Cross-Layer Protocol Combining Tree Routing and TDMA Slotting in Wireless Sensor Networks

Ronggang BAI†, Nonmember, Yusheng JI†(a), Member, Zhiting LIN†, Qinghua WANG†, Xiaofang ZHOU†, Yugui QU†, and Baohua ZHAO†, Nonmembers

1. Introduction

Wireless sensor networks (WSN) are widely used in many areas such as industry, environment, and military fields, but the limited amount of battery energy is the major constraint restricting the range of its applications. This motivates the proposal of many energy-saving strategies based on the hierarchical structure of protocols [1], [2]. However, the hierarchical structure is a local optimization scheme. Each layer only accomplishes its own functions without taking the new characteristics of WSN into account.

The nodes in WSN usually implement a common task with limited energy, and this is different from traditional wireless networks, in which nodes have their own, separated tasks. The nodes in WSN should access the channel on schedule to avoid contentions for the holistic performance and the energy saving purposes. This means the nodes are cooperative rather than competitive.

Therefore, the protocols need to adopt a contention-free mechanism such as scheduling and avoid contentions. In addition, the nodes in WSN have few point-to-point communications. Therefore, the direction and load of the data traffic are well-regulated. By adopting a suitable routing protocol with this situation in mind, the network performance is expected to be improved. Some routing protocols with simple structures were proposed such as dynamic tree routing protocols [3].

We propose a cross-layer protocol and architecture for wireless sensor networks in this paper called CLWSN, which is based on the above characteristics. CLWSN integrates the dynamic tree routing protocols with the TDMA MAC protocol such as those found in [4]–[6], and accomplishes holistic optimization by exchanging information between different layers to improve the global performance. CLWSN proposes an efficient slot allocation scheme using the routing information to reduce the number of nodes in the conflicting set to increase the throughput. In addition, it can avoid the disadvantage of a large delay in the original TDMA by rearranging the order of the slots. CLWSN’s structure is simple, energy-saving, and efficient.

2. Background

The cross-layer design for wireless sensor networks breaks the stack of the hierarchical structure of original protocols to achieve a better Quality of Service (QoS) for different applications. In recent years, various approaches on the cross-layer protocols have been proposed [7].

LESOP [8] is a cross-layer design for better efficiency. In LESOP, direct interactions between the application and MAC layers are exploited. The transport and network layers are excluded to simplify the protocol stack and to control the tradeoff between the inspecting error and the energy consumption. Akyildiz et al. [9] developed a unified cross-layer protocol that melts the traditional layers, such as the transport, routing, medium access, and physical layers, into a single layer protocol. In [10], Cui et al. proposed an iterative algorithm that performs adaptive link scheduling and computes optimal link rates and transmission powers for a fixed link schedule, given that the TDMA-based link schedules are non-orthogonal. Another related protocol is presented in [11]. Collision-free, deterministic, periodic flows are scheduled toward a data sink. This protocol uses cross-layer information to perform more efficient MAC layer transmission scheduling across the network. However, one disadvantage of this approach is that broadcasts cannot be scheduled since explicit ACKs are required to create a schedule, so this method is a poor choice in event driven networks.

MINA [12] is a cross-layer protocol that combines the routing and MAC protocols. The base station (BS) assigns slots for the nodes using a centralized algorithm to reduce...
Generally believed that the interference radius \((R_r)\) is beyond the communication range, and it is generally static. For each time slot, CLWSN selects only one transmitter per conflicting set, and therefore, all the nodes in the set of one-hop neighborhood of the transmitter are deemed to receive collision-free data. The slot assignment in the path of the tree is optimized to accelerate the data transmission.

The rest part of this paper is organized as follows. Section 3 describes the definition of the conflicting set in the protocol. Section 4 describes the protocol details including the self-organization, slot allocation, and slot optimization algorithms. A performance analysis and simulation results are given in Sect. 5, and Sect. 6 concludes the paper.

### 3. Conflicting Set

In this paper, the conflicting set of node \(i\) is defined as the set of nodes that will interfere with the i’s transmission when sending simultaneously. In MINA and some other protocols\,[5],[12], the nodes in the conflicting set of node \(i\) are its two-hop neighbors. However, in \[6\], it is pointed out that the signal of the sender could interfere with the receipt of some nodes beyond the communication range, and it is generally believed that the interference radius \((R_d)\) is twice that of the communication radius \((R_r)\).

In the real environment, the received signal strength decreases with the distance \((d)\) as:

\[
P_{\text{recv}} \propto P_{\text{send}} \cdot d^{-\beta}
\]

where \(P_{\text{send}}\) is the sending power, \(P_{\text{recv}}\) is the receiving power and \(\beta\) is an environment-dependent constant normally between 2 and 5\,[15]. Bai et al.\,[6] prove that

\[
R_d = \sqrt{10} \cdot R_r
\]

Generally, the nodes communicate with each other using the normal transmitting power at a fixed range \((R_r)\). However, when the nodes collect the information of interference neighbors in \(D(i)\) during the self-organization process, they enhance the transmitting power to increase the communication range according to Eq. (1), in order to be heard by the interference neighbors \((R_d)\). If the nodes have received the packet from their interference neighbors, they reply ACKs with the enhanced power too.

The conflicting set of node \(i\) (denoted by \(U(i)\)) is shown in Table 1 and Fig. 1. Confliction will be avoided as long as the slot of \(i\) is different from that of the other nodes in \(U(i)\).

In the TDMA frame structure, each slot contains two sub-slots, D-slot and A-slot. The DATA packets are sent during the D-slots, and the ACK packets are sent during the A-slots.

**Theorem:** Node \(i\) can communicate with its next hop \(i'\) without conflict when there aren’t any other nodes in \(U(i)\) sending DATA during the same D-slot.

**Proof:** We first prove that \(\forall j \in U(i), i\) will consequentially conflict with \(j\). Hereinafter we define “one node conflicts with another” as that there will consequentially be interference if two nodes send data to their next hop at the same time. For \(j \in D(i') \cup j' \in D(i)\), there are two cases.

1) If \(j \in D(i')\), when \(i\) sends data to \(i'\) and \(j\) sends data to \(j'\) simultaneously, the signal of \(j\) will interfere with the reception of \(i'\).

2) If \(j' \in D(i)\), when \(i\) sends data ACK to \(i\) and \(j'\) sends data ACK to \(j\) simultaneously, the signal of \(j'\) will interfere

<table>
<thead>
<tr>
<th>Notation</th>
<th>Signification</th>
</tr>
</thead>
<tbody>
<tr>
<td>(d_{ij})</td>
<td>Distance between node (i) and (j)</td>
</tr>
<tr>
<td>(i')</td>
<td>Next hop of node (i) in routing tree, e.g., (a'=b, b'=f)</td>
</tr>
<tr>
<td>(R(i))</td>
<td>Communication neighbors’ set of node (i), such that (R(i) = {j</td>
</tr>
<tr>
<td>(D(i))</td>
<td>Interference neighbors’ set of node (i) (called D-hop), such that (D(i) = {j</td>
</tr>
<tr>
<td>(U(i))</td>
<td>Conflicting set of node (i) (called U-hop), such that (U(i) = {j</td>
</tr>
</tbody>
</table>

**Table 1** Notations and terminologies.
with the reception of $i$.

Therefore, we conclude that $\forall j \in U(i)$, $i$ will consequently conflict with $j$.

Next, we prove that $U(i)$ is the minimum conflicting set of $i$. This proposition is equivalent to $\forall j \notin U(i)$, where $i$ does not conflict with $j$. We adopt the reduction to absurdity and suppose $\exists j \notin U(i)$, $i$ conflicts with $j$. Suppose the collision happens when $i$ sends data to $i'$ and $j$ sends data to $j'$ simultaneously. For the collision we can assume that $j' \in D(i)$ or $i' \in D(j)$. If $i' \in D(j)$, then $j \in D(i')$. So, $j \in D(i')$ or $j' \in D(i)$ can get $j \in U(i)$. This is contradictory to the assumption. So, we conclude that $U(i)$ is the minimum conflicting set of $i$.

4. CLWSN Protocol Details

This section describes the initialization and self-organization processes, as well as the slot allocation and data transmission process. When the nodes have been distributed in the region, we assume that they have already been synchronized.

The startup behavior of the network is divided into three phases. The self-organization phase includes two steps, and these steps will be implemented once at the beginning. The slot allocation phase includes five steps. There is a cycle, Steps 4 to 7, in the slot allocation phase. The repeating times of the cycle will be given in Step 7. The slot optimization phase includes five steps. The repeating times of the cycle (Steps 8 to 12) will be explained in Step 12. Then, the data transmission phase will follow these startup phases.

4.1 Self-Organization Phase

This phase includes the following steps (in Fig. 2):

**Step 1** Establish routing tree: Use the algorithm in [3] to establish the routing tree. The process of the tree construction starts with a token message generated at the BS. Upon receiving the token, the node broadcasts the token to its one-hop neighbors who have not received the token yet. Once it finds that all its neighbors have received the token, it sends the token back to its parent, which is the node from which it receives the token for the first time. At the end of the process, the token carries the information of the routing tree back to the BS.

We assume that each node only selects one father node in the tree as the next hop, and the algorithms described below are also appropriate for a situation where there are multiple next-hop nodes in the routing tree.

After establishing the routing tree, all nodes acquire the knowledge of its next and previous hop nodes, and the BS can broadcast directly to the entire network. In the data delivery from the nodes to the BS, a node only sends DATA to its next-hop node, and receives an ACK from its previous hop nodes (may be more than one).

**Step 2** Collect the neighbor information to identify the conflicting set. Now all nodes in the network have unique IDs and know their previous hop nodes’ IDs. First of all, all nodes broadcast their IDs and their previous hop nodes’ IDs to their interference neighbors. The nodes should increase their sending power to be heard by the interference neighbors. Here RF transceivers are assumed to be equipped with functions of changing transmitting power [16].

After collecting the information from their interference neighbors, the nodes establish the D-hop neighbor lists that include the IDs of their D-hop neighbors and their previous hop nodes’ IDs (i.e. collecting information on $\{j | j' \in D(i)\}$).

Then, all the nodes send their D-hop neighbors’ lists to their previous hop nodes (i.e. collecting information on $\{j | j \in D(i')\}$). At last, the nodes establish their conflicting set lists based on their D-hop neighbor lists and their next hop nodes’ D-hop neighbor lists (i.e. $U(i) = \{j | j \in D(i') \cup j' \in D(i)\}$).

We assume that the topology of the networks is static most of the time. If some nodes are added or removed, the BS should reset the network to rebuild the routing tree and assign the schedule again. Generally speaking, the data transmission phase takes about millions of seconds. The self-organization phase takes about 5 (s) respectively. The slot allocation phase and a round of the slot optimization phase take about 30 (s). The slot allocation phase and a round of the slot optimization phase take about 5 (s) respectively. Hence, the energy consumption of the startup phases could be ignored in the evaluations.

4.2 Slot Allocation and Optimization Phase

4.2.1 Slot Allocation Algorithm

After the self-organization phase, each node receives the U-hop information. In this phase, each node selects a conflict-free slot number. Steps 3 to 7 describe the distributed slot allocation algorithm, and these steps are synchronously implemented.

**Step 3** Each node $i$ computes the priority $p_j$ for the nodes $j \in U(i)$, such that

$$p_j = \text{Rand}(ID_j).$$  

(3)

$\text{Rand}(ID_j)$ is a pseudo-random function, $ID_j$ is the seed, and
$ID_j$ is the ID of node $j$. From the definition of $U(i)$, we can conclude that $j \in U(i)$ is equal to $i \in U(j)$. Function $Rand$ will produce the same result using the same seed, so all the nodes in $U(j)$ will get the same priority $p_j$. After calculating the priority, the distributed slot allocation cycle, Steps 4 to 7, will be implemented.

**Step 4** At the beginning of the cycle, each node $i$ that hasn’t been assigned a slot compares the priority of all the nodes in $U(i)$ to find whether its priority is the largest. If it is, node $i$ is the winner (called $w$ for short) among $U(i)$ during this circle. Otherwise, it is a non-winner. There will always be several winners during each cycle in the network, but there won’t be any other winners in the $U(w)$ except $w$.

**Step 5** Each winner $w$ selects the smallest slot number $SlotId(w)$, that is not used by other nodes in $U(w)$, and $SlotId(w)$ is a natural number.

**Step 6** Each winner $w$ broadcasts its $SlotId(w)$ to the nodes in $U(w)$ in a multi-hop fashion. First, $w$ sends $SlotId(w)$ to its next hop $w'$, and then $w'$ broadcasts $SlotId(w)$ to the nodes in $D(w')$. Second, $w$ broadcasts $SlotId(w)$ to the nodes in $D(w)$, and then the nodes in $D(w)$ send $SlotId(w)$ to their previous nodes. The nodes should increase their sending power to be heard by their interference neighbors.

In the end, the non-winners receive and save the winners’ $SlotIds$, and set the priority of the winner ($p_w$) to zero (i.e. the minimum).

**Step 7** Jump to Step 4. Since there will be at least one winner generated consequentially during every cycle in each conflicting set, the minimal number of cycles is equal to the largest number of nodes in the conflicting set of the network (i.e. $max(U(i))$). At the end of the slot allocation algorithm, each node has been assigned a slot number that is unique in its conflicting set.

### 4.2.2 Slot Optimization Algorithm

After running the slot allocation algorithm, the slots are randomly allocated along the branches of the routing tree. It is easy to compute that the mean delay for forwarding a packet is $T_{frame}/2$ per hop, where $T_{frame}$ is the length of the frame. Next, CLWSN rearranges the order of the slots in the branch based on the routing information to reduce the packet delivery latency. Steps 8 to 12 describe the distributed slot optimization algorithm:

**Step 8** Implement the same action as in Step 3. Then, enter a circle from Steps 9 to 12.

**Step 9** Same as Step 4. Each node whose slot hasn’t been rearranged determines whether it is the winner during the circle according to the priority calculated in Step 8.

**Step 10** Each winner $w$ reselects a $SlotId(w)$ that should be different from any slot number in $SlotId(U(w))$. The new $SlotId(w)$ attempts to increase from $SlotId(w')$ to $max(SlotId(U(w)))$, where $max(SlotId(U(w)))$ is the largest slot number of all the nodes in $U(w)$. Until it gets to $max(SlotId(U(w)))$, the $SlotId(w)$ attempts to increase from 1 to the original value $SlotId(w)$. If there is a value matching the condition (different from $SlotId(U(w))$), $w$ selects the new value as $SlotId(w)$. This will reduce the packet delivery latency from $w$ to $w'$. If there is not any value matching the condition, $w$ keeps the original value $SlotId(w)$.

**Step 11** Same as Step 6.

**Step 12** Go to Step 9. The termination condition of the cycle is the same as in Step 7.

We can see that the slot optimization algorithm is very similar to the slot allocation algorithm, except for Steps 5 and 10. In Step 5 of the slot allocation algorithm, the winner $w$ selects the smallest $SlotId(w)$, which is not used by other nodes in $U(w)$. The purpose of the slot allocation algorithm is to assign the $SlotId$ for all nodes in the network. Then every node $i$ knows the $SlotId$ of its next hop node i.e. $SlotId(i')$, and the largest $SlotId$ of all the nodes in $U(i)$, i.e. $max(SlotId(U(i)))$. They are useful in Step 10.

The slot optimization algorithm can be repeated many times to gradually reduce the network delay. The more times the networks are optimized, the better the performance. The maximum number of optimization times is equal to the maximum number of hops in the routing tree from the source nodes to BS.

Finally, the maximum value of $SlotId$ representing the size of the conflicting set is broadcasted to all nodes in the network. All the nodes in the network report their $SlotIds$ to the base station through the tree. Then, the base station finds the largest $SlotId$ (denoted as $MaxSlot$), and broadcasts the $MaxSlot$ to all the nodes. All the nodes compute ($MaxSlot − SlotId$) as their final $SlotIds$, so that the slot number descends along the routing branch of the tree (Fig. 6).

### 4.3 Data Transmission Phase

In the data transmission phase, the channel is divided into a succession of frames. Each frame is divided into several slots whose numbers increase from 1 to $MaxSlot$. Each node is assigned a slot whose number is the same as the node’s $SLOTID$. Moreover, each slot contains two parts, D-slot and A-slot. The frame structure is shown in Fig. 3.

A node sends DATA during its D-slot, and receives an ACK from its next-hop node during its A-slot. It also receives DATA during its previous nodes’ D-slots, and replies to the ACKs during their A-slots. In addition, the node sleeps to save energy. The transmission process is as follows. Before each slot, every node judges whether this slot is its own or its previous-hop nodes’ or not. There are several possibilities:

1) If the slot is its own and it has data to be sent, it sends DATA to its next-hop node during its D-slot and wait
for an ACK during its A-slot. The RTS and CTS packets of the traditional MAC protocol are not necessary, because the collision and the hidden node problem have been resolved in the slot allocation algorithm.

2) If the slot is its own and it has no data to send, it sleeps during the whole slot.

3) If the slot does not belong to its own or its previous-hop nodes, it sleeps during the whole slot.

4) If the slot belongs to its previous-hop nodes, it wakes up and listens to the channel for the DATA during the D-slot, and replies to the ACK during the A-slot.

5. Performance Evaluation and Simulation

5.1 Size of Conflicting Set

Since the number of nodes in the conflicting set determines the length of the frame (MaxSlot) and the throughput performance, this section calculates the average number of nodes in \( U(i) \) (denoted as \( N_U \)).

As shown in Table 1, the condition for \( j \in U(i) \) is \( j \in D(i) \) or \( j' \in D(i) \). The first part is \( \{ j| j \in D(i') \} \), which includes the nodes in the circle whose center is \( i' \) and whose radius is \( R_d \).

For the convenience of analysis, we assume that nodes are randomly placed on a plane according to two-dimensional Poisson distribution with density \( \rho \). The assumption is based on the network model in Refs. [17]–[19]. However, when we did simulations on both Poisson- and uniformly distributed network nodes, we found that the results are very similar. Therefore, we put both kinds of the results in Fig. 5, and use the more realistic uniform distribution in the simulations of Sect. 5.3. Due to Poisson distribution, the probability of having \( k \) nodes in an area of size \( S \) follows a Poisson distribution, according to Eq. (4).

\[
p(k, S) = \frac{(\rho S)^k}{k!} e^{-\rho S}. \quad (4)
\]

The average number of nodes in the area of size \( S \) is \( \rho S \). So, the average number of nodes in the first part \( \{ j| j \in D(i') \} \) is denoted as \( N_D \), and

\[
N_D = \rho \pi R_d^2. \quad (5)
\]

The second part is \( \{ j| j' \in D(i) \} \). As shown in Fig. 4, if \( j' \) belongs to \( D(i) \), the \( d_{i-j} \) should be less than \( (R_d + R_s) \). Excluding the nodes counted in the first part, node \( j \) can only be in \( Ring(r_1, R_d + R_s) \), shown as the shadowed part in the figure. The inside boundary of the ring (denoted as \( r_1 \)) is a function of \( \alpha \), and \( \alpha \) is the angle between the line connecting \( i \) and \( i' \), and the line connecting \( i \) and \( j \). Therefore,

\[
r_1 = r \cos \alpha + \sqrt{t^2 \cos^2 \alpha - t^2 + R_s^2},
\]

where \( d_{i-j} \) (denoted as \( t \) for short) is the distance between \( i \) and \( i' \).

Generally speaking, the relative direction from \( i \) to \( i' \) should be the same as that from \( j \) to \( j' \), because the next hop of the node should be closer than itself to the BS. To simplify our calculation, we ignore the direction factor in the routing tree. The probability of \( j' \in D(i) \) is \( S_1/\pi R_s^2 \), where \( S_1 \) is the overlapping area of the circle whose center is \( i \) and radius is \( R_d \), and the circle whose center is \( j \) and radius is \( R_s \), i.e.,

\[
\Pr[j' \in D(i)] = S_1/\pi R_s^2, \quad (6)
\]

where \( S_1 \) is a function of \( d_{i-j} \) (denoted as \( r \) for short), such that

\[
S_1 = R_s^2 \arccos \frac{X}{R_d} + R_d^2 \arccos \frac{r - X}{R_s} - r(R_s^2 - X^2)^{\frac{1}{2}},
\]

where \( X = \frac{R_s^2 + r^2 - R_d^2}{2r} \).

Integrate the probability of Eq. (6) in \( Ring(r_1, R_d + R_s) \) to calculate \( N_{j' \in D(i)} \), which is the average number of nodes in the second part \( \{ j| j' \in D(i) \} \), and then we get

\[
N_{j' \in D(i)} = \rho \int_0^{R_1} \int_0^{2\pi} \int_{\frac{\sqrt{(\pi + 1)R_s R}}{\pi R_s^2}} S_1 d\theta dt dr. \quad (8)
\]

Where \( \beta \) is an environment-dependent constant. When \( \beta = 3.32, R_d = 2 + R_s \). From Eqs. (5) and (8), the average number of nodes in \( U(i) \) is

\[
N_U = N_{j' \in D(i)} + N_D. \quad (9)
\]

The mathematical expression of \( N_U \) is very complicated, so we calculate the theoretical value of \( N_U \) by using Eq. (9) with MATLAB Numerical Integration Toolbox in the next section.

5.2 Performance Evaluation for Slot Allocation Algorithm

TDMA-WSN and MINA are both TDMA protocols. We counted the size of the conflicting set and MaxSlot for TDMA-WSN [6], CLWSN, and MINA in a network with 800 nodes.

As shown in Fig. 5, first we can see that the statistical value of \( N_U \) in CLWSN is close to the theoretical value calculated by Eq. (9). Then, the conflicting set of TDMA-WSN...
NU and MaxSlot in CLWSN and MINA is built based on the three-hop neighborhood information. We compare the $N_u$ of TDMA-WSN with that of CLWSN. For the TDMA protocols, $N_u$ increases with the density of the network and $MaxSlot$ increases with $N_u$. The slot allocation algorithm in CLWSN obtains a smaller $N_u$ to reduce the $MaxSlot$ compared to TDMA-WSN.

Next, to evaluate the performances of the distributed algorithm in CLWSN with the centralized algorithm in MINA, Fig. 5 also compares the values of the $MaxSlot$ between MINA and CLWSN by assuming the same $N_u$. Ergen et al. [20] proves that it is a NP problem to find the minimum $MaxSlot$, and it tries to find the approximate value of the smallest $MaxSlot$ using a centralized algorithm. Although the centralized algorithm of MINA can find a smaller $MaxSlot$ than the distributed algorithm of CLWSN with the same $N_u$, this gain is at the cost of more energy consumption (Sect. 5.4).

Simulation results using both Poisson and uniform distributions to model node distributions for CLWSN are shown in Fig. 5, in order to compare with theoretical analysis results. Since they don’t make much difference in the simulations (including the slot optimization algorithm), hereinafter we use uniform distribution, which may be more realistic.

5.3 Performance Evaluation for Slot Optimization Algorithm

In order to evaluate the performance of the slot allocation and optimization algorithms, we tested the delay performance in a network with 100 nodes. These nodes are uniformly distributed in a $200 \times 200$ m square area. Figure 6 shows the network topology, where the BS is the root of the routing tree. The connections between the nodes denote the routing links and the numbers beside the nodes denote the $SlotId$. We represent the set of nodes in the network by $V$. $Delay(v, BS)$ denotes the delay for forwarding a packet from node $v$ to BS by using the multi-hop fashion.

$$\text{sum}\_\text{delay} = \sum_{v \in V} Delay(v, BS)$$ (10)

We calculated the sum delay of the network using Eq. (10) in the two phases to compare the improvement. After running the slot allocation algorithm, the slots are approximately and randomly allocated, and the resulting total delay was 10285 slots. Figure 6 also shows the slot allocation after the optimization. The slots are regularly rearranged down the branches, and the sum of the delay is 2305 slots.

Figure 7 shows the performance of the slot optimization algorithm (SOA) in different network scales. We also found that the delay in the network decreases with the number of times of the optimization and remains stable after running a few times. Although the optimization algorithm is local, the performance was improved by 40–50%. The necessary number of optimizations increases with the number of nodes in the network. In practice, optimization can greatly reduce the total delay. The slot optimization algorithm finished after 8 and 18 rounds (shown as “After SOA” in Fig. 7) for networks with 100 and 200 nodes, respectively.

5.4 Simulation

In this section, we compare the performance of CLWSN against that in “TREE+SMAC” and MINA using NS2 [21]. “TREE+SMAC” means that the routing protocol is TREE routing and the MAC protocol is SMAC without a cross-layer design.

In the simulations, each node (100 in total) sends 5
packets, and the traffic load is varied by changing the packet interval which is based on uniform distributions with a minimum of 0, and averages varying from 0 to 2 seconds. When the average packet interval is $T$ (s), the minimum is 0 (s), and the maximum is $2 * T$ (s). And each value is calculated as the average of 10 random runs.

We tested the energy and latency performances for these protocols for two scenarios. In the first scenario, the traffic load decreases as the packet interval increases from zero to two seconds, and the average number of one-hop neighbors (denoted as $N1$) is five. The $R_r$ and $R_d$, are 25 m and 50 m, respectively. In the second scenario, the network density increases as $N1$ increases from 5 to 15, and the packet interval is one second. The parameters of the simulation are shown in Table 2.

In the simulation, the power consumptions in the transmission, receive, and sleep modes are set to 29 mW, 31.7 mW, and 11 $\mu$W respectively [16].

In the simulation, the power consumed for state transition, i.e., from wakeup state to idle state, or from idle state to wakeup state, is not considered. Because the duration of a slot is 20 ms, and the duration of the state transition of the radio is less than 0.5 ms [22], the energy consumption for state transition can be ignored. On the other hand, the power consumed by both effective and ineffective control packets is considered in the evaluation.

First, we simulated the total energy consumption of the three protocols.

Figure 8 shows the total energy consumption for delivering all the messages with different packet intervals when $N1$ is 5. Figure 9 shows the total energy consumption with different network densities when the packet interval is one second. The “TREE+SMAC” consumes the most energy, for SMAC is a contention-based MAC protocol that consumes too much energy while idle listening and colliding. The MINA and CLWSN both adopt the collision-free TDMA MAC protocols to save energy.

Since each node only transmits during its slot and receives during its previous nodes’ slots, the duty cycle and the energy consumption decrease with the length of the frame ($MaxSlot$). In the TDMA protocols, $MaxSlot$ increases with the density of the network nodes, which is showed as the average number of neighbors in Fig. 9. Therefore, the energy consumptions of the TDMA protocols decrease with the increase of the density of the network in Fig. 9.

From Fig. 8, when $N1$ is 5 and the packet interval is one second, the $MaxSlot$ of CLWSN and MINA are 25 and 21, respectively. Then, the energy consumption ratio of CLWSN to MINA is $1 : 1.19 = 21 : 25$. From Fig. 9, it is easy to identify that the energy consumption ratio of CLWSN to MINA in the simulation is close to the theoretical value shown in Fig. 5, when $N1$ varies from 5 to 15.

Figure 10 shows the average end-to-end latency for all packets versus the traffic load, when $N1$ is 5. Figure 11 shows the latency versus the node density, when the message interval is 1 (s). When the message interval is less than 0.2 (s), the traffic load is beyond the capacity of CLWSN, and the delivery ratio drops below 100% for CLWSN (as latterly shown in Fig. 13). Therefore, the maximum capacity of CLWSN is a little less than that of MINA, but in most

### Table 2  Simulation parameters.

<table>
<thead>
<tr>
<th>Protocols</th>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLWSN/MINA</td>
<td>Duration of slot</td>
<td>0.02 s</td>
</tr>
<tr>
<td>CLWSN/MINA</td>
<td>Duty cycle</td>
<td>10%</td>
</tr>
<tr>
<td>CLWSN/MINA</td>
<td>Duration of listening interval</td>
<td>115 ms</td>
</tr>
<tr>
<td>Common parameters</td>
<td>Data packet length</td>
<td>200 Bytes</td>
</tr>
<tr>
<td></td>
<td>Management packet length</td>
<td>20 Bytes</td>
</tr>
<tr>
<td></td>
<td>Transmission rate</td>
<td>250 kbps</td>
</tr>
<tr>
<td></td>
<td>Max packet in iqf in NS2</td>
<td>300</td>
</tr>
<tr>
<td></td>
<td>Confidence Interval</td>
<td>90%</td>
</tr>
</tbody>
</table>
of the cases, CLWSN has smaller latency than MINA and SMAC.

When the slot allocation algorithm is implemented, CLWSN reduces the conflicting set to increase the spatial reuse and reduces the delay. CLWSN further improves its end-to-end delivery latency over multiple hops by rearranging the slot order in the routing path, as CLWSN allows more nodes in a path to finish their transmissions in one frame. On average, CLWSN reduces the end-to-end latency by more than 50% in both scenarios compared to MINA. CLWSN achieves the same delivery latency as SMAC under a wide range of traffic loads when the node density is low. From Fig. 11, although the idle listening greatly reduces the end-to-end latency for SMAC when the density is high, this gain is at the cost of higher energy consumption.

The delay of SMAC is reduced with the average number of neighbors. The reason is that, in the routing tree, the average number of hops from the nodes to the BS decreases with the density of the network, so the transmission time of packets decreases with the density of the network. While in the TDMA protocols, since the length of the frame (MaxSlot) increases with the density of the network, the delay also increases.

Different from the existing schemes, CLWSN balances the tradeoff between energy efficiency and latency. Figure 12 calculates the per-packet cost of energy-time by multiplying the results from Figs. 9 and 11 to evaluate the tradeoff on energy and latency. The results show that the integrative performance of CLWSN is better than the other two protocols. Since the energy consumption of MINA is smaller than that of SMAC (Fig. 9), while the latency of SMAC is smaller than that of MINA (Fig. 11), the product of energy and latency for SMAC and MINA crosses when the average number of neighbors is around 11. On the other hand, CLWSN gives the best performance due to that both energy consumption and latency are comparatively small for CLWSN. On average, when the packet interval is one second, the integrative performance ratio of CLWSN : MINA : SMAC is $1 : 2.17 : 2.29$.

In the simulation, the length of the queue in NS2 is 300. The arrival rate of the data should be less than the capacity of the network. When the traffic load is too heavy, the nodes will discard the packets, and the delivery ratios (the percentage of packets that are successfully received by BS) will be less than 100%.

Figure 13 shows the delivery ratios under different size of networks and different traffic loads and the purpose of this test is to show the different capacity in different protocols. The three protocols maintained a 100% delivery ratio when the packet interval was larger than 0.14 (s). The delivery ratio of SMAC drops quickly when the load becomes heavier, since more collisions can be caused by transmissions from hidden nodes. MINA and CLWSN maintained close to 100% packet delivery ratio (100 nodes, $N_1 = 5$) and outperformed SMAC when the traffic load was heavy. In the case of the large network size (200 nodes, $N_1 = 5$), because there are no more collisions in the TDMA protocols, the delivery ratios will be the same as that in the case of the small network size. In the case of the high network density (200 nodes, $N_1 = 10$), because the traffic load increases with the density, the delivery ratios for MINA and CLWSN will drop a little, but much slower than SMAC.

6. Conclusion

Scheduling and duty cycle mechanisms have been used in sensor networks to improve energy efficiency, but they also introduce a significant increase in the delivery latency. We proposed CLWSN as a TDMA-based distributed scheme to avoid collisions in MAC layer. In each time slot, CLWSN selects only one transmitter per conflicting set, which is defined as $U(i)$ using the routing information. Therefore,
all the nodes in the one-hop neighborhood of the transmitter are deemed to receive collision-free data. We also designed an algorithm that is capable of multi-hop data delivery in a single frame to accelerate the transmission. CLWSN rearranges the slot assignment using the routing information to allow more nodes in a path to set up a multi-hop schedule for subsequent forwarding of a packet in one frame. Therefore, each node along the forwarding path then wakes up at the correct time to receive and forward the data packet. Our theoretical analysis and simulation evaluation showed CLWSN’s advantages in reducing delivery latency and avoiding contention. Moreover, CLWSN saves energy, and improves the delivery ratio and throughput as well. Our future work is to develop the protocol which is applicable for a more realistic propagation model and to do the experiment on the hardware platform.

Acknowledgments

This paper is supported by the National Natural Science Foundation of China under Grant No. 60872009, 60602016, Hi-Tech Research and Development Program of China 863 No. 2007AAA01Z428, 2009AA01Z148; the Natural Science Foundation of Anhui under Grant No. ZD2008005-2, ZD200904, JK2009A013, JK2009A025.

References


Ronggang Bai received the B.S. in School of Information Science & Technology, University of Science and Technology of China in 2004. He now is Ph.D. candidate in University of Science and Technology of China, interested in protocols of wireless sensor networks.

Yusheng Ji received B.Eng., M.Eng. and D.Eng. in Electrical Engineering from the University of Tokyo in 1984, 1986 and 1989 respectively. She joined the National Center for Science Information Systems in 1990. She is currently an Associate Professor at the National Institute of Informatics, and the Graduate University for Advanced Studies. Her research interests include network architecture, protocol, traffic control and performance analysis for quality of service provisioning in wired and wireless networks. She is also a member of IPSJ and IEEE.
Zhiting Lin received the B.S. in School of Information Science & Technology, University of Science and Technology of China in 2004. He now is a Ph.D. candidate in the University of Science and Technology of China. He is interested in protocols of wireless sensor networks.

Qinghua Wang received the B.S. in Shan Dong University, China in 2004. She now is a Ph.D. candidate in the University of Science and Technology of China. She is interested in algorithms of wireless networks.

Xiaofang Zhou received the B.S. in School of Information Science & Technology, University of Science and Technology of China in 2005. She now is a Ph.D. candidate in the University of Science and Technology of China. She is interested in protocol theory and engineering of wireless networks.

Yugui Qu is a professor and Ph.D. supervisor in the University of Science and Technology of China. Her research interests are protocol theory and engineering, and protocols of wireless sensor networks.

Baohua Zhao is a professor and Ph.D. supervisor at the University of Science and Technology of China. He currently assumes the duties of vice director of Software Research Center of the Department of Computer Science, vice chairperson of CIC Communication Software Technical Committee, member of Professional Committee of Software Engineering of Chinese Computer Federation, and editor of Journal of Computer Information and Technology. His research interests include protocols theory and engineering and algorithms of wireless sensor network.