DEIMOSBC: A Blockchain-based System for Crowdsensing after Natural Disasters

Raunak Sarbajna
University of Houston
Houston, USA
rsarbajna@uh.edu

Christoph F. Eick
University of Houston
Houston, USA
CEick@central.uh.edu

Aron Laszka
NTT Research
Sunnyvale, USA
alaszka@uh.edu

Published in the proceedings of the 3rd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS 2021).

***Abstract*—For first responders entering into a post-disaster situation, there is usually a severe lack of up-to-date ground truth. The initial period of time has multiple sources of conflicting information coming in and creating confusion about the situation. The most important immediate requirement is to create a traversal map, highlighting navigable paths to victims of the disaster and possible hazardous locations. Due to infrastructure damage, it is hard for existing centralized geospatial portals to quickly update and provide this information, which has become outdated. IoT solutions that can be deployed without extensive preparation provide the capability to quickly acquire and disseminate essential information to rescue teams. In this paper, we present a decentralized system, named DEIMOSBC, that is able to provide such a mapping service faster and more reliably, utilizing the work of volunteers and relying on a blockchain backend that is based on an IoT system. Our solution utilizes the availability of modern smartphones with GPS receivers and processing capabilities to collect sequences of GPS locations and chain them into trajectories. These trajectory data are submitted as entries into a blockchain after cleaning them through a purpose-built smart contract. DEIMOSBC relies on the inherent robustness and distributed nature of a blockchain to make collating and assembling a map from these paths more accurate and less susceptible to disruption. We describe how DEIMOSBC would work for a hypothetical disaster scenario of a Category 5 hurricane striking an area of the Gulf of Mexico.**

***Index Terms*—Blockchain, crowdsourcing, Internet of Things, disaster management, map building**

I. INTRODUCTION

When a natural disaster occurs, there is an initial rush of information coming from the affected area, which is riddled with inaccurate and often inconsistent datapoints that need to be processed urgently. In order to deal with such fast evolving disaster and post-disaster scenarios, we need to be able to create a first-order approximation of the ground details quickly. Such information is critical for first responders and government officials to divert aid to regions that are in most need of support. A disaster-affected region, however, is not conducive to thorough surveys by external observers who could create such a report. The best and most logical source of information are the affected people themselves.

Crowdsensing [1] involves collecting sensor data from devices such as smartphones, which are already present in significant numbers amongst regular people going about their daily lives. They have a multitude of sensors, including GPS receivers, accelerometers, gyroscopes, ambient light sensors, and magnetometers. Each of these devices can give crucial information about the current state of an area. Modern IoT relies on the existing global network of smart devices and sensor hardware, which work using various protocols. They are a source of massive amount of data, utilized by various crowdsourcing platforms [2]. A prime example of a crowdsensing system that deals with post-disaster scenarios is Bhattacharjee et al.'s *Post-Disaster Map Builder*, which is a crowdsensed system for constructing digital pedestrian maps using a smartphone based DTN. This system generates a digital pedestrian map of the disaster affected area using battery-powered mobile handheld devices. In particular, trajectory traces are collected by volunteers, which are then periodically shared with other volunteers within the disaster affected area through a custom network solution. Pedestrian maps of the affected areas are gradually constructed by combining these traces over time. The proposed system has been evaluated through simulation and an actual testbed implementation. The results show that the system is capable of constructing digital pedestrian maps of disaster affected areas with high accuracy at the cost of marginal delay [3].

A problem emerges with the collating of multiple types of data and keeping them organized and referenced. Erroneous data, caused by malfunctioning sensors or untrustworthy actors, will contribute to lowering the accuracy of the information. As the size of the dataset increases, maintaining integrity will become a challenge.

A blockchain based system that works atop an IoT technology [1] provides a clear solution to some of these foundational problems. DEIMOSBC, our proposed framework, is designed to help rescue teams traverse a complex terrain with shifting conditions, while retaining active knowledge and receiving specific information regarding the situation, by quickly and efficiently creating navigable maps of the area. The system will be easy to deploy and manage using non-specialists first responders. DEIMOSBC uses a novel blockchain based response that works with greater efficiency than existing traditional edge-computing based systems and features decentralization as a core component. This system will have greater robustness, fault tolerance, and resilience towards traditional crowdsourcing difficulties and be easily accessible to the end user due to its decentralized nature.

II. Example Application Scenario: Category 4 Hurricane on the US Gulf Coast

After a major hurricane strikes land, there will be gale force winds, localized flooding due to rain, life-threatening storm surges, and general flooding due to levee breaks [4]. The immediate priority is to construct digital pedestrian maps of the disaster affected areas, which are needed to initiate post-disaster relief operations. In modern times, existing digital maps with local GPS support help disaster-management agencies with decision making. However, the process of mobilizing human resources in the disaster affected areas is often piecemeal and delayed, due to the intermittent connectivity after large-scale natural disasters. Due to severe damage in communication infrastructure, existing web-based digital mapping systems may become inaccessible. One of the priorities of the response is to get technicians and engineers to repair cellphone towers, fallen electricity poles, etc. Another challenge arises because of the way the road network conditions in such areas change drastically due to flood-related waterlogging, structural collapse, or incidental destructions like a landslide. This makes previously reliable existing analog (paper based) route maps of such areas obsolete in critical paths.

III. Requirements

For any post-disaster scenario, there are certain conditions that are universal. Any system that aims to provide some form of aid to first responders must be able to address these challenges and surpass them.

A. Assumptions

1) Disruption to communication networks: We assume that the disaster has a high likelihood of destroying a significant [5] amount of telecommunication infrastructure in the affected region and thus [6] existing communication methods are no longer reliable. Various forms of disasters can alternately destroy underground cabling, overground towers and street poles [7]. This means that we cannot rely on existing network towers to accurately and precisely deliver messages.

2) Obstructions to area infrastructure: We can assume [6] that power outages have rendered street lights, de-icers, traffic lights ineffective for an uncertain period of time, as this is common side effect of major disasters. This disrupts automated collection of ground truth in the immediate aftermath of the disaster, when people are most vulnerable. This also means that we have no way of independently verifying the current physical condition of the area in question.

3) Availability of communication devices: We assume that everyone who works in the field has access to some sort of communication device with some limited computational capabilities such as a cell phone or a low-power IoT device [8]. The device must have at least a Bluetooth transmitter/receiver or a WiFi chipset, enabling it to have short-range, ad-hoc wireless communication with similar devices. The device would thus be able to communicate with other, similar, devices, through any available communication channel using simple message-passing protocols.

Given that the initial assumptions are in effect, the objective of a system to deal with a disaster must fulfill the following objectives.

B. Objectives

1) Robustness: The system must be, above all, robust. This means that the generated result, which can take the form of a navigation map, population density estimates, etc., must at the very least be reliable. The results must be compiled from sources that are verified and reliable and incorrect results must not invalidate the complete output. As it can be expected that real-world data collection is messy and error-prone when a few processing entities fail, it becomes essential that the entire system degrades gracefully. Thus, the system must be able to appropriately handle missing data and dismiss any corrupted data, to a reasonable extent, and must not completely collapse.

2) Collaborative problem solving: Due to prevailing near-chaotic conditions within a post-disaster region, the system must be designed to be capable of distributed problem solving using a number of participants. The system must be able to take advantage of a number of volunteers pooling their resources, both human and technical, and be able to provide up-to-date information to the victims so that they can take the most appropriate action. The system must be able to scale quickly to deal an evolving situation and be able to provide results to anyone asking regardless of where it is being asked from. However, the system must not also rely on any one unified controlling authority, thus creating a central point of failure.

3) Decentralization: Decentralization appears as a natural solution to the two objectives described previously. It can be assumed that the there are no authorities we that we can reliably trust once the disaster has occurred and we have to look for one that has existed since before the disaster. It is assumed that any such authority becomes unavailable in the aftermath of the disaster. The system must thus neither require nor rely upon a central authority of any kind that is set up afterwards. If we do attempt to set up new unified controlling authorities, we risk creating a central point of failure. Furthermore, none of the processes must require external verification. Decentralization also eases dispute resolution through consensus mechanisms, as every participant has a copy of the relevant information.

IV. Proposed Approach

A. System Architecture

The objective of DeimosBC is to generate a complete and correct graph of accessible paths through a disaster affected region using data collected by volunteers equipped with IoT enabled devices. The blockchain itself would at any time contain the current partially complete graph. Each node of the graph is a point, which is submitted by a *data collector* as part of a *trace*. When the collector submits their point, the *data processor* runs a smart contract to check whether the created route is valid and adds it to the chain if so. Traces are paths created from the set of GPS locations collected by a volunteer. Each consecutive trajectory point is separated by a thresh-holding distance. These sets of traces that indicate a real world traversable path form local maps. A combination of all the local maps create a network of interconnected paths that will allow people to navigate a difficult terrain in the real world. This global map may be subtly different from existing real world maps, as it will not contain those paths that have been blocked due to some disaster.

The parties involved in the framework are:

1) Data Collector: All the data collectors are volunteers who are pre-verified and registered into the blockchain used by the system before they move out into the field. Their main job is collect and submit traces to a local data processor node. Once a trace is collected, it is broadcast to all local nodes that are within range. Then, it waits for confirmation of delivery and successful integration from at least one Processor. Until the confirmation is received, all subsequent traces are held in a queue. This process repeats until the data collector voluntarily stops or the trace management system (IV-B2) detects errors and disables the data collector. If Processors send back error messages, the Trace Management system tries to identify it and sends an alert to the user. If the number of errors exceeds a pre-set limit, the user must be re-authenticated.

2) Data Processor: These are specialized nodes that have advanced data processing capabilities. They haves several overlapping responsibilities:

1) *Package Verification.* Checks for integrity of incoming data packets. This includes whether the data was packaged correctly, i.e. whether all the requisite information was included and in the correct order.
2) *Author Identification.* Verifies whether the incoming data packet is from a authentic Data Collector. This is done by checking whether they were registered into the blockchain. Then send a message to the sender certifying that the data has been received.
3) *Trace Processing.* Receives the traces generated by the Trace management system, as detailed in IV-B2, and submits it for further processing to the Data Summarization System, as detailed in IV-B1. Following this, they would then connect to the blockchain and run the smart contracts that syncs the map. In case of errors, they would also send an alert back to the Collector who submitted it.
4) *Result Provisioning.* Whenever any third party requests an updated map through the system, they would fetch the latest global map and return it. This requester need not be verified.

At any point of time, the blockchain can contain multiple local maps. These maps are represented as graphs with each vertex containing: (i) which vertices it is connected with, (ii) real-world location, and (iii) any additional sensor data. These are created by individual Data Collectors submitting their data. Once all the processors are satisfied that no more updates are required, we make the map available for general public use. This process will repeat as many times as needed, generating up-to-date maps while there are still volunteers. The system is halted when the Data Processors detect that the system has enough errors to compromise the integrity of the map.

B. System Components

DEIMOSBC is an extension of the Bhattacharjee et al. post-disaster map builder set to work with an IoT solution. The system will construct a comprehensive digital pedestrian map of an entire region in a progressive approach, which requires first building local pedestrian maps of the disaster affected areas and then inferring the global pedestrian map by collating those local pedestrian maps. The primary components of DEIMOSBC are:

1) Data Summarization System: This is the fundamental module which is run by every data processor in DEIMOSBC. The main functionality of map inference module is creation of pedestrian maps from the trajectory traces collected by the nodes. The system looks at incoming traces and checks for overlapping positions. In case of overlap, the systems attempts to chain the traces together to create a large, interconnected trace. In case of conflict, both traces are discarded and a message confirming such is sent back to the original Collectors. As the pedestrian maps are constructed using mobile handheld devices, efforts have also been made to make the map inference mechanism computationally less intensive and energy efficient.

2) Trace Management System: This system facilitates the periodic collection of consecutive GPS points through sensors present in each volunteer node and creation of trajectory traces from such points. In our system, this collection of points creates a trajectory based on the displacement of volunteer nodes rather than time, as pioneered by Bhattacharjee et al. To smooth out abrupt direction changes on part of the collectors, we use *trajectory segmentation* [9].

C. Workflow

The workflow for DEIMOSBC takes advantage of blockchain's anonymized and decentralized approach to create a distributed crowdsensing system, as described in Fig 1.

- *Register*. All collectors and processors will need to register with the blockchain system at initialization. Each of these registered collectors will be assigned a pair of cryptographic keys, and their identities are noted. All the registration information is recorded as a transaction in the blocks.
- *Task Assignment*. This is an automated process. Once registration is complete, all collectors receive instructions about what data they are collecting and a starting point based on their geographic location. They might collect just GPS points or additional sensor data. They are then free to start perfomring their tasks. The *task* includes the starting trajectory point (which may be changed as per their need), the direction of the trace, the current time and the status.
- *Upload Sensory Data*. Workers upload the sensory data to the blockchain. The miners validate the quality of

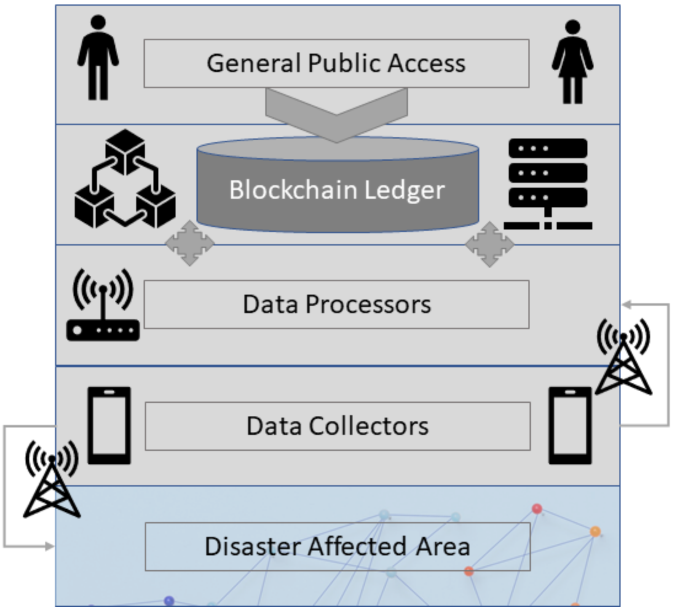

Fig. 1. Overview of the DEIMOSBC architecture.

the uploaded data. The qualified data is accepted and recorded, and the corresponding workers receive their rewards. If the sensory data is unqualified, the worker loses their deposit.

- *Smart Contract Creation*. To ensure correctness of the maps, the processor creates a smart contract, which runs automatically according to a predefined protocol. The processor also needs to define several rules for workers to ensure tasks are assigned appropriately and the quality of the uploaded sensory data, in accordance with the Data Summarization system IV-B1. As the collector needs to take the tasks from the public blockchain, they need to ensure a task is still available before assigning it to themselves. Availability in this case means a physical location that has not been mapped already. Therefore, a new smart contract is created to ensure each task can be assigned successfully.
- *Load tasks*. The processor downloads all the information related to the maps from the global map and posts all the information on their private blockchain network. The processor is responsible for maintaining the consistency of the task information on the blockchain (the global map and their own local map).
- *Broadcast*. If the current state of blockchain is satisfactory, any requester automatically gets a copy of the current product.
- *Retirement*. The processor constantly runs a check on the blockchain to ensure integrity is maintained. Once the number of errors exceeds a pre-arranged threshold, the blockchain is frozen and the final block is marked to ensure all subsequent blocks get automatically discarded. This freeze state is broadcast throughout the network from an authorized Processor and is amplified by every node who received it, whether Processor or Collector. Other Processors may run their own integrity tests to ensure the decision is correct. A majority vote is then undertaken to finalize the decision to freeze or not.

V. CONCLUSION AND FUTURE WORK

This paper presents DEIMOSBC, a framework for crowdsourced map-building following a major natural disaster using a blockchain based system. DEIMOSBC is designed to take advantage of the inherent robustness and distributed nature of a blockchain and places it within an IoT-enabled device. It can collate and assemble a map from crowdsensed data quickly and accurately without relying on a centralized authority of any kind. The system has two types of internal users, Data Collectors and Data Processors, who perform the basic tasks of collecting information form the real world, processing it, and then adding it to the blockchain for anyone to access. Internally, the system relies on two modules, the Data Summarization System and the Trace Management System.

The proposed system has not been deployed in a real-world setting yet. The area of blockchain-based crowdsensing remains an underexplored topic, so we are currently working on developing a minimum viable product and testing it through simulations to understand potential pitfalls. We aim to further introduce more advanced blockchain capabilities into the system, such as energy efficiency, complete confidentiality of users, and resource management. We also aim to eventually introduce an incentive mechanism within the system to reward volunteers for delivering better and more accurate reports.

REFERENCES

[1] K. Kucuk, C. Bayilmis, A. F. Sonmez, and S. Kacar, "Crowd sensing aware disaster framework design with IoT technologies," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 1709–1725, 2019.

[2] M. Durresi, A. Subashi, A. Durresi, L. Barolli, and K. Uchida, "Secure communication architecture for internet of things using smartphones and multi-access edge computing in environment monitoring," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, pp. 1631–1640, Apr. 2019.

[3] S. Bhattacharjee, S. Roy, and S. D. Bit, "Post-disaster map builder: Crowdsensed digital pedestrian map construction of the disaster affected areas through smartphone based dtn," *Computer Communications*, vol. 134, pp. 96–113, 2019.

[4] R. A. Pielke Jr, J. Gratz, C. W. Landsea, D. Collins, M. A. Saunders, and R. Musulin, "Normalized hurricane damage in the United States: 1900–2005," *Natural Hazards Review*, vol. 9, no. 1, pp. 29–42, 2008.

[5] World Economic Forum, "The global risks report 2020," tech. rep., World Economic Forum, January 2020.

[6] F. Kadri, B. Birregah, and E. Chatelet, "The impact of natural disasters on critical infrastructures: A domino effect-based study," *Journal of Homeland Security and Emergency Management*, vol. 11, May 2014.

[7] A. M. Townsend and M. L. Moss, "Telecommunications infrastructure in disasters: Preparing cities for crisis communications," tech. rep., Center for Catastrophe Preparedness and Response, Robert F. Wagner Graduate School of Public Service, New York University, April 2005.

[8] C. Han and S. Armour, "Computational complexity and energy consumption analysis of dynamic resource scheduling algorithms for LTE," in *75th IEEE Vehicular Technology Conference (VTC Spring)*, pp. 1–5, May 2012.

[9] U. Blanke, R. Guldener, S. Feese, and G. Troester, "Crowdsourced pedestrian map construction for short-term city-scale events," in *1st International Conference on IoT in Urban Space (Urb-IoT)*, pp. 25–31, ACM, November 2014.