SERENA: an energy-efficient strategy to schedule nodes activity in wireless ad hoc and sensor networks

Pascale Minet, Saoucene Mahfoudh

To cite this version:

Pascale Minet, Saoucene Mahfoudh. SERENA: an energy-efficient strategy to schedule nodes activity in wireless ad hoc and sensor networks. [Research Report] RR-6388, INRIA. 2007, pp.29. inria-00196094v2

HAL Id: inria-00196094

https://hal.inria.fr/inria-00196094v2

Submitted on 13 Dec 2007

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

SERENA: an energy-efficient strategy to schedule nodes activity in wireless ad hoc and sensor networks

Pascale Minet — Saoucene Mahfoudh

N° 6388

December 2007

Thème COM

Rapport
de recherche

ISSN 0249-6399 ISRN INRIA/RR--6388--FR+ENG

SERENA: an energy-efficient strategy to schedule nodes activity in wireless ad hoc and sensor networks

Pascale Minet , Saoucene Mahfoudh

Thème COM — Systèmes communicants
Équipe-Projet Hipercom

Rapport de recherche n° 6388 — December 2007 — 29 pages

Abstract: In wireless ad hoc and sensor networks, an analysis of the node energy consumption distribution shows that the largest part is due to the time spent in the idle state. This result is at the origin of SERENA, an algorithm to SchEdule RoutEr Nodes Activity. SERENA allows router nodes to sleep, while ensuring end-to-end communication in the wireless network. It is a localized and decentralized algorithm assigning time slots to nodes. Any node stays awake only during its slots and the slots assigned to its neighbors, it sleeps the remaining time. SERENA is based on distributed and localized two-hop coloring. The node's color is then mapped in time slot. Thus, each node is ensured to get at least one time slot, it also gets additional time slots proportionally to its traffic rate. Such a solution adapts to varying traffic rates and supports late node arrivals. A performance evaluation allows us to compare SERENA coloring algorithm with existing ones such as DLF, both in terms of number of colors and complexity. Simulation results show that SERENA enables us to maximize network lifetime while increasing the number of user messages delivered. We quantify the slot reuse and evaluate the impact of the frame size on network performance. We then study how to dimension buffers at the router nodes. Finally, we show how SERENA improves the node energy consumption distribution and maximizes the energy efficiency of wireless ad hoc and sensor networks.

Key-words: energy efficiency, node activity scheduling, network lifetime, sleeping node, spatial reuse, coloring algorithm, slot assignment, wireless ad hoc networks, sensor networks, energy consumption.

SERENA: une stratégie efficace en énergie pour ordonnancer l'activité des noeuds dans les réseaux sans fil ad hoc et les réseaux de capteurs

Résumé : Dans les réseaux sans fil ad hoc et les réseaux de capteurs, une analyse de la répartition de la consommation énergétique des noeuds montre que la plus grande part d'énergie est consommée dans l'état oisif. Ce résultat est à l'origine de SERENA, un algorithme pour ordonnancer l'activité des noeuds. SERENA permet aux noeuds routeurs de dormir tout en assurant la connectivité du réseau. C'est un algorithme localisé et décentralisé s'appuyant sur un algorithme d'assignation de slots temporels aux noeuds du réseau. Un noeud quelconque ne reste éveillé que durant ses slots et les slots attribués à ses voisins à un saut; il dort le reste du temps. SERENA utilise un algorithme distribué et localisé de coloriage à deux sauts. Un slot est ensuite associé à la couleur du noeud. Ainsi, chaque noeud est assuré de disposer d'au moins un slot. Il dispose de slots supplémentaires proportionnellement à son trafic. Cette solution s'adapte aux variations de trafic et supporte les arrivées tardives de noeuds. Une évaluation de performances permet de comparer SERENA à d'autres algorithmes de coloriage comme DLF, à la fois en termes de nombre de couleurs et complexité. Les résultats de simulation montrent que SERENA maximise la durée de vie du réseau tout en accroissant la quantité de données remises. Nous quantifions la réutilisation spatiale des slots et évaluons l'impact de la taille de la trame sur les performances du réseau. Finalement, nous montrons comment SERENA améliore la répartition de la consommation énergétique des noeuds et maximise l'efficacité énergétique des réseaux sans fil ad hoc et des réseaux de capteurs.

Mots-clés : efficacité énergétique, ordonnancement de l'activité des noeuds, durée de vie du réseau, noeud endormi, réutilisation spatiale, algorithme de coloriage, assignation de slots, réseaux mobiles ad hoc, réseaux de capteurs, consommation énergétique.

Contents

1 Introduction

With the increasing number of applications in many domains (such as detection of forest fire or seismic event, wild life protection, building and bridge monitoring, emergency rescue, target tracking, exploration mission in hostile environments and home monitoring), wireless ad hoc and sensor networks have a promising future. However, nodes in such networks can have a limited amount of energy. Moreover, this energy can be very expensive, difficult or even impossible to renew. That is why, energy efficient strategies are needed in order to maximize both network lifetime and the amount of data delivered.

The originality of this paper consists in identifying the highest energy consumption of a node and proposing a solution that improves network energy efficiency by scheduling node activity.

This paper is organized as follows. In Section 2, we first analyze the node energy consumption distribution in wireless ad hoc and sensor networks and identify the main energy costs. Section 3 is a brief overview of the state of the art related to the scheduling of wireless nodes activity. Section 4 presents a new algorithm SERENA, SchEduling RoutEr Nodes Activity, whose originality consists in allowing router nodes to sleep. Time slots are assigned to nodes in a decentralized and localized way. A node is awake only during its slots and the slots granted to its neighbors, it sleeps the remaining time. In Section 5, we evaluate the impact of SERENA on the network lifetime and the amount of user data delivered. Section 6 deals with slot reuse and the impact of the size of the periodic frame on network performance. Section 7 shows how to dimension buffers at the router nodes. Finally, Section 8 quantifies the ability of SERENA to improve energy efficiency in wireless ad hoc and sensor networks and we conclude in Section 9.

2 Analysis of the node energy consumption distribution

We focus on the distribution of the node energy consumption in a wireless ad hoc network, in order to highlight the main energy costs and then to propose a strategy for improving the network energy efficiency.

A wireless node can take four different states with regard to energy:

- *Sleeping*: the radio is turned off, and the node is not capable of detecting signals: no communication is possible. The node uses P_{sleep} that is largely smaller than any other power: the energy consumption is minimum;
- *Idle*: even when no messages are being transmitted over the medium, the nodes stay idle and keep listening the medium with P_{idle};
- *Transmitting*: node is transmitting a frame with transmission power $P_{transmit}$;
- *Receiving*: node is receiving a frame with reception power $P_{receive}$. This frame can be decoded by this node or not, it can be intended to this node or not.

State	Power value
Transmitting	$P_{transmit} = 1.3\text{W}$
Receiving	$P_{receive} = 0.9\text{W}$
Idle	$P_{idle} = 0.74\text{W}$
Sleeping	$P_{sleep} = 0.047\text{W}$

Table 1: Power value in each radio state.

In Table 1, we report the reference values of $P_{transmit}$, $P_{receive}$, P_{idle} and P_{sleep} taken from a Lucent silver wavelan PC card [1] for an IEEE 802.11b network. These values are used in the performance evaluation reported in this paper.

We now evaluate the energy consumption of wireless nodes, when a routing efficient strategy is used. This routing efficient strategy chosen consists in minimizing the energy consumed by the flow transmission in selecting the path with minimum energy dissipated, and avoiding depleted nodes.

For that, we will compute the energy dissipated by the end-to-end transmission of a flow packet, denoted $cost(flow)$. Instead of using the number of hops between source and destination to select the best path (i.e.; every link has a cost of one), as done in OLSR [2], we will use $cost(flow)$ as the criterion to choose the best path.

The energy cost of a flow on its path P is equal to:

$$cost(flow) = \sum_{i \in sender(flow)} cost_{transmission}(i),$$

where i is a sender of $flow$ on its path P.

When a transmitter transmits one packet to next hop, because of the shared nature of wireless medium, all neighbors of the source receive this packet even if it is intended to only one of them. Moreover, each node situated between transmitter range and interference range receives this packet but it cannot decode it. These two problems generate loss of energy. So to compute the energy dissipated by one transmission, we must take into account these losses as follows [3]:

$$cost_{transmission}(i) = E_{transmit} + n * E_{receive},$$

where:

- n is the number of non-sleeping nodes belonging to the interference area of the transmitter i,
- $E_{transmit} = P_{transmit} * Duration$,
- $E_{receive} = P_{receive} * Duration$,
- $Duration$ is the transmission duration of a packet.

The Routing Efficient strategy, called *RE* is based on the OLSR routing protocol [2]. In order to avoid depleted nodes, *RE* modifies the MPR (MultiPoint Relay) selection of OLSR to take into account the residual energy of nodes. These new MPRs are used to build the energy efficient routes, whereas the classical MPRs are used to optimize network flooding. For more details, the reader can refer to [4].

Simulations have been performed for different wireless networks, where the network density (the average number of neighbors per node) is fixed to 10. Nodes whose number varies from 50 to 200, are randomly distributed in the network area. Network throughput is set to 2 Mbps. The initial energy of nodes is equal to 100 Joules. User traffic consists of 30 flows, with randomly chosen sources and destinations, and a throughput of 16 Kbps. The size of a message is 512 bytes. Messages of the routing protocol are not taken into account. Each result is the average of 5 simulation runs. Figure 1 illustrates the average on all network nodes of the energy dissipated in the different states defined previously. Notice that the Receiving state has been splitted into three substates:

- Receive: when this node is the message destination,
- Overhearing: when this node is a one-hop neighbor of the transmitter but not the message destination,
- Interference: when this node is a two-hop neighbor of the transmitter.

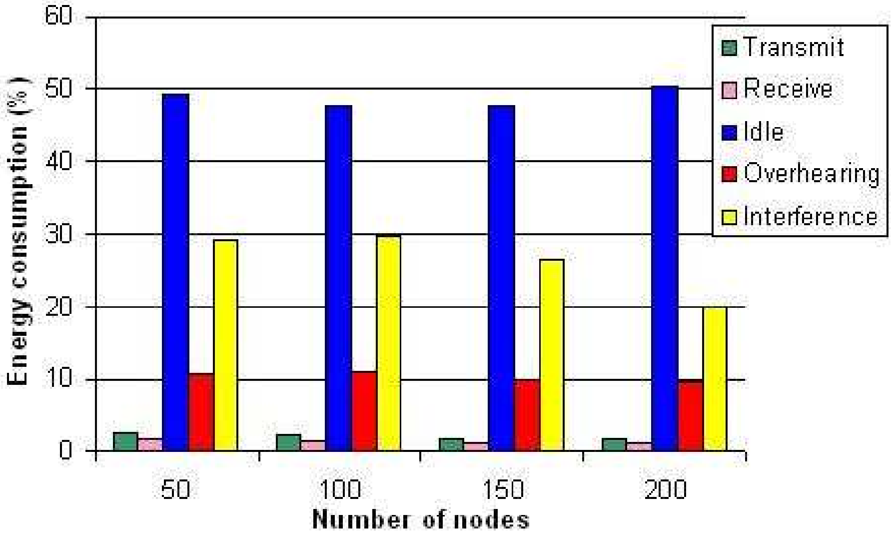

Figure 1: Distribution of node energy consumption without sleeping state.

Furthermore, it clearly appears that the highest part of energy (about 50%) is surprinsingly dissipated in the idle state. The second highest energy cost is due to interferences (about 30%), even if the *RE* routing actively contributes to energy saving. The third cost is due to overhearing (about 10%). Finally, the energy dissipated in the Transmit and Receive states are small (about 3%). From these results, we can conclude that the most efficient strategy consists in allowing nodes to sleep. Solutions already exist for non router nodes: see for instance ZigBee [5]. Our purpose is to propose a new solution allowing router nodes to sleep and to evaluate its benefit on the node enenergy consumption distribution.

3 State of the art

3.1 Energy efficient strategies

The energy constrained nature of wireless nodes requires the use of energy efficient strategies to maximize network lifetime. We can classify these strategies in four categories:

- Some strategies find the optimum node transmission power that minimizes energy consumption, while keeping network connectivity, like for instance [6, 7];
- Other strategies reduce the volume of information transferred by:
  - aggregating information with the use of clusters, like [8, 9] or without, like [10], [11], [12];
  - decreasing the frequency of information refreshment with distance, like [13];
  - avoiding to transfer information to uninterested nodes;
- Other strategies focus on energy efficient routing in order to minimize the energy consumed by the end-to-end transmission of a packet, to avoid depleted nodes and reduce the number of unsuccessful transmissions, like [14], [15], [16], [17], [18], [19], [20];
- Finally, the last category allow nodes to sleep in order to spare energy, provided that the network and application functionnalities are still ensured, like [21], [22], [23].

In this paper, we focus on the last category, nodes activity scheduling, for the reasons given in the previous section.

3.2 Node activity scheduling

All solutions scheduling node activity determine time intervals during which a node must be awake and those during which it can sleep, knowing that the application and network functions must always be satisfied. In IEEE 802.15.4 [5], the MAC layer can operate in two modes: beacon-enabled mode in which all nodes can sleep. A superframe is used to indicate to each router when it can send data and when it can sleep. This mode is used in tree and star topologies. The second mode, non-beacon mode, is used in mesh topology. In this mode only non-routing nodes can sleep. Data intended to these nodes must be kept by their parent. In this paper, we propose a solution allowing any node, even router, to sleep.

Some authors propose in [21] to extend network lifetime by dividing the network nodes in disjoint sets, such that each node set meets the network and application functions. These sets are activated successively, and at any time only the nodes of one set are active. All others nodes are in the sleep state. The problem consists in maximizing the number of disjoint sets. It has been shown NP-complete. The solution proposed is centralized. These authors have shown in [23] that network lifetime can be improved by allowing non-disjoint sets.

In [22], a distributed and localized solution is proposed. It consists in selecting a connected dominating set of sensor nodes (i.e. each node is either in this subset or is a neighbour of a node in this subset). Only the nodes of this set are active. All other nodes can change their state to sleep mode.

Some solutions take advantage of spatial reuse to determine the time intervals dedicated to node activity. Indeed, during the same time intervall two transmitters can transmit simultaneously and succesfully if they do not interfer. Spatial reuse can be obtained by means of a coloring algorithm. We will now focus on such solutions.

3.3 Centralized and distributed one-hop coloring

One-hop graph coloring has received a lot of attention from researchers (see [24, 25, 26, 27, 28, 34]). One-hop graph coloring consists in coloring each vertex of the graph such that two adjacent vertices have not the same color and the number of colors used is minimum. This problem has been shown NP-complete in [31] for the general case, whereas graphs with maximum vertex degree less than four, and bipartite graphs can be colored in polynomial time.

The first one-hop graph coloring algorithms proposed were centralized (see [24, 25]). Among the greedy algorithms (i.e. no color backtracking), Dsatur, presented in [24], where the vertex with the highest number of already colored neighbor vertices is colored first, exhibits very good performances, even if it is not optimal. It is then followed by Largest First, where the node with the highest degree is colored first.

Distributed one-hop graph coloring algorithms also exist. Some of them resort to randomization to select the color as for instance [32, 35] and [33]. The color selected by a node can be used only if it does not conflict with the colors chosen by its neighbors. Other algorithms are strictly deterministic. The authors of [28] require that the results of the distributed algorithm and its centralized version be identical for any graph. This constraint is not required in the case of wireless ad hoc and sensor nodes. In [27], it is shown that in some network configurations, Distributed Largest First uses more colors than Largest First and the reverse is true in some other network configurations. This is because with Distributed Largest First, a node is allowed to keep its selected color, not only if it has the highest degree among its neighbors (as Largest First does), but also if there is no color conflict among its neighbors. Another approach consists in finding maximum independent sets and then coloring these sets independently, as in [30] and [28], because both problems are related [34].

3.4 Complexity of a distributed coloring algorithm

The efficiency of a distributed coloring algorithm, [27, 34], can be evaluated by:

- the number of colors needed to color a graph G: closer this number to the chromatic number of G, more efficient the algorithm.
- its time complexity, expressed in the case of a distributed algorithm, by the maximum number of rounds needed to color each node. A round is defined such that every node can:
  - send a message to all its one-hop neighbors,
  - receive the messages sent by them,
  - perform some local computation based on the information contained in the received messages.

Let n be the number of vertices and Δ the largest vertex degree. For one-hop graph coloring, the algorithm proposed in [32] runs in $O(log\ n)$ rounds, but uses a number of colors close to Δ, whereas Distributed Largest First runs in $O(\Delta^2\ log\ n)$ rounds [27].

3.5 Distributed deterministic two-hop coloring

In this paper, we are interested in two-hop coloring. Indeed, in wireless ad hoc and sensor networks, interferences are generally assumed to be limited to two-hops. Hence, two transmitters at a distance strictly higher than two transmission range can simultaneously transmit. The coloring algorithm used must be deterministic and distributed.

To extend a deterministic algorithm of one-hop graph coloring to two-hop graph coloring can rise some difficulties. Indeed a node can communicate directly only with its one-hop neighbors. The information coming from its two-hop neighbors is received two rounds later. Let $\mathcal{N}^2(N)$ denote the set of nodes at a distance up to two-hop from N, we can distinguish two classes of algorithms:

- the simplest ones, such as the extension of Largest First, are based on identical rounds, called decision round. In a round, a node sends a message to its one-hop neighbors, this message contains its color and the colors of its one-hop neighbors. It receives the messages from its one-hop neighbors and takes a decision if it has the highest priority. Its decision is based on decisions already taken by nodes in $\mathcal{N}^2(N)$ having a higher priority than N. The priority of a node is fixed and does not depend on the round.
- the more complex ones, such as Distributed Largest First and Dsatur, alternate proposal rounds and decision rounds. To propose a color, a node N must know all the decisions taken in the previous decision rounds, by nodes in $\mathcal{N}^2(N)$. To decide, a node N must know all the decisions taken in the previous decision rounds, and all the proposals made in the previous proposal round, by nodes in $\mathcal{N}^2(N)$.

The solution we propose belongs to the first class of algorithms that is simpler to implement and requires less messages, as illustrated by the comparative performance evaluation in Section 4.2. According to the state of the art, our algorithm will be based on the highest cardinality of the set of nodes at a distance up to two-hop. In case of equal cardinality, the node identifier will be used as a secondary criterion.

3.6 Slot assignment algorithms

Slot assignment in wireless ad hoc and sensor networks consists in assigning slots to wireless nodes in such a way that the same slot is never used by two transmitters at a distance less than or equal to two hops. In such a context, slot assignment and two-hop coloring are very close. Probabilistic algorithms exist, such as [29] and [35], where a node randomly chooses a slot. It can use this slot, only if no conflict is detected on this slot up to two-hop. In [30], the number of slots allocated to a node depends on the number of colors seen by this node. All these solutions are inadequate in case of non-uniform traffic distribution, because the number of slots allocated to a node does not depend on its traffic rate.

The first deterministic solution based on slot assignment is TRAMA [36]. It consists in a neighborhood discovery protocol, a schedule exchange protocol and an adaptive election algorithm that selects the transmitter and receiver(s) for each time slot. Only nodes having data to send contend for a slot; notice however, that a node does not know which of its 1-hop and 2-hop neighbors have data to send. The node with the highest priority in its two-hop neighborhood wins the right to transmit in the slot considered. Each node declares in advance its next schedule containing the list of winning slots and for each winning slot its receiver(s). This new schedule is declared in the last slot of the current schedule. Hence, each neighbor node has to listen to this slot. In order to tolerate packet loss, a schedule summary is sent in each data packet. The adaptivity of TRAMA comes at a price: its complexity.

FLAMA [37] optimizes TRAMA for data gathering applications in sensor networks. In such an application, each node (except the sink) has one outgoing flow in the data gathering tree rooted at the sink and one incoming flow per child. The protocol is simplified both in terms of 1) message exchange (the schedule is no longer sent, the receiver is implicitly the parent of the transmitting node) and 2) processing complexity (the priority of a node is the weighted sum of its traffic rate and a pseudo-random function of the node identifier and slot number). The number of slots allocated by FLAMA to a node with a given traffic rate highly depends on node priority computation.

With regard to slot assignment, we focus on a deterministic algorithm, less complex than TRAMA and more generic than FLAMA, ensuring that:

- each node is guaranteed to receive at least one slot,
- the number of slots granted to a node depends on its traffic rate.

Moreover, late node arrivals can create conflicts. The impact of a late arrival should be limited to the neighborhood of the joining node.

4 Presentation of SERENA

4.1 Principles of SERENA

SERENA allows any router node to stay awake only during its slots and those assigned to its one-hop neighbors and to sleep all the remaining time. The number of slots assigned to a router node is proportional to its traffic rate. In any case, any router node is ensured to have at least one slot per frame period. SERENA is decentralized and localized. It adapts to traffic and topology changes. In SERENA, any node N has a priority equal to the cardinality of the set of nodes up to two-hop, denoted $\mathcal{N}^2(N)$. In case of equal cardinalities, the node with the smallest identifier wins. SERENA consists of a two-hop coloring algorithm and a slot assignment algorithm.

The aim of the two-hop graph coloring is to color all nodes with the smallest number of colors, in such a way that two different nodes at a distance less than or equal to two-hop have not the same color. A node N can select its color if and only if all nodes in $\mathcal{N}^2(N)$ with a higher priority have already selected their color. It then selects the smallest color unused in $\mathcal{N}^2(N)$.

As soon as all nodes in $\mathcal{N}^2(N)$ are colored, the slot assignment algorithm starts. Its purpose is to assign time slots to each node. Any node N computes its number of slots depending on its traffic. More precisely, node N receives the slot reserved to its color and it computes its number k' of additional slots according to the following formula:

$$k' = \lfloor \frac{traffic(N)}{\sum\limits_{i \in VisibleColor(N)} traffic(i)} * (Size - |VisibleColor(N)|) \rfloor.$$

Among these k' slots, $k' - k$ are requisitionable, where

$$k = \lfloor \frac{traffic(N)}{\sum\limits_{M \in \mathcal{N}^2(N)} traffic(M)} * (Size - |VisibleColor(N)|) \rfloor.$$

where:

- $Size$ is the size of the frame;
- $|VisibleColor(N)|$ denotes the cardinal of the set of colors visible by N up to two-hop;
- $traffic(N)$ is the bandwidth request of node N; it is computed from the traffic submitted by N on the last period and the traffic pending on N;
- $traffic(i)$ denotes the highest bandwidth request of nodes having color i up to two-hop from N. Notice that several nodes in $\mathcal{N}^2(N)$ can have the same color: this is perfectly acceptable insofar as these nodes are not at a distance less than or equal to two hops. That is why the maximum of the bandwidth request must be taken in the computation.

A node N selects its k' additional slots among the available ones if and only if all nodes in $\mathcal{N}^2(N)$ with a higher priority have already selected their additional slots. If all nodes succeed in assigning their additional slots, the algorithm is over. Otherwise, the node N that is unable to get at least its k additional slots requisitions them among the requisitionable slots used in $\mathcal{N}^2(N)$.

SERENA slot assignment algorithm is run periodically to adapt to traffic changes. The SERENA message sent by a node N to its one-hop neighbors contains the following fields:

- the node identifier,
- the node priority,
- the node color,
- the node traffic indication,
- the list of allocated slots and the indication whether they are requisitionable or not,
- and for each of its one-hop neighbors
  - the node identifier,
  - the node priority,
  - the node color,
  - the node traffic indication,
  - the list of allocated slots and the indication whether they are requisitionable or not.

Depending on the algorithm progress, some fields can be missing.

We can notice three important features of this algorithm:

- no node starvation: each node is ensured to get at least one slot per frame. In other words, the minimum throughput guaranteed to a node is equal to $Bandwidth/Size$, where $Bandwidth$ denotes the network bandwidth and $Size$ the size of the MAC slotted frame;
- node fairness: if traffic is uniformly distributed and all nodes have the same degree, they all receive the same amount of time slots;
- adaptivity to varying traffic rates and non-uniform traffic patterns, where a few number of nodes submit a high part of network traffic.

Some optimizations of SERENA are possible such as ordering the messages to transmit in a slot, in such a way that a node can detect the soonest possible that no message is destinated to it. For instance, we can:

- put the broadcast messages at the beginning of a slot,
- order the point-to-point messages by increasing destination identifier.

4.2 Performance evaluation of SERENA two-hop coloring

In this section, we compare the performance of SERENA with two other coloring algorithms: Distributed Largest First, DLF, and an algorithm close to SERENA, called the smallest identifier algorithm. In this algorithm, the priority of a node is equal to its identifier and the uncolored node N with the smallest identifier in $\mathcal{N}^2(N)$ is colored next. For each algorithm, we evaluate the number of colors used and the time required to color all nodes in the network.

Simulations have been performed for different wireless networks, where the network density (the average number of neighbors per node) is fixed to 10. Nodes whose number varies from 50 to 200, are randomly distributed in the network area. Each result is the average of 5 simulation runs.

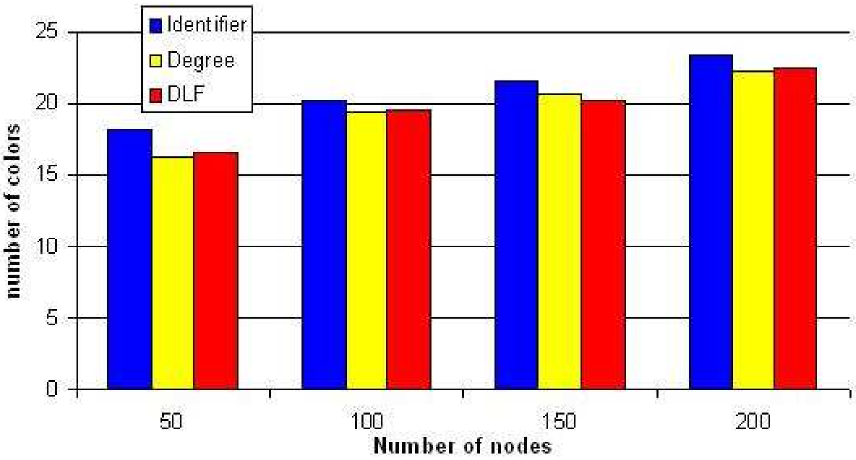

Figure 2: Number of colors used

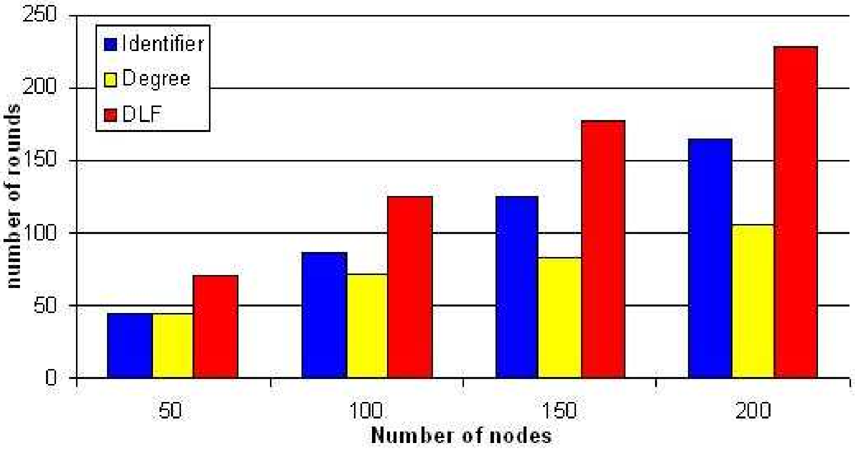

Figure 3: Number of rounds

Simulation results show that SERENA (the algorithm using the maximum degree) outperforms the algorithm with the smallest identifier, both in terms of number of colors used (see Figure 2) and time complexity expressed in number of rounds (see Figure 3). Intuitively, with SERENA (the maximum degree algorithm), the information (i.e. the selected colors) is propagated more quickly in the network: more nodes know the selected colors and can decide. If we focus on the first node selecting its color, let N be this node. Two rounds later, the maximum number of nodes up to 2-hop from N knows that this color is chosen. At least one of them, with the maximum degree, selects its color, and so on. With the identifier variant, the node N' with the smallest identifier has a degree generally less than the degree of N.

These simulation results validate the design choice of SERENA to define the priority of a node N as the cardinality of $\mathcal{N}^2(N)$, denoted $|\mathcal{N}^2(N)|$.

Compared with Distributed Largest First, DLF, SERENA shows very good performance. Both algorithms use a similar number of colors, whereas the time complexity of our algorithm is significantly lower. For 200 nodes with a density of 10, the number of rounds in DLF is 228, more than twice the number of rounds with SERENA, 105.

We now study the impact of network density, (i.e. the average number of neighbors per node) on the numbers of colors and rounds used by DLF and our algorithm. We consider a network of 100 nodes, with a node density varying from 5 to 20. Simulation results are averaged over 5 simulations and illustrated in Figures 4 and 5 for the number of colors and the number of rounds, respectively.

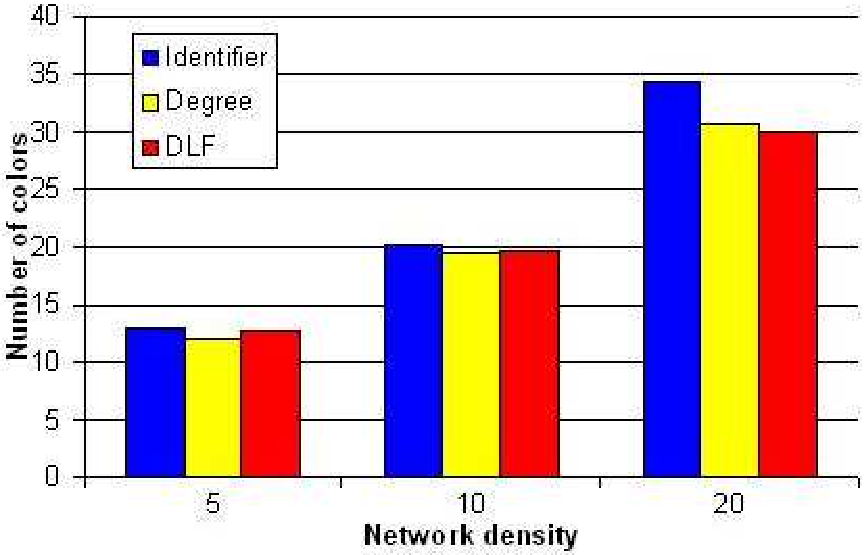

Figure 4: Number of colors used

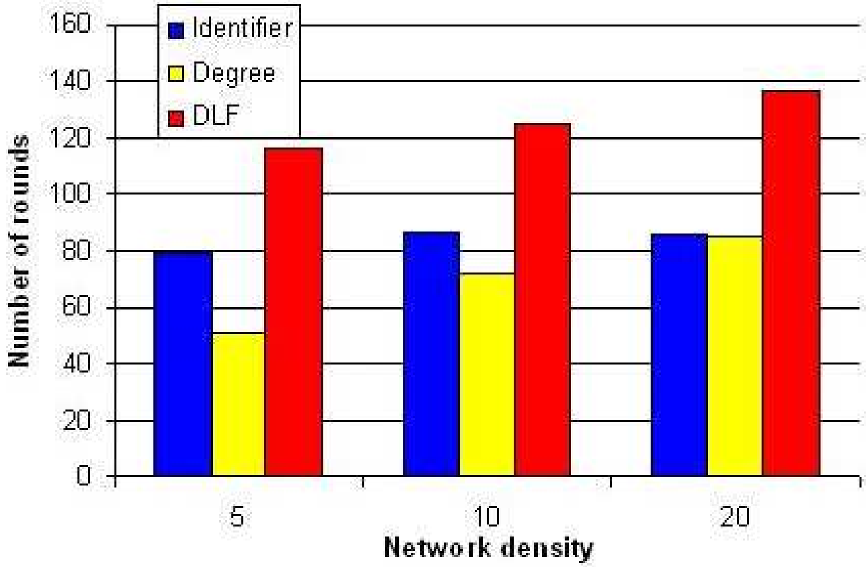

Figure 5: Number of rounds

With regard to these figures, SERENA provides an excellent performance, both in terms of:

- colors: it uses the smallest number of colors for densities 5 and 10 and is very close to the smallest number for density 20.
- rounds: the difference between SERENA and the smallest identifier algorithm tends to vanish, when the density increases. For a density of 20,

both algorithms have the same number of rounds. For all the densities studied, DLF exhibits the highest number of rounds, whereas SERNA provides the smallest one. Hence, it provides a shorter convergence time, a very interesting property in wireless ad hoc and sensor networks where energy matters.

Let c denote the number of colors assigned by SERENA, we have:

$$H1 + 1 \leq c \leq H2 + 1,$$

with $H1$ the maximum number of one-hop neighbors,
$H2$ the maximum number of neighbors up to two-hop.

With SERENA coloring algorithm, a fully connected graph is colored with its chromatic number, that is $c = H1+1$ colors. A bipartite graph is also colored with its chromatic number, but with $c = H2+1$ colors. In a one dimension network, the maximum number of rounds of SERENA coloring algorithm is reached when the node identifiers are attributed as illustrated in Figure 6. It is equal to $2 * diameter - 5$, where $diameter$ denotes the network diameter expressed in number of hops. In the best case, the node with the smallest identifier is located in the center, the number of rounds is then equal to $\lceil diameter/2 \rceil + 2$.

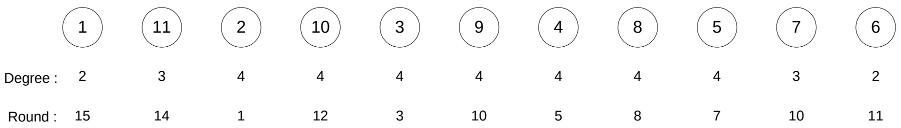

Figure 6: Number of rounds in the worst case

Compared with Figure 2, Figure 4 shows that the number of colors used by our coloring algorithm depends stongly on the network density and weakly on the number of nodes. This observation is also true to a lesser extent for the number of rounds.

4.3 Performance evaluation of SERENA slot assignment

We first justify, why in SERENA some slots are requisitionable during the slot assignment algorithm. If the slot assignment algorithm were centralized, slots would have been assigned per color. Hence, the number k' of slots would have been guaranteed to node N. Let us consider the following situation, illustrated in Figure 7.a, where color 1, used at node N_1, is reused three hops away, at node N_4. Let us assume a frame size of six slots. Node N_2 has color 2, whereas node N_3 has color 3. Three slots are allowed to node N_1, two slots to nodes N_2 and N_4, and one slot to node N_4. Colors receive their slots in increasing color order. Moreover, the number of slots allocated to color i is equal to the highest number of slots granted to a node with color i. We would have:

1. color 1 would receive slots 1, 2 and 3, node N_1 too, whereas node N_4 would only receive slots 1 and 2;
2. color 2 would receive slots 4 and 5, node N_2 too;

3. color 3 would receive slot 6 as well as node N_3.

```
   ...---N1-------N2-------N3-------N4---...
         col1     col2     col3     col1
         3 slots  2 slots  1 slot   2 slots
         prio=6   prio=3   prio=5   prio=6

a. Central.
   slots {1,2,3}  {4,5}    {6}      {1,2}


b. Distrib.
   order    1       3        2        1
   slots {1,2,3}   NOK     {4}      {5,6}
```

Figure 7: Distributed vs centralized slot assignment algorithm

However, to implement such an algorithm in a wireless network would be expensive in terms of message exchanges and computing power required by each node. That is why, we prefer a distributed and localized slot assignment algorithm, where each node locally assigns its slots. In order to limit the size of the messages exchanged, the priority used in slot assignment is this used in coloring. If the localized slot assignment assigns k' slots to node N, we can face the following situation, illustrated in Figure 7.b:

1. node N_1 selects slots 1, 2 and 3; meanwhile, node N_4 selects slots 5 and 6, because slots 1, 2 and 3 have already been attributed to neighbors with a higher priority than N_4;
2. node N_3 selects slot 4;
3. it is then impossible to node N_2 to select its two slots: there is no more available slot.

More generally, let us consider the following case where two nodes with a high priority (nodes N_1 and N_4 in the example), at a distance up to 4 hops reuse the same color and the union of the slots selected by these nodes is not equal to the slots selected by one of them (the set of slots $\{1,2,3,5,6\}$ is neither equal to $\{1,2,3\}$ nor to $\{5,6\}$. In such a case, nodes at a distance up to two hops from these two nodes (node N_2 in the example) will be unable to find k' available slots.

We now illustrate the behavior of our algorithm by a short example on a small network of 10 nodes. This network is illustrated in Figure 8, where the number besides the node identifier denotes the color assigned to this node by SERENA. The frame size is set to $2 * |\mathcal{N}^2(N)|$ slots.

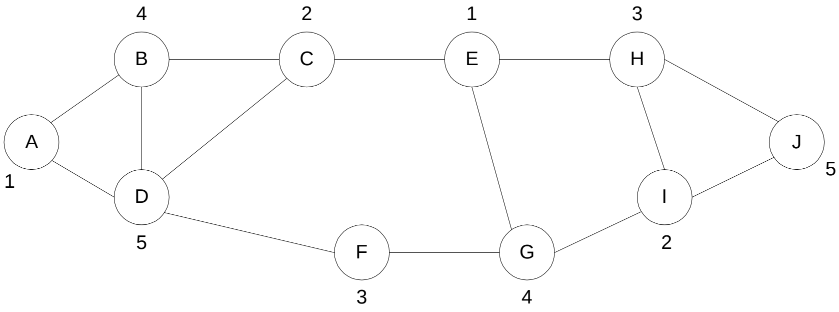

Figure 8: A wireless network and its coloring

The traffic for each node being given, the slot assignment algorithm computes the values of k and k', and assigns to each node a number of slots given in the last line of Table 2.

Node	A	B	C	D	E	F	G	H	I	J
Degree	4	5	7	6	8	7	7	5	5	4
Color	1	4	2	5	1	3	4	3	2	5
Traffic	10	20	30	10	50	10	30	50	30	10
k	0	1	1	0	2	0	1	3	2	0
k'	0	2	2	0	3	0	2	3	2	0
Slot	1	3	3	1	4	1	3	4	3	1

Table 2: Number of slots assigned per node

According to our algorithm, each node is awake during its slots and the slots attributed to its neighbors. The percentage of activity time is provided for each network node in Figure 9. The most active node is the node E that generates the highest traffic rate. Furthermore, its one-hop neighbors, C, G and H largely contribute to the network traffic, explaining why E has the highest activity. Meanwhile, nodes A and F are active at only 30%.

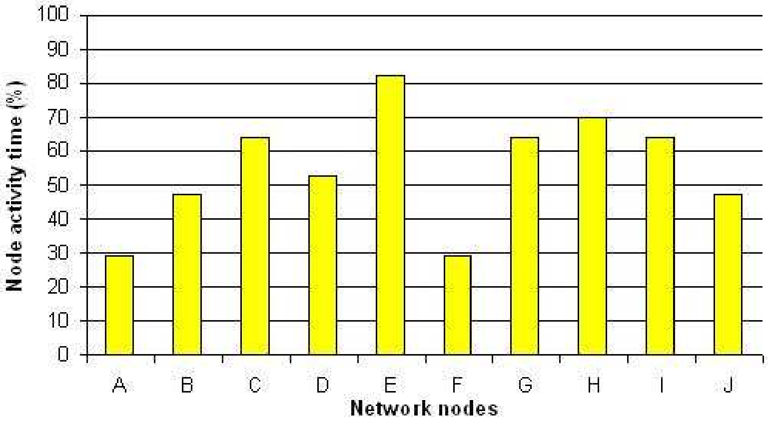

Figure 9: Node activity time

Simulation results with larger networks (50 to 200 nodes) are reported in Section 5.

4.4 Late node arrivals and node mobility

4.4.1 Assumptions

We focus on wireless and ad hoc sensor networks where:

- a high majority of nodes are present initially,
- topology changes are limited,
- mobility is also limited: only a few nodes can move and their speed is slow,
- the maximum number of nodes up to two hops is known.

It follows that the coloring algorithm is run once, at network initialization. However, late node arrivals or node mobility can cause color conflicts: two nodes at a distance less than or equal to two hops have the same color. Late node arrivals and node mobility are detected by topology changes. More precisely, some nodes detect changes in $\mathcal{N}^2(N)$. This set is refreshed by periodic exchanges of messages with the one-hop neighborhood (see rule RS3). A new joining node has no color assigned, whereas a moving node generally has one. Two types of conflict can be created:

- *first conflict type*: the moving node becomes the one-hop neighbor or two-hop neighbor of a node with the same color. Clearly, this conflict type is possible only in case of mobility.
- *second conflict type*: two nodes that are at a distance between one and two transmission ranges become two-hop neighbors due to the late arrival of a new node or node mobility.

4.4.2 General case

In the general case, the algorithm proceeds in three steps to deal with late node arrival or node mobility:

- step 1: Conflicts are first detected. The detection is based on the exchange of the colors granted to the node and its one-hop neighbors. If a node N detects that one of the node in $\mathcal{N}^2(N)$ has the same color as itself, a conflict exists.
- step 2: Detected conflicts are solved: the node with the highest priority in the conflict keeps its color, the other conflicting nodes apply rule RC1 to get a new color. In this step, the new node in case of late arrival will also receive its color.
- step 3: This will cause each node seeing a change in colors and bandwidth requests to reassign the slots according to SERENA slot assignment algorithm.

It follows that the impact of a new joining node N is limited to $\mathcal{N}^2(N)$.

This general algorithm can be simplified in some specific cases. Indeed, if there is no node mobility and any node knows all the nodes at a distance higher than one transmission range and less than two, then no conflict is created.

4.4.3 Specific case

In this specific case, we assume that there is no node mobility and any node knows all the nodes at a distance higher than one transmission range and less than two. This latter assumption can be met because:

- any node knows its GPS coordinates as well as the coordinates of any node at a distance up to two transmission ranges. This can be achieved by the routing protocol;
- or the network configuration is such that for any node N, there is no node M at a distance higher than one transmission range and less than two having no common one-hop neighbor with N.

In this case, the late arrival of a node N does not cause a color conflict and two steps are sufficient:

- step 1: A color is granted to this new node, all the other nodes keep their color. According to SERENA coloring algorithm, the new node N selects the smallest color unused in $\mathcal{N}^2(N)$.
- step 2: This will cause each node seeing a change in colors and bandwidth requests to reassign the slots according to SERENA slot assignment algorithm.

The impact of the changes are limited to the nodes in $\mathcal{N}^2(N)$.

5 Network lifetime and user data delivered

We now evaluate the performance of SERENA in various wireless ad hoc and sensor networks. More precisely, we quantify the network lifetime obtained with and without SERENA. In both cases, the routing strategy RE is used. The network lifetime is defined as the time at which a flow destination becomes unreachable. The simulation parameters are those given in Section 2. Furthermore, the slot size is fixed to 12ms and the frame consists of 80 slots. The frame size is equal to $2 * 4 * density$, where $4 * density$ is the average number of nodes up to two hops. Simulation results are averaged on 5 simulation runs.

It appears that with SERENA, the network lifetime can be increased by 100% (see for instance a network of 100 nodes in Figure 10). Notice that this excellent improvement would be useless, if during this extended network lifetime, the amount of information delivered to the user was not increased. Figure 11 confirms that with SERENA, the increase in network lifetime is followed by an increase in the delivered data.

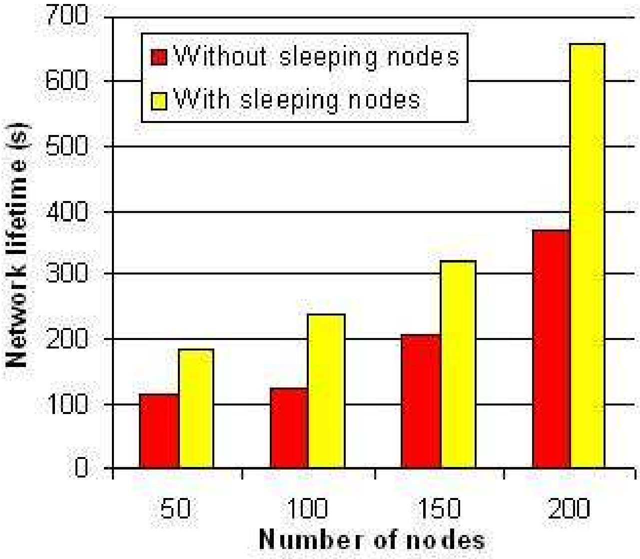

Figure 10: Network lifetime with and without SERENA.

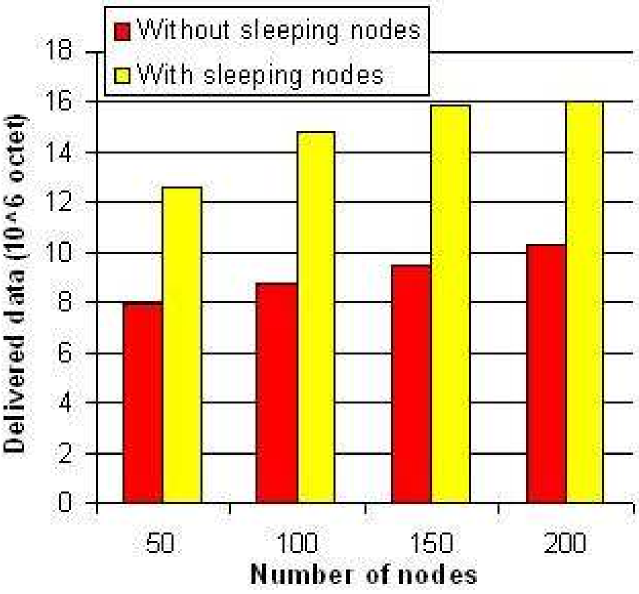

Figure 11: Amount of user data delivered with and without SERENA.

6 Slot reuse and impact of frame size

We now focus on the slot assignment algorithm and evaluate its performance by means of simulation. We consider a network of 50 nodes. The size of the node waiting queue is set to 50. The other simulation parameters are unchanged.

Traffic is not uniformly distributed over the nodes. We evaluate the number of slots obtained by a node with regard to its traffic rate. Figure 12 depicts the number of slots assigned to ten nodes.

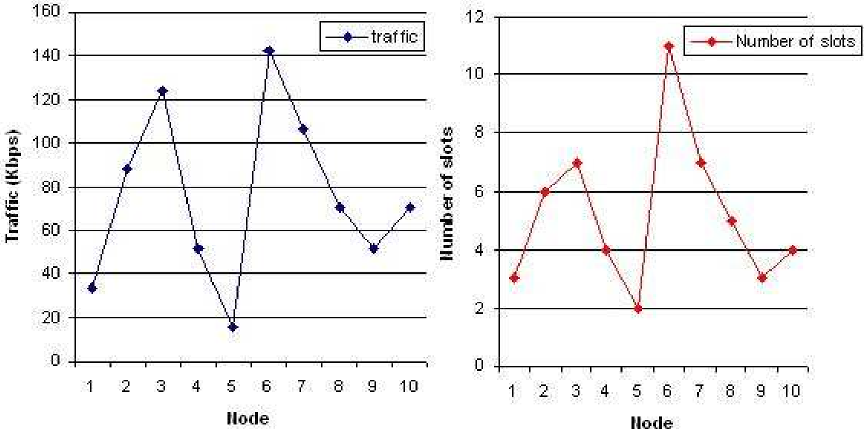

Figure 12: Slot assignment and traffic rate.

We notice that the nodes with the highest traffic rates (see nodes 3 and 6) receive the highest number of slots (7 and 11 slots respectively), whereas nodes with a small traffic rate (see nodes 1 and 9) receive few slots (3 slots for both of them). Simulation results show that as expected, SERENA assigns a slot number proportional to the traffic rate of the node.

Figure 13 depicts the slot reuse. It provides the number of slots shared by 5 nodes, down to 0 node. Two slots among the 80 are empty and 33 slots are reused by three nodes. Three quarter of slots are reused by several nodes. A slot is in average used by 2.43 nodes. This spatial reuse is obtained thanks to the SERENA two-hop coloring and slot assignment algorithms. It follows a better network efficiency.

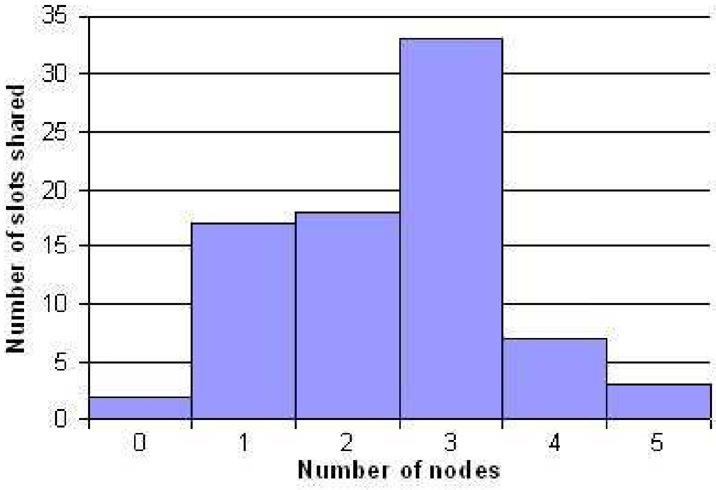

Figure 13: Slot reuse.

We now study the impact of frame size on both network lifetime and amount of messages delivered. The number of slots in the periodic frame takes the values of 40, 80, 160 and 240. As depicted on Figure 14, the highest lifetime is obtained with a frame size of 40 slots. Network lifetime then decreases and finally stabilizes for 160 slots. However, the size of 40 slots is unsatisfactory because of an unacceptable message loss rate, as shown in Figure 15. Indeed, nodes with a high traffic rate have not enough slots to send their messages and their waiting queue overflows. That is why, we recommend the use of a frame of 80 slots, corresponding to the highest lifetime obtained with no message loss due to queue overflow.

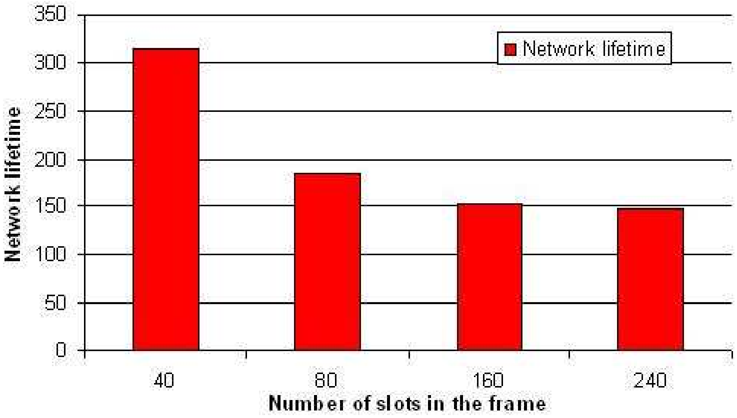

Figure 14: Impact of frame size on the network lifetime.

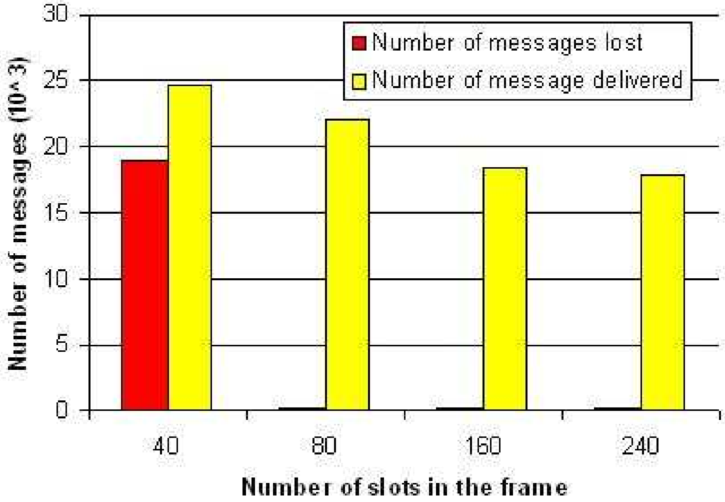

Figure 15: Impact of frame size on the delivered data.

From now on, the frame size is set to 80 slots. The impact of slot size is analyzed. We evaluate network lifetime and delivery rate, when the slot size ranges from 6ms to 48ms.

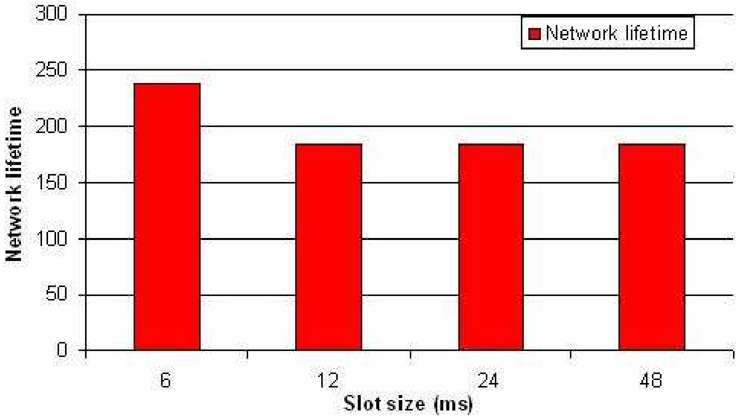

Figure 16: Impact of slot size on the network lifetime.

We observe in Figure 16 that the slot size has no impact on network lifetime, as soon as it is higher than or equal to 12 ms. A slot of 6ms provides the best network lifetime. However, as in the previous case, it leads to an unacceptable loss rate due to queue overflow (see Figure 17). Indeed, this slot size allows the transmission of only 3 messages. We also see that the size of 12 and 24ms give similar results with regard to the amount of user information delivered. With a size of 48ms, some messages are lost. Hence, the best value for the slot size is 12ms.

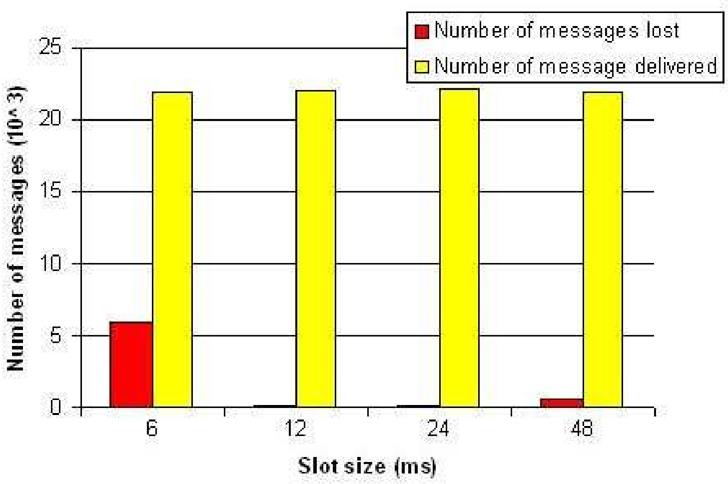

Figure 17: Impact of slot size on the delivered data.

7 Buffer dimensionning

We now evaluate the average and maximum number of messages in the waiting queue of a wireless node. The flow throughput ranges from 8kbps to 32kbps. Figure 18 depicts the results obtained, when the size of the node waiting queue is set to 50. As long as the flow throughput is lower than or equal to 16kbps, no message is lost due to queue overflow and the maximum number of waiting messages per node is less than 10 for a throughput of 8kbps and 20 for 16kbps. The average value is less than the half. With higher throughputs, the network is saturated.

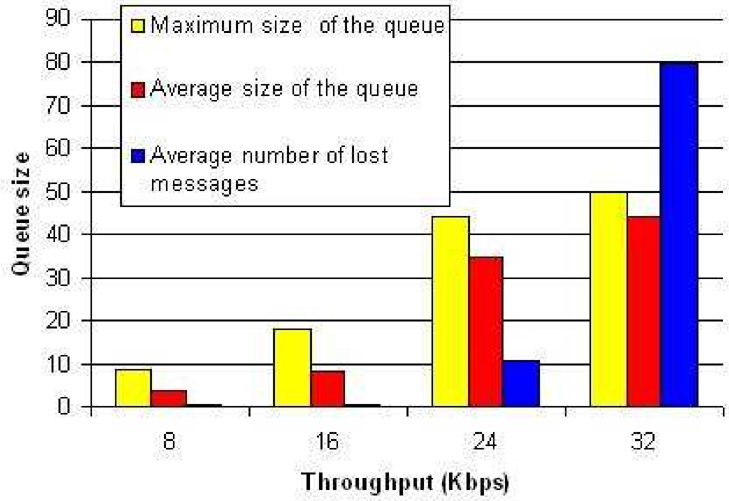

Figure 18: Average and maximum number of waiting messages per node versus flow throughput.

8 Distribution of node energy consumption

We now perform the same simulations as those described in Section 2, but now, using SERENA. The new distribution of node energy consumption is illustrated in Figure 19. The bar diagrams represent the energy dissipated in the Transmit, Receive, Idle, Overhearing and Interference states succesively for different size of networks: 50, 100, 150 and 200 nodes. The network density is set to 10.

Figure 19 shows that with SERENA, the energy dissipated in the idle state decreases to about 30% instead of 50%. This is the first benefit brought by SERENA. The second energy cost comes from overhearing: about 20% with SERENA instead of 10% without. We expect to improve the energy dissipated in the idle and overhearing states by optimizing SERENA as explained at the end of Section 4.1. The third enegy cost is due to interferences: about 1%. It was the second one without SERENA, with about 20%. This is explained by the fact that with SERENA, if neither the node nor its one-hop neighbors are transmitting, the node is sleeping. Hence, SERENA contributes to significantly reduce the interference phenomenon. This is the second benefit of SERENA.

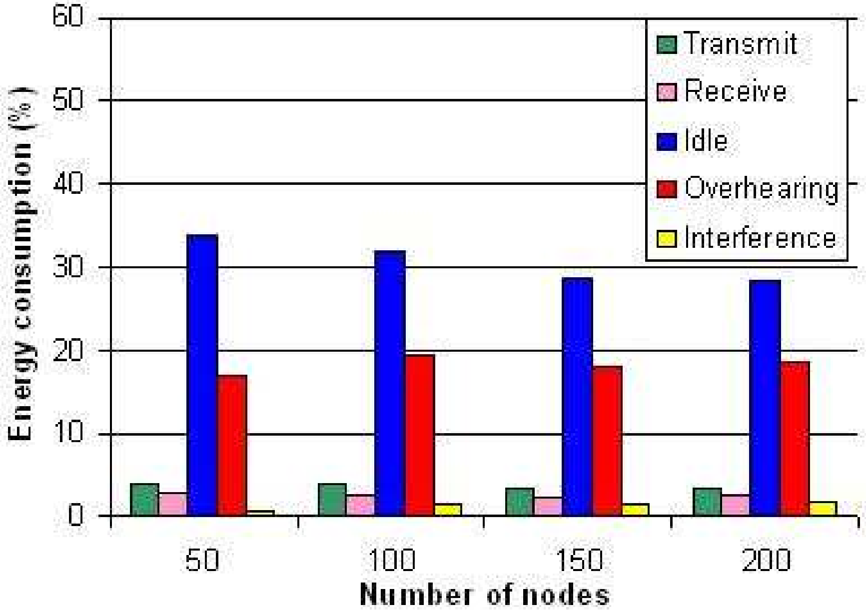

Figure 19: Distribution of energy consumption with SERENA.

9 Conclusion

In this paper, we have highlighted that in wireless and ad hoc sensor networks, most of the node energy is dissipated in the idle state. The second energy cost comes from interferences, even if a routing efficient strategy is used. SERENA in allowing a node to sleep while neither it nor one of its one-hop neighbor is transmitting, contributes to considerably improve the energy efficiency. First, the energy dissipated in the idle state is reduced up to 40%. Second, the energy loss due to interferences becomes negligible (about 1%). In parallel, the node spends more of its time in the Transmit and Receive states: the only useful states from the application point-of-view.

We have shown how to tune various parameters to maximize both the network lifetime and the amount of user data delivered. In conclusion, SERENA achieves a more efficient usage of node energy.

References

[1] L. Feeney, M. Nilson, *Investigating the energy consumption of a wireless network interface in an ad hoc networking environment*, IEEE INFOCOM 2001, Anchorage, Alaska, April 2001.

[2] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, L. Viennot, *Optimized Link State Routing Protocol*, RFC 3626, IETF, 2003.

[3] G. Allard, P. Minet, D.Q. Nguyen, N. Shresta, *Evaluation of the energy consumption in MANET* Adhoc-Now 2006, Ottawa, Canada, August 2006.

[4] S. Mahfoudh, P. Minet, *A comparative study of energy efficient routing strategies based on OLSR*, INRIA Research report 6374, https://hal.inria.fr/inria-00192899, November 2007.

[5] IEEE, IEEE 802.15.4: Wireless Medium Access Control (MAC) and Physical layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Computer Society, LAN/MAN standards committee, October 2003.

[6] F. Ingelrest, D. Simplot-Ryl, I. Stojmenovic, *Optimal Transmission Radius for Energy Efficient Broadcasting Protocols in Ad Hoc Networks*, IEEE Transactions on Parallel and Distributed Systems, June 2006.

[7] M. Cardei, J. Wu, S. Yang, *Topology Control in Ad hoc Wireless Networks with Hitch-hiking*, First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON04), October 2004.

[8] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, *Energy-efficient communication protocol for wireless microsensor networks*, the 33rd Annual Hawaii International Conference on System Sciences (HICSS '00), Maui, Hawaii, USA, vol. 2, pp. 3005â3014, January 2000.

[9] D. Xia, N. Vlajic, *Near-optimal node clustering in wireless sensor networks for environment monitoring*, AINA 2007, Niagara Falls, Ontario, Canada, May 2007.

[10] K. Kalpakis, K. Dasgupta, P. Namjoshi, *Maximum Lifetime Data Gathering and Aggregation in Wireless Sensor Networks*, University of Maryland Baltimore County (UMBC), August 2002.

[11] X. Tang, J. Xu, *Extending Network Lifetime for Precision-Constrained Data Aggregation in Wireless Sensor Networks*, IEEE INFOCOM 2006, Barcelona, Spain, April 2006.

[12] J. Kulik, W. Rabiner, H. Balakrishnan, *Adaptive Protocols for Information Dissemination in Wireless Sensor Networks*, 5th ACM/IEEE Mobicom Conference, Seattle, WA, August 1999.

[13] D.Q. Nguyen, P. Minet, *Scalabilty of the OLSR protocol with the Fish Eye extension*, ICN 2007, Sainte Luce, Martinique, April 2007.

[14] R.C. Shah, J.M. Rabaey, *Energy Aware Routing for Low Energy Ad Hoc Sensor Networks*, IEEE Wireless Communications and Networking conference (WCNC), Volume 1, pp. 17-21, March 2002.

[15] D. Ganesan, R. Govindan, S. Shenker, D. Estrin, *Highly-resilient,energy-efficient multipath routing in wireless sensor networks*, ACM SIGMOBILE Mobile Computing and Communications Review (MC2R), volume 1, no 2, 2001.

[16] A. Srinivas, E. Modiano, *Minimum Energy Disjoint Path Routing in Wireless Ad-Hoc Networks*, MOBICOM'2003 September 2003.

[17] S.-M. Senouci, G. Pujolle, *Energy efficient routing in wireless ad hoc networks*, IEEE International Conference on Communications (ICC 2004), volume 7, pp. 4057-4061, June 2004.

[18] S. Kwon, Ness B. Shroff, *Energy-Efficient Interference-Based Routing for Multi-hop Wireless Networks*, IEEE INFOCOM'2006, Barcelona, Spain, April 2006.

[19] H. Hassanein, Jing Luo, *Reliable Energy Aware Routing In Wireless Sensor networks*, Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems DSSNS, April 2006.

[20] N.Shresta, *Reception Awarness for Energy Conservation in Ad Hoc Networks*, PhD Thesis, Macquarie University Sydney, Australia, November 2006.

[21] M. Cardei, M. Thai, Y. Li, W. Wu, *Energy-efficient target coverage in wireless sensor networks*, IEEE INFOCOM 2005, Miami, Florida, March 2005.

[22] J. Carle, D. Simplot-Ryl, *Energy-Efficient Area Monitoring for Sensor Networks*, Computer, vol. 37, no. 2, pp. 40-46, Febuary, 2004.

[23] M. Cardei, D. Du, *Improving wireless sensor network lifetime through power aware organization*, ACM Journal of Wireless Networks, May 2005.

[24] D. Brelaz, *New methods to color the vertices of a graph*, Communications of the ACM, 22(4), 1979.

[25] J. Peemoller, *A correction to Brelaz's modification of Brown's coloring algorithm*, Communications of the ACM, 26(8), 1983.

[26] M. Kubale, L. Kuszner, *A better practical algorithm for distributed graph coloring*, PARELEC 2002, Warsaw, Poland, September 2002.

[27] J. Hansen, M. Kubale, L. Kuszner, A. Nadolski, *Distributed largest-first algorithm for graph coloring*, EURO-PAR 2004, Pisa, Italy, August 2004.

[28] A. Kosowski, L. Kuszner, *On greedy graph coloring in the distributed model*, Euro-Par 2006, Dresden, Germany, August 2006.

[29] L. Bao, J.J. Garcia-Luna-Aceves, *A new approach to channel access scheduling for ad hoc networks*, MOBICOM 2001, Rome, Italy, July 2001.

[30] T. Herman, S. Tixeuil, *A distributed TDMA slot assiognment algorithm for wireless sensor networks*, ALGOSENSORS 2004, Turku, Finland, July 2004.

[31] M. Garey, D. Johnson, *Computers and intractability: a guide to theory of NP-completeness*, W.H. Freeman, San Francisco, California, 1979.

[32] O. Johansson, *Simple distributed* $(\Delta + 1)$ *-coloring of graphs*, Information processing Letters, 70, 1999.

[33] I. Finoccho, A. Panconesi, R. Silvestri, *Experimental analysis of simple distributed vertex coloring algorithms*, SODA 2002, San francisco, California, January 2002.

[34] F. Kuhn, R. Watttenhofer, *On the complexity of distributed graph coloring*, PODC 2006, Denver, Colorado, USA, July 2006.

[35] C. Busch, M. Magdon-Ismail, F. Sivrikaya, B. Yener, *Contention-free MAC protocols for wireless sensor networks*, DISC 2004, Amsterdam, Netherlands, October 2004.

[36] V. Rajendran, K. Obraczka, J.J. Garcia-Luna-Aceves, *Energy-efficient, collision-free medium access control for wireless sensor networks*, Sensys'03, Los Angeles, California November 2003.

[37] V. Rajendran, J.J. Garcia-Luna-Aceves, K. Obraczka, *Energy-efficient, application-aware medium access for sensor networks*, IEEE MASS 2005, Washington, November 2005.

Centre de recherche INRIA Paris – Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Centre de recherche INRIA Futurs : Parc Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex
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 Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
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