

Routing based Roles Assignment for Monitoring 6LowPAN Networks

Abdelkader Lahmadi, Laurent Ciarletta, Alexandre Boeglin, Olivier Festor

▶ **To cite this version:**

Abdelkader Lahmadi, Laurent Ciarletta, Alexandre Boeglin, Olivier Festor. Routing based Roles Assignment for Monitoring 6LowPAN Networks. Third International Conference on Communications and Networking (ComNet), Mar 2012, Hammamet, Tunisia. 10.1109/ComNet.2012.6217745 . hal-00747002

HAL Id: hal-00747002

https://inria.hal.science/hal-00747002

Submitted on 30 Oct 2012

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.

Routing based Roles Assignment for Monitoring 6LowPAN Networks

Abdelkader Lahmadi, Laurent Ciarletta
Lorraine University - LORIA
Villers-Lès-Nancy, France
Email: {Abdelkader.Lahmadi, Laurent.Ciarletta}@loria.fr

Alexander Boeglin, Olivier Festor
INRIA Nancy - Grand Est Research Center
Villers-Lès-Nancy, France
Email: {Alexandre.Boeglin,Olivier.Festor}@inria.fr

***Abstract*—In this work, we present a novel approach to assign monitoring roles in constrained, low power and lossy networks using available local information provided by the routing layer. The resulting monitoring architecture is adaptive taking benefit from the reactivity of the routing protocol when dynamic changes occur due to connectivity or nodes movement. The simulation results reveal that our assignment approach is more efficient, less aggressive and less resources consuming than its competitors.**

***Index Terms*—6LowPAN, Monitoring, Piggybacking, Poller-Pollee, Placement**

I. Introduction

Low Power and Lossy Networks denote many embedded devices interconnected by a variety of links ranging from wireless technologies such as 802.15.4, bluetooth, Low Power Wifi to wired technologies such as low power PLC. The common property of such networks is the limited resources of their nodes in terms of power, computing, memory and communication. The network could be described with a thousands number of devices with very limited internal and external resources and their communication channels are low-bandwidth, high loss rate and volatile links subject to failure over time. A well known example of such networks is the wireless sensor networks which are made by sensing and actuation nodes interconnected by wireless links. These networks are associated with different applications for industrial automation, environmental monitoring, homeland security, weather and climate analysis and prediction, etc. In operational networks, these devices need to be constantly or temporally monitored to assure their functioning and detect relevant problems which will result in an alarm being forwarded to the enterprise network for analysis and remediation. However, LLN monitoring is an auxiliary activity to the primary function which is sensor data acquisition. Information delivering in an LLN network is costly and it has to be preserved for this primary function. Hence, monitoring of an LLN network to estimate its state is challenging and its overhead should be as small as possible.

In IP networks, monitoring architecture are often organized over the Poller-Pollee structure where a poller is a specific node which monitor other nodes called pollees [1]. In such architecture, two communication models are possible between the pollers and the pollees. In proactive monitoring, messages are sent periodically by the poller to a target pollee to retrieve monitoring data. In reactive monitoring, the pollee informs its poller about the values of monitoring variables. It monitors locally its variables and only sends a message to the poller when a value of a variable crosses a predefined threshold.

Monitoring low power and lossy networks is challenging. Firstly, the main concern in those networks is energy where networking and processing activities should be reduced to decrease nodes battery consumption. Secondly, LLN are dynamic and the connectivity is limited and fluctuating in time. A node may loss frequently its connectivity with its neighborhood nodes. In addition, link layer frames are constrained in size and in throughput. For example, the 802.15.4 protocol only leaves 102 bytes to upper layers data. Thus a monitoring solution need to be lightweight to reduce its resource consumption, it should be robust and fault tolerant to combat wireless channels effect, it should be distributed where it does not rely on specific nodes, and it should be adaptive that dynamically change operation to support channels errors and nodes movements.

Recently, a new framework for monitoring wireless sensor networks was described in [2]. Their approach is based on theoretical algorithms from graph theory to define optimal aggregation of monitoring reports and the placement of monitoring roles using a poller-pollee structure. The major drawback of such approach is its communication overhead in terms of maintenance and setup of the monitoring architecture.

In this work, we propose a novel approach to place a monitoring architecture for LLN networks aiming to minimize the monitoring communication cost and overhead on deployed nodes while being robust and adaptive. Our approach is mainly based on the routing graph build by the RPL protocol to place a two-tier monitoring architecture. The rest of the paper is organized as follows : Section II describes existing work about the placement of monitoring roles In IP and wireless sensor networks. In section III, we give an overview of the poller-pollee monitoring structure and its applicability in the context of a wireless sensor network. We also present the RPL routing protocol messages and operations. In section IV, we detail our algorithms to build a poller-pollee structure using information provided by the RPL protocol. Section V provides simulation results of the evaluation of our approach under different scenarios. In section VI, we draw some conclusions

978-1-4673-1008-6/12/$31.00 © 2012 IEEE

and identify future works.

II. Related work

The placement problem of monitors in IP-based networks has been addressed by several works [3], [1] where their goal was to minimize the number of monitor to reduce the overhead while maintaining a suitable monitoring quality to obtain a suitable network properties view. These works were conducted over fixed networks topologies which are different from LLN network where topologies are dynamic and links are unstable. In the case of LLN networks and wireless sensor networks in particular, several monitoring approaches have been proposed. In [4], authors present a distributed approach based on neighborhood monitoring. Where each node monitors its neighbors using two-phase timers to decrease the probability of false alarm. As they stated in their paper, their approach only works when the network is not partitioned and communication path exists to the control center. Our work is complementary to their since we propose an algorithm to place a small subset of pollers on the critical nodes to keep the routing graph connected. Thus nodes may apply a neighborhood monitoring using their approach and when an event occurs, they send the alarm to the nearest poller. Recently, Liu and al [2] propose an approach based on a distributed Maximal Independent Set-based algorithm. Their algorithm introduces two parameters k1 and k2 to control the geometrical distribution of the pollers and pollees. The parameter k1 denote the minimum number of hops between two pollers and k2 denotes the maximum number of hops between a poller and pollee. Their algorithm builds the monitoring architecture by exchanging messages between nodes according to the values of k1 and k2. However, our intuition is that in case of a low power and lossy network, using their algorithm raises an important overhead where building and maintaining the monitoring architecture requires that nodes exchange specific packets which will compete with application data. In our work, we propose an alternative approach where pollers are assigned to critical nodes in the routing graph to guarantee a path between pollees and their respective pollers to deliver network monitoring data.

III. Background and motivations

In this section, we give an overview of different notions used in the work to formulate and propose a solution for the assignment of monitoring roles in a LLN network.

A. Distributed Poller-Pollee monitoring model

In this work, we focus on a two tier monitoring model, where in lower tier we place pollees and in the upper tier we place pollers. The pollees report their monitoring variables values to their poller node along multi-hop paths [2]. Intermediate nodes may aggregate these reports to reduce overhead. Each poller node makes local decision based on the received monitoring messages and forward the decision to a centralized monitoring station. Figure 1 depicts an example of a poller-pollee deployment over a wireless sensor network. Each pollee

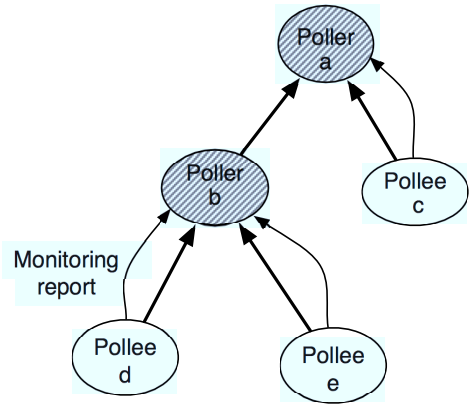


Fig. 1. An example of poller-pollee structure.

maintains a set of monitored variables with different monitoring periods. When a period expires or a monitoring threshold is crossed, a report is sent to the poller. If a link failure occurs or the connectivity with the destination node to the poller becomes bad then the report is lost and the poller will trigger a false alarm. A false alarm is triggered when the poller does not receive a pollee report within a specific time window. In the figure solid and dashed links denotes a routing path between nodes established using a running routing protocol. Designing a monitoring architecture following the poller-pollee model over an LLN network is challenging. A promising approach is to use dominating-sets to organize nodes into a monitoring hierarchy [2], [1].

We model a LLN network by a graph G(V,E^t) compromised of a set of vertices V and a time-varying edges E^t. For each pair of vertices $u, v \in V, (u, v) \in E^t$ if and only the nodes u and v are within communication range. Due to the lossy property, the topology of the network is dynamic and reflected by E^t.

Each node has an associated set of properties, including the following: the identifier of the node, its $rank^t$ at time t, its set of candidate parents, its preferred parent, its closed children set and its open children set.

From the graph theory, we used some definitions in this work.

Definition An independent set is a subset of V such that no two vertices within the set are adjacent in V.

For example, {d,e,c} is an independent set in Figure 1.

Definition A Maximal Independent Set (MIS) is an independent set such that adding any vertex not in the set breaks the independence property of the set. Thus, any vertex outside of the maximal independent set must be adjacent to some node in the set.

The independent set {d,e,c} is also a MIS.

Definition A Dominating set S is defined as subset of V such that no node in V - S is adjacent to at least one node in S.

Thus a node u dominates another node v if and only if u

= v or u and v are adjacent. A Connected Dominating Set is a dominating set of G which induces a subgraph of G. Thus for every vertex $v \in V - S$ there at least one vertex $u \in S$ that dominates v, and S is connected. One approach to construct a CDS is to find an MIS, and then add additional vertices as needed to connect the nodes in MIS. A CDS in Figure 1 is {a, b}. A minimal connected dominating set MCDS is a dominating set such that every proper subset is non-dominating. In Figure 1, {a, b} is also an MCDS.

Given, these definitions, we formulate the monitoring placement problem with the objective to find a CDS in an LLN network, such that every node in the network is either a poller (a dominating node) in the set or a neighbor of one or many nodes in the set. The main advantage of CDS is that it simplifies the monitoring process to the one in a smaller network. Only dominating vertices which are the pollers need to keep monitoring information to make a decision about the monitored set.

The efficiency of a CDS approach to build a monitoring hierarchy depends largely on the process of finding and maintaining the CDS and its size which is the number of pollers. However, finding an MCDS is NP-Hard problem. Many algorithms and techniques have been proposed in literature to construct a CDS [5] for different purposes : routing, clustering, network management. We believe that in a LLN network, CDS formation algorithm to build a monitoring architecture should ensure two properties: (i) it has to be localized which means it relies only on local information to minimize the monitoring cost in the network; (ii) it provides a MCDS to minimize the number of pollers within the network to minimize the overhead on selected nodes as pollers. In addition, we need to take care on the monitoring quality because, as stated in [2], in a lossy environment where links failure is frequent, the false alarm rates may also increase when the number of pollers is too small and pollees are so far from their pollers.

In our approach, we propose to only use information available from the routing protocol running over LLN nodes to construct a monitoring overlay following the poller-pollee structure. We focus in this work on the RPL protocol as an underlying routing protocol. The rational behind is to meet the first requirement that we have fixed which is to minimize the in network monitoring overhead.

B. RPL routing protocol overview

The Routing Protocol for Low Power and Lossy Networks (RPL) [6] is proposed by the ROLL working group of the IETF to provide an IPv6 routing protocol to fit the various requirements introduced by the working group in RFCs 5548, 5673, 5826 and 5867. In RPL when a node obtains an IPv6 address, it tries to join a Destination Oriented Directed Acyclic Graph (DODAG) by exchanging ICMPv6 control messages. The root of a RPL DODAG advertises itself as a parent node for other its neighborhood nodes using the DIO (DODAG Information Object) message. When a node receives a DIO from different nodes it computes its own rank value which represents its relative position within the DODAG from the root. It selects a parent which minimize the computed rank and it is attached to the DODAG through this parent. Once it selects a parent, a RPL node propagates its own DIO messages down the network to form its sub-DODAG. Each node stores its parent address as the next-hop to deliver data to the root of the DODAG. This state is only maintained using DIO message exchanging. On the other hand, RPL nodes use the DAO (Destination Advertisement Object) to provision routing tables with downward routes. The DAO message advertises routes for various destinations and prefixes. Each node locally store the route and forward this route information to the root of the DODAG. By collecting such information, a packet can travel in the tree until it reaches a node that has knowledge of the routing path to the destination. RPL was mainly designed to reduce the cost of routing state maintenance by using trickle based timers [7]. It also use expressive link metrics carried in DIO messages to deal with various types of applications requirements in challenging radio environments. The main assumption behind RPL is that multi-point to point traffic is dominating. Therefore, the protocol is optimized and relies on the notion of best parent which is selected by a node to transmit its data towards a sink. In this work, we take benefit from the virtual routing backbone build by RPL to place monitoring nodes following a poller-pollee structure. Our intuition is that network monitoring process has the same goal as a user-defined monitoring application where a set of pollees are aiming to deliver their status reports toward their respective pollers to provide a view of network properties (batteries, connectivity, density,...).

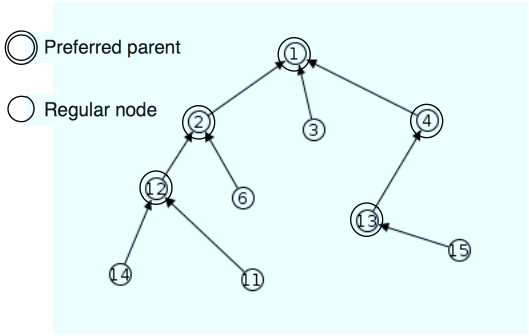


Fig. 2. RPL DODAG architecture.

IV. RPL-BASED MONITORING ROLES ASSIGNMENT

The RPL protocol mainly constructs a virtual routing backbone which is the DODOAG upon a set of nodes of a deployed wireless sensor network. The constructed DODOAG forms a CDS where each node is either in the CDS and acts as a router or it is a neighbor of a node in the CDS. In addition, over time RPL constructs k-CDS [8] since each node has k candidate parents and it selects a single parent among them according to a set of metrics. Figure 2 depicts an example of a DODAG built using the RPL protocol running over a wireless sensor network.

At a time t the current routing CDS is minimum since

a node has to select only one dominating parent (the best according a specific metric) among existing candidate parents available towards the root of the DODAG (Figure 2). We denote this set of parents as the set of candidate pollers available to a pollee. When a selected poller fails or its connectivity becomes unstable, the RPL routing protocol will react and selects a new routing parent which will be also the new poller. Our aim is develop a greedy algorithm executed on each node exploiting information provided by the routing protocol RPL, mainly the set of child and parents, also it piggybacks a minor specific information in RPL exchanged messages to build a monitoring architecture over an LLN network following the poller-pollee structure.

As described in III-B, the RPL protocol uses the DAO and DIO messages to build a DODAG over a set of nodes. Locally, each node maintains two sets of neighbors: the set of parents and the set of children. The set of children is composed of two subset. The first subset is the closed children which contains the direct attached children of a node. The second is the open children contains all the subsequent children of a node. For example in Figure 1, the node a has the nodes $\{$b,c$\}$ as a closed children set and $\{$b,c,d,e$\}$ as the open children set. In addition, each node maintains the number of direct edges emanating to it, which represents the size of its closed children. For example, the node a has a degree of 2. We propose a Greedy algorithm executed by each node to identify its role using the information provided by the routing layer, mainly the lists of open and closed children. These lists are computed by the routing layer each time a node receives a DAO message indicating that the receiver node is selected by another node as a parent towards the sink. Unlike other placement algorithms [2], there is no specific messages exchanged by nodes to make a role decision. Only the information provided by the routing layer RPL is used. As stated above, the RPL protocol build and maintain a DODAG and we take benefit from this DODAG to place monitoring roles up network nodes. The RPL DODAG have interesting properties to build the monitoring architecture. Firstly, the DODAG is self-organizing an auto-adaptive since when connectivity becomes unstable and links failure, nodes adapt their routes to select best parents toward the sink. Secondly, the selection process relies on well defined metrics (good connectivity, minimum number of hop-count) to guaranty the selection of a parent capable to deliver traffic towards the sink. In the following, we investigated different strategies to place the poller-pollee structure over a constructed DODAG. For each strategy we present its benefit and its drawbacks.

1) Trivial algorithm: All-Parents-Pollers: A trivial strategy for monitoring roles assignment is that each parent within the routing DODAG will select itself as a poller. A node can identify itself as a parent when it receives a DAO message from at least one node. The node becomes a pollee when it has not anymore children. Thus, only leaf nodes will be selected as pollees. The major drawback of this algorithm is that it places an important number of pollers over the network which introduces an important processing overhead on the nodes. In addition, some parents will be selected as pollers for only few pollees nodes. Figure 3 depicts a running example of this algorithm.

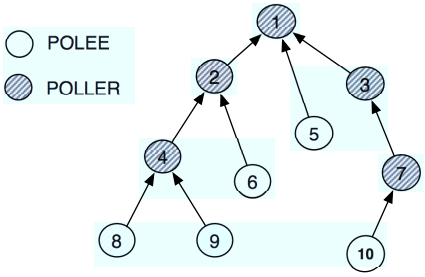


Fig. 3. A Poller-Pollee deployment using a trivial algorithm where all parents are POLLERs.

2) Critical parents strategy: A more interesting approach is to identify the critical nodes within the DODAG constructed by the routing process. Those nodes are critical because when they are removed, the DODAG will be disconnected. In graph theory a DODAG is a set of bi-connected directed graph where articulation nodes connect them [9]. In a routing DODAG, we define a critical parent v of the DODAG as a parent node that has at least a child who has no back parent other than v to reach the data sink. According to this definition, we can state that the root node is a poller. To identify critical parents, each node has to inform his selected parent about the number of its available parents. Let n_i denotes the number of candidate parents of a node i to reach the sink. Thus if a parent has a child with n_i equal or greater than 1, then it will select itself as a poller. In practical, we have modified the DAO message exchanged between an node and its selected parent to carry this number of candidate parents. The algorithm 1 presents the role assignment algorithm using the strategy of critical parents.

Algorithm 1 Critical-parent based monitoring roles placement algorithm.

Input: : N_i is the list of closed children of the current parent node
Input: : $\{n_j\}$ is the respective list of numbers of candidate parents of each child j of the current parent node i.
Function setRole ()
  Degree = size (N_i)
  if Degree == 0 **then**
    Role = POLLEE
  else if min($\{n_j\}$) == 1 **then**
    Role = POLLER
  end if

Applying the above algorithm on the example presented in Figure 3, we obtain monitoring roles assignment as depicted in Figure 4.

The described strategy introduces a small overhead into each DAO message where the node needs to inform his parent about its number of candidate parents.

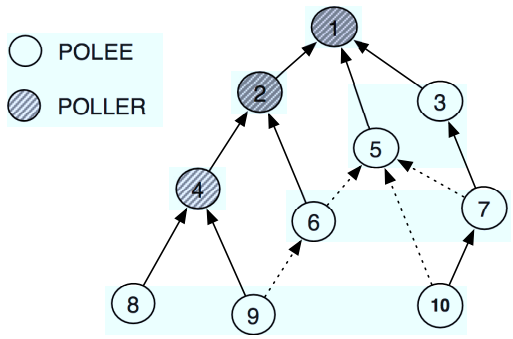


Fig. 4. A Poller-Polle deployment using the critical-parent algorithm. Solid lines represent link to the selected parent and dashed lines represent back links to candidate parents.

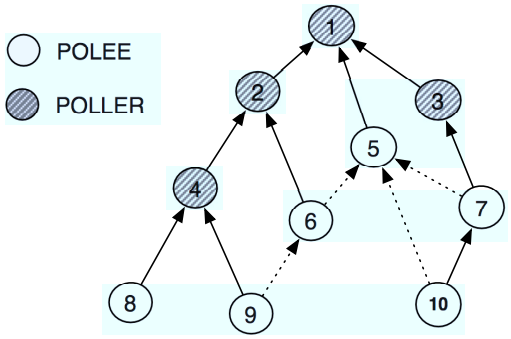


Fig. 5. A poller-polle deployment using the critical link strategy. Solid lines represent link to the selected parent and dashed lines represent back links to candidate parents.

3) Critical-edge placement strategy: A major drawback of the strategy based on the critical parents is that it may introduce an important monitoring traffic load on the single link between the critical parent and the child where it is the single parent since all the traffic will go through this link. To overcome such drawback, we propose another assignment strategy where a child with a single parent candidate will take the role of a poller. In such strategy, the child alleviates the monitoring load on the single link between him and his only parent toward the sink. This strategy needs only a local decision using the set of available parents for a node obtained from the received DIO messages. The algorithm 2 depicts the monitoring roles placement using this strategy. Figure 5

Algorithm 2 Roles placement strategy using critical links.

Input: CH is the list of closed children provided by the routing layer.
Input: CP is the list of candidate parents.
Function setRole ()
  Degree = size (CH)
  if Degree == 0 **then**
    Role = POLLEE
  else if size(CP) == 1 **then**
    Role = POLLER
  else
    Role = POLLEE
  end if

depicts the resulting roles assignment using the critical link strategy. We observe that with this strategy the node number 3 becomes a poller compared to the critical parent strategy where it was a pollee. Thus in this situation the distance between the pollee number 10 and its nearest poller is reduced to 1 hop.

4) k-distance POLLER-POLLEE rule: A major limitation of the the two strategies described above is that they may place multiple pollees between an edge pollee (routing tree leaves) and a poller. These strategies only ensure that a poller will be available on each articulation node of the constructed routing DODAG, but with no control over the maximum distance between a poller and pollee, .

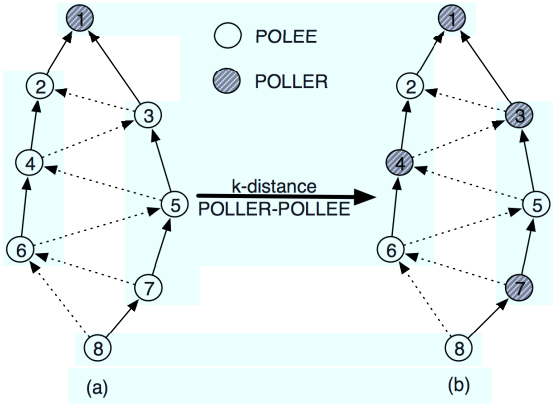


Fig. 6. Illustration of a Poller-Pollee deployment using the proposed algorithm 1. In (a), the distance between the poller 1 and the pollee 8 is equal to 3. (b) represents the resulting placement after applying the k-distance POLLER-POLLEE rule, where k is equal to 1.

Figure 6(a) depicts an example of such situation where we observe that the distance between the poller with the identifier 1 and the edge pollee 8 is equal to 3. In such situation a false alarm may raise where a monitoring data may be lost or delayed in the path between the poller and an edge pollee. A false alarm denotes that the poller does not receive a monitoring data from a node within a defined time window [2]. To overcome such situation we propose an expansion rule of the number of pollers along a routing branch to minimize the probability of a false alarm by enforcing a maximum tolerated distance between a polller and a pollee. The idea is to propagate a parameter k controlling the distance between pollers and pollees along the routing path. In practical, we piggybacked the parameter k in the DAO message between a node and its routing parent. Each pollee node along the routing path decreases the value of k. When a poller node receives the DAO it sets the value of k to its initial, increases it before sending its DAO. When k is equal to 0 and the node has a role of pollee, it modifies its role to poller and it re-initializes and increases the value of h before sending a DAO message. Algorithm 3 depicts the k-distance POLLER-POLLEE strategy.

Algorithm 3 k-distance POLLER-POLLEE strategy.

Input: Role is the role of the current node
Input: k is the maximum distance between a POLLER and a POLLEE
Function modifyRole ()
  if Role == POLLEE **then**
    piggybackDAO(k)
  end if
  if DAO is received **then**
    if Role == POLLEE **then**
      c = c -1
      if c == 0 **then**
        Role = POLLER
      end if
    end if
    if Role == POLLER **then**
      c=k
      c = c+1
    end if
    piggybackDAO(c)
  end if

V. Performance evaluation

Next, we present our methodology and results of the evaluation of the proposed algorithms of building a monitoring overlay based on the poller-pollee structure over a routing graph build using the RPL protocol.

A. Methodology

We conducted a set of simulations using the Cooja simulation tool provided by the Contiki project to evaluate the proposed placement algorithms and mechanisms. In our evaluation, we enabled on each simulated node the implementation of the protocols RPL and 6LoWPAN provided by Contiki. We have evaluated two types of networks: sparse and dense with a distance loss model. The sparse and the dense properties denote the density of the neighborhood of a node. We controlled the density using the transmission range value of a node. Each simulated network contains 25 nodes. The rank of each node in the RPL DODAG is based on the ETX metric [10]. The table I summarizes the different parameters of the simulated scenarios.

Scenario name	TX ratio	RX ratio	Transmission range	Inference range
Dense-Loss	100 %	0 %	100 m	120 m
Medium-Loss	100 %	0 %	70 m	90 m
Sparse-Loss	100 %	0 %	50 m	60 m

TABLE I
Evaluation parameters of simulated scenarios.

We assessed the performance metrics related to the overhead and the quality of our proposed algorithms for role placement. We mainly measured distribution of the distances between the pollers and their respective pollees over a particular deployed topology of wireless sensor network. This metric is composed of two sub metrics. The first metrics denotes the number of pollers in the network. The second sub-metric denotes the distribution of distances between pollers and pollees.

B. Comparison of the different assignment strategies

The communication overhead of the our proposed placement algorithms is minor, since it relies on messages used by the RPL protocol. Thus its cost is equal of the setup and maintenance cost of the RPL protocol. For the algorithm of critical parents strategy, as described in section IV, we piggybacked in each DAO message the number of candidate parents of a node which consumes 1 byte in each transmitted DAO message. The critical-link algorithm has no communication overhead since it only relies on the list of candidate parents available locally and provided by the routing process. The k-distance POLLER-POLLEE strategy consumes also 1 byte in each DAO message to encode the k distance number.

1) Critical-link strategy: Figure 7 shows the simulation results of the medium and sparse scenarios using the critical-link strategy to place the pollers and the pollees over a constructed DODAG by the RPL protocol. We observe that when the network becomes more sparse, the distance between pollers and pollees increases in consequence. For example, in a medium density network with packets loss, the number of pollers is 11, the average distance between a poller and a pollee is equal to 1.28 and 78% of pollers are within 1 hop from their nearest poller in the routing path. However, with a sparse topology the average distance is equal to 1.75 and only 50% of nodes are with a distance of 1 to their nearest poller.

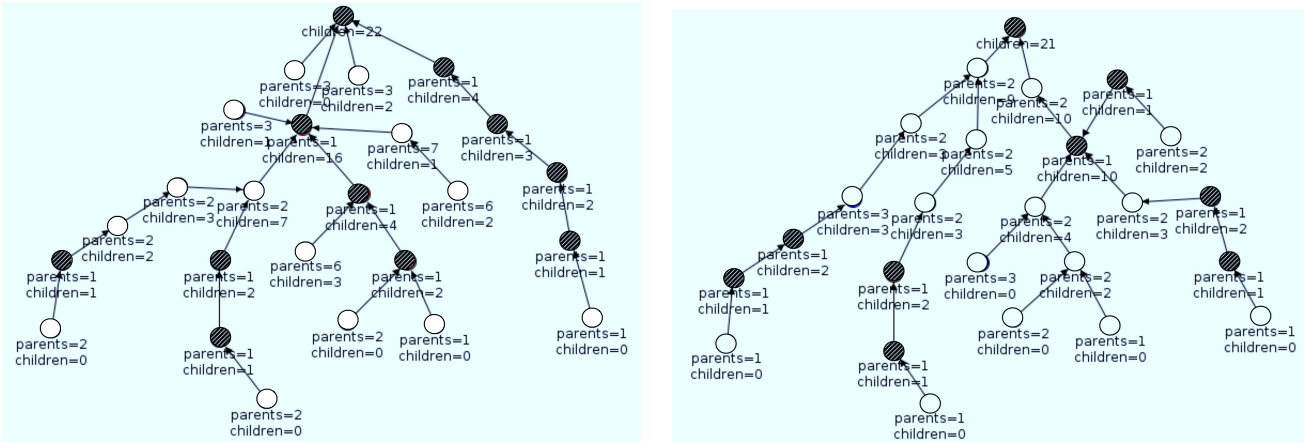
The table II summarizes the distribution of the pollers and pollees over the deployed network using the link-articulation strategy.

Scenario	Number of pollers	Average distance Poller-Pollee	Poller-Pollee distance distribution
Medium-Loss	11	1.28	78%: 1-hop 14%: 2-hop 8%: 3-hop
Sparse-Loss	9	1.75	50%: 1-hop 25%: 2-hop 25%: 3-hop

TABLE II
The poller-pollee distribution statistics when using an articulation-link strategy.

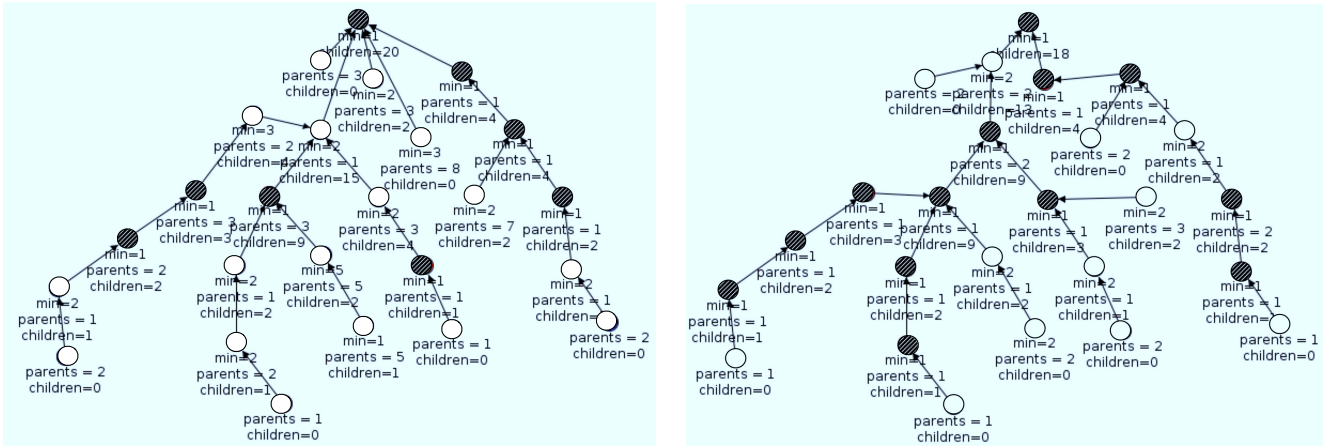
2) Critical-parents strategy: In a second step, we evaluated the critical-parents strategy without the k-distance POLLER-POLLEE rule, using the scenarios defined in Table I. We are interested in the distribution of the distance between a poller and pollee. Figure 8 shows a snapshot of the poller-pollee distribution in the dense scenario after running the articulation-parents strategy. We observe that a single poller is selected which is the data sink. This is due to the density of the network where each node has many candidate parents since the network is fully connected.

When the network becomes more sparse, we observed that the the strategy starts placing pollers over the DODAG build by the RPL routing protocol.



(a) Medium-Loss (b) Sparse-Loss

Fig. 7. A snapshot of the poller-pollee distribution over medium (a) and sparse (b) networks after running the critical-link strategy.



(a) Medium-Loss (b) Sparse-Loss

Fig. 9. A snapshot of the poller-pollee distribution over medium (a) and sparse (b) networks using the critical-parent strategy.

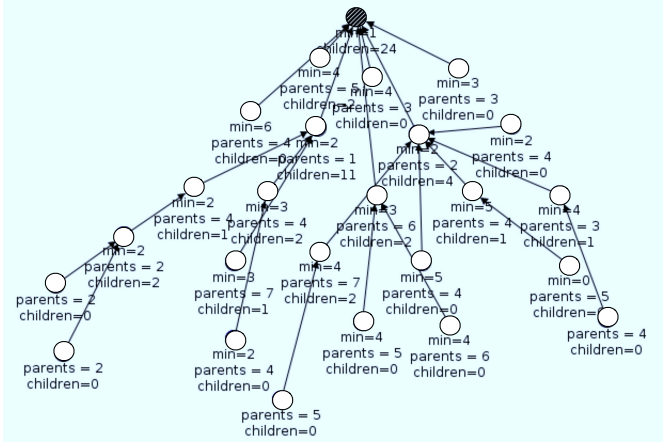


Fig. 8. A snapshot of the poller-pollee distribution after running the critical-parent strategy on a dense network.

Figure 9 shows the snapshots of poller-pollee distribution with the medium and sparse scenarios using the critical-parents strategy to place the pollers and the pollees over a constructed DODAG by the RPL protocol. For example, in a sparse density network with packets loss, the number of pollers is 13, the average distance between a poller and a pollee is equal to 1.25 and 75% of pollers are within 1 hop from their nearest poller in the routing path. The table III summarizes the statistics of the poller-pollee distribution using this strategy. In a second step, we enabled the $k - distance$ rule with k equal to 1, for the dense network. We observe as depicted in Figure 8 that only the data sink node was selected as poller using the parent-

Scenario	Number of pollers	Average distance Poller-Pollee	Poller-Pollee distance distribution
Medium-Loss	8	1.47	59%: 1-hop 35%: 2-hop 6%: 3-hop
Sparse-Loss	13	1.25	75%: 1-hop 25%: 2-hop

TABLE III
THE POLLER-POLLEE DISTRIBUTION STATISTICS WHEN USING AN ARTICULATION-PARENT STRATEGY.

articulation placement strategy. However, as shown in Figure 10, after applying the k-distance rule with $k = 1$, we observe that several nodes has been selected to be poller to ensure a distance equal to 1. In this scenario the number of pollers is equal to 8 with an average distance between a poller and pollee equal to 1.05. Only one pollee has a distance equal to 2 to the nearest poller. This distance is different from 1 due to the lost of the DAO message. In a last step, we compared our

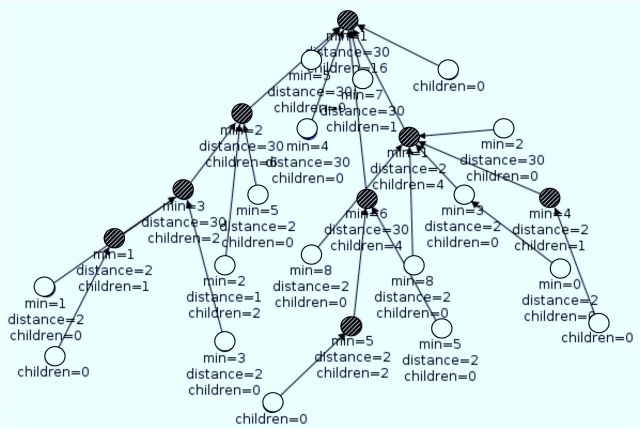


Fig. 10. A snapshot of the poller-pollee distribution after running the parent-articulation strategy and the k-distance rule on a dense network. $k = 1$.

critical-parents strategy with the hybrid algorithm proposed in [2]. We have implemented their proposed hybrid algorithm in Cooja and we selected the sparse network scenario to compare them. When implementing their hybrid algorithm, we have identified that it is theoretical and in practical it needs a mechanism to announce nodes ID over the neighborhood of each node according to the value of distance between a poller and a pollee. We have thus defined a flooding message to allow each node to announce its ID. When running the algorithm, we firstly observed that it introduces an important overhead in terms of the number of exchanged messages to maintain a setup the poller-pollee structure. As depicted also in Figure 11, we observe that the constructed structure ensure a maximum distance of 1 between a poller and pollee but within the connectivity Unit Disk Graph of each node. We observed also that the placement of the pollers does not match the routing overlay where some pollers are located in bottom the DODAG.

VI. Conclusion and future work

In this work, we have addressed the assignment of monitoring roles in a constrained low-power and lossy-network to monitor its nodes status (energy, connectivity, density,...). We have proposed a greedy algorithm to assign roles following a poller-pollee structure over a routing graph constructed using a running routing protocol. We have focused on the RPL protocol developed by the IETF. The main advantage of the RPL protocol is that it builds an adaptive and a robust routing overlay that we used to place a poller near the link path to a pollee based on the notion of critical links and parents. We

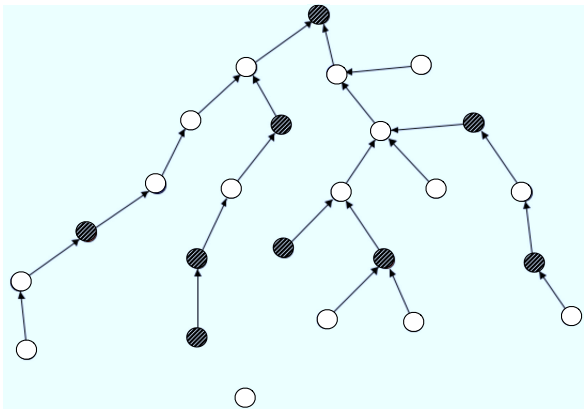


Fig. 11. A snapshot of the poller-pollee distribution after running the hybrid algorithm [2] where $k1 = k2 = 1$.

have simulated our proposed placement algorithm where we have shown that it is capable of placing the pollers nearest the pollee nodes with a critical link path. The placement algorithm maintains an average distance between each pollee and poller close to 1 while minimizing the exchanged traffic between nodes to place monitoring roles. In future work, we plan to extensively simulate our algorithms to better identify their overhead and scalability under different topologies and network densities.

References

[1] E. L. Li, M. Thottan, B. Yao, and S. Paul, "Distributed network monitoring with bounded link utilization in ip networks," in *INFOCOM 2003. he 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, San Franciso, CA, USA, March 30 - April 3, 2003*. IEEE, 2003.

[2] C. Liu and G. Cao, "Distributed monitoring and aggregation in wireless sensor networks," in *Proceedings of the 29th conference on Information communications*, ser. INFOCOM'10. Piscataway, NJ, USA: IEEE Press, 2010, pp. 2097–2105. [Online]. Available: http://portal.acm.org/citation.cfm?id=1833515.1833795

[3] K. Suh, Y. Guo, J. Kurose, and D. Towsley, "Locating network monitors: complexity, heuristics, and coverage," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 1. IEEE, 2005, pp. 351–361.

[4] C. Hsin and M. Liu, "A distributed monitoring mechanism for wireless sensor networks," in *Proceedings of the 1st ACM workshop on Wireless security*. ACM, 2002, pp. 57–66.

[5] J. Blum, M. Ding, A. Thaeler, and X. Cheng, "Connected dominating set in sensor networks and manets," *Handbook of Combinatorial Optimization, Springer*, pp. 329–369, 2005.

[6] T. Winter, P. Thubert, A. Brandt, T. Clausen, J. Hui, P. Levis, K. Pister, R. Struik, and J. Vasseur, "Rpl: Ipv6 routing protocol for low power and lossy networks," draf-ietf-rpl-19, 2011.

[7] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation-Volume 1*. USENIX Association, 2004, pp. 2–2.

[8] F. Dai and J. Wu, "On constructing k-connected k-dominating set in wireless networks," in *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*. IEEE, 2005, pp. 81a–81a.

[9] A. Gibbons, *Algorithmic graph theory*. Cambridge Univ Pr, 1985.

[10] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, ser. MobiCom '03. New York, NY, USA: ACM, 2003, pp. 134–146.