Exploiting Asymmetric Links in a Convergecast Routing Protocol for WSNs

Bilel Romdhani, Dominique Barthel, Fabrice Valois

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Exploiting Asymmetric Links in a Convergecast Routing Protocol

Bilel Romdhani — Dominique Barthel — Fabrice Valois

N° 7586

November 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

Exploiting Asymmetric Links in a Convergecast Routing Protocol

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

Thème : 1
Équipes-Projets SWING

Rapport de recherche n° 7586 — November 2011 — 24 pages

Abstract: Most of the existing routing protocol designed for WSNs assume that links are symmetric which is in contradiction to the reality of these networks. Indeed, asymmetric links cannot be ignored in WSNs as they can be predominant. The apparition of asymmetric links can dramatically decrease routing protocols that are not designed to support them by decreasing the delivery ratio and increasing the duplicated packet received at the destination. Obviously, most of the existing routing protocols prune the asymmetric links and only maintain the symmetric ones. From our point of view, the asymmetric links have to be considered as they can be effective in the network connectivity insurance. Moreover, they open new opportunities to improve the performance of routing protocols. From this perspective and in order to take benefit from asymmetric links, we propose a routing protocol for data collection in WSNs called AsymRP (Asymmetric Routing Protocol). AsymRP is a convergecast routing protocol which is based on a 2-hop neighbor knowledge combined with implicit and explicit source routing acknowledgment. Our proposal takes advantage of asymmetric links, enables the network to achieve higher delivery ratio while reducing significantly the number of duplicated packets and hop counts. Our simulation results show that our proposal AsymRP can significantly outperform traditional routing protocols in the presence of asymmetric links in the network.

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

Exploitation des liens asymétriques pour un routage convergecast pour les WSNs

Résumé : Plusieurs études et approches ont conçu des protocoles de routage pour les réseaux de capteurs sans fil (WSNs) et les réseaux de capteurs et actionneurs (WSANs). Néanmoins, la plupart d'entre eux supposent que les liens constituants le réseau sont symétriques qui est en contradiction avec la réalité de ces réseaux. En effet, les liens asymétriques ne peuvent pas être ignorés dans les WSNs et les WSANs, car ils peuvent être prédominants. L'apparition de liens asymétriques peut considérablement dégrader les performances des protocoles de routage qui ne considèrent pas ce type de liens.

De notre point de vue, les liens asymétriques doivent être considérés comme ils peuvent être efficaces dans l'assurance de la connectivité du réseau. Par ailleurs, ils ouvrent de nouvelles opportunités pour améliorer les performances des protocoles de routage.

Dans cette perspective et afin de tirer profit de ces liens asymétriques, nous proposons un protocole de routage pour la collecte des données dédiés aux WSNs et WSANs appelé AsymRP (Asymmetric Routing Protocol). AsymRP est un protocole de routage convergecast qui est basé sur une connaissance de voisinage à 2-sauts combiné avec l'utilisation des messages d'acquittements (ACKs) implicites et une technique de routage de messages ACKs explicites. Notre proposition tire profit des liens asymétriques, permet d'assurer un taux de livraison élevé tout en réduisant significativement le nombre de messages dupliqués et le nombre de sauts de bout-en-bout. Nos résultats de simulation montrent que notre proposition AsymRP surpasse nettement les protocoles de routage traditionnels lors de la présence de liens asymétriques dans le réseau.

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

1 Introduction

Several studies and approaches have designed routing protocols for Wireless Sensor Networks (WSNs). Nevertheless, most of them have assumed that links are symmetric which is in contradiction to the reality of these networks. Indeed, asymmetric links cannot be ignored in WSNs as they can be predominant. This asymmetric links can be caused by the diversity of the devices used (the existence of different transmission ranges in the network), by the real deployment (presence of interference in the network) and by the environment (presence of noise source for example). In this work, we are interested in the presence of asymmetric links whatever the source of the asymmetric links.

Several approaches have proposed protocols for WSNs to improve the network performance. Nonetheless, they are only based on symmetric links and ignore asymmetric links which are pruned. This selection can turn out inefficient as avoiding asymmetric links may cause performance fall [1]. In figure 1, we depict an example where the node S is sending data to the sink D. We display the radio range of each node by dotted semicircles and we assume that the transmission range of the node A is two times greater than those of the other nodes. We can highlight, from this example, two interesting observations. On the one hand, the link between the node A and the node C is not considered when a protocol relies on symmetric links. If the nodes have to exchange RTS-CTS-ACK packets, the node A does not receive neither CTS nor ACK from the node C. On the other hand, a reactive routing protocol as [2] or [3], the Route Request/Response would be lost. In fact, if the Route Request follows the route S-A-C-D, the Route Response is blocked at the node C as the node A is not within the range of the former node.

To overcome the constraints imposed by the asymmetric links, two solutions can be considered. The first one tends to prune all asymmetric links [4] [5]. Pruning asymmetric links can have several drawbacks: it can cause the lose of the network connectivity or it can deteriorate the performance of routing protocols for example. For the second solution, the aim is to reduce the path length to limit end-to-end transmission delays. For this, the use to long-range links is foreseen [6] [7]. The protocol presented in this report joins the concept of the second solution. In fact, it aims at ensuring efficient data collection in WSNs while exploiting asymmetric links whatever the source of this asymmetry . We target through this work to ensure high delivery ratio with minimizing the number of hop counts and the number of duplicated packets delivered at the destination.

The remainder of this report is as following. In Section II we present the related work. In section III, we present the problem statement. In Section IV and V, we respectively describe our proposal and evaluate its performance. Finally, we discuss and present two extensions of our proposal. Section VII concludes the report.

2 Related Work

As we mentioned previously, there are two categories of routing protocols dealing with asymmetric links: protocols which avoid the use of asymmetric links and protocols which promote the use of such links.

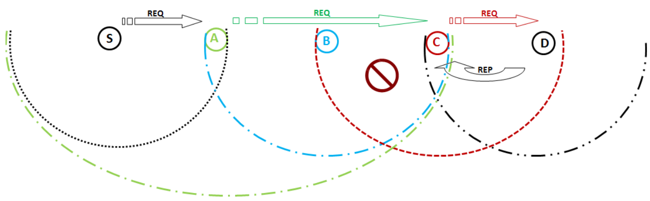


Figure 1: Asymmetric links and routing behavior

For the first category, several routing works consider WSNs with asymmetric links but they hide this kind of links caused specially by the different transmission ranges in the network [8] [5] [9]. First, some algorithms are based on 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. Second, there are protocols such as COMPOW [5] that suppose an heterogeneous network but calculate a common transmission range which will be used by all nodes in the networks. This transmission range is calculated to reduce the interference, to eliminate asymmetric links and to ensure connectivity between nodes. The drawback of such mechanism is that is centralized and its not dynamic with the environment.

For the second category, some protocols suppose networks with asymmetric links [10] [11] [12] [13]. [10] and [11] suppose that asymmetric links are caused by the existence of many power transmission ranges in the network. In [10], authors propose 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 asymmetric, and the received message will be 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 [11]. TRIF is a mechanism used jointly with RREQ/RREP-based routing protocols. 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. This process will continue until the data message reaches the final destination.

The drawback of these two protocols is that the estimation of the distance based on the signal to noise ratio is not a good metric to evaluate proximity [14]. Moreover, both protocols suppose only the different power transmission range as a source of the asymmetric links. Or as we mentioned previously, the asymmetric links are not only caused by this power transmission range heterogeneity but also caused by the environment and the deployment.

In [12], the authors propose *Volunteer Relaying* for information feedback. Each node needs to monitor its entire 1-hop links to identify asymmetric links among them. With *Volunteer Relaying*, when a node detects that it has two neighbors having an asymmetric link will volunteer itself to relay the link discovery and maintenance information to both neighbors. Such mechanism can cause unnecessary duplicated received packet when more than one neighbor may volunteer themselves. Some suppression techniques can be used to reduce this duplication but the performance will rely on the efficiency of those techniques.

In [13], the authors address the problem of exploiting asymmetric links in link layer. They propose DEAL [13] to discover and maintain asymmetric links. DEAL is based on two different mechanisms presented in [13]. First, DEAL is based on a feedback scheme called *Source-Specified Relay* (SSR). SSR is used as an information feedback mechanism. SSR uses local information at link layer to find the relay nodes for information feedback over the poor direction of asymmetric links. SSR have the same problem as in *Volunteer Relaying* mechanism. Second, DEAL is based on a link maintenance scheme called *Dynamic Driven Maintenance* (DDM). It supposes that the asymmetric links can be a temporal phenomenon. DDM adopts different strategies in order to use the most efficient links. DEAL addresses the problem of asymmetric links on the link layer. It supposes also that the network is dense and all the results presented in [13] suppose that neighborhood size is between 10 and 50 neighbors per node.

In this report, we propose a new routing protocol called AsymRP (Asymmetric Routing Protocol).We address the problem of the asymmetric links in a connected WSN without any constraint on the density of the network. With AsymRP, a node receiving a message to forward 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 each node neighbor list. AsymRP will be described in the next section.

3 Problem statement

The existence of asymmetric links can not be avoided in WSN. In fact, studies like in [15] [16] [17] [12] have demonstrated the presence of asymmetric links. The asymmetric links are caused by transmission power disparity, interference, real deployment, and radio irregularity [1].

With the presence of asymmetric links, the first challenge is how to detect the presence of asymmetric. To address this challenge, we present a simple mechanism based on the exchange of neighborhood table using Hello messages. Another challenge of link asymmetry in WSNs is how to exploit this links to forward data message to the sink node and in the same time how to backward an ACK message to the source node to avoid unnecessary retransmissions and reduce duplicated received message. To address this challenge, we present a mechanism based on implicit ACK and explicit ACK based on the detection of 2-hops common neighbors.

Our proposal is able to deliver the messages from source nodes to the destination sink node regardless of the topology and the network density. To verify how our proposal can exploit the asymmetric links in the network, we evaluate the number of this kind of links used in each path found to deliver data message.

4 AsymRP: Asymmetric Routing Protocol for Wireless Sensor Networks

In this report, we propose a convergecast routing protocol dedicated to heterogeneous WSN. AsymRP (Asymmetric convergcast Routing Protocol) benefits from the asymmetric links to ensure the data collection task 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.

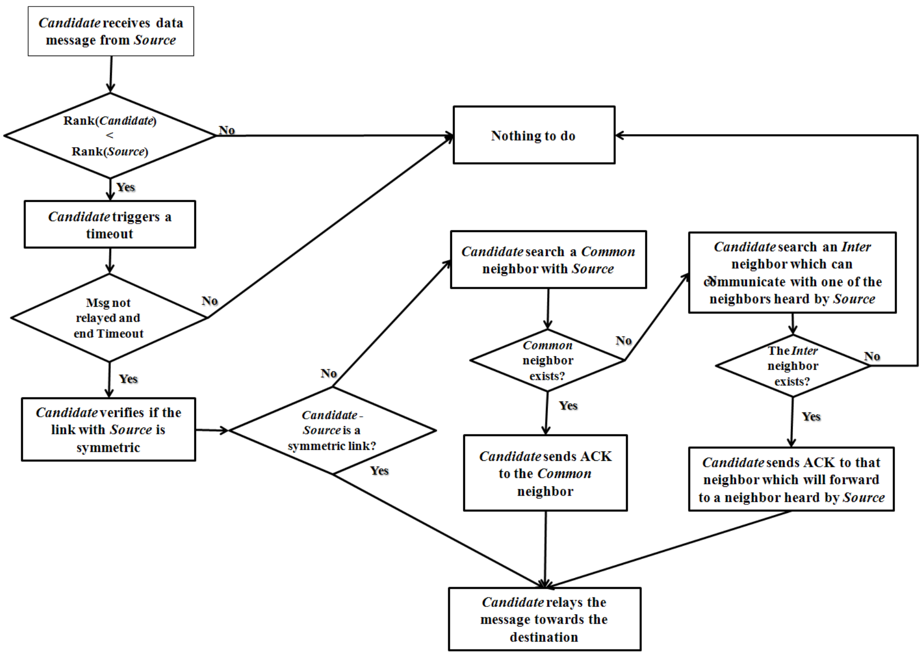


Figure 2: AsymRP: Data Collection Phase.

4.1 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 asymmetric links in the network. As we mentioned before, these links can be caused by the environment, the deployment or the heterogeneity of the power transmission range of sensors nodes in the networks. We assume also a low data traffic in the network. At t = 0, we suppose that all sensor nodes are deployed. We assume that the network is connected: each node can reach any other node in the network. No real geographic information is available for any network node. But we suppose that sensor nodes have gradient information called also rank. These ranks will be used as a gradient when there is a data to send to the sink node. We suppose that sensor nodes nearest the sink node will have a smaller rank. This rank can be obtained like in [18] or [19].

4.2 Neighbor discovery phase

The aim of this first phase is to allow to each node to have knowledge of its direct 1-hop neighbors and the neighbors of its neighbors. This phase can be divided into two steps.

- First, each node broadcasts a first message named `Hello_Msg` to discover its neighborhood. In this message, each node puts its own ID and its rank (which is equal to 0 for the sink node).

- Second, each node broadcasts a second message named `Heard_Nghb_Msg`. In this message each node puts the list of its heard neighbors. This message contains the ID, the rank of that node and the list of the IDs of the neighbors heard by this node.

At the end of this neighbor discovery phase, each node will construct a neighborhood table: this table contains the IDs of the 1-hop neighbors, their ranks and their neighbor list.

4.3 Data Collection Phase

The goal of this phase is to route data message from source sensor nodes to the sink node. In this phase, when a sensor node has data to send to the sink node, it broadcasts the data message in its neighborhood. In the header of this data message, the sender node, called *Source*, adds its ID, its rank and its neighborhood table. Each sender node starts a timer, `timeout_relayed`, during which it verifies if its message is relayed. If the timer expired and the sender node is not informed that its message was relayed by another node, it still tries a second time to broadcast its message. The calculation of this timer will be discussed in the next section (section 4.5).

When receiving a broadcast data message from a *Source* node, the receiving neighbors will apply the algorithm described in figure 2: a neighbor will first verify 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 will drop this received message. If it is a candidate to relay this message, the node computes a timer called `timeout_to_relay` and enters in a contention phase. The objective of this timer is to favor the node closest to the sink node (having a smaller rank). This timer is also discussed in the next section (section 4.5).

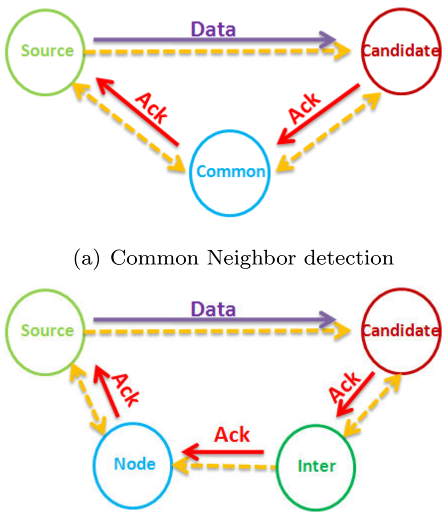
If a candidate node detects that the message was forwarded by another node, the contention phase is ended. If the timer expired and no other node has forwarded the message then this node, denoted *Candidate*, first verifies if the link between itself and the sender node is symmetric. This node can check the symmetry of the link by verifying in its neighbor table if the sender node can receive the messages sent by this *Candidate* node. If the link is symmetric, this node can relay the data message which will be used as an implicit acknowledgment message to the *Source* node. Else, if the *Candidate* node deduces that the link between the sender node and itself is an asymmetric link, two cases arises in our algorithm:

1. **First**, the candidate node tries to find in its neighbor table a common neighbor, called *Common*, with the sender node (i.e a node that can communicate with the two nodes (the *Source* node and the *Candidate* node) as in figure 3(a)). If such a node exists, the *Candidate* node forwards the data message and sends an explicit acknowledgment message to the *Common* neighbor. The latter forwards this acknowledgment until it reaches the *Source* node (see figure 3(a)).

2. **Second**, if such *Common* node does not exist, the *Candidate* node tries to find in its neighbor table whether any of its neighbors, called *Inter*, which satisfies the two conditions:

   - One of the neighbor heard by the *Source* node can receive the message sent by this *Inter* node.
   - The *Inter* node has a symmetric link with the *Candidate* node.

   If this *Inter* node exists (see figure 3(b)), the *Candidate* node forwards the data message and sends an explicit acknowledgment message to the *Inter* node which will forward that acknowledgment to one of the neighbor heard by the *Source* node which will forward that acknowledgment until it arrives to the *Source* node.



(a) Common Neighbor detection

(b) Inter Neighbor detection

Figure 3: Data collection and explicit ACK message using *Common* and *Inter* neighbors.

This algorithm is iterated until the message arrives at the sink node. The sink node, when receiving a data message, responds by broadcasting an acknowledgment message.

4.4 Example

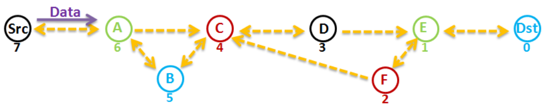
Consider the example shown in Figure 4. We assume a simple network composed of 7 sensor nodes (*Src*, *A*, *B*, *C*, *D*, *E* and *F*) and one sink node (*Dst*). We assume that each node has a rank that determines its relative position to the sink node (*Dst*). This rank is represented by the number written below each node in figure 4. We assume that links *A*-*C*, *D*-*E* and *F*-*C* are asymmetric links. After the first phase of neighbor discovery, each node has built its neighbor table. We do not represent the neighborhood tables because of space. We show a part of the table when used by AsymRP algorithm (as in 4(b) and 4(d)).

(a) The source node broadcast the data message (figure 4(a))
The source node *Src* wants to send a message to the sink node *Dst*. It broadcasts this message, and puts in the header its ID, its rank and its neighborhood table. The message is received by node *A* (figure 4(a)). Node *A* is a candidate to relay that message because it has a smaller rank and has not heard another node relaying it.

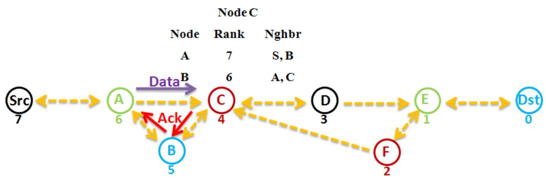
(b) *A* forwards the data message and receives explicit ACK (figure 4(b))
At the end of the timeout `timeout_to_relay`, node *A* broadcasts this message in its neighborhood (figure 4(b)) because *A* deduces that the link between itself and the source node is symmetric. Node *A* updates the information contained in the header (*A* puts its own ID, rank and its own neighborhood table). The source node *Src* will hear its message relayed by *A* (an implicit ACK), 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 starts a timeout `timeout_to_relay`. The timeout triggered at node *C* elapses first since node *C* has the smallest rank (*C* has a rank equal to 4 while the rank of *B* is 5). Node *C* check if the link between itself and node *A* is a symmetric or an asymmetric one. To verify that, node *C* checks in its neighbor table if it is in the neighbor list of node *A* (figure 4(b)). This is not the case, since the neighbors heard by *A* are *Src* 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 explicit ACK to node *B* that in turn forwards it to node *A* (figure 4(b)). By receiving this ACK, node *B*, which is in contention phase with node *C*, removes its timer and drops the message received from source node *A*.

(c) *C* forwards the data message and receives implicit ACK (figure 4(c))
The data message sent by node *C* will be received by node *D*, which will be the only candidate to forward this data message (figure 4(c)). The link between *C* and *D* is symmetric, so after the `timeout_to_relay` is elapsed, node *D* broadcasts this message in its neighborhood.

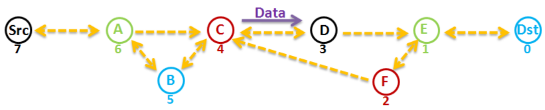
(d) *D* forwards the data message and receives explicit ACK (figure 4(d))
The data message sent by the node *D*, will be received by node *E*. Node *E* concludes that the link between itself and node *D* is asymmetric. *E* also notes that there is no common neighbor between itself and node *D*. Node *E* check if it has a neighbor which can communicate with a neighbor heard by the node *D*. Node *E* notes that its neighbor *F* can communicate with *C* which is a node heard by the source node *D* (figure 4(d)). Hence *E* forward the data message after



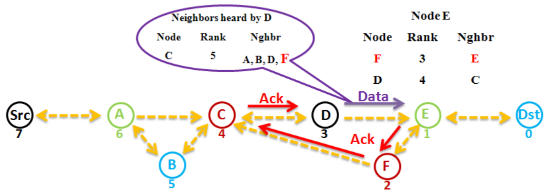
(a) The source node broadcast the data message



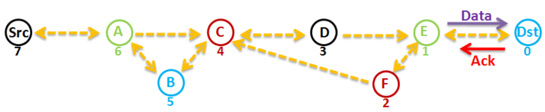
(b) *A* forwards the data message and receives explicit ACK



(c) *C* forwards the data message and receives implicit ACK



(d) *D* forwards the data message and receives explicit ACK



(e) *E* forwards the data message and receives explicit ACK

Figure 4: Example of a topology with asymmetric links caused by heterogeneous transmission range levels: Node *Src* sends data message to the *Dst* node.

the `timeout_to_relay` elapses, and sends an explicit acknowledgment message to F which will forward it to node C which in turn will send it to the source node D (figure 4(d)).

(e) *E* forwards the data message and receives explicit ACK (figure 4(e))
Finally, the broadcast message by node E will be received by the sink node Dst which replied with an acknowledgment (figure 4(e)).

4.5 Timeout calculation

Our proposal defines two timers: The first one `timeout_relayed` is calculated by sender node 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 which will relay the message toward the sink node. The goal is to favor nodes closer to the destination and to promote the use of the asymmetric links. Thus, the timeout calculated will be proportional to the rank of the candidate node (the smaller the rank, the shorter the timeout). In the case where the asymmetric links are caused by the heterogeneity in power transmission range, this timer will be also inversely proportional to the transmission range level of candidate node (the higher the transmission range, the shorter the delay before relaying). The goal of this second condition is to promote the use of longest links to reduce the number of hops.

- `Timeout_relayed`: This timer is initiated by sender nodes. It is used to ensure that the data message is relayed by another node toward the sink. This timer should be larger than the upper bound of `Timeout_to_relay` and three times the estimated propagation delay of the ACK message. 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 wait for an explicit ACK which can be send through three hop.

5 Performance evaluation

This section is divided into two parts: we begin with a theoretical study in which we evaluate the energy consumption of AsymRP and we compare it with the energy consumption of TRIF [11]. Second part, we evaluate AsymRP and TRIF using network simulation in more realistic assumptions. To be fair between the two protocols, it should be noted that, for TRIF protocol, we have added an acknowledgment message when the sink node receives a data message. This acknowledgment is used to prevent the nodes in the contention phase for it broadcasting the same message towards the sink node. Without loss of generality, in the two parts we consider the asymmetric links caused by different power transmission ranges. In this case we define two type of nodes:

- normal-nodes: which are sensor node having homogeneous power transmission range.
- super-nodes: which are heterogeneous nodes having a higher power transmission range.

5.1 Theoretical study: Numerical evaluation

In this section, we are interested in the evaluation of the energy consumption of both AsymRP and TRIF. Our proposal, AsymRP, 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 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 AsymRP and TRIF at the end of this section.

5.1.1 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. This is not true in reality since the data messages are larger than the ACK messages. But to facilitate the calculation we assume that they have the same size. For AsymRP and TRIF, the sink node is supposed to be super-node (having a higher transmission range). 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.
- R: represents the range of normal-nodes. Hence the number of neighbors for a normal-node is equal to $R^2 * V * \pi$.
- x: represents the super-node range. Hence the number of neighbors for a super-node is equal to $(x * R)^2 * V * \pi$.
- H: 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.

5.1.2 AsymRP: Cost of the Neighborhood Discovery Phase

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

Sent messages

- N messages of type `Hello_Msg` will be sent.
- N messages of type `Heard_Nghb_Msg` will be sent.

Received messages

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

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

$$\begin{gathered} Neighbor\_Discovery = \\ 2.(N + A.R^2.V.\pi + B.(x.R)^2.V.\pi) \end{gathered} \tag{1}$$

5.1.3 AsymRP: Cost of the Data Collection Phase

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

Sent messages

- S messages of type data message will be generated.
- For the S data messages relayed by the H intermediate nodes, so there will be $\underline{S.H}$ messages relayed.
- On the worst case, each relay of super-node will generate an ACK message. Hence, in the worst case, there will be $\underline{3.(B/N).S.H}$ ACK messages sent.
- For an ideal propagation, the sink node will send $\underline{S}$ acknowledgment messages when receiving all the S data messages.

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 $R^2.V.\pi$ nodes. So, all the data message sent by the normal-nodes will generate $\underline{(A/N).S.R^2.V.\pi}$ 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 $(x.R)^2.V.\pi$ nodes. So, all the data message sent by the super-nodes will generate $\underline{(B/N).S.(x.R)^2.V.\pi}$ reception.
- There will be an $(A/N).H.S$ data message relayed by normal-nodes. These messages will generate $\underline{(A/N).H.S.R^2.V.\pi}$ reception.
- There will be an $(B/N).H.S$ data message relayed by super-nodes. These messages will generate $\underline{(B/N).H.S.(x.R)^2.V.\pi}$ reception.
- On the worst case, the $3.(B/N).S.H$ ACK message sent will generate $\underline{3.(B/N).S.H}$ reception.
- There will be an S acknowledgment messages sent by the sink node. These messages will generate $\underline{S.(x.R)^2.V.\pi}$ reception.

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

$$\begin{aligned} Data\_Collection &= (1+H).(S+(A/N).S.R^2.V.\pi) \\ &+ (1+H).(B/N).S.(x.R)^2.V.\pi) \\ &+ S.(1+(x.R)^2.V.\pi) \\ &+ 6.(B/N).S.H \end{aligned} \tag{2}$$

5.1.4 TRIF: Cost of the Data Collection

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

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 $\underline{(A/N).H.S}$ data message relayed by normal-nodes.
- There will be $(B/N).H.S$ 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 $\underline{x.(B/N).H.S}$.
- There will be $\underline{S}$ acknowledgment messages sent by the sink node.

Received messages

- The $(A/N).S$ data message sent by normal-nodes will generate in the network $\underline{(A/N).S.R^2.V.\pi}$ reception.
- Each of the $(B/N).S$ super-nodes which generate a data message will sends x messages which will generate $\underline{(B/N).S.\sum_{i=1}^{x}(i.R)^2.V.\pi}$ total reception.
- The $(A/N).H.S$ data message relayed by normal-nodes will generate $\underline{(A/N).H.S.R^2.V.\pi}$ reception.
- Each of the $(B/N).H.S$ super-nodes which should relay a data message will sends x messages which will generate $\underline{(B/N).H.S * \sum_{i=1}^{x}(i.R)^2.V.\pi}$ total reception.
- The S acknowledgment messages sent by the sink node will be received by $\underline{S.V.(x.R)^2.\pi}$.

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

$$\begin{aligned} TRIF = &(A/N).S.(1 + R.V * \pi + H + H.R.V.\pi) \\ &+ (B/N).x.S.(1 + H) \\ &+ (B/N).S.R^2.V.\pi.(x.(x + 1).(2.x + 1)/6) \\ &+ (B/N).S.H.R^2.V.\pi.(x.(x + 1).(2.x + 1)/6) \\ &+ S.(1 + V.(x.R)^2.\pi) \end{aligned} \tag{3}$$

5.1.5 Numerical results

Here we fixed the number of nodes in the network (N=1000 nodes) and the range of heterogeneous nodes which is equal to 6 times regular range(x=6). We consider a low (V=2) and a high (V=10) density networks: the normal nodes have an average of 6 and 30 neighbors respectively. We evaluate the number of messages sent and received.

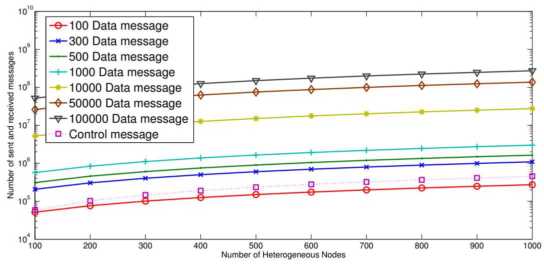
- <u>AsymRP: Neighbor Discovery vs. Data Collection Phases</u>

  Figure 5(a) and figure 5(b) represent the number of total sent and received messages for a low and a high density networks, respectively.

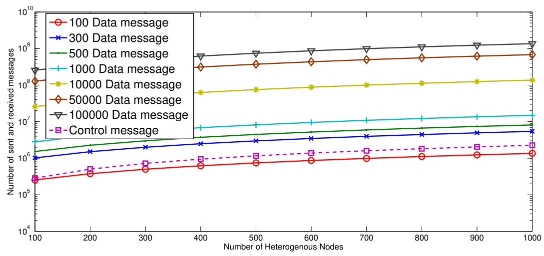
  We note that whatever the density in the network, when the number of data messages exceeds 1/3 of total nodes in the network, the cost of the neighbor discovery phase is covered by the data collection phase. By increasing the number of data messages, the cost of the neighborhood discovery phase in a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

- <u>AsymRP and TRIF energy consumption comparison</u>

  Here we compared the amount of sent and received message for AsymRP and TRIF when used on a high and a low density networks. Figure 6(a)

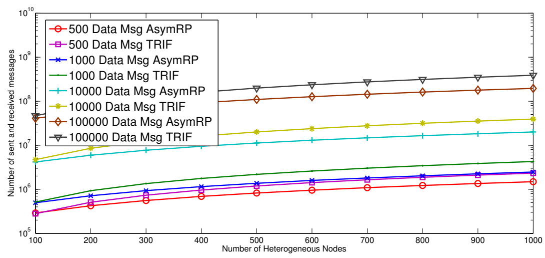


(a) Low Density Network

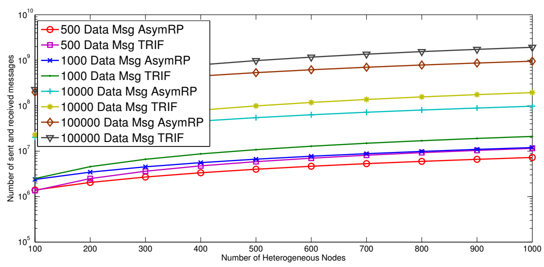


(b) High Density Network

Figure 5: AsymRP: Total number of message sent and received for neighbor discovery and data collection phases



(a) Low Density Network



(b) High Density Network

Figure 6: AsymRP vs. TRIF: Total number of message sent and received

and figure 6(b) respectively represent the total number of message sent and received in a low and in a high density networks. In both cases, AsymRP uses less messages than TRIF and hence consumes less energy. We see also that the difference between the two curves representing TRIF and AsymRP increases when we increase the number of data message generated in the network. AsymRP consumes less energy than TRIF because when a super-node sends one data message in addition it generates one message sent and $\underline{V.(x.R)^2\pi}$ receptions with AsymRP. Whereas with TRIF, when a super-node sends one data message in addition, it generates $\underline{x}$ messages sent and $\underline{\sum_{i=1}^{x}(i.R)^2.V.\pi}$ receptions.

5.2 Simulation study: Network simulation

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 AsymRP to TRIF [11].

5.2.1 Simulation Parameters

In this part, we consider a grid topology. We select a random number of source nodes in the network which will send independently data message periodically to the sink node. The sink node is placed at the center of the network. Here we consider the asymmetric links caused by the presence of different transmission range in the network. Hence, we assume that there are three kinds of nodes are deployed: normal-nodes having a regular transmission range level and super-nodes having a transmission range level equals to three and six times the transmission range level of normal-nodes. The percentage of total super-nodes varies from 10% to 50%. Table 1 summarizes the main characteristics of the network.

Parameter	Value
Sensor Nodes	120
Node range	1x, 3x and 6x regular range
Number of source nodes	1 .. 50
Number of packet sent	1 packet / minute / source node
Propagation	Two ray ground
MAC Protocol	802.15.4 (CSMA)
Confidence Interval	95%
Simulator	WSNet [20]

Table 1: Simulation Parameters

5.2.2 Duplication ratio evaluation

Figure 7 represents the amount of duplicated data message received on the sink node for AsymRP, TRIF using ACK sent by the sink node and TRIF without ACK for 50 source nodes in three scenarios: with 10%, 30% and 50% super-nodes deployed randomly in a grid topology. In figure 7, we see that, in all cases, the amount of duplicated received packet with our proposal AsymRP is less then the amount of duplicated received packet with the two variants of TRIF. We can also see on figure 7 that, when increasing the number of super-nodes with the two variants of TRIF, the average of the duplication ratio increases because each super-node repeatedly sends the message with decreasing transmission range. Whereas with our proposal AsymRP, the duplication ratio remains very low and constant (arround 10%) compared to TRIF because with AsymRP, each source node sends only one message using its transmission range.

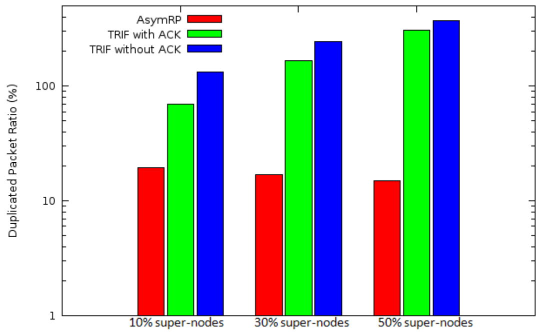


Figure 7: AsymRP vs TRIF: Duplicated received packet Ratio.

5.2.3 Delivery ratio evaluation

In figure 8, we represent the delivery ratio for each source node indexed by their rank. We verified by simulation that on average both our proposal AsymRP and TRIF provide a high delivery ratio between 90% and 100%. But when we evaluate the delivery ratio for each source node's rank, we see that AsymRP outperform TRIF. Indeed, more the path is long more the delivery ratio decreases. Since TRIF does not use asymmetric links, the delivery ratio for TRIF protocol is equal to 0% for further source nodes (with rank equal to 7 and 8 in figure 8). Moreover, we note in figure 8) that when we increase the number of asymmetric links in the network (by increasing the number of super-nodes) the delivery ratio of AsymRP increases: with AsymRP, we use the asymmetric links which will reduce the number of hops. We check this propriety in the section below.

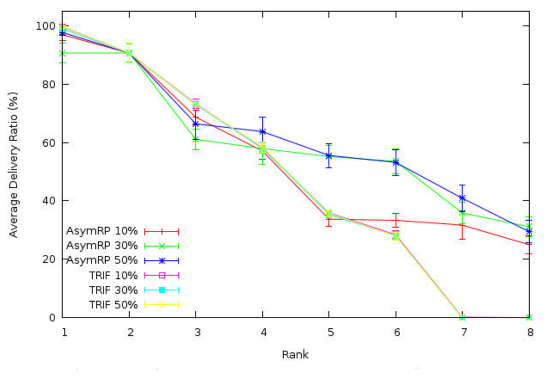
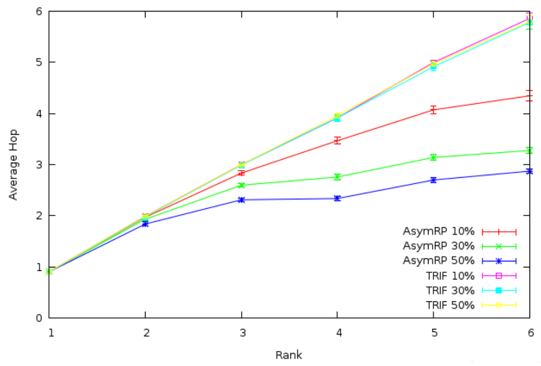


Figure 8: AsymRP vs TRIF: Delivery Ratio

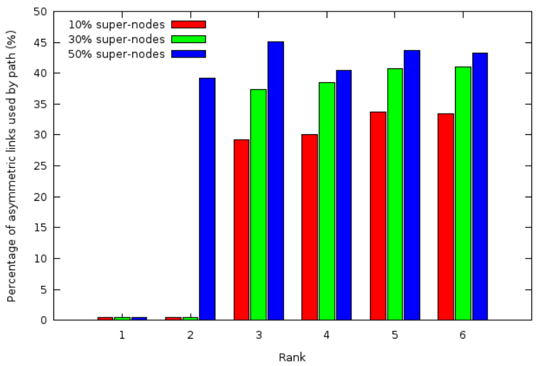
5.2.4 Comparison of the number of hops performed

Finally, we evaluate the average hop count that a packet makes to reach the final destination. Figure 9(a) represents the average hop count using TRIF and AsymRP proposal for each rank. As we can see in Figure 9(a), AsymRP offers a lower hop count when compared to the TRIF protocol. This is due to the fact that in AsymRP, we exploit the asymmetric links when gathering the data from sensors nodes to sink node. Indeed, with AsymRP, the packet can be transmitted using an asymmetric link so the number of hops to reach the destination is less than TRIF because with TRIF, packets only use symmetric links. We can see also, that when we increase the number of super-nodes, the number of hops decreases with AsymRP. This decrease in the number of hops is related to the appearance of more longer links when we increase the number of super-nodes in the network. In Figure 9(b), we can check the percentage of asymmetric links used with our proposal AsymRP. Indeed, the Percentage of asymmetric links used by AsymRP for each path vary between 30% and 45%. When we increase the number of asymmetric links (by increasing the number of super-nodes), the percentage of asymmetric links used by AsymRP also increases.

Thus we check that our proposal exploit the asymmetric links to ensure a high dilivery ratio with a low hop count and a low packet duplication ratio.



(a) Average hop count



(b) Percentage of asymmetric links used

Figure 9: AsymRP vs TRIF: Number of hops performed.

6 Discussions

In this section, we present some discussions on possible extensions of AsymRP. We look at the problem represented as in the figure 10 where the network is connected but no route is found to send an explicit ACK.

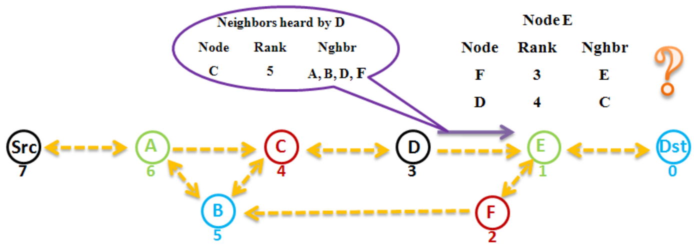


Figure 10: Problem of delivry of data message with AsymRP.

Here we propose two extensions of our proposal AsymRP to address this problem. In the first extension, the candidate relay node (node E in the example in figure 10) should look for a path to send an explicit ACK to the source node (node D in the example in figure 10). In the second extension, the source node, when it did not hear its message relayed, will initiate a recursive k-hop neighbor discovery until the candidate relay node find a common or an intermediate node to which it can send its explicit ACK.

- The idea in the first extension is to use a technique based on a source routing acknowledgment. Each candidate node, when it can not find a common or an intermediate neighbor which can forward an explicit ACK to the source node, initiates a source routing discovery to send its explicit ACK. To avoid flooding of route discovery, we propose to use optimization techniques as in [21] or [22]. An example of this extension based on a source routing acknowledgment is shown in figure 11. Figure 11 shows how the candidate node E initiates a path search to send an explicit ACK to node D. The source routing request traverses the path F-B-C until it reaches the final destination D.

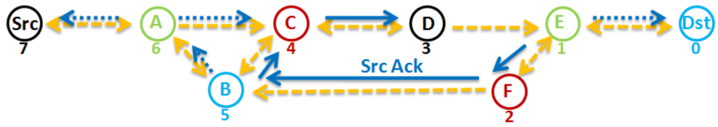
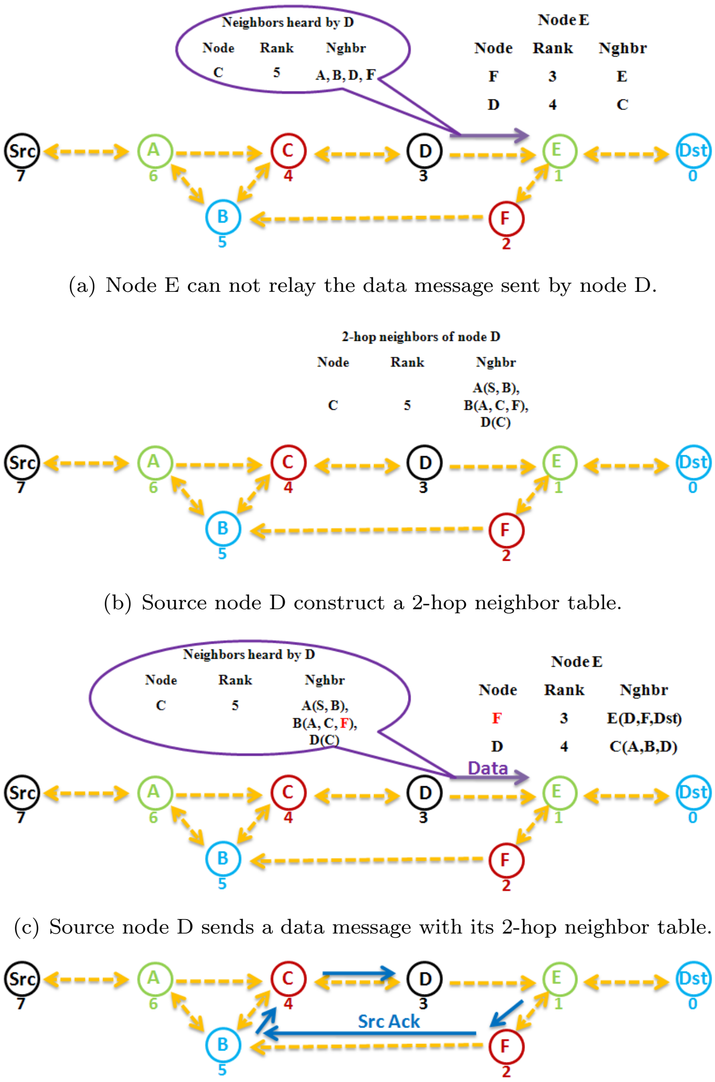


Figure 11: First extension: Source routing ACK.

- The second extension of AsymRP proposes that the source node initiates a recursive k-hop neighbor discovery, sends its data message with it k-hop neighbor table until its message will be relayed by a candidate node. An example of this extension is shown in figure 12. In figure 12(a), the source node D sends the data message, but the only candidate node E can not forward this data message because it can not find a common or an intermediate neighbor. After the `timeout_relayed` elapses, the source node concludes that it can not receive an explicit ACK from potential candidates to relay that message. So the source node D starts a 2-hop neighbor discovery to construct a 2-hop neighbor table as represented in figure 12(b). At the end of this neighborhood discovery the source node D sends again its data message with its new 2-hop neighbor table (as in figure 12(c)). By receiving this data message, the candidate node E deduces that it can send an explicit ACK via node F and this ACK should follow the path F-B-C-D as in figure 12(d).



(a) Node E can not relay the data message sent by node D.

(b) Source node D construct a 2-hop neighbor table.

(c) Source node D sends a data message with its 2-hop neighbor table.

(d) Node E can send an explicit ACK to node D.

Figure 12: Second extension: k-hop neighbor discovery.

These two extensions help to ensure the exploitation of asymmetric links for a convergecast routing. They improve specially data delivery. But there is a compromise between the additional cost incurred by the source routing discovery and the number of k-hop to discover by the source node. In addition, the first extension requires the use of optimization technique to reduce the overhead of the source routing request propagation to send the explicit ACK. The second extension may have an increased delay before delivery of data messages. Indeed, the discovery of recursive k-hop neighbor introduces a delay before the data message will be relayed by the next hop. Thus, the end-to-end delay with that extension will increase compared to that of the basic proposal and to that of the first extension.

7 Conclusion

In this report, we proposed a data collection algorithm for networks where the are asymmetric links caused by the presence of different transmission range. Our proposal, AsymRP, 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 static network may be insignificant compared to the cost of sending periodic data to the sink node. Hence, we compared the energy consumption of AsymRP and TRIF by calculating the amount of sent and received messages. It was shown that AsymRP consumes less energy than TRIF when the number of data message exceeds the 1/3 of total nodes in the network. We are working on evaluating the cost of the overhearing and also on taking into account other metrics to calculate the timeouts introduced by AsymRP 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. We will also evaluate the performance of the two extensions of our Proposals based on the source routing ACK and the recursive neighbor discovery.

References

[1] Gang Zhou, Tian He, Sudha Krishnamurthy, and John A. Stankovic. Impact of radio irregularity on wireless sensor networks. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, MobiSys '04, pages 125–138, 2004.

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

[4] Vikas Kawadia and P.R. Kumar. Power control and clustering in ad hoc networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pages 459 – 469 vol.1, march-3 april 2003.

[5] 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.

[6] S. Nesargi and R. Prakash. A tunneling approach to routing with unidirectional links in mobile ad-hoc networks. In *Computer Communications and Networks, 2000. Proceedings. Ninth International Conference on*, pages 522 –527, 2000.

[7] V. Shah and S. Krishnamurthy. Handling asymmetry in power heterogeneous ad hoc networks: A cross layer approach. In *Distributed Computing Systems, 2005. ICDCS 2005. Proceedings. 25th IEEE International Conference on*, pages 749 –759, june 2005.

[8] Mahesh K. Marina and Samir R. Das. Routing performance in the presence of unidirectional links in multihop wireless networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '02, pages 12–23, 2002.

[9] Gang Zhou, Tian He, Sudha Krishnamurthy, and John A. Stankovic. Models and solutions for radio irregularity in wireless sensor networks. *ACM Trans. Sen. Netw.*, 2:221–262, May 2006.

[10] 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.

[11] 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.

[12] Lifeng Sang, Anish Arora, and Hongwei Zhang. On exploiting asymmetric wireless links via one-way estimation. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '07, pages 11–21, 2007.

[13] Bin Bin Chen, Shuai Hao, Mingze Zhang, Mun Choon Chan, and A. L. Ananda. Deal: discover and exploit asymmetric links in dense wireless sensor networks. In *Proceedings of the 6th Annual IEEE communications society conference on Sensor, Mesh and Ad Hoc Communications and Networks*, SECON'09, pages 297–305, 2009.

[14] Karel Heurtefeux and Fabrice Valois. Distributed localization protocol for routing in a noisy wireless sensor network. In *Proceedings of the 2009 Fifth International Conference on Mobile Ad-hoc and Sensor Networks*, MSN '09, pages 223–230, 2009.

[15] Deepak Ganesan, Bhaskar Krishnamachari, Alec Woo, David Culler, Deborah Estrin, and Stephen Wicker. Complex behavior at scale: An experimental study of low-power wireless sensor networks, ucla/csd, technical report, 2002.

[16] Jerry Zhao and Ramesh Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 1–13, 2003.

[17] Kyu-Han Kim and Kang G. Shin. On accurate measurement of link quality in multi-hop wireless mesh networks. In *Proceedings of the 12th annual international conference on Mobile computing and networking*, MobiCom '06, pages 38–49, 2006.

[18] 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.

[19] 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.

[20] 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.

[21] Wei Peng and Xi-Cheng Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *First Annual Workshop on Mobile and Ad Hoc Networking and Computing. MobiHOC 2000*, pages 129 –130, 2000.

[22] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *Parallel and Distributed Systems, IEEE Transactions on*, 13(1):14 –25, jan 2002.

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