A Multifrequency MAC Specially Designed for Wireless Sensor Network Applications

GANG ZHOU, College of William and Mary, USA
VALERIE BÉRANGER, Inria Paris-Rocquencourt, France

Multifrequency media access control has been well understood in general wireless ad hoc networks, while in wireless sensor networks, researchers still focus on single frequency solutions. In wireless sensor networks, each device is typically equipped with a single radio transceiver and applications adopt much smaller packet sizes compared to those in general wireless ad hoc networks. Hence, the multifrequency MAC protocols proposed for general wireless ad hoc networks are not suitable for wireless sensor network applications, which we further demonstrate through our simulation experiments. In this article, we propose MMSN, which takes advantage of multifrequency availability while, at the same time, takes into consideration the restrictions of wireless sensor networks. Through extensive experiments, MMSN exhibits the prominent ability to utilize parallel transmissions among neighboring nodes. When multiple physical frequencies are available, it also achieves increased energy efficiency, demonstrating the ability to work against radio interference and the tolerance to a wide range of measured time synchronization errors.

ACM Reference Format:
Gang Zhou and Valerie Béranger. 2010. A Multifrequency MAC Specially Designed for Wireless Sensor Network Applications. *ACM Trans. Graph.* 9, 4, Article 39 (March 2010), **??** pages. https://doi.org/0000001.0000001_2

1 INTRODUCTION

As a new technology, Wireless Sensor Networks (WSNs) has a wide environment monitoring, smart buildings, medical care, industrial and military applications. Among them, a recent trend is to develop commercial sensor networks that require pervasive sensing of both environment and human beings, for example, assisted living

> "For these applications, sensor devices are incorporated into human health related information like EKG readings, fall detection, and voice recognition".

While collecting all these multimedia information sensor devices only provide very limited bandwidth in a single

In this article, we propose MMSN, abbreviation for Multifrequency Media access control for wireless Sensor Networks. The main contributions of this work can be summarized as follows.

Authors' addresses: Gang Zhou, College of William and Mary, 104 Jamestown Rd, Williamsburg, VA, 23185, USA, gang_zhou@wm.edu; Valerie Béranger, Inria Paris-Rocquencourt, Rocquencourt, France, beranger@inria.fr.

© 2010 Association for Computing Machinery.
This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *ACM Transactions on Graphics*, https://doi.org/0000001.0000001_2.

- To the best of our knowledge, the MMSN protocol is the first multifrequency MAC protocol especially designed for WSNs, in which each device is equipped with a single radio transceiver and the MAC layer packet size is very small.
- Instead of using pairwise RTS/CTS frequency negotiation we propose lightweight frequency assignments, which are good choices for many deployed comparatively static WSNs.
- We develop new toggle transmission and snooping techniques to enable a single radio transceiver in a sensor device to achieve scalable performance, avoiding the nonscalable "one

2 FEATURE EXTRACTION

Apart from the audio features that were included in the million song dataset (acousticness, tempo, instrumentalness, liveness, speechiness, valence, danceability) we also extracted the year and popularity from the Spotify API data. We also crafted 3 additional features using the modeling techniques described below.

2.1 Latent Dirichlet Allocation

We chose all playlists for which we had an overlap of at least 30 songs with our dataset. We then tokenized and removed all stop words from the lyrics of the songs from every playlist, in order for the playlists to be treated as documents and the lyrics as words in the latent Dirichlet allocation model. Latent Dirichlet allocation is a generative statistical model that posits that the lyrics from every playlist can be explained by a fixed number of unobserved groups, which would explain similarity between some playlists. In our case, we chose the number of common topics to be 3, and the process that the generative model describes is the following:
For the M playlists, each of length N_i we have the following parameters and distributions:

(1) Probability $\theta_i \sim Dir(\alpha)$, where $Dir(\alpha)$ is a Dirichlet distribution with parameter α and $i \in 1, ..., M$
(2) Probability $\phi_k \sim Dir(\beta)$, where $k \in 1, 2, 3$ is the index of the topic.
(3) For each word in the lyrics from all playlists, for i, j, where $i \in 1, ..., M$ and $j \in 1, ..., N_i$:
    (a) Chose a topic $z_{i,j} \sim Multinomial(\theta_i)$
    (b) Chose a word $w_{i,j} \sim Multinomial(z_{i,j})$

The probabilities that were output from the model for each of the three topics, were subsequently used as features by the final model.

$$P(t) = \frac{b^{\frac{t+1}{T+1}} - b^{\frac{t}{T+1}}}{b-1}, \quad (1)$$

where $t = 0, \ldots, T$, and b is a number greater than 1.

$$i = \lfloor (T+1) \log_b[\alpha(b-1)+1] \rfloor.$$

It can be easily proven that the distribution of i conforms to Equation

controls[1] for frequency negotiation and reservation are not suitable for WSN applications, even though they exhibit good performance in general wireless ad-hoc networks.

2.1.1 Exclusive Frequency Assignment. In exclusive frequency assignment, nodes first exchange their IDs among two communication hops so that each node knows its two-hop neighbors' IDs. In the second broadcast, each node beacons all neighbors' IDs it has collected during the first broadcast period.

Eavesdropping. Even though the even selection scheme leads to even sharing of available frequencies among any two-hop neighborhood, it involves a number of two-hop broadcasts. To reduce the communication cost, we propose a lightweight eavesdropping scheme.

2.2 Hidden Markov Model on timbre segments

Timbre is defined as the perceived sound quality of a musical note, sound or tone. The MSD dataset contains the time sequence of the timbre feature as a vector of 12 unbounded values centered around 0. These values represent different characteristics of the spectral surface, ordered by degree of importance. For example, the first dimension represents the average loudness of the segment, the second one describes brightness, the third one describes the flatness of a sound, the fourth describes sounds with a stronger attack etc. We averaged the vector features for every time segment for each song, in order to have a time series of the actual timbre of each segment. We subsequently trained a hidden Markov model on the timbre sequence of a random sample of 5,000 songs. The model is fit using the EM algorithm which is a gradient-based optimization method and can therefore get stuck in local optima. For this reason we fit the model with various initializations and selected the highest scoring one. The inferred optimal hidden states of the timbre segments of all songs were predicted by the model, employing the Viterbi algorithm. For each song the optimal hidden states were averaged to provide a single description of the path followed, which was used as an additional feature by the final model. For the training data the average value of the timbre which had a hidden value of 1 was 4.96, for hidden value of 1 they had an average value of the timbre of 13.76 and for a hidden value of 2 an average of -5.35.

[1]RTS/CTS controls are required to be implemented by 802.11-compliant devices. They can be used as an optional mechanism to avoid Hidden Terminal Problems in the 802.11 standard and

ALGORITHM 1: Frequency Number Computation

Input: Node α's ID (ID_α), and node α's neighbors' IDs within two communication hops.
Output: The frequency number ($FreNum_\alpha$) node α gets assigned.

```
index = 0; FreNum_α = -1;
repeat
    Rnd_α = Random(ID_α, index);
    Found = TRUE;
    for each node β in α's two communication hops do
        Rnd_β = Random(ID_β, index);
        if (Rnd_α < Rnd_β) or (Rnd_α == Rnd_β and
         ID_α < ID_β);
         then
            Found = FALSE; break;
        end
    end
    if Found then
        FreNum_α = index;
    else
        index ++;
    end
until FreNum_α > −1;
```

2.3 Sentiment analysis

number, each node calculates a random number (Rnd_α) for itself and a random number (Rnd_β) for each of its two-hop neighbors with the same pseudorandom number generator.

Bus masters are divided into two disjoint sets, $\mathcal{M}_{RT}$ and $\mathcal{M}_{NRT}$.

RT Masters $\mathcal{M}_{RT} = \{\vec{m}_1, \ldots, \vec{m}_n\}$ denotes the n RT masters issuing real-time constrained requests. To model the current request issued by an $\vec{m}_i$ in $\mathcal{M}_{RT}$, three parameters—the recurrence time (r_i), the service cycle (c_i), and the relative deadline (d_i)—are used, with their relationships.

NRT Masters $\mathcal{M}_{NRT} = \{\vec{m}_{n+1}, \ldots, \vec{m}_{n+m}\}$ is a set of m masters issuing nonreal-time constrained requests. In our model, each $\vec{m}_j$ in $\mathcal{M}_{NRT}$ needs only one parameter, the service cycle, to model the current request it issues.

Here, a question may arise, since each node has a global ID. Why don't we just map nodes' IDs within two hops into a group of frequency numbers and assign those numbers to all nodes within two hops?

3 MODELING

Two different methodologies were pursued. In the first one we tried different classification algorithms both for a single playlist prediction and multiclass prediction for 26 playlists. In the second

(1) Load state into microcontroller model.
(2) Determine assignments needed for resolving nondeterminism.
(3) For each assignment.

Fig. 1. Code before preprocessing.

(a) either call interrupt handler or simulate effect of next instruction, or
(b) evaluate truth values of atomic propositions.

(4) Return resulting states.

controls an automotive power window lift. The program is one of the At first sight, the programs looks like an ANSI C program. It contains function calls, assignments, if clauses, and while loops.

3.1 Classification

We first applied binary classification using a single playlist ('60s, 70s, 80s Classic Rock') as the target and randomly selecting songs that didn't belong to it as well. The training set consisted of 98 songs, while the test set included 34 songs. After standardizing the features we performed grid search on the hyper-parameters for a logistic regression and support vector machine classifier. For logistic regression the regularization strength was fine tuned employing 10-fold cross-validation, while for the support vector classifier the regularization was also optimized using grid search. The other parameters that were chosen during cross validation were the kernel (linear or exponential). For the exponential kernel $e^{-\gamma\|x-x'\|^2}$ the γ parameter was also optimized.

The objective of variable coalescence-based offset assignment is to find both the coalescence scheme and the MWPC on the coalesced graph. We start with a few definitions and lemmas for variable coalescence.

Definition 3.1 (Coalesced Node (C-Node)). A C-node is a set of live ranges (webs) in the AG or IG that are coalesced. Nodes within the same C-node cannot interfere with each other on the IG. Before any coalescing is done, each live range is a C-node by itself.

Definition 3.2 (C-AG (Coalesced Access Graph)). The C-AG is the access graph after node coalescence, which is composed of all C-nodes and C-edges.

LEMMA 3.3. *The C-MWPC problem is NP-complete.*

PROOF. C-MWPC can be easily reduced to the MWPC problem assuming a coalescence graph without any edge or a fully connected interference graph. Therefore, each C-node is an uncoalesced live range after value separation and C-PC is equivalent to PC. A fully connected interference graph is made possible when all live ranges interfere with each other. Thus, the C-MWPC problem is NP-complete. □

LEMMA 3.4 (LEMMA SUBHEAD). *The solution to the C-MWPC problem is no worse than the solution to the MWPC.*

PROOF. Simply, any solution to the MWPC is also a solution to the C-MWPC. But some solutions to C-MWPC may not apply to the MWPC (if any coalescing were made). □

4 RESULTS

Results for the two main methodologies followed are presented below.

4.1 Classification

The logistic regression model achieved training accuracy of 95.9% and CV accuracy of 94.8%. The SVC outperformed the logistic regression model, with the best model achieving training accuracy of 96.9% and CV accuracy of 95.9%. The best model's parameters used a linear kernel with regularization. The final test accuracy for the SVC was 94.1%. . The learning curve for the svm model is presented below in figure **??** and it shows that it generalizes well over unseen data.

The validation curve for the SVC model can be seen below in figure **??**, which justifies how the particular value for the regularization hyper-parameter was chosen. The cross-validation accuracy decreases as the regularization parameter C increases even though the training accuracy keeps increasing, which would indicate overfitting.

The confusion matrix for the SVC model on the test data can also be seen below in figure **??**.

The ROC curve is also plotted below in figure **??** which shows an excellent area under the curve.

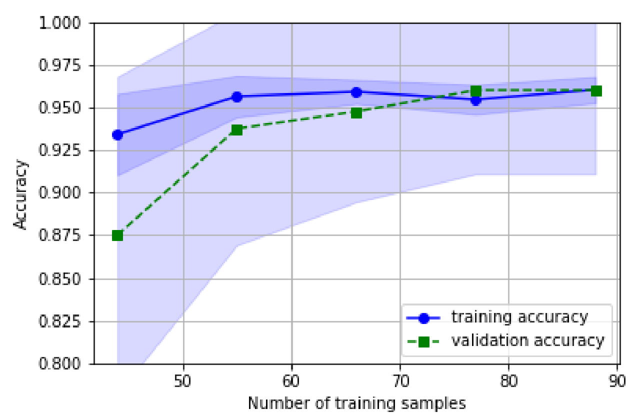

Fig. 2. Learning curve for SVC

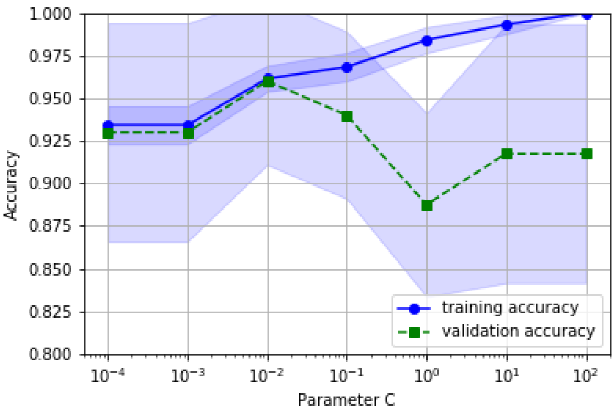

Fig. 3. Validation curve for SVC

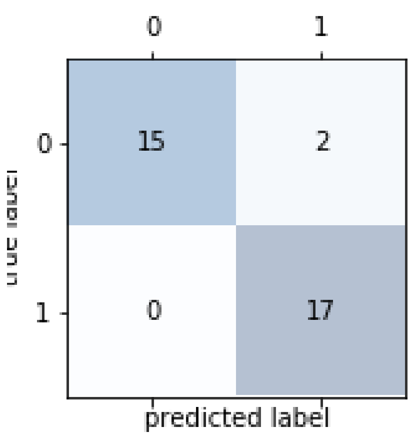

Fig. 4. Confusion matrix for SVC

Table 1. SVC performance on single playlist

Test accuracy	94.1%
Recall	100%
Precision	88.2%
F1-score	93.7%

Precision scores for SVC model

We also applied multi-class classification for 27 playlists for which we had more than 30 songs in our dataset. With this modeling technique called one-vs-all classification, one classifier is fitted for each class against all of the other classes. The training set consisted of 1265 songs, while the test set included 317 songs. Given the superior performance of the SVC model on one playlist, we optimized it in the multi-class setting using grid search and cross-validation.

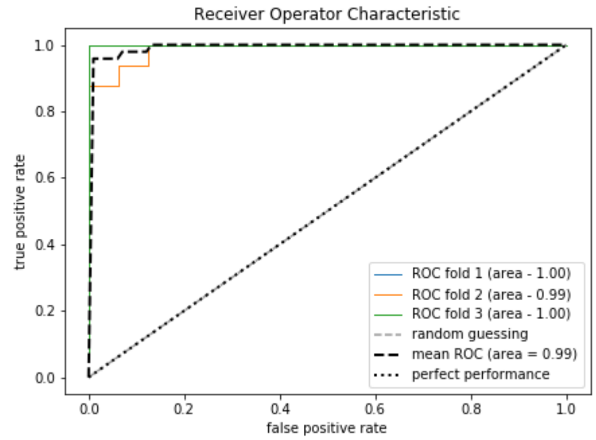

Fig. 5. ROC curve for SVC

5 CONCLUSIONS

In this article, we develop the first multifrequency MAC protocol for WSN applications in which each device adopts a single radio transceiver. The different MAC design requirements for WSNs and general wireless ad-hoc networks are compared, and a complete WSN multifrequency MAC design (MMSN) is put forth. During the MMSN design, we analyze and evaluate different choices for frequency assignments and also discuss the nonuniform back-off algorithms for the slotted media access design.

6 TYPICAL REFERENCES IN NEW ACM REFERENCE FORMAT

A couple of citations with DOIs: [**??**].

A SWITCHING TIMES

In this appendix, we measure the channel switching time of Micaz alternatingly switches between Channels 11 and 12. Every time after the node switches to a channel, it sends out a packet immediately and then changes to a new channel as soon as the transmission is finished. We measure the number of packets the test mote can send in 10 seconds, denoted as N_1. In contrast, we also measure the same value of the test mote without switching channels, denoted as N_2. We calculate the channel-switching time s as

$$s = \frac{10}{N_1} - \frac{10}{N_2}.$$

By repeating the experiments 100 times, we get the average channel-switching time of Micaz motes: 24.3 μs.

B SUPPLEMENTARY MATERIALS

See the supplementary materials in the online version

ACKNOWLEDGMENTS

The authors would like to thank Dr. Maura Turolla of Telecom Italia for providing specifications about the application scenario.

The work is supported by the National Natural Science Foundation of China under Grant No.: 61273304_a and Young Scientists' Support Program (http://www.nnsf.cn/youngscientists).

Received February 2007; revised March 2009; final version June 2009; accepted July 2009