

Copyright © 2006 IEEE. Reprinted from
International Conference on Advanced Information Networking and
Applications (20th : 2006 : Vienna, Austria)

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Adelaide's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org.

By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

Distributed Tuning Attempt Probability for Data Gathering in Random Access Wireless Sensor Networks

Haibo Zhang, Hong Shen
Graduate School of Information Science
Japan Advanced Institute of Science and Technology
1-1, Asahidai, Nomi, Ishikawa 923-1211, Japan
{haibo,shen}@jaist.ac.jp

Abstract

In this paper, we study the problem of data gathering in multi-hop wireless sensor networks. To tackle the high degree of channel contention and high probability of packet collision induced by bursty traffic, we introduce a novel model based on random channel access scheme for data gathering. In our model, both data delivery reliability and latency are considered, and our goal is to tune the attempt probability for each sensor node so that the data gathering duration can be minimized on condition that each link in the data gathering tree can provide guaranteed per-hop packet delivery reliability. We formulate this problem as an optimization problem and propose a distributed heuristic algorithm which exploits only two hop neighbors information to solve it for tree networks. We evaluate the algorithm and the model by simulations, and results show that our algorithm has low computational complexity and our model can provide a good trade-off between reliability and latency for data gathering.

1. Introduction

Wireless sensor networks (WSNs) are expected to be used in a wide range of applications, such as habitat sensing, healthcare monitoring, target tracking, etc.. An important task for such scenarios is data gathering, where each sensor must periodically report its sensed data to the central node, or base station (BS). For many applications, the sensed data from different locations is expected to be transported to the BS with high reliability and low latency.

The advances in low cost and low power RF communication have enabled the deployment of large-scale sensor networks. Due to the low transmission power, the distance between a sensor node and the BS may exceed its transmission range. Relaying packets by intermediate nodes needs to be performed and data gathering must be operated in a multi-hop fashion. During data gathering, the large burst of

data packets generated within a very short time will lead to high degree of channel contention and high probability of packet collision. The situation is further exacerbated by the fact that packet is delivered over multi-hop routes. Consequently, the packet loss rate is very serious in data gathering. For example, in [9], the authors observed that around 50% packets are lost for most events in Lites with the default radio stack of TinyOS.

To improve packet delivery reliability, the RTS/CTS handshake is employed in IEEE802.11 Distributed Coordination Function (DCF) to counter the packet loss. However, RTS/CTS handshake can improve system performance only when large data frames are considered. But in most data gathering WSNs, the packet generated by the sensor is usually small, for example, the message generated by sensors deployed for monitoring the environment temperature may only contain several bits data besides the header. Therefore, the short control frames may further increase the channel contention and cause more data loss. To avoid packet collision, efficient scheduling algorithms have also been proposed for data delivery in WSNs [2][6]. However, due to the large scale of WSNs, finding optimal collision-free schedules for packet transmission is not easy.

Random access scheme is seen as better solution to bursty packet networks due to its simplicity and low delay (under light load) for bursty traffic [8]. It is well known that the random access channel throughput increases initially with increasing aggregate traffic generation rate [5]. Moreover, in most random access schemes, no additional control frames are used for channel reservation. Therefore, the channel utilization may be greatly improved.

In our work, we study the data gathering problem under random channel access model. We introduce a novel mathematical model for data gathering which considers both packet delivery reliability and latency. In our model, each node v_i simply determines to transmit a packet with probability α_i and decides to receive a packet with probability $1 - \alpha_i$ at any time slot during data gathering, where α_i is called *attempt probability* [4] for node v_i . We do not

insist that all the sensors use the same *attempt probability* first because maintaining the same *attempt probability* at every sensor is difficult. More importantly, different sensors may need different values of α_i to counter the local inhomogeneities in nodes deployment.

To perform data gathering task, tree structure – a multiple-source single-sink communication paradigm – is always employed. In [1], a hop-tree is constructed to gather sensor data. In [10], an algorithm for constructing a data gathering tree which can provide good trade-offs between reliability and latency was proposed. In this paper, we assume that the tree for data gathering has already been constructed. Our goal is to compute the optimal attempt probability for each sensor node so that the data gathering duration can be minimized on condition that each link in the tree can provide guaranteed hop-by-hop delivery reliability. We formulate this problem as an optimization problem and propose a distributed heuristic algorithm to solve it.

The remainder of this paper is organized as follows. Section 2 reviews some related work. In section 3, we describe our model and formulate the problem as an optimization problem. Section 4 gives the distributed heuristic algorithm. Section 5 provides the simulation results and analysis. Finally, we conclude in Section 6.

2. Related Works

The most relevant works include [7,9,10]. In [4], a generalized gradient ascent algorithm was proposed to self-learn the optimal attempt probability to maximize the throughput for each node. Two iterative, decentralized algorithms were presented in [7] to compute the global optimal rates for the problem of max-min fair rate allocation in ALOHA networks. In [3], the author views the problem of optimizing the node attempt probability as a non-cooperative game and shows the existence of a Nash equilibrium giving an optimal channel access rate for each node. However, all these approaches only consider the performance for each single node, such as maximizing the throughput or obtaining fairness. In data gathering, each sensor node is not independent, and all the sensors must work in a cooperative manner since the quality of a link depends on the attempt probabilities of several nodes. Therefore, the approaches mentioned above are not suitable to solve our problem.

To the best of our knowledge, this is the first paper that addresses the attempt probability tuning problem for data gathering in random access wireless sensor networks. Since different applications of WSNs may have different requirements for reliability and latency, it is desired to propose a model in which reliability and latency tradeoffs can be tuned by the user. In our model, the tradeoff can be obtained by simply adjusting the per-hop packet delivery reliability.

3. Problem Formulation

In our model, we make the following assumptions about the wireless sensor networks.

- All sensors are equipped with the same radio transceiver and can transmit/receive on a common carrier frequency using omni-directional antennas. Each sensor node is assumed to be static and distinguishable by a unique identifier.
- Time is slotted and nodes are synchronized on time slots. The length of each time slot is equal to the duration of transmitting a packet. At each time slot, a node can be in two states: active and sleep. If node v_i is in active state, it decides to transmit a packet with probability α_i and decides to receive with probability $(1 - \alpha_i)$.
- Collisions occur if a node simultaneously receives transmissions from two or more of its neighbors. Although it is possible that the receiver may correctly decode the packet from the transmitter if the signal to interference ratio(SIR) exceeds a given threshold, in our model, we assume that all colliding packets will be lost.

We model the network as a directed graph $G = (V, E)$, where $V = (v_1, v_2, \dots, v_n)$ is the set of sensor nodes, and E is the set of wireless links. Let (i, j) denote the link from v_i to v_j . If v_i can receive packets from v_j , we also assume that v_j can receive packets from v_i . Let $N_1(i)$ and $N_2(i)$ be the set of one hop and two hop neighbors of node v_i respectively. In addition, we assume that there is only one BS in the network, and all the data generated by the sensor nodes should be reported to the BS during data gathering.

All sensor nodes should operate in a cooperative manner to perform the data collection task. Therefore, each node in the network should optimally attempt a transmission, which means that a node should neither be too aggressive in attempting a transmission (thereby risking a collision) nor be too conservative so as to miss the chance of successful transmission. To address the problem in detail, we first give some definitions.

Definition 1: At time slot t , $P(i, j)$ is defined as the probability that a packet issued by node v_i can be successfully delivered to node v_j .

$$P(i, j) = \alpha_i(1 - \alpha_j) \prod_{k \in N_1(j) \setminus \{v_i\}} (1 - \alpha_k) \quad (1)$$

It is obvious that node v_j can successfully receive a packet from v_i in time slot t if and only if node v_i is in transmit state, while node v_j and all its one hop neighbors except v_i are in non-transmit states at this time slot.

Let $S_{i,j}$ denote the number of time slots allocated to v_i for delivering its packet to v_j . $B_{i,j}$ denotes the number of

collision-free time slots among $S_{i,j}$ time slots. Since the states at different time slots are independent, $B_{i,j}$ can be modeled as a binomial random variable:

$$B_{i,j} \sim \text{Binomial}(S_{i,j}, P(i, j)) \quad (2)$$

Definition 2: $P(i, j, S_{i,j})$ is the function for computing the probability that v_j can receive at least one packet from v_i in the $S_{i,j}$ time slots.

$$\begin{aligned} P(i, j, S_{i,j}) &= P(B_{i,j} \geq 1) = 1 - P(B_{i,j} = 0) \\ &= 1 - [1 - \alpha_i(1 - \alpha_j) \prod_{k \in N_1(j) \setminus \{v_i\}} (1 - \alpha_k)]^{S_{i,j}} \end{aligned} \quad (3)$$

$P(i, j, S_{i,j})$ is called the *per-hop delivery reliability* (PHDR) in our model. Given threshold $\tau(i, j)$ for $P(i, j, S_{i,j})$, $S_{i,j}$ can be easily computed from $P(i, j, S_{i,j}) = \tau(i, j)$, i.e.,

$$S_{i,j} = \left\lceil \frac{\log(1 - \tau(i, j))}{\log[1 - \alpha_i(1 - \alpha_j) \prod_{k \in N_1(j) \setminus \{v_i\}} (1 - \alpha_k)]} \right\rceil \quad (4)$$

Let (V, T) be the data gathering tree constructed from G , where T is the set of directed links in the tree. Let $L(i)$ be the latency for node v_i to deliver a packet to the BS along the path in the tree. In our model, we assume that the packet generated by the sensor is small and each intermediate node can encapsulate the data it generates or receives into one packet. Since the time for data encapsulation is small and we ignore it when calculating the packet delivery latency. Denote by L the minimal duration for data gathering, therefore, $L = \max_{v_i \in V} L(i)$. Our goal is to compute the optimal attempt probability for each node that can minimize the duration of data gathering while providing guaranteed PHDR. Hence, we aim to solve the following optimization problem.

$$\begin{aligned} \text{Minimize} \quad & L = \max_{v_i \in V} L(i) \\ \text{s.t.} \quad & P(k, j, S_{k,j}) \geq \tau(k, j) \\ & 0 \leq \alpha_i \leq 1, \quad v_i \in V \end{aligned} \quad (5)$$

Where $\tau(k, j)$ is the user specified threshold for $P(k, j, S_{k,j})$. We refer this problem as *attempt probability tuning problem*(APTP).

4. Distributed algorithm for APTP

Let $F(i)$ be the father node of v_i , and $C(i)$ be the set of children of v_i . Each node v_i maintains a label $Ts(i)$, which denotes the maximum time slots for the descendants of v_i to transport a packet to v_i . Therefore, $Ts(i)$ can be defined as

$$Ts(i) = \max_{j \in C(i)} \{Ts(j) + S_{j,i}\} \quad (6)$$

Let v_1 be the BS. Obviously, $Ts(1) = \max_{i \in V} L(i)$, therefore, Problem 5 can be rewritten as

$$\begin{aligned} \text{Minimize} \quad & L = Ts(1) \\ \text{s.t.} \quad & P(k, j, S_{k,j}) \geq \tau(k, j) \\ & 0 \leq \alpha_i \leq 1, \quad v_i \in V \end{aligned} \quad (7)$$

Denote by α the vector of α_i , $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$. From Eq.4, the objective function in Problem 7 is a nonlinear function of α . Therefore, APTP is indeed a nonlinear optimization problem. It is well known that nonlinear models can be extremely difficult to optimize, and finding global optimum for this kind of problem still remains as an open problem. We propose a distributed heuristic algorithm to tune the attempt probability for each node in tree networks, which relies on only two hop local information.

In the following, we first describe the local data structure maintained at each node, the message communication between the nodes and the attempt probability update procedure performed at each node. Then, we present the distributed heuristic attempt probability tuning algorithm.

Local Data Structure: For each node in G , there may be multiple paths to deliver its packets to the BS. However, only the directed links in T are useful for data gathering.

Lemma 1 For tree networks, α_i will only influence $S_{k,j} \in \{S_{m,F(i)}, m \in C(F(i))\} \cup \{S_{n,i}, n \in C(i)\} \cup \{S_{l,p}, p \in C(i), l \in C(p)\}$.

Proof: From Eq.4, $S_{k,j}$ only depends on $\alpha_m, m \in \{j\} \cup N_1(j)$. Therefore, α_i will only influence $S_{n,l}$ where $i \in \{l\} \cup N_1(l)$. In the data gathering tree T , $i \in N_1(F(i)) \cup \{i\} \cup N_1(C(i))$. Hence, the lemma is proved.

From Lemma 1, all links that α_i influences are within two hop neighborhood of node v_i . In our algorithm, each node v_i in data gathering tree maintains a simple data structure $\langle PNTab \rangle$, which records the attempt probability for $v_j, j \in \{i\} \cup N_1(i) \cup N_2(i)$.

Communication: To maintain this table, at each time slot, if a node is in transmit state, it simply broadcasts $\langle PNTab \rangle$ to its neighbors. Otherwise, if node v_i is in receive state and successfully receives a $\langle PNTab \rangle$ message from its neighbor v_j , it only updates $\alpha_k, k \in \{j\} \cup N_1(j) \setminus \{i\}$ with the corresponding value in the received $\langle PNTab \rangle$.

Computation: During attempt probability computation, each node should adjust its attempt probability to minimize the data gathering duration L .

Lemma 2 Let $Ts^*(i)$ be the value of $Ts(i)$ when each node v_i has obtained the the optimal attempt probability α_i^* , then

$$Ts^*(j) + S_{j,i} = Ts^*(k) + S_{k,i}, \quad j \in C(i), k \in C(i). \quad (8)$$

Proof: Suppose that there are two node $v_j \in C(i), v_k \in C(i)$, where $Ts^*(j) + S_{j,i} \neq Ts^*(k) + S_{k,i}$ when each

node has been assigned with the optimal attempt probability. Without loss of generality, suppose $Ts^*(j) + S_{j,i} < Ts^*(k) + S_{k,i}$. From Lemma 1, α_k will only influence $\{S_{p,F(k)}, p \in C(F(k))\}$, $\{S_{q,k}, q \in C(k)\}$ and $\{S_{l,n}, n \in C(k), l \in C(n)\}$. For $S_{k,F(k)}$, it is a strict increasing function with respect to α_k since v_k acts as sender for link $(k, F(k))$. While for all other $S_{m,n}$ that α_k influences, $S_{m,n}$ is strict decreasing function with respect to α_k since v_k must be in non-transmit state for these links. Because $S_{j,i} \in \{S_{p,F(k)}, p \in C(F(k))\}$, there must be $Ts'(i)$ that $Ts'(1) < Ts^*(1)$ when α_k is assigned with a smaller value than α_k^* , which implies that α^* is not optimal. Hence, the lemma is proved.

Theorem 1 Let α_i^* be the optimal attempt probability for node v_i , then

$$\alpha_i^* = \begin{cases} 0, & i = 1 \\ \frac{1}{|C(F(i))|}, & i \neq 1 \end{cases} \quad (9)$$

Proof: If $i=1$, v_1 is the BS. During data gathering period, the BS needn't to transmit any packet and all the time slots should be used to receive packets from other nodes. Hence, $\alpha_1^* = 0$.

For any node $v_i \in C(F(i))$, if $\forall v_j \in C(F(i)), C(j) = \emptyset$, then according to Lemma 2 and Eq.10, we can get $S_{k,F(k)} = S_{m,F(m)}, k, m \in C(F(i))$.

Let $S_1 = C(F(i)) \setminus \{i\}$ and $S_2 = \{F(i)\} \cup N_1(F(i)) \setminus C(F(i))$, $S_{i,F(i)}$ can be rewritten as

$$S_{i,F(i)} = \left\lceil \frac{\log(1-\tau(i,F(i)))}{\log[1-\alpha_i \prod_{k \in S_1} (1-\alpha_k) \prod_{m \in S_2} (1-\alpha_m)]} \right\rceil$$

Since for all $S_{i,F(i)}, i \in C(F(i)), \prod_{m \in S_2} (1-\alpha_m)$ is the same. Therefore, $\alpha_k = \alpha_m = \alpha, k, m \in C(F(i))$, and $Ts(F(i)) = S_{i,F(i)}$. It is easy to prove that $Ts(F(i))$ is minimized only when $\alpha_i = \frac{1}{|C(F(i))|}$. From Lemma 1, α_i will only influence $\{S_{j,F(i)}, j \in C(F(i))\}$. Hence, the theorem is proved.

For the other nodes, we design a heuristic distributed algorithm to compute the attempt probability. The algorithm includes two phase: Self-stabilization and Latency Balance.

Definition 5: Given node $v_i, i \in V \setminus \{v_1\}, C(i) \neq \emptyset$, let $D(i) = j$, where $Ts(j) + S_{j,i} = \max_{k \in C(i)} (Ts(k) + S_{k,i})$. We call the path from $D(D(i))$ to $F(i)$ the critical path for node v_i . Then we define $f(\alpha_i)$ as the function for calculating the latency that a packet is delivered along the critical path.

$$f(\alpha_i) = S_{D(D(i)),D(i)} + S_{D(i),i} + S_{i,F(i)} \quad (10)$$

If $C(D(i)) = \emptyset$, then $D(D(i)) = D(i)$. From Eq.(10), the critical path has an important impact on the data gathering duration L . At the Self-Stabilization phase, each intermediate node tries to adjust its attempt to minimize the packet delivery latency along its critical path. Since $f(\alpha_i)$

is a strict concave function with respect to α_i , there is an optimal α_i' at each iteration that can minimize $f(\alpha_i)$, where α_i' satisfies $\frac{\partial f(\alpha_i)}{\partial \alpha_i} |_{\alpha_i'} = 0$. Therefore, once a node v_i receives a $\langle PNTab \rangle$ message from its neighbors, it simply updates α_i with α_i' .

According to Lemma 2, we use the latency balance phase to further reduce the data gathering duration L . Given node v_i , each node $v_j \in C(i) \setminus D(i)$ decreases α_j step by step as long as $Ts(j) + S_{j,i}$ is not larger than $Ts(D(i)) + S_{D(i),i}$. While for node $v_{D(i)}$, it just updates $\alpha_{D(i)}$ so that $f(\alpha_{D(i)})$ is minimized. The whole algorithm is given as follows.

Algorithm 2: Tune α for Tree Network(node u)

- 1: Initialize $\alpha_u, \bar{\alpha}_u$ and Δ ;
 - 2: state=GenerateState($\bar{\alpha}_u$);
 - 3: If state=Transmit then
 - 4: broadcast(u, $Ts(u)$, $\langle PNTab \rangle$);
 - 5: If state=Receive then
 - 6: If SuccessReceive(v, $Ts(v)$, $\langle PNTab \rangle$);
 - 7: Update $\alpha_m, m \in \{v\} \cup N_1(v) \setminus \{u\}$;
 - 8: Case Self-Stabilization:
 - 9: $\alpha_u = \alpha'_u$, where $\frac{\partial f(\alpha_i)}{\partial \alpha_i} |_{\alpha'_u} = 0$;
 - 10: Case Latency-Balance:
 - 11: If $v = F(u)$ then
 - 12: If $u=D(v)$ then
 - 13: $\alpha_u = \alpha'_u$, where $\frac{\partial f(\alpha_i)}{\partial \alpha_i} |_{\alpha'_u} = 0$
 - 14: else if $Ts(u)+S_{u,v} < Ts(v)$ then
 - 15: $\alpha_u = \alpha_u - \Delta$
 - 16: else $\alpha_u = \alpha_u + \Delta$;
 - 17: If $(u=F(v))$ then
 - 18: Update $Ts(u)$ and $D(u)$;
-

5. Simulation Results

5.1. Algorithm Convergence

We simulated the algorithm with a tree network consists of 10 sensor nodes(see Figure 1). All links in T were assigned with the same PHDR. Therefore, the attempt probability for each node is independent on PHDR. Table 1 gives the attempt probability computed by the algorithm for each node, and Table 2 shows $Ts(i)$ for each non-leaf node under different PHDR.

In our algorithm, each node v_i uses a fixed attempt probability $\bar{\alpha}_i$ when computing its attempt probability. This is because the attempt probability computed by the algorithm only considers the performance of the links that are useful for data gathering. However, during the attempt probability computation period, each node needs to broadcast the $\langle PNTab \rangle$ to its neighbors, and all the links are expected to have fair throughput to fasten the convergence speed of the algorithm. In [10], we considered the attempt probability computation problem for one hop networks on the assumption that each node has the same attempt probability.

Table 1. Attempt probability for each node

v_i	v_1	v_2	v_3	v_4	v_5	v_6	$v_{7\sim 10}$
α_i	0	0.058	0.26	0.28	0.167	0.389	0.5

Table 2. $Ts(i)$ with different PHDR

	$Ts(6)$	$Ts(5)$	$Ts(4)$	$Ts(3)$	$Ts(1)$
$\tau=0.85$	17	17	27	37	44
$\tau=0.90$	20	20	33	45	53
$\tau=0.95$	26	26	42	58	69
$\tau=0.98$	34	34	55	76	90

We proved that the $S_{j,i}$ is minimized only when $\alpha = \frac{1}{n+1}$, where n is the number of neighbors of node v_i . Therefore, in our simulation, we simply set $\bar{\alpha}_i$ with $\frac{1}{|N_i(i)|+1}$.

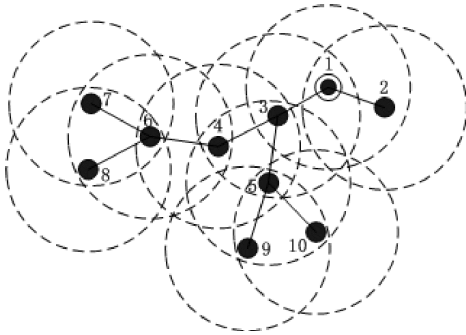


Figure 1. Data gathering tree with 10 nodes

Figure 2 shows the convergence speed of the Self-stabilization phase when each link in T is guaranteed with 90% PHDR. Due to the reason of random access, it is difficult to give a theoretical upper bound on the number of iterations. Figure 2 gives three runs, and it clearly indicates that the Self-stabilization phase has a fast convergence speed. Figure 3 shows the convergence speed of the Balance phase with different Δ . When $\Delta = 0.001$, the data gathering latency decreases stably but the convergence speed is slow. While $\Delta = 0.01$, the convergence speed is a little faster but there are many surges in the convergence curve.

From Figure 2 and Figure 3, it can be observed that the data gathering latency only decreases about 10 time slots by latency balance phase. However, the time spent by latency balance phase is several times of the Self-stabilization phase. Therefore, in real applications, the time allocated to balance latency can be reduced. We also did experiments on several tree networks in which all the leaf nodes have the same hops from the BS. Interestingly, we found that each leaf node nearly has the same data delivery latency after self-stabilization phase. To save energy, the balance phase for such networks can be ignored.

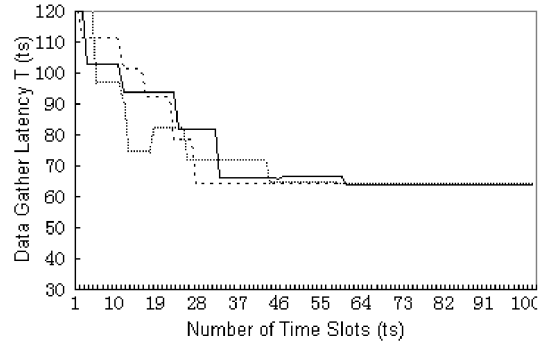


Figure 2. Convergence of Self-stabilization

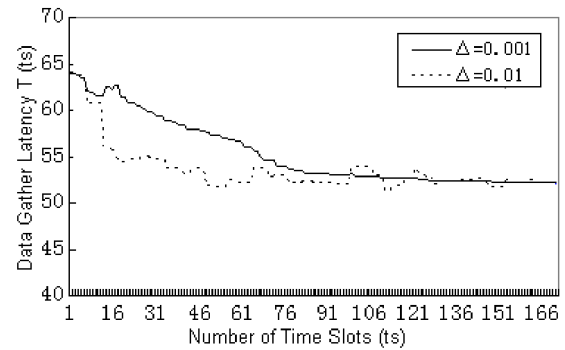


Figure 3. Convergence of Latency balance

5.2. Performance of the model

We first evaluate the data gathering reliability of our model by tuning PHDR. Since our model is based on random channel access model, we use average packet delivery ratio (APDR) as the evaluation metric for data gathering reliability. APDR is defined as the average percentage of received packets (n_{succ}) compared with the total number of nodes n , and it can be computed by the following equation.

$$APDR = \frac{1}{N} \sum_{i=1}^N \frac{n_{succ}}{n} \quad (11)$$

Where N is the number of runs. In this simulation, we simply used the same PHDR for each link, and we compare the results with the theoretical average delivery reliability (TADR) computed by Eq. 12.

$$TADR = \frac{1}{n} \sum_{i=1}^n \gamma^{h(i)} \quad (12)$$

where γ is the value for PHDR and $h(i)$ is the number of hops from node v_i to the BS. We perform the simulation on a data gathering tree composed of 88 sensors, which is constructed by the algorithm proposed in [10]. The results were

obtained by 5000 runs with the same configuration. Figure 4 gives the comparison of APDR and TADR by increasing PHDR step by step. It is clear that APDR is higher than TADR especially when PHDR is small. When PHDR=0.6, the APDR has increased more than 40%. The reason for this case is obvious. Because each immediate node can fuse its data in the packet it receives, the probability that an immediate node's data being delivered to sink by packet issued by its descendants becomes higher with the increase of PHDR. Thus, the data gathering reliability will be greatly improved.

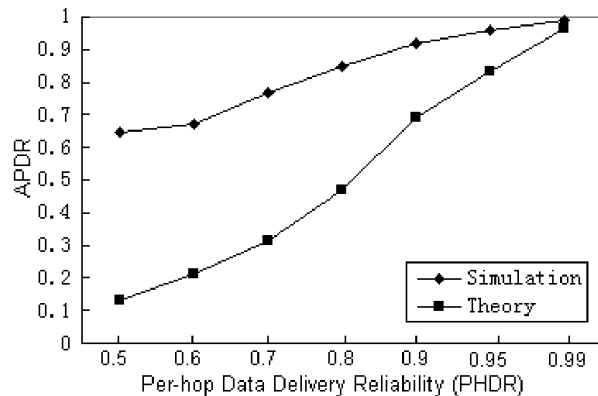


Figure 4. APDR with different PHDR.

Figure 5 depicts the average and max latency with different PHDR. Both of them become larger with the increase of PHDR. By analyzing Figure 5 together with Figure 4, the tradeoff between reliability and latency is clear, and different applications can select different tradeoff between them. For example, the improvement of APDR is only 4.7% when we change PHDR from 0.95 to 0.99, but the average latency and max latency have been prolonged by 63 and 116 time slots respectively. For some applications that require both high packet delivery reliability and low latency, fixing PHDR with 0.95 may be more suitable.

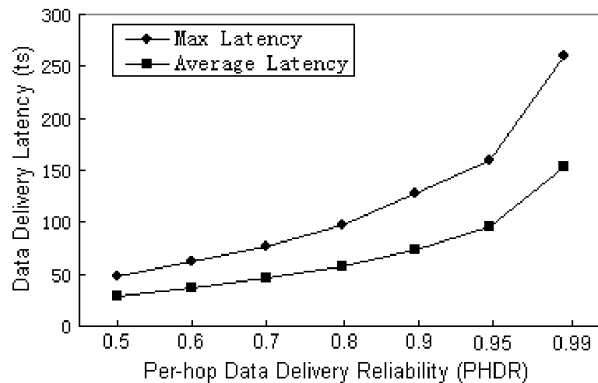


Figure 5. L with different PHDR.

6. Conclusion

In this paper, we study the problem of tuning attempt probability for data gathering in random access wireless sensor networks. We introduce a model for data gathering which considers both packet delivery reliability and latency. In our model, each node has its own attempt probability to counter the local inhomogeneities in node placement. We formulate the attempt probability computation problem as an optimization problem and propose a distributed heuristic algorithm to solve it for tree networks. Simulation results show that our algorithm is efficient and has low computational complexity and our model can provide a good tradeoff between reliability and latency for data gathering.

References

- [1] A. Boukerche, R. W. N. Pazzi, and R. B. de Araujo. A fast and reliable protocol for wireless sensor networks in critical conditions monitoring applications. In *Proceedings of the 7th International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems(MSWiM 2004)*, pages 157–164, 2004.
- [2] P. K. C. Divyasheel Sharma, Vladimir Zadorozhny. Data delivery in sensor networks using whirlpool. In *Second International VLDB Workshop on Data Management for Sensor Networks(DMSN)*, 2005.
- [3] E. Altman, V.S. Borkar, and A.A. Kherani. Optimal random access in networks with two-way traffic. In *Proc. IEEE PIMRC*, 2004.
- [4] A. Karnik and A. Kumar. Distributed optimal self-organisation in a class of wireless sensor networks. In *Proceedings IEEE INFOCOM 2004*, 2004.
- [5] J. H. Sarker and S. J. Halme. Auto-controlled algorithm for slotted aloha. In *IEE Proceedings - Communications*, pages 53–58, 2003.
- [6] P. K. V. Zadorozhny, PK Chrysanthis. A framework for extending the synergy between mac layer and query optimization in sensor networks. In *Proceedings of the International Workshop on Data Management for Sensor Networks(DMSN)*, 2004.
- [7] X. Wang and K. Kar. Distributed algorithms for max-min fair rate allocation in aloha networks. In *Proceedings of 42nd Annual Allerton Conference on Communications, Control and Computing*, 2004.
- [8] H. Yin and H. Liu. Distributed rate adaptive packet access (drapa) for multicell wireless networks. *IEEE Transactions on Wireless Communications*, vol.3:432–441, 2004.
- [9] H. Zhang, A. Arora, Y. ri Choi, and M. G. Gouda. Reliable bursty convergecast in wireless sensor networks. In *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing(MobiHoc2005)*, pages 266–276, 2005.
- [10] H. Zhang, H. Shen, and H. Kan. Reliability-latency tradeoffs for data gathering in random-access wireless sensor networks. In *Proceedings of 4th International Conference on Grid and Cooperative Computing(GCC2005)*, pages 701 – 712, 2005.