Distributed competition to compute localized scheduling

Quentin Lampin, Barthel Dominique, Valois Fabrice, Isabelle Augé-Blum

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Distributed competition to compute localized scheduling

Quentin Lampin — Dominique Barthel — Isabelle Augé-Blum — Fabrice Valois

N° 7522

october 2010

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7522--FR+ENG

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche GRENOBLE - RHÔNE-ALPES

Distributed competition to compute localized scheduling

Quentin Lampin* † ‡ , Dominique Barthel‡ , Isabelle Augé-Blum† , Fabrice Valois†

Thème : 1
Équipes-Projets SWING

Rapport de recherche n° 7522 — october 2010 — 15 pages

Abstract: Duty-cycled medium access protocols allow for long lasting autonomous networks by periodically putting nodes to sleep. However, this life expectancy improvement comes at the cost of a lesser network capacity and a poor adaptability to bursty traffic loads. Indeed, existing contention algorithms do not provide efficient algorithms to dynamically elect multiple senders per wake-up periods. In this paper, the medium is divided in several logical channels (eg. obtained by a time/frequency division of the communication medium) and we propose to allocate them dynamically among senders. For this purpose, we propose a joint contention/scheduling algorithm, named Extended Slot Selection (ESS), that schedules multiple sender/receiver pairs to available logical channels.

Key-words: medium sharing, contention, localized allocation.

* Corresponding author - supported by Orange Labs
† University of Lyon, INRIA, INSA Lyon, CITI, F-69621, France
‡ Orange Labs, Meylan, France

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Distributed competition to compute localized scheduling

Résumé : L'usage de protocoles MAC à endormissement cyclique permet un allongement de la durée de vie des réseaux de capteurs. Cet accroissement de la durée de vie du réseau est obtenue au dépend de la capacité en traffic des réseaux et entraîne une mauvaise adaptabilité aux charges de traffic variables, en particulier les traffics en rafale. En effet, les mécanismes de contention existants ne permettent pas une allocation dynamique et efficace de plusieurs paires de communications(récepteur/émetteur). Dans cet article, le médium est partagé en plusieurs canaux logiques (par exemple:division en temps/fréquence) et nous proposons d'allouer dynamiquement ces ressources aux noeuds en faisant la demande. A ces fins, nous proposons un algorithme, Extended Slot Selection (ESS), combinant contention et ordonnancement, qui alloue plusieurs paires émetteur/récepteur aux canaux logiques disponibles.

Mots-clés : partage de médium, contention, allocation localisée.

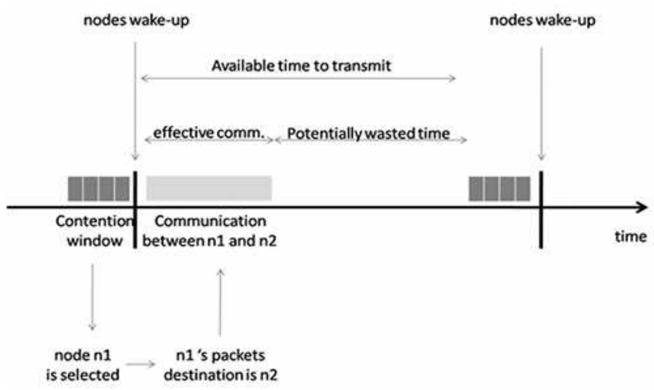

Figure 1: bandwidth utilization limitation

1 Introduction

Nowadays, wireless networks such as Wireless Mesh Networks, WiFi LANs or Wireless Low power Lossy Networks are ubiquitously deployed in both industrial and personal applications. Because such networks offer connectivity between nodes by the mean of a shared communication medium, simultaneous transmissions may interfere, causing collisions and possibly packet losses. Over the past decades, many MAC protocols have been designed to deal with this problem. The recent development of low power and autonomous wireless networks such as Wireless Sensor Networks (WSNs) revived this research field by introducing a new challenge to protocol designers, namely power efficiency. This effort led to the development of Low Power Listening (LPL) protocols that put nodes to sleep for long periods of time: eg. [1], [2], [3], [4], [5], [6]. However, existing localized contention algorithms for MAC protocols only allow for the selection of a single sender. In LPL protocols, this often implies that a single transceiver/receiver pair is allowed to communicate in a given wake-up period (as multiple contention phases are deemed too expensive). Consequently, the bandwidth utilization is very limited as shown in fig.1 and the latency is in the order of several sleep periods for congested networks. This cripples WSN performances and advocates for algorithms that permit to simultaneously select multiple sender nodes and allocates them to the available logical channels.

This paper proposes a localized medium sharing algorithm using a joint contention/scheduling scheme and is organized as follows: section 2 exposes the addressed problematic and the assumptions made in this work, section 3 reviews existing medium sharing algorithms, section 4 exposes our proposal and its design, section 5 presents simulations that assess the benefits of our proposal, section 6 investigates the energy cost of the proposal and its adaptability to dynamic topologies. Finally, section 7 unveils future work and summarizes this article.

2 Problem statement

As stated in sec. 1 and illustrated by fig.1, existing duty-cycled protocols present poor bandwidth utilization and poor latency since they allocate only one logical channel per contention phase and because the wake-up period is much larger than the typical packet duration. We believe that alleviating this limitation is equivalent to answering the following problem: *Given N nodes competing to access the medium, how to share the medium such that N' nodes get dedicated logical channels for their transmissions?*

2.1 Requirements

Considering WSNs specificities, the solution should exhibits the following properties:

- p_1 it must allow for the scheduling of multiple transmitter/receiver pairs in a given wake-up period.
- p_2 dimensioning of the algorithm must only depend on traffic load, not on network properties such as diameter and degree.
- p_3 it must only rely on localized information (localized algorithm).
- p_4 it must self-adapt to fast-varying, bursty traffic.
- p_5 it must cope with network dynamicity.
- p_6 it must grant fair access among nodes.

2.2 Assumptions

Most WSN applications strongly rely on network synchronization. For example, in metering applications as in urban networks [7] or industrial networks [8], data relevance is often limited to a given time-frame and outdated packets are dropped for energy and congestion considerations. Such mechanisms compel packet sources and all the forwarding nodes to share a common time reference so as to compare timestamps to their clock, thus requiring network synchronization. Network synchronization also allows for various optimizations such as synchronous sleep schedules that shorten wake-up guard times or time-spread transmission schedules for congestion mitigation. Therefore, we will make the assumption that a network time synchronization mechanism such as [9] or [10] is operating on the network.

3 Related work

Although quite different in nature, access control algorithms can be classified into three categories regarding the way nodes compete to access to the medium: deterministic, random and hybrid access.

3.1 Deterministic access algorithms

Deterministic access relies on pre-established schedules of node emissions such as in [11]: each wake-up period is dedicated to a specific node providing it with a collision free time slot. However, these algorithms poorly adapt to unpredictable, bursty traffics and to network changes or imply frequent re-schedules, at high energy cost. In order to mitigate this poor scalability, one could also think of extending the scheduling algorithm to address multiple logical channels in a single wake-up period. However, this would force nodes to listen to all available logical channels which is either not feasible (eg: multiple frequency channels at once) or extremely energy inefficient.

Therefore, these algorithms meet the key properties p_1 and p_6 but fail to achieve p_2, p_3 p_4 and p_5.

3.2 Random access algorithms

Random access relies mostly on CSMA algorithms and can also be sub-divided into two sub-types: unsynchronized and synchronized algorithms.

3.2.1 Unsynchronized algorithms

Unsynchronized algorithms such as described in [1], [4] and [5] are based on the preamble sampling algorithm. Nodes wake up periodically but at different times, the period being often the same for all nodes. When a node has data to send, it first listens to the channel: if it's idle, the node transmits either a long preamble whose size is larger than the wake-up period as in [1] or send a sequence of short preambles until the destination wakes up and responds to the preamble [4], [5]. If the medium is occupied, the node backs off and retries later. Thus, once a node initiates the algorithm by starting to send preambles, it inhibits other senders, that delay their transmissions. Therefore, this contention algorithm allows only for a single sender/receiver pair which violates desired key property p_1.

3.2.2 Synchronized algorithms

Synchronized algorithms benefit from a shared knowledge of time by setting rendez-vous points in time for congestion resolution algorithms. A widely-used algorithm in Low Power MAC protocols relying on nodes synchronization consists in the usage of fixed contention windows to grant access to the available channel as described in [12], [13] and [14]. The fixed contention window algorithm involves a fixed-size time frame subdivided into K time slots. Nodes willing to transmit on the medium, called contending nodes, will pick one or more of these slots and mark them by an occupancy signal, possibly random sequences of symbols. When not transmitting, nodes listen to check for occupancy of the other slots. Then, the decision to transmit is computed locally at each node based on its choices and other occupied slots. Therefore, these algorithms do not exhibit the key property p_1 but enforce p_2, p_3, p_4, p_5 and p_6[1]. As described in sec. 4, our proposal relies on this kind of algorithm so we deem necessary to review them here.

[1] under the assumption that all nodes share the same algorithm and parameters and that they do not depend on previous contention results

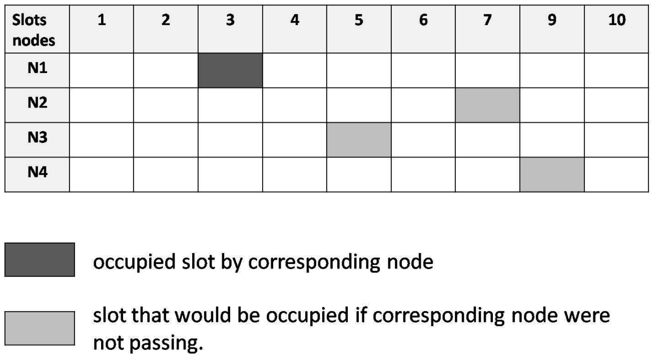

Figure 2: single slot choice algorithms

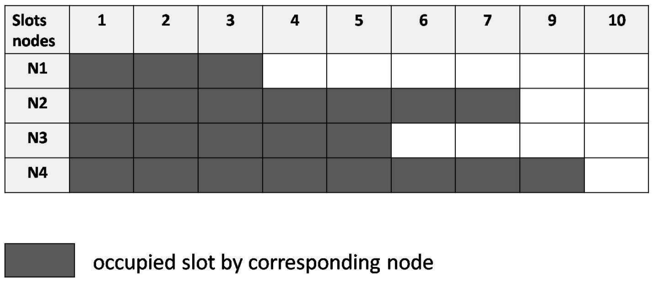

Figure 3: Longest burst algorithms

Single slot choice algorithms In congestion algorithms such as SMAC [2] and SIFT [12], nodes only pick one slot that they mark with an occupancy signal while they listen during the others. Node(s) occupying the first non-empty contention slot decide(s) to transmit, the others pass. Therefore, in fig.2, node *n1* decides to transmit in the emission slot while *n2*, *n3* and *n4* pass, hence the grayed chosen slot.

Longest burst algorithms The HIPERLAN [15] protocol defines a congestion resolution algorithm based on a longest bursts policy. Contending nodes pick an integer number i in the [1,CW][2] interval that defines their respective burst length and occupy the medium in the i first slots of the contention window. At the end of their own burst, nodes listen for other nodes' bursts. If the channel is idle until the end of the window, a node decide to transmit otherwise it passes. This is also equivalent to: *the node(s) with the longest burst decide(s) to transmit while the others pass.* Fig.3 illustrates this kind of policies.

Binary countdown algorithms The CONTI protocol [13] and its improvment [14] consider the K slots of the contention window as K rounds of selection. At the first slot, each contending node emits an occupancy signal with a probability p. When a node does not transmit, it listens for other signals: if there is one, it withdraws from the contention process otherwise it participates to the

[2] contention window slots number

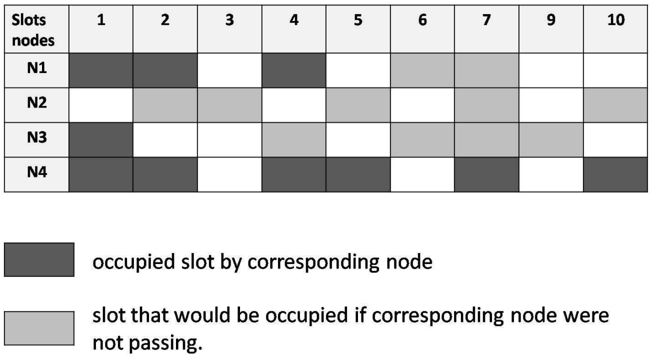

Figure 4: binary countdown algorithms

next round, and reiterates until the end of the contention window. Fig.4 gives an example of such a contention algorithm.

3.3 Hybrid algorithms

Hybrid algorithms such as [2], [3], [16] use a mixed TDMA-CSMA scheme: a slot assignment is first performed to allocate dedicated time slots to nodes as in TDMA algorithms. Then, if a slot owner does not use it, a random access algorithm takes place between nodes that need access to the medium. This way, such algorithms provide a more adaptive solution to the problem than purely deterministic algorithms. However, the dimensioning of the time slots and the time-frame remains problematic. In order to re-allocate slots that are not used by their owners, time-slots must be long enough to embed a contention window. Time-frames must also provide a dedicated slot to each node or face fairness issues. Therefore, hybrid algorithms fail to meet p_2, p_5 or p_6.

3.4 Existing solutions and adequacy to requirements

As seen previously, existing deterministic contention algorithms do not meet p_2, p_3, p_4 and p_5 whereas random access algorithms fail to meet p_1. Finally hybrid algorithms do not achieve p_2, p_5 or p_6. Therefore, none of the reviewed algorithms satisfies our requirements, thus prompting for a new algorithm that satisfies them all.

4 Proposed algorithm

4.1 Key Concept of the algorithm

In existing fixed congestion window algorithms, the decision process relies on the information that another node marked a preceding slot in the contention window. This leads to binary decisions, ie. *transmit* or *go back to sleep*. We propose to generalize this decision process by using the information on the number of occupied slots preceding current node's slot. This number is then used to build a localized scheduling of the nodes to the available logical channels.

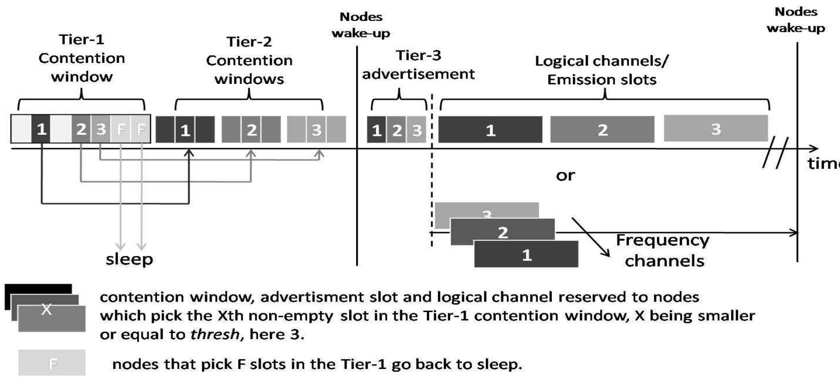

Figure 5: Detailed algorithm

More specifically, we propose to use a counter c that is managed locally at each node. This counter is used to keep track of the number of already occupied slots in the contention window, as seen by each node. Nodes emitting before the first increment of the counter decide to use the first logical channel, nodes that incremented the counter only once choose the second logical channel and so on until all the C_{log} logical channels get used (see fig.5 for more details).

4.2 Detailed algorithm

Our proposal relies on a 3-tiers extended selection algorithm called Extended Slot Selection (ESS) described below and illustrated by fig.5. Contending nodes participate in tier 1 and 2 while non-sending nodes sleep. In tier-3, all nodes are awake.

4.2.1 tier 1

Similarly to single slot choice algorithms, contending nodes pick a slot i in a tier-1 contention window in which they emit an occupancy signal. When a node does not transmit it listens for other nodes' transmissions. For each slot j, $j < i$ in which an occupancy signal has been heard, it increments a counter c of initial value 1.

4.2.2 tier 2

Each contending node whose counter c is less than a given threshold[3] C_{log} then competes in the c^{th} tier-2 contention window. When a node loses in one Tier-2 contention round, it may participates to the next available Tier-2 round if any. Hopefully, at the end of tier-2, up to C_{log} logical channels will be occupied by single nodes.

4.2.3 tier 3

The last step of this algorithm consists in the advertisement of packets' destinations by their senders. The node of the first logical channel will announce first,

[3] a discussion will be provided in 5.2

followed by the one of second logical channel and so on. This way, destination nodes will be able to deduce which logical channels they should listen to.

5 Performance evaluation: Simulation

5.1 Scenario and model

As depicted in [7] and [8], wireless sensor networks may present very dense node deployments and neighborhood of several hundreds sensors are to be expected. This performance evaluation will then consider a neighborhood of up to $N_{max} = 500$ nodes that are all within communication range of one another. All nodes in the neighborhood will want to access the medium. Moreover, we consider that two nodes emitting simultaneously results in a collision at the receiver (no capture effect).

5.1.1 Notations

We consider a N nodes neighborhood. The tier-1 contention window is of size K_1 and the tier-2 contention window of size K_2. C_{log} logical channels are available for transmission.

5.1.2 Simulation environment

Simulations are performed using the WSNet Simulator [17]. Relevant simulator parameters are given in table 1.

propagation model	infinite range
collision model	full
simulation time	$120000s$
wake-up period	$120s$
guard time (between slots)	$100\mu s$
slot time	$450\mu s$
data	$1packet/120s$, constant size
C_{log}	32

Table 1: simulation parameters

5.2 ESS parameter dimensioning

The SIFT protocol [12] recommends a contention window of size 32 for neighborhoods up to 512 nodes. So as to exhibit the benefit of the ESS algorithm, we have chosen to restrict the number of slots per logical channel to $NSlots_{log} = 16$, half of what is recommended in SIFT. For the simulations, we chose $K_1 = 128$, $K_2 = 12$ and $C_{log} = 32$, that configuration having proven to be efficient[4].

[4]the optimal ratio K_1/K_2 and $C_{log} = 32$ for a given collision rate and population N will be addressed in a future work.

5.3 Measurements

In order to evaluate the ESS algorithm, we have conducted measurements of the following indicators:

- *Throughput*: computed at the sink node. The packet size being constant, we consider the packet rate instead of the bit rate. We compute the average number of packets successfully sent per node. Obtained values are then compared to the theoretical maximum throughput for a $C_{log} = 32$ logical channels and for a single logical channel medium access protocol such as SCP-MAC [6] or SIFT [12].
- *Collision rate*: we keep track of the number of collisions and also compute the ratio #(packets.successfully.sent)/#(packets.sent).
- *Fairness*: illustrated by the computation of Jain's fairness index [18].

5.4 Simulation results

5.4.1 Throughput

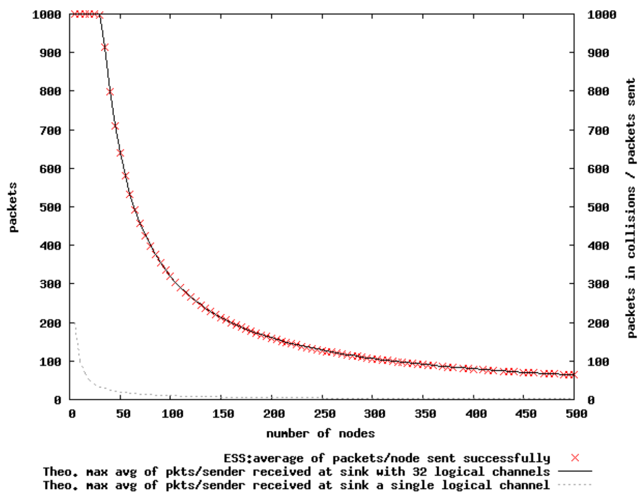

Figure 6: Throughput: simulation results

Fig.6 exposes the average number of packets received at the sink per sending node. Since the simulations last 1000 sleep periods, each node submits 1000 packets to the Medium Access Control protocol and up to 32 000 packets reach the sink node. Values obtained by simulation for the ESS algorithm are depicted by crosses. The upper curve represents the theoretical maximum average number of packets received at the sink per sending nodes for a $C_{log} = 32$ logical channels medium. The lower curve provides the same value for a single logical channel. Simulations show that our proposal is near-optimal regarding the throughput with chosen C_{log}, K_1 and K_2 and outperforms all protocols with a single logical channel allocation algorithm.

5.4.2 Collisions

Fig.7 illustrates the collision rate that was observed during the simulation. Vertical bars represent the number of lost packets due to collisions divided by the number of packets sent over all logical channels. With the chosen protocol parameters, this ratio is below 1.4% for all simulations and below 0.1% for in the 50 to 500 nodes range. Crosses report the raw number of collisions observed during the simulations and the line provides a linear approximation of the number of collisions. The correlation factor of 0,978 indicates that the collision rates follows a near linear progression for increasing neighborhood sizes in the range of interest.

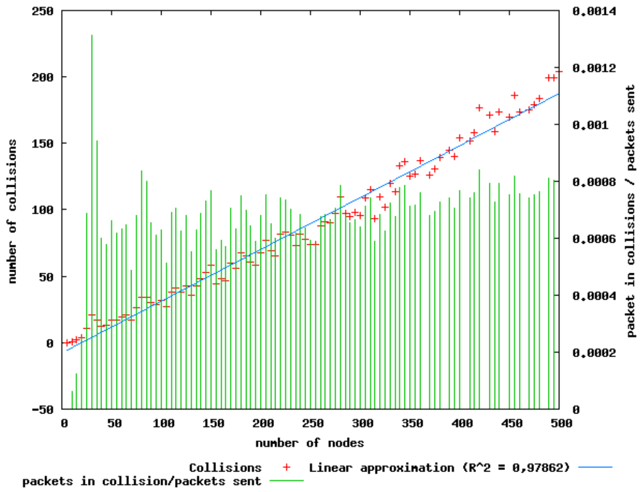

Figure 7: Collisions: simulation results

5.4.3 Fairness

Each round of contention/selection is independent of another, therefore the algorithm must be totally fair when all nodes use the same parameters. The simulations confirmed it: the Jain index stayed above 0.99 in all scenarios.

6 Discussions

6.1 Energy

As stated in 1, energy efficiency is a major concern in Wireless Sensor Networks therefore we deemed necessary to evaluate the cost of ESS and compare it to SCP-MAC [6] that is known to perform well under the same assumptions.

6.1.1 no traffic

When there is no traffic, the ESS energy cost is due to the synchronization mechanism and to the channel sampling. Since the synchronization mechanism is out of scope and is used in both ESS and SCP-like protocols, we will just consider the latter. This process requires all nodes to wake up periodically and sample the medium for activity, which cost energy. ESS and SCP-MAC require a different number of samplings per wake-up period. In SCP-MAC, receiver nodes must sample each and every slot of the post wake-up contention window. In ESS, a receiver node goes back to sleep as soon as it has determined the first advertisement slot to be empty.

6.1.2 Light traffic

Non-zero traffic adds cost in the case of both protocols. In ESS, there is a cost implied at the senders that contend in the tier-1 and tier-2 contention and a cost at the receivers that must listen to the advertisement of the destinations. In SCP-MAC, similar costs happen at the transmitters for the pre and post wake-up contention windows and at the receivers to listen to the post wake-up contention window and the packet header. Again, ESS cost is either less or equivalent to that of SCP, depending on contention windows sizes.

6.1.3 Heavy traffic, congestion

In case of heavy traffic, ESS energy cost increases because receivers listen to more advertisements whereas SCP-MAC cost for receivers stays constant. However, this increase in cost is factored over the number of transmissions that will occur in the same period. Indeed, ESS is able to schedule up to C_{log} concurrent communication pairs whereas SCP-MAC can only establish one. This means that SCP-MAC will need up to C_{log} wake-up periods to handle the same amount of traffic, each period causing additional cost for waking up the radio. Again, in case of heavy traffic, ESS is more energy efficient than SCP-MAC.

6.2 Network dynamicity

WSNs often present varying topologies during their life-time, such as node additions or removals. As stated in 3.1, this can cripple MAC performances (energy/collision rate), especially for deterministic and hybrid ones. As it does not rely on a reservation scheme, ESS is insensitive to the node IDs present in a given neighborhood, ie: changing a node MAC address or exchanging nodes won't affect its performances. ESS, despite using fixed size congestion windows, performs also very well in a network with varying neighborhood densities. As section 5.4.2 shows, ESS collision rate varies only slightly while the load (number of contending nodes) dramatically increases. Being insensitive to nodes IDs, ESS is therefore able to cope with varying densities, thus making it suitable for realistic conditions.

7 Conclusion and future work

ESS is an algorithm that allows for a better usage of the medium capacity by allocating multiple logical channels in a single contention/allocation process. Simulations have shown that ESS exhibits very low collision rates and almost optimal throughput. ESS is also fair, energy efficient and scales very well for neighborhoods up to hundreds of nodes. Therefore, ESS fulfills all requirements described in section 2.1, as illustrated by table 2.

algorithm	p_1	p_2	p_3	p_4	p_5	p_6
Deterministic access	✓	✗	✗	✗	✗	✗
Rand. access: unsynchronized	✗	✓	✓	✓	✓	✓
Random access: synchronized	✗	✓	✓	✓	✓	✓
Hybrid access	✓	✗	✓	✓	✗	✗
ESS	✓	✓	✓	✓	✓	✓

Table 2: Fulfillment of the requirements

We have proposed an algorithm and shown its efficiency by simulating a single neighborhood with increasing numbers of competing nodes. We are currently evaluating the same algorithm in multi-hop networks with realistic traffic patterns. We will also derive an optimal distribution for the first contention window and derive optimal K_1/K_2 and C_{log} values for a given maximum collision rate and population of nodes. This algorithm also offers interesting opportunities for QoS mechanisms that we will evaluate. Lastly, various optimizations can be envisioned on top of ESS.

References

[1] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *SenSys*, pages 95–107, New York, NY, USA, november 2004. ACM.

[2] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transaction on Networking*, 12(3):493–506, 2004.

[3] Injong Rhee, Ajit Warrier, Mahesh Aia, Jeongki Min, and Mihail L. Sichitiu. Z-MAC: a hybrid MAC for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 16(3):511–524, june 2008.

[4] A. Bachir, D. Barthel, M. Heusse, and A. Duda. Micro-frame preamble mac for multihop wireless sensor networks. In *ICC*, Istanbul, Turkey, June 2006. IEEE.

[5] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han. X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In *SenSys*, pages 307–320, Boulder, CO, USA, 2006. ACM.

[6] Wei Ye, Fabio Silva, and John Heidemann. Ultra-low duty cycle mac with scheduled channel polling. In *SenSys*, pages 321–334, New York, NY, USA, october, november 2006. ACM.

[7] M. Dohler, T. Watteyne, T. Winter, and D. Barthel. Routing requirements for urban low-power and lossy networks. RFC 5548, May 2009.

[8] K. Pister, P. Thubert, S. Dwars, and T. Phinney. Industrial routing requirements in low-power and lossy networks. RFC 5673, october 2009.

[9] Philipp Sommer and Roger Wattenhofer. Gradient clock synchronization in wireless sensor networks. In *IPSN*, pages 37–48, Washington, DC, USA, april 2009. IEEE Computer Society.

[10] Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi. The flooding time synchronization protocol. In *Sensys*, pages 39–49, New York, NY, USA, 2004. ACM.

[11] K. Pister and L. Doherty. TSMP: Time synchronized mesh protocol. In T. F. Gonzalez, editor, *PDCS*, Orlando, FL, USA, november 2008.

[12] Kyle Jamieson, Hari Balakrishnan, and Y. C. Tay. Sift: A MAC protocol for event-driven wireless sensor networks. In *EWSN*, volume 3868 of *Lecture Notes in Computer Science*, pages 260–275, Zurich, Switzerland, february 2006. Springer.

[13] Zakhia G. Abichar and J. Morris Chang. CONTI: Constant-time contention resolution for WLAN access. In *NETWORKING*, volume 3462 of *LNCS*, pages 358–369, Waterloo, ON, Canada, may 2005. Springer.

[14] J. Galtier. *Graphs and Algorithms in Communication Networks: Studies in Broadband, Optical, Wireless, and Ad Hoc Networks*, chapter Tournament Methods for WLAN: Analysis and Efficiency, page 379. Number 15 in Texts in Theoretical Computer Science: An EATCS Series. Springer-Verlag New York Inc, 2009.

[15] Standard. HIgh PErformance Radio Local Area Network(HIPERLAN) Type 1; Functional Specification. *ETSI Publications*, 1996.

[16] Lanny Sitanayah, Cormac J. Sreenan, and Kenneth N. Brown. Emergency response mac protocol (ER-MAC) for wireless sensor networks. In *IPSN*, pages 364–365, New York, NY, USA, april 2010. ACM.

[17] Elyes Ben Hamida, Guillaume Chelius, and Jean-Marie Gorce. Impact of the physical layer modeling on the accuracy and scalability of wireless network simulation. *Simulation*, 85(9):574–588, 2009.

[18] RK Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling.* Wiley, 1991.

Contents

1 Introduction **3**

2 Problem statement **4**
2.1 Requirements 4
2.2 Assumptions 4

3 Related work **4**
3.1 Deterministic access algorithms 5
3.2 Random access algorithms 5
3.2.1 Unsynchronized algorithms 5
3.2.2 Synchronized algorithms 5
3.3 Hybrid algorithms 7
3.4 Existing solutions and adequacy to requirements 7

4 Proposed algorithm **7**
4.1 Key Concept of the algorithm 7
4.2 Detailed algorithm 8
4.2.1 tier 1 8
4.2.2 tier 2 8
4.2.3 tier 3 8

5 Performance evaluation: Simulation **9**
5.1 Scenario and model 9
5.1.1 Notations 9
5.1.2 Simulation environment 9
5.2 ESS parameter dimensioning 9
5.3 Measurements 10
5.4 Simulation results 10
5.4.1 Throughput 10
5.4.2 Collisions 11
5.4.3 Fairness 11

6 Discussions **11**
6.1 Energy 11
6.1.1 no traffic 11
6.1.2 Light traffic 12
6.1.3 Heavy traffic, congestion 12
6.2 Network dynamicity 12

7 Conclusion and future work **12**

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399