CONET 2012

The Third International Workshop on Networks of Cooperating Objects

April 16th, 2012
Beijing, China

CONET 2012 Workshop was organized as part of the CPS Week 2012 (http://www.cpsweek.org/). The aim of CONET 2012 was to provide a discussion forum to foster synergies and convergence of complementary areas, towards the vision of networks of Cooperating Objects. This Workshop welcomed contributions reporting on mature research, and explicitly encouraged contributions with work-in-progress systems, provocative ideas, and position papers. The Workshop program is completed with a keynote address, an invited industry presentation and a panel discussion.

All submitted papers were peer-reviewed and, as a result of this review process, 10 papers (out of 13 received) were selected to appear in the Workshop. Out of these, 5 are long papers and 5 are short papers.

These electronic pre-proceedings were produced for the benefit of discussions during the Workshop. They will not be distributed widely and are not considered to be archived publications.

Table of Contents

Long Papers

Short Papers

An Internal Model Approach for Synchronization of Linear Multi-Agent Systems Using Relative Outputs ⋆

Ji Xiang, and Wei Wei

Department of System Science and Engineering, College of Electrical Engineering, Zhejiang University, P. R. China. Email: jxiang@zju.edu.cn.

Abstract. The synchronization problem of identical linear multi-agent systems where nodes have only the relative outputs of their neighboring nodes has been investigated in this paper. A dynamical controller is constructed by virtue of the internal model principle. A sufficient and necessary condition, followed by a simple sufficient condition, is presented for synchronization under the proposed controller. It is shown that the synchronizability is is related to the maximal *eigenratiovariant* of the underlying diagraph and the dynamics of agent itself. The maximal eigenratiovariant reduces to the eigenratio of Laplacian matrix if all eigenvalues of Laplacian matrix are real. Moreover, the sufficient condition is always feasible if the system matrix has no open right half-plane eigenvalues. The efficacy of proposed controller is illustrated by a simulation example.

1 Introduction

Systems consisting of many units are more interesting than a single unit system because the charming collective phenomena only happens among multiple units. Among various collective phenomena, synchronization can be thought as the fundamental one. For example, the synchronous paddling is a key for being the winner of kayak four race and the clock synchronization is necessary for the wireless sensor network.

Studies of synchronization might be traced back to the Huygens pendulums and firstly arose in physics literature. There the considered synchronization have experienced from phase synchronization, to limit-cycle synchronization, and to chaotic synchronization. More details can refer to a recent survey and references therein [1]. A feature of these synchronization studies is that the considered systems are nonlinear. By contrast, most synchronization studies in the field of control focus on linear systems. The seminal paper by Jadbabaie et. al., investigated the synchronization problem in networks of first-order discrete integrator

⋆ This research was supported by a grant from the National 863 Program of China (2011AA050204), the National Natural Science Foundation of China (60704030, 61074122), and the Fundamental Research Funds for the Central Universities (2011QNA4010).

agents [2]. There all the agents have their states converge a constant value if the network topology satisfies the jointly connected condition. In such a case, the steady state is a constant point depending on the initial condition so that the term "consensus" is utilized. Since then, the consensus problem of first-order multi-agent systems has been extensively studied. The consensus problem of continuous-time systems was investigated in [3], where three cases including switching topology and time-delay communication are discussed. In [4], the set-valued Lyapunov method was proposed for the discrete time consensus of the first-order multi-agent systems with time-dependent topology. The stochastic consensus problem was studied in [5].

When the agent has general linear dynamics with higher orders, the steady state is not a constant point but a trajectory determined by the self-dynamics of agent, being similar to the synchronization studies in the field of physics. In such a case it seems more preferable to use the term "synchronization" instead of "consensus". The synchronization problem is more difficult and challenging than the consensus problem, even if the agents are limited to linear dynamics. There are at least three factors influencing the synchronization of multi-agent systems: the self-dynamics of agent, the network topology and the inner-coupling way. The third factor determines the information representation of node at its neighboring nodes. In [6], the synchronization problem of the double-integrator multi-agent systems was studied with the static state feedback. It is show that the existence of a spanning tree is a necessary rather than a sufficient condition for achieving synchronization. The necessary and sufficient condition of synchronization of the double-integrator multi-agent systems was presented in [7], where the qualifiable feedback gains closely depend on the complex eigenvalues of Laplacian matrix. .

This paper studies the synchronization problem of linear multi-agent systems with the general high-order dynamics. Our focus is on the case that only the relative output information of neighboring nodes is available for synchronization seeking. For synchronization of such a case, the static feedback controller is in general not workable, recalling the fact that it is still an open problem to design a static output feedback controller. Lunze [8] shows that under static output feedback, the synchronization problem is equivalent to the static output feedback simultaneous stabilization problem. Tuna [9] proposed a static feedback controller by using the relative output information for synchronization of discrete-time linear multi-agent systems, under the assumptions that the input matrix is a unit matrix and the system matrix is neutrally stable. Scardovi and Sepulchre [10] presented a dynamic output feedback controller for synchronization of multi-agent systems with general linear dynamics in , where the inner state of dynamical controller of neighboring nodes should be available for seeking synchronization. Li, et. al., [11] proposed an observer-type dynamical controller that also requires the information exchanging of controller states. It should be pointed out that the requirement of controller states is different in essence from that of output states. The inner controller state is artificial and in general must rely on the communication channel to obtain; while the output state can be directly measured without any communication channel. Seo et. al. proposed a low

gain dynamical controller using only the relative output information for synchronization problem of linear multi-agent systems, under the assumption that the system matrix has no open right half-plane eigenvalues in [12]. This assumption is critical for the feasibility of the low gain approach.

In this paper, a dynamical controller based on internal model approach is constructed for synchronization seeking. It should be recognized that recently there are several works using internal model principle to build the dynamic controller for output synchronization. In [13], a dynamical controller based on the state information exchanging is presented for synchronized output regulation. In [14], a dynamical controller using the low gain approach is presented for output synchronization of heterogenous multi-agent systems. There the desired output trajectory is predefined by a neutrally stable internal model. In [15], a dynamical controller is presented for output synchronization of multi-agent systems with switching topology, but the exchanging of internal model state is required. Noticing that the internal model in general is used for the output manipulation, it is new here to use it for the state synchronization, especially the incorporated internal model in the feedback path is just the agent itself. More importantly, the proposed internal-model-based dynamical controller only requires the information exchanging of outputs and does not restrict that the system matrix has no open right half-plane eigenvalues, either or both of them are assumed in the most synchronization studies reported before.

The remainder of this paper is organized as follows. The problem formulation is presented in Section 2. The internal mode controller is given in Section 3. The synchronization analysis is made in Section 4, where a necessary and sufficient condition for synchronization, as well as a simple and tractable sufficient condition, is presented. It is shown that the eigenratiovariant of Laplacian matrix plays a key role for synchronization of multi-agent systems with general dynamics. The feasibility of sufficient condition, as well as the synthesis algorithm, is addressed in Section 5. A simulation example is made in Section 6, followed by a conclusion in Section 7. All the proofs of theorem are listed in the Appendix.

2 Problem formulation

Consider a network of N identical linear systems given by

$$\begin{cases} \dot{x}_i = Ax_i + Bu_i \\ y_i = Cx_i \end{cases}, \quad i = 1, \cdots, N, \tag{1}$$

where $x_i \in \mathcal{R}^n$, $u_i \in \mathcal{R}^m$ and $y_i \in \mathcal{R}^p$ are the state, input and output of the ith agent, respectively. The network topology is assumed to be fixed and is represented by a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \cdots, v_N\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. An edge of $\mathcal{G}$ is denoted by $\varepsilon_{ij} = (v_i, v_j)$. The jth node has the information of node i if and only if $\varepsilon_{ij} \in \mathcal{E}$.

Denote by $\mathcal{N}_i$ the in-neighboring set of node i. $j \in \mathcal{N}_i$ if and only if $\varepsilon_{ji} \in \mathcal{E}$. Let $\mathcal{A} = (a_{ij})$ be the adjacency matrix of $\mathcal{G}$. $a_{ij} > 0$ if $\varepsilon_{ji} \in \mathcal{E}$; or else $a_{ij} = 0$. The

graph Laplacian L associated with $\mathcal{G}$ is defined as $l_{ij} = -a_{ij}$ and $l_{ii} = \sum_{j\neq i} a_{ij}$. Let λ_i, $i = 1, \cdots, N$, be the eigenvalues of L with the order of $\lambda_1 \leq \mathrm{Re}(\lambda_2) \cdots \leq \mathrm{Re}(\lambda_N)$. Denote by $\mathbf{b}$ the vector with all elements being 1. It is known that $\mathbf{b}$ is the right eigenvector of L associated with the trivial eigenvalue $\lambda_1 = 0$. Denote by ϕ_i the expression of the information of neighboring nodes of set $\mathcal{N}_i$ at node i. If $\phi_i = \mathrm{col}(\cdots, x_{i_j}, \cdots)$, with $i_j \in \mathcal{N}_i$, then node i has the state information of its neighboring nodes; if $\phi_i = \sum_{j\in\mathcal{N}_i} a_{ij}(y_j - y_i)$, then node i has the sum of relative output information of its neighboring nodes.

The considered synchronization problem is formally defined as follow,

Definition 1 (Synchronization problem). *Design a distributed control law $u_i = f(x_i, \phi_i)$, such that the solution of (1) asymptotically synchronize to a solution of isolated node $\dot{x}_0 = Ax_0$.*

Most studies on the synchronization problem assume either that ϕ_i includes some inner controller states [11] or that A is neutrally stable [12] [9] [1]. In the current paper, the above two assumptions are removed with the penalty of some restrictions on the qualifiable network topologies. The following assumptions are required in this paper,

A1) (A, B) is stabilizable.
A2) (A, C) is detectable.
A3) Laplacian L has only one 0 eigenvalue.
A4) $\phi_i = \sum a_{ij}(y_j - y_i) = -\sum l_{ij} y_j$.

Assumption A3) means that $\mathcal{G}$ contains a spanning tree [18]. Assumption A4) indicates that for each agent, the available information of its neighboring nodes is only the weighted sum of relative outputs.

3 Internal model controller

The prototype form of proposed internal model controller is

$$\begin{cases} \dot{\xi}_i = A\xi_i + G\phi_i \\ u_i = K(\xi_i - x_i) \end{cases} \tag{2}$$

where $\xi_i \in \mathcal{R}^n$ is the state of controller, G and K are the feedback gain matrices. Define $\tilde{e}_i = x_i - \xi_i$, whose dynamics is governed by

$$\dot{\tilde{e}}_i = (A - BK)\tilde{e}_i - G\phi_i. \tag{3}$$

[1] There are also several works for the synchronization of nonlinear networked systems under output feedbacks with the assumption that the node self-dynamics is passive [16] [17]. The passivity means that the node dynamics is stable itself and has outputs of the same dimensions with inputs.

At the synchronization state, $\phi_i = 0$, the dynamics of closed-loop system of $\tilde{e}_i$ and ξ_i is governed by

$$\begin{cases} \dot{\tilde{e}}_i = (A - BK)\tilde{e}_i \\ \dot{\xi}_i = A\xi_i \end{cases}, \tag{4}$$

where the state of controller will be evolving by itself; while the state of agent will asymptotically converge to that of controller, if K is such that $A-BK$ is Hurwitz. This is the salient difference in contrast to the observer-type controller [11], where at the synchronization state the state of controller will converge to that of agent, like an observer. Here, the dynamical controller (2) plays an additional internal model, noting that the agent itself was such an internal model. The route to synchronization under (2) is that each agent follows its internal model and the internal models synchronize by adjusting its state according to the relative output errors.

Since the modes associated with the open left half-plane eigenvalues of A will exponentially decay to zero and will play nothings for the asymptotic behavior of an internal model, naturally we can remove these decaying modes to construct a reduced-order internal mode controller.

Without loss of generality, assume that A has $n_1 > 0$ (including the identical ones) eigenvalues with nonnegative real parts. Let T be a non-singular matrix such that

$$T^{-1}AT = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \tag{5}$$

where $A_1 \in \mathcal{R}^{n_1 \times n_1}$ has n_1 eigenvalues with nonnegative real part, and $A_2 \in \mathcal{R}^{(n-n_1)\times(n-n_1)}$ is Hurwitz. Partition T by $T = \begin{bmatrix} T_1 & T_2 \end{bmatrix}$, where $T_1 \in \mathcal{R}^{n \times n_1}$. It follows that $T_1 A_1 = A T_1$.

Now, we can construct the following reduced-order internal mode controller,

$$\begin{cases} \dot{\xi}_i = A_1\xi_i + G\phi_i \\ u_i = K(T_1\xi_i - x_i) \end{cases}, \tag{6}$$

where $\xi_i \in \mathcal{R}^{n_1}$ is the state of controller, $K \in \mathcal{R}^{m \times n}$ and $G \in \mathcal{R}^{n_1 \times p}$ are gain matrices to be designed later. When $n_1 = n$, $T_1 = I$ and then (6) becomes (2), and therefore the prototype controller (2) is a special case of (6).

4 Synchronization Analysis

In this section, we analyze the synchronization of (1) under controller (6). The obtained results are also valid for that under (2).

Define $e_i = x_i - T_1\xi_i$. Noticing $T_1A_1 = AT_1$, the closed-loop system of e_i and ξ_i is

$$\begin{cases} \dot{e}_i = (A - BK)e_i + \sum\limits_{j\in\mathcal{N}_i} l_{ij} T_1 GC(e_j + T_1\xi_j) \\ \dot{\xi}_i = A_1\xi_i - \sum\limits_{j\in\mathcal{N}_i} l_{ij} GC(e_j + T_1\xi_j) \end{cases}. \tag{7}$$

Define the matrix $\bar{A}_i$ dependent on λ_i by

$$\bar{A}_i = \begin{bmatrix} A - BK + \lambda_i T_1 GC & \lambda_i T_1 GCT_1 \\ -\lambda_i GC & A_1 - \lambda_i GCT_1 \end{bmatrix}. \tag{8}$$

Then the following result can be established,

Theorem 1. *The synchronization problem of (1) is solvable by controller (6) if and only if K and G are such that $A - BK$ and $\bar{A}_i$, $i = 2, \cdots, N$, are Hurwitz. And moreover, the synchronization state is governed by*

$$\dot{x}_0 = Ax_0, \quad \text{with} \quad x_0(0) = T_1 \sum_{j=1}^{N} \nu_j \xi_j(0) \tag{9}$$

where ν_j is jth element of left eigenvector ν of L associated with eigenvalue $\lambda_1 = 0$, satisfying $\sum_j \nu_j = 1$.

Theorem 1 gives the necessary and sufficient condition for the synchronization problem, which is important but less tractable for the synthesis problem, since the feedback gain matrices should simultaneously stabilize N subsystems. Below we will provide a sufficient but more tractable condition for the solvability of synchronization problem under controller (6).

Before to proceed, let us firstly introduce some notations about eigenvalues of Laplacian L. Let α_i and β_i be such that $\alpha_i + j\beta_i = \lambda_i$ is the ith eigenvalue of Laplacian L. If $\beta_i \neq 0$, certainly λ_i^* is also an eigenvalue of L. The superscript $*$ denotes the conjugate transpose throughout this paper. Let $|\lambda_i|$ be the modulus of λ_i, i.e., $|\lambda_i| = \sqrt{\alpha_i^2 + \beta_i^2}$. Let $\theta_i = |\lambda_i| \left(\frac{|\beta_i| + |\lambda_i|}{\alpha_i} \right)$ denote the *modulusvariant* associated with λ_i, where for $\lambda_1 = 0$, $\theta_1 = 0$ is set for the consistence of definition. Define by $r_i = \frac{\theta_i}{\alpha_2}$ the *eigenratiovariant* of λ_i. Denote the maximal eigenratiovariant of L by

$$\bar{r} = \max_i r_i. \tag{10}$$

Theorem 2. *The synchronization problem of (1) is solvable by controller (6) if there are two symmetric positive definite matrices $\bar{P} \in \mathcal{R}^{n \times n}$ and $\bar{Q} \in \mathcal{R}^{n_1 \times n_1}$, and matrix $\bar{Y} \in \mathcal{R}^{m \times n}$ such that the following two matrix inequalities hold*

$$A\bar{P} + \bar{P}A^T - B\bar{Y} - \bar{Y}^T B^T + T_1 \bar{Q} T_1^T C^T C T_1 \bar{Q} T_1^T + \bar{r}^2 \bar{P} C^T C \bar{P} < 0, \tag{11}$$

$$A_1 \bar{Q} + \bar{Q} A_1^T - \bar{Q} T_1^T C^T C T_1 \bar{Q} < 0. \tag{12}$$

Moreover, the qualifiable gain matrices are given by

$$K = \bar{Y}\bar{P}^{-1}, \quad G = \frac{1}{\alpha_2} \bar{Q} T_1^T C^T. \tag{13}$$

Remark 1. Theorem 2 simplifies the problem finding K and G that simultaneously stabilize N matrices to the problem solving two low-order matrix inequalities that are semi-coupled in the sense that the second inequality is independent on the first one, with the price of some conservatism. (34) illustrates the mechanism of such a simplification, the different Lyapunov functions dependent on the eigenratiovariant associated with λ_i are selected to show the stability of different A_is with the same feedback gain matrices.

Remark 2. The influence on synchronization of network topology is reflected by the eigenratiovariant of eigenvalues of Laplacian matrix, which points out that both the real part and the image part of eigenvalue will influence the synchronization seeking. The smaller the maximal eigenratiovariant, the more synchronizable the network. If L has no complex eigenvalues, such as the undirected graph, then $\theta_i = \alpha_i$ and $\bar{r}$ reduces to eigenratio α_N/α_2 that has been extensively utilized to depict the synchronizability of a network topology [19, 20].

5 Feasibility of sufficient condition and synthesis algorithm

In this section, we will discuss the feasibility of the sufficient condition in Theorem 2, and give a general synthesis algorithm for the controller (6).

Lemma 1. *Given Assumption A2), the matrix pair* (A_1, CT_1) *is observable.*

This can be straightly derived by the Popov-Belevitch-Hautus test [21]. Lemma 1 further implies that inequality (12) is always feasible by itself.

Theorem 3. *Given Assumptions A1)∼A3). If A has no eigenvalue with positive real part, then the proposed sufficient condition consisting of* (11) *and* (12) *is always feasible for any* $\bar{r}$.

Remark 3. In [12], it is shown that for the case that A has no eigenvalue with positive real part, the synchronization problem can be solved by an observer-based compensator of order n, which is designed by the low gain approach. The same result can be obtained by the proposed controller (6) with a less order of n_1. Moreover, how to get a qualifiable $\bar{\epsilon}$ is not clearly pointed out in [12]; while here inequality (37) gives a way to judge whether $\bar{\epsilon}$ is qualifiable.

As shown in [22], solution $Q(\epsilon)$ of equation (37) satisfies $\|Q(\epsilon_1)\| \leq \|Q(\epsilon_2)\|$ if $\epsilon_1 \leq \epsilon_2$. According to this and the above proof, the following synthesis algorithm of controller (6) for general triple (A, B, C) can be established,

Synthesis algorithm for internal model controller (6)

S1) Take the block diagonalization of A to obtain A_1 and T_1, and calculate the maximal eigenratiovariant $\bar{r}$.

S2) Set $\epsilon = 0$.

S3) Solve the algebraic Riccati equation

$$A_1 Q + Q A_1^T - Q T_1^T C^T C T_1 C Q + \epsilon I = 0, \tag{14}$$

to obtain solution $Q(\epsilon)$ and calculate $M(\epsilon) = T_1 Q(\epsilon) T_1^T C^T C T_1 Q(\epsilon) T_1^T$,

S4) Solve the following linear matrix inequality

$$\begin{bmatrix} AP + PA^T - BY - Y^T B^T + M(\epsilon) & PC^T \\ CP & -1/\bar{r}^2 \end{bmatrix} < 0 \tag{15}$$

.

S5) If (15) is infeasible with $\epsilon = 0$, then the synchronization problem might be not solvable for multi-agent system (1). Stop; or else the synchronization problem is solvable, go to next step.

S6) Arbitrarily select a $\epsilon > 0$ to run S3) and S4) to get a solution of Q, P and Y such that the norms of Q, Y and P^{-1} are relatively small.

S7) Obtain the feedback gain matrices by (13).

The state of agent itself x_i is assumed to be accessible in controller (6). This requirement can be weakened to that only the output of agent itself y_i is available. For such a case, the internal model controller has the following form

$$\begin{cases} \dot{\hat{x}}_i = A\hat{x}_i + Bu_i + H(y_i - C\hat{x}_i) \\ \dot{\xi}_i = A_1 \xi_i + G\phi_i \\ u_i = K(T_1 \xi_i - \hat{x}_i) \end{cases} \tag{16}$$

where the observer system of $\hat{x}_i$ is added to estimate the state of agent itself with $H \in \mathcal{R}^{n\times p}$ being the observer gain matrix. Notice that the dynamics of the estimated error $\hat{e}_i = x_i - \hat{x}_i$ is independent on the x_i and ξ_i. By the separation principle, the following result can be directly established,

Theorem 4. *The synchronization problem of (1) is solvable by controller (16) if there are two symmetric positive definite matrices $\bar{P} \in \mathcal{R}^{n\times n}$ and $\bar{Q} \in \mathcal{R}^{n_1 \times n_1}$, matrix $\bar{Y} \in \mathcal{R}^{m\times n}$ such that both (11) and (12) hold, and matrix $H \in \mathcal{R}^{n\times p}$ such that $A - HC$ is Hurwitz.*

6 Simulation example

We consider four agents with the same dynamics

$$\dot{x}_i = \begin{bmatrix} -1.8 & 0 & -1 \\ 4.2 & -1 & 2.2 \\ 5 & 0 & 2.2 \end{bmatrix} x_i + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_i, \quad y_i = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} x_i, \quad i = 1, \cdots, 4. \tag{17}$$

Suppose the Laplacian matrix to represent the information flow among four agents is the same as that in [12],

$$L = \begin{bmatrix} 6 & -1 & -2 & -3 \\ -3 & 4 & -1 & 0 \\ 0 & 0 & 2 & -2 \\ -1 & -2 & -3 & 6 \end{bmatrix}.$$

It can be verified that Assumption A1) $\sim$ A3) hold for the above multi-agent system. The eigenvalue of L is $\{0, 3.35, 7.32 \pm j1.12\}$. The maximal eigenratiovariant is $\bar{r} = 2.5772$. The block diagonalization of A is

$$T^{-1}AT = \begin{bmatrix} 0.2 & 1 & 0 \\ -1 & 0.2 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad \text{with,} \quad T = \begin{bmatrix} -1 & 0 & 0 \\ 1 & 1 & 1 \\ 2 & 1 & 0 \end{bmatrix}. \tag{18}$$

The self-dynamics of agent has one negative eigenvalue and a pair open right half-plane complex eigenvalues. Suppose the state of agent itself is available, then the internal model controller (6) can be constructed with

$$A_1 = \begin{bmatrix} 0.2 & 1 \\ -1 & 0.2 \end{bmatrix}, T_1 = \begin{bmatrix} -1 & 0 \\ 1 & 1 \\ 2 & 1 \end{bmatrix}, K = \begin{bmatrix} 26.1804 & 1.7444 & 27.4942 \end{bmatrix}, G = \begin{bmatrix} 0.0818 \\ 2.1720 \end{bmatrix} \tag{19}$$

where K and G are obtained by running the synthesis algorithm with $\epsilon = 20$. It should be noted that here (A, C) is not observable while (A_1, CT_1) is observable.

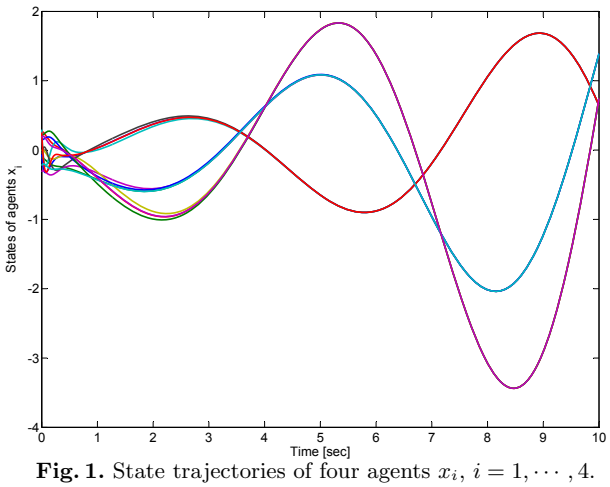


Fig. 1. State trajectories of four agents x_i, $i = 1, \cdots, 4$.

The simulation results are shown in Fig. 1 with the initial states of agent and controller produced randomly in the range of $(-0.5, 0.5)$. It can be seen that after transition time the synchronization is achieved under the proposed internal model controller (6), although the synchronization state is divergent due to the existence of right half-plane eigenvalues of A.

7 Conclusion

A dynamical controller constructed by the internal model approach has been proposed for the synchronization problem of multi-agent systems with general linear dynamics. Only the relative output information of neighboring agents has been used, and the exchanging of controller state of neighboring agents is not required. A sufficient and necessary condition for solvability of synchronization problem under the proposed controller, as well as a simple but tractable sufficient condition, has been given. These conditions are closely related to the maximal eigenratiovariant of Laplacian matrix, and are independent on the coupling strength. The sufficient condition always holds when the network has a spanning tree embedded and the system matrix has no eigenvalue with positive real parts. A simulation example illustrates the efficacy of analytic results.

8 Appendix

8.1 Proof of Theorem 1

Proof. Let $\bar{e}_i = \mathrm{col}(e_i, \xi_i) \in \mathcal{R}^{n+n_1}$ and let $\bar{e} \in \mathcal{R}^{(n+n_1)N}$ be the concatenate vectors of $\bar{e}_i$. The compact form of (7) is

$$\dot{\bar{e}} = \left(I_N \otimes \begin{bmatrix} A - BK & 0 \\ 0 & A_1 \end{bmatrix}\right) \bar{e} - \left(L \otimes \begin{bmatrix} -T_1 GC & -T_1 GCT_1 \\ GC & GCT_1 \end{bmatrix}\right) \bar{e} \tag{20}$$

Let $\bar{T}$ be the matrix to put L in the Jordan canonical form. Without loss of generality, let the first column of $\bar{T}$ be $\mathbf{b}$, the right eigenvector associated with $\lambda_1 = 0$. Correspondingly, the first row of $\bar{T}^{-1}$ is ν. Define $\bar{x} = (\bar{T}^{-1} \otimes I_{n+n_1})\bar{e}$. Let $\bar{x}_i \in \mathcal{R}^{n+n_1}$, $i = 1, \cdots, N$, denote the N partitions of $\bar{x}$. Then one has,

$$\begin{bmatrix} \dot{\bar{x}}_1 \\ \dot{\bar{x}}_2 \\ \ddots \\ \dot{\bar{x}}_N \end{bmatrix} = \begin{bmatrix} \bar{A}_1 \bar{x}_1 & 0 & \cdots & 0 \\ 0 & \bar{A}_2 \bar{x}_2 & \star & 0 \\ \vdots & \vdots & \ddots & \star \\ 0 & 0 & \cdots & \bar{A}_N \bar{x}_N \end{bmatrix} \tag{21}$$

where $\star$ denotes the possible nonzero coupled term.

(Sufficiency) Since $\bar{A}_i$ is Hurwitz for all $i = 2, \cdots, N$, $\bar{x}_i \to 0$ for all $i = 2, \cdots, N$, by which it follows that as $t \to \infty$,

$$\bar{e} = (\bar{T} \otimes I_{n+n_1})\bar{x} \to \mathbf{b} \otimes \bar{x}_1. \tag{22}$$

This means that $\bar{e}_i \to \bar{x}_1$ for all i, therefore the synchronization is achieved. The remainder is to show that the synchronization state satisfies the dynamical function $\dot{x}_0 = Ax_0$. Notice that

$$\dot{\bar{x}}_1 = \bar{A}_1\bar{x}_1 = \begin{bmatrix} A - BK & 0 \\ 0 & A_1 \end{bmatrix} \bar{x}_1 \tag{23}$$

Let $\bar{x}_1 = \text{col}(\bar{x}_{11}, \bar{x}_{12})$ with $\bar{x}_{11} \in \mathcal{R}^n$. Since $A - BK$ is Hurwitz, $\bar{x}_{11} \to 0$ and $\bar{x}_{12} \to e^{A_1 t}\bar{x}_{12}(0)$. Combination this with $\bar{e}_i \to \bar{x}_1$ obtains that $x_i \to T_1 e^{A_1 t}\bar{x}_{12}(0) = e^{At}T_1\bar{x}_{12}(0)$, where the equality arises from $T_1A_1 = AT_1$.

This means that all the agents will have their states converge to the synchronization state determined by $\dot{x}_0 = Ax_0$ with the initial condition $x_0(0) = T_1\bar{x}_{12}(0)$. Noting that $\bar{x}_1 = (\nu \otimes I_{n+n_1})\bar{e}$, it follows that $\bar{x}_{12}(0) = \sum_j \nu_j\xi_j(0)$. Thus, (9) holds.

(Necessity) At the synchronization state, $\phi_i = 0$ and

$$\begin{cases} \dot{x}_i = (A - BK)x_i + BKT_1\xi_i \\ \dot{\xi}_i = A_1\xi_i \end{cases}, \tag{24}$$

which firstly implies that $A - BK$ should be Hurwitz in order for decaying the action of different initial conditions and secondly implies that by output regulation theory [23], $x_i \to \Pi\xi_i$, with Π satisfying the Sylvester equation

$$\Pi A_1 = (A - BK)\Pi + BKT_1. \tag{25}$$

Because A_1 has all eigenvalues in the right half-plane, the above equation has a unique solution $\Pi = T_1$. Noting that T_1 is full of column rank, it follows that the state of controller has also achieved synchronization at the synchronization state, which together with the asymptotical convergence of synchronization state means, $\|\bar{e}_i - \bar{e}_j\| \to 0$. From $\bar{x} = (\bar{T}^{-1} \otimes I_{n+n_1})\bar{e}$ and $\bar{T}^{-1}\mathbf{b} = \text{col}(1, 0, \cdots, 0)$, one has $\bar{x}_i \to 0$ for all $i = 2, \cdots, N$, for any initial conditions. This shows that $\bar{A}_i$ is Hurwitz and thus the proof is completed.

8.2 Proof of Theorem 2

Proof. Firstly consider $\bar{A}_i$, $i = 2, \cdots, N$, which is Hurwitz if and only if the following dynamical system

$$\dot{\eta}_i = \bar{A}_i^*\eta_i = \begin{bmatrix} A^T - K^TB^T + \lambda_i^*C^TG^TT_1^T & -\lambda_i^*C^TG^T \\ \lambda_i^*T_1^TC^TG^TT_1^T & A_1^T - \lambda_i^*T_1^TC^TG^T \end{bmatrix} \eta_i \tag{26}$$

will exponentially converge to zero for any initial state $\eta_i(0) \in \mathcal{R}^{n+n_1}$. Partition η_i by $\eta_i = \text{col}(\eta_{i1}, \eta_{i2})$. Consider the Lyapunov function $V_i = \eta_{i1}^*P_i\eta_{i1} + \eta_{i2}^*Q\eta_{i2}$. With $G = QT_1^TC^T$, the time derivative of V_i along (26) is

$$\begin{aligned} \dot{V}_i = {} & \eta_{i1}^*[P_i(A - BK)^T + (A - BK)P_i + \lambda_i^*P_iC^TCT_1QT_1^T + \lambda_iT_1QT_1^TCC^TP_i]\eta_{i1} \\ & - \eta_{i1}^*\lambda_i^*P_iC^TCT_1Q\eta_{i2} - \eta_{i2}^*\lambda_iQT_1^TC^TCP_i\eta_{i1} + \eta_{i2}^*\lambda_i^*QT_1^TC^TCT_1QT_1^T\eta_{i1} \\ & + \eta_{i1}^*\lambda_iT_1QT_1^TC^TCT_1Q\eta_{i2} + \eta_{i2}^*(QA_1^T + A_1Q - 2\alpha_iQT_1^TC^TCT_1Q)\eta_{i2}. \end{aligned} \tag{27}$$

The four crossing terms can be enlarged by

$$\begin{aligned}
&\eta_{i1}^*(-\lambda_i^* P_i C^T + \lambda_i T_1 Q T_1^T C^T) C T_1 Q \eta_{i2} + \eta_{i2}^* Q T_1^T C^T (-\lambda_i C P_i + \lambda_i^* C T_1 Q T_1^T) \eta_{i1} \\
&\le \eta_{i1}^* \frac{\left(\lambda_i T_1 Q T_1^T C^T - \lambda_i^* P_i C^T\right)\left(\lambda_i^* C T_1 Q T_1^T - \lambda_i C P_i\right)}{\alpha_i} \eta_{i1} \\
&\qquad + \alpha_i \eta_{i2}^* Q T_1^T C^T C T_1 Q \eta_{i2} \\
&= \frac{|\lambda_i|^2}{\alpha_i} \eta_{i1}^* ((P_i C^T C P_i + T_1 Q T_1^T C^T C T_1 Q T_1^T) \eta_{i1} + \eta_{i2}^* Q T_1^T C^T C T_1 Q \eta_{i2} \\
&\qquad + \eta_{i1}^* \left(\frac{-(\lambda_i^*)^2}{\alpha_i} P_i C^T C T_1 Q T_1^T + \frac{-(\lambda_i)^2}{\alpha_i} T_1 Q T_1^T C^T C P_i \right) \eta_{i1},
\end{aligned} \tag{28}$$

where $\alpha_i > 0$, $i = 2, \cdots, N$, due to Assumption A3. Also one has

$$\begin{aligned}
&\eta_{i1}^* \left(\left(\lambda_i^* - \frac{(\lambda_i^*)^2}{\alpha_i} \right) P_i C^T C T_1 Q T_1^T + \left(\lambda_i - \frac{\lambda_i^2}{\alpha_i} \right) T_1 Q T_1^T C^T C P_i \right) \eta_{i1} \\
&\le \frac{\beta_i}{\alpha_i} |\lambda_i| \eta_{i1}^* (P_i C^T C P_i + T_1 Q T_1^T C^T C T_1 Q T_1^T) \eta_{i1}.
\end{aligned} \tag{29}$$

Notice that $\frac{|\beta_i|}{\alpha_i}|\lambda_i| + \frac{|\lambda_i|^2}{\alpha_i} = \theta_i$. Combining (27), (28) and (29), one has

$$\dot{V}_i \le \eta_{i1}^* W_1 \eta_{i1} + \eta_{i2}^* W_2 \eta_{i2} \tag{30}$$

where

$$\begin{aligned}
W_1 &= P_i (A - BK)^T + (A - BK) P_i + \theta_i (T_1 Q T_1^T C^T C T_1 Q T_1^T + P_i C^T C P_i), \\
W_2 &= Q A_1^T + A_1 Q - \alpha_i Q T_1^T C^T C T_1 Q.
\end{aligned} \tag{31}$$

Notice that both $W_1 < 0$ and $W_2 < 0$ simultaneously hold for all $i = 2, \cdots, N$ if

$$P_i (A - BK)^T + (A - BK) P_i + r_i \alpha_2 T_1 Q T_1^T C^T C T_1 Q T_1^T + \frac{\bar{r}^2}{r_i} \alpha_2 P_i C^T C P_i < 0, \tag{32}$$

and

$$Q A_1^T + A_1 Q - \alpha_2 Q T_1^T C^T C T_1 Q < 0 \tag{33}$$

simultaneously hold for some Q, K and P_i, which are nothing but (11) and (12) by taking

$$P_i = \frac{r_i}{\alpha_2} \bar{P}, \quad Q = \frac{1}{\alpha_2} \bar{Q}, \quad \text{and} \quad K = \bar{Y} \bar{P}^{-1}. \tag{34}$$

Thus, $\dot{V}_i < 0$ for all nonzero η_i. Subsequently, $\bar{A}_i$ is Hurwitz. On the other hand, inequality (32) implies that $A - BK$ is also Hurwitz. By Theorem 1, the synchronization problem of (1) is solvable by controller (6) with the gain matrices given by (13). The proof is completed.

8.3 Proof of Theorem 3

Proof. For any given L satisfying Assumption A3), $\bar{r}$ is well-defined and bounded. Since (A, B) is stabilizable, there exist P and Y such that for sufficiently large $\rho > 0$

$$AP + PA^T - BY - Y^T B^T + \frac{\bar{r}^2}{\rho} PC^T CP < 0. \tag{35}$$

Multiplying both sides by $\frac{1}{\rho}$ and Taking $\bar{P} = \frac{1}{\rho} P$ and $\bar{Y} = \frac{1}{\rho} Y$ yields

$$A\bar{P} + \bar{P}A^T - B\bar{Y} - \bar{Y}^T B^T + \bar{r}^2 \bar{P} C^T C\bar{P} < 0, \tag{36}$$

from which it follows that

$$A\bar{P} + \bar{P}A^T - B\bar{Y} - \bar{Y}^T B^T + \bar{r}^2 \bar{P} C^T C\bar{P} + \varepsilon I < 0, \tag{37}$$

where $\varepsilon > 0$ is a sufficiently small scalar. On the other hand, since (A_1, CT_1) is observable by Lemma 1, the following algebraic Riccati equation

$$A_1 Q + QA_1^T - QT_1^T C^T CT_1 Q + \epsilon I = 0 \tag{38}$$

has a unique solution $Q(\epsilon) \in \mathcal{R}^{n_1 \times n_1}$, where $\epsilon > 0$ is any positive scalar. Noticing that $\lim_{\epsilon \to 0} Q(\epsilon) = 0$ [22,24], when A_1 has no eigenvalue with positive part, there is a sufficiently small $\bar{\epsilon}(\varepsilon)$ such that for all $\epsilon < \bar{\epsilon}(\varepsilon)$ the solution $Q(\epsilon)$ of (37) satisfies

$$T_1 Q(\epsilon) T_1^T C^T CT_1 Q(\epsilon) T_1^T < \varepsilon I. \tag{39}$$

Combining (37), (38) and (39), it follows that the condition consisting of (11) and (12) holds for any $\bar{r}$. The proof is completed.

References

[1] Alex Arenas, Albert Daz-Guilera, Jurgen Kurths, Yamir Moreno, and Changsong Zhou. Synchronization in complex networks. *Physics Reports*, 469(3):93 – 153, 2008.

[2] A. Jadbabaie, Jie Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Autom. Control*, 48(6):988 – 1001, june 2003.

[3] R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control*, 49(9):1520 – 1533, sept. 2004.

[4] L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Trans. Autom. Control*, 50(2):169 – 182, feb. 2005.

[5] Y. Hatano and M. Mesbahi. Agreement over random networks. *IEEE Trans. Autom. Control*, 50(11):1867 – 1872, nov. 2005.

[6] Wei Ren and Ella Atkins. Distributed multi-vehicle coordinated control via local information exchange. *International Journal of Robust and Nonlinear Control*, 17(10-11):1002–1033, 2007.

[7] Wenwu Yu, Guanrong Chen, and Ming Cao. Some necessary and sufficient conditions for second-order consensus in multi-agent dynamical systems. *Automatica*, 46(6):1089 – 1095, 2010.

[8] Jan Lunze. Synchronizable nodes in networked systems. *Journal of Physics A: Mathematical and Theoretical*, 44(4):045103, 2011.

[9] S. Emre Tuna. Synchronizing linear systems via partial-state coupling. *Automatica*, 44(8):2179 – 2184, 2008.

[10] Luca Scardovi and Rodolphe Sepulchre. Synchronization in networks of identical linear systems. *Automatica*, 45(11):2557 – 2562, 2009.

[11] Zhongkui Li, Zhisheng Duan, Guanrong Chen, and Lin Huang. Consensus of Multiagent Systems and Synchronization of Complex Networks: A Unified Viewpoint. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 57(1):213 –224, 2010.

[12] Jin Heon Seo, Hyungbo Shim, and Juhoon Back. Consensus of high-order linear systems using dynamic output feedback compensator: Low gain approach. *Automatica*, 45(11):2659 – 2664, 2009.

[13] Ji Xiang, Wei Wei, and Yanjun Li. Synchronized output regulation of linear networked systems. *IEEE Trans. Autom. Control*, 54(6):1336 –1341, june 2009.

[14] Hongkeun Kim, Hyungbo Shim, and Jin Heon Seo. Output consensus of heterogeneous uncertain linear multi-agent systems. *IEEE Trans. Autom. Control*, 56(1):200 –206, jan. 2011.

[15] Peter Wieland, Rodolphe Sepulchre, and Frank Allgwer. An internal model principle is necessary and sufficient for linear output synchronization. *Automatica*, 47(5):1068 – 1074, 2011.

[16] A. Pogromsky and H. Nijmeijer. Cooperative oscillatory behavior of mutually coupled dynamical systems. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 48(2):152 –162, feb 2001.

[17] K.D. Listmann and C.A. Woolsey. Output synchronization of systems in chained form. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 3341 –3346, dec. 2009.

[18] Z. Lin, B. Francis, and M. Maggiore. Necessary and sufficient graphical conditions for formation control of unicycles. *IEEE Trans. Autom. Control*, 50(1):121–127, 2005.

[19] Mauricio Barahona and Louis M. Pecora. Synchronization in small-world systems. *Phys. Rev. Lett.*, 89:054101, Jul 2002.

[20] A. E. Motter, C. S. Zhou, and J. Kurths. Enhancing complex-network synchronization. *EPL (Europhysics Letters)*, 69(3):334, 2005.

[21] M. L. J. Hautus. Controllability and observerbility condition of linear autonomous systems. *Ned. Akad. Wetenschappen, Proc. Ser. A*, 72:443–448, 1969.

[22] J. Willems. Least squares stationary optimal control and the algebraic riccati equation. *IEEE Trans. Autom. Control*, 16(6):621 – 634, dec 1971.

[23] J. Huang. *Nonlinear output regulation: theory and applications*. Advances in design and control. Society for Industrial and Applied Mathematics, Philadelphia, 2004.

[24] Zongli Lin. *Low gain feedback*, volume 240 of *Lecture Notes in Control and Information Sciences*. Springer Berlin / Heidelberg, 1999. 10.1007/BFb0119076.

A Light-weight Approach to Online Detection and Classification of Interference in 802.15.4-based Sensor Networks

Frederik Hermans, Olof Rensfelt, Lars-Åke Larzon, Per Gunningberg

IT Department, Uppsala Universitet, Sweden
{frederik.hermans, olofr, lln, perg}@it.uu.se

Abstract. With a rapidly increasing number of devices sharing access to the 2.4 GHz ISM band, interference becomes a serious problem for 802.15.4-based, low-power sensor networks. Consequently, interference mitigation strategies are becoming commonplace. In this paper, we consider the step that precedes interference mitigation: interference detection. We have performed extensive measurements to characterize how different types of interferers affect individual 802.15.4 packets. From these measurements, we define a set of features which we use to train a neural network to classify the source of interference of a corrupted packet. Our approach is sufficiently light-weight for online use in a resource-constrained sensor network. It does not require additional hardware, nor does it use active spectrum sensing or probing packets. Instead, all information about interferers is gathered from inspecting corrupted packets that are received during the sensor network's regular operation. Even without considering a history of earlier packets, our approach reaches a mean classification accuracy of 79.8%, with per interferer accuracies of 64.9% for WiFi, 82.6% for Bluetooth, 72.1% for microwave ovens, and 99.6% for packets that are corrupted due to insufficient signal strength.

1 Introduction

The unlicensed 2.4 GHz ISM band is heavily populated by wireless devices as diverse as WiFi laptops, Bluetooth headsets, baby monitors and microwave ovens [6]. Consequently, sensor networks operating in this frequency band increasingly suffer from cross-technology interference. This is a severe problem because even unsuccessful communication attempts take a toll on the sensor nodes' scarce energy budgets. Mitigation strategies aim to warrant timely and energy-efficient communication even in the presence of interferers. Knowing the type of interference a network is exposed to can increase the effectiveness of mitigation, resulting in higher packet delivery ratio and ultimately lower power consumption [13, 8].

To facilitate useful mitigation decisions, we consider the problem of interference detection and classification in 802.15.4-based sensor networks. This is a challenging task, because 802.15.4 devices generally cannot decode transmissions from interferers using other radio technologies. Existing approaches actively sample the spectrum for activity from interferers [1, 16] or use dedicated hardware such as software-defined radios [12]. However, spectrum sampling is a very energy-consuming task for sensor networks, and adding dedicated hardware increases both price and complexity of sensor nodes.

In this paper, we describe an approach that enables resource-constrained sensor nodes to classify individual corrupted 802.15.4 packets according to the cause of corruption. Using data from our extensive measurements on how different interferers affect 802.15.4 communication, we show that each interferer has characteristic patterns that emerge from observing the (1) Link Quality Indicator (LQI) of an interfered 802.15.4 packet, (2) the signal strength during packet reception, and (3) information about what parts of the packet are corrupted. We define a set of *features* on these three observations that extract the essential information. Our features are sufficiently light-weight so that a sensor node can compute them for a given corrupted packet. A neural network maps features to an interference class, i.e., it allows to determine the type of interferer from the data collected about an individual corrupted packet. We implement a fixed-point neural network on the TelosB platform to demonstrate the feasibility.

A key strength of our approach is its resource efficiency: It does not require active spectrum sensing, additional hardware, or probing packets. Instead, it gathers information about interference only during the regular operation of the sensor network. Assuming that the network uses either forward error correction or retransmissions, our approach does not incur any communication overhead. The main energy cost of our approach comes from turning on the Micro Controller Unit (MCU) during packet reception, whereas usually it would be woken up only when packet reception is completed.

The contributions of this paper are:

- We identify interference patterns that distinguish interference sources, and show how from information that can be gathered during a sensor network's regular operation, we can define features that allow to capture the essence of these patterns.
- The features and the classification method we use are chosen to be sufficiently light-weight so they can be computed on a resource-constrained sensor node.
- We evaluate the classification accuracy of our approach with respect to four different causes of packet loss: interference from WiFi networks, Bluetooth networks, and microwave ovens, as well as packet loss due to insufficient sender TX power. The mean classification accuracy is 79.8%.

While in this paper we focus on three types of interferers (WiFi, Bluetooth, microwave ovens) and packet loss due to insufficient signal strength, our approach is not confined to detecting these. To classify an additional interferer type requires collecting appropriate training data, re-training the classifier, and updating the classification parameters on the sensor nodes. It may also be necessary to define additional features.

The paper is organized as follows: A brief summary of relevant radio technologies is given in Sec. 2. We describe our interference measurements in Sec. 3. Sec. 4 describes the core ideas of our approach: how features can be extracted from data about corrupted packets, and how these features can be used to classify packets according to the cause of corruption; we also present implementation and overhead considerations. In Sec. 5, we evaluate the classification accuracy. Related work is described in Sec. 6 and we present conclusions and an outlook on future work in Sec. 7.

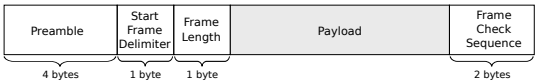


Fig. 1: Format of an 802.15.4 packet

2 Technical Background

We briefly summarize the technical aspects of 802.15.4, 802.11b/g, Bluetooth, and microwave ovens that are relevant to our goal of interference classification.

802.15.4 The 802.15.4 standard defines a physical layer and a MAC layer for low-power, low-rate wireless networks [10]. We consider 802.15.4 at 2.4 GHz in this paper, because most interference is faced in this popular ISM band. At 2.4 GHz, 16 channels of 2 MHz width are defined with an inter-channel spacing of 3 MHz. A maximum transmission power of 0 dBm is common. The standard implements direct sequence spread spectrum by mapping each four-bit symbol to be transmitted to a pseudo-random 32-chip sequence. Offset quadrature phase-shift keying is used for modulation. The data rate is 250 kbps, the symbol period is 16 µs.

The format of an 802.15.4 PHY packet is shown in Fig. 1. Each packet begins with a preamble, which consists of four zero bytes, followed by a one-byte start frame delimiter (SFD) field with a fixed value. The frame length field contains the number of the packet's payload bytes, which may be up to 127. The length includes the two-byte frame check sequence (FCS) field which trails the packet payload. The FCS field contains a checksum which is calculated over the length field and the payload bytes. A receiver synchronizes to incoming zero-bytes; after receiving four zero-bytes, it scans for an SFD. Only after correctly receiving the SFD, it reads the following payload field and then reads the specified number of payload bytes. If no SFD is received after four zero-bytes, the receiver synchronizes to incoming zero-bytes again. A receiver can detect transmission errors by comparing the FCS against the checksum calculated for the received packet.

802.11b/g The amendments b and g describe the two most prevalent physical layer implementations of the 802.11 standard for wireless local area networks [11]. Fourteen channels of 22 MHz width are defined, of which eleven are available worldwide. Nominal transmission power is between 15 dBm and 20 dBm, and bit rates range from 1 MBit/s to 54 MBit/s using a variety of modulations. The MAC layer is a variant of CSMA/CA and mandates the minimum time between two frames.

Bluetooth Bluetooth is a standard for short-range radio communication within personal area networks [9], and is often used for connecting peripherals. Bluetooth divides the 2.4 GHz band into 79 bands of 1 MHz each and employs frequency hopping over these bands. Devices in a network synchronize on a hopping sequence, and stay on each frequency for a slot duration of 625 µs. The most common class of Bluetooth devices has a maximum transmission power of about 4 dBm.

Microwave oven Residential microwave ovens are kitchen appliances that are used to heat food using non-ionizing microwave radiation in the 2.4 GHz band. Usually the whole band is affected by the emissions, with ovens exhibiting a frequency sweeping

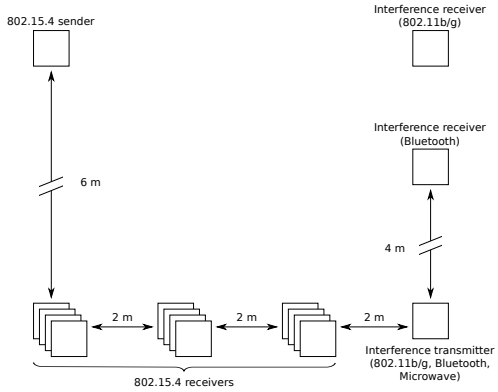


Fig. 2: Experiment layout

Fig. 3: The microwave interferer and four of the sensor nodes (on a tripod) in the anechoic chamber

behavior. Minor variations in the affected frequencies can be observed for different models. The ovens' emissions alternate between on and off phases every $\frac{1}{2f}$ s during the heating period, where f is the mains frequency. The equivalent isotropically radiated power (EIRP) has been reported to range between 16 dBm and 33 dBm [5].

3 Interference Measurements

We conducted a series of experiments in which an 802.15.4-based sensor network was exposed to radio interference. In each experiment, we activated one interference source and collected data on the corrupted 802.15.4 packets. The purpose of these experiments is twofold. First, we collected corrupted packets to gain a better understanding of the effects that different interferers have on individual 802.15.4 packets. Second, we use subsets of the data for training and evaluating our classification approach.

The experiments were carried out in an anechoic chamber, which is shielded from outside radio transmissions. Such a controlled environment gives us high confidence that a corrupted 802.15.4 packet recorded during an experiment was indeed corrupted by the source of interference that we have activated. In a less controlled environment, e.g., our university building, it is virtually impossible to prevent radio devices that are outside of our control from affecting the experiments. The anechoic chamber is also constructed to minimize multipath propagation. This is desirable, because multipath propagation is strongly dependent on the concrete physical layout of an environment, and we do not want to capture environment-specific effects in our measurements.

3.1 Experiment Setup

The experiment setup is shown in Fig. 2. We used TelosB sensor nodes [4], which have a 802.15.4-compliant CC2420 transceiver [20]. One node acted as a sender and twelve nodes were receivers. The receivers were divided into groups of four nodes and receivers within one group were placed close to each other. With this setup, receivers

within one group all have similar distance to the sender and interferer, which allows us to separate effects of distance from actual interference.

We placed different interferers in the anechoic chamber during the experiments. The position of the interfering transmitter, and the receiving counterpart (in case of WiFi and Bluetooth) is also shown in Fig. 2. We describe the interferers and their parameters in more detail in the next subsection.

During each experiment, the 802.15.4 sender periodically broadcasted packets. We added a jitter to the transmission period to avoid accidental synchronization with an interferer. The sender's transmission power was fixed to -15 dBm, unless mentioned otherwise. We confirmed that when no interferer was active, the packet reception rate was close to 100% at a transmission power of -15 dBm. As we want to focus exclusively on the effects of interference, no MAC layer was used in the sensor network, and the sender did not perform carrier sensing prior to sending. We varied the length of the payload of 802.15.4 packets between 16, 32, 64, 96 and 124 bytes, excluding the FCS field. The receiving nodes logged all packets they received, regardless of whether they passed the CRC check or not. Notice that nodes can only log packets for which they can successfully decode the preamble and the SFD field. For data collection and experiment control we use Sensei-UU, our nomadic sensor network testbed [17].

3.2 Interferers

We consider three different technologies that commonly cause interference to 802.15.4-based sensor networks: IEEE 802.11b/g networks, Bluetooth networks, and household microwave ovens. For each interferer, we ran experiments with different models from different vendors to avoid model-specific effects. Two different WiFi chipsets, three different microwave ovens, and three different Bluetooth devices were used.

IEEE 802.11b/g The devices operated in infrastructure mode. The access point (referred to as interference transmitter in Fig. 2) sent constant bitrate UDP traffic saturating the WiFi link to the receiver, which acted as a stationary client. We chose this scenario to resemble a video streaming session. The following parameters were varied between experiment runs: WiFi transmission power (4 dBm to 22 dBm), WiFi rate (1 MBit/s up to 54 MBit/s), 802.15.4 channel. The WiFi channel was fixed to 5, so by varying the 802.15.4 channel we controlled the frequency distance between WiFi and 802.15.4.

Bluetooth Bluetooth traffic was created by one Bluetooth dongle (interference transmitter) sending back-to-back L2CAP packets to another Bluetooth dongle (interference receiver). Since Bluetooth performs adaptive frequency hopping, the dongles were reset at the beginning of each experiment, such that the dongles were oblivious with regard to the frequency on which the 802.15.4 network would be active. Between experiment runs, we varied the 802.15.4 channel and 802.15.4 transmission power.

Microwave oven Using a spectrum analyzer, we confirmed that the ovens were mostly active around 802.15.4 channels 18 and 23. A bowl of water was heated in the microwave during each experiment run. The effect was set to maximum power in all runs, because changing power only resulted in varying length of idle periods between the oven's heating periods. Between experiment runs, we varied the 802.15.4 channel.

Insufficient signal strength We also ran experiments in which the 802.15.4 sender sent packets with a transmission power of -25 dBm without any interferer being active. At this transmission power, the 802.15.4 receivers experienced significant packet loss due to an insufficient SNR. This experiment serves as a reference case for packet loss that was not caused by interference.

4 Interference Detection and Classification

We describe a classifier that assigns each incorrectly received packet to an interference class. Each interference class represents a source of interference: WiFi, Bluetooth, or microwave. We further define an additional class to represent packets that have been received incorrectly in the absence of an interference source due to insufficient signal strength at the receiver.

In this section, we first consider what data can be feasibly gathered in a sensor network for such a classification task; we then consider how this data can be condensed into numerical features. We discuss suitable classification algorithms, and finally consider implementation, energy cost and overhead.

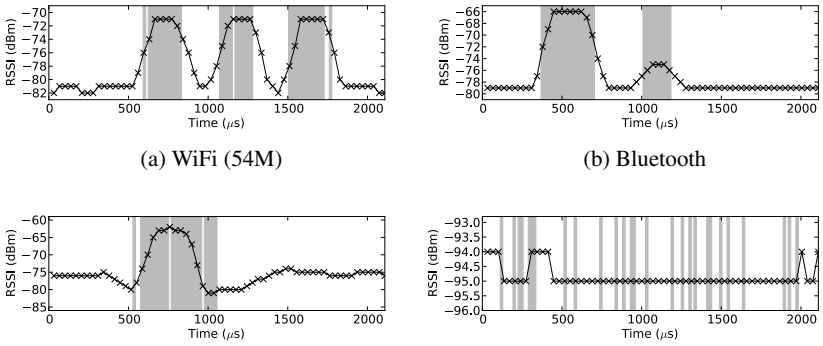
4.1 Extracting Information about Interference from Corrupted 802.15.4 Packets

As described in Sec. 2, an 802.15.4 sensor node can detect transmission errors by matching the payload against the CRC at the end of the packet. We consider what information a node could extract from a corrupted packet to help the goal of determining the cause of packet corruption. Only information that can be gathered with moderate energy costs and without additional hardware is regarded.

LQI The 802.15.4 standard requires transceivers to provide a Link Quality Indication (LQI) measurement for each received packet. Transceivers such as the CC2420 determine LQI in terms of the chip error rate for the first eight symbols following the SFD field. Thus, LQI describes how well the first eight received chip sequences match any of the known chip sequences. LQI is popularly used by link quality estimators. Obtaining the LQI for a received packet incurs no additional energy cost.

RSSI during packet reception The standard also requires transceivers to be capable of providing an estimate of the received signal power within the bandwidth of the currently selected channel, called Received Signal Strength Indication (RSSI). The CC2420 measures RSSI over a 128 µs window (8 symbol periods). For each packet, it provides an RSSI measured over the packet's first eight symbols by default. This RSSI measurement is also commonly used in link quality estimators as a metric of the channel's quality. Further RSSI measurements can be requested from the transceiver explicitly.

Rather than considering only one RSSI measurement taken at the beginning of reception, a more fine-grained view of how signal strength changes while a packet is being received potentially allows us to learn about the presence of high-power interferers. We therefore modify the CC2420 radio driver in the Contiki OS source code to continuously measure RSSI while a packet is being received. Our modified driver produces a series of RSSI values for each received packet, containing about one sample per payload byte. Examples of RSSI during packet reception are shown in Fig. 4.



(a) WiFi (54M)

(b) Bluetooth

(c) Microwave

(d) Insufficient signal strength

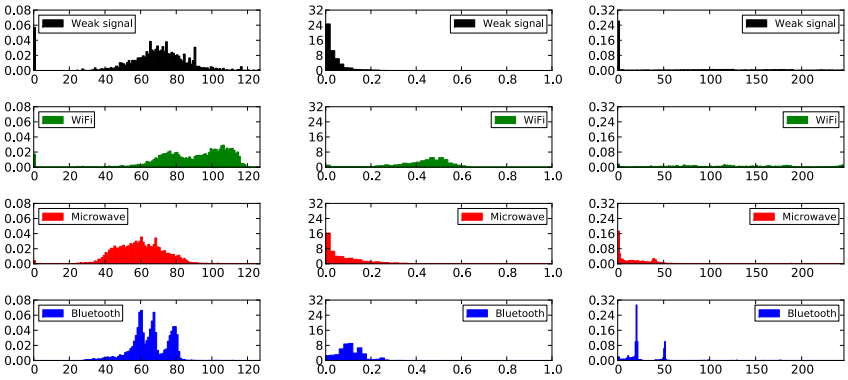
Fig. 4: Corruption of some 802.15.4 packets in the presence of different interferers. Dark gray areas show which parts of a packet have been corrupted. The RSSI during packet reception is also shown. Note that the Y axis' range differs between subfigures.

By measuring the signal strength only during packet reception, we ensure that we exclusively capture changes in channel conditions that could have an impact on the sensor network communication. In contrast, if we sampled the channel while there is no ongoing sensor network communication, we may capture emissions from a non-interfering radio devices, e.g., devices that back off during ongoing 802.15.4 transmission.

Incorrectly received symbols We assume that a sensor node can determine what parts of a received packet are corrupted, i.e., which symbols of the packet's payload have been decoded incorrectly. This assumption can be warranted if either forward error correction (FEC) or a retransmission protocol is used. If the sensor network applies FEC and a sensor node can successfully reconstruct a damaged packet, it can immediately determine which parts of the packet were corrupted. If no FEC is used, a sensor node may still be able to determine the corrupted parts of a packet if a retransmission protocol is used. In that case, the sensor node can keep recently received corrupted packets in a buffer, and probabilistically match successfully received packets against the corrupted packets in the history. In this way, it is possible for the node to identify which parts of the packets stored in the history are damaged. Specifying a method to perform this matching is outside of the scope of this paper and we leave it to future work, but from our experimental data, we are confident of the feasibility of such an approach.

4.2 Features

The data we can gather for each corrupted packet is not suitable as direct input to a classification algorithm. Not only is it too large for a classifier run on a resource-constrained sensor node, but the RSSI values are also dependent on the concrete transmission powers and distances of the given scenario. Thus, we define some features that represent the data more concisely and that abstract from transmission power and distance. These features will then be used as inputs to the classifier.



(a) Standard deviation of normalized RSSI

(b) Normalized number of corrupted symbols

(c) Bursts spanned

Fig. 5: Histograms of selected features of corrupted 802.15.4 packets. One histogram is shown for each group of packets corrupted by one type of interferer.

Our approach to feature selection is empirical; we aim to capture patterns that we observe in corrupted packets that we collected in our experiments. Note that the packets shown in Fig. 4 are exemplary, and we display them here to explain the reasoning behind our features—in practice, not all interfered packets show such well-defined patterns. (Readers may skip forward to Fig. 8 for an impression of hard-to-classify packets.)

None of the following features by itself will give a perfect distinction between interferers. Rather, it is the combination of multiple features that facilitates classification. Furthermore, we have evaluated a number of features in the course of this work that turned out to be of little use in classification. For the sake of brevity, we omit their description and concentrate on those features that we use in the evaluation in Sec. 5.

LQI threshold We define a binary feature that indicates whether the *LQI of a corrupted packet is higher than 90*. As described previously, LQI can be considered to represent the chip error rate over the first two bytes of a packet. If a packet is received with a high LQI, but the packet fails the CRC check, we take this to as an indicator that channel conditions where good when reception started, but then deteriorated. Such a sudden change in channel conditions can be observed when an interferer starts emission during packet reception. In contrast, packets that cannot be decoded correctly due to insufficient signal strength usually have a low LQI value, because the channel conditions are poor over the whole packet reception time. The threshold value 90 was chosen empirically from our measurement data.

RSSI-related features We define a binary feature to indicate whether the *range of RSSI values is greater than 2 dB*. For packets that are corrupted due to insufficient signal strength, the RSSI values often contain little variation, whereas distinct peaks in signal strength can usually be seen for interfered packets.

Before considering further RSSI-related features, we assume that the series of RSSI measurements is normalized by the maximum value in the series. As argued before, absolute RSSI measurements are dependent on distances and transmission powers of sensor nodes and interferers. Normalization allows us to abstract from the concrete distances and transmission powers, while preserving the series' shape. Before normalizing, we smooth the series using a moving average to reduce high-frequency noise which is due to the CC2420's internal quantization of the signal strength. We found a window size of eight to give reasonable filtering while preserving the shapes of interest.

We define the features *mean normalized RSSI* to be the average value of the smoothed, normalized RSSI readings, and similarly, we use the *standard deviation of normalized RSSI*. Our reasoning is that the RSSI series of packets corrupted by a given interferer often have similar, distinct shapes, and these features are a simple means of characterizing the shape of a series of RSSI values. Though obviously two different shapes can have the same mean and standard deviation, we observe distinct distributions for the mean and standard deviation of RSSI for different interferers. This is demonstrated in Fig. 5a, which shows the distribution of standard deviation for different interferers.

Finally, we define a feature in terms of the *difference between the maximum normalized RSSI value and the most common RSSI value*. This feature captures a pattern that can be seen in Fig. 4c and that we commonly observe in packets interfered by microwave ovens: the signal strength slightly drops, then peaks, and drops again. For such packets, this difference is smaller than the difference for packets without this pattern.

Number of corrupted symbols Knowing the true payload of a packet, we can count the number of symbols that have been decoded incorrectly. We normalize this number by the total number of symbols in the payload. This feature helps to distinguish different types of interferers. Radio technologies such as Bluetooth or 802.11g specify bit rates, minimum and maximum packet lengths, as well as inter-packet delays for medium access control. These specifications put a constraint on the amount of damage that can be done to a 802.15.4 packet. This can be seen in Fig. 5b, which shows the histogram of corrupted symbols per packet for different interferers. The distributions of WiFi and Bluetooth have distinct means, whereas the distributions for microwave-interfered packets and packets with insufficient signal strength are more similar.

Error bursts Comparing the received payload to the correct payload, we can define *error bursts* for each corrupted packet. An error burst is a sequence of corrupted symbols that may contain subsequences of at most four consecutive correct symbols. Allowing error bursts to contain correct symbols allows us to detect contiguous areas of interference. For example, we consider the packet in Fig. 4a to contain three error bursts, and the packet in Fig. 4c to contain one error burst.

For WiFi, we often observe regular bursts of specific lengths, exemplified Fig. 4a. We thus define the feature *mean burst length* for each packet. Bluetooth-interfered packets often contain two error bursts of specific length, whereas the pattern for packets that are corrupted due to insufficient signal strength is much more random. To capture these variations, we define the feature *standard deviation of burst length*. For Bluetooth, we often observe that the time between the start of the first burst and the end of the second burst roughly matches the Bluetooth slot length. Hence we define the feature *bursts*

LQI-based	$\mathrm{LQI} < 90$
RSSI series-based	$\max(\mathrm{RSSI}) - \min(\mathrm{RSSI}) > 2\mathrm{dBm}$
	$\mathrm{avg}(\mathrm{RSSI}_{\mathrm{normed}})$
	$\mathrm{stddev}(\mathrm{RSSI}_{\mathrm{normed}})$
	$\max(\mathrm{RSSI}_{\mathrm{normed}}) - \mathrm{mode}(\mathrm{RSSI}_{\mathrm{normed}})$
Error burst-based	Number of corrupted symbols/payload length
	avg(burst lengths)
	stddev(burst lengths)
	End of last burst − start of first burst
	avg(burst spacings)

Table 1: Overview of features used for classification

spanned, which counts the number of symbols between the first burst's start and the last burst's end. The distribution of this feature is shown in Fig. 5c.

The MAC protocols of WiFi and Bluetooth impose constraints on the time between two packets. Thus, considering the spacing between two error bursts, i.e., the number of symbols correctly received between two error bursts, can give away useful information about the interferer. We define the feature *burst spacing mean* to capture these variations. If a packet contains only one error burst, this feature is undefined. An overview of all features can be found in Tab. 1

4.3 Classification Algorithms

We use a supervised learning approach to train a classifier to assign each corrupted packet to a class representing either WiFi, Bluetooth or microwave interference, or corruption due to insufficient signal strength. In supervised learning, a classifier is trained on a set of examples for which the correct class (i.e., the interference source in our case) is given. A corrupted packet is represented by the features described in the previous section. The learning phase, which is computationally more costly than classifying individual packets after training is completed, is carried out on a regular PC, whereas the actual classification is performed online in the sensor network. We consider two different classification algorithms: Support Vector Machines (SVMs, [3]) and feed-forward neural networks [18].

An SVM transforms the feature vectors to a high-dimensional space and, during the learning phase, constructs (potentially non-linear) hyperplanes to separate the vectors from the learning set according to their classes. In the classification phase, the SVM determines the class of the input vector by considering on which side of the hyperplanes the vector lies. Unfortunately, it turned out that online classification in a sensor network with SVMs is not feasible in our case due to the limited amount of RAM available on the sensor nodes. We nevertheless present results for the SVM classification in the evaluation as a reference case, since they are often considered the best "out-of-the-box" classification algorithm [15]. Furthermore, SVMs find a global, unique solution, whereas neural networks may get stuck in local minima during the learning phase.

Feed-forward neural networks are a class of classification algorithms inspired by biological neural networks. They are represented by directed acyclic graphs, where each

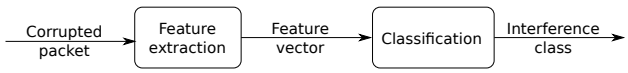


Fig. 6: Steps in classification on a sensor node

node represents a computational unit and weighted edges describe input/output relationships between the nodes. Each node represents either an input value (i.e., a component of a feature vector), an output value, or an activation function, which usually is a sigmoid function that takes as input the weighted outputs from incoming edges. During the learning phase, an optimal set of edge weights is found. In the classification phase, an input feature vector is propagated through the network and the output nodes indicate the classification result. After the costly learning phase, a neural network can be represented by a matrix and the classification cost is dominated by matrix multiplications. Thus, with careful implementation, the use of a feed-forward neural network is feasible even on a resource-constrained platform like TelosB.

4.4 Implementation and Overhead Considerations

We briefly consider some implementation and overhead aspects. Our approach requires two steps shown in Fig. 6: extraction of features from the data gathered for a corrupted packet and evaluation of the feed-forward neural network.

Measuring RSSI during packet reception incurs an additional energy cost, because the sensor node's MCU needs to be awake while the packet is being received. This cost is proportional to the length of the received packet. But since power consumption is usually dominated by radio operation, we believe the additional cost from the MCU being awake to be tolerable. The buffer for RSSI samples is 128 bytes of RAM.

The cost of determining the damaged parts of a packet depends on whether FEC or retransmissions are used. With FEC, there is no additional overhead. If retransmissions are used, RAM needs to be allocated to store a history of recent packets. The size of the history depends on the retransmission protocol and the network's traffic pattern, but we estimate the history to be on the order ten packets, corresponding to 1270 bytes of RAM, given the maximum payload length of 127 bytes imposed by 802.15.4.

Most operations that are necessary for feature extraction are trivial to perform, such as additions or finding the maximum value in a buffer. We note the following regarding smoothing, calculating mean, standard deviation and the most common value: smoothing can be efficiently implemented using an exponential weighted moving average. Calculating the mean and standard deviation of a series each require one iteration over the series of values. Finally, the most common value of a series can be easily found by sorting the series and then iterating over it. Since our series are small (< 128 bytes), the overhead is moderate.

The extensive use of floating point operations in the evaluation of a neural network poses a challenge on constrained platforms that do not support floating point operations. Thus, to make online classification feasible, we have implemented a fixed point neural network that operates on 32-bit integer values and does not require any floating point operations. The C implementation takes approximately 60 lines of code. The memory consumption is less than 1 kilobyte of RAM to represent the weights of the neural

network and less than 100 bytes of RAM for temporary variables. As described in the previous section, the weights of the neural network are determined during the neural network's learning phase, which is performed on a PC.

5 Evaluation

We now investigate classification accuracy to assess how well our features are capable to capture interference patterns, and whether the fixed point implementation of the neural network performs significantly worse than a floating point neural network or SVM.

5.1 Datasets

We constructed two datasets of interfered packets from the measurements described in Sec. 3: a training set for training the classifiers and a testing set for evaluation. The sets are balanced, i.e., each set contains an equal amount of corrupted packets for each interference class and an equal amount of packets corrupted due to insufficient signal strength. The sets are also balanced regarding the set of parameters for each interferer. E.g., each set has an equal amount of packets interfered by the various WiFi rates. For the interfered packets (as opposed to packets with low signal strength), we exclude all such packets from our sets that were received incorrectly by only one of the twelve receivers, since such packets may be corrupted due to transient errors in that receiver rather than due to interference.

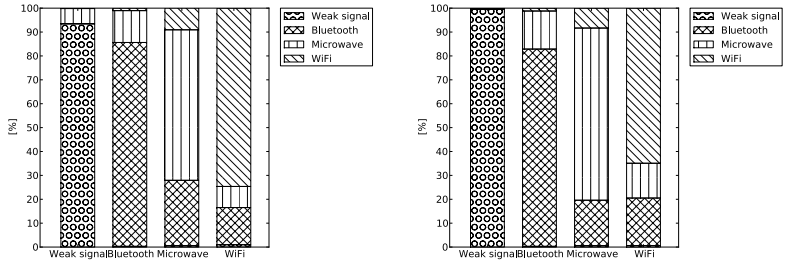
For each interference class, we used devices from different vendors for the testing set and the training set. E.g., the training set contains WiFi packets corrupted by devices using an Atheros AR9100 chipset, whereas the testing set contains WiFi packets corrupted by different devices using a Broadcom BCM5354 chipset. This is to ensure that the features we train on are sufficiently general and not model-specific. Furthermore, it also ensures that the training sets and testing sets are disjoint.

The training set consists of packets of 64 and 96 bytes length. To ensure that our features are not specific to packet length, the testing set additionally includes packets of 124 bytes. We exclude shorter packets as they have shown to carry insufficient information for meaningful classification.

5.2 Classification Accuracy

To evaluate how well the features we have selected can distinguish between different interferers and to establish a comparison case for the implementation for resource-constrained devices, we have trained an SVM, a fixed point neural network, and a floating point neural network to classify interferers. The result from the SVM serves as a base case for the neural networks' performance.

Support vector machine The classification accuracy for the SVM is shown in Fig. 7a. The mean classification accuracy is 79.1%. Packets corrupted by insufficient signal strength are correctly classified to 93.4%. Most of the misclassified weak-signal packets are attributed to microwave interference. The accuracy for Bluetooth and microwave



(a) Classification accuracy for a support vector machine

(b) Classification accuracy for a fixed point neural network

Fig. 7: Classification accuracy

interference is 85.3% and 63.0% respectively. Packets interfered by Bluetooth and microwave share some similarity in the corruption patterns, which shows in the fact that around 27% of microwave interference is incorrectly classified as Bluetooth interference. Regarding WiFi interference, 74.6% of the corrupted packets are correctly classified. Again, WiFi is most likely to be misclassified as Bluetooth interference (15%).

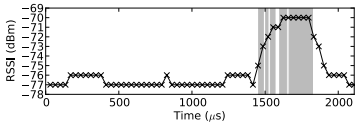
Fixed point neural network implementation The classification accuracy for the fixed point neural network is shown in Fig. 7b. The mean classification accuracy is 79.6%. The classification of non-interfered packets is above 99%, which is significantly better than the SVM. This is an important figure, since mistaking corrupted packets due to insufficient signal strength for actual interference may trigger costly interference mitigation strategies. The fixed point neural network correctly classifies Bluetooth interference in 82.6% of the cases—marginally worse than the SVM. It performs considerably better at classifying microwave interference (72.1% accuracy), but worse at WiFi (64.9%). WiFi is most commonly misclassified as Bluetooth (20.0%) or microwave interference (14.6%).

In total, the performance of the fixed point neural network is similar to the SVM's performance. This is an indication of its feasibility for the classification task at hand, despite the fact that it may theoretically reach non-optimal accuracy due to local minima in the error function.

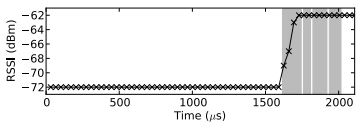
Floating point neural network A more resource-demanding floating point neural network implementation performed only marginally better than our fixed point implementation. Just 0.2% of the packets have been classified differently by the floating point neural network compared to the fixed point neural network. Thus, the fixed point implementation does not suffer from a worse accuracy.

5.3 Misclassified Packets

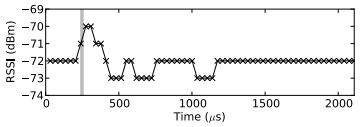
By inspecting the misclassified packet we can understand the shortcomings of our approach. In many cases microwave interference is being classified as Bluetooth interfer-



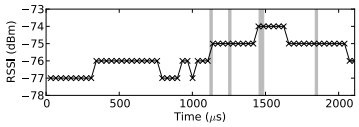
(a) Microwave interference misclassified as Bluetooth interference



(b) WiFi (6M) interference misclassified as microwave interference



(c) Bluetooth interference misclassified as microwave interference



(d) WiFi (5.5M) interference misclassified as microwave interference

Fig. 8: Examples of misclassified packets

ence and vice versa. From manual inspection, we realize that our features do not suffice to distinguish some of them; for an example, see Fig. 8a.

Another set of misclassified packets is due to the fact that in our approach, data is only collected during packet reception. If the transmission of an 802.15.4 packet is only partly interfered, we may have insufficient data to correctly classify the packets. An example of an overlapping transmission is shown in Fig. 8b, in which a 802.15.4 packet is overlapped by a WiFi transmission towards the end.

We also observe that some of the interfered packets have only little variation in signal strength and only a few symbols of the payload are corrupted (Fig. 8d, 8c). These packets cannot be classified with our approach, as too little information is available. We did not identify certain sets of parameters (e.g., WiFi bit rate and 802.15.4 channel) for which classification was significantly worse than for other parameter values.

As mentioned earlier, we exclude packets of 16 and 32 byte length from our evaluation. An early evaluation had shown the classification accuracy for packets of 16 bytes and 32 bytes length to be very low (50% and 58% respectively). This is not surprising, as for such packets we get only little data to extract features from.

6 Related Work

Relevant research in the WiFi domain includes Airshark [16], a system that uses standard 802.11 cards to sample the spectrum. The sampled data is analyzed using cyclostationary process methods to detect transmission patterns, which are then used to classify interferers. Another example is RFdump [12], which uses a software-defined radio to detect which devices are accessing the medium. RFdump aims to provide a tcpdump-like tool for the wireless communication. Gollakota et al. describe how antenna diversity in 802.11n can be exploited to reconstruct interfered signals [6]. All these approaches have in common that they require advanced signal processing capabilities that are usually not available in sensor networks. Cisco has developed a spectrum analyzer for network analysis that is capable of classifying radio devices [2].

In sensor networks, interference detection can help mitigation, which in turn increases the network lifetime by reducing unsuccessful communication attempts. Chowdhury et al. describe an approach to interference classification by actively scanning channels for characteristic spectrum usage [1]. In contrast to our work, this approach comes at a higher energy cost, because the radio needs to be turned on even when no sensor network communication is ongoing. Their work is also concerned with interference mitigation. Hauer et al. describe how detection of WiFi interference can be used for interference mitigation [8]. Similar to our approach, they also consider RSSI during packet reception for identifying interference. They do not take locations of corrupted symbols into account; rather than that, their work is concerned with selectively retransmitting parts of a packet that are suspected to be interfered, without having certain knowledge about the corruption.

A large body of works considers the effect of interference on sensor networks in terms of high-level metrics such as packet reception rate. We selectively refer to [7, 14, 19] for an overview.

7 Conclusion and Future Work

In this paper, we have investigated the feasibility of a light-weight interference detection and classification approach that only uses data that can be gathered during a sensor network's regular operation. The reason to only use such data is to keep energy consumption as low as possible. We described a set of features that allows us to define the characteristic patterns that we observe for different sources of interference. We also demonstrated that a fixed-point neural network reaches a mean classification accuracy of 79.8% for packets of 64 bytes and more.

For this paper, we have gathered training and testing data in an anechoic chamber, which is a highly controlled environment. While we introduce a certain variance into our data by using different hardware models for the interferers, we would expect a less controlled radio environment to contribute further variance due to multipath propagation. Thus, we also plan to perform experiments in the offices of our university to test the robustness of our features. However, we face the practical issue of establishing ground truth in such an environment—we could not with certainty say that a packet is interfered by the interferer that we have activated, because an RF device outside our control (e.g., the university WiFi) may also affect our experiments.

Interference detection and classification is an important tool for debugging network problems and mitigation strategies, but it is not an end in itself. Therefore, we plan to integrate our approach into an existing interference mitigation strategy. If our approach helps to make significantly better mitigation decisions in an uncontrolled environment, we may avoid the aforementioned issue of establishing ground truth for measuring the performance of our classification approach.

The aim of this work was to assess the feasibility of interference classification with the limited information that can be gathered from corrupted packets. We have focused on classifying the source of interference for individual packets. An interesting track of future work we plan to follow is to incorporate information about previously received, corrupted packets into the classification process. We believe this may yield a significant

increase in accuracy. However, incorporating previous information requires careful consideration, especially regarding situations in which multiple interferers are present, or in which the source of interference rapidly changes due to high mobility.

References

1. Chowdhury, K., Akyildiz, I.: Interferer classification, channel selection and transmission adaptation for wireless sensor networks. In: Proc. of ICC '09. pp. 1 –5 (Jun 2009)
2. Cisco Inc.: Cisco Spectrum Expert Wi-Fi (2012), http://www.cisco.com/en/US/products/ps9393/index.html
3. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20, 273–297 (1995)
4. Crossbow Inc.: TelosB datasheet. http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf
5. Gawthrop, P.E., Sanders, F.H., Nebbia, K.B., Sell, J.J.: Radio spectrum measurements of individual microwave ovens – volume 1 & 2, NTIA Report TR-94-303-1, NTIA Report TR-94-303-2
6. Gollakota, S., Adib, F., Katabi, D., Seshan, S.: Clearing the RF smog: making 802.11n robust to cross-technology interference. In: Proc. of SIGCOMM '11. pp. 170–181 (2011)
7. Hauer, J.H., Handziski, V., Wolisz, A.: Experimental Study of the Impact of WLAN Interference on IEEE 802.15.4 Body Area Networks. In: Procs. of EWSN '09 (2009)
8. Hauer, J.H., Willig, A., Wolisz, A.: Mitigating the Effects of RF Interference through RSSI-Based Error Recovery. In: Wireless Sensor Networks, Lecture Notes in Computer Science, vol. 5970, pp. 224–239. Springer (2010)
9. IEEE Computer Society: Local and metropolitan area networks, specific requirements, part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs) (2005)
10. IEEE Computer Society: 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) (2006)
11. IEEE Computer Society: Local and metropolitan area networks, specific requirements, part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (2007)
12. Lakshminarayanan, K., Sapra, S., Seshan, S., Steenkiste, P.: RFDump: An Architecture for Monitoring the Wireless Ether. In: Procs. of CoNEXT '09 (Dec 2009)
13. Liang, C.J.M., Priyantha, N.B., Liu, J., Terzis, A.: Surviving Wi-Fi Interference in Low Power ZigBee Networks. In: Procs of SenSys '10. pp. 309–322 (2010)
14. Musaloiu-E., R., Terzis, A.: Minimising the effect of WiFi interference in 802.15.4 wireless sensor networks. Int. Journal of Sensor Networks 3, 43–54 (2008)
15. Ng, A.: Support Vector Machines – CS229 Lecture Notes (2011), http://cs229.stanford.edu/notes/cs229-notes3.pdf
16. Rayanchu, S., Patro, A., Banerjee, S.: Airshark: detecting non-wifi rf devices using commodity wifi hardware. In: Procs. of IMC '11. pp. 137–154 (2011)
17. Rensfelt, O., Hermans, F., Larzon, L.A., Gunningberg, P.: Sensei-UU: a relocatable sensor network testbed. In: Procs. of WiNTECH '10. pp. 63–70 (September 2010)
18. Rojas, R.: Theorie der neuronalen Netze. Springer (1993)
19. Sikora, A.: Compatiblity of IEEE 802.15.4 (ZigBee) with IEEE 802.11 (WLAN), Bluetooth and Microwave Ovens in 2.4 GHz ISM-Band. Tech. rep., University of Cooperative Education Loerrach (2004)
20. Texas Instruments Inc.: CC2420 - 2.4 GHz IEEE 802.15.4, ZigBee-ready RF Transceiver. http://www.ti.com/lit/gpn/cc2420

Unstable Path Routing in Urban-Scale WSN

Farrukh Mirza, Mélanie Bouroche, and Vinny Cahill

Distributed Systems Group, School of Computer Science and Statistics, Trinity College Dublin, Ireland,
{mirzaf,Melanie.Bouroche,Vinny.Cahill}@scss.tcd.ie

Abstract. Wireless sensor networks (WSNs) may suffer from congestion at the nodes near the sink, and partition due to the failure of crucial nodes. In urban environments, mobile devices, such as vehicles and smart phones, present in the vicinity of the sensor field could be opportunistically used for data forwarding. Such devices, controlled by third parties, introduce paths that may appear for only very small intervals. This paper discusses how exploiting such unstable paths to shift the routing-related processing and communication load to more capable mobile devices can alleviate traffic congestion, improve fault tolerance and reduce WSN energy consumption.

Keywords: Unstable Path Routing, Urban Scale WSN, Data forwarding via participatory devices, WSN

1 Introduction

In large scale WSN, a number of challenges are faced, such as energy conservation, traffic congestion, fault tolerance, data delivery ratio, timeliness and scalability, among others. In an urban environment, individually-owned mobile devices that may have overall better resources may be present in the vicinity of a WSN. These mobile devices have the potential to forward data on behalf of the WSN. This research contributes to the routing domain of sensor networking by exploring the possibility of taking advantage of such mobile devices, for data forwarding towards the destination in an opportunistic and dynamic manner (Figure 1). In doing so, the data forwarding (routing) load can be shifted away from resource constrained sensor nodes towards more capable mobile devices. Network congestion in particular areas may also be mitigated due to the availability of more capable mobile devices for data forwarding. This strategy may improve fault tolerance by bridging potential network partitions. In this way, energy consumption and processing load due to packet forwarding requests, at intermediate sensor nodes, can also be reduced. Third-party mobile devices could also potentially reduce end-to-end message latency because of the availability of higher transmission rates and better radios, resulting in over-all better timeliness. Finally such mobile devices may have more computational, storage and communication resources, achieving overall better throughput.

The paths introduced by these mobile devices, however, are unstable, i.e., they appear and disappear with the movement of mobile devices, or when a

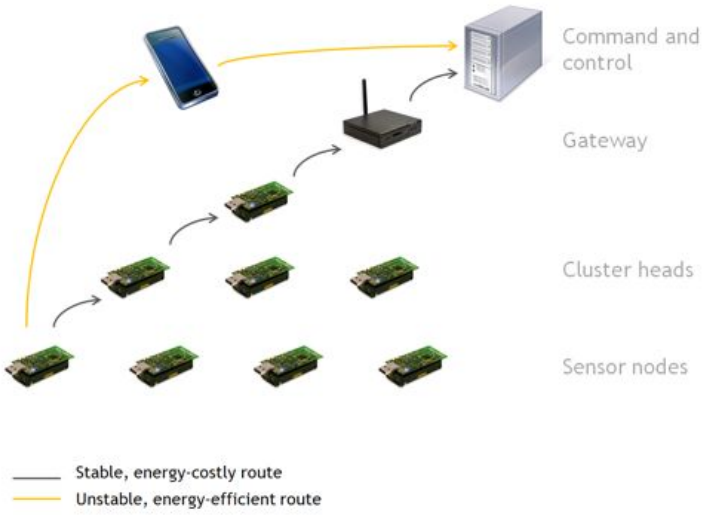


Fig. 1. Exploiting unstable paths via mobile device.

user turns on or off the radio responsible for communicating with the WSN. The instability of these paths may increase the risk of data loss. A trade-off may exist between energy conservation, end-to-end delay reduction, throughput, congestion avoidance and risk of data loss. In addition, completely reliable route performance history may not be available, and acquisition of route information with absolute certainty will increase the control overhead. Therefore, devices will need to make decisions based on very limited information available locally, while still meeting application requirements.

A device differentiation mechanism is also required to select the next hop according to its ability to serve the network. In an urban environment, different types of available devices can be classified according to their generic capabilities such as battery, radio, mobility etc. Stationary sensor nodes have limited and non-rechargeable batteries and can communicate only over short-range radio, e.g., IEEE 802.15.4 [1]. Mobile devices (laptops and smartphones) have limited but rechargeable batteries, can communicate over short-range as well as long-range radio, e.g., IEEE 802.16 [2] and show both walking and vehicular speeds (user in a car). Vehicles and stationary line-powered devices can have unlimited energy reserves. Vehicles may also exhibit varying speeds. By connecting a compatible radio, mobiles and vehicles can communicate directly with WSN. In addition, the devices can also be differentiated according to the roles that they perform. A sink requests data from the WSN, a gateway connects a WSN with some other network such as Internet, and a relay forwards data within the same network. A device may also change its role or may acquire additional role(s), e.g.,

if a user requests data from the WSN, a mobile relay may suddenly also become a sink. Therefore, this mechanism should support a vast variety of devices that dynamically change their roles and characteristics.

This research is closely related and complements EMMON [3], having similar network architecture and relying on COTS [1] components. The EMMON project shows promising results in achieving scalable and time-bounded communication. This research is expected to further enhance those results, by bypassing a number of hops when the network depth increases, which otherwise results in a linear increase of end-to-end latency (Figure 1), and by potentially reducing congestion at nodes that have many children. In this way, we expect to further improve scalability, energy conservation and end-to-end latency.

The essential problem to be solved to successfully exploit unstable paths is, given the mobility patterns of sink, relay and gateway, how can the best next hop be selected in a WSN in order to move data forwarding load to more capable devices and in doing so obtain the right trade off in terms of energy consumption, latency and risk of data loss.

The remainder of the paper is organised as follows. Section 2 gives an overview of the related work, Section 3 identifies and discusses challenges that need to be addressed to design a routing protocol that can efficiently take advantage of unstable paths and finally Section 4 concludes this paper.

2 Related Work

The majority of the protocols that consider the interaction of stationary WSN with mobile devices, only support delivering data from the WSN to mobile sinks. These protocols hide movement of the mobile sink from the WSN by employing a sensor node as proxy for the mobile sink, which is typically the nearest node to the mobile sink. These protocols first establish routes towards the initial proxy node, so that the data flows towards the proxy node. The proxy node then forwards the data to the mobile sink. TTDD, SEAD, EARM and IAR [4–7] extend existing routes from the proxy node towards the mobile sink via other sensor nodes, as the mobile sink moves away. The existing routes are extended unless new routes are absolutely necessary and/or old routes become significantly inefficient. Kusy et al. [8] take a different approach by continuously changing the proxy node with the movement of the mobile sink, by predicting the movement pattern of mobile sinks and identifying the future proxy nodes before they are needed. All these protocols only aim to deliver data to the mobile sinks but do not shift routing load away from sensor nodes.

ROME [9] supports mobile devices as relays, by creating logical gradients converging towards a stationary sink. ROME, however, only forwards data via mobile devices when they are not in motion, hence using them only a fraction of the time.

The closest related routing protocols to this work are those that differentiate among devices and favour more capable devices. DEAR [10] differentiates among

[1] Commercially available Off-The-Shelf

devices based on their energy reserves by classifying them into only two categories, i.e., battery powered and line powered. As the more capable devices are line-powered, therefore they are not mobile. The MC-Routing protocol [11] also classifies devices into only two categories. However, the classification is based on transmission range, data rate and processing capabilities. It creates a logical cellular structure and assumes at least one backbone (more capable) node in each cell. The backbone nodes are also assumed to communicate directly with other backbone nodes in adjacent cells. Similarly, hierarchical routing protocol such as HOLSR [12] and HCB [13] can also differentiate among devices, however, based solely on their number of radios. In hierarchical protocols, the traffic is routed towards more capable devices when the destination is outside the cluster.

These protocol rely on route discovery procedures, mostly initiated by the source node that may be resource constrained. In highly dynamic environments, due to the appearance and disappearance of devices, the route discovery procedures will be initiated frequently, increasing the control overhead. In addition the prospective route may break before any data has reached its destination, rendering the approach un-usable.

3 Design Challenges

Solving the above mentioned problem (Section 1) essentially requires answering the question of how can the forwarding capability of a newly-arrived device be judged and compared to that of existing neighbours, in a highly dynamic environment. Keeping in mind the unstable nature of the prospective routes, the protocol cannot rely on explicit route discovery procedures (Section 2). Similarly, pure geographic routing approach may not be viable, because more capable devices may not be the closest ones to the destination. Hence, the routing protocol should be able to react fast to the changing environment, as well as being able to identify and favour more capable devices.

The scale of the problem considered here, as well as the number of parameters in consideration mean that it is intractable to create a model of the environment. Indeed, the effect of selecting a particular next hop cannot be fully predicted as all operating conditions cannot be predefined and hence the correct behaviour cannot be decided at the design time. Reinforcement learning (RL) [14] is considered particularly suitable for such problems [15], and we are therefore exploring how it could be used in this context. However, it is not completely apparent if RL approach can react fast enough to cater the requirements of such a highly dynamic environment. In addition, set of actions may always be changing in contradiction to typical RL problems. The set of actions can be considered as set of individual devices acting as data forwarders. Hence, the solution is inspired by RL, however, some customization is required in accordance with the environment. RL has been previously used in the packet routing domain [16–18], however some notion of reaching a stable state is assumed, contrary to this research where the stable state remains elusive.

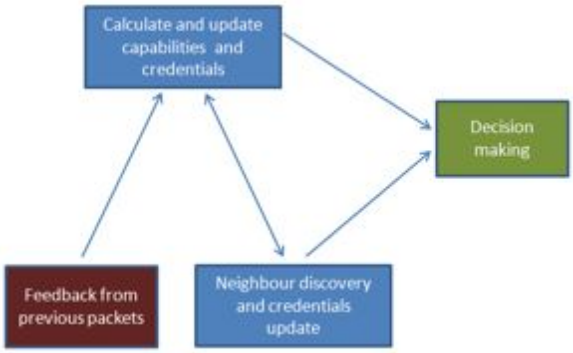


Fig. 2. Solution Components.

The problem is decomposed into four major components (Figure 2) and formulated as a RL problem. The arrows in Figure 2 show dependencies among different components. Each of the components are discussed below along with some of the design challenges that they incur.

3.1 Decision Making

This component is concerned with the next hop selection, enabling competition among potential forwarders, corresponding to action selection in RL. Depending on the action selection policy, this component may occasionally allow the current-hop (node holding the packet) to explore sub-optimal actions [14], allowing better routing opportunities to be explored in a dynamic environment, however, at the expense of overhead. The placement of this component is crucial as it dictates the design of the solution. This component can either be placed at newly-arrived devices that wish to take over the packet and become the next hop, or at the current hop that wish to find a potential next hop.

In the former case, the packet will be voluntarily taken over and forwarded towards the destination. The destination may receive multiple copies of the same packet. This mechanism places most of the routing-related load at the more capable devices. It increases the data delivery probability as multiple copies of the same packet will travel towards the destination, however, also increases the network traffic.

In the latter case, the newly-arrived device advertises itself along with its ability to deliver data to the destination. Taking advantage of the newly-arrived device is at the discretion of the current hop. This mechanism ensures that only a single copy of the packet reaches the destination, at the expense of slightly increasing processing load at the less capable devices. It reduces the over all traffic in the network, however, it also reduces data delivery probability.

3.2 Calculate and Update Capabilities and Credentials

As discussed in section 1, device differentiation and assessment mechanisms are needed. A device's packet delivery ability assessment can be divided into device's own *capabilities* dictated by the resources and roles that it possesses, and the device's *credentials* that reflect the "goodness" of a device for delivering packets to their destinations.

Device capability reflects the tasks that the device can perform. A device that has an active long-range radio can send packets to Internet destinations. Similarly, a device with rechargeable or un-limited energy reserves may forward packets more frequently.

Device credentials specify the trustworthiness and/or feasibility to deliver packets to a certain destination or type of destination. They will improve as a device successfully delivers packets to their destinations. They not only depend on performance statistics, but also on the current environment. The credentials can be further divided into its long-term and short-term components.

In terms of RL, this component will consist of a state-value function (e.g., a Q-value function), influenced by a number of factors (discussed below). Different weights can be associated with information from current environment and past performance. Weighted parameters can allow the application to change its delivery priority, however, they reduce the usability of the protocol.

Rewards from feedback Rewards are received via feedback from downstream nodes, as a device participates in data forwarding. Performance of next hops can be evaluated from these rewards. They influence both, short-term and long-term credentials.

Current state The current state is influenced by the short-term component of the device credentials. Factors such as remaining energy, distance to destination and velocity contribute to current state. Device capabilities such as active radios also influence current state.

In a highly dynamic environment, acquisition of route health information is very costly. Therefore, a (mobile) relay may store, update and advertise its own credentials. This strategy will contribute in shifting the routing-related load towards more capable devices. Calculating credentials at the potential next hop may, however, raise trust issues such that the next hop may provide false or unreliable information.

Past statistics Past statistics will be recorded and used as the device participates in data forwarding, influencing the long-term component of credentials. Successful deliveries to particular or similar destinations and frequency of visiting locations or areas near destinations, becomes part of these statistics. Although information reliability deteriorates with time.

Neighbourhood Neighbourhood's capabilities and credentials influence the short-term component of device credentials. The update mechanism is discussed in section 3.3.

3.3 Neighbour discovery and credential update

This component is concerned with the neighbourhood discovery and the credential propagation. More capable devices will scan radio channel and discover neighbourhood, to shift load from resource constrained devices. The newly-arrived devices will, however, have to synchronize with the waking period of the sensor nodes.

Newly-arrived devices will promiscuously listen and analyze all the packets in their vicinity. By doing so, a device can assess itself to deliver the same packet and will become aware of its neighbourhood.

A device can either influence its neighbourhood by periodically advertising its abilities, or be influenced by its neighbourhood by promiscuously listening to the packet transmissions in its vicinity. In the former case, a device will discover all neighbours, increasing the memory requirements of the protocol. In the latter case, a device will discover only those neighbours that are currently forwarding data and none if no data is currently transmitted.

3.4 Feedback

A device will acknowledge the reception of a data packet to the previous hop. The performance estimate is obtained from this acknowledgement containing the expected reward. This estimate will improve with experience and should converge towards an accurate reward.

The first accurate reward is received by the hop n-1 from the final destination (n) upon reception of the packet. The next time a packet is delivered to the same destination; hop n-2 receives an accurate reward from hop n-1, and so on. This mechanism allows intermediate nodes to become aware about their performance and make better decisions along the way. If congestion is detected, the previous hop can be advised to find alternative routes. This process will, however, take several iterations before all the nodes are fully aware about their real performance, reducing the protocol's reactivity. Optimizing this mechanism to react fast is crucial in this research.

4 Conclusions

This paper discussed our research in-progress that explores taking advantage of unstable routes created by third-party mobile devices and assessment of device's data forwarding ability in a highly dynamic environment. It also discussed the solution design approach for a routing protocol that can efficiently move the routing load away from less capable devices, in addition to the challenges faced while designing the protocol. Using these unstable paths promise great rewards

in terms of energy conservation, end-to-end latency reduction, throughput improvement and fault tolerance improvement. The unstable nature of these routes, however, may also increase the risk of data loss.

Currently, the mathematical basis upon which credentials can be updated and the state-value function can be influenced is being explored, so that the solution reacts quickly to the changing environment. Also the design of evaluation criteria is in progress along with the detailed design and specification of different mechanisms for components involved in the envisaged routing protocol.

The next steps in this research involves evaluation of the routes created by third-party mobile devices through an implementation of the routing protocol on a WSN testbed.

5 Acknowledgments

This work is supported by Enterprise Ireland, the National Development Plan (Ireland) and the European Union via the ARTEMIS Joint Undertaking, under grant agreement n.100036

References

1. IEEE802.15.4: Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (WPANs) (September 2006)
2. IEEE802.16: Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1 (2005)
3. Tennina, S., Bouroche, M., Braga, P., Gomes, R., Alves, M., Mirza, F., Ciriello, V., Carrozza, G., Oliveira, P., Cahill, V.: Emmon: A wsn system architecture for large scale and dense real-time embedded monitoring. In: Embedded and Ubiquitous Computing (EUC), 2011 IFIP 9th International Conference on, IEEE (2011) 150–157
4. Ye, F., Luo, H., Cheng, J., Lu, S., Zhang, L.: A two-tier data dissemination model for large-scale wireless sensor networks. In: Proceedings of the 8th annual international conference on Mobile computing and networking, ACM (2002) 159
5. Kim, H., Abdelzaher, T., Kwon, W.: Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks. In: Proceedings of the 1st international conference on Embedded networked sensor systems, ACM (2003) 193–204
6. Akkaya, K., Younis, M.: Energy-aware routing to a mobile gateway in wireless sensor networks. In: Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. IEEE, IEEE (2005) 16–21
7. Kim, J., In, J., Hur, K., Kim, J., Eom, D.: An intelligent agent-based routing structure for mobile sinks in WSNs. Consumer Electronics, IEEE Transactions on **56**(4) (2010) 2310–2316
8. Kusy, B., Lee, H., Wicke, M., Milosavljevic, N., Guibas, L.: Predictive qos routing to mobile sinks in wireless sensor networks. In: Information Processing in Sensor Networks, 2009. IPSN 2009. International Conference on, IEEE (2009) 109–120

9. Basagni, S., Nati, M., Petrioli, C., Petroccia, R.: Rome: Routing over mobile elements in wsns. In: Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE, IEEE (2010) 1–7
10. Avudainayagam, A., Lou, W., Fang, Y.: DEAR: A Device and Energy Aware Routing protocol for heterogeneous ad hoc networks* 1. Journal of Parallel and Distributed Computing **63**(2) (2003) 228–236
11. Du, X., Wu, D., Liu, W., Fang, Y.: Multiclass routing and medium access control for heterogeneous mobile ad hoc networks. Vehicular Technology, IEEE Transactions on **55**(1) (2006) 270–277
12. Villasenor-Gonzalez, L., Ge, Y., Lament, L.: HOLSR: a hierarchical proactive routing mechanism for mobile ad hoc networks. Communications Magazine, IEEE **43**(7) (2005) 118–125
13. Xia, Y., Yeo, C., Lee, B.: Hierarchical Cluster Based Routing for Highly Mobile Heterogeneous MANET. School of Computer Engineering, Nanyang Technological University, Singapore (2009)
14. Sutton, R., Barto, A.: Reinforcement learning: An introduction. The MIT press (1998)
15. Tesauro, G.: Reinforcement learning in autonomic computing: A manifesto and case studies. IEEE Internet Computing (2007) 22–30
16. Dowling, J., Curran, E., Cunningham, R., Cahill, V.: Using feedback in collaborative reinforcement learning to adaptively optimize MANET routing. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on **35**(3) (2005) 360–372
17. Forstert, A., Murphy, A.: FROMS: Feedback routing for optimizing multiple sinks in WSN with reinforcement learning. In: Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on, IEEE (2008) 371–376
18. Boyan, J., Littman, M.: Packet routing in dynamically changing networks: A reinforcement learning approach. Advances in neural information processing systems (1994) 671–678

Improving Device-level Electricity Consumption Breakdowns in Private Households Using ON/OFF Events

Christian Beckel[1], Wilhelm Kleiminger[1], Thorsten Staake[2], and Silvia Santini[3]

[1] Institute for Pervasive Computing, ETH Zurich, 8092 Zurich, Switzerland
{beckel,kleiminger}@inf.ethz.ch
[2] Information Management, ETH Zurich, 8092 Zurich, Switzerland
tstaake@ethz.ch
[3] Wireless Sensor Networks Group, TU Darmstadt, 64283 Darmstadt, Germany
silvia.santini@wsn.tu-darmstadt.de

Abstract. Digital electricity meters – also referred to as smart meters – measure the total electricity consumption of a household at a fine temporal granularity. Using this data, detailed information like the consumption of individual appliances can be retrieved and used to provide novel services, such as personalized energy consulting. In this paper, we build upon existing work in consumption data disaggregation by enriching smart meter data with additional information made available by networked sensors and household appliances. In particular, we investigate the use of ON/OFF events, which signal when appliances have been turned on or off, respectively. We analyze the performance of an existing algorithm that uses such events along with smart meter data to estimate the consumption of single appliances. Our results, obtained by applying the algorithm to a publicly available dataset, show that the accuracy of the algorithm quickly deteriorates as the number of available ON/OFF events decreases. We thus suggest possible countermeasures to cope with this limitation and to provide accurate electricity consumption breakdowns in private households.

1 Introduction

The energy sector is currently undergoing a massive paradigm shift by reducing the dependency on fossil fuels towards an increasing share of renewable energy sources. To manage integration of these highly dynamic energy sources into the electricity network, smart meters are being installed in millions of private households worldwide [1]. A smart meter is a sensing device that can measure electric power consumption (from here on also referred to as *electricity consumption*) and can report the collected readings at given time intervals, e.g., every second, through a communication interface. The availability of smart meter data (possibly along with additional sensory information) enables the design and development of novel services and applications [10]. For instance, household inhabitants can be provided with fine-grained data about the contribution of

individual appliances to the overall electricity consumption. The availability of this information can potentially motivate users to an overall more thrifty usage of electricity, e.g., by inducing them to purchase more efficient devices [2, 9, 13]. Also, knowledge about the use of appliances can help determining the occupancy state of a household and thus enable other services, like automatic heating control [7].

Recently proposed approaches try to derive the consumption of single devices by adequately processing the data collected by a smart meter. However, because there are many possible combinations of appliances contributing to the electricity consumption at the same time, these centralized, single-sensor approaches usually achieve limited accuracy in real deployments [8,11]. An alternative, fully distributed approach consists in instrumenting all household appliances in order to measure their individual electricity consumption. For instance, so-called smart power outlets can be used to measure the consumption signatures and report them to a central processing unit [4]. These solutions can clearly provide a very accurate consumption breakdown but they also result in high deployment and maintenance costs. A third possibility consists in combining knowledge of the total electricity consumption, retrieved through a smart meter, with additional sensory information, like the sequences of *ON/OFF events* of each device [5, 6, 12].[4] Clearly, the lesser the amount of additional sensory information that is needed to achieve a given disaggregation accuracy, the more attractive is its use in practical settings. Previous studies have indeed shown that the knowledge of the total electricity consumption along with the sequence of ON/OFF events of each appliance is sufficient to estimate the consumption breakdown accurately [5]. However, the accuracy of the estimation decreases significantly if only partial knowledge about the sequence of ON/OFF events is available.

In this paper, we focus on these latter hybrid approaches and provide two main contributions. First, we evaluate the performance of an existing state-of-the-art load disaggregation algorithm that relies on the use of ON/OFF events along with smart meter data. In particular, we quantify the deterioration of the algorithm's performance in terms of device identification accuracy as the number of collected ON/OFF events decreases. To this end, we use a publicly available dataset of electricity consumption data [8]. Building upon our evaluation study, we then introduce mechanisms that can contribute in making the considered load disaggregation algorithm more robust against appliances that do not provide ON/OFF events. For this purpose, cooperation between different sensors and devices (e.g., smart meters, intelligent power outlets, light sensors) plays a crucial role in capturing additional sensory information, which is required to disaggregate the overall electricity consumption and attribute it accurately to its individual contributors.

[4] An ON/OFF event signals whether the device has been turned on or off.

2 Related Work

Accurate device-level consumption breakdowns could easily be measured if appliances were equipped with built-in electricity meters. This would however require costly hardware enhancements of the devices and is thus an impractical solution. Alternatively, smart power outlets (such as the ones from Plugwise[5]) can measure the electricity consumption at the socket level. Nonetheless, equipping each socket of a private household would require purchasing and installing a large number of devices, causing high costs and deployment effort.

To overcome these limitations, several authors focused on the concept of non-intrusive load monitoring (NILM). NILM strategies typically apply machine learning algorithms to the aggregated consumption profile [14] and thereby retrieve an estimation of the consumption breakdown. For instance, Kolter et al [8] resort to a Factorial Hidden Markov Model (FHMM) to perform load disaggregation. The model is trained on consumption data from multiple households and when used on test data from a different household it achieves an accuracy of 47.7%. Their evaluation relies on a metric that measures accuracy every 10 seconds, thus avoiding that errors even out over time. Parson et al. focus on the three appliances that consume the most electricity and use Hidden Markov Models (HMM) to guess their current state [11]. The corresponding experimental evaluation shows that restricting the set of appliances leads to an accuracy of 83% in terms of consumption breakdown. Both Kolter et al.'s and Parson et al.'s approaches rely solely on smart meter data. Instead, we make use of additional information from other sensors such as infrared sensors or magnetic field sensors and thus trade off a slight increase in complexity of the system for a corresponding increase in estimation accuracy.

Jung and Savvides also abandon the idea of disaggregating electricity consumption solely based on single-point measurements and take into account knowledge about ON/OFF states of all appliances [5]. To evaluate their approach the authors gathered three days of data from both a central electricity meter and from sensors mounted next to the appliance's switches that recorded ON/OFF events. On this dataset, the consumption breakdown algorithm achieves an accuracy of 90%. However, the algorithm relies on the assumption that complete knowledge about occurring ON/OFF events is available. This assumption is unfortunately hard to meet in real deployments. As we show in the following section, the performance of this disaggregation algorithm decreases significantly when ON/OFF events are recorded for only a subset of the available appliances, instead of for all of them.

The ViridiScope system [6] leverages magnetic field sensors and light sensors to indirectly sense electricity consumption of appliances. Smart meter data is only used for the purpose of sensor calibration. Within ViridiScope, uninstrumented appliances whose consumption cannot be directly measured are defined as *ghost power* consumers [6]. While our approach to consumption breakdown relies on ON/OFF events of appliances as in [5], ViridiScope approximates elec-

[5] www.plugwise.com

tricity consumption through indirect sensing. Furthermore, while in our work we explicitly address the problem of ghost power, ViridiScope assumes that the ghost power of non-instrumented appliances is constant and rather small.

In order to detect ON/OFF events, high frequency event detection approaches measure electricity consumption at the order of kilohertz and classify characteristic physical effects in the electrical signal. In ElectriSense [3], for example, Patel et al. detect consumer electronics devices and fluorescent lighting based on the fact that these appliances use switch mode power supplies, generating measurable electromagnetic interference (EMI) during operation. Since this approach requires expensive hardware we do not rely on high frequency sensing but base our algorithm on low frequency data from an ordinary smart meter and assume that we obtain ON/OFF events from another source such as being directly communicated by the appliance. In [12], for example, Rowe et al. provide a low-cost sensor that detects state changes from appliances such as refrigerators, lights, desktop computers, or televisions through variations in electromagnetic fields when being placed next to them.

So far, single sensor approaches based on low frequency smart meter data cannot provide an accurate breakdown of electricity consumption based on data collected in real households. Jung et al., on the other side, achieve high accuracy but require that all devices are equipped with ON/OFF sensors.

Our approach employs ON/OFF events in addition to smart meter data to gain high accuracy. It differs from previous work by considering ghost power caused by non-instrumented appliances as a part of our model. This makes our approach applicable in real world scenarios, in which not all appliances in a household are equipped with sensors reporting their ON/OFF state.

3 Performance analysis of a load disaggregation algorithm based on ON/OFF events

In this section we discuss the performance of a reference algorithm to perform load disaggregation using ON/OFF events. In particular, we quantify the deterioration in terms of device identification accuracy as the number of collected ON/OFF events decreases. In the following, we first elaborate on the opportunity of using ON/OFF events to perform load disaggregation and then describe in detail the reference algorithm considered for our performance evaluation. Before presenting the final results, we also describe the dataset that we have used for running our experiments.

3.1 Collection of ON/OFF events

The availability of smart meter data alone is often not sufficient to achieve high load disaggregation accuracies [8, 11] and the use of additional information, like the sequence of ON/OFF events, is thus accordingly unavoidable. Capturing ON/OFF events is particularly interesting as it only requires a lightweight (i.e., cheap and easy to deploy) sensing infrastructure [5, 6]. However, instrumenting

all or most devices within an household would still cause too high deployment and maintenance costs. Therefore, algorithms that can cope with partial information about ON/OFF events would significantly increase the attractiveness of approaches based on this load disaggregation technique.

The sensing infrastructure that can be used to gather ON/OFF events is shown exemplarily in Figure 1. The system includes a smart meter that captures data at a frequency of 1Hz and a number of smart power outlets (e.g., from Plugwise) that can measure and notify when a device is turned on or off. Although the specific smart power outlets used in our experiments are able to capture the actual consumption curve of a device, we only assume the availability of a sensor that is able to determine the operational state of the appliance. Additional sensors may also be used to indirectly measure ON/OFF events. For instance, a light sensor can measure the state of a lamp while a microphone can give an indication of the current state of a washing mashine. However, in order to make the collection of ON/OFF events feasible in practical scenarios, it must be implemented as simple, cheap, and unobtrusively as possible. Considering the steadily growing trend towards embedding communication capabilities (like Ethernet or WiFi interfaces) in common appliances, we believe that it is reasonable to assume that (at least a subset of) devices within a household will be able to determine and communicate their operation status autonomously. However, a given number of devices is likely to remain unobservable, e.g., old or cheap appliances. Therefore, a given amount of the total power consumption will remain non-attributable to the actual devices causing it. Borrowing the definition reported in [6], we refer to this "non-attributable" power as *ghost power*. Unlike [6], however, we do not assume ghost power to be small and constant. Instead, we include ghost power as an explicit variable in our model so as to increase accuracy of the overall consumption breakdown.

3.2 Reference disaggregation algorithm

To the best of our knowledge, Jung and Savvides's algorithm [5] is the only currently available algorithm that combines smart meter data with ON/OFF events in a comprehensive way. It thus represents an obvious choice as a reference algorithm to analyze the performance of load disaggregation approaches based on ON/OFF events.

The disaggregation algorithm of Jung and Savvides [5] solves a linear optimization problem to estimate the contribution of each appliance to the overall power consumption. To this end, it maintains a trace of the total electricity consumption as well as a state vector of active appliances. States with the same set of active appliances are merged on the fly by averaging the total power consumption and increasing a state counter. Using this data, the algorithm computes the average consumption of each appliance by minimizing the mean square error between the sum of estimates of all active appliances and the total electricity consumption. To improve the estimation accuracy samples of the ON/OFF state vector that show fewer appliances in the ON state as well as samples that occur frequently are given higher weight in the estimation. Similarly, stationary

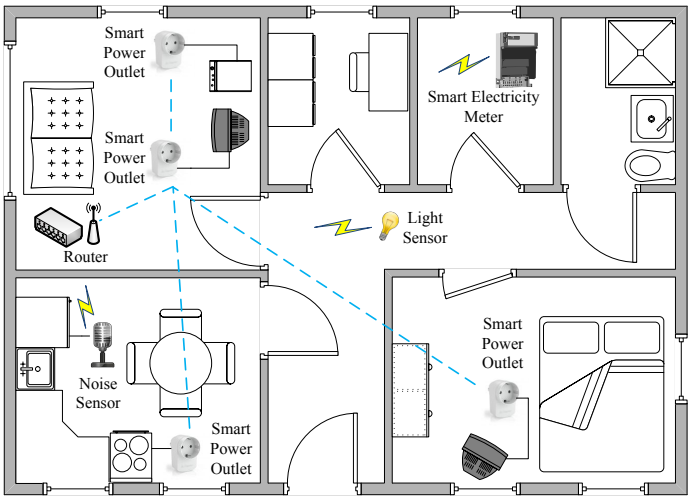


Fig. 1: Data collection architecture

loads are also given higher importance, as they can be estimated more accurately. The estimation procedure is performed over a specific time interval (e.g., one hour) and then restarted, whereas estimations from previous intervals are "remembered" for each successive iteration.

In order to perform the experimental study presented below, we implemented Jung and Savvides's algorithm in Matlab. For the sake of simplicity, however, we did not include the above mentioned weighting for stationary loads and we consider a single time interval only. We believe that the validity of our assertions about the robustness of the algorithm are however still valid, as our performance analysis focuses on a different issue (i.e. the effect of appliances that do not provide ON/OFF events on the estimation accuracy).

3.3 REDD dataset

In order to evaluate non-intrusive load monitoring algorithms using real world measurements, Kolter et al. released the REDD dataset [8], which is available at `http://redd.csail.mit.edu`. The initial release of the dataset (version 1.0) contains electricity consumption measurements from six households in the USA collected in April and May 2011. There are approximately 20 consecutive days of measurements available for each house. The REDD dataset provides data from the two main phases of each house at a granularity of one reading per second and measurements from 11 to 26 individual circuits – depending on the house – measured every 3–4 seconds. Some of the circuits contain a single appliance (e.g., a dishwasher) and thus qualify for device-level consumption breakdown. Other circuits contain multiple appliances (e.g., lights, kitchen outlets), which

Appliance	ON/OFF Events	Mean Power Consumption	Estimation Error
Oven	2	1,991W	12.9%
Fridge	29	193W	-0.9%
Dishwasher	2	552W	15%
Kitchen Outlets 2	11	56W	-61.8%
Kitchen Outlets 3	4	89W	12.6%
Kitchen Outlets 4	2	1,436W	29.6%
Lighting 1	5	90W	39.8%
Lighting 2	5	80W	4.7%
Washer / Dryer	4	1,897W	4.7 %
Microwave	20	1,239W	-4.3%
Bathroom	2	1,525W	21.2%
Total ON/OFF events: 86			
Mean of relative errors: 19.8%			
Relative error weighted by contribution: 14.3%			

Table 1: Estimation of consumption breakdown obtained by analyzing 24 hours of the REDD dataset.

can then only treated as a group of devices by the consumption breakdown algorithm. Since the REDD dataset represents to date the (or at least one of the) largest publicly available dataset of electricity consumption measurements.

3.4 Performance analysis

To evaluate the performance of Jung and Savvides's algorithm on the REDD dataset, we first extract ON/OFF events from the electricity consumption of each individual circuit. We then apply our version of the disaggregation algorithm on the dataset to obtain the electricity consumption of each appliance.

As a second step we analyze the estimation accuracy for each appliance using the circuit-level electricity consumption as ground truth information. Next, to investigate the effect of ghost power, we select a single base appliance as a test device and progressively filter out ON/OFF events of selected appliances. Clearly, this procedure does not affect the data relative to the aggregated electricity consumption. The choice of the appliances whose ON/OFF events are removed from the dataset is based on two strategies. The first strategy removes ON/OFF events of the appliance with the higher number of state transitions. The second strategy removes ON/OFF events of appliances that consume a large amount of electricity.

For the sake of simplicity and without loss of generality, we report results obtained from the REDD data relative to house 1 (which has a high number of individual circuits) and to April 24, 2011 (a day that exhibits many ON/OFF events). Table 1 shows the results obtained when applying our implementation of Jung and Savvides's disaggregation algorithm to the aggregated electricity

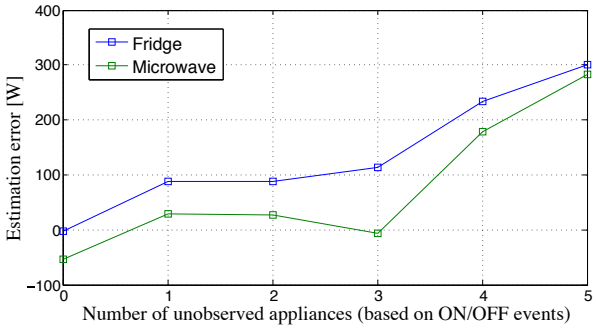


Fig. 2: Estimation accuracy based on ghost power of appliances with a large number of ON/OFF events.

consumption data and including all ON/OFF events. The first column lists appliances that provide ON/OFF events during the 24-hours long observation interval. The total electricity consumption is a result of the aggregation of the consumption of each of these devices. The second column shows that 86 state transitions occur during the whole time frame of 24 hours. The column *Mean Power Consumption* shows the average electricity consumption of each device during its ON phase, and the last column denotes the relative error of the estimated average consumption compared to its actual, measured value. The total relative error of 14.3% is obtained by comparing the aggregated values of the estimation and the actual mean electricity consumption of each appliance.

Figure 2 and Figure 3 show the resulting relative error depending on the number of appliances with missing ON/OFF events. Figure 2 illustrates the effect on the estimation error when following the first strategy by removing appliances with the higher number of state transitions. Similarly Figure 3 shows the graph that displays the relative estimation error obtained by removing appliances that consume a large amount of electricity. Both graphs illustrate that consumption from appliances with missing ON/OFF events is virtually spread over the rest of the appliances and thus highly reduces accuracy of the estimation.

4 Improving the accuracy of load disaggregation algorithms based on ON/OFF events

The results from the previous section show that the accuracy of Jung and Savvides's disaggregation algorithm decreases considerably as the number of reported ON/OFF events decreases. To perform load disaggregation in a real world setting, however, robust algorithms are required that can cope with missing ON/OFF events and thus run in houses that contain unobserved appliances. Unlike existing work [5, 6] we propose to explicitly model power consumption of uninstrumented appliances (i.e. ghost power). We thus integrate a simple virtual ghost power consumer into Jung and Savvvides' disaggregation algorithm.

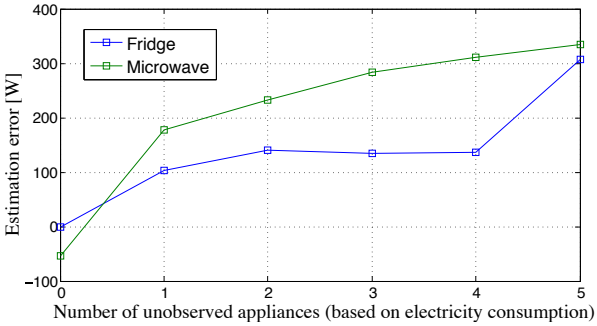


Fig. 3: Estimation accuracy based on ghost power of appliances with high electricity consumption.

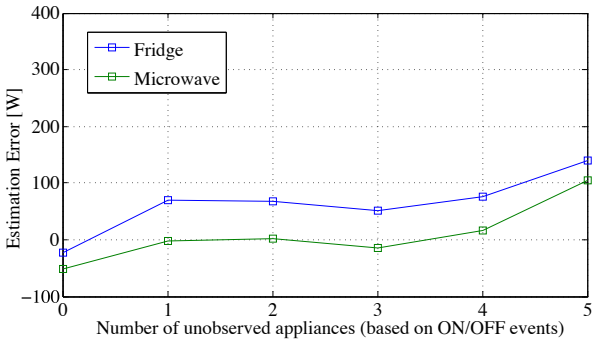


Fig. 4: Improved estimation accuracy using a simple model to attribute power consumption of unobserved appliances.

Acting as an "always-on" appliance it attributes for parts of the power consumption that is otherwise being (wrongly) assigned to other appliances. Similar to the procedure explained in Section 3.4 we apply the extended disaggregation algorithm on the REDD dataset and successively filter out ON/OFF events of appliances. As Figure 4 shows (compared to Figure 2), our version of Jung and Savvides's disaggregation algorithm outperforms the original version in case uninstrumented appliances are present.

Encouraged by these results we believe that the accuracy can be further improved by estimating power consumption of non-instrumented appliances more accurately. To this end, we integrate the following improvements into Jung and Savvides's algorithm: (1) Performing approximation over time; (2) Using characteristics of the load curve; (3) Employing state information gathered from sensors or the devices themselves; (4) Sharing consumption patterns. We describe these mechanisms in detail below, while the experimental evaluation of

the actual gains achieved by such measures in terms of estimation accuracy is planned for future work.

Value of time Over time it is possible to identify fractions of power consumption that are caused by non-instrumented appliances. This is done by observing specific (instrumented) appliances and discriminating ghost power from actual changes in the appliance's consumption pattern with a certain probability. In particular this works well for steady-state appliances, which exhibit a constant power consumption for a given time interval.

Appliances with a constant power consumption (such as lights) can be employed as an indicator for ghost power, in particular when the total power consumption exhibits a large variability while the set of running appliances remains constant. Observing such features allows an estimate on the proportion of non-instrumented appliances in a house. Such an estimate then supports calibration of the ghost power estimation itself, which is essential as we learned from our experiments performed on the REDD dataset.

Load curve characteristics The consumption patterns of many appliances exhibit characteristics such as periodicity (e.g., cooling appliances), a certain change in power when being switched on or off (e.g., lights, kettle), or a particular shape of the load curve once the appliance is running. This information, which is currently not included in Jung and Savvides' algorithm, could contribute to identifying ghost power consumption as follows:

- *Periodicity*: Some appliances (e.g., cooling appliances) exhibit a periodic consumption pattern that can be detected in the load curve. We first derive edges from the power consumption. In case the temporal occurence of these edges correlates with ON/OFF events observed from a sensor that reports such events, we assume that the edges are caused by an appliance whose events are reported by this sensor. Otherwise, we assume that they are caused by an appliance that is not instrumented to report ON/OFF events. In this case some part of the power consumption between these edges can be attributed as ghost power.
- *Change in power consumption*: Measuring the increase in power consumption of an appliance that provides ON/OFF events at the time it is being switched on gives information about the device's power consumption right at the beginning of its runtime. Similarly, the decrease in power consumption when an appliance is being switched off provides an estimate about the device's power consumption right before being switched off. First, this effect can be employed to estimate ghost power that disturbs the consumption pattern of appliances with a steady power consumption (e.g., a lamp). Second, ghost power caused by appliances with a steady power consumption can be estimated by investigating switching events of appliances that are instrumented to report ON/OFF events.

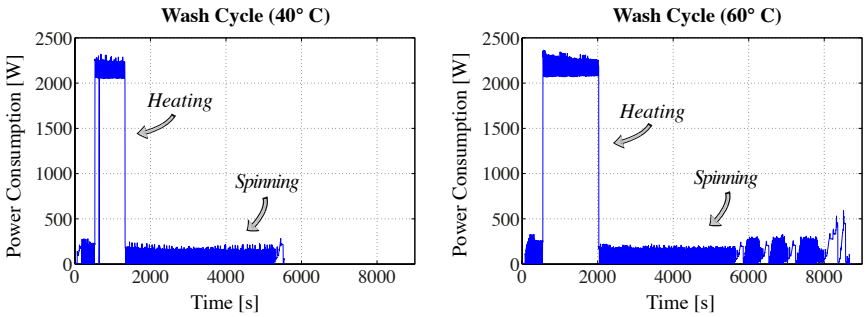


Fig. 5: Washing mashine cycles at 40°C (left) and 60°C (right).

– *Shape*: Over time, observing the shape of the load curve of a device that provides ON/OFF events reveals certain load curve characteristics. Therefore, changes in the consumption pattern that are not caused by observed appliances are possibly – but not necessarily – evidence for ghost power.

States Many household appliances exhibit a consumption pattern that is based on the states the appliance is running through, while having a constant power consumption throughout each state.Figure 5 compares the electric consumption of a washing machine at different temperature settings. Here we can distinguish between two main states – the heating of the water and the spinning of the drum. In order to measure the total power consumed by the washing machine accurately, knowing the time it spends in each state provides valuable input to the algorithm. Determining state transitions requires more sophisticated sensors than ON/OFF sensors such as vibration sensors or even network-connected appliances. The choice of these sensors depends on the appliance. Employing knowledge about state transitions of instrumented devices can increase accuracy of estimating the power consumption of this appliance. Hence, it improves the accuracy of ghost power estimation and thus the overall accuracy of the consumption breakdown.

Cooperative consumption analysis In order to differentiate between ghost power and the effects caused by instrumented appliances, comparing consumption patterns with more instrumented households potentially increases accuracy of less instrumented households, since devices might be identified based on their consumption pattern. Sharing consumption patterns of individual appliances comes at the expenses of an increased communication burden as well as potential privacy losses. The trade-off between these costs and the accuracy that is gained from cooperation must be thoroughly analyzed.

Integrating and evaluating these improvements on the REDD dataset highlights what proportion of appliances must be instrumented to provide an electricity

breakdown with high accuracy in a real world setting. Thus it contributes to making load disaggregation applicable in a real world environment. The information gained from analyzing the REDD dataset further provides the basis for our real world deployment, in which we collect electricity consumption information over a longer time period.

5 Conclusions

This paper presents the preliminary design of a novel approach to obtain electricity consumption breakdowns in residential settings. Our approach builds and improves upon Jung and Savvides's disaggregation algorithm, which relies on smart meter data and ON/OFF event reports to achieve a reliable consumption breakdown. We provide a quantitative analysis, based on the publicly available REDD dataset, to outline the limitations of Jung and Savvides's algorithm. In particular, we showed that the performance of the algorithm in terms of estimation accuracy quickly decreases as the number of missing ON/OFF events increases. Starting from this observation, we suggest a set of possible improvements, to perform disaggregation with only partial event knowledge.

Future work includes an extensive experimental evaluation of the proposed improvements and an analysis of what proportion – and type – of appliances needs to be instrumented in order to perform electricity consumption breakdown with high accuracy. In addition we investigate the scenario in which ON/OFF events are transmitted unreliably by extending the estimation algorithm to account for losses of events.

Acknowledgements

This work has been partially supported by the Hans L. Merkle Foundation and by the LOEWE research initiative of the State of Hesse, Germany, within the Priority Program Cocoon.

References

1. Burgess, J., Nye, M.: Re-Materialising Energy Use Through Transparent Monitoring Systems. Energy Policy 36(12), 4454–4459 (2008)
2. Fischer, C.: Feedback on Household Electricity Consumption: a Tool for Saving Energy? Energy Efficiency 1(1), 79–104 (2008)
3. Gupta, S., Reynolds, M., Patel, S.: ElectriSense: Single-Point Sensing Using EMI for Electrical Event Detection and Classification in the Home. In: Proceedings of the 12th International Conference on Ubiquitous Computing (UbiComp 2010), Copenhagen, Denmark (2010)
4. Jiang, X., Dawson-Haggerty, S., Dutta, P., Culler, D.: Design and Implementation of a High-fidelity AC Metering Network. In: Proceedings of the 8th International Conference on Information Processing in Sensor Networks (IPSN 2009), San Francisco, CA, USA (2009)

5. Jung, D., Savvides, A.: Estimating Building Consumption Breakdowns using ON/OFF State Sensing and Incremental Sub-meter Deployment. In: Proceedings of the 8th Conference on Embedded Network Sensing (SenSys 2010), Zurich, Switzerland (2010)
6. Kim, Y., Schmid, T., Charbiwala, Z., Srivastava, M.: ViridiScope: Design and Implementation of a Fine Grained Power Monitoring System for Homes. In: Proceedings of the 11th International Conference on Ubiquitous Computing (UbiComp 2009), Orlando, FL, USA (2009)
7. Kleiminger, W., Santini, S., Weiss, M.: Opportunistic sensing for smart heating control in private households. In: Proceedings of the 2nd International Workshop on Networks of Cooperating Objects (CONET 2011), Chicago, IL, USA (2011)
8. Kolter, J.Z., Johnson, M.J.: REDD: A Public Data Set for Energy Disaggregation Research. In: Proceedings of the KDD Workshop on Data Mining Applications in Sustainability (SustKDD 2011), San Diego, CA, USA (2011)
9. Mattern, F., Floerkemeier, C.: From the Internet of Computers to the Internet of Things. In: Kai Sachs, Ilia Petrov, Pablo Guerrero (Eds.): From Active Data Management to Event-Based Systems and More. LNCS, vol. 6462, pp. 242–259. Springer (2010)
10. Mattern, F., Staake, T., Weiss, M.: ICT for Green – How Computers Can Help Us to Conserve Energy. In: Proceedings of the ACM e-Energy Conference (2010)
11. Parson, O., Ghosh, S., Weal, M., Rogers, A.: Using Hidden Markov Models for Iterative Non-intrusive Appliance Monitoring. In: Machine Learning for Sustainability Workshop (MLSUST 2011), Sierra Nevada, Spain (2011)
12. Rowe, A., Berges, M., Rajkumar, R.: Contactless Sensing of Appliance State Transitions Through Variations in Electromagnetic Fields. In: Proceedings of the 2nd Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys 2010), Zurich, Switzerland (2010)
13. Weiss, M., Graml, T., Staake, T., Mattern, F., Fleisch, E.: Handy Feedback: Connecting Smart Meters with Mobile Phones. In: Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia (MUM 2009), Cambridge, UK (2009)
14. Zeifman, M., Roth, K.: Nonintrusive Appliance Load Monitoring: Review and Outlook. IEEE Transactions on Consumer Electronics 57(1), 76–84 (2011)

Modeling Building Thermal Response to HVAC Zoning

Virginia Smith, Tamim Sookoor, and Kamin Whitehouse

Department of Computer Science, University of Virginia,
Charlottesville, VA 22903, U.S.A.
{vs2z,sookoor,whitehouse}@virginia.edu

Abstract. HVAC systems account for 38% of building energy usage. Studies have indicated at least 5-15% waste due to unoccupied spaces being conditioned. Our goal is to minimize this waste by retrofitting HVAC systems to enable *room-level zoning* where each room is conditioned individually based on its occupancy. This will allow only occupied rooms to be conditioned while saving the energy used to condition unoccupied rooms. In order to achieve this goal, the effect of opening or closing air vent registers on room temperatures has to be predicted. Making such a prediction is complicated by the fact that weather has a larger effect on room temperatures than the settings of air vent registers, making it hard to isolate the influence of the HVAC system. We present a technique for dynamically estimating the heat load due to weather on room temperatures and subtracting it out in order to predict the effect of the HVAC system more directly.

Keywords: Building energy; energy; environment; sensing

1 Introduction

Buildings account for 75% [1] of the electricity and 43% of the greenhouse gas emissions in the United States [2] and the Heating, Ventilation, and Cooling (HVAC) system is the single largest energy consumer in residential buildings, accounting for 43% of the residential energy consumption in the US [3], and over 60% in Canada [4] and the UK [5], which have colder climates. This accounts for 38% of all the building energy used in the United States and over 15% of the total energy used in the U.S., making HVAC systems one of the nation's largest energy consumers. Studies have indicated that at least 5-15% of this waste is due to the course-grained, manual configuration of thermostats by users, whereby spaces are heated or cooled even if not needed by the occupants. Much of this wasted energy is used to heat or cool unoccupied spaces during long periods when people use only a small fraction of a house, such as when they work in an office or sleep in a bedroom. Our vision is to minimize this energy wastage through room-level zoning, where each room is conditioned individually based on its occupancy. This would allow most, if not all, of the energy used by

the HVAC system to be focused on maintaining occupied rooms at a comfortable temperature without wasting any energy conditioning unoccupied rooms. Many homes in the United States have centralized HVAC systems that have a single compressor or furnace. Such systems have to be configured for zoning during installation if homeowners want to minimize energy wastage. Most, if not all, zoned HVAC systems are implemented in multi-story houses where each floor is configured to be a separate zone. Due to the room usage generally being separated by floor, so that the bedrooms are on the upper floor and the living spaces on the lower floor, a coarse-grained zoning schedule can be manually configured for such a system allowing energy savings. For example, the system can be configured to condition the upper floor only during the night, when the bedrooms are in use, and the lower floor only during the day, when the living spaces are most likely to be used. Such a scheme cannot be used in a single-story house because the night and day living spaces are adjacent to each other. Also, the fact that rooms on a single floor are not as thermally isolated as rooms on separate floors reduces the energy savings that can be achieved through coarse-grained zoning within a floor.

Our goal is to implement a system that can retrofit the centralized HVAC systems that are in most homes in the United States so that air vents can be controlled individually and room-level zoning can be achieved. Such a system would require an automated controller that decided which rooms have to be conditioned and dynamically alters the zones based on occupancy and room temperature by opening and closing air-vent registers in rooms. In order for such a controller to be efficiently implemented, the affect of opening or closing registers on the temperatures in the room have to be predictable. Thus, in this paper we present and evaluate techniques to learn and predict the effect of opening or closing each vent register, in a set of R air vent registers, on the temperature at each sensor, in a set of T temperature sensors placed within a house.

The main challenge to modeling the thermal characteristics of a house is the effect of weather on the indoor temperature. For instance, wind, solar gain, and outdoor temperature have a greater influence on indoor temperature than any individual air vent register. It is difficult to build a model that completely captures the effect of weather on indoor temperatures because outdoor weather conditions constantly change and rarely repeat. The difficulty of attributing the influence on weather conditions on indoor temperature makes it difficult to isolate the effect of the state of any particular air vent register on the indoor temperature.

Our approach to overcoming this problem is to model the indoor temperature in two stages. In the first stage, we measure the rate of heat gain or loss due exclusively to outdoor weather conditions. This stage is modeled with data collected when the HVAC system is off using a linear function of current temperature. Then, when the HVAC system is turned on, we measure the *change* in the rate of heat gained or lost in a room due to the conditioned air provided by the HVAC system. We expect this change to be constant throughout the year because the HVAC system always outputs the same amount of conditioned air.

Thus, we isolate the HVAC effects by learning and subtracting out a dynamic estimate of weather effects over long periods of time.

In this paper, we present three iterations of a thermal model and analyze its accuracy in terms of predicting the effect of opening and closing various combinations of registers with a centralized HVAC system. An analysis of the HVAC system itself is beyond the scope of this paper. Performing ten-fold cross validation over three weeks of data sampled over three months, we demonstrate that even with the simplest model we can predict temperatures to within two degrees 30 minutes into the future. We focus on a 30 minute time window because longer time windows are not beneficial when making HVAC control decision. We also demonstrate that even the simplest of the three models we present in this paper is able to provide this level of accuracy allowing temperature prediction to be incorporated into an HVAC zoning controller easily and without much computation overhead.

2 Background

Heating, Ventilation, and Air Conditioning (HVAC) control systems are devised in order to maintain comfort within an enclosed space. In addition to meeting a desired temperature, this comfort is maintained by achieving a certain level of humidity, pressure, radiant energy, air motion, and air quality within a building [6]. The testbed in this study utilizes a centralized heat pump air conditioner. This is the most common method of residential air conditioning in the United States. Centralized HVAC systems do not permit fine-grained room-level control of the HVAC equipment beyond opening and closing air-vent registers that feed air into rooms. Thus, knowing the affect of opening or closing dampers is critical to the efficient retrofitting of a centralized HVAC system to enable room-level zoning.

2.1 Centralized HVAC System

The framework for the HVAC system is the air handling unit. The main responsibility of the air handling system is to deliver conditioned air throughout the building, while removing exhaust air and carbon dioxide (CO2) from the rooms. Most of the equipment is hidden from occupants, being located outside and in ducts within the building [6].

The air handling system may include fans, compressors, heating/cooling coils, and ducts, in addition to system controllers. The air handling process works in the following way: First, outdoor air is mixed with the return air of the system. The pressure of this air is determined by the supply air fan. The air is then heated/cooled to a preset temperature, and is released into specific spaces through the dampers. The exhaust air from the room is sent into the ducts according to the exhaust fan speed, and it is returned to begin the process again.

The damper is a mechanical device that allows for a variable amount of supply air to be released into a room. It consists of a thin metal sheet, rotated on an axis by an actuator. If the damper is set at 90 degrees, or 0% open, the damper is fully shut and no air is supplied to the room. When the damper is set to 0 degrees, or 100% open, the maximum amount of air is released [6].

2.2 System Logic

A centralized HVAC system can run in four possible states when heating/cooling: Float; Hold; Heat/Cool 1; and, Heat/Cool 2. Float causes the HVAC system to turn off, and hold tells the system to remain at the same temperature. Heat/Cool 1 corresponds to running the system at 67%, which provides a lower level of heating/cooling that can supply a base level of conditioned air throughout the day. Heat/Cool 2 turns on when temperature needs to be changed by a significant amount, and the system runs at 100%. The system in our testbed runs stage 2 conditioning if the current temperature is more than two degrees above/below the current setpoint [7].

2.3 Zoning

Most of the energy wasted by HVAC systems go towards heating or cooling unoccupied spaces during long periods when people use only a small fraction of a house. For instance, at night the bedrooms are used while the rest of the house is unoccupied and during the day the living room and kitchen maybe used with the bedrooms being unused for long periods of time. Zoning systems attempt to exploit this fact, and save energy for homeowners, by dividing a building into two or more zones that are controlled by separate thermostats, so that the occupants can schedule each zone to be heated or cooled separately. However, zoning systems are expensive, and are, therefore, typically only used for very course-grained zoning of the house: a typical configuration can condition the first floor living spaces separately from the second floor sleeping quarters for example. Such systems are both spatially and temporally course grained allowing large areas, in this case floors of a building, to be zoned separately and scheduled with a low frequency, for example switching between the living and sleeping areas only twice a day.

3 Problem Definition

Our problem is defined by a set of air vent dampers D and a set of temperature sensors T that are dispersed across a house (Figure 2). The dampers can be opened or closed, determining if conditioned air is delivered directly into a room. Due to the lack of thermal isolation between rooms, even if the air vent dampers of a room are closed, its temperature could still be affected by the HVAC system due to leakage from neighboring rooms. The temperature sensors monitor the temperatures at different points throughout the house. Figure 1 shows the

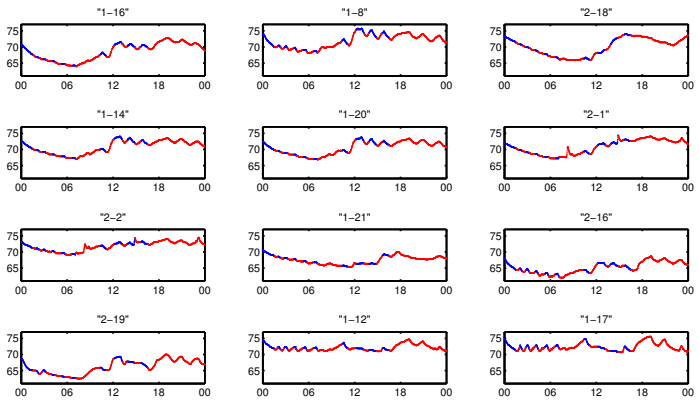


Fig. 1. The effect on temperature sensors, within a 24-hour period, of the HVAC system being on (red) and off (blue) when heating with all air vent dampers open. The locations of the twelve sensors are presented in Figure 2

readings at the twelve temperature sensors in our deployment during a day with all air vent dampers open. As the figure shows, the HVAC system being off (blue) causes drops in temperature while the HVAC system being on (red) usually causes temperature increases. We are attempting to learn and predict these effects on the temperature sensors when different sets of air vent dampers are opened and closed. In other words, we want to answer the question *"What effect does each register being open have on the reading of each temperature sensor?"* Being able to make such a prediction allows us to implement a fine-grained automated zoning controller that can dynamically alter zones within a single floor to maintain occupied rooms at a comfortable temperature while allowing unoccupied rooms to drift. Yet, answering this question is difficult due to the effect of the weather on the internal temperature of houses. Wind, solar gain, outdoor temperature, and other weather conditions have a much greater influence on indoor temperature than the conditioned air provided by an HVAC system. These weather conditions constantly change, and rarely repeat, therefore including it as part of a model is impossible without greatly increasing the complexity of the model. But, ignoring the effect of weather on internal temperature makes it impossible to isolate the effect of a particular register on a temperature sensor. Thus, a secondary question we are attempting to answer is *"Can we learn the effect of dampers on temperature sensors without knowing the weather during the training phase?"* In other words, we are attempting to capture the effect of the weather on the temperature sensor readings while ignoring the actual weather conditions, such as the external temperature or the position of the sun.

There have been a number of approaches proposed for learning the thermal response of buildings in order to control HVAC systems efficiently [8–13]. Yet, these approaches require a large amount of data or sophisticated sensors that will hinder our goal of developing a cheap and easy to install retrofit to enable room-level zoning of existing centralized HVAC systems.

4 Experimental Setup

The room-level zoning system described has been deployed in a single-story, 8-room, 1,200-square-foot residential building. A model of the home is shown in Figure 2. The hallway and porch are depicted, but not included within our analysis because of the inability to actuate temperature within these regions. The HVAC system setup is overlaid in order to show the position of vents, ducts, and the central air handler.

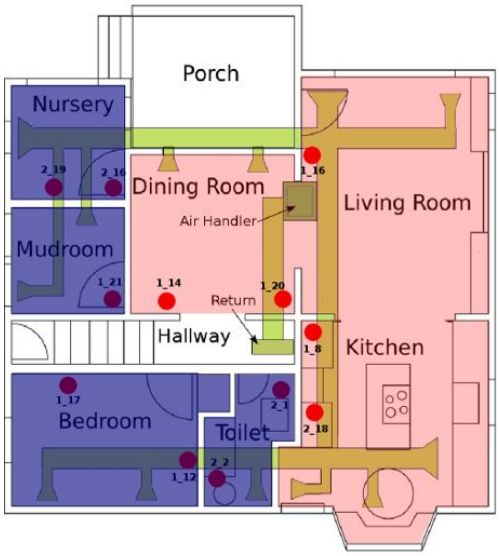


Fig. 2. The residential testbed used for this study. Red and blue overlays show an example of two room-level zones, the green ducts terminate in air vent registers that can be opened or closed, and the red circles show the locations of the twelve temperature sensors with the sensor IDs indicated.

Figure 2 shows the deployment from which data for this paper was collected. We used twelve temperature sensor deployed across the house and air vent registers that are remotely actuatable and collected data over a three month period.

Three weeks of the collected data was used for the analysis presented in this paper.

4.1 Temperature Detection

Sensors deployed throughout the building allow us to monitor the temperature and HVAC status within each room/zone. We collect temperature data at a fine granularity using temperature sensors placed at various points along the walls. In order to ensure the scalability of this system, we use 12 standard, off-the-shelf temperature sensors manufactured by La Crosse Technology [7].

One challenge with sensing temperature in this way is that temperatures are not uniform throughout the rooms/zones and along the walls. This can present problems when trying to determine the true temperature of each room. As shown in Figure 3, the placement of wall sensors has a large impact on the variability of the temperatures detected. While the sensors on the internal wall vary within the temperature range dictated by the return duct, the sensors on the external wall are subject to large temperature swings. This is because the external wall sensors pick up temperatures from outside of the building through windows, doors, and the wall itself. This is also compounded by the fact that most vents are placed on external walls, making these sensors subject to direct air from the duct [7].

Thus, we use two methods to ensure accuracy within our temperature data collection. The first is to only place sensors along the interior walls of the rooms. The second is to record the temperature as an average of these sensors, helping to detect the temperature more uniformly throughout the room.

4.2 HVAC Status Detection

The HVAC system used in this study can run in four possible states when heating/cooling: Float; Hold; Heat/Cool 1; and, Heat/Cool 2. Data on these system states are collected by interfacing with a standard internet-controlled thermostat manufactured by BAYweb. These stages are described in detail in the following section.

5 Model of Temperature Dynamics

The parameters of the model include the position of the damper, temperature, system status, and time. These values are recorded through a wireless sensor network deployed in the testbed and stored in a database. The temperature values are measured in degrees Fahrenheit, and the damper positions take one of two values: 0 (closed) or 1 (open 100%). The system status allows us to see whether the system is in off, heat/cool 1, or heat/cool 2 mode. An example of the damper, temperature, and system status for one room is shown in Figure 3.

In analyzing this system, we explore a number of different models. Three iterations of our final model are shown in the following sections. Each is a dynamic,

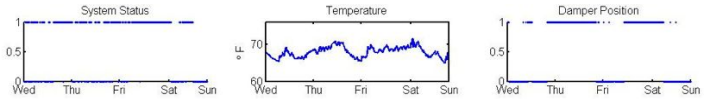


Fig. 3. The system status (on/off), temperature (°F), and damper position (open/closed) for one room in our testbed over the period 11/30/2011-12/04/2011.

linear model that is developed in two stages. The first stage aims to estimate the effects of heating/cooling due to external factors such as solar radiation, wind, and cloud coverage. This effect is calculated when the system is turned off, and the values are then used to develop the model when the system turns back on. This two-stage approach allows us to compensate for external factors without having to measure them directly. Furthermore, the results allow us to predict temperature dynamics due to the HVAC configuration with greater accuracy.

$$dT_k/dt = \alpha T + \beta D \tag{1}$$

The models we discuss follow the same format (Equation 1) in which the temperature of a specific room T_k over time t is a result of external factors (calculated through α), and the current damper configuration, D. The three iterations of this model differ in the way that the external factor coefficient, α, is calculated. These differences are as follows: 1) The first iteration calculates a universal α value by pooling the data when the system is off. 2) The second iteration calculates a constantly changing α value when the system is off, and uses this constantly calculated α value in the model when the system turns on. 3) The third iteration adds to the model complexity by using universal α values for all neighbors $T_1, ..., T_n$ of the temperature in room k, T_k.

5.1 Static α

The first iteration of the model we describe is one in which the α values, which estimate the temperature change due to weather patterns, are constant throughout the day. In order to calculate these values, we pool the data from times when the system is off together and fit one α value across all timesteps for each of the n rooms. This value is calculated through linear regression, and assumes that the heat load due to weather remains relatively constant throughout the day.

5.2 Dynamic α

In the second iteration, we explore the idea that the heat load due to weather conditions may be changing continuously throughout the day. In order to do this, we calculate a dynamically changing α value for each off segment, and include that value in the on segment that directly follows it. This method aims to compensate for weather by assuming that the heat load due to weather changes

significantly throughout the day, but by very little between one cycle of the system.

5.3 Adjacency Model

The third iteration increases the complexity of the first by including the other n rooms into the model. This assumes that the current temperature of the room is affected not only by its own weather conditions, but also by the temperature dynamics within the other rooms of the building. This model also calculates the α values universally through linear regression. The form of this room adjacency model is as follows:

$$dT_k/dt = \alpha_1 T_1 + \alpha_2 T_2 + \ldots + \alpha_n T_n + \beta D \tag{2}$$

6 Results

We compare the three iterations of our model described in section 5 using 21 days worth of data tested with 10-fold cross validation which involves randomly dividing the 21 days of data into ten equal sets, training the model using nine of those sets, and testing with the remaining set. All combinations of nine sets for training and one set for testing are used. The 21 days we have selected for model development and testing have been sampled from 3 months worth of data between October and December 2011. Using the training data, we develop the β values for the model. We then use these values with the α value scheme dictated by the model iteration in order to predict temperatures when the system turns on.

6.1 Prediction

Our predictions assume that temperature grows linearly when the system turns on as a result of the current damper configuration and the previous weather patterns estimated through α. Though temperature dynamics within a building are often nonlinear, we find a reasonable estimate by predicting temperature linearly into the future. This is because the temperature and airflow of the system operate within a narrow regime, making it reasonable to approximate change with a linear model. An example of a prediction 30 minutes into the future is shown in Figure 4. Here, the blue lines represent the actual temperatures and the red line plots our prediction. The solid blue line shows the temperature when the system is off, and the red/blue dashed lines show the predicted/actual temperatures when the system has just turned on.

6.2 Error Metric

One difficulty in determining the effectiveness of these models is that we aim to use them to predict temperatures at more than one timestep into the future.

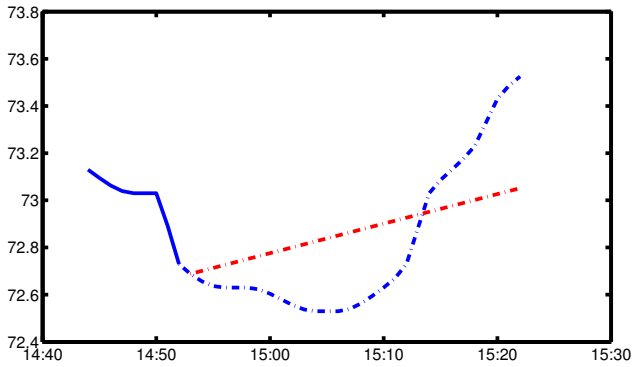


Fig. 4. An example of a prediction made for temperature up to 30 minutes into the future after the system turns on. The solid blue line shows the actual temperature when the system is off; the dashed blue line shows the actual temperature when the system is on; and the dashed red line shows temperature predicted after the system has just turned on.

This involves calculating predictions at each point that the system is on, up to t minutes into the future until the system turns off again.

The error metric that we have chosen for this comparison is to determine the distribution of prediction error as we predict t minutes into the future. For each minute, t, we calculate the mean and standard deviation of the prediction errors t minutes away from the initial time. The results from these analyses for the static α, dynamic α, and adjacency model are shown in Figure 5, Figure 6, and Figure 7 respectively. These results are calculated on a per-sensor basis for each of the 12 sensors in the 7 rooms of the building.

Visually examining the error distributions highlights a few important things about the model. One is that the variance of the errors tends to increase as we predict further into the future. The error can get quite large in some places, particularly in the dynamic α model. However, most of the values for each model remain within 2 degrees for the 30 minute prediction. This is a reasonable interval with which to enable the control of the system that we aim to accomplish.

The results from this analysis also indicate that the simple, pooled α model performs better than the dynamic model. This may be counterintuitive since weather tends to change significantly throughout the day. However, because of the window we are looking at and the narrow range of temperature change, it is reasonable that this model should perform well. It also has the added benefit of being computable and easy to implement within a control setting.

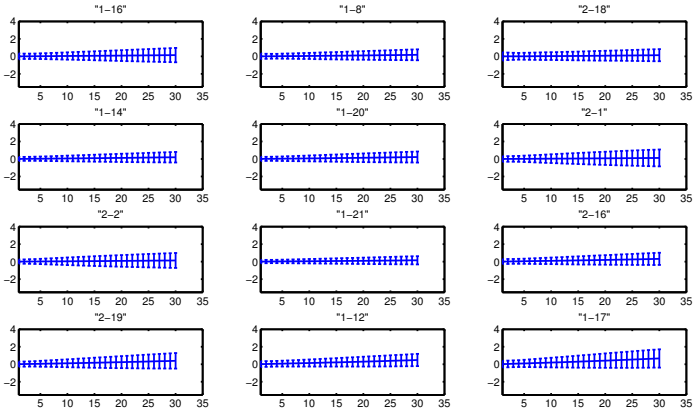


Fig. 5. Error distribtuions for the static α model, up to 30 minutes into the future. The locations of the twelve sensors are presented in Figure 2

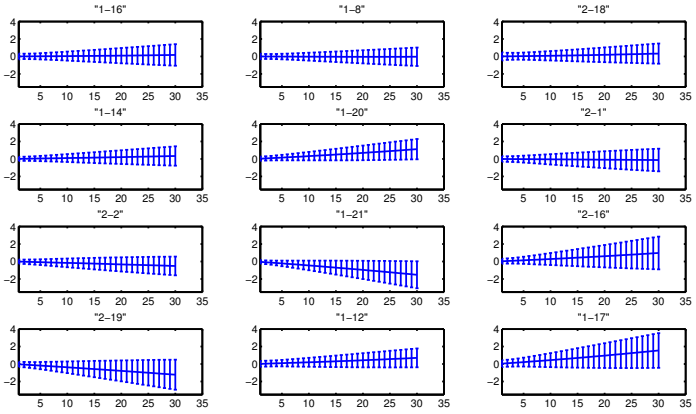


Fig. 6. Error distributions for the dynamic α model, up to 30 minutes into the future. The locations of the twelve sensors are presented in Figure 2

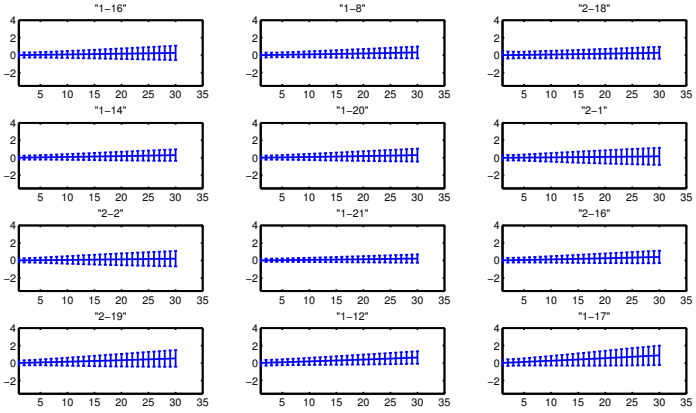


Fig. 7. Error distributions for the adjacency model, up to 30 minutes into the future. The locations of the twelve sensors are presented in Figure 2

7 Work-In-Progress

An observation we made with the model presented in this paper is that its linear nature fails to capture the mixing period experienced when the HVAC system first turns on. As Figure 4 shows, the temperature measured at a sensor continues to drop for about twenty minutes after the HVAC system is turned on before it begins warming up. This could be caused by the time taken for the conditioned air to sufficiently mix with the cold air in the room before the increase in temperature is detectable by a sensor and the absorption of heat by the structure of the room, such as walls and floors, as well as objects in the room such as furniture before the air get heated because these objects have a higher heat capacity than air. In order to capture this mixing period, we modify the thermal model by introducing a variable γ that varies with time and influences the effect the conditioned air from the HVAC has on the temperature sensor.

We estimate values of γ by creating a set of equations, such as the following, at various times from the time the HVAC turns on until 30 minutes into the future:

$$T_1 - T_0 = \alpha + \gamma_1 \beta D \tag{3}$$

$$T_2 - T_0 = \alpha + \gamma_2 \beta D \tag{4}$$

$$T_3 - T_0 = \alpha + \gamma_3 \beta D \tag{5}$$

Solving these equations for historical temperature data and HVAC state provides a set of γ values. Using these γ values a new iteration of the model can be specified as follows:

$$dT_k/dt = \alpha T + \gamma_k \beta D \tag{6}$$

We are currently in the process of training and evaluating this model.

8 Conclusions

We have presented our residential testbed, studied the characteristics of the dual stage HVAC, identified and analyzed mathematical models of the system, and discussed the impact of our results. The two-stage, dynamic model that we have developed provides an accurate way to predict the temperature in a zone based on a few, accessible parameters in the system. It also allows the calculation of highly variable terms, such as the heat load due to solar radiation, wind, and cloud coverage, without the need to explicitly measure these terms.

These results will be used in future work in order to develop a new, energy efficienct control scheme for the system. The model gives us better insight into the dynamics of the control scheme and allows for a more efficient design. This control scheme may then be used to create a more energy efficient design for similar HVAC units. This type of work is a crucial step in the developing the type of energy-agile systems that can ultimately be used to quell our dependency on fossil fuels.

References

1. U. DoE. Buildings energy data book. `http://buildingsdatabook.eren.doe.gov/`, 2011.
2. J. McQuade. A system approach to high performance buildings. Technical report, United Technologies Corporation, 2009.
3. Energy Information Administration. 2005 residential energy consumption survey. http://www.eia.doe.gov/emeu/recs/contents.html.
4. Energy Policy Branch Energy Sector Energy Forecasting Division. *Canada's energy outlook, 1996-2020*. Natural Resources Canada, 1997.
5. K Rathouse and B Young. Domestic heating: Use of controls. Technical Report RPDH 15, Building Research Establishment, UK, 2004.
6. N. Ben-Aissa. Heating, ventilation, and air conditioning (hvac) controls: Variable air volume (vav) systems. Johnson Controls, Inc.
7. Tamim Sookor, Brian Holben, and Kamin Whitehouse. Feasibility of retrotting centralized hvac systems for room-level zoning.
8. G. Henze, C. Felsmann, and G. Knabe. Evaluation of optimal control for active and passive building thermal storage. *International Journal of Thermal Sciences*, 43(2):173–183, 2004.
9. K. Deng, P. Barooah, P. Mehta, and S. Meyn. Building thermal model reduction via aggregation of state. In *American Control Conference*, pages 5118–5123, 2010.
10. F. Oldewurtel, A. Parisio, C. Jones, M. Morari, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and K. Wirth. Energy efficient building climate control using stochastic model predictive control and weather predictions. In *American Control Conference*, pages 5100–5105, 2010.
11. Y. Ma, G. Anderson, and F. Borrelli. A distributed predictive control approach to building temperature regulation. In *American Control Conference*, 2011.
12. T. Nghiem and G. Pappas. Receding-horizon supervisory control of green buildings. In *American Control Conference*, 2011.
13. A. Aswani, N. Master, J. Taneja, D. Culler, and C. Tomlin. Reducing transient and steady state electricity consumption in hvac using learning-based model predictive control. In *Proceedings of the IEEE*, 2011.

Do Sensor Networks need Mobile MAC Protocols?

Navid Hassanzadeh[1], Olaf Landsiedel[2,4], Frederik Hermans[3], Olof Rensfelt[3] and Thiemo Voigt[1]

[1] Swedish Institute of Computer Science (SICS), Kista, Sweden
[2] Chalmers University of Technology, Sweden
[3] Uppsala University, Sweden
[4] KTH Royal Institute of Technology, Sweden

Abstract. There exists a number of MAC protocols targeted for mobile scenarios. These include MMAC, MS-MAC and AM-MAC. These MAC protocols have in common that they seem to be evaluated only in simulation. This might indicate that these MAC are either too complex to use or they are not needed, at least for data collection, the major task of sensor networks. In this paper we show that extending a traditional data collection protocol with lightweight, carefully selected mechanisms is sufficient to provide reliable data collection at low energy cost for mobile sensor networks where both sinks and sources move.

1 Introduction

There are quite a few MAC layers specifically designed for mobile scenarios. These include MMAC [1], MS-MAC [2] and AM-MAC [3]. All of these MAC protocols are evaluated by simulation and it is unclear if implementations for real sensor node hardware exist. Given the complexity of some of these protocols and the absence of implemenations that work on real sensor nodes, we wonder if these mobile MAC protocols are really needed. We investigate this question for mobile scenarios where mobile sink nodes collect data from mobile data sources. We target scenarios where data is collected in real-time. This is in contrast to approaches that use mechanisms from delay-tolerant networking such as ZebraNet [4] or RatPack [5] where data is collected for off-line analysis and hence longer delays do not matter.

In this paper we advocate a different approach to tackle the challenging issue of mobile data collection. Rather than designing a new MAC protocol, we enhance an existing data collection protocol for static networks with lightweight mechanisms to improve performance in mobile scenarios. Towards this end, we modify the Contiki Collect protocol [6], a protocol similar to the Collection Tree Protocol (CTP) [7] for Tiny OS, to make it more suitable for mobile scenarios. We enhance Contiki Collect with mechanisms to detect and repair loops since these occur more often in mobile than in static scenarios. Furthermore, we enable nodes to quickly find new parents as nodes often move out of range in mobile scenarios. We provide an implementation for the Contiki operating system that we call Mobile Collect.

We evaluate Mobile Collect by both simulation and in a testbed that includes mobile robots. We also perform simulations with a random waypoint mobility model. Our experiments show that Mobile Collect performs very well in such a scenario. Even in scenarios with very high mobility (2 to 8 m/s), Mobile Collect still achieves a delivery rate of 70% and more at a low energy consumption of 10 mJ per received packet. Note that, for example, the simulations in MMAC were performed with an average speed of only 0.1 m/s [1]. Given these results, we conclude that in many scenarios with mobile nodes, there is no eminent need for new MAC protocols specifically designed for mobility.

2 Design and Implementation

As mentioned above we enhance Contiki Collect with two types of mechanisms. First, we enable nodes to more quickly find new parents as in mobile scenarios nodes often move out of range. Second, we need mechanisms to detect and repair loops since these occur more often in mobile than in static scenarios. Additionally, we integrate routing beacons into the receiver initiated MAC-layer.

MAC Layer Beaconing: Topology dynamics triggered by node mobility require nodes to frequently announce their presence and routing metrics to neighboring nodes. To avoid additional cost from this beaconing, Mobile Collect integrates its beacons into the MAC layer: In each probe of its receiver initiated MAC [8] a node also announces its current routing metric. As these probes are transmitted on each wakeup of our duty-cycled nodes, Mobile Collect achieves a high rate of routing beacons with essentially no additional cost.

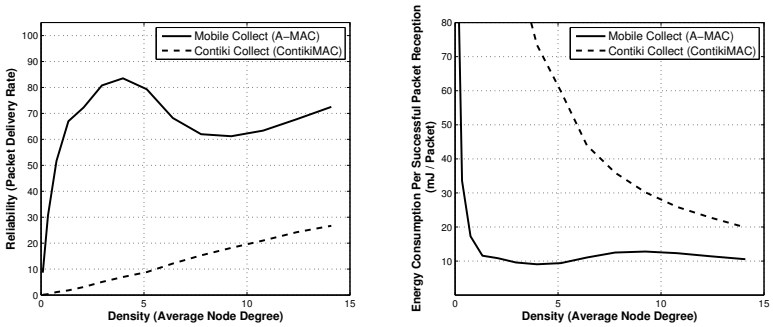
Parent Switch on Timeout: In mobile scenarios, we argue that a routing timeout indicates that the target node has disappeared from the communication range of the sending node. Thus, instead of punishing the timed-out route by slightly increasing its routing metric ETX, as done commonly in routing protocols, we increase the ETX to the maximum value which enforces a parent switch. Hence, the source node disconnects from its parent and (1) connects to alternate parents in its neighbor table, or (2) if no alternative parents are available it uses beaconing as fall-back to discover new parents.

Avoiding Routing Loops: The dynamic topology caused by the mobility of nodes and the agile parent change in Mobile Collect increase the risk of routing loops. Mobile Collect extends Contiki Collect by enabling a node to track the parents of all its neighbors. Thus, nodes in Mobile Collect announce the IDs of their parents in their routing beacons. This allows us to implement classic loop suppression mechanisms such as triangle suppression [9].

Our evaluation shows that we achieve high reliability and energy efficiency for data collection in mobile settings with these lightweight mechanisms.

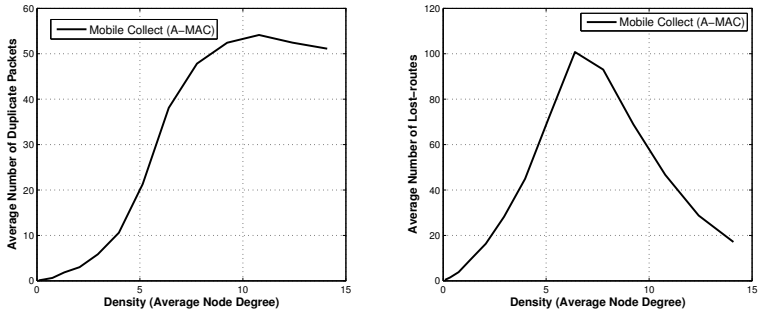
3 Evaluation

We perform simulations in COOJA [10] and use BonnMotion [11] to generate the mobility scenarios. We simulate 50 nodes out of which three are sink nodes.



(a) Reliability (b) Energy per received packet

Fig. 1: Reliability and energy performance of Mobile and Contiki Collect under the random waypoint mobility model. Mobile Collect performs very well.

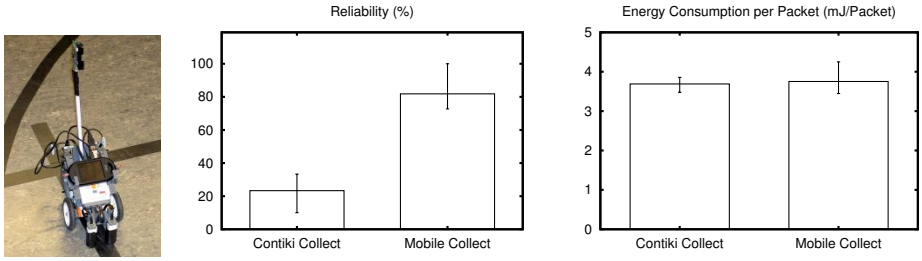


(a) Number of duplicate packets (b) Number of lost routes

Fig. 2: Microbenchmarks that explain the performance of Mobile Collect

We use a random waypoint mobility model with a node speed between 2 and 8 m/s.

Our results are shown in Figure 1. Mobile Collect achieves much higher packet delivery rates at lower energy consumption than Contiki Collect. Figure 1a shows that the packet delivery rate (PDR) first increases, then for node degrees between 4 and 8 decreases before it increases again. For very low densities, the network is sparse, so nodes do not find forwarders to forward their packets to the sink. This improves with higher density, but as shown in Figure 2a the number of duplicate packets then increases which is caused by packets not being acknowledged and larger than triangle loops. The negative trend does not continue for medium node degrees as shown in Figure 2b: the number of lost routes decreases as nodes can more easily find forwarders as the network becomes more dense. Figure 1b shows much better energy consumption per received byte for Mobile Collect than for Contiki Collect mainly because the PDR is much higher and there is less energy



(a) Robot (b) PDR for Robot's Packets (c) Energy per received packet

Fig. 3: Mobile Collect delivers more packets from the mobile robot at the same energy cost as Contiki Collect.

wasted on packets that do not make it to the sink. This is very apparent for Contiki Collect when the network is sparse.

We depict results from a robot sensor network in Figure 3. In this scenario, one robot acting as data source moves from one cluster of static nodes to a second cluster of static nodes. Both clusters have one sink each. Mobile Collect shows much better performance than Contiki Collect since it *(i)* is able to buffer data when moving from the first cluster to the second and *(ii)* with Mobile Collect it is able to find new routes quickly as it comes in range of the second cluster. The packet loss is mainly due to the early determination of the measurements: with Mobile Collect all packets have eventually arrived at the sink.

4 Conclusions

Given the promising results of Mobile Collect in demanding mobility scenarios and on real hardware, we doubt that sensor networks really need MAC layers specifically designed for mobile scenarios.

5 Acknowledgements

This work has been partly funded by SSF, VR, the SICS Center for Networked Systems (CNS), the Uppsala VINN Excellence Center for Wireless Sensor Networks WISENET, and partly supported the FP7 NoE CONET. CNS is funded by VINNOVA, SSF, KKS, ABB, Ericsson, Saab SDS, TeliaSonera, T2Data, Vendolocus and Peerialism. WISENET is funded by VINNOVA, Uppsala University, Banverket, CRL Sweden, FOI, Imego, JonDeTech, Pricer, SenseAir, SICS, TNT-Elektronik, TermoSense, VTT, Upwis, ÅAC Microtec, and WiseNet Holding. The authors thank Gunnar Karlsson (KTH) and Peter Stenlund (Vendolocus) for their input.

References

1. M. Ali, T. Suleman, and Z.A. Uzmi. MMAC: A mobility-adaptive, collision-free mac protocol for wireless sensor networks. In *IPCC*, Phoenix, USA, April 2005.
2. D. Zhang, Q. Li, X. Zhang, and X. Wang. De-ass: An adaptive mac algorithm based on mobility evaluation for wireless sensor networks. In *WiCOM*, 2010.
3. S.C. et al. Choi. An adaptive mobility-supporting mac protocol for mobile sensor networks. In *Vehicular Technology Conference*, May 2008.
4. T. Liu et al. Implementing software on resource-constrained mobile sensors: Experiences with Impala and ZebraNet. In *ACM MobiSys*, June 2004.
5. J.A.B. Link, K. Wehrle, O. Osechas, and J. Thiele. Ratpack: Communication in a sparse dynamic network. In *ACM Sigcomm*, Seattle, USA, August 2008.
6. J. Ko, J. Eriksson, N. Tsiftes, S. Dawson-Haggerty, M. Durvy, JP Vasseur, A. Terzis, A. Dunkels, and D. Culler. Beyond Interoperability: Pushing the Performance of Sensornet IP Stacks. In *ACM SenSys*, Seattle, USA, November 2011.
7. O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *ACM SenSys*, Berkeley, USA, November 2009.
8. P. et al. Dutta. Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless. In *ACM SenSys*, November 2010.
9. C. L. Hedrick. RFC 1058: Routing information protocol, June 1988.
10. F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. Cross-level sensor network simulation with cooja. In *SenseApp*, November 2006.
11. N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn. Bonnmotion: a mobility scenario generation and analysis tool. In *SIMUTools*, March 2010.

Zooming Into Radio Events by Bus Snooping

Zhitao He, Thiemo Voigt

Swedish Institute of Computer Science (SICS), Kista, Sweden

Abstract. In this position paper, we advocate the use of bus snooping to trace radio events. Highly precise and unintrusive, the technique leads to potentially more efficient code and more insightful protocol analysis.

1 Introduction

Communication between Cooperating Objects is typically carried out over a multiple-layer protocol stack. The communication interface resides at the bottom of the stack, and its device driver implements packet transmission and reception routines as well as certain MAC primitives. Performance analysis of the device driver is typically done by insertion of instrumentation code, which logs API calls and interrupt events with a timestamping function provided by the host OS. For example, the Contiki OS's radio driver for the Tmote Sky platform can timestamp incoming packets at a default precision of 2.44 *ms*. Due to CPU loading concerns, the timestamps have a limited resolution; they also incur extra latency in the code execution path. Furthermore, tracing interactions between a pair of communicating motes requires accurate time synchronization, which entails a considerable increase in communication overhead. Contiki's *timesynch* protocol, for example, piggybacks a 3-byte timestamp construct to every data packet.

To alleviate both the precision constraint and the measurement overhead for communication performance analysis, we advocate an unintrusive bus snooping technique that performs event tracing on the communication interface. We attach a logic analyzer to the communication bus, which samples the pins' logic levels at a high rate. A timeline of command strobes, interrupt signals, data bits, and extra test signals can then be constructed over a test run, providing a rich amount of information to the developer for performance analysis or debugging purposes. The ability to simultaneously trace a set of signals, potentially selected from a pair of communicating motes, makes it particularly easy to detect events or event sequences triggered by protocol state transitions. One can zoom into a particular region of the timeline view, either for searching an event sequence or for gauging a code block's processing latency. Furthermore, the logged signal traces can be exported to a data file for advanced offline processing.

We show two use cases of our technique to highlight the productivity enhancements to a communication software developer: optimizing bus latency and automatic gain control analysis. For all experiments, we use a USB logic analyzer [1] to snoop the pins of a pair of Tmote Sky motes running Contiki 2.4.

[1] Saleae Logic 16 logic analyzer. Web page: `http://www.saleae.com/logic16`

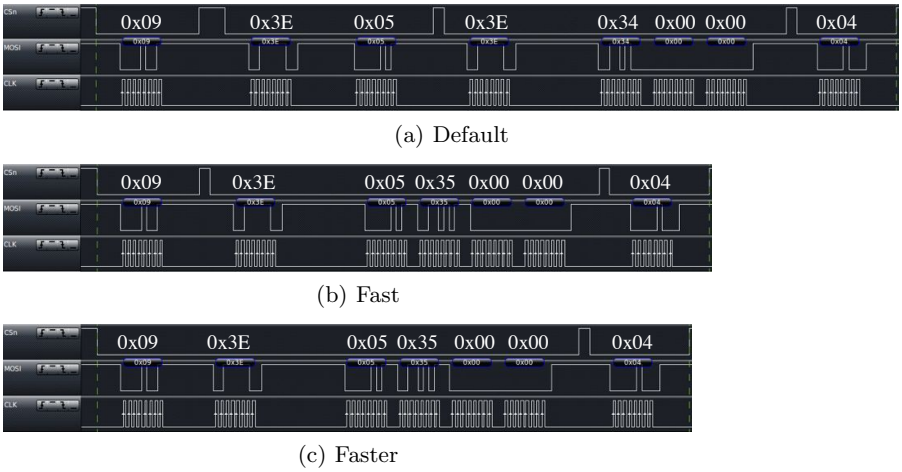
2 Optimizing Bus Latency

We analyze the bus events of a frequent elementary mote operation: packet transmission. A Contiki MAC protocol submits a packet to the radio driver for transmission, by calling a standard *radio_send(len, pkt)* method with a packet size parameter along with a payload pointer. The radio driver constructs a PHY frame by adding headers and footers around the payload, and then transmits the frame. A merit of this generic *radio send* API is that implementation details of PHY frame construction and decoding are completely abstracted away by the device driver. Whether any performance penalty in terms of bus latency is entailed by this layer separation requires analysis of the specific device driver that implements the API. In general, bus latency can be divided into two components: a marginal cost per data unit, as a result of copying between the MCU's packet buffer and the radio's frame buffer; a fixed cost per packet due to signaling overhead. We analyze the performance of the CC2420 driver's *radio send* by snooping the control and data commands over the SPI bus during a call to the method. A trigger function of the logic analyzer takes us right to the beginning of event sequence, from where we can zoom in for precise timing analysis of the trace. Figure 1(a) shows a 80 μs trace section captured over a transmission of a 3-byte packet. The CSn signal is set low during each MCU access to CC2420; The MOSI signal is updated at the rising edges of the CLK signal, indicating serially transferred command/data bytes from the MCU to CC2420. The whole section can be broken down visually into four bus accesses, bounded by the three CSn spikes:

1. Issuing a command strobe to flush the TX FIFO: **0x09** in the figure.
2. Writing a one-byte PHY header to the TX FIFO: **0x3E** followed by **0x05** (PHY payload size).
3. Writing 3 payload bytes to the TX FIFO: **0x3E** followed by 3 data bytes from user pointer. (The 2-byte CRC checksum will be appended automatically by hardware before transmission.)
4. Issuing a command strobe to instruct the radio to start frame transmission: **0x04**.

Step 2 and 3 issue the same **0x3E** command twice, one for writing the 1-byte PHY header and the other for writing the 3-byte payload, which results in a waste of bus bandwidth. Despite that the redundancy might as well be detected by careful code inspection through the CC2420 driver, our intuitive timeline view allows us to further measure the overhead to sub-μs precision, which would be unattainable with software instrumentation. By placing time markers at transition edges of the CSn signal, which signal the beginning and the end for each command access, we arrive at precise latencies of a FIFO write command: the fixed cost is 13.9 μs and the marginal cost per written byte is 6.6 μs. This means a 200% overhead for single byte accesses in this case.

To amortize the fixed cost for a FIFO write, we take full advantage of batch SPI transfers supported by CC2420 by combining step 2 and 3 into a single 4-byte write command. This however obliges allocation of a single, contiguous packet



(a) Default

(b) Fast

(c) Faster

Fig. 1. SPI bus latency for transferring a message of 3 bytes. SPI clock rate = 2 MHz, sampling rate = 16 MHz

buffer for storing both the PHY header and the payload ahead of the FIFO write. Such an optimization violates of the original PHY layer-independent API, but yields a considerable latency reduction, as shown in Figure 1(b). To achieve ultimate bus throughput, we further combine step 1's single-byte flush command strobe with step 2 write command into a continuous command sequence, by removing the chip deselect/select instructions between them, as shown in Figure 1(c). These two measures save us 20.5 μs per packet in total, which corresponds to 82 MCU cycles or 5 bit periods. We can make similar optimizations on the receiver path, reducing one-hop communication latency and maximizing throughput.

3 Events Pattern Mining

Our event analysis technique can also be applied to study random events, such as packet detections dependent on varying radio channel quality. Previous studies in reactive radio jamming have exploited CC2420's Start-of-Frame Delimiter (SFD) detection interrupt as a triggering signal for an eavesdropping jammer to transmit jamming signals [1] [2]. An important limitation of any reactive jammer though is a minimum switching time from listening mode to transmission mode, which sets a lower bound for the size of any jammable packet. A standard IEEE 802.15.4 acknowledgment frame consist of only 6 bytes, which is too short to be jammable by existing reactive jammers based on SFD decoding.

The problem can be alleviated by reducing the switching time, which depends on whether we can find a new triggering signal that becomes available earlier than the SFD interrupt, i.e., some sort of preamble energy indicator. During a

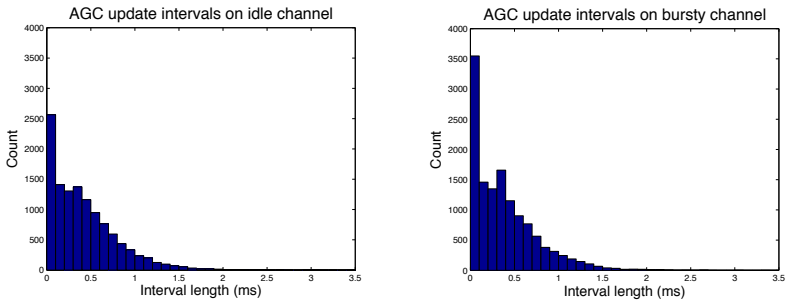
search of such an indicator, we focus our attention to a signal output by the radio receiver's automatic gain control (AGC) circuit. We conjecture that the frequency that the AGC circuit updates its gain correlates somehow to changes in the received signal strength. We configure the CCA pin of CC2420 to output the internal signal *AGC_UPDATE*, which manifests as a high one 16 MHz clock cycle each time the AGC gain is updated. The 4 MHz MCU on Tmote Sky is too slow to capture these narrow 16 MHz spikes. We instead tap a probe from our logic analyzer to the pin, thus are able to capture occurrences of this random signal with a precision of 0.01 μs.

We run a Tmote Sky in idle listening state. We observed 11822 AGC updates already after a brief period of 5 seconds. The time interval between each two consecutive updates ranges between [4.62 μs, 3.45 *ms*], with a mean value of 0.425 *ms*. We show the statistical distribution of this time interval in a histogram in Figure 2(a).

Despite the high frequency of AGC updates and apparently random intervals between them, we want to further investigate whether a packet triggers any extra updates. We configured another Tmote Sky to send a burst of 320 packets, at 64 pkts/s, while repeating the previous measurement on the listening mote. The histogram in Figure 2(b) shows an obvious increase of short update intervals. A close examination at regions in the trace where the SFD pin goes high, which indicates a detected frame, further reveals an interesting pattern of the AGC update intervals: proceeding almost every SFD interrupt by about 120 μs to 160 μs, there are two or more updates that are spaced at 4.62 μs to 20 μs intervals. As this burst pattern occurs during the known period of the 4-byte frame preamble, this burst pattern might qualify as preamble indicator. We set out to design a preamble sensing algorithm based on filtering of the AGC updates by a defined time interval. We test the algorithm's reliability by correlating each preamble prediction to a subsequent detected frame, and derive the prediction rate and false prediction rate. We step down the transmission power over successive test runs, to emulate 6 different signal-to-noise ratios at the receiver. We start with a narrow interval filter of 5 μs, then repeat the tests using a wider filter of 10 μs. Figure 3 shows that the prediction rate is close to 100% at presence of a strong signal, but drops as the signal weakens, while false prediction increases. Comparison between Figure 3(a) and Figure 3(b) shows that the wider filter yields a higher prediction rate for weak signals, albeit with higher likelihood of false predictions.

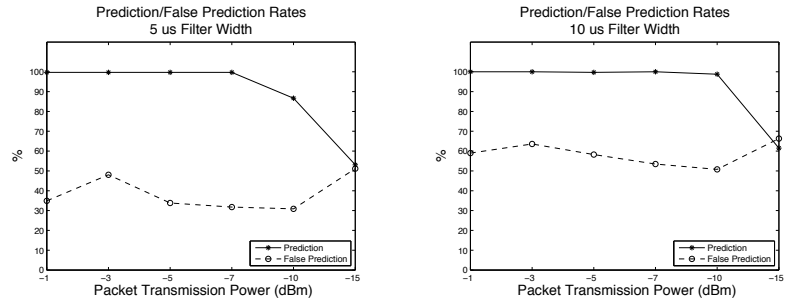
4 Limitations

There are a number of limitations imposed by the use of a logic analyzer. The number of available channels caps the number of concurrent test signals. Our 16-channel logic analyzer thus can monitor at most four 4-wire SPI buses at the same time. The bandwidth capacity of the logic analyzer limits the sampling rate of each channel, which will become an an issue if buses of higher data rates are to be snooped. The trace length is limited by the user's free disk space.



(a) Idle Channel (b) Bursty Channel

Fig. 2. Histogram of AGC updates intervals



(a) (b)

Fig. 3. The filtered AGC update intervals can be used as a predictor signal for an upcoming packet, with a very high prediction rate, given a strong packet signal.

5 Conclusions

In this paper, we show how bus snooping can gain us new insights into low level radio events. This bottom-up approach is precise and unintrusive, compared with conventional software instrumentation. We see potential opportunities to extend the use of this technique for debugging and performance analysis at higher layers.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 224282 and has been partially supported by the FP7 NoE CONET.

References

1. Wood, A., Stankovic, J., Zhou, G.: DEEJAM: Defeating Energy Efficient Jamming in IEEE 802.15.4-based Wireless Networks. Secon (2007)
2. He, Z., Voigt, T.: Precise Packet Loss Pattern Generation by Intentional Interference PWSN (2011)

Towards Collaborative Localization of Mobile Users with Bluetooth

Alexandre Barreira[1,2], Philipp Sommer[3], Brano Kusy[3], and Raja Jurdak[3]

[1] University of Technology of Compiègne
[2] Georgia Institute of Technology
abarreira3@gatech.edu
[3] Autonomous Systems Lab, CSIRO ICT Centre
{philipp.sommer,brano.kusy,raja.jurdak}@csiro.au

Abstract. Location awareness is a key requirement for many pervasive applications. Collaborative localization techniques are interesting because they help to improve accuracy and coverage indoors and improve power consumption by duty-cycling GPS outdoors. We use Bluetooth for collaborative localization of mobile personal devices. Specifically, we embed information in Bluetooth device names to improve latency of information exchange between participating nodes. We identify and demonstrate on real hardware two problems in the Bluetooth stack that negatively impact localization accuracy: a) device name caching that introduces significant device-specific delays in transmitting information between nodes, and b) poor accuracy of time synchronization in modern mobile devices. Our solution is to append additional time information to the device name and track time offsets between nodes. We verify experimentally that this helps to both detect outliers and correct for time-synchronization errors and thus mitigate localization errors.

Keywords: Localization, Bluetooth, Collaboration

1 Introduction

Precise location information serves as a basic building block for many pervasive computing systems enabling services tailored to the current position of mobile users. Location information outdoors is typically obtained from GPS modules on mobile devices, while indoor location information is delivered through specialized location tags that are attached to mobile users. Today's smart phones provide a compelling localization platform as they integrate GPS and multiple communication technologies capable of localizing users indoors (Bluetooth, Wi-Fi, etc). Bluetooth is commonly used for localization due to its pervasiveness in modern office environments [1][3][2]. These techniques typically embed location information in Bluetooth device names, which enables mobile devices to triangulate their location by inferring proximity to a number of infrastructure nodes. The reliance on infrastructure nodes, however, limits the utility of indoor Bluetooth localization in areas with sparse coverage of Bluetooth devices.

This paper proposes the use of Bluetooth device names for collaborative localization of mobile devices both indoors and outdoors. Collaborative Bluetooth localization can increase the density and coverage for indoor scenarios, where sharing location information between mobile devices enables devices further away from infrastructure nodes (laptop or desktop PCs) to more accurately determine their location. In outdoor scenarios, collaborative Bluetooth localization enables sharing of GPS location information among multiple smart phones. By splitting the energy burden of operating the GPS modules among multiple nodes, lifetime of individual nodes increases.

A key building block for collaborative localization using device names is to ensure that location information is shared between mobile devices in a timely manner. If the transmission or reception of location information is delayed, the mobile node might have changed its location significantly which would lead to inaccurate range estimates. Our experiments show that vendor-specific implementations of Bluetooth communication stack can create unpredictable delays in data reception shared through device names. These delays can be as high as tens of seconds which we conjectured to be caused by vendor-specific caching implementations. One simple solution is to embed the current time in device names to detect caching delays. This approach, however, requires all participating devices to be time-synchronized. We observe that modern mobile phones exhibit time offsets in the order of tens of seconds, introducing errors of the same magnitude as the device name caching.

We propose a simple algorithm that detects and corrects both the caching delays and synchronization errors by learning and tracking pairwise clock offsets between neighboring nodes. Our algorithm works under the assumption that the clock drifts are relatively small over short periods of time and that re-synchronization of devices is a relatively infrequent task. Given the accurate time information, we discover and discard stale location information due to device name caching. We evaluate our algorithm in experiments and show that it reduces the errors in device name timestamps to an average of one second, serving as an enabler for collaborative Bluetooth localization.

2 Collaborative Localization using Bluetooth

Bluetooth is a popular wireless interface that enables communication between computers and peripheral devices. The primary way of communicating information over Bluetooth is to establish a connection between devices followed by the actual data exchange. However, the Bluetooth network stack needs to first discover the neighboring devices which leads to delays. As mobile devices might only be within their radio range for a short time, location information is commonly embedded in Bluetooth device names and shared in the discovery phase. Our approach augments the device name with an estimate of the current location (building-specific cartesian coordinates), an estimate of location uncertainty based on the freshness and the algorithm's confidence in the location estimate, and the corresponding timestamp. Nearby devices can employ this information

together with the signal strength reading contained in Bluetooth packets to estimate their current position, which is then again reflected in the device name. Next we give a brief overview of the Bluetooth stack.

Bluetooth network stack. During the inquiry phase a device switches between different frequencies in a pseudo-random manner to transmit and listen for the responses from nearby devices. This will result in a list of device addresses in the inquirer's neighborhood. In a second step, remote devices are asked for their device name, which can contain up to 248 characters. Remote device names can be obtained without the need to establish an explicit connection between the two devices. A device inquiry will also report the received signal strength indication (RSSI) and the device class of each found device.

Bluetooth device name caching. Disseminating information through device names is particularly sensitive to caching of remote device names in the communication stack at the receiver. A cached version of the remote device name will also contain stale location information, negatively impacting the localization accuracy. Unfortunately, smartphone operating systems provide little control over the underlying Bluetooth protocol stack. In particular, a method to flush the cache of remote device names is not available, and therefore, we cannot guarantee that the contents of remote device names are up-to-date. Caching strategies also vary between different versions of the operating system and device models.

Rejection of cached device names. We include the current timestamp into the device name. This allows the receiver to estimate the time offset between the remote device and the local clock. Under the assumption that the clock of mobile phones remains stable over short time intervals, we can calculate a lower bound for the time offset. Consequently, a cached device name will result in a significantly larger time offset and can be discarded before passing it on to the localization algorithm. However, such an approach will not be able to mitigate the effect of small delays introduced by the Bluetooth transmission itself. Note that this simple clock offset estimation algorithm also corrects for global time-synchronization errors.

3 Evaluation

We demonstrate the feasibility of our approach using a setup consisting of two Samsung Nexus S phones placed in close proximity to each other. Both phones are running the Android OS version 2.3.3. Every phone continuously updates its Bluetooth device name once every second with the current local time. At the same time, each phone performs periodic Bluetooth device inquiries, which return the MAC addresses, device names, device classes and RSSI readings of nearby devices. For our evaluation, we use the local clocks of the devices which are only loosely synchronized exhibiting a clock offset of 9.5 seconds.

We measure the offset between the timestamp included in the remote device name and the local clock and plot results in Figure 1. The average offset is 11.2 seconds with a standard deviation of 3.2 seconds. At several instances the

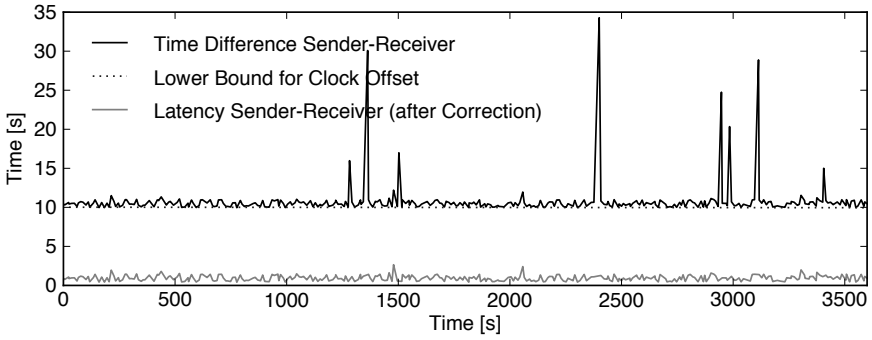


Fig. 1. Time errors observed in data dissemination through Bluetooth device names for Samsung Nexus S. The offset of 10 seconds is due to time-synchronization errors. The occasional spikes up of up to 30 seconds are due to device name caching.

measured offset was significantly larger than the average, indicating a cached version of the remote device name. We believe that this happens when the local device has detected a remote device during the inquiry phase but failed to query the device name. By using a threshold decision which is based on the minimal received offset, we are able to discard outdated location information included in remote device names. The resulting average latency for propagating a device name after applying our correction algorithm is roughly 1.0 seconds with a standard deviation of 0.6 seconds.

4 Conclusions

In this position paper, we describe the use of Bluetooth device names to enable collaborative localization for mobile users. We demonstrated that large time delays can be introduced by local caching of remote device names. We developed a filter that can reject this stale data using timing information embedded in device names. Our experimental results showed that the time errors decrease drastically using our algorithm, thus providing accurate and timely location data for collaborative localization algorithms.

References

1. Bandara, U., Hasegawa, M., Inoue, M., Morikawa, H., Aoyama, T.: Design and implementation of a Bluetooth signal strength based location sensing system. In: Proc. IEEE Radio and Wireless Conference (2004)
2. Raghavan, A.N., Ananthapadmanaban, H., Sivamurugan, M.S., Ravindran, B.: Accurate Mobile Robot Localization in Indoor Environments using Bluetooth. In: Proc. ICRA (2010)
3. Zaafir Barahim, M., Razvi Doomun, M., Joomun, N.: Low-Cost Bluetooth Mobile Positioning for Location-based Application. In: Proc. ICI (2007)

Practical Network Coding in Sensor Networks: Quo Vadis?

Thiemo Voigt[1], Utz Roedig[2], Olaf Landsiedel[3,4], Kasun Samarasinghe[1], Mahesh Bogadi Shankar Prasad[1]

[1] Swedish Institute of Computer Science (SICS), Kista, Sweden
[2] Lancaster University, Great Britain,
[3] Chalmers University of Technology, Sweden
[4] KTH Royal Institute of Technology, Sweden

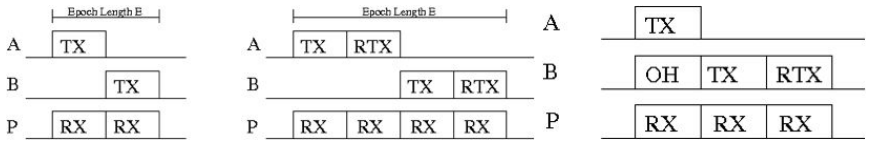
Abstract. Network coding is a novel concept for improving network capacity. This additional capacity may be used to increase throughput or reliability. Also in wireless networks, network coding has been proposed as a method for improving communication. We present our experience from two studies of applying network coding in realistic wireless sensor networks scenarios. As we show, network coding is not as useful in practical deployments as earlier theoretical work suggested. We discuss limitations and future opportunities for network coding in sensor networks.

1 Network Coding in Wireless Sensor Networks

Network Coding was introduced by Ahlswede et al. [1], proving that it can increase multicast capacity. Since then, it has been investigated in several different networked scenarios which demand different traffic characteristics. Most previous research has focused on theoretical aspects of applying network coding to sensor networks. There are, however, also more practical examples of applying network coding in wireless networks. Network coding has, for example, demonstrated its usefulness in the general networking domain (e.g., Katti et al. [2]). Hou et al. have proposed AdapCode [3], a network-coded variant of the Deluge protocol [4]. Aoun et al. specifically target real-time wireless sensor networks [5]. They use network coding together with packet skipping to improve on-time goodput in real-time WSNs. Their topology, however, is not applicable to data collection.

While network coding for data dissemination can increase reliability, it is harder to apply network coding for convergecast, probably the most important traffic paradigm in wireless sensor networks where data is collected from multiple sources and transported to one or more data sinks.

We have worked on two efforts to apply network coding in wireless sensor networks. In the first study, we have applied network coding to extend GinMAC, a state-of-the-art MAC protocol [6]. In the second, we have tried to apply network coding for reliable and low delay communication in network with very high packet loss rates [7]. In the second case it has turned out that network coding cannot live up to the expectations while in the first case in all scenarios we studied, network



(a) Schedules without NC (0 and 1 retransmissions) (b) Schedule with NC

Fig. 1: Schedules with and without Network Coding (NC). NC compresses the schedule while still enabling retransmissions.

coding can be replaced with simple overhearing and the retransmission of the lost packets which avoids the complexity of network coding. In the remainder of this paper, we detail our studies and discuss limitations and usefulness of network coding in sensor networks.

2 Network Coding in GinMAC

We have applied network coding in GinMAC [6], the MAC layer of the GINSENG system designed for industrial monitoring and control. In industrial process automation and control networks it is common that many nodes monitoring the same production process and are hence deployed in close proximity to each other. Thus, most nodes are in interference range of all other nodes in the network and many nodes are also capable of overhearing packets. GinMAC is a single channel state of the art TDMA-based MAC layer for performance-controlled sensor networks [8]. GinMAC uses an offline process to dimension a network before deployment. The heart of the dimensioned network is a TDMA schedule with an epoch length consisting of E slots. There are three types of slots, namely *basic* slots for transmission (TX) and reception (RX), *additional* slots for retransmissions (RTX) to increase reliability and *unused* slots to decrease the duty cycle. The schedule determines latency, power consumption and reliability. The latter is increased by adding more retransmission slots.

We employ overhearing as a mechanism for tuning network reliability and latency. Due to external interference a node may not be able to deliver a packet to a parent node. However, a neighbouring node may be able to overhear the packet and then try to deliver the packet on its behalf. Thus, instead of the sender retransmitting packets, neighbouring nodes that overhear the initial transmission take care of the retransmission. Obviously, a node that overhears packets will now need capacity to transmit these. In order to avoid allocation of additional transmission slots we propose network coding to combine several overheard packets into a single retransmission. Thus, we use overhearing and network coding for retransmissions reducing the overall number of transmission slots.

Figure 1 shows the resulting schedules with and without network coding for a binary tree with the leave nodes A and B. In the example node A has the first transmission slot. With network coding node B overhears A's transmission in this transmission slot. Thereafter, node B transmits its own packet in its

allocated slot. Finally, node B uses the retransmission slot in which it transmits a network coded packet that contains the information of the packet transmitted by A and B. For example, packets from A and B can be combined using simple binary XOR coding. The root node will receive A and B's packets if one original transmission is received and the network coded retransmission is received (both packets are also received if both original packet transmissions are received).

As a result we have now constructed a schedule in which both nodes A and B have a way to retransmit a packet despite that we provision only one retransmission slot. Thus, with network coding less slots are provisioned while we still provide a retransmission option for all nodes. The slight drop in reliability compared to the original scheme with one retransmission may be well acceptable for some application scenarios since a gain in transmission latency is achieved as the schedule is compressed. Figure 1 shows that the schedule compression for the binary tree is $3:4$ (0.75). In our previous work we have shown that we achieve approximately similar error packet reception rates for GinMAC with and without network coding and that for high packet loss rates we can also achieve a reduction in power consumption [6].

3 Network Coding for Reliable, Low-delay Communication

Many sensor network applications, in particular sensor actuator networks, require a certain delay and reliability to enable control loops on top of a wireless sensor network. Network coding promises highly reliable communication. We discuss our lessons learned from integrating it into a collection tree application [7]. While network coding increased reliability in some scenarios, we noticed the following two key limitations: (1) strongly increased delay, and (2) high overhead due to limited lack of adaptability:

Delay: Commonly, WSN applications operate at low data rates: often, a node generates packets in the order minutes and the whole network traffic is in the order of a couple of packets per second. However, to achieve a robust interleaving and mixing of packets for network coding, one commonly applies a coding scheme across multiple packets. Thus, nodes have to wait for these packets to accumulate, which significantly increases delay. For example, in today's collection protocols such as CTP [9] or Contiki Collect [10], one sees typical network delays in the order of milliseconds. However, waiting for enough packets for coding to accumulate takes a couple of seconds. Thus, the delay added by network coding is several order of magnitude higher than the network delay. Hence, network coding seems only beneficial when the applications do not have delay constraints.

Overhead and lack of adaptability: As network coding is an end-to-end scheme, its feedback is limited when compared to per-hop acknowledgments. Thus, it is difficult for the source to adjust its coding rate to changing network conditions. Reflecting this, we had to employ a redundancy level in our coding scheme that matches the maximum expected loss rate. As a result, this approach

adds a large overhead compared to the average scenario. Especially, when keeping in mind, that links in WSNs are often bursty [11], leading to high, short-term packet losses.

4 Limitations and Applicability of Network Coding

In the previous section we have already outlined some limitations of applying network coding for data collection. While our results with GinMAC (see Section 2) are encouraging, we later realized that we do not need network coding in this scenario. If either packet A or packet B is lost, we can simply retransmit the lost packet. If both packets A and B are lost, then network coding does not help either and it is better to retransmit only one of the lost packets. We have not found larger topologies where we could use network coding while it was not possible to do the same with overhearing and standard packet retransmissions only. For the single-sink scenario, we have found one use of network coding: when we do not apply acknowledgements which might be useful when the schedule needs to be compressed to a minimum but still some retransmissions should be enabled. In particular for trees where each node can have more than one sibling (e.g., ternary trees), network coding may be useful when acknowledgements are not enabled. Furthermore, in multi-sink scenarios, similar to the original network coding scenarios, we expect network coding to be beneficial.

5 Conclusions

In this paper, we have demonstrated some drawbacks of applying network coding in real-world sensor network scenarios. While we have pointed out some limitations we have also shown some potential use cases.

6 Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 224282, SSF and VR. This work has been partially supported by CONET, the Cooperating Objects Network of Excellence.

References

1. R. Ahlswede, N. Cai, S. Li, and R. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 2000.
2. S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft. Xors in the air: practical wireless network coding. In *SIGCOMM*, 2006.
3. I. H. Hou, Y. E. Tsai, T. F. Abdelzaher, and I. Gupta. AdapCode: Adaptive Network Coding for Code Updates in Wireless Sensor Networks. In *IEEE INFOCOM*, 2008.

4. J. W. Hui and D. Culler. The dynamic behavior of a data dissemination protocol for network programming at scale. In *SenSys'04*, November 2004.
5. M. Aoun, A. Argyriou, and P. van der Stok. Performance evaluation of network coding and packet skipping in ieee 802.15.4-based real-time wireless sensor networks. In *EWSN*, 2011.
6. K. Samarasinghe, T. Voigt, L. Mottola, and U. Roedig. Poster abstract: Network coding with limited overhearing. In *EWSN*, 2011.
7. M. Bogadi SP. Reliable, low delay communication in wireless sensor networks. Master's thesis, KTH, Stockholm, 2011.
8. J. Brown, B. McCarthy, U. Roedig, T. Voigt, and C. Sreenan. Burstprobe: Debugging time-critical data delivery in wireless sensor networks. In *European Conference on Wireless Sensor Networks (EWSN)*, Bonn, Germany, February 2011.
9. O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection Tree Protocol. In *ACM SenSys*, 2009.
10. J. Ko, J. Eriksson, N. Tsiftes, S. Dawson-Haggerty, M. Durvy, JP Vasseur, A. Terzis, A. Dunkels, and D. Culler. Beyond Interoperability: Pushing the Performance of Sensornet IP Stacks. In *ACM SenSys*, Seattle, USA, November 2011.
11. K. Srinivasan, M. Kazandjieva, S. Agarwal, and P. Levis. The β-factor: measuring wireless link burstiness. In *ACM SenSys*, 2008.

Accurate Sensor Node Energy Consumption Estimation Using EdiMote Prototyping Platform

Rinalds Ruskuls[1,2], Girts Strazdins[1,2], and Leo Selavo[1,2]

[1] Institute of Electronics and Computer Science, Riga, Latvia
[2] Faculty of Computing, University of Latvia, Riga, Latvia
{firstname.lastname}@edi.lv,

Abstract. Wireless sensor networks and other battery powered embedded systems often require long lifetime. To assess application efficiency, accurate energy estimation tools are required. Although software energy profiling provides scalable network simulation opportunities, it lacks accuracy. In this paper we propose a hardware-based energy measurement approach using EdiMote prototyping platform. Evaluation shows, that it is more accurate compared to software profiling while providing extra information about energy consumption of separate system modules: MCU and radio chip.

Keywords: energy consumption estimation, embedded systems, wireless sensor networks, prototyping

1 Introduction

Wireless sensor networks are highly diverse: different hardware platforms, software solutions, and applications are used. However, most of them do have one aspect in common: long lifetime is desired, usually in the order of weeks, months or even years. The requirement is motivated by limited battery capacity, large sensor node count and limited node physical accessibility.

To plan a deployment, lifetime must be estimated already in the development and testing phase. In this paper we propose an approach for sensor node energy metering, using EdiMote [4]: a prototyping system for cooperating object hardware platform incremental development and debugging (Section 3). EdiMote energy consumption estimation is compared to software energy profiling with Contiki powertrace [1]. Results show, that EdiMote is capable of measuring energy consumption with higher accuracy, while providing extra information about energy distribution between individual modules: MCU and radio. (Section 5).

2 Related work

One approach to energy estimation is simulations and software profiling, such as Contiki Powertrace [1]. Time of operation is logged for each sensor node module. It is then multiplied by energy consumption constants from chip datasheets. Advantage of this approach is usability in large scale simulations. However, the

accuracy is limited: simple implementations lack details of real deployment operation specifics and more complex chip emulations may impose high computational overhead.

Other method class is to measure physical values from sensor nodes by converting consumed current I to voltage V (with shunt resistor R). Voltage can be analyzed by using 1) analog to digital converter (ADC) or 2) voltage to frequency converter (VFC). The precision of the first method is determined by conversion circuit accuracy, and ADC resolution and sampling rate. The second method requires two counters: one reflects the used energy and another is used as reference to measure time span. In contrast to ADC, VFC method is suitable for long term estimation, while the accuracy in short term is lower.

Alternative approach is to count voltage oscillation cycles of switching regulator supplying the whole system [2]. However, this approach is not suited for separate module energy metering, as it requires single switching regulator and a MCU counter input pin for each module.

3 Our approach

Typical sensor node consists of standard modules: MCU, radio, sensors and external memories. EdiMote [4] allows to build a sensor node prototype by plugging separate modules on the main board and configuring the interconnections using a central programmable logic block (CPLD), Figure 1. Several predefined modules are available (MSP430F1611 MCU, CC2420 radio, Bluetooth, M25P80 flash) and any custom module is usable. EdiMote allows to measure energy consumption in real time, separately for each module: MCU, radio and sensors. In addition, it provides analog and digital signal interconnection and monitoring capabilities, not described in this paper. Such approach provides information for identifying deficiencies in new hardware platform prototypes and software solutions.

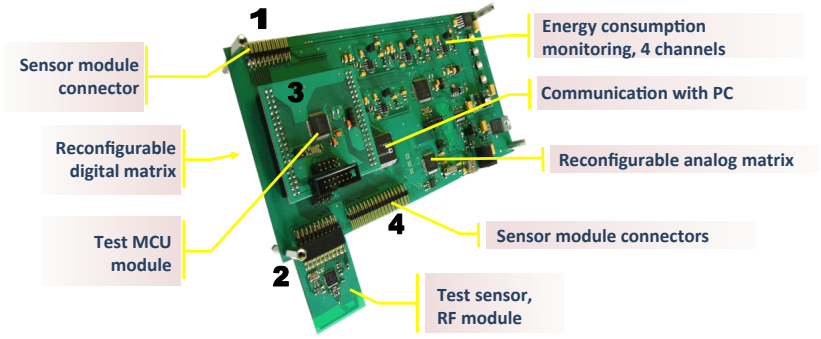


Fig. 1: EdiMote prototyping platform

EdiMote applies SPOT architecture [3] with VFC to provide accurate long term energy profiling. As the evaluation shows, the SPOT modules are also capable of measuring short term energy usage with high accuracy.

4 Test setup

Slightly modified Contiki `radio-test` application was used as test subject. LED indicators were switched off and null mac protocols were used to ensure determined and comparable application behavior between real TMote Sky nodes and its EdiMote prototype. Application contained one second long loop consisting of four different operation modes:

1. Sense: temperature sensor (connected to ADC) is polled 100 times ($\approx 14ms$)
2. Send: 100 packets with 20 byte payload are sent ($\approx 213ms$)
3. Receive: radio receive mode is turned on ($200ms$)
4. Sleep: the remainder of the second is spent in low-power mode ($\approx 570ms$)

Power consumption was measured using three different methods:

1. Shunt resistor voltage measurement using oscilloscope andpost-processing. Application was running on real TMote Sky sensor node. 10 test runs for each mode were analyzed.
2. Contiki powertrace software profiling. Application was running on real TMote Sky sensor node. Statistics were transmitted to PC (over UART) for post-processing. One hour long log was recorded (3600 application cycles).
3. EdiMote SPOT energy meters. The same Contiki application was running on simulated TMote Sky on the EdiMote platform. As in the second approach, one hour long log was recorded for each mode.

The first method using oscilloscope was considered *ground-truth*, and the energy consumption estimation accuracy of the other two methods evaluated against it.

5 Results and conclusions

Obtained energy consumption estimation using the three methods are displayed in Figure 2. It shows, that EdiMote approach provides significantly more reliable energy consumption estimation compared to pure software profiling. Multiple conclusions can be drawn:

1. Software energy accounting is reasonable in application modes with stable consumption pattern (radio idle listen, sensor sampling)
2. In dynamic application modes, such as radio transmission, energy estimation using SPOT provides higher accuracy compared to software methods
3. EdiMote SPOT energy profiling is a helpful tool in software debugging. In the particular test application it helped to identify ineffective MCU sleep mode, where ADC and UART modules were consuming significant amount of energy in idle mode (see Table 1)

6 Future work

To improve EdiMote, next steps are to implement oscilloscope-like short term energy metering with improved time resolution (using ADC) and automated energy consumption logging - different mode switching from the test software using GPIO pins on the test MCU.

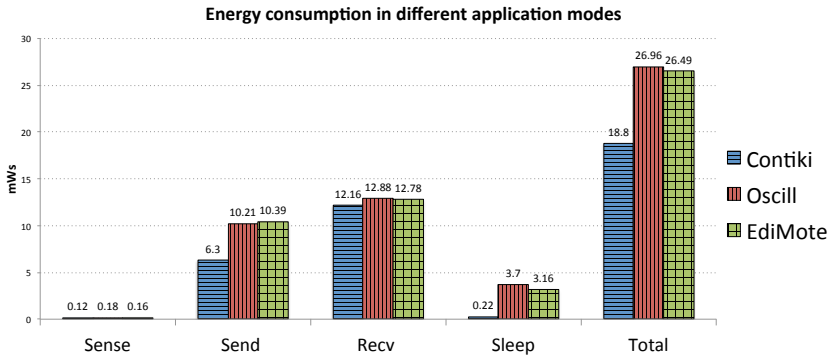


Fig. 2: Energy estimation in different application modes using three different metods: oscilloscope, Contiki software profiling and EdiMote SPOT modules.

Table 1: Energy consumption of MCU and radio modules in original test application and modified version with explicitly turning ADC and UART off

Application	MCU, mWs	Radio, mWs	Total, mWs
Original	2.448	0.769	3.156
ADC&UART off	0.758	0.323	1.071

Acknowledgement

This work has been supported by European Social Fund, grants Nr. 2009/0219/ 1DP/1.1.1.2.0/APIA/VIAA/020 and Nr. 2009/0138/ 1DP/1.1.2.1.2/09/IPIA/ VIAA/004. The authors would like to thank Gatis Supols, Arturs Selivanovs and Aivars Severdaks for help and advice in EdiMote design and development.

References

1. Dunkels, A., Eriksson, J., Finne, N., Tsiftes, N.: Powertrace: Network-level power profiling for low-power wireless networks. Tech. rep., SICS (2011)
2. Dutta, P., Feldmeier, M., Paradiso, J., Culler, D.: Energy metering for free: Augmenting switching regulators for real-time monitoring. In: IPSN '08. pp. 283–294 (2008)
3. Jiang, X., Dutta, P., Culler, D., Stoica, I.: Micro power meter for energy monitoring of wireless sensor networks at scale. In: IPSN'07. pp. 186–195 (2007)
4. Ruskuls, R., Selavo, L.: Edimote: A flexible sensor node prototyping and profiling tool. In: REALWSN'10, pp. 194–197 (2010)