# RSSI: Lost and Alone, a Case for Redundancy

Mohammad Imran Syed
*Sorbonne Université, CNRS, LIP6*
Paris, France
mohammad-imran.syed@lip6.fr

Anne Fladenmuller
*Sorbonne Université, CNRS, LIP6*
Paris, France
anne.fladenmuller@lip6.fr

Marcelo Dias de Amorim
*Sorbonne Université, CNRS, LIP6*
Paris, France
marcelo.amorim@lip6.fr

***Abstract*—The Received Signal Strength Indicator (RSSI) is a preferred metric for localization and distance estimation among researchers. Although it is easy to use, as it is available for most operating systems, it is not entirely reliable when used alone. We assess the inconsistency of RSSI values in an experimental setup. We deploy multiple sniffers to investigate whether redundancy can reduce measurement errors and, if so, by how much. Among other measures, we consider the total number of packets that individual sniffers miss as well as bursts of consecutive misses. We analyze their implications for distance estimation and the detection of mobile targets. As a major outcome of our analysis, we estimate that the redundancy of three sniffers yields a good balance between accuracy and cost.**

***Index Terms*—Distance estimation, wireless, passive measurement, RSSI, redundancy.**

## I. Introduction

The number of smartphone users is an ever-increasing figure, and it is expected to reach 7,516 billion by 2026 [1]. Furthermore, as more and more devices connect to the Internet [2, 3], forecasts suggest that, by the end of 2022, there will be approximately 362 million public Wi-Fi hotspots available worldwide [4]. These numbers lead to an amplification of the topology dynamics and more challenging network management issues. Consequently, our dependence on efficient measurement techniques to precisely characterize the network and understand the mobility of users also increases.

Although cellular operators produce a lot of location data, they are not publicly available. As a result, the research community still relies on a limited set of traces, which restrain the universe of possible observations. There is a need for wireless measurements to build traces that researchers can use to evaluate and improve networking approaches. In this paper, we focus on *estimating distances* separating two or more moving wireless devices using *passive measurements*.

Passive measurements are a non-intrusive data collection method and an effective way of tracing mobility and localization. It relies on *sniffers* (devices collecting wireless packets in monitor mode) placed throughout the desired testing area.[1] Passive measurements are easy to run and are preferable because they do not require to bother the infrastructure administrators or end-users [5, 6, 7, 8].

The Received Signal Strength Indicator (RSSI) values are of interest to detect the presence of a node in a target area and evaluate its position relatively to the sniffer [9, 10, 11, 12].

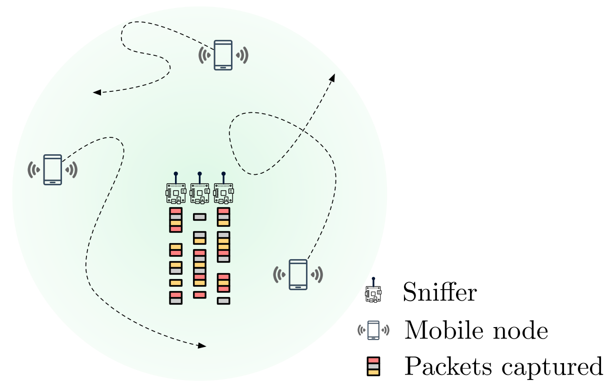



Fig. 1: Co-located sniffers may miss different packets.

Furthermore, they are available in most operating systems in a straightforward fashion. However, the downside of using the RSSI concerns its accuracy. Although there are some workarounds like machine learning to reduce the error in distance estimation [13, 14], the applicability of the RSSI is still questioned [15, 16, 17].

As a part of the ANR MITIK project [18], we collect Wi-Fi traces through passive measurements to capture the mobility of wireless nodes in a given area. In the project, we focus on probe request messages, the only messages a user's device sends when it is not associated with any Wi-Fi access point. The problem is that their sending rate can be as low as 55 packets per hour or as high as 2,000 packets per hour, depending on the device [19]. Missing these probe requests may have severe consequences on the quality of the measurement campaign, as it may lead to nodes not being detected or biased mobility estimations.

The capture ratio of a single sniffer is usually low because of packet losses due to the inherent characteristics of the wireless medium. We depict this scenario in Fig. 1, where the sniffers, although co-located, capture different sets of packets. We advocate that by combining traces of multiple co-located sniffers, we limit the number of missed packets and get closer to the complete trace. The question we address in this paper is *what redundancy (number of co-located sniffers) is needed to achieve a good balance between trace completeness and deployment cost*.

To this end, we conduct experimental work. We collect wireless traces using five Raspberry Pi 4B (RPi4 hereafter)

[1] It is, however, essential to define which data one can sniff in a given location to preserve the users' privacy.

978-1-6654-6975-3/22/$31.00 ©2022 IEEE

devices as sniffers and one as a source generating Wi-Fi traffic [20]. The source is the node we want to characterize. We co-locate the sniffers (i.e., introduce redundancy) to investigate the consistency of the RSSI values. We do the measurements outdoors and monitor the traffic generated by a controlled source for which we know the ground truth of its distance from the sniffers. In our experiments, we place the source node at multiple distances from the sniffers and make the following observations:

- Individual sniffers miss quite a lot of packets. However, the percentage of packets captured by the sniffers is homogeneous for a given distance.
- All sniffers do not necessarily miss the same packet. It means the packet is undoubtedly not lost because of the collisions at the sniffers.
- The number of consecutive packets missed by a sniffer can be huge at times. Moreover, the sniffers miss the capture for several seconds, which is problematic if one has to analyze mobility.
- At times, there is incoherence in RSSI values of the same packet captured by different sniffers, which leads to frequent errors in distance estimation.

Our experiments confirm that redundancy improves the capture quality by reducing the number of packets missed and the gaps due to consecutive packet misses. More importantly, we noticed that the redundancy of three seems to be the sweet spot for the type of nodes and antennas we used in our experimental campaign. Furthermore, we could also solve the problem of the incoherence of RSSI values and reduce the error in distance estimation.

The rest of the paper is organized as follows. We explain the complete experimental methodology and the dataset in Section II. In Section III, we present the analysis to show the need for redundancy. In Section IV, we present the results for distance evaluation using redundancy. Section V mentions the related work. We finally conclude the paper and indicate future directions in Section VI.

## II. Experimental methodology and datasets

In this section, we explain our experimental methodology and the datasets that we obtain as a result. As we mentioned in Section I, we collect Wi-Fi traces generated by our own source node, which lets us know the ground truth and rule out any privacy issues. We run the experiments outdoors and consider the redundancy of up to five sniffers. We perform several tests where we place the source node at different distances from the sniffers.

### A. Experimental set-up

**Nodes.** We use six RPi4 nodes in our measurement set-up, five as co-located sniffers and one as the source of Wi-Fi traffic. We use one external Wi-Fi module per sniffer (Alfa AWUS051NH [21]). The advantage of this specific external Wi-Fi module, contrarily to others commercially available, is that it can be easily set to monitor mode to capture the Wi-Fi traffic passively. In our experiments, the source and sniffers operate on channel 1 of the 2.4 GHz band.

Fig. 2: The layout of the sniffers for the experiment (placed on the ground).

**Scenario.** We capture traces outdoors while maintaining a line of sight between the source node and the sniffers. We place sniffers side by side with a distance of 20 cm between each of them (see Fig. 2), and they remain stationary for the duration of the experiments. We place the source node at distances of 1, 10, 20, 30, 40, and 50 meters from the sniffers. We perform four tests for each distance, each lasting five minutes.

### B. Methodology of the capture

**Trace generation.** We use `scapy` [22] at the sender node to send Wi-Fi probe requests. The source node transmits at a rate of 10 packets per second. This rate is the maximum value we can achieve without looping through the sequence numbers; in this way, we can identify the packets correctly and synchronize the sniffers' traces.

**Trace capture.** Sniffers run `tcpdump` to collect the traces [23]. We configure filters to gather only the traffic generated from the controlled source node. The final captured trace is one `pcap` file per individual sniffer and distance.

**Comparison between traces.** The sniffers' traces are not synchronized among them because the sniffers have their own local clocks. We need synchronization to be able to compare the traces that the sniffers collect at the same time. Synchronization is mandatory to identify the common frames, i.e., frames captured by at least two sniffers. We developed a Python tool called `PyPal` that performs such a synchronization operation [24]. Once the traces are synchronized, we concatenate the traces of all four tests to generate one large trace for each distance. We then perform multiple analyses to determine how many sniffers capture each packet. We store the RSSI values for each packet as seen by each sniffer.

**Steps involved in synchronization.** The beacons are the closest representatives of real-time clocks. Moreover, the IEEE 802.11 standard dictates that the beacon frames have a fixed timestamp in the header, added by the access point, further improving synchronization precision. We use these frames as a basis for the synchronization process as illustrated by Fig. 3. Two traces come as inputs: one as a reference

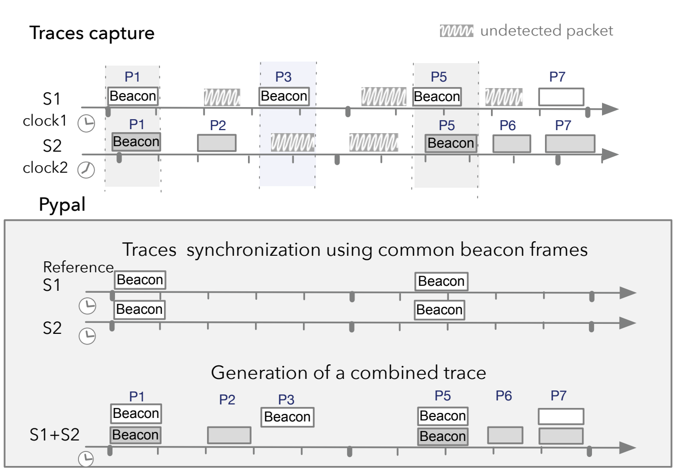



Fig. 3: PyPal's methodology for synchronizing traces.

trace and the second as the one to be synchronized. The first step consists of extracting the beacons that appear in both traces (common frames). Note that the coverage areas of the sniffers capturing these traces should overlap; otherwise, the traces will be disjoint. The common frames serve as reference frames. In the next step, the timestamps of reference frames are synchronized using linear regression over a sliding window of 3 frames. The synchronized reference frames are then the guidelines to synchronize the complete trace. The tool provides additional options to concatenate or merge the synchronized traces, which are out of the scope of this paper.

**Wi-Fi header fields that identify reference frames.** We use a combination of different header fields to identify reference frames that are present in both traces: (i) sender' MAC address, (ii) sequence number, (iii) frame sub-type, (iv) Frame Checksum Sequence (FCS), (v) fragment number, and (vi) fixed timestamp as per the IEEE 802.11 standard.

### C. Dataset

Each sniffer generates one trace per test for each distance. As discussed earlier, we run the test four times. So we obtain 120 traces in total. As the first step after time synchronization, we organize all traces within a single test – we perform this step separately for each distance. Once we have a single trace per test, we combine the traces from each test to get one single trace for each distance. At this stage, we have all the traffic captured in the four tests at a given distance in one final trace. We do the analysis to identify the packets that are captured by multiple sniffers and group them by their RSSI values. So, we have a single concatenated file per distance representing the RSSI values of each packet as seen by different sniffers. For the sake of illustration, we provide in Table I a snapshot of such a trace for the distance of 50 m. Note that, for example, sniffer $s_4$ misses the first packet, while sniffers $s_1$, $s_3$, and $s_5$ miss the second packet.

We use this dataset for all the analyses we present in the remainder of this paper.

TABLE I: Per-packet per-sniffer RSSI (dBm) at 50 m for a subset of the collected trace. "–" means that particular sniffer does not capture the packet.

| | **RSSI captured per sniffer** | | | | |
|---|---|---|---|---|---|
| **Packet** | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ |
| 1 | -48 | -54 | -48 | – | -46 |
| 2 | – | -78 | – | -48 | – |
| 3 | – | -72 | – | – | – |
| 4 | – | – | -64 | – | – |
| 5 | -48 | -56 | -48 | -48 | -44 |
| 6 | -48 | -54 | -50 | -48 | -46 |
| 7 | -46 | -54 | -50 | -48 | -44 |
| 8 | -46 | – | -48 | -48 | -44 |
| 9 | -48 | – | – | – | – |
| 10 | – | – | – | – | -44 |
| 11 | – | – | – | -50 | – |
| 12 | -46 | -54 | -50 | -48 | -44 |

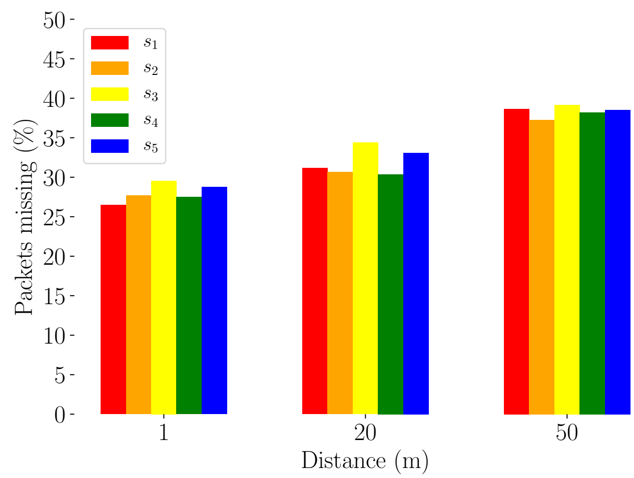



Fig. 4: Global error for individual sniffers at 1, 20, and 50 meters. Note that the longer the distance, the higher the ratio of packet misses.

## III. Is a single sniffer enough?

### A. Global error

A single sniffer is bound to lose some traffic because of the characteristics of the wireless medium. We show in Fig. 4 the percentage of packets missed by each sniffer at distances of 1, 20, and 50 meters.[2] We call it the *global error* because it represents the overall packet loss. We observe that between 26% and 30% of the packets are missed at a distance of 1 m, while the miss ratio increases to a maximum of nearly 35% and 40% for 20 m and 50 m, respectively. However, we observe that the error ratio is roughly the same for all five sniffers for the same distance.

### B. Burst of packets missing

We now look into the *burst size* of consecutive packets missing. It is an important parameter as it translates into a period during which the sniffers fail to detect the presence

[2]We only show three representative values for the sake of readability, as the results are similar for 10, 30, and 40 meters.

TABLE II: Maximum burst size (M.B.S.) for individual sniffers and the ratio between the number of maximum size bursts and the total number of bursts (Freq.).

| | **1m** | | **10m** | | **20m** | | **30m** | | **40m** | | **50m** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Sniffer** | **M.B.S.** | **Freq.** | **M.B.S.** | **Freq.** | **M.B.S.** | **Freq.** | **M.B.S.** | **Freq.** | **M.B.S.** | **Freq.** | **M.B.S.** | **Freq.** |
| $s_1$ | 7 | 1/1997 | 7 | 2/1926 | 10 | 1/2065 | 9 | 1/2120 | 9 | 1/2090 | 12 | 1/2098 |
| $s_2$ | 6 | 1/2063 | 8 | 1/1952 | 8 | 2/2030 | 7 | 5/2077 | 8 | 2/2042 | 9 | 3/2067 |
| $s_3$ | 9 | 1/2142 | 9 | 1/2078 | 259 | 1/2040 | 7 | 3/2147 | 11 | 1/2058 | 9 | 2/2110 |
| $s_4$ | 6 | 4/2077 | 8 | 1/1973 | 8 | 1/2035 | 11 | 1/2187 | 9 | 1/2081 | 11 | 1/2082 |
| $s_5$ | 54 | 1/2078 | 7 | 2/1931 | 7 | 3/2101 | 8 | 1/2079 | 13 | 1/2115 | 13 | 1/2073 |

of devices – depending on their mobility, they may traverse the monitored zone without being even detected. We show in Table II the maximum burst size (M.B.S.) of consecutive packet misses for each one of the five sniffers at each distance, along with the ratio between the number of maximum size bursts and the total number of bursts (Freq.). The numbers are far from negligible in certain cases. We see a gap of 259 consecutive packet misses by sniffer $s_3$ for the distance of 20 m. Recall that our source sends packets at a rate of 10 packets/s. Thus, $s_3$ misses packets for 25.9 seconds, which is massive! For contextualization, the average walking speed of an adult is between 1.2 m/s and 1.4 m/s, which means that a mobile user would have moved between 31.08 m and 36.26 m in 25.9 seconds [25, 26]. The phenomenon is even worse if we consider an average transmission rate of 1,028 probes per hour [19]. If a sniffer misses 259 consecutive packets, then it will end up missing traffic for 15.24 minutes. The user would have covered a distance between 1,097 and 1,280 meters. It means by the time this sniffer captures the next packet, the user is most likely to be out of the coverage area of several following sniffers.

### C. Introducing redundancy

Using a single sniffer can be risky. We highlight the need for more robust capture strategies. To this end, we combine the traces from different co-located sniffers to assess the advantages of redundancy. We take one sniffer and keep adding other sniffers one by one. We analyze all possible combinations of sniffers for a given redundancy, which gives a total of $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ combinations.

We do the calculations for all combinations and take the average for all combinations of the same size. We show in Table III the results for the average maximum burst size of consecutive packets missing. We see that even the average value of maximum burst size for single sniffers is quite high for all distances. However, the burst size decreases significantly for a combination of two sniffers for each distance. The results improve further for larger combinations of sniffers. The results are zero for the captured traffic when we consider the combination of all sniffers because each packet is captured by at least one sniffer.

TABLE III: Average burst size of consecutive packets missing for multiple sniffers.

| **Number of combined sniffers** | **Average burst size per distance** 1 m | 10 m | 20 m | 30 m | 40 m | 50 m |
|---|---|---|---|---|---|---|
| 1 | 16 | 8 | 58 | 8 | 10 | 11 |
| 2 | 4 | 5 | 5 | 5 | 5 | 6 |
| 3 | 3 | 3 | 3 | 4 | 3 | 4 |
| 4 | 2 | 2 | 2 | 2 | 2 | 3 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |

## IV. Distance evaluation

### A. Detection of a node

We use the Log-Distance Path Loss (LDPL) model to estimate the distance and put values into context. The formula for LDPL is as follows [27, 28]:

$$\texttt{RSSI}(d) = \texttt{RSSI}(d_0) - 10 \times \eta \times \log\left(\frac{d}{d_0}\right), \tag{1}$$

where $\texttt{RSSI}(d)$ is the RSSI value seen by the sniffers when they are at a distance $d$ from the source, $\texttt{RSSI}(d_0)$ is the RSSI value at a close-in reference distance $d_0$, and $\eta$ is the path loss index which is dependent on the propagation environment. We use the following values for our experimental set-up:

- $d_0$ = 1 meter,
- average RSSI at $d_0$ = -19.7 dBm, and
- $\eta$ = 1.75.

In our experiments, the average RSSI value at a reference distance of 1 m is -19.7 dBm. Although there are a few building walls on the side, we maintain Line of Sight (LoS) in our experiments, so we choose 1.75 as the value of $\eta$. The specifications of the hardware are as follows:

- transmission power: 27 dBm
- antenna gain: -5 dBi
- receiver sensitivity: -93 dBm

### B. Disparity in RSSI

We see in Table I that -46 dBm and -48 dBm are the most common RSSI values for all sniffers, but we see some values as -78 dBm, -72 dBm, -56 dBm, and -54 dBm. We classify these values as outliers by using the Majority Rule Scheme strategy [29]. These values are not coherent with other measures and result in large errors in distance estimation. We have such RSSI outliers for all distances in our experiments. We calculate the average RSSI value for each packet for all combinations of sniffers for redundancy of 1 to 5.

Table IV shows the average per-packet error in distance estimate in meters, using two sniffers with the least error, for combinations of all number of sniffers at 50 m. We see in row 1 that the average error of estimated distance for a combination of two sniffers is 6.24 m which is 2.35 m less than the average of individual sniffers. Similarly, if we consider only the average of single sniffers in the second row, the average

TABLE IV: Raw per-packet average distance error for the distance of 50 m.

| | Number of sniffers | | | | |
|---|---|---|---|---|---|
| **Packet** | **1** | **2** | **3** | **4** | **5** |
| 1 | 8.59 | 6.24 | 0.86 | 0.86 | 2.76 |
| 2 | 1051.8 | 8.59 | 8.59 | 128.32 | 248.05 |
| 3 | 924.03 | 924.03 | 924.03 | 924.03 | 924.03 |
| 4 | 289.96 | 289.96 | 289.96 | 289.96 | 289.96 |
| 5 | 8.59 | 6.24 | 0.86 | 2.76 | 3.99 |
| 6 | 6.24 | 2.76 | 0.86 | 0.45 | 1.5 |
| 7 | 6.24 | 2.76 | 0.86 | 1.6 | 6.35 |
| 8 | 8.59 | 8.59 | 10.41 | 13.66 | 16 |
| 9 | 8.59 | 8.59 | 8.59 | 8.59 | 8.59 |
| 10 | 25.53 | 25.53 | 25.53 | 25.53 | 25.53 |
| 11 | 3.88 | 3.88 | 3.88 | 3.88 | 3.88 |
| 12 | 6.24 | 2.76 | 0.86 | 1.6 | 6.35 |

TABLE V: Per-packet average distance error after removing the outliers for the distance of 50 m.

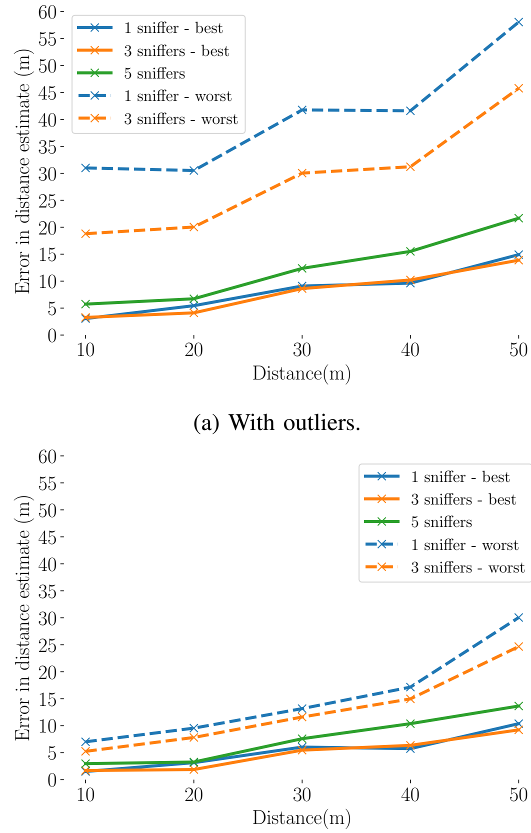
| | Number of sniffers | | | | |
|---|---|---|---|---|---|
| **Packet** | **1** | **2** | **3** | **4** | **5** |
| 1 | 8.59 | 6.24 | 0.86 | 0.86 | 2.76 |
| 2 | 8.59 | 8.59 | 8.59 | 8.59 | 8.59 |
| 3 | – | – | – | – | – |
| 4 | – | – | – | – | – |
| 5 | 8.59 | 8.59 | 8.59 | 6.24 | 3.88 |
| 6 | 6.24 | 2.76 | 0.86 | 0.45 | 1.5 |
| 7 | 6.24 | 3.32 | 1.81 | 0.66 | 0.45 |
| 8 | 8.59 | 8.59 | 8.59 | 10.41 | 12.23 |
| 9 | 8.59 | 8.59 | 8.59 | 8.59 | 8.59 |
| 10 | – | – | – | – | – |
| 11 | 3.88 | 3.88 | 3.88 | 3.88 | 3.88 |
| 12 | 6.24 | 3.32 | 1.81 | 0.66 | 0.45 |

error is huge, around 1050 m. However, the error goes down to 8.59 m when we take the average of the combination of two sniffers with the least error. Although the redundancy helps reduce the error, the presence of these outliers still results in massive errors, as we see for combinations of four and five sniffers in the second row. We clean the dataset to remove the outliers. The RSSI values below -56 dBm appear in only 2.86% of the cases; thus, we remove them from the 50 m dataset. Similarly, values greater than -46 dBm account for only 1.66% of the values. We recalculate the average values in Table V and we already see a huge improvement. We see in row seven that the average error with five sniffers goes from 6.24 m to only 0.5 m. The rows with no values mean that the sniffers presented outlier values.

In Fig. 5, we represent the best and worst-case results of average error in distance estimation for one, three, and all five sniffers, with and without outliers. We see that the worst-case error is massive for a single sniffer for all distances. Three sniffers yield an improvement, but the worst-case error is still significant. We also note in Fig. 5b that removing RSSI outliers significantly reduces the average error, including the worst case. Even a single sniffer gives low error for short distances but increases strikingly for longer distances, particularly for the worst-case scenario.

*C. Discussion*

The information of global error, bursts of packets missing, and outliers in the RSSI values lead us to the decision that using a single sniffer is not enough, especially in the case of measuring mobility. The results show that combining the sniffers improve all three issues that we highlight. On the one hand, the results of our experiments in Table V show that using five sniffers primarily results in a minimum error in distance estimate. On the other hand, the results in Table III show that even three sniffers (row 3) reduce the average burst size to 3 or 4, meaning the sniffers miss the traffic for 0.3∼0.4 seconds. The user could move between 0.36 and 0.56 meters in this period. Therefore, we propose that the redundancy of size 3 is good enough for an outdoor capture to rule out the anomalies in the trace capture and lead to more complete results and analysis. The best combination of 3 sniffers in Fig. 5 slightly outperforms the best case single sniffer which further strengthens our proposal of the use of redundancy of size 3.




(a) With outliers.



(b) Without outliers.

Fig. 5: Error in distance estimate for 1, 3, and 5 sniffers with and without outliers.

## V. Related Work

Adel et al. use ten maximum RSSI values for indoor localization using Bluetooth Low Energy (BLE). They use median, mean, mode, and single direction outlier removal to smooth the RSSI and improve the indoor distance estimate [30]. This method requires testing in a trilateration setup. Venkatesh et al. use the mean and median filters to stabilize the RSSI values to enhance the distance estimation accuracy for indoor localization in BLE [31]. Salomon et al. make a comparison for distance estimation by RSSI and Channel State Information (CSI) for Wi-Fi by doing experiments on RPi4 devices, one as an Access Point (sender) and one as receiver [32]. Forbes et al. use a single RPi4 as capturing device to perform distance estimation in Wi-Fi using CSI [33]. They capture the traffic generated by an Access Point in response to the packets it receives from a computer. Chuku et al. remove the RSSI outliers using clustering to improve the distance estimates [29] Our work stands distinctive as we introduce redundancy in the number of devices capturing the traffic, which reduces the error in distance estimation in Wi-Fi.

## VI. Conclusion and Future Work

We find out that individual sniffers miss many packets, impacting the quality of mobility traces. The problem is even more severe when we must rely on probe requests that are not very frequent. We measured the burst size of consecutive packet misses, and, in some instances, the number can be as high as a few hundred. Such misses translate into long data gaps and poor distance estimation when considering metrics such as the RSSI. In this regard, we observe fluctuations of RSSI values across multiple co-located sniffers. The use of a single sniffer can thus lead to incorrect distance estimations.

For this reason, we advocate for the introduction of redundancy to improve the quality of the traces and, consequently, the characterization of the nodes. In future work, we plan to investigate the inter-burst size distribution and its impact on long-term mobility analysis. We also intend to determine the value of the path-loss index from the RSSI value that appears the most for each distance and use that to find the error distribution in the distance estimation. Moreover, we plan to extend our work to accommodate mobility with the introduction of moving source nodes.

## VII. Acknowledgment

This work has been partially funded by the ANR MI-TIK project, French National Research Agency (ANR), PRC AAPG2019.

## References


[1] S. O'Dea, "Number of smartphone users from 2016 to 2021," https://statista.com/statistics/330695/number-of-smartphone-users-worldwide/.

[2] Statista Research Department, "Public wireless Internet usage in selected locations according to Internet users worldwide as of june 2015, by device," https://tinyurl.com/5y768zz6.

[3] Statista Research Department, "Wireless Internet access according to Internet users worldwide as of june 2015, by device," https://statista.com/statistics/463301/wireless-internet-access-by-device-worldwide/.

[4] L. S. Vailshery, "Number of public Wi-Fi hotspots worldwide from 2016 to 2022," https://tinyurl.com/4wfebd7j.

[5] Y.-C. Cheng, J. Bellardo, P. Benkö, A. C. Snoeren, G. M. Voelker, and S. Savage, "Jigsaw: Solving the puzzle of enterprise 802.11 analysis," in *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, New York, NY, USA, 2006.

[6] T. Claveirole and M. Dias de Amorim, "WiPal: Efficient offline merging of IEEE 802.11 traces," *SIGMOBILE Mob. Comput. Commun. Rev.*, p. 39–46, mar 2010.

[7] F. Garcia, R. Andrade, C. De Oliveira, and J. Souza, "EPMOSt: And energy-efficient passive monitoring system for wireless sensor networks," *Sensors (Basel, Switzerland)*, vol. 14, pp. 10 804–10 828, 06 2014.

[8] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Analyzing the MAC-level behavior of wireless networks in the wild," 2006.

[9] C. Bertier, "Quantification in Device-to-Device Networks : from Link Estimation to Graph Utility," Theses, Sorbonne Université, Jan. 2020. [Online]. Available: https://tel.archives-ouvertes.fr/tel-03347660

[10] T. I. Chowdhury, M. M. Rahman, S.-A. Parvez, A. K. M. M. Alam, A. Basher, A. Alam, and S. Rizwan, "A multi-step approach for RSSI-based distance estimation using smartphones," in *2015 International Conference on Networking Systems and Security (NSysS)*, 2015.

[11] S. Sadowski and P. Spachos, "RSSI-based indoor localization with the Internet of Things," *IEEE Access*, pp. 30 149–30 161, 2018.

[12] S. Yiu, M. Dashti, H. Claussen, and F. Perez-Cruz, "Wireless RSSI fingerprinting localization," *Signal Processing*, pp. 235–244, 2017.

[13] G. G. Anagnostopoulos and A. Kalousis, "A reproducible comparison of RSSI fingerprinting localization methods using LoRaWAN," in *2019 16th Workshop on Positioning, Navigation and Communications (WPNC)*, 2019.

[14] M. Anjum, M. A. Khan, S. A. Hassan, A. Mahmood, H. K. Qureshi, and M. Gidlund, "RSSI fingerprinting-based localization using machine learning in LoRa networks," *IEEE Internet of Things Magazine*, no. 4, pp. 53–59, 2020.

[15] K. Heurtefeux and F. Valois, "Is RSSI a good choice for localization in wireless sensor network?" in *2012 IEEE 26th International Conference on Advanced Information Networking and Applications*, 2012.

[16] Q. Dong and W. Dargie, "Evaluation of the reliability of RSSI for indoor localization," in *2012 International Conference on Wireless Communications in Underground and Confined Areas*, 2012.

[17] A. Parameswaran, M. I. Husain, and S. Upadhyaya, "Is RSSI a reliable parameter in sensor localization algorithms: an experimental study," Jan 2009.

[18] "ANR-MITIK," https://project.inria.fr/mitik/.

[19] J. Freudiger, "How talkative is your mobile device? An experimental study of Wi-Fi probe requests," in *Proceedings of the 8th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. New York, NY, USA: Association for Computing Machinery, 2015.

[20] "Raspberry Pi 4 model B," https://tinyurl.com/2p89uund.

[21] "AWUS051NH Wi-Fi adapter," https://tinyurl.com/yk8vk3vz.

[22] P. Biondi, "Scapy," https://scapy.net/.

[23] The Tcpdump Group, "Tcpdump and libpcap," https://tcpdump.org.

[24] M. I. Syed, A. Flandenmuller, and M. Dias De Amorim, "PyPal: Wi-Fi trace synchronization and merging python tool," LIP6 UMR 7606, UPMC Sorbonne Université, France, Technical Report, Mar. 2022. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03618014

[25] A. Mansfield, E. L. Inness, and W. E. Mcilroy, "Chapter 13 - stroke," in *Balance, Gait, and Falls*, ser. Handbook of Clinical Neurology, B. L. Day and S. R. Lord, Eds. Elsevier, 2018, vol. 159, pp. 205–228.

[26] O. Mohamed and H. Appling, "5 - clinical assessment of gait," in *Orthotics and Prosthetics in Rehabilitation (Fourth Edition)*, fourth edition ed. St. Louis (MO): Elsevier, 2020, pp. 102–143.

[27] A. Thaljaoui, T. Val, N. Nasri, and D. Brulin, "BLE localization using RSSI measurements and iRingLA," in *IEEE ICIT*, 2015.

[28] T. S. Rappaport, *Wireless communications - principles and practice.*, 1996.

[29] N. Chuku and A. Nasipuri, "RSSI-based localization schemes for wireless sensor networks using outlier detection," *Journal of Sensor and Actuator Networks*, vol. 10, 2021.

[30] A. Mussina and S. Aubakirov, "RSSI based Bluetooth Low Energy indoor positioning," in *2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT)*.

[31] V. R, V. Mittal, and H. Tammana, "Indoor localization in BLE using mean and median filtered RSSI values," in *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, 2021.

[32] E. Salomon, L. H. Botler, K. Diwold, C. A. Boano, and K. Römer, "Poster: Comparison of Channel State Information driven and RSSI-based WiFi distance estimation," in *Proceedings of the 2021 International Conference on Embedded Wireless Systems and Networks*. Junction Publishing, 2021.

[33] G. Forbes, S. Massie, and S. Craw, "WiFi-based human activity recognition using Raspberry Pi," in *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, 2020.