Phoenix: A Checkpointing Tool

STEVEN ROHR, Florida State University
MICHAEL MASCAGNI, Florida State University

Checkpointing is one way to increase computational resiliency in case of unexpected (likely external) sources of error such as power failure. While a number of checkpointing tools already exist, most will checkpoint the entire program stack. This is not always time or space efficient, especially when dealing with programs where the current state can be expressed in terms of few variables (e.g. many numerical or stochastic methods).

Phoenix was thus developed to be a checkpointing tool which makes partial checkpoints, and implements an OpenMP-inspired #pragma-driven metalanguage which the user may use in order to identify which constructs require checkpointing and when. By making use of C++'s naturally extensible file I/O operators, we offer a tool which could checkpoint any variable type (albeit some will require more work done by the end user).

CCS Concepts: •**Software and its engineering → Source code generation; Preprocessors;** *Macro languages;*

Additional Key Words and Phrases: Checkpointing, pragma-driven

ACM Reference format:
Steven Rohr and Michael Mascagni. 2010. Phoenix: A Checkpointing Tool. *ACM Trans. Graph.* 9, 4, Article 39 (March 2010), 4 pages.
DOI: 0000001.0000001_2

1 INTRODUCTION

As a new technology, Wireless Sensor Networks (WSNs) has a wide range of applications [5, 8, 13], including 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 [4, 12, 20] and smart homes [3, 12, 20].

> "For these applications, sensor devices are incorporated into human cloths [3, 8, 27, 39] for monitoring health related information like EKG readings, fall detection, and voice recognition".

While collecting all these multimedia information [4] requires a high network throughput, off-the-shelf sensor devices only provide very limited bandwidth in a single channel: 19.2 Kbps in MICA2 [8] and 250 Kbps 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 [3, 13, 38, 39], 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 [27].

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 [3, 5, 8, 13, 38, 39] 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.

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.

[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 [5] and [3].

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;
```

2.2 Basic Notations

As Algorithm 1 states, for each frequency 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 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.
   1. either call interrupt handler or simulate effect of next instruction, or
   2. evaluate truth values of atomic propositions.
4. Return resulting states.

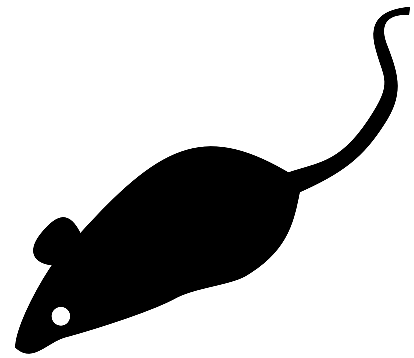

Fig. 1. Code before preprocessing.

Figure 1 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 3. 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 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 PERFORMANCE EVALUATION

During all the experiments, the Geographic Forwarding (GF) [5] 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

Table 1. 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.

in Table 1. 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.

6 TYPICAL REFERENCES IN NEW ACM REFERENCE FORMAT

A paginated journal article [2], an enumerated journal article [11], a reference to an entire issue [10], a monograph (whole book) [25], a monograph/whole book in a series (see 2a in spec. document) [19], a divisible-book such as an anthology or compilation [15] followed by the same example, however we only output the series if the volume number is given [16] (so Editor00a's series should NOT be present since it has no vol. no.), a chapter in a divisible book [36], a chapter in a divisible book in a series [14], a multi-volume work as book [24], an article in a proceedings (of a conference, symposium, workshop for example) (paginated proceedings article) [6], a proceedings article with all possible elements [35], an example of an enumerated proceedings article [17], an informally published work [18], a doctoral dissertation [9], a master's thesis: [7], an online document / world wide web resource [1, 30, 37], a video game (Case 1) [29] and (Case 2) [28] and [26] and (Case 3) a patent [34], work accepted for publication [31], 'YYYYb'-test for prolific author [32] and [33]. Other cites might contain 'duplicate' DOI and URLs (some SIAM articles) [23]. Boris / Barbara Beeton: multi-volume works as books [22] and [21].

A SWITCHING TIMES

In this appendix, we measure the channel switching time of Micaz [12] 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: 24.3 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 Scientsts' Support Program (http://www.nnsf.cn/youngscientsts).

REFERENCES

[1] Rafal Ablamowicz and Bertfried Fauser. 2007. CLIFFORD: a Maple 11 Package for Clifford Algebra Computations, version 11. (2007). Retrieved February 28, 2008 from http://math.tntech.edu/rafal/cliff11/index.html

[2] Patricia S. Abril and Robert Plant. 2007. The patent holder's dilemma: Buy, sell, or troll? *Commun. ACM* 50, 1 (Jan. 2007), 36–44. DOI:http://dx.doi.org/10.1145/1188913.1188915

[3] A. Adya, P. Bahl, J. Padhye, A.Wolman, and L. Zhou. 2004. A multi-radio unification protocol for IEEE 802.11 wireless networks. In *Proceedings of the IEEE 1st International Conference on Broadnets Networks (BroadNets'04)*. IEEE, Los Alamitos, CA, 210–217.

[4] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury. 2007. A Survey on Wireless Multimedia Sensor Networks. *Computer Netw.* 51, 4 (2007), 921–960.

[5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. 2002. Wireless Sensor Networks: A Survey. *Comm. ACM* 38, 4 (2002), 393–422.

[6] Sten Andler. 1979. Predicate Path expressions. In *Proceedings of the 6th. ACM SIGACT-SIGPLAN symposium on Principles of Programming Languages (POPL '79)*. ACM Press, New York, NY, 226–236. DOI:http://dx.doi.org/10.1145/567752.567774

[7] David A. Anisi. 2003. *Optimal Motion Control of a Ground Vehicle*. Master's thesis. Royal Institute of Technology (KTH), Stockholm, Sweden.

[8] P. Bahl, R. Chancre, and J. Dungeon. 2004. SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks. In *Proceeding of the 10th International Conference on Mobile Computing and Networking (MobiCom'04)*. ACM, New York, NY, 112–117.

[9] Kenneth L. Clarkson. 1985. *Algorithms for Closest-Point Problems (Computational Geometry)*. Ph.D. Dissertation. Stanford University, Palo Alto, CA. UMI Order Number: AAT 8506171.

[10] Jacques Cohen (Ed.). 1996. Special issue: Digital Libraries, *Commun. ACM* 39, 11 (Nov. 1996).

[11] Sarah Cohen, Werner Nutt, and Yehoshua Sagic. 2007. Deciding equivalances among conjunctive aggregate queries. *J. ACM* 54, 2, Article 5 (April 2007), 50 pages. DOI:http://dx.doi.org/10.1145/1219092.1219093

[12] CROSSBOW 2008. XBOW Sensor Motes Specifications. (2008). http://www.xbow.com.

[13] D. Culler, D. Estrin, and M. Srivastava. 2004. Overview of Sensor Networks. *IEEE Comput.* 37, 8 (Special Issue on Sensor Networks) (2004), 41–49.

[14] Bruce P. Douglass, David Harel, and Mark B. Trakhtenbrot. 1998. Statecarts in use: structured analysis and object-orientation. In *Lectures on Embedded Systems*, Grzegorz Rozenberg and Frits W. Vaandrager (Eds.). Lecture Notes in Computer Science, Vol. 1494. Springer-Verlag, London, 368–394. DOI:http://dx.doi.org/10.1007/3-540-65193-4_29

[15] Ian Editor (Ed.). 2007. *The title of book one* (1st. ed.). The name of the series one, Vol. 9. University of Chicago Press, Chicago. DOI:http://dx.doi.org/10.1007/3-540-09237-4
[16] Ian Editor (Ed.). 2008. *The title of book two* (2nd. ed.). University of Chicago Press, Chicago, Chapter 100. DOI:http://dx.doi.org/10.1007/3-540-09237-4
[17] Matthew Van Gundy, Davide Balzarotti, and Giovanni Vigna. 2007. Catch me, if you can: Evading network signatures with web-based polymorphic worms. In *Proceedings of the first USENIX workshop on Offensive Technologies (WOOT '07)*. USENIX Association, Berkley, CA, Article 7, 9 pages.
[18] David Harel. 1978. *LOGICS of Programs: AXIOMATICS and DESCRIPTIVE POWER.* MIT Research Lab Technical Report TR-200. Massachusetts Institute of Technology, Cambridge, MA.
[19] David Harel. 1979. *First-Order Dynamic Logic.* Lecture Notes in Computer Science, Vol. 68. Springer-Verlag, New York, NY. DOI:http://dx.doi.org/10.1007/3-540-09237-4
[20] Harvard CodeBlue 2008. CodeBlue: Sensor Networks for Medical Care. (2008). http://www.eecs.harvard.edu/mdw/ proj/codeblue/.
[21] Lars Hörmander. 1985. *The analysis of linear partial differential operators. III.* Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Vol. 275. Springer-Verlag, Berlin, Germany. viii+525 pages. Pseudodifferential operators.
[22] Lars Hörmander. 1985. *The analysis of linear partial differential operators. IV.* Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Vol. 275. Springer-Verlag, Berlin, Germany. vii+352 pages. Fourier integral operators.
[23] Markus Kirschmer and John Voight. 2010. Algorithmic Enumeration of Ideal Classes for Quaternion Orders. *SIAM J. Comput.* 39, 5 (Jan. 2010), 1714–1747. DOI:http://dx.doi.org/10.1137/080734467
[24] Donald E. Knuth. 1997. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms (3rd. ed.).* Addison Wesley Longman Publishing Co., Inc.
[25] David Kosiur. 2001. *Understanding Policy-Based Networking* (2nd. ed.). Wiley, New York, NY.
[26] Newton Lee. 2005. Interview with Bill Kinder: January 13, 2005. Video, *Comput. Entertain.* 3, 1, Article 4 (Jan.-March 2005). DOI:http://dx.doi.org/10.1145/1057270.1057278
[27] A. Natarajan, M. Motani, B. de Silva, K. Yap, and K. C. Chua. 2007. Investigating Network Architectures for Body Sensor Networks. In *Network Architectures*, G. Whitcomb and P. Neece (Eds.). Keleuven Press, Dayton, OH, 322–328. arXiv:cs/960935712
[28] Dave Novak. 2003. Solder man. Video. In *ACM SIGGRAPH 2003 Video Review on Animation theater Program: Part I - Vol. 145 (July 27–27, 2003).* ACM Press, New York, NY, 4. DOI:http://dx.doi.org/99.9999/woot07-S422
[29] Barack Obama. 2008. A more perfect union. Video. (5 March 2008). Retrieved March 21, 2008 from http://video.google.com/videoplay?docid=6528042696351994555
[30] Poker-Edge.Com. 2006. Stats and Analysis. (March 2006). Retrieved June 7, 2006 from http://www.poker-edge.com/stats.php
[31] Bernard Rous. 2008. The Enabling of Digital Libraries. *Digital Libraries* 12, 3, Article 5 (July 2008). To appear.
[32] Mehdi Saeedi, Morteza Saheb Zamani, and Mehdi Sedighi. 2010. A library-based synthesis methodology for reversible logic. *Microelectron. J.* 41, 4 (April 2010), 185–194.
[33] Mehdi Saeedi, Morteza Saheb Zamani, Mehdi Sedighi, and Zahra Sasanian. 2010. Synthesis of Reversible Circuit Using Cycle-Based Approach. *J. Emerg. Technol. Comput. Syst.* 6, 4 (Dec. 2010).
[34] Joseph Scientist. 2009. The fountain of youth. (Aug. 2009). Patent No. 12345, Filed July 1st., 2008, Issued Aug. 9th., 2009.
[35] Stan W. Smith. 2010. An experiment in bibliographic mark-up: Parsing metadata for XML export. In *Proceedings of the 3rd. annual workshop on Librarians and Computers (LAC '10)*, Reginald N. Smythe and Alexander Noble (Eds.), Vol. 3. Paparazzi Press, Milan Italy, 422–431. DOI:http://dx.doi.org/99.9999/woot07-S422
[36] Asad Z. Spector. 1990. Achieving application requirements. In *Distributed Systems* (2nd. ed.), Sape Mullender (Ed.). ACM Press, New York, NY, 19–33. DOI:http://dx.doi.org/10.1145/90417.90738
[37] Harry Thornburg. 2001. Introduction to Bayesian Statistics. (March 2001). Retrieved March 2, 2005 from http://ccrma.stanford.edu/~jos/bayes/bayes.html
[38] A. Tzamaloukas and J. J. Garcia-Luna-Aceves. 2000. *Channel-Hopping Multiple Access.* Technical Report I-CA2301. Department of Computer Science, University of California, Berkeley, CA.
[39] G. Zhou, J. Lu, C.-Y. Wan, M. D. Yarvis, and J. A. Stankovic. 2008. *Body Sensor Networks.* MIT Press, Cambridge, MA.

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