A Multifrequency MAC Specially Designed for Wireless Sensor Network Applications

GANG ZHOU, College of William and Mary
YAFENG WU, University of Virginia
TING YAN, Eaton Innovation Center
TIAN HE, University of Minnesota
CHENGDU HUANG, Google
JOHN A. STANKOVIC, University of Virginia
TAREK F. ABDELZAHER, University of Illinois at Urbana-Champaign

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.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Wireless sensor networks, media access control, multi-channel, radio interference, time synchronization

ACM Reference Format:
Zhou, G., Wu, Y., Yan, T., He, T., Huang, C., Stankovic, J. A., and Abdelzaher, T. F. 2010. A multifrequency MAC specially designed for wireless sensor network applications. ACM Trans. Embedd. Comput. Syst. 9, 4, Article 39 (March 2010), 3 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

As a new technology, Wireless Sensor Networks (WSNs) has a wide range of applications [**?**; **?**; **?**], including environment monitoring, smart buildings, medical care, industrial and military applications. Among them, a recent trend is to develop commercial

This work is supported by the National Science Foundation, under grant CNS-0435060, grant CCR-0325197 and grant EN-CS-0329609.
Author's addresses: G. Zhou, Computer Science Department, College of William and Mary; Y. Wu and J. A. Stankovic, Computer Science Department, University of Virginia; T. Yan, Eaton Innovation Center; T. He, Computer Science Department, University of Minnesota; C. Huang, Google; T. F. Abdelzaher, Computer Science Department, University of Illinois at Urbana-Champaign.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

sensor networks that require pervasive sensing of both environment and human beings, for example, assisted living [**?**; **?**; **?**] and smart homes [**?**; **?**; **?**].

> "For these applications, sensor devices are incorporated into human cloths [**?**; **?**; **?**; **?**] for monitoring health related information like EKG readings, fall detection, and voice recognition".

While collecting all these multimedia information [**?**] requires a high network throughput, off-the-shelf sensor devices only provide very limited bandwidth in a single channel: 19.2Kbps in MICA2 [**?**] and 250Kbps in MICAz.

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.

- 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 control channel + multiple data channels" design [**?**].

2. MMSN PROTOCOL

2.1. Frequency Assignment

We propose a suboptimal distribution to be used by each node, which is easy to compute and does not depend on the number of competing nodes. A natural candidate is an increasing geometric sequence, in which

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

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

In our algorithm, we use the suboptimal approach for simplicity and generality. We need to make the distribution of the selected back-off time slice at each node conform to what is shown in Equation (1). It is implemented as follows: First, a random variable α with a uniform distribution within the interval $(0, 1)$ is generated on each node, then time slice i is selected according to the following equation:

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

It can be easily proven that the distribution of i conforms to Equation (1).

So protocols [**?**; **?**; **?**; **?**; **?**; **?**; **?**] that use RTS/CTS 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.

[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 protocols based on those similar to **?**] 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;
```

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. Basic Notations

As Algorithm **??** states, for each frequency number, each node calculates a random number (*Rnd*$_\alpha$) for itself and a random number (*Rnd*$_\beta$) 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. SIMULATOR

If the model checker requests successors of a state which are not created yet, the state space uses the simulator to create the successors on-the-fly. To create successor states the simulator conducts the following steps.

(1) Load state into microcontroller model.
(2) Determine assignments needed for resolving nondeterminism.
(3) For each assignment.
    (a) either call interrupt handler or simulate effect of next instruction, or

ALGORITHM 2: DAG Construction

Input: Model Graphs g=$gm_1, gm_2, \ldots, gm_n$
Output:DAG D_g includes decomposed model graphs stored in the nodes
Initialize the iterator, i=1
Initializs the largest common subgraph, S_max=0
Create a NULL node as root for DAG
while Unprocessed Model Graph ¿ 0 **do**
  Set g_i=gm_i
  if i = 1 **then**
    Create a node for g_i
  end if
  if g_i is a singleton **then**
    create a node for g_1 as a child of the root of DAG
    Insert g_i in PriorityQueue
  end if
else
  if g_i - S_max = q_s **then**
    s_{max} is subgraph isomorphic to g_i
    Create node for g_i and set link from the parent of S_max to g_i.
    Decompose g_s
    S_max = $|g_s, S_{max}|$
    Insert g_i in PriorityQueue
  end if
  return
else
  if g - S_max = 0 **then**
    g_i is isomophic to S_{max}
    Set Link in DAG from the parent of g_i to S_{max}
    Delete g_i
  end if
  return
else
  if $g_i - S_{max}$ = q_i **then**
    RandomDecompose(g_i) into connected graphs $g_i^{'}$ and $g_i^{''}$
    set S_max=—$g_i^{`}$,$g_i^{''}$,S_max—
    Insert g_i in PriorityQueue
  end if
  return
end while

(b) evaluate truth values of atomic propositions.
(4) Return resulting states.

Figure **??** shows a typical microcontroller C program that controls an automotive power window lift. The program is one of the programs used in the case study described in Section **??**. At first sight, the programs looks like an ANSI C program. It contains function calls, assignments, if clauses, and while loops.

3.1. Problem Formulation

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

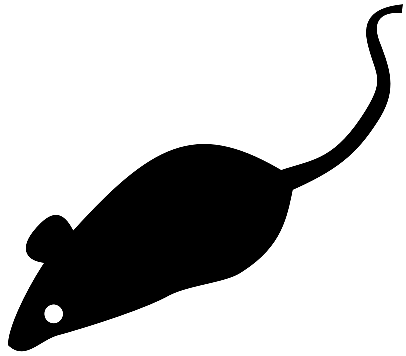

Fig. 1. Code before preprocessing.

ALGORITHM 3: Insert

```
Input: Graph g_i
let s := ∅, s_rest := ∅
if g_i is connected then
    for each graph g in DAG do
        if g ⊆_(induced) g_i then
            PriorityQuery.insert(g) — Insert g in PriorityQuery
        end if
    end for
end if
while PriorityQuery is not empty do
    s = PriorityQuery.dequeue() — s = dequeue PriorityQuery
    if s = g_i then
        let s be a child of g_i's parent and abandon g_i
        return
    else
        for all subgraph isomorphisms φ from s to g_i do
            s_rest = g_i −_(induced) s(decided by φ)
            if s_rest is connected then
                let s_rest be a child of g_i
                Insert(s_rest)
                return
            end if
        end for
    end if
end while
if g_i is not a singleton graph then
    if g_i is connected then
        (s, s_rest) = RandomPartition(g_i)
    else
        dustributes components of g_i to s and s_rest
    end if
    let s and s_rest be children of g_i
    Insert(s)
    Insert(s_rest)
end if
```

each other on the IG. Before any coalescing is done, each live range is a C-node by itself.

Table I. Simulation Configuration

TERRAIN[a]	(200m×200m) Square
Node Number	289
Node Placement	Uniform
Application	Many-to-Many/Gossip CBR Streams
Payload Size	32 bytes
Routing Layer	GF
MAC Layer	CSMA/MMSN
Radio Layer	RADIO-ACCNOISE
Radio Bandwidth	250Kbps
Radio Range	20m–45m

Source: This is a table sourcenote. This is a table sourcenote. This is a table sourcenote.

Note: This is a table footnote.

[a]This is a table footnote. This is a table footnote. This is a table footnote.

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. PERFORMANCE EVALUATION

During all the experiments, the Geographic Forwarding (GF) [**?**] routing protocol is used. GF exploits geographic information of nodes and conducts local data-forwarding to achieve end-to-end routing. Our simulation is configured according to the settings in Table **??**. Each run lasts for 2 minutes and repeated 100 times. For each data value we present in the results, we also give its 90% confidence interval.

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.

APPENDIX

In this appendix, we measure the channel switching time of Micaz [**?**] sensor devices. In our experiments, one mote 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: 24.3μs.

ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library.

ACKNOWLEDGMENTS

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

Received February 2007; revised March 2009; accepted June 2009

Online Appendix to: A Multifrequency MAC Specially Designed for Wireless Sensor Network Applications

GANG ZHOU, College of William and Mary
YAFENG WU, University of Virginia
TING YAN, Eaton Innovation Center
TIAN HE, University of Minnesota
CHENGDU HUANG, Google
JOHN A. STANKOVIC, University of Virginia
TAREK F. ABDELZAHER, University of Illinois at Urbana-Champaign

A. THIS IS AN EXAMPLE OF APPENDIX SECTION HEAD

Channel-switching time is measured as the time length it takes for motes to successfully switch from one channel to another. This parameter impacts the maximum network throughput, because motes cannot receive or send any packet during this period of time, and it also affects the efficiency of toggle snooping in MMSN, where motes need to sense through channels rapidly.

By repeating experiments 100 times, we get the average channel-switching time of Micaz motes: 24.3 μs. We then conduct the same experiments with different Micaz motes, as well as experiments with the transmitter switching from Channel 11 to other channels. In both scenarios, the channel-switching time does not have obvious changes. (In our experiments, all values are in the range of 23.6 μs to 24.9 μs.)

B. APPENDIX SECTION HEAD

The primary consumer of energy in WSNs is idle listening. The key to reduce idle listening is executing low duty-cycle on nodes. Two primary approaches are considered in controlling duty-cycles in the MAC layer.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000