ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019

NextWave Helper: Android App for Search and Rescue

Manilal D. L.[1], Anand Shekhar [2], Joe Davis Akkara[3], Saran Narayan [4]

Head of Department, Department of Computer Science, Model Engineering College, Kochi, Kerala, India[1]

U.G Student, Department of Computer Science, Model Engineering College, Kochi, Kerala, India [2, 3, 4]

ABSTRACT: Natural and man-made disasters can have devastating effects on human society, but they can largely be mitigated with the coordinated efforts of rescue workers. However, existing disaster response means have multiple obstacles like accessibility, ease of use, dependency on social media and requirement of special skill sets on the part of the public. There can also be cases when the whole network infrastructure fails, which can seriously hinder rescue efforts and endanger the lives of the affected. To overcome these limitations, we developed an Android-based application that offers user-friendliness and real-time data related to crowd-sourced information, which can provide missing persons data, show affected areas and relief centre locations, and handle requests. In cases when the networks fail, we use Mobile Ad-hoc Networks (MANET) that will serve to establish and provide communication and coordination among mobile devices during the emergency situation.

KEYWORDS: MANET; emergency; disaster; Android; search; rescue

I. INTRODUCTION

In mid-August 2018, Kerala, a state in India was hit by one of the worst floods in its recent memory. With over 483 dead and more than a million displaced, it remains a dark specter that looms over the state even today. 3,200 relief camps were set up during and after the floods. The property damage is estimated to be around ₹40,000 crores (or US $5.6 billion). Soon after the tragedy struck, technical experts and volunteers came together and built the website 'keralarescue.in'[1] to obtain a platform to connect online calls for help with people coordinating offline rescue operations. The website[2] played an integral role in search and rescue operations and helped saved lives[3].

The website had a plethora of features, which we took inspiration from, like showing the latest announcements, help requests registered by people affected by the flood, locations of relief camps, and a list of missing and found people. However, being built in a hurry, the website was not as well designed as it could have been. The user interface was cluttered and not mobile ready, this affected the usability and made it difficult for senior citizens, who were primarily using their mobile phones to navigate the website.

During the floods, the communication in disaster-affected areas was erratic, with some damage and destruction of the communication infrastructure. In order to remedy this, we developed an android application for smartphones, which are carried by most people. It will have 2 modes of operation - a normal mode and a disaster mode. The normal mode will allow communication through normal channels, while the disaster mode will rely on an ad-hoc peer to peer network in order to allow communication even when infrastructure is damaged. This will allow the end-user to choose how to operate the application, based on the current situation. The app will also crowdsource information to mark areas affected by the disaster, which is an effective method to obtain information during trying times such as these.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019

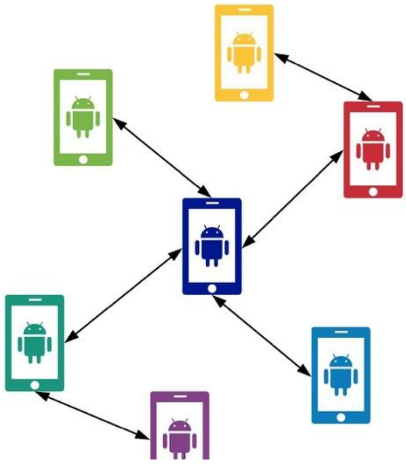

Fig 1. MANET formed by 7 devices

A mobile ad-hoc network (MANET) is an infrastructure-less self-configuring network, where each node acts as a router. There are no base stations, as in a normal network, and each node is free to move independently, making the network self-configuring and allowing devices in a MANET to leave or join the network dynamically. MANETs typically communicate at radio frequencies. All these features make a MANET a good solution to fix the communication issues that arise during disasters.

The rest of the paper is organized as follows. Section II discusses related works. Section III discusses the online features that are available in our application and the web portal, which allows people to coordinate with rescue workers to deliver supplies or find missing people. Section IV discusses the disaster mode feature, which creates a MANET with other devices in the area and its implementation. In Section V, we explain the algorithm used in this implementation. Section VI is concerned with the Performance review of the system. Finally, conclusions and plans for the future are discussed in section VII.

II. RELATED WORKS

A major issue is the possibility that communication infrastructure can be damaged during/after a disaster, leading to a loss of cellular connectivity or other standard means of communication. Several studies have been aimed to find remedies to this, but a large portion of them involve the deployment of additional infrastructure after the disaster, such as Satellite Access Points[4] and a wireless mesh infrastructure[5][6]. However, in case of a sudden, unexpected disaster such as an earthquake, it is often unrealistic to expect immediate mobilisation and deployment of this infrastructure and this delay, however slight, can lead to the loss of lives. Thus, it is imperative that an alternative to this which can be mobilised with almost no delay is developed.

[7] and [8] show that android applications are quite effective in aiding search-and-rescue efforts after a disaster, due to the number of people who have smartphones in this day and age. They show that usage of these applications can reduce response times of rescue workers and reduce obstructions in finding the victims, leading to lives saved. [9] and [10] show the value that crowdsourcing data can bring to situations such as these, with the lack of accurate information in the immediate aftermath of a disaster.

[11], [12] and [13] show the possibility of using a MANET in a situation where normal cellular connectivity is hampered and demonstrate its effectiveness. Another important feature discussed by these is the importance of reducing the power consumed by the application. Thus, implementing a MANET for mobile devices can be the solution to the possible inadequacy of communication methods after a disaster strikes. Along with this, implementing functionality that can be accessed via the internet can also help both victims and rescue workers with coordination and communication.

III. ONLINE FEATURES

With the current level of development of the internet, comprehensive internet connectivity is leading to an increasingly mobile reality. We are not tied to any single specific device, and everything is in the cloud. Further, coupled with Geographic Information System (GIS), extensive usage of these features has been made possible. This paper has two major components - the online web portal and android application. The web portal is only accessible to administrators and contains the backend of the application, while the Android Application is used by end users. There are 3 types of users throughout the paper the application:

1) Rescue Coordinator
2) Rescue Worker
3) Disaster Victim

Rescue coordinators are the administrators of the web portal who will receive information from the other 2 types of users through the internet and help to coordinate rescue efforts. Rescue workers will receive information from the coordinator through the application and work on their assigned help requests. Disaster victims can use the application to report/search for missing people, to register help requests, view map related information (relief camps locations, and areas affected by the disaster) and view relevant announcements.

A. *Android Application*:

The Android Application consists of 4 major parts, and is used by 2 types of end users - rescue workers and disaster victims.

1) Announcements: Announcements are posted by the rescue coordinator and users of the android application fetch these announcements through an Application Programming Interface (API). These announcements are cached in the application and updated whenever the app is opened and an internet connection is available (Fig. 2 Middle). The most recent announcement is also displayed in the home screen of the application.
2) Help Requests: Disaster victims can create requests for help, with their location (selected through a place picker on a map), and details of their requirements. The rescue coordinator can assign these requests to a rescue worker, who will try to work on them. The rescue worker can see the list of all registered help requests, and also help requests which have been assigned to him. 'Solved', 'Unsolved' and 'In-progress' help requests are assigned different colours (green, red and yellow respectively) when displayed in the application.
3) Missing Persons: Missing people can be reported and updated in the database. Once the person has been found, their status is updated by the rescue coordinator. 'Missing' and 'Found' people are assigned colours (red and green respectively) when displayed in the application (Fig. 2 Left).
4) Information Maps: The application also features maps which show a variety of information such as:
    - Affected Locations: Locations that are affected by the disaster can be marked on the map. In order to ensure that people don't maliciously mark locations as unaffected (Fig. 2 Right), we've implemented this in a way such that only people close to the area can mark it as affected. On selecting a road to mark as affected, the database will be updated and everyone who views the map will be able to see that particular road as an affected area.
    - Relief Camps: Relief camp locations can also be viewed in the application. The rescue coordinator must add relief camps through the web portal, following which they will be visible in the application. The user can view a list of relief camps, in which the closest camps will be displayed first. The user can also click on items in the list to view them on the map, or just view a map of their location, with nearby relief camps marked on it.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019

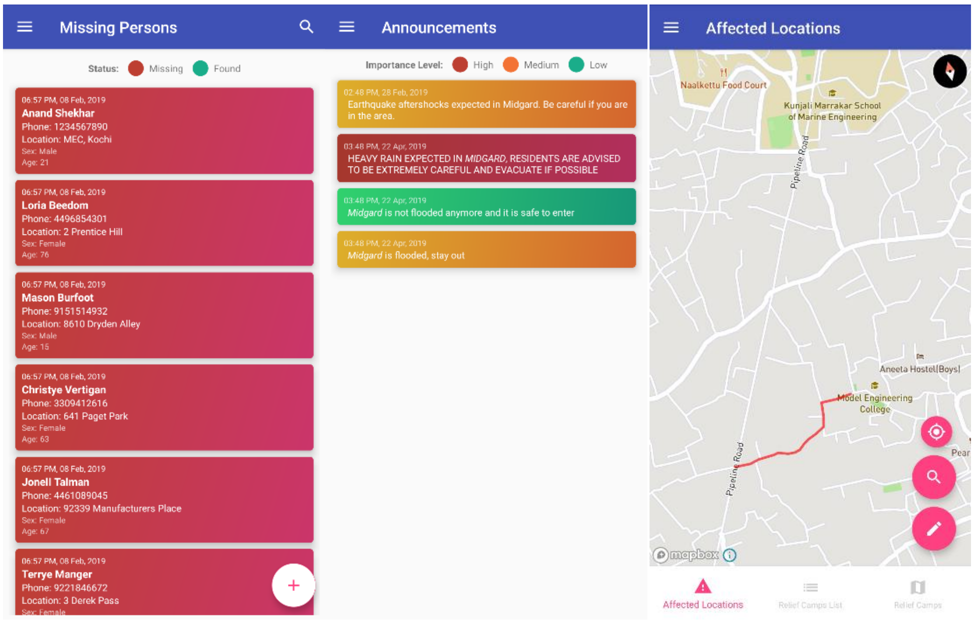

Fig.2. The android application

B. *Online Admin Portal*

The admin portal has an API used to upload data from end users and an interface for rescue coordinators to interact with the help requests - to change statuses of completed requests, missing persons or assign help requests to rescue coordinators. Other than adding announcements and relief camps, and marking missing people as found, the most important job for the rescue coordinator is to manage help requests.

1) Managing Help Requests: The rescue coordinator can view the help request on Google Maps, and then assign it to a particular rescue worker using the rescue worker's phone number (Fig. 3). The rescue worker can then view those help requests in the mobile application under the 'Assigned' section in "Help Requests". When allocating the help request to a rescue worker, the rescue coordinator sets the status of the request from 'Unsolved' to 'In-progress', letting the user know that their request has been assigned to a rescue worker.

2) Analytics: For visualization past data and increased accessibility, there is an option for representing the data from the website in the form of doughnut charts. In the analytics section of the website, the admin can view the history of requests - either all the requests together, or a subset of requests within 2 dates. The history of the selected request is shown, along with a doughnut chart to show the percentage of requests of different types - Solved/In-Progress/Completed in the case of help requests, or Missing/Found in the case of missing people. This allows us to analyse what fraction of the requests remained unfulfilled, and try and work out areas where requests remain uncompleted, for whatever reason and take steps to improve on them.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019

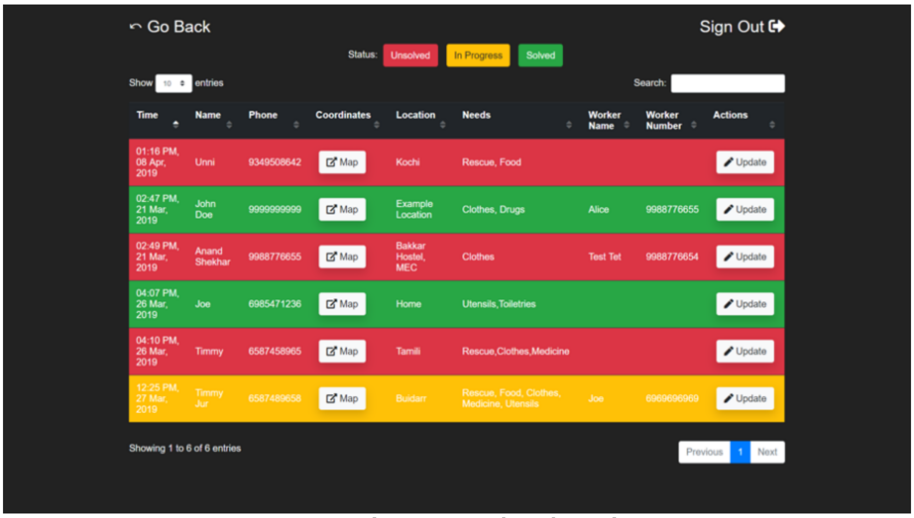

Fig.3. Help requests on the web portal

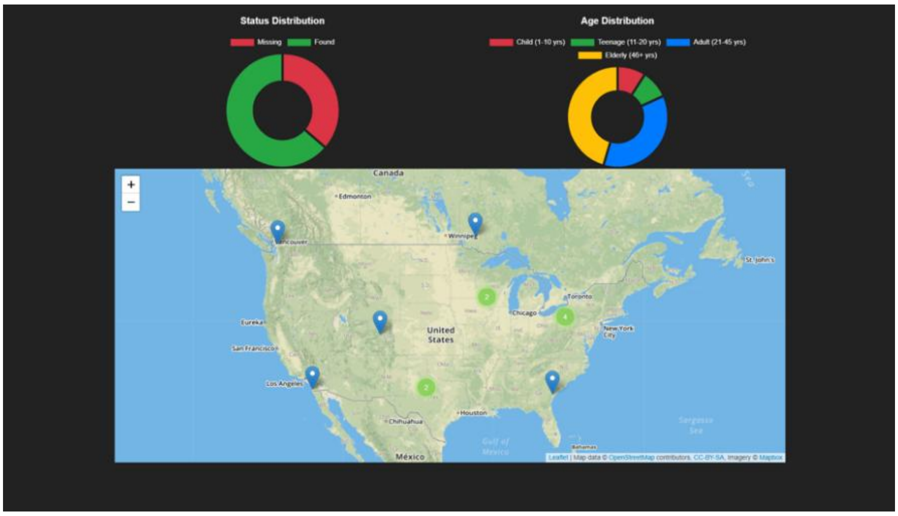

Fig.4.Analytics for missing persons on web portal

It is also possible to view the requests in cluster form on the map (Fig. 4), to identify where exactly the requests are coming from - to identify which areas have a high volume of requests and focus more efforts on those areas.

IV. OFFLINE MODE

The offline functionality of Application mainly revolves around Disaster mode. This mode employs the Google Nearby Communications API, which uses Bluetooth and Wi-Fi to search for and locate nearby peer devices. All the features are made available, once connection is established between 2 or more peer devices.

A. *Mobile Ad-hoc Network (MANET)*

A MANET is a type of ad-hoc network that can change locations and configure itself on the fly, and is a Delay Tolerant Network (DTN). Because MANETS are mobile, they use wireless connections to connect to various networks. They consist of set of mobile nodes connected wirelessly in a self-configured, self-healing network without having a fixed infrastructure. MANET nodes are free to move randomly as the network topology changes frequently. Each node behaves as a router as they forward traffic to other specified node in the network. This can be used in road safety, sensors for environment, home, health, disaster rescue operations, air/land/navy defense, weapons, robots, etc. We are using this network type for offline communication.

B. *Communication using MANET*

The steps to begin communication are as follows, the final screenshot is show in Fig. 5:

1) Switch to the Disaster Mode section of the application.
2) Click on the Discover Peers button in order to discover other devices that are connected/available to connect to the MANET.
3) Once the peers are discovered, the window will automatically switch to the chat.
4) Now, the user can send messages to other users connected via the MANET.
5) The user can also view the locations of other people in the chat by switching to the 'Find' tab.

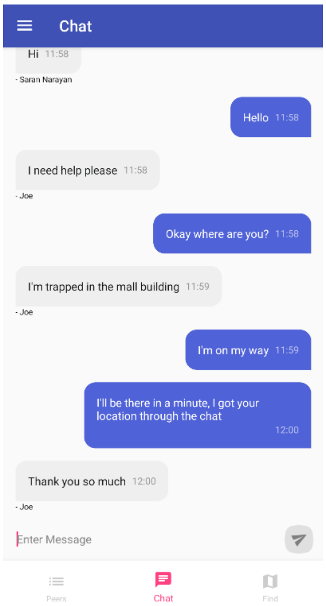

Fig.5.Disaster mode for offline chat

C. *Locating People using MANET*

There is also a provision to locate the people currently connected to the MANET, which can be accessed simply by switching to the 'Find' tab after connecting to a peer. This opens a map, which shows the location of the connected members of the MANET.

D. *Description of the Proposed Algorithm:*

The following algorithm highlights the basic steps involved in setting up the P2P connection between two Android devices using the Nearby Connections API.

Step 1: Start	
Step 2: startAdvertising()	//starts advertising service to nearby devices
Step 3: startDiscovering()	//starts discovering for nearby devices
Step 4: if onEndpointFound() called then	
Step 5: stopDiscovering()	//stops discovering nearby devices
Step 6: requestConnection()	//requests connection to the found endpoint

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019

Step 7: if acceptConnection() then
Step 8: onConnectionResult() callback //This is called on both devices
Step 9: Connection established
Step 10: else if rejectConnection() then
Step 11: startDiscovering() //starts discovering for any other devices
Step 12: End

V. PERFORMANCE REVIEW

A. Discovery time

In the discovery phase, time elapsed between initialization and the point at which all nodes are discovered is measured. We found in our results (Fig. 6) that delay increases non-linearly. This test was done with less than 10 peer devices. Any number of peers greater than that wasn't in the scope of our project. This behavior was observed due to the non-discovery of nearby devices in the first.

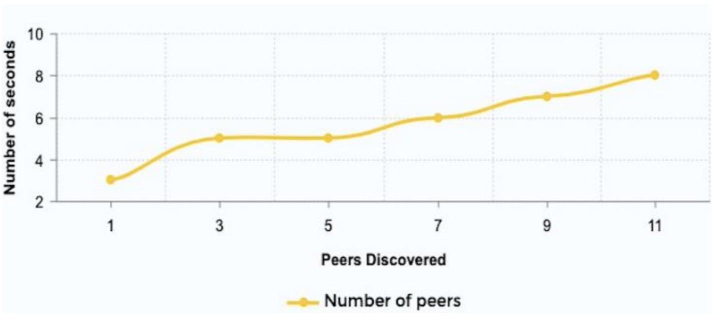

Fig. 6. Time to discover peers vs Number of peers

B. Bandwidth comparison

We tested the bandwidth of offline communications with two devices (Moto G5 Plus - Bluetooth 4.2 and Redmi 3S - Bluetooth 4.1). One device would act as the master and the other as slave. The slave echoes the messages it receives and was kept fixed at one location. This scenario gave us the round-trip time (RTT) from which we can calculate the speed in bytes per second easily.

While testing we noticed that there is little effect on the bandwidth for distances up to 100 feet. Also, that if we transmitted a 5KB message (two paragraph messages, which is unrealistic in our situation) the time remained almost constant. So, we can conclude that it has high bandwidth and the time it takes to transmit is more or less the delay to establish a communication link.

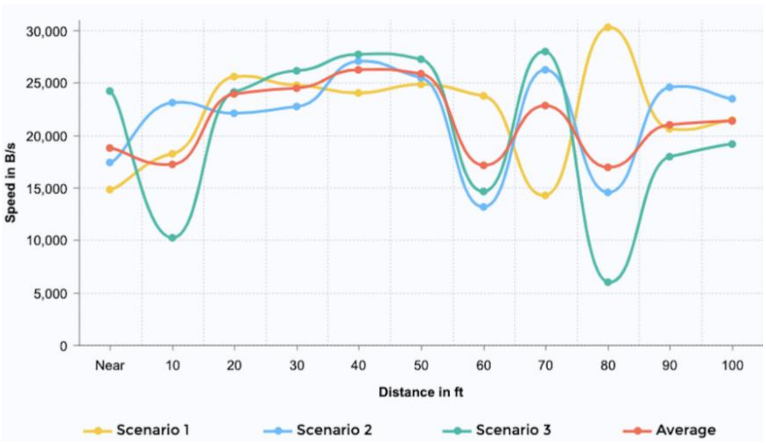

Fig. 7. Bandwidth comparison for different distances for a 1024 byte message

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 7, Issue 4, April 2019

The three scenarios mentioned in the graph (Fig. 7) is just three messages sent at the same distance but at different time intervals to get an average because of the variations we observed in the values for the same distances.

VI. CONCLUSION

In this paper, we developed an Android based application for smartphones, which can be used after a disaster to help the search and rescue efforts. We demonstrate its features, namely the ability to view announcements, add missing people and request for help and view affected areas and relief camps on the map. The application also allows for offline communication during disaster, which is for a situation where the standard communication infrastructure is damaged and the disaster victim cannot receive a mobile phone signal.

REFERENCES

1. IEEEKeralaSection, "Source code forhttps://keralarescue.in",Retrieved fromhttps://github.com/IEEEKeralaSection/rescuekerala.
2. Govt. of Kerala, Kerala State IT Mission and IEEE Kerala Section, "Kerala Rescue", Retrieved from https://keralarescue.in.
3. Shahim Baker, "Volunteers Come Together to Help Survivors of Floods in Kerala, India", Retrieved from http://sites.ieee.org/sb-uol/volunteers-come-together-to-help-survivors-of-floods-in-kerala-india/
4. F. Patricelli, J. E. Beakley, A. Carnevale, M. Tarabochia, and D. K.Von Lubitz, "Disaster management and mitigation: the telecommunications infrastructure", Disasters, vol. 33, no. 1, pp. 23–37, 2009.
5. R. B. Dilmaghani and R. R. Rao, "A wireless mesh infrastructure deployment with application for emergency scenarios", in 5th InternationalISCRAM Conference,Citeseer, 2008.
6. S. Mathur, "A rapidly deployable communications network architecturefor disaster management",Citeseer, Tech. Rep., 2009.
7. A. Hossain, S. K. Ray and R. Sinha, "A Smartphone-Assisted Post-Disaster Victim Localization Method", 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Sydney, NSW, 2016, pp. 1173-1179, 2016.
8. K. M. Rahman, T. Alam and M. Chowdhury, "Location based early disaster warning and evacuation system on mobile phones using OpenStreetMap", 2012 IEEE Conference on Open Systems, Kuala Lumpur, pp. 1-6, 2012.
9. Jane C. Samonte, Mary M. Rozario, Raymond Cleo, B. Aranjuez Nieland A. Maling Christopher,"Crowdsourced mobile app for flood risk management", pp 65-71, 2017.
10. L. Ma, X. Chen, Y. Xu, Y. Gao and W. Liu, "Study on crowdsourcing-compatible disaster information management system based on GIS", 2014 International Conference on Information Science, Electronics and Electrical Engineering, Sapporo, pp. 1976-1979, 2014.
11. HossmannTheus,Schatzmann Dominik, Carta Paolo and Legendre Franck,"Twitter in disaster mode: smart probing for opportunistic peers", Proceedings of the Third ACM International Workshop on Mobile Opportunistic Networks - MobiOpp '12, pp 93-94, 2012.
12. H. Yuze, Y. Qian and N. Suzuki, "Development of Smartphone Application for Off-Line Use in Case of Disaster", 27th International Conference on Advanced Information Networking and Applications Workshops, Barcelona,pp. 243-248, 2013.
13. S. Bhattacharjee, S. Kanta, S. Modi, M. Paul and S. DasBit, "Disaster messenger: An android based infrastructure less application for post disaster information exchange"2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Bangalore, pp. 1-5, 2016.