# Analytical Evaluation of Virtual Infrastructures for Data Dissemination in Wireless Sensor Networks with Mobile Sink

Elyes Ben Hamida, Guillaume Chelius

# Analytical Evaluation of Virtual Infrastructures for Data Dissemination in Wireless Sensor Networks with Mobile Sink

Elyes Ben Hamida
ARES INRIA / CITI
INSA Lyon, F-69621, France
elyes.ben-hamida@insa-lyon.fr

Guillaume Chelius
ARES INRIA / CITI
INSA Lyon, F-69621, France
guillaume.chelius@inria.fr

## ABSTRACT

In this paper, we address the problem of data dissemination in wireless sensor networks (WSN) with mobile sink(s). In such a context, the difficulty is for sensor nodes to efficiently track the sink and report the requested data to the sink location. As flat architectures and flooding-based protocols do not scale, overlaying a virtual infrastructure over the physical network has often been investigated as an interesting strategy for an efficient data dissemination in wireless sensor networks. This virtual infrastructure acts as a rendez-vous area for queries and data reports. The main contribution of this paper is to make an analytical comparative study of a variety of virtual infrastructure topologies. The communication cost and the path stretch are evaluated both in the worst and average cases. Finally, existing data dissemination protocols are compared on different applications scenarios.

## Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

## General Terms

Performance

## Keywords

Data Dissemination, Mobile Sink, Virtual Infrastructure, Rendez-vous-based protocols, Performance Analysis

## 1. INTRODUCTION

A Wireless Sensor Network (WSN) is a multi-hop wireless network consisting of a large number of sensors scattered randomly over an area of interest. A sensor node is generally a constrained device with relatively small memory, restricted computation capability, short range wireless transmitter-receiver, and nonrenewable battery energy. Furthermore, sensor networks usually operate on an N-to-1 communication paradigm, where sensors sense the environment and forward the measured data towards a control center. The sensor which generates data reports is called a *source-node*, while the control center is called a *sink*. Applications for wireless sensor networks fall in three major categories [1]: (i) *Periodic sensing*: sensors are always monitoring the physical environment and continuously report measurements to the sink, (ii) *Event driven*: sensors operate in a silent monitoring state and are programmed to notify about events, and (iii) *Query Based*: sensors react to the sinks' queries and return the corresponding data.

In many situations, a static sink may be unfeasible because of deployment or security constraints. Sink mobility may also improve the lifetime of a WSN by avoiding excessive transmission overhead at nodes that are close to the location that would be occupied by the static sink [12, 11]. The sink mobility assumption may be useful for many applications such as target tracking, emergency preparedness, and habitat monitoring. In such a context, the difficulty is for sensor nodes to efficiently track the sink and report the requested data. As flat architectures and flooding-based protocols do not scale, overlaying a virtual infrastructure over the physical network has often been investigated as an interesting strategy for an efficient data dissemination in wireless sensor networks. The mobile sink and sensors can then make use of this infrastructure for routing, data aggregation and data dissemination. This infrastructure acts as a rendez-vous area for queries and data reports.

This paper provides an analytical comparative study on virtual infrastructures. We focus on five main topologies that are used as virtual infrastructures by existing protocols, and we analyse the worst and average distances between the communicating entities (*i.e.,* {*source, rendez-vous-area*}, {*sink, rendez-vous-area*}, and {*rendez-vous-area, sink*}).

Then, the communication cost and the path stretch are analyzed and compared in the worst and average cases. As, there is no "one-fits-all" solution but rather *application-adapted* strategies, the second main contribution of this paper is to compare existing data dissemination protocols on different applications scenarios.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 presents a classification of existing data dissemination protocols. Section 4 describes an analytical study of several virtual infrastructures. The impact of these structures on data dissemination


*SANET'07,* September 10, 2007, Montréal, Québec, Canada.

Table 1: Data dissemination protocols.

| Protocol | Information | Where to? | How? |
|---|---|---|---|
| **GHT [7]** | data | 1 node | Geographic hash table |
| **GHT+SR [7]** | data | 1 out of N nodes | Geographic hash table |
| **TTDD [6]** | data | 1 node | Grid structure |
| **Railroad [9]** | meta-data | 1 out of N nodes | Rail structure |
| **Locators [8]** | sink location | 1 out of N nodes | Geographic hash table |

is discussed in Section 5. Section 6 analyzes the communication cost and path stretch, and existing data dissemination protocols are compared on two different scenarios. Section 7 summarizes the paper's contributions and presents some promising directions for future research.

## 2. RELATED WORKS

Several protocols have been proposed to implement a scalable and energy-efficient data dissemination architecture for WSNs. Directed Diffusion [4] has proposed the concept of data-centric routing. GHT (Geographic Hash Table) [7] has introduced the concept of data-centric storage (DCS). GHT hashes *sensed-data-type* into geographic coordinates and stores the corresponding data at the sensor node (or *home-node*) the closest to these coordinates. To avoid the hot spot problem where queries and sensed data are concentrated on few *home-nodes*, Structured Replication (SR) [7] may be used to distribute the load throughout the network. Shim and Park [8] have proposed a dissemination model where *locators* keep track of the sinks location and reply to queries from the source nodes asking for the sink location. These locators are selected using a deterministic geographic hash function and are replicated uniformly through the whole sensor field. Two-Tier Data Dissemination (TTDD) [6] provides a scalable and efficient data delivery to multiple mobile sinks. Each source pro-actively builds a grid structure by dividing the sensor field into cells with dissemination nodes located at the crossing points of the grid. Queries and data are then transmitted along the grid. Railroad [9] builds and exploits a virtual infrastructure, called a rail. This rail is placed in the middle area of the sensor field so that each node can easily access it. When a source detects a new event, the data remains locally stored and the corresponding meta-data is sent to the rail. This infrastructure is then used by the sink to retrieve meta-data, with the queries traveling around the rail. In HCDD [5], sensor-nodes are self-organized to find a route without the knowledge of node's location information.

## 3. A CLASSIFICATION OF DATA DISSEMINATION PROTOCOLS

Data dissemination protocols can be classified according to several criteria. First, they vary in the disseminated information: (i) ***Data dissemination***: the measured data is disseminated; (ii) ***Meta-data dissemination***: a meta-data is disseminated while the measured data remains stored locally; and (iii) ***Sink location dissemination***: the sink location is stored in the sensor field. When a node detects a new event, it determines the sink's location and the data is then forwarded to this location.

The protocols can also be classified depending on the target of the previous information dissemination: (i) ***a single node***: the disseminated information (data, meta-data, or sink's location) is stored on a particular node usually chosen in a deterministic and/or geographic way; (ii) ***a node out of a group of nodes***: a group of nodes is defined and the information is disseminated towards one node out of this group, generally the closest to the source, thus increasing the lookup cost and decreasing the dissemination one; and (iii) ***a set of nodes***: the information is replicated over a set of nodes, thus decreasing the lookup cost but increasing the dissemination one.

Finally, protocols vary in the virtual infrastructure formed by the set of - potential - storing nodes. It may be a single node, a rail, a grid, *etc.* These virtual infrastructure acts as a rendez-vous area for queries and data report. Table 1 classifies the existing approaches given in section 2 according to these different criteria we have just presented.

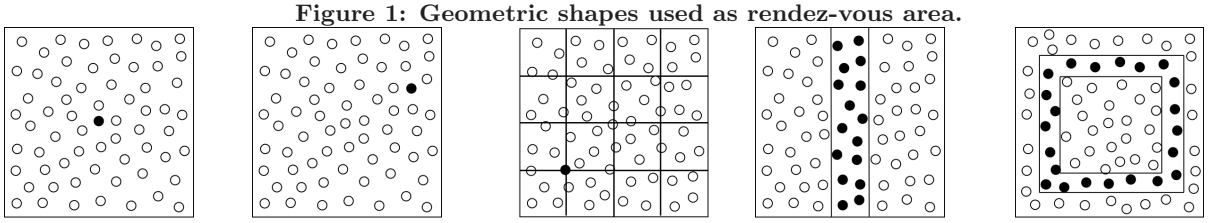
In this paper, we focus on rendez-vous based protocols and particularly on the virtual infrastructure they use. Our main contribution is to provide an analytical study of various virtual infrastructures. Indeed, in Section 4 we investigate five geometric shapes that can be used to build a virtual infrastructure. Then, in Section 6 we analyze the impact of these infrastructures on a protocol communication cost and path stretch in the worst-case and average-case.

## 4. ANALYTICAL STUDY OF VIRTUAL INFRASTRUCTURES

### 4.1 Assumptions

We consider a network with a large number of nodes deployed randomly over a square sensor field of size $1 \times 1$. As in [6, 7, 8, 9], we assume that each node knows its geographic location as well as the network geographic boundaries. Geographic routing is used to forward messages and the rendez-vous area is employed for data dissemination and data lookup. Once a sensor detects a new stimuli, a data report is sent towards the rendez-vous area. To acquire the data, a sink sends queries to the rendez-vous area: the requested data is then sent back to the sink. The communication cost of data dissemination and data collection is proportional to the distance between the different communicating entities (*i.e.,* {*source, rendez-vous area*}, and {*sink, rendez-vous area*}). Indeed, in non sparse networks, the number of hops between two nodes increases linearly with the euclidean distance [2, 3]. Thus, to compare the virtual infrastructures, it is important to evaluate the distances between these entities. For the rest of the paper we define the following metrics: (i) $\mathbb{D}_{src,rdv}$ is the distance between the source-node and the rendez-vous area; (ii) $\mathbb{D}_{sink,rdv}$ is the distance between the sink and the rendez-vous area; and (iii) $\mathbb{D}_{rdv,sink}$ is the distance between the rendez-vous area and the sink. We assume that $\mathbb{D}_{sink,rdv}$ and $\mathbb{D}_{rdv,sink}$ may be different as a query and the data transfer do not necessarily use the same path.

**Figure 1: Geometric shapes used as rendez-vous area.**



(a) Central (b) Random *(GHT [7])* (c) Grid *(TTDD [6])* (d) Line *(LBDD)* (e) Square *(RailRoad [9])*

In what follows, we evaluate these metrics in the worst and average cases for some main virtual infrastructures. Next, in Section 6, the communication cost and the path stretch are deduced and analyzed.

## 4.2 Central rendez-vous area

This rendez-vous area is introduced for comparison between virtual infrastructures. In this scheme, the virtual infrastructure is located in the square center and is composed of one node. The node the closest to the coordinate $(0.5, 0.5)$ acts as a rendez-vous point for data reports and replies to sink queries. This scheme is illustrated on Fig. 1(a).

**Worst-Case.** In the worst-case, a node is located at a distance $\frac{\sqrt{2}}{2}$ of the central rendez-vous node. The worst distance between the different entities is then defined as follows

$$\mathbb{D}_{src,rdv} = \mathbb{D}_{sink,rdv} = \mathbb{D}_{rdv,sink} = \frac{\sqrt{2}}{2} \approx 0.7$$

**Average-Case.** To evaluate the average distance between the communicating entities, we compute the average distance between a randomly chosen point in the unit square and the square center. Let $(X_1, Y_1)$ denote the coordinate of the random point. The distance between this point and the square center can be written as

$$D = \sqrt{|X_1 - 0.5|^2 + |Y_1 - 0.5|^2}$$

Let $Z_1 = |X_1 - 0.5|$ and $Z_2 = |Y_1 - 0.5|$. $Z_1$ and $Z_2$ are independent and have the following *Probability Density Functions* (PDF) [1]:

$$f_{Z_1}(z) = f_{Z_2}(z) = \begin{cases} 2 & 0 < z < 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Using $U_1 = Z_1^2 = (X_1 - 0.5)^2$ and $U_2 = Z_2^2 = (Y_1 - 0.5)^2$, we get the following PDF:

$$f_{U_1}(u) = f_{U_2}(u) = \begin{cases} \frac{1}{\sqrt{u}} & 0 < u < 0.25 \\ 0 & \text{otherwise} \end{cases}$$

Note that if $C = U_1 + U_2 = D^2$ and as $U_1$ and $U_2$ are independent, $f_C(c) = f_{U_1} \circledast f_{U_2}(c)$ (where $\circledast$ denotes the convolution operator). $f_C(c)$ can be written as:

$$f_C(c) = \int_{-\infty}^{+\infty} \frac{1}{\sqrt{x}} \frac{1}{\sqrt{c-x}} dx = [arccos(1 - \frac{2x}{c})]_{-\infty}^{+\infty}$$

[1] A PDF is any function $f(x)$ that describes the probability density in terms of the variable $x$ such that : $\forall x, f(x) \geq 0$ and $\int_{-\infty}^{+\infty} f(x)dx = 1$.

We get the following PDF:

$$f_C(c) = \begin{cases} \pi & 0 < c \leq 0.25 \\ \pi - 2arccos(\frac{1}{2c} - 1) & 0.25 < c \leq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Note that $D = \sqrt{C}$ and therefore $f_D(d) = 2df_C(d^2)$;

$$f_D(d) = \begin{cases} 2d\pi & 0 < d \leq \sqrt{0.25} \\ 2d(\pi - 2\ arccos(\frac{1}{2d^2} - 1)) & \sqrt{0.25} < d \leq \sqrt{0.5} \\ 0 & \text{otherwise} \end{cases}$$

The average distance is finally obtained by a numerical computation of $\mathbb{E}[D] = \int_0^{\sqrt{0.5}} x f_D(x)dx \approx 0.3825$. And the average distance between the different communicating entities is:

$$\mathbb{D}_{src,rdv} = \mathbb{D}_{sink,rdv} = \mathbb{D}_{rdv,sink} \approx 0.3825$$

## 4.3 Random rendez-vous node

In this scheme the rendez-vous area is placed randomly inside the sensor field, as shown on Fig. 1(b). This rendez-vous area is defined deterministically using for example a hash function as in GHT [7]. Once a source detects a new stimuli, a data report is sent to the node the closest to the coordinate $\{x, y\} = hash(data\_type)$. The sink's queries are sent to this rendez-vous point, and the data is forwarded back to the sink.

**Worst-Case.** In the worst-case, a node is located at a distance of $\sqrt{2}$ of the random rendez-vous area and the worst distance between the different entities is:

$$\mathbb{D}_{src,rdv} = \mathbb{D}_{sink,rdv} = \mathbb{D}_{rdv,sink} = \sqrt{2} \approx 1.41$$

**Average-Case.** The average distance between the different entities is evaluated by computation of the average distance between two randomly chosen points in a unit square. Let $(X_1, Y_1)$ and $(X_2, Y_2)$ denote the coordinates of two random points which are selected independently and uniformly. The distance between these two points can be written as $D = \sqrt{|X_1 - X_2|^2 + |Y_1 - Y_2|^2}$.

According to [10], the PDF of the random variable $D$ is defined as follows

$$f_D(d) = \begin{cases} 2\pi d - 8d^2 + 2d^3 & 0 < d \leq 1 \\ 2(\pi - 2)d + 8d\sqrt{d^2 - 1} & \\ -2d^3 - 4d\ arccos(\frac{2-d^2}{d^2}) & 1 < d \leq \sqrt{2} \\ 0 & \text{otherwise} \end{cases}$$

The average distance is obtained by a numerical computation of $\mathbb{E}[D] = \int_0^{\sqrt{2}} x f_D(x)dx \approx 0.52$. Finally, the average

distance between the different communicating entities is:

$$\mathbb{D}_{src,rdv} = \mathbb{D}_{sink,rdv} = \mathbb{D}_{rdv,sink} \approx 0.52$$

## 4.4 Grid

This virtual infrastructure is taken from TTDD [6] and shown on Fig. 1(c). Each source builds pro-actively a grid over the sensor field while the data remains stored locally. The source acts as the rendez-vous point and $\mathbb{D}_{src,rdv} = 0$ in the worst and average cases. To collect data, a sink sends a query to the closest *dissemination-node* (dissemination nodes are the crossing points of the grid). Next, the query travels along the grid in a two-tier way (*i.e.,* X-Y routing) to the source node. The data is then sent back to the sink along the reverse path. Given a grid built over a square sensor field of size $1 \times 1$ with a cell width equals to $c$, the total number of cells is $(\frac{1}{c})^2$ and the total number of crossing points is $(\frac{1}{c}+1)^4$.

**Worst-Case.** In the worst-case and given the X-Y routing scheme, a sink is located at a distance 2 of the source node. The worst distance between the different entities is:

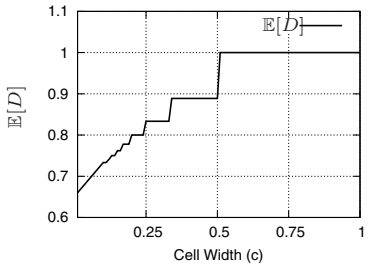
$$\mathbb{D}_{sink,rdv} = \mathbb{D}_{rdv,sink} = 2$$

**Average-Case.** The average distance between a sink and a source node is proportional to the average distance between two randomly chosen crossing points plus the average distance between the sink and the corresponding crossing point. The average distance between two randomly chosen crossing points can be written as

$$\mathbb{E}[D] = \frac{\sum_{i=1}^{(\frac{1}{c}+1)^2} \sum_{j=1}^{(\frac{1}{c}+1)^2} \mathbb{D}_{X,Y}(i,j)}{(\frac{1}{c}+1)^4}$$

where $\mathbb{D}_{X,Y}(i,j)$ is the euclidean distance between two node $i$ and $j$ according to the X-Y routing. This equation is solved numerically and plotted on Fig. 2 for different cell widths. We can notice that $\mathbb{E}[D]$ increases as the cell width $c$ increases. The possible values range from 0.66 to 1.

**Figure 2: Grid infrastructure: average cost.**




Given a cell width $c$, the average distance of a sink taken randomly inside a cell of size $c \times c$ to the nearest crossing point is equal to $\frac{\sqrt{2}}{4}c$. Thus, the average distance between the sink and the source is

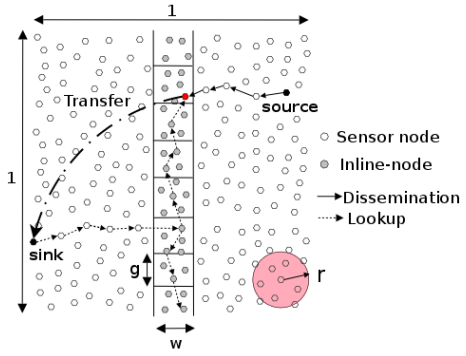
$$\mathbb{D}_{sink,rdv} = \mathbb{D}_{rdv,sink} = \mathbb{E}[D] + \frac{\sqrt{2}}{4}c$$

If we assume a cell width of $c = 0.25$, the average distance between the sink and the source node is $\mathbb{D}_{sink,rdv} = \mathbb{D}_{rdv,sink} \approx 0.88$.

## 4.5 Linear rendez-vous area

In this scheme, the virtual infrastructure is a vertical (or horizontal) virtual *line* of width $w$ which divides the sensor field into two equal parts, as shown on Fig. 1(d) and Fig. 3. This line is also divided into square of length $g$. The parameters $w$ and $g$ are used to address the hotspot problem and the scalability issue. The *line* is placed in the center of the sensor field so that each node can easily access it. Nodes within the boundaries of this wide line are called *inline-nodes*, while the other nodes are referred to as *ordinary nodes*. This *line* acts as a *rendez-vous* point for data storage and lookup. As each node knows its geographic position as well as the network boundaries, the eligibility of nodes as inline-nodes is then easily performed based on the geographic information. For the rest of this paper we will call LBDD (*Line-Based Data Dissemination protocol*) the protocol using this virtual infrastructure.

**Figure 3: The Line-Based Data Dissemination protocol.**




The operation of LBDD is composed of two mains steps: (i) ***Data Dissemination***: when ordinary sensor node generates some data, it forwards the data to the nearest inline-node; and (ii) ***Data Collection***: In order to retrieve a specific data, a sink sends a *query* towards the line in a perpendicular direction. The first inline-node which receives the query propagates it along both directions of the *line*. When the query reaches the inline-node storing the data, the data is forwarded directly to the sink.

To facilitate the data lookup process, two data-storage schemes are possible: the data can be either stored in all nodes of a group or just in the group-leader. The first scheme needs a fine-tuning of $w$ and $g$ to prevent an increase of the congestion under high traffic load conditions, while the second one requires a periodic group-leader election and a replication mechanism. We evaluate in what follows the worst and average distances between the different communicating entities. As in [9], we disregard the line's width $w$ and the group's size $g$.

**Worst-Case.** In the worst-case, a source node is located at a distance 0.5 from the virtual line for data dissemination, and the sink query's covers a distance of $0.5 + 1$ from the sink to the rendez-point (*i.e.,* the *inline-node* storing the requested data). The worst distance between this rendez-vous point and the sink, for sending the requested data, is then $\frac{\sqrt{5}}{2}$ (diagonal of a half square). The distances in the worst-case can then be written as: $\mathbb{D}_{src,rdv} = 0.5$, $\mathbb{D}_{sink,rdv} = 0.5 + 1$, and $\mathbb{D}_{rdv,sink} = \frac{\sqrt{5}}{2} \approx 1.11$.

**Table 2: Average distances: a summary.**

| Rendez-vous area | Dissemination ($\mathbb{D}_{src,rdv_p}$) | Lookup ($\mathbb{D}_{sink,rdv_p}$) | Transfer ($\mathbb{D}_{rdv_p,sink}$) | Protocols |
|---|---|---|---|---|
| **Central** | 0.38 | 0.38 | 0.38 | |
| **Random** | 0.52 | 0.52 | 0.52 | GHT [7] |
| **Linear** | 0.25 | 0.91 | 0.45 | LBDD |
| **Grid (c=0.1)** | 0.0 | 0.77 | 0.77 | TTDD [6] |
| **Grid (c=0.25)** | 0.0 | 0.88 | 0.88 | TTDD [6] |
| **Square (l=0.5)** | 0.13 | 1.11 | 0.46 | |
| **Square (l=0.7)** | 0.09 | 1.46 | 0.51 | RailRoad [9] |

**Average-Case.** Let $(X_1, X_2)$ denote the coordinates of the random source. The distance from this point to the virtual line can be expressed as $D = |X_1 - 0.5|$. It has the following PDF: $f_D(d) = 2$ for $0 < d < 0.5$ and 0 otherwise. The average distance between a random source and the line is

$$\mathbb{D}_{src,rdv} = \mathbb{E}[D] = \int_0^{0.5} x f_D(x) dx = 0.25$$

To compute the average distance from the sink to the rendez-vous point (*i.e.,* the *inline-node* which stores the requested data) we evaluate the average distance between the sink and the virtual line, which is equal to 0.25, plus the average distance covered by the query inside the line during the data lookup process.

Let $Y_1$ and $Y_2$ the Y-coordinates of two randomly chosen points in the unit square. We consider the following random variable: $D = |Y_1 - Y_2|$ with a PDF $f_D(d) = 2(1-d)$ for $0 < d < 1$ and 0 otherwise. The average distance covered by a query inside the virtual line is equal to $\mathbb{E}[D] + \frac{1-\mathbb{E}[D]}{2}$. The random variable $D$ is computed as

$$\mathbb{E}[D] = \int_0^1 x f_D(x) dx = 2[\frac{d^2}{2} - \frac{d^3}{3}]_0^1 = \frac{1}{3}$$

The average distance between the sink and the inline-node is finally equal to:

$$\mathbb{D}_{sink,rdv} = 0.25 + \frac{1}{3} + \frac{1}{3} \approx 0.9166$$

The average distance from the inline-node to the sink for the data transfer is equal to the average distance between two randomly chosen points in a unit square and the central line, respectively. This distance can be expressed as follows: $D = \sqrt{|X_1 - 0.5|^2 + |Y_1 - Y_2|^2}$, with $X_1$, $Y_1$ and $Y_2$ three independent and uniform random variables.

Let $Z_1 = |X_1 - 0.5|$ and $Z_2 = |Y_1 - Y_2|$. $Z_1$ and $Z_2$ are independent and have the following PDF:

$$f_{Z_1}(z) = \begin{cases} 2 & 0 < z < 0.5 \\ 0 & \text{otherwise} \end{cases}$$

$$f_{Z_2}(z) = \begin{cases} 2(1-z) & 0 < z < 1 \\ 0 & \text{otherwise} \end{cases}$$

With $U_1 = Z_1^2 = (X_1 - 0.5)^2$ and $U_2 = Z_2^2 = (Y_1 - Y_2)^2$, we get the following PDFs:

$$f_{U_1}(u) = \begin{cases} \frac{1}{\sqrt{u}} & 0 < u < 0.25 \\ 0 & \text{otherwise} \end{cases}$$

$$f_{U_2}(u) = \begin{cases} \frac{1}{\sqrt{u}} - 1 & 0 < u < 1 \\ 0 & \text{otherwise} \end{cases}$$

Note that if $C = U_1 + U_2 = D^2$ and as $U_1$ and $U_2$ are independent, $f_C(c) = f_{U_1} \circledast f_{U_2}(c)$ (where $\circledast$ denotes the convolution operator). $f_C(c)$ can be written as

$$f_C(c) = \int_{-\infty}^{+\infty} \frac{1}{\sqrt{x}}(\frac{1}{\sqrt{c-x}} - 1) dx = [arccos(1 - \frac{2x}{c}) - 2\sqrt{x}]_{-\infty}^{+\infty}$$

We get the following PDF

$$f_C(c) = \begin{cases} \pi - 2\sqrt{c} & 0 < c \leq 0.25 \\ arccos(1 - \frac{1}{2c}) - 1 & 0.25 < c \leq 1 \\ arccos(1 - \frac{1}{2c}) - arccos(\frac{2}{c} - 1) + & \\ 2\sqrt{c-1} - 1 & 1 < c \leq 1.25 \\ 0 & \text{otherwise} \end{cases}$$

Note that $D = \sqrt{C}$ and therefore $f_D(d) = 2d f_C(d^2)$;

$$f_D(d) = \begin{cases} 2d(\pi - 2d) & 0 < d \leq \sqrt{0.25} \\ 2d(arccos(1 - \frac{1}{2d^2}) - 1) & \sqrt{0.25} < d \leq 1 \\ 2d(arccos(1 - \frac{1}{2d^2}) & \\ -arccos(\frac{2}{d^2} - 1) & \\ +2\sqrt{d^2 - 1} - 1) & 1 < d \leq \sqrt{1.25} \\ 0 & \text{otherwise} \end{cases}$$

The average distance is evaluated numerically using a Riemann sum and is approximated as $\mathbb{D}_{rdv,sink} = \mathbb{E}[D] \approx 0.45$.

# 5. IMPACT OF AVERAGE PATH DISTANCES ON DATA DISSEMINATION

A summary of the average path distances computed in Section 4 is depicted on Tab. 2. To complete our analytical analysis, we performed monte-carlo simulations to evaluate the average path distances of the square virtual infrastructure (Fig. 1(e)). This type of virtual infrastructure is mainly used by RailRoad [9] with a square width of $l = 0.7$. This square structure was simulated using matlab and results were averaged over $10^9$ runs. At each run, the average distances between the different communicating entities (*i.e.,* $\mathbb{D}_{src,rdv}$, $\mathbb{D}_{sink,rdv}$, and $\mathbb{D}_{rdv,sink}$) are computed.

From the results summarized on Tab. 2, we can make two observations. First, we notice that the central and random rendez-vous area present a lower distance for data lookup than the other infrastructures. As the virtual infrastructure is limited to a single node, we don't need to search for the requested data. This is why we get a low distance for the data lookup from the sink to the rendez-vous node. However, this characteristic may induce a hotspot problem and causes congestion.

Second, the use of a large virtual infrastructure like a line, a grid or a square decreases the cost of dissemination (*i.e.,* $\mathbb{D}_{src,rdv}$) compared to the random and central rendez-vous area. Indeed, as the size of the rendez-vous area increases, source nodes get usually closer to this infrastructure. In

addition, using a large infrastructure allows to distribute the communication load through the nodes belonging to the rendez-vous area. However, the use of large infrastructures for data dissemination induces a higher distance for data lookup ($\mathbb{D}_{sink,rdv}$).

From this analysis a tradeoff emerges between the virtual infrastructure's size and the data dissemination/lookup path distances. Thereby there is no "one-fits-all" solution but rather application-adapted strategies. Large virtual infrastructures are more suitable to applications inducing a large number of data reports compared to the number of queries (*e.g., periodic sensing* or *event driven* applications), while small infrastructures surpass the latter in scenarios with a large number of queries compared to the number of data reports (*e.g., query based* applications). This fact is further analyzed in the following Section.

# 6. PERFORMANCE EVALUATION

This section provides a worst-case and average-case communication cost analysis. The path stretch is also investigated for the different virtual infrastructures. We first present the models and assumptions, then we evaluate the communication cost and the path stretch.

## 6.1 Models and assumptions

We consider a network made of nodes dispatched in a square field of size $1\times1$. We assume that nodes are uniformly and independently distributed in the region. The network is modeled by a stationary two-dimensional Poisson point process of constant spatial intensity $\lambda$. We define $H(l)$ as the number of hops on a path between two arbitrary nodes $x$ and $y$ such that $|x,y| = l$ is the euclidean distance of the path between the two nodes. According to [2, 3], given a geographic routing protocol, we have $H(l) = \zeta \frac{l}{r}$ with $r$ the communication range and $\zeta \geq 1$ a scaling factor depending on the spatial node density $\lambda$. For numerical applications, we will assume that: $\zeta = 1$.

According to the analysis proposed in [9], we consider four types of messages: event notification, query, data, and control messages, whose sizes are $p_e$, $p_q$, $p_d$, and $p_c$, respectively. We suppose that $p_e = p_c = p_q$ and that $p_d = 2 \times p_q$. There are $m$ sinks moving randomly in the sensor field as well as $n$ sources. Each sink generates an average number of queries equal to $\bar{q}$ and each source generates an average number of events equal to $\bar{e}$. Thus, the total expected number of queries and events can be written as $m\bar{q}$ and $n\bar{e}$.

## 6.2 Communication Cost

The communication cost represents the total amount of messages generated in the network during the data dissemination process and lookup. It is defined as: $C_{\text{protocol}} = C_{\text{DD}} + C_{\text{DL}} + C_{\text{DT}}$, where $C_{\text{DD}}$, $C_{\text{DL}}$, $C_{\text{DT}}$ are the costs of data dissemination, data lookup, and data transfer, respectively. The subscript "protocol" refers here to one of the solutions listed in Table 1.

In what follows, we compare the LBDD, GHT, TTDD and RailRoad protocols which correspond respectively to the linear, random, grid, and square (with $l = 0.7$) rendez-vous area.

### 6.2.1 Worst-case communication cost

**LBDD.** In the case of LBDD, upon the detection of a new event, the sensor node sends the measured data towards the line. In the worst case, this message meets about $H(\frac{1}{2})$ nodes. To retrieve the data, a mobile sink sends a query message which is forwarded greedily towards the line. This message is then propagated along the line until it is received by the corresponding inline-node. In the worst case, the query meets about $H(\frac{1}{2}+1)$ nodes. Then, the data is transferred from the inline-node to the sink, and meets in the worst-case $H(\sqrt{5}/2)$ nodes (diagonal of a half square). To avoid the transfer of duplicated data, we suppose that a sink receives a response to its query only if the *inline-node* owns a new data. The total communication cost of LBDD in the worst case is then

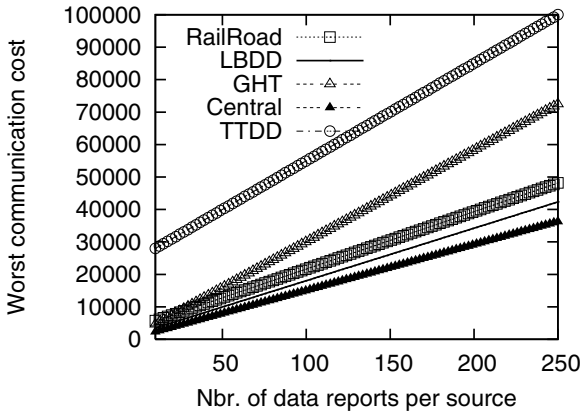
$$C_{\text{LBDD}} = (n\bar{e}\,p_d + m\bar{q}\,p_q)\,H\,(1/2) + H\,(1)\;m\bar{q}\,p_q + H\left(\sqrt{5}/2\right)\;n\bar{e}\,p_d. \quad (1)$$

**Central rendez-vous area.** The communication cost of the central rendez-vous area is evaluated in a similar way according to the worst distances computed in Section 4. This cost can be written as

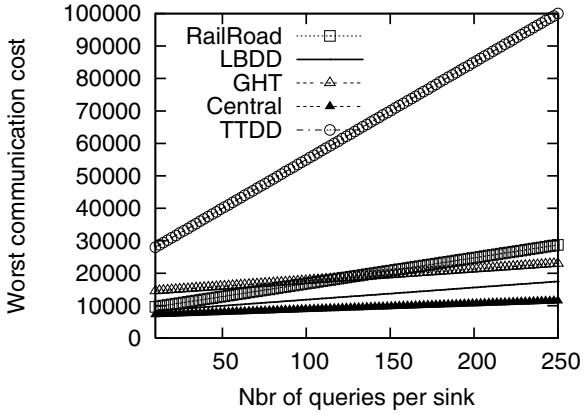
$$C_{\text{Central}} = (2\,n\bar{e}\,p_d + m\bar{q}\,p_q)\,H(\frac{\sqrt{2}}{2});$$

**GHT, Railroad, and TTDD.** The total communication cost of GHT, Railroad, and TTDD are computed in a similar

**Figure 4: Worst-case Communication cost ($m = 5$ sinks, $n = 10$ sources).**

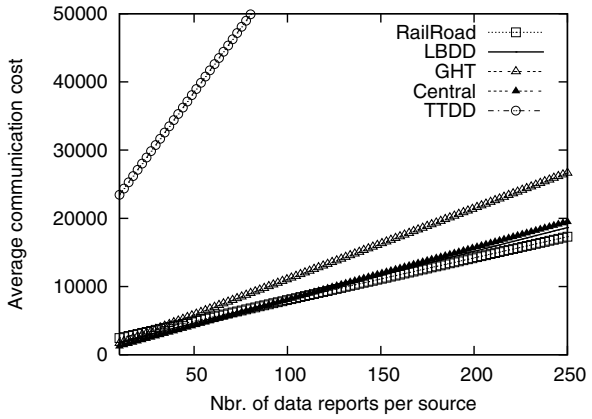



(a) $\bar{q} = 50$

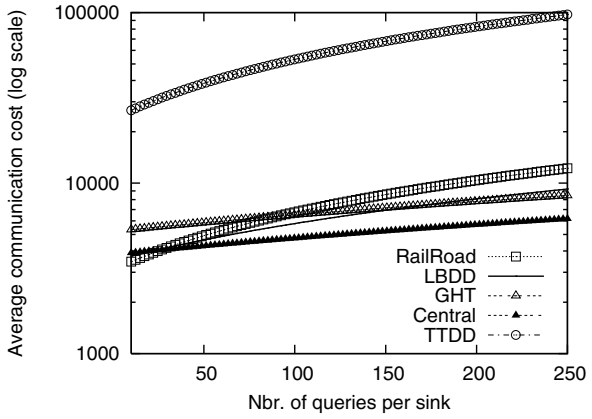



(b) $\bar{e} = 50$

**Figure 5: Average-case Communication cost ($m = 5$ sinks, $n = 10$ sources).**




(a) $\bar{q} = 50$




(b) $\bar{e} = 50$

way (for further details, refer to [9]):

$$C_{\text{GHT}} = (2\, n\bar{e}\, p_d + m\bar{q}\, p_q)\, H(\sqrt{2});$$

$$C_{\text{Railroad}} = [n\bar{e}(p_e + p_q + 4p_d) + m\bar{q}p_q]\; H\left(\sqrt{2}/4\right) + m\bar{q}\, H(2\sqrt{2})\, p_q; \quad (2)$$

$$C_{\text{TTDD}} = n\frac{4\lambda}{H(\frac{1}{c})}p_c + m\bar{q}\left[\lambda c^2 + H(2)\right] p_q + n\bar{e}\left[H(2) + H(\sqrt{2}/(2c))\right] p_d. \quad (3)$$

On Fig. 4 we compare the worst-case communication costs of all approaches for two scenarios. We consider $10,000$ sensor nodes deployed on a square sensor field of size $1 \times 1$. The sensor coverage area radius is $r = 0.1$ and we suppose that $c = 0.3$ (the size of a TTDD cell). The first scenario considers a fixed number of queries per sink ($\bar{q} = 50$) with a varying number of data reports per source node. The results for the first scenario are shown in Fig. 4(a). In the second scenario, we consider a fixed number of data reports per source ($\bar{e} = 50$) for a varying number of queries per sink. Results for the second scenario are shown in Fig. 4(b).

We notice on both scenarios that TTDD presents a rather high communication cost stemming from its need to build grids and its routing strategy along the grid. Moreover, as previously analysed in Section 5, we observe that scenarios with a high number of data reports are most suitable to protocols implementing a large virtual infrastructure like Railroad and LBDD. The reason is that the infrastructure reduces the communication length and thus cost between the source and the node storing the disseminated information. On the other side, scenarios with a large number of queries are more suitable to protocols like GHT and LBDD which propose a low lookup cost. Finally, as expected the central rendez-vous area present a lower communication cost in both scenario.

### 6.2.2 *Average-case communication cost*

Similarly to the worst-case study and according to the average distances computed in Section 4, we evaluate the average communication costs of the latter protocols. On Fig. 5 we compare the average-case communication costs of all approaches for two scenarios with the same parameters as defined in Section 6.2.1.

As expected TTDD presents on both scenarios a high average communication cost. We can notice on the first scenario (Fig. 5(a)) that GHT presents a high cost compared to the other protocols except TTDD. In addition the communication costs of the central rendez-vous area, LBDD and RailRoad are very close. However, we notice that for a high number of data reports per source node ($\geq$ 200 data reports per source), RailRoad performs slightly better than the others virtual infrastructures. As RailRoad implements a large infrastructure (*i.e.,* a square of width $l = 0.7$), sensor-nodes are closer to the rendez-vous area and the cost of data dissemination is reduced. However, on the second scenario (Fig. 5(a)) RailRoad presents a high communication cost compared to the other protocols. This scenario presents a large number of queries, and is more suitable to protocols like the central and random rendez-vous area. Finally, LBDD provides the best tradeoff among the evaluated approaches, leading to low communication costs in both scenarios.

## 6.3 Average Path Stretch

In this section, we analyze the impact of using a virtual infrastructure on the average path stretch. Let $\mathbb{S}$ be the path stretch. It is defined as the number of physical hops required for data dissemination, lookup and transfer using a virtual infrastructure over the number of hops on the path between the source and the sink without using any intermediate rendez-vous area. If we suppose that the sink knows the source's position, a sink's query travels $H(0.52)$ hops to reach to the source-node as the average distance between two randomly chosen points in a unit square is 0.52. The data is then sent back to the sink with an average number of hops also equal to $H(0.52)$. The total number of hops required for the data dissemination/lookup/transfer is thus $H(2 * 0.52)$. For the Central rendez-vous area, the total number of hops is $H(3 * 0.38)$: $H(0.38)$ hops from the source to the rendez-vous node, $H(0.38)$ hops from the sink to the rendez-vous node, and $H(0.38)$ hops from the rendez-vous node to the sink. Thus, the path stretch is $\mathbb{S} = \frac{H(3*0.38)}{H(2*0.52)} = 1.09$.

According to the average distances evaluated in Section 4, we compute in a similar way the path stretch of LBDD,

| Table 3: Average path stretch. | |
|---|---|
| **Rendez-vous area** | $\mathbb{S}$ |
| **Central** | 1.09 |
| **Linear [LBDD]** | 1.23 |
| **Random [GHT]** | 1.5 |
| **Grid (c=0.1) [TTDD]** | 1.48 |
| **Grid (c=0.25) [TTDD]** | 1.69 |
| **Square (l=0.7) [RailRoad]** | 1.98 |

GHT, RailRoad and TTDD. The results are shown in Tab. 3. We can make two observations. First, the use of a virtual infrastructure as a rendez-vous area for data dissemination/lookup increases the path stretch in comparison to a direct communication between a sink and a source-node. Second, we notice that the path stretch generally increases as the virtual infrastructure's size increases. Indeed, RailRoad presents a rather high path-stretch which is almost twice the optimal path. This is a direct consequence of the query path length. On the other side, the central rendez-vous area presents a low path-stretch which is close to the optimal path. Finally, LBDD presents the best tradeoff among the evaluated protocols.

# 7. CONCLUSIONS

In this paper we have presented an analytical comparison of several virtual infrastructures for data dissemination in wireless sensor networks. These infrastructures act as a rendez-vous area for the data reports and queries and can be leveraged by efficient and scalable protocols. Through the study of the different approaches, we have highlighted two tradeoffs. The first one is that if on one hand, the use of a large virtual infrastructure - such as railroad - reduces the dissemination cost, on the other hand it increases the data lookup and collection costs as well as the path stretch.

A second tradeoff is that the use of a small virtual infrastructures may reduce the energy cost of data dissemination and collection but it may also reduce the protocol reliability and robustness as it concentrates the traffic over a small structure, inducing congestion and premature death of nodes (overused). A solution for these tradeoffs is to adapt the protocol and its structure to the sensor network application and parameters. As shown by the performance evaluation, there is no "one-fits-all" solution but rather application-adapted strategies. It depends for example on the data report and data query frequencies. These two tradeoffs were also analyzed via realistic simulations considering realistic radio communications as well as the infrastructures' parameters such as the line/cell widths (*e.g.,* $w$ or $g$). Due to paper length limitations, we have chosen to let these results for a more complete paper. In the future we plan to extend our performance analysis of data dissemination protocols under irregular topologies and an inhomogeneous node density. We also plan to better characterize the congestion versus communication cost tradeoff in order to study the impact of the virtual infrastructures on the network lifetime.

# 8. REFERENCES


[1] A. Boukerche, R. W. N. Pazzi, and R. B. Araujo. A fast and reliable protocol for wireless sensor networks in critical conditions monitoring applications. *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems (MSWiM '04)*, pages 157–164, 2004.

[2] A. Busson, G. Chelius, and E. Fleury. From euclidian to hop distance in multi-hop radio networks: a discrete approach. Research Report RR-5505, INRIA, January 2005.

[3] S. De, A. Caruso, T. Chaira, and S. Chessa. Bounds on hop distance in greedy routing approach in wireless ad hoc networks. *International Journal of Wireless and Mobile Computing*, 1(2):131–140, 2006.

[4] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. *Proceedings of the sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00)*, Boston, Massachusetts, August 2000.

[5] C.-J. Lin, P.-L. Chou, and C.-F. Chou. Hcdd: hierarchical cluster-based data dissemination in wireless sensor networks with mobile sink. In *IWCMC '06: Proceeding of the 2006 international conference on Communications and mobile computing*, pages 1189–1194, 2006.

[6] H. Luo, F. Ye, J. Cheng, S. Lu, and L. Zhang. TTDD: Two-tier data dissemination in large-scale wireless sensor networks. *ACM Journal of Mobile Networks and Applications (MONET), Special Issue on ACM MOBICOM (2003)*, 2003.

[7] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu. Ght: A geographic hash table for data-centric storage in sensornets. *In Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, September 2002.

[8] G. Shim and D. Park. Locators of mobile sinks for wireless sensor networks. *Proceedings of the 2006 International Conference on Parallel Processing Workshops (ICPPW'06)*, pages 159–164, 2006.

[9] J. H. Shin, J. Kim, K. Park, and D. Park. Railroad: virtual infrastructure for data dissemination in wireless sensor networks. *Proceedings of the 2nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN'05)*, pages 168–174, Oct. 2005.

[10] E. Uysal-Biyikoglu and A. Keshavarzian. Throughput achievable with no relaying in a mobile interference network. In *ISCC '03: Proceedings of the Eighth IEEE International Symposium on Computers and Communications*, page 641, Washington, DC, USA, 2003. IEEE Computer Society.

[11] W. Wang, V. Srinivasan, and K.-C. Chua. Using mobile relays to prolong the lifetime of wireless sensor networks. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 270–283, New York, NY, USA, 2005. ACM Press.

[12] Z. M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli. Exploiting sink mobility for maximizing sensor networks lifetime. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS '05) - Track 9*, 09:287.1, 2005.