Computer Communications 134 (2019) 96–113

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Post-disaster map builder: Crowdsensed digital pedestrian map construction of the disaster affected areas through smartphone based DTN

Suman Bhattacharjee [a,*], Siuli Roy [b], Sipra Das Bit [a]

[a] Indian Institute of Engineering Science and Technology, Shibpur, Howrah, India
[b] Heritage Institute of Technology, Kolkata, India

ARTICLE INFO

Keywords:
Post-disaster communication
Delay tolerant networks
Pedestrian network
Pedestrian navigation services
Crowdsensing

ABSTRACT

Constructing digital pedestrian maps of the disaster affected areas is a vital requirement for post-disaster relief operation. Digital maps help disaster management agencies in decision making and mobilizing the manpower and resources in the disaster affected areas. However, due to the intermittent connectivity after large scale natural disasters, the existing web based digital mapping systems remain inaccessible in the disaster affected areas. Moreover, the road networks in such areas change drastically because of disaster (for instance waterlogging due to flood, structural collapse or incidental destructions like a landslide). Hence, existing analog (paper based) route maps of such areas become obsolete. In this work, we propose Post-Disaster Map Builder, a crowdsenced digital pedestrian map construction system over smartphone based DTN. The proposed system generates digital pedestrian maps of the disaster affected areas using battery powered mobile handheld devices. Here, trajectory traces collected by the volunteers are periodically shared among the other volunteers of the disaster affected area through smartphone based DTN. Pedestrian maps of the disaster affected areas are gradually constructed in the mobile handheld devices of the volunteers by combining the collected traces over time. The proposed system is thoroughly evaluated through simulation and testbed implementation. The results reveal that our proposed system can automatically construct digital pedestrian maps of disaster affected areas with high accuracy at the cost of marginal delay.

1. Introduction

Catastrophic natural disasters are responsible for substantial human fatality and massive destruction of assets. Post-disaster relief operations are essential to mitigate the impact of disaster related destructions on human life. Such operations are accomplished by diverse response/relief groups. Depending on their areas of expertise, the groups establish various types of camps. Dedicated groups of volunteers or relief workers are associated with each camp who provide a specific category of service relevant to that camp. For example, volunteers connected with the Medical Camps offer medical assistance and those associated with the Relief Camp offer foodstuff, clothes and blankets etc. to the victims residing in different shelters. Map based navigation support is a crucial requirement for the post-disaster relief operations. Maps guide volunteers to their targets, leads displaced population (victims) to the shelters through unfamiliar terrain, and assist disaster management authorities during resource deployment. However, in the aftermath of the massive natural disasters like flash flood in Uttarakhand, India (2013), the earthquake in Nepal (2015), the road networks of the disaster affected areas change drastically. Many existing roads become inaccessible because of waterlogging due to flood, structural collapse or incidental destructions like landslide etc. Due to the altered road networks, offering navigation support to the volunteers, victims and disaster management authorities through the existing analog route maps (drawn or printed on papers) of the disaster affected areas becomes infeasible.

In addition to that, enormous natural disasters severely impair existing communication infrastructure. As a result cell phone/internet services instantly become dormant due to the destruction of the underlying infrastructure [1]. Therefore, web based digital mapping systems like Google Maps, Open Street Map, Bing Maps etc. immediately becomes inaccessible after disasters in the affected areas. The communication infrastructure generally gets restored in the disaster affected areas within one to seven weeks after the occurrence of disaster depending on its severity [2]. Moreover, in a post-disaster scenario, while the emergency response activities are typically completed within one to two weeks, complete reconstruction and redevelopment of damaged infrastructure (such as roads, buildings, bridges etc.) may take many years [3]. The web based digital mapping systems like Google Maps updates the underlying satellite images of the maps within one to three years in non-US regions [4]. Therefore, the underlying route maps of

* Corresponding author.
E-mail address: sumanbhattacharjee@acm.org (S. Bhattacharjee).

https://doi.org/10.1016/j.comcom.2018.11.010
Received 28 June 2018; Received in revised form 13 October 2018; Accepted 29 November 2018
Available online 5 December 2018

such digital mapping systems are not up to date. Those backdated maps typically do not reveal the altered road conditions of the disaster affected areas. Hence, suitability of such web based digital mapping systems for providing navigation support in the disaster affected areas even after the communication infrastructure is restored remains doubtful.

The aforesaid discussion exposes the challenges of offering navigation support to the volunteers, deployed in the disaster affected areas, through both analog and digital mapping systems over traditional communication infrastructure. These limitations of the existing navigation systems necessitate the development of an alternative digital pedestrian map construction system through some unconventional communication techniques. We focus on constructing pedestrian maps as walking remains the predominant mode of movement in post-disaster scenarios when roads become inaccessible and transit systems shut down [5]. Constructing pedestrian maps of the disaster affected areas in absence of stable network connectivity involve the periodic collection and sharing of the trajectory traces of the volunteers walking around the disaster affected areas. However, due to the incapacitated network connectivity, periodic sharing of trajectory traces becomes challenging in a post-disaster scenario.

In the recent years, the networking research community has strongly advocated the application of Delay Tolerant Networks (DTN) [6] for establishing emergency communication networks. DTN is an overlay architecture intended to operate above the existing protocol stacks in various network architectures and provide a store-and-forward gateway function between them when a node physically touches two or more networks. DTNs are characterized by long delays and intermittent connectivity. Moreover, they may have power constraints, low and asymmetric bandwidth, and high bit-error rates. One common example of DTN is forming a network using the vehicles on the roads of a city where the vehicles are equipped with a radio transceiver, which allows them to communicate with each other, and also to access points, which have connectivity to the Internet and are planted strategically in different parts of the city. The other example is to form a DTN using people carrying smart phones equipped with sensors and transceivers. However, unlike vehicle based DTNs, smart devices have resource constraints such as limited battery life. Moreover, people will have varying contact duration and frequency. It implies that their movement pattern will be less predictable than vehicles.

Fortunately, now-a-days majority of the adult population possesses numerous wireless devices, like smartphones and tablets etc. with multiple communication interfaces (Wi-Fi, Bluetooth, etc.) and sensors. These devices with their alternative communication capabilities can be harnessed in a disaster aftermath in the form of DTN [7]. The volunteers from different disaster response groups equipped with such devices can collect trajectory information through the periodic accumulation of GPS traces and share them with other volunteers in multiple hops. Such collection and sharing of trajectory traces through the GPS receivers of mobile devices is termed as mobile crowdsensing [8]. Hence, DTN proves to be viable communication option for realizing mobile crowdsensing in a post-disaster scenario. Trace collection from entire disaster affected areas over incapacitated network connectivity is time consuming. As a result, this mechanism requires a substantial amount of time to construct complete pedestrian maps of the disaster affected areas which severely impedes the effectiveness post-disaster relief operation. Therefore, our proposed work is aimed at constructing pedestrian maps gradually in a progressive manner using mobile handheld devices (such as smartphones, tablets etc.) so that post-disaster relief operation can make use this gradually augmented map.

1.1. Motivation

In the aftermath of catastrophic natural disasters (for example flood, earthquake, storm), the existing road networks of the disaster affected areas get altered drastically. Hence, majority of the existing routes remain inaccessible and navigation support through existing analog route maps could be ruled out. Moreover, massive natural disasters affect traditional communication infrastructure resulting inaccessibility of web based digital mapping systems. Due to lack of navigation support in a post-disaster scenario, the volunteers and displaced population find difficulties in reaching to their targets and shelters respectively through unfamiliar terrain which may result in unwanted loss of lives. Hence, map based navigation support is a crucial requirement for the post-disaster relief operations. Inferring comprehensive pedestrian maps of the disaster affected areas from the trajectory traces of the volunteers generally require high power computing resources. The availability of such resources could be ruled out in the post-disaster scenarios of the developing countries like India due to the unstable power supply and lack of skilled manpower to handle them. Therefore, the response/relief groups still rely on the analog method of map creation in post-disaster scenarios. Such maps are drawn or printed on pieces of papers based on the information provided by the volunteers in face-to-face meetings. The accuracy of such analog maps could not be assured due to human errors. Apart from that, updating and sharing of such maps among the volunteers become time consuming which hampers the effectiveness of the post-disaster relief operation. This acute need of map based navigation support motivates us to propose a digital pedestrian map construction system with the help of crowdsenced trajectory traces of volunteers over smartphone based DTN. As the pedestrian maps are to be constructed using battery powered mobile handheld devices which suffer from scarce computing resources as well as unstable power supply for recharging their batteries in a post-disaster scenario, the efforts have also been made to make the map construction mechanism computationally less intensive and energy efficient.

1.2. Contribution and organization

In this work, we propose Post-Disaster Map Builder, a crowdsenced digital pedestrian map construction system over the smartphone based DTN. The proposed system generates digital pedestrian maps of the disaster affected areas (for instance the explored traversable paths) in the mobile handheld devices. The generated maps are then overlaid on top of pre-cached offline map image in order to identify existing traversable paths. To realize the proposed system, we contribute towards developing the subtasks as follows:

- Each volunteer's trajectory traces are periodically collected and shared with other volunteers over an intermittently connected network within an acceptable delay
- Pedestrian maps of the disaster affected areas are gradually constructed in the mobile handheld devices of the volunteers by combining the collected traces over time. This involves the following:
    - Pre-processing of raw trajectory traces in order to reduce noise
    - Significant point identification and clustering
    - Topology inference

The rest of the paper is organized as follows. Section 2, summarizes related works in this field. In Section 3, we describe the system model related to our work. The proposed crowdsenced digital pedestrian map construction system is elaborated in Section 4. Implementation feasibility of the proposed system is evaluated through simulation based on real life data set and presented in Section 5. Section 6 presents the implementation of the proposed system into an Android application (app) as well as the evaluation of such app (Post-Disaster Map Builder) using testbed. We conclude the paper with a direction towards future work in Section 7.

2. Related works

Generating crowdsensed digital pedestrian maps of disaster affected areas through smartphone based DTN is dependent on two major tasks. One such task is periodic collection and sharing of the trajectory traces of the volunteers walking around the disaster affected areas, over intermittently connected networks. The second task is inferring digital pedestrian maps from such traces using mobile handheld devices. The former task involves the data exchange among the nodes over intermittently connected networks. Performance of data exchange is greatly affected by DTN routing protocols which have been extensively studied over the years. The latter task is related to inferring route maps from GPS traces and has also been widely studied in the literature. In the following, we review literature connected to the aforesaid areas.

2.1. DTN routing protocols

DTN routing protocols (both unicast and multicast) can be broadly categorized into two — forwarding-based (single copy) routing and replication-based (multicopy) routing. The forwarding-based routing protocols consume fewer network resources as only a single copy of a message exists in the network at any instant of time. On the other hand, replication-based routing protocols offer superior message delivery as multiple copies of a single message exist in the network while only one copy will reach the destination. Hence, the replication-based routing protocols have been widely favored among the DTN research fraternity due to its ability of improved message delivery compared to forwarding-based routing protocols. Various unicast and multicast routing protocols are proposed in DTN till date. Some of the widely accepted replication-based (multicopy) routing protocols are Epidemic [9], PRoPHET [10], PRoPHETv2 [11], MaxProp [12], Bubble Rap [13], SDM, MDM [14], EASDM and EAMDM [15] etc. Out of these routing protocols, [9–13] are designed for the unicast mode of communication and [14,15] are designed for the multicast mode of communication. Aforesaid routing protocols are reviewed below:

Epidemic routing protocols emulate the spread of an infection in a crowded population, and are both reliable and stable under forms of stress that will disable most traditional routing protocols [9]. This routing protocol is modeled based on the concept of spreading epidemic diseases in real life. The central objective of this protocol is to enhance the likelihoods of message delivery and minimize latency in the network. It assumes random movement pattern of the DTN nodes, but in real world scenarios nodes do not move randomly. To address this limitation regarding movement pattern, Lindgren et al. [10] propose a probabilistic routing protocol using history of encounters and transitivity (PRoPHET) which does not assume random node mobility. PRoPHET employs a greedy algorithm, which assumes a node forwards a replica of a message to another node only if the latter node has a greater chance of meeting the destination of the message compared to the former node. This protocol forecasts the mobility of nodes using a probabilistic metric called delivery predictability. Grasic et al. [11], discover that the delivery predictabilities used in PRoPHET may not be able to replicate the real-world dynamics in some cases. The authors mention several instances where in spite of high values of delivery predictabilities of some nodes, the messages are not forwarded due to lack of direct contact. In order to address such limitations of PRoPHET, Grasic et al. propose an improved version of the protocol PRoPHETv2.

Routing protocols like Epidemic, PRoPHET etc. do not incorporate any buffer management feature. Therefore, buffer overflow is likely to occur, which causes unwanted message drop. In order to eliminate this problem, Burgess et al. [12] propose MaxProp protocol for efficient routing of messages over DTN. MaxProp is based on prioritizing both the schedule of packets transmitted to other nodes and the schedule of packets to be dropped. These priorities are assigned depending on the path likelihoods to reach to the adjacent nodes according to historical data and also on several other mechanisms, including acknowledgments, a head-start for new packets, and lists of previous intermediaries. The acknowledgment mechanism of MaxProp evades the possibility of buffer overflow and scale down network overhead by eliminating the replicas of the delivered messages from the network.

The aforesaid routing protocols [9–12] do not take into account the social properties of the nodes while making routing decisions. Hui et al. [13] exploit the idea of social associations among the people and consequently propose a message forwarding protocol termed as Bubble Rap. Two social metrics centrality and community are merged together in this protocol for taking message forwarding decisions. This protocol chooses high centrality nodes and community members of destination as relay nodes.

All the routing protocols discussed in [9–13] are designed for unicast communication. However, several multicast routing protocols have also been proposed in DTN. Gao et al. [14] study single, as well as multiple data multicasting problems with an aim to reduce the cost of data distribution utilizing social properties of the nodes. Delivery cost is minimized by minimizing the number of relays used. In this work, the relay selection problem is formulated as a unified knapsack problem. In order to deliver data to relays are selected depending on the centrality of the nodes for single data multicast (SDM). On the other hand, for multiple data multicast (MDM), relays are selected exploiting social community structures.

Galluccio et al. [15] propose a multicast polymorphic epidemic routing scheme to deliver data in DTN. In this work, the similarity of interest of the nodes is employed for the data forwarding and adaptive recovery strategies are introduced to decrease network overhead.

Roy et al. [16] propose an energy aware single as well as multiple data multicast scheme. The scheme chooses relay node depending on the centrality and residual energy. The proposed protocol is an improvement of [14] with respect to energy efficiency. In order to improve the lifetime of the network, a load balancing approach is incorporated in this work while selecting relays based on centrality. This mechanism makes the protocols energy efficient.

In our application domain (post-disaster scenario), trajectory traces collected by volunteers need to be shared with multiple destinations (other volunteers, camps or ferries) in multicast mode of communication to construct pedestrian maps of the disaster affected areas. Hence none of the existing DTN unicast routing protocols from [9–13] prove suitable for periodic sharing of trajectory traces. The routing protocols discussed in [14–16] support multicast communication which forward messages based on the social properties of nodes and thereby reduce network overhead. However, the efficiency in terms of reduced network overhead is achieved at the cost of decreasing delivery ratio of messages to the destinations as well as the increasing delay. Therefore, none of the aforesaid multicast routing protocols [14–16] fit well for periodic sharing of trajectory traces.

2.2. Inferring route maps from GPS traces

Several approaches have been proposed till date to automatically infer route maps from trajectory traces. The works describing such approaches can be classified into three major categories, namely point clustering [17,18], incremental track insertion [19,20] and intersection linking [21,22].

The point clustering approach clusters the trajectory traces in different ways to identify the cluster seeds which typically corresponds to the road intersections. The seeds are then linked to form road segments. Edelkamp et al. [17] introduce an approach to infer road maps with high geometric accuracy, containing detailed lane information from fine-grained GPS traces. This is the first approach to construct a route map based on k-means clustering method. Based on this work, Schrödl et al. [18] propose a process to enhance the intersections based on the transition relationships between traces and segments. This enhancement is achieved by identifying intersections and their bounding boxes. Traces through an intersection are then grouped by entry and exit points, and

a spline-fitting technique is used for each group to produce the final turn-lane geometry of the intersection.

Incremental track insertion approach constructs route maps by gradually inserting tracks into an empty map. Precisely, this approach uses the first track as a base map and enhances it cumulatively by inserting more tracks. With each insertion, new map details are added and the existing geometry is amended using interpolation. Cao et al. [19] propose an incremental track insertion approach that models the street map as a directed embedded graph with one directed edge per direction. The approach employs a clarification step which modifies the input tracks by applying physical attraction to group similar input tracks together. Then tracks are inserted incrementally by using local criteria such as distance and direction. Although their method exhibits satisfactory outcome on densely sampled trajectory traces, it proves to be non-robust towards noisy data. Based on the work of Cao et al., Wang et al. [20] present a novel approach for generating routable road maps from GPS traces. They introduce circular boundaries to isolate points near intersections and use k-means clustering method to refine the intersections. This method is also intended for densely sampled GPS traces.

Intersection linking approach emphasizes on the accurate identification of the intersections of the road and then links the intersections according to the raw traces. As opposed to other map construction approaches, the intersection vertices are identified initially in this approach and the vertices are linked together with edges in later stages. Intersections are identified based on movement characteristics (speed, direction) or point density. Fathi et al. [21] develop a classifier to find the intersections from GPS traces. After finding the intersections, they used the raw traces to prune and connect the intersections. Their method works well on road networks with simple intersections and densely sampled traces. Karagiorgou et al. [22] propose a similar algorithm consisting of the identification and connection of intersections and the network graph reduction. The algorithm uses a speed threshold and changes in direction as indicators to recognize the intersections. The traces around the intersections are then clustered in terms of the distance to determine the turns of the intersections. Finally, the links (road segments) are generated and compacted along the paths of the traces. Although this method uses comparatively less frequently sampled data (>30 s), this method is also sensitive to the sampling rates of the GPS traces.

Most of the aforesaid works [17–22] have various impractical assumptions about GPS data, including low noise, high sampling frequency and uniform distribution of traces (which implies passing through each road exactly once). Moreover, majority of the works discussed above collect densely sampled trajectory traces from moving vehicles using survey grade GPS receivers. But, in a post-disaster scenario, walking or using non-motorized vehicles (bicycles) are the predominant mode of trajectory trace collection. Moreover, the traces are collected by employing the GPS receivers of the mobile handheld devices (such as smartphones, tablets etc.). The GPS receivers of such devices do not offer fine-grained trajectory traces. Kasemsuppakorn et al. [23] first use GPS traces collected by pedestrians to construct route maps. They propose an algorithm for automatically constructing pedestrian networks using multiple GPS traces. However, in this work volunteers are instructed to walk continuously, resulting in clear trajectory traces. The assumption of continuous movement of volunteers in a post-disaster scenario is unrealistic. In order to overcome such limitations, Blanke et al. [24] propose a map construction approach from crowdsourced GPS traces which infer route maps from noisy GPS data with a high variation of walking behavior of the volunteers. However, this approach deals with large scale data set which includes around 25 million GPS data points from 28000 users. The availability of such huge data set could be ruled out in a post-disaster scenario. Qiu et al. [25] propose a map construction approach from the sparsely sampled GPS traces using a segmentation-and-grouping framework. The main idea of this work is to partition the points of the whole GPS traces to road segments, and then group the segments to form a long road. The algorithm is robust to noise and sampling rate.

All the aforesaid works [17–25] assume the existence of stable network connectivity while constructing route maps from GPS traces. To the best of our knowledge only one work by Trono et al. [26] has been reported for constructing route map from GPS traces under unstable network connectivity in disaster affected areas.

Trono et al. [26] propose an approach of constructing the pedestrian maps of the disaster affected areas across DTN assuming the existence of two types of nodes namely sensor nodes and computing nodes. The sensor nodes are mobile handheld devices employed to collect GPS traces. Such collected traces are forwarded to the computing nodes (PCs) over DTN, where route maps are constructed. In order to infer the maps, they use the map inference approach proposed by [24]. In this work authors also introduce a load balancing heuristic that distributes processing load to multiple computing nodes. The inferred route maps from the computing nodes are then delivered to the sensor nodes over DTN.

The works reported in [17–25] assume the existence of stable network connectivity and high power computing resources in order to construct route maps. However, in a post-disaster scenario such assumptions prove unrealistic. In [26] although the authors consider unstable network connectivity, the assumption regarding the existence of high power computing resources remains unaltered. Again, the availability of PCs/high power computing resources could be ruled out in the post-disaster scenarios especially in developing countries like India due to the unstable power supply and lack of skilled manpower (for example volunteers, local people) to handle such devices. Thus, none of the existing map inference approaches [17–26] prove suitable to automatically construct route maps from trajectory traces in a post-disaster scenario. The aforesaid shortcomings of the existing works motivate us to propose a digital pedestrian map construction system over smartphone based DTN which neither requires any high power computing resources nor requires stable network infrastructure.

3. System model

This section provides the description of the system model including network architecture considered in this work along with assumptions. In a post-disaster scenario, displaced population (victims) normally take shelter in nearby safe areas like school buildings, temporary tents in some highland areas and other risk free zones. Disaster response groups like Medical Support Group, Relief Group, Water and Sanitation Group, etc. establish camps in and around the disaster affected areas. Mobile devices with extended secondary storage and larger battery life (preferably tablets) are usually placed in the camps. Each camp has a dedicated group of volunteers equipped with smartphones. After setting up the camps, volunteers perform various tasks such as exploring adjacent areas for assessing damage, estimating resource requirements and rescuing victims to shelters. Apart from that, every volunteer periodically visits the shelters from their respective camps and provides specific type of services related to that camp. For example, volunteers associated with the Medical Camp provide medical aid and those associated with the Relief Camp provide foodstuff, clothes and blankets etc. to the victims residing in different shelters. They move predominantly on foot [5] or use low speed non-motorized vehicles (bicycles) to perform such tasks and return to their respective camps after completion of the task. A command and control centre is set up, away from the disaster struck areas, for allocating and mobilizing resources to different camps according to the needs of the corresponding camps. Supply vehicles equipped with smartphones periodically visit different camps form the command and control centre to distribute allocated resources. The following are the major types of nodes available in a post-disaster scenario.

- ***Volunteer nodes***: Volunteers equipped with mobile handheld devices working in the disaster affected areas.
- ***Camp nodes***: Mobile devices with extended storages placed at camps.

- ***Ferry nodes***: Supply vehicles equipped with mobile handheld devices visit camps from the command and control centre.

Natural disasters usually do not affect an entire region uniformly. Some parts of a region are severely affected due to their topography. Such areas are termed as disaster affected areas in disaster management vocabulary. Those disaster affected areas are often scattered within a region. In disaster affected areas, many existing roads become inaccessible due to incidental destructions. Thus, alternative routes to reach the shelters from the camps avoiding such inaccessible roads are often identified by the volunteers with the help of local people. The volunteers carrying smartphones collect trajectory traces periodically with the help of GPS receivers while moving to their destinations through both the accessible roads as well as the alternative routes.

3.1. Network architecture

Fig. 1 presents the framework for collection and exchange of trajectory traces among various nodes in the disaster affected areas. The figure reveals that there are two distinct disaster affected areas in the region. Each of such areas consists of a relief camp, a medical camp and several shelters. The command and control centre is situated at a faraway location from the disaster affected areas. There are total 10 *volunteer nodes* from different camps working in those two areas. A *ferry node* (supply vehicle in Fig. 1) visits different camps of the disaster affected areas from the command and control centre through the existing accessible roads. Trajectory traces (group of trajectory points) collected by the *volunteer nodes* are stored in their local storage (secondary storage of smartphones). The trajectory traces are generally shared among a group of nodes. The *volunteer nodes* belonging to a specific disaster affected area along with the *camp nodes* and *ferry nodes* typically form the potential group of receivers of such traces in a post-disaster scenario.

In this work, we consider Delay Tolerant Networks (DTN) with both static and mobile nodes. Out of the three types of nodes, *volunteer nodes* and *ferry nodes* are mobile nodes whereas the *camp nodes* are static nodes. The exchange of trajectory traces takes place when two such nodes come within the proximity of each other. The mobility characteristics of the *volunteer nodes* and *ferry nodes* are governed by the post-disaster mobility (PDM) model [27]. The *ferry nodes* enable the long range communication of trajectory traces over DTN.

3.2. Assumptions

We assume all the *volunteer* and *ferry nodes* are equipped with GPS receivers. The *ferry nodes* visit *camp nodes* from the command and control centre once in a day. In addition to that, all the nodes have same initial energy and identical energy requirement for transmission and reception of trajectory traces. In order to conserve the battery power, we assume the *volunteer nodes* collect trajectory points after a displacement of 20 m from its earlier location. The displacement between two consecutive trajectory points is calculated using Euclidean distance. The exchanges of trajectory trace among various nodes are governed by following assumptions.

- *Volunteer nodes* can exchange trajectory traces of a specific disaster affected area with other *volunteer nodes* and *camp nodes* belonging to the same area. They can also share trajectory traces with the *ferry nodes* as well as the *camp nodes* belonging to other disaster affected areas.
- *Camp nodes* can exchange trajectory traces of a particular disaster affected area with *volunteer nodes* belonging to the same area. They can exchange trajectory traces irrespective of the disaster affected areas with *ferry nodes*. However, the trajectory traces of other disaster affected areas are retained by the *camp nodes*. Those traces are not shared with any *volunteer nodes*.
- *Ferry nodes* can exchange trajectory traces irrespective of the disaster affected areas with other *ferry nodes* and *camp nodes*. They cannot share trajectory traces with *volunteer nodes*.

The aforesaid assumptions facilitate the gradual accumulation of trajectory traces from a particular disaster affected area at *volunteer nodes* of the same area. Moreover, such assumptions also enable the accumulation of trajectory traces from diverse disaster affected areas of an entire region at *camp nodes* as well as the command and control centre.

4. Proposed Post-Disaster Map Builder system

This proposed system comprises of *trace management* and *map inference* modules. This section describes the functionalities of both the modules along with their interoperability. Fig. 2 represents the modular architecture of our proposed system. Proposed system needs to be pre-installed in *volunteer nodes*, *camp nodes*, *ferry nodes* as well as in command and control centre. The *trace management* module facilitates periodic collection of trajectory traces by the volunteer nodes. In addition to that, this module also enables the exchange of trajectory traces among various nodes (*volunteer nodes*, *camp nodes* and *ferry nodes*) in disaster affected areas. This module performs two subtasks that is *trace collection* and *trace interchange*.

The *map inference* module is responsible for constructing pedestrian maps from the collected trajectory traces. This module works by performing three subtasks such as *pre-processing*, *significant point clustering* and *topology inference*.

4.1. Trace management

Construction of the comprehensive digital pedestrian map of an entire region in a progressive approach comprises of the following steps:

- Construction of local pedestrian maps of the disaster affected areas
- Inferring global pedestrian map by collating the local pedestrian maps

In order to construct the local pedestrian map of a specified disaster affected area, the *volunteer nodes* must share the trajectory traces collected by them with other *volunteer nodes* as well as the *camp nodes* belonging to that area. The exchange of trajectory traces within a disaster affected area is performed by employing a group based publish–subscribe mechanism [28]. In our application domain, *volunteer nodes* are the potential publisher of trajectory traces and other nodes (volunteer or camp) belonging to the same area are the potential subscriber of such traces. In this way the traces from a particular disaster affected area can be accumulated gradually at *volunteer nodes* and *camp nodes* of the same area. The local pedestrian map of a disaster affected area can be inferred by collating such traces.

In order to construct the global pedestrian map of the entire region, *camp nodes* share (publish) the trajectory traces retained by them with *ferry nodes*. They also subscribe the traces collected by the *ferry nodes* from other *camp nodes* belonging to the same or diverse disaster affected areas. In this manner the trajectory trace from diverse areas can be accumulated gradually at *camp nodes*. When *ferry nodes* return to their origin (command and control centre), the trajectory traces collected by them from different *camp nodes* reach to the command and control centre. Those traces can be collated to construct the global pedestrian map of the entire region. The global pedestrian map assists the disaster management authorities to assess the extent of damages to the road networks in the affected areas and plan the relief operation efficiently. Sharing of trajectory traces among various nodes with reduced network overhead requires group based multicasting (*groupcasting*) of such traces. Hence, in this module trajectory traces are shared in groupcast mode.

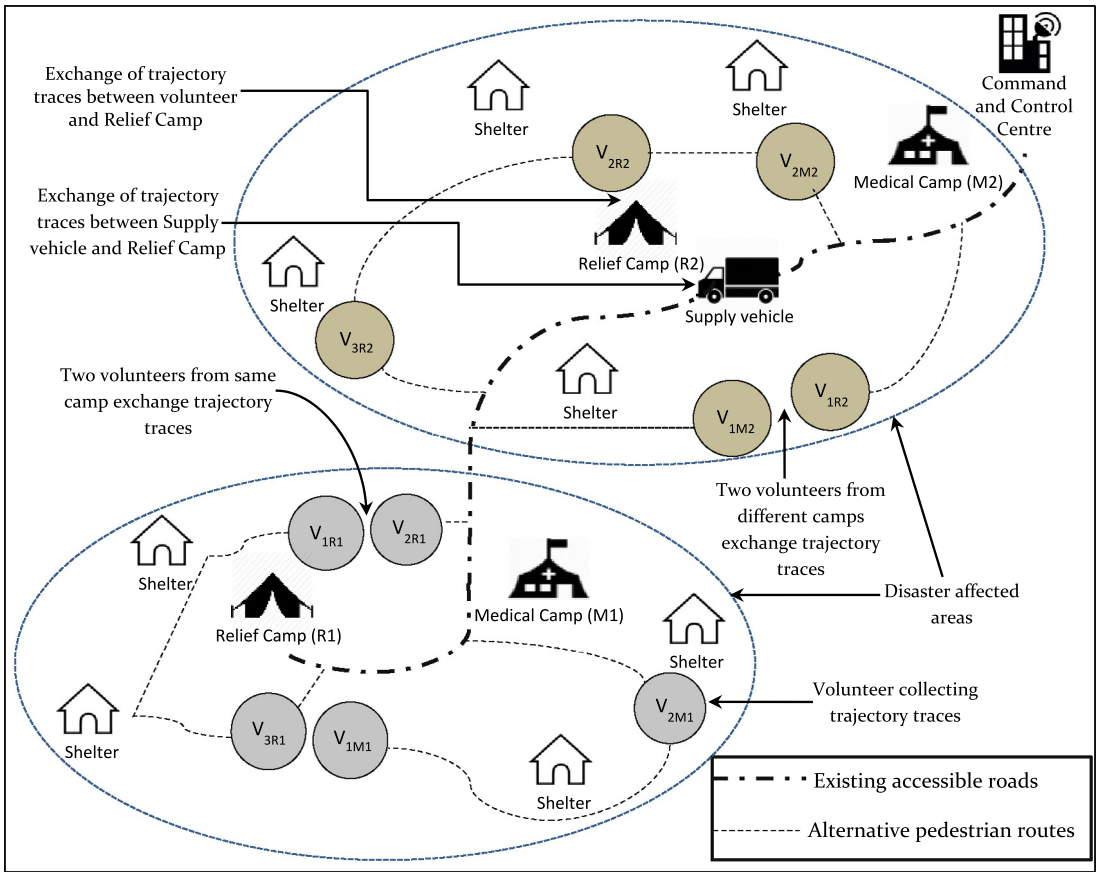


Fig. 1. Framework for collection and sharing of trajectory traces in a post-disaster scenario.

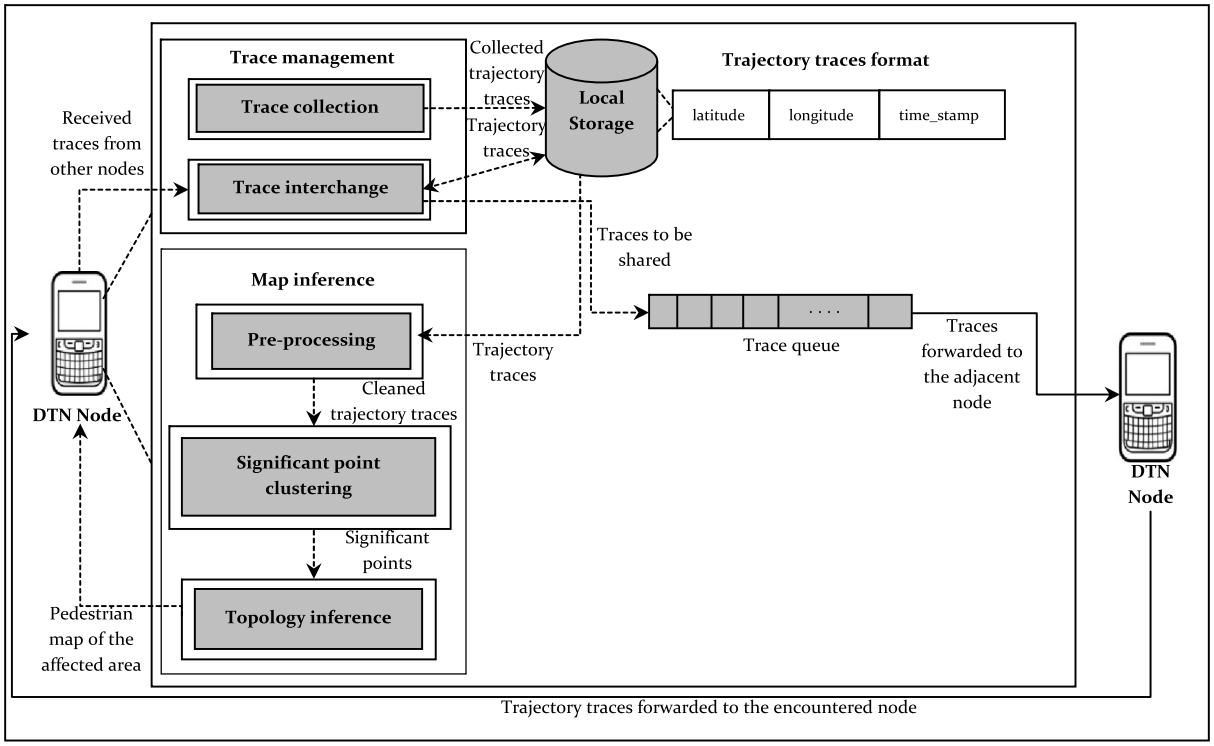


Fig. 2. Modular architecture of the proposed Post-Disaster Map Builder system.

4.1.1. Trace collection

This sub-module facilitates the periodic collection of trajectory points through GPS receivers present in each *volunteer node* and creation of trajectory traces from such points. As opposed to most of the existing works [23,24,26] on pedestrian map inference, the collection of trajectory points in our work is based on the displacement of *volunteer nodes* rather than time. In addition to that, the collected points are sparsely sampled. In reality, the *volunteer nodes* do not move at a uniform speed. During the course of their journey they tend to stop occasionally. These sporadic stoppages make the trajectory points susceptible to random noise. Hence, in this work, trajectory points are collected for each *volunteer node* depending on a predefined displacement (20 m) from their earlier locations. This mechanism of trace collection not only reduces the possibility of occurring random errors in the collected trajectory points due to arbitrary halts but also reduces energy consumption due to periodic polling of GPS receivers. In addition to this, sometimes displacement between two consecutive trajectory points becomes much greater than 20 m due to the missing/bad GPS signal. This sporadic gap between the consecutive trajectory points produces erroneous trajectory traces. In order to avoid such gaps in trajectory traces, we segment trajectory up to that gap and generate new trajectory traces. Inspired by [24] we use identical thresholding distance (30 m) between consecutive trajectory points for trajectory segmentation.

Trajectory traces generated at a node till a specific time can be uniquely identified by $Trace\_t_{creator\_id}$ where *creator_id* corresponds to the unique node identifier of the originator node and t is the time stamp of trajectory traces creation. Each $Trace\_t_{creator\_id}$ consists of a sequence of trace points (p). Each of such points is represented as 3-tuple $p = (lat, long, t)$ where *lat* and *long* correspond to the latitude and longitude of the point obtained at time stamp t.

The trajectory traces generated at *volunteer nodes* are retained in the local storages (secondary storage of mobile handheld devices) of the originator nodes. The traces received from other nodes during the trace exchange are also stored in the local storages of the respective nodes. Each node maintains a trace queue which holds the trajectory traces to be forwarded to the encountered node.

4.1.2. Trace interchange

This sub-module is responsible for exchanging trajectory traces among various nodes in the network and is present in every node. The sharing of trajectory traces in *groupcast* mode corresponds to the epidemic propagation, as all the *volunteer nodes* belonging to a specific disaster affected area along with the *camp nodes* and *ferry nodes* are the potential receiver of such traces. The Epidemic routing protocol [9] becomes beneficial in this scenario. Epidemic routing protocol is modeled based on the concept of spreading epidemic diseases in real life. This protocol distributes the replicated messages to each encountered node with an objective to enhance the likelihoods of message delivery and minimize latency in the network. However, in DTN, Epidemic routing protocol is designed primarily for unicast communication which results in exorbitant network overhead and energy consumption compared to the existing multicast routing protocols [14–16]. Hence, traditional Epidemic routing protocol does not prove suitable in our application domain.

In this work, we modify the existing Epidemic routing protocol and propose *Trace Route* protocol to facilitate the *groupcasting* of trajectory traces among various nodes. Unlike Epidemic routing protocol, in *Trace Route* protocol every $Trace\_t_{creator\_id}$ is forwarded to a predefined group of destinations (*destination group*) based on the location of the creator node. For instance, each *trajectory traces* collected by a *volunteer node* will be shared with the other *volunteer nodes* belonging to that disaster affected area as well as *camp nodes* and *ferry nodes*. Hence, all such nodes form the destination group of such trajectory traces. Each node in the network maintains a list of the trajectory traces ($Trace\_list_{node\_id}$) retained by them, where *node_id* is the unique identifier of the current node. The $Trace\_list_{node\_id}$ of a node comprises of a set $\{Trace\_t_{creator\_id} : Trace\_t_{creator\_id} \in Trace\_list_{node\_id}\}$. When two nodes ($A$ and B) encounter, they exchange their respective $Trace\_list_{node\_id}$. If A (source node) intends to transfer trajectory traces to B (encountered node) and vice versa, the set difference operation is performed on the trace lists at both the nodes as follows.

$$Trace\_list_{source} - Trace\_list_{encountered} = \{Trace\_t_{creator\_id} \in Trace\_list_{source} | Trace\_t_{creator\_id} \notin Trace\_list_{encountered}\} \quad (1)$$

After this operation, each node can determine the trajectory traces unavailable with the encountered node. Subsequently, the *destination groups* of the selected trajectory traces are checked against the encountered *node_id*. The trajectory traces whose *destination group* comprises the encountered *node_id* are dispatched to the trace queue for sequential transfer. Hence, the *Trace Route* protocol eliminates the possibility of unintended replication of trajectory traces as traces are forwarded only to their potential receivers. This mechanism reduces the communication overhead considerably and thereby decreases the average energy consumption of the entire network. The communication overhead is reduced at the cost of computation overhead.

The *trace interchange* sub-module employs the *Trace Route* protocol to exchange trajectory traces among different nodes. It is triggered automatically upon encounter with other nodes. The algorithm of the *trace management* module is given below:

4.2. Map inference

This module is present in every node (volunteer, camp and ferry) of the network and is triggered upon the request from the respective user. The main functionality of *map inference* module is creation of pedestrian maps from the trajectory traces collected by the nodes. As the pedestrian maps are constructed using mobile handheld devices, the efforts have also been made to make the map inference mechanism computationally less intensive and energy efficient.

4.2.1. Pre-processing

This sub-module is responsible for the cleaning and smoothing of trajectory traces from various noises. In this work, we propose a computationally less intensive and energy efficient data-driven approach for smoothing trajectory traces. The accuracy of trajectory traces collected by the GPS receivers of the mobile handheld devices in outdoor locations typically should lie between 5 and 8.5 m [29]. However, actually, it is much worse (sometimes up to 30 m) due to the multiple error sources. The following are the sources of errors in the trajectory traces obtained from the GPS receivers of the mobile handheld devices.

- **Warm start/cold start problem**: The GPS receivers generally require signals from at least four satellites to accurately calculate position. This signal acquisition process occasionally takes time. This results in missing points at the beginning of the trajectory traces.
- **Satellite clocks**: The inaccuracies in satellite clocks result in the reduced precision in positioning the GPS points.
- **GPS receiver clock**: The inaccuracy in receiver's clock reduces the precision of GPS points in trajectory traces.
- **Atmospheric effects**: Disturbances in the ionosphere and inconsistencies in the atmosphere affect the speed and direction of GPS signals. This results in inaccuracies in positioning the GPS receiver.
- **Multipath effects**: The GPS signal is often reflected by large buildings, walls or surfaces in urban areas. As a result, the corresponding GPS positions are usually displaced from the real position of the GPS receiver.
- **Signal blocking**: GPS signals are typically weak and hence can be blocked simply by materials like plastic, metal etc. The signal blocking results in missing GPS points.

Algorithm : Trace management
Input: Trajectory traces ($Trace\_t_{creator\ id}$) and Trace list ($Trace\_list_{source}$) of the source node
Output: Exchange of trajectory traces ($Trace\_t_{creator\ id}$) between encountered nodes
Algorithm Triggered: Upon request from user

```
Begin
    call trace_collection(p_last)                    */ collect trajectory traces based on displacement */
    call trace_interchange(node_id_encountered)
End

// Function
trace_collection(p_last)
displacement_limit = 20                              */ node displacement_limit is set to 20 meters */
segmentation_threshold =30                           */ segmentation_threshold is set to 30 meters */
d = Euclidean_Dist(p_now, p_last)                    */ d=Euclidean distance between p_now and p_last */
if (d <= segmentation_threshold) then
   if (d >= displacement_limit) then
      temp←p_now                                     */ Store current trajectory point (p_now) in temp*/
      p_last = p_now
   end if
end if
   Trace_t_creator id←temp
return

// Function
trace_interchange(node_id_encountered)
fetch Trace_list_encountered                         */ by exchanging trace list with encountered node */
trc_list[i ] = set_difference(Trace_list_source, Trace_list_encountered) */ list of trajectory traces using Equation (1) */
for each Trace_t_creator id in trc_list[i]
   fetch dest_group[j ]                              */ destination group of trajectory traces Trace_t_creator id */
   while dest_group[j ]is not empty
      if (node_id_encountered == dest_group[j]) then */ encountered node is in the destination list of T_id */
         queue← Trace_t_creator id                   */ insert trajectory trace into trace queue */
      end if
      j++                                            */ incrementing loop counter */
   end while
end for
while (queue is not empty) do
   Forward trajectory traces to node_id_encountered
end while
return
```

The complete elimination of errors from trajectory traces for all the six aforesaid error sources is infeasible in reality. The mechanism for elimination of errors from the satellite clocks and GPS receiver clocks are only included in the standard GPS receivers. As a result, the raw trajectory traces need to be cleaned before map construction. In practice, standard smoothing techniques, such as kernel smoothers, smoothing splines, generalized additive models and Kalman filters are widely used to clean and smooth trajectory traces collected from GPS receivers during map construction. However, these methods are model-driven (prediction based) which do not preserve the specific structure of the trajectory traces. Therefore, implementing data-driven (statistical) approach for smoothing trajectory traces is essential.

Chasel et al. [30] propose a data-driven trajectory smoothing technique which smooths trajectory traces by embedding each trajectory point p of the trace into a high dimensional space as p' using sample points preceding and following p. This process of transforming p into p' is termed as lifting of p. Each point p' in the lifted image of the trajectory is then moved towards its nearest neighbors in the lifted space (Laplacian smoothing). Finally, the original traces are recovered by averaging the relevant coordinates of the lifted points. But, lifting of each point in the trajectory traces require huge amount of computation overhead and consumes additional energy. Moreover, this technique assumes the existence of multiple trajectory traces for smoothing which is often unrealistic in a post-disaster scenario. In case of non-existence of multiple trajectory traces, this technique simply performs a moving average of nearby points of trajectory traces to obtain a smooth trajectory. This moving average mechanism alters the actual geometry of the underlying path by eliminating topographic details from the trajectory points of traces. Hence, in our proposed work, we modify the smoothing technique in [30] by proposing selective lifting of points in trajectory traces in order to reduce computation overhead, energy consumption and preserve the actual geometry of the underlying path.

Our proposed trajectory smoothing (pre-processing) technique first identifies the trajectory points containing noise in $Trace\_t_{creator\_id}$. The existence of noise in a trajectory point is identified based on the bearing change ($\Delta\alpha$) between consecutive trajectory points.

The GPS receivers present in mobile handheld devices provide bearing information. However, such bearing information is not employed in our work as it is not highly accurate, especially when walking is the predominant mode of trajectory trace collection (which implies traveling speed of the nodes are less than 3.0 m/s) [23]. Instead, the bearing change ($\Delta\alpha$), that is, the absolute value obtained from subtracting successive bearings using the great circle navigation formula is calculated based on Eqs. (2) and (3).

Here, bearing between two consecutive points α can be calculated as follows:

$$\alpha = \tan^{-1} 2(\sin \Delta long . \cos lat_2, \cos lat_1 . \sin lat_2 - \sin lat_1 . \cos lat_2 . \cos \Delta long) \quad (2)$$

where $(lat_1, long_1)$ and $(lat_2, long_2)$ are the latitude and longitude of two points respectively, $\Delta long = (long_1 - long_2)$ is the difference between longitudes of two points.

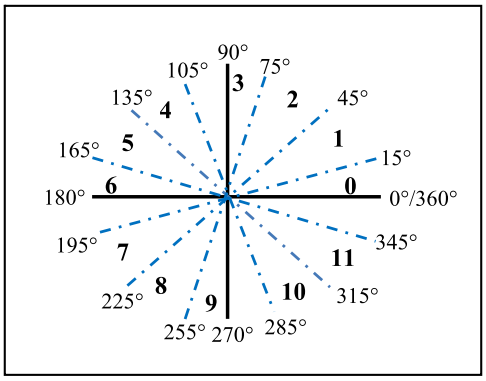


Fig. 3. 12-direction chain code [23].

The bearing change $\Delta\alpha$ is represented as:

$$\Delta\alpha_{12} = |\alpha_1 - \alpha_2| \tag{3}$$

where α_1 and α_2 are bearings. If consecutive bearing changes above a designated threshold (30° in our application domain) is present between the successive points of trajectory trace, such points are susceptible to noise.

The value for bearing change typically ranges between 0° and 360°. Inspired by the work of Kasemsuppakorn et al. [23], we have used 12-direction chain code in our proposed pre-processing technique. A 12-direction chain code is chosen because it is able to represent the sharpness of road turns clearly. Apart from that, it can also identify changes of directions accurately. Fig. 3 depicts an illustration of 12-direction chain code.

Our proposed pre-processing technique identifies a noisy point as follows.

- If more than one consecutive change of bearing beyond a designated threshold (30°) is present around 4 consecutive trajectory points in $Trace\_t_{creator\_id}$, then the third point is susceptible to noise.
- Such noisy points are only lifted to the higher dimensional space using their preceding and following points respectively.
- The noise is eliminated by performing arithmetic mean on the relevant coordinates of the lifted points.

Hence, our proposed pre-processing technique significantly reduces the amount of computation overhead and energy consumption at nodes by identifying and lifting only the noisy points to the higher dimensional space.

An illustrative example of our proposed pre-processing technique is provided in Fig. 4. It is clearly evident from Fig. 4(a) that, the bearing change from P_1 to P_2 and P_2 to P_3 ($\Delta\alpha_{12}$) is 37° which corresponds to chain code 1. Moreover, the bearing change from P_2 to P_3 and P_3 to P_4 ($\Delta\alpha_{23}$) is 65° which corresponds to chain code 2. Thus, there are two consecutive bearing changes (above threshold which is 30°) of 37° and 65° respectively around consecutive points P_1, P_2, P_3 and P_4. So, according to our proposed technique P_3 becomes a noisy point. Although there is another bearing change ($\Delta\alpha_{45}$) of 40° around point P_5, the point is not considered as noisy due to the nonexistence of consecutive points with bearing change. Fig. 4(b) depicts the trajectory trace after smoothing. The figure clearly presents the change in position of the noisy point P_3 after applying our proposed pre-processing technique.

The pre-processing sub-module is triggered automatically upon triggering the map inference module. Trajectory traces stored in the local storage of a node are inputs to this module. The outputs of this sub-module are the cleaned trajectory traces.

4.2.2. Significant point clustering

The significant point clustering sub-module performs two tasks. The primary task of this sub-module is to detect the trajectory points from each trajectory traces that contain the vital information about the geometry of the underlying pedestrian paths. A significant point in this work refers to a trajectory point in a trajectory traces with a high probability of determining the geometry of a pedestrian path.

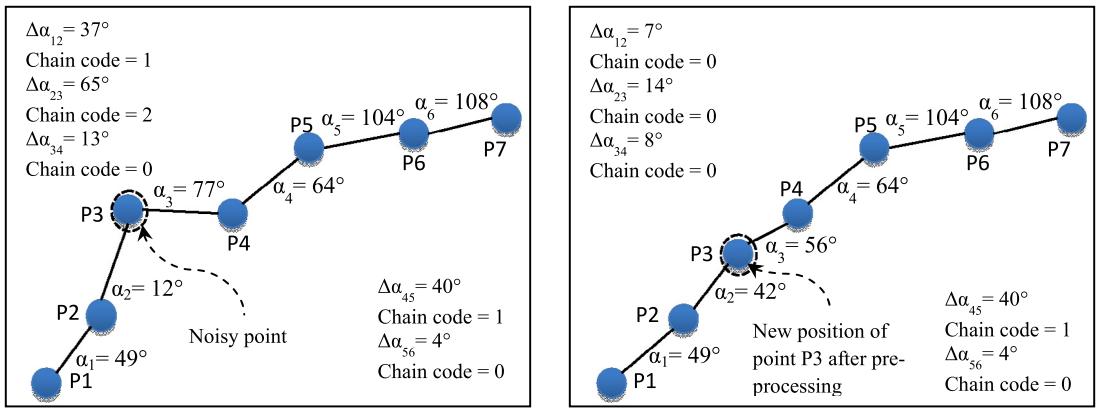
The other task of this sub-module is to generate representative significant points by grouping the significant points from multiple trajectory traces of a specific pedestrian path. In spite of pre-processing, the trajectory traces still may contain some inaccuracies. Such inaccuracies are eliminated by clustering significant points from multiple trajectory traces of identical pedestrian path and identifying a representative significant point from each cluster. Multiple trajectory traces of a particular path are obtained either from the movement of different *volunteer nodes* on the same path or from the movement of a particular *volunteer node* on the similar path at different times.

In majority of existing works on map inference, the trajectory traces are collected from moving vehicles which usually change their speeds and directions in road turns. Hence, identifying turns from the vehicular trajectory traces are comparatively easier. But, in our application domain, the trajectory traces are collected by *volunteer nodes* while walking. In such scenarios, the possibility of identifying road turns from the changes in movement speeds could be ruled out. Hence, we propose a technique to identify significant points from trajectory traces using bearing change ($\Delta\alpha$). We use 12-direction chain code proposed by [23], to set a threshold bearing change for significant point selection. As discussed in Section 4.2.1, the $\Delta\alpha$ is calculated based on Eqs. (2) and (3). The proposed technique identifies points as significant points as follows:

- If only a single bearing change above a designated threshold (30°) is present around a point in a trajectory trace, such point represents road turns and is identified as significant points.
- The starting and end points of every trajectory traces are considered as significant points.

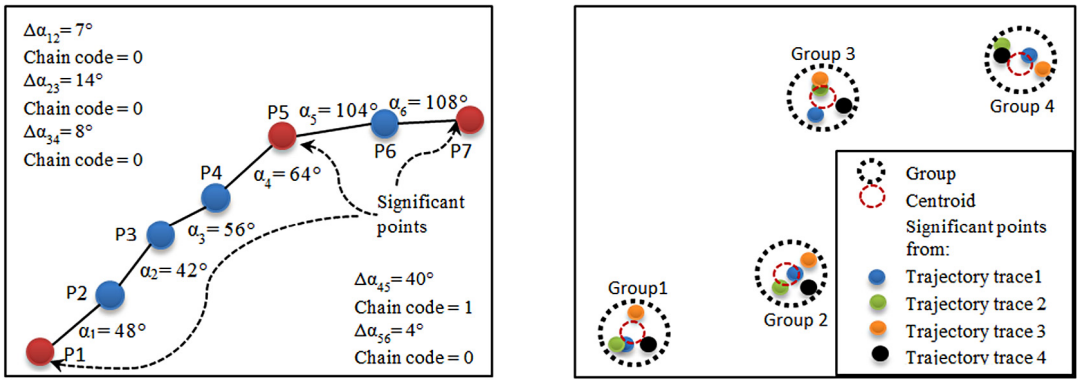
An instance of significant point identification is given in Fig. 5(a). In the figure, the bearing change from P_4 to P_5 and P_5 to P_6 is 40° which corresponds to chain code 1. Thus, there is a single bearing change ($\Delta\alpha_{45}$) around P_5 and no consecutive bearing changes above threshold (30°) are present around the adjacent points of P_5. So, P_5 represents a road turn and is considered as a significant point.

Finally, a grouping method is used to find the representative significant points from multiple trajectory traces of a pedestrian path. In this work, we do not use the well-known clustering methods due to their computation overhead. We simply calculate centroid from identical groups of significant points. The significant points of a particular trajectory trace are chosen as initial centroids of each group. The remaining significant points from other trajectory traces are added to the



(a) An instance of identifying noisy point (b) Sample trajectory trace after smoothing

Fig. 4. An illustration of the proposed pre-processing technique.



(a) Identification of significant point from a trajectory trace (b) Clustering significant points from multiple trajectory traces

Fig. 5. An illustration of significant point clustering.

corresponding groups depending on the maximum Euclidean distance of 10 m from the initial centroids. Fig. 5(b) depicts the method of clustering significant points from four trajectory traces. The centroids of each group are the representative significant points of such traces.

The significant point clustering sub-module is triggered automatically upon receiving the cleaned trajectory traces from the pre-processing sub-module. The outputs are the representative significant points from multiple trajectory traces of a pedestrian path. The representative significant points are forwarded to the next sub-module for further processing.

4.2.3. Topology inference

This sub-module connects the significant points and constructs the pedestrian map of the disaster affected areas. The significant points are connected based on the following assumptions:

- Two nearest significant points shall be connected if either of them is a starting or end points and both of them representing identical pedestrian path.
- Two nearest significant points shall be connected if there is a bearing change ($\Delta\alpha$) around both the points above a threshold (30°) and neither of them is a starting or end points of a pedestrian path.

Fig. 6(a) depicts an instance of topology inference without obstacle in the pedestrian path. The nearest points P_1 and P_2 are connected as both of them represent same path and P1 is the starting point of the path. Similarly, the nearest points P2 and P3 are connected as P_3 is the end point of the path.

Fig. 6(b) illustrates topology inference with an obstacle on the pedestrian path. Here, the nearest points P_1 and P_2 are connected as both of them represent the same path and P_1 is the starting point of the path. However, P_2 and P_5 are not connected although they are the nearest neighbors. The bearing change around P_2 is 6° ($\Delta\alpha_{15} = 6°$) which does not signify any turn. Hence, those points are not connected despite being nearest neighbors.

The topology inference sub-module is triggered automatically upon receiving the representative significant points from the significant point clustering sub-module.

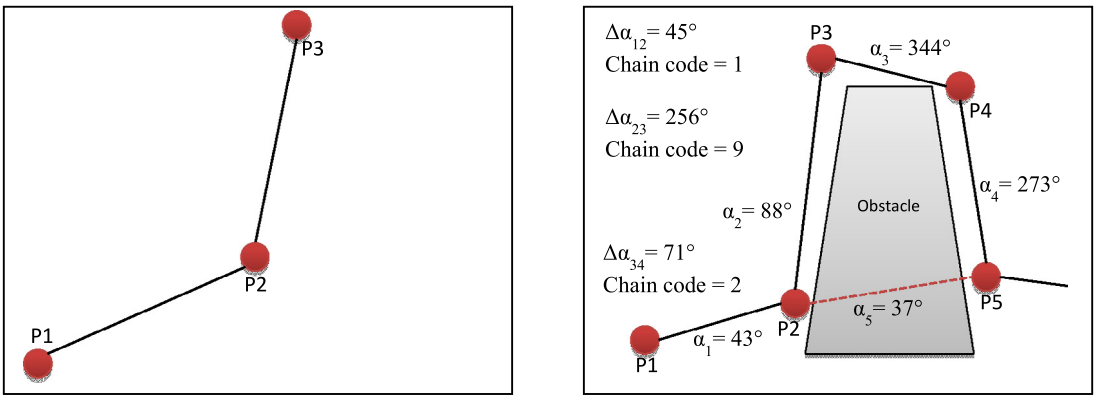
The algorithm of the *map inference* module is provided below:

5. Performance evaluation through simulation

This section evaluates the performance of the two main modules (*trace management* and *map inference*) of the proposed Post-Disaster Map Builder system through simulation. The simulation of the *trace management* and *map inference* modules are performed in ONE simulator [32] and the Mapping Toolbox of MATLAB 2017a respectively.

5.1. Simulation setup in ONE

In this work, in order to evaluate the *trace management* module, the simulation environment is set up on the modified ONE simulator [32], version 1.5.1-RC2, based on information provided by the unofficial crisis



(a) An instance of path without obstacle (b) An instance of path with obstacle

Fig. 6. An illustration of topology inference.

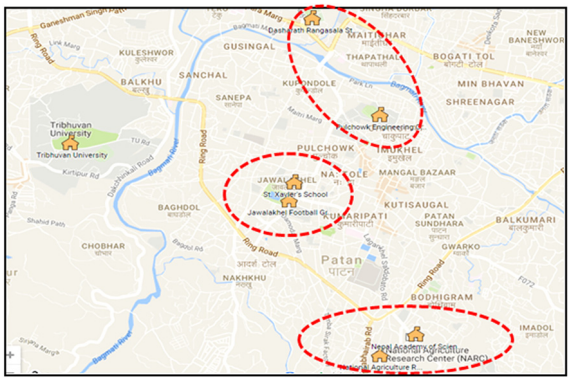


Fig. 7. Unofficial crisis map of Nepal earthquake [31].

map [31] regarding post-disaster relief operation carried out in Nepal, at the Kathmandu area on April, 2015. Fig. 7 displays the unofficial crisis map of Nepal earthquake. On the map of the disaster affected areas of Kathmandu, seven camps are set up within an area of 15 Square KM centered around the Patan area. The medical camps are set up in Durbar Square, St. Xavier's School, Pulchowk Engineering College Campus and Nepal Academy of Science. Relief camps are set up at National Agriculture Research Centre, Jawalakhel Football ground and the command and control centre is established at Tribhuban University. The dotted red ovals indicate the disaster affected areas in the map. We consider post-disaster mobility model (PDM) [27], as the underlying mobility model for our simulation. PDM proposed by Uddin et al. illustrates the movement pattern of victims and volunteers in post-disaster scenarios. PDM exposes that the mobility pattern of victims and volunteers are neither completely random nor completely periodic. We have chosen *Three-tier Architecture* [7] for our simulation as it offers superior message delivery with reduced network and energy overheads in post-disaster communication irrespective of routing protocols.

In the experiment, we assume that the trajectory traces are collected by the *volunteer nodes*. The collected traces are disseminated to the other nodes (*volunteer nodes*, *camp nodes* and *ferry nodes*) of the network as per system model. The disaster relief operations are generally performed for 12 h in a day from 6 a.m. to 6 p.m. Hence, simulation time is 12 h.

We assume the existence of similar types of devices (Samsung Galaxy J7 Prime) with extended secondary storage as camp nodes. Table 1 provides the Wi-Fi power consumption readings of Samsung Galaxy J7 Prime smartphone with Android v7.0 (Nougat) operating system having 3300 mAh, 3.85 V Li-ion battery. The readings of energy consumption are obtained using Trepn profiler [33] version 6.2 which provides the energy consumption of Android apps running on Android devices.

Table 1
Power consumption readings of Samsung Galaxy Grand 2.

Wi-Fi state	Power consumed (mW)	Energy consumption rate (J/s)
Wi-Fi off	0	0
Wi-Fi idle	164.79	0.165
Wi-Fi scanning	976.71	0.976
Wi-Fi receiving	1717.71	1.717
Wi-Fi sending	1786.24	1.786

Table 2
Parameter values used in simulation.

Parameters	Value
Simulation time	12 h
Number of nodes	82 (Camps: 7, Volunteers: 70, Ferry: 5)
Transmission range	80 m
Buffer Size	20 MB
Message TTL	720 min
Initial energy of each node	45 738 J
Trajectory trace generation	At the beginning of simulation
Mode of communication	Group based multicasting (*groupcasting*)

The total amount of stored energy in a battery is calculated as mentioned below,

$$E = QV/1000$$

where, E is energy in watt-hour (Wh), Q is electric charge in milliampere-hour (mAh) and V is voltage in Volt (V).

We also know that 1 Wh = 3600 J.

Hence, Total Stored Energy = 45 738 J. We consider the total stored energy of a smartphone battery as the initial energy of each node during simulation. We have applied the power consumption readings in various states of Wi-Fi mentioned in Table 1, in the *EnergyModel* module of ONE simulator in order to evaluate the average energy consumption of the nodes. Other important simulation parameters are provided in Table 2.

5.2. Simulation setup in MATLAB

The evaluation of the *map inference* module is performed on the Mapping Toolbox of MATLAB 2017a. This toolbox consists of inbuilt algorithms and functions for analyzing GIS data and construct maps from such data. Primarily, the trajectory points collected by volunteer are imported to the Mapping Toolbox using *gpxread()* function. Subsequently, our proposed *map inference* algorithm is implemented in MATLAB 2017a. Finally, the route map is constructed and placed on top of the Open Street Map of the disaster affected areas from the significant points using *geoshow()* and *webmap()* functions of the Mapping Toolbox.

Algorithm : Map inference
Input: Trajectory traces ($Trace\_t_{creator\ id}$) stored in a node
Output: Pedestrian map
Algorithm Triggered: Upon request from user

```
Begin
for each Trace_t_creator id in {Trace_list_node id }
   call preprocessing(Trace_t_creator id)
   call significant_point_clustering(Trace_t_creator id)
end for
  call topology_inference(sgp_table)
End
// Function
preprocessing(Trace_t_creator id)
for each p in Trace_t_creator id
  compute Δα                          */ Compute bearing change between consecutive points based on
                                      equation (3) */
  if (p containing noise)
     lift p to p'                     */ Lifting p to a higher dimensional space */
     p = arithmatic_mean(p')          */ perform arithmetic mean on p' and assign it to p */
     Trace_t_creator_id ← p           */store p in the trajectory trace */
 end if
end for
return Trace_t_creator_id

// Function
significant_point_clustering(Trace_t_creator id)
for each p in Trace_t_creator id
   compute Δα                         */ Compute bearing change between consecutive points based on
                                      equation (3) */
   compute C_c                        */ Compute the chain code */
   if (C_c > 0 || 6)                  */ identify significant point as per 12-direction chain code */
    sgp_table ← p                     */ store p in significant points table (sgp_table) */
end for
compute centroid from identical groups of significant points
return centroids[]

// Function
topology_inference(centroids[])
for each p in centroids[]             */ nearest neighbor of p from similar pedestrian path */
find p_near
  if (p or p_near is a starting point or end point)
     connect p and p_near
  end if
  if (p or p_near is not a starting point or end point)
     compute Δα
     compute C_c
     if (C_c >0 || 6 around both p and p_near)
       connect p and p_near
     end if
  end if
end for
return
```

5.3. Simulation metrics

Simulation performance of the proposed *trace management* module is evaluated depending on two categories of metrics such as metrics for design parameters and network parameters. The metrics influencing the performance of major design consideration are categorized as design parameters. The metrics influencing the performance of network are considered as network parameters. We consider *trace share ratio*, *overhead ratio* and *average energy consumption* as design parameters and *average delivery latency* as network parameter. The requirements of the selected parameters along with definition of each of the parameters are elaborated as follows:

Trace share ratio: We introduce this metric to measure the extent of trajectory traces spreading in the network within fixed time duration. This is an indicator of message delivery efficiency of the proposed module. The proposed *trace management* module employs Trace Route protocol to disseminate trajectory traces in the entire network within a fixed duration. Therefore, it is crucial to measure the delivery efficiency of the proposed Trace Route protocol. Hence, trace share ratio is chosen as an evaluation measure due its ability to measure the amount of trajectory traces disseminated in the entire network within a fixed duration. The metric is defined as follows:

$$Trace\ share\ ratio\ (\%) = \frac{number\ of\ nodes\ received\ trajectory\ traces\ within\ a\ fixed\ time\ duration}{total\ number\ of\ nodes\ in\ the\ network} \times 100\%$$

Overhead ratio: It is an assessment of the cost of delivering a trajectory trace to its destination in terms of replication. It reflects the number of trajectory traces that are relayed to deliver a single trajectory trace and

is defined as [7]:

$$Overhead\ ratio = \frac{number\ of\ trajectory\ trace\ relayed}{total\ number\ of\ trajectory\ trace\ delivered}$$

The *Trace Share Ratio* indicates the amount of trajectory traces disseminated in the entire network within a fixed duration. The proposed Trace Route protocol is a replication based protocol. Hence, the cost of transmission in terms replication need to be measured to evaluate the efficiency of the proposed protocol. Hence, overhead ratio is chosen as an evaluation measure due its capability to represent the cost of transmission. In this work we consider only multicopy routing. In case of multicopy routing overhead ratio should be always >1.

Average energy consumption: It is defined as the average energy consumed by each node in the network within fixed time duration. In our application domain, the trajectory traces are collected and exchanged among smartphones which are resource-constrained (for instance limited battery life). Immediately after natural disasters (such as flood, storm, earthquake etc.), the power supply is shut down in the affected areas to reduce the risk of electrocution. Hence, re-charging of smartphones in such scenarios remains challenging. Moreover, in our application domain the network lifetime is dependent on the number of active smartphones. Thus, it is essential to measure the energy consumption of the smartphones in order to analyze the lifetime of such devices. Hence, average energy consumption is chosen as an evaluation measure in this work.

Average delivery latency: It is defined as the average delay from the creation of the trajectory trace to its delivery. This metric indicates the speed of dissemination of trajectory traces in the network. The dissemination of trajectory traces in network with reduced latency is one of the primary goals of sharing trajectory traces in a post-disaster scenario. Hence, it is chosen as an evaluation measure in this work.

Apart from this, the performance of the proposed *map inference* module is assessed considering *Root mean square error*. The RMS error is treated as a qualitative measure [34] to evaluate the accuracy of the maps produced by employing *map inference* module. But, in our application domain the qualitative evaluation of the produced maps is not very pertinent due to the unavailability of the manually plotted route maps of the disaster affected areas as reference data. The road networks in disaster affected areas change drastically because of disaster (such as waterlogging due to flood, structural collapse or incidental destructions like a landslide). Hence, existing analog (paper based) route maps of such areas become obsolete. Therefore, the use of existing analog route maps as reference data might not be very useful in order to evaluate the quality of the produced route maps. However, for the completeness of the performance evaluation we use the offline Open Street Maps of the disaster affected areas as reference which is open source and convenient to import in mobile applications. We calculate the RMS error indicating the average distance between the constructed path segments and the actual path segments.

Root mean square error: Here, Root mean square (RMS) error indicates the average distance between the constructed path segments and the actual path segments. The smaller RMSE value signifies better geometric accuracy of the pedestrian path. The metric is defined [23] as follows:

$$RMS\ error = \sqrt{\frac{\sum_{i=1}^{l} d(cnst_i, act_i)^2}{l}}$$

Here, l = number of path segments, $d(cnst_i, act_i)$ = distance between constructed path segment and the actual path segment.

5.4. Results and discussion

In this section we analyze the experimental results acquired from different simulation scenarios.

5.4.1. Evaluating the performance of trace management module

Fig. 8 presents the comparison of Trace Route protocol in periodic sharing of trajectory traces with other well known DTN unicast routing protocols (Epidemic, PRoPHET and MaxProp) as well as a couple of multicast routing protocols such as SDM and EASDM with respect to the design parameters as well as a network parameter.

Fig. 8(a) exhibits the efficiency of different routing protocols in periodic sharing of trajectory traces. The results clearly indicate that each routing protocols exhibit a growth for an initial period and gradually reach to a steady state. Precisely, after the simulation period of 2 h, the values of the trace share ratio are 94.28%, 93.71%, 94.57%, 95.2%, 75.64% and 73.52% respectively in Epidemic, PRoPHET, MaxProp, Trace Route, SDM and EASDM routing protocols.

The aforesaid discussion reveals that the Trace Route protocol exhibits marginal improvements in trace share ratio compared to well known unicast routing protocols. However, it exhibits substantial improvements (around 25.72% and 29.48%) compared to multicast routing protocols (SDM and EASDM respectively). It can disseminate trajectory traces to the 95.2% (≈95%) of the nodes in the entire network within 2 h after generation of trajectory traces. Therefore, it is apparent that Trace Route routing protocol demonstrates superior performance in sharing trajectory traces.

Fig. 8(b) demonstrates the evaluation of the overhead ratio. The results exhibit insignificant growth of overhead ratio for all the well known routing protocols except Epidemic routing. After the simulation period of 2 h, the values of the overhead ratio are 76.61, 32.04, 23.07, 0.6, 0.57 and 0.58 in Epidemic, PRoPHET, MaxProp, Trace Route, SDM and EASDM protocols respectively. The figure exposes that in terms of message replication SDM, EASDM and Trace Route exhibits marginal amount of network overhead. Precisely, compared to Epidemic routing protocol each of the SDM, EASDM and Trace Route incur around 99.25%, 99.24% and 99.21% less network overhead respectively. The reason behind this significant variation is the application of multicasting in each of the three aforesaid protocols. All the other routing protocols such as Epidemic, PRoPHET and MaxProp support unicast communication only. Therefore, disseminating a single message to multicast destinations through such protocols require replication of the original message and delivering the replicas through unicast communication. Consequently, the overhead ratio grows abruptly.

Fig. 8(c) reveals the evaluation of average energy consumed by each node in the network under different routing protocols with respect to time. The figure clearly indicates that the average energy consumption grows linearly with time for all the routing protocols. After the simulation period of 2 h, the values of average energy consumption are 8505.92 J, 8263.67 J, 8213.95 J, 8094.36 J, 8088.69 J and 7793.63 J in Epidemic, PRoPHET, MaxProp, Trace Route, SDM and EASDM protocols respectively. The average energy consumption values clearly disclose negligible supremacy of EASDM, SDM and Trace Route compared to other routing protocol.

Fig. 8(d) demonstrates the comparison of Trace Route with other well known DTN routing protocols with respect to the average delivery latency (network parameter). The figure clearly reveals that after the simulation period of 2 h, the values of average delivery latency are 1037.7 s, 1511.73 s, 1721.64 s, 995.24 s, 1029.71 s and 1025.93 s in Epidemic, PRoPHET, MaxProp, Trace Route, SDM and EASDM protocols respectively. The figure clearly indicates that the Trace Route routing protocols out performs other protocols in terms of average delivery latency.

From the above discussion, it is evident that Trace Route protocol demonstrates its dominance in terms of trace share ratio and least average delivery latency which are the primary goals of sharing trajectory traces in a post-disaster scenario. However, these goals are attained at the cost of slightly elevated average energy consumption and overhead ratio compared to multicast routing protocols (EASDM and SDM). Although EASDM and SDM protocols demonstrate their dominance in average energy consumption and overhead ratio respectively, they do

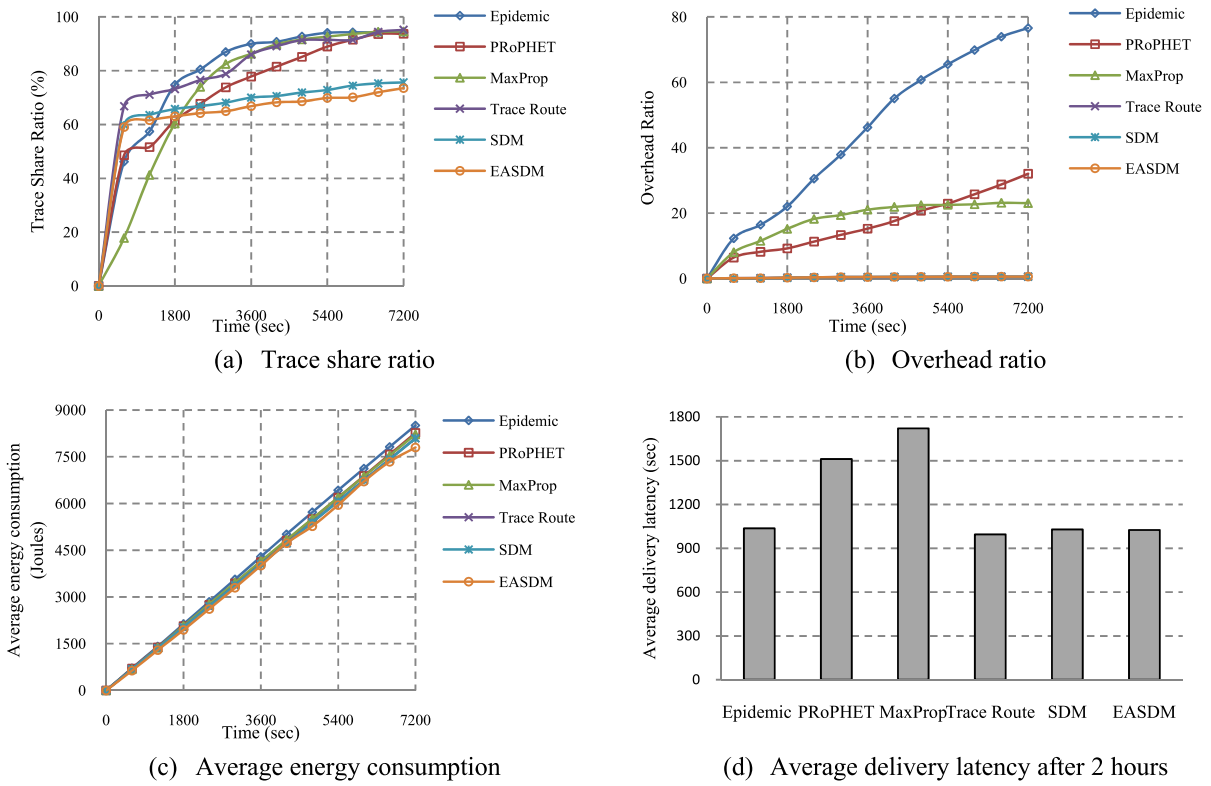


Fig. 8. Comparative performance of trace route protocol.

not yield desired outcomes in terms of trace share ratio and average delivery latency which are the crucial objectives of sharing trajectory traces in our application domain. Hence, Trace Route protocol proves suitable for the group based periodic sharing of the trajectory traces in post-disaster scenario.

5.4.2. *Feasibility and efficiency of map inference module*

In this section, we evaluate the feasibility and efficiency of our proposed *map inference module* in generating digital pedestrian maps from trajectory traces. Fig. 9 depicts the construction of a pedestrian path from multiple trajectory traces (4 trajectory traces containing 230 points) in the *map inference* module. Fig. 9(a) presents the raw trajectory traces and Fig. 9(b) depicts the pedestrian path constructed from such traces. Fig. 9(b) clearly reveals the existence of noise in the trajectory traces.

Fig. 9(c) presents the traces after *pre-processing*. The *pre-processed* traces contain the identical number of points as raw trajectory traces. Fig. 9(d) depicts the pedestrian path inferred from *pre-processed* traces. From the figure, it is evident that the path still contains some noisy points.

Fig. 9(e) presents the representative significant points after *significant point clustering*. The figure contains 29 representative significant points which are merely around 13% of the originally collected traces. Hence, there is a substantial reduction in the number of trace points after *significant point clustering*. The reduction in the number of trace points reduces the amount of computation overhead during *topology inference* (path inference). The reduced amount of computation demands lesser energy to process those points. Hence, energy consumption is minimized. Fig. 9(f) indicates the pedestrian path obtained after *topology inference* from representative significant points. The figure clearly shows that the path does not contain any redundant points. Hence, our proposed *map inference* module can construct pedestrian maps of the disaster affected areas from trajectory traces.

We evaluate the efficiency of our proposed *map inference* module by comparing it with a contemporary map inference scheme proposed by Karagiorgou et al. [22]. The comparison is performed in terms of RMS error which indicates the average distance between the constructed path and the actual path. The RMS error is treated as a qualitative measure [34] to evaluate the accuracy of the maps. Fig. 10 depicts the positional accuracy of the pedestrian path constructed through both the aforesaid map construction schemes with respect to the number of repeated trajectory traces (collected by traversing a predefined path multiple times). From the figure, it is evident that the number of repeated trajectory traces and geometric accuracy of the constructed path are interrelated. The accuracy increases gradually with the increasing number of repeated trajectory traces. The figure also reveals that our proposed map inference module exhibits almost identical accuracy in constructing pedestrian paths compared to the scheme proposed by Karagiorgou et al. The scheme proposed by Karagiorgou et al. assumes the existence of high power computing resources for map inference which could be ruled out in a post-disaster scenario.

Hence, even in absence of high power computing device, our proposed system achieves almost identical accuracy as in conventional scenario. Moreover, it is also evident that the significant improvements in accuracy are achieved with first five repeated trajectory traces. If we increase the number of repeated trajectory traces beyond five, the geometric accuracy of the pedestrian path segments does not improve significantly. Therefore, in order to construct high accuracy pedestrian map from trajectory traces using the proposed *map inference* module, at least five trajectory traces for each path segments are necessary.

The aforesaid discussions reveal the feasibility of our proposed Post-Disaster Map Builder system in constructing pedestrian maps in post-disaster scenarios in absence of high power computing resources. This motivates us to implement Post-Disaster Map Builder system as an Android application (app).

6. Implementation

We have developed the Post-Disaster Map Builder Android application (app) by implementing the proposed techniques as discussed

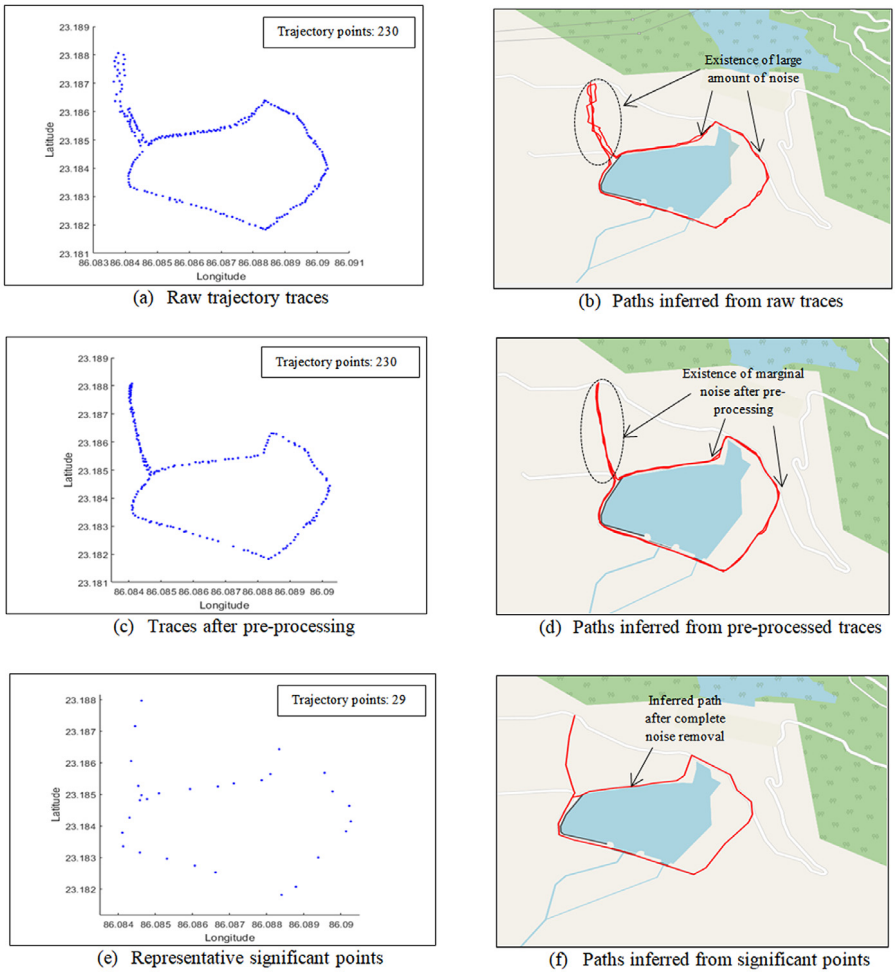


Fig. 9. An instance of the pedestrian path constructed by the map inference module.

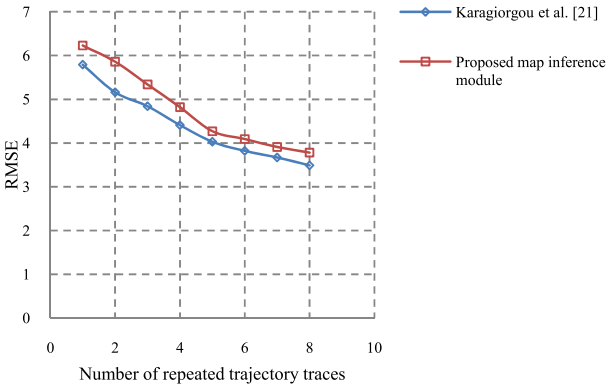


Fig. 10. Comparative evaluation of our proposed map inference module.

in Section 4. Every node (volunteer, camp and ferry) in the disaster affected areas run this application for periodic collection and sharing of trajectory traces automatically. In addition to that, the nodes are able to generate and view the pedestrian map based on the trajectory traces collected by them. Fig. 11 depicts some snapshots of the Post-Disaster Map Builder app. Immediately after launching the app in the Android devices the home page appears as shown in Fig. 11(a). The home page contains three buttons SHOW MAP, START SERVICE and STOP SERVICE. The START SERVICE button starts the periodic trace collection and sharing function. After initiation, the periodic trace collection and sharing function run in the background until it is stopped by the STOP SERVICE button.

Fig. 11(b) indicates the log of collected traces by the app over time after pressing the START SERVICE button. The collected traces are stored in .gpx files. Once a new node comes in the transmission range of the source node, the trajectory traces are forwarded to that node as per the system model. The SHOW MAP button on home page initiates the map inference functionality of the application. Fig. 11(c) illustrates the pedestrian map (marked by the dotted red lines) constructed by the app from the collected traces. The Post-Disaster Map Builder app overlays the generated pedestrian map onto pre-cached map image. Offline Open Street Map (OSM) is used as pre-cached map image in this app. The offline OSM is incorporated with the app by using *osmdroid* Android class.

The STOP SERVICE button on home page stops the app in background. Fig. 11(d) exhibits the app log after pressing the STOP SERVICE button.

6.1. Testbed setup

We have conducted several field trials in order to evaluate the efficiency of Post-Disaster Map Builder app under different scenarios. The field trials are conducted at Purulia and Bankura districts, West Bengal with eleven volunteers equipped with Post-Disaster Map Builder app preinstalled in their handheld devices (smartphones or Tablets). The volunteers are allocated to two groups (five volunteers in each group) corresponds to two disaster affected areas and placed in such locations that no groups can directly communicate with each other.

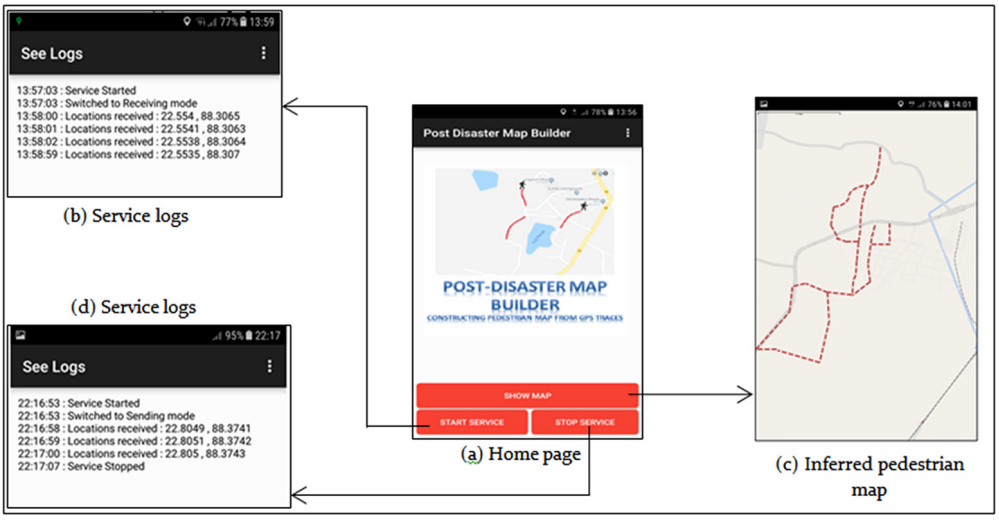


Fig. 11. Snapshots of the Post-Disaster Map Builder app.

Each group signifies a disaster affected area. One volunteer in each group equipped with tablet remains static and acts as *camp nodes*. One volunteer carrying a smartphone on his bicycle visit the static node in each group periodically to collect trajectory traces. Such moving volunteers act as *ferry nodes*. The trajectory traces are stored in the secondary storage of the *ferry node*. Upon reaching a new group the *ferry node* exchanges trajectory traces with *camp node*. Fig. 12 indicates the location of two volunteer groups during the field trial conducted at Purulia, West Bengal, India.

We have used heterogeneous devices (smartphones and Tablets from different manufacturers) for the field trials in order to assess the interoperability of the app. The performance of the Post-Disaster Map Builder app is assessed based on two aspects such as the extent of dissemination of trajectory traces in the network over time and energy efficiency of the app.

The Trepn profiler [33], an Android app is employed to measure the energy consumption of Post-Disaster Map Builder app. The energy consumption readings from Power Tutor have been recorded during the exchange of trajectory traces through Post-Disaster Map Builder. The average energy consumption is measured by calculating the arithmetic mean of the energy consumption readings from all the devices involved in the field trial.

6.2. Evaluating the performance of Post-Disaster Map Builder app

In this section, we evaluate the performance of the Post-Disaster Map Builder app based on the readings captured during the field trial conducted at Purulia district, West Bengal, India. Fig. 13(a) depicts the characteristic graph of trace share ratio over time. It is clearly apparent from the figure that, the trace share ratio increases progressively over time and gradually attains a steady state. The results also reveal that around 97.5% trajectory traces are delivered to their destinations within 2 h after generation of such files. Therefore, it is evident that Post-Disaster Map Builder app exhibits competence in sharing trajectory traces.

Fig. 13(b) exhibits the average energy consumed by each node in the network for continuously running the app. The figure reveals that the average energy consumed by employing Post-Disaster Map Builder app after 2 h is approximately 2601 J, which is around 5.7% of the total battery capacity. Hence, we can mathematically estimate that the nodes remain functional approximately for 35 h provided the mobile device is utilized only for running the Post-Disaster Map Builder app. So we can safely predict that the nodes employing our proposed app remain functional at least for a day after fully recharged. This is a vital requirement for the post-disaster scenario.

7. Conclusion

In this paper, we propose Post-Disaster Map Builder, a crowdsenced digital pedestrian map construction system over the smartphone based DTN. The proposed system involves two modules that is *trace management* and *map inference*. The *trace management* module collects trajectory traces through GPS receivers of the nodes. It employs the publish–subscribe mechanism to periodically exchange trajectory traces among the nodes belonging to a specified disaster affected area. Such trajectory traces serve as inputs to the *map inference* module which facilitates the construction of the pedestrian map of the disaster affected areas in the mobile handheld devices of the volunteers in a progressive manner. Due to the scarcity of computing resources and unstable power supply in post-disaster scenarios, attempts have also been made to make the map inference mechanism computationally less intensive and energy efficient. Evaluation through simulation and testbed implementation reveals that the proposed system can deliver more than 95% trajectory traces to their destinations within 2 h after generation. It is also apparent that the nodes employing our proposed system remain functional at least for 24 h after fully recharged, which is very much acceptable in a post-disaster scenario. Comparison of the proposed Post-Disaster Map Builder system with one of the contemporary map inference schemes proposed by Karagiorgou et al. [22] reveals that even in absence of high power computing devices, Post-Disaster Map Builder system still exhibits almost identical accuracy in constructing pedestrian maps.

Finally, we believe that our proposed Post-Disaster Map Builder system still has rooms for improvement. The system at present is only implemented in Android platform. The cross platform implementation of the proposed system may be studied as a part of our future work. The system can be improved further by introducing offline navigation assistance which will instruct volunteers to reach their destinations through the shortest paths.

Acknowledgments

This work is supported by Information Technology Research Academy (ITRA), Digital India Corporation, Government of India under, ITRA-Mobile grant [ITRA/15(58)/Mobile/DISARM/01/Rev/2015]. We also acknowledge all support and cooperation rendered by two NGOs Doctors For You (http://doctorsforyou.org/) and Society for Promotion of Appropriate Development Efforts (http://www.spadeglobal.org/) that has been instrumental in this work.

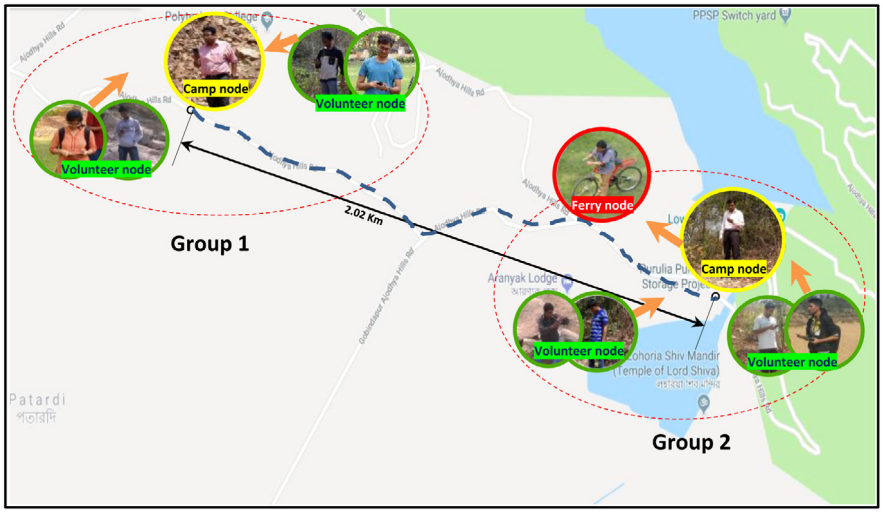
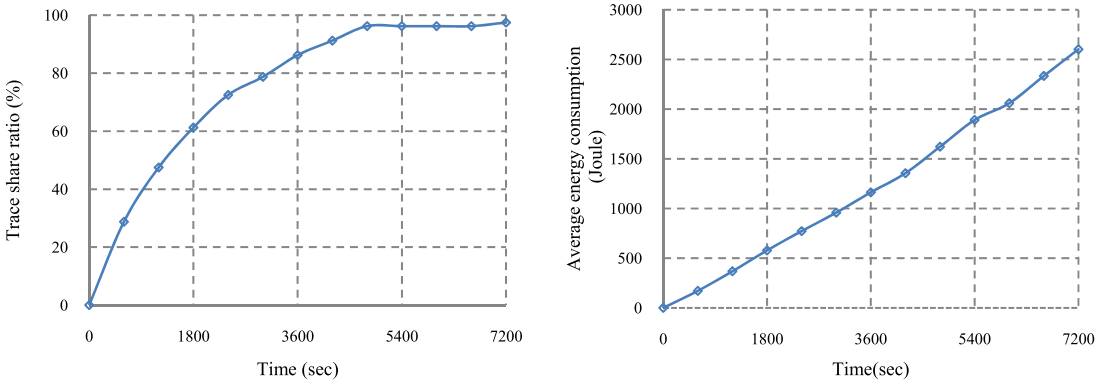


Fig. 12. Field trial conducted at Purulia, West Bengal, India.



(a) Trace share ratio (b) Average energy consumption

Fig. 13. Performance of Post-Disaster Map Builder application.

References

[1] S. Basu, S. Roy, S. Bandyopadhyay, S.D. Bit, A utility driven post disaster emergency resource allocation system using DTN, IEEE Trans. Syst. Man Cybern. Syst. (2018) http://dx.doi.org/10.1109/TSMC.2018.2813008.

[2] A. Kwasinski, Lessons from field damage assessments about communication networks power supply and infrastructure performance during natural disasters with a focus on Hurricane Sandy, in: Proceedings of the FCC Workshop on Network Resiliency, 2013, pp. 1–36.

[3] K. Rozario, Making progress disaster narratives and the art of optimism in modern America, in: L.J. Vale, T.J. Campanella (Eds.), The Resilient City: How Modern Cities Recover from Disaster, Oxford University Press, New York, 2005, pp. 27–54.

[4] C. Vandeviver, Applying google maps and google street view in criminological research, Crime Sci. 3 (1) (2014) 13.

[5] J.M. Ercolano, Pedestrian Disaster Preparedness and Emergency Management: White Paper for Executive Management, Technical Report, New York State Department of Transportation, 2007.

[6] K. Fall, G. Iannaccone, J. Kannan, F. Silveira, N. Taft, A disruption-tolerant architecture for secure and efficient disaster response communications, in: Proceedings of the ISCRAM, 2010.

[7] S. Bhattacharjee, S. Roy, S. Bandyopadhyay, Exploring an energy-efficient DTN framework supporting disaster management services in post disaster relief operation, Wirel. Netw. 21 (3) (2015) 1033–1046.

[8] R.K. Ganti, F. Ye, H. Lei, Mobile crowdsensing: current state and future challenges, IEEE Commun. Mag. 49 (11) (2011) 32–39.

[9] W. Vogels, R.V. Renesse, K. Birman, The power of epidemics: Robust communication for large-scale distributed systems, in: Proceedings of th e ACM SIGCOMM, 2003, pp. 131–135.

[10] A. Lindgren, A. Doria, O. Schelén, Probabilistic routing in intermittently connected networks, ACM SIGMOBILE Mob. Comput. Commun. Rev. 7 (3) (2003) 19–20.

[11] S. Grasic, E. Davies, A. Lindgren, A. Doria, The evolution of a DTN routing protocol – PRoPHETv2, in: Proceedings of the ACM Workshop on Challenged Networks, 2011, pp. 27–30.

[12] J. Burgess, B. Gallagher, D. Jensen, B.N. Levine, MaxProp: Routing for vehicle-based disruption-tolerant networks, in: Proceedings of the IEEE INFOCOM, 2006, pp. 1–11.

[13] P. Hui, J. Crowcroft, E. Yoneki, BUBBLE Rap: Social-based forwarding in delay-tolerant networks, IEEE Trans. Mob. Comput. 10 (11) (2011) 1576–1589.

[14] W. Gao, Q. Li, B. Zhao, G. Cao, Social-aware multicast in disruption-tolerant networks, IEEE/ACM Trans. Netw. 20 (5) (2012) 1553–1566.

[15] L. Galluccio, B. Lorenzo, S. Glisic, Sociality-aided new adaptive infection recovery schemes for multicast DTNs, IEEE Trans. Veh. Technol. 65 (5) (2016) 3360–3376.

[16] A. Roy, S. Bose, T. Acharya, S. DasBit, Social-based energy-aware multicasting in delay tolerant networks, J. Netw. Comput. Appl. 87 (2017) 169–184.

[17] S. Edelkamp, S. Schrödl, Route planning and map inference with global positioning traces, in: R. Klein, H.W. Six, L. Wegner (Eds.), Computer Science in Perspective, in: Lecture Notes in Computer Science, vol. 2598, Springer, Berlin, 2003, pp. 128–151.

[18] S. Schrödl, K. Wagstaff, S. Rogers, et al., Mining GPS traces for map refinement, Data Min. Knowl. Discov. 9 (1) (2004) 59–87.

[19] L. Cao, J. Krumm, From GPS traces to a routable road map, in: Proceedings of the 17th International Conference on Advances in Geographic Information Systems, 2009, pp. 3–12.

[20] J. Wang, X. Rui, X. Song, X. Tan, C. Wang, V. Raghavan, A novel approach for generating routable road maps from vehicle GPS traces, Int. J. Geogr. Inf. Sci. 29 (1) (2015) 69–91.

[21] A. Fathi, J. Krumm, Detecting road intersections from GPS traces, in: Proceedings of the 6th International Conference on Geographic Information Science, 2010, pp. 56–69.

[22] S. Karagiorgou, D. Pfoser, On vehicle tracking data-based road network generation, in: Proceedings of the 20th International Conference on Advances in Geographic Information Systems, 2012, pp. 89–98.

[23] P. Kasemsuppakorn, H.A. Karimi, A pedestrian network construction algorithm based on multiple GPS traces, Transp. Res. C Emerg. Technol. 26 (2013) 285–300.

[24] U. Blanke, R. Guldener, S. Feese, G. Tröster, Crowdsourced pedestrian map construction for short-term city-scale events, in: Proceedings of the International Conference on IoT in Urban Space, 2014, pp. 25–31.

[25] J. Qiu, R. Wang, Inferring road maps from sparsely sampled GPS traces, J. Locat. Based Serv. 10 (2) (2016) 111–124.
[26] E.M. Trono, M. Fujimoto, H. Suwa, Y. Arakawa, K. Yasumoto, Generating pedestrian maps of disaster areas through ad-hoc deployment of computing resources across a DTN, Comput. Commun. 100 (2017) 129–142.
[27] M.Y.S. Uddin, D.M. Nicol, T.F. Abdelzaher, R.H. Kravets, A post-disaster mobility model for delay tolerant networking, in: Proceedings of the Winter Simulation Conference, 2009, pp. 2785–2796.
[28] Y. Cao, Z. Sun, Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges, IEEE Commun. Surv. Tutor. 15 (2) (2013) 654–677.
[29] P.A. Zandbergen, S.J. Barbeau, Positional accuracy of assisted GPS data from high-sensitivity GPS-enabled mobile phones, J. Navigation 64 (3) (2011) 381–399.
[30] F. Chazal, D. Chen, L. Guibas, X. Jiang, C. Sommer, Data-driven trajectory smoothing, in: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2011, pp. 251–260.
[31] Google unofficial crisis map. https://www.google.com/maps/d/viewer?mid=zjU92k9XcdHk.kGerGuVhu2R0. Last Accessed: 2017-08-14.
[32] S. Bhattacharjee, R. Chatterjee, T. Pal, S. DasBit, Implementing multicasting and broadcasting of multimedia data in ONE simulator, in: Proceedings of the SIMUTOOLS, 2017.
[33] L. Ben-Zur, Developer Tool Spotlight - Using Trepn Profiler for Power-Efficient Apps. https://developer:qualcomm:com/blog/developer-tool-spotlight-using-trepn-profiler-powerefficient-apps. Last Accessed: 2017-08-17, 2011.
[34] C. Wiedemann, External evaluation of road networks, Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 34 (3/W8) (2003) 93–98.