S and Shtern S. Alternating minimization based first-order method for the wireless sensor network localization problem. *IEEE T Signal Proces* 2020; 68: 6418–6431.