Proceeding of the IEEE
International Conference on Information and Automation
Shenyang, China, June 2012

Adaptive and Reliable Transmission Scheduling with Low-Cost Estimation of Channel States *

Xiaoling Zhang, Wei Liang, Haibin Yu, and Xisheng Feng
Shenyang Institute of Automation, Chinese Academy of Sciences
Shenyang, Liaoning Province, China
{*zhangxiaoling, weiliang, yhb, xsfeng*}*@sia.cn*

***Abstract*— An adaptive and reliable transmission scheduling algorithm for wireless sensor networks based on the low-cost estimation of channel states is proposed to jointly optimize the superframe length and reliability. We establish a hierarchical scheduling framework, which includes a global centralized timeslot scheduling (GCTS) and a local distributed channel scheduling (LDCS). On one hand, GCTS aims to guarantee the global optimality of timeslot allocation, during which a mathematical reliability model is built to avoid the resource waste and to improve the transmission reliability. On the other hand, LDCS allocates channel resource according to actual electromagnetism environment. During LDCS, the channel model is established by the dynamic programming method and takes both probing cost and channel quality into consideration, which alleviates the uncertain and time-varying interference and overcomes the blindness of traditional methods. In contrast with previous works that do not consider link reliability and channel probing cost and often assume two channel states, our scheduling algorithm performs reliably for arbitrary number of channels and arbitrary number of channel states. Extensive simulations under a variety of network environments have been conducted to validate our theoretical claims.**

***Index Terms*— Wireless sensor networks. Transmission scheduling. Multi-channel. Reliability**

I. INTRODUCTION

The need for reduced cabling, which leads to both the possibility of faster setup times for equipment and the possibility of communication in areas too harsh for using cables, and the added mobility have triggered research on the use of wireless communication in industrial systems like production systems including robots, control loops, and other automation applications [1]. Even Wireless Sensor Networks (WSNs) add a lot of benefits to the context of industrial communication, they also suffer from a number of disadvantages, such as high error probability. Lost or delayed data may cause industrial applications to malfunction.

The transmission scheduling in the context of Time Division Multiple Access (TDMA) can achieve robust and collision-free communications[2]. Time is slotted into intervals of equal length in TDMA Medium Access Control (MAC) protocols, which is called timeslots. The duration of one timeslot is equal to the time required to transmit a packet and return an ACKnowledgement (ACK). A collection of timeslots that repeat cyclically are grouped into **superframes**. The radio spectrum is divided into channels with small frequency bands, which can realize parallel transmissions and enhance network throughput. Related works have been studied in [3-6]. Although these TDMA scheduling approaches can find minimum length schedule, minimum energy schedules or fairness-based schedule, they do not account for reliability. However, reliability in WSNs is particularly significant. The reliability expresses the probability of a successful packet delivery from a source node to the destination node [7]. On each wireless link, the Packet Loss Rate (PER) of 10%-30% are common [8], which significantly decrease the end-to-end reliability. An example in [7] shown that assuming 10% PER and 3 transmission attempts on each link, 99% of packets are received over one hop. After 10 hops, the success probability is only 76%. Previous works in [9-11] conclude that hop-by-hop retransmission is very important for achieving end-to-end reliability.

Furthermore, in order to communicate more reliably and efficiently, the knowledge of Channel-State Information (CSI) is required to estimate and model the channels. It is very likely that a network can perform better if more information is available and effectively utilized. A straightforward method to obtain all the channel states is to probe all channels in the network. The challenge in exploiting multiple channels is that each node will likely have only limited information about the instantaneous transmission quality of the individual channels. This problem can be overcome by sending control packets on all channels and informing the quality of channel by the receivers [12,13]. However, this probing process will incur overhead, consume more energy, waste more communication resources, and prevent neighboring sensors from simultaneously utilizing channels. Thus, each probe process is associated with a probing cost. However, channel probing must be done efficiently to realize trade-off between obtaining useful channel information and consuming communication resources (timeslots and channels). Related works have been studied in [14-16]. In [17], the combination of timeslot scheduling and channel scheduling is studies for WirelessHART network. However, the method in [17] is

*This work is partially supported by NSFC Grant #60725312 and #61174026, 863 Program #2011AA040103, and the Special Program for Key Basic Research Founded by MOST #2010CB334705.

978-1-4673-2237-9/12/$31.00 ©2012 IEEE

static and does not take the actual wireless environment into consideration.

This paper proposes a framework including both Global Centralized Timeslot Scheduling (GCTS) and Local Distributed Channel Scheduling (LDCS), which could support reliable communication to an application. The commonality and differences between our work and previous works are highlighted below within the context of our main contributions:

First, we propose a joint strategy of centralized timeslot scheduling and distributed channel scheduling, which is the first attempt to the best of our knowledge. The reliability are considered in the scheduling, while study in [16,17] does not. However, the requirement of reliability is required in many applications due to the brittle wireless medium.

Second, we do not restrict the number of channels to be finite as in [14]. Meanwhile, our method of channel modeling doesn't need much resource to calculate it and is simpler than [15]. We probe channels before transmissions and associate each probe process with a well defined probing cost that is more accurate. The probing cost shall be updated on line.

The remainder of this paper is organized as follows: The network model and interference model are described in Section 2. The adaptive and reliable transmission scheduling with low-cost estimation of channel states for WSNs is specified in Section 3. We present the numerical results in Section 4 and conclude the paper in Section 5.

II. System models and assumptions

We model the network as a directed graph *G= (V, E)*. *V* is the set of network nodes, which includes one centralized control node V_0 called Network Manager (NM), and *N* normal nodes V_i $(i = 1, 2, ...N)$. $E = (e_1, e_2, ...e_M)$ is the set of directed links. We assume that if node *i* can receive data from node *j*, they establish a link (*i*, *j*).

Due to the broadcast nature of wireless links, a link may interfere with other links when they transmit on the same channel. We use an interference model to define which set of links can be active simultaneously without interfering. Because the physical interference model is less restrictive and entails more capacity than the pair-wise interference model, we introduce the physical interference model in this paper [18]. In the physical interference model, successful transmission over a link (*i*, *j*) depends on the Signal-to-Interference-Noise-Ratio (SINR) at node *j*. Given P_{ij} be the transmission power of node *i*, G_{ij}be the channel gain between node *i* and node *j*, and η_j be the thermal noise at node *j*, the SINR at node *j* in the presence of other transmissions is:

$$SINR_{ij} = \frac{G_{ij}P_{ij}}{\eta_j + \sum\limits_{k,m \in V\setminus\{i,j\}} P_{km}G_{kj}}. \quad (1)$$

In (1), $\sum\limits_{k,m\in V\setminus\{i,j\}} P_{km}G_{kj}$ is the accumulated interference with respect to link (*i*, *j*); G_{ij} is calculated by the wide used far-field model $G_{ij} = d_{ij}^{-\alpha}$, where d_{ij} is the Euclidean distance between node *i* and node *j*; $\alpha \in [2, 4]$ is the path loss index. If $SINR_{ij} \geq \gamma_0$ is satisfied, node *j* will successfully receive data from node *i*. γ_0 is a given threshold determined by some QoS requirements such as Bit Error Rate (BER).

We assume that all nodes are synchronized and have same transmission range *R*. Every node can transmit data by using *Q* channels in set $\Omega = \{1, 2, \ldots, Q\}$. If node *i* communicates with node *i*, they must have same channels, i.e., $\Omega_i \cap \Omega_j \neq \emptyset$, where Ω_i denotes the set of usable channels by node i.

III. Adaptive and reliable transmission scheduling with low-cost estimation of channel states

The Adaptive and Reliable transmission Scheduling with low-cost estimation of Channel states (ARSC) is executed by two-steps: GCTS and LDCS.

A. Global Centralized Timeslot Scheduling (GCTS)

We utilize Integer Linear Programming (ILP) to formulate the timeslot scheduling problem. Initially, we develop a set of necessary and sufficient conditions that ensure all links can be scheduled conflict-freely with minimum superframe length; then, we extend the conflict-free timeslot scheduling to account for the reliability.

Firstly, decision variables x_{ijt} and y_t are defined as follows.

$$x_{ijt} = \begin{cases} 1, & if\ timeslot\ t\ is\ assigned\ to\ link\ (i,j) \\ 0, & else \end{cases}$$

$$y_t = \begin{cases} 1, & if\ timeslot\ t\ is\ assigned \\ 0, & else \end{cases}$$

x_{ijt} is used to indicate whether timeslot t has been allocated to link (i, j) and y_t is used to indicate whether timeslot t has been allocated. From the definitions of x_{ijt} and y_t, we can conclude that $x_{ijt} \leq y_t$.

Secondly, the ILP formulation of the conflict-free timeslot scheduling is stated as follows.

ILP 1: conflict-free timeslot scheduling problem

$$minimize \sum_{t=1}^{M} y_t \quad (2)$$

subject to:

$$\sum_{j:(i,j)\in E} x_{ijt} + \sum_{j:(j,k)\in E} x_{jkt} \leq 1, \forall t \in [0, M], \forall j \in V; (3)$$

$$x_{ijt} \leq y_t, \forall t \in [0, M], \forall (i, j) \in E; \quad (4)$$

$$G_{ij}P_{ij} - \gamma_0\eta_j - \gamma_0 \sum\nolimits_{k,m\in V\setminus\{i,j\}} P_{km}G_{kj} \geq \Phi(1 - x_{ijt}),$$
$$\forall t \in [0, M], \forall (i, j) \in E; \quad (5)$$

$$x_{ijt} \in \{0,1\}, \forall t \in [0,M], \forall (i,j) \in E; \tag{6}$$

$$y_t \in \{0,1\}, \forall t \in [0,M], \forall (i,j) \in E. \tag{7}$$

Objective (2) is to minimize the superframe length over *M* timeslots (*M* is the total number of links in the network, which is the maximum number of timeslots for communication). Constraint (3) ensures that two adjacent links that shares a common node must be assigned different timeslots and one node can not transmit and receive simultaneously. Constraint (4) is to represent the timeslot status. Constraint (5) is the transformation of SINR requirement in equation (1). Φ is a big integer number and is chosen to reduce the number of judgments. If link (*i*, *i*) is active in timeslot *t*, (5) is equal to (1); however, if link (*i*, *i*) is inactive in timeslot *t*, (5) shall not be satisfied and the judgment stops. Constraints (6) and (7) are the definitions of x_{ijt} and y_t.

Then, based on **ILP 1**, we will take account to the end-to-end reliability. The end-to-end reliability can be satisfied by MAC retransmissions [7]. However, more retransmissions cause larger delay and energy consumption. Therefore, the number of retransmissions should be optimized. We consider the situation that node *i* returns an ACK after correctly receiving a packet from node *j* in a timeslot, and node *j* will stop transmitting the packet after receiving ACK. We assume that ACK will not be lost. Let r be the end-to-end reliability and r_k be the reliability on kth hop ($k = 1, 2 \ldots, h$). The end-to-end reliability r on a path consisting of h hops can be represented as

$$\prod_{k=1}^{h} r_k = r. \tag{8}$$

Let $A(k)$ and T_k respectively denote the number of transmission attempts and the average number of retransmissions needed on kth hop. Based on $A(k)$, T_k can be calculated as:

$$T_k = 1 + \sum_{l=1}^{A(k)} (l+1)(1-r_l)^l r_l. \tag{9}$$

Supposing the per-hop reliability is equal and per-hop packet loss rate p is constant, that is $r_k = r^{1/h}$ (*k*=1, 2, …, *h*). Therefore, $A(k)$ is also a constant, which can be simply denoted as $\overline{A}$. r_k can be represented as

$$r_k = r^{1/h} = 1 - p^{\overline{A}}, \tag{10}$$

and $\overline{A}$ is $\log_p(1 - r^{1/h})$. Replacing r_l in (9) with (10), equation (9) shall be rewritten as

$$T_k = 1 + (r^{1/h} + \frac{1}{r^{1/h}})[1-(1-r^{1/h})^{\overline{A}}] - \overline{A}(1-r^{1/h})^{\overline{A}+1}. \tag{11}$$

We add a reliability constraint to **ILP 1** to guarantee the end-to-end reliability. The timeslot scheduling considering the reliability is denoted as **ILP 2** and is exactly same as **ILP 1**, except for adding a reliability constraint. **ILP 2** is formulated as follows and the reliability constraint is shown in (12).

ILP 2:

$$minimize \sum_{t=1}^{M} y_t$$

subject to:

$$Constraints\ (3) \sim (7)\ in\ \textbf{ILP 1};$$

$$T_k \le \sum_{t=1}^{M} x_{ijt} \le T_{max}, \forall (i,j) \in E. \tag{12}$$

In (12), T_{max} is the limitation for transmission attempts.

In order to find a simple method to solve **ILP 2**, we introduce a timeslot allocation table *Ar* to obtain a much simpler parameterized formulation denoted as **ILP-2(t, Ar)**. In *Ar*, each entry a_{ij} corresponds to a link and its value records how many timeslots have been allocated to link (*i*, *j*).

ILP-2(t, Ar):

$$minimize \sum_{t=1}^{M} y_t$$

subject to:

$$Constraints\ (3) \sim (5),\ (7)\ in\ \textbf{ILP 1};$$

$$Constraint\ (12)\ in\ \textbf{ILP 2};$$

$$x_{ijt} \begin{cases} = 0, & if\ a_{ij} < T_k; \\ \in \{0,1\}, & else. \end{cases} \tag{13}$$

It is well known that timeslot scheduling is NP-hard and solving its **ILP** may take long time especially for large networks. As pointed out in [4], a state-of-art solver (CPLEX 7.0) could solve the timeslot scheduling with no more than 10 nodes to optimality. Thus, we need to design a polynomial time heuristic algorithm to approach the optimal solution regardless of the network scale. By relaxing the integer variable x_{ijt} to [0, 1], we can obtain the corresponding LP-Relaxation (LPR) formulation. We use the solution of LPR as a guideline to schedule the timeslots. Process is repeated until no more timeslots can be scheduled. We define a new variable Y_{ijt} as

$$Y_{ijt} = \begin{cases} -1, & initial\ value \\ 0, & link(i,j)\ is\ not\ to\ be\ scheduling \\ 1, & link(i,j)\ has\ been\ considered \end{cases}.$$

As an entry, Y_{ijt} is used to fill with a scheduling table *Y*. *Y* is indexed by 3-tuples (*i*, *j*, *t*), in which *i* and *j* are used to denote link $(i,\ j)$ and *t* is used to denote the correspondent timeslot. The relaxed formulation of the timeslot scheduling problem is respresented as **LP(Y)**.

LP (Y):

$$minimize \sum_{t=1}^{M} y_t$$

subject to:

$$Constraints\ (3) \sim (5),\ (7)\ in\ \textbf{ILP 1};$$

$$Constraint\ (12)\ in\ \textbf{ILP 2};$$

$$x_{ijt} \begin{cases} = Y_{ijt},\ if\ Y_{ijt} \geq 0; \\ \in [0,1],\ else. \end{cases} \tag{14}$$

Based on **LP (Y)**, a LP-relaxation Heuristic Algorithm (LPHA) can be executed through five steps.

Step 1: initialization

$$\Gamma = \emptyset, \Gamma_L = E; y_t = 0; for\ t = 1\ to\ M, L_t^{\Gamma} = \emptyset;$$

$$for\ all(i,\ j,\ t) \in \Gamma_L, Y_{ijt} = -1.$$

Step 2: solve **LP(Y)**

If **LP(Y)** is infeasible, output Γ and $\sum_{t=1}^{M} y_t$; stop; endif

Step 3: examining optimality in decreasing order of X_{ijt}

Let (*i*, *j*, *t*) be the first to be selected in the solution, do

$$\Gamma = \Gamma + \{(i,j,t)\}; \Gamma_L = \Gamma_L - \{(i,j,t)\};$$

$$L_t^{\Gamma} = L_t^{\Gamma} + \{(i,j)\}; Y_{ijt} = 1; y_t = 1;$$

Step 4: check reliability

a_{ij}++; if $a_{ij} \geq T_k$, Y_{ijt}=0; endif

Step 5: loop from **Step 2**

The heuristic algorithm LPHA is a polynomial time algorithm. The second step solves the **LP(Y)** with the number of variables and constraints bounded by **O**(M^2). The running time is **O**(M^2) in the third step. And, this heuristic algorithm has **O**(M^2) iterations. Hence, the total running time is bounded by **O**(M^4).

After receiving the dissemination of the timeslot scheduling result from V_0, each node extracts its sub-schedule and records it. In an allocated timeslot, every node should choose one channel for each transmission, which is called channel scheduling. The detailed channel scheduling is described in next subsection.

B. Local Distributed Channel Scheduling (LDCS)

In order to overcome interference and enhance reliability, every node needs to probe channels and obtains the quality information of channels before transmissions. Obviously, the probing process will waste time and will bring more collisions to neighbor nodes using the same channels. In this paper, we take the probing cost into consideration.

We assume that all channel states vary independently and identically from timeslot to timeslot and from node to node. Let Π be the set of time-variant strategies of probing channels. Let π be one of the time-variant strategies in Π, which probes channels in the order of $\pi(1)$, ..., $\pi(\tau-1)$, and then transmits over channel $\pi(\tau)$. Let $c(k)$ denote the probing cost of probing channel $\pi(k)$ and c_π denote the overall probing cost that probes all channels under strategy π. Then, the overall probing cost c_π is calculated by:

$$c_\pi = \sum_{k=1}^{\tau-1} c(k). \tag{15}$$

Because the number of channels is finite in networks, the optimal strategy π^* must exists.

We set $c_\pi = 0$, if $\tau = 1$. Meanwhile, since the longest probing process is done through all channels and chooses one for transmission, we can derive

$$P(1 < \tau \leq |\Omega| + 1) = 1, \tag{16}$$

where $|\Omega|$ is the cardinality of channel set Ω.

We model the probing cost as a function of probing time and probing collisions. The probing time under strategy π is $(\tau-1)T$, where T is a constant that represents the time percentage of one timeslot used for probing one channel. T should satisfy that $T < Duration\ of\ one\ timeslot/|\Omega|$. Denote T_s as the duration of one timeslot, the probing time of one channel should satisfy that $T < T_s/|\Omega|$.

The probing collisions are figured up based on the idea of the load-based model [18]. We assume that the final superframe length calculated in the above subsection is $\widehat{M}$. Node *i* records the number of packets transmitted and received at channel $\pi(k)$, noted as U_{ik} every $\widehat{M}$ timeslots ($k = 1, 2 \ldots, |\Omega|$). Denote S_i as the set of neighbor nodes of node *i*. Then, the total number of packets transmitted and received at channel $\pi(k)$ by node i is

$$U_{total} = U_{ik} + \sum_{j \in S_i} U_{jk}, \tag{17}$$

where U_{total} denote the total number of packets transmitted and received at channel $\pi(k)$ by node i and all its neighbor nodes in S_i.

The average number of packets $\overline{U}_{total}$ transmitted and received at channel $\pi(k)$ by node i in one timeslot is

$$\overline{U}_{total} = (U_{jk} + \sum_{i \in S_j} U_{ik})/\widehat{M}. \tag{18}$$

Furthermore, the collision strengths are different between node *i* and its neighbor nodes due to the different distances. In order to model the collisions more accurately, we introduce $d_{ij}^{-\alpha}$ as a weight value. The meaning of $d_{ij}^{-\alpha}$ is same as that of in Section 2. The average weighted number of packets transmitted and received at channel $\pi(k)$ is further represented as (19).

$$\overline{U}_{total} = (U_{ik} + \sum_{j \in S_i} \frac{U_{jk}}{d_{ij}^{-\alpha}})/\widehat{M} \tag{19}$$

Because longer probing time and stronger collisions will result in larger probing cost, we define that the probing cost for probing channel $\pi(k)$ by node i is proportional to the probing time and collisions. Therefore, the probing cost for probing channel $\pi(k)$ by node i can be calculated as

$$c(k) = \frac{T_s}{|\Omega|} \times \overline{U}_{total} = \frac{T_s \times (U_{jk} + \sum\limits_{i \in S_j} \frac{U_{ik}}{d_{ij}^{-\alpha}})}{|\Omega| \times \widehat{M}}. \quad (20)$$

Before finding the optimal channel scheduling strategy, we define three strategies. (a) **Probing policy**: at each step, the sender S has a set of un-probed channels $C \subseteq \Omega$ that would be Ω at the beginning of the super-frame, and has determined the transmission success rate of the probed channels. It continues probing channels in set C and decides the probing order. (b) **Selection policy**: choosing the best probed channel in set $\Omega - C$ for transmission. (c) **Backup policy**: using an un-probed channel in set C for transmission.

During the channel scheduling, we aim to achieve the following objective:

$$J^* = \max_{\pi \in \Pi} J^{\pi} = \max_{\pi \in \Pi} E[r_{\pi(\tau)} - c_{\pi}]. \quad (21)$$

In (21), J^{π} is the network utility under strategy π, which is defined as the difference between the transmission success rate $r_{\pi(\tau)}$ by using channel $\pi(\tau)$ and the overall probing cost c_{π} under strategy π. J^* is the maximum network utility. $\{r_k\}_{k \in \Omega}$ are independent random variables.

Let r_{max} denote the maximum value of transmission success rate among probed channels in $(\Omega - C)$ and $0 \leq r_{\max} \leq 1$. **Probing (*k*)** denotes the action that the sender probes channel k for $k \in C$; **Selection (r_{max})** denotes the action that the sender utilizes the best channel in $(\Omega - C)$ for transmission; **Backup (*k*)** denotes the action that the sender utilizes the un-probed channel k for transmission, $k \in C$; $\pi(r_{max}, C)$ denotes the action taken by strategy π when the channel state is (r_{max}, C).

We use the dynamic programming [20] to represent the decision process of (21), which is shown in (22).

$$V(r_{\max}, C) = max\{\max_{k \in C}\{E[V(max\{r_{\max}, r_k\}, C-k)] - c(k)\}, r_{\max}, \max_{k \in C} E[r_k]\}. \quad (22)$$

In (22), $c(k)$ is the probing cost of probing channel $\pi(k)$; $C \subseteq \Omega$ is the set of un-probed channels. The first one on the right side of (22) represents the network utility of probing the best channel in C; the second one represents the network utility of using the best channel in $(\Omega - C)$; and the last one represents the expected network utility of using the best channel in C for transmission. Meanwhile, it is easy to proof that $V(\bullet, C)$ and $V(r_{max}, \bullet)$ are non-decreasing functions.

From (22), we know that the channel scheduling can be decided according to the network utilities. If we choose **Probing (*k*)**, it means that the network utility of **Probing (*k*)** is the largest. This situation can be formulated as:

$$\max_{k \in C}\{E[V(max\{r_{\max}, r_k\}, C-k)] - c(k)\} > r_{\max} \&\&$$
$$\max_{k \in C}\{E[V(max\{r_{\max}, r_k\}, C-k)] - c(k)\} > \max_{k \in C} E[r_k]$$

Analyzing (22) and referring to [16], we can deduce the following lemmas.

Lemma 1[16]. Consider any state (r_{max}, C). If $V(r_{max}, C) = r_{max}$, then $V(\widetilde{r}_{\max}, C) = \widetilde{r}_{\max}$ for all $\widetilde{r}_{\max} \geq r_{\max}$.

Lemma 2[16]. If $V(r_{max}, C) = E[r_k]$ for some $k \in C$, then $V(\widetilde{r}_{\max}, C) = E[r_k]$ for all $\widetilde{r}_{\max} \leq r_{\max}$.

Lemma 1 and **Lemma 2** give a threshold structures with respect to r_{max}. According to **Lemma 1** and **Lemma 2**, we define two bound values a_C and b_C as

$$\begin{cases} a_C = \inf\{r_{\max} : V(r_{\max}, C) = r_{\max}\} \\ b_C = \sup\{r_{\max} : V(r_{\max}, C) = E[r_k], k \in C\} \end{cases} \quad (23)$$

Then, the optimal strategy π^* can be formulated as

$$\pi^*(r_{\max}, C) = \begin{cases} Selection(r_{\max}) & r_{\max} \geq a_C \\ Probing(k) & b_C \leq r_{\max} \leq a_C \\ Backup(k) & r_{\max} \leq b_C \end{cases} \quad (24)$$

Note that, if $r_{max} = b_C$ and $b_C > 0$, $\pi^*(r_{\max}, C) = Backup(k)$; if $r_{max} = b_C$ and $b_C = 0$, $\pi^*(\mathrm{r_{max}}, C) = Probing(k)$.

Next we will analyze the case that C has only one element k and then extend the conclusion to the case that C has more than one element.

If $C = k$, the two bound values in (23) can be written as follows.

$$\begin{cases} a_k = \min\{r_{\max} : r_{\max} \geq E[r_k], c(k) \geq E[r_k - r_{\max} | r_k > r_{\max}] P(r_k > r_{\max})\} \\ b_k = \max\{r_{\max} : r_{\max} \leq E[r_k], c(k) \geq E[r_{\max} - r_k | r_k < r_{\max}] P(r_k < r_{\max})\} \end{cases} \quad (25)$$

Meanwhile, when $C = k$, the network utilities under three strategies are

$$\begin{cases} f(Probing(k)) = \\ -c(k) + r_{\max} P(r_k \leq r_{\max}) + E[r_k I_{\{r_{jk} > r_{\max}\}}] \\ f(Selection(r_{\max})) = r_{\max} \\ f(Backup((k)) = \mathrm{E}[r_k] \end{cases} \quad (26)$$

where function $f(x)$ indicates the network utility when adopting strategy x.

Supposing that r_k is uniformly distributed in [0,1] and $c(k)$ takes the values of 1/18, 1/8, and 1/4, we can obtain the network utilities as (27).

$$\begin{cases} f(Probing((k)) \\ = -c(k) + r_{\max}^2 + \int_0^{r_{\max}} (0 \times r_k) dr_k + \int_{r_{\max}}^1 (1 \times r_k) dr_k \\ = 0.5 r_{\max}^2 + 0.5 - c(k) \\ f(Selection(r_{\max})) = r_{\max} \\ f(Backup(k)) = 0.5 \end{cases} \quad (27)$$

Equation (27) is visualized in Fig.1.

From Fig.1, if $c(k) = 1/8$, $a_k = b_k = E[r_k] = 0.5$, any strategy is optimal; Thus, we can choose **Selection(r_{max})** for simplicity; if the probing cost is greater than 1/8, the

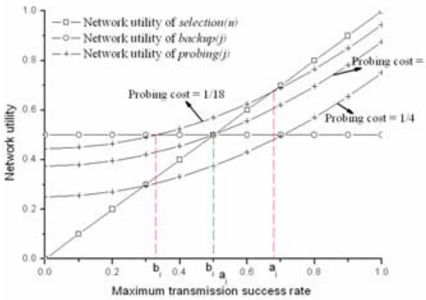


Fig. 1. Network utilities with different probing costs.

strategy of **Probing (**k**)** is never be selected. Otherwise, if $c(k) > 1/8$, $b_k < a_k$ and we can choose strategies according to (24). In conclusion, the probing cost $c(k)$ determines the optimal strategy of channel scheduling.

To the case that C has more than one element, we calculate the values of a_i, b_i ($i \in C$), and probing cost, and choose the optimal strategy according to the conclusion of Figure 1 and (24). A Policy-based Channel Scheduling algorithm (PCS) is proposed to find optimal strategies.

The PCS is executed by the following rules.

Step 1: searching k^* as (28).

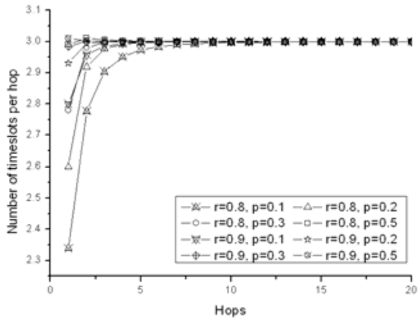
$$R = \{k^* \in C : a_{k^*} = \max_{k \in C} a_k\} \tag{28}$$

Step 2: replacing C with $C - k^*$ and repeating first step, with l denoting the result of this step. The strategy $\pi^*(r_{\max}, C)$ is defined as follows for state (r_{max}, C):

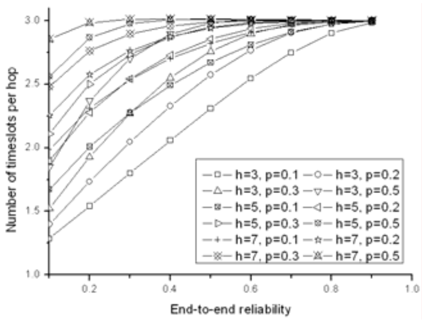
1) If $r_{\max} \geq a_{k^*}$, then $\pi^*(r_{\max}, C)$ = **Selection(**r_{max}**)**.

2) If $a_{k^*} > r_{\max} > \max\{b_{k^*}, b_l\}$, then $\pi^*(r_{\max}, C)$ = **Probing(**k^***)**.

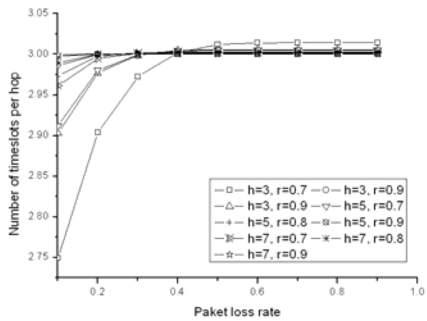
3) If $r_{\max} < \max\{b_{k^*}, b_l\}$ and $b_{k^*} \geq a_l$, $\pi^*(r_{\max}, C)$= **Backup(**k^***)**.



(a)



(b)



(c)

Fig. 2. Relationship among end-to-end reliability, packet loss rate and hops

IV. Performance Evaluation

The performances of the ARSC algorithm are evaluated by simulation and physical experiment. We utilize OPNET10.0 and MATLAB 7.5 to solve the ARSC algorithm and the optimal strategy. Furthermore, the physical experiment platform is established in a factory and the ARSC algorithm is applied in real environment.

A. Reliability Analysis

T_k in (11) is analyzed in cases of different end-to-end reliability *r*, packet loss rate *p*, and path hops *h*. The variation of T_k is shown in Fig.2.

Fig.2(a) depicted the influence on T_k because of the changeable values of *p* and *r* with the increasing *h*. The influence on the number of timeslots per hop is small and the fluctuation range is from 2.33904 to 3.01086. When the hops are bigger than 5, the value of T_k becomes stable. Fig.2(b) depicted the influence on T_k because of the changeable values of *h* and *p* with the increasing *r*. The value of T_k varies more greatly with R than in Fig.2(a). The fluctuation range is from 1.28469 to 3.01299 and the value of T_k converges to 3 finally. Fig.2(c) depicted the influence on T_k because of the changeable value of *h* and *r* with the increasing *p*. The fluctuation of T_k is stable when $p > 0.4$ and the fluctuation range is from 2.749183 to 3.01415.

Fig.2 demonstrates that the reliability model in (11) is robust to the variation of end-to-end reliability, packet loss rate, and path hops. The number of retransmissions could be 2 and 3 timeslots, and 3 in most of situations.

B. Simulation Results

We evaluate the ARSC algorithm from three aspects: total number of timeslots in a superframe, satisfaction degree of reliability, and network throughput. The total number of timeslots in a superframe is the number of timeslots that can guarantee all links to communication without interference. The satisfaction degree of reliability s_r is defined as

$$s_r = \frac{Number\ of\ allocated\ timeslots}{Number\ of\ timeslots\ for\ guaranteing\ reliability}. \quad (29)$$

The network throughput $N(Tr)$ is defined as the average traffic per timeslot, which is represented as

$$N(Tr) = c(\sum_{(i,j)\in E} \sum_{t=1}^{M} x_{ijt})/ \sum_{t=1}^{M} y_t, \quad (30)$$

where c is the electromagnetic propagation velocity, whose value is 3×10^8m/s. Since c is a constant, we can simply use $((\sum_{(i,j)\in E} \sum_{t=1}^{M} x_{ijt})/ \sum_{t=1}^{M} y_t)$ to represent the network throughput.

We utilize MATLAB 7.5 to find the optimal solution for **ILP 2(*t*, *Ar*)** and utilize OPNET 10.0 to solve the heuristic algorithm ARSC: joint LPHA and PCS. The simulation parameters of OPNET 10.0 are listed in Table I. The simulation results are shown in Table II.

TABLE I
SIMULATION PARAMETERS

Parameters	Settings
$Terrain$	$100 \times 100m^2$
$Communication\ radius$	25 m
$Thermal\ noise\ power$	$\eta_j = -190dB$
$SINR\ threshold$	$\gamma_0 = 1dB$
$Maximum\ packet\ length$	128 bytes
$Transmitting\ power$	15.16 mW
$Receiving\ power$	35.28 mW
$Channel$	16 channels IEEE 802.15.4
$Reliability\ requirement$	uniformly distributed in [0.7, 1]

The number of timeslots in a superframe and the network throughput obtained by the optimal strategy and the heuristic algorithm ARSC are shown in Fig.3 and Fig.4, respectively.

The following conclusions can be got by analyzing the simulation results.

TABLE II
OPTIMAL RESULTS AND SIMULATION RESULTS

Network scale	Timeslot number		S_r
	ILP 2(t, A_r)	LPHA	LPHA
1	1	1	1
2	2	2	1
3	6	6	1
4	9	11	0.91
5	33	32	0.83
6	76	84	0.85
10	193	232	0.83

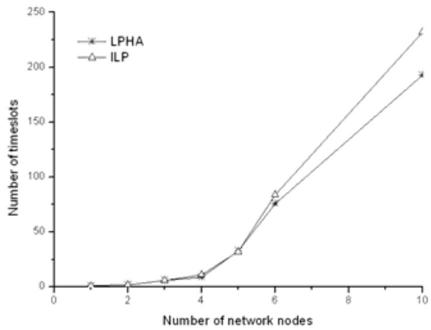


Fig. 3. Timeslot number - optimal strategy and heuristic algorithm.

(1) LPHA is close to optimal strategy when the network scale is smaller than 5 nodes. When the network has 10 nodes, the timeslot number of the heuristic algorithm is 1.2 times than that of the optimal strategy. When the network scale is big than 10, the variable of both algorithms grow rapidly, and even the CPLEX 7.0 can not find an optimal solution in 10 hours.

(2) Because the optimal algorithm can not be found when the number of nodes is bigger than 10 in 10 hours, we can only obtain some values of network throughput when the network scale is small. However, the heuristic algorithm can be performed in a large network. From Figure 5, the network throughput of our heuristic algorithm is more than 85% of the optimal strategy when the number of nodes is smaller than 5. And the network throughput of our heuristic algorithm increases near linearly when the network has more than 5 nodes.

(3) For small network, e.g., 6 nodes, the reliability can be satisfied in most of applications. However, when the network scale is bigger than 10 nodes, the number of timeslots grows rapidly. In order to satisfy the deadline, the reliability is sacrificed and the value of S_r is small. Trade-off between timeliness and reliability should be considered in future work.

In Fig.5, ARSC algorithm is compared with the non-adaptive algorithm. Two interference nodes are placed near the simulated network and randomly transmits data on IEEE STD 802.15.4 channels. In Fig.5, the transmisson success rate of ARSC algorithm is higher because of the adaptive channel

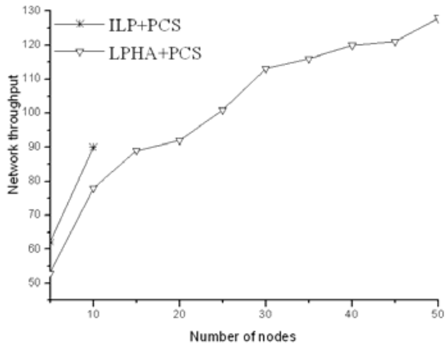


Fig. 4. Network throughput - optimal strategy and heuristic algorithm.

estimation.

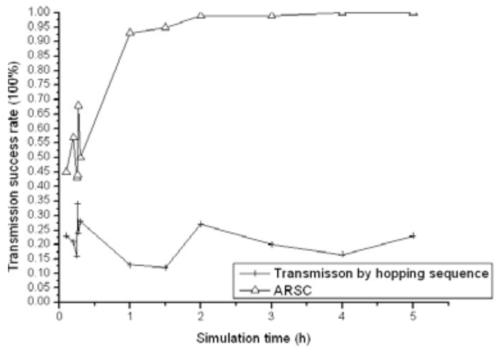


Fig. 5. Comparison between adaptive and non-adaptive algorithms.

V. Conclusion and Future Work

In this paper, we proposed an adaptive and reliable transmission scheduling algorithm with low-cost estimation of channel states for multi-channel WSNs. Our algorithm aims at maximizing the network throughput. In particular, our scheme effectively addressed the problem of end-to-end reliability. We modeled the reliability and the channel probing process more accurately, and proposed a combination scheduling algorithm of global centralized timeslot scheduling and local distributed channel scheduling. Through simulation and experiment, we showed that the proposed algorithm is effective and significantly gets closer to the optimal strategy.

There are still a number of challenging questions for future research. First, we considered the reliability in average retransmission number sense. However, statistic or maximum number of retransmissions should be considered rather than the average number of retransmissions for strict QoS. The second question is how to probing the channel effectively. Previous work overcomes this problem by sending control packets on all channels and informing the quality of channel by the receivers. However, this probing process will incur overhead, consume more energy, waste more communication resources, and prevent neighboring nodes from simultaneously utilizing the channel.

References

[1] W. Andreas, "Recent and emerging topics in wireless industrial communication: a selection", IEEE Transactions on Industrial Informatics, 2008, vol. 4(2), pp.102-124.

[2] P. Djukic and S. Valaee, "Delay aware link scheduling for multi-hop TDMA wireless networks," IEEE/ACM Transactions on networking, 2009, vol. 17, (3), pp. 870-883.

[3] W. Wang, Y. Wang, X. Li, W. Song, et al, "Efficient interference-aware TDMA link scheduling for static wireless networks," IEEE Transactions on Parallel and Distributed System, 2008, vol. 19(12), pp. 1709-1726.

[4] P. Bjorklund, P. Varbrand, and D. Yuan, "A column generation method for spatial TDMA scheduling in ad hoc networks", Ad Hoc Networks, 2004, vol. 2(4), pp. 405-418.

[5] W. Liang, X. Zhang, Y. Xiao, et al, "Survey and experiments of WIA-PA speccification of industrial wireless network", Wireless Communication and Mobile Computing, 2010, vol. 10, pp. 1-16.

[6] X. Zhang, H. Yu, W. Liang, et al, "Joint routing, scheduling, power control for multichannel wireless sensor networks with physical interference", Journal of Control Theory and Applications, 2011, vol. 9(1), pp. 93 - 105.

[7] J. Suhonen, T. Hamalainen, and M. Hannikainen, "Availability and end-to-end reliability in low duty cycle multihop wireless sensor networks", Sensors, 2009, vol. 9, pp. 2088-2116.

[8] A. Cerpa, N. Busek, and D. Estrin, "SCALE: A tool for simple connectivity assessment in lossy environments", CENS Technical Report, Los Angeles, CA, USA, September 2003.

[9] C. Wan, A. Campbell, and L. Krishnamurthy, "Psfq: a reliable transport protocol for wireless sensor network", Proceedings of the 1st ACM international workshop on Wireless Sensor Networks and Applications, Atlanta, Georgia, USA, 2002, pp. 1-11.

[10] F. Stann and J. Heidemann, "RMST: reliable data transport in sensor networks", Proceedings of the First International Workshop on Sensor Protocols and Applications, Anchorage, Alaska, USA, May 2003, pp.102-112.

[11] S. Kim, R. Fonseca, and D. Culler, "Reliable transfer on wireless sensor networks", First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE SECON, Santa Clara, CA, October, 2004, pp. 449-459.

[12] W. Gardner, "Exploitation of spectral redundancy in cyclostationary signals', IEEE Signal Processing Magazine", 1991, vol. 8(2), pp. 14-36.

[13] F. Fitzek, and M. Katz, "Cognitive wireless networks: concepts, methodologies and visions inspiring the age of enlightenment of wireless communications", New York: Springer, 2007.

[14] S. Guha, K. Munagala, and S. Sarkar, "Optimizing transmission rate in wireless channels using adaptive probes", Poster Paper in ACM Sigmetrics/Performance Conference, Saint-Malo, France, June 2006.

[15] A. Alizad, B. Alipanahi, M. Shiva, et al, "Channel estimation for time-hopping pulse position modulation ultra-wideband communication systems", IET Communications, 2008, vol. 2(8), pp. 1051-1060.

[16] B. Nicholas, and L. Mingyan, "Optimal channel probing and transmission scheduling in a multichannel system", Proceedings of the Second Workshop on Information Theory and Applications, Feb. 2007.

[17] H. Zhang, P. Soldati, and M. Johansson, "Optimal link scheduling and channel assignment for convergecast in linear wirelessHART networks", The 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT), 2009, pp. 1-8.

[18] M. AI-Ayyoub and H. Gupta, "Joint routing, channel assignment and scheduling for throughput maximization in general interference models", IEEE Transactions on Mobile Computing, 2010, vol. 9(4), pp. 553-565.

[19] X. Zhang, Q. Liu, S. Dai, et al, "Traffic load-based interference-aware routing protocol for mobile ad hoc networks", Journal of Software, 2009, vol. 20(10), pp. 2721-2728.

[20] P. Kumar and P. Karaiya, "Stochastic system: estimation, identification, and adaptive control", Prentice-Hal, Inc, Englewood Cliffs, NJ, 1986.