Clusters Construction Mechanism for Strictly Linear Wireless Sensor Networks

Abdourakhmane Fall, Moussa Dethié Sarr(✉), and Cheikh Sarr

Université de Thiès, BP 967 Thiès, Sénégal
{abdourakhmane.fall,mdsarr,csarr}@univ-thies.sn

Abstract. A wireless sensor network (WSN) is a set of interconnected sensors arranged in a given topology. When these sensors are arranged in a linear form, they are called linear wireless sensors networks. These particular cases of Linear WSN are today the subject of several applications like for instance monitoring border, watercourse, oil, road or rail infrastructure, gas pipe etc.

Several self-construction clusters algorithms for WSN have been proposed in the literature, these solutions, for the most part of them, are not adapted to a linear topology because having been thought under the base of nonlinear topologies. On the other hand, solutions allowing the organization in clusters for the linear WSN have been proposed. However, they do not work completely autonomously and require human intervention such as the choice of Cluster Head (CH) or the membership cluster in advance by the network administrator.

In this paper, we propose a new mechanism for self-construction clusters for strictly linear wireless sensor networks. This approach allows clustered wireless sensor organization to facilitate the routing of collected data to the sink.

The algorithm was tested on the Castalia/Omnet++ simulator and the results obtained provide a linear WSN with zero orphan node and zero singleton cluster, whatever the linear WSN cardinality, the results were also compared with the LEACH algorithm applied to a linear topology that gives a very large number of singletons nodes.

Keywords: Linear wireless sensor networks · Clustering algorithms · Automatic topology construction

1 Introduction

A wireless sensor network (WSN) is a collection of interconnected sensors for a given application. These sensors, after data collection, route them to a base station for operation. These WSN, today, are applied in several fields such as environmental (meteorology, ocean acidification, dispersion of pollutants, etc.), commercial [1], medical (implantation of micro-sensors in the human body), military (detection of chemical, biological or radiation agents) [2], etc. The WSN are arranged in a given way depending mainly

J. P. R. Thorn et al. (Eds.): InterSol 2020, LNICST 321, pp. 224–237, 2020.
https://doi.org/10.1007/978-3-030-51051-0_16

on their application. Thus, there are several topologies of wireless sensor networks. Cluster topologies are probably the best organization in terms of energy optimization of sensors [5].

Linear wireless sensor networks constitute a special case of WSN where the nodes are arranged in a linear pattern. These types of WSNs are subject of several applications: monitoring of bridges, road, rail, gas, oil and water streams [3, 4]. Linear WSNs are subject to the same constraints than non-linear WSNs. Consequently, the topologies of WSN (cluster topology among others) that have been proposed are also applicable in principle.

Regarding the self-construction algorithms proposed in the literature [5], they all rely on nonlinear topologies. Therefore, even if they are applicable to linear topologies, they are, however, not adapted to linearity. With these constraints, caused by linearity, new self-construction mechanisms in clusters specially adapted to the linear WSNs must be designed.

In this paper we propose a self-constructioning mechanism for strictly linear wireless sensor networks while avoiding the creation of orphan nodes and singletons clusters. The rest of the document will be organized as follows. In Part 2, we will present some clustering algorithms that have been proposed in the literature. In Part 3, we present the different topology proposals that have been made for linear WSNs. In Part 4, we will detail our algorithm. In Sect. 5, we will present the different simulation results we obtained. Finally in Part 6, we will end with the conclusion and perspectives for our research.

2 Clustering Algorithms in WSN

In a clustered topology, the nodes are organized into groups of nodes, each group constituting a cluster. In each cluster a node is designated as CH (Cluster Head) which role is to coordinate the cluster, to aggregate the data coming from the other nodes of the cluster, to serve as a gateway for inter-cluster transmissions etc.

WSNs are classified into two categories according to the nature of the nodes that compose them. We distinguish the homogeneous WSNs, in which all the sensors are of the same type, initially, they have the same capacities, the same responsibilities, the same reserve of energy, and we have with them the same probability of becoming CH. In these types of WSN, the CHs can periodically be changed according to selection criteria of the CHs defined by the algorithms. The second category concerns heterogeneous WSNs in which nodes do not all have the same capabilities. In these types of WSN, the CHs are chosen from among the nodes of high capacities, the other nodes are considered as simple nodes.

The clustering algorithms proposed are mainly aimed at increasing the lifetime of the WSNs. The CHs are chosen either randomly or, under the parameter base such as the residual energy of the nodes, the distance between the Sink and the nodes, the size of the clusters, etc.

In [7] the authors propose LEACH, an algorithm using a probabilistic method, composed mainly of two phases, a phase of construction of set-up phase clusters and a phase of steady-phase transmission. In the set-up phase, the CHs are randomly elected as follows: each node randomly choose a number that it compares to a threshold value T(n)

(1), if the number chosen is less than T(n) (1) the node becomes CH. The CHs send broadcast messages with maximum power, the non-CH nodes receiving these messages choose their home cluster by joining the CH with the highest transmission power.

$$T(n) = \begin{cases} \frac{P}{1-P*\left(r mod \frac{1}{P}\right)} & \text{if n} \in \text{G} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

P represents the desired percentage of clusters, r is the current round, G represents the set of nodes that have not yet been elected CH on the last 1/P rounds. In [9], LEACH-C was proposed which is an extension of LEACH. In LEACH-C all the nodes, send, initially, their position and their energy reserve to the Sink which is based on this information to elect the CHs and determine, thus, the nodes members of each cluster. With this approach the selection of CHs is done centrally. Among the extensions of LEACH are LEACH-V [10], LEACH-R [11], LEACH-M [12], TL-LEACH [15], ACTH-LEACH [17] and LEACH-WDN [18]. In [16], the authors propose HEED, where the CHs are selected according to two criteria: the residual energy of the node and the cost of CH. HEED takes place in three phases: an initialization phase, a repetition phase and a finalization phase. In the initialization phase, each node calculates the probability of becoming CH based on a probabilistic parameter and its residual energy. In the repetition phase, each node goes through several iterations until it finds the least cost of CH. In the finalization phase, the nodes decide whether they will be CH or join a cluster. In [6], the authors propose EEHC. With EEHC we have two types of CHs, necessarily selected CHs and other CHs which have volunteered, the latter are selected probabilistically. In [8], the authors propose MHRPUC in which the selection of CHs is essentially based on the residual energy of the nodes. In [13] the authors propose P-LEACH. The P-LEACH algorithm partitions the WSN into several sectors through a calculation system performed by the Sink. Indeed with P-LEACH the Sink calculates an optimal value which will constitute the number of CH of the WSN and thus the number of sectors. All sectors contain the same number of nodes, the distance between the CHs and the nodes of the same cluster is less than the radius of the network. The same sectorization concept of the WSN is used with the PASCAL algorithm [14].

In [19], the authors carry out a detailed and comparative study on the set of clustering algorithms proposed in the literature for both homogeneous and heterogeneous WSNs. In [20] also, the authors classify the different algorithms studied based on criteria such as data aggregation, the CHS rotation system, the equalization of the size of the clusters, the centrality of the CHs, the energy consumption of nodes.

However, all these clustering algorithms are essentially based on nonlinear topologies, which makes them unsuitable for linear topology WSNs. Indeed, with these types of WSNs we have additional constraints that do not exist in classical topologies. Faced with this, it becomes necessary to think about new mechanisms more adapted to these types of WSNs.

3 Topologies in Linear WSN

Several types of topologies have been proposed for the linear WSN in order to provide them with a better organization. The WSNs are categorized into two types (Fig. 1): We have strictly linear WSNs (a), in these networks we have a single linear line to sink direction. The second category of networks constitutes the linear WSNs with junction zones (b) where we have several linear lines with or without direction of the Sink.

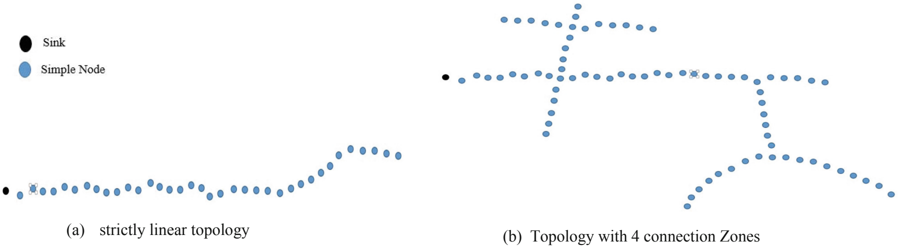

Fig. 1. Linear wireless sensor network topologies

A k-Redundant Topology

In [21] the authors propose topologies based on a k-redundant architecture in which each node of the k-neighbor network in the sense of Sink and in the opposite sense if it is a strictly linear network each node has at most 2 * k neighbors and at least k neighbors (Fig. 2). In this type of topology the availability of the WSN strongly depends on the value of k the more this value is large, the better is the network in terms of availability.

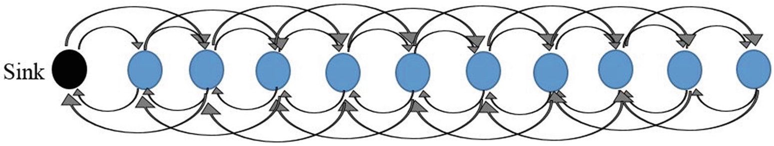

Fig. 2. 2-redondant topology [21]

B N-Level Hierarchical Topologies

In hierarchical topologies [22] not all nodes have the same functionality so we have a heterogeneous network. In [22] the authors propose a three-level hierarchical topology (Fig. 3) in which we have three types of nodes: Single sensors (NCS) that act as a data sensor only, these nodes send their data to Relay Nodes (NRD) of which role is to collect data from NCS in their vicinity. The NCSs, after collection, send the aggregated packets to Data Routing Nodes that forward the data to the Sink.

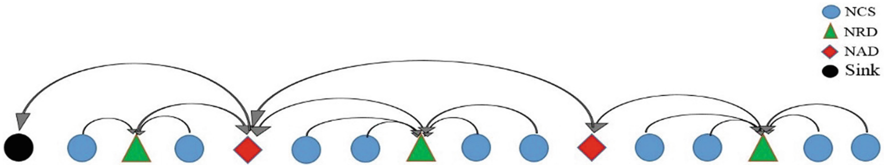

Fig. 3. Hierarchical topology with 3 levels [22]

C Cluster Topology

In [23] the authors propose a topology in logical grouping of clusters called Long Thin Wireless Sensor Network (LT-WSN). In this type of topology the linear WSN is clustered, each cluster consists of single nodes, a Cluster Head (CH) and a cluster bridge (PC). The cluster bridge and the CH define the cluster (Fig. 4). The cluster bridge is the only one capable of hosting data entering the cluster. The CH is the closest node to the Sink in terms of the number of jumps.

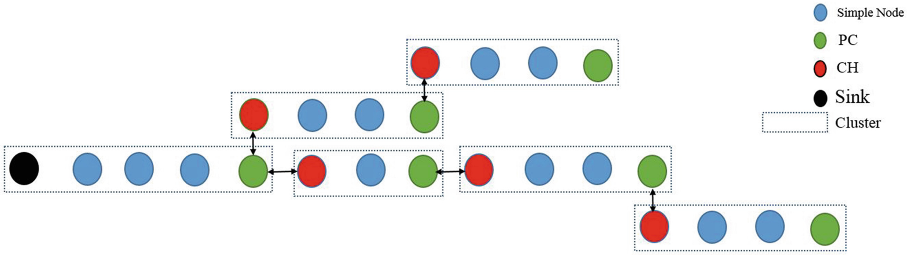

Fig. 4. Long Thin Wireless Sensor Networks [23]

The major disadvantage with cluster tree topologies is that CHs and PCs are manually selected by the network administrator.

In this session we presented the different topologies proposed for the linear WSN. This article proposes a cluster self-construction mechanism for linear WSNs that regulates among other things the problem of manual deployment of CHs.

4 Mechanism for Clusters Construction

All clustering algorithms studied in the literature (to our knowledge) are all based on non-linear architectures. Cluster tree topology proposals made in the literature require human interventions such as the choice of CHs by the network administrator. In this article, we propose **M2CRL**, a cluster self-construction mechanism for strictly linear WSNs. The algorithm assumes a 2-redundant topology where all nodes are of the same type and initially have the same capabilities and functionality.

The M2CRL algorithm comprises two phases: a first phase called the Discovery phase and a second phase called the Cluster Construction phase.

In the discovery phase (Algorithm 1), the sink discovers its neighbors and classifies them according to their proximity. Initially the sink sends in broadcast, a Hello message, in order to trigger the process of discovery, these messages are received by the neighbors of the sink. These ones in turn broadcast Hello messages that will be acknowledged, which will allow the nodes to know exactly the number of neighboring nodes v they have. After determining *v*, neighbor nodes of sink send a response to sink with the value *v*. The sink is based on the different values of v received, to evaluate the proximity with its neighbors. After the discovery phase, the sink sends a construction message to its nearest neighbor to initiate the cluster construction phase.

In the cluster construction phase, all clusters are constructed in a linear fashion. Nodes choose their cluster of membership thanks to three values: a constant Pi which constitutes the maximum number of nodes per cluster (defined by the sink), Cid which is the identifier of the cluster and Pn which constitutes the position of the node in its cluster. The construction is done thanks to a construction packet containing the three values Pi, Pn and Cid. All issued construction packets are acquitted.

Initially (Algorithm 2) the sink defines Pi that it takes in the interval [3, N/2] N being the cardinality of the WSN, it fixes Cid and Pn to 1. After defining the values it sends to the closest node a constructioning packet with the values Pi, Pn and Cid. As soon as it receives the acknowledgment from this node, it arms a new construction packet which it sends to the second closest node with the values Pi, Cid and Pn + 1. After receiving the construction packet issued by the sink, the neighbors of the sink put their Pi Pn and Cid to the values contained in the construction packet and broadcast a construction packet with the values Pi, Cid and Pn + 1. After choosing its home cluster, each node broadcasts a construction packet to its neighbors with the values Pi, Cid and Pn + 1. The other Linear WSN nodes choose their home cluster using the three principles of the following algorithm:

1. First principle of the algorithm (Algorithm 3): the position Pn of the node is equal to the maximum of Pn contained in the received construction packets Pn = max (Pn_received) and the Cid of the node is equal to the received Cid, Cid = Cid_received. This first principle is executed by the node if and only if all the received Pn are different from Pi.
2. Second principle of the algorithm (Algorithm 3): executed if one of the received Pn is equal to Pi. This principle says that if the node receives a Pn equal to Pi, it then compares the Cid contained in the packets, two scenarios are possible.

   a. if the packets contain the same Cid then the node auto-elects as Cluster Head (CH), puts its Pn to 1 and broadcast a construction packet to its neighbors with the values Pi, Cid and Pn + 1.
   b. If the received packets do not contain the same Cid then the node in question is the first neighbor of the CH, it puts its Pn to 2, keeps the maximum of Cid received Cid = max (Cid_received) and broadcasts a construction packet to its neighbors with the values Pi, Cid and Pn + 1.

3. Third principle of the algorithm (Algorithm 4): It aims to avoid singleton clusters. Indeed, after having broadcast a construction packet, the node triggers a timer which stops upon receipt of an acknowledgment. When the Timer is exhausted, i.e. no acknowledgment is received, the node self-elects as Cluster Tail (CT). If the latter is a CH, then it is a singleton Cluster, therefore the node is positioned at the father cluster by decreasing its Cid and putting its Pn to Pi + 1.

Figures 5 and 6 respectively illustrate the discovery and construction phases of the algorithm executed on a topology of eleven linearly arranged nodes.

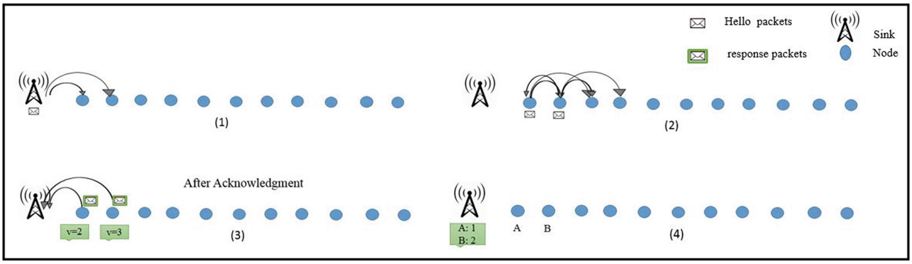

Fig. 5. M2CRL: discovery phase

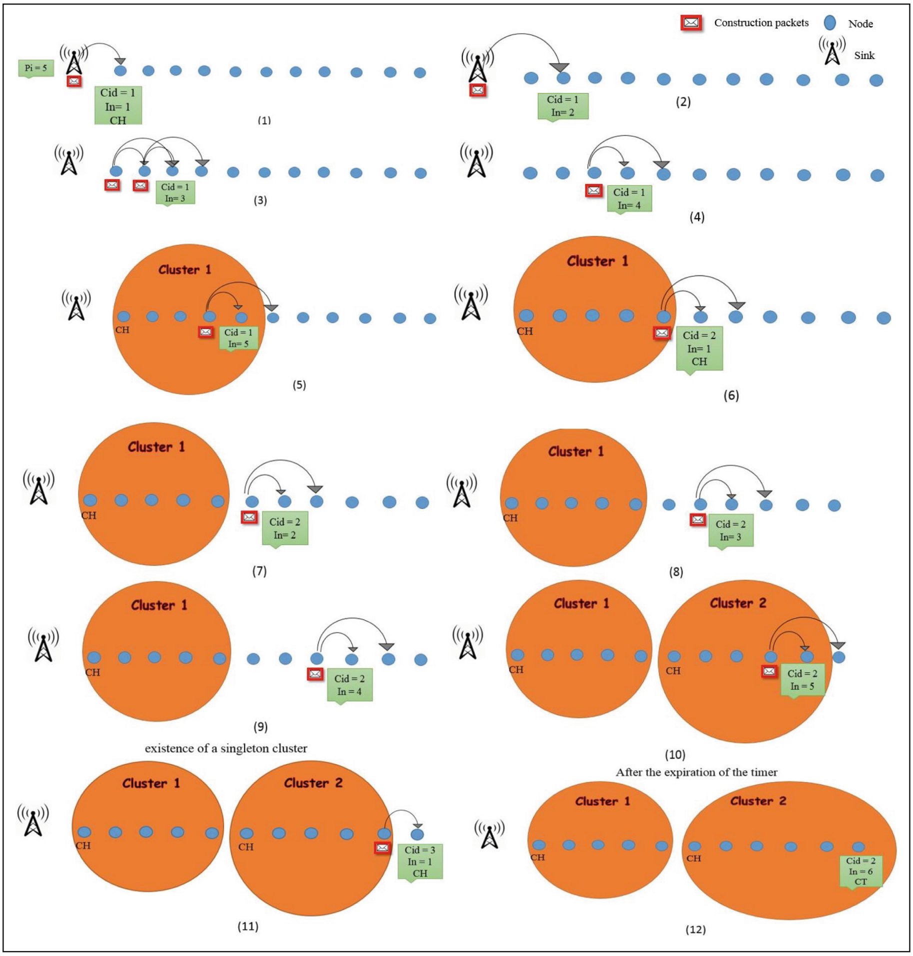

Fig. 6. M2CRL: clusters construction phase

Algorithm 1: Discovery phase

```
Result: First phase: Allow the Sink to discover its neighbors and
        their degree of proximity
Data: neighbor: Sink_neighbors[ ]  int: v
Initialization: v=0
struct {
    MACAddress source
    int v
} neighbor
begin
    if ($isSINK == true$) then
        Pcst=Packet(Discovery)
        Send(Pcst, Broadcast)
        if (Receive_Response_Discovery) then
            neighbor x
            x.source= Address_Packet
            x.v= v_Packet
            Add(x, Sink_neighbors)
    else
        if ($Receive\_Packet\_Discovery$) then
            Pcst=Packet(Hello)
            Send(Pcst, Broadcast)
            start(neighborhood_discovery_Timer)
        if (Receive_Hello) then
            Pcst=Packet(Ack_hello)
            Unicast(Pcst, Sender_Packet)
        while (Receive_Ack_Hello) do
            $v++$
        if (End of neighborhood_discovery_Timer) then
            Pcst=Packet(Response_Discovery)
            Unicast(Pcst, SINK)
```

Algorithm 2: Cluster construction phase

```
Result: This phase makes the creation of the clusters it starts
        immediately after the discovery phase of the Sink
Data: Int:3< max ≤ N/2 (N is the number of nodes in WNS) ack,
      max, Pi, Pn, Cid, neighborsID[2], neighborsCid[2]
Data: $MACAddress : neighbors\_List[]$
Initialization: ack=0
begin
    if ($isSINK == true$) then
        $Pi = random(3, max)$
        $Cid = 1$
        $Pn = 1$
        Pcst=Packet(Pi,Pn,Cid)
        Send(Pcst, A)  // A is the first closest node to SINK:
         with the greatest value of v

        if (Receive Ack) then
            Pcst=Packet(Pi,2,Cid)
            Send(Pcst, B)    // B is the second closest node to
             SINK

    else
        if ($Receive\_Construction\_Packet$) then
            EXECUTE Algorithm 3
        if (Receive_Ack) then
            $ack = 1$
            $Cancel(Ack\_Timer)$
```

Algorithm 3: Behavior of a node following a receipt of a Construction Packet(Pcst)

```
begin
    $Add(MACAddress\_Packet, voisinList)$
    if $(ID\_Receive\_Packet == idSINK)$ then
        $cpt = 0$
        $Pi = Pi(Pcst)$
        $Pn = Pn(Pcst)$
        $Cid = Cid(Pcst)$
        if $(Pi == Pn)$ then
            isCH=true
            Send(Ack, SINK)
        Pcst=Packet(Pi,Pn+1,Cid)
        Send(Pcst, Broadcast)
    else
        if $(cpt == 0)$          // the node already belongs to a cluster
        then
            break
        else if $(cpt == 1)$          // second packet received
        then
            neighborsID[2]=Packet(Pn)
            neighborsCid[2]=Packet(Cid)
            if $(neighborsID[1] == Packet(Pi)) || (neighborsID[2] == Packet(Pi))$ then
                if $neighborsCid[1]==neighborsCid[2]$ then
                    $Pn = 1$
                    $Cid = Packet(Cid) + 1$
                    isCH = true
                    $Pi = Packet(Pi)$
                else
                    $Pn = 2$
                    $Cid = max(neighborsCid[1], neighborsCid[2])$
                    $Pi = Pi(pcst)$
                Pcst=Packet(Pi,Pn + 1,Cid)
                Send(Pcst, Broadcast)
                Send(Ack)
                $cpt = cpt - 1$
                start (Ack_Timer)
            else
                Pn= max(neighborsID[1], neighborsID[2])
                Cid= Packet(Cid)
                Pi=Packet(Pi); Pcst=Packet(Pi,Pn + 1,Cid)
                Send(Pcst, Broadcast)
                Send(Ack)
                $cpt = cpt - 1$
                start (Ack_Timer) 3
        else if $(cpt == 2)$ then
                                          // first packet received
            $cpt = cpt - 1$
            $neighborsID[1] = Packet(Pn)$
            $neighborsCid[1] = Packet(Cid)$
            Send(Ack)
```

```
Algorithm 4: Behavior of a node at the end of Ack Timer
  Result: This process aims to eliminate singleton clusters
begin
    if (End of Ack Timer) then
        if ($ack == 0$) then
            isCT=true
                // Custer Tail(CT) is the most distant node of
              the chain
            if ($Pn == 1$) then
                Pn= Pi+1
                $Cid--$
```

The cluster number of the Linear WSN is given by Nc (2) and the position of a node in the linear WSN is given by Ps (3).

$$N_c = \frac{N}{P_i} \tag{2}$$

$$P_S = P_i * (C_{id} - 1) + P_n \tag{3}$$

N is the cardinality of the Linear WSN.

5 Simulations and Results

To evaluate our mechanism, we performed our simulations on the Castalia environment, which is a wireless sensor network simulator based on the OMNet++ platform. The simulation parameters used are listed in Table 1.

Table 1. M2CRL: simulation parameters

Distance between nodes	Number of nodes in different topologies	MAC protocol	Routing protocol	Radio module	Power Tx	Simulation time
25 m	[5, 10, 15, 20, 25, 30, 35, 40]	TMAC	M2CRL	CC2420	0 dBm	100 s

We performed M2CRL on eight linear topologies ranging from five to forty nodes. The topology used is 2-redundant. Pi is set to 5 in all topologies. The results obtained give a good construction of homogeneous clusters (5 nodes per cluster) with zero singleton cluster. The linear WSN contains zero orphan nodes, each node belongs to a cluster.

We compared M2CRL with LEACH to highlight the inadaptability of existing cluster construction algorithms with a linear topology. The same linear topologies used with M2CRL were used with LEACH. The results obtained confirm the thesis of inadaptability. The construction of the clusters is not complete which implies the existence of several orphan nodes.

Figure 7 gives the number of cluster Head (CH) and thus the number of cluster per topology running the LEACH and M2CRL algorithms. It shows that our M2CRL mechanism creates clusters for the entire network regardless of size. In Fig. 8, LEACH creates many orphan nodes, unlike our M2CRL mechanism, which succeeds in putting all the nodes in the different clusters, without giving orphan nodes, that is to say, nodes that belong to no cluster. Figure 9 gives us the evolution of the construction time of M2CRL on the different topologies used. The results obtained show a decreasing evolution of the construction time of M2CRL clusters on different topologies. Indeed, with the first topology (5 nodes), the construction time is 9.8 s, that is to say, the cardinality of the network, Nc, increased by 4.8 (4). Formula (4) is used to determine the reference time

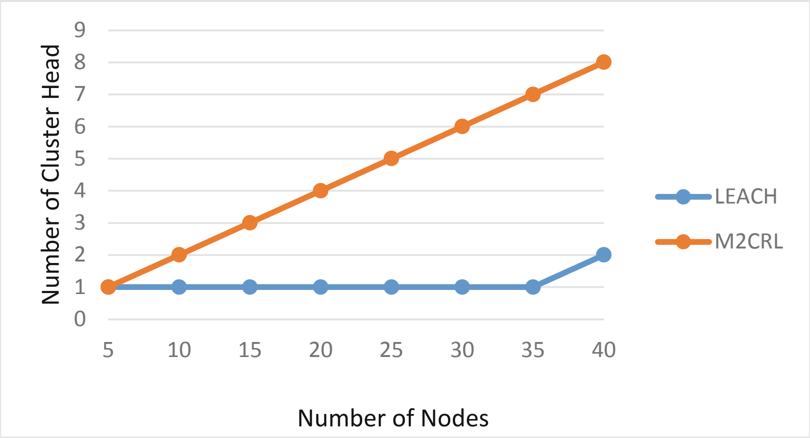

Fig. 7. Number of Cluster Head (CH) per topology

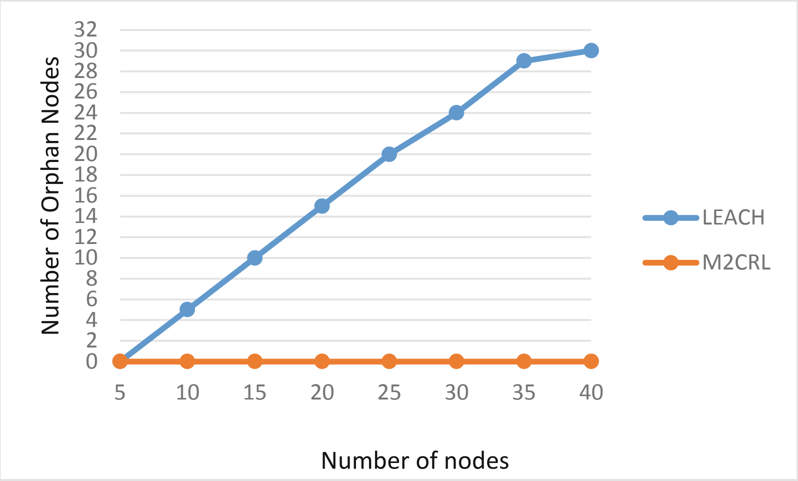

Fig. 8. Number of orphaned nodes per topology

in terms of constructioning clusters on a topology. This reference time will be compared to the different construction times of M2CRL on the different topologies. The graph of Fig. 9 shows that the construction times of M2CRL on the different topologies gradually decrease with respect to the established reference time.

$$Time_{reference} = N_c + 4.8 \tag{4}$$

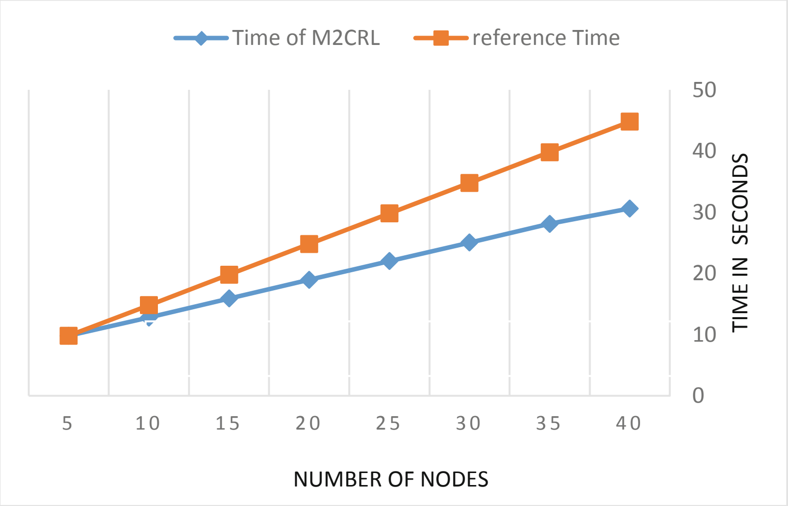

Fig. 9. Evolution of the M2CRL construction time

6 Conclusion and Perspectives

In this paper we have proposed M2CRL, a mechanism of self-construction of clusters for strictly linear networks, which is, to our knowledge, the first mechanism for self-construction of clusters for strictly linear WSNs. Comparative simulations with the previous algorithms have shown perfectly the inadaptability of the latter with linear topologies, and the important need to reflect on mechanisms that are perfectly suited to the particular case of Linear WSNs.

In perspective we plan to improve the algorithm in order to extend its possibilities so that it can take into account linear topologies with junction zones. We also plan to set up an addressing mechanism for these types of junctional linear networks. The algorithm will be improved to foresee the cases of additions of new nodes and deletions of nodes it will be necessary to set up, in the algorithm, systems of automatic reactions in case of topological modification. We also expect a round-robin system of the algorithm to avoid fixed clusters. So in each round of new clusters will be created.

References

1. Kacimi, R., Dhaou, R., Beylot, A.-L.: Using energy-efficient wireless sensor network for cold chain monitoring. In: IEEE Consumer Communications and Networking Conference (CCNC 2009), Las Vegas, Nevada, USA, pp. 1–5. IEEE (January 2009)
2. Sentilles, S.: Architecture logicielle pour capteurs sans-fil en réseau Rapport de recherche, Université de Pau et des Pays de l'Adour, juin 2006
3. Kim, S., et al.: Health monitoring of civil infrastructures using wireless sensor networks. In: Proceedings of the 6th International Conference on Information Processing in Sensor Networks, IPSN 2007, pp. 254–263. ACM Press (2007)
4. Pan, M.-S., Fang, H.-W., Liu, Y.-C., Tseng, Y.-C.: Address assignment and routing schemes for zigbee-based long-thin wireless sensor networks. In: IEEE Vehicular Technology Conference 2008, Spring 2008, pp. 173–177 (2008)
5. Stoianov, I., Nachman, L., Madden, S., Tokmouline, T.: Pipenet: a wireless sensor network for pipeline monitoring. In: ACM IPSN (2007). http://db.csail.mit.edu/pubs/ipsn278-nachman.pdf
6. Abbasi, A.A., Younis, M.: A survey on clustering algorithms for wireless sensor networks. Comput. Commun. **30**, 2826–2841 (2007)
7. Bandyopadhyay, S., Coyle, E.: An energy efficient hierarchical clustering algorithm for wireless sensor networks. In: Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003), San Francisco, California (April 2003)
8. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy efficient communication protocol for wireless microsensor networks. In: 2000 Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, vol. 2, p. 10, 4–7 January (2000)
9. Li, B., Gong, L., Wang, S., Zhou, X.: Multihop routing protocol with unequal clustering for wireless sensor networks. In: International Colloquium on Computing, Communication, Control, and Management (ISECS 2008), pp. 552–556 (2008)
10. Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H.: An application-specific protocol architecture for wireless microsensor networks. IEEE Trans. Wirel. Commun. **1**(4), 660–670 (2002)
11. Ahlawat, A., Malik, V.: An extended vice-cluster selection approach to improve V LEACH protocol in WSN. In: 2013 Third International Conference on Advanced Computing and Communication Technologies (ACCT). IEEE (2013)
12. Li, Y.-Z., Zhang, A.-L., Liang, Y.-Z.: Improvement of leach protocol for wireless sensor networks. In: 2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC). IEEE (2013)
13. Mhatre, V., Rosenberg, C.: Homogeneous vs heterogeneous clustered sensor networks: a comparative study. In: International Conference on Communications, vol. 6. IEEE (2004)
14. Young, H., Wan, Y., Haosong, G., Zeng, H.: A partition based LEACH algorithm. In: IEEE Ninth International Conference on Computer and Information Technology, pp. 40–45 (2009)
15. Mirza, M.A., Garimella, R.M.: PASCAL: power aware sectoring based clustering algorithm for wireless sensor networks. In: The International Conference on Information Networking (ICOIN), Chiang Mai, Thailand, January 2009 (2009)
16. Loscri, V., Morabito, G., Marano, S.: A two-level hierarchy for low-energy adaptive clustering hierarchy. Proc. Veh. Technol. Conf. **03**, 1809–1813 (2005)
17. Fahmy, S., Younis, O.: HEED: a hybrid energy-efficient distributed clustering approach for AdHoc sensor networks. IEEE Trans. Mob. Comput. **3**, 366–379 (2004)
18. Guo, L.-Q., Xie, Y., Yang, C.-H., Jing, Z.-W.: Improve by LEACH by combining adaptive cluster head election and two-hop transmission. In: International Conference on Machine Learning and Cybernetics (ICMLC), vol. 4, pp. 1678–1683 (2010)

19. Gupta, G., Younis, M.: Fault-tolerant clustering of wireless sensor networks. In: Proceedings of the IEEE Wireless Communication and Networks Conference (WCNC 2003), New Orleans, Louisiana, March 2003 (2003)
20. Wang, A., Yang, D., Sun, D.: A clustering algorithm based on energy information and cluster heads expectation for wireless sensor network. Comput. Electr. Eng. **38**, 662–671 (2012)
21. Rostami, A.S., Keshavarz, H., Mohanna, F., Hosseinabadi, A.A.R.: Survey on clustering in heterogeneous and homogeneous wireless sensor networks. J. Supercomput. (2017). https://www.researchgate.net/publication/319155887
22. Wei, D., Chan, H.A., Kaplan, S.: Energy efficient clustering algorithms for wireless sensor network. Conference Paper. IEEE (June 2008). https://doi.org/10.1109/ICCW.2008.50
23. Ndoye, E.H.M., Jacquet, F., Misson, M., Niang I.: Using a token approach for the MAC layer of linear sensor networks: impact of the node position on the packet delivery. In: 2014 IFIP Wireless Days (WD), pp. 1–4 (2014). (cf. p. 38, 70)
24. Jawhar, I., Mohamed, N., Agrawal, D.P.: Linear wireless sensor networks: classification and applications. J. Netw. Comput. Appl. **34**(5), 1671–1682 (2011). (cf. p. 3, 5, 35, 38, 39, 59, 70)
25. Pan, M.-S., Fang, H.-W., Liu, Y.-C., Tseng, Y.-C.: Address assignment and routing schemes for zigbee-based long-thin wireless sensor networks. In: IEEE Vehicular Technology Conference 2008, Spring 2008, pp. 173–177 (2008). (cf. p. 1, 3–5, 35, 40, 56, 66, 67)