# Social Distancing using Bluetooth Low Energy to Prevent the Spread of COVID-19

Satyam Kumar
Department of Computer Science & Engineering
Darbhanga College of Engineering
Darbhanga, India
Email: satyamkr.dce@gmail.com

Vikas Gautam
Department of Computer Science & Engineering
LNJPIT Chapra
Chapra, India
Email: vgautum@gmail.com

Amit Kumar
Department of Electrical and Electronics Engineering,
Darbhanga College of Engineering
Darbhanga, India
Email: amit.kumarc210@gmail.com

Puja Kumari
Department of Physics,
Darbhanga College of Engineering
Darbhanga, India
Email: pp.phy.ism@gmail.com

2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence) | 978-1-6654-1451-7/20/$31.00 ©2021 IEEE | DOI: 10.1109/Confluence51648.2021.9377096

***Abstract*—Maintaining Social Distancing among people is very crucial at present when the whole world is facing the COVID-19 pandemic, and yet there is no effective cure or antidote is available. In this pandemic situation, we need a robust solution to flatten the curve of COVID-19 so that people do not get sick, so our economy and supply chain stays running, and people's risk can be minimized. Some techniques are available to maintain social distancing, but most are based on visual recognition, which is not effectively applicable everywhere and comparatively expensive. To overcome this issue and make it more effective and smoother, an android application has been developed for the Social Distancing Alert System (SDAS). The system is based on Bluetooth Low Energy (BLE) proximity detection technology, which uses the Received Signal Strength Indicator (RSSI) and Transmission Power (TxPower) of user's android handsets for the estimation of real-time social distancing status and to alert them. It notifies users through real-time popup notifications on the screen of the handsets and by vibration and notification sound when another app user gets closer to a range of 2 meters with four levels of granularity. This application also provides other compelling features like past social distancing status tracking of the past 30 days, power-saving, and QR scanning to ignore specific people without an internet connection.**

***Keywords—Bluetooth Low Energy (BLE), Bluetooth Plugins, Social Distancing Alert System (SDAS), Received Signal Strength Indicator (RSSI), Transmission Power (TxPower), Android, QR Scanning.***

## I. Introduction

The entire world is facing the COVID-19 pandemic currently, but to fulfilling daily life needs, people are compelled to go to markets, offices, travel, and do a lot of stuff in a crowded place. Companies are also continuing their work with almost 50% of employees, and people need to get back to work eventually. We need them not to get sick, so our economy and supply chain stays running, and we can flatten the curve of COVID-19. According to the World Health Organization (WHO) guidelines and suggestions given by medical experts, social distancing is the most compulsory precaution until a vaccine or antidote is not developed for the COVID-19 virus [1, 2]. Social Distancing Alert System Application (SDAS) helps protect the people and workforces everywhere like public transport, companies, market and crowded public places, etc., through maintaining social distancing so that the risk of infection can be minimized [3].

SDAS is an android application built-in Kotlin and based on Bluetooth Low Energy (BLE) proximity detection technology that helps people maintain social distancing at a range of 2 meters with different levels of granularity, the curve of COVID-19 can be flattened. It notifies the user with the help of notifications alerts and shows real-time social distancing status when other users are detected in the proximity of the BLE (android handsets) [4, 5].

At present, ~2.5 billion active android handsets users are available globally, and android handsets provide built-in support for BLE technology with very little power consumption. Therefore this application can be effectively used by a large population without any extra cost of implementation.

## II. Methodology

This technique for maintaining social distancing takes permissions for accessing Bluetooth and Camera of the handset and broadcast, advertises the signal with the help of Generic Attribute Profile Server (GATT) and other utilities to establish the connection and scan the presence of app user in the proximity of BLE shown in Fig. 1. After establishing a link, it calculates the resulting strength of the signal using Algorithm1 with taking into consideration the type of handset (Android or IOS), reported Received Signal Strength Indicator (RSSI), and Transmission Power (TxPower) [6–8]. Based on the resulting signal strength, the application provides real-time notification alerts to the user with 4 levels of granularity displayed in Fig. 2 as "Safer" = Green, "Caution" = Yellow, "Warning" = Orange, "Danger" = Red. Individuals must turn ON the application to get real-time alerts on the screen with vibration and notifications sound, and it will continue in the background unless paused.

This application also facilitates ignoring some android handsets (specific users) by scanning the QR code on their respective app. This feature mutes the alerts for customized people to whom users decide are safe or want to ignore. This feature is helpful at home for family and specific members.

The application needs only Bluetooth and camera access to give real-time alert and customization. All data is stored locally on the user handset to maintain the privacy of individuals. We have used Firebase as a database for storing information locally. It also keeps the user's social distancing status on the application locally with date and time for the past 30 days and reset automatically over the period. To manage the whole process, the application uses some Bluetooth plugins and protocols and Application Programming Interface (API) like Bluetooth GATT Server, etc., and QR scanner plugins [4, 5]. There is no requirement for an active internet connection to run the application. Individuals can use the SDAS application by installing it on their android phones and by turning it on.

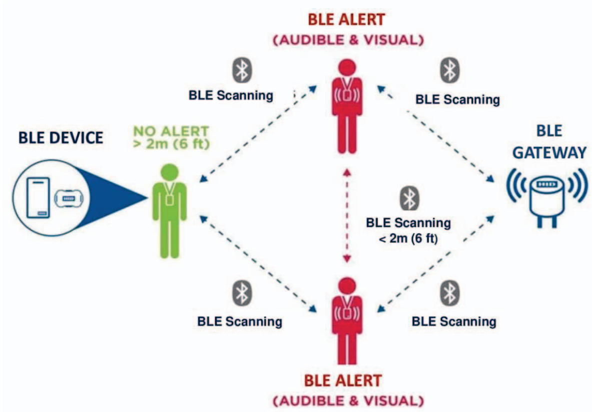


Fig. 1. Scanning process of BLE Android handsets


Fig. 2. Visual representation of the Signal Strength

## A. Bluetooth Terminologies

*a) Bluetooth Low Energy:* BLE is a wireless Personal Area Network (PAN) technology and an extension of the Bluetooth standard in version 4.0 that enables low-power, low-cost, and short-range wireless communication [6]. Android handsets provide built-in support for BLE in the central role to discover devices, services, and transmit information.

*b) Bluetooth GATT Server:* It is a public API for the Bluetooth GATT Profile server role. This class provides Bluetooth GATT server role functionality, allows applications to create Bluetooth smart services and characteristics. Bluetooth GATT server is a proxy object for controlling the Bluetooth Service via inter-process communication (IPC).

*c) RSSI:* The signal strength reported from the scan depends on the distance and broadcasting power value. RSSI is used to estimate the approximate distance between users using another value defined by the BLE standard: TxPower [8–12].

*d) Transmission Power (TxPower):* The TxPower reported from the scan indicates the expected RSSI at a distance of 1 meter to the device. It is also known as a 1 meter RSSI [9, 10].

*e) Signal Propagation Constant (n):* The constant depends on the environmental factors and varies from 2-4. In free space, its value is taken as n=2.

## B. Used Constants

*a) Scanning Interval of the signal*

- Scan Period = 1000 ms
  (It scans the device for every second.)
- Rebroadcast Period = 30000 ms
  (It stores the status of the user in the history section for every 30 seconds).
- Background Trace Interval=2000 ms
  (It scans devices in the background at an interval of every 2 seconds).
- Foreground Trace Interval = 2000 ms
  (It scans devices in the foreground at an interval of every 2 seconds).

*b) Bounds Signal Strength (based on Algorithm1)*

- Signal Distance Safer = 27 dBm
  (This or lower is socially distant and indicating "Green" status, i.e., Safer).
- Signal Distance Caution= 37 dBm
  (Between Signal Distance Safer and Caution displays "Yellow" alert, i.e., Caution).
- Signal Distance Warning = 52 dBm
  (Between Signal Distance Caution and Warning display "Orange" alert, i.e., Warning).
- And above 52 dBm, displays "Red" alert, i.e., Danger, Not socially distant.
- Assumed TxPower = 127 dBm
  (It refers to predefined static transmission power for reference purpose).

Fig. 3 shows the different signal alerts generated based on the signal strength corresponding to the users' distance.

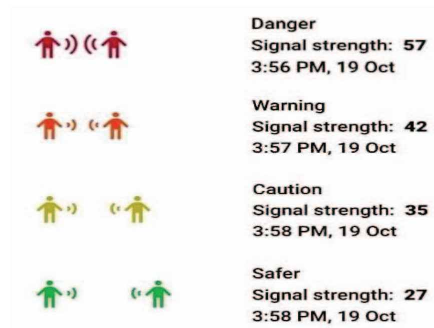


Fig. 3. Social Distancing Status of active users

## C. Flowchart

The flowchart of the Social Distancing Alert System android application to generate alert is shown in Fig. 4.

# III. Algorithms

The SDAS android application works on Algorithm1, which is derived practically using experiments done on different android handsets using RSSI tracker applications and considering the variation in signal strength of the BLE handset during the whole scanning process. We applied the theoretical formula for RSSI to distance conversion, as shown in Algorithm2. Testing is done on different android handsets for proximity estimation. Still, it doesn't give appropriate and effective results due to the propagation constant (n) that depends on various environmental factors varying between 2 and 4 [8]. Therefore, to remove this dependency and provide real-time alerts to the user with the help of Algorithm2 was a big challenge.

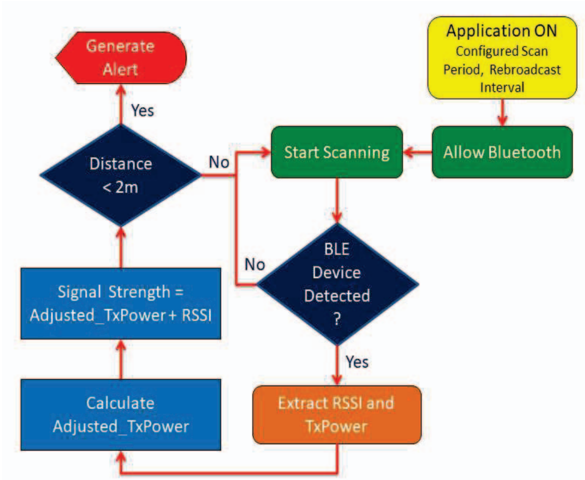

Fig. 4. Flowchart of the android application

To resolve the above problem, an optimized algorithm (Algorithm1) is proposed that calculates the signal strength from Android BLE using reported RSSI and TxPower without considering any constants. The distance is mapped with resulting signal strength and a range of signal strength displayed in correspondence to distance with four granularity levels and notifies the user about their real-time social distancing status and alerts. Since Algorithm1 doesn't have a propagation constant, so it works effectively on every condition whether there is any obstruction between mobile devices or not with a variation of ±2 dBm (depends on different versions of BLE handsets).

For deriving Algorithm-1, samples have been taken with the help of different android mobile handsets that indicated the variation of signal power strength in dBm with the interpretation of distance observed using the RSSI Tracker applications. We have averaged various samples' response and integrated the signal strength value corresponding to multiple alerts displayed in Fig. 2 and 3.

### *A. Algorithm1: Calculation of Modified Signal Strength*

***Input:*** BLE handset scans another app user handset in proximity to receive RSSI value and considering parameters: RSSI, TxPower.

***Output:*** Returns the signal's strength from another handset taking into consideration the type of handset and the reported TxPower.

```
Calculate_Signal_Strength (RSSI,  TxPower){
// Fix for older handset that don't report power
if (TxPower + RSSI < 0)
    adjusted TxPower  = Assumed TxPower
else
    adjusted TxPower  = TxPower
// Notify the user when we are adding a device
signal = adjusted TxPower + RSSI
return signal
if (signal<=27)
    return Safer
else if (signal>27 && signal<=37)
    return Caution
else if (signal>37 && signal<=52)
    return Warning
else
    return Danger
}
```

### *B. Algorithm2: RSSI to Distance Conversion*

Algorithm2 calculates the distance between users using the following formula of Signal Strength calculation, which considers signal propagation constant to provide notification alerts.

***Input:*** The algorithm takes parameters as RSSI (dBm), TxPower (dBm) at 1 meter.

***Output:*** Returns distance in the meter of other users which is closest using BLE proximity [9, 10].

```
getDistance( rssi, TxPower ){
/*
RSSI = TxPower - 10 * n * log(d)
n = 2 (in free space)
d = 10 ^ ((TxPower - RSSI) / (10 * n))
*/
return Math.pow(10, ( TxPower – RSSI ) / (10 * 2))
}
```

## IV. FEATURES

- ***Detecting Active Users in the Proximity:*** The app broadcasts the signal nearby to its range and notifies the user when another app user enters in a range of 2 meters. This process continues in the background unless paused.
- ***Real-time Notification Alerts***: Whenever the application detects any user in the proximity range, it shows the signal strength of the user handsets that are the closest one with 4 levels of granularity: "Safer" = Green, "Caution" = Yellow, "Warning" = Orange, "Danger" = Red for the variable distance in the range of BLE (shown graph in Fig. 2) and notifies the user's by sound notifications, vibrations, and displaying alerts on the screen of the app.
- ***Stores History***: The app stores scanning status of past 30 days at every 30 seconds with date and time for future use when anyone was getting infected.
- ***Pause Scanning***: Users can pause the app's scanning when not in use like sleeping time.
- ***User customization***: It provides the facility to ignore the specific person and mutes alert for them by scanning the QR displayed in this section. This feature is handy for family members.
- ***Low Energy Consumption***: BLE consumes very little power from handsets and runs continuously in the background without affecting other applications.
- ***No Extra cost for Implementation:*** The android handsets have inbuilt BLE technology. Therefore one need not install any extra equipment for its working.

- ***Maintain the user's privacy*****:** The app does not store any data on the server and keeps all data locally on the user's handset to maintain their privacy.

## V. Disadvantages

- The minimum supported version of Android to run this application is 4.2 (API 19).
- The RSSI values returned might differ depending on the Bluetooth version of handsets and their surroundings.

## VI. Applications

- ***Public Transport (Bus/ Railways/ Metros etc.)*****:** In this pandemic, when passengers are waiting for the bus at a bus stop or for a train at the platform or while travelling, they can use this application to be socially distant from others and minimize the risk of infection.
- ***Crowded Public Places*****:** People are doing a lot of stuff in crowded places like government offices, markets, and malls in daily life. At those places, individuals can turn on the application to get alerts. Even this application can be used everywhere to maintain social distancing.
- ***Company's workplaces*****:** Several companies and manufacturing units have started their productions and other works with 50% of employees, but infection protection is a big challenge in this pandemic. So at the workplace to maintain social distance, they can use this application.
- ***Healthcare workers***: In this pandemic, we all know frontline corona warriors like doctors, nurses, and medical staff doing their duties 24x7 tirelessly at hospitals. There is a high risk of getting them infectious. So they can use this application to minimize the risk.
- ***Construction workers*****:** Now, construction works are also resumed after a long time, there is a chance that workers might get infected at construction places. This application is useful for them too.
- ***Education institutions*****:** After some days, when education institutions will also be resumed with offline classes, then there will be a big challenge to maintain the hygiene of students as well as teaching and non-teaching staff. This application will be helpful to maintain safe social distancing status at the campus.
- ***Warehouse workers:*** At the warehouse, where products are stored for delivery to the customers, and at the grocery shop wherever there is a chance of physical contact, this application can be used.
- ***Road and municipal workers*****:** Road and municipal workers can also use this application to be socially distant at their workplace.

## VII. Result

After implementing and testing Algorithm1 on various android handsets, we analyzed that it gives relevant results with a maximum variation of ±2 dBm with resulting signal strength due to different versions and configurations of the BLE. The graph between resulted signal strength vs distance is shown in Fig. 5. Individuals can install and turned ON the application to get notifications and alerts by scanning the QR code of the APK file shown in Fig. 6.

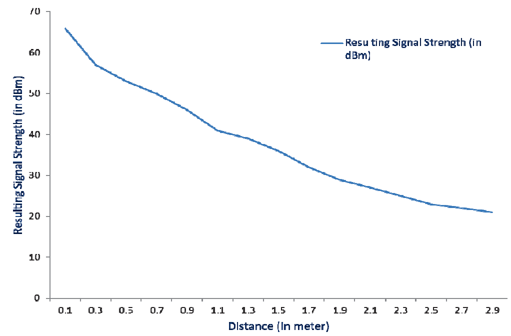


Fig. 5. Graph representation of Resulting Signal Strength vs Distance

Fig. 6. QR code for APK installation

## VIII. Conclusions

We have developed and presented an Android application "Social Distancing Alert System" with an optimized algorithm to maintain social distancing using Bluetooth Low Energy (BLE) RSSI and Transmission Power (TxPower) that is independent of signal propagation constant. Due to no such environmental-based constraints, Algorithm1 provided relevant results for BLE proximity detection to maintain social distancing effectively and prevent the spread of COVID-19. Android-supported BLE-based technology makes it available for many users (~2.5 billion) without any cost of implementation along with minimal power consumption that makes it convenient and durable. We have also concluded that RSSI-based BLE depends on several conditions, which make the whole work complex from the mathematical point of view and to resolve this, we can use Algorithm1. In the future, these techniques should be integrated as a pandemic mode with smartphones and also with a wide range of Bluetooth and Global Positioning Systems (GPS) supported low-cost micro devices like smart-watches, bands, etc. to minimize the risk of a pandemic and improvement in the healthcare system.

## Acknowledgement

We want to acknowledge TEQIP-III a joint project of NPIU (a unit of MHRD, Govt. of India) and World Bank for their support and encouragement to carry out this research.

## REFERENCES


[1] L. Matrajt and T. Leung, "Evaluating the Effectiveness of Social Distancing Interventions to Delay or Flatten the Epidemic Curve of Coronavirus Disease," *Emerg. Infect. Dis.*, vol. 26, no. 8, pp. 1740–1748, Aug. 2020, doi: 10.3201/eid2608.201093.

[2] C. T. Nguyen *et al.*, "A Comprehensive Survey of Enabling and Emerging Technologies for Social Distancing - Part II: Emerging Technologies and Open Issues," *IEEE Access*, vol. 8, pp. 154209–154236, 2020, doi: 10.1109/ACCESS.2020.3018124.

[3] C. T. Nguyen *et al.*, "A Comprehensive Survey of Enabling and Emerging Technologies for Social Distancing—Part I: Fundamentals and Enabling Technologies," *IEEE Access*, vol. 8, pp. 153479–153507, 2020, doi: 10.1109/ACCESS.2020.3018140.

[4] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology," *Sensors (Switzerland)*, vol. 12, no. 9, pp. 11734–11753, 2012, doi: 10.3390/s120911734.

[5] "Bluetooth Smart." https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics/low-energy.

[6] Y. Zhuang, J. Yang, Y. Li, L. Qi, and N. El-Sheimy, "Smartphone-based indoor localization with bluetooth low energy beacons," *Sensors (Switzerland)*, vol. 16, no. 5, pp. 1–20, 2016, doi: 10.3390/s16050596.

[7] J. Paek, J. Ko, and H. Shin, "A Measurement Study of BLE iBeacon and Geometric Adjustment Scheme for Indoor Location-Based Mobile Applications," *Mob. Inf. Syst.*, vol. 2016, 2016, doi: 10.1155/2016/8367638.

[8] M. Kaczmarek, J. Ruminski, and A. Bujnowski, "Accuracy analysis of the RSSI BLE SensorTag signal for indoor localization purposes," *Proc. 2016 Fed. Conf. Comput. Sci. Inf. Syst. FedCSIS 2016*, vol. 8, pp. 1413–1416, 2016, doi: 10.15439/2016F501.

[9] P. S. Helode, Dr. K. H. Walse, and Karande M.U., "An Online Secure Social Networking with Friend Discovery System," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 5, no. 4, pp. 8198–8205, 2017, doi: 10.15680/IJIRCCE.2017.

[10] M. Al Qathrady and A. Helmy, "Improving BLE distance estimation and classification using TX power and machine learning: A comparative analysis," *MSWiM 2017 - Proc. 20th ACM Int. Conf. Model. Anal. Simul. Wirel. Mob. Syst.*, vol. 2017-November, pp. 79–83, 2017, doi: 10.1145/3127540.3127577.

[11] A. A. Kherani, S. K. Bhogi, and B. Shin, "Hybrid location tracking in BLE beacon systems with in-network coordination," *2016 13th IEEE Annu. Consum. Commun. Netw. Conf. CCNC 2016*, pp. 814–815, 2016, doi: 10.1109/CCNC.2016.7444890.

[12] J. R. Martínez-De Dios, A. Ollero, F. J. Fernández, and C. Regoli, "On-Line RSSI-range model learning for target localization and tracking," *J. Sens. Actuator Networks*, vol. 6, no. 3, 2017, doi: 10.3390/jsan6030015.