Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi
International Journal of Engineering Research and Development

UMAGD, (2021) 13(1), 13-21.

10.29137/umagd.664730

Cilt/Volume:13 Sayı/Issue:1 Ocak/January 2021

Araştırma Makalesi / Research Article

Developing a Message Broadcasting System for Natural Disasters

Veysel Harun Sahin[*,1,2], **İsmail Oztel**[3]

[1] *Department of Software Engineering, Faculty of Computer and Information Sciences, Sakarya University, 54050, Sakarya, TURKEY*
[2] *Disaster Management Application and Research Center, Sakarya University, 54050, Sakarya, TURKEY*
[3] *Department of Computer Engineering, Faculty of Computer and Information Sciences, Sakarya University, 54050, Sakarya, TURKEY*

Başvuru/Received: *26/12/2019* ***Kabul / Accepted***: *11/10/2020* ***Çevrimiçi Basım / Published Online:*** *18/01/2021*

Son Versiyon/Final Version: *18/01/2021*

Abstract

Natural disasters may cause fatal results on earth and often, it concludes with death and great destruction. One of the ways to cope with these effects is effective and continuous communication when the communication infrastructure destroyed with disasters. In this study, a message broadcasting system is proposed to enable the relief organizations to broadcast messages to the people in a disaster area even the communication infrastructures collapsed. The system uses smartphones for communication because of its widespread usage in the community. It consists of two mobile applications: master node and neighbor node. The master node is used to broadcast messages to disaster sufferers in the disaster area by relief organizations. The neighbor node receives the broadcasted message and also broadcasts the received message to other neighbor nodes. In the system, the communication between the smartphones were achieved by using Google Nearby Connections API. This API allows Android operating system smartphones to broadcast/discover each other and exchange messages regardless of network connectivity with the help of Bluetooth, Bluetooth Low Energy and Wi-Fi. In the context of this study, the proposed system was successfully developed and tested in different scenarios by using smartphones.

Key Words

"Disaster, Emergency Communication, Google Nearby Connections API, Mobile Application, Wireless Communication"

***Corresponding Author:** vsahin@sakarya.edu.tr*

1. Introduction

Many natural disasters have occurred in the world, which cause great losses and affect human life. According to an article of EM-DAT, 315 disasters, 11.804 deaths, 68.5 million affected people and $132 billion economic damages are reported in just 2018 all around the world (CRED, 2019). Many scientific studies have been carried out to reduce these effects.

Natural disasters such as earthquakes, floods, landslides, avalanches, storms, hoses, volcanic eruptions, fires can cause great damage to infrastructures that make people's lives easier. Unfortunately, the occurring time and place of some of these natural disasters cannot be predicted. Even if predicted, they cannot be prevented. In addition, most of the disasters catch people unawares. It is observed that 50% of deaths happen in the first two hours after a natural disaster (Shao et al., 2011). Taking into account all of these facts, communication becomes an important requirement for the effective response of the rescue teams and relief organizations on disasters.

Rescue, recovery, and reconstruction operations during and after disasters are major duties for the rescue teams, relief organizations, and the governments (Menon et al., 2016). An efficient communication network is needed for the healthy progress of these operations (Townsend & Moss, 2005). With reliable and continuous communication during and after disasters, it is possible for the relief organizations to carry out effective aiding activities to provide food, water, and shelter equipment to the disaster victims (Mase et al., 2010). Unfortunately, the communication infrastructure may collapse during natural disasters and this situation may cause the late and ineffective response to the disaster area which means that life and financial loss may increase (Krock, 2011). Uchida et al. (2011) shared the status of networks after the 9.0 magnitude earthquake in Japan on March 11, 2011. Fixed phone, the network of local government, LAN in city hall were not functional; 3G and GSM were partly functional but jammed; wireless LAN and internet of the satellite system were functional.

There are three main causes of the failure of telecommunication networks during natural disasters (Townsend & Moss, 2005): physical destruction, corruption in supporting the infrastructure, and network overload. It can take a great deal of time to renew or repair the crashed communication network (Tuna et al., 2012). Even the communication infrastructure is not damaged, it may not be usable because of the large number of calls occurring at the time of disaster which can be more than 60 times compared to normal times (Kim et al., 2015; Ministry of Internal Affairs and Communications, 2019).

The communication between the relief organization and the disaster sufferers is very important. There is a critical need to develop systems that can provide reliable and continuous communication during and after disasters. A message broadcasting system is proposed in this paper. The proposed system aims to enable relief organizations to broadcast messages to disaster sufferers even when the current communication infrastructure is not functional. The system uses smartphones for communication. There are mainly two advantages of using smartphones in such a system: The widespread usage of smartphones in the community and not requiring additional hardware. Two mobile applications have been developed to realize the system: one for relief organizations to broadcast the messages, and one for the disaster sufferers to receive and rebroadcast the messages. Owing to the message broadcasting system, the relief organizations can provide their assistance more effectively to the victims of the disasters.

The remainder of the paper is organized as follows. In Section 2, an overview of the related works on the topic is given. The proposed message broadcasting system is explained in Section 3, and implementation information is shared in Section 4. After giving information about the evaluation of the system and discussion in Section 5, the paper is concluded with Section 6.

2. Related Works

Many research have been carried out on natural disasters on different topics like the effect of the disasters (Chang-Richards et al., 2019; Kousky, 2016; Lesk et al., 2016), reducing the destructive effects or results of disasters (Aschenbruck et al., 2006; Kim et al., 2015; Noguchi & Fujii, 2000), effective and rapid response methods (Schultz et al., 1996), communication behavior in case of disaster (Lu, 2018), etc.

The communication and messaging between rescue teams, relief organizations and the survivors is essential for an effective and rapid response. Murugeswari and Radhakrishnan (2015) proposed a network architecture that uses Wireless Mesh Network (WMN) for emergency communication. Together with the network, they introduced an algorithm (Report Message Forwarding Algorithm) to provide reliable communication in case of a disaster. Shao et al. (2011) proposed a communication system using aero crafts. Several aerial base stations establish a network in order to ensure emergency communication. Also, some issues like coverage zone and capacity were analyzed in their study. According to the results, just one aerial base network is adequate for a county. Menon et al. (2016) proposed a technique, namely Reliable Routing Technique (RRT), in order to provide safe data delivery during disaster recovery. Their simulation results showed that the RRT achieved high performance for delivering data through the network using different topology and node number scenarios. Guo et al. (2010) investigated disaster area communication by using mobile nodes and relay nodes. In the paper, they introduced a novel mobility model and studied relay management. The approaches were evaluated with simulations and the results showed that their mobility model can help the disaster scenarios. Menon (2019) introduced an opportunistic routing protocol named Optimized Opportunistic Routing. The protocol aims to guarantee Quality of Service (QoS) and high transmission efficiency in highly dynamic mobile ad hoc networks. The efficiency of the protocol was shown by simulation. Ahmed et al. (2017) proposed a routing method for Wireless Sensor Networks (WSNs) that provides a reliable environment for communication during disasters. The method isolates misbehaving nodes during communication. By using this method, communication is achieved by using trusted nodes and shorter paths. This also leads to reduced energy consumption.

Tuna et al. (2012) proposed a post-disaster temporary communication system using unmanned aerial vehicles. They studied the system from three angels: peer to peer communication, coordination, navigation and localization. The navigation and localization were investigated with simulation studies and the simulation results showed that the proposed system can be used after disasters on the purpose of the communication. Pogkas et al. (2007) proposed an ad hoc network for post-disaster operations. The system can provide data about people who are inside a collapsed building and the state of the building. Also, it can be used for communication purposes. Energy efficiency and low cost are the two primary objectives of the study. According to the simulation results, energy consumption is reduced, user connection life is increased, but the network performance is not significantly affected using the proposed method.

Location detection is very important in case of a disaster and many studies are available in the literature about location detection for disasters. The study of Pogkas et al. (2007) mentioned above can be given as an example. Sun et al. (2011) developed a location-based vehicular ad hoc network for efficient rescue planning. Their proposed system, named RescueMe, retrieves the location information which was stored before the disaster and uses that information for the rescue planning. Also, the system pays attention to security and privacy issues. Arbia et al. (2017) presented an IoT end-to-end system with ORACE-Net routing protocol for disaster relief operations using wearable wireless sensor networks. With the help of the system, the information from a disaster area is collected using different types of wireless devices such as smartphones, Raspberry Pi modules, etc. The collected information is then sent to a command center. Mezghani and Mitton (2017) proposed a method for determining the location of disaster victims using their mobile devices by taking into consideration their phone energy levels. The system aims to provide maximum network coverage by using high-level energy nodes. The researchers applied the method on two laptops and according to their experience, the system has been observed to perform effectively. Khan et al. (2020) proposed a system that reports the location of the victims to the rescue teams. The system is rapidly deployable for the first response after a disaster situation. It uses a novel routing approach which includes Dynamic ID Assignment (DIA) algorithm and the Minimum Maximum Neighbor (MMN) algorithm. According to their experiments, the system reduces the message delivery delay and improves the message delivery ratio.

The developed system in the context of this study uses Google Nearby Connections API (Google, 2019) for communication between Android operating system (OS) smartphones. Google Nearby Connections API is used in many different research studies. Eichinger et al. (2019) proposed a new method for finding similar peers and exchanging items among them. The method was implemented by using this API. Takasuka et al. (2018) introduced a delay, disruption, disconnection tolerant networking system which provides a messaging infrastructure for the communication of a social networking service community. The system creates ad-hoc network with the help of the same API. Meftah et al. (2019) presented a test framework that aims to help the testing of nearby peer-to-peer applications. The framework targets the applications which are based on also this API. In a different study, Antonioli et al. (2019) reversed engineered the API and performed a security analysis of it.

3. The Message Broadcasting System

During the development phase of the message broadcasting system, the traditional software development process was followed. Firstly, the requirements were analyzed. Then the system was designed by considering the requirements. After the design phase, the implementation was realized. In the following subsections, these phases are explained in detail.

3.1. Requirement Analysis

The main aim of this study is to develop a system that helps relief organizations to broadcast messages to the people, especially to the disaster victims in the disaster area. For this system, three requirements were identified.

(1) The system should enable broadcasting messages (sending bulk messages) to the people.
(2) The system should operate even the communication infrastructure collapsed.
(3) The system should use commercial of the shelf (COTS) technology.

3.2. System Design

In the design phase by considering the requirements stated above we chose to use smartphones for communication as they are one of the most common technologies. For the operating system, Android OS was chosen.

As the system should work without current communication infrastructure, we planned to use Google Nearby Connections API (Google, 2019). This API allows Android OS smartphones to broadcast-discover each other and exchange messages regardless of network connectivity with the help of Bluetooth, Bluetooth Low Energy (BLE) and Wi-Fi. The API automatically chooses the most suitable protocol for communication according to each scenario without developer and user intervention.

For the message broadcasting system, M to N connection (mesh topology) between nodes is needed. Google Nearby Connections API supports star and mesh topologies. These are called connection strategies in the API, and their names are “P2P_STAR” and “P2P_CLUSTER”. Therefore, “P2P_CLUSTER” connection strategy was chosen for the operation of the API.

The main aim of this system is to send messages between smartphones. Google Nearby Connections API supports 3 different types of payloads: bytes payload, file payload, and stream payload. For exchanging messages between smartphones, we chose to use bytes payload.

Other information needed for this kind of communication is the identification of smartphones. In this system, all nodes have to have unique identifiers for seamless communication. There are several ways of getting a unique identifier in Android OS (Android, 2019). One way is using hardware identifiers. We chose to use secure settings Android Id (SSAID) which is one type of hardware identifiers.

After making these choices the design of the system was started. In the developed system, all of the smartphones are called nodes. The smartphone of the relief organization which is the main message sender is called **master node** and all other smartphones are called **neighbor nodes**.

The system works as follows. The relief organization (master node) broadcasts a message. The smartphones of the people (neighbor nodes) who are nearby the broadcast center receive the broadcasted message. After receiving the message neighbor nodes become broadcasters and start broadcasting the message they just received to other neighbors. The aim of this process is spread the message to the disaster area.

To create this system, two mobile applications were developed: master node application and neighbor node application. For the communication of these applications, two algorithms were designed.

The algorithm of the master node is shown in Algorithm 1:

Algorithm 1 Master Node Communication Algorithm

```
1:  broadcast (advertise)
2:  if there is a connection request from a neighbor node then
3:      accept connection
4:      send the message
5:      disconnect from the node
6:  end if
7:  go to step 1
```

In step 1 of the Algorithm 1, the master node starts broadcasting. This is written also as advertising as the API documentation uses this term. In this paper, we use broadcast and advertise terms interchangeably. When there is a connection request from a neighbor node (which is discoverer), the master node accepts the connection, sends the message and disconnects from the node. After that, it jumps to step 1 and continues to broadcast.

Here it should be noted that the master node does not check repetitive sending of messages to the same neighbor node. It just broadcasts. When a connection request is made by a neighbor node it accepts the request and sends the message. Repetitious message control is performed by a neighbor node upon receiving the message. To help this control, a unique identifier is given to each different message. The identifier is the current timestamp -in millisecond resolution- which is set by the master node when starting the broadcast operation.

The algorithm of the neighbor node is shown in Algorithm 2:

Algorithm 2 Neighbor Node Communication Algorithm

```
1:  discover
2:  if there is a broadcaster (advertiser) then
3:      send connection request
4:      connect to the node
5:      get the message from the node
6:      disconnect from the node
7:      stop discovery
8:      if the message is received before then
9:          exit
10:     end if
11:     set message count to 0
12:     broadcast (advertise)
13:     if there is connection request from a neighbor node then
14:         accept connection
15:         send the message
16:         disconnect from the node
17:         increase message count
18:         if message count < maximum node count then
19:             go to step 12
20:         end if
21:     end if
22: end if
```

Algorithm 2 is a little bit long because neighbor nodes both discover and broadcast messages. In the first step, the neighbor node starts discovery. If it finds a broadcaster, it connects to the node, gets message and disconnects from the node. Then it stops discovery. Afterward, it checks if the same message is received before or not. It compares the timestamp of the new message with the timestamp of the last message it received. If they are different, this means a new message has arrived. If they are the same, this means the message has already been received before. If the message is received before, it exists. If the message is a new message, it starts broadcasting the message. During this phase, we decided to set a limit on the number of neighbor nodes to deliver the message. We chose this because of energy saving purposes of smartphones. This limit is named as “maximum node count” in the algorithm. Each neighbor node stops broadcasting after delivering the message to the number of neighbor nodes which is set by the maximum node count.

4. Implementation

According to our design, two mobile applications were developed: master node and neighbor node. The applications were developed for Android OS. For the development platform, we used Android Studio 3.6. Supported minimum API level is 21 (Android 5.0). With this API level, the system will work on 85% of Android devices[#]. The applications were developed using Java programming language.

The screenshots of the mobile applications are shown in Fig. 2, Fig. 3 and Fig. 4. Log message interfaces were added intentionally in both applications to show the communication procedures. In the deployment phase, these log messages will be removed.

In Fig. 2 the screenshot of the master node is shown. As seen from the figure it is self-explanatory. The user (the officer of the relief organization) writes the message which will be delivered to the neighbor nodes in the message text box of the application. After the “start sending message” button is clicked the application starts broadcasting the message according to the algorithm written above. The logs of the broadcasting procedure are seen on the log text box.

In Fig. 3 and Fig. 4, the screenshots of the two different neighbor nodes are shown. These figures show the entire procedure from discovery to broadcast. When the user of the neighbor node clicks on the “start receiving message” button the application starts discovery. During discovery, it looks for a node that broadcasts a message. If it finds a broadcaster, then connects and gets the message from it. After that, it shows the message to the user in the message text box and begins to broadcast the message. The logs of the discovery and broadcasting procedures are seen on the log text box. The neighbor is shown in In Fig. 3 gets the message from the master node and also delivers the message to another neighbor node. The neighbor node is shown in In Fig. 4 gets the message from another neighbor node (the neighbor node which is shown in Fig. 3).

5. Evaluation and Discussion

In this section, we first explain the evaluation of the developed system. Then we discuss the advantages and challenges of implementing and deploying of such a communication mechanism.

5.1. Evaluation

After the implementation of the system, we performed several tests on real hardware (multiple smartphones running Android OS). Tests were performed both indoors and outdoors. During the tests, smartphones were not connected to any wireless networks or cell towers. Phones were operated in flight mode or without a SIM card. The connection between phones and message exchange were performed by using Bluetooth. The system was tested in three different scenarios. The scenarios are shown in Fig. 1. For each scenario, multiple tests were performed in different distances between nodes ranging from 1 meter to 10 meters.

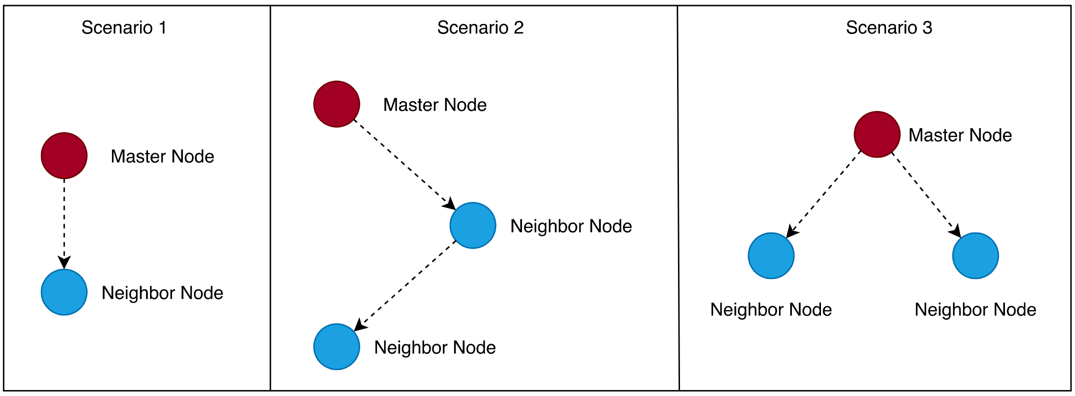

Figure 1. Test scenarios.

[#] This information was obtained from Android Studio during the API level selection.

In scenario 1, we performed tests with two smartphones: one master node and one neighbor node. The main aim of the tests in this scenario was the validation of the basic communication and messaging between the master node and neighbor node. During the tests broadcast, discovery, and message exchange operations were performed successfully.

In scenario 2, we performed tests with three smartphones: one master node and two neighbor nodes. The main aim of the tests in this scenario was the validation of the communication of multiple nodes in the form of master-neighbor and neighbor-neighbor. We tried to send the message to neighbor node from the master node and then to another neighbor node from receiving neighbor node. During the tests, the first neighbor node successfully discovered the master node and got a message from it. Then the same neighbor node again successfully delivered the newly received message to another neighbor node.

In scenario 3, we performed tests with three smartphones: one master node and two neighbor nodes. The main of the tests in this scenario was the validation of the communication of multiple neighbor nodes with one master node. We tried to send the message to multiple nodes from one master node. During the tests, message exchange between the master node and neighbor nodes was performed successfully.

The screenshots taken in scenario 2 are shown in Fig. 2, Fig. 3, and Fig. 4. In Fig. 2, the screenshot of the master node is shown. In the log window of the screenshot, the communication process of the master node with neighbor node is seen. In Fig. 3, two screenshots of the first neighbor node -the neighbor node which is connected to the master node- are shown. In the log window of these screenshots, the whole process of discovering the master node, getting message, broadcasting, and sending the message to another neighbor are seen. In Fig. 4, the screenshot of the second -last- neighbor node is shown. In the log window of the screenshot, the process of discovering, finding the first neighbor node, and getting a message is seen.

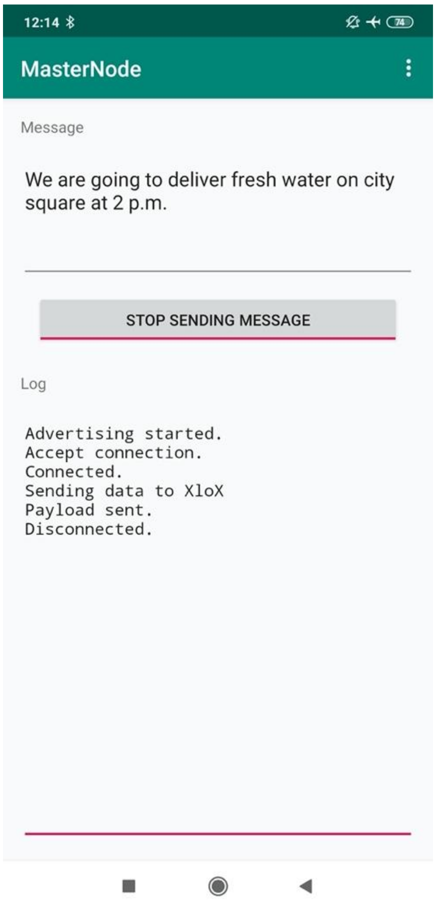

Figure 2. Screenshot of the master node.

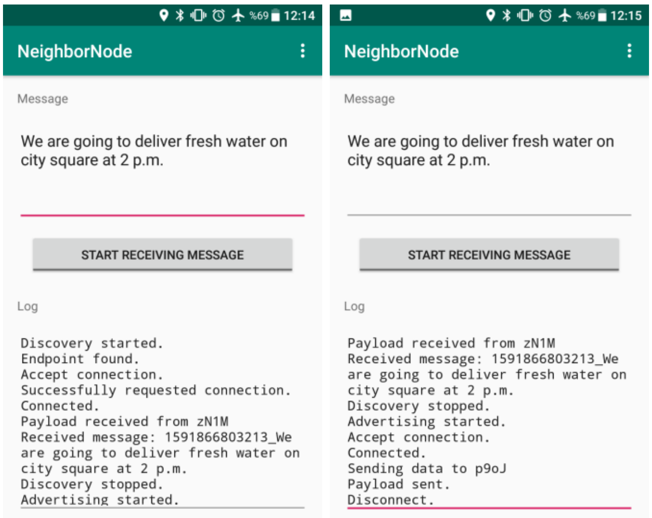

Figure 3. Screenshots of the first neighbor node.

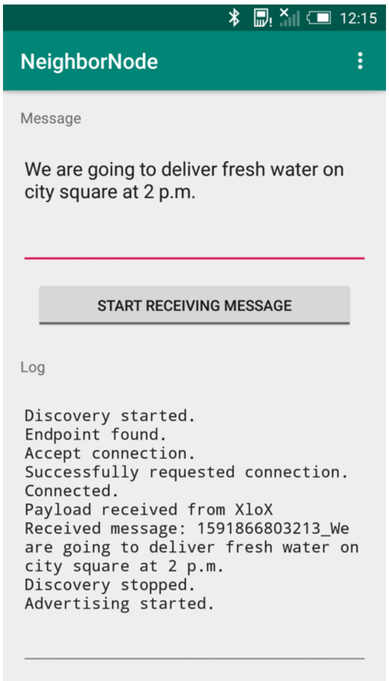

Figure 4. Screenshot of the second neighbor node.

5.2. Discussion

One of the key advantages of such a communication system that allows communication even the communication infrastructure is collapsed. The second advantage of the developed system is that it does not require additional specific hardware. This lowers the costs and eases the adaptation. The third advantage is using smartphones. Today smartphones are ubiquitous technology which most of the community has access. We believe that this will also ease the adaptation and will increase the coverage of the system. One of the important things of developing this kind of system is selecting the Android API level. As this is a solution for disasters, the API level should be selected to cover as many smartphones as possible.

As mentioned above Google Nearby Connections API was chosen to create the communication mechanism. Like all other API's, this API also may change over time. Therefore, the developers should be careful during the maintenance and updating of the applications. A permanent solution can be to communicate with the manufacturers and request from them to provide API's for disasters with long term support.

In these kinds of systems, nodes have to have unique identifiers. Android allows us to use different types (Android 2019). One of them is hardware identifiers. Currently, our system uses secure settings Android Id (SSAID) which is one type of hardware identifier. An alternative is to use instance id, but it requires an internet connection. Another one is to use advertising id whose main target is advertisement use cases. This id is resettable so it is not appropriate for this kind of use.

6. Conclusion

In this study, we designed and developed a message broadcasting system which helps communicating the relief organizations and community. The main purpose of the system is to enable the relief organizations to broadcast messages to the people in disaster area by using smart phones. Smart phones are everyday technology which most people in the community have access and this kind of coverage is very important for communication on disasters.

The developed system run and tested by using Android smart phones successfully. The results and experiences are shared. Different kind of communication systems which use special hardware can also be developed and used in disasters. But we believe that using COTS hardware is important because of their accessibility and prevalence. Therefore, getting help from COTS technology on disasters should be studied more.

References

Ahmed, A., Bakar, K. A., Channa, M. I., Khan, A. W., & Haseeb, K. (2017). Energy-aware and secure routing with trust for disaster response wireless sensor network. Peer-to-Peer Networking and Applications, 10(1), 216–237. https://doi.org/10.1007/s12083-015-0421-4

Antonioli D., Tippenhauer N. O., Rasmussen K. B. (2019). Nearby Threats: Reversing, Analyzing, and Attacking Google's 'Nearby Connections' on Android. Network and Distributed Systems Security (NDSS) Symposium 2019 (pp. 1-14). https://dx.doi.org/10.14722/ndss.2019.23367

Arbia, D. Ben, Alam, M. M., Kadri, A., Hamida, E. Ben, & Attia, R. (2017). Enhanced IoT-based end-to-end emergency and disaster relief system. Journal of Sensor and Actuator Networks, 6(3), 1–18. https://doi.org/10.3390/jsan6030019

Android. (n.d.). Best practices for unique identifiers. Retrieved August 7, 2019, from https://developer.android.com/training/articles/user-data-ids

Aschenbruck, N., Gerharz, M., Frank, M., & Martini, P. (2006). Modelling Voice Communication in Disaster Area Scenarios. In Proceedings. 2006 31st IEEE Conference on Local Computer Networks (pp. 211–220). IEEE. https://doi.org/10.1109/LCN.2006.322102

Chang-Richards, A., Seville, E., Wilkinson, S., & Walker, B. (2019). Effects of Disasters on Displaced Workers (pp. 185–195). https://doi.org/10.1007/978-3-319-92498-4_14

CRED. (2019). Natural Disasters 2018. Brussels, Belgium.

Eichinger T., Beierle F., Papke R., Rebscher L., Tran H. C., Trzeciak M. (2019). On gossip-based information dissemination in pervasive recommender systems. RecSys 2019 - 13th ACM Conference on Recommender Systems (pp. 442–446). https://doi.org/10.1145/3298689.3347067

Google. (n.d.). Google Nearby Connections API. Retrieved August 7, 2019, from https://developers.google.com/nearby/connections/overview

Guo, W., Huang, X., & Liu, Y. (2010). Dynamic relay deployment for disaster area wireless networks. Wireless Communications and Mobile Computing, (10), 1238–1252. https://doi.org/10.1002/wcm.679

Khan, A., Munir, A., Kaleem, Z., Ullah, F., Bilal, M., et al. (2020). RDSP: Rapidly deployable wireless ad hoc system for post-disaster management. Sensors (Switzerland), 20(2), 1–21. https://doi.org/10.3390/s20020548

Kim, S., Urata, Y., Koizumi, Y., & Hasegawa, T. (2015). Power-saving NDN-based message delivery based on collaborative communication in disasters. In The 21st IEEE International Workshop on Local and Metropolitan Area Networks (pp. 1–6). IEEE. https://doi.org/10.1109/LANMAN.2015.7114733

Kousky, C. (2016). Impacts of Natural Disasters on Children. The Future of Children, 26(1), 73–92. https://doi.org/10.1061/(asce)0733-9402(2009)135:2(25)

Krock, R. E. (2011). Lack of emergency recovery planning is a disaster waiting to happen. IEEE Communications Magazine, 49(1), 48–51. https://doi.org/10.1109/MCOM.2011.5681014

Lesk, C., Rowhani, P., & Ramankutty, N. (2016). Influence of extreme weather disasters on global crop production. Nature, 529(7584), 84–87. https://doi.org/10.1038/nature16467

Lu, X. (2018). Online communication behavior at the onset of a catastrophe: an exploratory study of the 2008 Wenchuan earthquake in China. Natural Hazards, 91:785–802. https://doi.org/10.1007/s11069-017-3155-1

Mase, K., Azuma, N., & Okada, H. (2010). Development of an Emergency Communication System for Evacuees of Shelters. In 2010 IEEE Wireless Communication and Networking Conference (pp. 1–6). IEEE. https://doi.org/10.1109/WCNC.2010.5506628

Meftah L., Rouvoy R., & Chrisment I. (2019). Testing Nearby Peer-to-Peer Mobile Apps at Large. In 2019 IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems (MOBILESoft) (pp. 1–11). IEEE. https://doi.org/10.1109/MOBILESoft.2019.00009

Menon, V. G., Pathrose, J. P., & Priya, J. (2016). Ensuring Reliable Communication in Disaster Recovery Operations with Reliable Routing Technique. Mobile Information Systems, 2016, 1–10. https://doi.org/10.1155/2016/9141329

Menon, V. G. (2019). Optimized Opportunistic Routing in Highly Dynamic Ad hoc Networks. https://doi.org/10.20944/preprints201810.0478.v1

Mezghani, F., & Mitton, N. (2017). Alternative opportunistic alert diffusion to support infrastructure failure during disasters. Sensors (Switzerland), 17(10). https://doi.org/10.3390/s17102370

Ministry of Internal Affairs and Communications. (n.d.). Retrieved August 7, 2019, from http://www.soumu.go.jp/main_content/000117676.pdf

Murugeswari, R., & Radhakrishnan, S. (2015). Reliable data delivery for emergency and disaster recovery in wireless mesh network. 2014 International Conference on Information Communication and Embedded Systems, ICICES 2014, (978), 1–6. https://doi.org/10.1109/ICICES.2014.7033950

Noguchi, T., & Fujii, T. (2000). Minimizing the Effect of Natural Disasters. Japan Railway & Transport Review, 10(March), 52–59.

Pogkas, N., Karastergios, G. E., Antonopoulos, C. P., Koubias, S., & Papadopoulos, G. (2007). Architecture Design and Implementation of an Ad-Hoc Network for Disaster Relief Operations. IEEE Transactions on Industrial Informatics, 3(1), 63–72. https://doi.org/10.1109/TII.2007.891307

Schultz, C. H., Koenig, K. L., & Noji, E. K. (1996). A Medical Disaster Response to Reduce Immediate Mortality after an Earthquake. New England Journal of Medicine, 334(7), 438–444. https://doi.org/10.1056/NEJM199602153340706

Shao, Z., Liu, Y., Wu, Y., & Shen, L. (2011). A Rapid and Reliable Disaster Emergency Mobile Communication System via Aerial Ad Hoc BS Networks. In 7th International Conference on Wireless Communications, Networking and Mobile Computing (pp. 1–4). IEEE. https://doi.org/10.1109/wicom.2011.6040479

Sun, J., Zhu, X., Zhang, C., & Fang, Y. (2011). RescueMe: Location-based secure and dependable VANETs for disaster rescue. IEEE Journal on Selected Areas in Communications, 29(3), 659–669. https://doi.org/10.1109/JSAC.2011.110314

Takasuka H., Hirai K., & Takami K., (2018). Development of a social DTN for message communication between SNS group members. Future Internet, 10(4), 1–20. https://doi.org/10.3390/fi10040032

Townsend, A. M., & Moss, M. L. (2005). Telecommunications Infrastructure In Disasters: Preparing Cities for Crisis Communications. New York, NY, USA. Retrieved from https://www.nyu.edu/ccpr/pubs/NYU-DisasterCommunications1-Final.pdf

Tuna, G., Mumcu, T. V., & Gulez, K. (2012). Design strategies of unmanned aerial vehicle-aided communication for disaster recovery. High Capacity Optical Networks and Emerging/Enabling Technologies. IEEE. https://doi.org/10.1109/HONET.2012.6421446

Uchida, N., Takahata, K., & Shibata, Y. (2011). Cognitive Wireless Network for Large Scale Disaster. In 2011 Third International Conference on Intelligent Networking and Collaborative Systems (pp. 362–366). IEEE. https://doi.org/10.1109/INCoS.2011.64