



HAL
open science

Moving Beyond Testbeds? Lessons (We) Learned about Connectivity

Keoma Brun-Laguna, Pedro Henrique Gomes, Thomas Watteyne, Pascale Minet

► **To cite this version:**

Keoma Brun-Laguna, Pedro Henrique Gomes, Thomas Watteyne, Pascale Minet. Moving Beyond Testbeds? Lessons (We) Learned about Connectivity. IEEE Pervasive Computing, 2018. hal-01968647

HAL Id: hal-01968647

<https://inria.hal.science/hal-01968647>

Submitted on 2 Jan 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Moving Beyond Testbeds? Lessons (We) Learned about Connectivity

How realistic is the connectivity in a testbed? The question determines whether a protocol that performs well in a testbed also works in a real-world environment. We answer this question by gathering 11 datasets, a total of 2,873,156 connectivity measurements, on both testbeds and real-world deployments. We determine that real-world deployments feature some quantifiable level of external interference, multi-path fading and dynamics in the environment. We further show that some testbeds do not have all three components. We develop a 5-point check-list to assess the realism of a testbed deployments and introduce the visual "waterfall plot" tool to instantaneously evaluate the connectivity characteristics of a deployment.

Keoma Brun-Laguna
Inria, France

Pedro Henrique Gomes
Ericsson Research, Brazil &
University of Southern
California

Pascale Minet
Inria, France

Thomas Watteyne
Inria, France

Designing a low-power wireless networking protocol typically involves early back-of-the-envelope analysis, high-level simulation, and experimental evaluation to benchmark its performance. After that, the protocol is considered ready to be deployed in real-world applications. *But is it really?*

How can one be sure the conditions in the testbed were varied enough that the protocol was actually tested in real-world conditions? We focus on connectivity (the characteristics of the wireless links between nodes) of IEEE802.15.4-based low-power wireless networks and want to (1) compare the connectivity between testbeds and real-world deployments, and (2) propose a methodology to verify that the testbed evaluation includes all key connectivity characteristics seen in the real world. The goal of this methodology is to ensure that a protocol that performs well on a testbed also does so when moving beyond testbeds.

The methodology we adopt is the following. We start by gathering a large set of connectivity traces, on both testbeds and real-world deployments we have access to. Then, we extract from the real-world deployments the three main connectivity characteristics: presence of external interference, level of multi-path fading, and amount of dynamics in the environment. In the process, we show that some testbeds do not feature all three characteristics. Finally, we propose a methodology for ensuring testbed results are realistic and describe the associated tools.

This article makes the following contributions:

1. A methodology to collect dense connectivity datasets.
2. Two tools for collecting dense connectivity datasets: Mercator for testbeds, and Sol-System for real-world deployments. Both tools are fully described in this article and the related code is published as open-source.
3. Eleven connectivity datasets available to the research community, from both testbeds and real-world deployments, containing 2,873,156 Packet Delivery Ratio (PDR) measurements gathered over a cumulative 170,037 mote-hours of operation.
4. A check-list to assess the realism of a (testbed) deployment.
5. The visual “waterfall plot” tool to instantaneously evaluate connectivity characteristics.

RELATED WORK

This article focuses on the evaluation of protocol proposals. In particular, on making sure that a protocol which has proven to perform well in a testbed, also does so when used in real-world applications.

New protocol propositions typically are evaluated through analysis, simulation, and experimentation. Experimentation is done mostly on testbeds: permanent instrumented deployments focused entirely on protocol benchmarking. There is no shortage of open-access testbeds, including **Indriya** [1], **IoT-Lab** [2], and **Tutornet** [3]. Typical testbeds consist of between 50 and 400 nodes deployed in an indoor environment, usually a university laboratory. **Tonneau et al.** put together an up-to-date survey of over a dozen open-access testbeds [4].

Since each testbed is different, it is important to understand the connectivity between nodes in a particular site. It is equally important to make sure that this connectivity has the same key characteristics as real-world deployments.

Papadopoulos et al. study the connectivity in the IoT-Lab Strasbourg testbed, and show how the shape/structure of the building, WiFi interference, and time of day impact experimental results [5].

Watteyne et al. perform a similar analysis on IoT-Lab Grenoble, a 350-node testbed deployed in a 65 m × 30 m office building [6]. Each node transmits 100-frame bursts while all others listen and record received frames. This process is repeated for all 16 IEEE802.15.4 channels at 2.4 GHz. The authors quantify multi-path fading and show that WiFi beaconing significantly impacts network performance.

With such variety of testbeds, being able to conduct reproducible experiment becomes important. **Papadopoulos et al.** show that only 16.5% of the studied experimental-based work propose reproducible results [5].

Somewhat more fundamentally, it is of paramount importance to ensure that a solution evaluated on the testbeds “looks like” real-world deployments. If that is not the case, a solution might work perfectly on a testbed, but fail when deployed beyond testbeds.

Zhao et al. did some early work on measuring the connectivity between nodes deployed in realistic environments [7]. They deployed 60 nodes in a building, a forest and a parking lot.

More recently, **Dong et al.** proposed a methodology for collecting data traffic and analyzing the impact of packet delivery ratio in different protocols [8]. The data is collected from a real-world deployment in the wild, with 343 nodes deployed for 10 days. All nodes generate three types of packets every 10 min, containing application raw values, link quality, and routing statistics. Even though the experiment was performed in a large-scale deployment, the network runs on a single channel, and from the results, it is clear that the links were very stable and not influenced by external interference.

Doherty et al. deployed a 44-node low-power wireless network in a 76 m × 69 m industrial printing facility for 26 days [9]. Authors show that the PDR varies over frequency (i.e. because of multi-path fading) and time (i.e. because of dynamics in the environment).

CRAWDAD (crawdad.org) is a community archive that gathers wireless traces, including on connectivity, since 2002 [10]. To this date, the platform has 121 datasets on different application and technologies. For instance, the dataset used in [11] is available. This dataset is the results of the analysis of different IEEE802.15.4 parameters for a network deployed in an indoor environment for 6 months. *As an online addition to this article, the 11 datasets gathered (see Section “Published Datasets”) are available on the CRAWDAD platform.*

With such datasets available, some run simulations on them, i.e. replacing the propagation model at the PHY layer of simulators.

Watteyne et al. analyze multi-channel networks based on frequency hopping [12]. The authors deploy 46 TelosB nodes in a 50 m × 50 m office environment. The results are based on simulations that take into account the connectivity datasets (more precisely, the Packet Delivery Ratio) obtained from a deployment in a working office. Even though the datasets utilized are realistic, the chosen environment is very limited in size, and the results may not be applicable to other scenarios, such as large-scale and/or outdoor deployments.

We make two main observations from surveying related work. First, only very few connectivity traces are gathered on testbeds, and their connectivity is not studied well. Most often, protocols are being evaluated, without really knowing whether the connectivity in the testbed resembles that in the real-world scenarios. Very little is done in related work to show the completeness of the evaluation, i.e. demonstrate that the testbed(s) used for evaluation contains the same connectivity characteristics as real-world deployments. Second, very little has been done to verify that the connectivity in these testbeds resembles real-world deployment connectivity. The impact of this second point is particularly important, as, without it, one cannot really trust that a solution that works on a testbed will also work in a real-world deployment.

DENSE CONNECTIVITY DATASETS

Methodology and Terminology

Our goal is to gather *dense* connectivity datasets and learn lessons from them.

We are interested in the connectivity between the nodes in an IEEE802.15.4-based low-power wireless network, and quantify the “quality” its links by (link-layer) Packet Delivery Ratio (PDR). We operate at 2.4 GHz, the most commonly used IEEE802.15.4 frequency band. The PDR of a link between nodes A and B can be measured as the ratio between the number of link-layer acknowledgments frames received by node A , and the number of link-layer frames sent from node A to node B . A link with PDR = 50% means that, on average, node A has to transmit the same frame twice to node B to receive an acknowledgment and consider the communication successful. We consider the PDR of a link to be a good indicator of the “quality” of a link, and prefer it over other indicators such as the Received Signal Strength Indicator (RSSI), which are related but hardware-dependent.

We call **PDR measurement** the measurement of the PDR (a number between 0% and 100%) between two nodes, at a particular time. A **dataset** consists of all the PDR measurements collected during one experiment.

We want the dataset to be dense along 3 axes: (1) dense in *time*, as we want to analyze the PDR of a link evolving over time, (2) dense in *frequency*, as we want to see the impact of the communication frequency used on the PDR at a particular time, (3) dense in *space*, i.e. collected over as many environments as possible to draw conclusions that apply to various use cases and are relevant.

Datasets are collected on both testbeds and real-world deployments. While the data contained in the datasets are equivalent (and can be compared), the hardware and tools in both cases are different. We hence use two tools (Mercator and SolSystem), both creating equivalent data.

Mercator: Testbed Datasets

The 3 testbeds we use offer the ability to load arbitrary firmware directly on IEEE802.15.4-compliant nodes. These nodes are deployed in a particular location (detailed in Section “Deployments”), and while our firmware executes, we have access to the serial port of each device. This means we are able to (1) receive notifications from the nodes, and (2) send commands to the nodes, without interfering with the radio environment.

We developed Mercator, a combination of firmware and software specifically designed to collect connectivity datasets in testbeds¹. The same firmware runs on each node in the testbed; the software runs on a computer connected to the testbed, and drives the experiment. The firmware allows the software to control the radio of the node, by sending commands to its serial port. The software can send a command to a node to either transmit a frame (specifying the frequency to transmit on), or switch the remote node to receive mode (on a particular frequency). In receive mode, the node issues a notification to the software each time it receives a frame.

All frames are 100 B long, and contain the necessary fields (unique numbers, addressing fields, etc.) to filter out possible IEEE802.15.4 frames sent by nodes outside the experiment.

At the beginning of an experiment, the same firmware is loaded on all nodes. The software is responsible for orchestrating the experiment, which has a pre-set duration. The software starts by having a particular node transmit a burst of 100 frames, on a particular frequency, while all other nodes are listening to that frequency. By computing the portion of frames received, each listening node measures the PDR to the transmitting node, at that time, on that frequency. The PDR ranges from 100% if the node received all frames, and 0% if it received none. The software repeats this over all 16 available frequencies, and all nodes, in a round-robin fashion, until the end of the experiment. The dataset resulting from the experiment contains the PDR measured over all source-destination pairs, all frequencies, and throughout the duration of the experiment.

Mercator has been used on 3 testbeds (see Section “Deployments”), resulting in 5 datasets (see Section “Published Datasets”).

SolSystem: Real-World Deployment Datasets

In real-world deployments, nodes are standalone, and each node’s serial port is *not* connected to testbed infrastructure, so we cannot use Mercator. We instead use network statistics as the primary source of data to create the datasets.

We deploy SmartMesh IP based networks for real-world applications. SmartMesh is the market-leading commercial low-power wireless networking solution, with over 60,000 networks deployed. A SmartMesh IP network offers over 99.999% end-to-end reliability, over a decade of battery lifetime, and certified security [13]. In addition, once it has joined a network, each SmartMesh IP node automatically generates “Health Reports” (HRs) every 15 minutes. The HRs contain networking/node statistics, and allow a network administrator to have a clear view over the health of a network, in real-time.

¹ The Mercator source code is published under a BSD open-source license at github.com/openwsn-berkeley/mercator.

We use the SolSystem back-end² to collect all HRs from 4 real-world deployments (see Section “Deployments”). These built-in SmartMesh IP statistics are equivalent to the information gathered using Mercator.

Note that we had to develop two (equivalent) solutions because the hardware on the testbeds and real-world deployments is different. SmartMesh IP firmware only runs on the LTC5800 chip, which is not present in the testbeds. Similarly, it is technically impossible to run the Mercator firmware on the LTC5800-based real-world deployment nodes.

Deployments

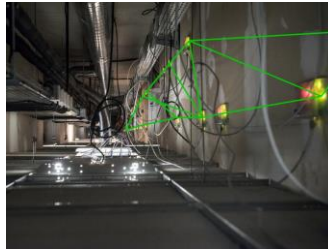
Figure 1 shows pictures of the 7 deployments used to generate the datasets.

IoT-Lab [1] is a 2728-node testbed, deployed across different sites in France. We run Mercator on the Lille, Grenoble and Strasbourg sites. On the Lille site (Fig. 1a), nodes are deployed on the ceiling and walls of a single large, mostly empty, room in an office building. On the Grenoble site, nodes are deployed along four interconnected corridors of an office building, hidden between the dropped ceiling and the roof. On the Strasbourg site (Fig. 1c), nodes are deployed inside a single room in the basement of an office building. In all cases, the distance between adjacent nodes does not exceed 1 m. On each site, we run two types of experiments: an 18 h experiment on 50 nodes, and a multi-day experiment on 5 nodes.

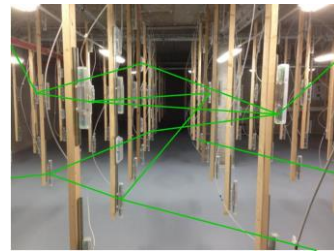
From a hardware/system point of view, the three IoT-Lab deployments are equivalent. We run Mercator on the same hardware (the “IoT-Lab_M3” node - www.iot-lab.info/hardware/m3/), and use the exact same procedure for all experiments on these three sites.



a) [testbed] Lille



b) [testbed] Grenoble



c) [testbed] Strasbourg



d) [real-world] EvaLab and Inria-C



e) [real-world] SmartMarina



f) [real-world] Peach

Figure 1: Pictures of the testbeds and real-world deployments we collect dense connectivity datasets in. Green lines are added to suggest wireless communication between nodes. They show the position of the nodes, but do not per-se represent the exact connectivity collected in the datasets.

² solsystem.io, source code at github.com/realms-team/sol

In the real-world case, we collect the connectivity traces from already-deployed SolSystem networks. Each of these networks has been installed for an end-user application; they were not deployed as part of this article.

The 22-node EvaLab SolSystem deployment is done across a single 40 m × 10 m office building floor. About 200 people work in that building, many of them using WiFi extensively. Nodes are not attached to external sensors, each node reports temperature data every 30 s.

The 19-node SmartMarina SolSystem deployment is done as part of a “smart parking for boats” project at the Cap d’Agde marina, in Southern France (www.smartmarina.org). Nodes are attached to ultrasonic sensors to detect the presence of boats on the different moorings. The network is deployed along a 50-boat pier. WiFi is deployed across the marina and used extensively by boat owners.

The 19-node Peach SolSystem deployment is done as part of a project to predict frost events in fruit orchards (www.savethepeaches.com). Nodes are attached to air and soil temperature/humidity sensors, and deployed on top of 5 m high poles. These poles are installed in 100 m × 50 m peach orchard in Mendoza, Argentina. Each node generates sensor data every 30 s. There is no WiFi connectivity in the orchard.

The 21-node Inria-C SolSystem deployment is done across a single 27 m × 10 m section of an office building floor. About 200 people work in that building, many of them using WiFi extensively. Nodes are not attached to external sensors, each node reports temperature data every 1 s. Unlike all other SolSystem deployments, the Inria-C network is forced to form a star topology (only leaf nodes). This is a requirement for the network to produce the per-frequency statistics we need for Sections “Witnessing Instantaneous Multi-Path Fading” and “Witnessing Dynamics in the Environment”.

Published Datasets

Table 1 lists the 11 datasets produced by the deployments listed in Section “Deployments”. They contain a total of 2,873,156 PDR measurements, gathered over a cumulative 170,037 mote-hours of operation. These datasets are made publicly available as part of this article³, and are one of its main contributions. To the best of our knowledge, they are, to date, the most comprehensive set of multi-frequency connectivity datasets gathered over a wide variety of environments.

dataset	# nodes	duration	# PDR measurements	associated figures
lille_1	5 nodes	15 days	367,293	Figs. 4a
lille_2	50 nodes	18 h	274,392	Figs. 2a, 3a, 6a
grenoble_2	50 nodes	18 h	284,068	Figs. 2b, 3b, 6b
strasbourg_1	5 nodes	3 days	81,900	Figs. 4b
strasbourg_3	49 nodes	21 h	300,938	Figs. 2c, 3c, 6c
evalab_1	22 nodes	3 days	9,422	Figs. 6d
evalab_2	22 nodes	3 days	58,895	Figs. 2d
smartmarina_1	18 nodes	4 months	1,122,177	Figs. 2e
smartmarina_2	19 nodes	4 months	183,939	Figs. 6e
peach_1	19 nodes	4 months	166,927	Figure 6f
inria-c	20 nodes	30 h	23,205	Figs. 3d, 4c, 4d
11	170,037-hours of operation		2,873,156 measurements	

Table 1: Summary of the published datasets

³ **TEMPORARY NOTE TO THE REVIEWERS:** the datasets will be contributed to the crawdad.org archive when this manuscript is accepted. We prefer not to publish it before to keep our “first-mover” advantage. We are happy to provide the datasets to any reviewer interested, as part of the review process.

Each dataset represents one experiment, and consists of a single Comma Separate Values (CSV) file which first line contains a JSON formatted set of meta information. The data format is the same whether it is generated by Mercator or SolSystem, allowing the same analysis tools to be used on both.

OBSERVATIONS FROM THE DATASETS

The datasets presented in Section “Dense Connectivity Datasets” contain a wealth of information. The goal of this section is to contrast/compare the connectivity in testbeds and real-world deployments. We highlight the lessons (we) learned when “moving beyond testbeds”, and believe these are interesting to the readership.

Clearly, the points we discuss do not necessarily apply to every testbed, nor do we claim to even know what “realistic” connectivity means (see discussion in Section “”). That being said, we believe the datasets to be comprehensive enough that we can extract clear connectivity characteristics in real-world cases that are not per-se present in testbeds. Our main message is that protocol evaluation should be done also in the presence of these different phenomenon.

Specifically, this section answers the following questions: *What are the phenomena related to connectivity that are typically seen in real-world deployments? How can these be measured? Are those phenomena present in most testbeds?*

Mercator was created specifically to gather dense datasets; all testbed datasets are hence used in each section below. SolSystem was not created to create these datasets, we hence cannot use all real-world datasets in each analysis. The specificities are: (1) the Peach network does not generate per-frequency information because of outdated firmware, (2) the EvaLab and SmartMarina deployment do generate per-frequency information, but not on a link-by-link basis, (3) the Inria-C dataset is the only one that contains per-link and per-frequency PDR measurements, but is constrained to a star topology. Based on these constraints, we pick the right datasets to fuel the different discussions points below.

Node Degree

Average node degree, or the average number of neighbors of the nodes in the network, is very typically used to quantify topologies. Table 2 shows the node degree in the 6 deployments, using a 0 dBm output power in the testbeds and +8 dBm in real-world deployments. We declare two nodes as being neighbors when the link that interconnects them has a PDR of at least 50%. We borrow this rule from SmartMesh IP (www.linear.com/dust_networks/).

	Lille	Grenoble	Strasbourg	EvaLab	SmartMarina	Peach
Average Node Degree	49.00	38.67	48.00	11.32	5.94	9.04

Table 2: Average degree of a node.

While there is certainly no rule for what a “realistic” node degree is, typical real-world deployment operators try to cut cost by the deploying the least amount of nodes possible. Analog Devices, for example, recommends that each node has at least 3 neighbors; if given the choice, network operators will not exceed that number. In that case, a node degree around 3 is a lower bound.

Table 2 shows that the testbeds used exhibit a very high node degree, at least 5 times that of the real-world deployments. Testbed operators typically recommend lowering the output power of the nodes to lower the average node degree. Section “A Word about Output Power Tuning” argues that this is not a good idea, but that the real solution is to spread the testbed nodes.

The lesson learned is that testbeds may be too densely deployed (e.g. all nodes in the same room) and that reducing the output power is not a valid workaround.

Witnessing External Interference

External interference happens when a different technology – or a different deployment of the same technology – operates within the same radio range. In the types of networks considered in this article, the most common case of external interference is IEEE802.11 WiFi interfering with IEEE802.15.4 at 2.4 GHz. WiFi interference causes a portion of the packets sent by the low-power wireless nodes to fail, requiring re-transmissions.

External interference can be shown by plotting the PDR, averaged over all measurements, grouped by frequency. This is done, for all deployments⁴, in Figure 2.

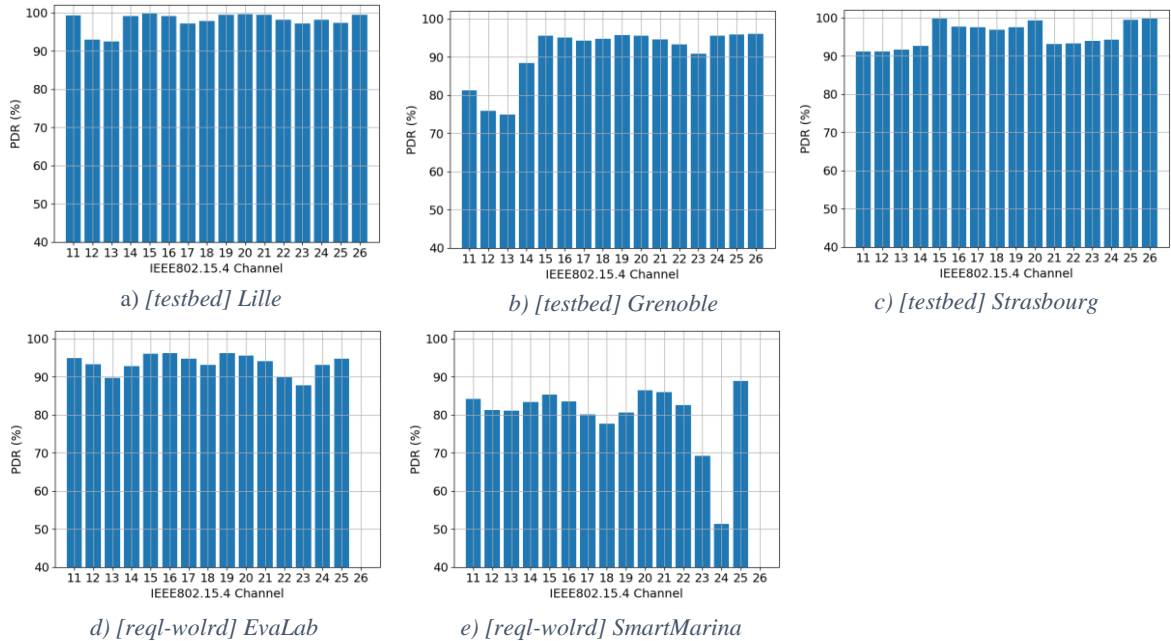


Figure 2: [External Interference] PDR per frequency, averaged over all measurements. IEEE802.15.4 channel 26 (2.480 GHz) is not used by SmartMesh IP, and hence does not appear in the real-world plots.

Figure 2 shows some level of external WiFi interference on all deployments, except for IoT-Lab Lille. In a multi-Access-Point WiFi configuration, different APs typically operate on 3 different frequencies, centered on IEEE802.15.4 channels 13, 18 and 23. This is clearly the case in the EvaLab deployment (Figure 2a). It appears from Figure 2b that IEEE802.11 channel 1 (2.412 GHz) is mostly used in IoT-Lab Grenoble. In the SmartMarina deployment (Figure 2e), the very high interference on IEEE802.15.4 channels 23-24 is due to a continuously streaming WiFi security camera next to the deployment site, operating on IEEE802.11 channel 11 (2.462 GHz).

The lesson learned is that external interference from WiFi is typically present in real-world deployments, and is also most often present in testbeds, as those are typically deployed in office buildings.

Witnessing Instantaneous Multi-Path Fading

Multi-path fading is both less intuitive and far more destructive than external interference. It is entirely caused by the environment around nodes that communicate. When node *A* sends a frame

⁴ The appropriate HRs data was not gathered on the SolSystem Peach deployment; we are hence unable to plot the figure for that deployment.

to node B , what B receives is the signal that has traveled over the line-of-sight path between A and B , but also the “echoes” that have bounced off nearby objects. Depending on the relative position of nodes A and B and the objects around, these different components can destructively interfere. The result is that, even though A and B are close, and that A transmits with a high output power, B does not receive any of its frames. This “self-interference” pattern depends on the frequency used. What typically happens is that node A can send frames to node B on most of the available frequencies, except on a handful of frequencies on which communication is impossible. The impact of multi-path fading is higher when the deployment area is cluttered by highly reflective (e.g. metallic) objects.

What we are looking for in the datasets is hence how many frequencies are usable ($\text{PDR} > 50\%$) for each link. If all frequencies are usable, there is no multi-path in the environment. Figure 3 plots, for each PDR measurement, how many frequencies have a PDR higher than 50%.

In the IoT-Lab Lille case (Figure 3a), almost all PDR measurements show that all frequencies are usable: there is very little multi-path fading in that environment. This is expected, as the deployment is done in one large uncluttered room (see Figure 1a). In contrast, multi-path fading is very present in the IoT-Lab Grenoble site (Figure 3b). This is expected, as the deployment is done in a tight space between the dropped ceiling and the roof, a space cluttered with metallic structure and wiring (see Figure 1b). Multi-path fading is also very present in the Inria-C deployment (Figure 3d). This deployment spans multiple rooms, with the 20 m long links crossing several walls and rooms filled with white boards, chairs, tables, ventilation piping, etc., all opportunities for multi-path fading to occur.

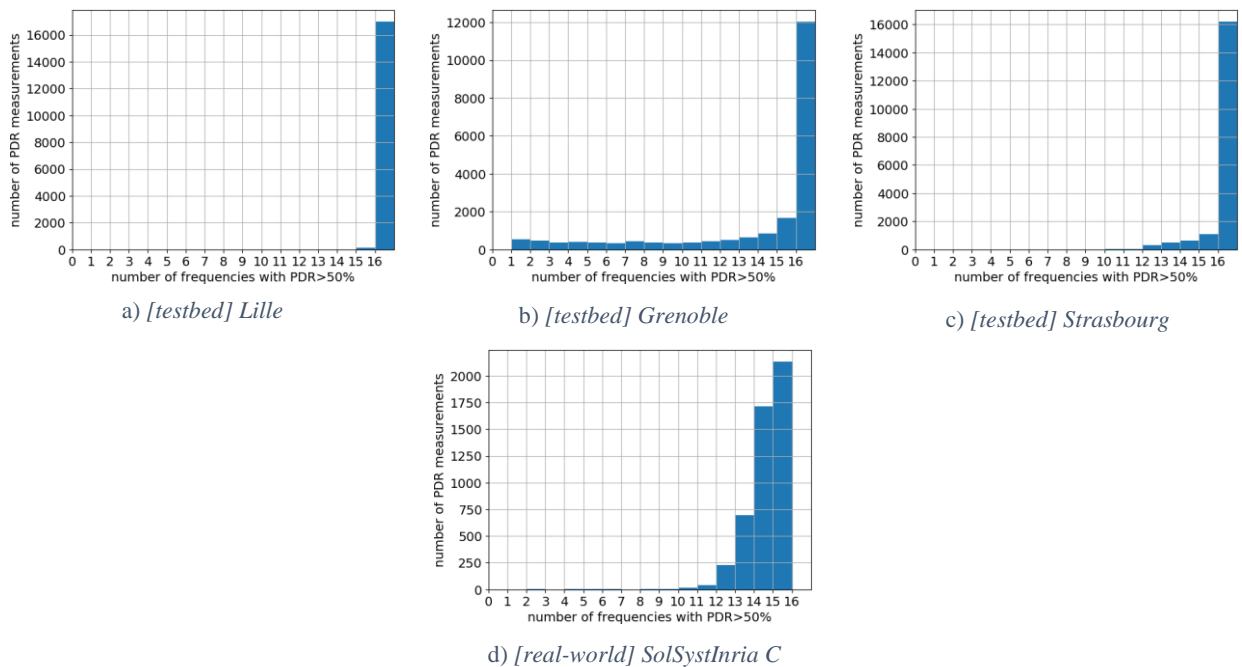


Figure 3: [Instantaneous Multi-Path Fading] Measurements with number of frequencies with $\text{PDR} > 50\%$.

Multi-path fading takes place in varying degrees in virtually all deployments. It is in particular present in an environment cluttered with highly reflective (e.g. metallic) objects, or simply when links are long (over 10 m). It causes the PDR of a link to vary significantly with frequency, and it is essential to test networks in testbeds in which there is a lot of multi-path. The lesson learned is that it is essential to deploy a testbed across a large area, e.g. across an entire floor rather than in a single room.

Witnessing Dynamics in the Environment

In virtually any real-world deployment, the environment changes over time: people move across buildings, WiFi traffic continuously changes, machines are switched on and off, doors are opened and closed, forklifts zip around factory floors, etc. This means that the level of both external interference and multi-path fading changes over time. From a connectivity point of view, this means that the PDR of each link varies over time, and across all frequencies.

Figure 4 shows how the PDR of particular links varies over time, on each IEEE802.15.4 frequency. The gray zones highlight daily business hours. While we had to choose specific links for each deployment, we make sure they are representative of the other links.

In the Inria-C deployment, Figure 4 c) and d) show the PDR variation over time for the link from nodes *TX1* and *TX2* sending to node *RX*, respectively. Nodes *TX1* and *TX2* are both placed 27 m away from *RX*. Even though *TX1* and *TX2* are only separated by 50 cm, the per-frequency PDR variations on their links to node *RX* evolve in very different manners, which is expected.

Figure 4 a) and b) show the variation of PDR on a particular link in the IoT-Lab Lille and IoT-Lab Strasbourg deployment, respectively. Even over many days, there are no significant changes in PDR. This has severe consequences, as a networking solution validated on a testbed like this might fail in the real world, in which the environment (and the PDR) changes frequently.

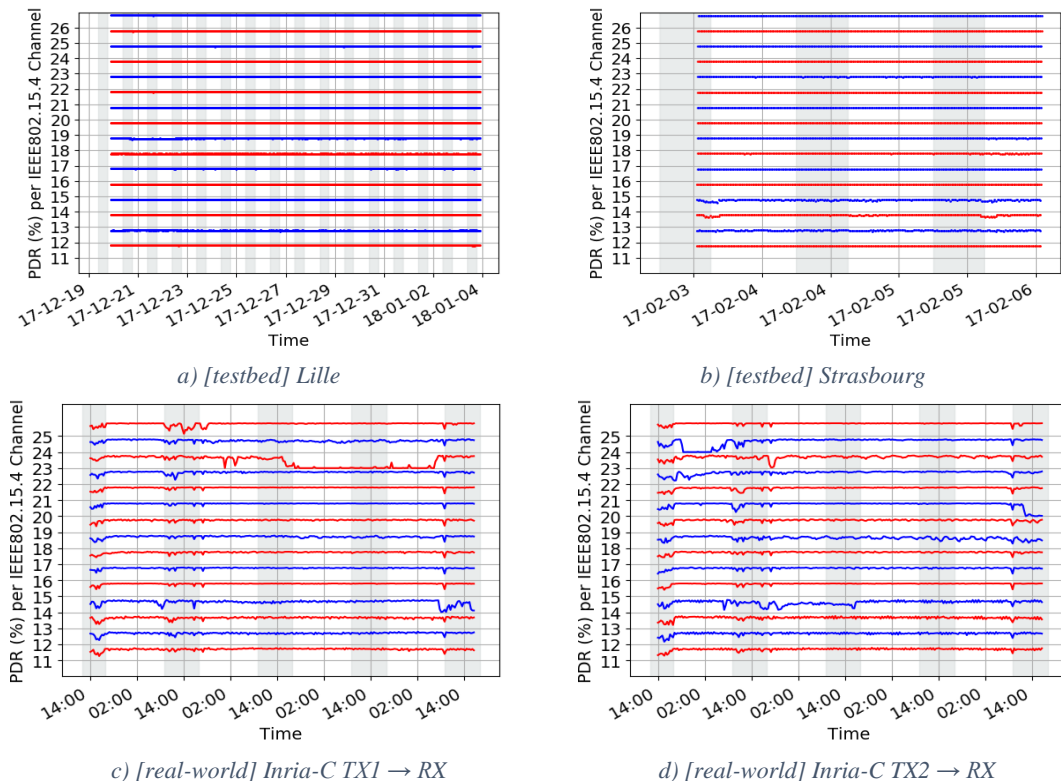


Figure 4: [dynamics in the environment] PDR evolving over time for specific links.

In virtually all real-world deployments, the environment in which nodes are deployed changes, resulting in dynamics in the connectivity between nodes, on each frequency. Testbeds often do not capture these effects, as nodes may be deployed in basements. This has a severe impact on the validity of evaluations in these testbeds, and solutions working perfectly on them might not work at all in the real world. The lesson learned is that the evaluation of a networking solution on a testbed without dynamics has very limited validity.

DISCUSSION

The goal of this section is to discuss what changes when going from testbeds to real-world deployments. In particular, we discuss some of the steps one needs to take to ensure a solution is properly tested in a testbed so it succeeds in real-world deployments.

What is Realistic?

We do not claim to know what “realistic” connectivity looks like. Every deployment is different, and a dense deployment in a small basement room is as realistic as a deployment on an entire factor floor. It all depends on the application. This article *does not* argue in favor or against particular testbeds.

Rather, this article lists the 3 phenomena that are most common in real-world deployment, and which have a deep impact on connectivity: external interference, multi-path fading, dynamics in the environment. Any deployment will exhibit a combination of these three phenomena. **When evaluating a networking solution, it is hence essential to do so in environment(s) which exhibit all three. Without this, you run the risk of having your solution fail during a real-world deployment.**

Before evaluating a solution in a testbed, we recommend you go through the following 5-point checklist:

1. Gather connectivity traces that are dense in time, frequency and space, by using Mercator, SolSystem, or an equivalent tool.
2. Compute the average node degree (as in Section “Node Degree”), and ensure that you are testing your solution also on very sparse deployments (down to a degree of 3).
3. Plot the average PDR for each frequency (as in Section “Witnessing External Interference”), and ensure that you see variation across different frequencies, indicating the presence of external interference.
4. Plot a histogram of the number of frequencies with PDR>50% (as in Section “Witnessing Instantaneous Multi-Path Fading”), and ensure that a significant portion of the links in your deployment have one or more “bad” frequencies, indicating the presence of multi-path fading.
5. Plot, for each link, the evolution of its PDR over time, for each frequency (as in Section “Witnessing Dynamics in the Environment”), and ensure that a significant portion of the links see the PDR switch from 0% to 100% on multiple frequencies, indicating the presence of dynamics in the environment.

It is our experience that a solution evaluated on a testbed in which the check-list above passes performs well in real-world deployments.

A Word about Output Power Tuning

Some testbeds are often too densely deployed, and to limit the node degree (number of neighbors) and increase the network radius (number of hops), testbed operators often reduce the output power of the radios (e.g. -55 dBm). This is not a good idea. The reason is that this also limits the amount of multi-path fading, as the echoes that reach the receiver antenna are so weak that self-interference is not happening. The result is a deployment that looks more like free space.

Instead, we recommend installing MAC address filters on the nodes so they artificially drop packets from neighbors not in the list. This is a way to force a topology while maintaining the same level of multi-path fading and dynamics.

Waterfall Plot

Each PDR measurement in the datasets also contains the average RSSI over the 100 frames received in that burst. Plotting a scatterplot of PDR as a function of RSSI reveals a large number of insights about the connectivity in the network. Because of its shape, we call this a “waterfall plot”. In the absence of external interference and multi-path fading, the waterfall plot is at $\text{PDR} \approx 0\%$ 10-15 dB below the radio chip’s sensitivity, at $\text{PDR} \approx 100\%$ above sensitivity, and with an almost linear ramp between the two.

You can apply the tools detailed in this section both on your testbed (to verify it is “realistic”), and on your real-world deployment (to quantify its connectivity).

We assume you have generated a waterfall plot from the connectivity dataset gathered in your deployment. Figure 5 shows such a waterfall plot. Each cross represents a PDR measurement; the mean value with standard deviation is also depicted. Figure 5 contains annotations on how to “read” it:

1. Make sure the left-hand side of the waterfall plot is complete, i.e. it reaches 0%. Not having this left-hand side indicates that your nodes are very close to one another. On a testbed, this means you are not testing your solution close to sensitivity.
2. Any discontinuity in the plot indicates that your deployment contains either very good links, or bad links, but no in-between. This is typically the case for networks in which nodes are deployed in clusters.
3. A waterfall plot shifted to the right indicates the presence of external interference and multi-path fading.
4. A “dip” in the waterfall plot indicates strong interference on specific links.
5. The spread of PDR measurements around the mean value indicates dynamics in the environment.

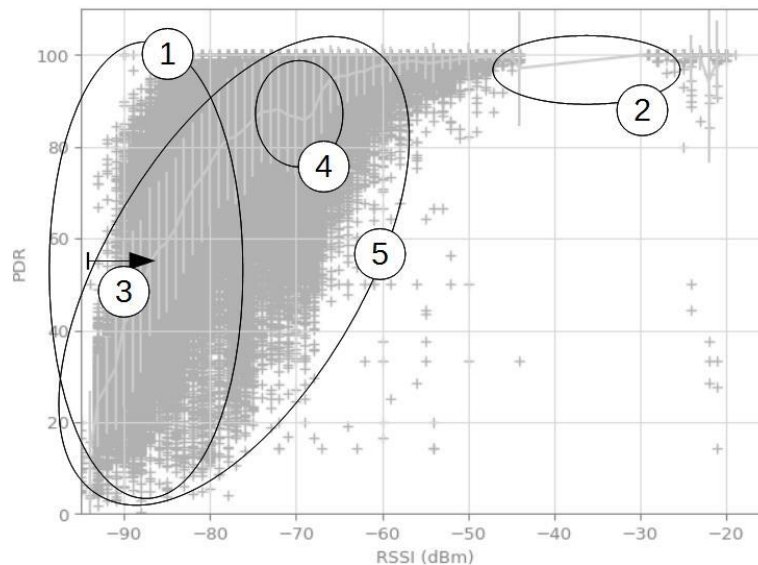


Figure 5: Five elements to look at when assessing the connectivity in a deployment by “reading” its waterfall plot (detailed in Section “Waterfall Plot”).

Five elements to look at when assessing the connectivity in a deployment by “reading” its waterfall plot (detailed in Section “Waterfall Plot”).

Given these rules, just looking at a waterfall plot allows one to determine how close together nodes are deployed, and whether external interference, multi-path fading and dynamics are present.

We show the waterfall plots for all deployments in Figure 6. The rules above allow us to get good insights into the connectivity in the deployments (circled number refer to the rules above). The IoT-Lab Lille and Strasbourg testbeds (Figure 6a and Figure 6c) suffer from the fact that nodes are deployed too close to one another (1). Nodes are deployed in clusters in SmartMarina, as shown by the discontinuity in the plot (2). The fact that the EvaLab and SmartMarina waterfall plot are shifted right compared to Peach indicates external interference in the former two, very little in the latter (3). A WiFi camera interferes with a small number of links in SmartMarina; this can be seen by the “dip” in the plot (4). Nodes in the IoT-Lab Grenoble testbed are deployed far enough apart from each other, but lacks dynamics in the environment (5).

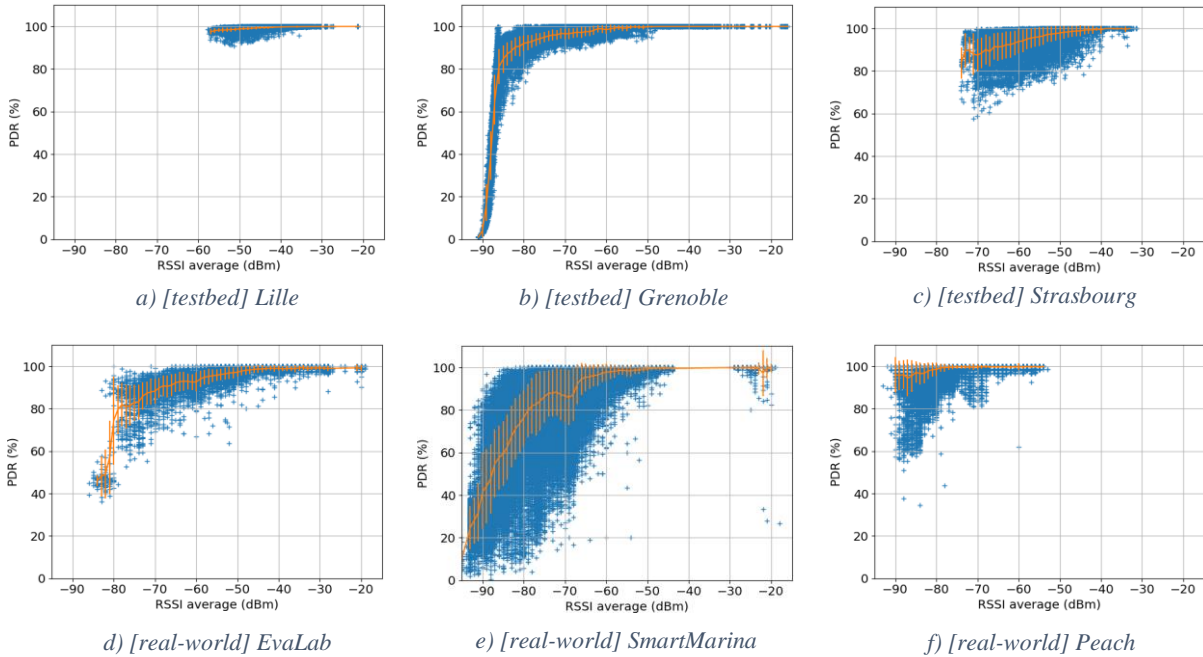


Figure 6: Waterfall plots for the different deployments.

Directions for Future Work

The format of the dataset highlighted in Section “Published Datasets” is generic enough that additional datasets can be added. There would be great value in creating a standardized connectivity evaluation kit and deploy it in various environments for several weeks, in order to generate a comprehensive set of connectivity datasets. Simulation platforms could be modified to “replay” these connectivity datasets, rather than relying on propagation models at the physical layers. The benefits would be that (1) this would increase the realism and confidence in the simulation results, and (2) the same simulation could be run against a number of datasets, which would serve as connectivity scenarios.

There would be great value in defining a set of metrics to quantify how much external interference, multi-path fading and dynamics there is in a network. Networking solution could be benchmarked against several deployments, covering a range of metrics.

Similarly, it would be interesting to evaluate how much the type of connectivity impacts the performance of networking solution, such as those proposed by the academic community.

REFERENCES

1. M. Doddavenkatappa, M. C. Chan, and A. L. Ananda, "Indriya: A Low-Cost, 3D Wireless Sensor Network Testbed," in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Springer Berlin Heidelberg, 2012, pp. 302–316.
2. C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele, and T. Watteyne, "FIT IoT-Lab: A Large Scale Open Experimental IoT Testbed," in *World Forum on Internet of Things (WF-IoT)*. IEEE, 2015.
3. P. H. Gomes, Y. Chen, T. Watteyne, and B. Krishnamachari, "Insights into Frequency Diversity from Measurements on an Indoor Low Power Wireless Network Testbed," in *Global Telecommunications Conference (GLOBECOM), Workshop on Low-Layer Implementation and Protocol Design for IoT Applications (IoT-LINK)*. Washington, DC, USA: IEEE, 4-8 December 2016.
4. A.-S. Tonneau, N. Mitton, and J. Vandaele, "How to Choose an Experimentation Platform for Wireless Sensor Networks?" *Elsevier Adhoc Networks*, 2015.
5. G. Z. Papadopoulos, A. Gallais, G. Schreiner, and T. Noël, "Importance of Repeatable Setups for Reproducible Experimental Results in IoT," in *Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, 2016.
6. T. Watteyne, C. Adjih, and X. Vilajosana, "Lessons Learned from Large-scale Dense IEEE802.15.4 Connectivity Traces," in *International Conference on Automation Science and Engineering (CASE)*. IEEE, 2015.
7. J. Zhao and R. Govindan, "Understanding Packet Delivery Performance in Dense Wireless Sensor Networks," in *Embedded Networked Sensor Systems (SenSys)*. ACM, 2003, pp. 1–13.
8. W. Dong, Y. Liu, Y. He, T. Zhu, and C. Chen, "Measurement and Analysis on the Packet Delivery Performance in a Large-Scale Sensor Network," *IEEE/ACM Transactions on Networking*, no. 6, 2014.
9. L. Doherty, W. Lindsay, and J. Simon, "Channel-Specific Wireless Sensor Network Path Data," in *International Conference on Computer Communications and Networks (ICCN)*. IEEE, 2007.
10. Kotz, David and Henderson, Tristan and Abyzov, Ilya and Yeo, Jihwang, "CRAWDAD dataset dartmouth/campus (v. 2009-09-09)" Downloaded from <https://crawdad.org/dartmouth/campus/20090909>, 2009.
11. S. Fu, Y. Zhang, Y. Jiang, C. Hu, C. Y. Shih, and P. J. Marran, "Experimental Study for Multi-layer Parameter Configuration of WSN Links," in *International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2015.
12. T. Watteyne, A. Mehta, and K. Pister, "Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense," in *Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*. ACM, 2009.
13. T. Watteyne, J. Weiss, L. Doherty, and J. Simon, "Industrial IEEE802.15.4e Networks: Performance and Trade-offs," in *International Conference on Communications (ICC)*. IEEE, June 2015, pp. 604–609.