Journal of
*Sensor and
Actuator Networks*

MDPI

Article

A Bluetooth-Based Architecture for Contact Tracing in Healthcare Facilities

Piergiuseppe Di Marco [1,*], Pangun Park [2,*], Marco Pratesi [1] and Fortunato Santucci [1]

[1] DISIM, University of L'Aquila, 67100 L'Aquila, Italy; marco.pratesi@univaq.it (M.P.); fortunato.santucci@univaq.it (F.S.)
[2] Department of Information, Communications Engineering, Chungnam National University, Daejeon 34134, Korea
* Correspondence: piergiuseppe.dimarco@univaq.it (P.D.M.); pgpark@cnu.ac.kr (P.P.)

Abstract: With the latest standard releases, Bluetooth technology is becoming more and more relevant for building and industrial automation. At the same time, Bluetooth is now becoming fundamental for contact tracing applications, to support monitoring and containment of the COVID-19 pandemic. Critical facilities such as nursing homes and hospitals have been severely exposed to the pandemic, but the currently available short-range wireless technology still faces the fundamental limits of proximity accuracy, battery lifetime, and privacy in those complex indoor environments. The aim of this paper is to investigate effective ways of building an architecture with heterogeneous devices to support contact tracing in critical scenarios such as healthcare facilities, while meeting the required level of accuracy and privacy. A framework based on standard Bluetooth mesh networking technology is proposed, and the research challenges are discussed.

Keywords: bluetooth; mesh networking; contact tracing

check for
updates

Citation: Di Marco, P.; Park, P.; Pratesi, M.; Santucci, F. A Bluetooth-Based Architecture for Contact Tracing in Healthcare Facilities. *J. Sens. Actuator Netw.* **2021**, *10*, 2. https://doi.org/10.3390/jsan10010002

Received: 1 December 2020
Accepted: 26 December 2020
Published: 31 December 2020

1. Introduction

The COVID-19 pandemic is having tremendous consequences on our lives and the economy. As the world was unprepared for the first wave of the pandemic, the research community has focused on how technology can better support the containment of the disease. Contact tracing is an essential component to support the early identification of new cases among the population and to contain outbreaks of the disease. There is currently a great interest in the development of mobile apps to facilitate COVID-19 contact tracing. In April 2020, Apple and Google worked together to build an opt-in and decentralized way of allowing individuals to know if they have come into contact with confirmed cases based on Bluetooth technology [1]. Apps developed for contact tracing allow a handheld device to scan for other devices in the background, storing data locally. If one has tested positive for coronavirus, the user may authorize the data to be provided to the health authorities, trace others who happened to be in proximity (e.g., spending more than 15 min within a 1 m range from the person who tested positive), and notify about the risk of exposure.

The benefits of the use of contact tracing apps on a smartphone rely on the assumption that the majority of the population installs and uses the app regularly [2,3]. However, as of November 2020, the adoption and use of governmental contact tracing apps among the population is relatively low (i.e., around 15% of the population). The slow penetration rate of the technology is not only due to privacy, but also to the fundamental limitations of proximity detection accuracy. During the first wave of the COVID-19 pandemic, the Italian National Institute of Health (ISS) reported more than 40% of analyzed COVID-19 cases in Italy to be linked to nursing homes and 10% of cases were related to outbreaks in hospitals. Figure 1 shows the distribution of COVID-19 cases of Italy based on the dataset of the health authority. Protecting critical facilities with a targeted efficient contact tracing solution may provide great benefits for the containment of this pandemic and the prevention and control of future pandemics.

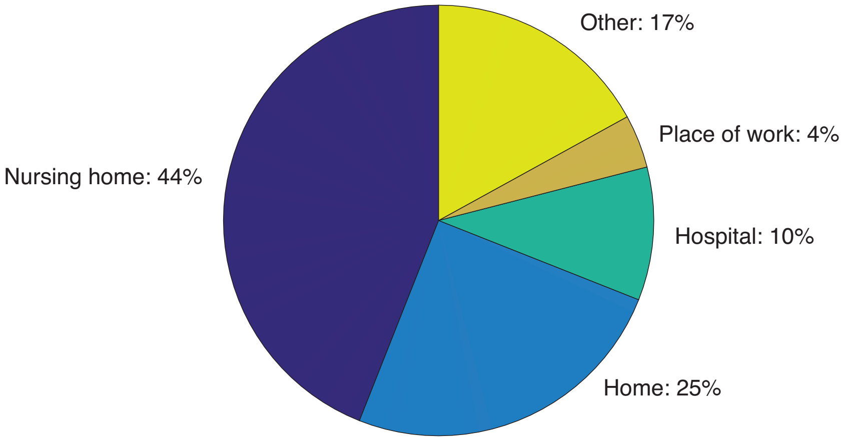

Figure 1. Distribution of COVID-19 cases in Italy as of April 2020, according to the Italian National Institute of Health (ISS).

Bluetooth mesh networking [4] offers a low cost, flexible, standard infrastructure to support contact tracing that is compatible with additional profiles and applications in use in hospitals (lighting, heating and ventilation control), without the need to deploy an ad-hoc infrastructure of devices only dedicated to the scope of monitoring the pandemic. The infrastructure can be additionally used for better care, including staff being promptly alerted when a patient falls or tries to move from the room against medical recommendations. In many emergencies, timeliness is essential, and location-based services can help staff keep track of assets and find the needed equipment faster.

In this paper, we first analyze the potentials and limitations of current Bluetooth-based contact tracing solutions. Then, we propose a framework for building a Bluetooth mesh standard-based infrastructure to support the contact tracing services for critical healthcare facilities. The infrastructure relies on a hierarchical architecture and standardized protocols for exchanging proximity data instead of a typical peer-to-peer fashion using smartphone apps. We show the results of experimental campaigns and simulations to validate the proposed solution, then suggest standardization guidelines and address research challenges to promote the adoption of such an architecture.

The rest of the paper is organized as follows. Section 2 provides an overview of Bluetooth technology and the Bluetooth mesh networking standard. In Section 3, we describe the contact tracing scenario. We survey the technical challenges associated with the existing contact tracing framework in Section 4 and describe the proposed approach for contact tracing in Section 5. Simulation results are presented in Section 6. Finally, we provide standardization directions in Section 7 and summarize our conclusions and future work in Section 8.

2. Bluetooth Technology Overview

In this section, we provide an overview of Bluetooth technology, distinguishing the core specification [5] and the mesh networking specification [4].

2.1. Core Specification

The major use case for Bluetooth has for a long time been the wireless connection between a mobile phone and a headset. Bluetooth Low Energy, which was released in 2010 as part of the Bluetooth 4 radio specification, represented an effective step to expand the ecosystem of Bluetooth to the Internet of Things. Bluetooth technology is seen as the key technology for contact tracing apps, thanks to the large ecosystem of devices. Bluetooth supports connection-oriented and connection-less data transfer modes. In the former mode, devices negotiate dedicated resources for data communication. In the latter mode, denoted as advertising mode, short messages (advertising packets or beacon messages) are transmitted directly over random access channels and can be used to support the quick estimation of the position of a device, using the received signal strength indicator (RSSI) at the receiving side. Each advertising packet consists of a short fixed header, followed by a small payload. The payload contains an identifier of the device making the broadcast plus a short message (generally up to 31 bytes long). This advertising payload is typically used to indicate that the packet is associated with a particular app or service, e.g., to associate it with a contact tracing app. Bluetooth technology provides better accuracy in determining the proximity if compared to GPS or cellular-based localization. For these reasons, most of the apps developed today use proximity data provided by the Bluetooth radio in a mobile phone to support contact tracing services.

As a recent advancement of Bluetooth core technology, the release of the Bluetooth 5.1 specification in 2019 introduced enhanced localization services, with the so-called direction-finding feature. Combined with enhanced broadcasting capabilities and data transfer modes [6], a boost in the adoption and deployment of Bluetooth beacons and location-based services is introduced. Moreover, a modified connection-less approach, denoted as the advertising extension, has been introduced in Bluetooth 5 to enhance advertising and beaconing by the use of additional channels previously not utilized for regular advertising. The increased data rates, in addition to the rather obvious benefits of increased throughput and reduced latency, decreases the time on air. This, in turn, saves the battery lifetime and improves the coexistence with other technologies, such as Wi-Fi, that operate in the same band and often coexist within the same device [7,8].

The Bluetooth security architecture includes five distinct features: pairing, bonding, device authentication, encryption, and message integrity; this is designed to provide protection against passive eavesdropping and protection against man-in-the-middle attacks. The standardized full stack architecture guarantees strong protection at the device level, although there is no user authentication [9]. Furthermore, Bluetooth provides privacy by supporting a feature that reduces the ability to track a device over a period of time by changing the Bluetooth device address on a frequent basis.

2.2. Mesh Networking Specification

Bluetooth mesh introduced the networking specification on top of Bluetooth technology to take additional market shares in the Internet of Things (IoT) connectivity space. Formally, Bluetooth mesh is defined as a profile that can run on top of any device compatible with Bluetooth Low Energy. Bluetooth profiles define the required functions and features of each layer in the Bluetooth stack from the physical layer to the application. A profile defines the vertical interactions between the layers, as well as the peer-to-peer interactions of specific layers between devices forming a mesh network.

The design of the Bluetooth mesh profile aims at creating a simple, efficient, and flexible wireless mesh networking protocol solution. The Bluetooth mesh profile standardizes a layered protocol architecture, as illustrated in Figure 2. Each layer has its own functions and responsibilities and provides services to the layer above. The access to the radio is handled by the Bluetooth core technology, with the advertising data format defined by the bearer layer. The network and transport layers are functional for the network design and strategies for deployment. The network layer handles aspects such as addressing and relaying of messages, as well as encryption and authentication within the network. The lower transport layer handles segmentation and reassembly and provides acknowledged or unacknowledged transport of data end-to-end. The upper transport layer encrypts and authenticates access messages and defines transport control mechanisms including the management of low power nodes. The access layer is responsible for application-to-network traffic management and end-to-end data transfer. An application interacts with the protocol stack via standardized models, which define the message format and configuration of the application parameters [4].

With the introduction of Bluetooth mesh, potentially thousands of nodes can interact with each other without establishing a connection. After provisioning the devices in the network, there is no need for centralized algorithms, nor coordination, and most importantly, there is no single point of failure in the network. A group of nodes can be addressed with a single command, so that the dissemination and collection of information are efficient and reliable. Devices that need low power support can associate themselves with an always-on device that stores and relays messages on their behalf, using the so-called friendship. Friendship is a special relationship between a low power node and a neighboring "friend" node. Friendship is established by the low power node, but, once established, the friend node is responsible for taking actions that help with the reduction of the power consumption for the low power node. The friend node holds a cache that stores incoming messages destined to the low power node and, upon request, delivers those messages to the low power node. In addition, the friend node delivers security updates to the low power node.

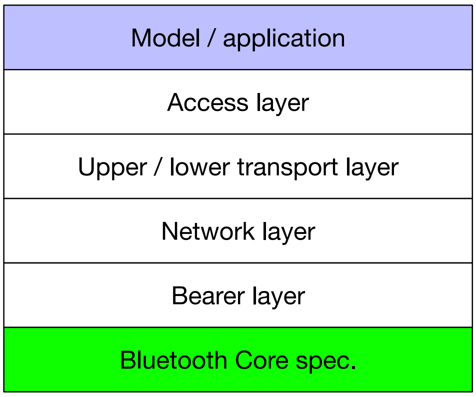

Figure 2. Bluetooth mesh layered architecture.

Currently, Bluetooth mesh is supported for a smartphone only through proxy connections via a dedicated node of the network, which is called the proxy and takes the responsibility of injecting and collecting traffic in the network from and towards the smartphone. Determining the correct number of proxies in a network, the number of relays, and efficient solutions for interference mitigation is a challenging task that is under discussion in the research community and standardization bodies. The performance of Bluetooth mesh has been evaluated though simulations in [10,11] and through a set of experimental campaigns in [12].

3. Contact Tracing Scenario

A generic contact tracing scenario in the context of pandemic containment is as follows. A person that has tested positive to COVID-19 needs to inform the public health authorities about all people that have been in close contact in the previous 14 days and that are at risk of being infected. There are no definitive studies on the exposure and transmission of the disease, and it is up to public health authorities to define what a close contact means. The definition depends on the context where the potential exposure happens and the use of personal protective equipment (PPE). As a general rule, permanence within proximity for more than 15 min is considered as a contact at risk.

The contact tracing scenario defined in this study aims at complying with the most recent guidelines from the WHO, in particular when referring to healthcare environments [13]. In such a scenario, the WHO considers persons at risk anyone within 1 m of a COVID-19 patient in any part of the healthcare facility for more than 15 min. Additionally, the hospital staff in direct contact with a COVID-19 patient, where the use of PPE has failed, is considered at risk no matter the duration of the contact. Any patient hospitalized in the same room or sharing the same bathroom as a COVID-19 patient, visitors of the patient, or other patients in the same room or other rooms visited by the COVID-19 patient (e.g., common dining facilities) should be listed as a potential person at risk. It is evident that the requirements are much stricter than in the general case for contact tracing, as it is not sufficient to perform only seldom measurements in a time frame of 15 min, as current exposure notification services do [1], but many critical interactions that depend on location (people in the same room) need to be captured more frequently.

The isolation of patients that have tested positive for COVID-19 in restricted areas follows strict protocols in most of the hospitals and healthcare facilities. However, a critical scenario for contact tracing and exposure notification is in the non-COVID areas and grey areas, where patients that are not considered at risk later result in being positive.

Bluetooth technology is playing a pivotal role to ease the contact tracing procedure. Various devices can read and store anonymous identifiers associated with other users in the proximity area using the Bluetooth technology. If a patient has tested positive to COVID-19, this information can be shared with public authorities, and the users that have been in close contact with the infected person can receive an exposure notification and are invited to perform the test. After a number of days from the contact, the data are removed from the servers.

4. Technical Challenges

There are major concerns about the feasibility of reliable and robust contact tracing solutions based on Bluetooth technology. The main challenges are classified into four categories, namely proximity accuracy, device resources, privacy, and technology penetration rate.

4.1. Proximity Accuracy

Distance estimation based on RSSI measurements is motivated by conceptual simplicity and by the spread of suitable devices. The usual variability of wireless channel behavior is not particularly marked for line-of-sight short-range distance estimation; however, the attenuation may deviate from free space due to many environmental factors. Multipath may have a strong influence, especially in closed environments with reflecting surfaces; on the other hand, averaging over frequent measurements should be a solution in most cases. Other inconveniences may arise from propagation without direct or clear electromagnetic visibility. As an example, a smartphone may be in a pocket, with the human body obstructing direct visibility; the smartphone itself may be protected by a more or less robust cover that can also be built using non-plastic materials. Furthermore, a Bluetooth chip can be employed by devices with different kinds of body/shell materials: as an example, aluminum frames involve a larger electromagnetic attenuation than plastic materials. Moreover, barrier protections and PPE drastically reduce the risk of contagion, but their impact on proximity measurements is not easy to account for.

In [14], the authors showed the results of an experimental study to validate the impact of various factors such as distance, orientation, obstacles, and time of the day on the Bluetooth RSSI measurements, showing that a limited number of RSSI measurements is not sufficient for the purpose of accurate distance estimation. In [15], the authors studied the impact of channels, distances, and user’s orientation in the positioning phase on the accuracy of Bluetooth positioning. The accuracy obtained was within 1.5 m, and the precision was 90% within the range of 1.5–2.5 m.

Radio frequency fingerprinting in support of RSSI measurements was proposed in [16] as a means to enhance the positioning accuracy of Bluetooth. Reference [17] analyzed the performance of Bluetooth beacons in terms of their accuracy in proximity estimation. Three Bayesian filtering techniques to improve the BLE beacon proximity estimation accuracy were analyzed. With this approach, the achieved proximity error range was 0.27 m at a distance of 3 m, using 1000 RSSI measurements. In [18], indoor localization measurements were conducted using Bluetooth mesh device implementations. The study concluded that a careful design of the network and interference management are important as they affect the positioning accuracy of existing algorithms. While several machine learning techniques have been proposed to improve proximity accuracy, recent works still show the fundamental boundaries of the proximity estimation of off-the-shelf devices [19,20].

4.2. Device Resources

Transmitting short beacon messages periodically (say, every 250 ms) to enable contact tracing application is not a problem even for a resource-constrained device. A device with Bluetooth connectivity enabled typically advertises its supported services roughly every 100 ms with little impact on the resources (i.e., battery, computation capabilities) [5]. However, in order to perform proximity measurements, each device needs to keep its Bluetooth receiver continuously on for sufficiently long time to collect unsolicited beacon messages from peer devices. Methods to optimize the scanning cycle of Bluetooth devices have been proposed in the literature (e.g., [21]), but this is not always possible in the case of battery powered devices such as smartphones since radio resources are shared among the various active services, including active Bluetooth connections, and even the use of Wi-Fi connectivity impacts the availability of the Bluetooth receiver [6].

When turned on, a Bluetooth receiver consumes approximately 15 mA of current. If continuously active during 24 h, it depletes about 12% of charge in a typical 3000 mAh battery in a smartphone. Indeed, apps running in the background on a smartphone are typically allowed to scan for incoming Bluetooth beacons only for a few seconds every 5 min [1]. Therefore, when compared to devices that have a duty cycle close to 100% (e.g., relay nodes in a Bluetooth mesh), a smartphone may take more than 10 times to collect the same number of RSSI measurements for use in proximity detection and contact tracing.

4.3. Privacy

Contact tracing services handle critical data such as medical condition and potentially the position of a user at any time. The user needs to trust the central authority and the owner of the contact tracing service to preserve the privacy of the information, as the system can be in theory always misused for mass surveillance or unclear scopes, beyond the original purpose of pandemic containment. Indeed, a relevant problem for some of the contact tracing apps available for our smartphones is third-party information sharing [22]. Several contact tracing apps analyzed by [22] mention outsourcing data to third parties, and it is not clear what data are shared with whom and and how they are processed by these parties.

Both centralized and decentralized approaches have been proposed for privacy-preserving contact tracing. The recent Google/Apple solution implements a decentralized approach, where each device protects its identity by using random temporary exposure keys and encrypted identifiers that are frequently refreshed [1]. Theoretically, nobody except the user can easily understand if two temporary codes have been transmitted by the same device, not even the central authority. Each user connects periodically to the server and receives the entire list of infected users. At that point, the user's device can securely check if this list contains one of the previously observed random identifiers and notify the user if at risk.

Although governmental contact tracing apps are designed to guarantee a certain level of privacy, there can be intentional or unintentional risks for infected users, resulting from the fact that positive users reveal all the codes transmitted to the server during the period in which they are presumed to have contracted the virus. This potentially allows the server or a malicious attacker to make the person identifiable. This is especially problematic if the implementation of the framework is not open-source. To overcome this limitation, distributed approaches for preserving privacy are investigated in the research community. Blockchain is currently evaluated as a privacy-preserving method for data logging and retrieval (e.g., as described in [23]), but the impact in terms of cost and computation for these solutions is yet to be determined in consideration of the scale of the application.

4.4. Technology Penetration Rate

The technology penetration rate is a critical factor since a contact tracing service requires that the majority (60%) of the population must have the underlying technology with regular usage of the app [2]. Bluetooth technology has a great potential, as it is available in all modern smartphones and there is a large ecosystem of compatible devices. Ninety-five percent of people in South Korea own a modern smartphone. However, in Europe and the U.S., the percentage varies from 60% to 80% of the population, and the number reduces drastically below 40% in developing countries. These percentages would be in theory still good for the success of contact tracing technologies. However, there is some skepticism around the potential success of contact tracing solutions among the general public, whenever it is not enforced, and it is only partially attributable to the technical limitations discussed above. Indeed, the adoption and use of governmental contact tracing apps among the population is surprisingly low across the world (i.e., typically below 15% of the population). This fact encourages a more focused effort and targeted use of the technology in key areas, where the containment of the pandemic is critical and the use of the technology can be enforced.

5. Contact Tracing Architecture

In this section, we describe the system architecture and deployment guidelines for building a Bluetooth mesh standard-based infrastructure to support the contact tracing services for critical healthcare facilities.

5.1. System Architecture and Devices

The main elements of the proposed system architecture for contact tracing are illustrated in Figure 3 with reference to the Bluetooth mesh profile protocol stack [4]. We define four different roles of the devices in the architecture, depending on the functionality and resource capabilities:

- Beaconing tag: A beaconing tag is a relatively simple device that should be tracked using the minimum amount of energy possible to provide the longest battery life. Beaconing tags should work for years without having to be recharged or have a battery replaced. To implement a beaconing tag, only the model associated with the contact tracing apps is required above the underlying Bluetooth core implementation, and there is no need to implement states and behaviors associated with the mesh profile stack.
- Proximity detector: A proximity detector is a device that is configured to monitor and send messages based on the presence, absence, or change in the position of a beaconing tag. The proximity detector needs to have resources to scan the advertising channels regularly and perform mesh procedures, thus implementing the full stack, including the model associated with the contact tracing app. The states and behaviors associated with the relay feature standardized in Bluetooth mesh are not mandatory to implement.
- Mesh relay: A relay node forwards messages received from other nodes over the advertising bearer. It is the equivalent role of a generic relay in the Bluetooth mesh standard. Therefore, it is provided with the network security credentials associated with the mesh network, but it does not need to implement the higher layers of the Bluetooth mesh protocol stack and does not need to be provided with the security credentials associated with the contact tracing application in order to participate in the traffic forwarding.
- Gateway: The gateway node is a generic mesh node that provides an interface with the contact tracing server, where the advanced computation of the proximity data is performed and relevant data are stored. The gateway needs to be provided with an implementation of the model associated with the contact tracing apps and support APIs to a communication infrastructure with the contact tracing server.

Since we consider healthcare facilities such as nursing homes and hospitals, it is envisioned that the use of beaconing tags is enforced for doctors and patients. Proximity detectors and the gateway may be smartphones and existing Bluetooth mesh-capable devices deployed within the premises, where the contact tracing model is installed. Mesh relays can be existing Bluetooth mesh relay-capable devices deployed within the premises, with no need for software upgrades.

The main advantage of implementing such an architecture in healthcare facilities such as nursing homes and hospitals, compared to the general purpose scenario, is that the use of devices such as beaconing tags and proximity detectors can be imposed on personnel and patients, and potentially registered visitors, via internal regulation or by the local authority, so that the adoption of the technology is enforced. Privacy aspects and its implication are easier to handle in restricted scenarios for contact tracing in healthcare facilities compared to other scenarios, since access and the presence of staff and patients are already monitored and visitors may need to register to access the premises. To improve accuracy and robustness to the presence of obstacles, patients and users may be provided with instructions on how to properly carry the tag. As an example, a tag carried as a visible necklace (or a wristband) preserves high correlation between electromagnetic visibility and face-to-face contact, which is a situation at risk. Proximity accuracy and the use of resources can be addressed by opportunely configuring the protocols of the stack, as we illustrate in the next section.

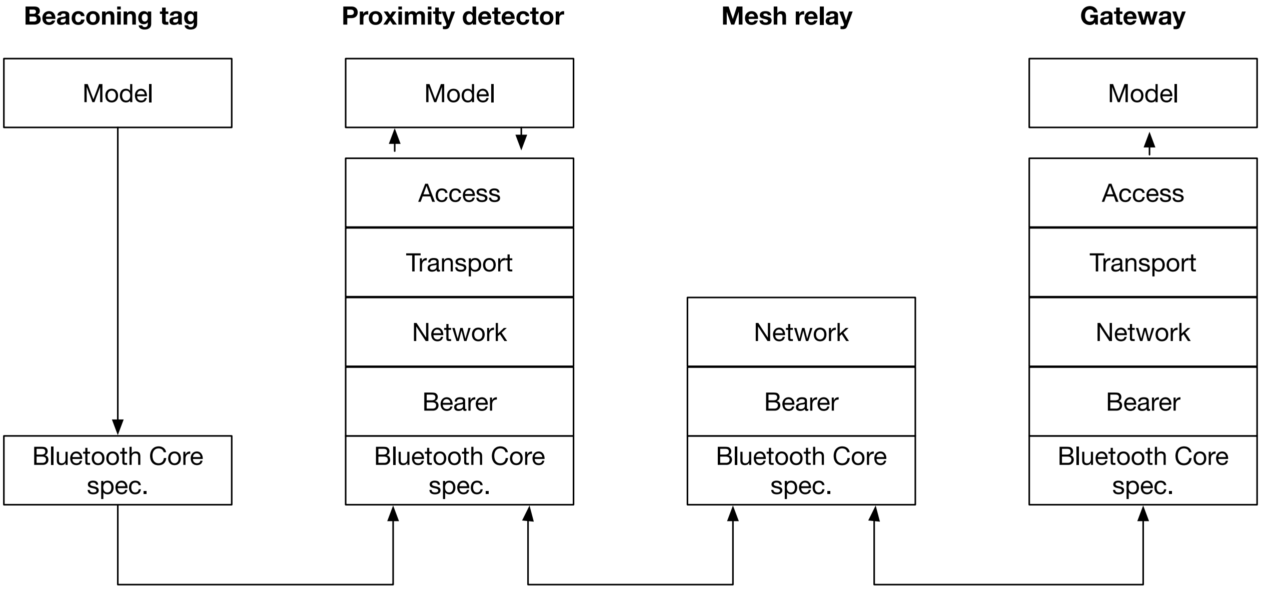

Figure 3. System architecture for the proposed Bluetooth mesh infrastructure for contact tracing.

5.2. Protocol Setup and Configuration

An example of a deployment scenario on a reference topology is illustrated in Figure 4. Beaconing tags can be transmit-only devices. This implies that they may not be able to receive regular network security updates as required for a generic mesh node. Therefore, each beaconing tag needs its own separate security credentials defined in the model. Some of the nodes that have a Bluetooth receiver with a low duty cycle may be configured as proximity detectors, as they can read and process messages from beacons in their surrounding proximity area. Other nodes with a high duty cycle can be instantiated as proximity detectors with respect to surrounding beaconing tags, but also implement the relay feature of Bluetooth mesh to forward data received by proximity detectors and other mesh nodes in the network. This infrastructure is integrated with legacy Bluetooth mesh nodes that only serve as relay nodes carrying traffic towards the contact tracing engine, but do not perform proximity detection and do not need to implement any application associated with the contact tracing application.

With reference to the hospital facility use case, people admitted in the hospital facility may be provided with a beaconing tag (e.g., a transmit-only wristband) that is temporarily associated with their social security number or mobile number. Personnel in the hospital may be given a beacon or a smartphone that behaves as a proximity detector. Bluetooth mesh enabled lights and other sensors/gates in the hospital may behave as proximity detector and/or relay nodes. Without the requirement of complete anonymity, different risk profiles can be attributed to people, based on role, performed activity, diligence and expertise in usage of PPE, and previous clinical conditions.

The beaconing tags are programmed to transmit a short advertising message every 250 ms. The frequency is in line with the current recommendation for notification exposure [1]. Each proximity detector continually reports back to the proximity engine all tags it can hear above a predefined RSSI threshold, as well as the RSSI measurement from each. The proximity engine uses that information, as well as the known position of each proximity detector, to estimate the position of tags based on trilateration. To be able to reuse this architecture for different situations and to identify more accurately hazardous exposures, the proposed framework can be contextualized to the specific environment, including the presence of physical barriers (e.g., plexiglass panels) and the use of PPE.

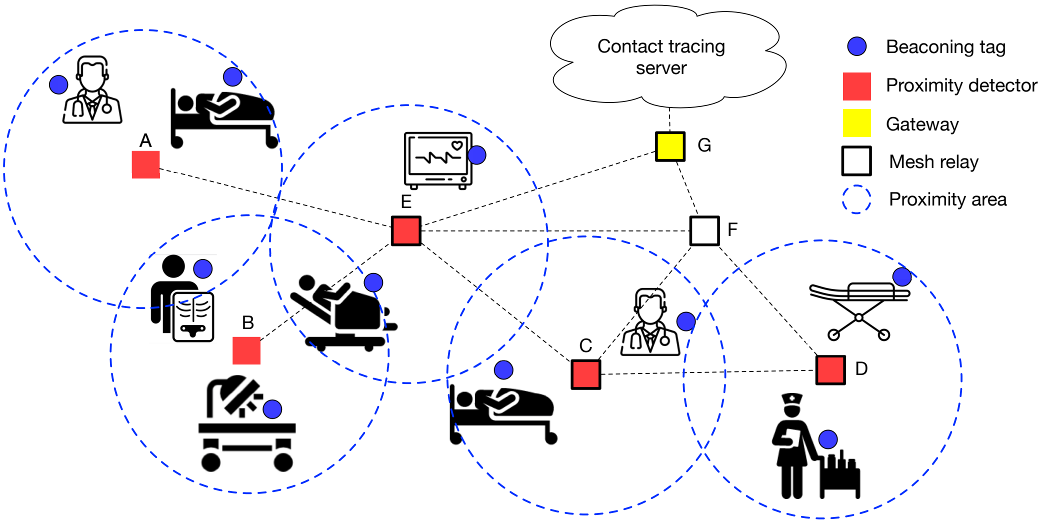

Figure 4. Bluetooth mesh infrastructure with the example topology. Beacons are identified with small colored circles. Nodes A and B are simple proximity detector, as they can read and process messages from beacons in their surrounding proximity area, which is limited by the dotted circles. Nodes C, D, and E behave as proximity detectors with respect to surrounding beacons, but also implement the relay feature of Bluetooth mesh to forward data received by the proximity detector and other mesh nodes in the network. Nodes F and G only serve as the relay node and the gateway, respectively, carrying traffic toward the contact tracing engine, but do not perform proximity detection.

The dependence between RSSI measurements and distances can be tuned accounting for the peculiarities of the environment, through deductive/deterministic methods, or through machine learning techniques. Safe distance and risky exposure time can be specifically identified for each room/environment, based on parameters such as the availability of forced ventilation and the density of people. The whole system could even be refined and enhanced by data fusion techniques, e.g., using surveillance cameras to identify protective dividers, the number of persons within a room and their usage of PPE, and so on. Moving forward from the approach followed by the Google/Apple solution, we envision that the RSSI threshold and the advertising interval can be configuration parameters that are set by the configuration manager. As we show in the next section, this is a critical aspect that affects the feasibility of contact tracing solutions.

6. Results

We conducted a measurement campaign using the nRF Connect for mobile app [24], running on iOS smartphones, one acting as a beaconing tag and one acting as a proximity detector, placed at various distances. The advertising interval was set to 250 ms, and the total scanning period was set from 15 to 60 min. Furthermore, to account for multiple access interference given for the presence of multiple beaconing tags and proximity detectors in the same network, we considered various levels of advertising packet loss probability.

In Figure 5, we report a representative sample of the measurement campaign for a beaconing tag placed at 0.5, 1, 1.5, and 2 m from the proximity detector. The result of the measurement is in line with the conclusions from previous studies that the RSSI value at a fixed distance can vary significantly, up to around 10 dB of difference [19]. A Bluetooth receiver scans the three advertising channels one at a time, and there is no coordination between the transmitter and the receiver in connectionless mode. In this way, the RSSI measurement can be taken at any of the three advertising channels. Part of the variability in the measurement is then due to different radio propagation conditions in the advertising channels. In addition, the radio environment may change during the measurement period. Figure 5 shows that a limited number of transient RSSI measurements is neither sufficient nor accurate for distance estimation. However, there is a significant correlation between the presence of a proximity detector within 1 m of distance and the long-term average RSSI above a certain value.

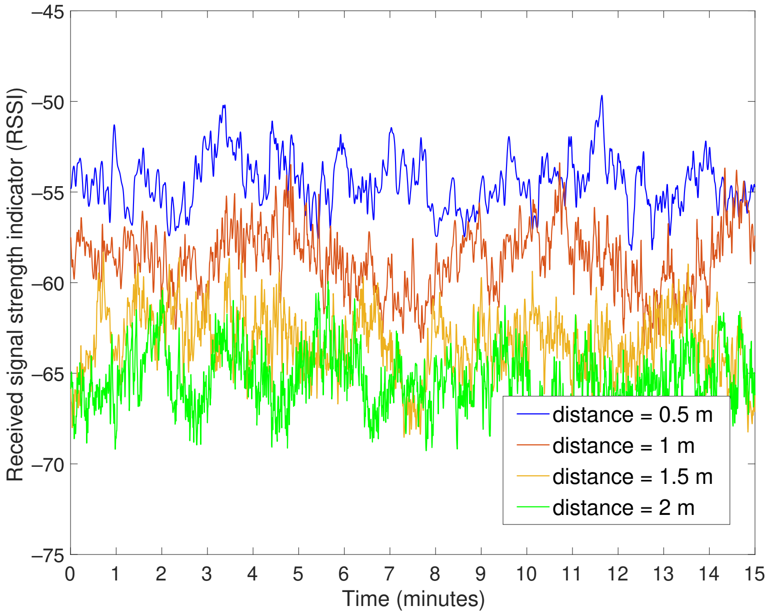

Figure 5. Bluetooth measurements in a static indoor scenario at various distances.

In Figures 6 and 7, we report the probability of exposure notification. The exposure notification probability was obtained by analyzing the RSSI measurement trace of nRF Connect and computed as the duration of the RSSI levels above the threshold for a period of at least 15 min within the observation. Missed RSSI measurements were not considered in the computation. The observation period in Figures 6 and 7 varies from 15 min to 60 min. The probability is reported for various values of the threshold in the interval from −54 dBm to −66 dBm, for a distance between the beaconing tag and the proximity detector of 1 m in Figure 6 and 2 m in Figure 7. For a given threshold, by looking at Figure 6, we can observe both the probability of the correct notification of an exposure (true positive) and its complement to one that measures the probability of missing the notification of exposure for a beaconing tag that is in proximity (false negative). By looking at Figure 7, we can observe the probability of the notification of exposure triggered by a beaconing tag that is actually not in proximity (false positive) and its complement to one that measures the probability that the exposure is not notified (true negative).

The probability of a false detection of proximity, which triggers an exposure notification for a device that has not been actually in close contact, needs to be low to avoid a person being put in isolation and tested based on inaccurate information. However, it is more critical to have a device that has been in close contact that does not exhibit long-term average RSSI capable of triggering an exposure notification (false negative), as this is dangerous for the containment of the pandemic. Based on our measurement campaign, a threshold at −63 dBm guarantees a low probability of false positive events and false negative events in the scenario evaluated and can be used as a default value set by the Bluetooth model.

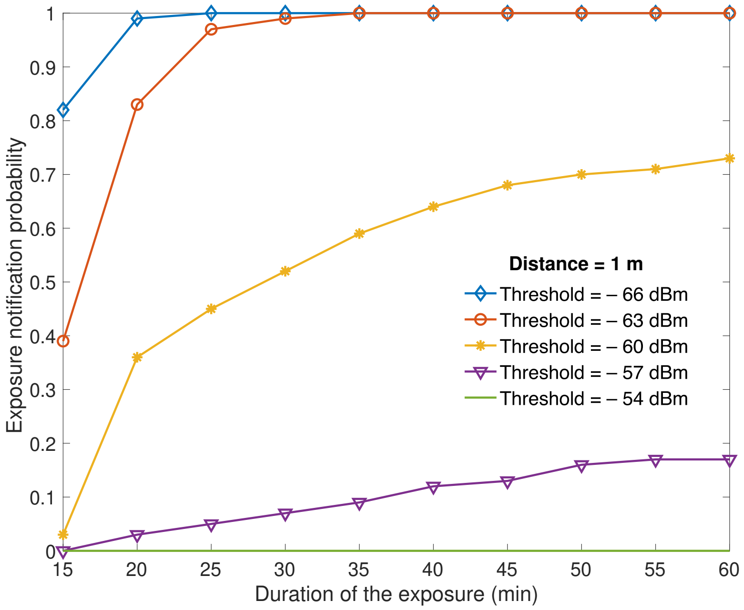

Figure 6. The probability of exposure notification vs. the duration of the exposure at a proximity detector for a beaconing tag at a distance of 1 m.

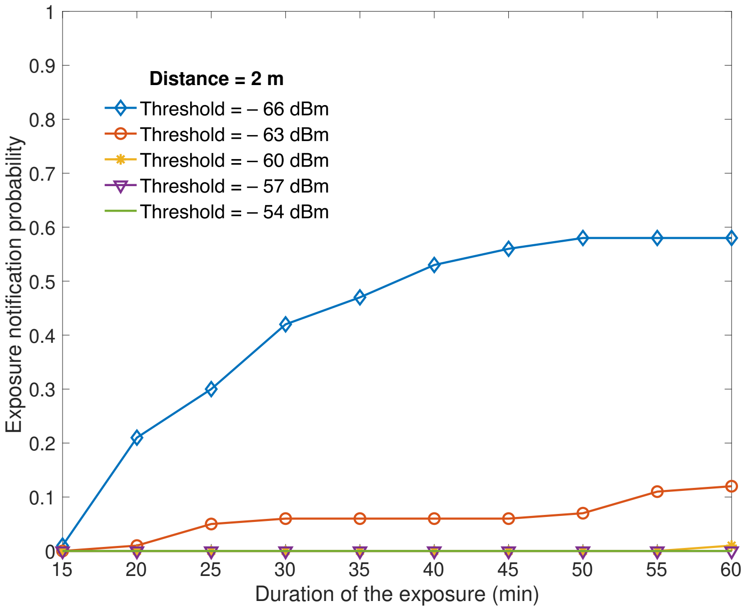

Figure 7. The probability of exposure notification vs. the duration of the exposure at a proximity detector for a beaconing tag at a distance of 2 m.

The range of thresholds presented in Figures 6 and 7 was chosen based on the results of a preliminary measurement campaign, considering only values that are significant for the contact tracing scenario. Indeed, an RSSI threshold lower than −66 dBm gives a significant probability that a device that is farther than 2 m is detected as an exposure, whereas on the contrary, a threshold above −54 dBm gives a very low probability of correct exposure notification. A high sensitivity of the results with respect to the threshold is also observed in Figures 6 and 7, which corresponds to a sensitivity to inaccurate RSSI estimation and scenario-dependent bias on the RSSI values. A 3 dB constant bias in the measurement of the received power at a given distance may increase the probability of false negatives within a 30 min observation period from 2% to 48%. Therefore, it is fundamental that the threshold can be changed by the configuration manager based on the knowledge of the deployment area. Walls can affect proximity measurements by 6–12 dB. However, such additional path loss is not negative for proximity measurements as close contacts with a wall in between should not be counted.

While a higher sampling rate in transmission and a long scanning period in reception may provide high proximity accuracy, they considerably increase the cost of in terms of energy consumption. Hence, there is a clear fundamental tradeoff between proximity accuracy and device resources depending on the sampling rate and scanning cycles. One of the main advantages of the proposed asymmetric architecture with a combination of beaconing tags and a network of more proximity detectors is that it can balance proximity accuracy and the battery lifetime of constrained devices. Repeated measurements within an interval give high accuracy in the detection of close contacts. If the transmission interval is 250 ms and a smartphone can scan for 30 s every 5 min, the proximity detection can be based on 360 RSSI measurements. A mesh node, such as a monitor, can scan the advertising channels with a 100% duty cycle, except when the device is occupied in re-transmitting messages. Therefore, there are potentially 3600 measurements available at a proximity detector in 15 min if scanning continuously. Considering 15 mA of current consumption, a rechargeable proximity detector can run full-day operations, with the appropriate battery size. As a novel paradigm that we introduce with our scheme, compared to the peer-to-peer smartphone-based approach, we have an asymmetric architecture, with beaconing tags that are transmit-only devices and can be provided with button-cell batteries, so that they can be produced in a scalable way at low cost.

The number of proximity detectors needs to be sufficient to detect beaconing tags at any location within a few meters of distance, to perform accurate distance estimation. Therefore, in an indoor healthcare facility, depending on the geometry of the map, the recommended density of proximity detectors is one node every 10 m^2. The number of relay nodes may be much lower, as it is sufficient that every node can be reached via Bluetooth advertising. Therefore, depending on the geometry of the map, the recommended density of the proximity detector is one node every 50 m^2.

To validate the claims in a general mesh scenario with multiple access interference, we include an additional advertising loss probability that is applied to the RSSI measurement trace obtained with the experimental campaign. In Figure 8, we report the probability of exposure notification for a duration of the exposure of 20 min and a distance of 1 m, for increasing advertising loss probability up to 20%. It can be seen that the exposure notification probability is rather insensitive to the additional packet loss probability, in particular for typical values of the advertising loss probability in the context of Bluetooth mesh networks. In fact, although the broadcasting nature of Bluetooth mesh may suggest that multiple access interference is an issue for the reliability of the advertising data transfer, especially in a large-scale scenario, this is not entirely correct for a properly configured mesh network, as experimentally validated for an 879 device building automation scenario in [11], with more than 99% successful data transfers. This is due to the short duration of mesh advertising packet transmissions (around 300 μs) and the use of channel and spatial redundancy [5]. Indeed, with an advertising interval of 250 ms, the channel occupation for a beaconing tag is about 0.12%.

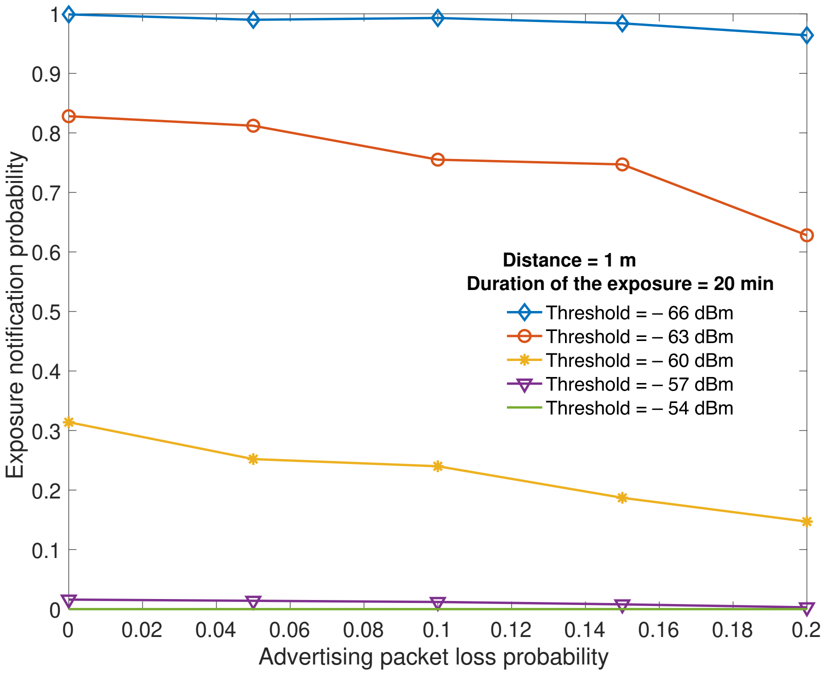

Figure 8. Probability of exposure notification vs. advertising loss probability at a proximity detector for a distance of 1 m and 20 min of exposure.

7. Standardization Directions

The Bluetooth mesh networking specifications were published in 2017 and first updated in 2019. In order to participate in a Bluetooth mesh network, devices need to implement standard or vendor specific models, which define the basic functionality of a node, including states, messages, and behaviors. Currently, most of the certified products are associated with the home automation domain, in particular lighting applications. Lighting has two important attributes to leverage as the tracking infrastructure: the availability of power and the fixed location. All lighting fixtures are powered by mains power, which in a hospital includes emergency back-up. Moreover, they are mounted at fixed known positions.

For single-hop broadcasting, Bluetooth core supports the so-called proximity profile [5], which defines the behavior when a device moves away from a peer device so that the connection is dropped or the path loss increases above a predefined level, causing an alert. The proximity profile can also be used to define the behavior when the two devices come closer together, but a connection is required to perform the RSSI measurements associated with the profile. This is not sufficient to cover the use case proposed in this paper. Mesh specifications extend Bluetooth core by providing a set of functionalities and features that handle multi-hop data transfer, end-to-end secure transport, and management of heterogeneous devices in the same network. The full-stack standardization process guarantees interoperability among devices, which is not possible with solutions that are developed on top of the core specifications (e.g., Apple iBeacon). The availability of standard profiles and interoperable networking operations is critical for the success of contact tracing applications.

There is not yet an available model for Bluetooth mesh that supports proximity detection of transmit-only devices, and this is mainly due to the existing security architecture of Bluetooth mesh that requires nodes to be listening when security updates are disseminated in the network. The framework proposed in this paper can provide relevant use case requirements for specifying a suitable model that covers application in proximity detection and contact tracing in critical facilities. In particular, the concept that introduces the role of the proximity detector that is separated from the beaconing tag role can be standardized, with the possibility to separate the security domains between the beaconing tag and the proximity detector and the proximity detector and the rest of the mesh.

The great advantage of exploiting a standardized solution is that the facility manager will not be forced to deploy a full-sized infrastructure dedicated only to contact tracing and proximity detection, but it can reuse existing Bluetooth devices already deployed in the infrastructure as the proximity monitor and can deploy Bluetooth mesh devices such as lights and sensors that can be used for other purposes beyond the contact tracing needs, which will hopefully not be necessary for a long time. Applications would even be able to perform different risk assessments for different pathologies, without being forced to rely on a "black-box" framework, specifically designed for only one type of risk, whose parameters cannot be set by applications built on top of it. Furthermore, a deeper understanding of the mechanisms of contagion might involve updates in the definition of exposure to risk, and in the future, profiles may be updated considering different mechanisms of contagion, eventually not only related to some safety distance and exposure time.

8. Conclusions

Bluetooth is a global short-range radio communications standard with a large ecosystem and with strong potential for the success of contact tracing applications. There are ongoing discussions on the effectiveness of contact tracing in large-scale scenarios, and concerns have emerged about the technology penetration, proximity accuracy, efficient use of device resources, and privacy of such solutions. However, critical facilities such as hospitals and nursing homes can benefit from the deployment of simple, flexible, cost-effective, local infrastructures to support contact tracing, especially if based on a standardized networking solution.

We propose a system architecture for contact tracing based on the Bluetooth mesh standard and addressed challenges and opportunities for its adoption in critical facilities. We show through experiments the sensitivity of the exposure notification service to the RSSI threshold and duration of the exposure, showing that this is a critical component compared to the multiple access interference that is present in a dense mesh network. As a result, we advocate for standardizing Bluetooth mesh models that include configurable parameters for contact tracing purposes. As future work, we plan to extend the test bed for the evaluation of the proposed system architecture in large-scale scenarios.

Author Contributions: Conceptualization, P.D.M.; data curation, P.D.M.; investigation, P.P. and M.P.; methodology, P.D.M. and P.P.; supervision, F.S.; writing original draft, P.D.M.; review & editing, P.P., M.P. and F.S. All authors have read and agreed to the published version of the manuscript.

Funding: The work of P. Park was supported in part by the National Research Foundation under Grant NRF-2019R1H1A1101803 and in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) under Grant No. 2020-0-00187 funded by the Korea government. The work of P. Di Marco was partially supported by the PON Ricerca & Innovazione 2014-2020 (AIM 1877124). The work of M. Pratesi and F. Santucci was supported by the RIA2020 project at the University of L'Aquila and by the Centre of EXcellence on Connected, Geo-Localized and Cybersecure Vehicles (EX-Emerge), funded by the Italian Government under CIPE resolution n. 70/2017 (Aug. 7, 2017).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Apple and Google. Exposure Notification—Bluetooth Specification v1.2. 2020. Available online: https://www.blog.google/documents/62/Exposure_Notification_-_Bluetooth_Specification_v1.2.pdf (accessed on 1 December 2020).
2. Ferretti, L.; Wymant, C.; Kendall, M.; Zhao, L.; Nurtay, A.; Abeler-Dörner, L.; Parker, M.; Bonsall, D.; Fraser, C. Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing. *Science* **2020**, *368*, eabb6936. [CrossRef] [PubMed]

3. Hernández-Orallo, E.; Manzoni, P.; Calafate, C.T.; Cano, J. Evaluating How Smartphone Contact Tracing Technology Can Reduce the Spread of Infectious Diseases: The Case of COVID-19. *IEEE Access* **2020**, *8*, 99083–99097. [CrossRef]
4. Bluetooth SIG. Bluetooth Mesh Profile Specifications 1.0.1. 2017. Available online: https://www.bluetooth.com/specifications/mesh-specifications/ (accessed on 1 December 2020).
5. Bluetooth SIG. Bluetooth Core Specification 5.2. 2019. Available online: https://www.bluetooth.com/specifications/bluetooth-core-specification/ (accessed on 1 December 2020).
6. Di Marco, P.; Skillermark, P.; Larmo, A.; Arvidson, P.; Chirikov, R. Performance Evaluation of the Data Transfer Modes in Bluetooth 5. *IEEE Commun. Stand. Mag.* **2017**, *1*, 92–97. [CrossRef]
7. Collotta, M.; Pau, G.; Talty, T.; Tonguz, O.K. Bluetooth 5: A Concrete Step Forward toward the IoT. *IEEE Commun. Mag.* **2018**, *56*, 125–131. [CrossRef]
8. Di Marco, P.; Chirikov, R.; Amin, P.; Militano, F. Coverage analysis of Bluetooth low energy and IEEE 802.11ah for office scenario. In Proceedings of the 2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Hong Kong, China, 30 August–2 September 2015; pp. 2283–2287.
9. Lonzetta, A.; Cope, P.; Campbell, J.; Mohd, B.; Hayajneh, T. Security Vulnerabilities in Bluetooth Technology as Used in IoT. *J. Sens. Actuator Netw.* **2018**, *7*, 28. [CrossRef]
10. Rondon, R.; Mahmood, A.; Grimaldi, S.; Gidlund, M. Understanding the Performance of Bluetooth Mesh: Reliability, Delay, and Scalability Analysis. *IEEE Internet Things J.* **2020**, *7*, 2089–2101. [CrossRef]
11. Di Marco, P.; Skillermark, P.; Larmo, A.; Arvidson, P. *Bluetooth Mesh Networking*; Ericsson AB - White Paper 284 23-3310 Uen; 2017. Available online: https://www.ericsson.com/en/reports-and-papers/white-papers/bluetooth-mesh-networking (accessed on 1 December 2020).
12. Hernández-Solana, A.; Pérez-Díaz-De-Cerio, D.; García-Lozano, M.; Bardají, A.V.; Valenzuela, J. Bluetooth Mesh Analysis, Issues, and Challenges. *IEEE Access* **2020**, *8*, 53784–53800. [CrossRef]
13. World Health Organization (WHO). Contact tracing in the context of COVID-19 (Interim Guidance). 2020. Available online: https://apps.who.int/iris/bitstream/handle/10665/332049/WHO-2019-nCoV-Contact_Tracing-2020.1-eng.pdf?sequence=1&isAllowed=y (accessed on 1 December 2020).
14. Gu, Y.; Ren, F. Energy-Efficient Indoor Localization of Smart Hand-Held Devices Using Bluetooth. *IEEE Access* **2015**, *3*, 1450–1461. [CrossRef]
15. De Blasio, G.; Quesada-Arencibia, A.; García, C.R.; Rodríguez-Rodríguez, J.C.; Moreno-Díaz, R. A Protocol-Channel-Based Indoor Positioning Performance Study for Bluetooth Low Energy. *IEEE Access* **2018**, *6*, 33440–33450. [CrossRef]
16. Ng, P.C.; She, J.; Ran, R. A Compressive Sensing Approach to Detect the Proximity Between Smartphones and BLE Beacons. *IEEE Internet Things J.* **2019**, *6*, 7162–7174. [CrossRef]
17. Mackey, A.; Spachos, P.; Song, L.; Plataniotis, K.N. Improving BLE Beacon Proximity Estimation Accuracy Through Bayesian Filtering. *IEEE Internet Things J.* **2020**, *7*, 3160–3169. [CrossRef]
18. Jürgens, M.; Meis, D.; Möllers, D.; Nolte, F.; Stork, E.; Vossen, G.; Werner, C.; Winkelmann, H. Bluetooth Mesh Networks for Indoor Localization. In Proceedings of the 2019 20th IEEE International Conference on Mobile Data Management (MDM), Hong Kong, China, 10–13 June 2019; pp. 397–402.
19. Leith, D.; Farrell, S. Coronavirus Contact Tracing: Evaluating the Potential of Using Bluetooth Received Signal Strength for Proximity Detection. *arXiv* **2020**, arXiv:2006.06822.
20. Hatke, G.F.; Montanari, M.; Appadwedula, S.; Wentz, M.; Meklenburg, J.; Ivers, L.; Watson, J.; Fiore, P. Using Bluetooth Low Energy (BLE) Signal Strength Estimation to Facilitate Contact Tracing for COVID-19. *arXiv* **2020**, arXiv:2006.15711.
21. Song, S.W.; Lee, Y.S.; Imdad, F.; Niaz, M.T.; Kim, H.S. Efficient Advertiser Discovery in Bluetooth Low Energy Devices. *Energies* **2019**, *12*, 1707. [CrossRef]
22. Azad, M.A.; Arshad, J.; Akmal, S.M.A.; Riaz, F.; Abdullah, S.; Imran, M.; Ahmad, F. A First Look at Privacy Analysis of COVID-19 Contact Tracing Mobile Applications. *IEEE Internet Things J.* **2020**. [CrossRef]
23. Garg, L.; Chukwu, E.; Nasser, N.; Chakraborty, C.; Garg, G. Anonymity Preserving IoT-Based COVID-19 and Other Infectious Disease Contact Tracing Model. *IEEE Access* **2020**, *8*, 159402–159414. [CrossRef]
24. Nordic Semiconductor. nRF Connect for Mobile. Available online: https://www.nordicsemi.com/Software-and-tools/Development-Tools/nRF-Connect-for-mobile (accessed on 1 December 2020).