



**HAL**  
open science

# Multi-channel clustering algorithm for improving performance of large-scale Wireless multi-sink sensor networks

Cheick-Tidjane Kone, Michael David, Francis Lepage

► **To cite this version:**

Cheick-Tidjane Kone, Michael David, Francis Lepage. Multi-channel clustering algorithm for improving performance of large-scale Wireless multi-sink sensor networks. 6th International Wireless Communications and Mobile Computing Conference, IWCMC 2010, Jun 2010, Caen, France. pp.CDROM. hal-00600554

**HAL Id: hal-00600554**

**<https://hal.science/hal-00600554>**

Submitted on 15 Jun 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Multi-channel Clustering Algorithm for improving Performance of Large-Scale Wireless Multi-Sink Sensor Networks

Cheick Tidjane Kone, Michael David, and Francis Lepage

Research Centre for Automatic Control of Nancy (CRAN), UMR 7039 – Nancy-Université, CNRS  
Campus sciences – BP 70239 - 54516 Vandœuvre Cedex, France  
+33(0)383684419

{Cheick-tidjane.kone, michael.david, francis.lepage} @ cran.uhp-nancy.fr

## ABSTRACT

This paper presents a simple and distributed clustering algorithm suitable for large-scale wireless sensor networks (WSNs) consisting of several thousands of sensor nodes and few sink nodes. A two-tiered hierarchical architecture is used to increase scalability and ensure performances and durability of such a system: Level 1 called sensor network is partitioned into several equilibrate clusters with one leader or sink by cluster; Level 2 also called sink network is composed by  $N$  sink nodes placed in planned manner into monitored region and is connected through IEEE 802.11 radio interfaces. A multi-channel system is used to create a cellular structure by assigning one frequency channel per cluster. We use simulation technique to evaluate and compare the impact of two distributed schemes (a single channel one and a multi-channel one) on network capacity like traffic load, energy consumption, medium access delay, end-to-end delay and data delivery ratio.

**Categories and Subject Descriptors:** C.2.1 [Computer-Communication Networks]: Network Architecture and Design – *Network topology*; C.2.3 [Computer-Communication Networks]: Network Operations – *Network management*

**General Terms:** Algorithms, Performance.

**Keywords:** Wireless sensor networks, Multiple sinks, Multiple channels, Clustering algorithm, Network performance

## 1. INTRODUCTION

Wireless Sensor Networks (WSNs) typically have little or no fixed infrastructure as compared with wired network. Their deployment is often random and could be done into large monitoring region. Assuming for example that a chemical or nuclear plant explodes and that civil security Department want to quickly and accurately monitor the area. A spreading of sensor nodes with an average density of one per 100 m<sup>2</sup> leads to deploy 10000 nodes in a monitoring area of one square kilometer but one

million nodes if this area is 10 square kilometers. In such an application case where the network density is high, the whole network traffic must remain limited to be able to ensure a high availability of such large-scale networks. Moreover, the lifetime duration, the connectivity and the scalability of the network are important design goals to ensure network performance over time and provide services at every part of the deployment area [1], [2]. Hierarchical architectures or clustering techniques are used to solve the scalability **problem**. Moreover, the use of multiple sink nodes in large-scale WSNs have been advocated as one possible way to facilitate the manageability of the network, to shorten the communication path in number of hops and to scale the network capacity and its lifetime duration. So, a WSN with multiple sinks could be partitioned into several groups or clusters with one leader (a sink node) per cluster. Also, a lot of works focus on the number and/or positions of sinks according to the application's needs in order to maximize the network capacity and its lifetime [3], [4].

For significantly optimizing energy and communication efficiency in the clusters, the multi-channel system can be used because it reduces the radio interferences, the contention to access the wireless channel and the bandwidth occupation [5]. So, we can use the cellular network approach that assigns one frequency channel per cluster while respecting the spatial channel reuse. So, all radio transceivers within a cluster operate on the same channel.

The main contribution of this paper is to propose a simple and distributed clustering algorithm which use multiple sinks and frequency channels to improve the performance and ensure a high availability of large-scale WSNs (in order of ten thousands of nodes). It creates a cellular structure by assigning one sink and one frequency channel per cluster. Then, we show that this algorithm by using multiple channels provides better network performances (like traffic load, energy consumption, medium access delay, end-to-end delay and packet delivery ratio) compared to the single channel scheme.

The rest of the paper is organized as follows: Section 2 gives an overview of the related work and specifies our objectives. Section 3 defines the network model. Section 4 describes our clustering algorithm using multiple sink nodes and multiple channels. Section 5 evaluates the performance of our algorithm. Conclusion and future work are given in Section 6.

## 2. RELATED WORK AND OBJECTIVES

In this section, we summarize the related work regarding the **problem** of heterogeneous WSNs and distinguish our approach from other works.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.  
"IWCMC'10, June 28– July 2, 2010, Caen, France. Copyright © 2010 ACM 978-1-4503-0062-9/10/06/...\$5.00"

Some works focus on a multi-sink WSN, consisting of sensor and sink nodes. In a multi-sink WSN, the number of sinks represents the number of clusters. Authors discuss the optimal number and the sink nodes' location for maximizing the coverage and the lifetime of the network. Reference [4] proposes an off-line and centralized solution for calculating the sinks' optimal number and their location while having a prior knowledge on the network's lifetime constraint and **all sensors' locations**. Another solution based on a linear programming formulation is proposed in [6]. By using a centralized graph partitioning algorithm, the reference [7] proposes to partition the network into clusters of size relatively equal in number of nodes while having also a prior knowledge of all sensors' positions. Then, they randomly define one sink per cluster and each sink can move in the area of its cluster.

For maximizing the network lifetime too, another type of heterogeneous WSN called "Wireless Sensor and Actor Network" is used **in [8]**. Actor nodes collect the data of sensor nodes and have two radio transceivers like sinks. Unlike sink nodes, actor nodes can perform other actions (such as mechanical actions) and can move. They propose to determine the number of cluster-heads (actor nodes) and their location through a mechanism based on the k-hop Independent Dominating Set (k-IDS). They suppose that all sensor and actor nodes know their location (with a mechanism like GPS). After having constructed a k-IDS, each actor must move in order to be relocated close to a dominator node. This process is highly expensive in terms of message exchanges.

We remark that the main objectives of these above proposals are to find the locations and/or the number of sink/actor nodes which maximize the network's lifetime. Moreover, their approach is centralized and the location system of all sensor nodes is a strong assumption. Such a location system integrated on each node is highly expensive for large-scale WSNs. Note that most of these proposals are tested for hundreds of nodes and that no paper describes a test bed for larger WSNs (our target is 10000 nodes).

Our approach is different from previous ones because our goal is to improve the performance of large-scale networks and also extend its lifetime duration but in a distributed manner and **regardless of the location of sensor and sink nodes**. In addition, data traffic is generated over network to address the network **operation** issue. We propose a distributed clustering algorithm which considerably reduces the traffic in large-scale WSNs. Then, we combine a multi-channel system to our approach because it has been advocated as one way of significantly scaling capacity and lifetime of the network [9]. Indeed, if the node density is very high, the level of interference is also very high, and the overall network capacity is compromised [10]. Thus, a cellular structure can be created by assigning one frequency channel per cluster while respecting the spatial channel reuse [11]. Note that a single radio transceiver might suffice for using dynamic channel switching [12] and a radio transceiver switching delay has no impact on the network performance [13]. **For instance, the latency of CC2420 radio transceiver to switch channel is about 200  $\mu$ s.**

### 3. NETWORK MODEL / ASSUMPTIONS

We consider a two-tiered heterogeneous WSN as illustrated in Figure 1. The level 1 called sensor network operates on 802.15.4 channels and the level 2 called sink network on a single 802.11 channel. All radio transceivers of network of level 1 have the same radio range. Unlike sensor nodes, we assume that sink nodes have less energy-constrained and have two radio transceivers: one for level 1 and the other for level 2. We suppose that the sensor

nodes are randomly deployed to detect, track and transmit/relay some physical events to an end-user through the sink network. The sink nodes are assumed to be placed in planned manner on the area while forming a connected sink network. We assume that the sink network uses a MANET routing protocol such as OLSR to route the physical **events** to an end-user. We also assumed that all nodes in overall network are stationary, which is **typical** for WSNs. Each sink can construct a cluster at most k hops. For combining the multi-channel system to our clustering protocol, we suppose only sinks know their frequency channel  $C_x$  which will be assigned to their cluster. This will enable sensors of a same cluster to switch their radio transceiver on the channel dedicated to cluster in order to have different intra-clusters communications (Figure 1). **Note that four 802.15.4 frequency channels ( $C_x$ ) suffice to create a pattern and facilitate the spatial channel reuse.**

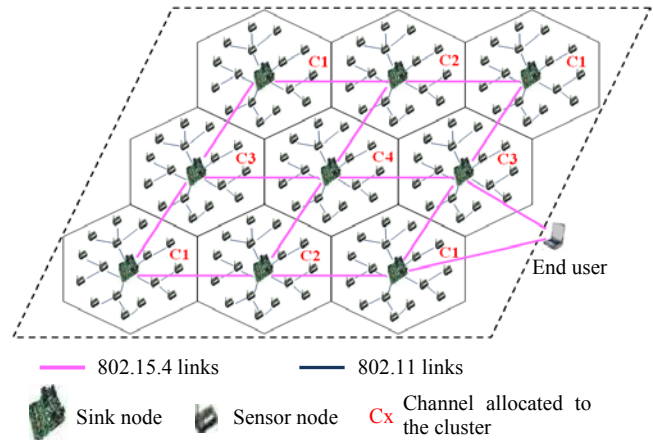


Figure 1. Proposed topology inside a large-scale WSN with multiple sinks and multiple channels.

## 4. PROPOSED ALGORITHM

### 4.1 Definition of Parent Rules

Let us consider  $Id(v)$  defining the identifier of node  $v$ . We use the **MAC address as identifier** because it is unique.  $Parent(v)$  represents the current parent of node  $v$  in a cluster-tree. Its value is **-1 when a node does not belong to a cluster-tree**.  $Hops(v)$  denotes the **current** distance in number of hops between the node  $v$  and its clusterhead (a sink);  $Hops(v) = -1$  if node  $v$  does not belong to a cluster-tree;  $Energy(v)$  represents the energy consumed rate of node  $v$ ; i.e., the ratio between the consumed amount of energy and the initial amount of energy. The parameter  $\lambda \in ]0; 1[$  defines the predefined threshold for the energy consumed rate of nodes. The  $1-Density(v)$  (also noted  $Density(v)$ ) defining the link density of node  $v$ ; i.e., the ratio between the number of links between  $v$  and its 1-neighbors, the number of links between  $v$ 's 1-neighbors and the number of node inside  $v$ 's 1-neighborhood. We choose the **link density criteria** because, in [14], the authors showed that it provides a better stability of the virtual topology than other metrics (degree, identity) facing the topological changes (i.e., a slight evolution in the neighborhood of a node).  $Weight(v)$  defining the weight of node  $v$  is a linear combination of  $Energy()$ ,  $Hops()$ ,  $Density()$ ,  $Parent()$ ,  $Id()$ . For comparing the weights, we use a lexicographical order of  $Energy()$ ,  $Hops()$ ,  $Density()$ ,  $Parent()$ ,  $Id()$ . So, this  $Weight$  function guarantees that two nodes have different weights. Given two nodes ( $u$  and  $w$ ) inside the neighborhood table of a node  $v$ . Then, the node  $v$  selects as parent between both nodes  $u$  and  $w$  the node having the highest weight.

For example the node  $v$  will pick  $u$  as parent if  $Weight(u) > Weight(w)$ , i.e. if and only if :

- $Energy(u) \leq \lambda \wedge Energy(w) > \lambda, \vee$
- $Energy(u) \leq \lambda \wedge Energy(w) \leq \lambda \wedge Hops(u) < Hops(w), \vee$
- $Energy(u) \leq \lambda \wedge Energy(w) \leq \lambda \wedge Hops(u) = Hops(w) \wedge Density(u) > Density(w), \vee$
- $Energy(u) \leq \lambda \wedge Energy(w) \leq \lambda \wedge Hops(u) = Hops(w) \wedge Density(u) = Density(w) \wedge (u = Parent(v), \text{ i.e., node } u \text{ was the parent of node } v \text{ just before this comparison}), \vee$
- $Energy(u) \leq \lambda \wedge Energy(w) \leq \lambda \wedge Hops(u) = Hops(w) \wedge Density(u) = Density(w) \wedge (Parent(v) = -1, \text{ i.e., node } v \text{ had no parent}) \wedge Id(u) < Id(v), \vee$
- $Energy(u) > \lambda \wedge Energy(w) > \lambda \wedge (u = Parent(v), \text{ i.e., node } u \text{ was the parent of node } v \text{ just before this comparison}), \vee$
- $Energy(u) > \lambda \wedge Energy(w) > \lambda \wedge (Parent(v) = -1, \text{ i.e., node } v \text{ had no parent}) \wedge Id(u) < Id(v)$

## 4.2 Cluster-tree Construction

We propose a clustering algorithm suitable for large-scale WSNs exploiting the capabilities of multiple sinks and channels to enhance the network performances. To construct a cluster-tree of the network, we define four states for a node (see Figure 2):

- Ordinary: the initial state of any node (except sink node) or a node does not belong to a cluster-tree.
- Leader: the state of a sink. It is the root node or the chief of a cluster.
- Member: node belonging to a cluster-tree.
- Border: node belonging to a cluster-tree and intermediate between clusters.

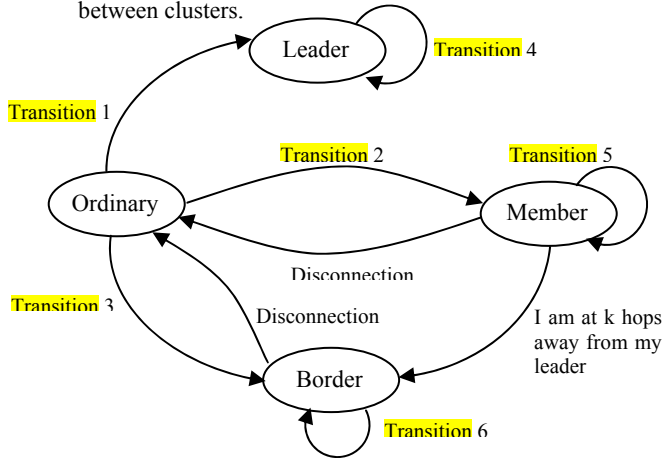


Figure 2. States changes' diagram.

The proposed clustering algorithm constructs k-cluster-tree where each sensor node within a cluster can reach the *Leader* of this cluster within at most k hops. The cluster-tree construction proceeds by waves and is message-triggered. We use only four broadcast messages types namely: *CLUSTER\_INITIATE*, *CLUSTER\_ACCEPT*, *CLUSTER\_FINISH* and *HELLO*. Our heuristic proceeds as follows. First, each node performs the neighborhood discovery process by periodically sending a packet *HELLO* to signal its presence to close nodes, construct and maintain its neighborhood table. Thus, each node will be able to compute its own metric (as the link density) and update the information (identifier, weight, etc.) of its close neighbors. After a delay, each *leader* (i.e., sink node) passes through transition 1

(Figure 2) and initiates the construction of its cluster-tree by inviting (through a broadcast of a message *CLUSTER\_INITIATE*) its close neighbors to join the cluster to which it is the chief. Then, it triggers a timer to retransmit this message if it has received no messages *CLUSTER\_ACCEPT* during this time. The message *CLUSTER\_INITIATE* contains the information about the sink node (its identifier that is also the identifier of its cluster, state, channel assigned to its cluster).

Upon receiving a *CLUSTER\_INITIATE* message from a neighbor *leader*, sensor records the information about *leader*, selects it as its parent, passes through the transition 2 (Figure 2) and becomes *member* node. Then, it broadcasts a message *CLUSTER\_ACCEPT* in order to advertise to the *leader* its membership to the cluster and also invite other close nodes to join this cluster. Then, it triggers a timer to retransmit its message *CLUSTER\_ACCEPT* if during this time, it has received no messages *CLUSTER\_ACCEPT* coming from nodes which are farther (in number of hops) of *leader* of this cluster than it. *CLUSTER\_ACCEPT* message contains the information about the source (identifier, state, density, identifier of cluster, channel assigned to cluster, parent, number of hops from clusterhead and consumed energy rate). Based on the received information of *CLUSTER\_ACCEPT* message, any node (sink or sensor) executes a procedure according its state and passes through any transition of figure 2:

- If it is a *leader* (or sink) then it updates the node's information which has sent this message. It stops the retransmission of message *CLUSTER\_INITIATE*. Finally, it passes through transition 4 and switches the current channel of its radio transceiver on the frequency channel assigned to its cluster if it had not yet changed.
- If it is a sensor node and is *member* or *border* then:
  - It updates the node's information which has sent this message. Then, it will select as parent, the node having the highest weight between its current parent and the node which has sent this message.
  - It stops the retransmission of *CLUSTER\_ACCEPT* message and pass through transition (transition 5 if it is *member* or through transition 6 if it is *border*) if and only if it is closer (in number of hops) of *leader* of this cluster than the node which has sent this message. After passing through transitions 5 or 6, it switches the current channel of its radio transceiver on the frequency channel assigned to the cluster if it had not yet changed.
- If it is sensor node and is a node *Ordinary* then :
  - It updates the node's information which has sent this message. Then, it selects as parent, during a timeout, the node having the highest weight between its close *members* neighbors.
  - If during this timeout, it had received of messages *CLUSTER\_ACCEPT* of at least two *member* nodes belonging to distinct clusters then it passes through transition 3, becomes a *border* node after this time and sends a message *CLUSTER\_FINISH*. Otherwise, it passes through transition 2, becomes a *member* and sends a message *CLUSTER\_ACCEPT* to notify its membership to the cluster and also invite its close *ordinaries* neighbors to join the cluster.

*CLUSTER\_FINISH* message contains the same information as the *CLUSTER\_ACCEPT* message except that it is only transmitted by a *border* node in order to notify its membership to a cluster and



the end to this **cluster-tree**. Upon receiving the message *CLUSTER\_FINISH*, any node executes a procedure according to its state **and passes through any transition**. For example:

- If it is a *leader* node then it updates the information of the node which has sent this message.
- If it is a sensor node and is *member* or *border* then:
  - It updates the node's information which has sent this message. Then, it stops, according to its state, the retransmission of messages *CLUSTER\_ACCEPT* or *CLUSTER\_FINISH*.
  - It passes through transition (transition 5 if it is *member* or through transition 6 if it is *border*). Then, it switches the current channel of its radio transceiver on the frequency channel assigned to the cluster if it had not still changed.

Note that some nodes (*member* or *border*) cannot switch their radio transceivers on the channels assigned to the clusters **owning** them. Thus, these nodes will disconnect of the clusters over time and will become *Ordinaries*. As these nodes know the frequency channels assigned to their clusters, they must switch their radio transceivers on these channels before their disconnection. Then, they can reconnect to a cluster upon receiving of *HELLO* messages and **thereafter can** become *member* or *border* nodes.

## 5. PERFORMANCE EVALUATION OF THE SINGLE AND MULTI-CHANNEL SCHEMES

### 5.1 Simulation Parameters

We assess the performance of our approach on the test bed developed using OPNET Modeler 15.0 [15]. In the network of level 1, we use the physical layer defined in the IEEE 802.15.4-2003 standard which operates at the 2.4 GHz frequency bands with a data rate of 250 Kb/s, a radio range of 50 meters. **The Mac sub-layer does not implement the 802.15.4 MAC protocol but a non-slotted CSMA/CA in non-beacon mode that performs better in such configuration**. We use the power consumption model from the Open-ZB web page [16]. This model is based on the Crossbow MICAz motes' specifications [17]. The network of level 2 implements the IEEE 802.11g-2003 standard with a data rate of 1Mb/s, a radio range of 500 m, in DCF mode and OLSR as routing protocol. The network consists of one end-user, 9 sinks uniformly placed and several static sensors (between 1000 and 10000 to address the scalability issue) randomly distributed over a same area of 1000 m by 1000 m. Each sink must construct a cluster at most 5 hops. All simulations are run during 7200s and over the same area. *HELLO* packets are sent periodically within a time interval  $[f_{HELLO} - 5, f_{HELLO} + 5]$ s with  $f_{HELLO} = 60$ s. For reducing network congestion, we use a crossing time of packets within the queue of the MAC layer of sensors to remove the packets delayed. **Its** value is 2s. The timeout for a sensor to switch from a state to another is 5s. In order to estimate the end-to-end delay and the packets delivery ratio of both schemes, we only consider the network consisting of 10000 sensors, 9 sinks and one end-user. Among the 10000 sensors, we randomly pick sources (between 200 and 1000 to address the network **operation** issue) which perform independent event readings. The packet generation process performed independently on each source follows an exponential distribution of average 300s, where an event-driven monitoring application is assumed. **Application packet payload is 5 bytes**. The results obtained are within 95% confidence interval.

## 5.2 Simulation results and analysis

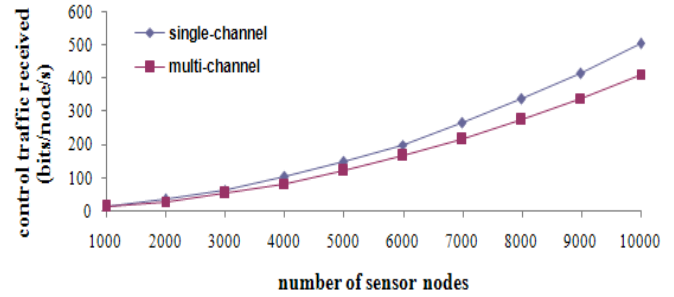


Figure 3. Average control traffic received per sensor node per second versus the network density: single channel vs multi-channel.

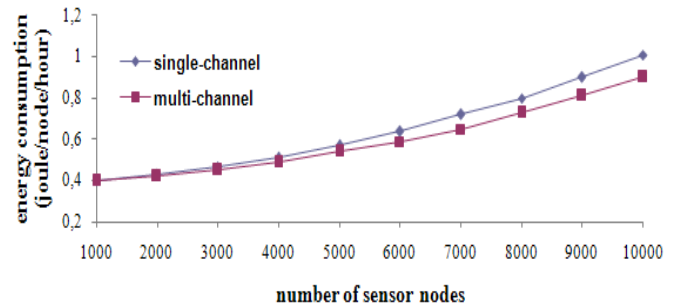


Figure 4. Average energy consumed per sensor node per hour versus the network density: single channel vs multi-channel.

We study the performance of our algorithm on both schemes (single channel **one** and multi-channel **one**). Figure 3 depicts the control traffic in bits received per sensor per second versus the network density. The control traffic consists of messages *HELLO*, *CLUSTER\_INITIATE*, *CLUSTER\_ACCEPT* and *CLUSTER\_FINISH*. We remark that the control traffic received per each sensor remains acceptable in both schemes, even with an increase of the network density. However, the control traffic observed in both schemes rapidly grows when the number of sensors increases. This suggests saturation for larger network (100000 nodes) used in the same conditions. The network traffic must remain limited to be able to ensure a high availability of such large-scale networks (low losses of packets and low delays). So, if the network density increases, the radio range of sensors or the number of clusters (sinks) should be adjust to reduce the network traffic. On the same figure, we notice that the control traffic received per each sensor is significantly lower (~18% less) in the multi-channel scheme compared to the single-channel scheme. In fact, after having switched on another frequency channel, a node has fewer neighbors (~12% less of neighbors than a node in a single channel network in our simulation results) and therefore receives fewer packets. So, the multi-channel scheme minimizes better than the single-channel scheme the bandwidth consumption, the energy consumption (Figure 4) and the medium access delay (Figure 5), since these above properties depend of the network load and the node degree (number of neighbors) over time. On the Figure 5, we remark that during 1 hour, a node in a multi-channel network consumes on average 7% less energy than a node in a single channel network. On the Figure 5, the media access delay is approximately similar in both schemes for the networks with low density. But, we also observe a difference of media access delay when the network density is high. The media access delay varies

very little in the multi-channel scheme even with an increase of the network density, whilst it rapidly grows in the single channel scheme. Thus, it is appropriate to implement a multi-channel system unlike a single channel when the network density is high.

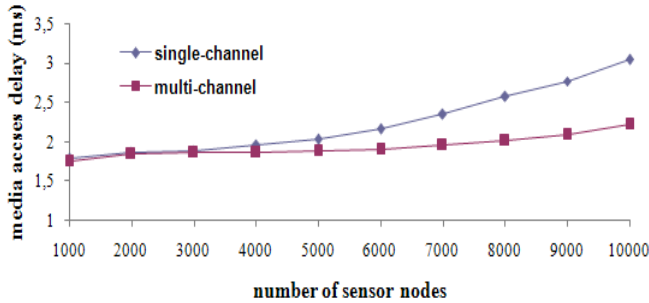


Figure 5. Average medium access delay of sensors network versus the network density: single channel vs multi-channel.

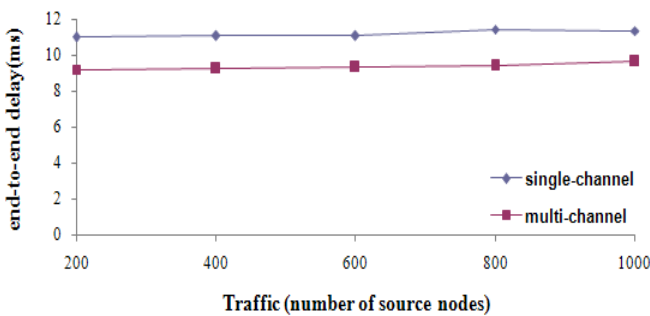


Figure 6. Average end-to-end delay: single channel vs multi-channel.

Figure 6 illustrates the average end-to-end delay of data traffic from some sensors to an end-user. This delay in the multi-channel network is better (at almost 2 ms whatever the number of source nodes) than in the single-channel network because the multi-channel scheme assures a low media access delay compared to the single channel scheme (Figure 5). This shows that multi-channel scheme performs better than the single channel one. However, the overall performances observed on both schemes are good for such a WSN (10000 nodes) since they guarantee a small overall system delay (under 12 ms) and a good delivery packets ratio over 93%.

## 6. CONCLUSION AND PERSPECTIVES

In this paper, we used multiple sinks and channels to ensure the performance of the large-scale WSN (our target was 10000 sensor nodes). We propose a simple and distributed clustering algorithm which creates a cellular system by assigning one sink and one frequency channel per cluster. Then, we discuss the performance of both schemes (a single channel one and a multi-channel one) based on this algorithm. Simulation results have shown that the multi-channel scheme performs generally better than the single channel scheme. The multi-channel scheme has a considerable influence on the energy consumption of the overall network. Moreover, it assures a low media access delay compared to the single channel scheme, even with a high network density. It also guarantees a smaller overall network delay than the single channel scheme. Thus, the multi-channel approach contributes in enhancing the performance and the lifetime duration of network.

As future works, we plan to evaluate the performance of our algorithm on a largest network topology (100000 sensor nodes). A

theoretical study should be done to determine the upper working limit of such large networks. We also intend to propose a distributed algorithm which dynamically assigns one frequency channel per cluster when the sinks are assumed randomly deployed into a monitoring region.

## 7. ACKNOWLEDGMENTS

This work was supported by OPNET Technologies, Inc. Through "Teaching with OPNET" program.

## 8. REFERENCES

- [1] I. F. Akyildiz, Y. Sankarasubramaniam, W. Su, and E. Cayirici, "Wireless sensor networks: a survey," *Computer Networks (Elsevier)*, vol. 38(4), pp. 393–422, March. 2002.
- [2] R. Verdone, D. Dardari, G. Mazzini, and A. Conti, "Wireless Sensor and Actuator Networks: Technologies, Analysis and Design," *Academic Press*, 2008.
- [3] B. McLaughlan, and K. Akkaya, "Coverage-based clustering of wireless sensor and actor networks," In: Proc. of IPCP, Istanbul, Turkey, July 2007.
- [4] E. I. Oyman, and C. Ersoy, "Multiple sink network design problem in large scale wireless sensor network networks," In : Proc. of ICC, Paris, June 2004.
- [5] X. Wang, and T. Berger, "Spatial channel reuse in wireless sensor networks," *Wirel. Netw.* 14, 2, 133-146, March. 2008.
- [6] H. Kim, Y. Seok, Y. Choi, and T. Kwon, "Optimal Multi-sink Positioning and Energy-efficient Routing in Wireless Sensor Networks," In: Kim, C. (ed.) *ICOIN 2005. LNCS*, vol. 3391, pp. 264-274. Springer, Heidelberg, 2005.
- [7] I. Slama, B. Jouaber, and D. Zeghlache, "Multiple Mobile Sinks Deployment for Energy Efficiency in Large Scale Wireless Sensor Networks," ICETE, Porto, Portugal, July 26-29, 2008.
- [8] K. Akkaya, F. Senel, and B. McLaughlan, "Clustering of wireless sensor and actor networks based on sensor distribution and connectivity," *J. Parallel Distrib. Comput.* 69, 6, pp. 573-587, June 2009.
- [9] D. Vassis, G. Kormentzas, and C. Skianis, "Performance evaluation of single and multi-channel actor communication for wireless sensor actor networks," *Ad Hoc Networks*, 4, 487-498, 2006.
- [10] P. Gupta, and P. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theo.* 46, 2, 388-404, 2000.
- [11] I. Katzela, and M. Naghshineh, "Channel assignment schemes for cellular mobile telecommunication systems: a comprehensive survey," *IEEE Personal Communications*, 3, pp. 10-31, 1996.
- [12] B. Darties, F. Theoleyre, and A. Duda, "A Divide-and-Conquer Scheme for Assigning Roles in Multi-Channel Wireless Mesh Networks," In: Proc. of IEEE LCN, Zurich, Switzerland, October 23-23, 2009.
- [13] P. Kyasanur, and N. H. Vaidya, "Capacity of multichannel wireless networks under the protocol model," *IEEE/ACM Trans. Netw., IEEE Press*, 17, pp. 515-527, 2009.
- [14] N. Mitton, A. Busson, and E. Fleury, "Self-organization in large scale ad hoc networks," In Med-Hoc-Net, Brodum, Turkey, June 2004.

[15] OPNET Technologies, Inc., Opnet Modeler Wireless Suite – ver. 15.0.A PL1, <http://www.opnet.com>.

[16] IEEE 802.15.4 OPNET Simulation Model, <http://www.open-zb.net>

[17] Crossbow motes and Stargate-Xscale, <http://www.xbow.com>