HAL
open science

Routing for Data-Collection in Heterogeneous Wireless Sensor Networks

Bilel Romdhani, Dominique Barthel, Fabrice Valois

To cite this version:

Bilel Romdhani, Dominique Barthel, Fabrice Valois. Routing for Data-Collection in Heterogeneous Wireless Sensor Networks. [Research Report] RR-7586, 2011. inria-00581603v1

HAL Id: inria-00581603

https://inria.hal.science/inria-00581603v1

Submitted on 4 Apr 2011 (v1), last revised 30 Nov 2011 (v3)

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.

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Routing for Data-Collection in Heterogeneous Wireless Sensor Networks

Bilel Romdhani — Dominique Barthel — Fabrice Valois

N° 7586

March 2011

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7586--FR+ENG

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche GRENOBLE - RHÔNE-ALPES

Routing for Data-Collection in Heterogeneous Wireless Sensor Networks

Bilel Romdhani*†‡ , Dominique Barthel‡ , Fabrice Valois† §

Thème : 1
Équipes-Projets SWING

Rapport de recherche n° 7586 — March 2011 — 18 pages

Abstract: In this report, we focus on Wireless Sensor Networks (WSNs) in a more realistic setting than classical studies and previous works: we consider wireless sensor nodes having different transmission ranges according to the environment and/or to the wireless chipset. The main consequence of this heterogeneity is the existence of asymmetric links. Such links in a WSN degrade the performance of most protocols which have not been designed to benefit from this heterogeneity and to deal with asymmetric links: most often, these links are essentially pruned. We propose a routing protocol for data collection from sensors nodes to the sink node in heterogeneous WSNs. Our proposal detects and takes benefit from asymmetric links caused by this heterogeneity. Our proposal, denoted MURA (MUltiple RAnges), (1) provides a high delivery ratio, (2) reduces the number of duplicated packets and (3) reduces the number of hop counts by exploiting the asymmetric links.

Key-words: Heterogeneous Networks, Wireless Sensor Networks, Asymmetric Links, Data-Collection.

§ This work was partially funded by the ANR VERSO ARESA2 project

* Corresponding author
† University of Lyon, INRIA, INSA Lyon, CITI, F-69621, France
‡ Orange Labs Meylan France

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Routage convergcast pour les réseaux de capteurs hétérogénes

Résumé : Dans ce rapport, nous nous concentrons sur les réseaux de capteurs sans fil (WSNs) dans un cadre plus réaliste que les études classiques et antérieures : nous considérons des nœuds capteurs sans fil avec des portées de communication différentes. Cette différence au niveau des portées de communication peut être causée par l'environement où le capteur est deployé ou en relation avec le matériel dont dispose chaque capteur. La principale conséquence de cette hétérogénéité est l'existence des liens asymétriques.
L'apparition des liens asymétriques dans un WSN dégrade les performances de la plupart des protocoles qui n'ont pas été con cus pour supporter cette hétérogénéité. Dans ce rapport, nous proposons un protocole de routage pour la collecte de données à partir des nœuds capteurs au nœud Sink dans les réseaux de capteurs hétérogénes. Notre proposition détecte et bénéficit des liens asymétriques causés par cette hétérogénéité. Notre proposition, notée MURA (MUlti-RAnge convergcast routing protcol), (1) prévoit un taux de livraison élevés, (2) réduit le nombre de paquets dupliqués et (3) réduit le nombre de sauts de bout-en-bout en exploitant les liens asymétriques.

Mots-clés : Réseau de capteurs, Hétérogénéité, Liens asymétriques, Routage convergcast

1 Introduction

Currently, we observe that several industrials are interested in the design and in the manufacturing of wireless sensors. Such diversity implies the cohabitation of several types of hardware and leads to heterogeneity: energy heterogeneity, computational heterogeneity and radio link heterogeneity. In this work, we focus on this last type of heterogeneity. Hence, in this paper, we consider heterogeneity as a radio link heterogeneity where some sensor nodes have a more important transmission range. In WSNs, link heterogeneity can be especially caused by the variation of power levels in deployed nodes' devices. This heterogeneity leads to the existence of asymmetric links in the network.

The existence of asymmetric links induced by the presence of several transmission ranges in the network may cause a performance limitation of protocols that do not take into account this kind of links because asymmetric links are essentially pruned. Consider the example in Figure 1, where node S wants to send data to the destination D. The dotted semicircles represent the radio range of nodes. In this example, node A has a radio transmission range equal to twice that of the other nodes.

First, the A-C link is an invalid link according to symmetric MAC protocols. Indeed, consider the example where nodes need to exchange control packets such as RTS-CTS-ACK: node A cannot receive either CTS or ACK from node C.

Second, even when asymmetric MACs are used, reactive routing protocols which use Route Request/Response as AODV [1] or DSR [2] fail on a such topology. Indeed, the request will follow the path S-A-C-D while the response can not follow the opposite path because A is not within communication range of C.

To address the heterogeneity problem and especially the problem of asymmetric links caused by heterogeneous transmission ranges, there are two different solutions. The first category avoids the use of these asymmetric links by pruning them. In the second category, long-range links, even if unidirectional, are exploited. This tends to decrease the number of hops in the path which reduces the end-to-end transmission delay. Our proposal follows the second category. The aim of our proposal is to ensure reliable data collection in an heterogeneous network while taking advantage of the presence of different transmission ranges. The goal of our proposal is to increase the delivery ratio, to decrease the hop counts and to minimize the number of duplicate packets delivered.

This paper is organized as follows. In the next section, we summarize the related work. Section 3 and 4 respectively present our proposal and its performance evaluation. Section 5 concludes this work.

2 Related Work

As we previously mentioned, there are two categories of protocols for heterogeneous networks: protocols which avoid the use of the large transmission range of certain nodes and protocols which promote the use of such nodes.

First, several works that consider heterogeneous network basically hide this heterogeneity by eliminating the asymmetric links caused by the different transmission ranges of heterogeneous nodes. Some of these algorithms are based on

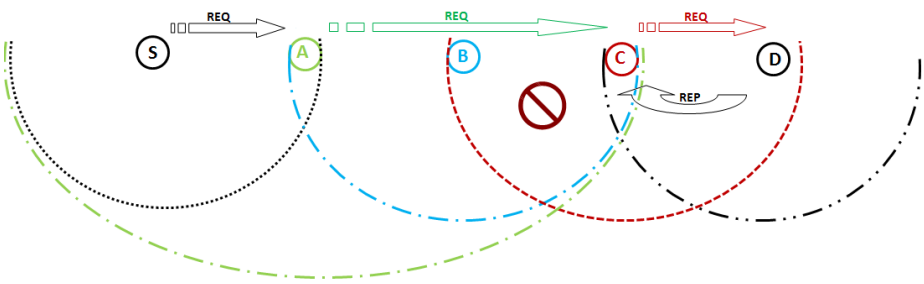


Figure 1: Asymmetric links and routing behavior

the detection of asymmetric links by exchanging neighbor list. By receiving this message, each node can deduce the list of symmetric links. Then nodes will use only these symmetric links in the routing phase. Some others, such as COMPOW [3], understand that the network is heterogeneous in nature but calculate a transmission power for each node such that their transmission ranges are identical. This transmission range is calculated to reduce the interference and to ensure connectivity between nodes.

Second, some protocols exploit heterogeneous nodes to take advantage of their different transmission ranges. In [4], authors propose Early Unidirectionality Detection and Avoidance EUDA routing protocol. The idea of EUDA is to exchange informations such as transmission range, noise level, minimum Signal to Noise Ratio, etc. When an intermediate node B receives a message from a source node A, it compares its highest transmission range to estimate the distance between itself and the source node A. If the value of the estimated distance from node B to A is larger than the transmission range of node B, node B considers the radio link to A as unidirectional, and the received message is dropped. When a transmission range of the intermediate node B is equal to or larger than the estimated distance towards source node A, then the message from A will be processed. TRIF proposes a similar approach [5]. TRIF is a mechanism used jointly with RREQ/RREP-based routing protocols. It relies on the idea that a source node (or a relay node that forwards a request) sends its RREQ with different transmission range levels. TRIF assumes that the transmission range is adjustable, thus TRIF sends each RREQ successively with decremented transmission range level. The source node adds in the header of the RREQ the transmission range level used when sending this request. The receiver processes the RREQ if the level mentioned in the header packet is less than or equal to its own transmission range level. If the power level used to send the RREQ is higher than the power level available at the receiving node, then this request is dropped: the receiving node concludes that it has received this request via an asymmetric link (it just can hear messages from this source, but its response may not reach that node). A receiving node forwards the query using its transmission range level. However, this receiver uses at least a transmission power equal to the minimal power transmission used by the previous forwarder. This minimum transmission power will be used to send the RREP response: the use of the minimum transmission power is justified by the minimization of the energy consumed during the response and to reduce interferences.

Our proposal, denoted MURA, is similar to TRIF. With MURA, a node receiving a message that needs to be forwarded to the sink decides whether to participate or not in the process of the data collection. This decision will be made based on the information contained in the received message and based on its neighbor list. Indeed, instead of sending each message with several levels of transmission range to avoid asymmetric links (as TRIF does), our proposal MURA sends one message, but it requires knowledge of the neighborhood. MURA will be described in the next section.

3 Convergecast Routing with Asymmetric Links in Heterogeneous Wireless Sensor Network

In this paper, we propose a convergecast routing protocol dedicated to heterogeneous WSN. MURA (MUlti-RAnge convergecast routing protocol) benefits from the large transmission ranges to collect data while avoiding redundant messages and reducing the hop count from source sensor nodes to the sink destination node. Our proposal can be divided into two phases: a neighbor discovery phase and a data-collection phase.

3.1 Network model and hypothesis

We consider a WSN with a large number of sensor nodes and one static sink node. We assume that there are several types of nodes having different transmission ranges. At t = 0, we suppose that all sensor nodes are deployed. We assume that the transmission range level of the sink node is equal to the highest transmission range level of any node in the network. No real geographic information is available for any network node. But we suppose that sensor nodes have information about their rank to the sink. These ranks will be used as a gradient when there is a data to send to the sink node. It is used as pseudo-geographic information to route data to the sink node. We suppose that sensor nodes nearest the sink node have a smaller rank. This rank can be obtained as decribed in [6] or [7].

3.2 Neighbor discovery phase

This phase can be divided into two steps, triggered by the sink node. Supposing that sensor nodes have a rank, each node broadcasts two message.
First, each node sends a `Hello_Message` to discover its neighborhood. In this message, each node puts its own ID, rank (which is equal to 0 for the sink node) and transmission range. The sink node is the first to send this message. Other nodes, by receiving such a message from a neighbor, will send a unique `Hello_Message`. Upon receiving this message, each node adds to its neighbors table the node that just advertised. This table contains the ID, the rank and the transmission range level of that neighbor.
At the second step, each node broadcasts a second message named `Heard_Neighbor_Message`. This step is also started by the sink node. This message contains the ID of the sending node, its rank, its transmission range level and a list of the IDs of the neighbors known to it. At the end of this phase, each node will

have a list of its neighbors, their ranks, their transmission range levels and their neighbor list.

3.3 Data-collection phase

In this phase, when a sensor node has data to send to the sink node, it broadcasts the data message to its neighborhood. In the header of this data message, the sender node puts its rank and its transmission range level. Each sender node starts a timer, `timeout_relayed`, during which it verifies if its message is relayed. If the timeout elapses while the sender node is not informed that its message was relayed, the sender node still tries a second time. The calculation of this timeout will be discussed in the next section. Upon reception of a broadcast message, a node applies the algorithm described in Figure 2: it first verifies if it is closer to the sink node by comparing its rank with that of the sender node. If this is not the case, that node is not a candidate to relay that message because it is farther than the sender from the sink. This non-candidate node drops the received message. If it is a candidate to relay this message, the node computes a timeout called `timeout_to_relay` and enters in a contention phase. The objective of this timeout is to favor the node closest to the sink node (having a smaller rank) and the node with the highest transmission range. This timeout is also discussed in the next section. If a candidate node detects that the message is being forwarded by another node, the contention phase is ended. If the timeout elapses and no other node has forwarded the message, then this node proceeds to the comparison of its transmission range level with that of the sender node. If the transmission range level of that node is greater than or equal to the transmission range level of the sender node, then this node will forward the message. It will broadcast this message to its neighborhood by changing the header (ID of the sender node, rank and transmission range level). If the transmission range level of this candidate node is smaller than that of the source node, then it first verifies if the link between himself and the sender node is symmetric. This node can check the symmetry of the link by verifying in its neighbor table if the sender node can hear the messages sent by that node. If the link is symmetric, this node can relay the message. Else, if the candidate node deduces that the link between the sender node and itself is an asymmetric link, this node tries to find a common neighbor node with the sender node (i.e a node that can communicate with both the sender and the candidate node). If such a node exists, the candidate node forwards the data message and sends an acknowledgment message to the common node. The latter will forward this acknowledgment to the sender node. This algorithm is iterated until the message arrives at the sink node. The sink node, when receiving a data message, responds by broadcasting an ACK message.

3.4 Example

Consider the example shown in Figure 3. We assume a basic network composed of a linear arrangement of four evenly spaced sensor nodes (S, A, B and C) and one Sink node (D). Node A has a transmission range level equal to two times the other nodes transmission range levels. We assume that each node has a rank that determines its relative position to the sink node (D). This rank is represented by the number written below each node in Figure 3. After the

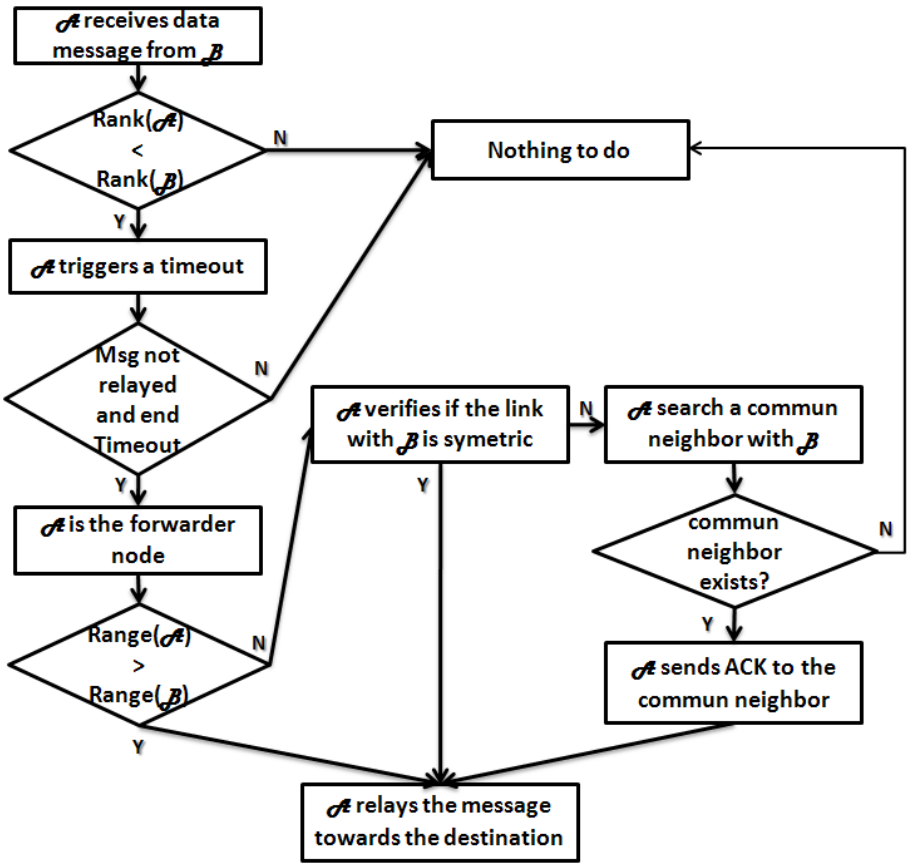


Figure 2: Data-collection Algorithm

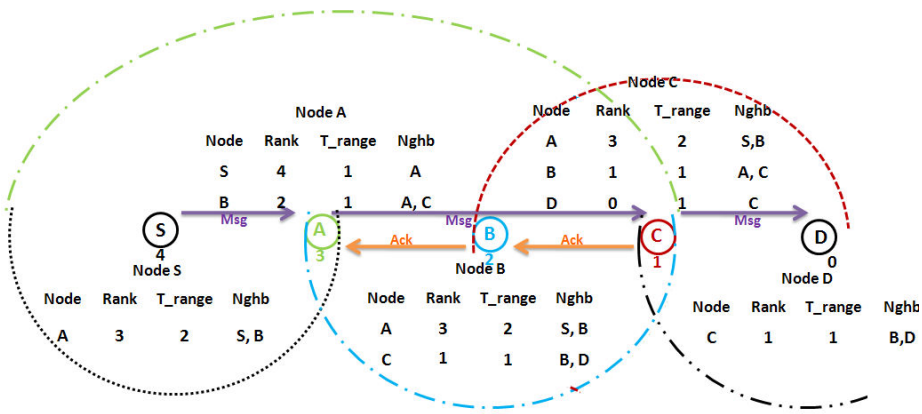


Figure 3: Example: 4 linear sensor nodes (S, A, B and C) and a Sink node (D). Node A has a transmission range level equal to two times the other nodes transmission range levels.

first phase of neighbor discovery, each node has built its neighbor table (shown next to each node in Figure 3). The source node S wants to send a message to the Sink node D. It broadcasts the message, putting in the header its ID, rank and transmission range. The message is received by node A. Node A is a candidate to relay that message because it has a smaller rank and has not heard another node relaying it. So, at the end of the timeout `timeout_to_relay`, node A broadcasts this message in its neighborhood by updating the information contained in the header (A puts its own ID, rank and transmission range level). The source node S will hear its message relayed by A, so it will stop its timeout (`timeout_relayed`). The message relayed by A will also be received by nodes B and C. Each of them will launch a timer `timeout_to_relay`. The timer of node C elapses first since node C has the smallest rank (C has a rank equal to 1 while the rank of B is 2). Node C notices that its transmission range is smaller than that of node A, so C first verifies if the link between itself and A is symmetric. To verify that, node C checks in its neighbor table if it belongs to the neighbor list of node A. This is not the case, since the neighbors heard by A are S and B only. So C checks its neighbor table to find if there is a common neighbor between itself and source node A. Node C finds that node B is a common neighbor between itself and node A. So C broadcasts the data message towards the sink and sends an ACK to node B that in turn sends it to node A. By receiving this ACK, node B, which is in contention phase with node C, removes its timer and drops the message received from node A. The sink, which is the final destination, upon receiving the data message, will broadcast an ACK to the last relay node C.

3.5 Timeout calculation

Our proposal defines two timeouts: The first timeout `timeout_relayed` is calculated by the sender nodes and the second one, `timeout_to_relay`, is calculated by the forwarder candidate nodes.

- `Timeout_to_relay`: The `timeout_to_relay` is calculated by the candidate nodes which could relay the data message and which enter into the contention phase. The purpose of this timeout is to introduce priorities to candidate nodes. The node with the highest priority will be the next hop to relay the message toward the Sink node. The goal is to favor nodes closer to the destination and to promote the use of the longest links to reduce the number of hops. Thus, the timeout is made proportional to the rank of the candidate node (the smaller the rank, the shorter the timeout). Furthermore, this timeout is made inversely proportional to the transmission range level of candidate node (the higher the transmission range, the shorter the delay before relaying).

- `Timeout_relayed`: This timeout is initiated by sender nodes. It is used to ensure that the data message is relayed by another node toward the Sink. `Timeout_relayed` is the sum of two components. The first one is the upper bound of `Timeout_to_relay`, the second one is the estimated propagation time of the two ACK messages. Indeed, the maximum time that a node could wait to hear its message relayed by a direct neighbor is equal to the upper bound of the waiting time the candidate node computes to relay

the message. If the message is relayed by a node that the sender node can not hear, the sender node must additionally wait twice the propagation delay of the ACK message before possibly receiving a confirmation.

4 Performance evaluation

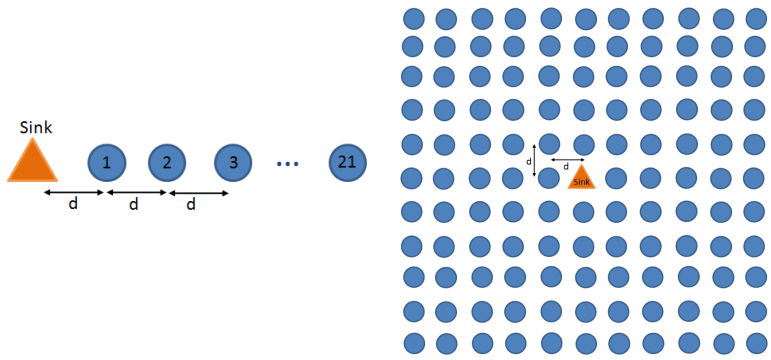
In this section, we describe the parameters used in simulation to evaluate the performance of our proposal. Then we present the main results of our simulations when comparing MURA to TRIF [5].

4.1 Simulation Parameters

We consider two kinds of networks: First, a linear topology as represented in Figure 4(a). Second, a grid topology as represented in Figure 4(b). In both networks, we select a random number of source nodes in the network which will send data messages periodically to the sink node. Three kinds of nodes are deployed in each network: normal-nodes having a regular transmission range and super-nodes having a transmission power range equals to three and six times the transmission range of normal-nodes. Table 1 summarizes the main characteristics of the network.

Parameter	Value
Sensor Nodes	21 and 120
Node range	1x, 3x and 6x regular range
Number of source nodes	1 .. 9
Number of packet sent	1 packet per minute /source node
Propagation	Unit Disk Graph
MAC Protocol	CSMA/CA
Confidence Interval	95%
Simulator	WSNet [8]

Table 1: Simulation Parameters



(a) Linear Topology (b) Grid Topology

Figure 4: Topologies Simulated

4.2 Linear Topology

Figure 4(a) represents the simulated topology. The distance between each node is constant and uniform. We represent in Figure 5 the results of our simulation for a network with 10% and with 50% of super-nodes randomly deployed in the network.

4.2.1 Delivery Ratio

In figure 5(a), we represent the delivery ratio of our MURA and TRIF. Both protocols offer a high delivery ratio, close to 100%.

4.2.2 Duplication Ratio

In this section, we calculate the duplication ratio which is defined as the amount of duplicated packets received at the final destination (the Sink node) divided by the the total number of received packet. In figure 5(b), we represent the duplication ratio. Our proposal offers a low duplication ratio compared to TRIF protocol. This is due to the fact that in MURA the sink, when receiving a packet, responds by an ACK using the same transmission range as the last relay node. Hence, nodes in a contention for that message which did not hear the message being relayed, will stop their contention phase and so will not send the packet after they heard the ACK from the sink node.

4.2.3 Average Hop Count

Figure 5(c) represents the average number of hops a packet makes to reach the final destination. We see that MURA offers a low average hop count compared to TRIF. This is due to the fact that, in MURA, the packet can be transmitted using an asymmetric link so the number of hops to reach the destination is less than TRIF. In TRIF, packets only use symmetric links.

4.3 Grid Topology

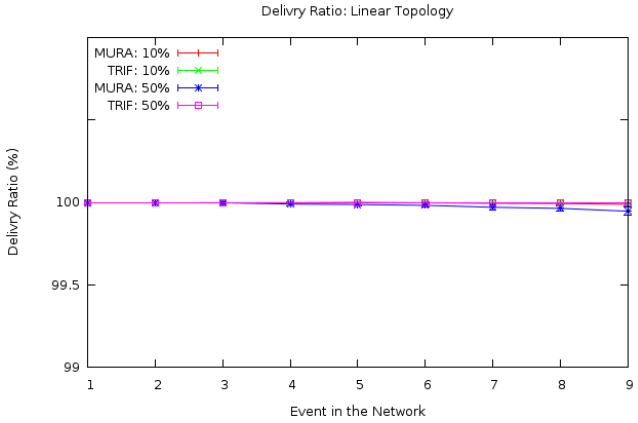
In the following, we generate a grid topology as shown in Figure 4(b). The percentage of total super-nodes varies from 10% to 50%.

4.3.1 Delivery ratio

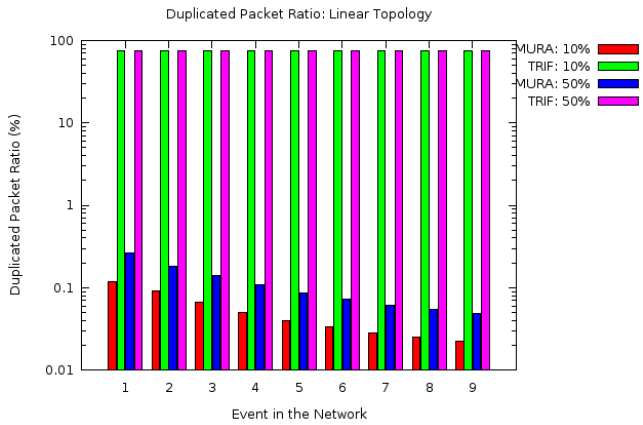
As shown in figure 6(a), both our proposal and TRIF provide a high delivery ratio, close to 100%

4.3.2 Duplication Ratio

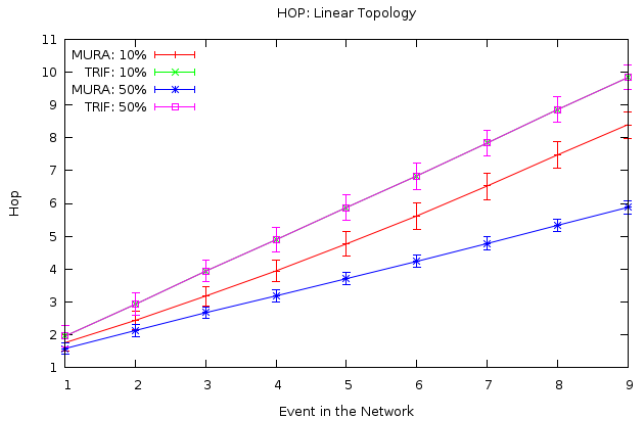
Figure 6(b) represents the amount of duplicated data for MURA and TRIF in two scenarios: with 10% and 50% super-nodes randomly deployed in the grid represented in Figure 4(b). In Figure 6(b), we see that, in all cases, the amount of duplicated received packets with our proposal MURA is less then the amount of duplicated received packets with TRIF. We can also see on Figure 6(b) that, when increasing the number of super-nodes with TRIF, the average of the duplication ratio increases because each super-node repeatedly sends the message with decreasing transmission range. Whereas with our proposal MURA, the duplication ratio remains very low compared to TRIF because with MURA, each node only sends only one message using its own transmission range level.



(a) Delivery Ratio

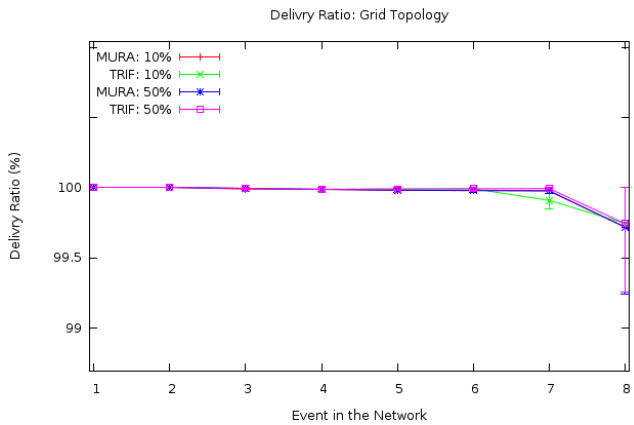


(b) Duplicated received packet Ratio

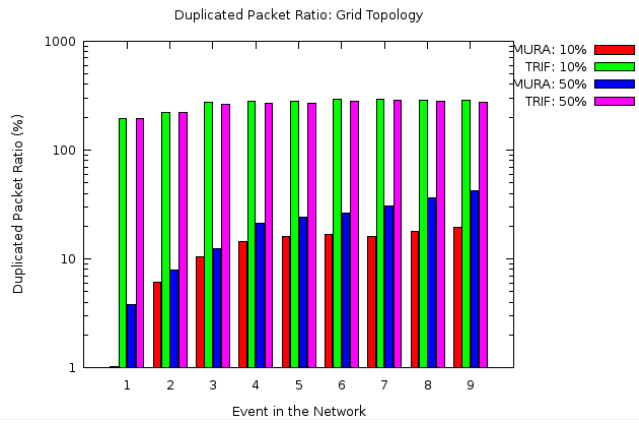


(c) Average Hop Count

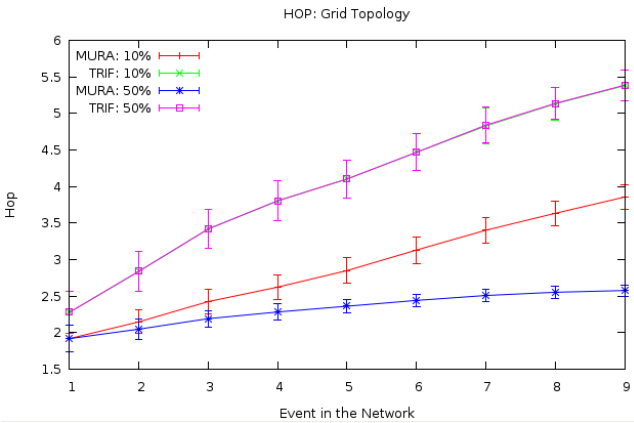
Figure 5: Linear Topology Performance



(a) Delivery Ratio



(b) Duplicated received packet Ratio



(c) Average Hop Count

Figure 6: Grid Topology Performance

4.3.3 Average Hop Count

Finally, we evaluate the average hop count that a packet makes to reach the final destination. Figure 6(c) represents the average hop count using TRIF and MURA proposal with different scenarios. As we can see in Figure 6(c), MURA offers a lower hop count than TRIF. This is due to the fact that in MURA, we exploit the asymmetric links when gathering the data from sensors to sink nodes.

4.4 Energy Consumption

In this section, we are interested in the evaluation of the energy consumption of both MURA and TRIF. Our proposal, MURA, requires neighborhood knowledge and there is a tradeoff between the energy cost to get this information and the energy cost of the data collection phase. Indeed, for frequent data collection applications, the cost of the neighborhood discovery in a quasi-static network may be insignificant compared to the cost of sending periodic data to the sink node. In this section, we start by evaluating the cost of the neighborhood discovery phase and the data collection phase. We calculate the number of messages sent and received for each phase for a high and a low density in the network. We compare the energy consumption of both MURA and TRIF at the end of this section.

4.4.1 Parameters and Hypothesis

We assume that we have a uniform deployed network. We also assume that data packets and control packets have the same size. We suppose:

- N: Number of nodes in the network.
- A: Number of normal-nodes.
- B: Number of super-nodes (heterogeneous nodes).
- V: the geographical density. We assume that the geographical density is uniform.
- x: represents the super-node range. Hence the number of neighbors for a super-node is equal to $x^2 * V * \pi$.
- R: Average hop counts to reach the sink node.
- S: Number of data messages to send to the sink node. We suppose that the number of data messages is evenly divided between the two types of node: normal and super nodes.

4.4.2 Cost of the Neighborhood Discovery Phase

In this section, we calculate the number of messages sent and received during the neighborhood discovery phase.

1. Sent messages
   - $\underline{N}$ messages of type `Hello_Message` will be sent.
   - $\underline{N}$ messages of type `Heard_Neighbor_Message` will be sent.
2. Received messages
   - Each message of type `Hello_Message` (resp. `Heard_Neighbor_Message`), sent by a normal-node, will be received by $V * \pi$ nodes. Since there is 'A' normal-nodes, there are $\underline{A * V * \pi}$ messages of type `Hello_Message` (resp. `Heard_Neighbor_Message`) received.

- Each message of type `Hello_Message` (resp. `Heard_Neighbor_Message`), sent by a super-node, will be received by $x^2 * V * \pi$ nodes. Since the is 'B' super-nodes, there are $\underline{B * x^2 * V * \pi}$ messages of type `Hello_Message` (resp. `Heard_Neighbor_Message`) received.

Hence the number of messages sent and received for the neighborhood discovery phase is expressed in 1

$$Cost_Neighbor_Discovery = 2 * (N + A * V * \pi + B * x^2 * V * \pi) \quad (1)$$

4.4.3 Cost of the Data-Collection Phase

In this section, we calculate the number of message sent and received during the data collection phase.

1. Sent messages
   - $\underline{S}$ messages of type data_message will be generated.
   - For the S data messages relayed by the R intermediate nodes, so there will be $\underline{S * R}$ messages relayed.
2. Received messages
   - The number of data_message generated by the normal-nodes is equal to $(A/N) * S$. Each data_message sent by a normal-node will be received by $V * \pi$ nodes. So, all the data_message sent by the normal-nodes will generate $\underline{(A/N) * S * V * \pi}$ reception.
   - The number of data_message generated by the super-nodes is equal to $(B/N) * S$. Each data_message sent by a super-node will be received by $x^2 * V * \pi$ nodes. So, all the data_message sent by the super-nodes will generate $\underline{(B/N) * S * x^2 * V * \pi}$ reception.
   - There will be an $(A/N) * R * S$ data_message relayed by normal-nodes. These messages will generate $\underline{(A/N) * R * S * V * \pi}$ reception.
   - There will be an $(B/N) * R * S$ data_message relayed by super-nodes. These messages will generate $\underline{(B/N) * R * S * x^2 * V * \pi}$ reception.

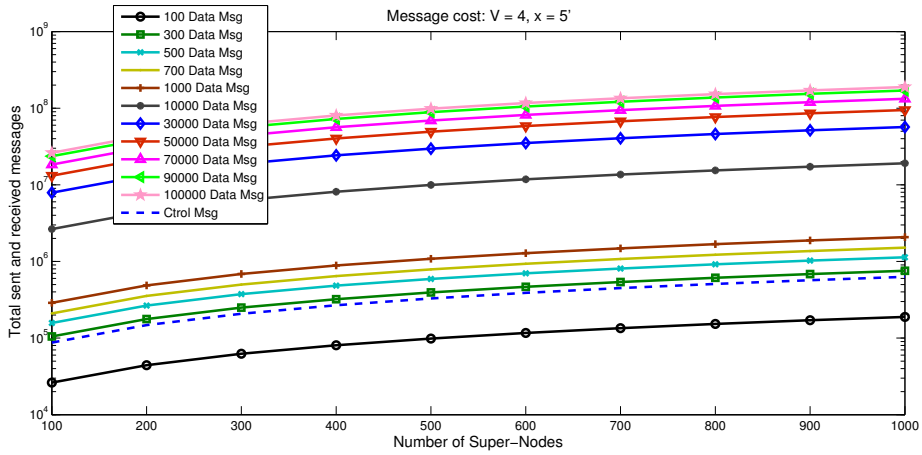
Hence the number of message sent and received for the data collection phase is represented as in 2

$$Cost_Data_Collection = (1+R)*(S+(A/N)*S*V*\pi+(B/N)*S*x^2*V*\pi) \quad (2)$$

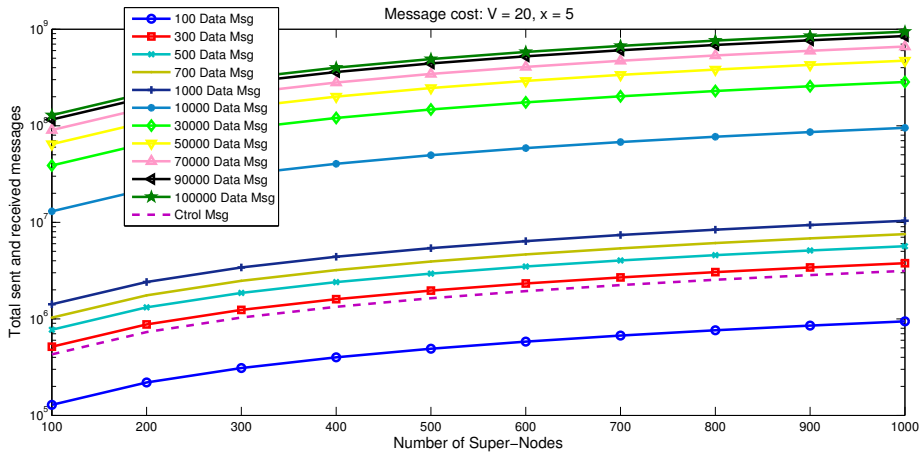
4.4.4 Simulation results

Here we fixed the number of nodes in the network (N=1000 nodes) and the range of heterogeneous nodes which is equal to 5 times regular range(x=5). We evaluate for a low and a high density (V=5 and V=20) networks the number of messages sent and received. Figure 7(a) and Figure 7(b) represent the number of total sent and received messages for a low and a high density networks, respectively.

We note that whatever the density in the network, when the number of data messages exceeds 1/3 of total nodes in the network, the cost of the neighbor discovery phase is covered by the data collection phase. By increasing the number of data messages, the cost of the neighborhood discovery phase in a quasi-static network is insignificant compared to the cost of the data collection phase.



(a) Low Density Network



(b) High Density Network

Figure 7: Total Number of Message Sent and Received for neighbor discovery and data collections phases

4.4.5 MURA and TRIF energy consumption comparison

Cost of TRIF protocol In this section, we calculate the number of messages sent and received when using TRIF protocol.

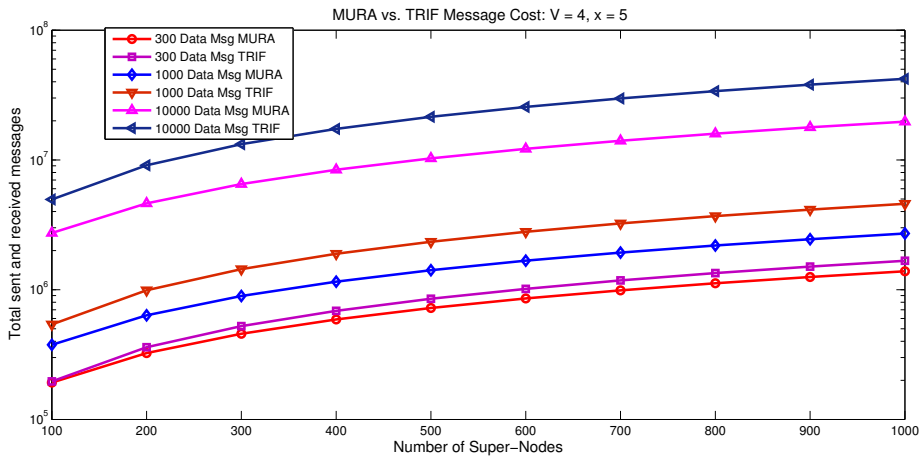
1. Sent messages
   - $\underline{(A/N) * S}$ messages of type data_message will be generated by the normal-nodes.
   - There will be $(B/N) * S$ messages of type data_message will be generated by each super-node. Since a super node sends x messages, the total number of messages generated by super-nodes will be $\underline{x * (B/N) * S}$.
   - There will be $\underline{(A/N) * R * S}$ data_message relayed by normal-nodes.
   - There will be $(B/N) * R * S$ messages of type data_message which should be relayed by super-nodes. Since a super node sends x messages, the total of message relayed by super-nodes will be $\underline{x * (B/N) * R * S}$.
2. Received messages
   - The $(A/N) * S$ data_message sent by normal-nodes will generate in the network $\underline{(A/N) * S * V * \pi}$ reception.
   - Each of the $(B/N) * S$ super-nodes which generate a data_message will sends x messages which will generate $\underline{(B/N) * S * \sum_{i=2}^{x} i^2 * V * \pi}$ total reception.
   - The $(A/N) * R * S$ data_message relayed by normal-nodes will generate $\underline{(A/N) * R * S * V * \pi}$ reception.
   - Each of the $(B/N) * R * S$ super-nodes which should relay a data_message will sends x messages which will generate $\underline{(B/N) * R * S * \sum_{i=2}^{x} i^2 * V * \pi}$ total reception.

Hence the number of message sent and received for TRIF protocol is calculated as the sum of the previous sent and received data messages. The total number of message sent and received is calculated as in 3

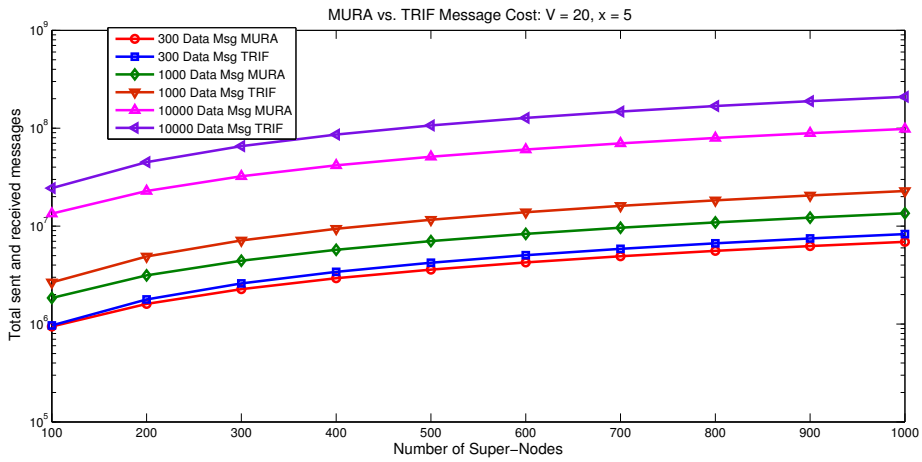
$$\begin{aligned} Cost_TRIF = A/N * S * (1 + V * \pi + R + R * V * \pi) + \\ (B/N) * x * S * (1 + R + ((V * \pi * (x+1) * (2 * x + 1))/6) + \\ ((R * V * \pi * (x+1) * (2 * x + 1))/6)) \end{aligned} \quad (3)$$

Simulation results Here we compared the amount of sent and received message for MURA and TRIF when used on a high and a low density networks. Figure 8(a) and Figure 8(b) respectively represent the total number of message sent and received in a low and high density network. In both cases, MURA uses less messages than TRIF and hence consumes less energy.

We see also that the difference between the two curves representing TRIF and MURA increases when we increase the number of data_message generated in the network. MURA consumes less energy than TRIF because sending one data_message in addition generate one message sent and $\underline{V * \pi}$ reception with MURA. Whereas with TRIF sending one data_message in addition generate $\underline{x}$ messages sent and $\underline{\sum_{i=2}^{x} i^2 * V * \pi}$ reception.



(a) Low Density Network



(b) High Density Network

Figure 8: MURA vs. TRIF: Total Number of Message Sent and Received

5 Conclusion

In this report, we proposed a data collection algorithm for networks composed of nodes having different transmission ranges. Our proposal benefits from asymmetric links caused by heterogeneity in transmission ranges of sensor nodes. Simulations highlight that our proposal meets the requirements of providing a high delivery ratio, a lower hop count and a low duplication ratio compared to TRIF protocol. We studied and evaluated the energy consumption of the neighborhood discovery and data-collection phases. Our proposal requires neighborhood knowledge and there is a tradeoff between the energy cost to get hold of this information and the energy cost saved on data traffic. Indeed, for periodic data collection applications, the cost of doing the neighborhood discovery in a quasi-static network may be insignificant compared to the cost of sending periodic data to the sink node. Hence, we compared the energy consumption of MURA and TRIF by calculating the amount of sent and received messages. It was shown that MURA consumes less energy than TRIF when the number of data message exceeds the 1/3 of total nodes in the network. Currently we are also working on taking into account other metrics to calculate the timeout `Timeout_to_relay` such as the amount of energy available in each node. The use of such metrics will avoid over-exploiting some nodes during the routing phase. The goal is to spread the energy consumption and therefore to increase the lifetime of the network.

References

[1] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing. United States, 2003. RFC Editor.

[2] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.

[3] Swetha Narayanaswamy, Vikas Kawadia, R. S. Sreenivas, and P. R. Kumar. Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the compow protocol. In *in European Wireless Conference*, pages 156–162, 2002.

[4] Young bae Ko, Sung ju Lee, and Jun beom Lee. Ad hoc routing with early unidirectionality detection and avoidance, personal wireless communications. In *In Proc. of International Conference on Personal Wireless Communications (PWC 04*. Springer, 2004.

[5] Thang Le, Prasun Sinha, and Dong Xuan. Turning heterogeneity into an advantage in wireless ad-hoc network routing. *Ad Hoc Netw.*, 8(1):108–118, 2010.

[6] B. Romdhani, D. Barthel, and F. Valois. Strategy of self-organization in sensors and actuators networks. In *WiMob'10: IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, Niagara Falls, Canada, 2010.

[7] Azlan Awang, Xavier Lagrange, and David Ros Sanchez. RSSI-based forwarding for multihop wireless sensor networks . *Lecture notes in computer science*, 5733:138 – 147, august 2009.

[8] E. Ben Hamida, G. Chelius, and J. M. Gorce. Scalable versus accurate physical layer modeling in wireless network simulations. In *PADS'08: Proceedings of the 22nd Workshop on Principles of Advanced and Distributed Simulation*, pages 127–134, Washington, DC, USA, 2008. IEEE Computer Society.

Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399