A two-channel slotted sense multiple access protocol for timely and reliable data transmission in industrial wireless sensor networks

Md Abul Kalam Azad, Chi Trung Ngo and Hoon Oh

Abstract
In designing a real-time MAC protocol for wireless sensor networks, it is desirable to shorten the data acquisition cycle time to satisfy more applications with a tighter time constraint in data collection. To achieve this, one often tries to improve throughput by employing a slot reuse technique. Meanwhile, the slotted sense multiple access protocol that uses a sharable slot was proven to be advantageous for timely and reliable data delivery. However, since the nodes at lower tree levels process more packets or bigger packets with data aggregation, they need a bigger sharable slot. Thus, it is not easy to employ a slot reuse technique due to the size variation of sharable slots. To tackle this problem, we try to reduce the size of an aggregated packet by limiting the number of packets for aggregation using two channels. If every node is allowed to send a single packet during data acquisition cycle time, an equal-sized sharable slot can be allocated to every tree level. It is shown that the proposed approach can reduce data acquisition cycle time significantly while maintaining good reliability in data transmission compared with slotted sense multiple access and also reduce energy consumption greatly compared with another multi-channel MAC protocol.

Keywords
Slotted sense multiple access, slot reuse, multiple channels, real time, sharable slot, wireless sensor networks

Date received: 16 November 2016; accepted: 17 December 2019
Handing Editor: Carlos Calafate

Introduction
Due to the low cost and the ease of installation and maintenance, wireless sensor networks (WSNs) can be used for data and command transmission in industrial applications.1 One important application is a safety monitoring and control system (SMOCS) in which a server collects data from sensor devices or nodes in work fields, judges whether or not the field involves danger based on the analysis of those data, and sends a warning message to workers about danger, if any.2 This application not only requires a reliable data delivery, but also constrains a data acquisition cycle time (DACT), defined as the time required for every node to send one data packet to a sink, to satisfy the temporal requirements of data. To improve DACT, this article extends the slotted sense multiple access (SSMA) protocol3 by enabling the reuse of a sharable slot with two channels.

Many protocols based on time division multiple access (TDMA) were proposed to improve throughput as well as reliability in data transmission. In terms of a
slot reuse, they can be classified into three broad groups: a spatial slot reuse, a channel-assisted slot reuse, and an opportunistic slot reuse. A slot can be spatially reused if two transmitting nodes are sufficiently distanced not to interfere with each other, as used by some TDMA-based MAC protocols. In TreeMAC, two nodes reuse a slot if they are three levels away from each other in a tree topology; according to some studies, they suffer from an irregular interference because the interference range is always farther than the transmission range. The weighted incremental ranking for convergecast with aggregation scheduling (Wires) protocol tries to construct a balanced shortest path tree that distributes load among nodes to reduce the transmission delay of data packets to a sink. This approach suffers from the waste of slot time since every node is assigned a slot big enough to accommodate the packets delivered from the largest possible number of descendant nodes. Z-MAC combines the advantages of carrier sense multiple access (CSMA) and TDMA to maximize channel utilization. It allocates slots to every node such that two-hop neighbors are not assigned the same slot in order to prevent interference. This protocol is not free from the irregular interference problem either.

With the channel-assisted slot reuse, nodes can reuse a slot by having different channels. In TFMAC, two-hop neighbors are allowed to reuse the same slot using two channels; however, as the number of allocated channels increases, it requires more slots per frame for correct scheduling. Y-MAC allows each node to own an exclusive receiving slot within two-hop neighborhood and permits all potential senders to contend for the receiving slot. However, the protocol fails to avoid the hidden terminal problem despite it uses a preamble-based channel sensing to suppress competing transmissions. A slot in MC-LMAC consists of a control period and a data transmission period (DTP). During the control period, all nodes switch to a common channel so that a sender can notify the receiver of the incoming packet and the channel on which the packet will be transferred. In this approach, one channel has to be solely reserved for negotiation. MuChMAC utilizes a pseudo random number generator to synchronize a frequency hopping sequence between a sender and a receiver for the selected slot, and allows parallel transmission using both the spatial slot reuse and the channel-assisted slot reuse. The authors in Pyeon and Yoon propose the multi-path pipeline transmission protocol that reuses time slots using multiple channels, and also tries to distribute transmission load among multiple established paths. The protocol could reduce transmission delay and balance energy consumption of nodes; however, it focuses on transmitting a large-sized data from source to destination. The authors in Swain et al. propose an advertisement-based multi-channel MAC (Adv-MMAC) protocol that uses two types of channels, one advertisement channel (ACH) and multiple data channels (DCHs). If a node acquires ACH through contention, it sends an advertisement message to a receiver using the acquired ACH. If it receives the acknowledgment (ACK) from the receiver, it sends data using any available free DCH. This approach can improve throughput with parallel transmission; however, every node can participate in one route, thereby limiting parallel transmission and the frequent flooding of the advertisement message can increase energy consumption.

Some approaches reuse slots opportunistically such that two or more nodes transmit data packets simultaneously if they do not interfere with each other through the channel contention process that often relies on the carrier sense multiple access with collision avoidance (CSMA/CA) protocol. In the SSMA protocol, a sharable slot is allocated to each tree level and the nodes at the same level compete for data transmission using CSMA. Then, if two sending nodes do not interfere with each other, they can transmit data simultaneously, thereby reusing the sharable slot opportunistically. However, since a node allocates a sharable slot of varying size depending on its tree level, it is not easy to reuse the sharable slots spatially. Another hybrid MAC (HyMAC) protocol combines TDMA and CSMA/CA with the grouping of nodes. One distinct time slot is allocated to each group and the nodes in the same group compete for data transmission using CSMA/CA, thus reducing the possibility of collision. In this approach, a node can adjust its sleep period dynamically by considering network density, collision status, and channel failure, to save energy. This approach also allows an opportunistic slot reuse within the same group.

According to the discussion so far, even though the contemporary protocols improve throughput reusing slots in various ways, they suffer from an irregular interference, an inefficiency in bandwidth utilization, the complexity of slot scheduling, and so on. SSMA suffers from a relatively long DACT due to the difficulty of its slot reuse. In this article, the proposed protocol using two channels allows every node to use a sharable slot of an identical size regardless of its tree level, thereby enabling the use of all three-slot reuse techniques. Two channels allow nodes at every other level to transmit packets in parallel. Thus, the number of packets that a node has to process at a time is significantly reduced since it can finish transmitting its own data before it receives the packets produced by the nodes at two higher levels. Therefore, if a node aggregates all the received packets into a single packet whose size is not greater than one MAC protocol data unit (MPDU), it can transmit only a single packet within one sharable slot. In consequence, different sharable
slots of an identical size can be assigned to different tree levels. Furthermore, a slot scheduling can be performed in a distributed manner as SSMA does. According to simulation, the protocol could improve DACT by 40% over its ancestor SSMA. It could also improve packet delivery rate by reducing the number of data transmissions and energy efficiency by reducing DACT.

The rest of the article is organized as follows. In section “Background,” we give a background for the proposed approach. In section “Two-channel SSMA protocol,” we formally describe the proposed protocol. A mathematical and experimental evaluation of the proposed protocol is given in section “Protocol evaluation,” followed by concluding remarks in section “Conclusion.”

Background

Network model

In the industry fields such as the construction site of a liquefied petroleum gas (LPG) ship, workers can be exposed to hazardous situations due to toxic gas or smoke from firing or explosion, or physical injuries from incautious works. Since the signal transmission in the environment is significantly affected by high signal attenuation and multi-path fading effect, a sensor network of 25–40 nodes with a transmission range of 10–25 m is required in spite of relatively small working space for the implementation of an SMOSCS.

A WSN consists of one data collection and control server (hereafter, referred to as a sink) and a number of sensor communication devices (SCDs) (hereafter, referred to as a sensor node or a node simply), each including one or more sensor modules. Every node is required to send one packet to a sink periodically. Network topology can be changed due to ambulant workers who carry nodes or due to transient link failure by multi-path fading effects, battery depletion, or the intervention of some obstruction, and so on. Sensor nodes form a tree originating from a sink. A node is said to be a tree-node if it belongs to a tree. Otherwise, it is an orphan-node. Two nodes that can directly and mutually communicate with each other are said to have a link or an ordinary link. Especially, a link between a node and its parent is called a tree-link.

Figure 1 shows a tree topology of five levels where the solid lines and the dashed lines indicate tree-links and links, respectively. A sink indicated by a thick circle has four children, enumerated as 1, 2, 3, and 4.

Motivation

The SSMA protocol allocates a sharable slot to each tree level and uses CSMA for data transmission within a sharable slot. Data transmission is performed from the highest tree level to the lowest tree level (a sink) progressively. Every node competes only with its adjacent neighbors at the same level within the same sharable slot, thus reducing the degree of competition. Assuming that nodes are distributed randomly, the number of competing nodes can be assumed to be approximately same regardless of the location. However, a node has to process all data packets generated by its descendants. Therefore, a sharable slot is assigned to each tree level such that the size of a sharable slot decreases exponentially as tree level increases. Therefore, SSMA undergoes some difficulties in exploiting the slot reuse technique since it uses the sharable slots of different sizes from level to level. Suppose that in SSMA, a node can aggregate all packets generated by its descendants into a single packet. Thereby sending only one packet by a node is sufficient during one DACT. Then, a sharable slot of an identical size can be allocated to every level. However, if a network size is large, a certain node can have too many descendants, creating a big aggregated packet that cannot be transmitted at one time.

The use of two channels can resolve the problem. Suppose that nodes at level $l$ and the nodes at level $l + 2$ can transmit packets simultaneously within the same sharable slot. Then, the nodes at level $l$ will transmit their own packets before they receive the packets from their children (at level $l + 1$) and its grandchildren (at level $l + 2$). Consider a subtree whose root is $s$ as the child of a sink. Then, since all the packets generated at two paired levels at the subtree can gather at node $s$ to be forwarded to a sink, the maximum number of packets that a node has to process at one time is limited to the number of nodes at two paired levels in the subtree.

Figure 2 illustrates the reduction of DACT by parallel transmission compared to DACT by SSMA for a single child tree of depth 5 in Figure 2(a). With SSMA in Figure 2(b), every node processes its own packet and all the packets received by its descendants one by one, starting from node $D$ at level 5. With SSMA using two channels, named TC-SSMA, as shown in Figure 2(c),
node $B$ can transmit its own data packet $b$ while node $C$ receives packet $d$ from node $D$ in the first slot since they use different channels. Then, node $C$ has to process packet $d$ received from its child and its own packet $c$, requiring two slots. Now, node $B$ only needs to process two packets, $c$ and $d$, from its child since it had already transmitted its own packet in the first slot. Therefore, the maximum number of packets that a node has to process is two in this topology. With TC-SSMA using aggregation, named TC-SSMA$^a$, a node can aggregate two packets into one before transmission, as shown in Figure 2(d), thereby reducing DACT greatly.

With TC-SSMA, the maximum number of packets that a node has to process corresponds to the number of nodes that belongs to two adjacent levels for any subtree of a sink. In Figure 1, if nodes at odd levels transmit data first, node 1 can have maximum of five packets generated by nodes 12, 13, 14, 15, and 22. If five packets are aggregated, node 1 sends only one packet within one sharable slot. If the size of an aggregated data is greater than that of the MPDU, it can be transmitted twice after it is segmented into two packets. The size of the aggregated data can be further reduced in real network deployment, considering that the data are generally spatial-correlated$^{16}$ and some data can be filtered out.

**Notations**

We use some notations as follows:

- $S$: a set of nodes participating in a considered network;
- $L$: the maximum level of a tree;
- $\text{level}(i)$ or $l$: the level of node $i$;
- $\text{nNbrs}(i)$: a set of neighbors of node $i$;
- $C(i)$: a set of children of node $i$;
- $D(i)$: a set of descendants of node $i$;
- $i(X)$: the transmission time of message $X$;
- $\text{slotLen}$: the length in time of a slot;
- $\text{sslotLen}$: the length in time of a sharable slot;
- $\text{PCL}(i)$: a parent candidate list of node $i$ is a list of nodes such that for every node $j$ in $\text{PCL}(i)$, level($i$) is equal to level($j$) + 1 and link ($i, j$) is reliable.

### Two-channel SSMA protocol

#### Protocol structure

The protocol structure consists of the initial construction period (ICP) and the repeating cycle that includes the DTP and the control transmission period (CTP) as illustrated in Figure 3. During ICP, the participating nodes perform time synchronization, tree construction, and channel-slot scheduling using a common channel.
Among the existing approaches for time synchronization, the flooding time synchronization protocol (FTSP) is employed since it is suitable for the operational characteristics of the proposed MAC protocol. The channel-slot scheduling distributes scheduling parameters to the participating nodes so that every node can allocate a channel and a sharable slot according to its tree level. During DTP and CTP, data packets are delivered from sensor nodes to a sink and command messages are delivered from a sink to sensor nodes, respectively. In this article, only a data transmission protocol is discussed. For the detail discussion on the reliable broadcast protocol for command transmission, refer to the study by Van Vinh and Oh.

**Reliable tree construction and time synchronization**

A reliable tree is constructed by estimating link quality. The protocol adopts the following compound link-quality metric, \( linkq(t) \), that combines received signal strength indicator (RSSI) and link quality indication (LQI) associated with a received packet:

\[
linkq(t) = \begin{cases} 
1, & \text{if } (RSSI(m, t) > RSSITh) \quad \text{and} \quad (LQI(m, t) < LQITh) \\
0, & \text{otherwise}
\end{cases}
\]

where \( linkq(t) = 1 \) indicates that the corresponding link is good at time \( t \), and \( RSSI(m, t) \) and \( LQI(m, t) \) are the RSSI and the LQI of message \( m \) at time \( t \), respectively. \( RSSITh \) and \( LQITh \) are the threshold values of RSSI and LQI, respectively. Based on this quality function, every node \( i \) maintains a reliable neighbor set, RNS\( (i) \), which records the link quality of its neighbors. During tree construction, a node, say \( i \), can also maintain a parent candidate list, PCL\( (i) \), such that for any node \( j \) in PCL\( (i) \), link\( (i, j) \) is reliable and the level of node \( j \) is lower than that of node \( i \) by unity.

For time synchronization, our protocol relies on a local time synchronization scheme to remove a run-time overhead while it uses the FTSP only once during ICP. It has been known that FTSP secures an accuracy of less than 30 ms with a resynchronization period of 30 s. Then, a node performs time synchronization locally by making use of the control messages such as CTS, ACK, or RTR used in our protocol. A sink piggybacks its global time on RTR while other tree-nodes include their global time in CTS or ACK. Upon receiving the control messages, a node updates its global time.

**Channel and sharable slot allocation**

Based on a constructed tree and a synchronized time, a channel and sharable slot scheduling is performed. Unlike SSMA, the proposed TC-SSMA can allocate the sharable slots of an identical size to different levels by means of parallel transmission and aggregation. Thus, a slot reuse technique can be used. We discuss data transmission method and the functions for the allocation of channels and slots that is based on both a spatial slot reuse and a channel-assisted slot reuse.

**Channel allocation**

**Channel-assisted slot reuse technique.** A node cannot receive and transmit data simultaneously. Thus, if two adjacent levels use the same channel for transmitting and receiving, \( L/2 \) channels are sufficient to achieve maximum parallelism among different levels considering a tree of depth \( L \). Then, during the first sharable slot, the nodes at even levels, \( I = 2, 4, \ldots, (2 \times \lfloor L/2 \rfloor - 2), (2 \times \lfloor L/2 \rfloor) \), can transmit data packets simultaneously to their respective parents in one upper odd levels using different channels. Note that if \( L \) is odd, there is no node at level \( (2 \times \lfloor L/2 \rfloor + 2) \). Nodes at the same even level compete for data packet transmission using CSMA within the same sharable slot. When the first sharable slot ends, all nodes at the highest even level, \( 2 \times \lfloor L/2 \rfloor \) complete data transmission during the current DACT and thus go to sleep till the end of DACT. During the second sharable slot, the nodes at odd levels, \( I = 3, 5, \ldots, (2 \times \lfloor L/2 \rfloor + 1) \), start data transmission in parallel. When the second sharable slot ends, the nodes at the highest odd level, \( 2 \times \lfloor L/2 \rfloor + 1 \), complete data transmission and go to sleep till the end of DACT. Note that if \( L \) is even, there is no node at level \( 2 \times \lfloor L/2 \rfloor + 1 \). In this way, whenever one sharable slot ends, the nodes at one level complete data transmission.

**Spatial slot reuse technique.** Suppose that, for \( k \geq 0 \), levels \( 4k + 1 \) and \( 4k + 2 \) use channel \( x \), and levels \( 4k + 3 \) and \( 4k + 4 \) use channel \( y \). Then, we assume that levels \( 4(k + 1) + 1 \) and \( 4(k + 1) + 2 \) can reuse the same channel \( x \) since a transmitting node at level \( 4k + 2 \) (transmitters of the former) and a receiving node at level \( 4(k + 1) + 1 \) are three hops away. This is illustrated in Figure 4 where node \( F \) can reuse the same channel as of node \( B \) in data transmission by means of

![Figure 4](image-url)
spatial slot reuse. TreeMAC assumes that a signal does not reach other nodes that are two-hop away from its location. According to one study in Malhotra et al., it is verified that the interference of signal from three-hop away nodes is negligible.

According to the principle of channel allocation, the functions to determine a receiving channel and a transmitting channel for a node is developed. Let \( Q(x) \) and \( R(x) \) represent the Quotient and Remainder of integer \( x \) with divisor 2.

\[
R(x) = x \mod 2, \text{ and } Q(x) = x \div 2
\] (1)

The addition of \( R(l) \) and \( Q(l) \) falls into either an even integer set or an odd integer set. By utilizing this property, the nodes with even level are given one channel while the nodes with odd level are given another channel. Thus, node \( i \) can determine the receiving channel, \( CH_i^{RX} \) and the transmitting channel, \( CH_i^{TX} \) as follows

\[
CH_i^{RX} = ((Q(l_i) + R(l_i)) \mod 2) + 1
\]

\[
CH_i^{TX} = ((Q(l_i - 1) + R(l_i - 1)) \mod 2) + 1, \ l_i \neq 1
\] (2)

where the use of \( l_i - 1 \) in \( CH_i^{TX} \) indicates that when node \( i \) is in transmitting mode, it always follows the channel of its parent level. In equation (2), the addition of 1 allows a channel number to become 1 or 2, instead of 0 or 1. Note that a sink does not need a transmitting channel.

**Sharable slot allocation.** Let \( sslen \) and \( W_1 \) be the size of one sharable slot and DACT, respectively. During tree construction process, a sink disseminates these two parameter values to all nodes. Then, every node can calculate the number of sharable slots, \( nsslots \), within \( W_1 \) as follows

\[
nsslots = \left\lfloor \frac{W_1}{sslen} \right\rfloor
\] (3)

Then, each sharable slot can be sequentially numbered from 1 to \( nsslots \). Let \( ssn \) denote a sharable slot number. A node can use each sharable slot for one of sending, receiving, and idle. Without loss of generality, suppose that a sink (at level 1) receives data packets from the nodes at level 2 during the first sharable slot (with \( ssn = 1 \)) as illustrated in Figure 5. Then, the nodes at even levels and at odd levels can use the first slot as sending and receiving activities, respectively. Consequently, the nodes at even levels alternate between sending activity and receiving activity, and those at odd levels do the same in a reverse way.

Since a sink starts with a receiving sharable slot and ends with the receiving sharable slot, the required number of total sharable slots becomes an odd number. Accordingly, the required number of sharable slots, \( nsslots \), can be adjusted as follows

\[
nsslots = nsslots - 1 \text{ if } (nsslots \mod 2) = 0
\] (4)

Next, let us calculate the total number of sharable slots that each node needs for data transmission. Whenever each of the sharable slots is used, the nodes at one level starting from the highest level finish data transmission. Finally, the nodes at level 2 finish data transmission in the last slot with \( ssn = nsslots \). Therefore, the maximum supportable level, \( MaxSupportLevel \), with \( nsslots \), is given as follows

\[
MaxSupportLevel = nsslots + 1
\] (5)

The nodes at the \( MaxSupportLevel \) (is equal to \( L \) in Figure 5) need only one sending sharable slot without requiring any sharable slot for forwarding. It is obvious that the nodes at one lower level need two sharable slots, one for receiving and another for sending. Since the use of one sharable slot results in the completion of data transmission for one level, \( SD(i) \) as the sharable slot demand of node \( i \) is given as follows

\[
SD(i) = nsslots - level(i) + 2, \ level(i) \neq 1
\] (6)

For convenience, let (receiving, sending, idle) be enumerated as (0, 1, 2). A node \( i \) uses the following function \( sslotUse(i) \) to determine the use of a sharable slot by its \( ssn \)

For \( level(i) = 1, sslotUse(i) = \begin{cases} 0, & \text{if } (ssn \mod 2) = 1 \\ 2, & \text{if } (ssn \mod 2) = 0 \end{cases} \)

For \( level(i) \neq 1, \ sslotSDN(i) = \begin{cases} (level(i) + ssn) \mod 2, & \text{if } ssn \leq SD(i) \\ 2, & \text{Otherwise} \end{cases} \) (7)
Figure 5 shows a tree model of level $L$ and the schedule of two channels and sharable slots according to equations (2) and (7).

**Data transmission time scheduling**

Once tree construction and synchronization are finished, every node knows its slot demand and the channel number for each slot. Then, a node schedules data transmission time in a distributed manner. Let us denote the start time of DACT by $sTime$. Then, assuming that time is globally synchronized, node $i$ derives $RxTime(i)$ and $TxTime(i)$ that indicates the start times of receiving slot and transmitting slot, respectively, as follows.

If $ssn \leq SD(i)$ and \( l_i \mod 2 = 0 \)

\[ TxTime(i) = sTime + \left(2 \times \frac{ssn}{2} - 2\right) \times sslotLen \]

\[ RxTime(i) = sTime + \left(2 \times \frac{ssn}{2} - 1\right) \times sslotLen \]  \hspace{1cm} (8)

If $ssn \leq SD(i)$ and \( l_i \mod 2 \neq 0 \)

\[ TxTime(i) = sTime + \left(2 \times \frac{ssn}{2} - 1\right) \times sslotLen \]

\[ RxTime(i) = sTime + \left(2 \times \frac{ssn}{2} - 2\right) \times sslotLen \]  \hspace{1cm} (9)

where $sTime = GT + ICP$, and $GT$ is a globally synchronized time. Node $i$ goes to sleep if it has no data packet to send or forward, or $ssn > SD(i)$.

A sink starts $ICP$ at some arbitrary time, referred to as global time ($GT$) as shown in Figure 6. Then, suppose that node $x$ can finish two processes, tree construction and time synchronization, before time, $GT + t_x < GT + ICP$ where $t_x$ is the time elapsed until node $x$ finishes the processes. Then, every node can start DTP at $sTime = GT + ICP$. Based on this $sTime$, every node $x$ can obtain $RxTime(x)$ and $TxTime(x)$ using equations (8) and (9) in a distributed manner.

**Reliable data transmission**

In the industrial environment in which the channel condition is time-varying, the success of data transmission cannot be guaranteed. Thus, our approach includes two features to improve the reliability of data transmission. First, once a node detects the failure of transmission, it changes its parent using its parent candidate list before retrial. Second, the protocol employs the CSMA mechanism using the MAC control messages such as RTS, CTS, and ACK for data transmission within a sharable slot. In CSMA, if a node detects a channel is idle, it takes a random delay, $rdelay$, before it transmits RTS to avoid collision.

\[ rdelay = \text{rand}(0, \text{MaxRndDelayCnt}) \times \text{delaySlot} \]  \hspace{1cm} (10)

where $\text{MaxDelayCnt}$ indicates the maximum delay count and $\text{delaySlot}$ indicates the amount of time elapsed from the time when a node sends one bit at the time when its neighbor detects the bit.

A detailed data transmission process is given in Algorithm 1. The procedure starts with a flag, $\text{succeed}$ set to $False$. In line 3, node $x$ tries to send data with its primary parent, $p$. A node aggregates all data packets received from its children and its own data packet into one packet, $\text{aggrPacket}$. In line 5, function $\text{sendPacket()}$ uses CSMA/CA to send $\text{aggrPacket}$ to $p$. During this time, the control packets RTS, CTS, and ACK are used not only for reliable data transmission but also for time resynchronization between node $x$ and $P(x)$. If data transmission fails after two retires, a

![Figure 6. Data transmission time scheduling.](image-url)
node tries data transmission with a new candidate parent (line 10) until either it succeeds, or the sharable sending slot ends.

**Data aggregation**

According to our protocol that allows parallel transmission, nodes at level 2 complete the transmission of their own data during the first sharable slot with no aggregation. Therefore, the number of packets to be aggregated corresponds to the number of nodes that is located at two adjacent levels, but belongs to one of the subtrees of a sink, starting from level 3 up to level $L$.

For a subtree of a sink, let $maxAggrNodes$ be the maximum number of nodes at two adjacent levels. Then, it is given as follows

$$maxAggrNodes = \max\{(n_{i-1} + n_i) | i = 2 \times j, j = 2 \cdot \lceil L/2 \rceil\}$$

A node should be able to aggregate the number of data packets that corresponds to $maxAggrNodes$ at maximum. Suppose that every node produces a payload that consists of node address of 2 bytes and data of 4 bytes. Then, $size(packet)$ including the header and trailer of a packet can be given as follows

$$size(packet) = 18 + maxAggrNodes \times 6 \quad (11)$$

If the size of a packet goes over 127 bytes, it can be segmented into multiple packets for transmission. This may require the size of a sharable slot to be increased.

For example, consider the subtree whose root is node 1 in Figure 1. Seven nodes 5, 6, 7, 12, 13, 14, and 15 are located at levels 3 and 4. In this case, $size(packet)$ is 60 bytes from equation (11).

**Protocol evaluation**

**DACT analysis**

**DACTs for different protocols.** Let us examine DACT for different approaches. In TreeMAC,\(^4\) since a node is assigned one distinct frame that consists of three slots, DACT is given for a tree with $n$ nodes, as follows

$$DACT(TreeMAC) = (n \times 3) \times slotLen \quad (12)$$

In I-MAC,\(^2\) the size of $SF$ varies according to the level of a tree such that a node at level $l$ requires $(l - 1)$ slots to send one data packet to a sink. Thus, DACT is given as follows

$$DACT(I-MAC) = \sum_{l=2}^{L} (n_l \times (l - 1)) \times slotLen \quad (13)$$

In TC-SSMA, $SF$ is calculated easily since nodes at every level from 2 to $L$ require one sending sharable slot. Referring to Figure 5, the DACT of TC-SSMA for a tree of level $L$ is calculated as follows

$$DACT(TC - MAC) = (L - 1) \times slotLen \quad (14)$$

**Lower bound of sharable slot size.** In this section, we estimate the size of a sharable slot, $sslotLen$. A node has its neighbors over three levels; however, it only competes with its neighbors at the identical level when it wants to send data packet to its parent. Assuming that its neighbors are uniformly distributed over three levels, the lower bound of $sslotLen$ is expressed as follows

$$sslotLen \geq T_{\mu} \times \left(\frac{1}{3} \times nNbrs + 1\right) \quad (15)$$

where $nNbrs$ indicates the average number of neighbors of a node, and $T_{\mu}$ is the total time taken to successfully transmit one data packet in the worst case. In fact, a node can fail to receive $CTS$ due to the loss of $RTS$ or $CTS$ and be aware of this failure only if it does not receive $CTS$ by the end of $CTSTimeout$ period. Furthermore, it also can fail to receive $ACK$ due to the loss of $DATA$ or $ACK$. We assume that the protocol makes only one more trial after each failure. Then, the process of successful data transmission in the worst case is illustrated in Figure 7. Assuming that $t(ACK)$ is equal to $ACKTimeout$ and the propagation delay is negligible, $T_{\mu}$ is expressed as follows

$$T_{\mu} \geq MaxDelayCnt \times delaySlot + 2 \times \left(2 \times t(RTS + CTS) + CTSTimeout + t(DATA + ACK)\right) \quad (16)$$

**Estimation of tree depth and the number of neighbors.** Using the cumulative distribution function in Bettstetter et al.,\(^{23}\) we can estimate how many nodes are located at each level. The probability distribution of tree level, $Pr(level = l)$, is obtained with the transmission range of $R$ and the dimension of $a \times a$. Figure 8 shows $Pr(level = l)$ with $a = 100\, m$ and $R = 25\, m$ where the maximum tree level is bounded to 6.
Using the probability mass function in Lifang and Huimin et al., we can estimate the number of neighbors that a node can have. The probability distribution, $Pr(n Nbrs = k)$, is obtained with the transmission range of $R$, the dimension of $a \times a$, and the number of nodes, $n$. Figure 9 shows $Pr(n Nbrs = k)$ with $a = 100$ m, $R = 25$ m, and $n \in \{25, 30, 35, 40\}$ where $n Nbrs$ is about 12 in the worst case when $n = 30$.

**Determination of sharable slot size.** To determine the sharable slot size from equation (15), $T_{tx}$ has to be determined using equation (16). A node goes through the following six steps for successful message transmission: (1) generate a packet and transfer it from the micro control unit (MCU) to a radio buffer ($t_{tx}$); (2) turn the radio on ($t_{turnon}$); (3) perform a clear channel assessment (CCA) ($t_{CCA}$); (4) take the physical layer processing delay ($t_{ppd}$); (5) transmit the packet to the medium ($t_{tx}$); and (6) receive the packet and transfer it from the radio chip buffer to the MCU at the receiver side ($t_{tr}$).

In simulation, $t_{tx}$, $t_{turnon}$, and $t_{tr}$ can be ignored. Therefore, the transmission time of a message $m$ is given as follows

$$t(m) = t_{CCA} + t_{ppd} + t_{tx}(m) \quad (17)$$

Based on the 802.15.4 physical layer standard with a transmission rate of 250 Kbps (=0.032 ms/byte) and a synchronization header, PHY header, MAC header and footer of 11 bytes, we can calculate $t_{tx}$ in equation (17) as follows

$$t_{tx}(m) = 0.032 \times (11 + \text{size}(m)) \quad (18)$$

Since $t_{CCA} = 0.128$ ms and $t_{ppd} = 0.192$ ms, $t(m) = 0.672 + 0.032 \times \text{size}(m)$ (ms). Furthermore, size($RTS$) = size($CTS$) = size($ACK$) = 5 bytes and max(size($DATA$)) = 100 bytes. Thus, we get

$$t(RTS) = t(CTS) = t(ACK) = 0.832 \times (ms)$$
$$t(DATA) = 3.872 \times (ms) \quad (19)$$

With $MaxDelayCnt = 20$, $delaySlot = 0.32$ (ms), and $CTSTimeout = 0.832$ (ms), from equation (16), we get

$$T_{tx} \geq 24.128 \times (ms) \approx 25 \times (ms) \quad (20)$$

Substituting equation (20) into equation (15), we obtain

$$sslotLen \geq 25 \times \left( \frac{1}{3} \times n Nbrs + 1 \right) \times (ms) \quad (21)$$

For example, according to Figure 9, $nNbrs = 12$ in the network of $100 \times 100$ (m$^2$), $R = 25$ m, and $n = 35$. Substituting $nNbrs = 12$ into equation (21), we obtain $sslotLen \geq 125$ (ms).

**Comparison of DACTs.** Suppose that $sslotLen = 20$ ms. From Figure 8, $L = 6$ and $n_l$ can be calculated according to the change of the number of nodes. From equations (12)–(14), the SF sizes for the three protocols, TC-SSMA, I-MAC, and TreeMAC, can be calculated with varying the number of nodes.

In Figure 10, the SF sizes of three protocols are compared by changing the number of nodes from 25 to 40. It is shown that the SF size of TC-SSMA is much smaller than the other single-channel approaches overall since it allows parallel transmission. As the number of nodes increases, I-MAC and TreeMAC show a rapidly increasing curve compared to the slow increase in TC-SSMA. This is because the number of neighbors increases slowly with the increase in the number of nodes as shown in Figure 9.

**Performance evaluation**

**Simulation setup.** The QualNet simulator version 5.0.2 was used for simulation. The proposed protocol, TC-SSMA, is compared with its ancestor SSMA using single-channel and another protocol MC-LMAC using multiple channels under two different network scenarios. We used two performance metrics: packet delivery ratio (PDR) that is defined as the ratio of the total number of data packets successfully received at a sink to the total number of data packets sent by all sensor nodes, and energy consumption that is given as the average energy consumed by each sensor node for all sorts of data transmission activities during simulation time. Table 1 shows the key simulation parameters and values.

**Evaluation scenarios.** Simulation scenarios were made with two deployment patterns: a square dimension of $100 \times 100$ m$^2$ and a rectangular dimension of $50 \times 200$ m$^2$, shown in Figure 11(a) and (b), respectively. All sensor nodes were uniformly distributed, and
a sink is placed at the middle of the top of the area. Each node is required to transmit one packet of 127 bytes per DACT.

**Scenario I: static network with fading.** All sensor nodes are immobile; the links can be broken by the channel fading. The Ricean fading model was used with the driving parameter $K$ that is defined as the ratio between $P_{LOS}$ and $P_{other\_paths}$, where $P_{LOS}$ is the receiving power in a line-of-sight path, and $P_{other\_paths}$ is the receiving power in other paths. This implies that the smaller value of $K$ has the higher degree of the fading effect.

**Scenario II: dynamic network with mobile nodes.** One to three nodes are selected randomly and are allowed to move along the predefined tracks as illustrated in Figure 12. A mobile node has the maximum speed of 1 m/s that models the movement of workers who carry sensor nodes, and moves continuously without a pause time. The length of each arrowed line segment is about 25 m.

**Determination of DACT.** In order to examine the appropriate size of DACT, PDR was obtained with decreasing DACT from 2.0 to 0.6 s. Figure 13 shows that the PDR of TC-SSMA starts decreasing very slowly at DACT $< 0.8$ with $nNodes = 25$. This value can be compared with 0.6 from the analytical data in Figure 10. Note that the PDR of SSMA decreases at DACT $< 1.2$. This implies that TC-SSMA reduces DACT considerably. Also, with TC-SSMA, the DACT of 1.0 seems quite reasonable with $nNodes = 25$. Thus, with DACT = 1.0, the PDRs of TC-SSMA and SSMA

| Table 1. Simulation parameters and values. |
|--------------------------------------------|
| Parameter                                  | Value                                             |
| Number of nodes (nNodes)                   | 1 sink and 25 sensor nodes                        |
| Maximum size of a packet                   | 127 bytes                                         |
| Slot size (in MC-LMAC)                     | 50 ms                                             |
| Number of slots (in MC-LMAC)               | 12                                                |
| Tx range                                   | 25 m                                              |
| Channel frequency                          | 2.4 GHz                                           |
| Path loss model                            | 2-ray                                             |
| Shadowing model                            | Constant (mean = 4 dB)                            |
| Fading model                               | Ricean                                            |
| Noise factor                               | 10 dB                                             |
| Sensor energy model                        | MicaZ                                             |
| Current draw in MicaZ                      | Transmitting = 17.4 mA (0 dBm), Receiving = 19.7 mA, Idle = 20 μA, Sleep = 1 μA |
| Battery model                              | Linear                                            |
| Number of packets                          | 600 packets                                       |
were obtained by increasing the number of nodes. In Figure 14, it is shown that the PDR of TC-SSMA is sustained against the increase in $nNodes$ from 25 to 50 while that of SSMA decreases notably from over 30 nodes.

Evaluation with scenario I. In Figure 15, three protocols, SSMA, TC-SSMA, and MC-LMAC, are compared for PDR and energy consumption with $DACT$ fixed to 1.0 and $nNodes$ fixed to 25. It is shown that all protocols show much lower PDRs in the rectangular dimension that can produce the higher tree depth. However, TC-SSMA does not make a notable difference of PDRs for two dimensions. Furthermore, referring to the slopes of the PDR graphs, SSMA and TC-SSMA are much less sensitive against the increasing fading effect (i.e. the decrease in $K$) than MC-LMAC owing to the use of multiple parents with a sharable slot. In conclusion, TC-SSMA is highly dependable against high fading effect and various shapes of network dimension.

Referring to Figure 16, TC-SSMA has rather higher energy consumption than SSMA. It makes sense since nodes in TC-SSMA has to wake up at every receiving sharable slot to receive packets from their children, while in SSMA, nodes wake up only once at a sharable slot and go to sleep immediately after they succeed in data transmission. Note that MC-LMAC consumes much more energy than the other two protocols.

Evaluation with scenario II. In this simulation, we compare three approaches with varying the number of mobile nodes ($nMNodes$) while $DACT$, $nNodes$, and $K$ are fixed to 1.0, 25, and 12, respectively. In Figure 17, it is shown that SSMA and TC-SSMA using a sharable slot are not sensitive to the increase in $nMNodes$ compared
to MC-LMAC. The graphs in Figure 17 is quite similar to those in Figure 15, since both the decrease in $K$ and the increase in the number of mobile nodes increase link instability. For energy consumption, the graphs of Figure 18 are quite similar to those of Figure 16 since energy consumption is less affected by link instability.

**Conclusion**

With the use of two channels and packet aggregation, TC-SSMA is able to shorten DACT significantly by employing the spatial slot reuse and channel-assisted slot reuse techniques. It is proven by analysis that TC-SSMA could reduce DACT largely compared to I-MAC and TreeMAC. Thus, TC-SSMA will be able to satisfy more tightly time-constrained industrial applications. Simulations were performed with varying fading effects and different number of mobile nodes. It was shown by simulation that TC-SSMA achieves low energy consumption comparable to its single-channel ancestor, SSMA, and also that TC-SSMA far outperforms the multi-channel MAC protocol, MC-LMAC, and SSMA in DACT and PDR.

In future studies, the use of multiple channels can be considered within a sharable slot to further improve throughput. The channel hoping mechanism can also be considered for improving the reliability of data transmission. Finally, experiments with implementation have to be conducted to examine the applicability of the proposed protocol in real work fields.

**Declaration of conflicting interests**

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

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the 2018 Research Fund of University of Ulsan.

**ORCID iDs**

Chi Trung Ngo [https://orcid.org/0000-0003-3810-2611](https://orcid.org/0000-0003-3810-2611)

Hoon Oh [https://orcid.org/0000-0003-4257-4531](https://orcid.org/0000-0003-4257-4531)

**References**

1. Ajith Kumar AS, Ovsthus K and Kristensen LM. An industrial perspective on wireless sensor networks — a
survey of requirements, protocols, and challenges. *IEEE Commun Surv Tut* 2014; 16(3): 1391–1412.

2. Oh H and Vinh P. Design and implementation of a MAC protocol for timely and reliable delivery of command and data in dynamic wireless sensor networks. *Sensors* 2013; 13: 13228–13257.

3. Oh H and Ngo CT. A slotted sense multiple access protocol for timely and reliable data transmission in dynamic wireless sensor networks. *IEEE Sens J* 2018; 18(5): 2184–2194.

4. Song W, Huang R, Shirazi B, et al. TreeMAC: localized TDMA MAC protocol for real-time high-data-rate sensor networks. In: *Proceedings of the 2009 IEEE international conference on pervasive computing and communications*, Galveston, TX, 9–13 March 2009, pp.1–10. New York: IEEE.

5. Malhotra B, Nikolaidis I and Nascimento M. Aggregation convergecast scheduling in wireless sensor networks. *Wirel Netw* 2011; 17: 319–335.

6. Rhee I, Warrier A, Aia M, et al. Z-MAC: a hybrid MAC for wireless sensor networks. *IEEE/ACM T Network* 2008; 16(3): 511–524.

7. Jovanovic MD and Djordjevic GL. TFMAC: multi-channel MAC protocol for wireless sensor networks. In: *Proceedings of the 2007 8th international conference on telecommunications in modern satellite, cable and broadcasting services*, Nis, Serbia, 26–28 September 2007, pp.23–26. New York: IEEE.

8. Kim Y, Shin H and Cha H. Y-MAC: an energy-efficient multi-channel MAC protocol for dense wireless sensor networks. In: *Proceedings of the 2008 international conference on information processing in sensor networks (Ipsn 2008)*., St. Louis, MO, 22–24 April 2008, pp.53–63. New York: IEEE.

9. Incel O, Hoevel L, Jansen P, et al. MC-LMAC: a multi-channel MAC protocol for wireless sensor networks. *Ad Hoc Netw* 2011; 9: 73–94.

10. Borns J, Steenhaut K and Lemmens B. Low-overhead dynamic multi-channel MAC for wireless sensor networks. New York: Springer, 2010, pp.81–96.

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

12. Zhou M, Harada H, Kong P, et al. Interference range analysis and scheduling among three-hop neighborhood in maritime WiMAX mesh networks. In: *Proceedings of the 2010 IEEE wireless communication and networking conference*, Sydney, NSW, Australia, 18–21 April 2010, pp.1–6. New York: IEEE.

13. Kaixin X, Gerla M and Sang B. How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks. In: *Proceedings of the global telecommunications conference*, Taipei, Taiwan, 17–21 November 2002, vol. 1, pp.72–76. New York: IEEE.

14. Pyeon D and Yoon H. An efficient multi-path pipeline transmission for a bulk data transfer in IEEE 802.15.4 multi-hop networks. *Wirel Netw* 2019; 25(1): 117–130.

15. Swain RR, Mishra S, Samal TK, et al. An energy efficient advertisement based multichannel distributed MAC protocol for wireless sensor networks (Adv-MMAC). *Wirel Pers Commun* 2017; 95(2): 655–682.

16. Cheng H, Su Z, Xiong N, et al. Energy-efficient node scheduling algorithms for wireless sensor networks using Markov Random Field model. *Inf Sci* 2016; 329(c): 461–477.

17. Ganeriwal S, Kumar R and Srivastava M. Timing-sync protocol for sensor networks. In: *Proceedings of the First international conference on embedded networked sensor systems Sensys’03*, Los Angeles, CA, 5–7 November 2003, pp.138–149. ACM.

18. Maróti M, Kusy B, Simon G, et al. The flooding time synchronization protocol. In: *Proceedings of the 2nd international conference on Embedded networked sensor system*, Baltimore, MD, 3–5 November 2004, pp.39–49. ACM.

19. He J, Cheng P, Shi L, et al. Time synchronization in WSNs: a maximum-value-based consensus approach. *IEEE T Automat Control* 2014; 59(3): 660–675.

20. Van Vinh P and Oh H. RSBP: a reliable slotted broadcast protocol in wireless sensor networks. *Sensors* 2012; 12(11): 14630–14646.

21. Kim S and Eom D. Link-state-estimation-based transmission power control in wireless body area networks. *IEEE J Biomed Health Inform* 2014; 18(4): 1294–1302.

22. Entezami F, Tunicliffe M and Politis C. Find the weakest link: statistical analysis on wireless sensor network link-quality metrics. *IEEE Veh Technol Mag* 2014; 9(3): 28–38.

23. Bettstetter C, Hartenstein H and Costa-Pérez X. Stochastic properties of the random waypoint mobility model. New York: Springer, 2002, p.7.

24. Lifang KHG and Huimin X. Distribution of the Node Degree in MANETs. In: *Proceedings of the fourth international conference next generation mobile applications, services and technologies (NGMAST)*, Amman, Jordan, 27–29 July 2010, pp.162–167. New York: IEEE.