

# A Sensor Network Protocol for Automatic Meter Reading in an Apartment Building

Tetsuya Kawai<sup>1</sup> and Naoki Wakamiya<sup>1</sup> and Masayuki Murata<sup>1</sup> and Kentaro Yanagihara<sup>2</sup> and Masanori Nozaki<sup>2</sup> and Shigeru Fukunaga<sup>2</sup>

<sup>1</sup> Graduate School of Information Science and Technology, Osaka University, Japan  
{t-kawai, wakamiya, murata}@ist.osaka-u.ac.jp

<sup>2</sup> Corporate R&D center, Oki Electric Industry Co., Ltd., Japan  
{yanagihara726, nozaki765, fukunaga444}@oki.com

**Abstract.** A wireless sensor network for automatic meter reading needs to satisfy two contradicting requirements, i.e., long lifetime and prompt detection and notification of emergency. We propose a sensor network protocol for this purpose, in which sensor nodes operate on a low duty cycle while the latency of transmission is guaranteed to be less than a certain bound. In this protocol, each node is assigned a time slot in which it receives messages from other nodes. To accomplish slot assignment where nodes further from a BS are assigned earlier time slots for a packet to be transmitted to the BS in one cycle, we propose a slot assignment function with which a node can determine its own slot in a distributed way. We explore several slot assignment functions to find one which gives low and homogeneous contention over a grid network. The simulation results show that our protocol performs well close to the optimal case.

## 1 Introduction

Wireless sensor network (WSN) is a key technology to realize our safe, secure, and comfortable future life. Among its wide variety of applications, we focus our attention on automatic meter reading (AMR) in a large-scale apartment building which consists of hundreds of apartments. In such a building, meters are attached to pipes and cables in each apartment to monitor consumption of water, gas, electricity, and so on. A water, gas, or electricity company hires and sends personnel to collect meter readings once a month in most cases. In these days, those companies consider to adopt a WSN for meter reading, since it is costly to hire many personnel to cover all houses and buildings of customers and it becomes difficult for outsiders to enter modern apartment buildings for security reasons.

Such a WSN for AMR consists of meters equipped with a radio transceiver operated on battery power supply and a gateway server, i.e., base station (BS), connected to a monitoring station of a company through a regional wired or wireless network. Since a meter is usually stored in a meter box or pipe shaft, radio signal is heavily disturbed and the range of transmission is relatively small. Therefore, a WSN is sparse, where the average number of neighboring nodes, i.e., nodes in the range of radio signals, is a few. Monthly meter reading does not put too strict restriction on the delay requirement as far as meter reading is collected from all meters in a building. On the contrary, urgent

information such as detection of gas leakage must be transmitted to the BS immediately, e.g., within 10 seconds, once it happens.

In this paper, we consider a network protocol for AMR, which can collect meter reading from all meters at a predetermined interval, e.g., once a month, and transmit urgent information from a meter to a BS within the specified delay bound. We assume that a contention-based MAC is adopted and that timers of nodes are synchronized by a certain synchronization protocol, such as proposed in [5, 6, 11]. In addition, a WSN is maintained by a certain topology control protocol, together with a health check mechanism conducted once a day, for example. Under these assumptions, we consider sleep scheduling for energy-efficient operation of a WSN.

The trade-off between latency and energy-efficiency has been discussed in some literatures [2–4, 9]. Sleep scheduling to reduce latency is also proposed in [1, 8, 10]. However, they do not consider reliability of data transmission. Both loss for unreliable wireless communication and latency caused by sleeping nodes are considered in the dynamic switch-based forwarding [7], which optimizes the expected delivery ratio, expected communication delay, or expected energy consumption. However, this forwarding method does not provide any guarantee on end-to-end delay. We aim to satisfy both requirements on bounded end-to-end delay and low duty cycle by waking up nodes from the edge of a WSN to the BS adopting a distributed and self-organizing slot assignment mechanism.

The rest of this paper is organized as follows. First we describe the details of our protocol in Sect. 2. In Sect. 3, we define the contention degree, which represents the intensity of contention at each node, and expected contention degree in a grid network is presented. Next we evaluate four slot assignment functions in Sect. 4 and the best one is further tuned for a grid network in Sect. 5. We conclude this paper in Sect. 6.

## 2 Sensor Network Protocol for AMR

To save energy consumption, making nodes sleep is one of the primary techniques. To detect an event and notify a BS of it, a node has to wake up at least once per the delay bound, denoted as  $d_{\max}$  hereafter. Then, all nodes on the path from the detecting node to the BS must wake up at appropriate timing so that the urgent information is relayed to the BS immediately.

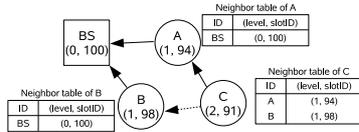
Now consider the case that the duration of one operation cycle,  $d_{\max}$ , is divided into time slots of duration  $t_s$ . In most of TDMA-based scheduling, a slot is assigned to each node for packet transmission. A node can occupy the wireless channel during the assigned slot and send a packet without being disturbed by other nodes. This scheduling enables efficient usage of the wireless channel, but it is inefficient from an energy point of view. Since a node does not know which node has a packet to relay, it must listen to the wireless channel in all of slots assigned to its neighbors. In a WSN, such idle listening is the major drain of energy, especially when packets are generated intermittently as for AMR.

Our approach to shorten this idle listening is to assign time slots not for sending but for receiving to lower the duty cycle of nodes. With this slot assignment, each node has to be awake during only one slot per cycle even if it has two or more child nodes. In

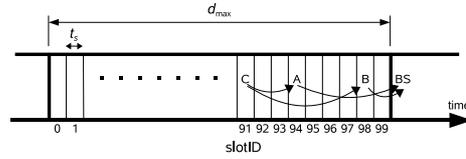
our sensor network protocol for automatic meter reading, called Sleep Scheduling by Distributed Slot Assignment (SSDSA), every node keeps awake for  $t_s$  at an interval of  $d_{\max}$  for possible packet reception. The total number  $N$  of slots is given by  $N = d_{\max}/t_s$ . We define slotID  $0 \leq k \leq N - 1$  where a smaller slotID corresponds to an earlier slot in a cycle of  $d_{\max}$ . The duration  $t_s$  is determined to be long enough for MAC layer to deliver a packet to a next-hop node, including carrier sense, MAC level acknowledgement, and retransmissions. For example, its typical value would be between 100 ms and 200 ms for IEEE 802.15.4 and smaller for IEEE 802.11. If  $t_s = 100$  ms and  $d_{\max} = 10$  seconds, the duty cycle is 1/100. Even lower duty cycle can be achieved by employing an energy-aware low duty MAC protocol such as [12]. In SSDSA, a node further from a BS obtains an earlier slot with a smaller slotID. As far as every node has a next-hop node which has a slot of a larger slotID, a packet originating at any node can reach the BS within  $d_{\max}$ .

The details of SSDSA are as follows:

1. New node  $i$  first tries to discover neighbor nodes by a neighbor discovery protocol employed. Neighbor discovery is out of scope of this paper.
2. When node  $i$  discovers neighbor  $j$ , node  $j$  notifies node  $i$  of its ID, level  $l_j$ , which corresponds to the hop distance from the BS, and slotID  $k_j$ . We assume that the BS has a power supply and does not sleep. The BS declares slotID  $N$  and the level 0. Node  $i$  stores the received information in a neighbor table. Neighbors are listed in ascending order of the level as the first key and the slotID as the second key (see Fig. 1). A time synchronization process could be conducted in this stage. In the rest of this paper, we refer a parent or child node as a direct neighbor of a node which is one hop closer or further to the BS than the node itself, respectively. A next-hop node is one of parent nodes to which a node sends a packet and a preceding node is one of child nodes from which a packet is received.
3. After completing the neighbor discovery process, node  $i$  determines its own level  $l_i$  and slotID  $k_i$  as  $l_i = l_i^1 + 1$  and  $k_i = f(k_i^1)$ , where  $l_i^1$  and  $k_i^1$  are the level and slotID of the node at the top of the neighbor table, respectively.  $f(k)$  is called the slot assignment function (SAF), which gives a smaller value than  $k$  following the slot assignment probability distribution function (SAPDF), which will be discussed later. Level  $l_i + 1$  nodes and level  $l_i$  nodes with slotID equal or smaller than the maximum among level  $l_i - 1$  nodes are removed from the neighbor table. In Fig. 1, node B does not have node C in its neighbor table for this reason. Then, node  $i$  sends these information to its neighbor nodes of the same level at their slots. A neighbor node which receives this notification adds node  $i$  to its neighbor table, if slotID  $k_i$  is greater than the maximum slotID of its parent nodes.
4. Once node  $i$  determines its time slot  $k_i$ , node  $i$  wakes up at the slot, keeps active for  $t_s$ , and goes back to sleep.
5. With the slot assignment stated above, a node can have two or more next-hop candidates for improving reliability of transmission [7]. If node  $i$  receives a packet during its time slot or it has a packet to send, it wakes up again at the next-hop node's time slot, i.e.,  $k_i^1$ , and sends the packet. If node  $i$  fails in transmitting a packet to the first next-hop node, it tries the second next-hop node in the neighbor table at slot  $k_i^2$  (see node C in Fig. 2). If the transmission fails again, it tries third one. It repeats this



**Fig. 1.** An example of level and slotID assignment in SSDSA.



**Fig. 2.** Scheduled packet transmission.

procedure until the transmission succeeds or fails at the last next-hop node in the neighbor table.

Possible congestion around the BS could be avoided, for example, by giving level 1 nodes even slotIDs and having them transmit a packet at the following time slot of an odd slotID. Note that, in SSDSA, sleep scheduling of nodes and routing are integrated in the time slot assignment. Slot assignment is done in a fully-distributed and self-organizing manner by using neighbor information.

### 3 Contention Degree

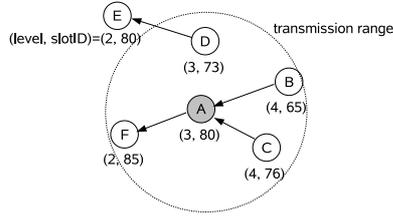
#### 3.1 Definition

In SSDSA, it is possible that two or more nodes transmit packets at the same time, since time slots are assigned not for transmission but for reception. Moreover, since each node determines its own slotID in a distributed manner, two or more nodes could have the same slotID. These could degrade the reliability of transmission and lead to extra energy consumption in the MAC layer. Therefore, we need to balance the degree of contention among nodes.

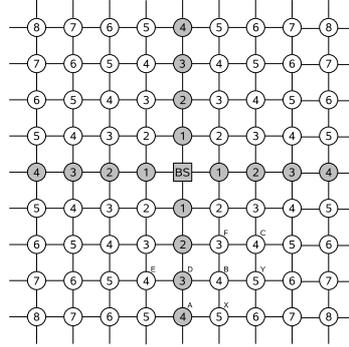
We define the contention degree  $C$  of a node as the number of neighbors which can transmit a packet at the time slot given to the node. For example, in Fig. 3, the contention degree of node A is three, where neighboring nodes B, C, and D compete for slot 80. Although we consider only the first next-hop node to compute the contention degree in the following discussion, it is straightforward to extend this idea to involve the second and more next-hop nodes by weighing contribution to the contention degree, 1.0 for the first next-hop node and 0.1 for the second next-hop node for example.

#### 3.2 Contention Degree in a Grid Network

First, we calculate the expected contention degree in a grid network, considering the arrangement of apartments in a building. In a grid network shown in Fig. 4, each circle represents a node and a number inside denotes its level. Each line corresponds to a bidirectional link among nodes. The contention degree  $C_B$  of node B having two child nodes X and Y is 0, 1, or 2, depending on slotIDs of nodes A, B, and C. For example, for  $k_A > k_B > k_C$ ,  $C_B = 1$ , since node X chooses node B and node Y chooses node C as their first next-hop node.



**Fig. 3.** An example of the contention degree.



**Fig. 4.** A grid network.

Now let  $q$  the probability that two nodes of level  $l$  have the same slotID. The probability that two level  $l$  nodes have different slotIDs becomes  $(1 - q)/2$ . Then,  $P(k_A > k_B > k_C) = \{(1 - q)/2\}^2$ . Calculating the probability of all nine combinations of  $k_A$ ,  $k_B$ , and  $k_C$ , the expected contention degree of node B is given as  $E(C_B) = 1 + q$ . Next we consider nodes represented by a gray circle in Fig. 4. They have three child nodes and one parent node. For example, node D has child nodes A, B, and E. We can apply similar discussion as above for nodes B and E. On the other hand, node A always chooses node D as its next-hop node. Therefore, the expected contention degree of node D is given as  $E(C_D) = 2 + q$ . We refer a gray node as “m-node” (main stream node) and an open-circle node as “b-node” (branch stream node), and we define the expected contention degree of m-nodes,  $E(C_m)$ , and of b-nodes,  $E(C_b)$ , as

$$E(C_m) = 2 + q, E(C_b) = 1 + q. \quad (1)$$

## 4 Slot Assignment Probability Distribution Functions

### 4.1 Evaluation Metrics

In the rest of this paper, we explore SAFs which lead to lower and more homogeneous contention degree. We employ following four evaluation metrics.

- $E_l(C)$ . The average contention degree of level  $l$  nodes.
- $V_l(C)$ . The variance of the contention degree of level  $l$  nodes.
- $p_{\text{empty}}$ . The ratio of unassigned time slots.
- $q_{\text{isolated}}$ . When all of parent nodes are assigned the time slot 0, a node cannot obtain its time slot and it is isolated.  $q_{\text{isolated}}$  is the ratio of isolated nodes.

### 4.2 Examples of SAPDF

In this subsection, we consider SAPDF  $g^k(x)$  which determines SAF  $f(k)$ .  $g^k(x)$  gives the probability that slotID  $x$  is assigned to a node when the slotID of its first next-hop

node is  $k$ . Letting  $F_l(x)$  denote the probability distribution of nodes (PDN) that a node of level  $l$  is assigned slotID  $x$ , we have

$$F_l(x) = \frac{1}{n_l} \sum_{i \in \mathcal{S}_l^*} g^{k_i}(x), \quad (2)$$

where  $n_l$  and  $\mathcal{S}_l^*$  are the number of level  $l$  nodes and a set of their first next-hop nodes, respectively. Assuming the distribution of slotIDs of nodes in  $\mathcal{S}_l^*$  is identical to that of all level  $l-1$  nodes,  $F_{l-1}(x)$ , (2) can be rewritten as

$$F_l(x) = \sum_{k=0}^{N-1} g^k(x) F_{l-1}(k). \quad (3)$$

The duplication probability  $q$  in (1) can be derived as

$$q = \sum_{x=0}^{N-1} \{g^k(x)\}^2. \quad (4)$$

We consider the following four typical examples of SAPDFs.

**K-1** If the slotID of the first next-hop node is  $k$ , slotID  $k-1$  is assigned.

$$g^k(x) = \begin{cases} 1 & (x = k-1) \\ 0 & (0 \leq x \leq k-2, k \leq x \leq N-1). \end{cases} \quad (5)$$

According to K-1, all level  $l$  nodes have the identical slotID  $N-l$ . Therefore, K-1 leads to the worst case scenario.

**L-BOUND** The lower bound  $L_l$  of slotID for level  $l$  nodes is predetermined. A slotID is randomly chosen between  $L_l$  and  $k-1$ .

$$g^k(x) = \begin{cases} \frac{1}{k-L_l} & (L_l \leq x \leq k-1) \\ 0 & (0 \leq x \leq L_l-1, k \leq x \leq N-1). \end{cases} \quad (6)$$

**LINEAR** A slotID is randomly chosen between 0 and  $k-1$  according to linear distribution as shown in Fig. 5(a).

$$g^k(x) = \begin{cases} \frac{2}{k(k+1)}(x+1) & (0 \leq x \leq k-1) \\ 0 & (k \leq x \leq N-1). \end{cases} \quad (7)$$

PDN  $F_l(k)$  of LINEAR for level 1 through 3 calculated by (3) and (7) is shown in Fig. 5(b).

**EXPONENTIAL** A slotID is randomly chosen between 0 and  $k-1$  following the exponential distribution as shown in Fig. 6.

$$g^k(x) = \begin{cases} e^{-\lambda_k\{k-(x+1)\}} - e^{-\lambda_k(k-x)} & (k \neq 1, 0 \leq x \leq k-1) \\ 0 & (k \neq 1, k \leq x \leq N-1). \end{cases} \quad (8)$$

SlotID 0 is assigned for  $k=1$ .

Now consider the probability  $q$  for  $k=100$ . In K-1, all nodes of the same level have the same slotID, thus  $q=1$ . In L-BOUND,  $q$  heavily depends on  $L_l$  and is given as  $1/(100-L_l)$ . In LINEAR,  $q$  becomes 0.013. For EXPONENTIAL, when we set  $\lambda_k$  to  $11.5/(k-1)$  to have the ratio of isolated nodes at level 10 less than 0.01 % following the discussion in Sect. 4.3,  $q$  becomes 0.058.

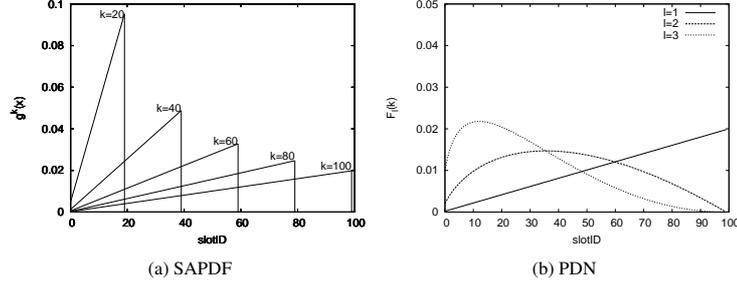


Fig. 5. (a) SAPDF and (b) PDN of LINEAR.

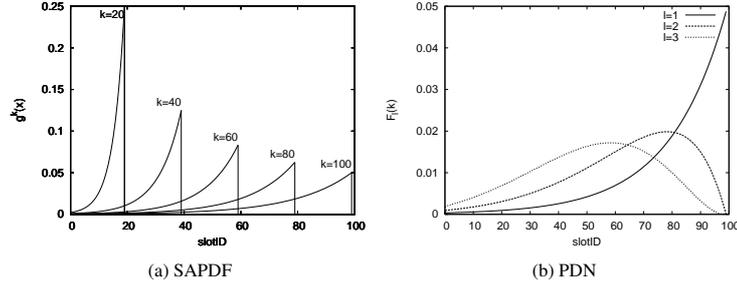


Fig. 6. (a) SAPDF and (b) PDN of EXPONENTIAL ( $\lambda_k = 5/(k-1)$ ).

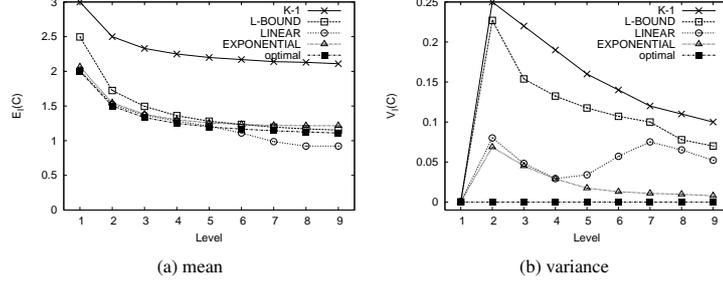
### 4.3 Parameter Determination of EXPONENTIAL SAPDF

In EXPONENTIAL, parameter  $\lambda_k$  determines the distribution of slotIDs. With small  $\lambda_k$ , slotIDs are widely distributed, with which probability  $q$  becomes small and the contention degree becomes low. However, as an adverse effect, it generates many isolated nodes by assigning a small slotID to a low level node. Therefore, we need to find appropriate  $\lambda_k$  which guarantees the maximum ratio of isolated nodes of level  $l$  at the desired value  $1 - P^*$ . We define  $P_l$  as the probability that slotID  $k = [1, N-1]$  is assigned to a level  $l$  node, namely,

$$P_l = \sum_{i=1}^{N-1} F_l(i) = 1 - F_l(0). \quad (9)$$

Among level 2 nodes, those who have a level 1 node with slotID  $k = 0$  as their next-hop node cannot obtain a valid slotID and they are isolated. For simplicity, ignoring a case that a level 2 node has two or more parent nodes,  $1 - P_1$  of level 2 nodes are isolated. Among the rest,  $P_2$  get slotID  $k = [1, N-1]$  and  $1 - P_2$  get slotID  $k = 0$ . Therefore, among all level 2 nodes, the probability of getting slotID  $k = [1, N-1]$  and  $k = 0$  are  $P_1 P_2$  and  $P_1(1 - P_2)$  respectively. Assuming  $P = P_1 = P_2 = \dots = P_l$  for simplicity, the ratio of level  $l$  nodes with a valid slotID is  $P^{l-1}$ . Since this must be larger than  $P^*$ , we have

$$P > \sqrt[l-1]{P^*}. \quad (10)$$



**Fig. 7.** The contention degree.

From (3) and (9), we get

$$P = \sum_{i=1}^{N-1} F_l(i) = \sum_{i=1}^{N-1} \left( \sum_{k=0}^{N-1} g^k(i) F_{l-1}(k) \right) = \sum_{k=0}^{N-1} \left( F_{l-1}(k) \sum_{i=1}^{N-1} g^k(i) \right). \quad (11)$$

If

$$\sum_{i=1}^{N-1} g^k(i) > l^{-1} \sqrt{P^*} \quad (12)$$

holds,

$$P = \sum_{k=0}^{N-1} \left( F_{l-1}(k) \sum_{i=1}^{N-1} g^k(i) \right) > l^{-1} \sqrt{P^*} \sum_{k=0}^{N-1} F_{l-1}(k) = l^{-1} \sqrt{P^*}. \quad (13)$$

Therefore, (12) is a sufficient condition for (10). For EXPONENTIAL, from (8),

$$\sum_{i=1}^{N-1} g^k(i) = 1 - e^{-\lambda_k(k-1)}. \quad (14)$$

Substituting (14) into (12), we finally get

$$\lambda_k > - \frac{\log(1 - l^{-1} \sqrt{P^*})}{k-1}. \quad (15)$$

#### 4.4 Simulation Experiments

We compare the four SAPDFs by simulation experiments. 220 nodes up to level 10 are arranged in a grid network centered at a BS. The number of time slots  $N$  is set to 100. For L-BOUND, the lower bound  $L_l$  is determined according to the number of nodes belonging to level  $l$ ,  $(L_{l-1} - L_l)/(L_l - L_{l+1}) = l/(l+1)$  where  $L_0 = N$  and  $L_{10} = 0$ . For EXPONENTIAL,  $\lambda_k = 11.5/(k-1)$ . Shown results are averaged over 500 simulation runs.

In the optimal case, the contention degree of all level  $l$  nodes should be equal to the average number of level  $l+1$  nodes per node, thus

$$E_l^{\text{opt}}(C) = (l+1)/l, V_l^{\text{opt}}(C) = 0. \quad (16)$$

**Table 1.** Evaluation metrics for four SAPDFs.

	K-1	L-BOUND	LINEAR		EXPONENTIAL		
					$r = 2.0$	3.0	4.0
$p_{\text{empty}} (\%)$	90.0	11.8	65.9	29.4	31.9	33.7	35.1
$q_{\text{isolated}} (\%)$	0	0	41.0	0.008	0.006	0	0

The results are shown in Fig. 7 and Table 1. As shown in Fig. 7(a), K-1 gives the largest contention degree, whereas there is no isolated node as shown in Table 1. L-BOUND can efficiently assign slots, but the contention degree and its variance are not sufficiently small. LINEAR accomplishes the smallest contention degree at high level nodes. However, it is due to isolated nodes. In Fig. 5(b), the peak of PDN for level 3 is at  $k = 12$ , which means that higher level nodes have little choice of slotIDs. Consequently, 41 % of nodes are isolated and do not contend for the wireless channel. We can see that the graph of EXPONENTIAL is very close to that of the optimal and the best among four SAPDFs.

The variance  $V_l(C)$  of L-BOUND is higher than that of LINEAR or EXPONENTIAL as shown in Fig 7(b). Basically, an m-node has larger contention degree than a b-node as indicated by (1) if they are equally chosen as a next-hop node. However, at higher levels in LINEAR or EXPONENTIAL, a b-node has a smaller slotID than an m-node for having more parent nodes and it is chosen as a next-hop node more often. Then, the difference of the contention degree between m-nodes and b-nodes becomes smaller. On the contrary, in L-BOUND, a node randomly selects a slotID within a pre-determined range of time slots. EXPONENTIAL gives the smallest variance among the four SAPDFs.

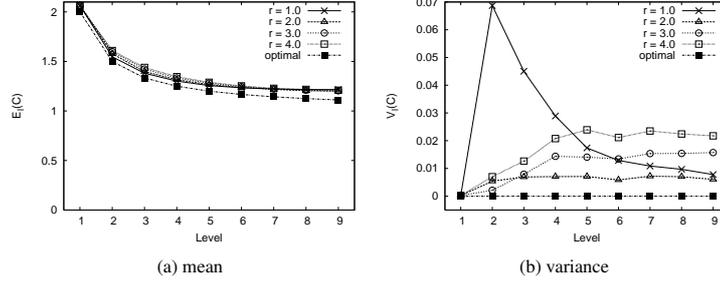
## 5 Optimization of Exponential Distribution

As shown in the previous section, EXPONENTIAL gives the near-optimal performance. However, m-nodes have higher contention degree than b-nodes for having more child nodes especially at lower levels. We consider to make the contention degree same or similar among m-nodes and b-nodes by using different parameters.

### 5.1 Stochastic Analysis

The basic idea is to make a node which has both m-nodes and b-nodes as its parents choose a b-node as its next-hop node with a higher probability. For this purpose, we employ a larger coefficient of the exponential distribution for m-nodes to get a larger slotID than those of b-nodes.

In a grid network shown in Fig. 4, introducing probability  $p$  that an m-node has a larger slotID than that of a b-node of the same level, probability  $q_1$  that an m-node has an equal slotID to that of a b-node, and probability  $q_2$  that two b-nodes have the identical slotID, we obtain the expected contention degree  $E(C_A)$ ,  $E(C_B)$ , and  $E(C_C)$  of m-node A, b-node B which has an m-node of the same level as its two-hop neighbor, and



**Fig. 8.** The contention degree with the parameter tuning.

b-node C which does not have any m-nodes of the same level as its two-hop neighbors, respectively. After similar calculation as in Sect. 3.2, we get

$$E(C_A) = 3 - 2p, E(C_B) = p + q_1 + \frac{1 + q_2}{2}, E(C_C) = 1 + q_2. \quad (17)$$

For  $E(C_A) = E(C_B) = E(C_C)$ , erasing  $q_2$ , we obtain

$$2p + q_1 = \frac{3}{2}. \quad (18)$$

This is a necessary condition for the identical contention degree among nodes of the same level. Now, let us assume that an m-node and a b-node have their next-hop node with slot  $k$  and their slotIDs are assigned by SAPDFs  $g_m^k$  and  $g_b^k$  respectively. The probabilities  $p$  and  $q_1$  can be calculated as

$$p = \sum_{x=1}^{N-1} \left( g_m^k(x) \sum_{i=0}^{x-1} g_b^k(i) \right), q_1 = \sum_{x=0}^{N-1} g_m^k(x) g_b^k(x). \quad (19)$$

In order to assign a larger slotID to an m-node, we employ a larger coefficient for  $g_m^k$  than  $g_b^k$  by introducing a parameter  $r \geq 1$  as,

$$g_m^k(x) = \begin{cases} e^{-r\lambda_k\{k-(x+1)\}} - e^{-r\lambda_k(k-x)} & (0 \leq x \leq k-1) \\ 0 & (k \leq x \leq N-1) \end{cases} \quad (20)$$

while  $g_b^k(x)$  is the same as (8). Using (19) and (20), we can derive the optimal  $r$  to have (18), although results are not shown for space limitation.

## 5.2 Simulation Experiments

The mean  $E_l(C)$  and variance  $V_l(C)$  of the contention degree at level  $l$  are plotted in Fig. 8 by changing  $r$ .  $E_l(C)$  slightly increases with  $r$ , because  $q_1$  in (17) increases.  $V_l(C)$  dramatically decreases by changing  $r$  compared to the case without the parameter tuning ( $r = 1.0$ ). It means that the contention degree is well equalized among m-nodes and b-nodes. At level 3 or above,  $r = 2.0$  gives better results than  $r = 3.0$  does. As mentioned

in Sect. 4.4, even in the case without the parameter tuning, slotIDs of b-nodes tend to be smaller than those of m-nodes as the level increases. Increasing  $r$  enhances this effect even further, thus  $p$  becomes too large to satisfy (18) with  $r \geq 3.0$  at higher levels. Table 1 shows  $p_{\text{empty}}$  and  $q_{\text{isolated}}$  for each of four  $r$  values. As  $r$  increases,  $p_{\text{empty}}$  also increases, since m-nodes are distributed within a narrower range of time slots, in other words, more packed.

## 6 Conclusion

In this paper, a WSN protocol for AMR in a large-scale apartment building was proposed. For accomplishing both a low duty cycle and delay-bounded transmission, the operation interval is set at the delay bound and divided into time slots. A node determines its time slot for packet reception in a distributed and stochastic manner, so that every node has a next-hop node with a larger slotID. It was shown that EXPONENTIAL SAPDF enabled less and more identical contention degree at all level nodes in a grid network and the results were near-optimal. Our future work includes evaluation of energy efficiency of our protocol in practical experiments, including comparison to a TDMA-type protocol, which we will report in near future.

**Acknowledgement** This research was supported in part by “Global COE (Centers of Excellence) Program” of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

## References

1. Cao, Q., Abdelzaher, T., He, T., Stankovic, J.: Towards optimal sleep scheduling in sensor networks for rare-event detection. In: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN 2005), pp. 20–27. Los Angeles, California, USA (2005)
2. Chiasserini, C.F., Garetto, M.: Modeling the performance of wireless sensor networks. In: Proceedings of 23rd Annual Joint Conference of the IEEE Computer and Communication Societies (INFOCOM 2004), pp. 220–231. Hong Kong (2004)
3. Cohen, R., Kapchits, B.: An optimal algorithm for minimizing energy consumption while limiting maximum delay in a mesh sensor network. In: Proceedings of the 26rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2007), pp. 258–266. Anchorage, Alaska, USA (2007)
4. Dousse, O., Mannersalo, P., Thiran, P.: Latency of wireless sensor networks with uncoordinated power saving mechanisms. In: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '04), pp. 109–120. Tokyo, Japan (2004)
5. Elson, J., Girod, L., Estrin, D.: Fine-grained network time synchronization using reference broadcasts. In: Proceedings of the 5th symposium on Operating systems design and implementation (OSDI '02), pp. 147–163. Boston, Massachusetts, USA (2002)

6. Ganeriwal, S., Kumar, R., Srivastava, M.B.: Timing-sync protocol for sensor networks. In: Proceedings of the 1st international conference on Embedded networked sensor systems (SenSys '03), pp. 138–149. Los Angeles, California, USA (2003)
7. Gu, Y., He, T.: Data forwarding in extremely low duty-cycle sensor networks with unreliable communication links. In: Proceedings of the 5th international conference on Embedded networked sensor systems (SenSys '07), pp. 321–334. Sydney, Australia (2007)
8. Keshavarzian, A., Lee, H., Venkatraman, L.: Wakeup scheduling in wireless sensor networks. In: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '06), pp. 322–333. Florence, Italy (2006)
9. Lai, W., Paschalidis, I.C.: Sensor network minimal energy routing with latency guarantees. In: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '07), pp. 199–208. Montreal, Quebec, Canada (2007)
10. Lu, G., Sadagopan, N., Krishnamachari, B., Goel, A.: Delay efficient sleep scheduling in wireless sensor networks. In: Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005), pp. 2470–2481. Miami, Florida, USA (2005)
11. Maróti, M., Kusy, B., Simon, G., Ákos Lédeczi: The flooding time synchronization protocol. In: Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys '04), pp. 39–49. Baltimore, MD, USA (2004)
12. Ye, W., Silva, F., Heidemann, J.: Ultra-low duty cycle mac with scheduled channel polling. In: Proceedings of the 4th international conference on Embedded networked sensor systems (SenSys '06), pp. 321–334. Boulder, Colorado, USA (2006)