Exploiting Ephemeral Link Correlation for Mobile Wireless Networks

Wei Liu[§†]
liuwei@iie.ac.cn

Yunhuai Liu[‡]
yunhuai@trimps.ac.cn

Ziguo Zhong[∓]
zzhong@cse.unl.edu

Limin Sun[† *]
sunlimin@iie.ac.cn

Hongsong Zhu[†]
zhuhongsong@iie.ac.cn

Tian He[◇]
tianhe@cs.umn.edu

[†]Institute of Information Engineering, Chinese Academy of Science, China
[§]Graduate University of Chinese Academy of Sciences, China
[‡]Third Research Institute of Ministry of Public Security, China
[∓]Computer Science and Engineering, University of Nebraska - Lincoln
[◇]Computer Science and Engineering, University of Minnesota

Abstract

In wireless mobile networks, energy can be saved by using dynamic transmission scheduling with pre-knowledge about channel conditions. Such pre-knowledge can be obtained via profiling as proposed by several existing systems which assumed that the existence of spatial link correlation makes the measured channel status at one location reusable over a long period of time. Our empirical data, however, tells a different story: spatial link correlation only maintains well within a short duration (from seconds to tens of seconds) while decreases significantly afterwards, a phenomena we call *ephemeral link correlation*. By leveraging this observation, we design and implement a real-time transmission scheduling system, named *PreSeer*, on the railway platform for cargo transportation, where the transmission of a sink (to cellular towers) can be scheduled intelligently by utilizing *future* channel status measured by sinks located in front of it on the same train. We have implemented and evaluated the PreSeer system on the collected data extensively over 7,000-kilometer railway routes during a period of one and half years. Results reveal that PreSeer can help save as much as 40% energy, comparing with three base-line algorithms. More importantly, lessons learned from this major effort provide useful guidelines for transmission scheduling in highly-dynamic mobile environments, where (i) channel measurements cannot be perfectly aligned due to varying vehicle velocities, and (ii) the accuracy of channel measurements is subject to hardware discrepancy as well as environment irregularity.

*Limin Sun is the corresponding author.

SenSys'12, November 6–9, 2012, Toronto, ON, Canada.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms

Algorithms, Design, Performance, Experimentation

Keywords

Mobile Wireless Networks, Ephemeral Link Correlation, Transmission Scheduling

1 Introduction

This paper presents our experience and lessons learned from a mobile data collection application driven by demanding requirements for safe and secure cotton transportation from the Chinese Ministry of Railways. In China, every year millions of tons of cotton are transported from inner provinces to coastal harbors with hundreds of thousands of freight trains. As shown in Fig. 1, we have instrumented carriages with various types of sensor nodes to collect real-time monitoring data such as temperature and humidity during the transportation. Sink nodes (or sinks for short) equipped with cellular data communication capabilities (e.g., GPRS) are attached to each carriage to forward collected data to remote servers for on-line event detection and analysis.

In such applications, not only normal sensor nodes, but also sink nodes are subject to tight energy constraints because of the high-energy-profile cellular TX/RX operations and the lack of appropriate power supply from the cargo train. Situations get even worse when we consider the fact that wireless links could become more unreliable and irregular in mobile scenarios, especially at a high velocity [8]. To prolong the system lifetime, ideally a sink node should transmit only when it has an excellent wireless channel. However, this simple scheme will have two issues in practice. On one hand, accurate real-time channel prediction in mobile environments is quite challenging, and errors from the prediction can have significant impact on the scheduling decision making. On the other hand, we cannot simply wait for a good channel because data forwarding usually has a stringent delay constraint, i.e.,

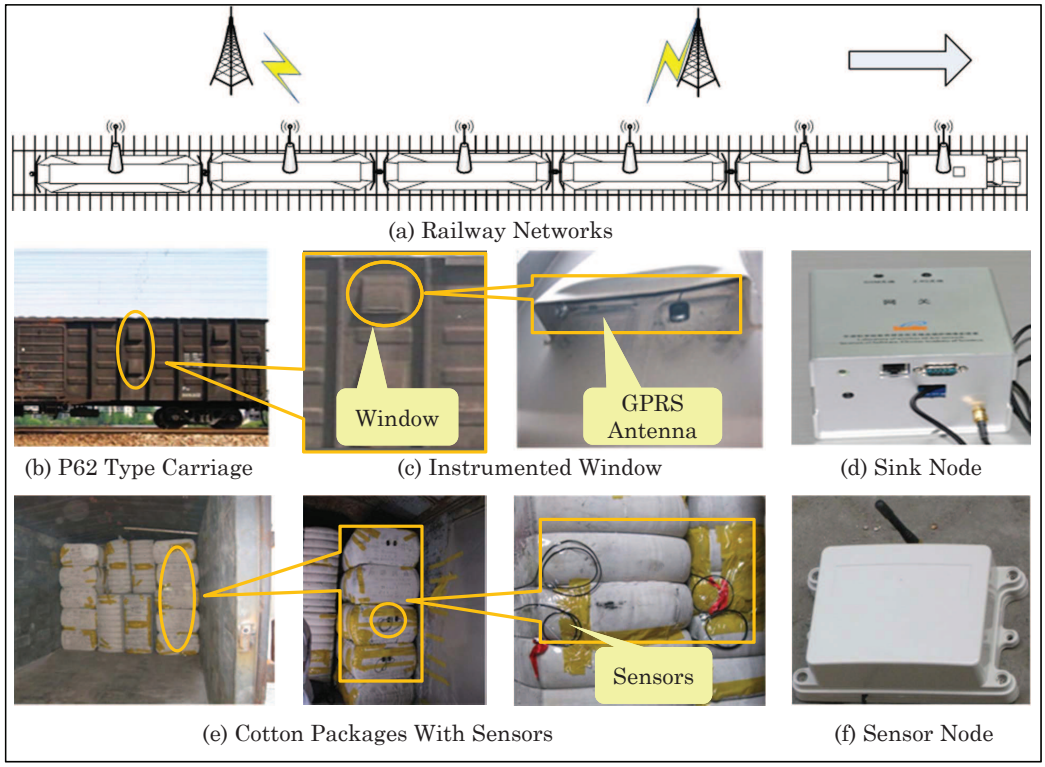

Figure 1. System Overview

a deadline for processing and forwarding, and transmissions after the deadline can be of little value.

To address similar challenges in various systems, researchers have developed many ideas to balance between energy efficiency and networking performance by using pre-knowledge about channel conditions [4, 9]. Recently, an important work Bartender [17] showed that such pre-knowledge can be obtained via RSS profiling. The rational in behind is the existence of spatial link correlation which makes the measured channel status at one location reusable over a long period of time. Our empirical data, however, tells a different story: in high-speed mobile environments, spatial link correlation only maintains well within a short duration (from seconds to tens of seconds), while decreases significantly afterwards, a phenomenon we call *ephemeral link correlation*.

Realizing the limitations of prior work, in this paper we propose *PreSeer*, a real-time transmission scheduling algorithm for mobile wireless networks. PreSeer is motivated by the observation of ephemeral link correlation that can be utilized for efficient networking in systems with well-structured network topology. In such systems, such as a train with sink-embedded carriages, sink nodes pass the same location consecutively during a short period of time. Cellular channel conditions experienced by the sink nodes located in the front part of the train can be exploited by sink nodes behind to forecast their incoming channel conditions in the immediate future, and then their transmissions can be better scheduled to meet the forwarding deadline and the requirement of energy efficiency simultaneously. In short, the intellectual contributions of this paper may include the following.

- To the best of our knowledge, this work is the first to report the phenomena of *ephemeral link correlation* in mobile wireless (cellular) networks.
- In the framework of PreSeer, the real-time link quality prediction in mobile environments becomes possible without the excessive overhead of traditional out-of-band wireless channel profiling. After inspecting experimental results, we found that the prediction errors have a significant impact on the transmission scheduling, a practical concern largely ignored by existing scheduling designs. To address this issue, we design and implement a hybrid scheduling algorithm which utilizes prediction results to guide wireless transmission to cellular towers. This algorithm is robust to different levels of prediction errors.
- We implement the PreSeer design on a railway platform as shown in Fig. 1. In addition to online experiments, more than 400,000 data records were collected from 6 different sink nodes traveling over 7,000 km in different regions of China during the last one and a half years. Results show that PreSeer is practical and effective in reducing energy consumption at sink nodes.

The rest of the paper is organized as follows. Section 2 first gives the background and motivation of this work. Section 3 introduces the main ideas. Then, Section 4 and 5 present the details on wireless signal prediction and transmission schedule optimization. The system implementation and extensive evaluations are reported in Section 6 and 7 respectively. Section 8 briefs the related work, and finally Section 9 concludes the whole paper.

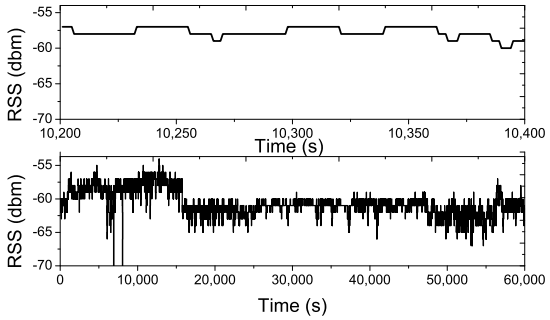

Figure 2. RSS Variation at Identical Location.

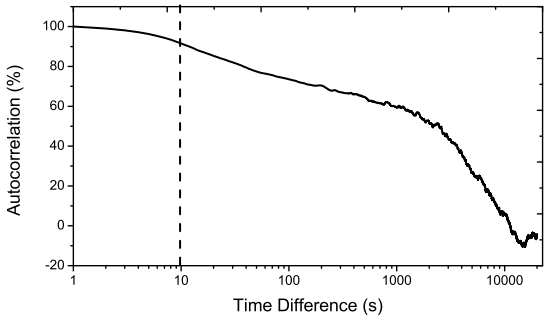

Figure 3. Correlation with Different Intervals

2 Background and Motivation

The uneven distribution of cotton cultivation and consumption leads to massive long-distance transportation using freight trains in China [10]. For years, fire accidents have caused severe damages and casualties during cotton transportation. Since detection during the incipient stage is very difficult, fire is normally detected at the beginning of the growth stage, indicated by temperature, flume, and smoke. Once reaching the growth stage, fire escalates quickly with a logarithmic trend [11]. Therefore, realtime detection and delivery of warning messages are very important in our system.

To prevent and warn the fire hazard in real time, a wireless system is deployed to monitor the status of cotton during transportation. Because the carriages could be re-arranged for different destinations at the transfer stations, each carriage has to be instrumented with (i) a dedicated sink node with GPRS capability and (ii) tens of normal sensor nodes to monitor about 120 cotton packages, as shown in Fig. 1. Sensors collect the readings of temperature, humility, and fume density in the environment and on surfaces of cotton packages periodically, and transmit these readings to their sinks via ZigBee radio at the rate of 1Kbps. Sink nodes are equipped with both GPRS radios for data uploading and ZigBee radios for information exchange between sink nodes mounted on adjacent carriages.

Typically, the duration of a transportation process ranges from 7 days to 30 days, depending on the railway traffic. Sink nodes and normal sensors both are powered only by batteries for three reasons. First of all, sensor nodes need to be inserted into or between cotton packages for effective data gathering, which requires flexibility and simplicity for system deployment. Second, wired nodes with power lines in the carriage are prohibited due to potential fire risks. Third, since carriages would head for different destinations and be disassembled and marshaled during the journey, wiring with power lines increases the operational cost. As a result, not only sensor nodes but also sink nodes are subject to severe energy constraints, especially for sinks with energy-hungry GPRS operations. In cellular networks, the transmission power and data rates are automatically adjusted at mobile terminals based on sensed channel status such as the Received Signal Strength (RSS). To save energy, sink nodes only need to determine the best time to transmit. Therefore the first key problem in our work is an effective prediction for future channel conditions.

2.1 Limitation of Prior Work

To predict the future RSS, profile based approaches have been widely studied in previous research [7, 15, 17]. The basic assumption is that radio signals at the same location are strongly correlated at different time instances. In this case, a previously collected data trace is very useful for predicting channel conditions when driving along the same road again. To verify this assumption, we carried out an experiment in which a sink node was deployed at a fixed location to collect the RSS over 17 hours. We repeated this experiment in different environments. Results are similar which are shown in Fig. 2.

In Fig. 2, the bottom sub-figure shows the RSS variation during 60,000 seconds and the upper sub-figure provides a zoom-in view of the bottom sub-figure from 10,200s to 10,400s. In both figures, the x axis is the time line and the y axis gives the RSS values. From the bottom sub-figure, we can see that RSS values vary dynamically over long time with a maximum peak to peak difference of 15dbm. And from the upper sub-figure, we can see that the RSS looks much more stable and its variation is only 2dbm during 200s.

To show the correlation of RSS at different time, we calculated the autocorrelation of RSS records along the time line with the following equation

$$\theta(\tau) = \frac{E[(X_t - \mu)(X_{t+\tau} - \mu)]}{\delta^2} \quad (1)$$

where τ stands for the time difference, $E[\cdot]$ is the expected value, X_t is the RSS record at time t; μ is the mean of the X, and δ is the variance. Results are shown in Fig. 3 where the x axis is the time in a log scale, and y axis is θ. From this figure, we can see that the autocorrelation is decreased as the time difference increase. When the time difference is smaller than 10s, the autocorrelation is as much as 95%. When the time difference reaches 1,000s, the autocorrelation decreases to 60%. There is little autocorrelation when the time difference reaches 10,000s, which is only about 3 hours.

Based on above results, we argue that the assumption of spatial link correlation only holds partially. On one hand, long-term spatial link correlation at one location is limited due to changes of the environment over time such as humidity and temperature. On the other hand, short-term spatial link correlation does exist and can be very useful for channel prediction. In the following, we first explain the basic idea of PreSeer based on the short-term (ephemeral) link correlation.

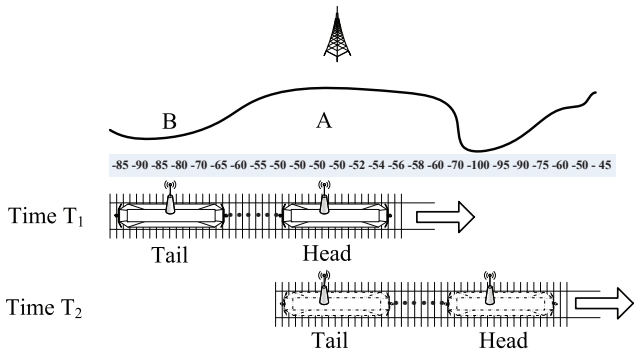

Figure 4. The High-Level Main Idea

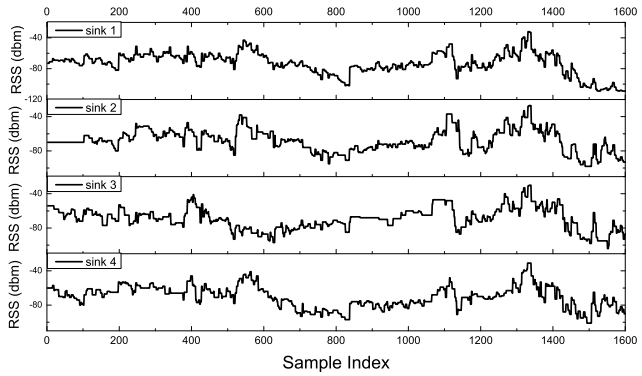

Figure 5. RSS Collected on a Car.

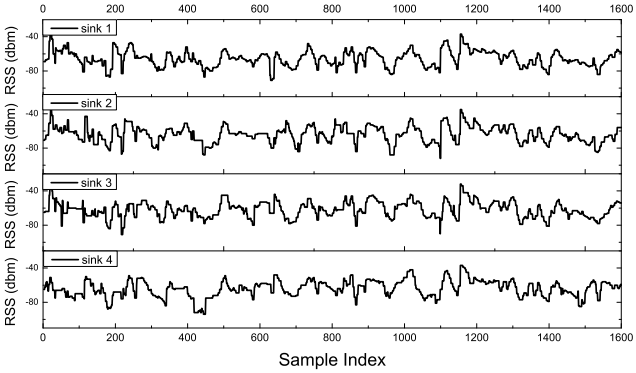

Figure 6. RSS Collected on a Train.

3 Main Idea

By leveraging the characteristics of train, we have the main idea of PreSeer design as follows.

A sink node can decide whether to transmit or not by collecting and analyzing its "future" RSS information provided by other sink nodes located in the front of the same train (along the moving direction).

Fig. 4 depicts an example of this idea, where two sink nodes denoted as *head* and *tail* are deployed on a train. The train travels from left to right. At time T_1, the head and tail are at position A and B respectively. The head sends the RSS measurements between location A and B to the tail. Then the tail can leverage this information to predict its RSS between time T_1 and future time T_2 which is corresponding to the locations B and A respectively. This process iterates as the head updates its RSS to the tail periodically. Note that the energy cost of RSS sharing is negligible because the power of short distance communication such as ZigBee is much lower than that of cellular communication. Furthermore, the RSS sampling as a part of the normal operation of the mobile module (GSM/GPRS module in our system) does not cost additional energy.

Considering the rich literatures on ZigBee-based networking [3, 6], we do not labor on the sharing design among sink nodes and their communication with sensor nodes. Instead, this paper only focuses on energy-efficient transmission scheduling between sinks and cellular base stations using "future" RSS information.

3.1 Observation

A number of studies [7, 17] have demonstrated that in cellular networks, for a specific receiver, the signal strength has a significant correlation with its locations.

The intuition behind is that radio signals suffer from *location-dependent* fading and multi-path effects. However, such location-dependent correlation among multiple receivers is not well studied in the literature. To verify such a correlation, we carried out two sets of experiments by deploying four GPRS receivers with GPS devices. In the first experiment, we setup four receivers in a car. We drive the car along a road segment, and recorded the RSS values from the cellular network. The second experiment has a similar setting except that GPRS receivers are attached to a train. The recorded RSS values in both experiments are shown in Fig. 5 and Fig. 6, respectively, where the x axis is the sample index and the y axis is the RSS. From these two figures, we have an key observation as follows.

Observation: *the RSS is correlated with the sampling location even with different receivers. The RSS varies slowly, but can change considerably after a while.*

3.2 Correlation Evaluation

To quantify the correlation between two signal series, we adopt two evaluation metrics: (i) the Correlation Coefficient (ρ), and (ii) the Average Absolute Relative Deviation (ω), both of which are defined in the following.

Definition Given two series $X = x_i$, $Y = y_i$, $i = 1,...n$, their Correlation Coefficient, denoted as $\rho(X,Y)$, is calculated with

$$\rho(X,Y) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}} \quad (2)$$

where $\bar{x}$ and the $\bar{y}$ are the mean values of two series.

Correlation Coefficient measures how well two series "fit" with each other. A similar trend between two series can be considered as having similar increasing (or decreasing) rates at the same time. The $|\rho|$ is always between 0 and 1. A bigger ρ indicates higher correlation and vice versa.

Definition Average Absolute Relative Deviation (AARD), denoted as ω, is an average of absolute values of relative deviations between two series. For two series $X = x_i$ and $Y = y_i$,

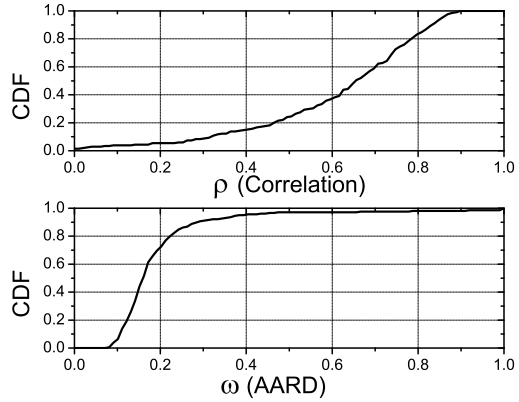

Figure 7. CDF Curves of ρ and ω.

it is calculated with

$$\omega(X,Y) = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - x_i}{y_i}\right| \quad (3)$$

where n is the length of the series.

The AARD ω is often used to evaluate the prediction accuracy, for example, to measure the deviation between the real values (say Y) and the predictions (say X).In general, a smaller ω indicates a higher similarity between two data series.

The correlation coefficient ρ can effectively meter the similarity in *trend* between two series, but without capturing the absolute deviation information. In contrast, ω shows the absolute deviations between two series without trend information. We therefore use both of them to investigate the correlation among series from different sink nodes. To get a quantitative impression, we align the samples in the spatial dimension and pair each sink node with another on the same train to calculate ρ and ω. The CDF for ρ and ω are shown in Fig. 7, which tells that over 60% of sink pairs have $\rho > 0.6$, and over 80% of sink pairs have $\omega < 0.2$. In other words, the ephemeral correlation does exist among sink nodes in mobile environments, if the RSS values are collected within a short interval (e.g., tens of seconds).

3.3 Design Overview

From the above analysis, we can see that there are two steps involved in energy efficient scheduling at the sink nodes: (i) predicting future channel status of a sink node based on RSS information of sink nodes ahead; and (ii) scheduling its transmission in a real-time manner based on the channel prediction. We use Sec. 4 and 5 to present these two steps respectively. Specifically, Sec. 4 explains how to improve prediction accuracy by mitigating impacts of temporal and amplitude offsets in raw samples. In Sec. 5, we developed an error-tolerant scheduling algorithm to decide the right timing for transmission in presence of prediction error.

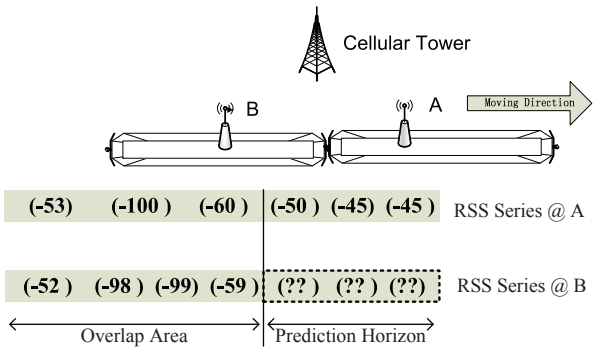

Figure 8. Channel Prediction

4 Channel Prediction

In this section, we will show how to leverage the ephemeral link correlation to predict the future RSS for sinks. We start from the general architecture, and then describe how to predict in the simple cases of two sinks. At last, we extend to more general cases of an arbitrary number of sinks.

4.1 General Architecture and Challenges

For ease of presentation, we start by considering two sinks named as the *current sink* and *ahead sink* and the algorithm description will be with the respect to the current one. In a nutshell, the prediction algorithm works as follows. As shown in Fig. 8, the ahead sink A and current sink B keep collect the RSS values and generate two series of RSS values at same time interval. A portion of the two RSS series are collected at the overlapped area, which are used to train parameters for prediction. Fig. 8 also shows that the ahead RSS series collected by A has additional RSS values, which are exploited by B as inputs to predict the future coming RSS of the current sink node.

During the transportation, sinks are well managed to collect the RSS values at the same time and therefore RSS series are time aligned. We apply a simple time synchronization mechanism that sinks are synchronized before the departure, and are configured to collect the RSS values with a fixed, predefined time interval. By this scheme, we can easily achieve the synchronization [18, 14] on the order of tens of milliseconds which is sufficient for our channel prediction purpose.

When predicting based on the ahead RSS series, we are facing several practical challenges. The first challenge is the lack of location information. In our system, sinks are equipped with the GPS devices but seldom active them. This is mainly because GPS devices are very energy-hungry. When continuously opened, they will consume more energy than the core sensing and communication components and thus are not an energy-economic approach. GPS devices are for emergency purpose only and cannot be used for continuous localization. However, for accurate prediction purpose, we desires RSS series are *spatially* aligned, so that for a given current sink RSS, we can identify the corresponding RSS values obtain by the ahead sink at the *same* location (may be collected a while ago). We call this *RSS series mapping*.

As shown in Fig. 9, though sinks are configured to collect

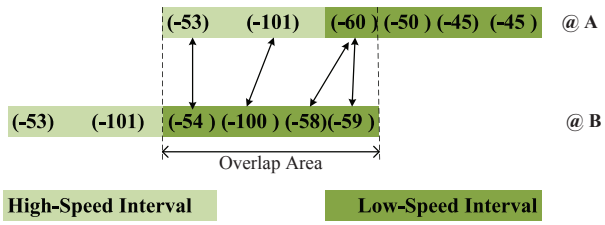

Figure 9. RSS Series Mapping

Algorithm 1 Prediction with RSS sample series

Input: Two RSS series $S_a[1,..,T_a]$ and $S_b[1,..,T_b]$, $T_a > T_b$
Output: Predicted RSS series $S_b[T_b+1,..,T_a]$

1: //Coarse-grained mapping
2: Find all handover in S_a and in S_b
3: Map the handover in S_a and S_b based on the BS info.
4: Divide S_a and S_b to partitions based on handover RSS values
5: //Fine-grained mapping
6: **for** RSS series in each pair of mapped partitions **do**
7: **for** $i \in [1 : T_a]$ **do**
8: **for** $j \in [1 : T_b]$ **do**
9: $\gamma_{i,j} = (S_a[i] - S_b[j])^2 + min\{\gamma_{i-1,j-1}, \gamma_{i-1,j}, \gamma_{i,j-1}\}$
10: $f(i) = \arg\min_j(\gamma_{i,j})$
11: **end for**
12: **end for**
13: **end for**
14: //Parameter estimation and final prediction
15: With $S_b[1 : T_b] = S_a[f(1) : f(T_b)] \cdot \beta + \varepsilon$, apply linear regression
16: $S_b[T_b+1 : T_b+L] = S_a[T_a+1 : T_a+L] \cdot \beta$

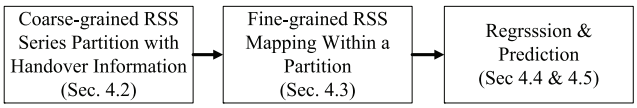

Figure 10. Prediction Procedures

RSS with a fixed time period, a train could change its speeds and travels different distances in the different time slots. A dense RSS series is obtained when a trains runs at a low speed and vice versa. Consequently, it is likely that different sinks measure RSS values at locations with spatial offsets. Moreover, the mapping between RSS series may not be one-to-one but any-to-any as shown in Fig. 9. We will address this issue by (i) using a handover-based localization approach to divide RSS series into matching partitions between cellular towers in Sec.4.2, and within a partition, we use Derivative Dynamic Time Warping (DDTW) algorithm in Sec.4.3 to minimize RSS matching difference.

The second challenge is the hardware discrepancy [20]. Because of the variation during the manufacturing, sinks often own different hardware specifications. In other words, even under the same location in the same radio circumstance, different sinks may obtain different RSS values. As a result, the channel measurements will be deviated, leading to a fixed amount of offsets between RSS amplitudes. We address this issue in Sec. 4.5.

With these challenges, upon receiving the ahead RSS series, the prediction algorithm mainly has three procedures as illustrated in Fig. 10. We first explore the cell handover information to get a coarse-grained mapping between two series and divide them into aligned partitions (segments). Secondly, we find a mapping between RSS values of one or multiple pairs of partitions that are closest to the current time, i.e., the fine-grained mapping. After this mapping, the last step is to establish regression parameters between the two series and then apply these parameters on the ahead RSS series to predict the future RSS values of the current sink. Alg. 1 gives the pseudo-code of the prediction algorithm. We will explain the details of these procedures in the next subsections.

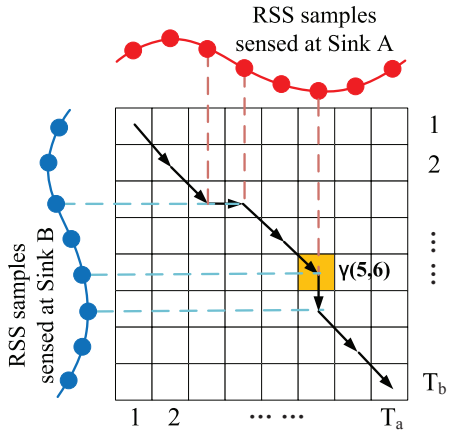

Figure 11. Alignment with DDTW.

4.2 Coarse-Grained RSS Series Mapping

The first task is to roughly map the RSS series, i.e., to obtain a coarse-grained RSS series mapping function. Though the final RSS series mapping function may not be one-to-one, the coarse-grained can be a simpler one-to-one function. For this, roughly speaking the RSS series will be divided to partitions and we will derive an one-to-one mapping function for these partitions. Accurate partition methods are essential as RSS of different partitions will not be mapped in later fine-grained mapping stage.

Towards this goal, an important observation from cellular network is that different sinks may exhibit similar handover behaviors [13]. Because of the ephemeral link correlation and the cellular network standards, sinks often switch from one base station to another at similar locations. As a result, the handover location (i.e., RSS values with a handover operation) can be used as *landmarks* to partition the RSS series.

The algorithm works as follows. Each sink node receives RSS samples with corresponding Base Station ID (BS) from the sink nodes ahead. When the BS ID of the RSS value is changed, it can be considered as a handover. When handing over, the current sink searches the similar handover in the ahead RSS series. Experimental results show that this handover-based RSS mapping can partition the RSS series with reasonable accuracy, while the limitation is that the handover happens infrequently (e.g., once every 10s to 30s). It thus can only provide a coarse-grained mapping and the fine-grained mapping algorithms will be presented in the next subsection. We provide the pseudo-code for the coarse-grained RSS mapping in Alg. 1 from line 1 to line 4.

4.3 Fine-Grained RSS Series Mapping

In this part, we will present how to obtain a fine-grained RSS series mapping with the granularity of each individual RSS measurements. By the coarse-grained mapping, the input of the fine-grained mapping is two RSS series partitions, one is from the ahead RSS series and the other is from the current one. And the two partitions may have different lengths (numbers of RSS values). It is worth noting that by coarse-grained mapping in sec. 4.2, the very beginning and end of the RSS values of the two series partitions are well mapped. Therefore the task of the fine-grained mapping is to map the intermediate RSS values. For ease of presentation, we still

use the term RSS series to represent the RSS series partition.

According to the ephemeral link correlation phenomenon, RSS values of different sinks sampled at the near locations shall have the high correlation and small amplitude difference (called amplitude offset). By this principle, the appropriate mapping shall results in the minimal of the sum of the amplitude offset. The problem can be stated as follows.

Given two RSS series S_a and S_b whose values are $S_a[i], i = 1,..,T_a$ and $S_b[i], i = 1,..,T_b$ respectively, we are seeking a monotonic mapping function $f : Z[1,T_a] \rightarrow Z[1,T_b]$ between S_a and S_b such that

$$minimize: \quad \sum_{i=1}^{T_a} (S_a[i] - S_b[f(i)])^2 \tag{4}$$

$$subject\ to: \quad \forall 1 < i < j < T_a, f(i) \leq f(j) \tag{5}$$

where $Z[1,T_a]$ is the integers from 1 to T_a.

To solve this problem, an intuitive approach is to list all the possible mapping functions and then compute their accumulative distances according to Eq. 4. From all those possibilities, we select the one with the minimal accumulative distance and following the constraint defined in Eq. 5. This simple algorithm, however, has a computation complexity of $O(T_b^{T_a})$ that is costly. To effectively reduce the complexity, we proposed to apply the Derivative Dynamic Time Warping (DDTW) algorithm [12]. DDTW is a dynamic programming method as shown from line 7 to line 12 in Alg. 1, the complexity of which is only $O(T_a \cdot T_b)$.

As illustrated in Fig. 11, the "grid sheet" stands for the accumulative distance matrix Γ corresponding to Eq. 4. Each element $\gamma(i,j)$ in Γ stands for the minimum accumulative distance from $S_a[1:i]$ to $S_b[1:j]$. The basic idea is to reuse the calculated distance $\gamma(i,j)$ to reduce the computational complexity. In Fig. 11, the horizontal red samples are sensed at sink *A*; while the vertical blue ones are sensed at sink *B*. Recall that sink *A* is in front of sink *B* as shown in Fig. 8. A path needs to be found to satisfy the requirement of minimizing the accumulative distance from (1,1) to (T_b, T_a) in the "grid sheet". At each step, the path can be extended one unit to the east (right), or the southeast, or south, which corresponds to the situations of many-to-one, one-to-many and one-to-one mapping respectively. When the entire path is found, the mapping has been established. We denote the mapping relationship as $f(\cdot)$. For example, $f(1) = 1$, $f(2) = 2$, ... $f(T_b) = T_a$. Once the mapping is established, samples are well aligned in the spatial dimension, which is the precondition for estimating the parameters in the linear regression.

4.4 Prediction with Two RSS Series

When the mapping function f between S_a and S_b is obtained by DDTW, we are ready to predict the future S_b values. With newly received $S_a[T_a, \ldots, T_a + L]$, where L is the prediction length, we are able to predict at most a number of L values for S_b. We hereby call L the prediction horizon.

When predicting, we take a simple model that assumes a linear relationship between S_a and S_b. After considering the hardware discrepancy, we can have

$$S_b[1,..,T_b] = S_a[f(1), \ldots, f(T_b)] \cdot \beta + \alpha + \varepsilon \tag{6}$$

where β is the linear coefficient and ε is the error residual.

We apply the linear regression method to compute the parameter α and β with the minimal ε. Due to space constraints, here we omit the details about the linear regression that can be found in reference [2]. Then, the linear Eq. 7 is applied to give the prediction.

$$S_b[T_b + 1, \ldots, T_b + L] = S_a[T_a + 1, \ldots, T_a + L] \cdot \beta + \alpha \tag{7}$$

However, Eq. 7 assumes that two sink nodes have identical velocities when passing the same route segment. In practical systems, this assumption may not hold, which could lead to unexpected temporal offsets with RSS predictions. We mitigate this problem by leveraging results from multiple periodic predictions. The pseudo-code for this part is shown as line 15 and 16 in Alg. 1.

4.5 Prediction with Multiple RSS Series

Indeed, we are not limited to use the RSS series of a single sink to predict that of another sink only. All sinks in front of the current sink can be referred to help prediction. The Multiple Linear Regression is helpful to decide the correlation degree of different sinks with the current one.

More specifically, suppose the current is the K-th sink, and there are $K-1$ ahead sinks. Let the K RSS series be $S_i, i = 1,..,K$ with the mapping function f_i between S_K to S_i where $i = 1,...,K-1$. A multiple linear model between the current S_K and the ahead S_i can be expressed as

$${S_K}^T = \begin{bmatrix} S_1[f_1(1),...,f_1(T_b)]^T \\ S_2[f_2(1),...,f_2(T_b)]^T \\ \cdots \\ S_{K-1}[f_{K-1}(1),...,f_{K-1}(T_b)]^T \end{bmatrix} \beta + \alpha + \varepsilon \tag{8}$$

where ${S_K}^T$ is S_K's transpose. Here, ${S_K}^T$ is a $K \times 1$ vector, $[{S_1}^T, ...{S_{K-1}}^T]$ is a $K \times T_b$ matrix, and β is a $K \times 1$ coefficient matrix. We apply the traditional multiple linear regression method to obtain α, β and ε and again use the model function Eq. 8 to predict the future S_b RSS values. The code is similar to line 15-16 in Alg. 1. More details of multiple linear regression can be found in reference [2].

5 Scheduling Algorithm

In practical environments, scheduling data transmissions based on real-time channel conditions can dramatically save energy. In this section, we present how to well utilize the RSS prediction in transmission scheduling.

In literatures, energy efficient scheduling algorithms can be classified into two categories, off-line scheduling (e.g.,[4, 21]), and on-line scheduling (e.g.,[16, 23]). Off-line methods are designed for ideal environments with the given full knowledge such as the past and future channel conditions and the data arrival rates. They are mainly for theoretical curiosity and can hardly be applied in real environments directly. On-line algorithms exploit statistic information for better scheduling. As we mentioned before, in practice the link correlation is an ephemeral phenomenon and may change significantly in the long term. Statistic information thus may hardly reflect the instant link condition. Be aware of these limitations, in this paper we attempt to combine the intelligence from both worlds by utilizing the prediction results to guide an online scheduling algorithm.

In the remainder of this section, we first introduce the modeling of the scheduling problem, and then give the details of the algorithm design.

5.1 Modeling

For practical concerns we apply a time-slot based scheduling scheme. The sink measures the RSS at 1Hz frequency, i.e., 1 second for one slot. With each measurement, the sink determines whether to transmit until the new RSS measurement. Each scheduling decision (i.e., transmit or not) thus works for one second. We assume that sensing data arrives at a constant rate of r per time slot. The scheduling can be formulated as the following optimization problem

$$minimize \ : E = \sum_{i=1}^{N} e(S_i) \cdot \frac{d_i}{R(S_i)} \tag{9}$$

where E is the cumulative energy cost, N is the total number of the time slots, S_i is the RSS at the i-th slot, $e(S_i)$ is the energy cost if we transmit during a time slot with the RSS value S_i, $R(S_i)$ is the maximum data transmission rate with RSS S_i and d_i is the amount of data to be transmitted at the i-th slot. This objective function is in subject to the following constrains.

First, all the arriving data must be transmitted during the transportation, namely,

$$\sum_{i=1}^{N} d_i = r \cdot N \tag{10}$$

where N is the number of the current time slot.

Second, it must satisfy the causal constraint that the data should not be transmitted before its arrival. We can use a data queue to represent this constraint where the length of the queue at any time must be no less than 0, i.e.,

$$Q_i \geq 0 \quad \text{where} \quad Q_i = Q_{i-1} + r - d_i \tag{11}$$

is the length of the queue at the i-th time slot. The Q_i is determined by three factors, the length of the queue at the $(i-1)$th time slot , the data arriving rate r, and the transmitted data at the ith time d_i .

Because the data arrives one by one, by the Little's Law [5] the mean length of the queue equals to the mean delay. And if the queue is longer than the deadline value, deadline may be missed. To guarantee the satisfied deadline, we formulate the deadline constraint as follows

$$\forall Q_i \leq D \tag{12}$$

The last constraint is that the transmitted data at the ith slot cannot exceed the maximum data rate as shown, i.e. $d_i < R(S_i)$. Integrating all aforementioned objectives and constraints, the scheduling problem can be formulated as

$$minimize: \quad E = \sum_{i=1}^{N} e(S_i) \cdot \frac{d_i}{R(S_i)} \tag{13}$$

$$subject\ to: \quad \begin{array}{c} \sum_{i=1}^{N} d_i = r \cdot N \\ Q_i = Q_{i-1} + r - d_i \\ 0 < Q_i < D \\ d_i < R(S_i) \end{array} \tag{14}$$

Algorithm 2 Scheduling Algorithm

Input: Queue(Q), Deadline(D), Current RSS(S_i)
Input: Predicted RSS($\hat{S}_t, t \in [i, i+min(D,W)]$).
Output: Transmitting or not at current slot i.

```
1: //dynamic programming based on predictd RSS
2: for each packet k in queue do
3:   for each time slot t, t ∈ [i+1, i+min(D,W)] do
4:     if d̂_t + r ≤ R(Ŝ_t) then
5:       Ê_k(t) = min_{j=k-1}^{t-1} {Ê_{k-1}(j) + e(Ŝ_t) · r/R(Ŝ_t)}
6:     end if
7:     Log the index of e with minimum Ê_k(t)
8:   end for
9: end for
10: Caculate d̂_t based on the logged index above;
11: //threshold-based part
12: if Q_i ≥ D then
13:   th = 0
14: else
15:   th = min(Ŝ_t)(∀t ∈ [i+1, i+min(D,W)], d̂_t > 0);
16: end if
17: if S_i > th and Q > 0 then
18:   d_i = min{Q, R(S_i)}
19: else
20:   d_i = 0
21: end if
22: Q_{i+1} = Q_i + r − s_i
```

5.2 Algorithm Description

Predictions always contain errors in practice, which have to be well dealt with during the scheduling to reduce the impact of such errors on the energy consumption. Basically, there are two kinds of RSS prediction errors. One is the amplitude deviation, i.e., the predicted RSS is different from the real RSS. The second is the time offsets of the RSS, e.g., an RSS appears but at a different time slot.

In literatures there are two kinds of methods to solve this problem, namely dynamic programming and threshold based scheduling. Dynamic programming can achieve optimal results when the input is complete and accurate. It determines how much data should be transmitted in each time slot based on the RSS. The main limitation is that dynamic programming is vulnerable to time offset errors. When the good link condition arrives at a different time, data might be misscheduled at a wrong time in which case the channel is poor. On the contrast, threshold based algorithm provides more error-resistance to time offset as it is not based on the time, while the limitation is that in practice, the optimal threshold is environmental-dependent and thus can hardly be obtained. Realizing pros and cons of both methods, we design a *hybrid* algorithm that takes advantages from the both sides.

The basic idea is not complex. Given the predicted RSS, we apply the dynamic method to guide the selection of a threshold. The pseudo-code is shown in Algorithm 2. Suppose the current time is i, and by prediction we get the RSS for the next W time slots (Recall in Sec. 4.4, W is the prediction length). So the input will be the current RSS value S_i, the predicted future RSS value $\hat{S}_t,\ t \in [i+1, i+min(D,W)]$. We also have the queue which may store some data before, and the required deadline is D. The output will be a decision whether to transmit at this i-th slot.

From line 1 to 10, we firstly apply the dynamic program-

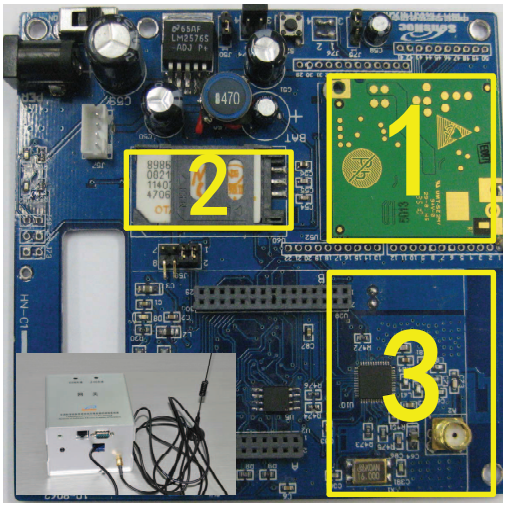

Figure 12. The Sink Node Hardware

ming on the predicted RSS results to obtain a scheduling result. The core line is 5 in which $E_k(t)$ is the cumulative energy cost for transmitting the previous k packets at time slot t; $e(\hat{S}_t)$ is the transmitting energy using the RSS value S_t. The output of this dynamic programming process is $\hat{d}_t$, the amount of data to be transmitted in each of the time slot. We here use $\hat{d}_t$ but not d_t because this is an intermediate result. We then apply the threshold-based scheduling algorithm. From all the future time slots with transmissions (i.e., $\hat{d}_t > 0$) according to the dynamic scheduling results, we look for the one with the weakest predicted RSS and set it as the threshold(line 15). In other words, if the current RSS is greater than this RSS (threshold), we transmit. Otherwise, we wait (lines 17 to 21). Besides, we need to take boundary conditions into account: When the queue length $Q_i > D$, it implies the deadline will be missed, in which case whatever the RSS we transmit by setting the threshold as 0 (line 13). When the queue $Q = 0$, there is no data to transmit. After all these, in line 22 we update the queue Q after the transmission.

The benefits of integrating the two scheduling schemes are two-folds. First, the errors in individual RSS measurements do not impact on the results much as the decisions are made according to a number of RSS values. In other words, the errors in RSS measurements are amortized to a number of RSS samples. Second, the threshold based method is less sensitive to the varying velocity. By applying the threshold, it is equivalent to get an average velocity in the dynamic scheduling space, which is more stable than the instantaneous speed at each time slot.

6 Implementation and Experimentation

This section explains PreSeer hardware design, data collection schemes, experiment setups and runtime performance.

6.1 Hardware Design

In the Chinese railway system, a freight train usually has 30∼70 carriages, and the carriage used to transport the cotton is the P62 type as shown in Fig. 1(b) with a dimension of $15.4 \times 2.8 \times 2.7$ (in meters). Thus, a freight train can be 1,000-meter long. A freight train is operated to travel at speeds of 10∼25 meters per second that is much slower than passenger trains. Because of its low priority, it also needs to make way for passenger trains if necessary. As a result, the transportation time of a freight train can be uncertain.

In our system, a sink node alone with 50∼180 sensor nodes are deployed in each carriage as shown in Fig. 1(f). There are several types of sensors such as temperature sensor, humidity sensor, regain sensor, and smoke sensor. Some of the sensor nodes are inserted into the cotton package to monitor the cotton status, and some of them are placed among cotton packages to monitor the environment. Each sensing record has 6 bytes of payload including sensor ID, sequence number, and data. The incoming data rate at the sink varies from 0.3 KB to 1.08 KB during a sampling period of 10 seconds, depending on the number of sensors in the carriage. The collected data is required to be sent out before a deadline specified by the user, which is 20 seconds in our system.

As shown in Fig. 12, each sink node employs a ZigBee short distance radio module (marked as “3” in Fig. 12) for data collection from the sensor nodes and a Huawei EM310 GPRS module (marked as “1” in Fig. 12) for cellular networking. The number “2” denoted in Fig. 12 shows the SIM card from China Mobile. We also design another version of sink nodes which is equipped with WiFi module.

For energy supply, we used heat-proof lithium batteries with a capacity of 17Ah to reduce potential risks of fire and explosion. All devices were carefully packaged with metal shells that can prevent damages caused by battery fire. Without recharging, the maximum transmission time of a sink node to the cellular network is around 8 hours, which is much shorter than the average cotton transportation duration.

6.2 Experiment Setup

To verify our design in realistic scenarios, we first ran a starter configuration in a train with two sink nodes which are 200 meters apart. With this distance, the prediction length varies from 7 second to 20 seconds, depending on the train speeds. In our setting, a CC2420 chip with ZigBee protocol can transmit 100m in the motionless outdoor environments and 70m in the moving train using the maximum TX power 0dbm. And a RT3070 chip with WiFi protocol can transmit about 180m and 110m respectively when using the TX power 20dbm with the same packet loss rate. To simplify the prototype implementation, we use a WiFi module between the sinks to share the RSS information, which includes sink ID, sequence number, RSS value and Cell ID. A relay node was placed in the middle of the two sinks.

We note that the energy overhead introduced by sharing RSS information is negligible compared with energy consumed in GPRS communication. For CC2420, the Tx and Rx current are 17.4mA and 19.7mA respectively. Accordingly each packet with average length of 50 byes would consume about 10^{-4}J when transmitting. If a WiFi chip is used instead for sharing, the Tx and Rx current are 31mA and 37mA respectively using RT3070. The energy consumption is less than 10^{-4}J. In contrast, the energy cost for an average 500-byte GPRS packet is about 0.1J, which is more than hundred times larger. Clearly, it is worthwhile to spend energy for RSS sharing in exchange of a better GPRS communication.

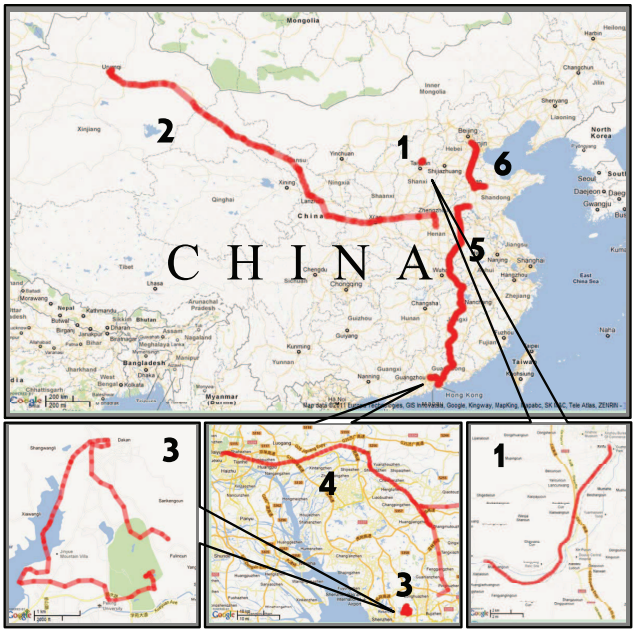

Figure 13. The Experiment Journeys.

Table 1. Experiments

No.	From	To	Distance (km)	Num of Sinks
1	Taiyuan	Xinzhou	88	1
2	Urumqi	Xuchang	3165	1
3	Shenzhen	Shenzhen	14	4
4	Shenzhen	Guangzhou	139×4	6
5	Shenzhen	Jining	1912	5
6	Zibo	Beijing	605	5

In addition to the starter configuration, we also employ six sink nodes during other experiments to collect RSS trace data. To obtain the empirical data in different natural environments with diverse cellular tower densities, we conducted experiments in various regions throughout the country as shown in Fig. 13. More than 400,000 data records along approximately 7,000 kilometers of railway routes were collected from 6 different railway segments as listed in Table 1. In all the experiments, we have no control over the cellular network or the train itself.

6.3 Experiment Results

Because the energy cost of the GPRS transmission is hard to measure accurately and the result is easily affected by other parts of the circuit, we use the RSS distribution when sending packets as the metric to evaluate energy saving performance of our design in real experiments. More packets transmitted under high RSS indicates that more energy has been saved. To show the performance, a Baseline method is used as the comparative in which the packets are transmitted as soon as they arrival. Fig. 14 shows the experimental results. From the figure we can observe that more than 60% packets are sent when RSSs were lower than 50 using Baseline method, while only 30% packets are sent when RSSs were lower than 50 using PreSeer.

To show the effect of our design, we draw the dynamic

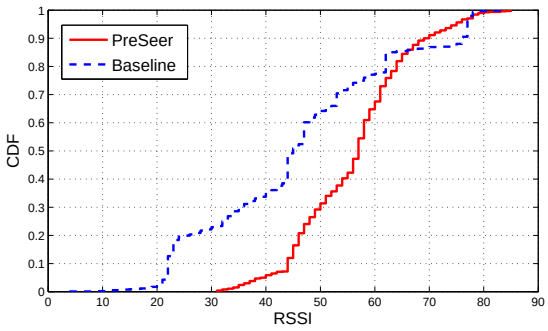

Figure 14. Comparing RSS CDF during Transmission.

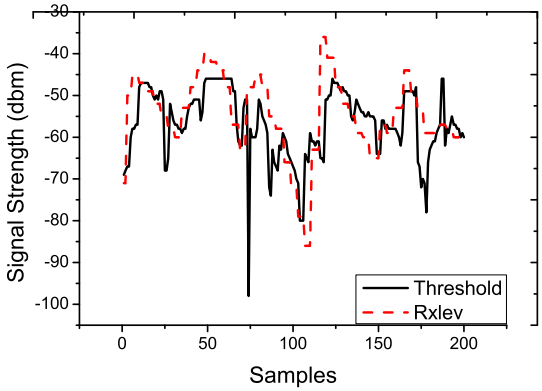

Figure 15. The Varying Thresholds with RSS.

threshold with the RSS variations. The result is shown in Fig. 15. The two curves are the RSS and the threshold respectively. The sink transmits data only when the RSS is bigger than the threshold. From the figure we can see the threshold varied with the RSS and data were always transmitted at signal's peak positions.

7 Trace Driven Evaluation

To compare the PreSeer design with other baseline approaches under exactly the same network environments, we use the same sets of RSS traces collected during the journeys as the inputs for evaluation, effectively avoiding performance discrepancy introduced by different runs.

7.1 Approaches

One experiment goes through thousands of kilometers' railways. We divided the collected about 400,000 records into 87 small data sets with approximately 4,000 records each. Identical data sets are used as inputs for both our design and base-line algorithms for channel prediction and transmission scheduling.

With those data sets, PreSeer is compared with three reference scheduling algorithms. The first one does not apply any transmission optimization, in which the aggregated sensing data sent out as soon as data arrives at the sink node. We call such a method as *Baseline Scheduling* that provides the worst-case upper bound for the energy cost. The second algorithm, on the contrary, is *Ideal Scheduling* in which future channel conditions and data rates from sensor nodes are assumed to be known to the scheduler. With ideal scheduling, which is impossible in practice, we can obtain the best-case lower bound

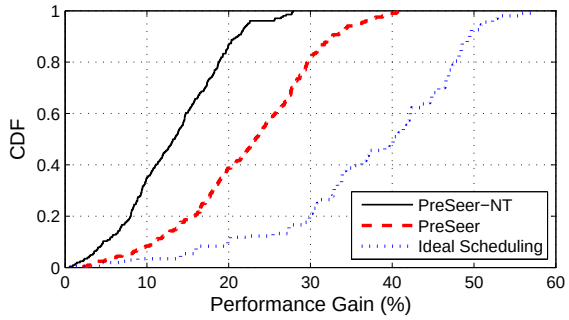

Figure 16. Comparing the Performance Gains.

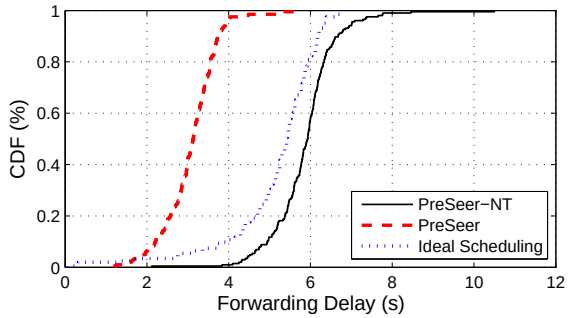

Figure 17. Comparing the Forwarding Delays

for the energy cost. The third method used for comparison is a simplified version of PreSeer itself that is called *PreSeer without error tolerance*, or *PreSeer-NT* in short. The difference between PreSeer-NT and PreSeer is that PreSeer-NT directly uses results from the channel prediction step for transmission scheduling without conducting the threshold-based error processing. It schedules data transmission solely based on the results from dynamic programming discussed in Sec. 5.2.

To evaluate the system energy cost, without loss of generality, we adopted the widely used power model from [1] as Eq. 15.

$$P_{CH} = \min(G - \alpha \times (RSS + 48), PMAX) \qquad (15)$$

where P_{CH} is the transmission power; G and α are system parameters used to adjust the power control performance in different environments. RSS is the received signal strength measured by the mobile terminal, i.e., sink nodes in our case. PMAX is the maximum transmission power. In this model, power variables are expressed in dBm and typical values for these parameters ($G = 21$, $\alpha = 0.7$ and $PMAX = 36$) were used for energy cost calculation. We note that our design does not depend on this model, and other power models can be similarly applied here.

We applied the metric of "performance gain" (in term of energy reduction) to evaluate the energy efficiency of each algorithm, which is defined as follows.

$$gain = \frac{E_s - E}{E_s} \qquad (16)$$

where E_s is the energy cost when applying Baseline Scheduling, and E can be the energy cost of any other algorithm including Ideal Scheduling, PreSeer, and PreSeer-NT.

7.2 Performance of PreSeer

In this section, we focus on two key metrics: energy efficiency and forwarding delay.

7.2.1 Energy Efficiency

Fig. 16 shows CDF curves for the performance gain of PreSeer, PreSeer-NT and Ideal Scheduling over Baseline Scheduling with all the data sets, respectively. Fig. 16 tells that energy can be saved by looking into the future, i.e., predicting channel conditions based on ephemeral link correlation. In this figure, all three algorithms, PreSeer, PreSeer-NT and Ideal Scheduling, bring about significant performance gains, where Ideal Scheduling (the blue dotted curve) has the best performance as the lower bound of energy consumption with an average performance gain of 37.57%. PreSeer (the red dashed curve) achieves an average performance gain of 22.59% and a maximum gain of 40.68% despite practical conditions such as dynamic train velocity, irregular radio channel, hardware discrepancy. Even PreSeer-NT (the black solid curve), a degraded version of PreSeer, improves energy efficiency over Basic Scheduling by an average of 13.25%.

In Fig. 16, the performance gap between Ideal Scheduling and PreSeer is expected. Ideal Scheduling assumes accurate and real-time knowledge about not only the future channel conditions and but also incoming sensor data rates, while Preseer faces double challenges of inevitable channel prediction errors and unpredictable incoming data rates. The improvement of PreSeer over PreSeer-NT tells that the threshold-based conversion scheme successfully mitigates the negative impacts of channel prediction error.

7.2.2 Forwarding Delay

Fig. 17 shows the CDF curves of forwarding delays for PreSeer, PreSeer-NT and Ideal Scheduling. Note that we did not consider Baseline Scheduling in this case because Baseline Scheduling transmits the aggregated sensing data immediately upon its arrival without any optimization. In other words, it has almost zero forwarding delays when there is cellular coverage. In this experiment, the default deadline constraint is set to be 20 seconds along with an 1Kbps sensing data rate, a 30-second prediction duration, and a 4Kbps maximum cellular transmission data rate.

From Fig. 17, we can observe that PreSeer has the shortest delay on average. This is because the threshold-based scheduling scheme in PreSeer encourages to use the first time slot with *sufficiently* good channel quality for transmission, and this slot may not be the best choice in terms of energy efficiency. With perfect knowledge, Ideal Scheduling can make better use of the allowed delay to achieve optimal energy efficiency, which explains why Ideal Scheduling has a longer average forwarding delay than that of PreSeer in Fig. 17. For PreSeer-NT, without the threshold-based scheme, it becomes more conservative than PreSeer by trying to employ only the best predicted transmission opportunity in the near future. However, unlike Ideal Scheduling, PreSeer-NT does not know the long-term uplink quality, so each time it can only schedule a transmission based on its currently buffered sensing data, and thus sensing data collected after the scheduling operation will be delayed. As a result, PreSeer-NT has a longer average forwarding delay as it is shown in Fig. 17.

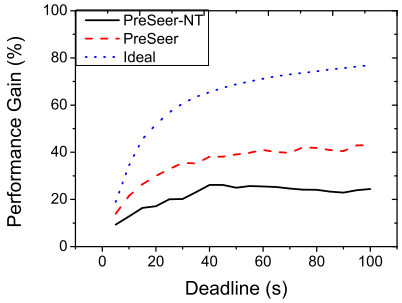

Figure 18. Impact of the Transmission Deadline Constraint.

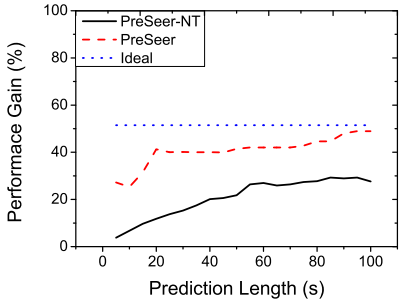

Figure 19. Impact of the Prediction Length.

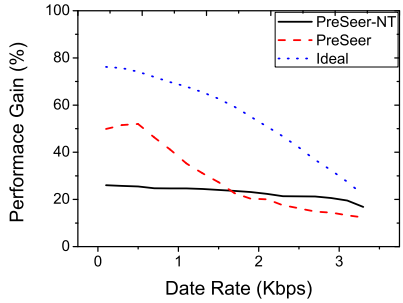

Figure 20. Impact of the Arrival Rate of Sensing Data.

7.3 Impact of Working Conditions

This section evaluates the impacts of three key factors in the system: the deadline constraint, the prediction length, and the sensing data rate.

7.3.1 Impact of the Deadline Constraint

The transmission deadline is one of the most important constraints affecting the scheduling operation. Intuitively, the more relaxing of this constraint, the better energy efficiency can be achieved, because in this case the scheduler has a larger "maneuver space" to optimize the transmission. To verify this observation, we applied different deadline constraints and the result is shown in Fig. 18 (in this experiment, a 100-second prediction length and an 1Kbps incoming (senor) data rate were used so that they cannot be the bottleneck of the system). Fig. 18 tells that the energy gain of all three algorithms improves with relaxed deadline constraints (from 0 to 100s). However, it becomes less significant when the deadline is bigger than 40s. By analyzing the original data, it is found that intervals between RSS peaks usually ranged from 20s to 40s, during which aggregated sensing data can be delivered. In other words, transmission deadline extension beyond 40s in this system may not help too much.

7.3.2 Impact of the Prediction Length

Carriages of a train have different distances to the locomotive. As a result, the number of forward RSS samples that can be leveraged for channel prediction is different at each sink node, i.e., different prediction length. Generally speaking, the further away of a sink node from the locomotive, the more information it can collect for channel prediction (or the further it can look into the future). Intuitively, a long prediction can contribute to energy efficiency. Evaluation results reported in Fig. 19 confirm such an observation when prediction lengths vary from 5 seconds to 100 seconds (in this experiment, a 100-second deadline constraint and an 1Kbps incoming data rate were applied to remove their impacts). Fig. 19 tells that with increasing prediction length, both PreSeer and PreSeerNT get improved. In contrast, the performance of Ideal Scheduling does not change over varying prediction length, which is expected since Ideal Scheduling does not need to predict at all.

7.3.3 Impact of the Arrival Rate of Sensing Data

The data arrival rate also affects the scheduler since we have a capacity-limited cellular uplink. Intuitively, if only a small amount of data needs to be delivered, there is a high probability that those data can be transmitted under good channel conditions with high energy efficiency. In this experiment, we investigated the system performance under different data arrival rates with a loose delay constraint of 100s and a long prediction length of 100s. Results are shown in Fig. 20. the performance gain. Fig. 20 tells that overall the performance gain decreases with increasing data arrival rates from 0.1Kbps to 3.5Kbps.

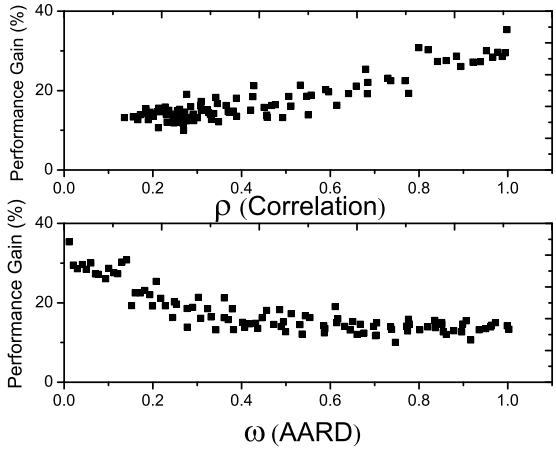

Figure 21. Performance Gain with ρ and ω.

Interestingly, PreSeer is not better than PreSeer-NT when the data arrival rate is more than 1.8Kbps. By digging into the data, we found that in this case the threshold-based scheduling scheme proposed in Sec. 5 for PreSeer degrades because of very limited "maneuver space" for scheduling. As a result, more data was sent over "acceptable" channels instead of optimal conditions. However, note that the practical data rate in our system is only about 1Kbps.

7.4 Accuracy of Channel Prediction

In this section, we report the impact of prediction accuracy to scheduling and key factors affecting the channel prediction.

7.4.1 Impact of the Prediction Accuracy

Accurate channel prediction is the precondition of effective scheduling. To investigate the impact of prediction accuracy to the performance gain in terms of energy, we carefully generated "polluted" prediction results by adding noise with different amplitudes to collected RSS samples in a controlled manner. With increasing noise, the ephemeral link correlation weakens; in other words, we have decreasing ρ (i.e., correlation coefficient) and increasing ω (i.e., relative

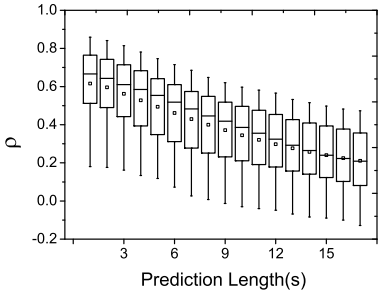

Figure 22. ρ with Different Prediction Length Using BP.

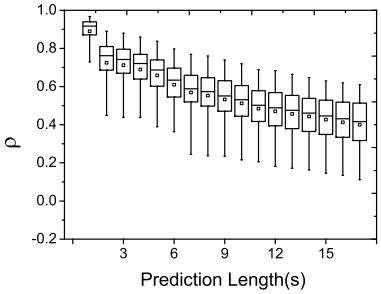

Figure 23. ρ with Different Prediction Length Using MLR.

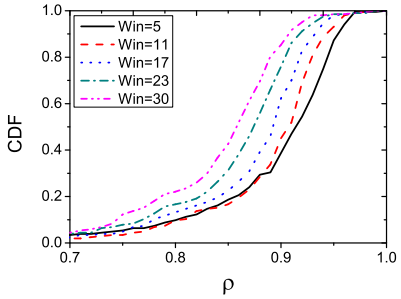

Figure 24. ρ with Different Slip Window Size.

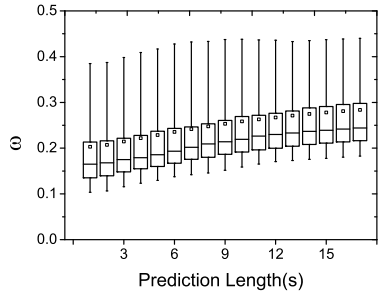

Figure 25. ω with Different Prediction Length Using BP.

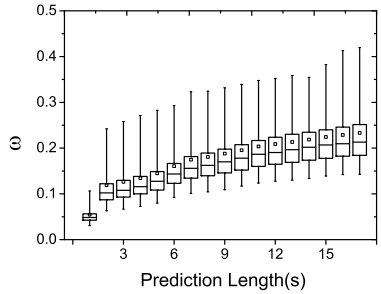

Figure 26. ω with Different Prediction Length Using MLR.

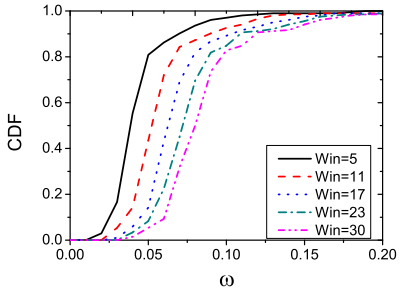

Figure 27. ω with Different Slip Window Size.

deviation) that are defined in Sec. 3. The result regarding energy performance of PreSeer in this situation is depicted in Fig. 21, which tells that the system performance improves almost monotonically with increasing ρ (and decreasing ω), showing that the correlation between the predication and its true value has a direct impact on the system performance. In the rest of this section, we further investigate the impact of key factors affecting prediction accuracy.

7.4.2 Prediction Methods and Prediction Length

In Sec. 4.5, we proposed the use of MLR (Multiple Linear Regression) to predict future RSS based on the information obtained from sink nodes in the front. A more intuitive and simpler method is to directly use such information as prediction results, which is called Basic Prediction or BP in short. We compared the accuracy of MLR and BP by focusing on ρ and ω as metrics with all empirical data sets collected in practical environments using uncontrolled vehicles.

Fig. 22, 25 and Fig. 23, 26 give boxplots (95% confidence range) of ρ and ω for BP and MLR, respectively, with different prediction lengths. By comparing Fig. 22 and Fig. 23, we can have two observations. First of all, MLR provides higher ρ values, indicating that MLR is more effective than BP. Secondly, with increasing prediction length, ρ decreases for both methods. This phenomenon is expected since the auto-correlation between a pair of RSS samples weakens with a larger temporal-spatial distance between them, especially in our case where the velocity of a train may keep varying. Note that this result does not contradict with the previous conclusion of better energy performance with longer prediction length because each sink node predicts future channel conditions based on all RSS samples in its prediction window instead of only furthest samples.

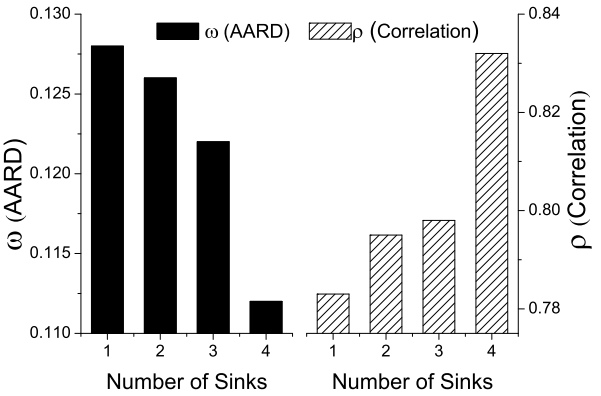

Figure 28. Impact of Sinks Used in Prediction

Fig. 25 and Fig. 26 tell similar stories from the perspective of ω, where MLR provides lower ω values than that of BP, and ω increases with extending prediction length. The above figures all confirm that the proposed MLR-based prediction method is superior to the simple baseline method.

7.4.3 MLR Key Parameters

The size of slip window is an important parameter in the MLR-based channel prediction. We evaluated the impact of this parameter with varying window sizes from 5 to 30, and CDF curves for ρ and ω are plotted in Fig. 24 and Fig. 27, respectively. From these two figures we can see that a slip window with smaller size can bring in a larger ρ and a smaller ω, namely, potentially enhanced prediction accuracy, which we consider is mainly because of the proximal effect.

Another key parameter in MLR is the number of ahead sink nodes from which RSS samples are used for channel

prediction. We evaluated its impact on ρ (correlation coefficient) and ω (relative deviation), respectively, as shown in Fig. 28 where the *x* axis is the number of sink nodes involved in prediction and *y* axes are ρ (on the left) and ω (on the right). Fig. 28 tells that a bigger ρ and a smaller ω can be obtained simultaneously with RSS samples from more sink nodes. This observation coincides with our previous discussion about the prediction length, and indicates that a sink node should make use of channel sensing results from as many sink nodes located in front it as possible.

8 Related Work

Due to space constraints, we only cover closely related literatures here. Pioneering work [24] and [19] contributed early observations of wireless link correlation in stationary WIFI and ZigBee networks. To provide better energy efficiency in mobile scenarios, a profile based method is proposed in [17] that uses the historical RSS samples to predict future link quality when passing the same road segment. It is not suitable in our application for two reasons. First, manual profiling may not be feasible due to the long distance of railway systems. Second, the cellular link quality at one location may change slowly but with considerable accumulated difference after a long period as shown in Fig. 2 and the temporal correlation decreases with increasing interval as illustrated in our study (Fig. 3). Therefore, it makes more sense to employ only short term (i.e., ephemeral) link correlations.

Many investigations [4, 16, 21, 23] have studied energy efficient scheduling methods with delay constraints. Two classes of methods are developed, i.e., on-line algorithm [16, 23] and off-line algorithm [4, 21]. The off-line algorithm assumes the knowledge of global signal conditions and data arrival rate, and the optimal results can be obtained by using the dynamic programming. In contrast, the on-line algorithm only depends on the current and historical information. And statistical methods with Markov modeling are usually applied to solve this problem. As one type of on-line method, the proposed PreSeer design departs from the state-of-the-art by proposing a new idea of channel estimation for the immediate future with historical, current, and "future" information provided by entities in the network. Importantly, unlike previous works largely ignoring prediction errors, PreSeer can tolerate different levels of prediction errors as indicated by our analysis and evaluation.

9 Conclusions

We present PreSeer, a versatile scheduling algorithm for last-hop wireless networking by leveraging ephemeral link correlations observed from an extensive empirical study. For the example application of freight train transportation, the main idea is that a mobile sink node can predict its channel conditions for the immediate future, according to channel sensing results from other sink nodes in front of it. In addition to the basic design, an error-tolerant mechanism is proposed to work with PreSeer to mitigate uncertainties caused by mobile operations. The proposed design is evaluated with real-live systems for which 400,000 data records were collected from railway routes over 7,000 km. Results show that the PreSeer design is practical, reliable, and reduces energy consumptions at sink nodes by as much as 40%.

ACKNOWLEDGEMENT

The authors thank our shepherd Dr. Utz Roedig and the anonymous reviewers for providing valuable feedbacks to this work. The work was supported in part by China NSF (Grant No. 60933011), the State Key Development Program for Basic Research of China (Grant No. 2011CB302902), US NSF grants CNS-0845994, and K. C. wong Education Foundation, Hong Kong.

10 References

[1] 3GPP TS 05.08 v8.16.0, 3rd Generation Partnership Project (3GPP), 2003-04, Radio Subsystem Link Control, 2003.

[2] L. Aiken, S. West, and S. Pitts. Multiple linear regression. *Handbook of psychology*, 2003.

[3] N. Baccour, A. Koubaa, L. Mottola, M. Zuniga, H. Youssef, C. A. Boano, and M. Alves. Radio link quality estimation in wireless sensor networks: a survey. *ACM TOSN*, 8(4):A1 –A35, nov. 2011.

[4] R. Berry and R. Gallager. Communication over fading channels with delay constraints. *IEEE Tran. on Info. Theory*, 48(5):1135–1149, 2002.

[5] D. Bertsekas and R. Gallager. *Data networks*, volume 2. Prentice-hall New Jersey, 1987.

[6] G. W. Challen, J. Waterman, and M. Welsh. Idea: integrated distributed energy awareness for wireless sensor networks. In *MobiSys '10*.

[7] G. Chandrasekaran and T. V. et al. Tracking vehicular speed variations by warping mobile phone signal strengths. In *PerCom'11*.

[8] D. Cichon, T. Zwick, and W. Wiesbeck. Radio link simulations in high-speed railway tunnels. In *ICAP'95*.

[9] A. Goldsmith and P. Varaiya. Capacity of fading channels with channel side information. *IEEE TIT*, 43(6):1986–1992, 1997.

[10] H. Hsu and F. Gale. Regional shifts in china's cotton production and use. *Cotton and Wool Situation and Outlook*, 2001.

[11] F. Jia, E. Galea, and M. Patel. The numerical simulation of the non-charring pyrolysis process and fire development within a compartment. *Applied Mathematical Modelling*, 23(8):587 – 607, 1999.

[12] E. Keogh and M. Pazzani. Derivative dynamic time warping. In *SDM'01*.

[13] A. LaMarca, Y. Chawathe, S. Consolvo, et al. Place lab: Device positioning using radio beacons in the wild. *Pervasive Computing*, pages 301–306, 2005.

[14] L. Li, G. Xing, L. Sun, W. Huangfu, R. Zhou, and H. Zhu. Exploiting fm radio data system for adaptive clock calibration in sensor networks. In *MobiSys'11*.

[15] X. Ma, J. Liu, and H. Jiang. Energy-efficient mobile data uploading from high-speed trains. *Mobile Networks and Applications*, 2011.

[16] N. Salodkar, A. Bhorkar, A. Karandikar, and V. Borkar. An on-line learning algorithm for energy efficient delay constrained scheduling over a fading channel. *Selected Areas in Communications, IEEE*, 26(4):732–742, 2008.

[17] A. Schulman, V. Navda, R. Ramjee, N. Spring, P. Deshpande, and e. a. Grunewald, C. Bartendr: a practical approach to energy-aware cellular data scheduling. In *MobiCom'10*.

[18] F. Sivrikaya and B. Yener. Time synchronization in sensor networks: a survey. *Network, IEEE*, 18(4):45–50, 2004.

[19] K. Srinivasan, M. Jain, J. I. Choi, T. Azim, E. S. Kim, P. Levis, and B. Krishnamachari. The κ; factor: inferring protocol performance using inter-link reception correlation. In *MobiCom '10*.

[20] W. Struble, F. McGrath, I. Harrington, and P. Nagle. Understanding linearity in wireless communication amplifiers. *Solid-State Circuits, IEEE*, 32(9):1310–1318, 1997.

[21] H. Wang and N. Mandayam. A simple packet-transmission scheme for wireless data over fading channels. *IEEE TOC*, 52(7), 2004.

[22] H. Zhang. Em310 datasheet. 2009.

[23] X. Zhong and C. Xu. Online energy efficient packet scheduling with delay constraints in wireless networks. In *INFOCOM'08*.

[24] T. Zhu, Z. Zhong, T. He, and Z.-L. Zhang. Exploring link correlation for efficient flooding in wireless sensor networks. In *NSDI'10*.