RFID BASED BUS MANAGEMENT SYSTEM

S.Saritha[1], Sai Meenakshi Raja[2], Noor Mohammed.S[3]

[1]*Assistant Professor, Electronics and Communication Engineering, Nandha College Of Technology, Tamil Nadu, India*
[2]*UG Student, Electronics and Communication Engineering, Nandha College Of Technology, Tamil Nadu, India*
[3]*UG Student, Electronics and Communication Engineering, Nandha College Of Technology, Tamil Nadu, India*

ABSTRACT

The bus arrival time is primary information to most city transport travelers. Excessively long waiting time at bus stops often discourages the travelers and makes them reluctant to take buses.This paper presents a bus detection system using RFID technology that aims to ease the traveling and movement of people. In the bus detection subsystem, the nearby stations will be easily detected and then display by LCD inside the bus. The RF transmitter in every bus can read the code from RF receiver which is present in the bus stops. And also the security system is managed inside the bus using GSM module. For example, if the gas leaked in bus, using the gas sensor warn the bus driver and also send the message to the main office through the GSM.

Keyword: *Radio Frequency Identification (RFID), Peripheral Interface Controller (PIC), Global System for Mobile Communication (GSM),Global Positioning System(GPS)*

1. INTRODUCTTION

PUBLIC transport, especially the bus transport, has been well developed in many parts of the world. The bus transport services reduce the private car usage and fuel consumption, and alleviate traffic congestion. As one of the most comprehensive and affordable means of public transport, in 2011 the bus system serves over 3.3 million bus rides every day on average in Singapore w**ith around 5 million residents**.

When traveling with buses, the travelers usually want to know the accurate arrival time of the bus. Excessively long waiting time at bus stops may drive away the anxious travelers and make them reluctant to take buses. Nowadays, most bus operating companies have been providing their timetables on the web freely available for the travelers. The bus timetables, however, only provide very limited information (e.g., operating hours, time intervals, etc.), which are typically not timely updated. Other than those official timetables, many public services (e.g., Google Maps) are provided for travelers. Although such services offer useful information, they are far from satisfactory to the bus travelers. For example, the schedule of a bus may be delayed due to many unpredictable factors (e.g., traffic conditions, harsh weather situation, etc). The accurate arrival time of next bus will allow travelers to take alternative transport choices instead, and thus mitigate their anxiety and improve their experience. Towards this aim, many commercial bus information providers offer the realtime bus ar**rival time to the public**. Providing such services, however, usually requires the cooperation of the bus operating Companies (e.g., installing special location tracking devices on the buses), and incurs substantial cost.

In this paper, we present a novel bus arrival time prediction system based on crowd-participatory sensing. We interviewed bus passengers on acquiring the bus arrival time. Most passengers indicate that they want to instantly track the arrival time of the next buses and they are willing to contribute their location information on buses to help to establish a system to estimate the arrival time at various bus stops for the community. This motivates us to design a crowd-participated service to bridge those who want to know bus arrival time (querying users) to those who are on the bus and able to share the instant bus route information (sharing users). To achieve such a goal, we let the bus passengers themselves cooperatively sense the bus route information using commodity mobile phones. In particular, the sharing passengers may anonymously upload their sensing data collected on buses to a processing server, which intelligently processes the data and distributes useful information to those querying users.

Our bus arrival time prediction system comprises three major components: (1) Sharing users: using commodity mobile phones as well as various build-in sensors to sense and report the lightweight cellular signals and the surrounding environment to a backend server; (2) Querying users: querying the bus arrival time for a particular bus route with mobile phones; (3) Backend server: collecting the instantly reported information from the sharing users, and intellectually processing such information so as to monitor the bus routes and predict the bus arrival time. No GPS or explicit location services are invoked to acquire physical location inputs.

In the following of this paper, we first introduce the background and motivation in Section 2. In Section 3, we detail the challenges of our system and describe our technical solutions. The evaluation results are presented in Section 4. We perform a trial study in London and the results are shown in Section 5. The related works are described in Section 6. We summarize this paper in Section 7.

2. BACKGROUND AND MOTIVATION

The bus companies usually provide free bus timetables on the web. Such bus timetables, however, only provide very limited information (e.g., operating hours, time intervals, etc.), which are typically not timely updated according to instant traffic conditions. Although many commercial bus information providers offer the realtime bus arrival information, the service usually comes with substantial cost. With a fleet of thousands of buses, the installment of invehicle GPS systems incurs tens of millions of dollars. The network infrastructure to deliver the transit service raises the deployment cost even higher, which would eventually translate to increased expenditure of passengers.

For those reasons, current research works explore new approaches independent of bus companies to acquire transit information. The common rationale of such approaches is to continuously and accurately track the absolute physical location of the buses, which typically uses GPS for localization. Although many GPS-enabled mobile phones are available on the market, a good number of mobile phones are still shipped without GPS modules. Those typical limitations of the localization based schemes motivate alternative approaches without using GPS signal or other localization methods. Besides, GPS module consumes substantial amount of energy, significantly reducing the lifetime of power-constrained mobile phones. Due to the high power consumption, many mobile phone users usually turn off GPS modules to save battery power. The mobile phones in vehicles may perform poorly when they are placed without line-of-sight paths to GPS satellites.

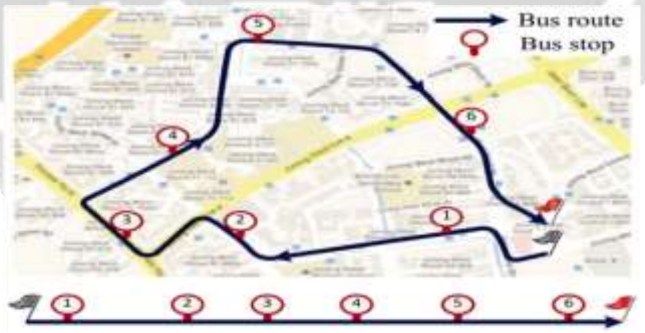


Fig -1: Absolute localization is unnecessary for arrival time prediction.

2.1 SYSTEM DESIGN

Though the idea is intuitive, the design of such a system in practice entails substantial challenges. In this section, we describe the major components of the system design. We illustrate the challenges in the design and implementation, and present several techniques to cope with them.

2.1.1 SYSTEM ARCHITECTURE

Querying user. As depicted in Fig. 2 (right bottom), a querying user queries the bus arrival time by sending the request to the backend server. The querying user indicates the interest bus route and bus stop to receive the predicted bus arrival time .Sharing user. The sharing user on the other hand contributes the mobile phone sensing information to the system. After a sharing user gets on a bus, the data collection module starts to collect a sequence of nearby cell tower IDs. The collected data is transmitted to the server via cellular networks. Since the sharing user may travel with different means of transport, the mobile phone needs to first detect whether the current user is on a bus or not. As shown in Fig. 2 (left side), the mobile phone periodically samples the surrounding environment and extracts identifiable features of transit buses. Once the mobile phone confirms it is on the bus, it starts sampling the cell tower sequences and sends the sequences to the backend server. Ideally, the mobile phone of the sharing user automatically performs the data collection and transmission without the manual input from the sharing user.

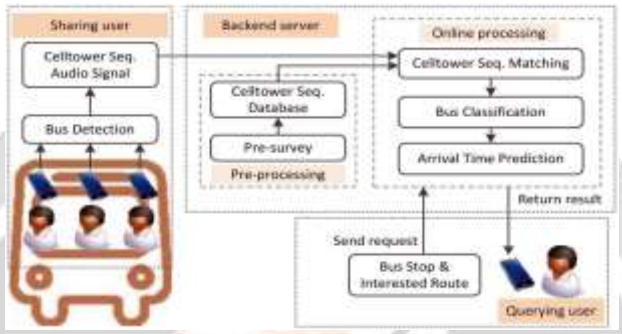


Fig -2: System architecture

2.1.2 BUS DETECTION

Am I on a Bus? During the on-line processing stage, we use the mobile phones of sharing passengers on the bus to record the cell tower sequences and transmit the data to the backend server. As aforementioned, the mobile phone should intelligently detect whether it is on a public transit bus or not and collect the data only when the mobile phone is on a bus. Some works [16], [18] study the problem of activity recognition and context awareness using various sensors. Such approaches, however, cannot be used to distinguish different transport modes (e.g., public transit buses and non-public buses). In this section, we explore multi-sensing resources to detect the bus environment and distinguish it from other transport modes. We seek a lightweight detection approach in terms of both energy consumption and computation complexity.

2.1.3 BUS CLASSIFICATION

When a sharing user gets on the bus, the mobile phone samples a sequence of cell tower IDs and reports the information to the backend server. The backend server aggregates the inputs from massive mobile phones and classifies the inputs into different bus routes. The statuses of the bus routes are then updated accordingly.

2.1.3.1.CELL TOWER SEQUENCE MATCHING

We match the received cell tower sequences to those signature sequences store in the database. Fig. 9 shows an illustrative example where a sharing passenger gets on the bus at location A. The backend server will receive a cell tower sequence of 7, 8, 4, 5 when the sharing user reaches location B. Say that the cell tower sequence of the bus route stored in the database is 1, 2, 4, 7, 8, 4, 5, 9, 6, then the sequence 7, 8, 4, 5 matches the particular bus route as a sub-segment as shown in Table 1. In practical scenarios, the sequence matching problem becomes more complicated due to the varying cell tower signal strength. Recall that for each sub-route we record the top-*3* cell tower IDs instead of the connected cell tower ID in the pre-processing period. We let each mobile phone send back the sequence of cell towers that the mobile phone has connected to. In the matching process on the server, we accordingly devise a top-*k* cell tower sequence matching scheme by modifying the Smith-Waterman algorithm [28]. Smith-Waterman is a dynamic programming algorithm for performing local sequence alignment which has been widely used in bioscience (e.g., to determine similar regions between two nucleotide or protein sequences).

We make concrete modifications on the original algorithm to support the top-*k* cell tower sequence matching. We weigh a matching of a cell tower ID with a top-*k* set according to the cell tower signal strength. Say that in a top-*k* set $S = \{c_1, c_2, ..., c_k\}$ ordered by signal strength (i.e., $s_i \geq s_j$, $1 \leq i \leq j \leq k$), where c_i and s_i denote cell tower i and its signal strength, respectively

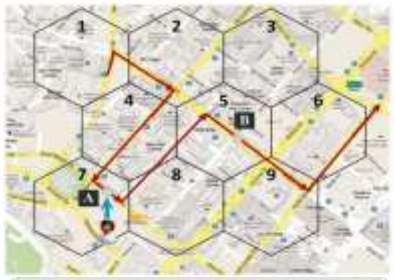
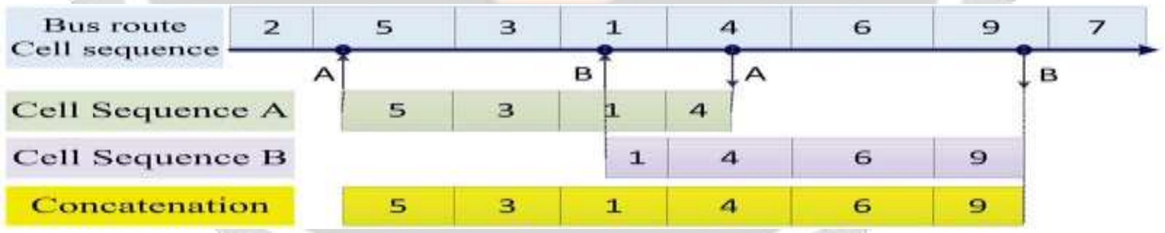


Fig- 3. Cell tower sequence matching.

We denote the uploaded cell tower sequence from a sharing user as $Seq_{\text{upload}} = u_1u_2 ...u_m$ where m is the sequence length. We also denote a cell tower set sequence in database as $Seq_{\text{database}} = S_1S_2 ...S_n$ where n is the set sequence length. If $u_i = c_w \in S_j$, u_i and S_j are considered matching with each other, and mismatching otherwise. We assign a score $f(s_w)$ for a match, where $f(s_w)$ is a positive non-decreasing scoring function and w is the rank of signal strength. In practice, we use $f(s_w) = 0.5^{w-1}$ as the scoring function according to the signal strength order in the set. The penalty cost for mismatches is set to be an empirical value of −0.5 which balances the robustness and accuracy in practice.

We choose top-*3* cell tower IDs with strongest cell tower signal strength to form a set based on our initial observations (Sectio3.2). The distinctive advantage of the proposed classification algorithm is its robustness to the variation of cell tower signal strength. Table 2 shows a cell tower set sequence matching instance. In the example, the uploaded cell tower sequence is Seq_{upload} = 1, 8, 10, 15, 16, and the cell tower ID set is shown in the first three rows sorted in decreasing order of the associated cell tower signal strength.



Bus route Cell sequence	2	5	3	1	4	6	9	7
		A		B	A		B	
Cell Sequence A		5	3	1	4			
Cell Sequence B				1	4	6	9	
Concatenation		5	3	1	4	6	9	

Table 1 : Cell Tower Sequence Matching

After running the sequence matching algorithm across all bus route sequences in the database, the backend server selects the bus route with the highest score. If the highest matching score is smaller or the sequence length is shorter than our empirical thresholds, the backend server postpones the updates to avoid errors. Intuitively, the small highest matching score would be due to mistriggering of sharing phones uploading cell tower sequence not from interested bus routes (e.g., rapid trains, private cars, etc). Too short cell tower sequence may not be informative since the misclassification rate of such short sequence is high and thus the backend server postpones the classification and the updating process until the sequence excesses the empirical threshold (which will be elaborated later).

One problem of the cell tower sequence matching is that some bus routes may overlap with each other. The mobile phones on the overlapped road segments are likely to observe similar cell tower sequences. Since many buses typically arrive at and depart from several major transit centers, such overlapping road segments among different bus routes are common.

We survey 50 bus routes in Singapore and measure their overlapped road segments using Google Maps. Fig. 10(a) plots the distribution of the lengths of overlapped road segments, which suggests that over 90% of the overlapped route segments are shorter than 1400 meters, and over 80% are less than 1000 meters. Considering that the coverage range of each cell tower in urban area is about 300-900 meters, we set the empirical threshold of cell tower sequence length to 7.

The below graph plots the cell tower sequence matching accuracy in classifying the bus routes. We vary the length of uploaded cell tower sequence from 2 to 9. We find that the matching accuracy is low when the cell tower sequence length is small (e.g, <4) largely because of the problem of route overlap. We observe that when the cell tower sequence length reaches 6, the accuracy increases substantially to around 90%. When the cell tower sequence length is larger than 8, the experimental results are reasonably accurate and robust.

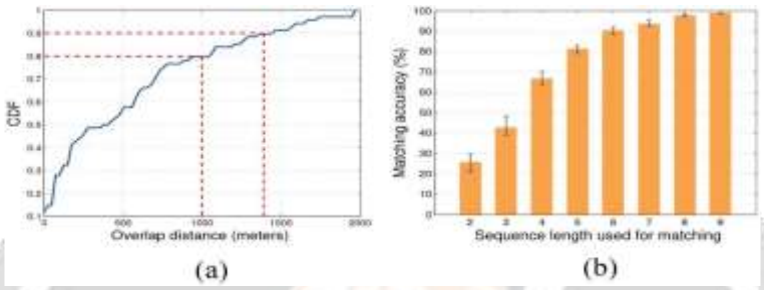


Fig-4. Overlaped routes and matching accuracy with varying sequence length. (a) CDF of the overlapped route length. (b) Matching accuracy with varying sequence length.

2.1.3.2 CELL TOWER SEQUENCE CONCATENATION: SOLVING JIGSAW PUZZLES

In many practical scenarios, the length of the cell tower sequence obtained by a single sharing user, however, may be insufficient for accurate bus route classification. An intuitive idea is that we can concatenate several cell tower sequences of different sharing users on the same bus to form a longer cell tower sequence. In Fig. 11, both cell tower sequences of sharing user A and B are short, while by concatenating the two cell tower sequences the backend server may obtain an adequately long cell tower sequence which can be used for more accurate bus classification. A simple way of concatenating the cell tower sequences is to let the mobile phones of sharing passengers locally communicate with each other (e.g., over Bluetooth) [20]. This approach, however, mandates location exposure among sharing passengers and might raise privacy concerns. We thereby shift such a job to the backend server.

Recall that the mobile phone needs to collect audio signals for bus detection (Section 3.3.1). Here, we reuse such information to detect whether the sharing passengers are on the same bus for cell tower sequence concatenation. At each bus stop, normally several passengers enter a bus and multiple beeps of the IC card readers can be detected. The time intervals between the consecutive beep signals fingerprint each bus in the time domain. Fig. 5 depicts an instance of the audio signals captured by three different mobile phones on the same bus. We depict the raw audio signals in Fig. 5(a), and corresponding frequency domain signals in Fig. 5(b)–(d). Compared with the time domain signal, the frequency domain signal is robust against the background noise (e.g., though signal strength increases are observed in 1kHz frequency band around 0.8s, the signal strengths in 3kHz frequency band remain low). We can see that in the frequency domain the signals are highly cross correlated and thus can be used to determine whether the phones are on the same bus. Specifically, the time intervals observed by three mobile phones are all approximately $dT1$ and $dT2$ in Fig. 5. We therefore use the time intervals between the detected beeps to determine whether multiple mobile phones are on the same bus. In our system, the mobile phones of sharing users keep sampling the audio signal and record the time intervals between the detected beeps. Such beep interval information is reported along with the cell tower sequences to the backend server.

Receiving the uploaded sensing data from sharing passengers, the backend server detects and groups the sharing passengers on the same bus by comparing both cell tower sequences and the time intervals of the beep signals. The backend server concatenates the pieces of cell tower sequences from the same bus and forms a longer cell tower sequence.

2.2 ARRIVAL TIME PREDICTION

After the cell tower sequence matching, the backend server classifies the uploaded information according to different bus routes. When receiving the request from querying users the backend server looks up the latest bus route status, and calculates the arrival time at the particular bus stop.

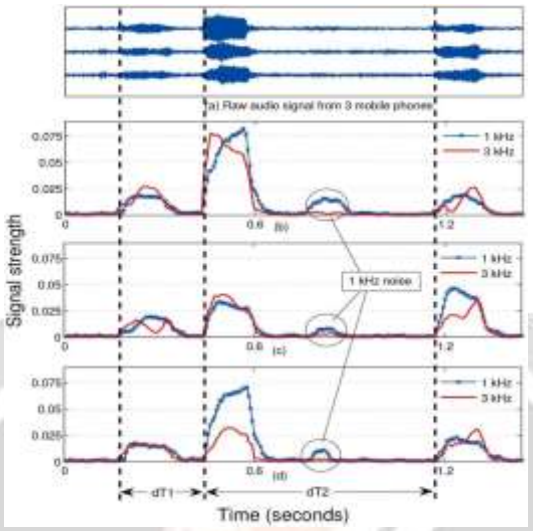


Fig-5. Time intervals of audio indication signals.

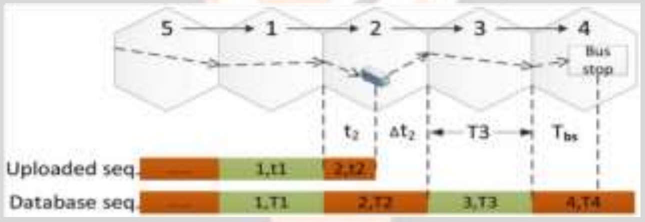


Fig-6. Bus arrival time prediction.

Fig.6 illustrates the calculation of bus arrival time prediction. The server needs to estimate the time for the bus to travel from its current location to the queried bus stop. Suppose that the sharing user on the bus is in the coverage of cell tower 2, the backend server estimates its arrival time at the bus stop according to both historical data as well as the latest bus route status. The server first computes the dwelling time of the bus at the current cell (i.e., cell 2 in this example) denoted as t_2. The server also computes the traveling time of the bus in the cell that the bus stop is located denoted as t_{bs}. The historical dwelling time of the bus at cell 3 is denoted as T_3. The arrival time of the bus at the queried stop is then estimated as follows,

$$T = T_2 - t_2 + T_3 + t_{bs}.$$

Without loss of generality, we denote the dwelling time in cell i as T_i, $1 \leq i \leq n$, the bus's current cell number as k, and the queried bus stop's cell number as q.

The server can estimate the arrival time of the bus as follows,

$$T = \sum_{i=k}^{q-1} T_i - t_k + t_q.$$

The server periodically updates the prediction time according to the latest route report from the sharing users and responds to querying users. The querying users may indicate desired updating rates and the numbers of successive bus runs to receive the timely updates.

3. IMPLEMENTATION AND EVALUATION

We implement a prototype system on the Android platform with different types of mobile phones, and collect the real data over a 7-week period. We first present the experiment environment and methodology. We test the performance of each system component individually to evaluate the design feasibility. We test the bus detection techinique and route classification method. When we evaluate the whole system performance, i.e., the accuracy of arrival time prediction, all the components are working together.

3.1 EXPERIMENTAL METHODOLOGY

Mobile phones. We implement the mobile phone applications with the Android platform using Samsung Galaxy S2 i9100 and HTC Desire. Both types of mobile phones are equipped with accelerometers and support 16-bit 44.1kHz audio signal sampling from microphones. The Samsung Galaxy S2 i9100 has a 1GB RAM and Dual-core 1.2GHz Cortex-A9 processor, while the HTC Desire has a 768MB RAM and 1GHz Scorpion processor. For most of our experiments, we base on the SingTel GSM networks in Singapore.

Experiment environment. Public bus transit system serves millions of bus rides every day covering most parts of Singapore. The public bus transit system is supervised by Land Transport Authority (LTA) of Singapore and commercially operated mainly by two major public transport providers, SBS Transit and SMRT Corporation. Many other transit means coexist with the public bus system. Mass Rapid Transit (MRT) trains form the backbone of the railway system. There are also tens of thousands of taxicabs operated by commercial companies and individual taxi owners. IC cards are widely used for paying transit fees. Several card readers are deployed for collecting the fees on SBS and SMRT public buses and at entrance gates of MRT stations.

3.2 BUS DETECTION PERFORMANCE

The experiment results suggest that the audio based method effectively detects the beep signal on the bus when the distance between the IC card reader and the mobile phone is within 3 meters. Considering that the entrance gate of the bus is about 1.4 meters wide, when a sharing user enters a bus, the mobile phone would be normally less than 1 meter away from the IC card reader. Notice that our system tolerates some missing beeps because there are multiple opportunities to detect the audio when other passengers are tapping their cards.

3.3 ARRIVAL TIME PREDICTION

We present the final bus arrival time prediction results based on above estimations. We collect the campus bus traces using a high accurate vehicle GPS navigator as the benchmarks. In the same buses, we collect cell tower sequences using two mobile phones and stored the sequence in memory stick for our later trace-driven study.

3.4SYSTEM OVERHEAD

Mobile phone. In order to maintain the sample resolution and remove the noise, we extract the audio signal with sliding widows with the window size of 32. We record the audio signal at the sampling rate of 8kHz, and use n = 128pt FFT to convert the time domain audio signals to frequency domain signals. The major computational complexity is attributed to performing FFT on mobile phones which is $O(n \log n)$. Current mobile phones can finish the computation task in realtime. For example, it takes approximately 1.25ms and 1.8ms on average to finish to 128pt FFT on Samsung Galaxy S2 i9100 and HTC Desire, respectively.

4. CONCLUSIONS

In this paper, we present a crowd-participated bus arrival time prediction system. Primarily relying on inexpensive and widely available cellular signals, the proposed system provides cost-efficient solutions to the problem. We comprehensively evaluate the system through an Android prototype system. Over a 7-week experiment period, the evaluation results demonstrate that our system can accurately predict the bus arrival time. Being independent of any support from transit agencies and location services, the proposed scheme provides a flexible framework for participatory contribution of the community. For a particular city, the only requirement of our system implementation is that there exist a backend server and an IC card based bus system.

Future work includes how to encourage more participants to bootstrap the system because the number of sharing passengers affects the prediction accuracy in our system. This common issue of crowd-sourced solutions is largely influenced by the penetration rate and popularity of the services. One may actively promote the service to reach a critical penetration rate so as to ensure that at least one sharing user is on the bus willing to report the bus status. At the initial stage, we may encourage some specific passengers (like the bus drivers) to install the mobile phone clients.

5. REFERENCES

[1] India's missing children, by the number. Link available at: http: //blogs.wsj.com/indiarealtime/2012/10/16/indias-missing-children-by-the-numbers/

[2] Children injured after drunk driver rams school bus into railing of bridge.chandigarh/children-injured-after-drunk-driver-rams-school-bus-into-railing-of bridge

[3] Saranya. J, Selvakumar. J, "Implementation of children tracking system on android mobile terminals," Communications and Signal Processing International Conference, Vol., no., pp.961, 965, 3-5 April 2013.

[4] Mori, Y.; Kojima, H.; Kohno, E.; Inoue, S.; Ohta, T.; Kakuda, Y.; Ito, "A Self-Configurable New Generation Children Tracking System Based on Mobile Ad Hoc Networks Consisting of Android Mobile Terminals," Autonomous Decentralized Systems (ISADS), 2011, 10th International Symposium , vol., no., pp.339,342, 23-27 March 2011.

[5] Shu, C., "Guardian Uses Bluetooth Low Energy Tech To Keep Your Child Safe" uses-bluetooth-low-energy-tech-to-keep-your-child-safe/