PAPER Special Section on Parallel and Distributed Computing and Networking

A Distributed TDMA Scheduling Algorithm with Distance-Measurement-Based Power Control for Sensor Networks

Koji SATO[†a)], ***Member and*** **Shiro SAKATA**[†], ***Fellow***

SUMMARY This paper proposes a distributed TDMA slot scheduling algorithm with power control, which the slot allocation priority is controlled by distance measurement information. In the proposed scheme, Lamport's bakery algorithm for mutual exclusion is applied for prioritized slot allocation based on the distance measurement information between nodes, and a packet-based transmission power control scheme is combined. This aims at achieving media access control methods which can construct a local network practically by limiting the scope. The proposed scheme can be shown as a possible replacement of DRAND algorithm for Z-MAC scheme in a distance-measurement-oriented manner. The scheme can contribute to the efficient TDMA slot allocation.

key words: *wireless sensor networks, media access control, TDMA, distance measurement, power control*

1. Introduction

The more the fields of wireless sensor networks have been expanded, the more active on the area of associated ad-hoc research has been. Not only applications in home networks or environmental monitoring, but various control techniques for wireless sensor networks in various fields have been presented [1]. In environments in which a variety of devices can be linked with each other, the realization of media access control methods which can construct a local network quickly and efficiently is strongly expected. Configuring the network in accordance with the particular context such as a distance enables to limit the scope of the target devices and to set up specific ad-hoc services and applications autonomously.

As a general requirement for communication scheme, efficient data delivery to multiple devices is an important issue. Even in such a large-scale environment, it can be considered to be common that multiple devices are scattered within a certain range. Therefore, if we can treat devices which exist in certain areas as a chunk of a group of specific categories, more efficient communications in the system can be achieved, and the construction of the network depending on the particular context is also feasible. It can lead to building a QoS-controlled network for a specific application, which can guarantees a response within a limited time, for example.

In this paper, we proposed a distributed TDMA slot scheduling algorithm by referring inter-device distance under such circumstances, with power control scheme named L-DRAND+, aiming at achieving media access control to construct an ad hoc network. The proposed scheme can be regarded as an extension of DRAND algorithm [3] for Z-MAC [2] combined with distance measurement. The method can contribute not only to be faster TDMA slot allocation than DRAND, but to reduce energy consumption.

The rest of this paper is organized as follows. Section 2 describes background research in sensor networks MAC protocols. Section 3 explains a proposed scheme in details. Section 4 gives an evaluation of proposed scheme by showing simulation results. Finally in Sect. 5, the summary and the future plans are illustrated.

2. Related Research

On media access control (MAC) protocols for sensor networks, various protocols have been proposed [4], for example, B-MAC [5] is a CSMA-based protocol which targets idle listening reduction by periodically receiving packets including preambles. Its transmission period is set longer than the sleep period of receiving node, in combination with LPL (Low Power Listening). CSMA scheme is outstanding in terms of bandwidth scalability in general, but it tends to increase unsolicited packets and header information for the specific node, and redundant active period.

On the other hand, TDMA scheme can reduce the redundant active period for each assigned nodes, because TDMA is a communication scheme with time-divided slot management. As an example of TDMA, we can pick out LEACH [6]. LEACH is the communication protocol which performs clustering in the network first, and then performs communications for slots independently after assigning a slot to each node in the cluster. Despite the efficiency of bandwidth, TDMA has a characteristic that it cannot easily follow against the topology changes. In such an aforementioned environment with a number of devices, frequent slot allocation will be necessary to be polled to the devices which have data to be transmitted in their own equally. In that sense, CSMA-based communication protocol is considered to be useful, but if we can specify the scope of the area locally, quick response to the operation via TDMA can be guaranteed. Therefore, a hybrid MAC equipped with TDMA control scheme to suppress the process overhead is desirable.

Z-MAC is a hybrid protocol which combines the advantages of CSMA and TDMA MAC protocols and has en-

Manuscript received January 10, 2012.
Manuscript revised May 8, 2012.
[†]The authors are with the Graduate School of Advanced Integration Science, Chiba University, Chiba-shi, 263–8522 Japan.
a) E-mail: sato-koji@sharp.co.jp
DOI: 10.1587/transinf.E95.D.2879

hanced in terms of bandwidth utilization compared to other protocols. Z-MAC protocol switches TDMA and CSMA depending on the contention situation to use the bandwidth effectively. Z-MAC slot assignment algorithm, DRAND, was implemented by a node conflict resolution procedure based on randomized ODP [7], but the calculation cost of running the algorithm tends to be high. And if the number of nodes increases, time for TDMA slot assignment would increase significantly. Therefore, DRAND has a problem in terms of scalability on the number of nodes.

Otherwise, the frequency of slot allocation process is also an issue in DRAND. In proportion to the increase of slot assignment opportunities, the time needed for the slot relocation is expected to be shortened as much as possible†.

In view of distance sensing devices, Cricket [8], [9] is an example of actual sensor hardware device which has a feature to measure the distance to other devices. This device has a feature which enables the position estimation especially in indoor environments using distance measurement with Time of Arrival (ToA) method realized by ultrasonic and RF devices. Cricket MAC protocol is configured based on B-MAC protocol, but the distance measurement information is only provided as data for the application (e.g., [10]). We cannot have seen yet any proposals which distance information can be fed back into media access control mechanism itself.

Designing a MAC protocol itself which can determine its behavior according to the distance measurement information will be significant. Because many kinds of devices, including Cricket, tend to have a function which can measure the distance, the function shipment cost will be declined. In addition, the usages of the RSSI (Received Signal Strength Indicator) of the RF radio without additional hardware have increased for location estimation (such as in [11]). Consequently, distance measurement technology would be easily achievable at no extra cost. Authors recognize there are various advantages such as time reduction of slot allocation by limiting the area, improvement of the process efficiency by autonomous control, or the interference avoidance from other networks, by referring the practical distance information.

In the following chapters, a slot allocation algorithm which aims at priority control in the network with distance measurement information for constructing TDMA MAC, is described.

3. Proposed Scheme

3.1 Preliminaries

Definition 1. This work assumes that a wireless sensor network comprises a group of nodes through a common broadcast channel with the same transmission range. Thus the topology of the network is represented by a uni-directed graph $G = (V, E)$, where V is the set of vertices (*nodes*) and $E \subseteq V \times V$ is the set of edges giving the available communications: if node v is a physical neighbor of node u, then there exists $(u, v) \in E$. If we assume that all nodes have the same communication range, denoted by R, then the set of links E is defined by:

$$E = \{(u, v) \in V \times V | dist(u, v) \leq R\} \tag{1}$$

$dist(u, v)$ is the Euclidean distance between node u and node v. If a link $(u, v) \in E$ exists, and that node u and node v are within the packet-reception range of each other, node u and node v are called one-hop neighbors of each other.

If a link $(u, v) \in E$ does not exist, but links (u, w), $(w, v) \in E$ exist $s.t. \exists w \in V$, node u and node v are called two-hop neighbors of each other. The node w is used as a relay node in this paper hereinafter. This is used to describe node relationship in terms of number of hops which is simply the minimum number of edges when a message has to cross to travel from node u to node v, *via node* w.

If node u and node v are two-hop neighbors via node w, the inter-node distance between node u and node v is defined by the sum of the Euclidean distance via node w:

$$dist(u, v) = dist(u, w) + dist(w, v) \tag{2}$$

3.2 DRAND-related Premises

L-DRAND+ is defined as a distributed slot allocation algorithm which enhanced DRAND characteristics further by adding features for localization with referring distance information between devices, including power controls. In L-DRAND+, following characteristics from DRAND are retained:

1. *No two nodes within a two-hop neighborhood will be assigned the same slot*

One of the premises in multi-hop DRAND environment shall be the same in L-DRAND+. This means that nodes in a two-hop neighborhood are assumed to interfere mutually in the same network.

2. *The maximum slot size of L-DRAND+ for the node assignment will be the same as that of DRAND*

As described hereinafter, L-DRAND+ is designed to combine the priority control algorithm with distance measurement information with original DRAND, when a slot assignment occurred. Therefore the maximum slot size will be the same as DRAND.

3. *Neighbor Discovery (ND) is the same as DRAND*

In L-DRAND+, the same Hello procedure in DRAND is used in ND phase, therefore the power control is not applied during the period. In order to collect accurate information of adjacent nodes, sufficient time is needed and there is a tradeoff between the observation time and accuracy. In this paper, this optimization issue is, however, out of scope. As described below, L-DRAND+ Hello message includes

†In Z-MAC, DRAND phase is separately designed under condition that each node position is fixed statically.

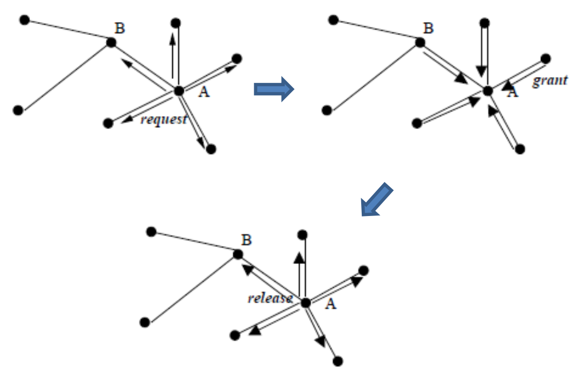

Fig. 1 DRAND: A successful round where a node A is allocated a time slot after receiving *grant* messages from its one-hop neighbors.

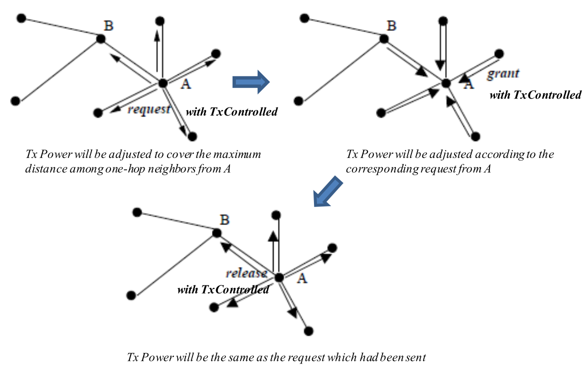

Fig. 2 L-DRAND+: A successful round where a node A is allocated a time slot after receiving *grant* messages from its one-hop neighbors.

distance measurement information which the sending node had held on its nodes within a one-hop neighborhood. This information is referred when the node determines the processing timing for slot assignment. The extended items of L-DRAND+ against DRAND are described in the following sections.

3.3 Prioritized Slot Assignment Control Based on Lamport's Bakery Algorithm

3.3.1 Overview

In DRAND, slot allocation control based on randomized ODP is implemented. The objective of the implementation is simply an exclusive control which only one node can issue a slot allocation request at the same time among multiple nodes. The exclusive control is conducted using slot allocation control packets such as *request*, *grant*, *reject*, *release*, and *fail*.

When a node A tries to acquire a time slot, A broadcasts a *request* message to its one-hop neighbors. If adjacent nodes of A, in the IDLE state for example, are ready to respond to it, each node sends a *grant* message. After A receives a *grant* from its entire one-hop neighbors for the *request*, it decides on its time slot to be the minimum of the time slots that have been taken by its two-hop neighbors before this round. Then A broadcasts a *release* message that contains selected time slot of A to inform its one-hop neighbors. Figure 1 shows a successful round where a node A is allocated a time slot after receiving *grant* messages from its one-hop neighbors†.

Figure 2 shows the successful round example in L-DRAND+. In L-DRAND+, packet-based Tx power controls have been introduced to reduce system power consumption. When A sends a *request* message, Tx power will be adjusted to cover the maximum distance among one-hop neighbors from A. The suitable distance will be selected among one-hop neighbors' information which had been collected when in ND phase. Any nodes which can be ready will respond a *grant* message with adjusted its Tx power to cover enough the distance from A, after respective node received a *request* message. For example, after detecting the

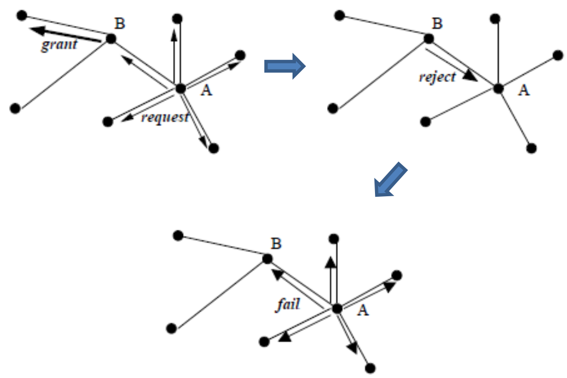

Fig. 3 DRAND/L-DRAND+: A failed round for a node A because a node B has sent a *grant* message to one-hop neighbors of a node B before receiving a *request* from A.

distance from A, B as well as A would adjust its Tx power to transmit a packet to A reactively. Then A broadcasts a *release* message with the same Tx power as it transmitted a *request*. In contrast, all the packets in DRAND are transmitted without Tx power controls.

Figure 3 shows a failed round for a node A in DRAND, because a node B has sent a *grant* to its one-hop neighbors before receiving the *request* from A. Other nodes except the one which had already sent a slot allocation *request* would be rejected its *request* from other adjacent nodes.

When receiving a *request* from A, if B is not ready to respond to it, because B is in the state of waiting a response to the former *request* which had already been sent from B for example, B sends a *reject* message to A. When A receives a *reject* from any node, A sends a *fail* message to all its one-hop neighbors to inform that the status of A will be changed.

In L-DRAND+, a fail round is the same as in Fig. 3, besides A can receive a *reject* with its Tx power control. Even when A received a *reject* with Tx power control, A will send a *fail* with its Tx power level normal. It means that Tx power will not be adjusted when transmitting a *fail* message to inform the one-hop neighbors of the message certainly. In case its Tx power level had been adjusted lower, the level

†Fig. 1 and Fig. 3 are referred from [3].

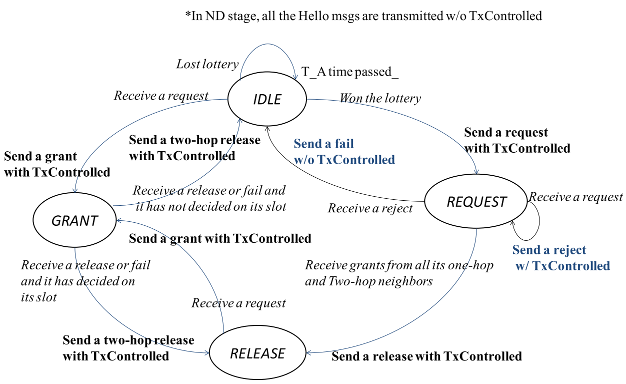

Fig. 4 State machines.

Table 1 Outline difference between DRAND and L-DRAND+.

items	DRAND	L-DRAND+	details
prioritized sequencing on slot allocation	- (randomized ODP)	○ (Lamport's bakery algorithm extension)	•controls using distance measurement information
Tx Power controls	-	○	•packet-based Tx power controls based on the distance info •proactive & reactive controls
Hello msg with distance measurment info	- (Hello without distance info)	○	•to be used to determine the priority

will be reverted to the normal.

Figure 4 shows that state machines of L-DRAND+. These state machines of L-DRAND+ determine the node's behaviors as shown in Fig. 2 and Fig. 3.

State machines of the nodes go back to IDLE state and wait until the next request is enabled to transmit with random backoffs. Consequently, the process will be delayed because a number of backoffs occur in a common condition when there are many unslotted nodes in the same network.

L-DRAND+ adopts an exclusive control algorithm which is based on Lamport's bakery algorithm [12] in place of randomized ODP. L-DRAND+ is designed to enable to be controlled under the existence of multiple N-threads simultaneously. In original Lamport's bakery algorithm, all the numbers which are assigned to the nodes themselves will be incremented when a new node as a "guest" joins, and the thread whose number is the smallest will be processed with priority, by checking the numbers.

In L-DRAND+, by adding the number according to the rule which is combined with the acquired distance measurement information, an effective prioritized order control for slot assignment is achieved. The details are described in Sects. 3.3.3 and 3.3.5.

In summary, the general differences between DRAND and L-DRAND+ are shown in Table 1.

By incorporating Tx controls, retransmission frequency can be changed according to the change of the transmission range. It may cause differences between the two schemes consequently. Details are described in Sect. 4.

```
integer array choosing[1:N], number [1:N]
*the relation "less than " on ordered pairs of integers is defined by
   (a,b) < (c,d)  or  if a < c, or a = c and b < d

begin integer j;
   L1 :   choosing[i] := 1;
          number[i] := 1 + maximum(number[1],...,number[N]);
          choosing[i] := 0;
          for  j = 1 step 1 until N do
               begin
                    L2: if choosing[j] ≠  0 then goto L2;
                    L3: if number[j] ≠ 0 and (number [j],j) < (number[i], i) then goto L3;
               end;
          critical section;
          number[i] := 0;
          noncritical section;
          goto L1;
end
```

Fig. 5 Pseudocode of Lamport's bakery algorithm.

3.3.2 Lamport's Bakery Algorithm

Lamport's bakery algorithm is one of many mutual exclusion algorithms which are designed to prevent concurrent threads entering critical sections to eliminate the risk of data corruption. The algorithm solves the following conditions assuming N asynchronous threads:

1. At any time, at most one thread may be in its critical section.
2. Each thread must eventually be able to enter its critical section (unless it halts).
3. Any thread may halt in its noncritical section

Lamport borrowed a bakery concept with a numbering machine at its entrance so each customer is given a unique number. Numbers increase by one as customers enter the store, the holder of the lowest number is the next one to be served. In Lamport's bakery algorithm, each thread chooses its own number, and waits until all the threads with smaller numbers finish their work. If two threads choose the same number, then the one with the lowest name goes first.

Figure 5 shows the pseudocode of Lamport's bakery algorithm†.

Lamport's bakery algorithm is designed for multiple threads from the beginning, and order control by reference to the number as shared information is available. In L-DRAND+, this algorithm is extended to allow the node from the near distance to enter the critical section preferentially. This achieves a prioritized order control based on the distance measurement information.

3.3.3 Rules for Prioritized Sequencing Control Using Distance Measurement Information

The basic rules of the proposed method are as follows:

i. *The slot allocation priority is given to the node if there is a node within a two-hop neighborhood, whose inter-node distance to relay node is less than the one of the selected*

†Fig. 5 is referred from [12].

node, and it has not been assigned a slot

Within a two-hop neighborhood, if there is a node where the inter-node distance to the relay node is shorter than the one of the selected node for applying the rule, the node in a closer range would be slotted prior to the others by making adjustments to it to give priority. Thus, the local node which does not exist adjacently to the node but is closer to the relay node than the selected one can join the network earlier.

ii. *The slot allocation priority is given to the relay node in the case of above and if the relay node has not been assigned a slot*

This rule allows the process order to be adjusted so that a key node within a one-hop neighborhood will join a network in order to build a local network as soon as possible.

iii. *The slot allocation priority is given to the node if there is a node within a one-hop neighborhood, whose inter-node distance in the two-hop is less than the one of the selected node, and it has not been assigned a slot*

This corresponds to the above case *i*, when viewed from the reverse side of a node within a two-hop neighborhood from a relay node. By applying these rules, the adjacent nodes would join the network rapidly, and these nodes would be assigned to the slot position closer to each other.

3.3.4 Hello Message with Distance Measurement Information

In L-DRAND+, apart from DRAND, the sending node has the distance information which the sending node had held in its nodes in a one-hop neighborhood, and the information is shipped with a Hello message. This includes information of multiple nodes according to the circumstances around the sender node. Figure 6 shows L-DRAND+ HelloMsg format[†]. The array interNodeDist stores the distance information of the nodes within a one-hop neighborhood from the sender.

When the node receives a Hello message, the receiver node measures the Euclidean distance to the sender node and store it to its internal DB which has kept distance information within a two-hop neighborhood. And then the receiver merges the distance measurement information acquired from the sender node with its internally managed information. The node can manage all the nodes within the two-hop neighborhood from its own node. The distance measurement information is referred to determine its protocol behavior, for example, when the node sends a slot assignment request, or what to do next after it received a reject message from other nodes.

```
typedef struct helloMsg{
    uint8_t sendID;
    uint8_t OneWayLen; // length of one way id array
    uint8_t OneWayId[OneWayLen];
    double interNodeDist[OneWayLen]; // 1 dimentional
} helloMsg;
```

Fig. 6 L-DRAND+ HelloMsg format.

3.3.5 Prioritized Sequencing Control Algorithm for Slot Allocation

By keeping the numbering rules prescribed to reflect the distance measurement information, the sequencing of nodes is determined according to the distance measurement information, as given in ascending priority order. The algorithm when slot allocation is requested is shown in Algorithm 1, and the other when receiving reject is shown in Algorithm 2. These algorithms are based on 3.3.3 descriptions. Presented

Algorithm 1 send request(slot allocation request):

```
0: ticket_number[self]++;
   // ticket_number is the number which all the nodes have ,
   // and then increment of the node on process for starters
1: while has_unslotted_two-hop_node &&
         the_unslotted_two-hop_node_has_smaller_inter-node_distance:
   // checks if there're unslotted two-hop nodes
   // whose the inter-node distance to their relay node are nearer than that of
   // the distance within one-hop from the node on process
       ticket_number[unslotted_two-hop_node]++;
2:   if relay_node_of_above_two-hop_node_is_unslotted :
       ticket_number[relay_node]++;
3: while has_unslotted_one-hop_node &&
       the_unslotted_one-hop_node_has_smaller_inter-node_distance
   // checks if there're unslotted one-hop nodes
   // whose the inter-node distance to another nodes are nearer than that of
   // the distance within one-hop from the node on process
       ticket_number[unslotted_one-hop_node]++;
       ticket_number[self]++;
4: if (max(ticket_number[]) != ticket_number[self]):
   // checks if the ticket_number of the node is the largest among others elements.
   // If not then,
       backoff := [regular round duration] * sum(ticket_numbers[])* rand[0,1];
       wait_until_next_round_with_backoff(backoff);
   // wait until the next send request round adding backoff duration calculated by the sum of
   // all the ticket_numbers which are not zero:
   else:
       send request
```

Algorithm 2 receive reject (backoff toward next slot allocation request)

```
0: ticket_number[self]++;
   // ticket_number is the number which all the nodes have ,
   // and then increment of the node on process for starters
1: while has_unslotted_two-hop_node &&
         the_unslotted_two-hop_node_has_smaller_inter-node_distance:
   // checks if there're unslotted two-hop nodes whose the inter-node distance to their relay
   // node are nearer than that of the distance within one-hop from the node on process
       ticket_number[unslotted_two-hop_node]++;
2:   if relay_node_of_above_two-hop_node_is_unslotted :
       ticket_number[relay_node]++;
3: while has_unslotted_one-hop_node &&
       the_unslotted_one-hop_node_has_smaller_inter-node_distance
   // checks if there're unslotted one-hop nodes whose the inter-node distance to another
   // nodes are nearer than that of the distance within one-hop from the node on process
       ticket_number[unslotted_one-hop_node]++;
       ticket_number[self]++;
4: backoff := [regular round duration] * sum(ticket_numbers[])* rand[0,1]
   wait_until_next_round_with_backoff(backoff);
   // wait until the next send request round adding backoff duration calculated by the sum of
   // all the ticket_numbers which are not zero:
```

[†]The structure was configured on 32-bit Linux (Ubuntu 9).

variable ticket_number is an array whose element is assigned for respective node in a two-hop neighborhood which the node managed the distance information to count a value (ticket). By applying the rules sequentially, ticket_number value for each node has been operated, and finally on the judging phase, priority for assigning a slot will be determined by referring the value. Apart from the original Lamport's bakery algorithm, this proposed algorithm functions as the node with the highest ticket_number goes first.

Respective node calculates the timing of slot allocation request transmissions or the next processing after the receipt of the refusal based on the algorithms, to determine the processing in the local node.

When a node had judged to delay a request and to calculate the backoff timing, the number multiplied by the sum of ticket_number of which counted in the node will be used to set the next slot allocation request timing.

This aims to reduce the interference among adjacent nodes, and to optimize the start timing of the subsequent process in the local node, while proceeding another node with a higher priority than itself.

3.3.6 Slot Assignment Example by Proposed Method

Figure 7 shows a slot assignment result example of applying the proposed method when the number of nodes is six. The number in parentheses (x,y) means coordinates and indicates the position of the nodes in a plane coordinate system.

In Fig. 7, a node group A-B and another D-F are formed apart from a group B-C-D-E by having executed the slot assignment algorithm independently. Figure 7 shows that any node in the two-hop neighborhood is allocated to different slot for sure.

In DRAND slot assignment process, a node is randomly selected from the group that time conditions are met, to carry out the time slot assignment. But in this proposed method, the node behavior is determined by the rule that refers predefined distance measurement information according to surrounding environmental situation.

In the case of the topology shown in Fig. 7, the slot allocation request procedure is executed for each group in parallel. The final slot orders of each node group are D-F, A-B, and D-B-E-C, respectively. We can observe that the maximum slot size is optimized, if any two-hop nodes were not allocated in the same slot.

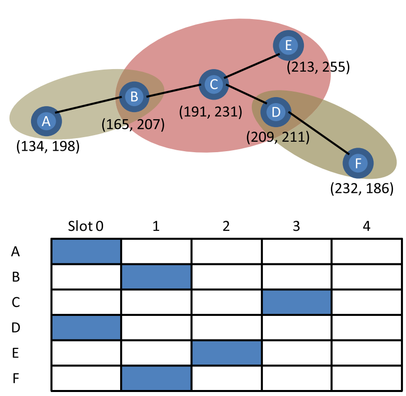

Fig. 7 Slot assignment result example: 6 nodes with X, Y –coordinates.

4. Evaluations

4.1 Conditions

To evaluate the proposed scheme, the above described algorithm was implemented on the network simulator ns-2 [13].

The network topology consists of nodes placed randomly on a 300×300 m surface. Nodes have a radio range of 40 m initially, and a link capacity of 2 Mbps†.

Basic simulation parameters†† are configured according to [3] in order to compare with a reference DRAND implementation. The major simulation parameters are shown in Table 2. The experiments are conducted with 20 repetitions of trials, varying the number of nodes between from 10 to 70 at run-time.

4.2 Average Number of Message Transmissions per a Node

Figure 8 shows a graph of the average number of message transmissions per a node during slot scheduling.

As the number of neighbor nodes increases, the increase in the number of sent messages can be confirmed on both DRAND and L-DRAND+. Totally, the frequency of transmissions of L-DRAND+ greatly exceeds that of DRAND. This is clearly shown in both cases it is getting harder to allocate slots as the number of neighbors becomes

Table 2 Simulation parameters.

Parameters	
Phy	802_11
Datarate	2Mbps
Propagation	TwoRayGround
Antenna/OmniAntenna(Gr_Gt_)	1.0
(hr, ht)	1.5
Capture Threshold(CPThresh_)	10.0 dB
Carrier Sense Threshold(CSThresh_)	1.559e-11 W
Transmission Power(Pt_: for 40m Xmit range initially & maximally)	8.5872e-4 W
Frequency(freq_)	914e+6
RXThresh_	3.652e-10 W
Pr_consume	395 mW
Pt_consume	660 mW
P_idle	35 mW
Sample time	700s

†Simulation parameters are configured according to the 914 MHz Lucent Wavelan DSSS radio Interface.

††Instead of TwoRayGround model, Freespace is practically used to calculate the transmission power to be adjusted because the inter-node distance will not exceed the crossover distance in this case.

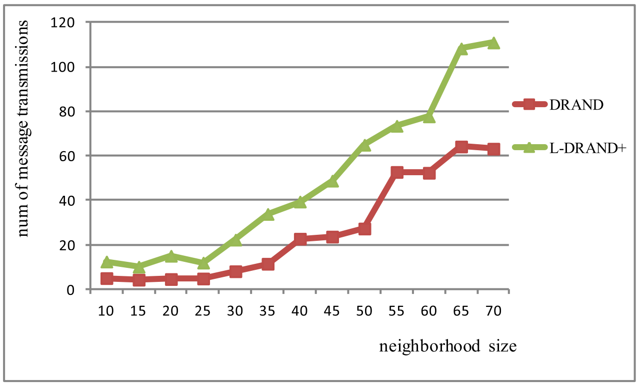

Fig. 8 The average number of message transmissions per node during slot scheduling.

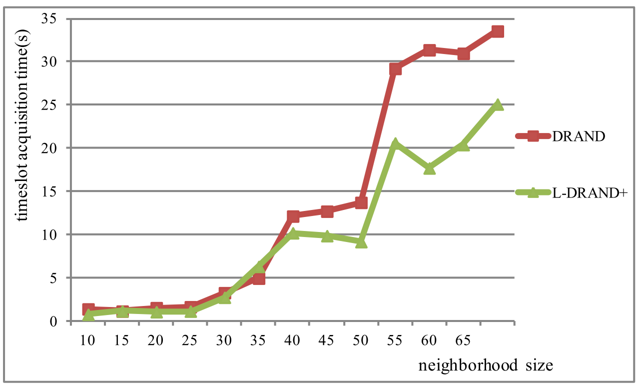

Fig. 9 The average time taken for a node to acquire a time slot.

large.

It should be noted that the special cases were observed in this simulation scenario set, at the points of 40 and 55 in the x-axis in Fig. 8. Detailed analysis of the transmissions revealed that the cases, which the slot allocations were not converged easily, i.e., slots were not assigned for a long time, were included. In these cases, the amount of transmissions tends to be large in common in both DRAND and L-DRAND+, because the state machine of L-DRAND+ is a descendant of DRAND. We can see the effect in other graphs such as Fig. 9, Fig. 10, and Fig. 11, because of using the common simulation scenario set.

In L-DRAND+, slot allocation request timing can be adaptively adjusted depending on the situation of adjacent nodes in a short period compared to DRAND. Therefore, a tendency to increase the number of sent messages significantly in response to the difficulty of slot assignment process can be observed. Practically, additional methods to reduce the frequency of transmissions are needed to be utilized, such as adding another protocol function such as constraining flows for adaptive control.

4.3 Average Time for a Node to Acquire a Time Slot

Figure 9 shows a graph of the average time taken for a node to acquire a time slot.

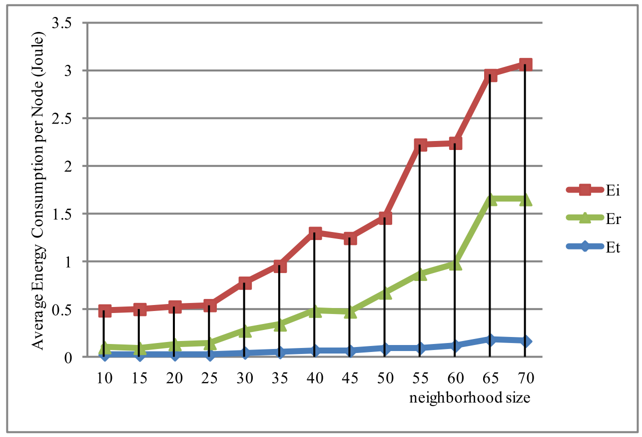

Fig. 10 The average energy consumption per node in L-DRAND+.

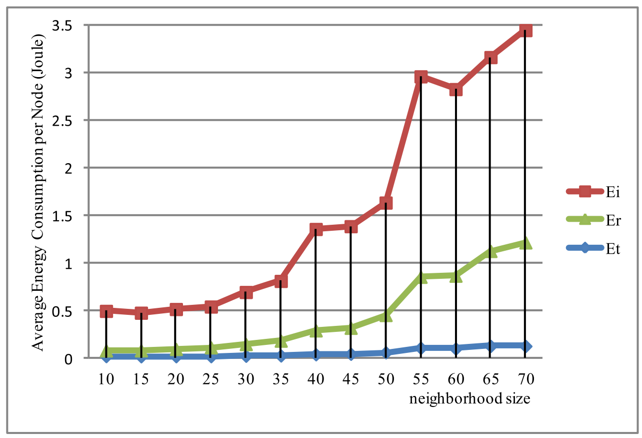

Fig. 11 The average energy consumption per node in DRAND.

By referring Fig. 9, DRAND and L-DRAND+ can be seen to complete their processes within nearly the same duration up to the neighborhood size 35.

In case of larger number of the nodes, L-DRAND+ can reduce its slot allocation time to around 78 percent compared to that of DRAND. The calculation cost of running the algorithm depends on its time typically, thus the proposed method can contribute to the reduction of the time and the calculation cost.

As described in Sect. 4.2, the cases which the slot allocations were not converged can be seen at the points of 40 and 55 in the x-axis in Fig. 9. Therefore, the corresponding time values are prominently high in comparison with the cases whose x positioning is adjacent to 40, or 55. If the above two cases are handled as special ones, the time values increase monotonously as x increases.

This result shows that the exclusive control based on Lamport's bakery algorithm with the distance measurement information is effective, under the condition that the slot assignment process becomes complicated according to the increase of the number of nodes. Further enhancement will be needed to be used in a practical environment, because quite a

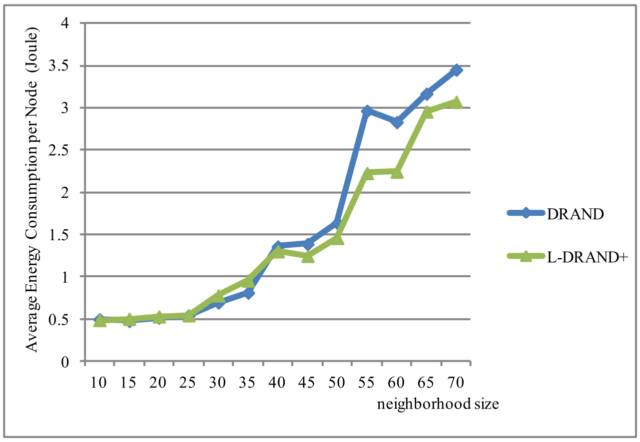

Fig. 12 The average energy consumption per node.

little time is still needed to process for slot assignment with a number of neighbors, and to deal with the special cases previously described.

4.4 Energy Consumption

By referring the energy model in ns-2, we've conducted energy consumption analysis based on the simulation result. Figure 10 shows a cumulative graph of the average energy consumption per node in L-DRAND+, and Fig. 11 is in DRAND.

In ns-2 energy model, total energy consumption is given by:

$$E_{total} = E_i + E_s + E_t + E_r \tag{3}$$

E_i is energy consumption in IDLE state, E_s is in SLEEP state but not used in this work ($E_s = 0$), E_t is consumed in transmitting packets, and E_r is in receiving packets. Thus top lines of the graph in Fig. 10 and Fig. 11 illustrate the total energy consumption of a node on average until the end of the simulation period (Fig. 12 shows the two top graphs).

By referring Fig. 10 and Fig. 11, E_i, E_r in L-DRAND+ are smaller than those of in DRAND, when the neighborhood size is large (beyond 40). It can be observed that L-DRAND+'s slot assignment process works appropriately even if the longer packets with distance information were handled with high frequency[†,††].

Additionally, it should be noted that high E_i values can be observed against the increasing rate on E_t and E_r at the points of 40 and 55 in the x-axis in both Fig. 10 and Fig. 11. These results illustrate that various nodes could not transmit from IDLE state for a long time, and the cases increased the energy consumption in the system. Even in the above cases, L-DRAND+ shows a better characteristic on energy consumption than DRAND.

Furthermore, L-DRAND+ can eliminate the redundant packets or control total amount of packets by limiting the scope of the network. In view of total energy consumption, we can conclude that L-DRAND+ shows good characteristics by optimizing packet transmissions in the system.

L-DRAND+ can be observed that it enables to limit the range of influence of communications in comparison with DRAND. To improve the energy efficiency further, a method that can reduce the duration in IDLE state more efficiently must be combined.

4.5 Miscellaneous Issues

L-DRAND+ can be expected to reduce time for slot allocation, and that the resulting network will be constructed in accordance with the order which is determined by the distance measurement information. In this scheduling, the adjacent nodes allow to be allocated in the closed slot positions in the early stage. Therefore, by shifting the set of slots from the head, for example, conflict resolution can be expected in the case of multiple congested networks. In case of conflicts caused by hidden terminals, the interference often occurs in the marginal area of the network. In that perspective, a method which does not degrade the performance by holding a series of slots with no impact can be feasible.

Additionally, limiting the area of a network has a possibility to lead to a construction of a QoS-controlled network, which is expected as one of the methods for environmental-/context-oriented network applications.

Lastly, it should be noted that the proposed scheme could be applicable to mobile nodes if slot allocation time is improved.

By referring Fig. 9, the slot allocation process finishes less than 10 s, in case of up to the neighborhood size 50. When considering the maximum Xmit range as 40 m, and the additional time to ND less than 1 s in the simulation, one slot re-allocation time within (at most) 11 s is needed after detecting whether the nodes have moved. Moreover, considering the case that the two nodes go away in the opposite direction, as a result, the proposed scheme can follow up to the speed of 1.8 m/s of mobile nodes under such conditions (note that this case does not include the net data transmission time).

$$\frac{40}{2 \times 11} \fallingdotseq 1.8\,\mathrm{m/s} \tag{4}$$

Slot allocation time improvement is effective in case of many nodes. The modification of the state machines (Fig. 4) by adding a function of adjusting retry timing depending on the situation of adjacent nodes, for example, will be a future work.

5. Conclusion

In this paper, a distributed TDMA slot scheduling with prioritized control based on Lamport's bakery algorithm is produced. The scheduling is applicable to media access control

†Energy consumption of receiving packets is observed to be significant compared to the one of sending packets.

††See Fig. 6.

methods which can constitute a locally limited network by measuring inter-device distances with efficiency. By using this proposed scheme, priority control for nodes in the network can be performed in the MAC layer according to the collected distance measurement information. It can also increase efficiency for slot allocation by reducing the processing time for it.

L-DRAND+ has a possibility to determine its behavior according to the environmental situation around the node, therefore, adaptive flow control with adjacent node information will be one option to be considered in view of the improvement of the protocol behaviors in the future.

In addition, a distance-oriented network can have benefits such as reducing the interference from other sensor networks, or even building a context-oriented network autonomously. In parallel, the approaches to context-aware network applications, that the proposed scheme is applicable, e.g., to distance information-used user interaction feedback model [14], would be important.

References

[1] A. Boukerche, Algorithms and protocols for wireless sensor networks, John Wiley & Sons, 2009.
[2] I. Rhee, A. Warrier, M. Aia, and J. Min, "Z-MAC: A hybrid MAC for wireless sensor networks," Proc. 3rd International Conference on Embedded Networked Sensor Systems (SenSys '05), pp.90–101, 2005.
[3] I. Rhee, A. Warrier, J. Min, and L. Xu, "DRAND: Distributed randomized TDMA scheduling for wireless Ad-Hoc networks," Proc. 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp.190–201, Florence, 2006.
[4] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC protocols for wireless sensor networks: A survey," IEEE Commun. Mag., vol.44, no.4, pp.115–121, 2006.
[5] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," Proc. 2nd International Conference on Embedded Networked Sensor Systems (SenSys'04), pp.95–107, Baltimore, MD, USA, Nov. 2004.
[6] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-effcient communication protocol for wireless microsensor networks," Proc. Hawaii International Conference on System Sciences (HICSS), Maui, Hawaii, Jan. 2000.
[7] I. Rhee, A. Warrier, and L. Xu, "Randomized dining philosophers to TDMA scheduling in wireless sensor networks," Technical Report, Computer Science Department, North California State University, Raleigh, NC, 2004.
[8] The Cricket Indoor Location System: http://cricket.csail.mit.edu/
[9] Memsic: http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html
[10] N.B. Priyantha, A.K.L. Miu, H. Balakrishnan, and S. Teller, "The Cricket Compass for context-aware mobile applications," Proc. 6th ACM MOBICOM Conf., Rome, Italy, July 2001.
[11] D. Anzai and S. Hara, "An RSSI-based MAP localization method with channel parameters estimation in wireless sensor networks," Vehicular Technology Conference, 2009 (VTC Spring 2009) IEEE, pp.1–5, April 2009.
[12] L. Lamport, "A new solution of Dijkstra's concurrent programming problem," Commun. ACM, vol.17, no.8, pp.453–455, Aug. 1974.
[13] The Network Simulator – ns-2: http://www.isi.edu/nsnam/ns/
[14] K. Sato and S. Sakata, "ManueverXML: Distance-measurement based operation event description model and user interaction interpretation," Proc. 10th Annual Symposium on Application and the Internet (SAINT2010) WS-1 (EUCASS), July 2010.

Koji Sato received the Master of Management Sciences and Public Policy Studies from Tsukuba University in 1994. He has joined Sharp Corporation since 1994, and has been engaged in research on mobile ubiquitous computing and its services. He is currently a graduate student of Chiba University. He is a member of IPSJ.

Shiro Sakata received B.E., M.E. and Dr.E., degrees in electronic communication engineering from Waseda University, Japan in 1972, 1974 and 1991, respectively. He joined NEC Corp. in 1974. He was engaged in research and development in the field of computer networks, Internet, multimedia communications, mobile communications and digital broadcast systems. He served as Director of Internet Systems Research Lab., NEC Corp. in 1996 to 2003. He joined Chiba University, Japan in 2004 and is currently a Professor at the Graduate School of Advanced Integration Science. His current research covers QoS control, reliability, energy-efficiency, multicasting and interoperability issues for ubiquitous communication and networking. He received Yamashita Memorial Research Award in 1994. He is an IEICE Fellow for his research contributions to ubiquitous network technologies, and an IPSJ Fellow for his research contributions to multimedia and mobile communication technologies.