Cascading Tournament Algorithm: Low Power, High Capacity Medium Sharing for Wireless Sensor Networks

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

To cite this version:

Quentin Lampin, Barthel Dominique, Isabelle Augé-Blum, Valois Fabrice. Cascading Tournament Algorithm: Low Power, High Capacity Medium Sharing for Wireless Sensor Networks. [Research Report] RR-7705, INRIA. 2011, pp.33. inria-00584936

HAL Id: inria-00584936

https://hal.inria.fr/inria-00584936

Submitted on 8 Aug 2011

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Cascading Tournament Algorithm: Low Power, High Capacity Medium Sharing for Wireless Sensor Networks

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

N° 7705

April 2011

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7705--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
GRENOBLE - RHÔNE-ALPES

Cascading Tournament Algorithm: Low Power, High Capacity Medium Sharing for Wireless Sensor Networks

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

Thème : 1
Équipes-Projets SWING

Rapport de recherche n° 7705 — April 2011 — 30 pages

Abstract: Existing Medium Access Control protocols for Wireless Sensor Networks reduce the radio activity to improve network lifetime, at the expense of a reduced network capacity. Those protocols are ill-suited for energy constrained sensor networks that must support spatially and temporally heterogeneous traffic loads.

This paper proposes a novel multi-ressource allocation algorithm and describes its implementation as a medium access control protocol for Wireless Sensor Networks.

The algorithm, named Cascading Tournament (CT), is a localized, dynamic, joint contention/allocation algorithm. It relies on cascading iterations of tournaments to allocate a multiplicity of ressources to a multiplicity of winners.

CT-MAC is an implementation of CT as a medium access protocol. By allocating multiple logicals channels allocation at each competition, CT-MAC improves the network capacity at a given duty-cycle or decreases the energy expenditure of the MAC layer at a given network capacity.

Extensive simulations highlight the benefits of CT-MAC in both single-hop and multiple-hop scenarios through the computation of relevant performance metrics: power consumption, network capacity, delay and retransmissions.

CT-MAC offers an unprecedented trade-off between network capacity, energy efficiency and delay and stands out as a solid candidate for energy constrained sensor networks that must support heterogeneous traffic loads. Our simulations show that CT-MAC significantly outperforms the state-of-the-art SCP-MAC protocol.

Key-words: Medium Sharing, Contention, Localized Allocation, Medium Access Protocol, capacity/energy trade-off.

---

[*] Corresponding author - supported by Orange Labs
[†] University of Lyon, INRIA, France
[‡] 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

Cascading Tournament Algorithm: Low Power, High Capacity Medium Sharing for Wireless Sensor Networks

Résumé : Les protocoles d'accès au medium radio existants pour réseaux de capteurs sans-fil réduisent l'activité de la radio afin d'améliorer la durée de vie du réseau, et ce, au prix d'une diminution de la capacité du réseau.

Ces protocoles sont peu adaptés pour les réseaux de capteurs contraints en énergie qui doivent supporter des trafics spatialement et temporellement hétérogènes. Ce rapport propose un algorithme d'allocation multi-ressources et décrit son implémentation sous forme de protocole de contrôle d'accès (MAC) au canal radio pour réseaux de capteurs. L'algorithme, appelé Cascading Tournament (CT), est un algorithme combiné de gestion de la contention/allocation localisé, dynamique et localisé. Il se repose sur des itérations de tournois en cascade pour allouer une pluralité de ressources à une pluralité de vainqueurs. CT-MAC est une implémentation de CT en tant que protocole MAC. En allouant plusieurs canaux logiques à chaque compétition, CT-MAC améliore la capacité du réseau pour un cycle d'endormissement donné ou diminue la consommation énergétique de la couche MAC pour une capacité du réseau donnée. Une étude complète par simulation montre l'intérêt de CT-MAC dans des scénarios de voisinage unique et multi-sauts. Ces simulations ont permis le calcul de métriques de performances pertinentes: consommation énergétique, capacité du réseau, délai et retransmissions. CT-MAC offre un compromis entre capacité du réseau et efficacité énergétique qui n'a pas de précédent. Il se présente donc comme un candidat sérieux pour les réseaux de capteurs contraints en énergie qui doivent supporter des trafics hétérogènes. Nos simulations ont montré que CT-MAC surpasse le protocole de l'état de l'art SCP-MAC.

Mots-clés : Partage de medium, Contention, allocation localisée, protocole d'accès au canal radio, compromis capacité/énergie.

1 Introduction

The network lifetime is a critical issue for commercial and industrial deployments of Wireless Sensor Networks as it determines their economic viability. Energy efficient Medium Access Control (MAC) protocols, e.g. [1], [2], [3], [4], [5], have been proposed to meet lifetime requirements of such networks.

However, the energy savings achieved by aggressively duty-cycling the radio come at the expense of network capacity. Existing localized competition algorithms for MAC protocols allocate a single resource. Some that allocate multiple resources do exist, however they either rely on non-localized reservation algorithms, e.g. [6], or require to iterate localized competition algorithms for each resource, e.g. [2]. In either case, allocating multiple resources comes at a significant energy cost.

Sensor networks must handle heterogeneous traffic loads, either in the temporal sense (bursty traffic) or in the spatial domain (funneling effect close to the sinks), while being drastically energy efficient. This is even more drastic in “Smart City” [7] applications where multiple applications are deployed on a single sensor network, e.g. pollution monitoring, water and gas metering, parking spots management etc. Traditionnally, in such energy-constrained networks, duty-cycle is adjusted to meet lifetime requirements and to support a periodic traffic load. Heterogeneous traffic loads may therefore saturate the network capacity and causes congestion, packet loss and significant delays. A naive solution would consist in setting the active time according to the maximum traffic load of the network, i.e. to match the network capacity to the traffic peak load. This solution is unsuitable when the traffic load is expected to vary widely. Adaptive duty-cycle MAC protocols may address the specific issue of the heterogenenous traffic loads but still incur energy costs for predicting/adapting the duty-cycle to traffics bursts.

This paper describes a novel localized medium-sharing algorithm: Cascading Tournament (CT) and its implementation as a Medium Access Protocol (CT-MAC). CT is a joint contention/allocation algorithm that both grants access to a multiplicity of winners and allocate them independant resources. In this paper, the ressources are logical channels obtained by partitioning the physical channel, e.g. in time (TDMA), frequency (FDMA) or code (CDMA).

Thanks to the multiple-ressource allocation feature of CT, CT-MAC supports higher traffic loads at a given duty-cycle than single-logical channels protocols do, e.g. [5], [3] and [4]. This allows CT-MAC to operate at lower duty-cycles and therefore to save energy at lower loads. Furthermore, by factoring the allocation of multiple channels in one large competition, CT-MAC reduces the allocation energy cost per channel allocated.

Contrary to existing multiple logical channels protocols, e.g [6], [8], CT-MAC is a localized protocol and it does not rely on a priori traffic pattern assumptions to operate. CT-MAC therefore adapts to heterogeneous and unpredictable traffic patterns.

A performance evaluation through extensive simulations assesses the benefits of CT-MAC over the existing state-of-the-art SCP-MAC protocol [5].

Simulation results outline the following main properties of CT and CT-MAC:

- CT is close to optimal in terms of logical channels usage, i.e. medium capacity.

- When using 32 logical channels, CT-MAC provides a capacity equivalent to that of SCP-MAC operating with a wake-up cycle 32 times faster while consuming up to 5.5 times less energy. However, this comes at the expense of some extra delay.
- CT-MAC outperforms SCP-MAC in terms of capacity and delay when operating at the same wake-up cycle period.

This paper makes three main research contributions. First, it describes a novel ressource sharing algorithm, Cascading Tournament (CT), that efficient allocates multiple resources. Second, it proposes an implementation of CT as a multi-channel MAC protocol, CT-MAC, specifically designed for energy constrained, realistic Wireless Sensor Networks. Third, it compares SCP-MAC to upper theoretical bounds and provides a fair and complete comparison against the state-of-the-art SCP-MAC protocol.

The rest of this paper is organized as follows: section 2 underlines the design objectives and requirements that outlined our proposal, section 3 states assumptions made in this study, section 4 reviews existing medium sharing algorithms, section 5 exposes CT design and CT-MAC implementation and section 6 and 7 presents simulations that assess the benefits of CT and CT-MAC. Finally, sections 8 and 9 unveils future work and summarizes this article.

2 Design Objectives and Requirements

2.1 Energy efficiency

- CT's energy cost to allocate multiple logical channels must be less than that of repeated existing contention algorithms allocating on logical channel each.
- CT must be distributed and must solely rely on local information, i.e. it must be a localized algorithm, because control messages exchange cost is considered prohibitive.

2.2 Adaptativity

- CT must not rely on a priori or dynamically discovered traffic pattern assumptions. It must self-adapt to fast-varying, bursty traffic.
- CT must cope with the traffic load heterogeneity of a collection network.
- CT-MAC's dimensioning must not depend on network topology, e.g. network diameter or connectivity.

2.3 Fairness

- CT must grant a fair medium access to all nodes.

2.4 Requirements

Requirements to compare CT to existing contention algorithms have been derived from this set of objectives:

$r_{heterogeneous}$ the contention algorithm must cope with the spatially heterogeneous load of a collection network.

$r_{dimensioning}$ dimensioning of the contention algorithm must not depend on network topology, e.g. network diameter or connectivity.

$r_{localized}$ the contention algorithm must exclusively rely on local information and operate in a decentralized manner.

$r_{assumption}$ the contention algorithm must not depend on traffic pattern assumptions.

$r_{adaptability}$ the contention algorithm must self-adapt to fast-varying, bursty traffic.

$r_{fairness}$ the contention algorithm must grant a fair medium access to all nodes.

3 Assumptions

Most Wireless Sensor Network applications strongly rely on network synchronization. For example, in metering applications as in urban networks [7] or industrial networks [9], 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 [10] or [11] is operating on the network.

4 Related work

Access control algorithms can be classified into three categories regarding the way nodes compete to access the medium: pre-reservation, random and hybrid access.

4.1 Pre-reservation access algorithms

Pre-reservation access relies on pre-established schedules of node emissions such as in [6] [8]: each wake-up period is dedicated to a specific communication providing it with a collision free logical channel. This access method allows for optimal support of periodic traffic loads. However, these algorithms adapt poorly to unpredictable traffics bursts and to network topology alterations. Indeed, each variation requires a new schedule, which implies energy cost and possibly outage during the dissemination of the new schedule.

Therefore, these algorithms meet the key requirement $r_{heterogeneous}$. $r_{fairness}$ requirement may also be met, although it would require to schedule transmission for all nodes in the network which violates $r_{dimensioning}$ requirement. Finally, such algorithms do not meet $r_{localized}$, $r_{assumptions}$ and $r_{adaptability}$ requirements.

4.2 Random access algorithms

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

4.2.1 Unsynchronized algorithms

In unsynchronized algorithms such as described in [1], [3] and [4], energy savings are achieved by a channel sampling mechanism. The sampling frequency is the same for all the nodes while the instants at which sampling occurs are different accross nodes. The competition algorithm is similar to those used in always-on radio protocols: Aloha, CSMA, etc. It consists in an optional random backoff to spread channel access attempts, followed by a Clear Channel Assessment (CCA). Such mechanisms cannot handle more than one packet per duty-cycle. Therefore, the wake-up period must be tuned according to the maximum expected traffic load or the network may face congestion, packet loss and/or delay issues. These preamble-sampling mechanisms also require sending a preamble prior to the packet to transmit. This preamble is either longer than the sampling period [1] or composed of a strobe of small frames [3], [4]. Its goal is to tell the destination node that a packet will be sent. In any case, such preamble holds the channel for a long time, which is detrimentatl to network capacity and delay. Therefore, these mechanisms do not meet the $r_{heterogeneous}$ and $r_{adaptability}$ requirements while respecting $r_{assumptions}$, $r_{localized}$ and $r_{dimensioning}$.

4.2.2 Synchronized algorithms

Synchronized algorithms benefit from a shared knowledge of time by setting rendezvous times to perform the channel competition. A widely-used algorithm in energy efficient MAC protocols relying on nodes synchronization consists in the usage of a fixed contention window to grant access to the medium as described in [12], [13] and [14]. The fixed contention window algorithm involves a fixed-size time frame subdivided into K time slots. Nodes competing to access the medium, called contending nodes, choose one or more of these slots and mark them using a busy tone. When not transmitting, nodes perform CCAs. The decision to access the medium is computed locally at each node based on slot choices and CCAs returns.

These algorithms inherently meet $r_{assumptions}$, $r_{localized}$ and $r_{dimensioning}$ requirements. $r_{fairness}$ requires all nodes to share the same algorithm parameters and independent successive channel access attempts, which are reasonable assumptions. However, existing algorithms lead to a binary decision, i.e. to access or to not access the medium. Such mechanisms can only allocate a single logical channel and thus need to be repeated for each logical channel, which is inefficient and not implemented in practice. As a result such algorithms do not adapt efficiently to the traffic variations that occurs in a collection network and do not meet $r_{heterogeneous}$ and $r_{adaptability}$.

As described in section 5, CT use fixed contention window to manage the channel access. A description of existing fixed contention window algorithms is therefore provided here.

Single slot choice algorithms In congestion algorithms as in SIFT [12], each node n_i chooses a slot t_i and marks it with a busy tone. Node n_i listens to slots preceding t_i. The first nodes to mark a slot gains access to the medium while the others retire until the next access competition.

Longest burst algorithms The HIPERLAN [15] protocol defines a congestion resolution algorithm based on a longest burst policy. Contending nodes n_i choose an integer number l_i in the interval [1,K] where K is the size of the contention window. l_i represents the size of n_i's burst. Node n_i marks the l_i first slots of the contention window with a busy tone. At the end of its burst, node n_i performs a CCA. If the CCA returns a positive, then n_i gains access to the medium, otherwise n_i retires until the next competition. In other words, nodes with the longest burst are granted access to the medium.

Binary countdown algorithms The CONTI protocol [13] and its improvment [14] consider the K slots of the contention window as K rounds of competition. At each round, nodes n_i mark the current slot with a busy tone with a probability p and perform a CCA with probability $1-p$. If n_i's CCA is negative then n_i retires until the next competition by not participating in the following rounds. Nodes that haven't retired by the end of the contention window gain access to the medium.

All those algorithms inherently meet $r_{assumptions}$, $r_{localized}$ and $r_{dimensioning}$ requirements. $r_{fairness}$ requires all nodes to share the same algorithm parameters and independent successive channel access attempts, which are reasonable assumptions. However, existing algorithms lead to a binary decision, i.e. to access or not to access the medium. Such mechanisms can only address a single logical channel and thus need to be repeated for each logical channel. Therefore, existing energy-efficient MAC protocols considered such overhead as unaffordable. As a result such algorithms do not adapt efficiently to the traffic variations that occurs in a collection network and do not meet $r_{heterogeneous}$ and $r_{adaptability}$.

4.3 Hybrid algorithms

Hybrid algorithms as in [2], combine both reservation and random access mechanisms. Logical channels are assigned to requesting nodes based on traffic predictions as done in reservation based access mechanisms and random access mechanisms take place when a logical channel is left unused by its legitimate owner. Such mechanisms provide a more adaptive solution to variable traffic patterns than pure reservation based algorithms. However, it still requires establishing and updating a communication schedule. Therefore those mechanisms suffer from the same limitations as pure reservation algorithms do. These

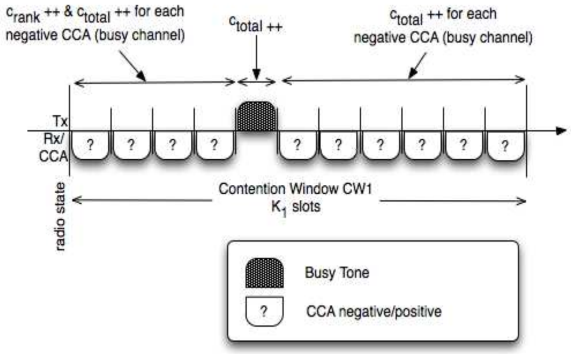


Figure 1: Tier-1 competition structure

algorithms meet the key requirement $r_{heterogeneous}$ and possibly $r_{fairness}$ but do not respect $r_{localized}$, $r_{assumptions}$ and $r_{adaptability}$ requirements.

4.4 Existing solutions and adequacy to requirements

As previously seen, existing reservation based access algorithms (the hybrid ones included) do not meet $r_{localized}$, $r_{assumptions}$ and $r_{adaptability}$. Random access algorithms fail to meet $r_{heterogeneous}$. Therefore, none of the reviewed algorithms satisfies all of our requirements, thus prompting for a new algorithm.

5 Cascading Tournament Algorithm

5.1 Core algorithm

The Cascading Tournament Algorithm core algorithm consists in a 3-tiers access competition algorithm and a data transmission window. Nodes competing to access the medium execute all tiers while others only execute tier 3.

5.1.1 Tier 1

This section refers to fig.1 for details. Tier 1 involves a contention window: CW_1. This contention window is divided into K_1 time slots. Each slot is long enough to perform a radio state transition Tx/Rx or Rx/Tx and a Clear Channel Assessment. Successive time slots are spaced by a time guard to prevent synchronization issues. See fig.2 for details.

- Nodes participating in tier-1, $N_{i\in 1..n}$, first randomly choose one of the K_1 slots $t_{i\in 1..K_1}$ and initialize two counters, c_{rank} and c_{total} to 0. c_{rank} purpose is to store the number of occupied slots preceding t_i while c_{total} stores the total number of occupied slots.

- Participating nodes $N_{i\in 1..n}$ perform CCAs in slots preceding t_i and increment c_{rank} and c_{total} values for each negative CCA (channel is busy).

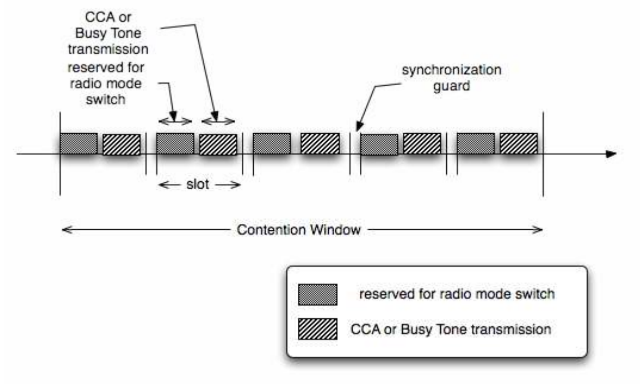


Figure 2: Structure of a contention window

- If c_{rank} and c_{total} values reach the number of available logical channels, C_{log}, then corresponding nodes retires of the competition, i.e. cancel further participation to tier-1 and tier-2, and schedule tier-3. In the meantime, these nodes switch their radio off.
- Nodes $N_{i \in 1..n}$ eventually send a busy tone (non coherent transmission) in the t_i slot after switching their radio to TX and incrementing c_{total}.
- Then, these nodes switch the radio to RX and perform CCAs in the $t_{j \in i+1..K_1}$ slots if any. Only the c_{total} counter is further incremented for each new negative CCA. If c_{total} reaches C_{log}, it is no longer incremented for new negative CCAs.
- At the end of tier-1, participating nodes know the number of occupied slots prior to their own: c_{rank}, and the total number of occupied slots: c_{total} (capped at C_{log}).

  Nodes whose counters verify $c_{rank} == c_{total} == C_{log}$ cancel their participation to the channel access competition.

  The others nodes then compete for the c_{rank}^{th} logical channel in the c_{rank}^{th} tier-2 contention window.

5.1.2 Tier 2

This section refers to fig.3 for details.

Tier 2 involves C_{log} contention windows: $CW_{2,j \in 0..(C_{log}-1)}$ of size K_2 slots.

- Each slot $t_{i \in 1..K_2}$ consists in a competition round similar to CONTI [13], i.e. nodes n_i mark the current slot with a busy tone with a probability p and perform a CCA with probability $1-p$.
- If n_i's CCA is negative then it cancels its participation to the current contention window and schedules a new competition in the first “free” $CW_{2,j}$ window if any. If no “free” $CW_{2,j}$ is found then n_i renounces to send its packet in the current period. A $CW_{2,j}$ is considered “free” when c_{total} value is strictly less than j.

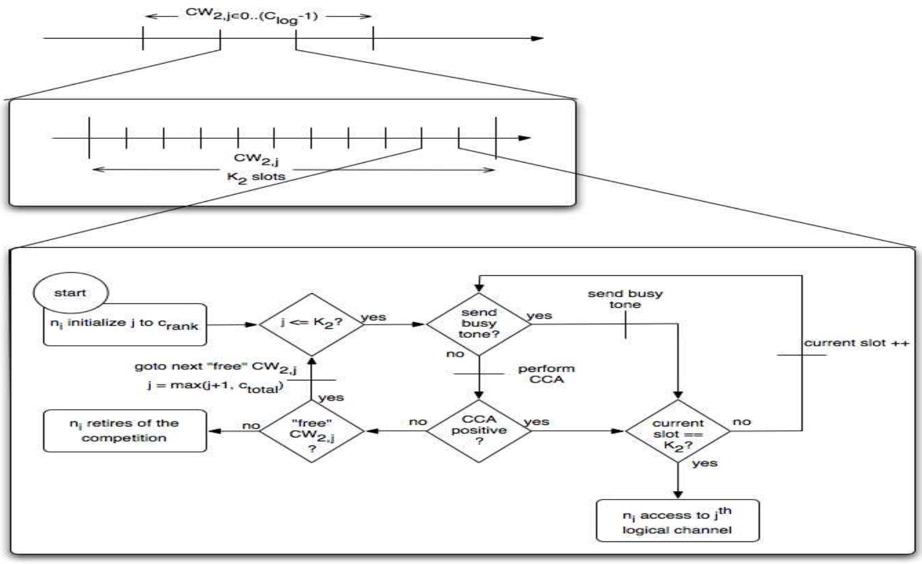


Figure 3: Tier-2 competition structure and algorithm.

5.1.3 Tier 3

Tier 3 involves an advertisement window of size C_{log}. Each advertisement slot, $ADV_{j \in 0..C_{log}-1}$ is long enough to operate a radio switch state, 2 frequency corrections (to adapt receiver oscillator period to the sender), time guards for synchronization issues and finally an advertisement payload. This payload is composed of the MAC addresses of the source and destination and a packet id. Nodes that won access to a logical channel j send an advertisement packet in the j^{st} advertisement slot and listen to remaining slots. Other nodes listen to all slots. When a node receives an advertisement in the ADV_j slot and is the destination, it schedules a radio switch to RX in the j^{th} data slot to receive the data.

5.1.4 Transmission window

Sending nodes transmit their data packet in the data slots and then switch their radio to RX, waiting for a data acknowledgment packet. The data packet does not contain a full MAC header but only the packet id and the data itself. When a node successfully receives a data packet, it sends back a data acknowledgement. The data acknowledgment contains only the packet id. If a data acknowledgment is received, sending nodes may dispose of the sent packet and free the packet buffer.

5.2 Illustration of CT algorithm: detailed chronogram

Fig. 4 depicts a chronogram that shows how CT-MAC is operated. 6 nodes, $N_{1..5}$ and N_r, compose a set of nodes in a one-hop neighborhood. $N_{1..5}$ have packets to transmit. N_r has no packet to send. The tier 1 contention window CW_1 has 6 slots, the tier 2 contention windows $CW_{2,j}$ have 5 slots and the

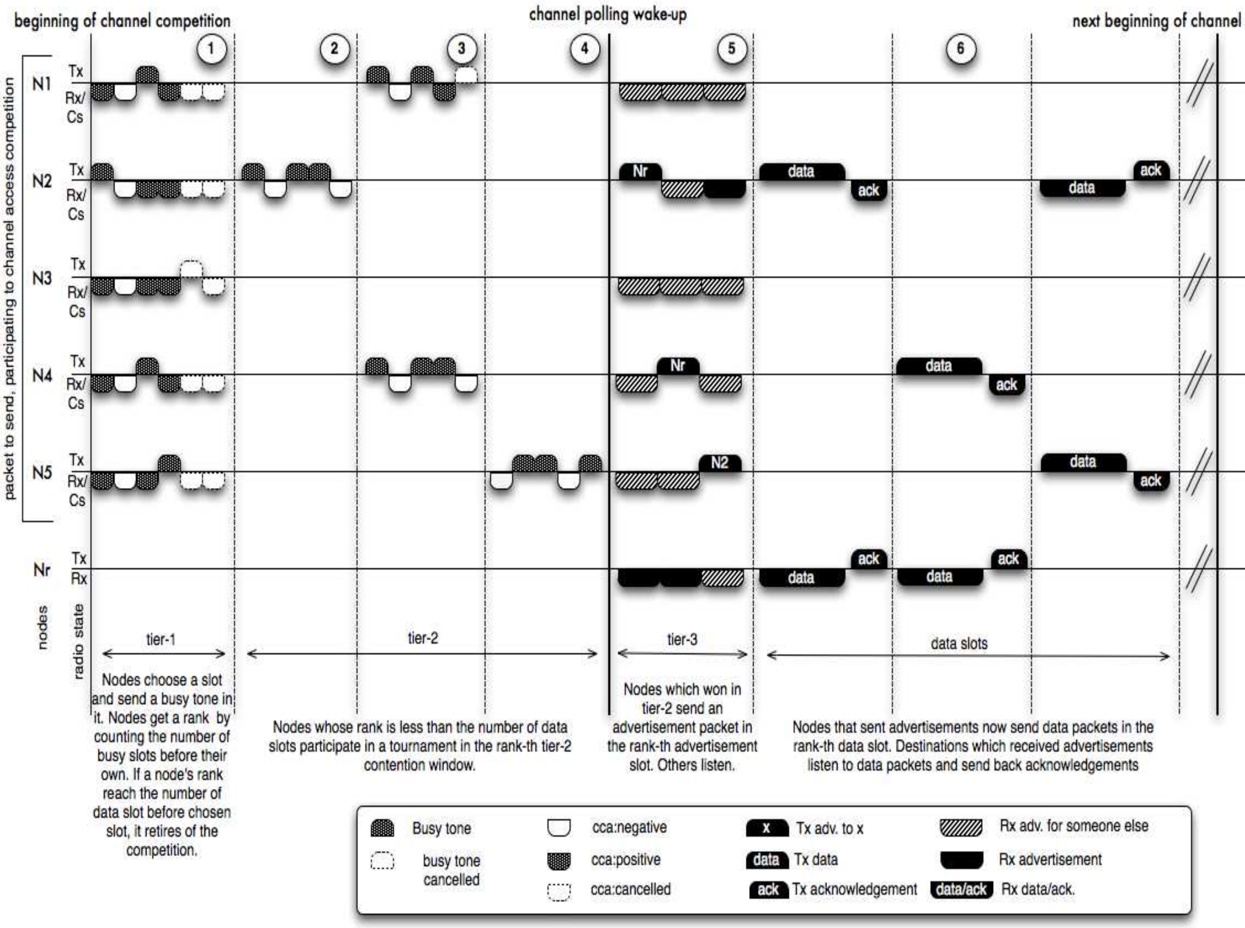


Figure 4: CT algorithm chronogram: a detailed example

number of available logical channels (data slots): C_{log}, is 3. On fig. 4, the 6 stamp numbers correspond to the 6 following items.

(1) As described in 5.1, $N_{1..5}$ choose a random slot in the tier 1 contention window. In this scenario, Nodes $Ni \in 1..5$ compete to access the medium and respectively choose slots 3, 1, 5, 3, 4.

- N_2 performs a busy tone in slot 1, its c_{rank} is therefore 0. N_2 then increments c_{total} to 1. $N_{i\in\{1..6\}\setminus\{2\}}$ increment their c_{rank} and c_{total} counters to 1.
- There is no busy tone in slot 2. Neither c_{rank} nor c_{total} is incremented.
- N_1 and N_4 send a busy tone in slot 3, their c_{rank} value is therefore 1. $N_{i\in\{1..6\}\setminus\{1,2,4\}}$ increment c_{rank} and c_{total}. $N_{\{1,2,4\}}$ increment c_{total}.
- N_5 performs a busy tone in slot 4. It's c_{rank} is therefore 2. N_3 increments both c_{rank} and c_{total}. Its c_{rank} and c_{total} values verify

$c_{rank} == c_{total} == C_{log}$, N_3 therefore retires of the medium access competition and turns its radio off. $N_{i\in\{1..6\}\setminus\{3\}}$ increment c_{total}, which reach C_{log}. $N_{i\in\{1..6\}\setminus\{3\}}$ thus cancel CCAs scheduled in slot 5 and 6.

(2) N_2 competes in the first tier 2 contention window: $CW_{2,0}$. All CCAs performed by N_2 returns are positive. N_2 therefore decides to send an advertisement in the first advertisement slot.

(3) N_1 and N_4 compete in the second tier 2 contention window: $CW_{2,1}$. In the first 3 slots, N_1 and N_4 perform the same actions therefore CCAs returns are positive. In slot 4, N_1 performs a CCA while N_2 send a busy tone. N_1's CCA return is thus negative and N_1 cancels its participation to this contention window. Since c_{total} value is equal to C_{log}, N_1 sees no "free" tier-2 contention window and retires of the competition. N_4 finally decides to send an advertisement in the second advertisement slot.

(4) Similar to (2). N_5 decides to send an advertisement in the third, and last, advertisement slot.

(5) N_2 sends an advertisement to N_r. Upon reception of the advertisement packet, N_r schedule a radio wake-up on state Rx at the first data slot. Other nodes ignore the advertisement. Similarly N_4 and N_5 send their advertisement packets in the 2^{nd} and 3^{rd} advertisement respectively. N_r and N_6 schedule radio wake-up at the 2^{nd} and 3^{rd} data slots respectively.

(6) N_2, N_4 and N_5 send the data packets and N_r and N_2 send back acknowledgments.

5.3 Optimizations

Depending on the traffic load and constraints, CT-MAC implementation may includes the following optimizations.

5.3.1 Advertisement acknowledgment

When receiving an advertisement, a destination node may send back an acknowledgment to the sender. This optimization eventually prevents the sender to waste the necessary energy to send the data payload at the cost of sending an acknowledgment by the receiver for each packet. The trade-off lies in the ratio of the respective costs to send a data payload and an acknowledgment and in the channel error rate.

5.3.2 Adaptive listening

This section refers to fig.5 for details. CT Core algorithm specifies that nodes listen to all C_{log} advertisement slots. In case of light traffic conditions, this causes unwanted idle listening and therefore energy wastes. The adaptive listening optimization consists in a listening time-out. When nodes wake up to probe the channel for activity, they first schedule a listening period. This listening period consists in $s_{adaptive}$ advertisement slots for receiving-only nodes. Nodes that send advertisements schedule a listening period of $s_{adaptive} + s_{latest\_slot}$

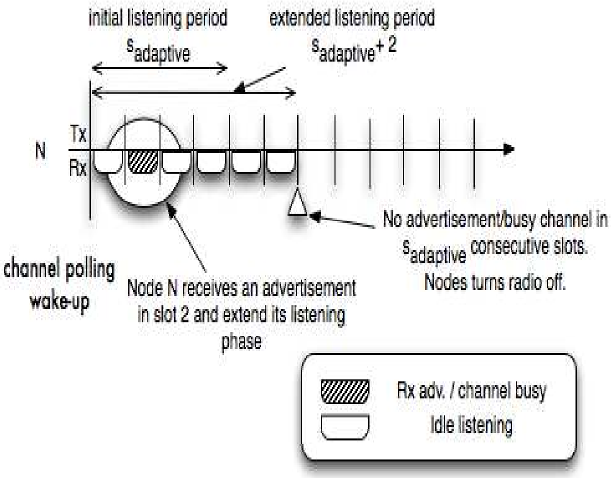


Figure 5: Adaptive listening optimization

slots, where s_{latest_slot} is the latest slot in which they send their advertisement. When an advertisement is received or an advertisement slot is found busy, nodes extend the listening period to listen to the next $s_{adaptive}$ slots. The time-out is thus postponed to $s_{current} + s_{adaptive}$ if it's not already scheduled later.

This optimization helps preventing energy wastes due to the idle listening phenomenon. However, it may also induce early radio OFF switches and eventually make nodes unavailable for advertisements and data reception.

5.3.3 Multiple channels access per node

This discussion relates to fig6.

The notion of "free" logical channel has been introduced in 5.1. When such logical channels exist, i.e. $c_{total} < C_{log}$, nodes that have multiple packets to send are given the right to compete for these "free" logical channels. Such nodes may therefore compete in the corresponding tier 2 contention windows $CW_2, j \in c_{total}..C_{log}$.

This optimization rely on the assumption that the communication stack buffer can store more than one packet, which is not always true, e.g. the Contiki operating system [16]. When this assumption is verified, this allows bottleneck sensor nodes to use more data slots and thus to handle higher traffic loads, which is particularly useful for nodes close to the network sink node.

5.3.4 Channel reuse

Section 5.2 shows that when multiple nodes compete in a tier 2 contention window, some of them will likely retire from the competition, i.e. when a node performs a negative CCA. When operating in a multi-hop wireless network, such situations may lead to unused data slots in given neighborhoods. Fig.7(a) and Fig. 7(b) expose such a scenario. In this case, node 2 obtains the first logical channel. Node 1 and 4 competes for the second logical channel. Node 4 wins

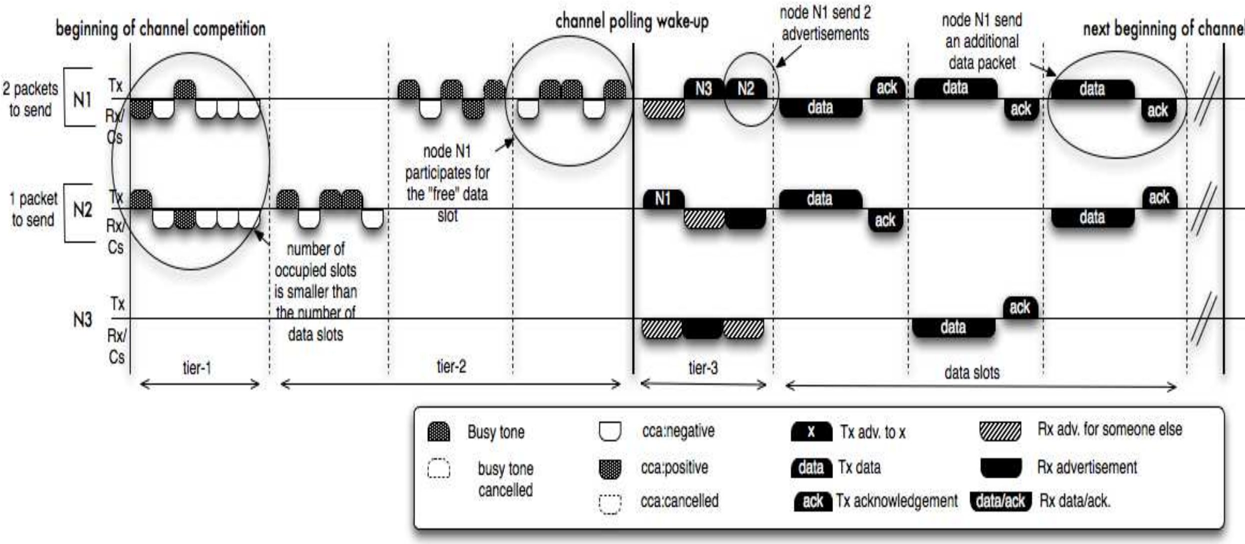


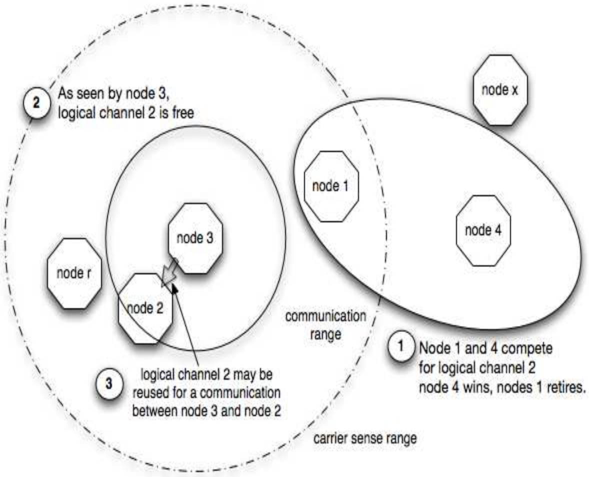
Figure 6: Multiple channels access per node optimization

access to this channel in the tier 2 second contention window $CW_{2,1}$ while node 1 cancels its participation at slot 1. Therefore, if no channel reutilization method is used, the second logical channel is seen empty by node 3. One method to efficiently reuse these channels is that nodes schedule CCAs at the median slot in all tier-2 contention $CW_{2,j}$ windows where they would not participating if strictly following the core algorithm. If the CCA is negative then these nodes do not further participate to the window (1). However, if the CCA is positive then the nodes may compete in the second half of the window for the associated logical channel (2). In order to guarantee that “legitimate” nodes, i.e. nodes that have chosen the $CW_{2,j}$ window during tier 1, have priority over the “opportunistic” nodes, “legitimate” nodes that have not canceled their participation before the median slot, send a busy tone in the median slot. See. fig.7(b) for details. This optimization helps maximizing the channel usage and thus the network throughput. However, it increases the collision probability for packets sent in these reused logical channels as their tournament access process relies only on the second half of the corresponding $CW_{2,j}$ windows.

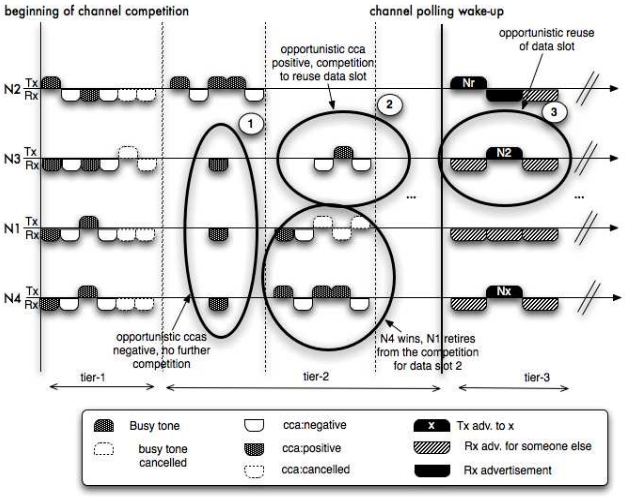
6 Evaluation

This section evaluates the CT algorithm and its implementation, CT-MAC. For comparison purposes, results are also provided for SCP-MAC [5]. SCP-MAC is an ultra-low duty-cycle MAC protocol designed towards energy savings and it stands out as one of the most energy efficient protocol for Wireless Sensor Networks.

First, this section provides simulation results for the main CT-MAC (without optimizations) in a single-hop scenario with a perfect channel. This scenario



(a) scenario



(b) chronogram

Figure 7: A channel reuse scenario

outlines CT's main feature, i.e multiple logical channel allocation, and shows it near-optimality.

Second, we simulated CT-MAC with optimizations in multi-hop scenarios. These results show the behavior of CT-MAC in a data collection network, i.e. with a converge-cast traffic, when facing a non-perfect channel. Throughput, power consumption, delay and retransmission rate results are provided and discussed.

6.1 Simulation settings

Simulations are conducted on the WSNET network simulator [17].

radio bitrate	20 kbits/s
radio output power	0 dBm
radio RX, TX power consumption	53.7, 65.7 mW
radio wake-up cost	0.16 mJ
carrier-sense sensitivity	-97 dBm
tournament slot	1 ms
advertisement slot	8 ms
data payload	200 bits
wake-up period (CT-MAC)	10 s
wake-up periods (SCP-MAC)	312.5 ms, 10 s
C_{log}	32 (TDMA)
CW_1, CW_2	128, 12
SCP-MAC pre, post wake-up CW	32, 12
$s_{adaptive}$	6
propagation model	perfect channel, 2-ray ground
simulation duration	10 000 s

Table 1: simulation parameters

Table 1 lists relevant simulation parameters. CT-MAC operates on a 10 s wake-up/sleep cycle and allocates 32 TDMA-based logical channels. The 10 s duty-cycle is inspired from existing commercial urban sensor deployments [7] while the number of logical channels has been chosen such that packets up to 80 bytes (including physical and routing headers) can be handled without fragmentation, which we deem sufficient for collection networks. The reference protocol, SCP-MAC, is simulated with both a 10 s and a 312.5 ms wake-up period, i.e. 1 and 32 operation cycles every 10 s. As verified in 6.2, this means that we compare our CT-MAC with one version of SCP-MAC operating at the same duty-cycle and with another version of SCP-MAC operating at the same theoretical capacity. In this paper, we will refer to these two specific versions of SCP-MAC as SCP-MAC(10 s) and SCP-MAC(312.5 ms). For SCP-MAC, we use the values $K_1 = 32$ and $K_2 = 12$ often seen in the litterature. To be fair in our comparison, we have implemented the obvious improvement to SCP-MAC that consists in using a CONTI algorithm in its tier-2 contention window, instead of a uniform backoff. Our algorithms therefore only differ in the tier-1 algorithm. Both CT-MAC and SCP-MAC use a uniform probability distribution for ther tier-1 contention window. For CT-MAC, we naturally picked the same value $K_2 = 12$ as for SCP-MAC. Because we compare CT-MAC to versions of SCP-MAC with a sleep period spanning more than an order of magnitude, we expect the population of contending nodes per CT-MAP period to be generally larger than that of SCP-MAC(10 s). We therefore picked $K_1 = 128$, which brings the collision occurence low enough that other practical factors such as channel error rate will dominate. Chosing a lower K_1 reduces the protocol overhead but increases the collision rate, and vice-versa. Optimizing K_1, K_2, C_{log} for given application requirements (traffic load, collision rate, etc) is left as future work. The simulated time is a fixed 10 000 s. As an example, this means that a network of 100 nodes each generating 0.01 packet/second processes about 10 000 packets during the simulation run. Furthermore, for the multi-hop scenarios described in 6.3.1, one simulation is run on each of four different routing topologies of the same physical network, and results are averaged across these four simulation.

6.2 Single-hop scenario

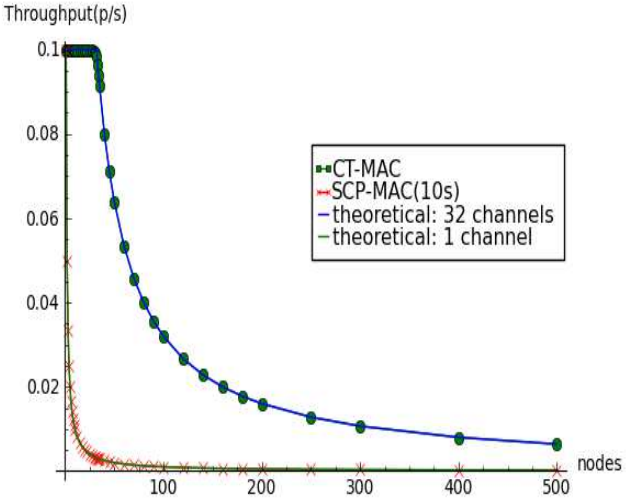


Figure 8: Throughput: CT allocation is near-optimal

In this scenario, n nodes share a common medium and compete to send packets to the sink node N_r. All nodes are in one another's radio range. The channel is error-free. Packets that overlap in time generate collisions and are all lost. All the nodes generate one packet every wake-up period of CT-MAC and SCP-MAC(10s), i.e. every 10 s. They therefore all compete in each and every period. We ran a range of simulations to sweep values from 1 to 500 for n.

Fig.8 depicts the average throughput, computed as the rate of packets successfully received at the sink N_r divided by the number of source nodes, averaged over the simulation time. Results are provided for four scenarios: CT-MAC, SCP-MAC and, in order to serve as upper theoretical bounds, a perfect algorithm for a medium comprising 32 logical channels and a perfect algorithm for a medium providing 1 single logical channel. The graph confirms our intuition that a globally shared medium is unable to deliver the requested load of 1 packet per source node per wake-up period when the number of source nodes exceeds the number of channels available, hence the sharp knee in the CT-MAC curve for n close to 32 and the hyperbolic decay in the right part of the graphs.

Both CT-MAC and SCP-MAC(10s) perform near-optimally, considering their respective number of logical channels. In both cases, the distance to the optimal value is due to residual collisions stemming from identical decisions being taken at different nodes. For example, two or more nodes send their advertisements in the same slot (with CT-MAC), or two nodes send their packets at the same time

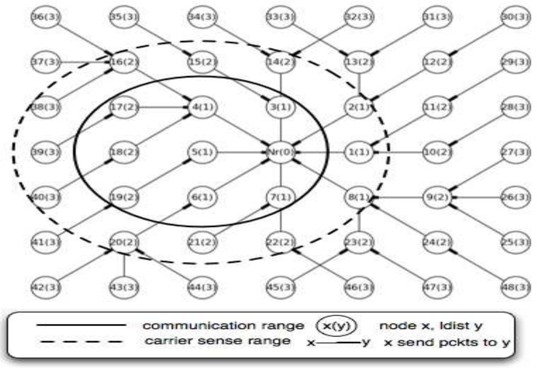


Figure 9: Multiple-hops topology example

(with SCP-MAC). The collision ratio (number of collisions divided by number of packets sent), lies below 10^{-3} for either protocol.

Because CT-MAC is near-optimal in allocating C_{log} logical channels, it provides a network capacity up to C_{log} times that of SCP-MAC operating at the same wake-up period. Conversely, at same network capacity, CT-MAC can use a much longer wake-up period, providing significant energy savings when and where the traffic is low. This will be verified in 6.3.5. This, however, comes at the cost of higher delays, as shown in 6.3.7.

6.3 Multiple-hops scenarios

6.3.1 Physical and routing topologies

This set of scenarios involves 3 physical topologies. Each physical topology consists in n source nodes, $n \in \{48, 80, 120\}$, and one sink node N_r. The nodes are placed on a fixed pitch grid centered on N_r. Topologies with more nodes span a larger physical area, with an identical spatial density. On each physical topology, we ran four times a distance-vector routing algorithm to build four different (logical) routing topologies, over which below-mentionned results were averaged.

We define *ldsit*, the logical distance of a node, as the number of routing hops from the node to the sink. The sink node N_r has an *ldist* of 0 while nodes at x hops from N_r get an *ldist* value of x. Fig.9 depicts an example of such a topology.

6.3.2 Offered load

The n nodes all send their packets to the sink N_r, potentially over multiple hop routes. Over multiple simulations, their packet generation interval is varied from 1000 s downto 34 s, i.e they all uniformely generate from 0.001 to 0.029 packets per second. Compared with the contention cycle of 10 s of CT-MAC and SCP-MAC(10 s), this means that, at the highest simulated offered load, each node injects a packet into the network about every third contention cycle.

6.3.3 Performance metrics

This study evaluates the protocols based on the following performance metrics:

Throughput As in 6.2, we define the throughput as the aggregate rate of non-duplicate packets that are received at the sink node N_r, divided by the number of source nodes. This is averaged over the full simulation time.

Power consumption The power consumption is the total power expenditure of all nodes during the simulation, divided by the number of nodes. This is averaged over the full simulation time. For further insight, we actually compute and display this metric for each *ldist* bin, i.e. we average and display the average power consumed by the nodes at $ldist = 1$ separately from the average power of the nodes at $ldist = 2$, etc. Indeed, the amount of traffic is significantly different across successive network rings, and it reflects in the power consumption. The consumption of edge nodes in low traffic scenarios depicts the base energy consumption of the protocol. Indeed, nodes rarely compete to access the medium and therefore they consume most of their energy probing it for incoming transmissions.

Traffic delay The traffic delay is the difference between the time a packet was received at the sink node N_r and the time it was created at the application layer of the source node. This delay is higly influenced by the sleep cycle period and the network capacity. For the same reason as mentioned above, we display average Traffic delays for each *ldist* bin.

Data retransmission rate Data retransmission rate is the rate of data packets that have been sent by a given node but not acknowledged by the on-link destination (not the final destination!), whether these packets originated at that node or were simply forwarded. We also display data retransmission rates averaged over each *ldist* bin of the sending nodes.

6.3.4 Throughput

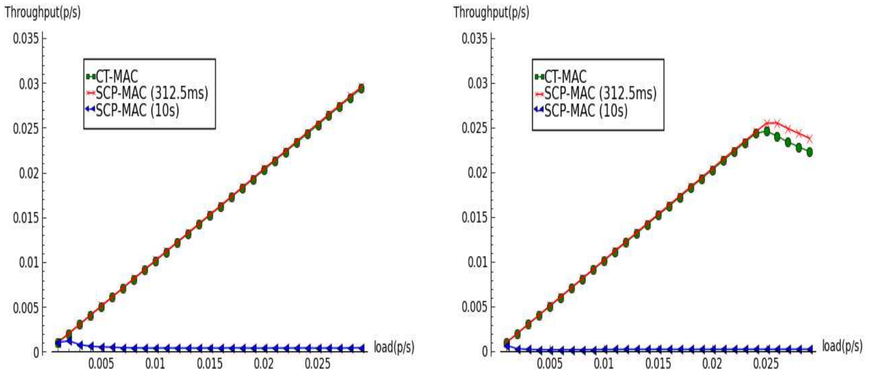
This section discusses Fig. 6.3.4, which represents the average rate of non-duplicate packets received at the sink node N_r per source node, versus the load offered by each node, under our three different MAC protocols and on three different network sizes. For all three network sizes, CT-MAC (with a 10 s sleep-cycle) and SCP-MAC with a 312.5 ms cycle offer a similar network-wide throughput: it increases linearly with the load, then falls off when the network gets saturated. The difference between CT-MAC and SCP-MAC(312.5ms) doesn't exceed 5%. All nodes don't fare equally well with respect to saturation, depending of their position in the network, but space limitations prevent us from presenting further details here.

Clearly, CT-MAC hugely outperforms SCP-MAC(10 s). This is consistent with the single-hop simulations of section 6.2.

In Fig. 6.3.4, the 95% confidence interval, not displayed for clarity, never exceeds a width of 4% of the value plotted.

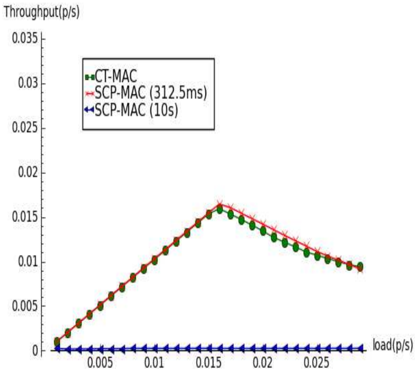
6.3.5 Power consumption

The energy consumption results are shown in figures 6.3.5 , 6.3.5 and 13



(a) 48 nodes network

(b) 80 nodes network

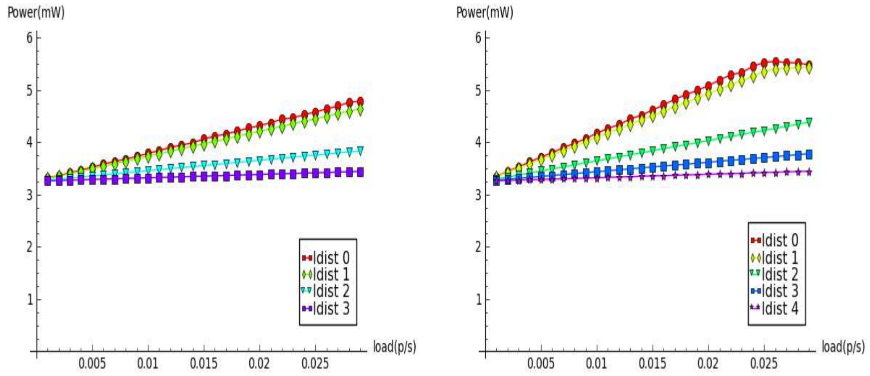


(c) 120 nodes network

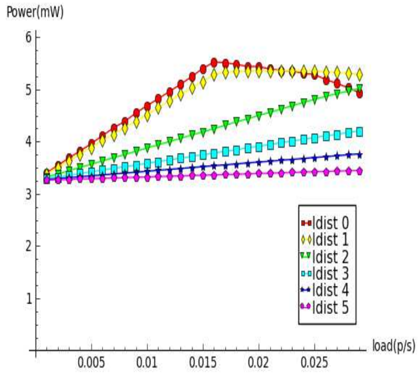
Figure 10: Throughput versus offered load. X and Y units are per node.

Figures 11(a), 11(b) and 11(c) show the average power consumed by nodes in different rings around the sink node, when running SCP-MAC(312.5 ms). The nodes in the inner rings consume more power because, in addition to sending their own data packets, they also forward traffic coming from outer rings. At low offered loads (few nodes in the network or low data generation rate), the average power consumption per node in each ring is an affine function of the load. When the offered load increases, either by adding more source nodes or by increasing the data generation rate, the network reaches saturation and this reflects in a flattening and even in a reduction of the power consumption at the sink and at the nodes of the inner rings. The ring where choking takes place actually moves outwards as the offered load increases.

Fig. 12(a), 12(b) and 12(c) show that our CT-MAC protocol behaves very similarly, except it consumes less energy than SCP-MAC(312.5ms). By allocating 32 ressources in one contention mechanism, the CT algorithm factors in some of the contention cost. The resulting saving is about 10% at high load. At low load, one would expect a similar relative saving. In fact, CT-MAC consumes about 5.5 times less energy. This is the benefit of the adaptive listening optimization described in 5.3.2: instead of listening to the quiescent medium 32



(a) 48 nodes network (b) 80 nodes network



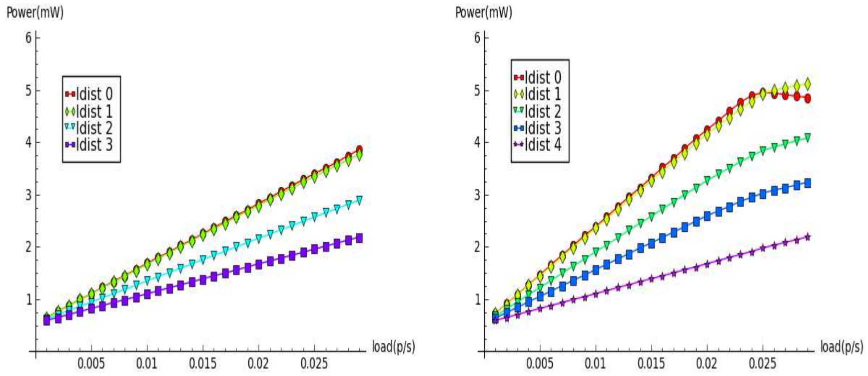
(c) 120 nodes network

Figure 11: Power consumption: SCP-MAC(312.5 ms). The units are per node.

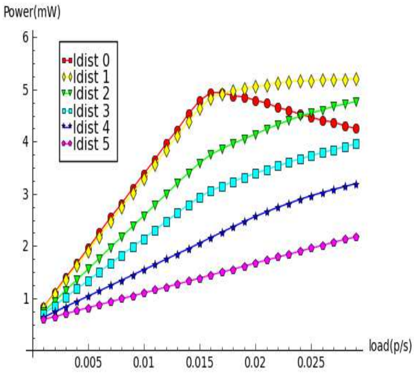
times as is the case with SCP-MAC, the nodes running CT-MAC only listen to the first few $s_{adaptive}$ slots of the announcement phase.

At the other extreme, let's compare CT-MAC with SCP-MAC(10 s). Fig. 13 is the counterpart to Fig. 12(a) and 11(a). The energy consumption is much less but saturation appears at very low loads. For loads approaching zero, CT-MAC consumes 3.5 times more than SCP-MAC(10 s). This is the price to pay for the 32x flexibility in network capacity. With a few simple calculations, we can conclude that our protocol CT-MAC consumes at very light loads about the same as SCP-MAC with a 1.8s sleep period, yet it provides a capacity 5.5x greater when and where it is needed.

For legibility, the 95% confidence intervals are not plotted on the graphs, but are discussed here. For CT-MAC and SCP-MAC(312.5 ms), their width is always less than 8% of the plotted value, for all data points. For high loads, SCP-MAC(10 s) shows an interval width up to 27% of the mean value at the sink node ($ldist = 0$). In high load scenarios, less packets indeed reach the sink node due to congestion and packet collisions, and the number of reception varies more significantly. Consumption values for other $ldist$ values are less affected since they are averaged on all nodes sharing the same logical distance. The



(a) 48 nodes network (b) 80 nodes network



(c) 120 nodes network

Figure 12: Power consumption: CT-MAC. The units are per node.

95% confidence interval width is therefore smaller, less than 10% of the plotted values.

6.3.6 Data retransmission rate

Figures 14(a) and 14(b) show the data retransmission experienced at nodes in the 120 nodes network. In the saturated regime, CT-MAC boasts only about half the data retransmission rate of SCP-MAC(312.5 ms) at nodes belonging to the first ring. This is because , in the saturated regime, all nodes always contend for the medium. CT-MAC benefits from its wider tier-1 contention window (128 slots compared to 32 for SCP-MAC). Conversely, at low offered loads, SCP-MAC has a lower retransmission rate. The explanation is that, at low loads, nodes only seldom contend for the medium. SCP-MAC organizes thirty-two separate contests spread over time whereas CT-MAC organizes just one contest with 32 times as many contenders. This 32x downsampling of the contender population with SCP-MAC turns out to be more beneficial than the wider tier-1 contention window of CT-MAC. Compared to SCP-MAC(10 s), the downsampling effect does not kick in, and CT-MAC at very low loads exhibits

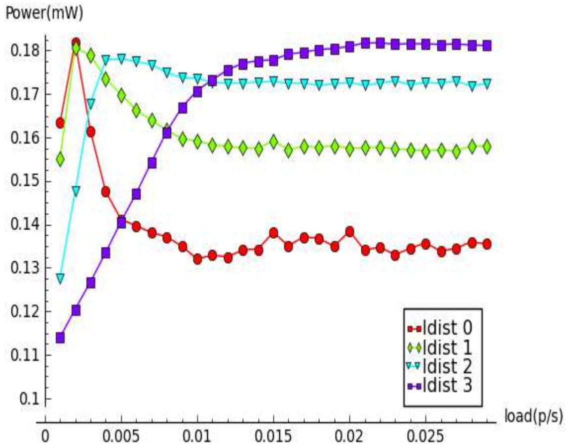
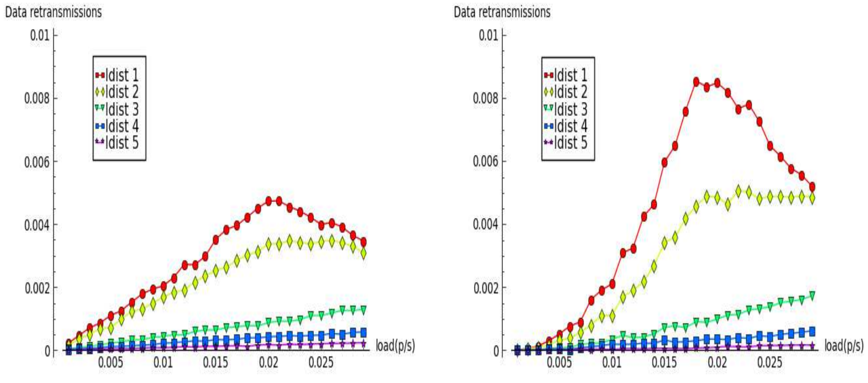


Figure 13: Power consumption: SCP-MAC(10 s), 48 nodes network. The units are per node.



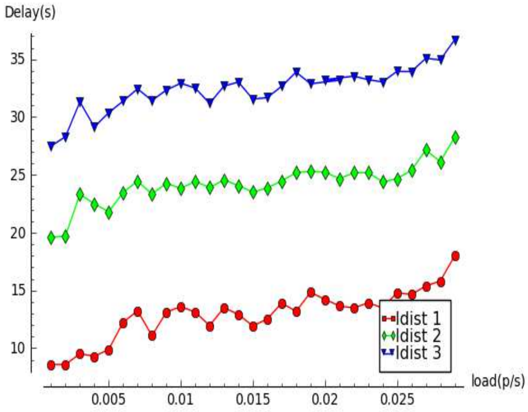
(a) CT-MAC

(b) SCP-MAC(312.5 ms)

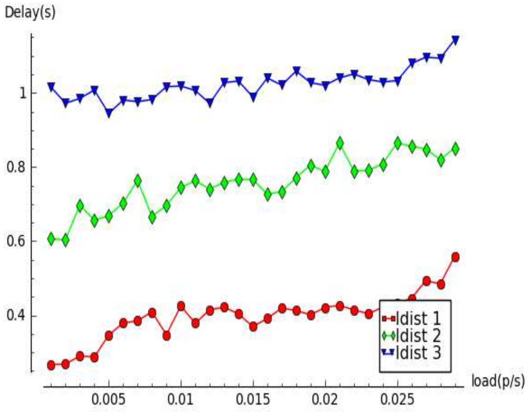
Figure 14: Data retransmission rates: 120 nodes network

a much lower retransmission rate, due to the wider tier-1 contention window. Since the channel model is the same for CT-MAC and SCP-MAC simulations, the difference in the retransmission rates directly relates to the difference in the number of collisions, which are direct effects of the underlying algorithms. Another source for retransmissions is the adaptive listening optimization described in 5.3.2: if $s_{adaptive}$ is too small, receiver nodes might stop listening for advertisements in the announcement phase, therefore ignoring ensuing transmissions. Our $s_{adaptive}$ value for this study was adjusted empirically. Its optimization is still to be explored.

The 95% confidence interval on data retransmission rates varies significantly from one scenario to another. We observe a maximum interval confidence width of up to 97% of the mean value ($ldist = 5$ for 120 nodes) with an average width of 24% of the mean value for all scenarios and $ldist$. As shown in figures 14(a) and 14(b), retransmissions are rare events (at most 50 retransmissions on all



(a) CT-MAC



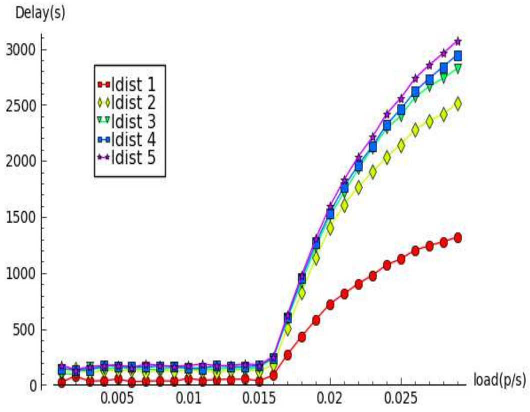
(b) SCP-MAC

Figure 15: Traffic delay (48 nodes)

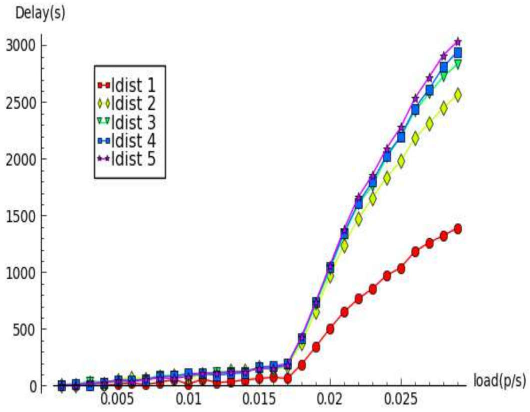
ring-1 nodes over 10 000 s for CT-MAC), and the large confidence intervals are related to the smallest numbers of retransmissions ($< 5 retransmissions$). This comforts our confidence in our results, which are accurate enough to back our claims above.

6.3.7 Traffic Delay

Figures 15 and 16 illustrate the traffic delays observed in simulation on 48 and 120 nodes networks. SCP-MAC(312.5 ms) processes the transmission requests thirty-two times in a 10 s interval, while CT-MAC regroups those requests for one more sophisticated allocation mechanism every 10 s. It therefore comes as no surprise that, at low loads and with packets generated at random instants, one hop with CT-MAC requires about 10 seconds (Fig. 15(a)) while SCP-MAC needs only about 300 ms per hop (Fig. 15(b)). SCP-MAC(10 s), not shown here, results in delays similar to that of CT-MAC. When the network goes into saturation, the end-to-end delay is related to queue lengths more than



(a) CT-MAC



(b) SCP-MAC

Figure 16: Traffic delay (120 nodes)

anything else, and SCP-MAC and CT-MAC fare equally badly under similar implementations (see Fig. 16(a) and 16(b)).

On these delay results, all the 95% confidence intervals have a width of less than 7% of the plotted value.

6.4 Results synthesis

Both single-hop and multiple-hops simulations show that the CT algorithm allows for allocating multiple logical channels to nodes competing to access the medium.

Single-hop scenario results assert of the CT algorithm near-optimality for a perfect channel scenario.

Multiple-hops scenarios show that CT-MAC is able to allocate as much logical channels, i.e. C_{log}, as SCP-MAC with a sleep-cycle C_{log} times shorter. Because CT-MAC groups the medium access contention in one concentrated phase, it reduces energy consumption when compared to a fast running SCP-MAC providing similar capacity, and it also drastically increases the capacity compared to a slow running SCP-MAC, with similar idle energy consumption. Most important, it dynamically adapts, spatially and temporally, to the load, and can do both extremes mentioned above with no parameter changes.

7 Discussion: fulfillment of the requirements

As stated in section 2.4, CT-MAC has to comply with the following requirements: $r_{heterogeneous}$, $r_{dimensioning}$, $r_{localized}$, $r_{assumption}$, $r_{adaptability}$ and $r_{fairness}$.

$r_{heterogeneous}$ the contention algorithm must cope with the traffic load heterogeneity of a collection network.

CT-MAC allows for a high network capacity while saving energy by using long sleep-cycles. Its high network capacity allows it to cope with large traffic load heterogeneity, as seen in collection networks. CT-MAC therefore meet the $r_{heterogeneous}$ requirement.

$r_{dimensioning}$ dimensioning of the contention algorithm must only depend on application requirements: traffic load requirement, delivery rate and not on network topology, e.g. network diameter.

CT-MAC does not depend on network topology information such as the network diameter and neighbors IDs. CT-MAC dimensioning only relies on traffic load, energy and delay requirements and therefore meet the $r_{dimensioning}$ requirement.

$r_{localized}$ the contention algorithm must rely exclusively on local information and operates in a decentralized manner.

CT-MAC does not rely on neighbors informations of any sort and operates in a localized manner. Therefore, CT-MAC also meet the $r_{localized}$ requirement.

$r_{assumption}$ the contention algorithm must not depend on traffic assumptions.

CT-MAC does not rely on traffic forecasts and/or traffic assumptions to operate. Therefore, CT-MAC meets the $r_{assumption}$ requirement.

$r_{adaptability}$ the contention algorithm must self-adapt to fast-varying, burst traffic.

CT dynamically allocates channels and therefore adapts to fast-varying, burst traffic.

$r_{fairness}$ the contention algorithm must grant a fair medium access to all nodes.

Each iteration of CT-MAC is independent of the others. All nodes participating to the tournaments share the same parameters and algorithm and have therefore the same probability to access the medium. CT-MAC is thus inherently fair and meet the $r_{fairness}$ requirement.

8 Future Work

CT-MAC relies on a novel competition algorithm that allocates multiple logical channels to nodes competing to access the medium. Studies on existing contention algorithms. e.g [14] and [12], show that significant improvements on collisions can be achieved by tuning the probability distribution used for choosing slots in the contention window. A near-optimal probability law has been proposed by [12] for single-slot choice algorithms that allocate a single resource. [14] exposes a similar result for binary countdown mechanisms, which we use in the tier 2 contention windows of CT-MAC. Therefore, we are currently addressing the optimization problem of reducing the collision probability in the CW_1 contention window that is used for allocating C_{log} resources.

We are also investigating how to prioritize the medium access such that we can differentiate traffic loads depending on their relative application requirements. This study includes discussions on how to bound the channel access time and how to derive deadline requirements into probability distributions for the CW_1 and $CW_{2,j}$ contention windows.

Finally, further work will address the adaptive listening time-out mechanism and the dimensioning of the $s_{adaptive}$ value for given channel conditions and traffic assumptions.

9 Conclusion

This paper makes three main research contributions. First it describes a novel distributed resource-sharing algorithm, Cascading Tournament (CT), that allows for efficient allocation of multiple resources. Second, it proposes an implementation of CT as a MAC protocol, CT-MAC, specifically designed for energy constrained, realistic Wireless Sensor Networks. Simulations compare CT-MAC against theoretical bounds and provide a fair and complete comparison with the state-of-the-art SCP-MAC protocol, under diverse conditions.

The simulation results show that CT-MAC dynamically adapts to the load, providing an unprecedented trade-off between maximum throughput and quiescent power consumption.

CT-MAC is therefore an excellent candidate to handle the spatially and temporally heterogeneous traffic of real-world Wireless Sensor Networks. Optimizations to fit various application areas are in the work.

10 Acknowledgments

This work has been partially founded by French Agence Nationale de la Recherche under contract VERSO 2009-017.

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] 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.

[3] 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.

[4] 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.

[5] 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.

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

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

[8] Sinem Coleri Ergen and Pravin Varaiya. Pedamacs: Power efficient and delay aware medium access protocol for sensor networks. *IEEE Transactions on Mobile Computing*, 5(7):920–930, 2006.

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

[10] 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.

[11] 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.

[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] The contiki operating system, http://www.sics.se/contiki/.

[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.

Contents

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