L-DRAND: A Localized Distributed TDMA Scheduling Algorithm with Distance-Measurement for Sensor Networks

Koji Sato, and Shiro Sakata

***Abstract*—This paper proposes a distributed TDMA slot scheduling algorithm which the slot allocation priority is controlled by distance measurement information. In the proposed scheme, named L-DRAND, Lamport's bakery algorithm for mutual exclusion is applied for prioritized slot allocation based on the distance measurement information between nodes. This method aims at the realization of media access control methods which can construct a localized 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.**

***Index Terms*—Wireless sensor networks, media access control, TDMA, distance measurement**

I. 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 the home network 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 localized network quickly and efficiently is strongly expected. Configuring the network in accordance with the particular context like distance enables to limit the scope of the target devices and to set up specific ad-hoc services and applications autonomously, even when a variety of devices can be linked with.

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 group of category of devices, a local optimization of communications in the system can be achieved, and the construction of the network depending on the particular context is also expected.

In this paper, a distributed TDMA slot scheduling algorithm is introduced aiming at achieving media access control scheme which can construct a localized network by referring inter-device distance under such circumstances. 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 to be faster TDMA slot allocation than DRAND.

The contents of this paper are as follows. Section II describes background research in sensor networks MAC protocols. Section III explains a proposed scheme in details. Section IV gives an evaluation of proposed scheme by showing simulation results. Finally in Section V, the summary and the future plans are illustrated.

Manuscript received September 10, 2011.

K. Sato and S. Sakata are with Graduate School of Advanced Integration Science, Chiba University, 1-33 Yayoi-cho, Inage-ku, Chiba, 263-8522 Japan (e-mail:ksato@graduate.chiba-u.jp, sakata@faculty.chiba-u.jp).

II. 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 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 secured. Therefore, a hybrid MAC which equips 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 enhanced 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 need to decrease time needed for the slot relocation is expected to be shortened as much as possible[1].

In view of distance sensing sensor 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 ToA (Time of Arrival) 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 any proposals which distance information can be fed back into media access control mechanism itself yet.

Consideration of a scheme that MAC protocol itself 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, and the function shipment cost will be declined. 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.

III. Proposed Scheme

A. 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 a node v is a physical neighbor of a 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:

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

$dist(u, v)$ being the Euclidean distance between nodes u and v. If a link $(u, v) \in E$ exists, and that nodes u and v are within the packet-reception range of each other, u and 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 $s.t. \exists w \in V$, nodes u and v are called two-hop neighbors of each other. The node w is used as 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 u to v, via w.

B. DRAND-related Premises

Localized DRAND(L-DRAND, hereinafter) is defined as a distributed slot allocation algorithm which enhanced DRAND characteristics further by adding features for localization with referring distance information between devices. 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[2].

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

In L-DRAND, the same Hello procedure in DRAND is used in ND phase. In order to collect accurate information of

[1] In Z-MAC, DRAND phase is separately designed under condition that each node position is fixed statically

[2] Maximum slot size adaptation according to the nodes situation is in the future plan

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 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 DRAND for L-DRAND are described in the following sections.

C. Prioritized slot assignment control based on Lamport's bakery algorithm

1) Overview

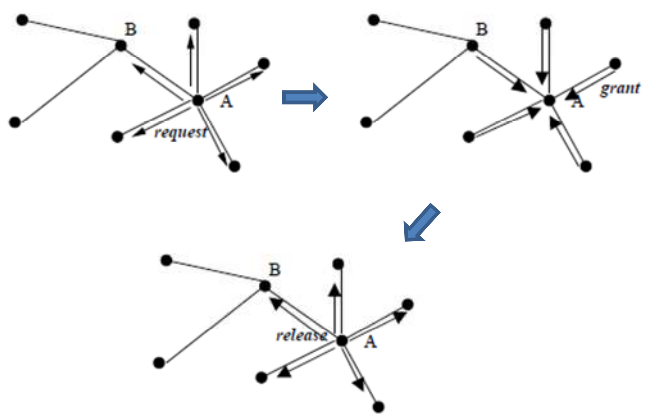

Fig. 1 DRAND: A successful round where a node A is allocated a time slot after receiving grant messages from its one-hop neighbors

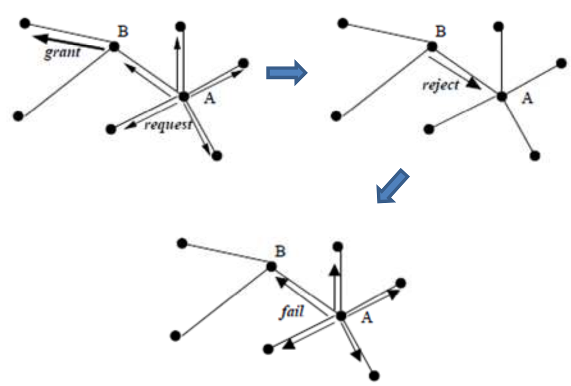

Fig. 2 DRAND: A failed round for a node A because a node B has sent a *grant* message to one-hop neighbors of a node B before receiving a *request* from A

In DRAND, slot allocation control based on randomized ODP is implemented. The objective of the implementation is a 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 broadcast a *release* message that contains selected time slot of A to inform its one-hop neighbors. Fig. 1 shows a successful round where a node A is allocated a time slot after receiving grant messages from its one-hop neighbors.

Fig. 2 shows a failed round for a node A, 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.

State machines of the nodes go back to WAIT 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 (Fig.1 and Fig.2 are from [3]).

L-DRAND adopts an exclusive control algorithm which is based on Lamport's bakery algorithm [11] 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 realized.

2) 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 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) 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. Fig. 3 shows L-DRAND HelloMsg format[3]. The array interNodeDist1 stores the distance information of the nodes within a one-hop neighborhood from the sender.

```
typedef struct helloMsg{
  uint8_t sendID;
  uint8_t OneWayLen; // length of one way id array
  uint8_t OneWayId[OneWayLen];
  double interNodeDist1[OneWayLen]; // 1 dimentional
} helloMsg;
```

Fig. 3 L-DRAND HelloMsg format

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.

4) Prioritized sequencing control algorithm for slot allocation

[3] Simulation environment was configured on 32-bit Linux(Ubuntu 9)

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 III.C.2) descriptions. Presented 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.

Algorithm 1 send request(slot alloc request)

```
1 :  ticket_number[self]++;
2:  if has_unslotted_two-hop_node &&
      has_smaller_inter-node_dist(unslotted_two-hop_node):
          ticket_number[unslotted_two-hop_node]++;
          ticket_number[self]++;
3:  if has_unslotted_one-hop_node &&
          has_smaller_inter-node_dist(unslotted_one-hop_node):
          ticket_number[unslotted_one-hop_node]++;
          ticket_number[self]++;
4:  if min(ticket_number[]) != ticket_number[self]:
        random_backoff(sum(less_than(ticket_number[self]))
    else:
      send request
```

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

```
1 :  ticket_number[self]++;
2:  if has_unslotted_two-hop_node &&
        has_smaller_inter-node_dist(unslotted_two-hop_node):
            ticket_number[unslotted_two-hop_node]++;
            ticket_number[self]++;
3:  if has_unslotted_one-hop_node &&
        has_smaller_inter-node_dist(unslotted_one-hop_node):
            ticket_number[unslotted_one-hop_node]++;
            ticket_number[self]++;
4:  random_backoff(sum(less_than(ticket_number[self]))
```

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.

5) Slot assignment example by proposed method

Fig. 4 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. 4, 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. Fig. 4 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. 4, 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 provided any two-hop nodes were not allocated in the same slot.

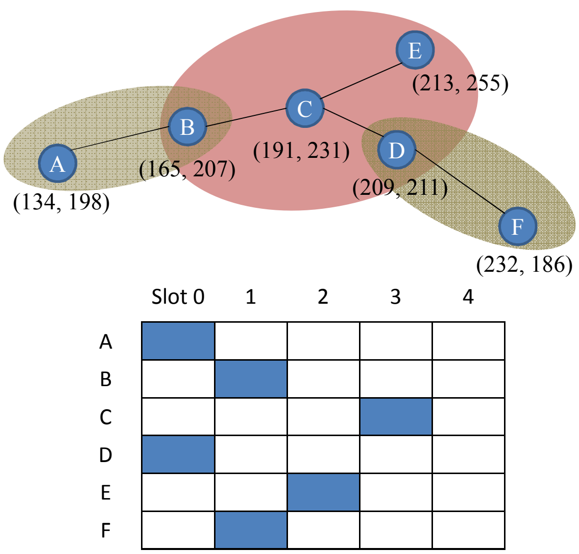

Fig. 4 Slot assignment result example: 6 nodes with X,Y –coordinates

IV. Evaluations

A. Conditions

To evaluate the proposed scheme, the above described algorithm was implemented on the network simulator ns-2 [12].

The network topology consists of nodes placed randomly on a 300x300m surface. Nodes have a radio range of 40m, and a link capacity of 2Mbps[4].

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 1. The experiments are conducted with 20 repetitions of trials, varying the number of nodes between from 10 to 70 at run-time.

[4] Simulation parameters are configured to work like the 914MHz Lucent Wavelan DSSS radio Interface

Table 1 Simulation parameters

Parameters	
Mac Type	MAC/802_11
Datarate	2Mbps
Capture Threshold(CPThresh_)	10.0 dB
Carrier Sense Threshold(CSThresh_)	1.559e-11
Transmit Power(Pt_: for 40m range)	8.5872e-4
Frequency(freq_)	914e+6
Receive Power Threshold(RXThresh_)	3.652e-10
rxPower	395 mW
txPower	660 mW
idlePower	35 mW
Sample time	700 s

B. Average number of message transmissions per a node

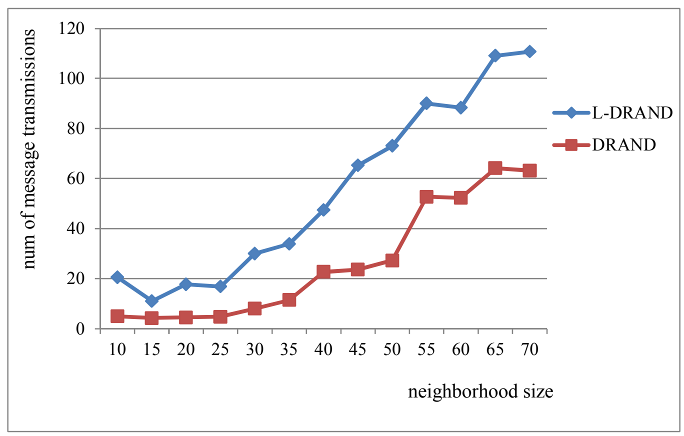

Fig. 5 The average number of message transmissions per node during slot scheduling

Fig. 5 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 large.

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 like constraining flows for adaptive control.

C. Average time for a node to acquire a time slot

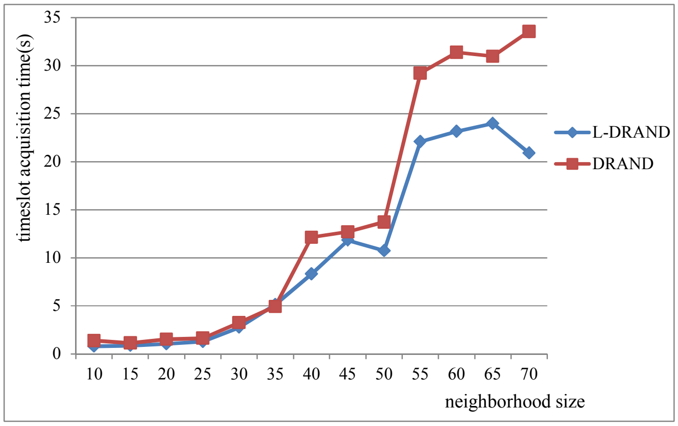

Fig. 6 The average time taken for a node to acquire a time slot

Fig. 6 shows a graph of the Fig. 6 The average time taken for a node to acquire a time slot.

By referring to Fig. 6, both 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 65 percent compared to that of DRAND. This result shows that the exclusive control based on Lamport's bakely algorithm with the distance measurement information make an effect, 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 little time is still needed to process for slot assignment with a number of neighbors.

D. Energy consumption

By referring the energy model in ns-2, we've conducted energy consumption analysis based on the simulation result. Fig. 7 shows a cumulative graph of the average energy consumption per node in L-DRAND, and Fig. 8 is in DRAND.

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

$$E_{total} = E_i + E_s + E_t + E_r \quad (2)$$

, where 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. 7 and Fig. 8 illustrate the total energy consumption of a node on average until the end of the simulation period.

By referring Fig. 7 and Fig. 8, IDLE duration in L-DRAND is shorter than that of in DRAND, but $E_t + E_r$ is bigger because the longer packets than DRAND's with distance information were handled with high frequency[5][6]. As a result, in view of total energy consumption, both are nearly the same.

If the method which can eliminate the redundant packets or control total amount of packets can be combined with L-DRAND, good characteristics will be expected on energy consumption.

L-DRAND has a possibility to determine its behavior according to the environmental situation around each node, adaptive flow control with localizing the network will be one option to be considered.

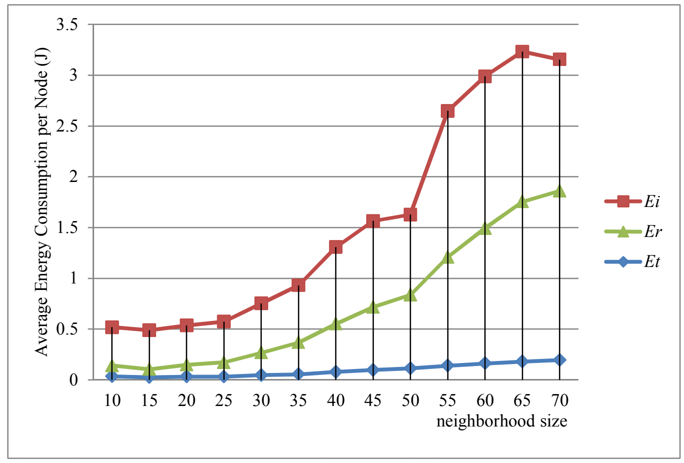

Fig. 7 The average energy consumption per node in L-DRAND

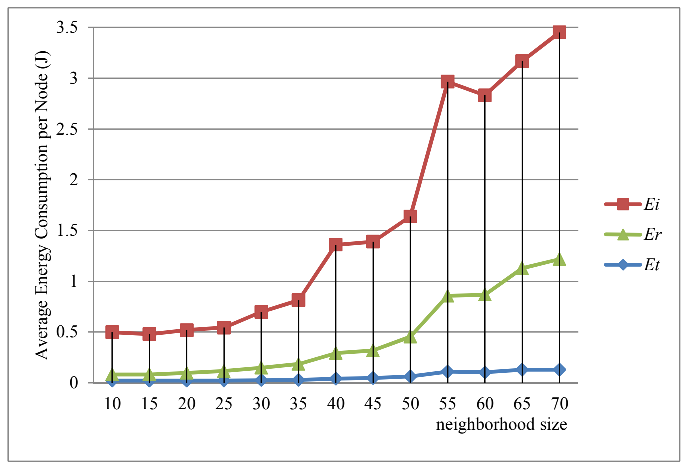

Fig. 8 The average energy consumption per node in DRAND

[5] Energy consumption of receiving packets is observed to be significant compared to the one of sending packets

[6] See Fig. 5

E. Network Construction Example

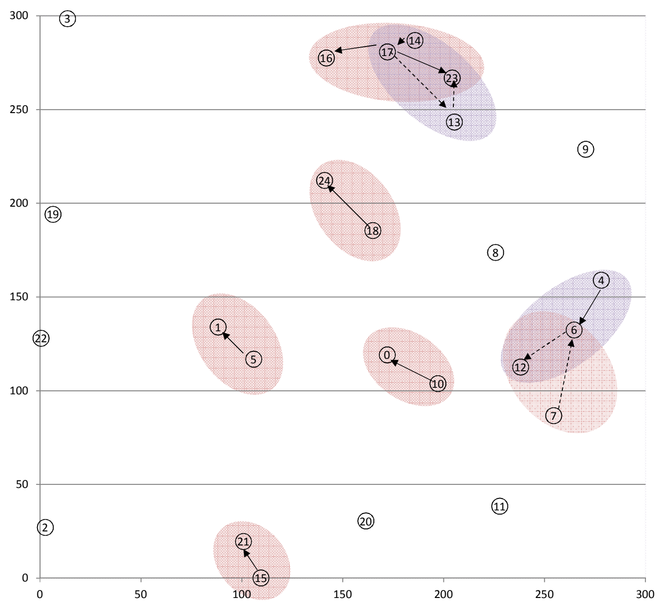

Fig. 9 An example of L-DRAND execution snapshot when neighborhood size is 25

Fig. 9 shows an L-DRAND execution snapshot as a network construction result when neighborhood size is 25.

In Fig. 9, when focusing on the set of nodes around the coordinate position (180,280), the node group 14-17-16-23 and the other group 14-17-13-23 are finally constructed as the third slot of their network is only different. In the process, the slot allocation process starts with the nodes 14-17 whose pair is in the shortest route of the topology. Node selection when running DRAND will be completely random process, therefore another mechanism must be needed to adapt to the environmental situation.

Additionally, when the number of neighborhoods is small in particular, L-DRAND slot allocation process can be finished within slight different number of message transmissions, with joining adjacent nodes with higher priority to the network (see Fig. 6).

F. 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 thinking of a series of slots for example, conflict resolution can be expected by shifting all the slots together. In case of conflicts by hidden terminals, the interference often occurs in the marginal area of the network.

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.

V. 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 achieve media access control 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, a variety of rule sets such as the only adjacent nodes can be collected on a priority basis, or the combination of plane partitioning algorithms for example, should be considered and evaluated.

In addition, a distance-oriented network can have benefits such as reducing the interference from another sensor networks, or even building a context-oriented network autonomously. In parallel, considering of context-aware network applications that the MAC can be effectively applicable, such as an active feedback model to user interactions by using the distance measurement information [13], should be significantly 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,” in 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,” in Proc. 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp.190-201, Florence, 2006

[4] I. Demirkol, C. Ersoy, F. Alagoz, “MAC protocols for wireless sensor networks: a survey,” IEEE Communications Magazine, vol 44(4), pp.115-121, 2006.

[5] J. Polastre, J. Hill, and D. Culler, ”Versatile low power media access for wireless sensor networks," in 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,” in Proc the 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,” in 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, S. Teller, ”The Cricket Compass for context-aware mobile applications,” in Proc. 6th ACM MOBICOM Conf., Rome, Italy, July 2001.

[11] L. Lamport, “A new solution of Dijkstra's concurrent programming problem,” Communications of the ACM 17, 8, pp.453-455, Aug. 1974.

[12] The Network Simulator – ns-2: http://www.isi.edu/nsnam/ns/

[13] K. Sato, and S. Sakata, "ManueverXML:Distance-Measurement based Operation Event Description Model and User Interaction Interpretation," Proc. the 10th Annual Symposium on Application and the Internet(SAINT2010) WS-1(EUCASS), July 2010.