

RESEARCH

Open Access



# Energy-aware MAC protocol for data differentiated services in sensor-cloud computing

Shun Liu<sup>1</sup>, Guosheng Huang<sup>2</sup>, Jinsong Gui<sup>1</sup>, Tian Wang<sup>3</sup> and Xiong Li<sup>4\*</sup>

## Abstract

The energy-aware MAC protocol is the basic communication protocol for device-to-device communication in sensor-cloud computing and facilitates data sensing, computing, and sharing for applications. Transmitting high-priority data to control centers quickly to manage emergencies is essential for sensor-cloud applications. In this paper, a Dynamic Wait time-based MAC (DWT-MAC) protocol is proposed for data differentiated services of a sensor-cloud system (SCS). The DWT-MAC protocol is novel in that it changes the receiver wait time, which has been fixed in previous MAC protocols, and it reduces the delay in data transmission by dynamically adjusting the wait time. In the DWT-MAC protocol, the wait time changes according to the number of senders, which can ensure that it always approaches the optimal value. A dynamic time adjustment algorithm, which causes the wait time to always trail the optimal value, is proposed to reduce delays. While the DWT-MAC protocol cannot ensure wait times of the optimal value, it can ensure that it quickly comes close to the optimal value, which is suitable for dynamically changing networks. Extensive experiments show that the DWT-MAC protocol reduces the average delay in the transmission of data of the highest priority by 49.3%.

**Keywords:** Edge cloud, Energy-aware, DWT-MAC protocol, Delay differentiated services, Lifetime

## Introduction

Cloud computing has been widely used by the scientific community and industry, because users can benefit from computing infrastructures at low costs [1–3]. At the same time, big data networks have attracted the attention of numerous researchers [4–6]. However, the development of big data networks is attributable to the development of numerous applications with several sensing devices [7–9]. A sensor-cloud system (SCS) combines cloud computing [10–12] with sensor nodes [13–15] to sense, collect, process, analyze, store and share data for various applications; this combination has largely promoted the development of big data networks [16–18].

In recent years, with the development of microprocessor technologies [19], devices connected to the Internet have become more prevalent [20–22] with the number of such devices anticipated to increase to 23 billion by 2020 according to relevant research [19]. The computing, storage, and communication power of these devices has also developed considerably with network edges accumulating numerous resources for computing, storage, and communication, forming the so-called edge network [23–25]. This has resulted in a computation shift from the cloud to the edge, and the combination of edge and cloud networks has formed the edge cloud computing mode, which has considerably changed existing networks [26, 27]. However, most of these large devices are embedded sensing devices, which are the main source of big data networks. We can obtain large amounts of data because these sensing devices can sense surrounding environments anytime and anywhere. A Cisco report

\* Correspondence: [lixiongzhq@163.com](mailto:lixiongzhq@163.com)

<sup>4</sup>Institute for Cyber Security, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

Full list of author information is available at the end of the article



© The Author(s). 2020 **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

shows that data flows generated by the Internet of Things (IoT) account for 69% of data flows generated by the Internet, reflecting a 30-fold increase relative to levels in 2000. Data flows are still increasing, continuously promoting the development of big data networks [28]. Data transmission, data protection and intrusion detection technologies can help guarantee the security of IoT data [29].

However, numerous challenges still emerge when massive data enter big data networks. Communication between sensing devices follows a basic protocol [30] and creates challenges in big data networks due to its influence on every aspect of big data network performance [31–36]. Device-to-device (D2D) communication has attracted considerable research interest as a basic mode of network computing [30, 36]. The MAC protocol is an important protocol of D2D communication [30, 36]. It influences every aspect of communication performance. Communication performance is also affected by other communication constraints such as bandwidth and communication noise [34–36]. The two most important aspects of communication it influences are as follows. (1) Energy efficiency [37]. Generally, most sensing devices are powered by batteries, which provide very limited energy, creating a need for high energy efficiency [38]. Under normal circumstances, node energy consumption is divided into four states as follows: (a) sending data; (b) receiving data; (c) idle listening; and (d) sleeping. The first three states consume the most energy. Therefore, network nodes should sleep when possible to save energy. In most cases, nodes work based on a duty cycle [38], i.e., nodes periodically shift between the active and sleep modes to save energy [39]. (2) Delay in data transmission. Transmitting and storing data reliably and quickly is critical [40]. The delay in data transmission is the expected delay in communication between devices and is mainly attributable to communication protocols [3, 9, 30]. As a node only sends and receives data when it is active, when a node is sleeping, the sender must wait until the receiver wakes for data operation [39], shortening the node's active time and increasing delays. At the same time, when transmitting many-to-one data through a sensor-cloud, different MAC protocols can greatly influence delays.

The MAC protocol for sensor-cloud systems is divided into two MAC protocols for actively initiating communication: (1) the sender-initiated MAC protocol and (2) receiver-initiated MAC protocol. Under the sender-initiated MAC protocol, when a sender has data to be transmitted, the sender listens to and polls the channel, and if the receiver is awake, the receiver can receive the data transmitted by the sender and establish a

communication, resulting in data transmission. Under the receiver-initiated MAC protocol, when the sender must transmit data, it does not initiate data transmission and must wait for a beacon from the receiver indicating that the receiver is ready to receive data [41]. When the receiver wakes, it actively sends the beacon to receive data and initiate communication. As the sender waits for the beacon, it can communicate with other receivers until achieving data transmission [41].

In most sensor-cloud systems, sensing devices are relatively dense, and there are always multiple sensing devices with data to transmit at the same time within a communication range; in other words, there are a many-to-one cases of communication in which multiple senders must transmit data to the same receiver [39]. Data from different senders usually is of varying priority [41], and of course high priority data must be transmitted first, as this is good for the system [41]. For example, in an emergency, alarm data should be transmitted first while less important data such as daily life data can be transmitted later. Therefore, researchers have designed data transmission protocols that prioritize data in the MAC protocol layer. Such protocols are helpful, as in many cases of sensor-cloud system communication, there is only one channel, meaning that high priority data must be transmitted first when multiple senders must transmit data. Sarang et al. [41] developed an MAC protocol allowing a receiver-initiated network to transmit high priority data faster. Under this protocol, when the receiver wakes up, it broadcasts a wake up beacon to indicate that it is ready for data transmission; then, it initiates a wait time of a fixed duration for sender nodes to reply to it. The wait time is usually determined by the number of sender nodes. Thus, if there are  $n$  sender nodes, the receiver's wait time is equal to the duration required to transmit  $n$  Tx-Beacons. If the sender node has data to transmit, it will listen to the channel. If it receives a wake up Beacon from the receiver, it transmits a Tx-beacon to the receiver. Because a Tx-Beacon contains priority data, after the receiver receives Tx-Beacons from all sender nodes, it communicates with the sender with data of the highest priority to transmit this information faster.

While past studies have realized ways of transmitting prioritized data, such methods present the following shortcomings: (1) Long delays. When there are numerous sender nodes, wait times are long, causing establishing communication before data transmission to take a considerable amount of time for each set of data and resulting in long delays. In a multi-hop network, multi-hop data transmission results in considerable delays. (2) Inefficient energy consumption. While there are many sender nodes in a network, not every sender node has data to transmit most of the time. However, a receiver

must reserve time to transmit Tx beacons for each sender node. When some sender nodes do not have data to transmit, some wait time is wasted because the receiver must reserve time for these sender nodes, resulting in long periods of empty consumption, invalid energy consumption, and long network delays.

The above analysis shows that the current MAC protocol for prioritized data is mainly limited in that the wait time set by the receiver is not equal to the number of sender nodes, and the fixed wait time results in long delays and low energy efficiency. This is the case because while a receiver may have multiple sender nodes, not every sender node has data to transmit. The method reserves time for each sender node, which limits the protocol's flexibility and performance. Thus, a more energy efficient Dynamic Wait time-based MAC (DWT-MAC) protocol with shorter delays is proposed in this work. The paper makes the following contributions:

- (1) In this paper, a Dynamic Wait time based MAC protocol is proposed to reduce delays and enhance the energy efficiency of data differentiated services of a sensor-cloud system (SCS). The DWT-MAC protocol differs from the previous strategy with a fixed wait time by reducing data transmission delays by dynamically adjusting the wait time. Under the DWT-MAC protocol, wait times change depending on the number of senders, which can ensure that they are always close to optimal. This dynamic wait time strategy with wait times always close to the optimal value is intended to reduce delays. While the protocol cannot ensure that wait times are always of the optimal value, it ensures that wait times are close to the optimal value, making it very suitable for networks with changing data.
- (2) Our numerous studies and experiments show that the strategy proposed here is better than the previous strategy. Under the DWT-MAC protocol, when wait times exceed the number of senders (i.e., for  $f$  senders, the wait time exceeds the period required to transmit  $f$  Tx-Beacon), this has almost no effect on high priority data. When wait times fall below what is required for the number of senders (i.e., wait times fall below what is required to transmit  $f$  Tx-Beacon), this may influence some high priority data, but network data transmission delays can be reduced. In both cases, the wait time is adjusted before the next round of data transmission, causing wait times to approach optimization. Our numerous experiments show that the DWT-MAC protocol is better than the previous strategy by 49.3%.

The rest of this paper is organized as follows. In "Related work" section, we review the related work. "System

model and problem statement" section describes the network model and formulate the optimization problem to reduce delay. In "The design of the DWT-MAC protocol" section, we propose DWT-MAC protocol. Simulation results are presented in "Experimental results and analysis" section and the entire paper is concluded in "Conclusion" section.

## Related work

With the development of the Internet of Things (IoT) [15], more and more sensor devices are deployed in various applications, such as vehicles network [18, 42], traffic monitoring, healthcare [43], security monitoring [44–46], and industrial networks [47–49]). Numerous sensor devices are used to sense and collect data, providing rich data for various applications [50]. As the computing and storage power of these devices has improved and with the development of microprocessor technology, a major shift in current network architectures has occurred. The large number of computing and storing resources gathered at the edges of networks have formed Edge networks. The center of network computing has thus shifted from the cloud to the edge. Edge networks generate large amounts of data and have formed big data networks. The combination of these forms has led to a considerable shift in network computing modes. Networks are developing from those of single cloud computing those combining edge and cloud computing [1, 2, 5, 7].

Among sensor devices, device-to-device communication plays an important role in network performance. Data transmission delays and network energy consumption are two main issues of concern, and the basic MAC protocol plays an important role in network performance. This section introduces research conducted in this area.

## MAC protocol

The MAC protocol is vital for device-to-device communication and directly influences network performance through energy consumption, delays, etc. [3, 9, 30]. The protocol has attracted considerable attention due to its importance. Past works related to the research presented this paper is introduced below.

In most networks, the time ratio for devices used in data operations is not high. Therefore, most devices use a duty cycle scheme, which causes devices to periodically alternate between the awake and sleep modes. As devices consume 1/1000 of their energy capacity when asleep, a duty cycle scheme can save considerable amounts of energy [39]. However, while a duty cycle scheme can increase network lifetimes, it can have negative impacts on other areas of network performance. Its influence on delays is the most significant. In a network that does not use a duty cycle, multiple senders must transmit data at the same time, increasing delays. In a

network that uses the duty cycle, when a sender must transmit data, it must wait until the receiver wakes, also increasing delays. The Sensor-MAC (S-MAC) protocol employs a duty cycle based the MAC protocol [51]. Under the S-MAC protocol, while nodes save energy by periodically waking and sleeping, even when there is no data operation, the amount of time spent awake is stationary. Therefore, researchers have proposed the Time out MAC (T-MAC) protocol [52], which keeps time spent awake constant even during times without data operation under the S-MAC protocol. A node listens the channel for a period of time, and when there is no data operation, it goes to sleep to save energy, increasing delays. Under duty cycle-based MAC protocol, the length of a duty cycle influences delays most. Usually, the proportion of awake time is greater and delays are shorter, but energy consumption is higher [52–55].

The above analysis is based on the contention-based MAC protocol, which requires devices to be periodically awake even when there is no data operation. The Time Division Multiple Access (TDMA) protocol [56] determines device work slots in advance. Devices are awake during data operation and go to sleep when data operation is complete, saving energy. However, this MAC protocol must plan work slots for each device beforehand. Therefore, it is not suitable for dynamically changing networks [56].

Under the divided MAC, receiver-initiated MAC and sender-initiated MAC protocols, one receiver and one sender form the proactive side of data transmission [41]. The receiver-initiated and sender-initiated MAC protocols are similar and differ only in the timing of data operation. Under the sender-initiated MAC protocol, when a sender has data to transmit, the sender uses to non-stop polling channel, and the receiver listens to the channel after it wakes; if the receiver finds a sender polling channel, it can communicate with the sender. It is apparent that under this MAC protocol, both the polling frequency of the sender and the time required for the receiver to listen to the channel will affect data transmission delays [41]. When the polling channel frequency of the sender is high, the receiver receives polling signals easier after it wakes, causing it to communicate with the sender earlier and shorten data transmission delays. However, the energy consumed by the sender increases. The amount of time required for the receiver to listen to the channel also affects delays, because when this period is short, the receiver will go to sleep after listening; when the sender has data to transmit in this period and a data request must be sent, it must wait until the receiver wakes, resulting in long delays.

On the other hand, under the receiver-initiated MAC protocol, when a sender has data to transmit, it stays silent and listens to the channel. The receiver broadcasts a

wake up beacon showing that it is ready to receive data transmission requests from senders after wakes. Therefore, after the sender receives the wake up beacon from the receiver, the sender sends a Tx-Beacon requesting data transmission. Thus, under the receiver-initiated MAC protocol, the receiver's wake up-Beacon influences delays most. When a receiver only needs to send a wake up-Beacon once, a sender waking before the receiver sends a wake up-Beacon can receive the wake up-Beacon and thus initiate data communication. Senders waking after the receiver sends a wake up-Beacon and wanting to establish data communication must wait for the next cycle, increasing delays. When a receiver intermittently sends a wake up-Beacon after it wakes, this shortens delays but requires more energy consumption.

#### Priority data transmission

As delay sensitivity varies for different perceived data, when large amounts of data must be transmitted at the same time, prioritizing emergency data transmission is essential for applications. For example, emergency data related to fires, alerts and etc. must be transmitted as soon as possible so that the decision center can address the emergency quickly. Some works have thus focused on priority data transmission. Priority data transmission can be divided into the following types based on different methods used:

- (1) Priority data transmission in the MAC layer.

Priority data transmission based on the MAC layer can transmit high priority data to the channel first in the bottom layer of the network or transmit high priority data after determining priorities, lowering system costs and improving system performance. The contention window is the amount of time required for random backoff when multiple devices are using a channel; devices randomly select a period of time to wait for and listen to the channel, and data transmission starts if the channel is still idle after this period of time [57]. The length of the contention window is critical to the contending channel. Generally speaking, a contention window has following effects on devices using a channel. When a contention window is large, in terms of probability, devices must wait a long time, so it a device selects a shorter contention window, shortening the wait time, the probability of winning the competition for devices selecting a shorter contention window is higher, increasing the likelihood of data from these devices being transmitted first [57]. Obviously, devices use smaller contention windows for high priority data and larger ones for low priority data. Therefore, high priority data are likely to be successful in the

contending channel, ensuring that high priority data are sent first. Setting a smaller contention window not only ensures that high priority data are sent first, but also reduces delays because the contending channel uses less time, reducing energy consumption and increasing network lifetimes. However, when the size of the contention window is reduced, there will be more collisions channels compete, degrading system performance. Priority data transmission that involves setting the size of the contention window is most advantageous due to its low costs. With this method, the MAC protocol need not pay extra costs for priority data transmission. It only needs to change the size of the contention window. However, for this method, data priorities are undetermined, meaning that it is impossible to determine which data are transmitted first when there are multiple senders. This the case because even when the contention window is small, the wait time randomly selected by devices with a small contention window may still be shorter than that chosen by devices with a large contention window, and thus devices with lower data priority access the channel earlier. Therefore, this priority data transmission method is effective, but it cannot guarantee that high priority data are transmitted first.

S. Sarang et al. [41] proposed another reliable strategy for priority data transmission where high priority data are transmitted first. In a receiver-initiated network, after a receiver wakes, it broadcasts a wake up-Beacon which indicates that a sender with data to transmit can send a request to transmit data. A sender sends a Tx-Beacon to request data transmission, and priority data it wants to transmit is contained in the Tx-Beacon. The receiver returns the Rx-Beacon to select the sender with the highest priority data after it receives Tx-Beacons from all senders, and thus devices with high priority data can transmit data first. This method guarantees that high priority data are transmitted first. However, the system's costs are high, as a period of time is set for each Tx-Beacon.

- (2) Priority data transmission in the network routing layer. Priority data transmission based on a network routing layer is another commonly used and diverse method. The priority queue method is frequently used for priority data transmission and is based on the network routing layer. The method sets multiple queues by data priority, and senders choose data to send by polling when they can send. Polling starts with the highest priority queue, and if there are high priority data to transmit, these data

are selected and sent; otherwise, the lower priority queue is queried. Using this method can guarantee that high priority data are transmitted first. However, while the sender can ensure that its high priority data are transmitted first, it cannot guarantee that its data will be sent. When a sender cannot send data, its high priority data cannot be sent, limiting this method.

In the routing layer, data transmission delays not only affect delays in each hop of data transmission, but also the number of hops required for routing. If the number of hops needed for data transmission routing is large, delays will be long as well. Thus, a routing path with few hops should be used for emergency data. Interference emergency data routing should be avoided by choosing the path with more routing hops for data that are not urgent. To select different routing paths based on data priority, Zhang et al. [58] proposed the integrity and delay differentiated routing (IDDR) strategy. Under the IDDR scheme, delay-sensitive data choose the shortest routing path for data transmission to guarantee delay minimization. Longer routing paths for data that are delay-insensitive but with high integrity requirements are selected to prevent the shortest routing path from being chosen by emergency data. Thus, integrity is guaranteed, and network performance and data transmission with different requirements can be ensured by selecting different routing paths.

**Table 1** Symbols used in this paper and their definitions

Symbol	Definition
$V$	All ordinary nodes in the network
$W$	All nodes in the network
$P_{ack}$	All data generated in the network
$p_{ac}^i$	$i$ -th data generated in the network
$P_{hi}$	All data of the highest priority generated in the network
$N_{hi}$	Total volume of data of the highest priority in the network
$T_g^i$	The time when $i$ -th data are generated in the network
$T_e^i$	The time when $i$ -th data generated in the network are successfully transmitted to the sink node
$T_t^i$	The packet delay of $i$ -th data generated in the network
$T_{to}$	The total delay of the network
$T_{ave}$	The average delay of highest priority packets in the network
$T$	The time required for a cycle with the exception of the wait time
$T_w$	The wait time
$T_{tx}$	The amount of time required to transmit a Tx-Beacon
$T_{cur}$	The amount of time required for the network to initiate a current

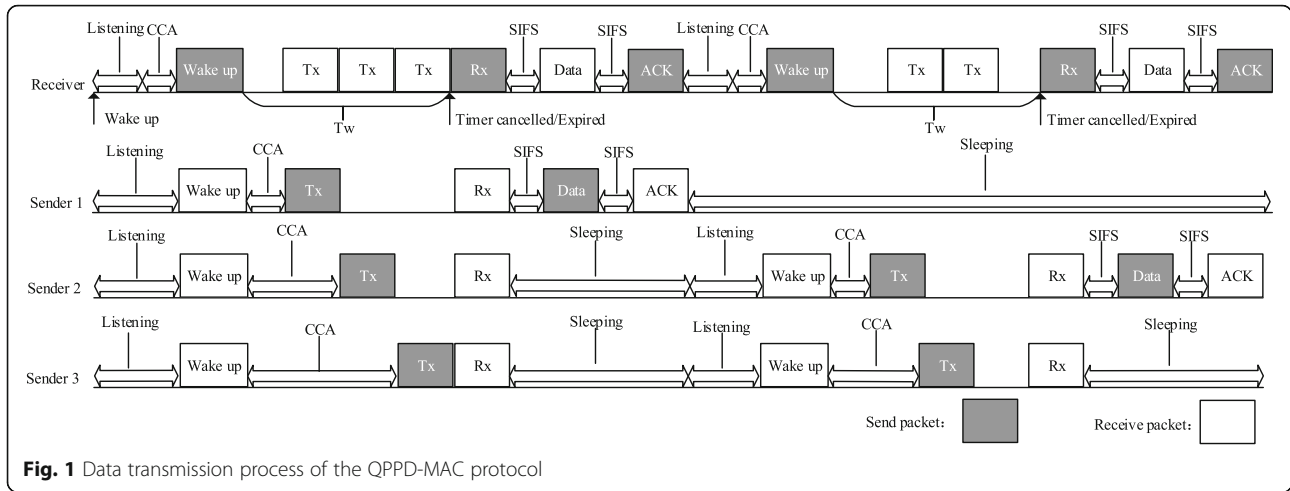


Fig. 1 Data transmission process of the QPPD-MAC protocol

### System model and problem statement

#### The network model

The network model used in this paper is similar to that used in Ref. [41] as is described as follows: a wireless sensor network contains a *sink* node and multiple ordinary nodes with the same characteristic  $V = \{v_1, v_2, \dots, v_{(n-1)}, v_n\}$ ; for ease of expression, the network described in this paper is abstracted to a plane tree network, and each ordinary node in the tree network has sole parent nodes but can have multiple child nodes. A tree network with  $n$  ordinary nodes can be expressed as  $W = \{sink, v_1, v_2, \dots, v_{(n-1)}, v_n\}$ , and the *sink* node is expressed as the root node in the tree network. Each node in the sensor network can sense its surrounding environment and can send sensing data to the node in the next hop. The next hop node makes fewer hops to the *sink* node, and the node sends data to the *sink* node by multi-hop routing. Ordinary nodes in the network are powered by batteries of the same limited capacity, and they can harvest energy from the surrounding environment; the energy of the *sink* node is infinite.

Data priority is divided to four levels:  $P_4$  (emergency),  $P_3$  (most important),  $P_2$  (important), and  $P_1$  (not important).  $P_4$  denotes the highest level of data

priority. Data in the network are expressed as  $P_{ack}$ , and data of the highest priority in the network are expressed as  $P_{hi}$ .  $P_{hi} \subseteq P_{ack}$ .

Some symbols used in this paper and their definitions are given in Table 1.

#### The problem statements

##### Definition 1

Packet delay. The time required for the data is successfully transmitted to the *sink* node. If a data  $P_{ac}^i$  is generated at time  $T_g^i$ , it is successfully transmitted to the *sink* node at time  $T_e^i$ . The packet delay  $T_t^i$  of these data can be expressed as

$$T_t^i = T_e^i - T_g^i \tag{1}$$

##### Definition 2

Total delay. The sum of the packet delay of multiple data. If there are  $x$  data in the network, the total delay  $T_{to}$  of these  $x$  data can be expressed as

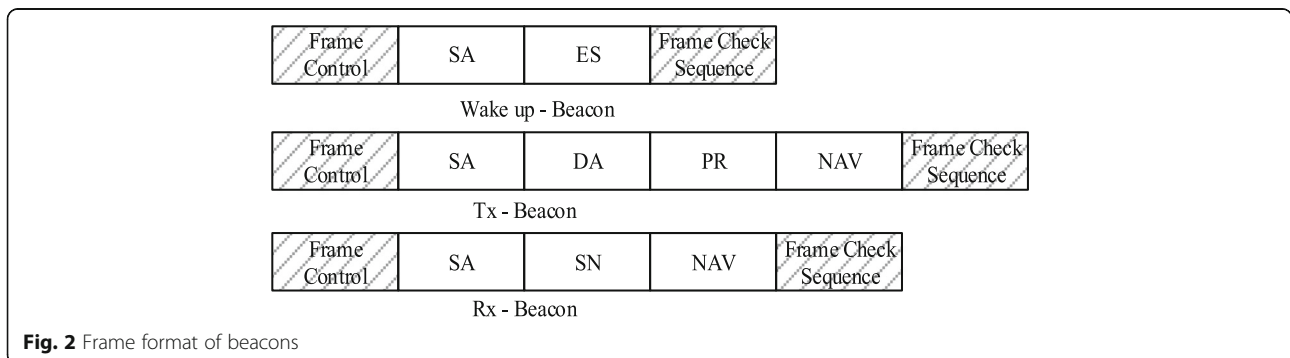


Fig. 2 Frame format of beacons

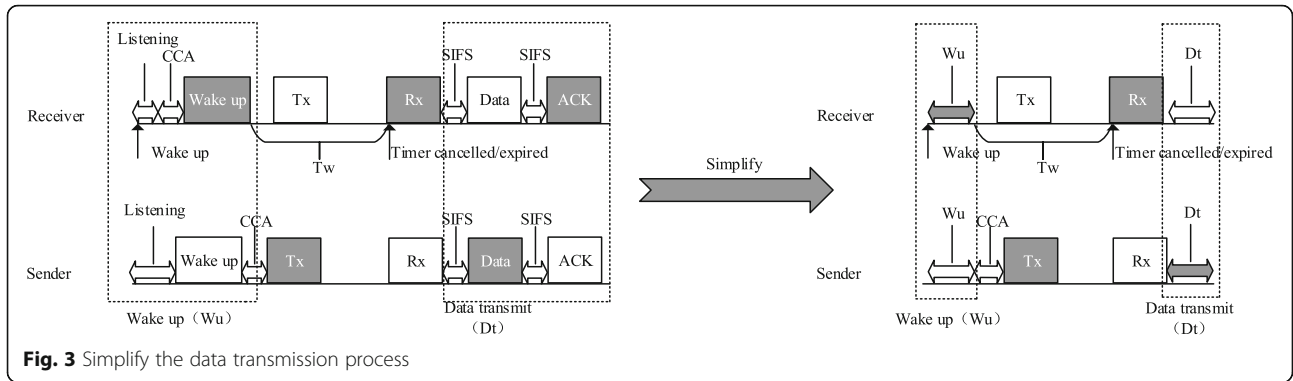


Fig. 3 Simplify the data transmission process

$$T_{to} = \sum_{i=1}^x T_t^i \tag{2}$$

In the network, the total delay should be as low as possible to ensure the data can be transmitted in time and processed quickly. Thus, the first focus of this paper can be expressed as

$$\begin{cases} \min(T_{to}) = \min(\sum T_t^i) \mid i \in P_{ack} \\ s.t. T_t^i \neq 0 \mid i \in P_{ack} \end{cases} \tag{3}$$

**Definition 3**

Average highest priority packet delay. The average packet delay for multiple data of the highest priority. If there are  $x$  data of the highest priority in the network, the average delay of highest priority packets  $T_{ave}$  of these  $x$  data can be expressed as

$$T_{ave} = (\sum_{i=1}^x T_t^i) / x \tag{4}$$

In the network, the average delay of highest priority packets should be as low as possible to ensure that the highest priority data can be transmitted as soon as possible and so that emergencies can be addressed earlier. Thus, the second goal of this paper can be expressed as

$$\begin{cases} \min(T_{ave}) = \min(\frac{\sum T_t^i}{N_{hi}}) = \min(\sum \frac{T_e^i - T_g^i}{N_{hi}}) \mid i \in P_{hi} \\ s.t. T_t^i \neq 0 \mid i \in P_{hi}, N_{hi} \neq 0, P_{hi} \neq \emptyset \end{cases} \tag{5}$$

Overall, we seek to minimize the total delay,  $D$ , and the average delay of highest priority packets,  $A$ , which can be summarized as follows:

$$\begin{cases} \min(D) = \min(\sum_{i \in P_{ack}} (T_e^i - T_g^i)) \\ \min(A) = \min(\sum_{i \in P_{hi}} \frac{T_e^i - T_g^i}{N_{hi}}) \end{cases} \tag{6}$$

**The design of the DWT-MAC protocol**

In improving the previous receiver-initiated MAC protocol, we first introduce the QoS MAC protocol for Prioritized Data (QPPD-MAC) below and improve it. After discussing QPPD-MAC protocol communication, our proposed protocol is given.

**Background information**

Communication under MAC protocols for prioritized data such as the QPPD-MAC is illustrated in Fig. 1. The network shown in Fig. 1 is a many-to-one data transmission network with three sender nodes that want to transmit data to the same receiver. Nodes in the network

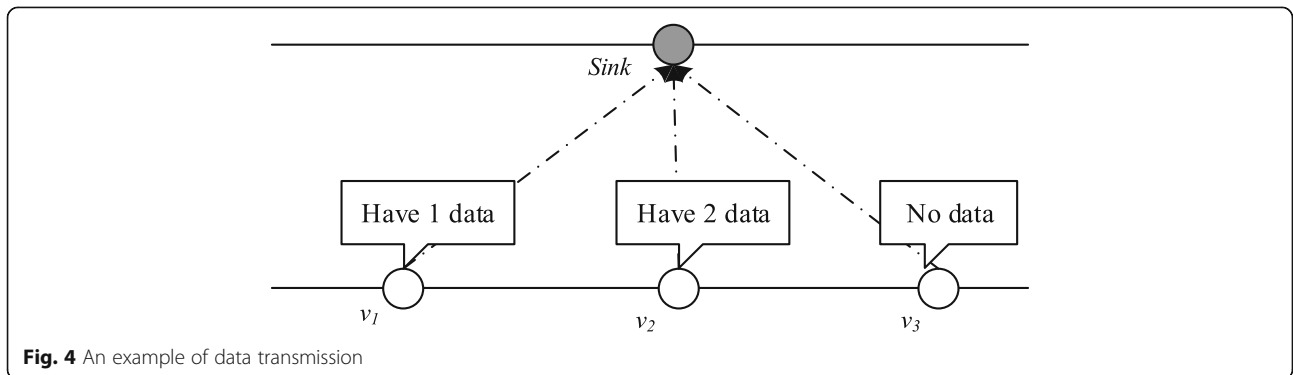


Fig. 4 An example of data transmission

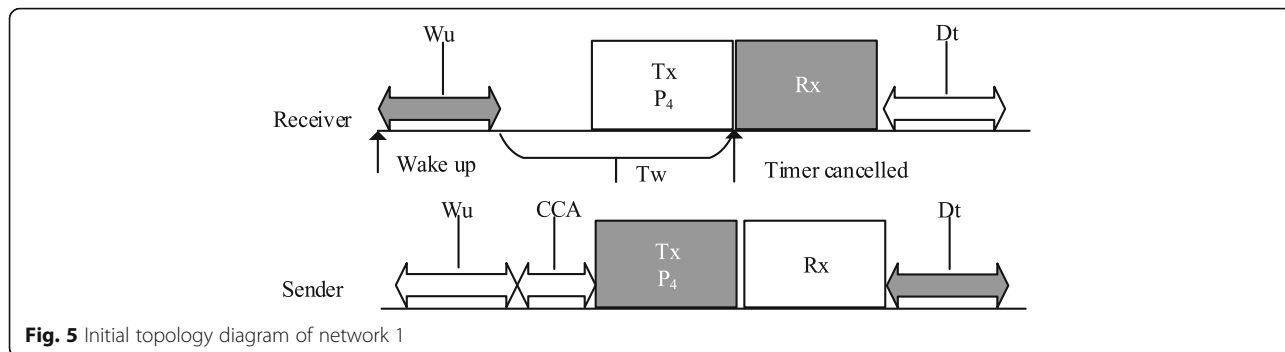


Fig. 5 Initial topology diagram of network 1

with data to transmit wake and listen to the channel. The receiver performs clear-channel assessment (CCA) once it wakes to determine the status of the medium; if the medium is idle, it broadcasts a wake up-Beacon indicating that it is ready to receive data. The wake up-Beacon contains a Source Address (SA) and Energy State (ES) as shown as Fig. 2. The receiver then initiates a wait time of duration  $T_w$  to receive Tx-Beacons from the sender nodes, and the Tx-Beacon contains an SA, Destination-Address (DA), Priority (PR), and Network-Allocation Vector (NAV) (see Fig. 2). The duration of wait time  $T_w$  is proportional to the number of sender nodes  $\mathcal{N}$ , and  $T_w$  not only contains the time length for CCA but also the time length for the  $\mathcal{N}$  Tx-Beacon. After sender nodes receive the wake up-Beacon from the receiver, they send Tx-Beacons to the receiver after they successfully compete for the channel. The receiver broadcasts the Rx-Beacon with the address of the selected sender when one of the following two conditions is met. The Rx-Beacon contains an SA, Selected Node (SN), and NAV (see Fig. 2). The two conditions are as follows:

- (1) the timer expires and the wait time reaches the predetermined value  $T_w$ ;
- (2) Timer cancelled, receiver received the Tx-Beacon with the highest data priority. Sender nodes compare the SN contained in Rx-Beacon to itself address after successfully received Rx-Beacon, if the SN is same as itself address, this indicates that this sender node is selected for data transmission, and this sender node transmits data packet after a Short Interframe Space (SIFS), finally receiver reply a ACK packet to the selected sender after successfully received data packet. The above process completes the data transmission of a data. Know from the process, when there are multiple sender nodes which have data need to be transmitted, it can guarantee the data which obtain the chance for data transmission must be the data with the relative highest priority among all sender nodes, and finally this data was transmitted to receiver. The subsequent data transmission process is same as above

process. The different just is there is no data to be transmitted in sender 1, so sender 1 does not send Tx-Beacon no longer, only senders which have data need to be transmitted still send Tx-Beacon. But receiver did not know the sender 1 has no data to transmit,  $T_w$  remains unchanged. The transmission process is shown in Fig. 1. The frame formats of beacons in the transmission process are shown as Fig. 2.

**Research motivation**

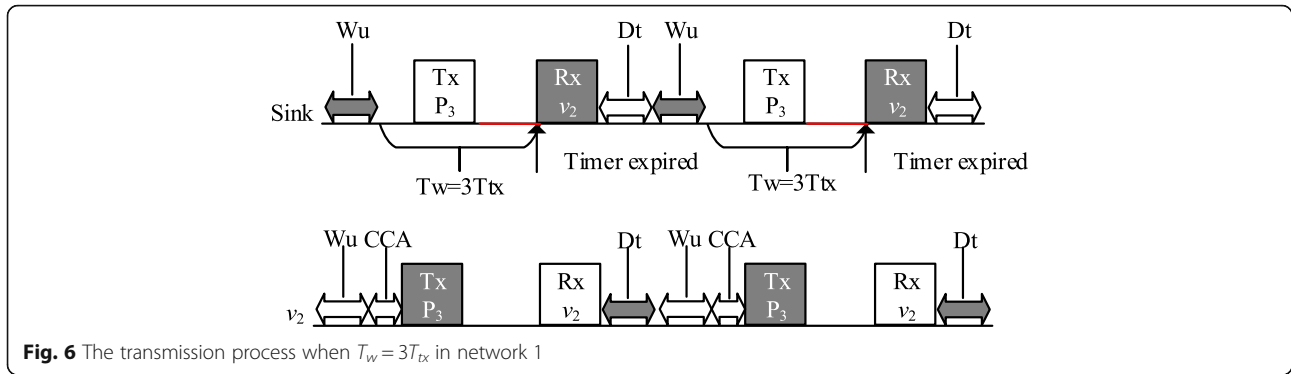
From the MAC protocol for prioritized data, to transmit high priority data first, it takes a long period of  $T_w$  to determine what data to receive from which sender. A time length of  $T_w$  can ensure that each sender sends a Tx-Beacon for data transmission, ensuring that the receiver selects the sender with the highest priority data among  $\mathcal{N}$  senders. This method guarantees that high priority data are transmitted first, but it has the following limitations:

- (1) Long data transmission delays. The previous MAC protocol based on data priority is similar in that time length  $T_w$  is equal to the number of sender nodes when there are multiple sender nodes. This requires data to take a long time to establish communication before data transmission, increasing delays.
- (2) Ineffective energy consumption. As mentioned above, because a receiver needs set a period of time for each sender node, it waits for sender nodes sends a Tx-Beacon in this period of time. However, while there may be multiple sender nodes in a network, most of the time, only some sender nodes have data to transmit, resulting in long periods of empty

Table 2 The priority levels of data of each node in network 1

Node	Data priority
$v_1$	$P_4$
$v_2$	$P_3, P_2$





**Fig. 6** The transmission process when  $T_w = 3T_{tx}$  in network 1

consumption, ineffective energy consumption, and long delays.

We use a concrete example to analyze the impact of  $T_w$  on network performance. For ease of analysis, for networks described in section 4, assume that no new data are generated after data transmission in each cycle. The index of node  $v_c$  in the network (i.e.,  $c$ ) is larger, and this node can compete for the channel faster.

The MAC protocol proposed in this paper focuses on the time used for high priority data, and the transmission process is the same after communication is established. Therefore, to make the transmission process shorter and clearer, it is simplified as shown in Fig. 3. Transmission from the receiver broadcasts a wake up-Beacon until the receiver initiates the wait time through a simplified Wake up (Wu) process. The transmission process whereby the receiver broadcasts an Rx-Beacon until data are successfully transmitted to the receiver is simplified as the Data transmit (Dt) process.

To accurately compare the optimization of  $T_w$  under different MAC protocols, we do not use time units to represent delays and instead use the number of cycles and the required number of Tx-Beacons to represent delays, which is more accurate. A delay in this paper is defined as time used for multiple cycles. The basic transmission process is generally the same for each data transmission process. A node wakes, a receiver sends a wake up-Beacon, the receiver sends a Rx-Beacon, the sender node sends a data packet and the receiver returns a ACK packet. The time required for each cycle is the same. The cycles differ in transmission processes required for receivers to initiate wait times and receive Tx-Beacons. The wait time stops when the time expires or is cancelled, and the timer is stopped depending on how many Tx-Beacons are transmitted, resulting in different periods of data transmission between cycles. To facilitate comparisons, the time required for

the current cycle can be expressed as the total number of cycles passed and the total number of Tx-beacons transmitted. The current time  $T_{cur}$  can be expressed as:

$$T_{cur} = yT + zT_{tx}$$

$y$  indicates the total number of cycles passed until the current cycle is reached, and  $z$  denotes the total number of Tx-Beacons received by the receiver until the current cycle is reached. We provide an example to explain our delay calculations. For the data transmission process shown in Fig. 4, using the above delay calculation method, for data generated at time  $0T + 0T_{tx}$ , a sender first sends a Tx-Beacon for the highest priority data. Thus, the receiver's wait time is cancelled early. When the receiver successfully receives data of the highest priority, the network passes a cycle, and the receiver receives 1 Tx-Beacon. The total delay of this transmission process is expressed as  $1T + 1T_{tx}$ . The packet delay of the data of the highest priority is  $1T + 1T_{tx}$ , as only one set of data of the highest priority is successfully transmitted. The average delay of highest priority packets is expressed as  $1T + 1T_{tx}$ .

In the following, we analyze limitations of the previous MAC protocol and describe the DWT-MAC protocol proposed here. We use network 1 as an example to illustrate limitations of the previous MAC protocol. An initial topology diagram of network 1 is shown in Fig. 5, and the priority levels of data from each node in network 1 are listed in Table 2. Delays of the three cases experienced when the length of  $T_w$

**Table 3** Symbols used in Algorithm 1 and their definitions

Symbol	Definition
$Cycle\_num$	The current cycle number
$if\_timercan()$	This function is used to determine when the wait time is stopped by the timer or expires
$thiscycleTx()$	This function is used to calculate the total number of Tx-Beacons received in the current cycle

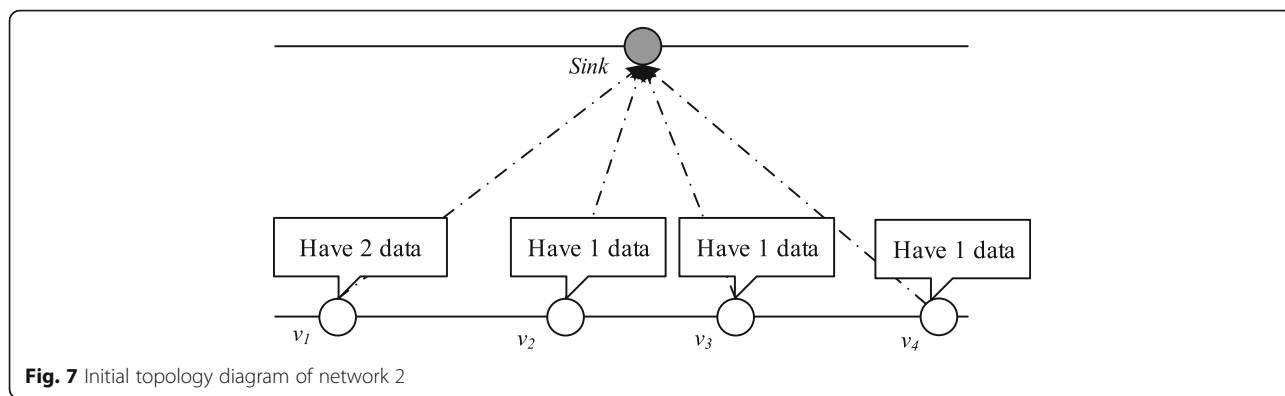


Fig. 7 Initial topology diagram of network 2

is fixed are analyzed to illustrate limitations of the previous MAC protocol.

- (1) The length of  $T_w$  is equal to the number of sender nodes. In network 1 shown in Fig. 5, when  $T_w = 3T_{tx}$ , during the transmission of data from  $v_1$  and  $v_2$  to the sink node,  $v_1$  successfully transmits its data of the highest priority to the sink node in the first cycle, and only  $v_2$  still has data to transmit in the next 2 cycles. Thus, just one sender node  $v_2$  will wake and transmit data in the second and third cycles. As shown in Fig. 6, in these 2 cycles,  $T_w$  must be long enough to transmit one Tx-Beacon, but  $T_w$  must be long enough to transmit three Tx-Beacons, as no high priority data must be transmitted. The wait time is thus not cancelled early, and the receiver must wait for the wait time of duration  $T_w$  to expire, thus wasting time while waiting to receive a Tx-Beacon (see the red lines in Fig. 6). In this period, the sink node and  $v_2$  are waiting. This increases the average and total delays of highest priority packets, network energy consumption increases and the network lifetime is shortened.
- (2) The length of  $T_w$  is lower than the number of sender nodes, and assuming  $T_w = 1T_{tx}$ , it loses the feature of priority data transmission. While network data are still prioritized, the receiver only can receive one Tx-Beacon from the sender node at a time; in fact, the receiver starts data transmission after the first Tx-Beacon is received in each cycle. The transmission process at this time is more consistent with the transmission process for unprioritized data than the common protocol, and the QPPD-MAC protocol must transmit a Tx-Beacon, Rx-Beacon and others. At this point, the QPPD-MAC cannot ensure that high priority data are transmitted first, and thus it cannot reduce the average delay of highest priority packets. Instead, the

average and total delays of the highest priority packets increase, and network energy consumption increases as well.

- (3) Finally, when  $T_w = 2T_{tx}$  (i.e., the length of  $T_w$  is equal to the number of sender nodes), multiple nodes wake up at the same time,  $v_1$  completes data transmission in the first cycle, and after this cycle, when  $v_2$  transmits data in the next 2 cycles, some time is wasted while waiting to receive a Tx-Beacon. While less time is wasted than when  $T_w = 3T_{tx}$ , invalid waiting time is unavoidable.

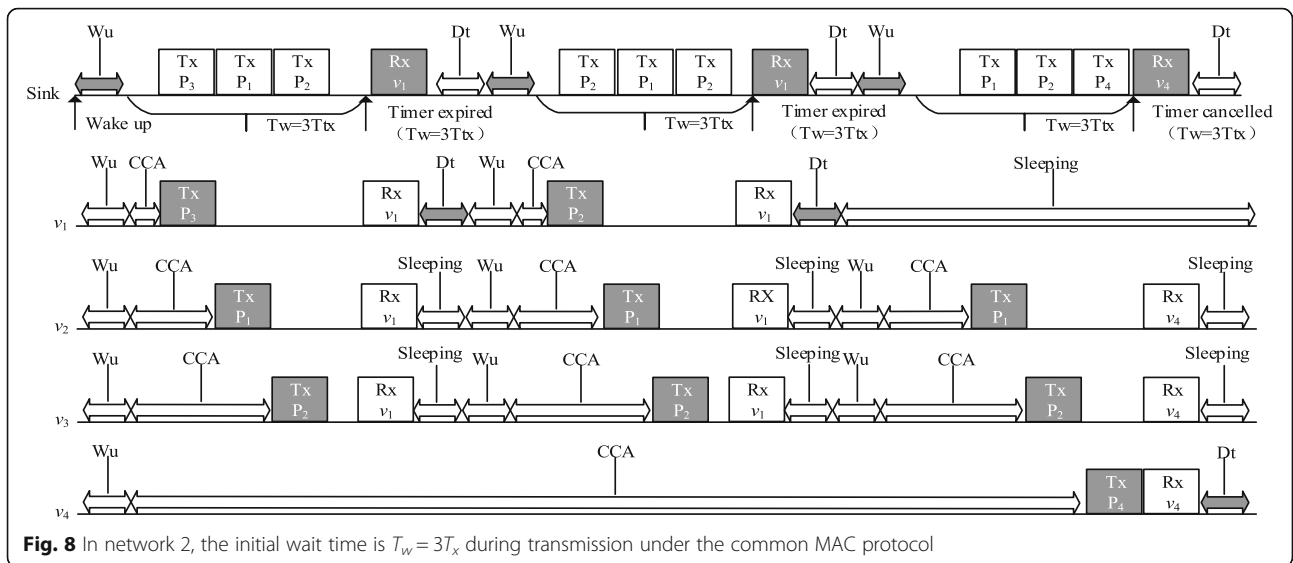
Our analysis of the above three cases shows that whatever the length of  $T_w$ , it will influence delays and energy consumption. If  $T_w$  is too long, high-priority data are transmitted first, but delays increase significantly. When  $T_w$  is too short, delays decrease, but the transmission of high priority data first cannot be guaranteed.

Thus, is it possible to develop a method that transmits high priority data as early as possible while limiting delays?

Under the previous MAC protocol, the wait time is fixed, and its length  $T_w$  does not change with the state of the network, resulting in high network energy consumption and long delays. When there are  $k$  sender nodes with data to transmit, if the value of  $T_w$  can be adjusted to  $T_w = kT_{tx}$  before the timer starts, this value can ensure that all sender nodes participate in transmission and that sender nodes with the highest priority data send Tx-beacons in this transmission cycle. This also ensures that  $T_w$  is not too large, shortening invalid wait

Table 4 The prioritization of data of each node in network 2

Node	Data priority
$v_1$	$P_3, P_2$
$v_2$	$P_1$
$v_3$	$P_2$
$v_4$	$P_2$



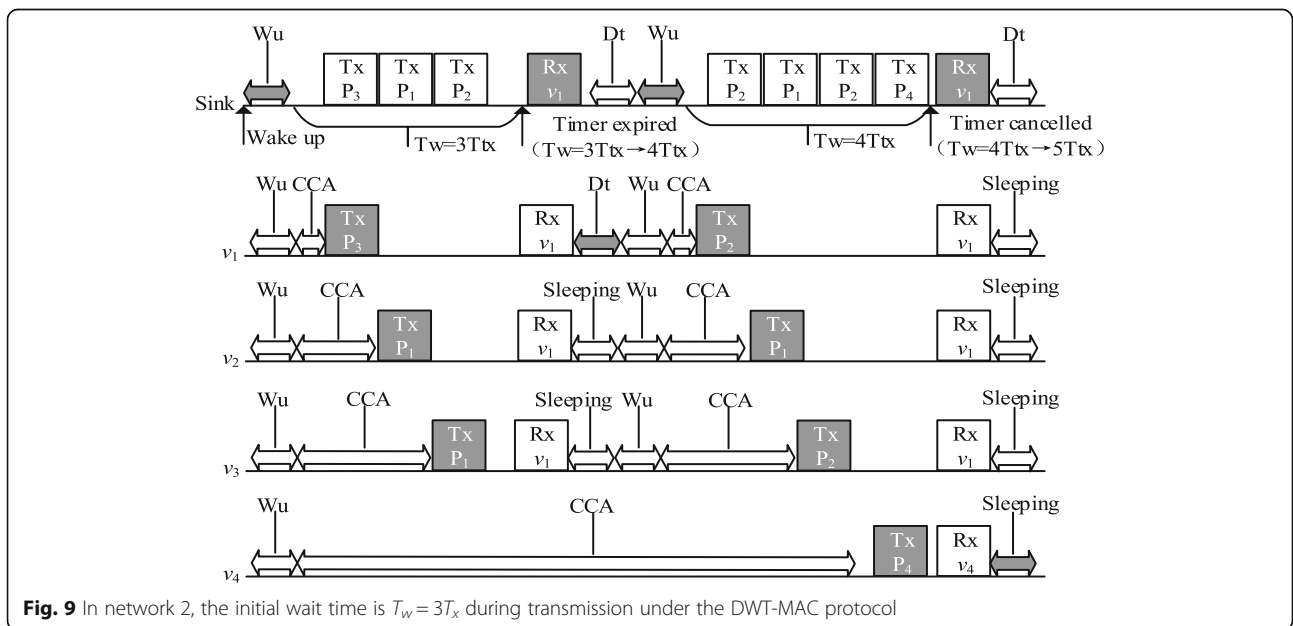
times and solving the above problem. If the value of  $T_w$  in each cycle can be adjusted to the optimal value  $kT_{tx}$ , the desired goal can be met. However, because Tx-Beacons are transmitted after the wait time is initiated, the value of  $T_w$  cannot be adjusted according to the number of Tx-Beacons received in the current cycle and can only be adjusted by estimating how many Tx-Beacons may be received in the current cycle according to the number of Tx-Beacons received in the previous cycle.

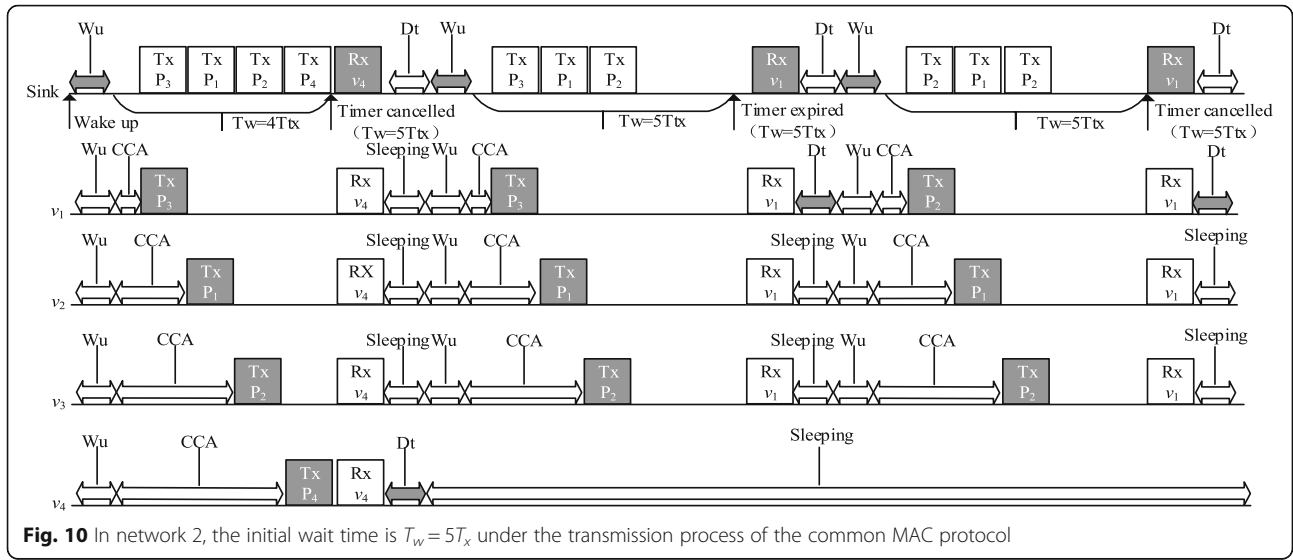
To our knowledge, no MAC protocol can dynamically adjust wait times according to the number of Tx-Beacons successfully received. Therefore, there is a need for an MAC protocol that can reduce delays by

dynamically adjusting wait times according to the number of Tx-Beacons successfully received. A novel dynamic timer adjustment algorithm is proposed in section 4.3, and an MAC protocol is proposed based on this algorithm.

**DWT-MAC protocol**

As the wait time is fixed throughout transmission in the previous QoS MAC protocol, a novel Dynamic Wait time based MAC (DWT-MAC) protocol is proposed in this section based on the ideas analyzed in section 4.2. Under the DWT-MAC protocol, the wait time is adjusted according to the following rules after the wait period stops in each cycle to bring the value





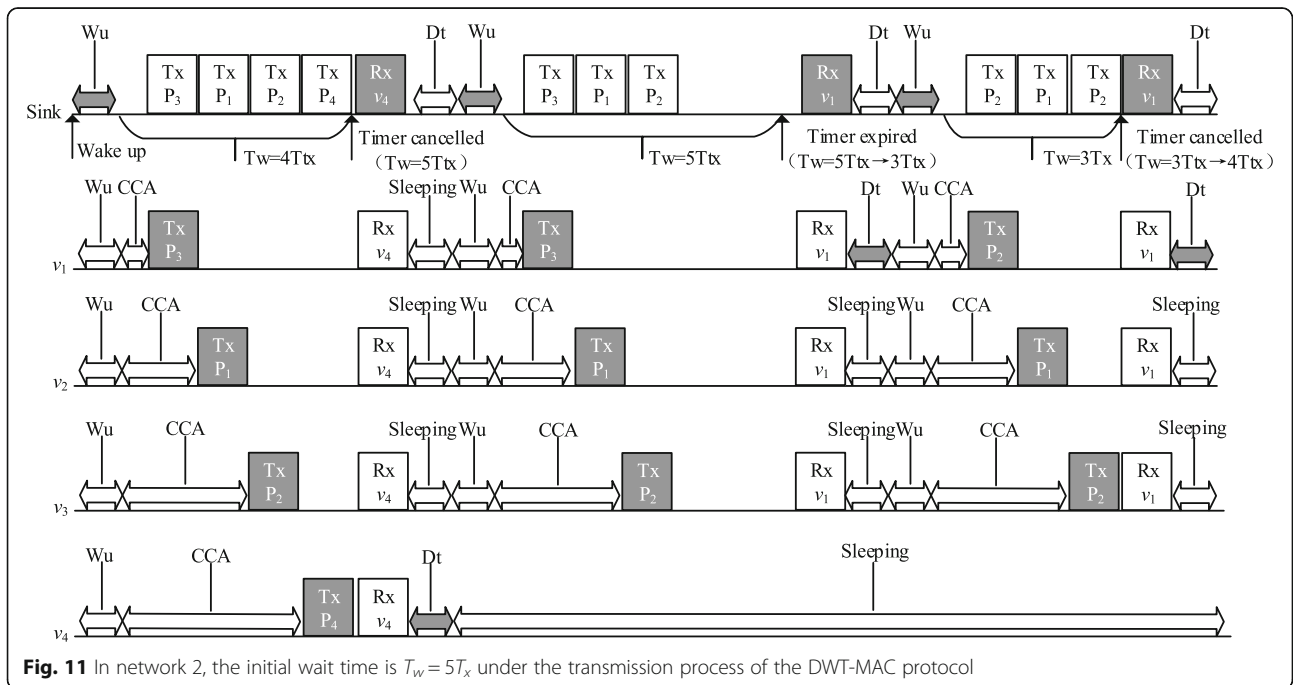
of  $T_w$  as close to the optimal value as possible before the next cycle:

- (1) The initiated value of  $T_w$  must be set before transmission starts,  $T_w = nT_{tx}$ .
- (2) If the wait time is stopped by the timer in a cycle, do not change the value of  $T_w$  in this cycle, and the wait time stays at its original time length.
- (3) If the wait time expires in a cycle, the current wait time is  $T_w = iT_{tx}$  and thus count the total number of Tx-beacons successfully transmitted in this cycle as  $j$ .

If  $j < i$ , adjust the value of  $T_w$  to  $T_w = jT_{tx}$ , and if  $j = i$ , adjust the value of  $T_w$  to  $T_w = (j + 1)T_{tx}$ .

- (4) If an exception occurs during transmission and data transmission is not completed, do not change the value of  $T_w$  in this cycle, and thus keep the wait time at its original length.

A dynamic timer adjustment algorithm is designed based on the above rules, and the algorithm is written as Algorithm 1. Symbols used in Algorithm 1 and their definitions are given in Table 3.



Next, in comparing the data transmission processes of the DWT-MAC and previous MAC protocols, how the protocol proposed in this paper reduces the average delay of highest priority packets and total delays by dynamically adjusting the wait time is analyzed. In the following, the transmission process of the other MAC protocol is called the common MAC protocol.

Algorithm 1: **Dynamic timer adjustment algorithm**

```

1: BEGIN
2:  $Cycle_{num} = 1$ 
3:  $T_w = nT_{tx}$ 
4: FOR each cycle DO
5: IF  $Cycle_{num} = 1$  DO
6:  $i = n$ 
7: END IF
8: IF  $if\_timercan() = 0$  DO
9:  $j = thiscycleTx()$ 
10: IF  $j < i$  DO
11:  $T_w = jT_{tx}$ 
12:  $i = j$ 
13: ELSE IF  $j = i$  DO
14:  $T_w = (j + 1)T_{tx}$ 
15:  $i = j + 1$ 
16: END IF
17: END IF
18:  $++ Cycle\_num$ 
19: END FOR
20: END

```

The network for comparing network 2 (the initial topology diagram is shown as Fig. 7) and the priority of data in each node in network 2 is listed in Table 4.

A case in which the initial wait time is short is analyzed first, setting the initial value of  $T_w$  as  $T_w = 3T_{tx}$ . The transmission process of this initial wait time is shown in Fig. 8.

From the transmission process of the MAC protocol, in the first 2 cycles,  $v_4$  cannot transmit data of the highest priority. As  $T_w$  cannot change under the common MAC protocol until third cycle  $v_1$  has successfully transmitted all data it contains to the *sink* node,  $v_4$  can only transmit its Tx-Beacon of the highest priority data to the receiver and participate in data transmission. Sender

nodes in the network continuously generate new data, preventing  $v_4$  from transmitting a Tx-Beacon to the receiver and leading to a very long delay for these high-priority data. Sometimes, such data may not be transmitted at all. The packet delay of these data is  $3T + 9T_{tx}$ , as there is just one high priority dataset and the average delay of highest priority packets is thus  $3T + 9T_{tx}$ .

For the same network (i.e., in network 2, the initial value of  $T_w$  is  $T_w = 3T_{tx}$ ), the transmission process that occurs until the highest priority data are transmitted to the *sink* node is shown in Fig. 9.

Under the transmission process of the DWT-MAC protocol,  $T_w$  is dynamically adjusted in each cycle according to the dynamic timer adjustment algorithm. The wait time is stopped once the timer completes the first cycle, and the number of Tx-Beacons received by the *sink* node within  $T_w$  is  $i = 3$ . The wait time was  $T_w = 3T_{tx}$ , so  $j = 3$  and  $i = j = 3$  at this time, and thus the value of  $T_w$  is adjusted to  $T_w = 4T_{tx}$  after the wait time ends in the first cycle. Thus, all four sender nodes can transmit their Tx-Beacons in the second cycle, and  $v_4$  is transmitted the highest priority data in the second cycle, and the highest priority data are successfully transmitted to the *sink* node early. The packet delay of this highest priority data and of high priority packets is  $2T + 7T_{tx}$  at this point.

Relative to the common MAC protocol, the DWT-MAC protocol decreases the average delay of high priority packets as follows

$$3T + 9T_{tx} - 2T - 7T_{tx} = 1T + 2T_{tx}$$

In network 2, when the initial wait time is long, the common MAC protocol generates longer wait times, causing sender and receiver nodes to waste time waiting. How the DWT-MAC protocol reduces delays by dynamically adjusting the wait times is analyzed next. We set the initial value of  $T_w$  as  $T_w = 5T_{tx}$ , and the first 3 cycles of the common MAC protocol are shown in Fig. 10.

Under the common MAC protocol, in the second and third cycles, after the *sink* node wakes, as the timer sets a duration of  $T_w = 5T_{tx}$  while only three sender nodes must transmit data, time is wasted after the third Tx-Beacon is transmitted, and *sink* and sender nodes must wait for the wait time to end, increasing packet delays and wasting energy. The total delay of the first cycles is  $3T + 14T_{tx}$ .

For the same network state (i.e., in network 2, the initial value of  $T_w$  is  $T_w = 5T_{tx}$ ) and DWT-MAC protocol, the transmission process is shown in Fig. 11.

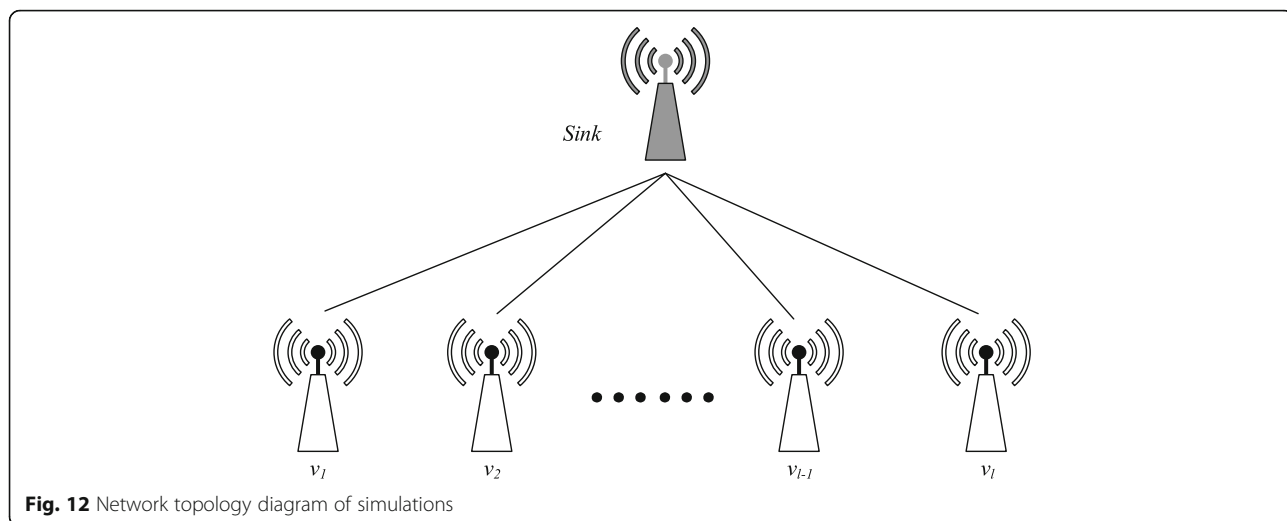


Fig. 12 Network topology diagram of simulations

The first 2 cycles of the DWT-MAC protocol are the same as the common MAC protocol, but in the second cycle, the wait time stops once the timer ends, the number of Tx-Beacons received by the *sink* node is  $j = 3$ , and the wait time is  $T_w = 5T_{tx}$ . Thus,  $i = 5$  and  $j < i$ , so  $T_w$  is adjusted to  $T_w = 3T_{tx}$  after the wait time ends in the second cycle. Thus, in the third cycle, when there are three sender nodes in the network, the wait time is just long enough to receive the Tx-Beacons transmitted by these three sender nodes, and no time is wasted waiting. The total delay of the first 3 cycles is  $3T + 12T_{tx}$ . In this network state, relative to the common MAC protocol, the DWT-MAC protocol reduces total delays by

$$3T + 14T_{tx} - 3T - 12T_{tx} = 2T_{tx}$$

From above comparisons, the DWT-MAC protocol can reduce the average delay of high priority packets and total delays by dynamically adjusting the wait time.

### Experimental results and analysis

In this section, we compare the DWT-MAC protocol to the QPPD-MAC protocol based on a series of simulations. In the simulations, the transmission failure rate is set to 1/10000, and we calculate the total delay and the average delay of highest priority packets. The DWT-MAC and QPPD-MAC protocols mainly differ based on whether wait times can be dynamically adjusted in each cycle. The performance of the DWT-MAC protocol is evaluated through these simulations.

The network topology of our simulations has two levels: the *sink* node and ordinary nodes. When the number of ordinary nodes in network is  $l$  (i.e.,  $W = \{sink, v_1, v_2, \dots, v_{l-1}, v_l\}$ ), the topology of this network is as shown in Fig. 12.

Our simulation proceeds as follows:

We first determine if the current cycle must generate new data from the minimum value of total data ( $M_n$ ). If the current volume of data ( $C_n$ ) is equal to or greater than  $M_n$ , no new data need be generated, and otherwise new data must be generated one by one until  $C_n$  is equal to  $M_n$ . Which ordinary node generates data and how

Table 5 Symbols used in simulations and their definitions

Symbol	Definition
$M_n$	The minimum total volume of data used in simulations
$C_n$	The current total volume of data used in simulations
QPPD AD( $T$ )	Under the QPPD-MAC protocol, the coefficient of $T$ of the average delay of highest priority packets
QPPD AD( $T_{tx}$ )	Under the QPPD-MAC protocol, the coefficient of $T_{tx}$ of the average delay of highest priority packets
DWT AD( $T$ )	Under the DWT-MAC protocol, the coefficient of $T$ of the average delay of highest priority packets
DWT AD( $T_{tx}$ )	Under the DWT-MAC protocol, the coefficient of $T_{tx}$ of the average delay of highest priority packets
QPPD TD( $T_w$ )	Under the QPPD-MAC protocol, the coefficient of $T_{tx}$ of the total delay
DWT TD( $T_w$ )	Under the DWT-MAC protocol, the coefficient of $T_{tx}$ of the total delay

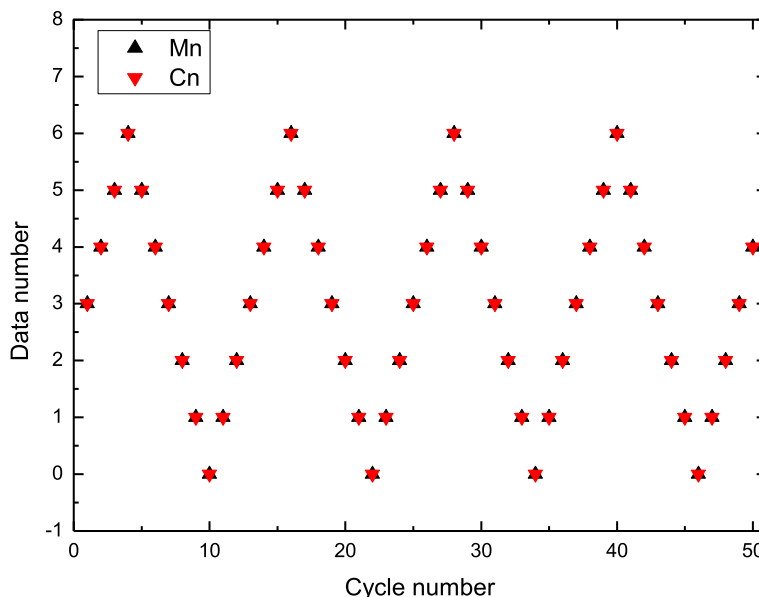


Fig. 13 The total number of periodic data changes increases from 0 to 6; Mn and Cn in the first 50 cycles

the data are prioritized as randomly determined. Then, each sender node generates a random value to compete for the medium. Sender nodes transmit Tx-Beacons in descending order of these random values until the wait time is stopped by timer expiry or cancelling. After transmitting Tx-Beacons, the sender node with the highest priority data transmits data to the sink node.

Our simulations compare the DWT-MAC protocol to the QPPD-MAC protocol based on different

volumes of data in each cycle, different numbers of ordinary nodes in each network and different changes in the total volume of data. The period is set to  $T_w = 3T_{tx}$  in each simulation. The simulations calculate the total and average delays of high priority packets for the first 5000 cycles. Because the number of cycles passed under the DWT-MAC protocol is always the same as the number of cycles passed under the QPPD-MAC protocol, when comparing total delays, only wait times are compared.

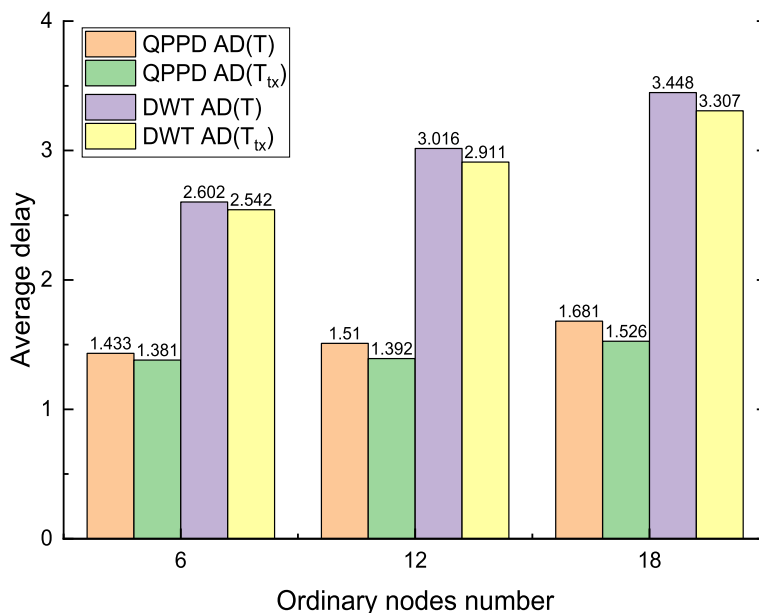
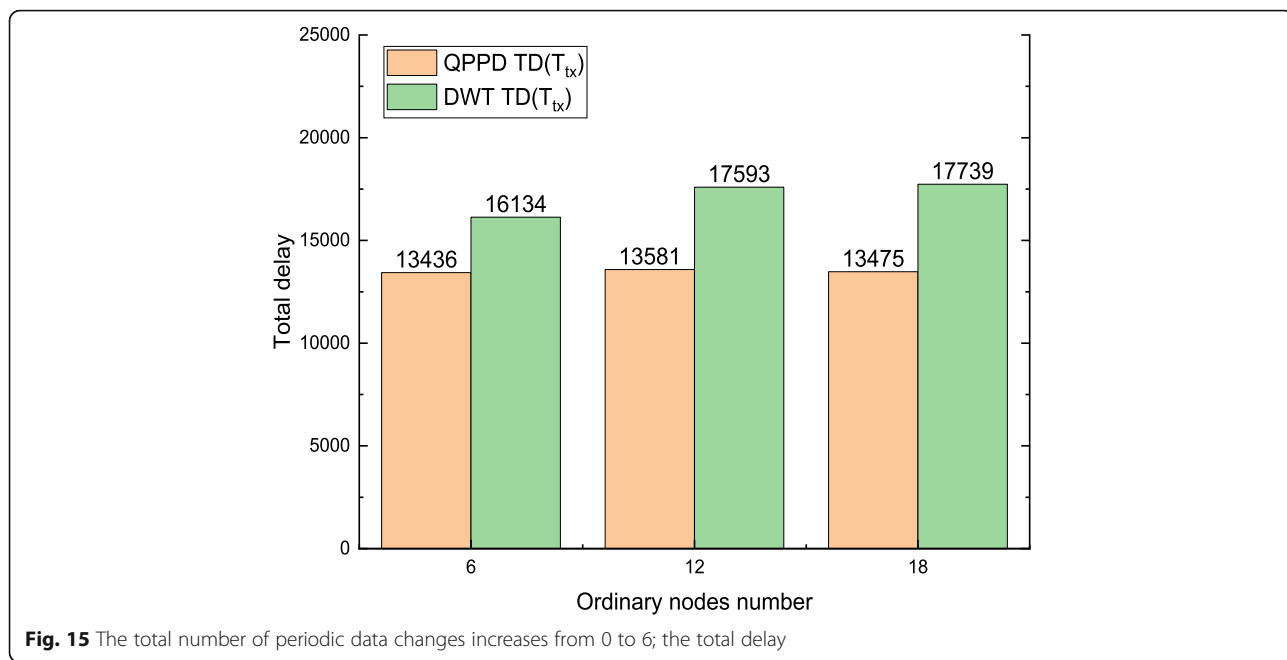


Fig. 14 The total number of periodic data changes increases from 0 to 6; the average delay of highest priority packets



**Fig. 15** The total number of periodic data changes increases from 0 to 6; the total delay

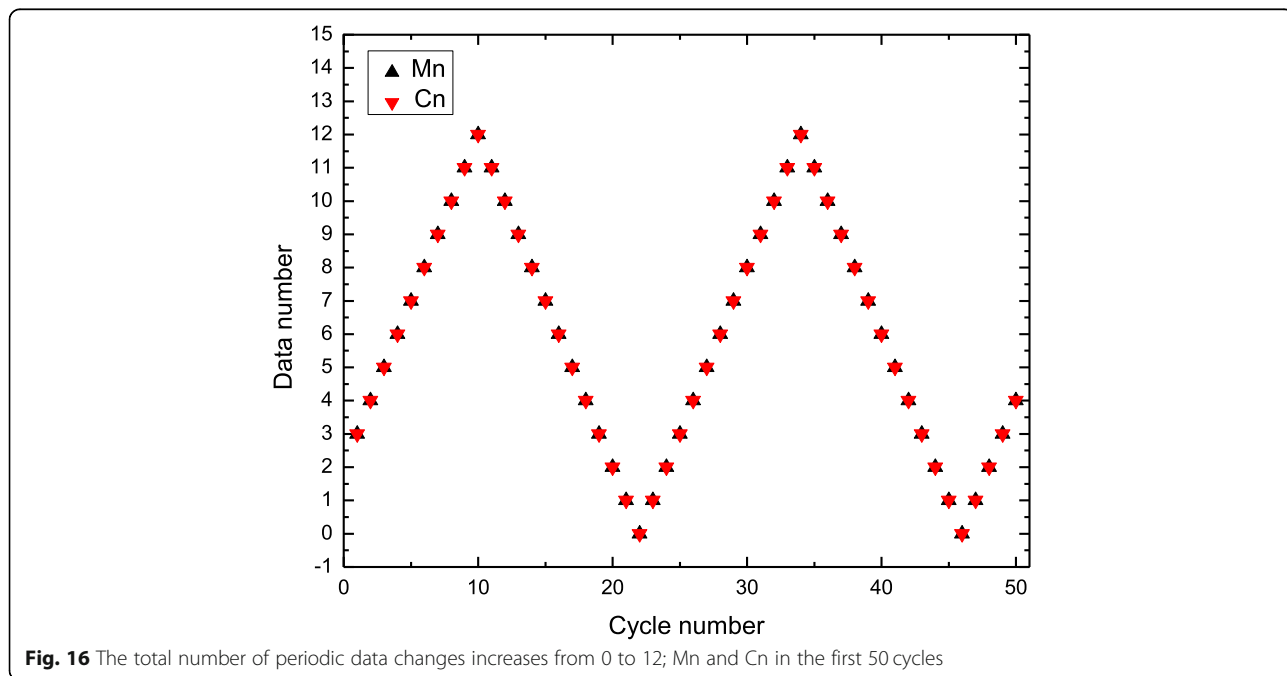
Symbols used in the simulations and their definitions are shown in Table 5.

**Delay comparisons**

**The total number of periodic data changes**

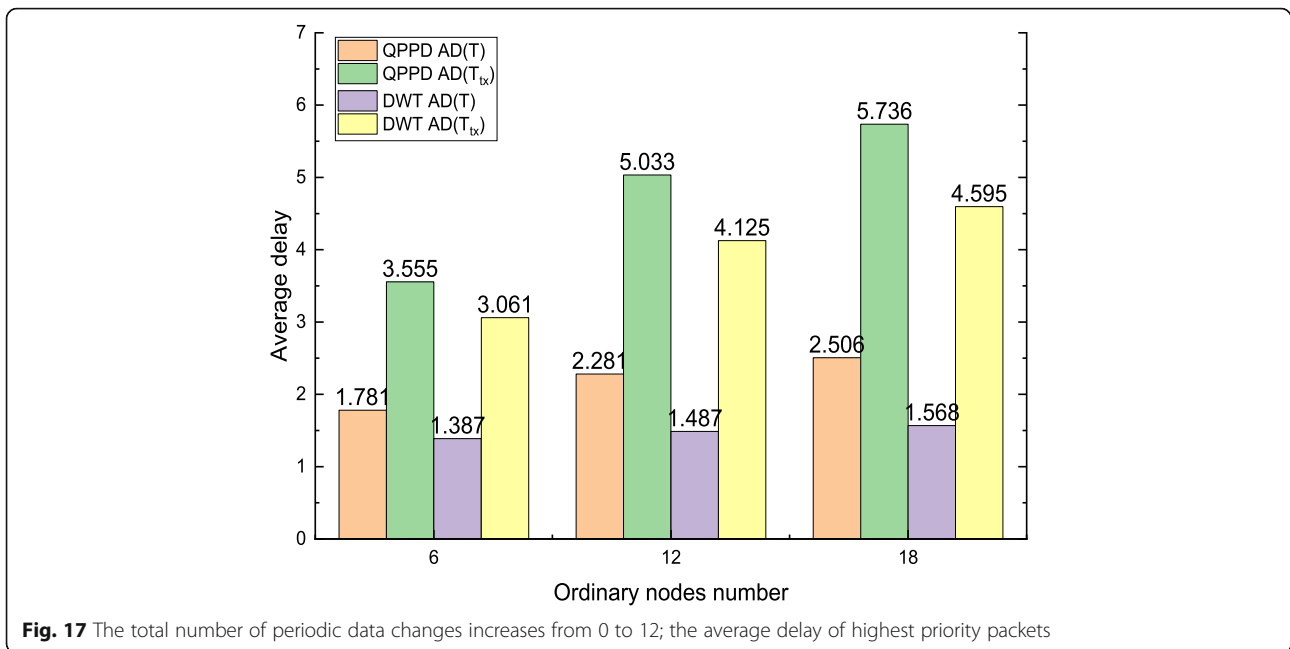
This section shows what results from periodic data changes in each cycle. The transmission delays of the DWT-MAC and QPPD-MAC protocols are evaluated under different conditions.

When the total number of periodic data changes shifts from 0 to 6, Mn and Cn for the first 50 cycles are shown in Fig. 13. The average delay of high priority packets is shown in Fig. 14. At this point, despite the total number of ordinary nodes, the average delay of highest priority packets is reduced under the DWT-MAC protocol. When the total number of ordinary nodes is 6, 12 or 18, the average delay of high priority packets for the DWT-MAC protocol



**Fig. 16** The total number of periodic data changes increases from 0 to 12; Mn and Cn in the first 50 cycles



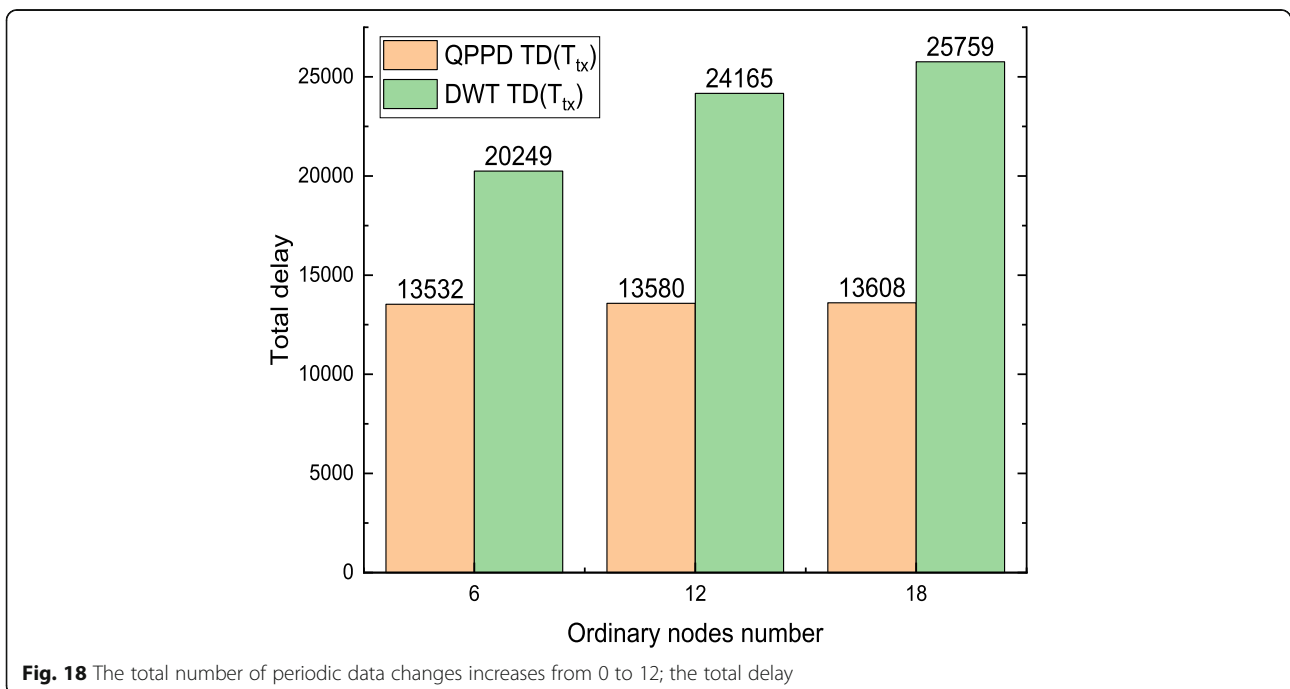


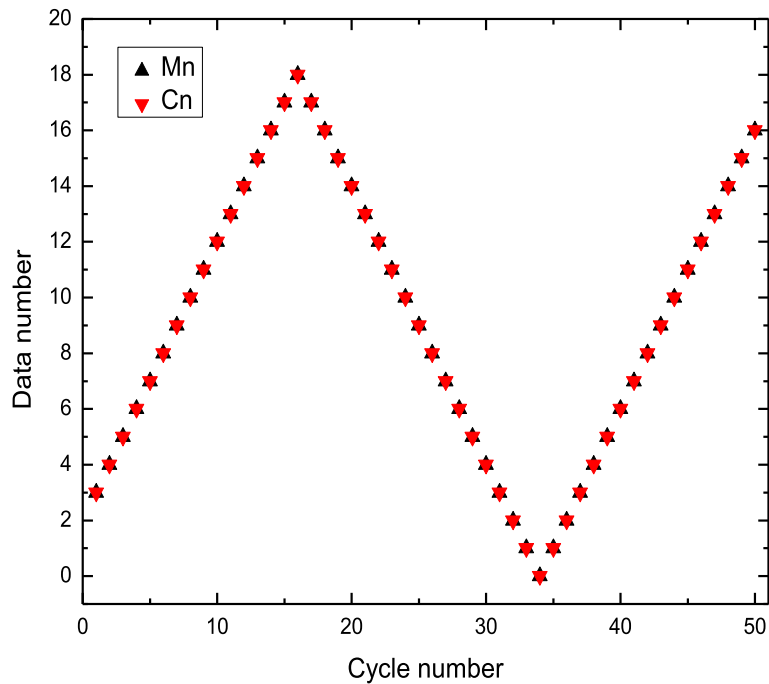
respectively decreases by 3.6%, 8% or 9.6% relative to the QPPD-MAC protocol. The average delay of highest priority packets is lower for the DWT-MAC protocol, but the effect is not significant, with the total number of ordinary nodes increasing and the average delay of high priority packets decreasing more. The total delay for this case is shown in Fig. 15. From the results, the total delay of the DWT-MAC protocol is increased despite the total

number of ordinary nodes relative to the QPPD-MAC protocol. As the number of ordinary nodes increases, the total transmission delay increases more, demonstrating a different trend from the average delay of highest priority packets.

As data in the network increase, the value periodically changes from 0 to 12.

Mn and Cn for the first 50 cycles are shown in Fig. 16. Average delays of high priority packets for

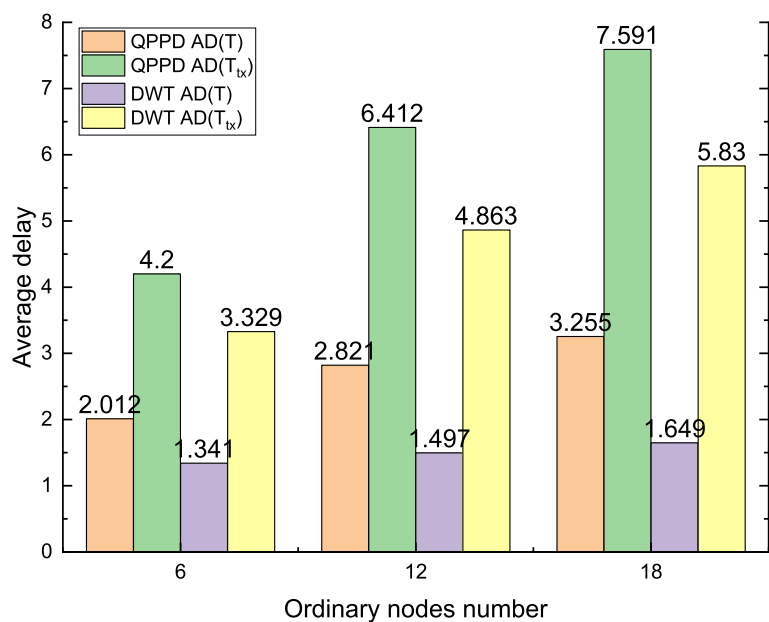




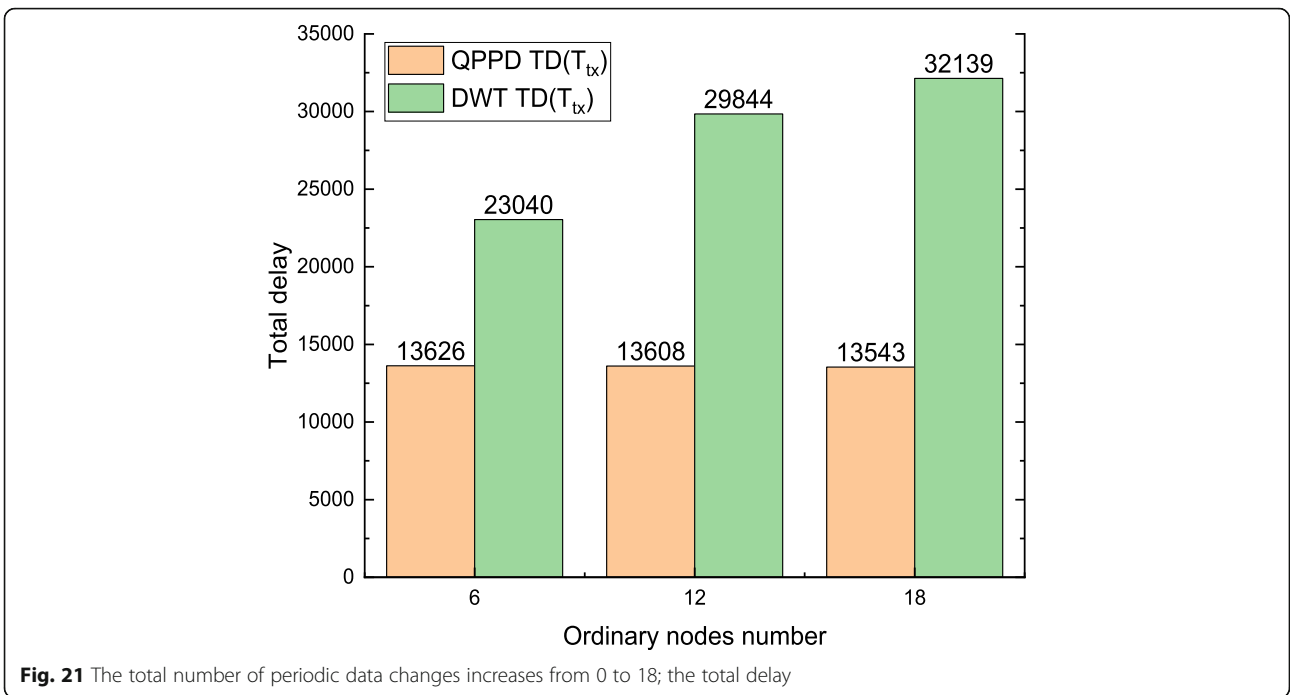
**Fig. 19** The total number of periodic data changes increases from 0 to 18; Mn and Cn in the first 50 cycles

different total numbers of ordinary nodes are shown in Fig. 17. In this case, when the number of ordinary nodes is 6, 12 or 18, the average delay of high priority packets for the DWT-MAC protocol is decreased by 22.4%, 35% or 37.8%, respectively, relative to the QPPD-MAC protocol. The average delay

of high priority packets still decreases despite the total number of ordinary nodes, but the reduction is significant. As the volume of data periodically changes from 0 to 6, the number of ordinary nodes increases, and the average delay of high priority packets is reduced more. The total delays of the



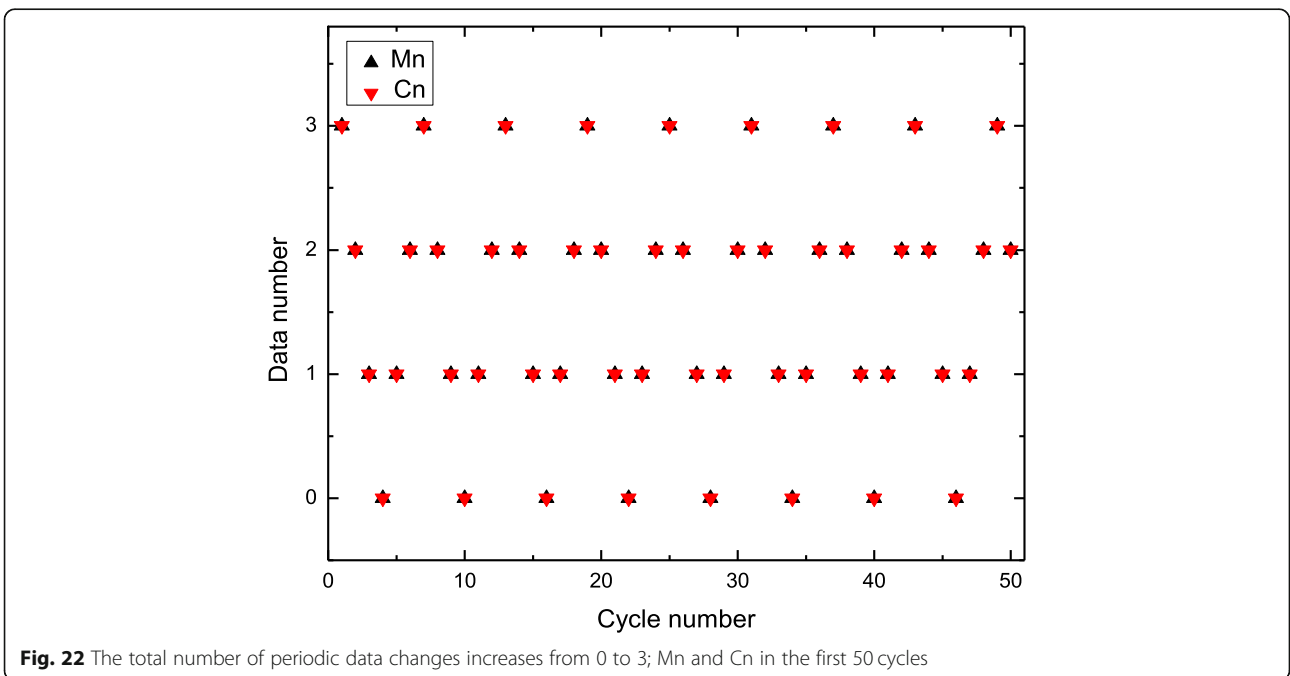
**Fig. 20** The total number of periodic data changes increases from 0 to 18; the average delay of highest priority packets



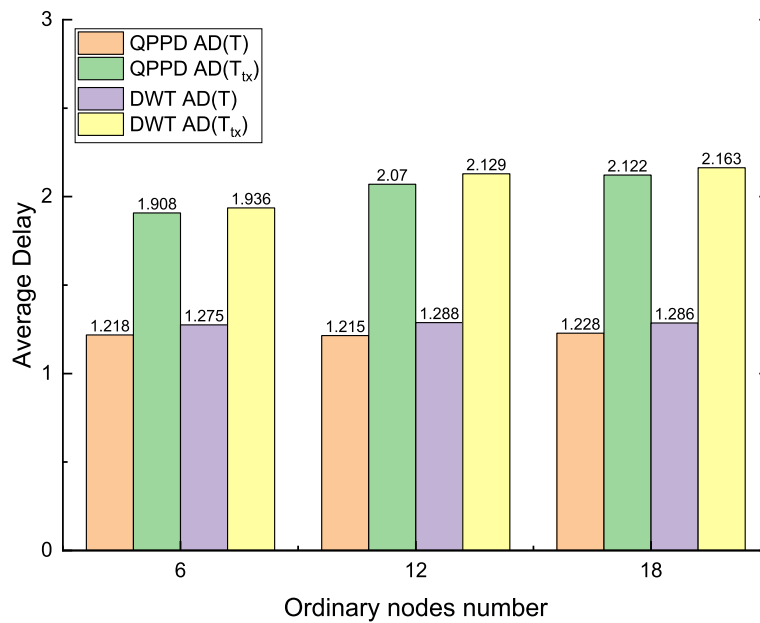
**Fig. 21** The total number of periodic data changes increases from 0 to 18; the total delay

DWT-MAC and QPPD-MAC protocols are shown in Fig. 18. The total transmission delay of the DWT-MAC protocol is higher than that of the QPPD-MAC protocol, and as the number of ordinary nodes increases, the total transmission delay increases more. Thus, the average delay of high priority packets is decreased more, and the total delay also increases more in this case.

The change in the total volume of data increases more when it periodically changes from 0 to 18. Mn and Cn for the first 50 cycles are shown in Fig. 19. The average delays of high priority packets under the DWT-MAC and QPPD-MAC protocols are shown in Fig. 20. In this case, when the number of ordinary nodes is 6, 12 or 18, the average delay of high priority packets under the DWT-MAC protocol is respectively decreased by 33.4%,



**Fig. 22** The total number of periodic data changes increases from 0 to 3; Mn and Cn in the first 50 cycles

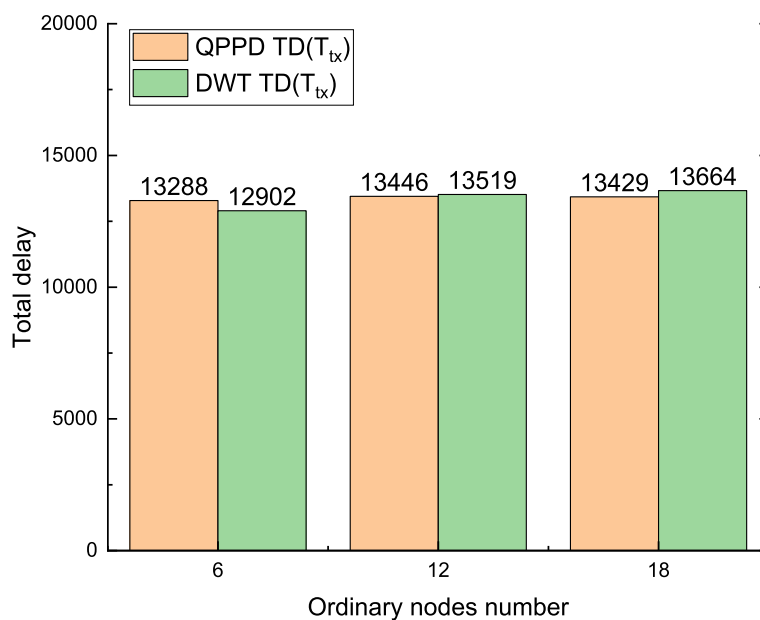


**Fig. 23** The total number of periodic data changes increases from 0 to 3; the average delay of highest priority packets

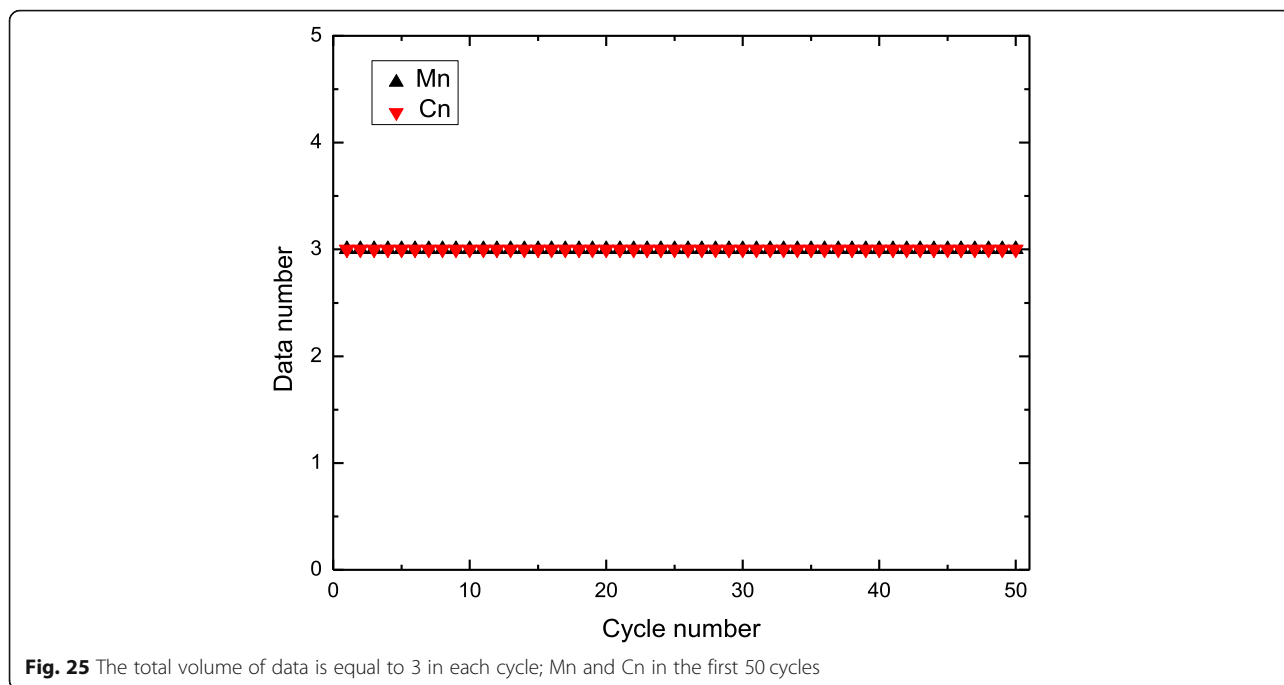
47% or 50% relative to the QPPD-MAC protocol. The total delays of the DWT-MAC and QPPD-MAC protocols are shown in Fig. 21, and that of the former increases more.

When the total number of periodic data changes increases from 0 to 3, the change in the volume of data is less than the maximum number of Tx-Beacons that can be transmitted during the initial wait time. Mn and Cn for the first 50 cycles are shown in Fig. 22.

The average delays of high priority packets under the DWT-MAC and QPPD-MAC protocols are shown in Fig. 23. The average delay of high priority packets is greater for the former in this case. When the total number of ordinary nodes is 6, 12 or 18, the average value of highest priority packets increases by 4.7%, 6% or 4.6%. Thus, in this case, the average delay of high priority packets is slightly higher under the DWT-MAC protocol. The total delays of DWT-MAC and



**Fig. 24** The total number of periodic data changes increases from 0 to 3; the total delay

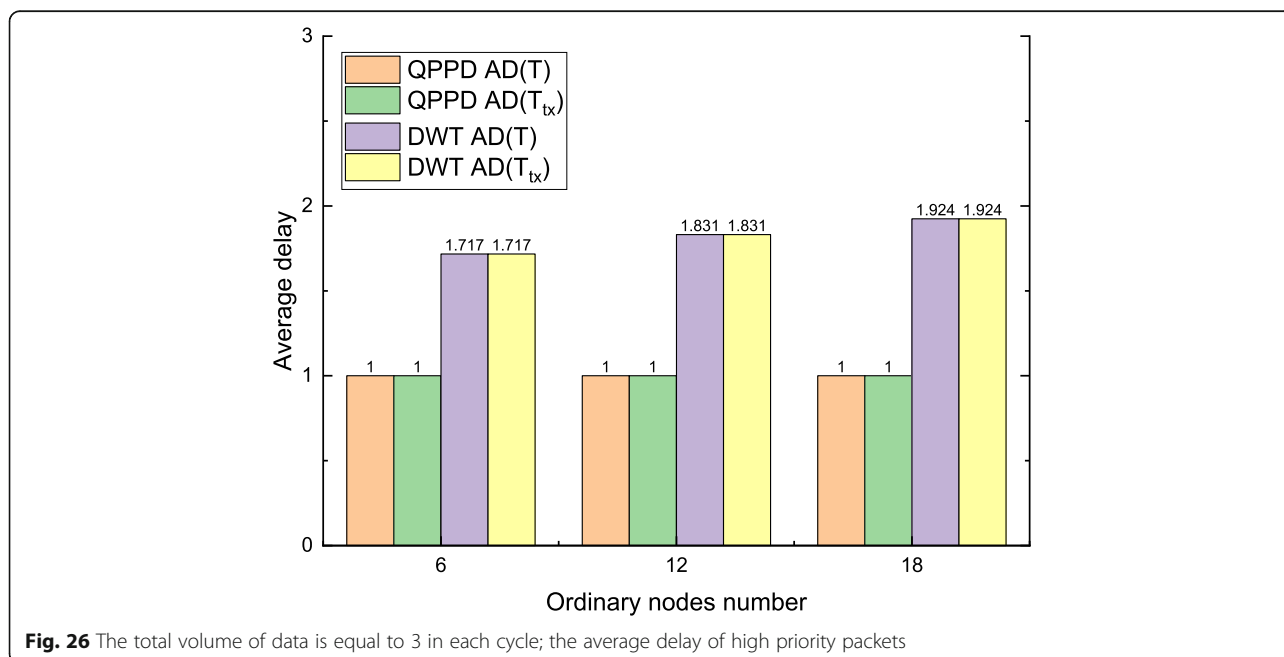


**Fig. 25** The total volume of data is equal to 3 in each cycle; Mn and Cn in the first 50 cycles

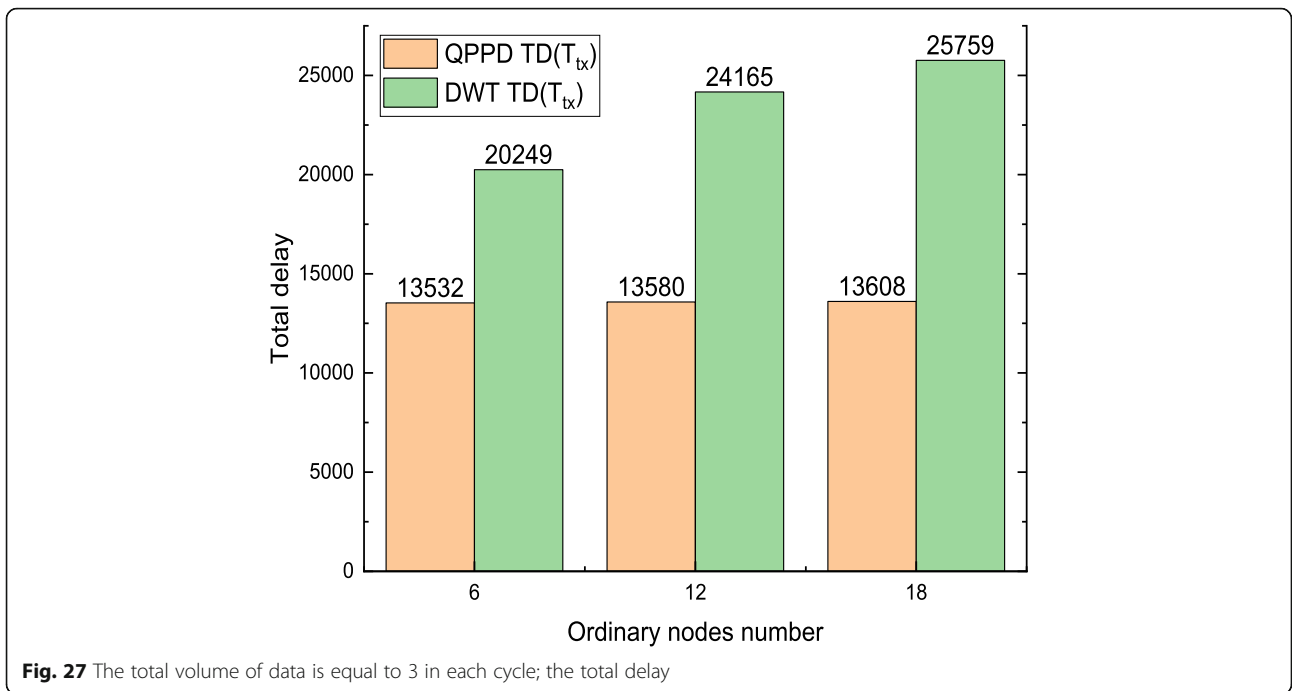
QPPD-MAC protocols are shown in Fig. 24. When the number of ordinary nodes in network is 6, the total delay for the DWT-MAC protocol is higher than that of the QPPD-MAC protocol, but when the number of ordinary nodes is 12 and 18, the total delay of the DWT-MAC protocol is less than that of the QPPD-MAC protocol. When the volume of data periodically changes from 0 to 3, as the number of ordinary nodes increases, data are more likely to be

distributed to different ordinary nodes. The number of cycles with at least three sender nodes increases as well. This adjusts the wait time of the DWT-MAC protocol to  $T_w = 4T_{tx}$  more often, leading the total delay of the DWT-MAC protocol to be higher.

Thus, the total volume of periodic data changes by a value less than the maximum number of Tx-Beacons that can be transmitted in the initial wait period. The average delay of high priority packets

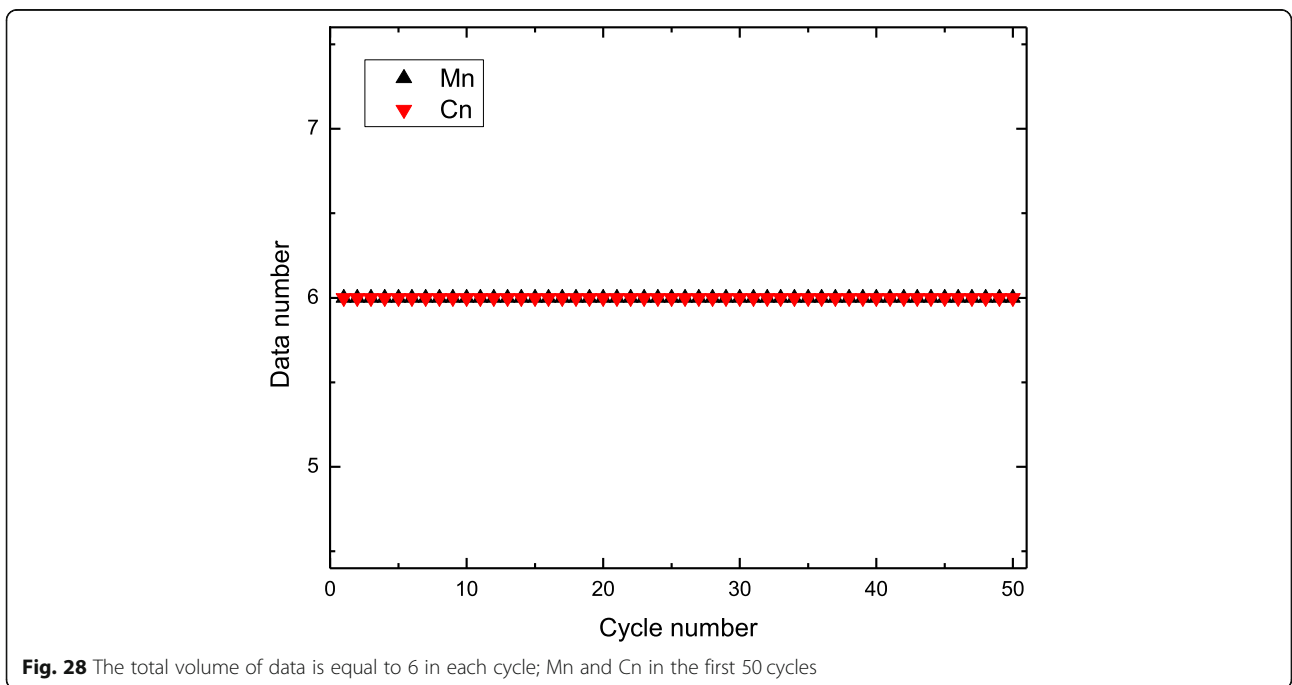


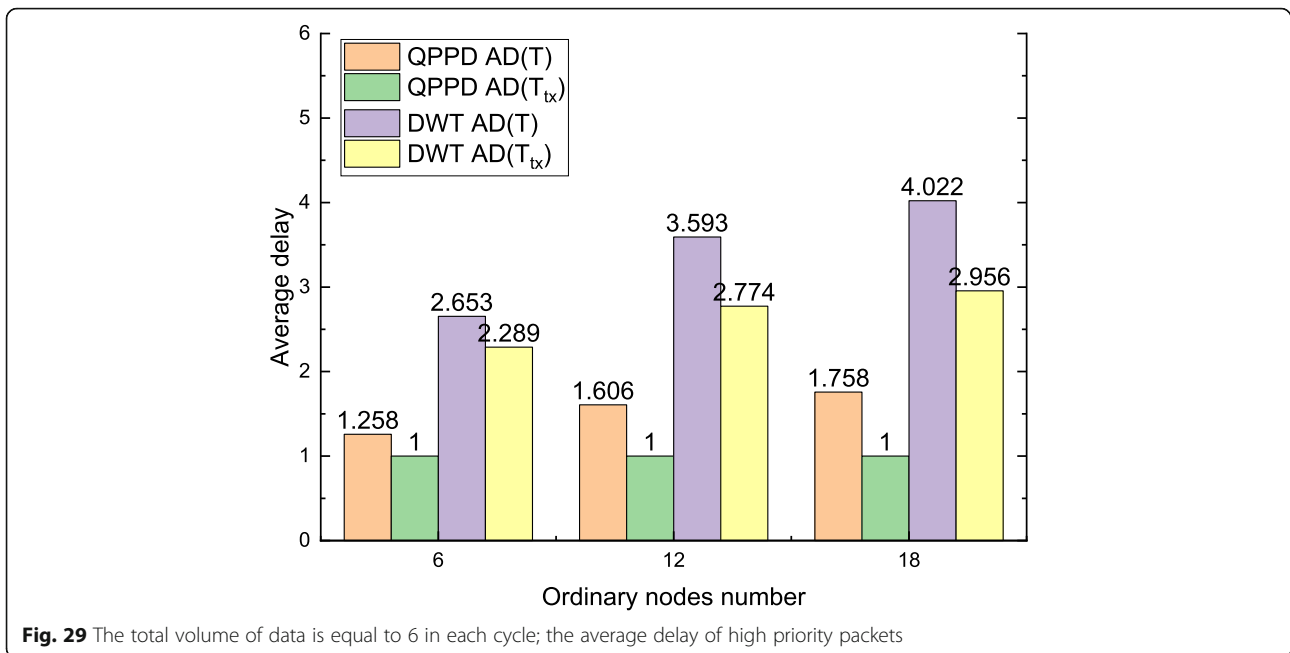
**Fig. 26** The total volume of data is equal to 3 in each cycle; the average delay of high priority packets



under the DWT-MAC protocol is less than that of the QPPD-MAC protocol. As this change becomes more significant and as the total number of ordinary nodes in the network increases, the average delay of high priority packets decreases more under the DWT-MAC protocol, but the total delay of the DWT-MAC protocol is greater than that of the QPPD-MAC protocol. When the total volume

of data in each cycle changes periodically by less than the maximum number of Tx-Beacons that can be transmitted in the initial wait period, the delay corresponds to different numbers of ordinary nodes. Whether the delay of the DWT-MAC protocol is less than that of the QPPD-MAC protocol depends on the total number of ordinary nodes. As the total number of ordinary nodes



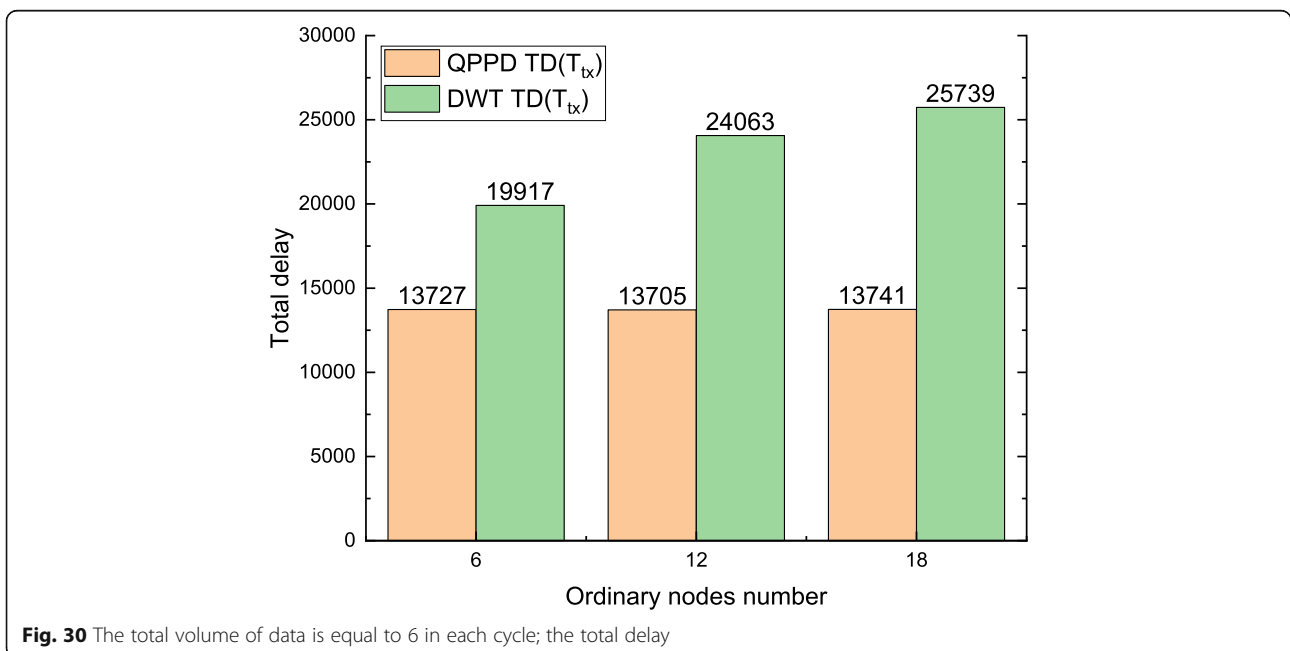


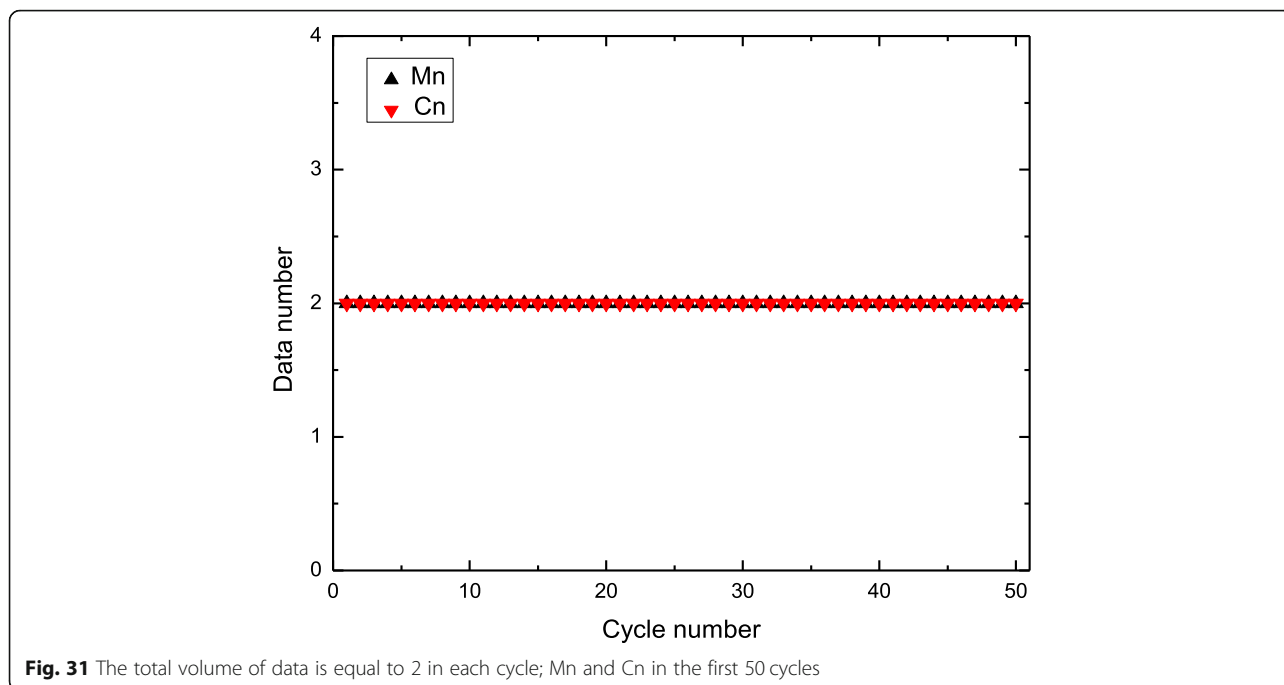
increases, the delay of the DWT-MAC protocol is more likely to become higher than that of the QPPD-MAC protocol.

**The total volume of data is constant**

This section illustrates delays that occur when the total volume of data remains the same in each cycle. DWT-MAC and QPPD-MAC protocol delays are evaluated under different conditions.

First, the total volume of data for each cycle is equal to 3, and  $M_n$  and  $C_n$  for the first 50 cycles are shown in Fig. 25. In this case, average delays of high priority packets for the DWT-MAC and QPPD-MAC protocols are shown in Fig. 26. Regardless of the total number of ordinary nodes, the average delay of high priority packets under the DWT-MAC protocol is equal to that of the QPPD-MAC protocol. The total delays of the DWT-MAC and QPPD-MAC protocols are shown in Fig. 27. The total delay of the



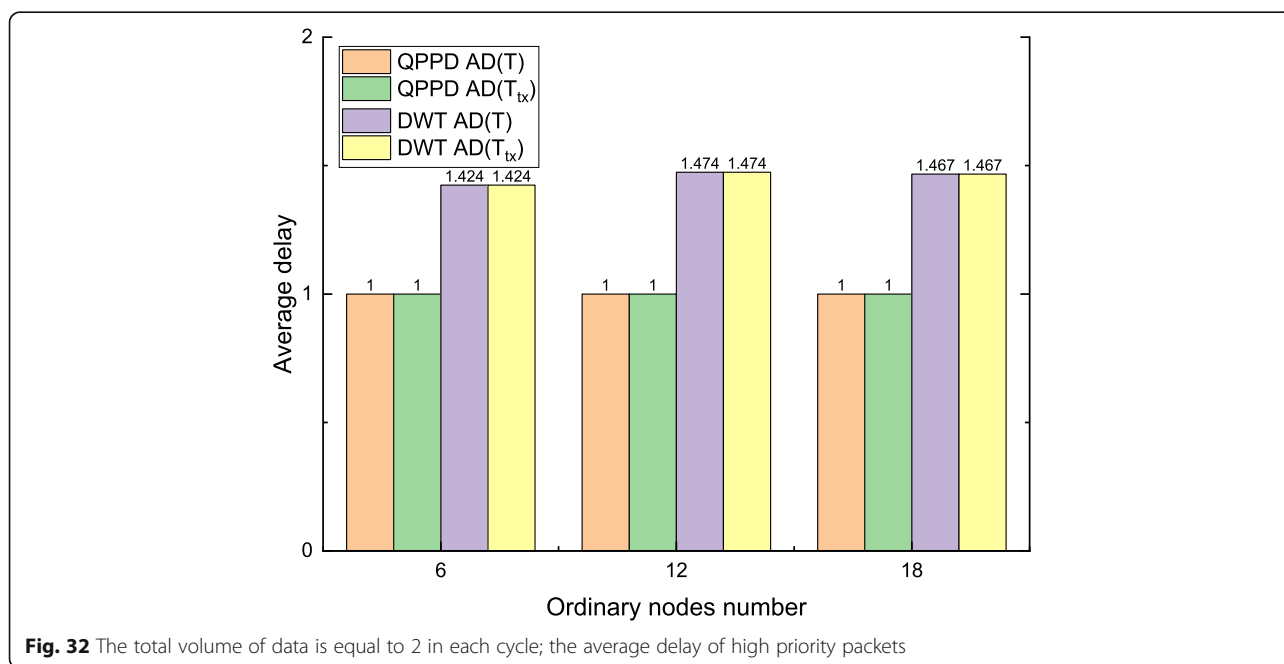


**Fig. 31** The total volume of data is equal to 2 in each cycle; Mn and Cn in the first 50 cycles

DWT-MAC protocol is higher than that of the QPPD-MAC protocol, as the wait time of the DWT-MAC protocol dynamically varies between  $2 T_{tx}$ ,  $3 T_{tx}$  and  $4 T_{tx}$  in this case, causing the total delay of the DWT-MAC protocol to increase. As the total number of ordinary nodes increases, the total delay of the DWT-MAC protocol increases more, and as the total number of ordinary nodes increases, new data are more likely to be distributed to different

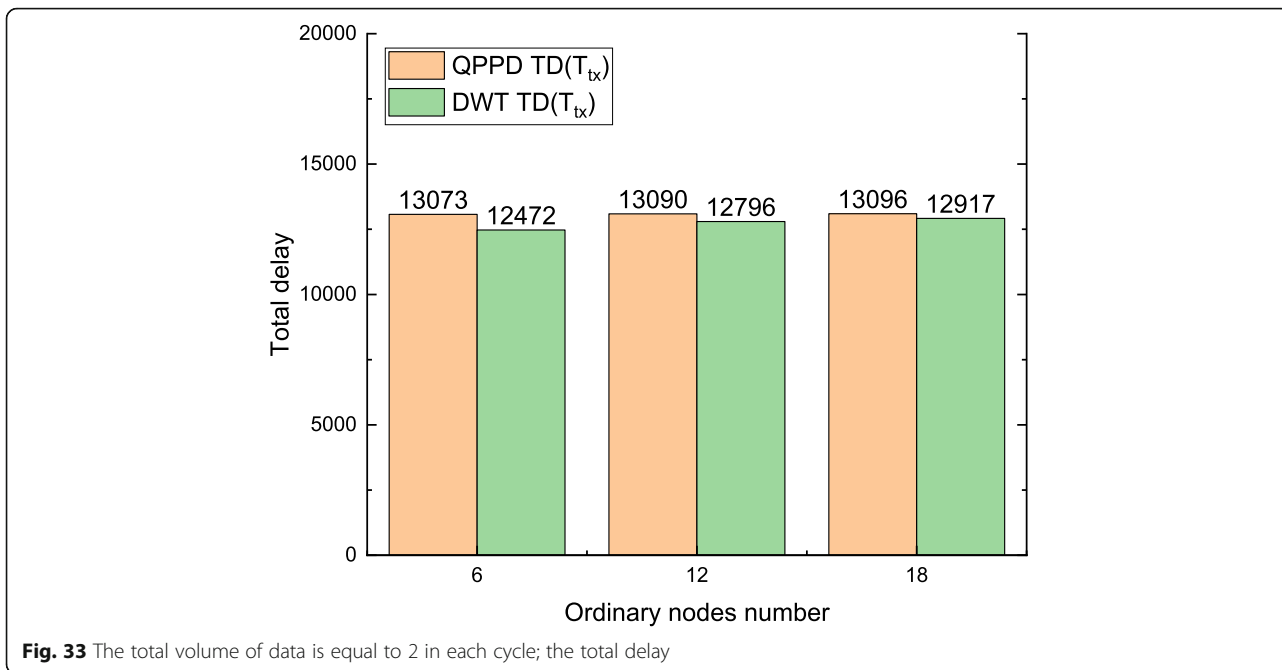
ordinary nodes. The number of cycles with at least three nodes participating in transmission increases as well.

We increase the total volume of data in each cycle to 6. Mn and Cn for the first 50 cycles are shown in Fig. 28. The average delays of high priority packets of the DWT-MAC and QPPD-MAC protocols are shown in Fig. 29. When the volume of data in each cycle is greater than 3, the average delay of high priority



**Fig. 32** The total volume of data is equal to 2 in each cycle; the average delay of high priority packets

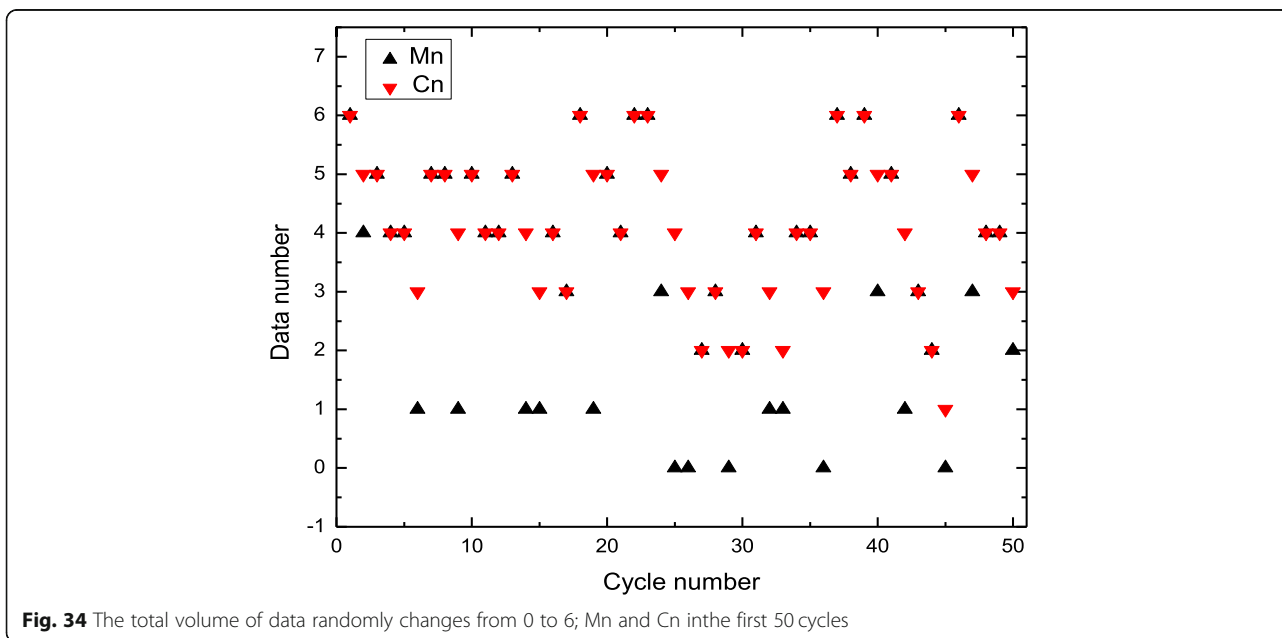


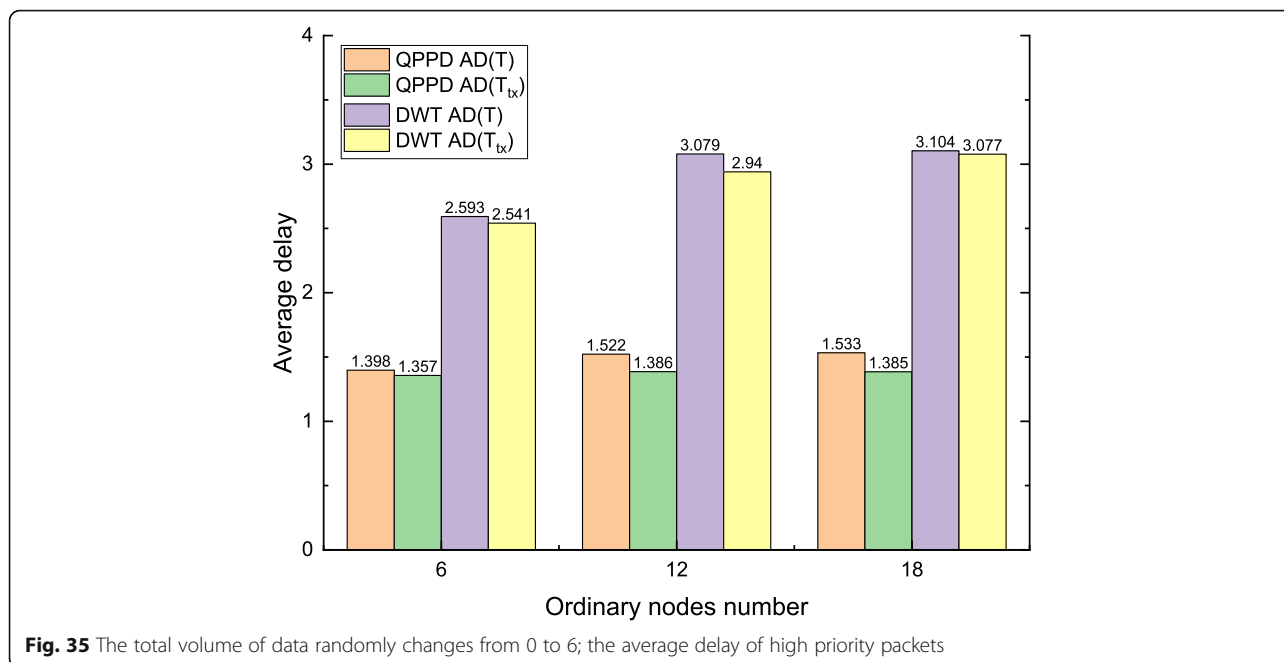


packets of the DWT-MAC protocol is less than that of the QPPD-MAC protocol. When the total number of ordinary nodes is 6, 12 or 18, the average delay of high priority packets of the DWT-MAC protocol respectively decreases by 20.6%, 37.7% or 43.1% relative to the QPPD-MAC protocol. The average delay of high priority packets of the DWT-MAC is decreased significantly in this case. The total delays of the DWT-MAC and QPPD-MAC protocols are shown in Fig. 30, and that of the DWT-MAC protocol is higher because its

wait times are always longer to transmit high priority data first, thus increasing delays.

We set the total volume of data for each cycle to less than 3 and equal to 2, and  $M_n$  and  $C_n$  for the first 50 cycles are shown in Fig. 31. The average delays of high priority packets of the DWT-MAC and QPPD-MAC protocols are shown in Fig. 32 and they are equal in this case as observed when the total volume of data is equal in each cycle. The total delays of the DWT-MAC and QPPD-MAC protocols in this



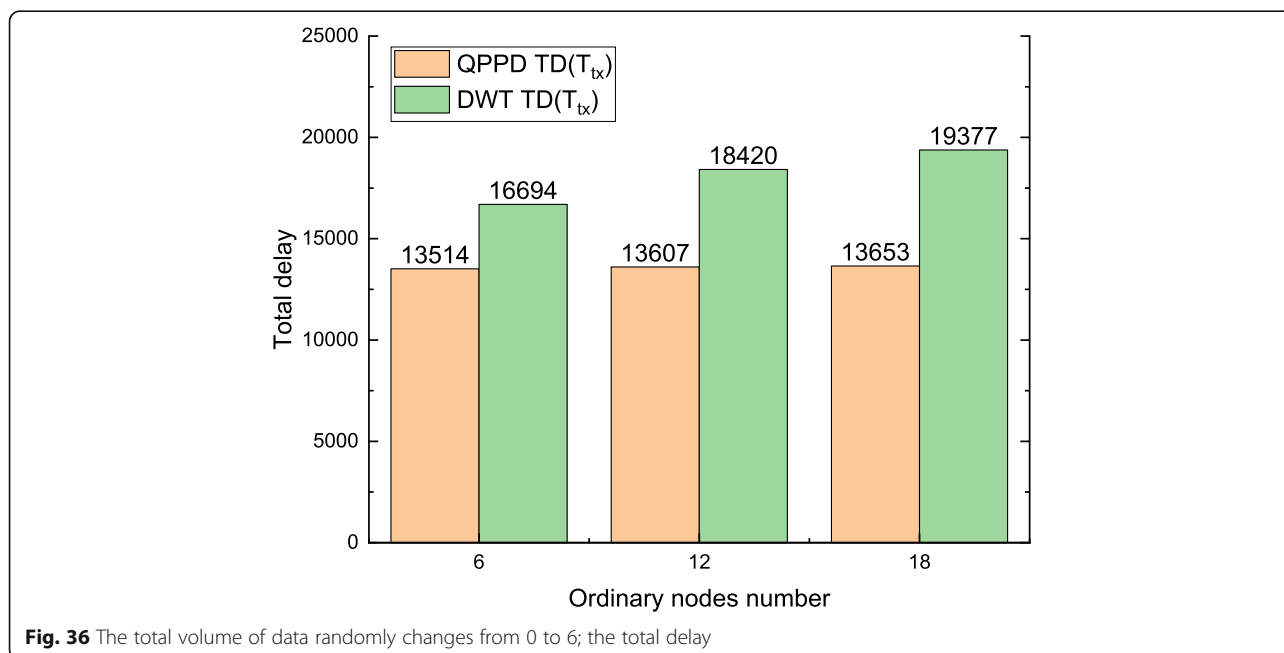


**Fig. 35** The total volume of data randomly changes from 0 to 6; the average delay of high priority packets

case are shown in Fig. 33, and that of the DWT-MAC protocol is lower since under the QPPD-MAC protocol, each cycle has two sender nodes with data to transmit, wasting time waiting, while the DWT-MAC protocol can adjust the wait time to  $2T_{tx}$ , leading to a lower average delay of high priority packets and lower total delays.

Therefore, when the total volume of data per cycle remains the same and is equal to the maximum number of Tx-Beacons that can be transmitted in the

initial wait period, the average delay of high priority packets under the DWT-MAC protocol is equal to that of the QPPD-MAC protocol, and the total delay of the DWT-MAC protocol is greater than that of the QPPD-MAC protocol. When the total volume of data is less than the maximum number of Tx-Beacons that can be transmitted in the initial wait period, the average delay of high priority packets under the DWT-MAC protocol is still equal to that of the QPPD-MAC protocol, but the total delay of



**Fig. 36** The total volume of data randomly changes from 0 to 6; the total delay

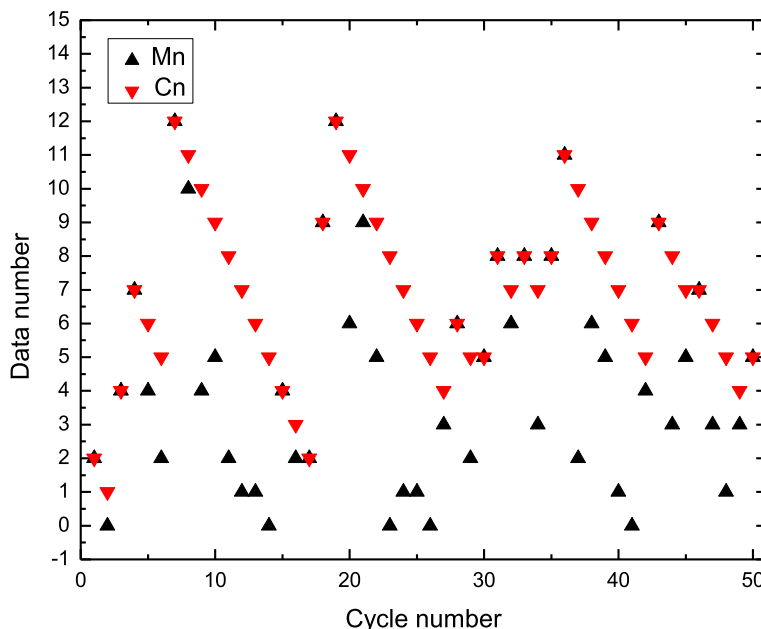


Fig. 37 The total volume of data randomly changes from 0 to 12; Mn and Cn in the first 50 cycles

the DWT-MAC protocol is less than that of the QPPD-MAC protocol.

When the volume of data is greater than the maximum number of Tx-Beacons that can be transmitted in the initial wait period, the average delay of high priority packets under the DWT-MAC protocol is less than that under the QPPD-MAC protocol, but the total delay of the DWT-MAC protocol is higher than that of the QPPD-MAC protocol.

**The total number of data irregularly changes**

This section illustrates the delay that occurs when the total volume of data randomly changes. The delays of the DWT-MAC and QPPD-MAC protocols are evaluated under different conditions.

The total volume of data randomly generated is increased from 0 to 6. Mn and Cn in the first 50 cycles are shown in Fig. 34. The average delays of high priority packets under the DWT-MAC and QPPD-

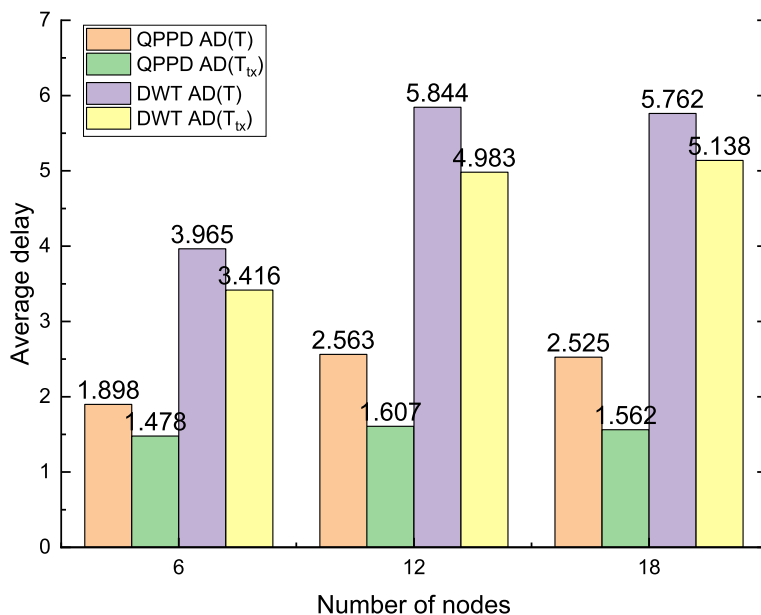


Fig. 38 The total volume of data randomly changes from 0 to 12; the average delay of high priority packets

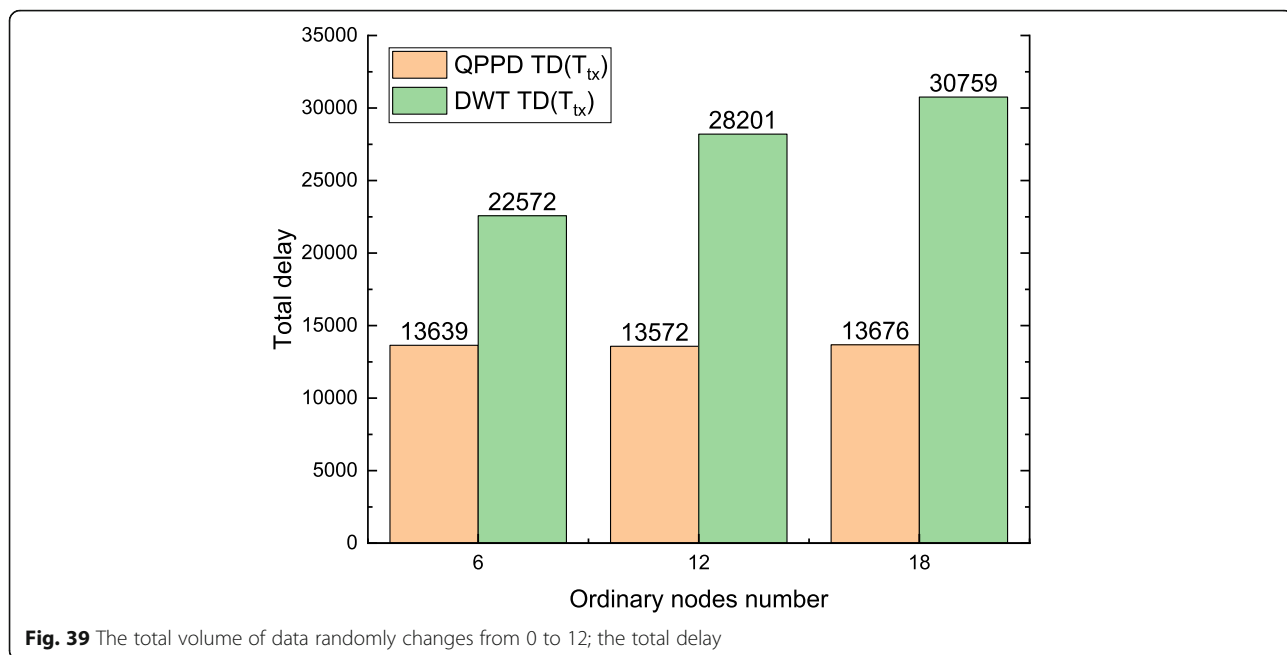


Fig. 39 The total volume of data randomly changes from 0 to 12; the total delay

MAC protocols are shown in Fig. 35; when the total number of ordinary nodes is 6, 12 or 18, the average delay of high priority packets under the DWT-MAC protocol is decreased by 3.2%, 10.8% or 11%, respectively, relative to the QPPD-MAC protocol. The total delays of the DWT-MAC and QPPD-MAC protocols are shown in Fig. 36, and that of the DWT-MAC protocol is higher in this case.

We increase the scope of total data volume change with volumes randomly increasing from 0 to 12. Mn and Cn in the first 50 cycles are shown in Fig. 37. The average delays of high priority packets for the DWT-MAC and QPPD-MAC protocols are shown in Fig. 38. In this case, the average delay of high priority packets under the DWT-MAC protocol is further reduced, and when the total number of ordinary nodes is 6, 12 or 18, delays respectively decrease

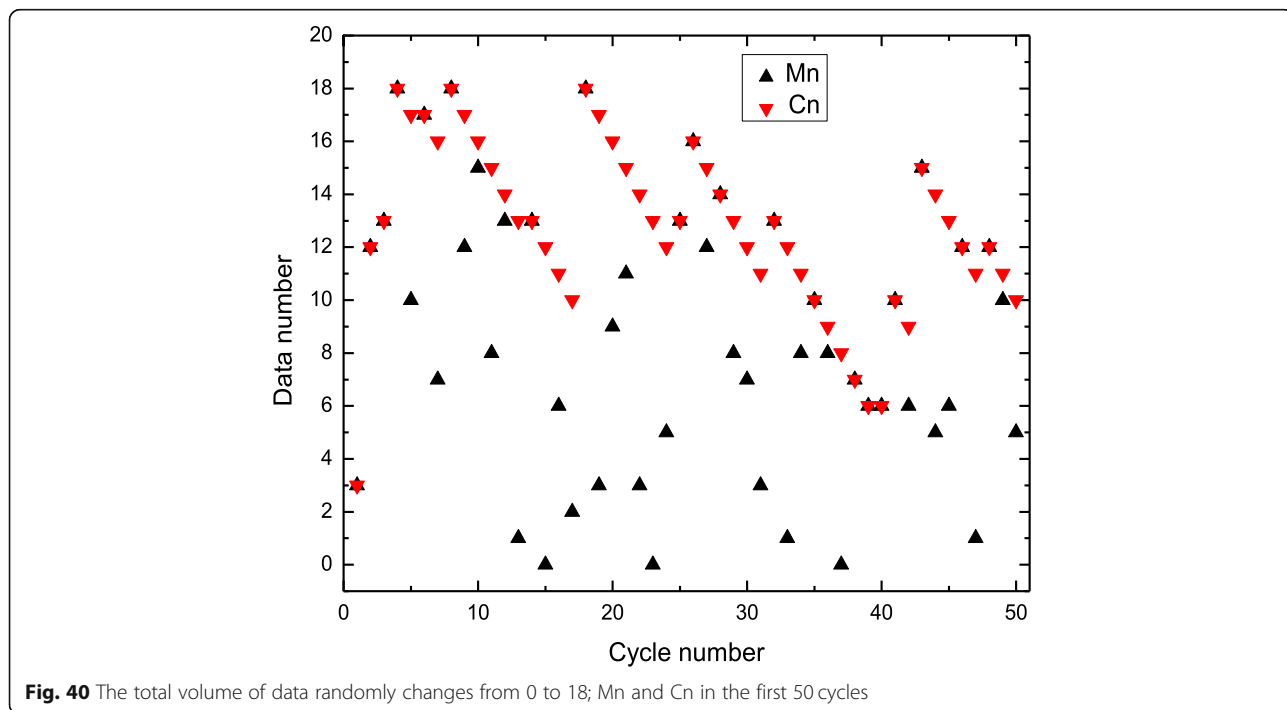
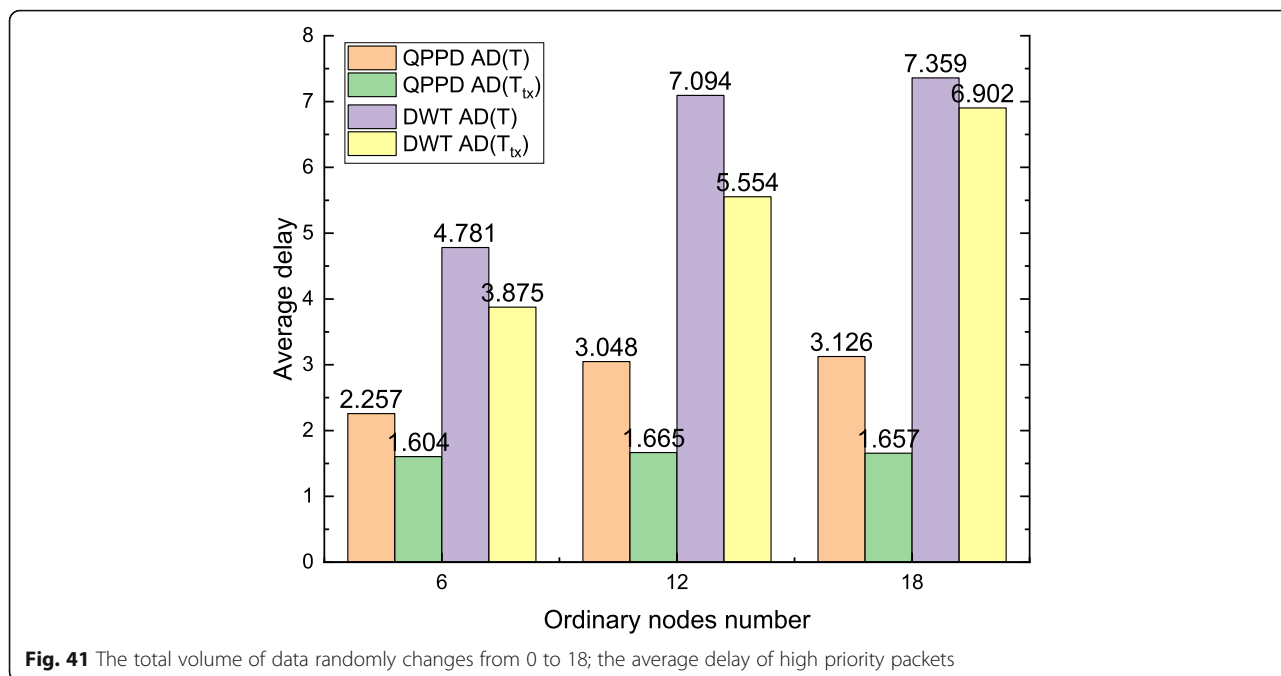


Fig. 40 The total volume of data randomly changes from 0 to 18; Mn and Cn in the first 50 cycles

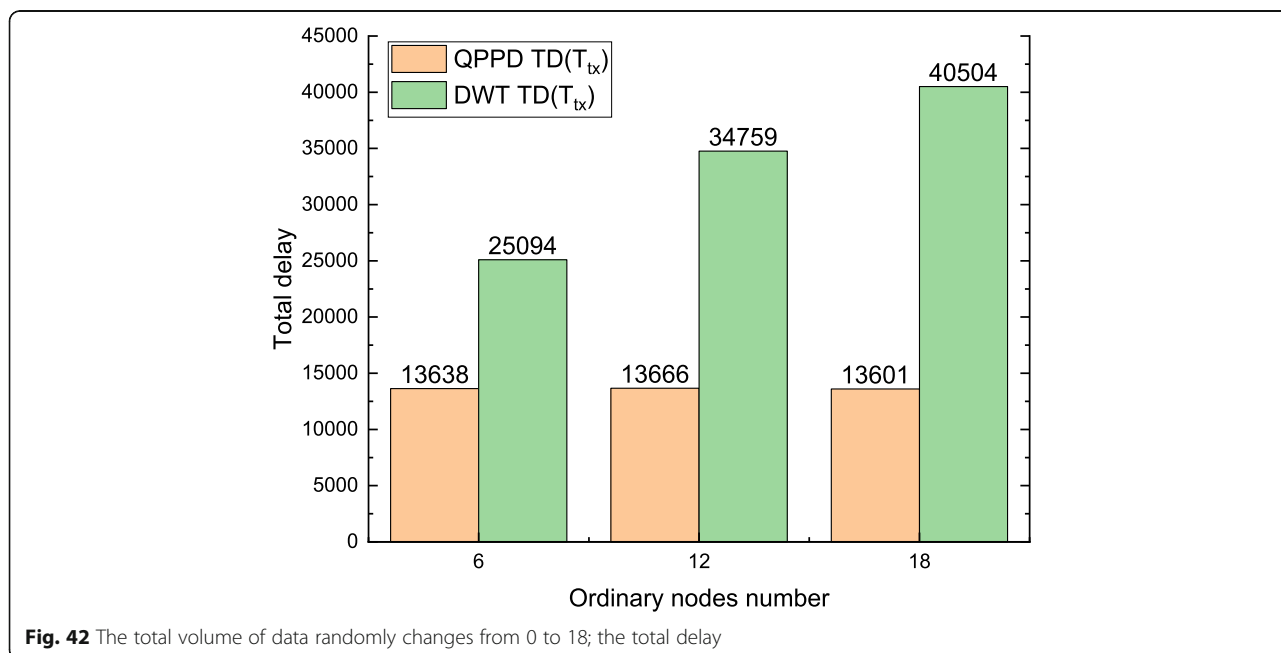


**Fig. 41** The total volume of data randomly changes from 0 to 18; the average delay of high priority packets

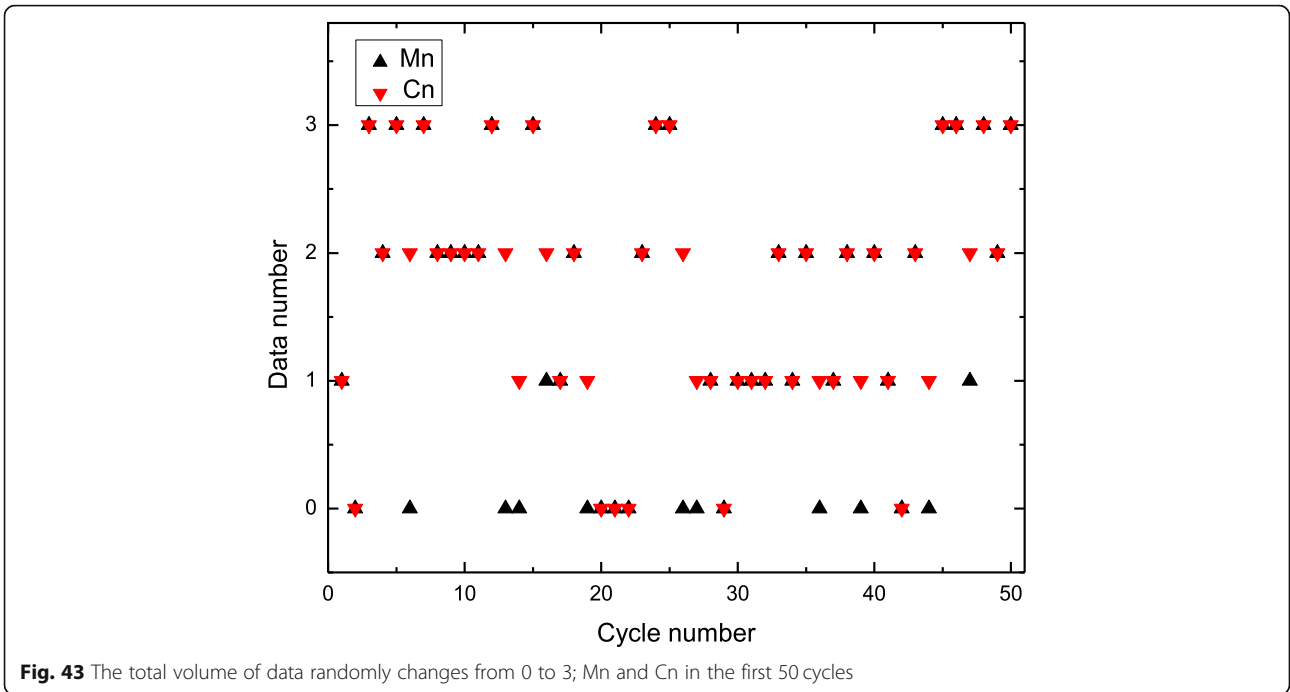
by 22%, 37.6% or 38% relative to the QPPD-MAC protocol. The total delays of the DWT-MAC and QPPD-MAC protocols are shown in Fig. 39, and that of the DWT-MAC protocol is higher in this case.

We further increase the scope of data volume changes with volumes randomly increasing from 0 to 18. Mn and Cn in the first 50 cycles are shown in Fig. 40. The average delays of high priority packets for the DWT-MAC and QPPD-MAC protocols are shown in Fig. 41, and when the total number of ordinary nodes is 6, 12 or 18,

delays respectively decrease by 30%, 45.6% and 47% under the DWT-MAC protocol relative to the QPPD-MAC protocol. Under the DWT-MAC protocol, the average delay of high priority packets decreases significantly in this case. Additionally, as the number of ordinary nodes increases, the average transmission delay of highest priority packets decreases more under the DWT-MAC protocol. The total delays of the DWT-MAC and QPPD-MAC protocols are shown in Fig. 42, and that of the former is still higher in this instance.

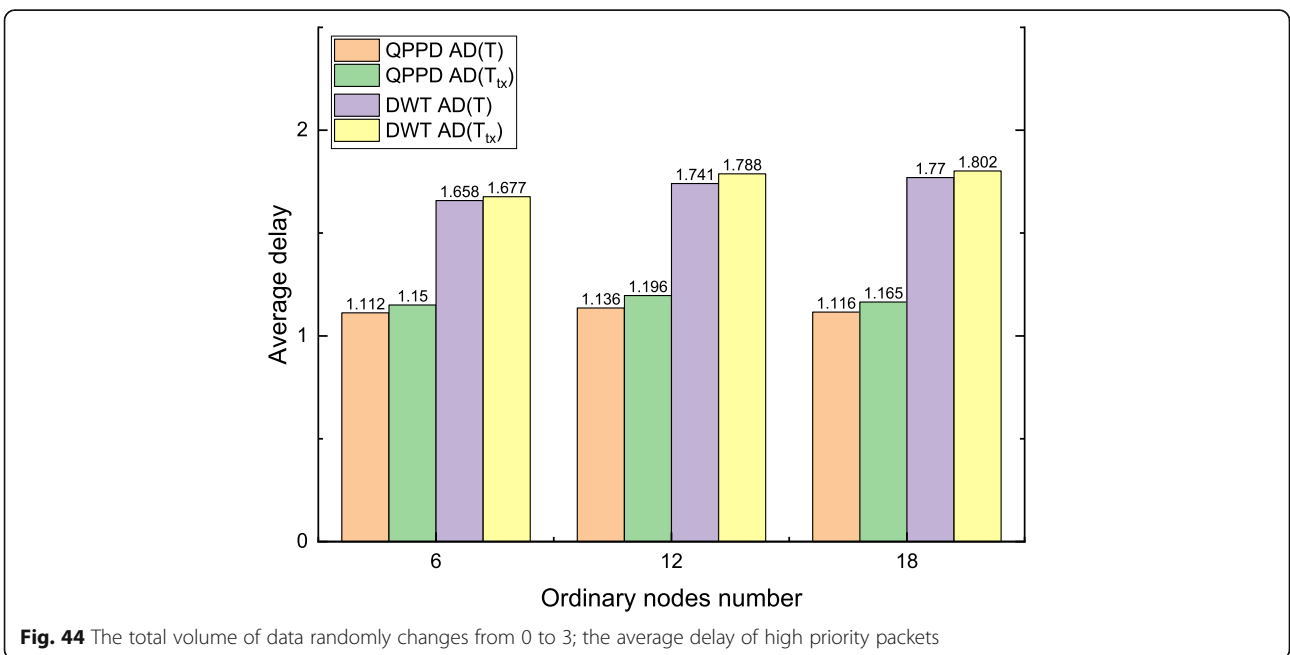


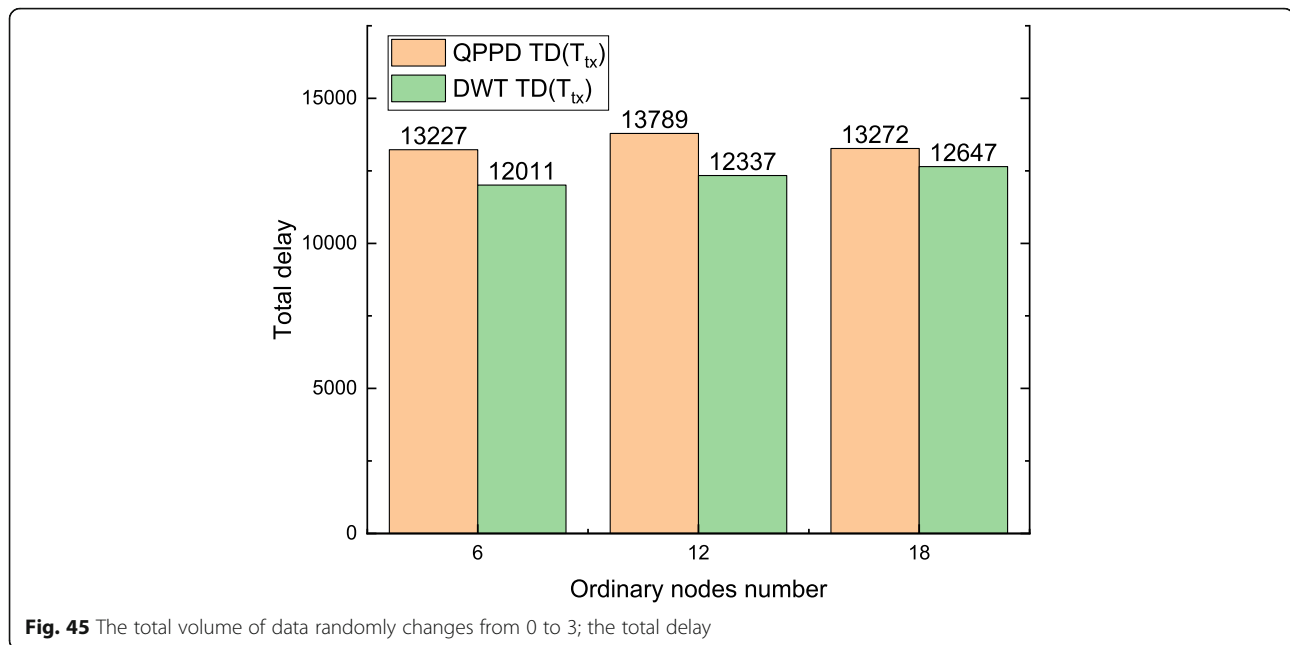
**Fig. 42** The total volume of data randomly changes from 0 to 18; the total delay



We decrease the scope of data volume changes with volumes randomly increasing from 0 to 3. Mn and Cn in the first 50 cycles are shown in Fig. 43. The average delays of high priority packets for the DWT-MAC and QPPD-MAC protocols are shown in Fig. 44, when the volume of data is less than or equal to the maximum number of Tx-Beacons that can be transmitted in the initial wait period, the average delay of high priority packets under the

DWT-MAC protocol is higher than that under the QPPD-MAC protocol, and when the total number of ordinary nodes is 6, 12 or 18, delays respectively increase up 2.6%, 5.2% or 4.3% under the DWT-MAC protocol relative to the QPPD-MAC protocol. The total delay for this case is shown in Fig. 45. From the results, the total delay of the DWT-MAC protocol is decreased relative to the QPPD-MAC protocol.





### Experiment summary

The experimental results are summarized as follows: no matter the volume of data changes in what way, when the total volume of data changes by a value greater than the maximum number of Tx-Beacons that can be transmitted in the initial wait period, the average delay of high priority packets under the DWT-MAC protocol is less than that under the QPPD-MAC protocol, but the total delay of the DWT-MAC protocol is higher relative to the QPPD-MAC protocol, as the scope of data volume or the total number of ordinary nodes increase, the greater the proportion of the average delay of high priority packets and the total delay change under the DWT-MAC protocol relative to the QPPD-MAC protocol. When the total volume of data changes by a value less than or equal to the maximum number of Tx-Beacons that can be transmitted in the initial wait period, experimental results show that even when the average delay of high priority packets under the DWT-MAC protocol is increased relative to the QPPD-MAC protocol, the proportion of increase is not large. In most cases, when the average delay of high priority packets under the DWT-MAC protocol is decreased relative to the QPPD-MAC protocol, the total delay under the DWT-MAC protocol is increased relative to the QPPD-MAC protocol.

### Conclusion

In the previous receiver-initiated MAC protocols for prioritized data, the wait time is set as a fixed duration and thus leads to long delays in data transmission and high network energy consumption. The DWT-MAC protocol is proposed in this paper for reducing the

delay in data transmission as well as network energy consumption, the main innovation of the DWT-MAC protocol is that it is based on a novel dynamic timer adjustment algorithm, it can dynamically adjust the wait times in each cycle according to the number of Tx-Beacons received by receiver. The performance of the DWT-MAC protocol is evaluated in simulations, the results show that by dynamically adjusting the wait time, when the total volume of data changes greatly in the network, the DWT-MAC protocol can significantly reduce the average end-to-end delay of high priority packets, and when the total volume of data changes not greatly in the network, the DWT-MAC protocol can also significantly reduce the total end-to-end delay.

### Acknowledgements

There is no acknowledgement.

### Authors' contributions

SL and XL are the main researcher, GH, JG, and TW are coauthor. All authors read and approved the final manuscript.

### Authors' information

Shun Liu is currently a Master student at the School of Computer Science and Engineering, Central South University, China. His research interest is wireless sensor networks. Email: [shunliu@csu.edu.cn](mailto:shunliu@csu.edu.cn).

Guosheng Huang received his M.S. and Ph.D. degrees in computer science from Central South University, China, 2001 and 2010 respectively. He was a visiting scholar with Sun Yat-Sen University, From 2017 to 2018. Currently, he is an associate professor of School of Information Science and Engineering, Hunan First Normal University, China. His major research interests include MIMO techniques, wireless sensor network and mobile computing. Email: [hgsheng@mail.sysu.edu.cn](mailto:hgsheng@mail.sysu.edu.cn)

Jinsong Gui. He received the BE from the University of Shanghai for Science and Technology, China, in 1992, and the MS and PhD from Central South University, China, in 2004 and 2008, respectively. He is currently a Professor in School of Computer Science and Engineering, Central South University. He is the member of China Computer Federation (CCF) and the IEEE member.

He published over 50 international journal papers and over 10 international conference papers. His research interests cover the general area of distributed systems, as well as related fields such as wireless network topology control and network security. Email: [jsgui2010@mail.csu.edu.cn](mailto:jsgui2010@mail.csu.edu.cn)

**Tian Wang** received his BSc and MSc degrees in Computer Science from the Central South University in 2004 and 2007, respectively. He received his Ph.D. degree from the City University of Hong Kong in 2011. Currently, he is an associate professor at the National Huaqiao University of China. His research interests include wireless sensor networks, social networks and mobile computing. E-mail: [wangtian@hqu.edu.cn](mailto:wangtian@hqu.edu.cn).

**Xiong Li** now is a Professor at School of Computer Science and Engineering of University of Electronic Science and Technology of China (UESTC), China. He received his master's degree in mathematics and cryptography from Shaanxi Normal University (SNNU), China in 2009 and Ph.D. degree in computer science and technology from Beijing University of Posts and Telecommunications (BUPT), China in 2012. He has published more than 100 referred journal papers in his research interests, which include cryptography, information security, cloud computing security etc. He is currently an Editor of Telecommunication Systems and KSII Transactions on Internet and Information Systems. He was the recipient of 2020 IEEE Systems Journal best paper award and 2015 Journal of Network and Computer Applications best research paper award. E-mail: [lixiongzq@163.com](mailto:lixiongzq@163.com).

#### Funding

This work has been institutionally supported by the National Key Research and Development Program of China (No. 2018YFE0126300), the National Natural Science Foundation of China with grant number 61772554, 61803387.

#### Availability of data and materials

There is no supporting data available.

#### Competing interests

The authors declare that they have no competing interests.

#### Author details

<sup>1</sup>School of Computer Science and Engineering, Central South University, Changsha 410083, China. <sup>2</sup>School of Information Science and Engineering, Hunan First Normal University, Changsha, China. <sup>3</sup>College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China. <sup>4</sup>Institute for Cyber Security, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China.

Received: 20 January 2020 Accepted: 14 August 2020

Published online: 31 August 2020

#### References

- Wang T, Zhao D, Cai S, Jia W, Liu A (2020) Bidirectional prediction based underwater data collection protocol for end-edge-cloud orchestrated system. *IEEE Trans Ind Inform* 16(7):4791–4799
- Zhao Y, Wang T, Zhang S, Wang Y (2020) Towards minimum code dissemination delay through UAV joint vehicles for smart city. *IET Commun*. <https://doi.org/10.1049/iet-com.2019.1205>
- Liu Y, Zeng Z, Liu X, Zhu X, Bhuiyan M (2020) A novel load balancing and low response delay framework for edge-cloud network based on SDN. *IEEE Internet Things J* 7(7):5922–5933
- Zhang N, Yang P, Ren J, Chen D, Yu L, Shen X (2018) Synergy of big data and 5G wireless networks: opportunities, approaches, and challenges. *IEEE Wirel Commun* 25(1):12–18
- Li T, Liu W, Wang T, Zhao M, Li X, Ma M (2020) Trust data collections via vehicles joint with unmanned aerial vehicles in the smart internet of things. *Trans Emerg Telecommun Technol*. <https://doi.org/10.1002/ett.3956>
- Huang M, Zhang K, Zeng Z, Wang T, Liu Y (2020) An AUV-assisted data gathering scheme based on clustering and matrix completion for smart ocean. *IEEE Internet Things J*. <https://doi.org/10.1109/JIOT.2020.2988035>
- Harb H, Makhoul A (2019) Energy-efficient scheduling strategies for minimizing big data collection in cluster-based sensor networks. *Peer-to-Peer Netw Appl* 12(3):620–634
- Li T, Liu A, Zhang S, Wang T, Xiong N (2020) A trustworthiness-based vehicular recruitment scheme for information collections in distributed networked systems. *Inf Sci*. <https://doi.org/10.1016/j.ins.2020.07.052>
- Jiang B, Huang G, Wang T, Gui J, Zhu X (2020) Trust based energy efficient data collection with unmanned aerial vehicle in edge network. *Trans Emerg Telecommun Technol*. <https://doi.org/10.1002/ett.3942>
- Xiong H, Zhang H, Sun J (2018) Attribute-based privacy-preserving data sharing for dynamic groups in cloud computing. *IEEE Syst J* 13(3):2739–2750
- Ren Y, Zeng Z, Wang T, Zhang S, Zhi G (2020) A trust-based minimum cost and quality aware data collection scheme in P2P network. *Peer-to-Peer Netw Appl*. <https://doi.org/10.1007/s12083-020-00898-2>
- Li X, Ma J, Wang W, Xiong Y, Zhang J (2013) A novel smart card and dynamic ID based remote user authentication scheme for multi-server environment. *Math Comput Model* 58(1–2):85–95
- Tan J, Liu W, Wang T, Zhao M, Liu A, Zhang S (2020) A high-accurate content popularity prediction computational modelling for mobile edge computing by using matrix completion technology. *Trans Emerg Telecommun Technol*. <https://doi.org/10.1002/ett.3871>
- Wang X, Liu Z, Gao T, Zheng X, Dang Z, Shen X (2019) A near-optimal protocol for the grouping problem in RFID systems. *IEEE Trans Mob Comput*. <https://doi.org/10.1109/TMC.2019.2962125>
- Teng H, Ota K, Liu A, Wang T, Zhang S (2020) Vehicles joint UAVs to acquire and analyze data for topology discovery in large-scale IoT systems. *Peer-to-Peer Netw Appl*. <https://doi.org/10.1007/s12083-020-00879-5>
- Hafeez KA, Zhao L, Ma B, Mark JW (2013) Performance analysis and enhancement of the DSRC for VANET's safety application. *IEEE Trans Vehicular Technol* 62(7):3069–3083
- Sultana A, Zhao L, Fernando X (2017) Efficient resource allocation in device-to-device communication using cognitive radio technology. *IEEE Trans Veh Technol* 66(11):10024–10034
- Chen M, Wang T, Ota K, Dong M, Zhao M, Liu A (2020) Intelligent resource allocation management for vehicles network: an A3C learning approach. *Comput Commun* 151:485–494
- Huang M, Liu A, Wang A, Liu A, Zhang S (2020) A cloud-MEC collaborative task offloading scheme with service orchestration. *IEEE Internet Things J* 7(7):5792–5805
- Li X, Xiong Y, Ma J, Wang W (2012) An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards. *J Netw Comput Appl* 35(2):763–769
- Liu Q, Hou P, Wang G, Peng T, Zhang S (2019) Intelligent route planning on large road networks with efficiency and privacy. *J Parallel Distributed Comput* 133:93–106
- Li X, Niu J, Khan MK, Liao J (2013) An enhanced smart card based remote user password authentication scheme. *J Netw Comput Appl* 36(5):1365–1371
- Ren Y, Wang T, Zhang S, Zhang J (2020) An intelligent big data collection technology based on micro mobile data centers for crowdsensing vehicular sensor network. *Pers Ubiquit Comput*. <https://doi.org/10.1007/s00779-020-01440-0>
- Zeng D, Gu L, Pan S, Cai J, Guo S (2019) Resource management at the network edge: a deep reinforcement learning approach. *IEEE Netw* 33(3):26–33
- Gui J, Hui L, Xiong N, Wu J (2020) Improving spectrum efficiency of cell-edge devices by incentive architecture applications with dynamic charging. *IEEE Trans Ind Inform*. <https://doi.org/10.1109/TII.2020.2987089>
- Liu Q, Tian Y, Wu J, Peng T, Wang G (2019) Enabling verifiable and dynamic ranked search over outsourced data. *IEEE Trans Serv Comput*. <https://doi.org/10.1109/TSC.2019.2922177>
- Li T, Zhao M, Wong KKL (2020) Machine learning based code dissemination by selection of reliability mobile vehicles in 5G networks. *Comput Commun* 152:109–118
- Liu X, Lin P, Liu T, Wang T, Liu A, Xu W (2020) Objective-variable tour planning for mobile data collection in partitioned sensor networks. *IEEE Trans Mob Comput*. <https://doi.org/10.1109/TMC.2020.3003004>
- Liang W, Huang W, Long J, Zhang K, Li K-C, Zhang D (2020) Deep reinforcement learning for resource protection and real-time detection in IoT environment. *IEEE Internet Things J* 7(7):6392–6401
- Kuang Z, Li G, Zhang L, Zhou H, Li C, Liu A (2020) Energy efficient mode selection, base station selection and resource allocation algorithm in D2D heterogeneous networks. *Peer-to-Peer Netw Appl*. <https://doi.org/10.1007/s12083-020-00915-4>



31. Zhang Y, Xu C, Li H, Yang K, Zhou J, Lin X (2018) Healthdep: an efficient and secure deduplication scheme for cloud-assisted ehealth systems. *IEEE Trans Ind Inform* 14(9):4101–4112
32. Xie K, Li X, Wang X, Cao J, Xie G, Wen J, Qin Z (2018) On-line anomaly detection with high accuracy. *IEEE/ACM Trans Networking* 26(3):1222–1235
33. Peng M, Liu W, Wang T, Zeng Z (2020) Relay selection joint consecutive packet routing scheme to improve performance for wake-up radio-enabled WSNs. *Wirel Commun Mob Comput*. <https://doi.org/10.1155/2020/7230565>
34. Chen CY, Guan ZH, Chi M, Wu Y, Liao RQ, Jiang XW (2017) Fundamental performance limitations of networked control systems with novel trade-off factors and constraint channels. *J Franklin Inst* 354(7):3120–3133
35. Chen CY, Hu B, Guan ZH, Chi M, He DX (2016) Optimal tracking performance of control systems with two-channel constraints. *Inf Sci* 374: 85–99
36. Guan ZH, Chen CY, Feng G, Li T (2013) Optimal tracking performance limitation of networked control systems with limited bandwidth and additive colored white Gaussian noise. *IEEE Trans Circuits Syst I: Reg Papers* 60(1):189–198
37. Huang S, Liu A, Zhang S, Wang T, Xiong N (2020) BD-VTE: a novel baseline data based verifiable trust evaluation scheme for smart network systems. *IEEE Trans Netw Sci Eng*. <https://doi.org/10.1109/TNSE.2020.3014455>
38. Wang T, Luo H, Zeng X, Yu Z, Liu A, Sangaiah AK (2020) Mobility based trust evaluation for heterogeneous electric vehicles network in smart cities. *IEEE Trans Intell Transp Syst*. <https://doi.org/10.1109/TITS.2020.2997377>
39. Wang T, Liang Y, Yang Y, Xu G, Peng H, Liu A, Jia W (2020) An intelligent edge-computing-based method to counter coupling problems in cyber-physical systems. *IEEE Netw* 34(3):16–22
40. Liang W, Fan Y, Li K-C, Zhang D, Gaudiot J (2020) Secure data storage and recovery in industrial blockchain network environments. *IEEE Trans Ind Inform* 16(10):6543–6552
41. Sarang S, Drieberg M, Awang A, Ahmad R (2018) A QoS MAC protocol for prioritized data in energy harvesting wireless sensor networks. *Comput Netw* 144:141–153
42. Wang T, Cao Z, Wang S, Wang J, Qi L, Liu A, Xie M, Li X (2020) Privacy-enhanced data collection based on deep learning for internet of vehicles. *IEEE Trans Ind Inform* 16(10):6663–6672
43. Wang T, Qiu L, Sangaiah AK, Liu A, Bhuiyan MZA, Ma Y (2020) Edge-computing-based trustworthy data collection model in the internet of things. *IEEE Internet Things J* 7(5):4218–4227
44. Li X, Niu JW, Ma J, Wang WD, Liu CL (2011) Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart card. *J Netw Comput Appl* 34(1):73–79
45. Ge J, Liu B, Wang T, Yang Q, Liu A, Li A (2020) Q-learning based flexible task scheduling in a global view for internet-of-things. *Trans Emerg Telecommun Technol*. <https://doi.org/10.1002/ETT.4111>
46. Zhang N, Cheng N, Gamage A, Zhang K, Mark JW, Shen X (2015) Cloud assisted HetNets toward 5G wireless networks. *IEEE Commun Mag* 53(6):59–65
47. Gui J, Dai X, Deng X (2020) Stabilizing transmission capacity in millimeter wave links by Q-learning-based scheme. *Mob Inf Syst* 7607316:17
48. Zhuo C, Luo S, Gan H, Hu J, Shi Z (2020) Noise-aware DVFS for efficient transitions on battery-powered IoT devices. *IEEE Trans Comput Aided Des Integr Circuits Syst* 39(7):1498–1510
49. Wang J, Wang F, Wang Y, Wang L, Qiu Z, Zhang D, Lv Q (2019) HyTasker: hybrid task allocation in mobile crowd sensing. *IEEE Trans Mob Comput*. <https://doi.org/10.1109/TMC.2019.2898950>
50. Xie K, Li X, Wang X, Xie G, Wen J, Cao J, Zhang D (2017) Fast tensor factorization for accurate internet anomaly detection. *IEEE/ACM Trans Networking* 25(6):3794–3807
51. Ye W, Heidemann J, Estrin D (2002) An energy-efficient MAC protocol for wireless sensor networks. In: Proceedings. Twenty-first annual joint conference of the IEEE computer and communications societies, November 07, 2002. <https://doi.org/10.1109/INFCOM.2002.1019408>
52. Dam TV, Langendoen K (2003) An adaptive energy-efficient MAC protocol for wireless sensor networks. In: Proceedings of the 1st international conference on embedded networked sensor systems, pp 171–180. <https://doi.org/10.1145/958491.958512>
53. Gu K, Wu N, Yin B, Jia W (2020) Secure data query framework for cloud and fog computing. *IEEE Trans Netw Serv Manag* 17(1):332–345
54. Gu K, Zhang W, Lim S-J, Sharma PK, Al-Makhadmeh Z, Tolba A (2020) Reusable mesh signature scheme for protecting identity privacy of IoT devices. *Sensors* 20(758):1–23
55. Gu K, Dong X, Wang L (2020) Efficient traceable ring signature scheme without pairings. *Adv Math Commun* 14(2):207–232
56. Li Z, Liu Y, Liu A, Wang S, Liu H (2018) Minimizing convergencast time and energy consumption in green internet of things. *IEEE Trans Emerg Top Comput*. <https://doi.org/10.1109/TETC.2018.2844282>
57. Syed I, Shin SH, Roh BH, Adnan M (2017) Performance improvement of QoS-enabled WLANs using adaptive contention window backoff algorithm. *IEEE Syst J* 12(4):3260–3270
58. Zhang J, Ren F, Gao S, Yang H, Lin C (2014) Dynamic routing for data integrity and delay differentiated services in wireless sensor networks. *IEEE Trans Mob Comput* 14(2):328–343

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)