EIM: Efficient online Interference Measurement in Wireless Sensor Networks

Xiao Qi*, Yongcai Wang*, Yuexuan Wang*, Liwen Xu*, Changjian Hu†
* Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
{qixiao0113, kyoi.cn}@gmail.com, {wangyuexuan, wangyc}@tsinghua.edu.cn
† NEC Labs. Beijing, China. hu_changjian@nec.cn

***Abstract*—In wireless sensor networks, knowing real signal interferences (received signal strength or RSS) from other sensors to cared sensor is critically important for protocol design and for many applications. However, the real interferences generally differ much from those calculated by theoretical or empirical models, because these models cannot capture the dynamic impacts of the environments. This paper presents EIM, an efficient method for sensors to online measure their interference vectors, where the interference vector of a sensor indicates the real interferences from other sensors to it when other sensors are transmitting packets. EIM conducts interference vector measurement passively during the natural working process of the sensor network. Because a sensor may be interfered by quite a few neighbors and the environments may change over time, the efficiency i.e. latency for real-time interference vector estimation is a very challenging issue. EIM exploits three facts to improve the efficiency of interference vector calculation. First, by exploiting the additivity property of the received signal strength, a linear interference vector reconstruction model is developed. Secondly, EIM exploits not only the received and overheard packets from other sensors, but also the collided packets to construct the observation matrix of the linear model. Third, compressive sensing technologies are developed to recover the interference vector even when the observation matrix is partly determined. These methods help sensors much reduce the latency for calculating real-time interference vectors. Extensive simulations were carried out to verify the effectiveness and efficiency of the proposed methods.**

I. Introduction

In wireless sensor networks, knowing real interferences at sensors is crucially important for protocol design and practical applications[1, 2], such as scene analysis based indoor localization. In the conventional interference models, such as graphic model or physical model [4, 5], the interference from a sensor to another sensor is calculated according to transmission power and the distance among nodes, which cannot capture the field effects in real environments, such as signal blocked by obstacles or multi-path fading effects in indoor environments.

Some recent empirical works [3, 5, 6, 9] studied the impacts of the environments to the signal propagation, such as the multi-path effect, shadowing effect or temperature effect. Because these environment impacts change dynamically and are unpredictable, the signal interferences are hard to predict or calculate. Therefore, online interference measurement in real environments has attracted increasing research attentions[1, 9, 13]. It is crucial not only for improving the accuracy and efficiency of existing interference models, but also important for designing high performance protocols.

But online interference measurement is generally a very challenging problem. First, real-time interference measurement should not interrupt the normal working process of the network. Therefore, the interference measurement should be conducted passively during the natural working process of the network. Second, because a sensor may be interfered by quite a few neighboring sensors and the environments may change over-time, calculating the real-time interferences of these neighbors should have low latency for keeping adaptivity to the environment dynamics.

In this paper, we define *interference vector* for each sensor, which tracks its real-time interferences posed by other sensors. Each item in this vector indicates the disaggregated interference caused by an individual sensor when other sensors are silent. Because a sensor can only passively listen to online messages to infer its interference vector, the efficiency of interference vector calculation is generally a very challenging issue.

In traditional methods, a receiver measures the RSS from a sender only when it receives or overhears packets from the sender. But the measured RSS value may be impacted by other interfering signals from multiple transmitters, which can not reflect the real signal strength of the sender. The worse case is that when the undergoing interference is high, although a receiver can measure an aggregated signal strength, it cannot decode any packet and cannot know who are generating the interference. In such cases, the efficiency of online interference measurement is limited.

To address these challenges, we present following contributions to improve the efficiency of online interference vector measurement.

1) A novel linear interference vector detection model is developed based on the generally assumed additive property of RSS. We conduct hardware experiments to verify the additive property using sensor networks of RF230 radio chips, which validates the proposed model.
2) To resolve the linear model, rank-N observation matrix is required to be constructed for disaggregating interferences from N interfering neighbors. This can be done by exploiting the working logs (transmission and measurement logs) of the sensor nodes, which associates the transmission events of the neighbors to

978-3-901882-54-8/2013

the signal detection events at the cared sensor. However, this step is generally time consuming in the normal working process of the network. EIM proposes to use not only the received and overheard messages from other sensors, but also the collided messages that cannot be decoded to construct the observation matrix. This scheme improves the efficiency of observation matrix construction effectively.

3) To solve the linear equations model, two recovery schemes are developed. When the observation matrix has rank N, least square estimation (LSQ) is exploited to calculate the interference vector. In case the observation matrix is undetermined, compressive sensing method [23, 24, 26] is proposed to treat the problem as a sparse signal recovery problem to calculate the major interferences. The effectiveness of both schemes are validated by extensive simulations, which improves the online interference measurement efficiency by 20% than existing methods at least.

The rest of the paper is organized as follows. Section II introduces the problem model. Section III presents the hardware experiments of the interference additive property. Section IV and V show the methodology of constructing observation matrix and two schemes for recovering interference vector. Section VI presents the simulation results. SectionVII and VIII review the related works and conclude the paper with future works.

II. Problem Model

A. Network model

We consider a sensor network of $N+1$ sensors, which are deployed in a sensing field to perform some specific tasks. The sensors are labelled from 0 to N, which are assumed time synchronized. Let's consider time is divided into discrete time slots with same length. At the beginning of a time slot, a sensor can choose to transmit a packet following some MAC and routing layer protocols. Each sensor is equipped with a RF transceiver, which can detect received signal strength when the sensor receives packets or detects undergoing interferences.

We assume sensors will not change their positions or their transmission powers frequently and consider the environments change slowly. In such conditions, the signal interference conditions change slowly. Interference measurement under such assumptions can capture environment effects from temperature, indoor furniture or outdoor slowly moving objects etc. The highly dynamic effects, such as impacts by quickly moving people are hardly to be captured even using online interference measurement.

Since all the nodes can be treated equally in the interference measurement, we take node 0 as the cared node to consider its interference vector measurement. It is called "host node", and the other nodes are called "guest nodes", which may generate interferences to it.

Definition 1 (interference vector). *The interference vector of the host node at a time t is donated by $\mathbb{S}_t = \{s_i | 1 \leq i \leq N\}$ where s_i indicates the interference value generated by the ith guest node when the guest node transmits packet alone.*

Definition 2 (measurement vector). *The host node online tracks all the signal detection events. Such events can be packet receiving, overhearing or collision at the host node. The events from $t - \Delta$ to t compose a length L event vector $E = \{e_i\}$. Each event is a triple of $\langle \mathrm{Eventtime}, \mathrm{PacketID}, \mathrm{RSSvalue} \rangle$. $\mathrm{PacketID} = \mathrm{null}$ if the packet is not successfully decoded. Let y_i denote the RSS value in event e_i. Let $\mathbb{Y}_t = \{y_j | 1 \leq j \leq L\}$ denote the RSS value vector of all events. Δ is the period of interference vector updating.*

Corresponding to each measurement event, there must be some sensors transmitted packets at $\mathrm{Eventtime}$, which cause the signal detection event at the host node. We define a matrix to indicate the transmission states of the guest nodes corresponding to the event vector of the host node.

Definition 3 (transmission log matrix or TLM). *Let A_t $= \{a_{i,j}, 1 \leq i \leq L, 1 \leq j \leq N\}$ be the transmission state matrix of guest nodes corresponding to the L detected events. $a_{i,j} = 1$ if the jth sensor transmits packet at the time of event e_i; $a_{i,j} = 0$, otherwise. Each row of TLM indicates the transmission states of the guest nodes at a measurement event e_i, which is called transmission log vector and is denoted as $\mathbb{V}_i = \{a_{i,j} | 1 \leq j \leq N\}$.*

The host node can construct its TLM via the transmission logs and its own measurement log. The detail of the TLM associating process will be introduced in section IV.

B. Linear Interference Vector Measurement Model

With above definitions, the key problem for real-time interference measurement is how to calculate $\mathbb{S}_t$ based on $\mathbb{Y}_t$ and A_t. If the signal interference has additive property, i.e. if the RSS of the mixed signals is equal to the sum RSS of each signal, we can have following linear model:

$$\sum_{j=1}^{N} a_{i,j} s_j = y_i \tag{1}$$

The vector type presentation is:

$$\begin{pmatrix} a_{11} & \cdots & a_{1N} \\ a_{21} & \cdots & a_{2N} \\ \vdots & \vdots & \vdots \\ a_{L1} & \cdots & a_{LN} \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_L \end{pmatrix} \tag{2}$$

Using this linear model to estimate interference vector, the key problems to be addressed are as following:

1) The validity of the model depends on the additive property of RSS, which needs to be verified.
2) How to efficiently construct transmission log matrix?
3) Efficient method to solve the linear equations to calculate the interference vector.

We address these problems respectively in the following sections.

III. Interference Additivity Experiments

The validity of the proposed model depends on the interference additivity property, which has been verified in many previous experiments based on the CC1000 and CC2420 commodity radio chips[1, 5, 10]. We conduct further experiments on IRIS node equipped with RF230 radio chip, which further confirms the additivity property of the interfering energy.

A. The Metric of the Experiments

RSS value is generally adopted to indicate the signal interference energy, which measures the received signal strength at the signal detection circuits. For IRIS node, the range of RSS value is $0 \sim 28$ in steps of $3dBm$, whose granularity is relatively coarse in presenting the signal strength. In order to measure signal strength in finer-grained method, we use ED (Energy Detection) value as an indicator of the interference energy to replace the RSS, which is calculated by averaging RSS values over eight symbols($128\mu s$) and has 85 energy levels with a resolution of $1dBm$, which is better than RSS.

B. The Design of the Experiments

Experiments were conducted in a controlled indoor environment where surrounding objects are static. 18 IRIS nodes compose an interference measurement testbed. The communication channels of these sensor nodes are set to Channel 26 to avoid the influences from WiFi. The environment noises are negligible, because they are less than $-91dBm$ in this environment. In experiment, the CCA (Clear Channel Assessment) mode of each node is disabled to disable the embedded collision avoidance in IRIS, which makes sure the sensors can transmit packets simultaneously.

In the experiment, the sensor nodes are classified into 1)*coordinator node* , 2)*host node*, and *3) guest node.* The coordinator node is responsible for time synchronization and process control. The host node is to measure the interference values, and the guest nodes transmit packets using the same transmission power to the host node under the control of the coordinator node. The guest nodes have the same distances to the host node.

C. Experiment Results

The experiment begins with each guest node transmitting 100 test packets following the order of their node IDs, meanwhile, the host node records 100 measured ED values for each guest node. The signal strengths of each guest node without interference is stored by the most frequently appeared ED value in these measured values of the guest node. In the following experiments, the number of simultaneously transmitting guest nodes will be doubled in each time. The host node measures and records the measured ED values to indicate the interfering signal strengths. The experiment process is shown in Fig.1[5].

This experiment results are shown in Fig.2. The signal strengths without interferences are shown in Fig.2(a). In the following experiments, the ED value detected by the host node will increase $2 \sim 3$ (i.e.$2 \sim 3dBm$) in most times from

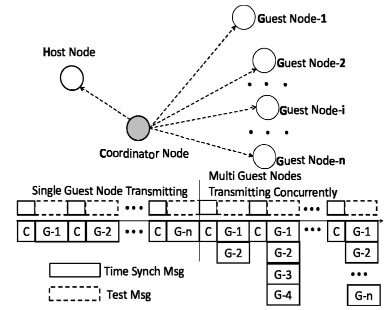
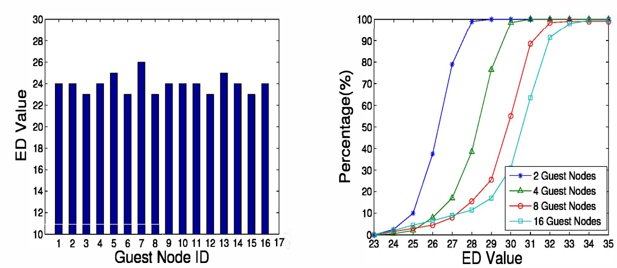


Fig. 1. Interference Additive Experiment in Single-Hop WSN

Fig.2(b), which indicates that the interference value generated by concurrent transmissions is roughly doubled due to the formula (3)

$$P = 10^{(x/10)}/1000 \ and \ x = 10log_{10}(1000P) \qquad (3)$$

where P is the power in W and x is the power ratio in dBm. We also observe that there are a little part of values are less than the expectation in every curve. The reason is due to imprecise time synchronization, because some guest nodes maybe transmit messages a little earlier or later sometimes.



(a) Interference Value of Single Guest Node

(b) CDF of Guest Nodes Interference Values Sum

Fig. 2. The Hardware Experiment Results

The experiments verified the additivity of the interference signal strengths, which helps us to validate the proposed interference vector detection model.

IV. EIM: Efficient Interference vector Measurement

Based on the verification of the observation model, we design and develop the Efficient Interference Measurement (EIM) scheme.

A. Overview of EIM

Both the time latency and the communication cost issues are considered in the design of EIM. The overview of EIM is shown in Fig. 3. In EIM, each sensor calculate its online interference vector distributively, based on the fact that only sensors within a limited range of a cared node can cause interference to the node. Therefore, the host node collects information only from neighborhood within k hops to construct its transmission log matrix (TLM). In such case, the number of interfering nodes, i.e., N in model (2) indicates the number of considered neighbors within k hops.

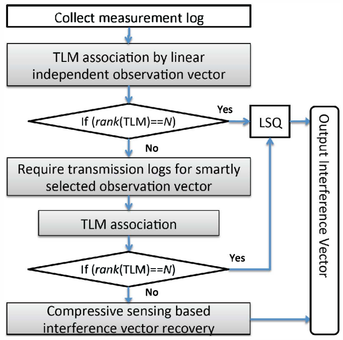


Fig. 3. Overview of EIM

All sensors run EIM with following local information:

1) *transmission log*, which records the time that the sensor transmitted a message;
2) *measurement log*, which stores a vector of $\langle \text{Eventtime}, \text{PacketID}, \text{RSSvalue} \rangle$ triples of the RSS detection events.

The transmission logs of guest nodes are smartly queried by the host node in EIM to efficiently construct the TLM at the host node. The TLM are refreshed after each period Δ for sensors to update the online interference. For a host node to calculate its interference vector at time t, its workflow is as following:

1) The host node firstly checks its measurement log to find the most number of linear independent observation vectors. If these vectors can form a rank N TLM matrix, least square estimation (LSQ) is used to calculate the interference vector (will be introduced soon). But constructing TLM by this method is not efficient, because it needs the host node hears or overhears every neighbor's transmission by at least once.
2) To improve the time efficiency, EIM exploits the signal detection events caused by the collided messages to construct TLM. To reduce the communication cost, we explore the Capture Effect [3] (will be introduced soon) of the linear independent observation vectors, which can determine partly entries of the TLM. This step is designed to avoid requiring unnecessary or redundant transmission logs entries, which reduces the communication cost.
3) The host node construct TLM by associating collected transmission logs with measurement logs. It calculates interference vector by LSQ if $rank(TLM) == N$, and calculates by compressive sensing based recovery algorithm if $rank(TLM)$ is less than N.

B. TLM Association by Independent Observation Vector

We firstly introduce how the host node associates the TLM by independent observation vectors in the measurement logs.

*1) **Capture Effect**:* In our network model, we can consider the "Capture Effect" as followings: if the received power of the transmission is greater than the sum of the interference power of other simultaneous transmissions, then this transmission will be successful. This property can be explained by the $SINR$ model [4, 5].

$$SINR = \frac{P}{I_{sum} + \mathcal{N}} > \beta \tag{4}$$

where P and I_{sum} represent the received power and the sum of the interference powers of other simultaneous transmissions respectively, $\mathcal{N}$ (usually $\leq -91dBm$) is the noise power, and β (usually ≥ 1) is a threshold. We suppose $\beta = 1$, and neglect $\mathcal{N}$ since it is too small to be detected by the sensor node. Obviously, satisfying the $SINR$ model, i.e., the requirement of "Capture Effect" can make the transmission successful.

*2) **Not-collided Observation Event (NOE)**:* Based on the Capture Effect, an measurement event is called NOE, if in this event, the host node successfully receives or overhears the packet from a guest node, so that the host node can successfully decode a dominating sender of this signal detection event. Correspondingly, collided observation events (COE) are defined as measurement events that the host node cannot decode a sender from the collided packets.

For a NOE, suppose it is the lth event in the measurement vector, i.e., e_l. If the dominating sender of this event is decoded as neighbor i, we denote the dominating sender as $D(e_l) = i$, where $D(\cdot)$ is the function to calculate the dominating sender. Two properties can be inferred by the Capture Effect:

- $a_{l,i} = 1$, $a_{l,i} \in \mathbb{V}_l$
- $s_i > s_j, \forall 1 \leq j \leq N$ if $a_{l,j} = 1$

The first fact is obviously which means guest node i must be transmitting at the event e_l, and $\mathbb{V}_l$ is called Independent Observation Vector (IOV). The second fact is because the signal from the dominating sender is the strongest in the concurrent signals.

*3) **Partially TLM filling**:* Based on these properties, we can fill the TLM matrix partially after getting a set of NOEs. Let the NOE set among the measurement events be $E = \{e_l | l \in [1, ..., L]\}$. Suppose the dominating sender of the lth event is $i_l = D(e_l)$. We can get the following information of TLM:

- $a_{l,i_l} = 1$.
- if $e_x \in E$ and $y_x \geq y_l$, then $a_{l,i_x} = 0$, where $i_x = D(e_x)$.

The second point is because the sender i_x has larger interference power than i_l. It must not transmit in event e_l, otherwise signal from i_l will be interfered and cannot be decoded in event e_l. Further more, for an arbitrary COE event e_c, $\forall e_l \in E$, if $y_l > y_c$, we can partially infer the observation vector by:

- $a_{c,i_l} = 0$, where $i_l = D(e_l)$.

It because the senders $\{i_l = D(e_l)\}$ must have larger interference power than $\frac{y_c}{2}$. If $a_{c,i_l} = 0$, then e_c will become NOE, so node i_l must not transmit in event e_c.

*4) **Independent Observation Vectors(IOV)**:* Grounded on these properties, we give a useful result about the $NOEs$.

Result 1. *For a set of NOEs $E = \{e_l | l \in [1, ..., L]\}$, if the dominating sender of each event $\{i_l = D(e_l)\}$ is different from each other, then the observation vectors of these NOEs are linear independent.*

Proof: Without loss of generality, we assume that $|E| = L \leq N$ and make $a_{i,i} = 1$. Other items of each observation vector are unknown, which are represented by squares, then the observation vectors of set E will be

$$\begin{pmatrix} a_{1,1} = 1 & \square & \cdots & \square & \cdots & \square \\ \square & a_{2,2} = 1 & \cdots & \square & \cdots & \square \\ \vdots & \vdots & \vdots & \vdots & \cdots & \square \\ \square & \square & \cdots & a_{L,L} = 1 & \cdots & \square \end{pmatrix}$$

Any two column vectors v_i, v_j will be linear independent, because $a_{i,j}$ and $a_{j,i}$ cannot be equal to 1 simultaneously, because in such case, one event in e_i and e_j will not be NOE, contradicting the assumptions. Since any two column vectors are linear independent, the observation vectors of these NOEs are linear independent. ■

Based on this result and above properties, we can pick up the most number (denoted by n) of linear independent observation vectors. With these n linear independent observation vectors, more than $\frac{n(n+1)}{2}$ entries of the TLM matrix can be determined via the Capture Effects.

C. TLM Association

The unknown entries of the observation vectors need to be filled by the host node requiring transmission logs from its neighbors in the past period Δ.

1) Collect transmission logs based on network protocols: The transmission logs of neighbors can be collected in different ways according to the running protocols of the sensor networks. We restrict sensors to record its transmission logs during run-time working process. How the transmission logs are collected by the host node depends on the running protocols of the network. If the neighbors have routing paths to the host node, the log messages can be piggyback on the normal messages to be transmitted to the host node. Otherwise, the host node can request the logs from the first hop neighbors and the other nodes' logs can be forwarded to the host by the first hop neighbors. We will not specify a log collection process for a specified protocol. Despite of the different implementations of transmission logs collection, a critical problem for the host node is how does it efficiently construct the TLM matrix based on the collected transmission logs.

2) TLM Construction using both NOE and COE: We propose to construct the TLM efficiently using both the NOE and COE events at the host nodes. Since both the transmission logs and measurement events use time as the unified index, when a host node gets the transmission logs from neighbors, it queries the logs to associate each signal detection events with the corresponding transmission events at that time.

Entry $a_{i,j} = 1$ if the jth neighbor was transmitting at the time of the ith signal detection event. Therefore, even for the COE events in which the host node cannot decode a sender, the events can still be efficiently utilized to construct the TLM.

Fig.4 shows a simple example which explains how the host node constructs TLM via associating the transmission logs with the signal detection events. In this example, the host

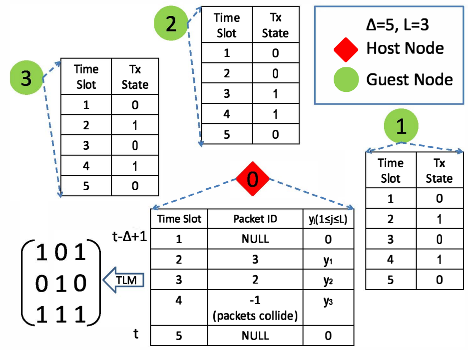


Fig. 4. Construct Transmission Log Matrix via Work Logs

node (node 0) detects three interference events. Two of them are NOE and one is COE. By associating these events with the collected transmission logs from neighbors, The host node construct its TLM. We can see the logs of collided messages are efficiently utilized to construct the TLM.

V. Interference Vector Recovering Schemes

After TLM association, the remaining problem is how to resolve the linear model to calculate the interference vector. Depending on the rank property of the TLM matrix, we provide two schemes to recover the interference vector.

1) If TLM has rank N, we adopt the method of least square estimation (LSQ) to solve the linear model to estimate the real-time interference vector.
2) If TLM is underdetermined, we employ compressive sensing to recover only the major interferences from close by neighbors.

A. LSQ for Solving Linear Equations

A generic method is to directly apply LSQ to calculate the interference vector if TLM has rank N:

$$\mathbb{S}_t = (A_t^T A_t)^{-1} A_t^T \mathbb{Y}_t, \text{if } rank(A_t) == N \tag{5}$$

By LSQ, we can get an interference vector with the minimum square errors, even the measured RSS values have some noises.

However if $rank(A_t)$ is less than N, LSQ can give infinite number of interference vectors $\mathbb{S}_t$ that can satisfy $A_t\mathbb{S}_t = \mathbb{Y}_t$, which cannot determine a unique solution.

B. Recover the Interference Vector via Compressive Sensing

In the practical WSN system, we are generally hard to get a full rank TLM to solve the system of linear equations, it prompts us to find a new scheme to recover the interference vector.

We consider $\mathbb{S}_t$ as a discrete signal vector. If $\mathbb{S}_t$ is sparse, i.e.$\| \mathbb{S}_t \|_0 = K \ll N$, then it is possible to reconstruct $\mathbb{S}_t$ by solving an underdetermined linear observation model using l_1 norm minimization [27] by the recent advantage of compressive sensing[23]. In other words, if $\mathbb{S}_t$ is sparse, we can use a compressive sensing model, i.e., $A_{M \times N}\mathbb{S}_t = \mathbb{Y}_{M \times 1}$, where $M < N$. The matrix $A_{M \times N}$ and $\mathbb{Y}_{M \times 1}$ are the subsets of the TLM A_t and measurement vector $\mathbb{Y}_t$ respectively. Only if $A_{M \times N}$ satisfies Restricted Isometry Property (RIP) and

$M > Klog(N)$, then $\mathbb{S}_t$ can be recovered accurately with high probability[23].

A remained crucial problem is that $\mathbb{S}_t$ is usually non-sparse, whose general pattern can be seen from Fig.5(b). So we find a basis to sparsely represent $\mathbb{S}_t$. We use *Discrete Cosine Transform Basis* (DCT) [25] and *Difference Basis Matrix* (DBM)[26] as the representation basis, and find that the interference vector generally become sparse after the DCT or DBM transformation. The former is generic representation basis in compressive sensing; The latter is usually apply to the case which many entries of the signal vector are same or almost same, the interference vector satisfies this requirement. By a sparse representation, $\mathbb{S}_t$ is further written as $\mathbb{S}_t = \Psi_{N\times N}\mathbb{Z}$, where $\Psi_{N\times N}$ is the DCT or DBM transformation matrix. $\mathbb{Z}$ is the $N\times 1$ coefficient vector in the Ψ-domain with $\parallel \mathbb{Z} \parallel_0 = K$, where $K \ll N$. After such transformation, the measurement model can be rewritten as:

$$A_{M\times N}\Psi_{N\times N}\mathbb{Z} = \mathbb{Y}_{M\times 1} \tag{6}$$

$A_{M\times N}$ and $\mathbb{Y}_{M\times 1}$ are known since both are the subsets of A_t and Y_t respectively. The remaining problem is how to efficiently solve this underdetermined model.

We employ the idea of (Orthogonal Matching Pursuit) OMP[27] algorithm to solve above underdetermined equations to recover the interference vector. The designed algorithm is shown in Algorithm1.

To identify the sparse vector $\mathbb{Z}$, the algorithm determines in a greedy manner which columns of $\mathcal{A}$ $(= A_{M\times N}\Psi)$ contribute most to the measurement vector $\mathbb{Y}_{M\times 1}$. It picks columns of $\mathcal{A}$ in a greedy fashion. At each iteration, it chooses the column of $\mathcal{A}$ that is most strongly correlated with the remaining part of $\mathbb{Y}_{M\times 1}$. Then it subtracts off the contribution to $\mathbb{Y}_{M\times 1}$ and iterates on the residual. The algorithm terminates when a K-sparse vector is determined, which needs only polynomial time computation.

Algorithm 1 Recover the Interference Vector via OMP

Input:

An $M\times N$ matrix $\mathcal{A}$ $(= A_{M\times N}\Psi)$ and An M-dimensional measurement vector $\mathbb{Y}_{M\times 1}$;

The sparsity level K

Output:

The estimate vector $\widehat{\mathbb{Z}}$ for $\mathbb{Z}$;

The set Λ_K containing K elements from $\{1, ..., N\}$;

An M-dimensional approximation $\overline{\mathbb{Y}}_M$ of the measurement vector $\mathbb{Y}_{M\times 1}$

An M-dimensional residual $\overline{\mathbb{R}}_M = \mathbb{Y}_{M\times 1} - \overline{\mathbb{Y}}_M$

1: Initialize the residual $\overline{\mathbb{R}}_0 = \mathbb{Y}_{M\times 1}$, the index set $\Lambda_0 = \emptyset$, and the iteration counter $\gamma = 1$.

2: Find the index λ_γ that solves the easy optimization problem, φ_j is one column of $\mathcal{A}$.

$$\lambda_\gamma = \arg \max_{j=1,\dots,N} |\langle \overline{\mathbb{R}}_{\gamma-1}, \varphi_j \rangle|$$

If the maximum occurs for multiple indices, break the tie deterministically.

3: Augment the index set $\Lambda_\gamma = \Lambda_{\gamma-1} \cup \{\lambda_\gamma\}$ and the matrix of chosen atoms $\mathcal{A}_\gamma = [\mathcal{A}_{\gamma-1}\ \varphi_{\lambda_\gamma}]$. We use the convention that $\mathcal{A}_0$ is an empty matrix.

4: Solve a least-squares problem to obtain a new vector estimate:

$$x_\gamma = \arg\min_x \parallel \mathcal{A}_\gamma x - \mathbb{Y}_{M\times 1} \parallel_2$$

.

5: Calculate the new approximation of the data and the new residual:

$$\overline{\mathbb{Y}}_\gamma = \mathcal{A}_\gamma x_\gamma,\ \overline{\mathbb{R}}_\gamma = \mathbb{Y}_{M\times 1} - \overline{\mathbb{Y}}_\gamma$$

6: Increment γ, and return to step 2 if $\gamma < K$

7: The estimate $\widehat{\mathbb{Z}}$ for $\mathbb{Z}$ has nonzero indices at the components listed in Λ_K. The value of the estimate $\widehat{\mathbb{Z}}$ in component λ_j equals the jth component of x_γ.

VI. SIMULATION RESULTS

With above design, in this section, we will give the simulation results about the performances of the proposed schemes of the interference vector recovery. For LSQ, our focus locates on the number of consumed time slots in the process of constructing TLM. For the compressive sensing method, we pay more attentions to the accuracy of the recovered interference vector.

A. The Configuration of the Simulation

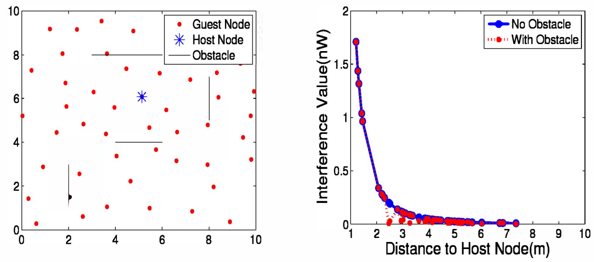
We assume there are 50 sensor nodes be uniformly deployed in a $10m \times 10m$ plane, and some obstacles locate in this area, which influence the interference values generated by some nodes. We can see it is a strong interference environment. We randomly select a node as the host node like Fig.5(a). The signal propagation model used in the simulation is based on the log-normal shadowing path loss model [28]

$$P_r = P_t - PL(d_0) - 10\alpha \log_{10}(\frac{d}{d_0}) - X_\sigma - P_n \tag{7}$$

where P_r is RSS in dBm at a distance d (sender-receiver distance); P_t the output power of the transmitter $(= 0dBm)$; d_0 a reference distance; $PL(d_0)$ is a constant value $(= 55dB)$; α the pass loss exponent $(= 3)$; X_σ a zero-mean Gaussian random variable (in dBm) with standard deviation $\sigma(= 4)$; and P_n the noise power floor $(= -105dBm)$. For convenience, we transform the RSS from dBm to nW following the formula (3). We assume the obstacles are plasterboard or cinder block wall and office window. The signal power will drop down $3 \sim 4dBm$ [29] if it is blocked by an obstacle. The interference values generated by guest nodes to the host node are shown in Fig.5(b). We can observe that some values in red dotted line deviate the blue solid line because of the influence of obstacles.

B. Latency of TLM Construction

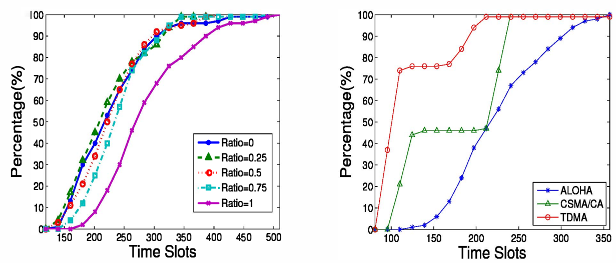
Two groups of simulation are conducted to estimate the latency of full rank TLM construction. Each node in the network is considered as the host node in turn in both groups of simulation. In the first group of simulation, we exploit



(a) The Deployment of the Sensor Nodes (b) The Interference to the Host Node

Fig. 5. The Sensor Deployment and the Interference

ALOHA(Additive Link On-Line Hawaii System) as the MAC protocol, where a node simply transmits the packet with the probability $P = 1/N$ at the beginning of every time slot, and set the $ratio = \frac{\sharp\ of\ IOV}{N}$ in the process of constructing full rank TLM. From Fig.6(a), we can see that only using IOV



(a) The CDF of Latency for Different Ratio (b) The CDF of Latency for Different MAC Protocols

Fig. 6. The Latency for Different Ratio and MAC Protocols

($ratio = 1$) to construct full rank TLM consumes more time slots by 20% than mixed using IOV and non-IOV at least, which means the new scheme utilizing the collide packets improve the efficiency by 20% than the traditional method only using non-collide packets.

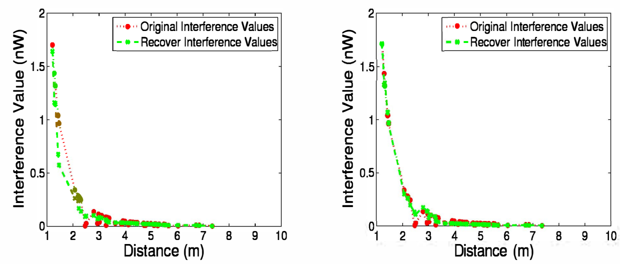
In the second group simulation, the sensor network is assumed to performance some data collecting task, and each node can generate packet with some probability and transmit or forward packets under some routing and MAC protocol. We fix the routing protocol and adopt three MAC protocols (ALOHA, CSMA/CA and TDMA) to schedule the transmissions respectively. For CSMA/CA and TDMA, we pick up the concurrent transmissions in a greedy fashion. Each node performs as the host node to do the interference vector measurement under every MAC protocol, and we do the statistic of the latency of constructing full rank TLM. The simulation results are shown in Fig.6(b), which can be observed that TDMA has the best performance, where almost all the nodes can construct their own full rank TLM in 200 time slots. The reason is that under the TDMA the needed IOVs can appear in less time slots than ALOHA, and the needed non-IOVs can appear in less time slots than CSMA/CA.

C. The Recovery Accuracy via Compressive Sensing

We conduct two simulations by exploiting two representation basis (DCT and DBM) respectively. From Fig.7, we can observe that the interference vector can be recovered by compressive sensing after making this vector sparse via the representation basis, especially the interference value generated by the guest nodes which are closer to the host node can be almost exactly recovered. Further more, we compare the recovery accuracy of both representation basis via calculating the SNR value between the original vector ($\mathbb{S}_t$) and recovered vector ($\widehat{\mathbb{S}_t}$) following formula (8)

$$SNR_{dB} = 10 \log_{10}^{\frac{\|\mathbb{S}_t\|_2^2}{\|\mathbb{S}_t - \widehat{\mathbb{S}_t}\|_2^2}} \tag{8}$$

, and find that DBM performances better than DCT (the SNR value of DBM is usually greater than DCT $6dB$)



(a) Discrete Cosine Transform Basis (b) Difference Basis Matrix

Fig. 7. Recover Interference Vector via Compressive Sensing

VII. RELATED WORK

Previous studies related to the interference measurement can be roughly divided into following three aspects:

1. The relationship between the interference and the link quality estimation and improvement: From the sensor networks perspective, the most important and basic aspect of the communication is the packet delivery performance, i.e., the spatio-temporal characteristics of the packet loss [8]. Some previous studies [7, 12, 14, 15] focused on how to mitigate the interference influence to improve the performance of the packet delivery in the sensor networks. In [14], the authors proposed a measurement-based approach to model the interference and link capacity in 802.11 networks. In [16], the authors proposed an analytical model to estimate the transitional region in the communication range of CC1000 radio, and they found that the root causes of the variable link-level performance were the external noise, random interference, and the transitional region in the PRR-SINR relationship of radio transceivers.

2. The accuracy of interference model and the methodologies of interference measurement: In the work of [9], the authors compared the accuracy of different interference models which vary from oversimplified graphic-based models to fairly realistic SINR physical models via extensive experiments based on the CC2420 radio chip and gived the conclusion that the physical interference model provides the best accuracy. But it is still far from being perfect. Son et al.[5] studied the PRR-SINR model of CC1000 radio and found that the SINR threshold was not a constant value. It depends on the transmitter hardware and the signal strength level. This result, however, is inconsistent with the findings on other radio chip (e.g.,CC2420)[9, 13]. In [1, 11], the authors proposed two approaches respectively to improve the accuracy of the PRR-SINR modeling and evaluated the efficiency via extensive experiments on testbeds. For the interference measurement, there are basically two methods: Offline and Online measurement. In [17–19], Offline measurements were introduced which are to take the network temporarily offline and to inject synthetic traffic for interference measurements. For the online measurement, two methods were given respectively in [20, 21], but both don't utilize the collide packets.

3. The MAC protocol based on the interference models: The design of the MAC protocol under the physical interference model has received significant attention. In [22], the authors showed that the problem of finding a minimum-length collision-free schedule was a NP-complete problem. In some earlier works [13], the authors proposed a new MAC protocol called C-MAC to maximize the aggregate throughput of a wireless node based on the empirical PRR-SINR model.

VIII. Conclusion and Future Works

In this paper, we proposed EIM, an efficient online interference vector measurement scheme based on the interference additive property of the concurrent interferences. We validated the proposed model by the hardware experiments based on the 18 IRIS nodes testbed. EIM utilizes both non-collided observation event (NOE) and collided observation event (CO-E) to construct the observation matrix, which improves the efficiency of linear observation model construction during natural working process of the network. Two schemes for recovering the interference vector have been developed. The first one uses the method of LSQ to solve linear equations, which can recover the interference vector more accurately but needs relatively high latency. The second scheme employs the compressive sensing, which is more scalable for the large scale sensor networks.

The future works may include 1) to refine the physical interference model under the on-line interference measurement; 2) to study on the enhancement of radio-map training in the scene analysis based indoor localization methods, etc.

Acknowledgment

This work was supported by in part by National Natural Science Foundation of China Grant 61202360, 61073174,61033001 61061130540, and the National Basic Research Program of China Grant 2011C-BA00302.

References

[1] S. Liu, G. Xing, H. Zhang and J. Wang, "Passive interference measurement in wireless sensor networks," in *ICNP*, 2010, pp.52-61

[2] P. Gupta and P. Kumar, "The Capacity of Wireless Networks," in *IEEE Transactions on Information Theory*, vol. 46, no. 2, mar 2000.

[3] K. Whitehouse, A. Woo, F. Jiang, J. Polastre and D. Culler, "Exploiting the Capture Effect for Collision Detection and Recovery," in *EmNetS-II*, 2005, pp.45-52.

[4] Q. Chen, "Overhaul of IEEE 802.11 Modeling and Simulation Architecture in NS-2," in *MSWiM'07*, 2007, pp.159-168.

[5] D. Son, B. Krishnamachari and J. Heidemann, "Experimental study of concurrent transmission in wireless sensor networks," in *SenSys'06*, 2006, pp.237-250.

[6] D. Son, J. Heidemann and B. Krishnamachari, "Towards Concurrent Communication in Wireless Networks," in *Technical Report ISI-TR-646,Information Sciences Institute*, 2007.

[7] R. Gummadi, D. Wetherall, B. Greenstein and S. Seshan, "Understanding and Mitigating the Impact of RF Interference on 802.11 Networks," in *SIGCOMM'07*, 2007, pp.385-396.

[8] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *SenSys*, 2003, pp.1-13.

[9] R. Maheshwari, S. Jain and S. R. Das, "A measurement study of interference modeling and scheduling in low-power wireless networks," in *SenSys'08*, 2008, pp.141-154.

[10] R. Maheshwari, S. Jain and S. R. Das, "On Estimating Joint Interference for Concurrent Packet Transmissions in Low Power Wireless Networks," in *WiNTECH'08*, 2008, pp.89-94.

[11] J. Huang, S. Liu, G. Xing, H. Zhang, J. Wang and L. Huang, "Accuracy-aware Interference Modeling and Measurement in Wireless Sensor Networks," in *ICDCS'11*, 2011, pp.172-181.

[12] S. Rangwala, R. Gummadi, R. Govindan and K. Psounis, "Interference-aware fair rate control in wireless sensor networks," in *SIGCOMM'06*, 2006, pp.63-74.

[13] M. Sha, G. Xing, G. Zhou, S. Liu and X. Wang, "C-mac: Model-driven concurrent medium access control for wireless sensor networks," in *InfoCom'09*, 2009, pp.1845-1853.

[14] A. Kashyap, S. Ganguly and S. R. Das, "A Measurement-Based Approach to Modeling Link Capacity in 802.11-based Wireless Networks," in *MobiCom'07*, 2007, pp.242-253.

[15] N. Ahmed, S. S. Kanhere and S. Jha, "Mitigating the effect of interference in wireless sensor networks," in *LCN*, 2010, pp.160-167.

[16] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in low power wireless links," in *SECON'04*, 2004, pp.517-526.

[17] R. Mahajan, M. Rodrig, D. Wetherall and J. Zahorjan, "Analyzing the MAC-level Behavior of Wireless Networks in the Wild," in *SIGCOMM'06*, 2006, pp.75-86.

[18] L. Qiu, Y. Zhang, F. Wang, M. K. Han and R. Mahajan, "A general model of wireless interference," in *MobiCom'07*, 2007, pp.171-182.

[19] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall and J. Zahorjan, "Measurement-based models of delivery and interference in static wireless networks," in *SIGCOMM'06*, 2006, pp.51-62.

[20] K. Cai, M. Blackstock, M. J. Feeley and C. Krasic, "Non-Intrusive, Dynamic Interference Detection for 802.11 Networks," in *IMC*, 2009, pp.377-383.

[21] N. Ahmed, U. Ismail, S. Keshav and K. Papagiannaki, "Online estimation of RF interference," in *CoNext*, 2008.

[22] O. Goussevskaia, Y. A. Oswald and R. Wattenhofer, "Complexity in Geometric SINR," in *MobiHoc'07*, 2007, pp.100-109.

[23] D. L. Donoho, "Compressed Sensing," in *IEEE Transactions on information theory*, vol. 52, no. 4, pp.1289-1323, 2006.

[24] W. Bajwa, J. Haupt, A. Sayeed and R. Nowak, "Compressive Wireless Sensing," in *IPSN'06*, 2006, pp.134-142.

[25] S. A. Khayam, "The Discrete Cosine Transform (DCT): Theory and Application," Michigan State University, 2003.

[26] X. Wu and M. Liu, "In-situ Soil Moisture Sensing:Measurement Scheduling and Estimation using Compressive Sensing," in *IPSN'12*, 2012, pp.1-12.

[27] J. A. Tropp and A. C. Gilbert, "Singal Revovery from Random Measurements via Orthogonal Matching Pursuit," in *IEEE Transactions on Information Theory*, vol. 53, no.12, pp.4655-4666, December 2007.

[28] T. Rappaport, "Wireless Communications: Principles and Practice," Prentice-Hall, Englewood Cliffs, NJ, 1996.

[29] http://www.moonblinkwifi.com/attenuation_at_24ghz_.cfm