PERFORMANCE COMPARISON OF DISTRIBUTED FREQUENCY ASSIGNMENT ALGORITHMS FOR WIRELESS SENSOR NETWORKS

Sonia Waharte and Raouf Boutaba
University of Waterloo, School of Computer Science, 200, University Avenue West, Waterloo, ON, Canada N2L 3G1
{swaharte, rboutaba}@bbcr.uwaterloo.ca

Abstract: Minimizing energy consumption of network operations remain a major concern in wireless sensor networks due to the limited energy capacity embedded in sensor nodes. Clustering has been proposed as a potential solution to address this issue, some nodes being responsible for the data gathering of nodes located in their vicinity. However, in order to avoid inter-cluster interference, neighboring clusters must acquire different frequencies. As the specific constraints of wireless sensor networks favor a distributed approach, we analyze modified versions of distributed backtracking, distributed weak commitment and randomized algorithms with a focus on energy consumption. In this context, we find that a heuristic may achieve better results than backtracking-based algorithms.

Keywords: wireless sensor networks; frequency assignement.

1. INTRODUCTION

In exploring the possibilities of interconnecting the world but alleviated of the constraints of physical infrastructure, wireless sensor networks stand out as a promising technology. To embed sensing, communication and processing capabilities in tiny devices engendered a significant potential of applications, in fields as diverse as environment monitoring, target tracking, surveillance systems, etc.

A bulk of research activities in wireless sensor networks has focused on reducing energy consumption of network operations. By considering that the nodes at close proximity have redundant information and therefore, by limiting the number of nodes simultaneously active, significant energy savings can be achieved. Another solution consists in implementing energy-efficient network organization mechanisms such as clustering. With cluster formation, one major concern is to allow simultaneous transmissions between neighboring cells while minimizing data collisions. Hence, the problem consists in allocating different frequencies (or different codes) to neighboring clusters.

Although the frequency assignment problem has been largely addressed in the literature, new constraints pertaining to wireless sensor networks have been introduced, necessitating an evaluation from different perspectives. First, the lack of a centralized administration calls for a solution based on distributed algorithms. Second, the limited power supply embedded in sensor nodes necessitates the development of energy-efficient mechanisms. Current works in this context have emphasized on achieving optimal running time (i.e. minimizing overall delay), regardless the cost of energy consumption. However, although the evaluation of an algorithm may appear beneficial in terms of running time, it can be detrimental to applications such as environment monitoring, due to an excessive energy consumption resulting from a high processing cost or by a significant number of message exchanges.

Considering the critical importance of energy conservation in such sensor network applications, we present in this paper a bi-criteria analysis of distributed frequency allocation algorithms, based not only on the overall delay of frequency allocation, but also on the overall energy consumption during the data transmission process.

The rest of the paper is organized as follows. In Section II, we describe previous works addressing the frequency assignment problem in wireless sensor networks. An overview of the algorithms implemented in our study follows in Section III. The results of our evaluations are presented in Section IV. We conclude this paper with a description of some still unaddressed issues.

2. RELATED WORKS

Frequency assignment is a well-known NP-hard combinatorial problem, which can be formalized as an instance of the Graph Coloring problem. As this subject has been widely addressed in the literature and mainly for cellular networks, we refer the reader to previous publications for further details[6,8,10].

In wireless sensor networks, frequencies can be assigned in two different ways. The centralized approach considers that one node (e.g. the base station), at the root of the topology, can efficiently proceed with the frequency allocation process and distribute the results of the operation to the concerned nodes[2]. However, this implies that the node positions are known by the root node, which introduces significant scalability issues.

A more scalable approach consists in distributing the frequency allocation problem among the sensor nodes and in seeking for a solution locally. Yokoo and Hirayama[9] propose several algorithms for distributed constraints satisfaction problems: Asynchronous Backtracking, Asynchronous Weak Commitment[5] and Distributed Breakout Algorithm, adapted from the well-known backtracking algorithm to accommodate the constraints of distributed environments. The two first algorithms are studied in this paper. In Distributed Breakout Algorithm (DBA), after an initial setup phase during which the constraints are weighted according to some predetermined parameters, the nodes exchange information on the possible weight reduction resulting from a modification of their current frequency assignment. The node maximizing the weight reduction proceeds with the modification. As an extension, Distributed Stochastic Algorithm[1] improves over DBA by a stochastic change of the local frequency. However, these two algorithms present the drawback of requiring a global synchronization, which we believe is not suitable for sensor networks. Therefore, they will not be considered in our study.

An evaluation of algorithms developed for distributed constraints satisfaction problems in wireless networks has been conducted[7], with a focus on three specific problems: partition into coordinating cliques, distributed Hamiltonian cycle formation and conflict-free channel scheduling. For the latter problem, the authors implemented the asynchronous backtracking algorithm with a network topology composed of 25 nodes and they analyzed the number of satisfied instances according to the number of channels available with a variation of the transmission radius. However, no evaluation has been conducted on the convergence time or on the number of message exchanges.

Finally, a heuristic was proposed by Guo[3], in which each node sends its randomly chosen frequency to its two-hop neighbors. Upon reception of this information, the channel is removed from the local channel pool. The appropriateness of this algorithm for sensor networks is difficult to evaluate as no simulation has been performed. There is also no mechanism to handle the situation where the frequency pool of a node becomes empty. In this paper, we analyze a heuristic based on a similar idea, but modified and completed in order to satisfy the constraints of our specific problem.

3. ASYNCHRONOUS DISTRIBUTED ALGORITHMS

3.1 Problem Formulation

The frequency assignment problem can be stated as follows. Given a wireless network composed of n nodes and its topology graph, a frequency is assigned to each node with respect to the following characteristics:

- n nodes: $x_1..x_n$
- m frequencies: $f_1..f_m$
- 1 constraint: two adjacent nodes can not be allocated the same frequency
- each node is allocated only one frequency

We implemented three asynchronous algorithms, two of them being modified versions of the distributed backtracking and the weak commitment algorithms. The third algorithm is a heuristic, used as a comparison basis in order to evaluate the real benefit of the algorithms both in terms of convergence delay and energy consumption (represented as the number of messages sent and received by all the nodes). More details on the implementation are provided in the following subsections.

3.1.1 Distributed Backtracking Algorithm.

This algorithm has been adapted from the backtracking algorithm to accommodate the constraints of distributed environments. After having informed its neighboring nodes of its frequency assignment, a node waits for any message indicating either a constraint violation or a new frequency allocation. The lowest priority node must change its value first. If no satisfactory value can be found, the node notifies a higher priority node to change its local frequency. The process is repeated until the derivation of a satisfactory solution or until the realization that no solution can be found. A node priority can be set according to its identifier, the smallest identifier having the lowest priority. Each node keeps track of the frequency assignment of all the surrounding nodes (located in the same neighborhood).

Each node is characterized by a tuple (x_i, f_i), where x_i is the identifier and f_i the frequency allocated to the node. The algorithm is described in Figure 1.

3.1.2 Asynchronous Weak-Commitment search.

Static priorities introduce limitations that the Weak Commitment search tried to overcome. The approach consists in dynamically adapting nodes priorities according to their local constraints, in order to avoid an exhaustive search. The priorities are set in increasing order (the higher the value, the higher the priority). Each time a node modifies its frequency (and thus consumes energy to inform its neighbors of its new frequency allocation), it increases its priority by one. Compared to an identifier-based priority assignment, this mechanism guarantees fairness (based on the number of frequency modifications). Further extensions are envisioned by setting priorities based on the energy level or on a node willingness to participate in the frequency allocation process.

A description of the algorithm is given in Figure 2.

3.1.3 Heuristic Algorithm.

In order to analyze the efficiency of the previous algorithms, we implemented an algorithm based on a random frequency assignment. It allowed us to relax our system from the constraint of node identification which can generate significant overhead. The algorithm is based on the following principle:

```
if received $\mathrm{msg}(\mathrm{x_j}, \mathrm{f_j})$ then
    update local_table
    check_constraints
end if

procedure check_constraints
if local_table and current_value are not consistent then
    if no value in possible_values consistent with local_table then
        backtrack
    else
        select new_value from possible_values
        current_value=new_value
        send msg(identifier, new_value) to neighbors
    end if
end if

procedure backtrack
if no solution possible then
    send(no solution) with empty value;
else
    send$\mathrm{msg}(\mathrm{identifier}, \mathrm{x_i}, \mathrm{f_i}))$ where identifier refers to the local node and $x_i$ is the agent with the lowest priority
    remove $(\mathrm{x_i}, \mathrm{f_i})$ from local_table
end if
```

Figure 1. Distributed Backtracking Algorithm

```
procedure check_constraints
if local_table and current_value are not consistent then
    if no value in possible_values consistent with local_table then
        backtrack
    else
        select new_value from possible_values while minimizing the number of constraints violation with lower priority agent
        current_value=new_value
        send msg(identifier, new_value) to neighbors
    end if
end if

procedure backtrack
if no solution possible then
    send(no solution) with empty value;
else
    send msg$(\mathrm{identifier}, \mathrm{x_i}, \mathrm{f_i}))$ where $x_i$ is the agent with the lowest priority
    remove $(\mathrm{x_i}, \mathrm{f_i})$ from the local table
    current priority++
    send msg $(\mathrm{x_i}, \mathrm{f_i}, \mathrm{current\_priority})$ to neighbors
end if
```

Figure 2. Distributed Weak Commitment Algorithm

the nodes randomly choose a frequency among a predefined set, after a random waiting period. In the meanwhile, if a node receives a frequency assignment notification from a neighboring node, it updates its local table and randomly chooses a frequency in the remaining frequencies pool. If a node receives a conflicting frequency notification after it has already chosen a frequency, the receiver node has to change its frequency and has to inform its neighbors of its new frequency. If no frequency is available, the node re-initializes its local table (i.e. it removes all values previously registered from the table), picks up a random frequency and informs its neighbors of its new value. Once no local constraint is violated, the algorithm stops. A sketch of the algorithm is given in Figure 3.

```
Initialization
wait random time
if received $\mathrm{msg}(f_j)$ then
    update local_table
    check_constraints
else
    choose random frequency among possible values
end if

procedure check_constraints
if local_freq and value in local_table are not consistent then
    if no value in possible_values consistent with local_table then
        reinitialize local_table
        choose new_fr and send msg(new_fr) to neighbors
    else
        select new_fr from possible_values
        local_freq=new_fr
        send msg(local_freq) to neighbors
    end if
end if
```

Figure 3. Heuristic Algorithm

4. EVALUATION

Whereas previous studies mainly evaluated the performance of distributed algorithms based on the execution time, we believe that, in sensor networks, energy consumption is a more critical criterion. Although technological advances have been achieved in the domain of energy supplies, a sensor node's lifetime still remains constrained. It thus becomes necessary to focus not only on the execution time but also on the energy consumption during the network operations.

In the following section, we first analyze the impact of the network size on the execution time of the algorithms and the energy consumption (in terms of number of messages sent and received). Then, we study the influence of the size of the frequency group.

4.1 Assumptions

In our study, we made the following assumptions:

- The propagation delay is considered negligible compared to the transmission delay. This hypothesis is justified if we consider that the transmission radius is around 20 meters and that the propagation speed is the speed of light. The propagation delay is then in the order of the nanosecond.
- The propagation distance can not exceed 20 meters.
- For the algorithms needing node identification (distributed backtracking and weak commitment), we assume that the node identifiers are set before the network deployment and are unique.
- All the packets are successfully delivered.

4.2 Simulation Results

The following results represent an average over 50 simulation runs, with test topologies uniformly generated at random.

The energy consumption is evaluated according to the radio propagation model described by Heinzelman et al.[4], where the energy to transmit a packet $\mathrm{E_{Tx}}$ and to receive a packet $\mathrm{E_{Rx}}$ can be computed such as:

$$E_{Tx} = lE_{elec} + l\epsilon d^2$$

$$E_{Rx} = lE_{elec}$$

where $E_{elec} = 50\mathrm{nJ/bit}$, l is the packet size, $\epsilon = 100\mathrm{pJ/bit/m}^2$ and d is the transmission distance.

In the simulations, we assume a 25-byte packet size, with a 2-byte node identifier and 5 bits reserved for the frequency assignment.

Figure 4 depicts the impact of the network size on the execution time of the studied algorithms. The Backtracking algorithm and the Weak Commitment algorithm significantly outperform the Heuristic algorithm, and achieve an improvement in term of execution time around five times higher than the Heuristic algorithm for a topology of 100 nodes. As expected, the Weak Commitment algorithm also performs on average better than the Backtracking algorithm, especially when the size of the network increases.

However, from an energy perspective, the Heuristic algorithm achieves excellent results compared to the two other algorithms (Figure 5). The difference

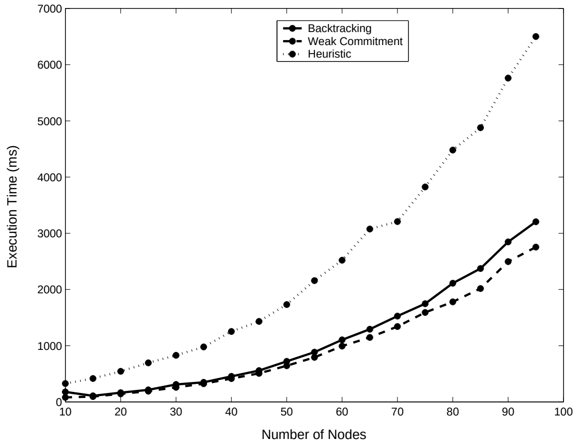

Figure 4. Impact of the network size on the execution time

with the backtracking-based algorithms becomes more significant with the increase of the network size.

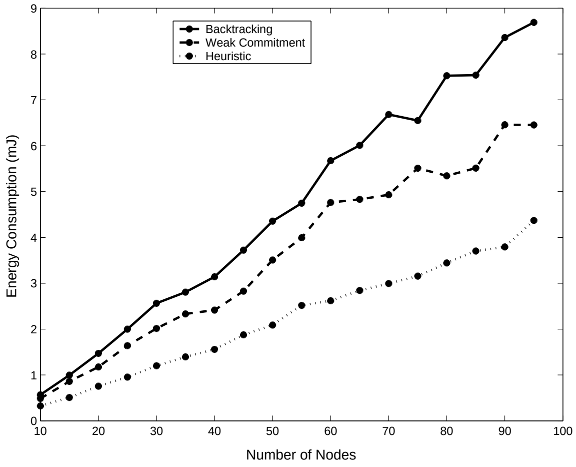

Figure 5. Impact of the network size on the energy consumption

For an average of four neighbors per node, the increase of the number of frequencies in the frequency pool has little impact on the relative performance of the algorithms (Figure 6), the Weak Commitment algorithm still performing the best.

When the number of frequencies increases, the energy performance of the Heuristic algorithm remains excellent, independently of the number of frequencies (Figure 7). The exponential decrease of the energy consumption can be explained by the fact that the probability that two nodes choose the same frequency progressively decreases when the size of the frequency pool increases.

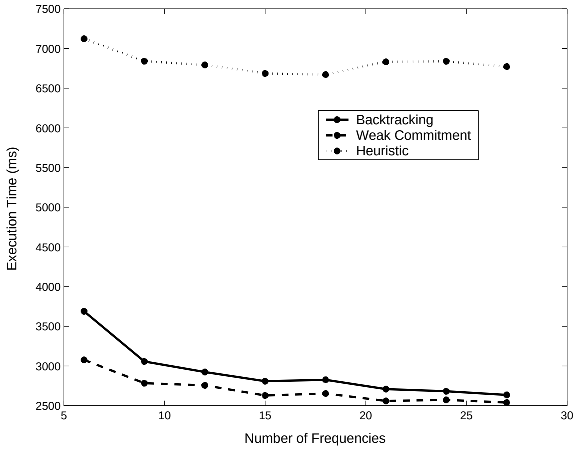

Figure 6. Influence of the number of frequencies on the execution time

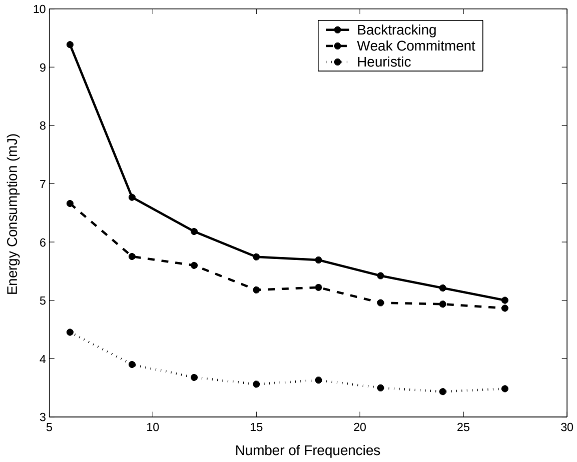

Figure 7. Influence of the number of frequencies on the energy consumption

The increase of the network density does not have any impact on the performance of the Heuristic (Figure 8). Even if its performance is relatively poor compared to the Weak Commitment and Backtracking algorithms, it presents a more stable behavior than the other two algorithms. The performance of the backtracking algorithm deteriorates with the increase of the network density. The Weak Commitment algorithm in turn shows a relatively stable behavior.

When the network density increases, the Heuristic algorithm achieves the best performance in terms of energy consumption (Figure 9).

The results obtained in the simulations are summarized in Table 1. The algorithms which perform best for each category of tests are pointed out.

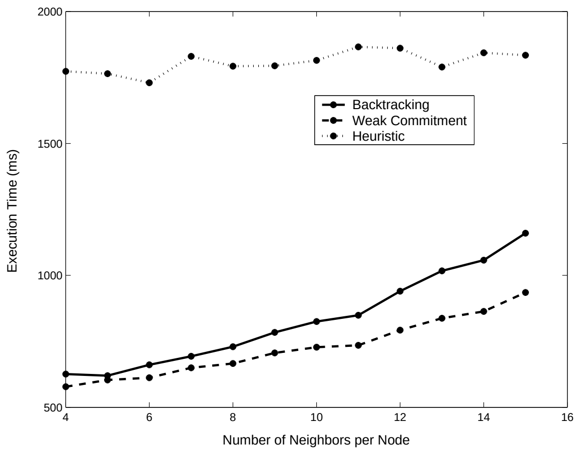

Figure 8. Influence of the network density on the execution time for a network topology of 50 nodes and 20 frequencies available

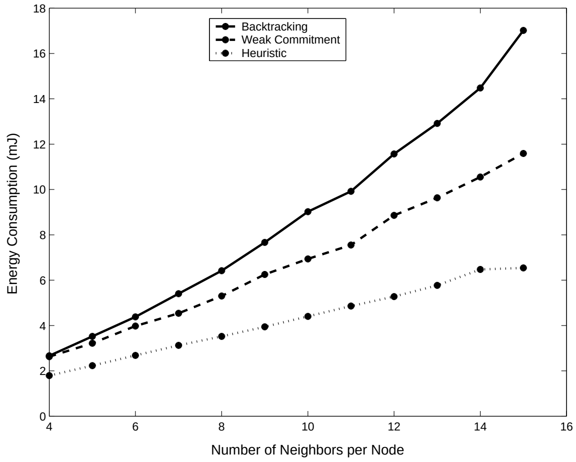

Figure 9. Influence of the network density on the energy consumption for a network topology of 50 nodes and 20 frequencies available

Table 1. Best performing algorithms according to the tested criteria

		Weak Commitment	*Heuristic*
Scalability	Execution Time	X	
	Energy		X
Frequencies	Execution Time	X	
	Energy		X
Network Density	Execution Time	X	
	Energy		X

5. CONCLUSION AND FUTURE WORK

In the context of Wireless Sensor Networks, the frequency allocation problem does not yield a unique solution. From an execution time perspective, the Weak Commitment algorithm performs the best whereas the Heuristic algorithm achieves the best results in term of energy consumption. A tradeoff between energy conservation and speed of convergence seems to be necessary. A fast running time may be critical for some applications whereas for some other applications a longer lifetime may suffice.

To determine which algorithm would be preferable, one solution would be to extend the experiments by introducing some criteria which have been ignored so far. Indeed, as stated previously, the transmission time has been neglected but may have an influence for low bandwidth implementations (in the order of several kb/s). Moreover, packet loss should be introduced for a more realistic representation of network operations. These criteria can impact the results especially when the number of messages increases. A shift of performance of the execution time may occur between the Weak Commitment algorithm and the Heuristic algorithm, but tests and further simulations are needed for a complete validation.

REFERENCES

1. *Distributed Sensor Networks, a multiagent perspective*, edited by V. Lesser, C.L. Ortiz, Jr and M. Tambe (Kluwer Academic Publications, 2003).

2. V. Annamalai, S.K.S. Gupta, , and L. Schwiebert, On tree-based convergecasting in wireless sensor networks, in: *IEEE Wireless Communications and Networking*, Vol.3, $1942 - 1947$ (2003).

3. C. Guo, L.C. Zhong, and J.M. Rabaey, Low power distributed mac for ad hoc sensor radio networks, in *IEEE Global Telecommunications Conference*, Vol. 5, $2944 - 2948$ (2001).

4. W.B. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, in *IEEE Transactions on Wireless Communications*, Vol. 1, 660670 (2002).

5. K. Hirayama and M. Yokoo, The effect of nogood learning in distributed constraint satisfaction, in *20th IEEE International Conference on Distributed Computing Systems*, $169 - 177$ (2000).

6. I. Katzela and M. Naghshineh, Channel assignment schemes for cellular mobile telecommunication systems: a comprehensive survey, *IEEE Personal Communications*, 3(3), $10 - 31$ (1996).

7. B. Krishnamachari, S. Wicker, R. Bejar, and C. Fernandez, On the complexity of distributed selfconfiguration in wireless networks, *Telecommunication Systems, Special Issue on Wireless Networks and Mobile Computing*, $169 - 177$ (2000).

8. M. Sengoku, H. Tamura, S. Shinoda, and T. Abe, Graph and network theory and cellular mobile communications, in *IEEE International Symposium on Circuits and Systems*, Vol. 4, $2208 - 2211$ (1993).

9. M. Yokoo and K. Hirayama, Algorithms for distributed constraint satisfaction: A review, *Autonomous Agents and Multi-Agent Systems*, 3(2), $185 - 207$ (2000).

10. D. Youngs, Frequency assignment for cellular radio networks, in *Fifth IEEE Conference on Telecommunications*, $179 - 183$ (1995).