NAIST-IS-DD1061206

Doctoral Dissertation

Information Collection in Disaster Areas Using Participatory Sensing and Delay Tolerant Network

Jovilyn Therese B. Fajardo

November 28, 2013

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Jovilyn Therese B. Fajardo

Thesis Committee:

Professor Minoru Ito	(Supervisor)
Professor Keiichi Yasumoto	(Co-Supervisor)
Professor Suguru Yamaguchi	(Co-Supervisor)
Associate Professor Naoki Shibata	(Co-Supervisor)

Information Collection in Disaster Areas Using Participatory Sensing and Delay Tolerant Network*

Jovilyn Therese B. Fajardo

Abstract

When a large-scale disaster strikes, situational awareness over the disaster area is important because first responders need timely and accurate information to assess the situation in the affected area and to provide an effective and immediate assistance. During this time, the communications infrastructure is usually unavailable and a mobile ad-hoc network is utilized to gather critical information using the mobile phones of people in the disaster zone as sensing nodes. In order to achieve situational awareness, text-based or photograph-based information is collected from the disaster area. However, instant communication of information is unrealistic thus different approaches to disaster information collection are proposed to collect as much information from the disaster area in the shortest possible time.

In the first part of this thesis, a disruption tolerant network (DTN)-based data aggregation method is proposed to achieve maximum coverage of an area of interest (AoI) within the disaster zone while minimizing delay. In this method, mobile phone users create messages containing disaster-related numerical information (e.g., 8 injured persons) and merge them with their respective coverage areas resulting in a new message with the merged coverage. The merging or aggregation of multiple messages will reduce message size and minimize the overall message collection delay. However, simply merging the messages can result in

*Doctoral Dissertation, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1061206, November 28, 2013.

duplicate counting; to prevent this, a Bloom filter is constructed for each message. The Bloom filter prevents messages containing the same information on a given location from being aggregated. In addition, to reduce further the message delivery time, the expected reaching time of a node to its destination (ert) is introduced as a routing metric. Through computer simulation with a real geographical map, we confirmed that the proposed method achieved a smaller delay with a smaller number of total exchanged messages in collecting disaster information covering the AoI than a conventional method based on epidemic routing.

In the second part of this thesis, a content-based image prioritization technique in delay tolerant network is proposed to achieve fast collection of images with high priorities. Geotagged disaster images are collected by mobile users and assigned priorities based on their content through image processing using OpenCV. A preinstalled mobile application on the users' phones detects critical content (e.g., fires, road blocks) in the images. The detected images with critical content are sent over the network faster than images with noncritical content in order to address the critical events immediately. Through testing and computer simulation, we confirmed that the proposed method with image prioritization achieved a higher percentage of received images with critical content corresponding to a smaller delivery latency than without image prioritization.

Keywords:

data aggregation, data prioritization, delay tolerant network, disaster communication system, information collection, participatory sensing

Contents

List of Figures

List of Tables

1. Introduction

Based on data from EM-DAT: the OFDA/CRED International Disaster Database, disasters that include earthquakes, droughts, floods, and storms caused 1.7 trillion dollars (USD) in damages with 2.9 billion people affected and 1.2 million people killed from the year 2000 to 2012 as shown in Fig. 1. Indeed, these disasters have a large economic and human impact including the unavailability or destruction of the communications infrastructure. As a result, one of the challenges in a disaster situation is establishing effective communication and at such times, situational awareness over the disaster area is at its lowest point. However, situational information is important because first responders rely on this information to assess the situation in the affected area and to provide an effective and immediate response. It is then critical to gather a comprehensive and reliable data from the disaster area and deliver it to the appropriate recipients for action. In this kind of situation, a mobile ad-hoc network can be deployed to provide a means for communication.

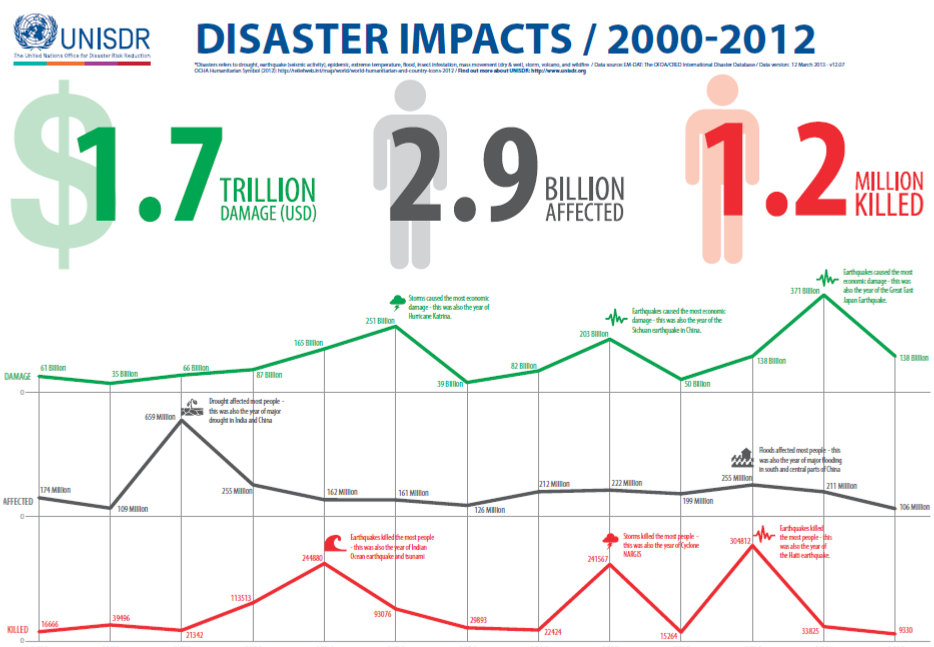


Figure 1: Disaster Statistics[2]

In a mobile ad-hoc network, the nodes are not constantly connected to each other, so information cannot be consistently delivered from one node to another, affecting network reliability. In networks where an end-to-end connectivity between users is not guaranteed, a delay or disruption tolerant network (DTN) architecture can be utilized [13]. In this type of architecture, information is delivered via a store-carry-and-forward approach as shown in Fig. 2. Based on Fig. 2, each node stores the information (e.g., messages, pictures, and the like) if there are no nodes available nearby (e.g., nodes 1 and 2 at $t = 0$). The node then carries the information until it meets another node along the way (e.g., nodes 1 and 2 at $t = 10$, nodes 2 and 3 at $t = 20$) and during node contact, information is forwarded. Information hops from one node to another until it is eventually delivered to the destination node. Although this approach is expected to incur delay in sending information, the delay that can be tolerated is still limited, especially in time-constrained networks such as a network in the area of a disaster. In the aftermath of the disaster, the first $36 - 48$ hours are critical and this limit is even shortened to 2 hours if people are wounded or trapped in buildings thus, information must be delivered within this period.

There are numerous studies that used DTN architecture in information collection. One particular study used DTN for the collection of disaster information by deploying additional message ferries (e.g., unmanned aviation vehicles, robotic insects) to the disaster zone [26]. However, this approach needs more time in setting up additional utilities; a more suitable approach is to use the personal devices (e.g., smartphones) of people in the disaster zone to gather disaster-related information and forward it to other users [9]. In disaster scenarios, it is of the utmost importance that there is access to a wide range of information such as information about the environment like the location of fires, flooded areas, and available resources. The mobile phones of people can then be used to gather these types of information in a variety of formats like text messages and photos. Currently, most mobile phones have a rich set of sensors, including a camera, microphone, global positioning system (GPS), gyroscope, accelerometer, and proximity sensor, resul-

[2] This image is taken from the UNISDR photo gallery and licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 2.0 Generic license. Source: http://www.flickr.com/photos/isdr/8567182347/

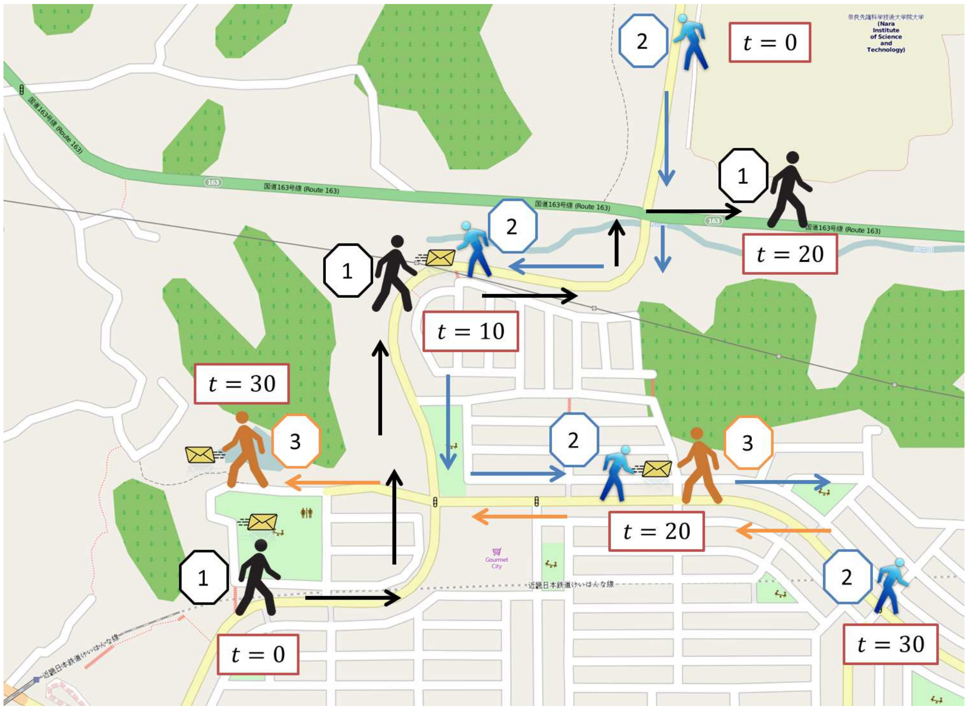


Figure 2: Store-Carry-and-Forward Approach

ting in the rise of participatory sensing applications in a variety of domains [16]. Mobile phones have been used in data gathering applications such as creating a noise map from sound samples collected using the phone's microphone [21]. They have also been used in achieving maximum coverage of a specific area [2] and in determining the optimum route for rescuers to save the most people in disaster-stricken areas [7]. Numerous studies describe the use of participatory sensing in achieving an effective and prompt information sharing for situational awareness in a disaster context [1]. A communication network relying on the use of mobile phones carried by people already within the disaster area was also proven to be effective in disseminating emergency information to the population in a DTN [11]. However, its effectiveness relied on the number of devices and the maximum allowed delay [5].

In addition to communication failures, there is also a surge in network traffic because of large amounts of data generated by people during the occurrence of a disaster leading to the congestion of the network, which can significantly hamper communications and emergency response efforts. Studies addressing the issue on network congestion in DTNs during a disaster either introduces a DTN architecture for an efficient disaster response network [8] or eliminates redundancies in disaster-related content [25] for fast message delivery outside the disaster zone. Unlike these existing studies, in this study, we focused on determining methods for collecting disaster-related information in the shortest possible time to provide immediate situational awareness over the disaster area.

In the first part of this thesis, we propose a data aggregation method using DTN to provide timely coverage of an area of interest (AoI) within the disaster area by collecting disaster-related messages in the shortest possible time. In the second part of this thesis, we propose a content-based image prioritization technique with DTN to send disaster-related images with critical content, collected from an AoI, in a timely manner to the appropriate recipients (or sinks).

First, in Chapter 2, we propose a DTN-based data aggregation method to achieve timely collection of information from an AoI within the disaster area. Due to the surge in network traffic in the disaster area, it is not possible to quickly collect all the messages created in the AoI with DTN. In our proposed method, we apply data aggregation to reduce data size by merging individual messages into one message using aggregation functions like max, min, sum, count, and average. Each message covers a specific area within the AoI and the aggregated message contains the merged coverage areas of the individual messages. The reduction of data size allows the message to traverse the network faster minimizing the time for information delivery. However, during aggregation, message duplication may be present in the aggregated message; users may create and share the same information on a location resulting in duplicate information. A Bloom filter is used to ensure that duplicate messages are not aggregated resulting in the accuracy of the aggregation results. An aggregation granularity (ϕ) metric is also introduced to ensure the resolution of the collected messages. In addition, to further reduce the message delivery time, the expected reaching time of a node to the sink (ert) metric is used in determining the node to forward the stored messages. We conducted computer simulations with a real geographical map and compared the proposed method with a conventional method based on epidemic routing in terms of information collection delay for varying number of mobile nodes. Based on their average latencies, the proposed method achieved a 9.8%, 14.9%, 9.2%, 5.9%, and 9.7% decrease in information collection delay than the conventional method using 200, 400, 600, 800, and 1000 mobile nodes, respectively. Also, based on the cumulative distribution graphs of the proposed method and the conventional method, 70% of all runs take less than $t = 2000$ s to attain 80% coverage of the AoI with 1000 mobile nodes using the proposed method whereas only 50% of all runs take less than $t = 2000$ s with the same parameters using the conventional method. From $t = 0$ to 2000 s, there is a 40% increase in the number of runs that achieved 80% coverage of the AoI using the proposed method compared with the conventional method further proving the effectiveness of the proposed method in collecting disaster information quickly.

Second, in Chapter 3, we propose a content-based image prioritization technique that enables disaster-related images with critical content to be collected from an AoI within the disaster area to be sent in a timely manner to the sinks. Due to the limited data transfer capacity of a DTN, there is a need to implement a prioritization technique to the collected images enabling the command posts to address critical data as soon as possible. In our proposed method, we apply image processing to detect important contents like fires or road blocks on the images and label them as images with critical content to be forwarded over the network prior to other images with noncritical content. A statistical fire color model was incorporated to the fire detection algorithm while a probabilistic Hough transform and some image transforms to detect edges were incorporated to the road block detection algorithm. We tested randomly chosen disaster-related images to detect critical content and achieved a detection rate of more than 80% using the proposed image processing algorithm. We also conducted computer simulations with a real geographical map and compared the proposed prioritization technique with the non-prioritization technique in terms of percentage of critical images received at the sink for varying number of images in the AoI. There was a 9.9%, 4.3%, and 10.1% increase when the number of images in the AoI are 100, 500, and 1000 images, respectively. This confirms that the proposed image prioritization reduced the delay in collecting critical images compared with the non-prioritization method.

2. Timely Collection of Messages in Disaster Areas

2.1 Introduction

In disaster scenarios, accurate and timely information on the locations of the people in need and what they urgently need are important for effective and immediate assistance. Even with the unavailability of the communication infrastructure, information must be collected with a small delay and with maximum coverage of an area of interest (AoI) within the disaster area. An efficient communication system, such as a delay tolerant mobile ad-hoc network, is needed to relay information between survivors and rescue teams hastening disaster relief efforts.

In this chapter, we describe the use of data aggregation over a DTN to achieve timely coverage of the AoI. People with mobile phones in the vicinity of the AoI create disaster-related messages at various places in the AoI and collect messages by exchanging them among other nodes through short-range wireless communication. However, due to the surge in network traffic in the disaster area, it is not possible to quickly collect all the messages created in the AoI with DTN. In our proposed method, we apply data aggregation to reduce data size thereby minimizing the time for information delivery. However, one common problem with data aggregation is the presence of duplicates. To prevent the aggregation of duplicate information, a Bloom filter is constructed for each created message (atomic or raw message). The Bloom filter consists of an n-bit array that maps the location information of a message to one of the array positions and sets the corresponding array position to 1. Before aggregating messages, their corresponding Bloom filters are checked and messages with Bloom filters having the same set of array positions equal to 1 or partly overlapping array positions equal to 1 are not aggregated. Moreover, the aggregation granularity metric is introduced to ensure the resolution of the collected information. Messages are only aggregated if the merged coverage area of the messages is less than the aggregation granularity. To reduce further the time for information delivery, the expected time of a node to reach the destination node (ert) is used to determine which node to forward the stored message. Information is opportunistically forwarded to the node with the lowest ert.

2.2 Data Aggregation Problem for Disaster Areas

2.2.1 Target Environment

Consider a disaster scenario during the early period of the recovery phase. In this period, knowledge about the situation in the disaster area is at its lowest point. There is lack of information or perhaps none at all and even if information about the affected area is available, it still needs to be up-to-date. Also, most of the time, the communication infrastructure is destroyed resulting in the difficulty of information collection from the affected area.

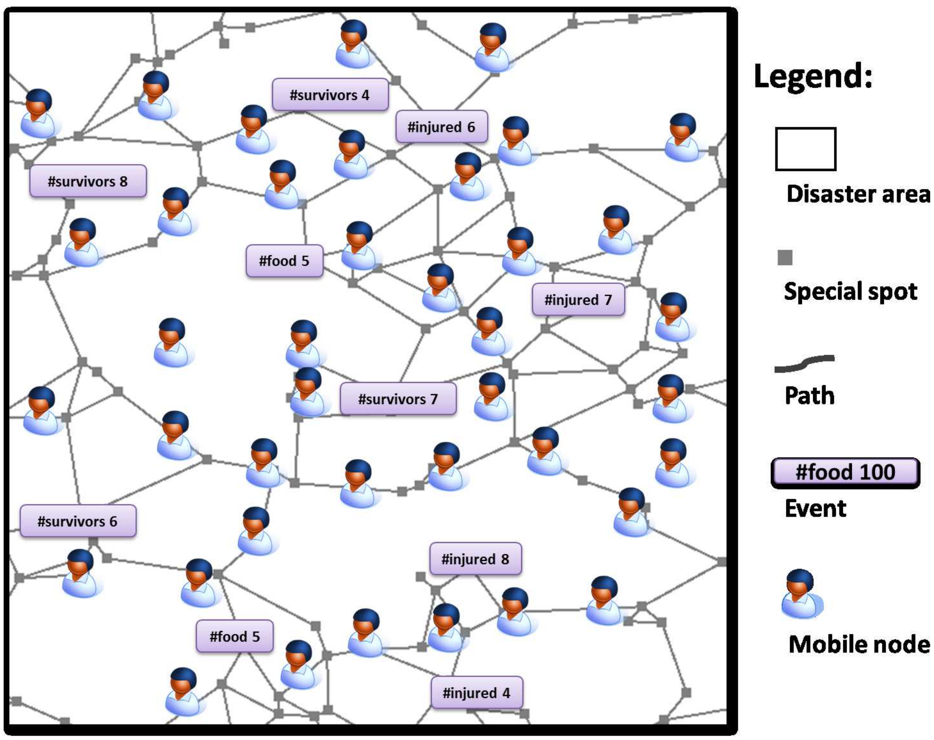


Figure 3: Disaster Area Scenario

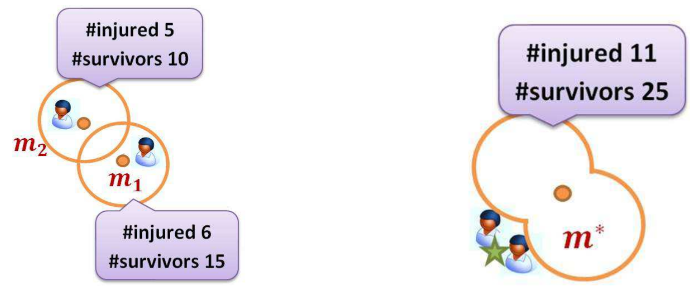
As shown in Fig. 3, a disaster area denoted by A_d consists of links (roads) between *special spots* (e.g., evacuation centers, hospitals, city halls and so on) and mobile nodes. The set of mobile nodes[3] existing in A_d is denoted by U. These mobile nodes move towards the nearest evacuation center within A_d, and

[3]The terms user, mobile node, and node are used interchangeably.

after arrival at the center, the nodes move only in its vicinity as modeled in [10], wherein the evacuation center is considered home by most users during this time. A stationary node at a special spot can send a snapshot query about a particular area of interest denoted by $AoI \subseteq A_d$. A query q is denoted by $\langle u_0, t_0, AoI, \phi \rangle$, where u_0 is the query node, t_0 is the time when the query is issued, and AoI is the area of interest from which information is to be collected with an aggregation granularity ϕ, the maximum area within which each message is effective. Information is then collected and aggregated from different users who pass through the AoI.

Each node receiving the query and existing in the AoI creates a message m when it finds a disaster-related event within the AoI and its current location is not covered by its retained messages. Each message m has an effective circular area denoted by $m.area$ with radius r_s. The messages are numerical information on survivors, shelter capacity, and available resources such as food, first responders, and utilities. Nodes exchange messages with other nodes upon contact so that the set of collected messages covers the AoI and is delivered to the sink. AoI coverage is determined from the union of the effective areas of the messages received by the sink.

Moreover, data aggregation in each node is based on the aggregation granularity metric ϕ, which specifies the maximum area (or radius) of the effective area of aggregated messages. We adopt our definition of aggregation granularity from the concept of information granularity, which by definition refers to the extent of detail within the information. At lower levels, information exhibits fine granularity and at higher levels, information becomes coarser because it is summarized or aggregation. Thus, in our case, a lower aggregation granularity ϕ value means that the query sender needs a more detailed data from the area while a higher aggregation granularity ϕ value means that the sender only wants a summary data from the area. Fig. 4 shows an example of aggregating messages. In Fig. 4(a), two users create *atomic messages* m_1 and m_2 with information on the number of injured persons and survivors from different areas. *Atomic messages* are nonaggregated messages (or raw messages) created by the user. As shown in Fig. 4(b), upon contact, the messages of the two nodes are aggregated into one message m^* containing information on the total number of injured persons and survivors from the two atomic messages. The effective areas of the atomic messages are merged and the resulting merged area is the effective area of the aggregated message $m^{*}.area$. However, a node only aggregates the messages if the effective area of the aggregated messages is less than the aggregation granularity ϕ.



((a)) Atomic Messages ((b)) Aggregated Messages

Figure 4: Message Aggregation

2.2.2 Assumptions

Each user in U is assumed to have a certain role in the disaster area, which affects its movement within the area. It is assumed that each user falls into a certain type of group in the disaster area: a first responder, an official, or an ordinary person. In disaster areas, there are spots that need urgent attention from the rescuers and these special spots are called *hotspots*. Examples of hotspots are buildings on fire, collapsed infrastructure, wounded persons, and the like. Naturally, first responders tend to move towards these hotspots while ordinary persons tend to move away from these hotspots. Also, officials tend to move towards the hotspots but only stay at the hotspot for about a minute or less while first responders stay at the hotspot for a longer time around 15 minutes, which is their average time at the scene during emergencies [3]. Using this special role-based disaster mobility model, any point in A_d can eventually be covered by some users in U, that is, the point will be visited by some user at some time in the future. Each user in U will also have direct or indirect contact with any other user in U eventually, wherein the direct contact represents the situation of two users existing in their common communication range and the indirect contact is defined as the transitive closure of the direct contact.

Each node $u \in U$ is able to *create*, *send*, *receive*, or *aggregate* messages in different locations. Its location at time t is denoted by $u.pos(t)$ and determined either through GPS or estimated based on some other means. The create action of node u includes sensing information and creating an atomic message m containing the information that covers a circular area $m.area$ with radius r_s. When creating an atomic message, its center location $m.center$ is the event place and $m.area$ does not contain any other events. Each node also has a limited storage for the collected information.

Each node is capable of short-range wireless communication, such as Bluetooth or WiFi, and a unit disc model is adopted with every node having the same communication range r_c. Within the range r_c, each node can send or receive data from other nodes with limited *contact duration*, defined as the available time when a contact (communication opportunity) occurs between two nodes. The contact duration between nodes u and u' starting from time t is denoted by $cd(u, u', t)$. In addition, the maximum transmission speed of the available short-range wireless communication is denoted by BW and the transferrable data amount of one contact is defined by $cd(u, u', t) \times BW$.

When a node creates or receives messages, it aggregates its stored messages depending on the aggregation granularity ϕ set by the query node. Let $\phi.a$ and $\phi.r$ denote the maximum area size and radius that a message can cover, respectively. Let m_1 and m_2 denote atomic messages with corresponding coverage areas $m_1.area$ and $m_2.area$. As shown in Fig. 4(a), m_1 and m_2 will only be aggregated to m^* if the following conditions are met: (a) the coverage areas of the two messages are overlapping,

$$m_1.area \cap m_2.area \neq \emptyset \tag{1}$$

(b) the merged area is smaller than the maximum area size,

$$|m_1.area \cup m_2.area| \leq \phi.a \tag{2}$$

(c) the farthest distance from the center point of the merged area is not greater than $\phi.r$,

$$\textsc{Radius}(m_i.area \cup m_j.area) \leq \phi.r \tag{3}$$

and (d) the center locations of m_1 and m_2 are not within the radius r_s of each other,

$$\textsc{Dist}(m_i.center, m_j.center) > \textsc{Max}(m_i.r_s, m_j.r_s) \tag{4}$$

wherein this is the condition to avoid the merging of atomic messages with the same event.

Let t denote the time elapsed since a query is issued ($t = 0$ when the query is issued). As time t progresses, the node may move from one location to another or perform a certain action type $A_t = \{create, send, receive, aggregate\}$. Time is divided into time periods $T_0, T_1, \ldots$ with length P. Each period is also divided into two parts: *active interval* and *sleep interval*. In each time period, the first pP portion is assigned as the active interval and the remaining $(1-p)P$ portion as the sleep interval, where p is a system parameter such that $0 < p < 1$. Each node is assumed to have a sufficiently accurate clock to synchronize with other nodes in active and sleep intervals wherein it sends a beacon message for finding other nodes only in the active interval. During the sleep interval, each node turns its wireless communication device to sleep mode if there is no contact with other nodes to save energy consumption.

2.2.3 Problem Definition

Given a query specifying the AoI, our problem is to derive the set of actions taken by each node in U that collects and delivers the set of messages M_0 covering the AoI to the sink u_0 in minimal time.

Each node $u_i \in U$ has the set of action tuples Act_i, where $a_{i_j} = \langle u_i, a_t, M_i, t, m\rangle \in Act_i$ refers to the jth action performed by u_i. The tuple $\langle u_i, a_t, M_i, t, m\rangle$ represents that node u_i performs action a_t on message set M_i at time t and the resulting (created or aggregated) message is m.

For every send action of node u_i, there is a corresponding receive action of node u_j. The send and receive actions must be performed during the contact duration, and the entire message must also be transferred within the duration.

Thus, the following equation must hold.

$$\begin{aligned}\forall u_i \in U, \quad \forall \langle u_i, send(u_j), M_i, t, m\rangle \in Act_i, \quad \exists u_j \in U,\\ \exists \langle u_j, receive(u_i), M_i, t', m\rangle \in Act_j \wedge \exists t'' cd(u_i, u_j, t'') > 0\\ \text{such that} \quad t'' \leq t \leq t' \leq t'' + cd(u_i, u_j, t'')\\ \wedge\, Size(M_i) \leq cd(u_i, u_j, t'') \times BW \quad (5)\end{aligned}$$

where

$Size(M_i)$ is the total byte size of the messages in message set M_i.

The set of messages M_0 delivered to sink u_0 is defined as follows,

$$M_0 = \bigcup_{\langle u_0, receive, M, t, null\rangle \in Act_0} M \quad (6)$$

and must achieve full coverage of the AoI.

$$\bigcup_{m \in M_0} m.area \supseteq area(AoI), \quad (7)$$

Each message in M_0 must have been created or aggregated by some nodes according to the following condition:

$$\forall m \in M_0, \quad IsCreated(m) \vee IsAggregated(m) \quad (8)$$

where

$IsCreated(m) \stackrel{def}{=} \exists u_i \in U,$
$\exists \langle u_i, create, \emptyset, t, m\rangle \in Act_i$

$IsAggregated(m) \stackrel{def}{=} \exists u_i \in U,$
$\exists \langle u_i, aggregate, M_a, t, m\rangle \in Act_i \quad \wedge \quad \forall m' \in M_a,$
$IsCreated(m') \vee IsAggregated(m')$.

Data collection delay D at u_0 is defined as follows.

$$D = max\{t | \langle u_0, receive, M, t, m\rangle \in Act_0\} \quad (9)$$

Thus, given A_d, U, and a query q with u_0, t_0, AoI, and ϕ, the problem is defined as the minimum time data aggregation (MTDA) problem to decide the set of actions Act_i for each node u_i with the objective function defined as:

$$\textbf{minimize } D, \quad \textbf{subject to } (5), (7), \textbf{and } (8)$$

The MTDA problem is an NP-hard problem since it implies the minimum geometric disk cover problem as a special case, even when we do not apply aggregation to messages. Thus, we devise a heuristic solution to solve the MTDA problem in the next section.

2.3 Data Aggregation Algorithm

In this section, a greedy algorithm is presented to solve the MTDA problem described in Section 2.2.3. The NodeAction algorithm shown in Algorithm 1 is our main algorithm. This algorithm is executed at each node $u_i \in U$ independently of the other nodes and determines action $a_t \in \{create, send, receive, aggregate\}$ of u_i on its message set M_i over time. Each node runs the algorithm when it receives the query containing u_0, t_0, AoI, and ϕ. The query is initiated by a stationary node at a special spot (e.g. command post) known as the sink wherein the Time to Live (TTL) of the query is set and can be varied depending on the situation. These neighbor nodes, which may be mobile, forward the query to its subsequent neighbor nodes and so on. Query distribution is sent through flooding wherein most nodes in A_d receive the query at some point. Moreover, each node has an electric map of the target disaster area A_d and its time period P (e.g., 10 s) for the duty cycle is set, with an active interval (e.g., 1 s) and a sleep interval (e.g., 9 s). When a query is received, a node sends a beacon message to find its neighbor nodes during its active interval. However, during its sleep interval, it creates a message, exchanges messages with its neighbor nodes, aggregates messages, or just sleeps if there are no neighbor nodes or there are no messages that can be aggregated.

The variables used in the algorithms with their corresponding meanings are shown in Table 1 for readability purposes. As shown in Algorithm 1, it is assumed that the location of node u_i is known and node u_i has received the query consisting of the identity of sink u_0 and its position $u_0.pos(0)$, time t_0 when the query is issued, area of interest AoI, and aggregation granularity ϕ. During this instance, time t is set to the time elapsed from the query sending time t_0 (Algorithm 1 line 1). The variable M_i represents the message set retained by u_i and is initialized to be empty (Algorithm 1 line 2). At this point, node u_i is not in contact with any node as represented by a null N_i (Algorithm 1 line 3). Node

Table 1: Algorithm Variables

Variable	Meaning
$ert(u_i)$	Expected reaching time of u_i to u_0
m_i	Atomic or aggregated message
$m_i.area$	Coverage area of message m_i
$m_i.replica$	Replica number of message m_i
M_i	Current message set of mobile node u_i
M'	Aggregated message set
N_i	Neighbor node set of mobile node u_i
ϕ	Aggregation granularity
q	Query message
$replica$	Replica number of message
t	Elapsed time
t_0	Time when query was issued
T	Predetermined deadline of the query q
u_i, u_j	Mobile nodes
$u_i.pos(t)$	Location of mobile node u_i at time t
u_m	Mobile node with the lowest ert
u_0	Query sender or sink

u_i then enters into a loop performing lines 4–18 (Algorithm 1) until reaching the predetermined deadline T, which may be equivalent to the time that the query is not needed anymore preventing node u_i from going into an infinite loop, or node u_i receives an acknowledgement, piggybacked in the messages, that node u_0 has received M_i.

The following subsections explain in detail the different parts of our main algorithm: neighbor discovery, message creation, message exchange, and message aggregation.

2.3.1 Neighbor Discovery

During the active interval in each time period, node u_i sends a beacon message for neighbor node discovery (Algorithm 2 line 5). The active interval period is divided

Algorithm 1 NodeAction($u_i.pos(t), q$)

Input: $u_i.pos(t)$, $replica$, q=$\langle u_0, t_0, AoI, \phi \rangle$
Output: Node action schedule of u_i

$t \leftarrow$ current time - t_0
$M_i \leftarrow \emptyset$
$N_i \leftarrow \emptyset$
while $t \leq T$ **or** M_i covering AoI is received by u_0 **do**
  $N_i \leftarrow$ NeighborDiscovery(u_i, N_i, t, t_0)
  while t is in sleep interval **do**
    if $u_i.pos(t)$ is within AoI, within $m_i.area$, **and** outside the covered area of M_i **then**
      u_i creates m_i
      $m_i.replica \leftarrow replica$
      $M_i \leftarrow$ Aggregate($M_i, \{m_i\}, \phi$)
    end if
    if u_i has a new message **then**
      $M_i \leftarrow$ MessageExchange(u_i, u_0, N_i, M_i)
      $M_i \leftarrow$ Aggregate(M_i, M_i, ϕ)
    end if
    $t \leftarrow$ current time - t_0
  end while
end while

into slots (e.g., 10 ms). Each node sends a beacon message during its active interval after the randomly decided backoff time (up to the active interval length). When node u_i receives the beacon message from another node u_j (Algorithm 2 line 8), nodes u_i and u_j are assumed to be in contact and node u_j is added to the neighbor node set N_i (Algorithm 2 line 9). All of the nodes within r_c of node u_i are added to its neighbor node set N_i. In addition, the clocks of all nodes are synchronized with the global clock using GPS or by some other means such that the active and sleep intervals of all nodes are synchronized with sufficient accuracy.

2.3.2 Message Creation

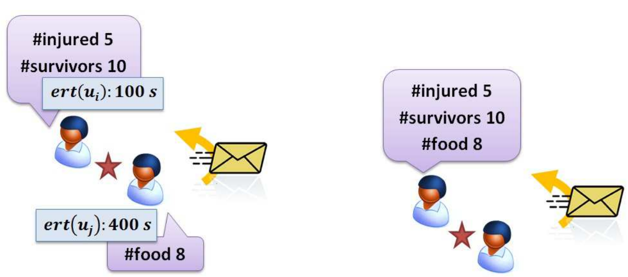
When t is within the sleep interval, node u_i checks whether its current location is within the AoI, within the effective area $m_i.area$ of an event, and outside the

Algorithm 2 NeighborDiscovery(u_i, N_i, t, t_0)

Input: Mobile node u_i, current neighbor node set N_i
Output: Updated neighbor node set N_i

```
t ← current time - t_0
backoff ← random number
while t is in active interval do
    if backoff = 0 then
        u_i sends a beacon message
        backoff ← random number
    end if
    if u_i receives a beacon message from another node u_j then
        N_i ← N_i ∪ {u_j}
    end if
    backoff ← backoff − (current time−t)
    t ← current time - t_0
end while
return N_i
```

covered area of its message set M_i (Algorithm 1 line 7). If it is true, u_i creates a message m_i with a center location and an effective area the same as the said event (Algorithm 1 line 8). A message replica count is also initialized to *replica* (a constant number given in advance) to increase the probability of message delivery (Algorithm 1 line 9). The newly created message m_i is then aggregated with the other local messages M_i of node u_i, which is explained in detail in Section 2.3.5 (Algorithm 1 line 4).



(c) Before (d) After

Figure 5: Message Exchange

2.3.3 Message Exchange

When node u_i has determined its neighbor set N_i, it decides whether to retain or to forward its stored messages (Algorithm 1 line 13). The expected reaching time of each contact node to sink u_0 denoted by $ert(u_i)$ is used to determine the node action. The expected reaching time $ert(u_i)$ can be determined from the node's speed $v(u_i)$, moving direction, and the distance of the node from u_0, $d(u_i, u_0)$, that can be computed from the positions of nodes u_i and u_0 as well as the links of disaster area A_d (Algorithm 3 line 4). Assume that node u_i is travelling at a speed $v(u_i) = 1\ m/s$ to a spot (intersection) 30 m far away from the current location and the spot is 70 m from node u_0, then $ert(u_i) = \frac{30+70}{1} = 100\ s$. This means that it will take 100 s for u_i to come into the possible earliest contact with node u_0. Node u_i then determines the node with the lowest ert from its N_i (Algorithm 3 lines 5–13), say u_m. If node u_i is not the node with the lowest ert (Algorithm 3 line 14), node u_i sends its stored messages to node u_m (Algorithm 3 line 15) and node u_m receives the stored messages of node u_i (Algorithm 3 line 16). The replica number of each stored message is also decremented every time it is sent to another node (Algorithm 3 line 18). When node u_i sends its stored messages, it retains its messages until the replica number of a message reaches 0 wherein the message is deleted from the node's buffer (Algorithm 3 lines 19–22). Let us take the example shown in Fig. 5 and assume that the replica number of both messages is 1. In Fig. 5(c), $ert(u_i) < ert(u_j)$ thus, node u_j sends its stored messages to node u_i and removes the messages from its buffer as shown in Fig. 5(d) since the replica number of the message after the exchange becomes 0. Moreover, if node u_i is the node with the lowest ert and it receives a message set M_j from a neighbor node (Algorithm 3 line 24), node u_i retains its stored messages and adds the received messages to its local messages (Algorithm 3 line 25).

2.3.4 Message Structure

Each message contains disaster-related information with a corresponding area coverage, a replica number, and a Bloom filter. The disaster-related information contained in the message follows a Tweet-based syntax with a certain limit (e.g., 140 characters) and containing statistical information on survivors, shelter capa-

Algorithm 3 MESSAGEEXCHANGE(u_i, u_0, N_i, M_i)

Input: Mobile node u_i, sink u_0, neighbor node set N_i, current message set M_i
Output: Updated message set M_i

$M_j \leftarrow null$
$u_j \leftarrow null$
$u_m \leftarrow u_i$
$minert \leftarrow ert(u_i)$ calculated from the locations of u_i and u_0
while $N_i \neq \emptyset$ **do**
  $u_j \leftarrow$ select one node from N_i at random
  Determine $ert(u_j)$ based on the locations of u_j and u_0
  if $ert(u_j) < minert$ **then**
    $minert \leftarrow ert(u_j)$
    $u_m \leftarrow u_j$
  end if
  $N_i \leftarrow N_i - \{u_j\}$
end while
if $u_m \neq u_i$ **then**
  u_i sends M_i to u_m
  u_m receives M_i from u_i
  for all $m_i \in M_i$ **do**
    $m_i.replica \leftarrow m_i.replica - 1$
    if $m_i.replica = 0$ **then**
      delete m_i
      $M_i \leftarrow M_i - \{m_i\}$
    end if
  end for
else if $u_m = u_i$ and u_i receives a message set M_j **then**
  $M_i \leftarrow M_i \cup M_j$
end if
return M_i

city, and available resources. The information consists of a keyword marked by a hash tag followed by a number that describes the keyword. Examples of Tweet-like messages are shown in Fig. 5. Let us take a message containing the following: `#survivors 10`. In this message, the keyword is survivors and the number is 10. This means that there are 10 survivors around the message's location. An atomic (raw) message will contain only one keyword with a corresponding number. However, aggregated messages will contain a number of keywords marked by hash tags with their corresponding numbers. To aggregate messages, the sums of the numbers corresponding to the same keywords are determined. Let us take the atomic messages containing the following: `#survivors 10` and `#survivors 5`. The resulting aggregate message of the two atomic messages is `#survivors 15`. However, messages are only aggregated if conditions (1) to (3) in Section 3 are satisfied. The equivalent coverages of the atomic messages are also merged by approximating a circle that will contain the coverages of the messages.

2.3.5 Aggregation Granularity

When a new message set M_j is received by u_i, it aggregates or concatenates message set M_j with the local messages M_i depending on ϕ. Let M' contain the aggregated messages, which is set to be empty initially (Algorithm 4 line 1). In the aggregation process, the entire message pairs of M_i and M_j are tried to be merged (Algorithm 4 lines 2–8). However, only the message pairs that satisfy conditions (1) to (3) in Section 3 are aggregated (Algorithm 4 line 3). If these conditions are met, messages m_i and m_j are aggregated depending on the aggregation function f_a resulting in the aggregated message m_a (Algorithm 4 line 4), which is then added to the set of aggregated messages M' (Algorithm 4 line 6). However, if m_i and m_j are atomic messages, the following condition must also hold before aggregation: the center locations of m_i and m_j are not within the radius r_s of each other.

Consider node u_1 in the AoI as shown in Fig. 4(a), in which node u_1 receives a query. During the sleep interval of u_1, it creates an atomic message m_1 with $r_s = 5\ m$. Then, as node u_1 becomes active, it sends beacon messages to discover its neighbor nodes. Suppose that nodes u_1 and u_2 come into contact, node u_1 compares its expected reaching time $ert(u_1)$ with the expected reaching time of u_2, $ert(u_2)$. Let us suppose that the expected reaching time of u_1 $ert(u_1)$ is less than the expected reaching time of u_2 $ert(u_2)$, message m_2 of node u_2 is sent to node u_1. Aggregation of the messages occurs if conditions (1) to (3) of Section 3 hold. The aggregated message m^* will then contain the information on the total number of survivors from the two messages m_1 and m_2.

Algorithm 4 AGGREGATE(M_i, M_j, ϕ)

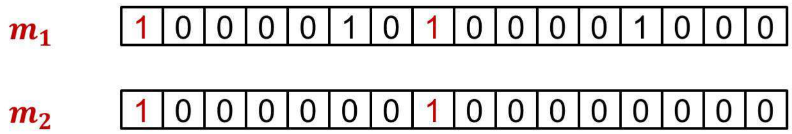
Input: Message sets M_i and M_j, aggregation granularity ϕ
Output: Aggregated message set M'

1: $M' \leftarrow \emptyset$
2: **for** each pair $(m_i, m_j) \in M_i \times M_j$ **do**
3:     **if** $m_i.area \cap m_j.area \neq \emptyset \wedge |m_i.area \cup m_j.area| \leq \phi.a \wedge \text{RADIUS}(m_i.area \cup m_j.area) \leq \phi.r \wedge \text{AND}(\text{BF}(m_i), \text{BF}(m_j)) \neq \text{BF}(m_i) \wedge \text{AND}(\text{BF}(m_i), \text{BF}(m_j)) \neq \text{BF}(m_j)$ **then**
4:         $m_a \leftarrow$ aggregate messages m_i and m_j
5:         $\text{BF}(m_a) \leftarrow \text{OR}(\text{BF}(m_i), \text{BF}(m_j))$
6:         $M' \leftarrow M' \cup \{m_a\}$
7:     **end if**
8: **end for**
9: **return** M'

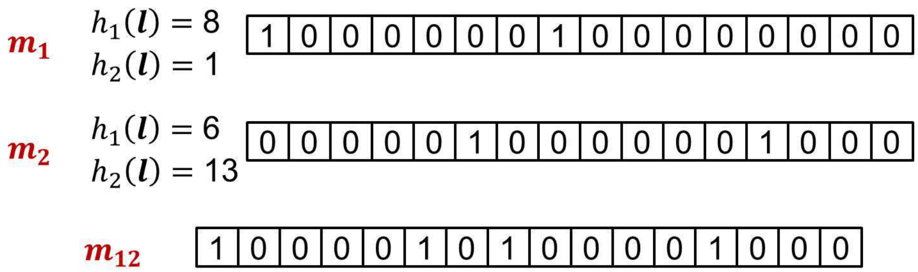
2.3.6 Bloom Filter

In data aggregation, there are instances that duplication occurs. To prevent this, a Bloom filter is constructed for each created message (atomic or raw message). It determines if a particular message is already part of the aggregated message. The Bloom filter is an array of n bits representing a set of messages. Initially, all the bits in the filter are set to zero and when an atomic message is added, k bits in the filter are set to 1 depending on the chosen number of hash codes. The Bloom filter uses a hash function to map the messages to random numbers within the index range of the filter. For the hash function, the MD5 hash algorithm is used since it is a popular choice for the hash function of Bloom filters [23]. Each atomic message is directly mapped to a bit sequence using its creation location. Let us take the example shown in Fig. 6(b). The creation location l of atomic message m_1 is hashed and the resulting index h_i is equivalent to its array position b_j in the Bloom filter. In the example, for atomic message m, there are two resulting hash indices $h_1(l)$ and $h_2(l)$. The resulting indices 8 and 1 correspond to the array positions of its Bloom filter thus, the positions b_8 and b_1 of its Bloom filter are set.

Before aggregating atomic messages, their corresponding Bloom filters are checked (Algorithm 4 line 3). If they contain the same set of array positions equal to 1, the atomic messages are not aggregated. If otherwise, the messages



(a) Duplicate Detection



(b) Aggregation

Figure 6: Bloom Filter

are aggregated and the Bloom filter of the aggregated message is the result of the OR operation between the corresponding Bloom filters of the atomic messages (Algorithm 4 line 5). Let us take for example Fig. 6(b), wherein m_1 has the resulting hash indices of 8 and 1 while m_2 has 6 and 13. This means that the two messages do not contain any identical atomic message and can be aggregated. To determine the bit sequence of the aggregated message m_{12}, an OR operation of the Bloom filters of the atomic messages m_1 and m_2 is executed that is, $\textsc{BloomFilter}(m_{12}) = \text{OR}(\textsc{BloomFilter}(m_1), \textsc{BloomFilter}(m_2))$. The resulting Bloom filter of the aggregated message m_{12} has positions b_1, b_6, b_8, and b_{13} set to 1. Moreover, in our proposed data aggregation, there are situations when atomic messages and/or aggregated messages are to be merged. To determine if the messages can be aggregated, their corresponding Bloom filters are checked. If their Bloom filters have the same set of array positions equal to 1 or partly overlapping array positions equal to 1, the messages are not aggregated just like in Fig. 6(a), wherein the Bloom filters of m_1 and m_2 have an overlapping bit sequence of k bits (k=2) that is, both positions b_8 and b_1 are set for the two Bloom filters. Therefore, there is an atomic message contained in m_1 and m_2 that are identical and the messages are not merged. If otherwise, the atomic message becomes part of the aggregated message and its corresponding Bloom filter is merged with the aggregated message Bloom filter using the OR operation. Thus, the resulting message set after aggregation consists of local atomic/aggregated messages and received atomic/aggregated messages.

In relation to the correctness or the accuracy of the message aggregation, let us take for example aggregated messages m_{12} and m_{13}, which contain atomic messages m_1 and m_2 and atomic messages m_1 and m_3, respectively. The Bloom filters of the aggregated messages will have an overlap because they contain a common atomic message m_1 thus, these messages would not be merged because it will result in the double counting of the information contained in atomic message m_1. These messages are then delivered to the sink as separate messages instead of one aggregate and when the sink receives these messages, it knows that they have an overlapping area but it cannot exactly extract the information in the overlapping area. Resolving this problem is part of the future work.

2.4 Performance Evaluation

The performance of the proposed algorithm is evaluated using a custom simulator. In order to evaluate the proposed algorithm, the time for the aggregated messages covering 80% of the *AoI* to arrive at its destination is determined.

2.4.1 Simulation Configuration

In this study, a custom simulator is used since the contact times of nodes are considered to be more important than its physical and link layer details. Therefore, we implemented the simulator so that a node can discover and connect with another node at transmission range instantly. Simulation consists of mobile nodes placed uniformly at random over a two-dimensional plane with a $3\ km \times 3\ km$ area. The area is based on an actual map within the vicinity of Takayama Science Town as shown in Fig. 7. The skeleton map represents the road network, which is composed of streets between special spots (e.g. hospitals, universities, and the like). In addition, the locations of the evacuation centers are based on actual evacuation centers published by the Ikoma City government. A special role-based disaster mobility model (Section 2.2.2) is adopted wherein the nodes only travel along the road network. Each node moves at a random speed with destinations conforming to its role and with routes based on Djikstra's shortest path algorithm. Disaster-related events with varying radii were generated with locations selected randomly inside the AoI depending on the type of information. For example, information about an injured person has an effective radius of 1 m while information about a shelter has an effective radius of 4 m. Two queries were also issued at the start of the simulation with TTL = 1 hr and were distributed by flooding. The locations of the query senders (or sinks) were randomly selected among the locations of the special spots within the disaster area A_d. The flooding overhead was considered by specifying corresponding data sizes for query and beacon messages. A simulation warming time was set wherein during this time, the nodes move according to its mobility model without performing any action. All nodes also have the same buffer size and transmission range. A total of 5 runs were taken with two queries for each simulation setting. Table 2 shows a summary of the default values used in the simulation. In order to facilitate a congested scenario in the network, the network bandwidth was set to 1 Mbps, the nonaggregated or raw message size to 5 KB, and the number of events within the AoI to 1000 events. This is because there is usually a surge in network traffic within the disaster area as a result of the large amount of messages generated by the people.

2.4.2 Message Delivery Latency

We evaluated the time for the aggregated messages to arrive at the sink in response to a query. Different parameters were varied such as the number of mobile nodes present in the disaster area A_d wherein the average message delivery latencies and its corresponding variances were determined. Fig. 8 shows the effect on message delivery latency with the increase in the number of mobile nodes present in A_d. Based on the figure, there is a 34.1% decrease in message delivery latency when the number of mobile nodes present in A_d is increased from 200 nodes to 1000 nodes. The significant decrease is due to the increase in contact opportunities between nodes. During these contact opportunities, messages are exchanged and aggregated to achieve the required coverage of the AoI and the increase in contact opportunities leads to a higher probability that more of the exchanged

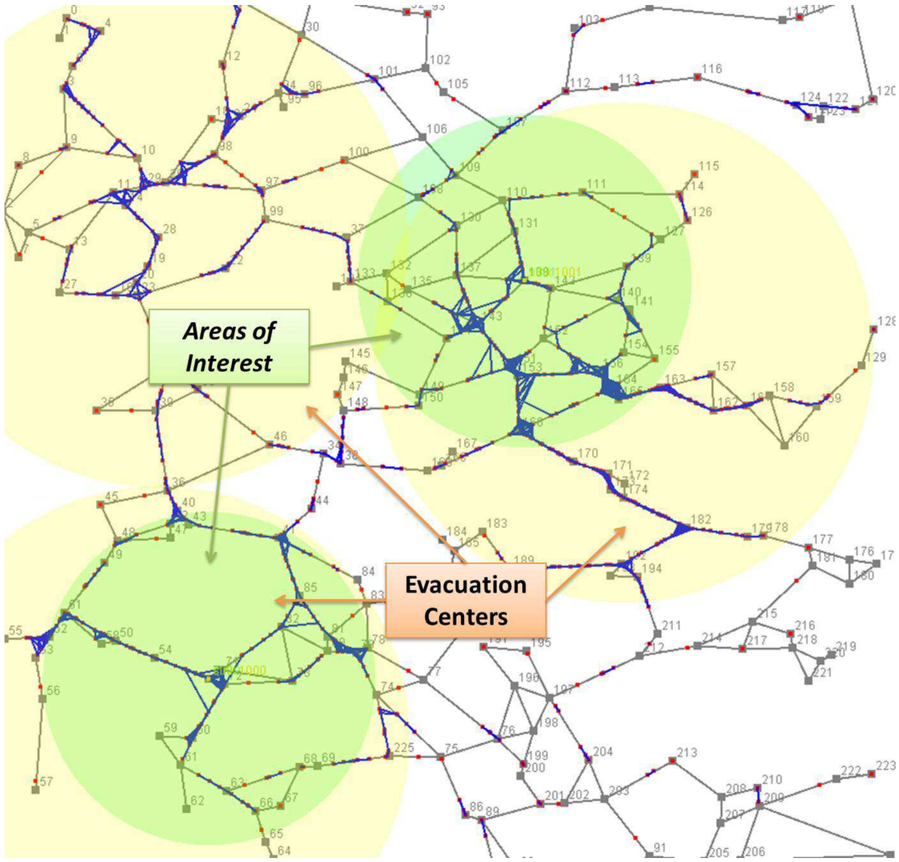


Figure 7: Simulation Environment

and aggregated messages reach the sink faster.

2.4.3 Aggregation Granularity

Another parameter that needs to be taken into consideration is the aggregation granularity ϕ. This sets the level of detail in the aggregated messages. The simulation result for the variation of the aggregation granularity metric is shown in Fig. 9 with 1000 mobile nodes present in A_d. Based on the results, aggregation granularity has an effect on message delivery latency. As shown in the figure, the average message delivery latency when $\phi = 10\ m$ is 2216.6 s while the message delivery latency when $\phi = 50\ m$ is 1913.2 s, which is a 13.7% decrease in latency. This decrease is due to the increase in aggregation granularity wherein more messages are merged into a single message thereby reducing message size and the

Table 2: Simulation Parameters

Parameter	Default Value
Network	
Bandwidth	1 Mbps
Buffer size	100–500 KB
Transmission range	50 m
Map	
Disaster area size	3 km x 3 km
No. of map points	226
No. of hotspots	120
No. of evacuation centers	3
Evacuation center range	750 m
AoI radius	500 m
No. of events within AoI	1000
No. of queries	2
No. of mobile nodes	200–1000
Node ratio (citizens:officials:responders)	3:1:1
Node	
Speed	1.0 - 1.4 m/s
Duty cycle period	10 s
Active interval	1 s
Sleep interval	9 s
Warming time	1000 s
Message	
Size	5 KB
Beacon message size	10 bytes
Query message size	300 bytes
Replica number	1–7
Aggregation granularity ϕ	0–100 m
Bloom filter size	256 bits

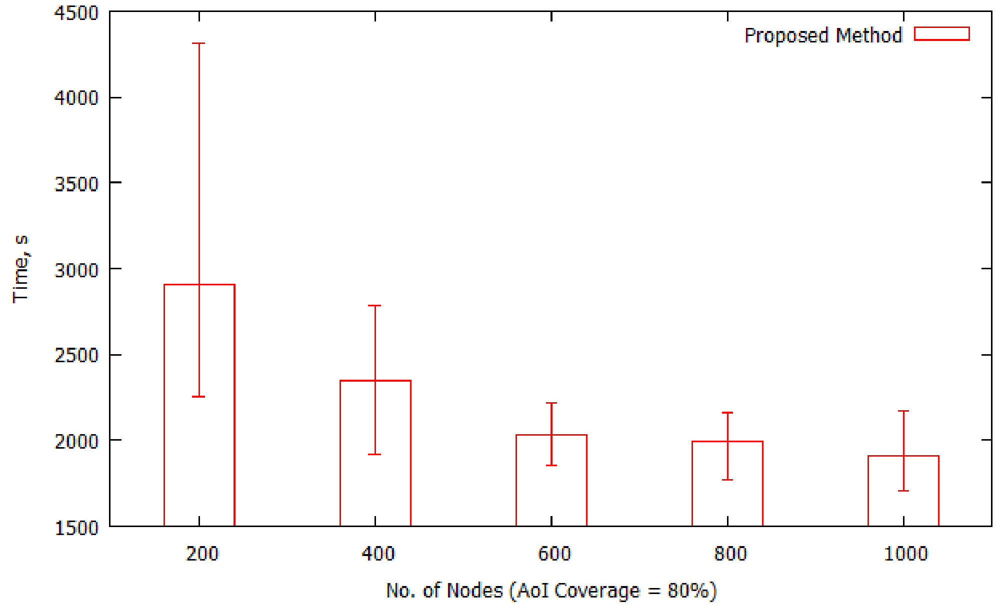


Figure 8: Effect of Node Density on Message Delivery Latency (Buffer Size = 500 KB, Replica = 5)

aggregated message is transmitted over the network faster than non-aggregated messages. Moreover, as shown also in the figure, there is a boundary on the decreasing effect of the aggregation granularity on message delivery latency. When ϕ is greater than 50 m, there seems to be no significant effect on the message delivery latency because of the absence of non-aggregated messages that satisfies the set conditions or other uncontrolled factors like the mobility of the nodes.

2.4.4 Comparison with Epidemic Routing Limited

The proposed algorithm was also compared to epidemic routing limited. In epidemic routing limited, the node's stored messages are copied to its neighbor nodes based on the set replica number that is, the number of copied messages are limited to the replica number just like in the proposed method. We used epidemic routing limited instead of the general epidemic routing to prove the effectiveness of the proposed aggregation method in decreasing message delivery latency.

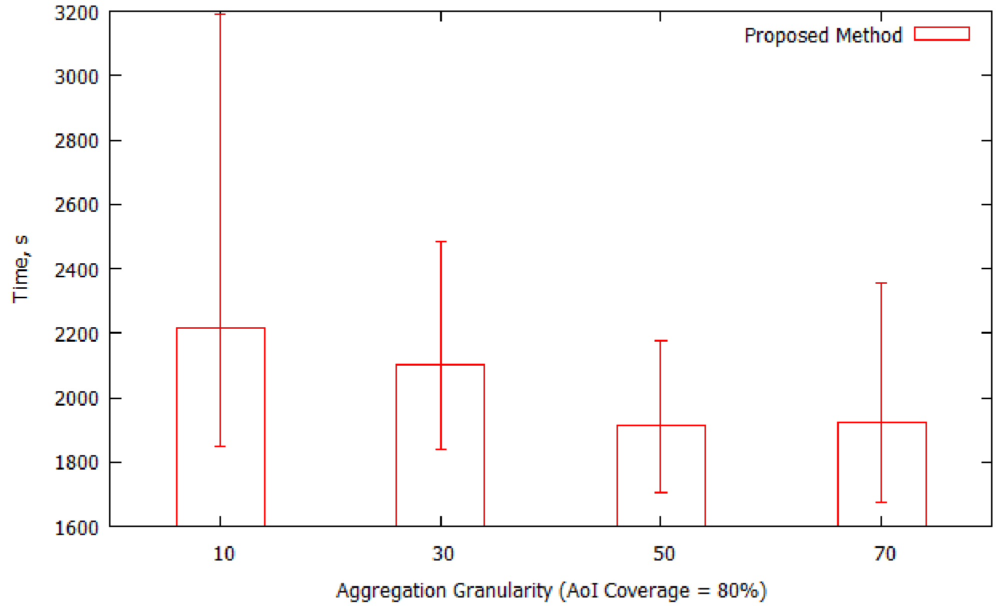


Figure 9: Effect of Aggregation Granularity on Message Delivery Latency (Buffer Size = 500 KB, No. of Nodes = 1000, Replica = 5)

2.4.5 Node Density

The average message delivery latencies of varying node density were plotted including their corresponding variances. Fig. 10 compares the performance of the proposed method and epidemic routing limited in terms of message delivery latency by varying the number of mobile nodes in A_d. The figure shows that with varying mobile nodes present in A_d, the proposed method achieves a better performance in message delivery latency compared to epidemic routing limited. Using the proposed method, the information collection delay decreased by 9.8%, 14.9%, 9.2%, 5.9%, and 9.7% compared to epidemic routing limited with 200, 400, 600, 800, and 1000 mobile nodes, respectively.

2.4.6 Replica Number and Buffer Size

We also evaluated our proposed algorithm with varying replica numbers and buffer sizes to further show the efficiency of the proposed method over epidemic routing limited. Fig. 11 compares the performance of the proposed method and

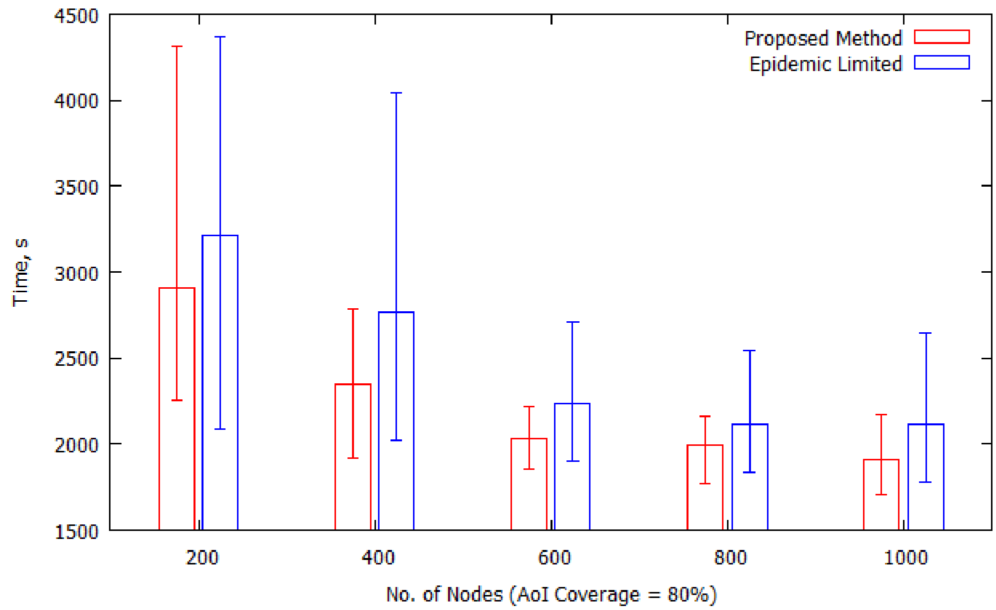


Figure 10: Comparison of Message Delivery Latency with Epidemic Routing vs. Node Density (Buffer Size = 500 KB, Replica = 5)

epidemic routing limited in terms of message delivery latency by varying the replica numbers of the message. The figure shows that with replica number = 1, the average message delivery latencies of the proposed method and epidemic routing limited were almost equal. However, as the replica number is increased, the proposed method outpasses epidemic routing limited in terms of achieving a lower latency. There was a 5.0%, 9.7%, and 10.1% decrease in message delivery latency using the proposed method as compared to epidemic routing limited with replica numbers 3, 5, and 7, respectively. Fig. 12 also compares the performance of the proposed method and epidemic routing limited in terms of message delivery latency by varying the buffer size allocated to each node. The figure clearly shows that even with varying buffer sizes, the proposed method achieves a better performance in message delivery latency compared to epidemic routing limited. There was a 6.0%, 8.5%, 9.7%, and 4.6% decrease in message delivery latency using the proposed method as compared to epidemic routing limited with buffer sizes 100 KB, 300 KB, 500 KB, and 700 KB, respectively. The decrease in message delivery latencies is due to the message buffer overflow that occurs when epidemic routing limited is used.

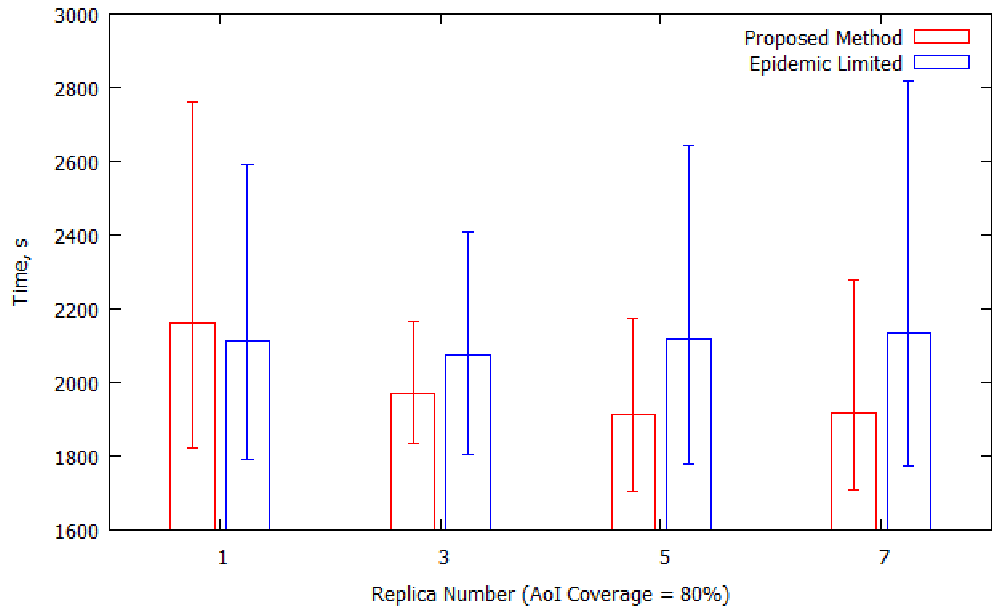


Figure 11: Comparison of Message Delivery Latency with Epidemic Routing vs. Replica Number (Buffer Size = 500 KB, No. of Nodes = 1000)

2.4.7 Buffer Overflow

In our simulation, a 500 KB buffer is allocated to each node and when the buffer space is consumed, old messages are removed to make buffer space for new messages. As shown in Fig. 13, buffer overflow happens immediately during the first 500 s of simulation time after the warming period. However, this does not occur using the proposed method. The figure also shows that as the number of mobile nodes present in A_d increases, the amount of buffer overflow increases at a rapid pace. Even if this did not cause a big impact on the result in Fig. 10, it is expected that when the number of nodes or as the network gets more congested due to the increase in message size, there will be a rapid growth in message delivery latency using epidemic routing.

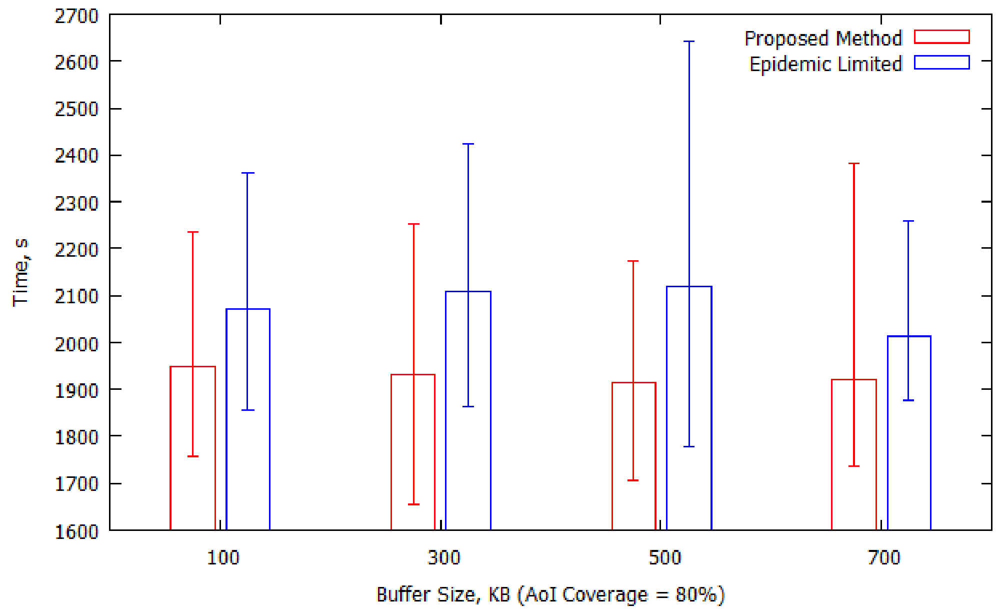
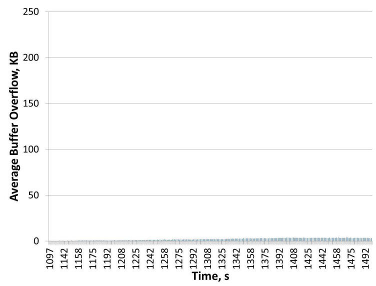


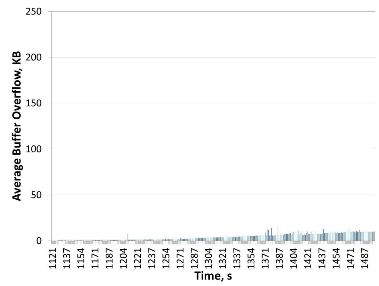
Figure 12: Comparison of Message Delivery Latency with Epidemic Routing vs. Buffer Size (No. of Nodes = 1000, Replica = 5)

2.4.8 Cumulated Distribution Function

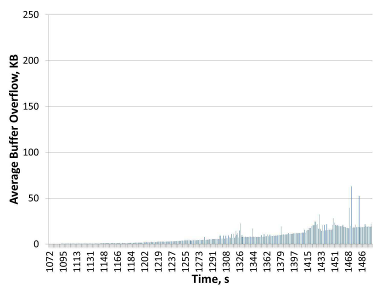
Moreover, cumulated distribution function (CDF) graphs of the time that it takes for the messages to reach its destination using the proposed method and epidemic routing are plotted. The CDF graph of message delivery latency with 1000 mobile nodes is shown in Fig. 14. A CDF graph shows the percentage of data falling between two points. In the said figure, the CDF graph shows the number of runs that attain 80% coverage of the AoI within a specified period of time. Based on Fig. 14, from $t = 0$ to 2000, 70.0% of all runs achieved 80% coverage of the AoI using the proposed method while 50.0% of all runs achieved 80% coverage of the AoI using epidemic routing from $t = 0$ to 2000. This is a 28.6% decrease in the number of runs using the proposed method between $t = 0$ to 2000 showing the effectiveness of using the proposed method to decrease message delivery latency in collecting disaster information.



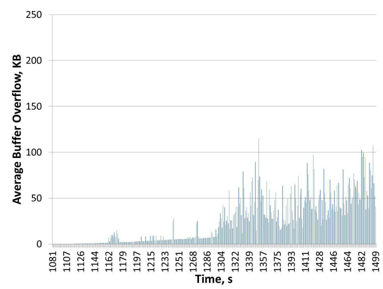
((a)) 200 nodes



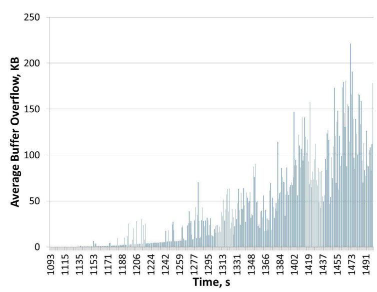
((b)) 400 nodes



((c)) 600 nodes



((d)) 800 nodes



((e)) 1000 nodes

Figure 13: Average Buffer Flow using Epidemic Routing (Buffer Size = 500 KB, No. of Nodes = 1000, Replica = 3)

2.5 Conclusion

The absence of communication infrastructure is common in disaster areas. To provide sufficient information regarding the affected area, a DTN is implemented using the participatory sensing framework. Even though the network is disruption tolerant, delay must still be minimized. Thus, a DTN-based data aggregation algorithm is proposed to minimize this delay. Information are collected, aggregated, and sent through DTN by users present in the affected area. A query sender in

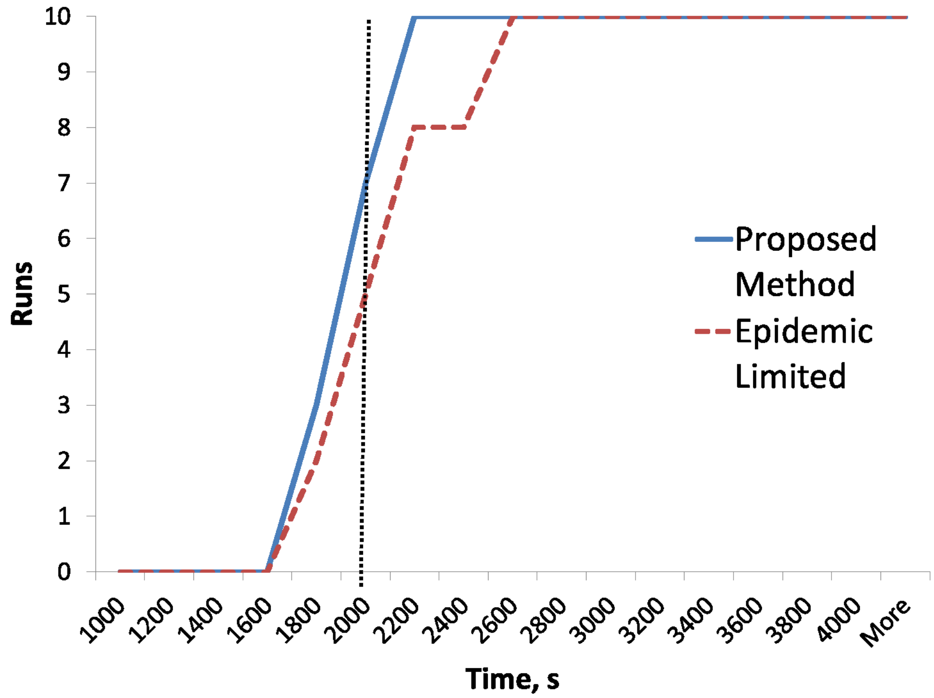


Figure 14: Comparison of Message Delivery Latency CDF with Epidemic Routing (Buffer Size = 500 KB, No. of Nodes = 1000, Replica = 5)

a known location issues a query on an *AoI* and the time when the aggregated messages covering 80% of the *AoI* reaches the query sender is evaluated. Results show that the proposed method achieved a lower message delivery latency and a higher percentage of runs that attained 80% *AoI* coverage than epidemic routing limited proving that the proposed method is indeed effective in decreasing delay for message delivery in disaster areas. Moreover, using the proposed method, the maximum delay with the lowest number of mobile nodes was 4316 s (approximately 1 hr and 12 mins), which is within the critical limit in the aftermath of the disaster (first 36 - 48 hours and the first 2 hours if people are wounded or trapped in buildings).

3. Fast Collection of Images in Disaster Areas

3.1 Introduction

In the aftermath of the disaster, situational awareness over the disaster area is at its lowest point but information about the affected area is important. One way to achieve situational awareness is by collecting photograph-based information. Collecting and sharing geotagged photos is more informative than regular text messages because of its vivid portrayal of unfamiliar objects. However, instant communication of images is unrealistic in an infrastructureless environment such as a disaster area. Thus, a feasible solution is to deploy a delay tolerant mobile ad-hoc network to provide a means for communication.

In this chapter, we describe the use of a content-based prioritization technique so that images collected from an area of interest (AoI) within the disaster area that have critical content are received in a timely manner by the intended image recipients (or sinks). The sinks may be the fire department or on-site command posts. People with mobile phones within or near the vicinity of the disaster zone serve as nodes for the DTN. They gather disaster-related images or photos with location information at various places in the disaster area. They also collect images by exchanging them among other nodes through short-range wireless communication such as Bluetooth or Wi-Fi. The collected images are eventually delivered to the command posts. However, due to the limited data transfer capacity of a DTN, there is a need to implement a prioritization technique to the collected images enabling the command posts to address critical data as soon as possible. In our proposed method, we apply image processing to detect important contents like fires or road blocks on the images and label them as images with critical content to be forwarded over the network prior to other images with noncritical content.

3.2 Image Collection Problem in Disaster Areas

3.2.1 Target Environment

Consider a disaster scenario denoted by A_d with a set of mobile nodes denoted by U and a mobility as described in Section 2.2.1. Just like in the previous chapter, a stationary node at a special spot (e.g., command post) can send an image collection request about a particular area of interest denoted by $AoI \subseteq A_d$. An image collection request q denoted by $\langle u_0, t_0, AoI \rangle$, where u_0 is the sink, t_0 is the time when the request is issued, and AoI is the area of interest from which images are to be collected. The collected images are sent towards the sink by different users who pass through A_d.

Each node receiving the request and existing in the AoI takes a picture p and the image is categorized as either *critical* or *noncritical* based on its content. Images with critical data have to be sent with higher priority than images with noncritical data. Nodes exchange pictures with critical content first to other nodes upon contact for them to be delivered to the sink with minimal delay.

3.2.2 Assumptions

Each user in U is assumed to have a certain role in the disaster area as described in Section 2.2.2. Each node $u \in U$ is also assumed to be carrying a mobile phone, in which the proposed image collection method and DTN protocols are installed in advance. The mobile application is automatically executed when the phone detects a disaster or receives an image collection request. Then, each node $u \in U$ takes pictures at different locations, wherein its location at time t is denoted by $u.pos(t)$ and determined either through GPS or estimated based on some other means. Each node has a limited storage for the collected images. Each node is capable of only short-range wireless communication (3G/4G communication is not available) as described in Section 2.2.2 with the same energy saving technique, i.e., the active-sleep transitions of the node.

3.2.3 Problem Definition

Given an image collection request specifying the AoI, our problem is to determine the order of image transmission taken by each node in U that collects and delivers the set of images with critical content taken in the AoI to the sink u_0 in minimal time.

3.3 Content-Based Image Prioritization Method

In this section, a content-based image prioritization technique is presented to solve the problem described in Section 3.2.3. The proposed algorithm is executed at each node $u_i \in U$ independently of the other nodes and determines the order of image transmission of node u_i over time. Each node runs the algorithm when it receives the request containing u_0, t_0, and AoI. Moreover, each node has an electric map of the target disaster area A_d and its time period P (e.g., 10 s) for the duty cycle is set, with an active interval (e.g., 1 s) and a sleep interval (e.g., 9 s). When a request is received, a node sends a beacon message to find its neighbor nodes during its active interval. However, during its sleep interval, it takes pictures of the AoI, performs image processing to detect disaster events in the picture, categorizes pictures into images with critical or noncritical content, exchanges images with its neighbor nodes or just sleeps if there are no neighbor nodes.

The following subsections explain in detail the different parts of our proposed algorithm: disaster event detection, image prioritization, image exchange, and image collection.

3.3.1 Fire Event Detection

In this study, we focused on detecting *fires*. Fires are important to be detected and extinguished immediately because they grow and spread rapidly leading to serious damages, injuries, or losses. The proposed image processing was done using the Open Source Computer Vision Library (OpenCV), an open source library with optimized algorithms for image processing.

A statistical fire color model described in [6] is used to detect if the image content is depicting an ongoing fire. Each pixel in the image is analyzed and its RGB value is determined. Based on the RGB value, the image pixel is categorized as a fire pixel or not. The image pixel is labeled as a fire pixel if the following conditions are met:

$$R(x, y) > R_{mean} \tag{10}$$

$$R_{mean} = \frac{1}{K} \sum_{i=1}^{K} R(x_i, y_i) \tag{11}$$

$$R(x,y) > G(x,y) > B(x,y) \tag{12}$$

$$0.25 \leq \frac{G(x,y)}{R(x,y)+1} \leq 0.65 \tag{13}$$

$$0.05 \leq \frac{B(x,y)}{R(x,y)+1} \leq 0.45 \tag{14}$$

$$0.20 \leq \frac{B(x,y)}{G(x,y)+1} \leq 0.60 \tag{15}$$

where $R(x, y)$, $G(x, y)$, and $B(x, y)$ are the red, green, and blue values (ranging from 0 to 255) of a pixel located spatially at (x, y) in the image, K is the total number of pixels in the image, and R_{mean} is the mean of red channel values of the pixels. The following condition must hold to determine if fire is detected in the image content:

$$F_d = \begin{cases} \text{fire is detected} & f_p \geq f_{th} \\ \text{fire is not detected} & \text{otherwise} \end{cases} \tag{16}$$

where f_p is the ratio of the total number of fire pixels to the total number of pixels in the image and f_{th} is the threshold parameter to determine the presence of fire in the image content. The threshold parameter f_{th} was statistically determined based on initial experiments.

3.3.2 Road Block Detection

In this study, we also focused on detecting *road blocks*. Road blocks are important because damaged infrastructures cause traffic jams. For example, collapsed buildings on the road lead to delay in rescue activities by first responders. The proposed image processing was also done using OpenCV as described in Section 3.3.1.

The original image (Fig. 15(a)) is converted to a grayscale image (Fig. 15(b)). Using the grayscale image as input, morphological dilation is applied to enlarge the road or lane regions and a top hat transform is performed to extract elongated objects whiter than their background. The resulting image is then converted to a binary image (Fig. 15(c)), which shows the edges of the road regions. To detect road blocks, a probabilistic Hough transform is performed on the binary image to determine the lines belonging to the road. The intersection points of the lines (red dots in Fig. 15(d)) are then determined and based on their locations, a *closeness* measure among the intersection points is calculated. If the intersection points are closer to each other or the closeness value is lower then, the road is clear and if otherwise, there must be a road block in the image. The following condition must hold to determine if a road block is detected in the image content:

$$B_d = \begin{cases} \text{road block is detected} & c_m \geq c_{th} \\ \text{road block is not detected} & \text{otherwise} \end{cases} \tag{17}$$

where c_m is the closeness measure and c_{th} is the closeness threshold parameter to determine the presence of a road block in the image content. The threshold parameter c_{th} was also statistically determined based on initial experiments.

3.3.3 Image Prioritization

Image processing is applied to the collected images from the *AoI* to determine if the images are depicting critical contents. If either a fire or a road block is detected in the image content based on the algorithms described in the previous section, the image is labeled as critical and will be given a higher priority during data exchange between neighbors.

3.3.4 Image Exchange

As mentioned, this study uses the forwarding method based on the node's expected reaching time (*ert*) to the destination calculated using Djikstra's shortest path algorithm. The node with the lowest *ert* receives the stored images with critical content as described in Section 2.3.3. Also, during a contact, only images with high priority are sent to the neighbor nodes and images with low priority are only sent when there are no high priority images existing in the buffer.

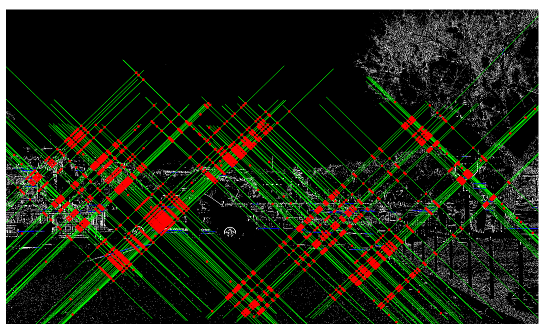
3.3.5 Image Collection

To provide a global view of the disaster to victims, first responders, and those outside the affected area, a situational awareness map is created as shown in Fig.

(a) Original Image

(b) Grayscale Image

(c) Binary Image



(d) Hough Lines Image

Figure 15: Image Flow of the Road Block Detection Algorithm

16 from the gathered images (red dots in Fig. 16) received by the sink, wherein these images were taken in the *AoI* (green circle in Fig. 16)

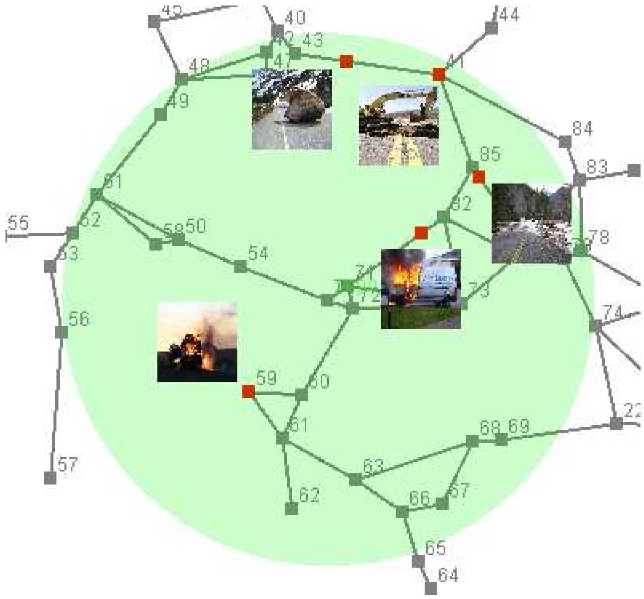


Figure 16: Situational Awareness Map Example

3.4 Performance Evaluation

In order to evaluate the performance of the proposed algorithm, two measures are determined: event detection rate and image delivery latency.

3.4.1 Event Detection Rate

The event detection rate measures the correctness of the used image processing algorithm to determine the event depicted in the image. To determine the event detection rate of the proposed algorithm, 20 sample images for each scenario[4] were collected. The following scenarios are: (1) images with road blocks, (2) images with clear roads, (3) images with fire, and (4) disaster-related images without fires. The event detection rate is related to the number of detected images by the proposed algorithm with the correct events to the total number of images in each scenario. Moreover, the average image processing times for the proposed image processing algorithms were determined on a 32-bit Windows 7 machine with 4 GB RAM, and an Intel Core i7 CPU operating at 3.40 GHz. An average

[4]The sample images used were labeled for reuse by Google Image Search. Some images were also taken from the photo collection of the Federal Emergency Management Agency (FEMA).

time of 978 ms was calculated for the road block detection algorithm based on the images in Scenario 1 while an average time of 1878 ms was calculated for the fire detection algorithm based on the images in Scenario 3. This shows that the proposed image processing algorithm is a rather light application and therefore can be executed in a mobile phone.

3.4.2 Road Block Detection Rate

The road block detection algorithm was performed on the images for scenarios (1) and (2). Figs. 17 and 18 show the detected and undetected images with road blocks for the images in scenarios 1 and 2, respectively. Based on Table 3, the road block detection algorithm was able to detect 80% of the images with road blocks and 85% of the images with clear roads from the sample images resulting in a false positive rate of 15% and a false negative rate of 20% for road blocks.

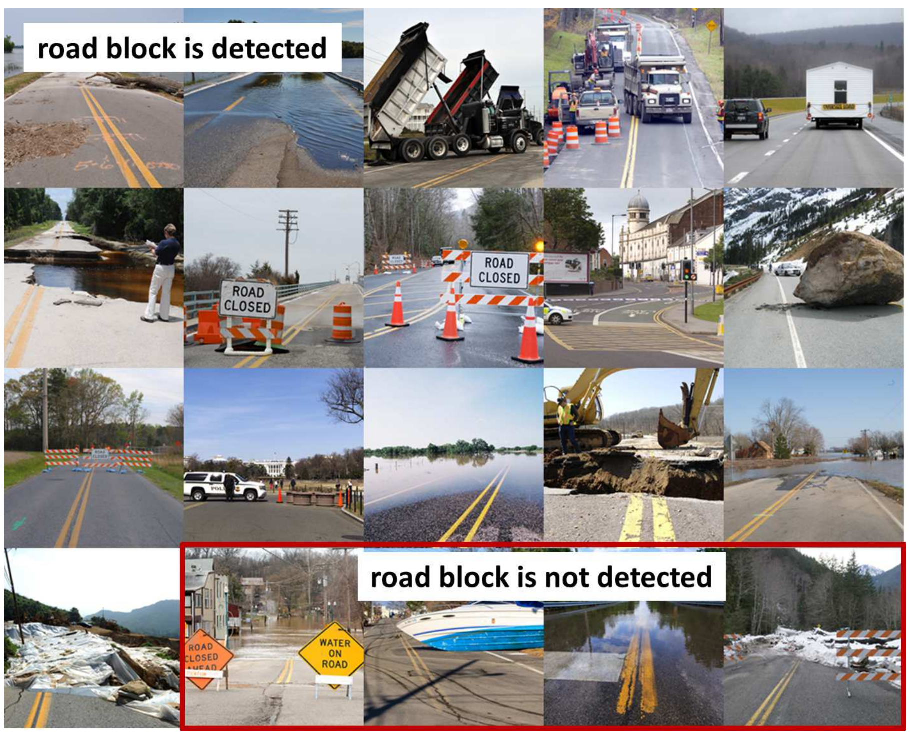


Figure 17: Scenario 1 - Images with Road Blocks

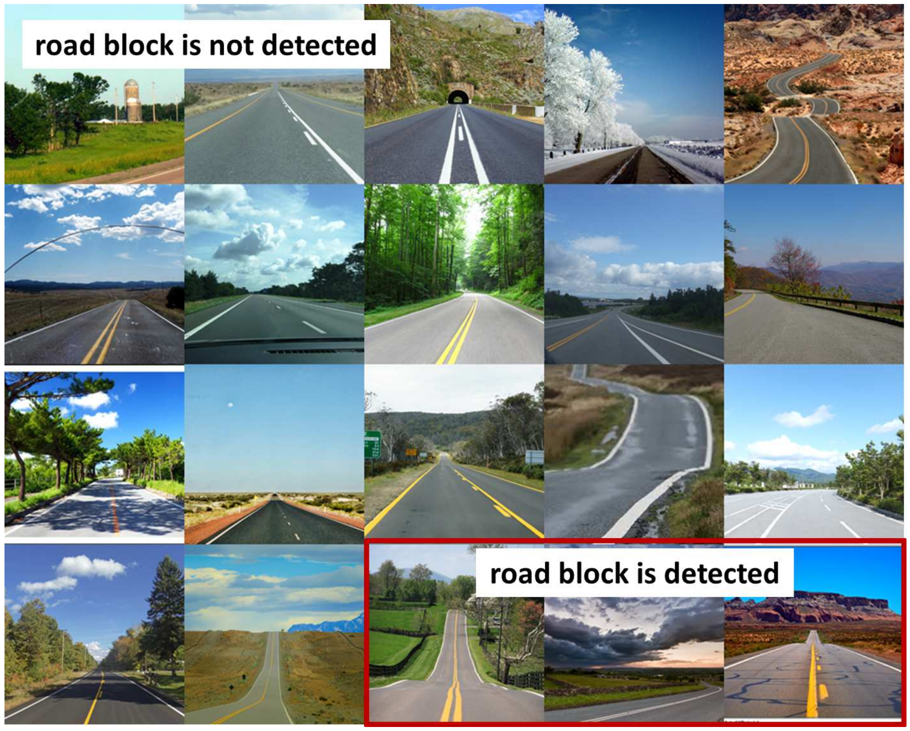


Figure 18: Scenario 2 - Images with Clear Roads

3.4.3 Fire Detection Rate

The fire detection algorithm was performed on the images for scenarios (3) and (4). Figs. 19 and 20 show the detected and undetected images with fire for the images in scenarios 3 and 4, respectively. Based on Table 3, the fire detection algorithm was able to detect 95% of the images with fires and 80% of the disaster-related images without fires from the sample images resulting in a false positive rate of 20% and a false negative rate of 5% for fires.

Table 3: Event Detection Rates

Algorithm	Scenario	Detection Rate
Road Block Detection	Scenario 1	80% (False Negative: 20%)
	Scenario 2	85% (False Positive: 15%)
Fire Detection	Scenario 3	95% (False Negative: 5%)
	Scenario 4	80% (False Positive: 20%)

Figure 19: Scenario 3 - Images with Fires

3.4.4 Image Delivery Latency

The image delivery latency of the proposed content-based image prioritization technique is determined to prove the effectiveness of using DTN coupled with an image prioritization technique in decreasing the delivery latency of images with critical content.

3.4.5 Simulation Configuration

In this study, the simulation configuration is similar to Section 2.4.1 with some minor changes discussed in this section. Images were generated with locations selected randomly inside the A_d. One image collection request was issued at the start of the simulation wherein the location of the command post (or sink) was randomly selected among the locations of the special spots within the disaster area A_d. A simulation warm up time was set wherein during this time, the nodes move according to its mobility model without performing any action. All nodes

Figure 20: Scenario 4 - Disaster-Related Images without Fires

also have the same buffer size and transmission range. A total of 5 runs were taken with a total simulation time of 10000 s for each setting. Table 4 shows a summary of the default values used in the simulation aside from those mentioned in Table 2. In order to facilitate a congested scenario in the network, the network bandwidth was set to 5 Mbps, the buffer size to 1 GB, and the image size to be randomly selected between 100-4000 KB, which is a typical size of an image taken using a mobile phone.

3.4.6 Comparison

The proposed method with image prioritization was compared to the proposed method without image prioritization. The average image delivery latencies of varying image density are plotted with their corresponding variances. Figure 21 compares the performance of using the prioritization technique and not using the prioritization technique in terms of the image percentage received by the sink. The figure shows that with the prioritization technique, images with critical con-

Table 4: Simulation Parameters

Parameter	Default Value
Network	
Bandwidth	5 Mbps
Buffer size	1 GB
Map	
No. of images within A_d	100–1000
No. of images with critical content	50–500
No. of queries	1
Node	
Critical event detection rate	100%
Image size	100–4000 KB
Total simulation time	10000 s

tent are delivered faster over the network achieving a higher percentage of images received. With 100 images, the proposed method with prioritization achieved a 100% in terms of the percentage of images received as compared to 91% without prioritization at $t = 7500\ s$. The proposed method with prioritization also achieved a 100% at an average simulation time of 5102 s, which clearly shows the decrease in delivery latency of images with critical content. Also, even with the increase in the number of images in the AoI, there is still a significant difference in terms of the received image percentage of those images with critical content with a 9.9%, 4.3%, and 10.1% increase when the number of images in the AoI are 100, 500, and 1000, respectively. This shows the effectiveness of the prioritization technique in terms of decreasing the delivery latency of images with critical content.

3.5 Conclusion

In times of disasters, communication networks have difficulty operating because of possible damage. However, situational awareness of the affected area is critical for an effective response. Thus, this study proposes the use of a participatory sensing framework to gather photos of the disaster area and sending the photos

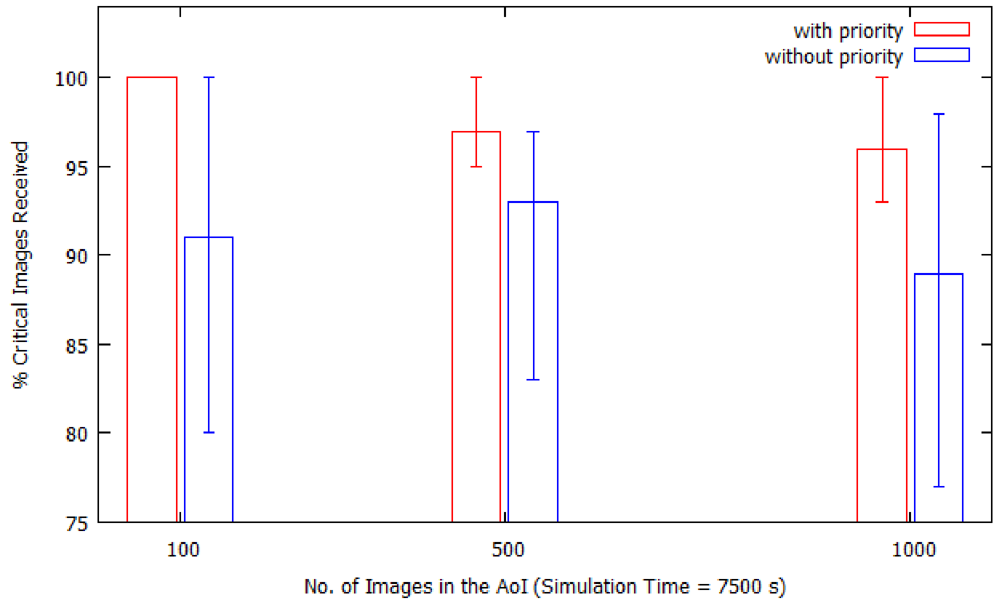


Figure 21: Effect of Image Density on Delivery Latency

over a delay tolerant network. Even though the network is delay tolerant, delay must still be minimized thus, a content-based image prioritization technique is proposed to transmit images with critical content over the network faster. Using image processing to detect critical events like fires and road blocks, the proposed algorithm achieved a detection rate of 80% and 95% for road blocks and fires, respectively. Moreover, results also show that using the prioritization technique achieved a smaller delay in the delivery of images with critical content.

4. Related Work

4.1 DTN Routing Protocols in Disaster Scenarios

In general, DTN routing protocols are classified in two categories: flooding-based protocol and forwarding-based protocol [17]. In flooding-based protocols, each node sends its messages to the nodes that it comes into contact. A modified Spray and Wait flooding-based protocol known as DTN message priority routing protocol was used in the delivery of messages in a disaster-stricken area [14]. A cluster mobility model designed for a disaster scenario was also evaluated using flooding-based routing protocols [22]. However, in flooding-based protocols, there is a possibility that a node has a copy of all the generated information exchanged in the network adding to the current network traffic. Perhaps, a forwarding-based protocol, such as a geographic routing protocol, is more appropriate to use in a disaster scenario. One such protocol is the Location-aware Message Delivery (LMD) protocol that forwards a single copy of a message to its neighbors based on their forwarding benefit, which is defined as the reduced amount of distance of the message from the destination caused by the forwarding [12]. The mentioned DTN studies focus more on achieving a high message delivery ratio than on minimizing delivery latency. Although a DTN is expected to incur a certain delay in sending the information, there is still a need to minimize the delivery latency because the survival rate of people in need decreases over time. In a recent survey evaluating DTN forwarding methods in a disaster setting including delivery latency, MaxProp forwarding is the best method in terms of delivery performance in almost all scenarios [19]. However, this forwarding method uses information on the future contacts of the nodes, which may not be available under disaster situations. Thus, this study uses a forwarding method based on a node's expected reaching time (ert) to the destination instead.

4.2 Data Aggregation in a DTN

One of the major causes of delay in a DTN is its limited data capacity and this is affected more greatly in emergency situations because of the increase in network traffic. One particular study examined Twitter usage during the 2011 South

East Queensland floods. Their report states that 1100 tweets were created per hour during the onset of the floods [4]. It is then critical to address the issue on network congestion especially in a DTN and this can be achieved through data aggregation. Data aggregation has been used in wireless sensor networks to reduce a large amount of raw sensor data to a small number of messages [20]. Also, aggregating data at various nodes results in the reduction of the amount of bits transmitted over the network. Hence, delay can be shortened by reducing the demand for wireless resources and expediting data transmissions [15]. Thus, in this paper, we investigate the use of in-network data aggregation in a disruption tolerant mobile ad-hoc network using a participatory sensing framework for information collection in a disaster area. Specifically, instead of sending all the disaster information or raw data to another node, only aggregates or statistics (summaries) of the collected information need to be sent.

4.3 Data Prioritization in a DTN

Aside from data aggregation, another approach is to implement a prioritization technique for critical data in order to reach emergency responders and key personnel with minimal delay. This was implemented in TRIAGE, a framework that prioritizes data based on user context, message content, and role [18]. However, this framework is limited to first responders that manually add context information with the data and send them to the emergency operations center.

4.4 Image Collection During Disasters

Moreover, with the rise of camera-integrated mobile phones, people in disaster zones often take pictures for documentation but these pictures tend to be repetitive and redundant resulting in network bottlenecks that prevent image delivery to the destination. A novel architecture called CARE (Content-Aware Redundancy Elimination) aims to increase the number of unique images and send these images faster by employing image similarity detection algorithms [25]. A similar study that uses a picture delivery service for camera sensor networks, called PhotoNet, alleviates this problem by assigning priorities to images for forwarding based on its degree of similarity to other images. This results in the delivery of diverse pictures and the reduction of redundant content belonging to the same event [24]. Thus, in this paper, we investigate the use of a content-based image prioritization technique in a disruption tolerant mobile ad-hoc network using a participatory sensing framework for information collection in a disaster area.

4.5 Contribution

In Chapter 2, we describe our method to achieve the minimum delay of data collection in an *AoI* within the disaster area by using data aggregation and employing mobile nodes that are part of a participatory sensing framework. Unlike in previous studies, we implemented an aggregation-based delay tolerant network using the mobile phones of the people with minimal delivery latency for collecting information from an *AoI* in a disaster scenario. Simulations proved that aggregating messages has an effect in reducing message delivery latency. An intensive evaluation of the proposed method was done to prove the efficiency of the proposed algorithm for reducing delay in collecting disaster information.

In Chapter 3, we describe our method to achieve the minimum delay of image collection within the disaster area by using a content-based image prioritization technique and employing mobile nodes that are part of a participatory sensing framework. Unlike previous studies, the proposed method detects critical events in the collected images, prioritizing those with critical contents, and sends them to the command post in the shortest possible time.

5. Conclusion

In this thesis, we studied different methods in collecting information from disaster areas using the participatory sensing framework in a delay tolerant network.

In Chapter 2, a DTN-based data aggregation algorithm is proposed to minimize this information collection time. Information are collected, aggregated, and sent through DTN by users present in the affected area. Results show that the proposed method achieved a lower message delivery latency and a higher percentage of runs that attained 80% *AoI* coverage than epidemic routing limited proving that the proposed method is indeed effective in decreasing delay for message delivery in disaster areas.

In Chapter 3, a content-based image prioritization technique is proposed to transmit images with critical content over the network faster. Using image processing to detect critical events like fires and road blocks, the proposed algorithm achieved a detection rate of 80% and 95% for road blocks and fires, respectively. Moreover, results also show that the prioritization technique achieved a 10.1% increase in received images with critical content when the number of images in the *AoI* is 1000. This shows the effectiveness of using the prioritization technique to achieve a smaller delay in the delivery of images with critical content.

6. Future Work

In this thesis, two approaches were described to achieve situational awareness over the disaster area in the shortest possible time: data aggregation and content-based image prioritization. These two approaches were able to cover a significant part of disaster information collection that is, collecting disaster information via text messages and images. However, there are still other mediums that can be used for disaster information collection such as videos and its collection and transmission via DTN can be tackled in future studies. Other future work includes an improvisation of the proposed data aggregation algorithm by extracting information in overlapping areas of the message coverage areas. Other image processing algorithms can also be added to our proposed content-based prioritization algorithm that will detect other important disaster events depicted in images. Finally, the proposed algorithms should be implemented on a real hand-held device, in which people collect information by creating messages or taking pictures from an area. This will further prove that the proposed algorithm can be implemented in a real environment.

The use of participatory sensing in this study has great potential in the disaster research field especially since there is a shift in paradigm towards a community-based disaster network. During the recent disasters, ordinary citizens played a role in providing information on the damages and needs in the affected area. With the help of ordinary people, basic information of the affected area can be immediately gathered for assessment, shortening the information gap when disaster strikes. This type of network is also useful for the communications/information management feature of the Incident Command System (ICS), an essential component of disaster response. The network for this feature must establish a system in gathering and sharing disaster-related information. However, there are still unsolved problems that need to be addressed for the successful use of such kind of disaster network.

First, even with the numerous disaster management or disaster support applications, most of them have not been implemented in the communities. There is a gap between the scientific community and the implementing agencies (e.g., local government, disaster-related public organizations, etc.). People in the implementing agencies still sometimes use conventional methods even if there are other technologies that are much more efficient in disaster mitigation, rescue, and recovery. People in the scientific community also have relatively few knowledge on the immediate needs of the implementing agencies during the occurrence of a disaster. Moreover, even if disaster support applications have been implemented in the communities, the civilians sometimes do not have knowledge on how to use the system. Thus, a system especially a disaster network can never be effective without an education component. The education component will serve as a bridge between the scientific community, the implementing agencies, and the ordinary citizens bringing about awareness between the different groups for a much more effective disaster network.

Second, in the technical aspect of disaster applications, security, privacy, and device interoperability still remain as open issues. Security and privacy cannot be underestimated since the lives of civilians are at stake and if the data falls into the wrong hands, it can have adverse effects on people's lives. Moreover, in a disaster response network, information from different sources is collected and one or more communication channels are used to send information from a central authority to the intended recipients and vice-versa. Thus, there needs to be a seamless integration of technologies (e.g., navigation, traffic system, public communication media, etc.) for a more effective disaster communication network.

Lastly, countries with small economies still cannot afford to have their own full-fledged disaster communication network that may consist of unmanned aviation vehicles, satellite mobile network, and the like thus, there is a need to explore affordable ways to implement an efficient disaster communication network. One such medium is the use of participatory sensing and a delay tolerant network for a disaster communication network. However, there is not one medium that is suitable for every disaster situation. Thus, a combination of technology is suggested. It is not a competition between different technologies but using the best combination depending upon the circumstances. Perhaps, a better combination for a disaster communication network can be the subject of future studies particularly, tapping the most readily available medium, which are the people.

Acknowledgements

First and foremost, I would like to thank God for His continued guidance and strength. For without Him, I would not be where I am now.

I would also like to thank Professor Minoru Ito for giving me a chance to be a part of the Foundations of Software laboratory. Without his support, I would not be able to pursue my graduate studies in Japan.

I deeply thank Professor Keiichi Yasumoto for his constant support and encouragement throughout my doctoral studies. Thank you for imparting on me a great amount of knowledge and advice especially in doing research work. I am truly grateful.

I am also deeply grateful to Professor Suguru Yamaguchi for his insightful comments.

I would also like to thank Associate Professor Naoki Shibata and Assistant Professor Weihua Sun for their support and valuable technical discussions.

I would also like to thank Assistant Professor Yukiko Yamauchi for her help during my first year as a PhD student.

I would also like to thank all current and previous members of the Foundations of Software laboratory.

I would also like to thank the administration of Nara Institute of Science and Technology for selecting me as one of the NAIST International Scholars and it is through this scholarship that I was able to fulfill my dream of pursuing a doctoral degree in Japan.

I would also like to thank my fellow International students at NAIST, especially the Filipino community, for it is with them I share the ups and downs of life as a graduate student.

Finally, I would like to thank my parents and the people close to my heart for being patient with me and just being present when I need them.

To everyone who has been there for me, thank you.

References

[1] Nabil R. Adam, Basit Shafiq, and Robin Staffin. Spatial computing and social media in the context of disaster management. *IEEE Intell. Syst.*, 27(6):90–96, 2012.

[2] Asaad Ahmed, Keiichi Yasumoto, Yukiko Yamauchi, and Minoru Ito. Probabilistic coverage methods in people-centric sensing. *J. Inform. Process.*, 19:473–490, 2011.

[3] Ali S Al-Ghamdi. Emergency medical service rescue times in riyadh. *Accident Anal. Prev.*, 34(4):499–505, 2002.

[4] Axel Bruns, Jean E Burgess, Kate Crawford, and Frances Shaw. # qldfloods and@ qpsmedia: Crisis communication on twitter in the 2011 south east queensland floods. *ARC Centre of Excellence for Creative Industries and Innovation, Queensland University of Technology*, 2012.

[5] Angela Sara Cacciapuoti, Francesco Calabrese, Marcello Caleffi, Giusy Di Lorenzo, and Luigi Paura. Human-mobility enabled wireless networks for emergency communications during special events. *Pervasive Mob. Comput.*, 9(4):472–483, 2013.

[6] Turgay Celik, Hasan Demirel, Huseyin Ozkaramanli, and Mustafa Uyguroglu. Fire detection using statistical color model in video sequences. *J. Vis. Commun. Image R.*, 18(2):176–185, 2007.

[7] Jovilyn Therese B. Fajardo and Carlos M. Oppus. A mobile disaster management system using the android technology. *WSEAS T. Commun.*, 9(6):343–353, 2010.

[8] Kevin Fall, Gianluca Iannaccone, Jayanthkumar Kannan, Fernando Silveira, and Nina Taft. A disruption-tolerant architecture for secure and efficient disaster response communications. In *Proc. ISCRAM*, 2010.

[9] Akihiro Fujihara and Hiroyoshi Miwa. Real-time disaster evacuation guidance using opportunistic communications. In *Proc. 2012 IEEE/IPSJ 12th Int. Symp. on Applications and the Internet (SAINT)*, pages 326–331, 2012.

[10] Akihiro Fujihara and Hiroyoshi Miwa. Homesick levy walk and optimal forwarding criterion of utility-based routing under sequential encounters. In Nik Bessis, Fatos Xhafa, Dora Varvarigou, Richard Hill, and Maozhen Li, editors, *Internet of Things and Inter-cooperative Computational Technologies for Collective Intelligence*, volume 460 of *Studies in Computational Intelligence*, pages 207–231. Springer, 2013.

[11] Gokce Gorbil and Erol Gelenbe. Disruption tolerant communications for large scale emergency evacuation. In *Proc. 11th IEEE Int. Conf. on Pervasive Computing and Communications (PERCOM 2013)*, pages 18–22, 2013.

[12] Md. Nurul Huda, Farzana Yasmeen, SShigeki Yamada, and Noboru Sonehara. An approach for short message resilience in disaster-stricken areas. In *Proc. 2012 Int. Conf. on Information Networking (ICOIN)*, pages 120–125, 2012.

[13] Yasuhiro Ishimaru, Weihua Sun, Keiichi Yasumoto, and Minoru Ito. Dtn-based delivery of word-of-mouth information with priority and deadline. In *Proc. 5th Int. Conf. on Mobile Computing and Ubiquitous Networking (ICMU 2010)*, pages 179–185, 2010.

[14] Inwhee Joe and Sang-Bo Kim. A message priority routing protocol for delay tolerant networks (dtn) in disaster areas. In Tai-hoon Kim, Young-hoon Lee, Byeong-Ho Kang, and Dominik Slezak, editors, *Future Generation Information Technology*, volume 6485 of *Lecture Notes in Computer Science*, pages 727–737. Springer Berlin / Heidelberg, 2010.

[15] Changhee Joo, Jin-Ghoo Choi, and Ness B. Shroff. Delay performance of scheduling with data aggregation in wireless sensor networks. In *Proc. 2010 IEEE Conf. on Computer Communications (INFOCOM)*, pages 1–9, 2010.

[16] Salil S. Kanhere. Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces. In *Proc. 2011 IEEE 12th Int. Conf. on Mobile Data Management (MDM)*, volume 2, pages 3–6, 2011.

[17] Mengjuan Liu, Yan Yang, and Zhiguang Qin. A survey of routing protocols and simulations in delay-tolerant networks. In Yu Cheng, Do Eun, Zhiguang

Qin, Min Song, and Kai Xing, editors, *Wireless Algorithms, Systems, and Applications*, volume 6843 of *Lecture Notes in Computer Science*, pages 243–253. Springer Berlin / Heidelberg, 2011.

[18] Faisal Luqman, Feng-Tso Sun, Heng-Tze Cheng, Senaka Buthpitiya, and Martin Griss. Prioritizing data in emergency response based on context, message content and role. In *Proc. 1st Int. Conf. on Wireless Technologies for Humanitarian Relief*, pages 63–69, 2011.

[19] Abraham Martn-Campillo, Jon Crowcroft, Eiko Yoneki, and Ramon Mart. Evaluating opportunistic networks in disaster scenarios. *J. Netw. Comput. Appl.*, 36(2):870–880, 2013.

[20] A.S. Peng, D.M. Moen, J.A. Spinks, L.M. Meredith, Tian He, and D.J. Lilja. Reliable data aggregation and dissemination framework in tactical network architecture. In *Proc. 2010 Military Communications Conf. (MILCOM 2010)*, pages 569–574, 2010.

[21] Rajib Kumar Rana, Chun Tung Chou, Salil S. Kanhere, Nirupama Bulusu, and Wen Hu. Ear-phone: an end-to-end participatory urban noise mapping system. In *Proc. 9th ACM/IEEE Int. Conf. on Information Processing in Sensor Networks (IPSN '10)*, pages 105–116, 2010.

[22] Sujoy Saha, Sushovan, Anirudh Sheldekar, Rijo Joseph C., Amartya Mukherjee, and Subrata Nandi. Post disaster management using delay tolerant network. In Abdulkadir zcan, Jan Zizka, and Dhinaharan Nagamalai, editors, *Recent Trends in Wireless and Mobile Networks*, volume 162 of *Communications in Computer and Information Science*, pages 170–184. Springer Berlin Heidelberg, 2011.

[23] Sasu Tarkoma, Christian Esteve Rothenberg, and Eemil Lagerspetz. Theory and practice of bloom filters for distributed systems. *IEEE Commun. Surv. Tut.*, 14(1):131–155, 2012.

[24] Md Yusuf Sarwar Uddin, Hongyan Wang, Fatemeh Saremi, Guo-Jun Qi, Tarek Abdelzaher, and Thomas Huang. Photonet: a similarity-aware picture

delivery service for situation awareness. In *Proc. 2011 IEEE 32nd Real-Time Systems Symposium (RTSS)*, pages 317–326, 2011.

[25] Udi Weinsberg, Qingxi Li, Nina Taft, Athula Balachandran, Vyas Sekar, Gianluca Iannaccone, and Srinivasan Seshan. Care: content aware redundancy elimination for challenged networks. In *Proc. 11th ACM Workshop on Hot Topics in Networks*, pages 127–132, 2012.

[26] Jiazhen Zhou, Jiang Li, and Legand Burge. Efficient scheduling of pigeons for a constrained delay tolerant application. *EURASIP J. Wirel. Commun.*, 2010:1–7, April 2010.

List of Major Publications

Journal Papers

1. Fajardo, J.T.B., Yasumoto, K., Shibata, N., Sun, W., and Ito M.: "Disaster Information Collection with Opportunistic Communication and Message Aggregation," to appear in the *Special Issue on Network Services and Distributed Processing, Journal of Information Processing (IPSJ)*, Vol. 22, No. 2, (April 2014). (corresponding to Chap. 2)

International Conference

1. Fajardo, J.T.B., Yasumoto, K., Shibata, N., Sun, W., and Ito M.: "DTN-Based Data Aggregation for Timely Information Collection in Disaster Areas," *Proc. 2012 IEEE 8th Int. Conf. on Wireless and Mobile Computing, Networking, and Communications (WiMob)*, pp. 333–340, (Oct. 2012). (corresponding to Chap. 2)

2. Fajardo, J.T.B., Yasumoto, K., and Ito M.: "Content-Based Data Prioritization for Fast Disaster Images Collection in Delay Tolerant Network," to appear in the *7th Int. Conf. on Mobile Computing and Ubiquitous Networking (ICMU 2014)*, (Jan. 2014). (corresponding to Chap. 3)

Other Publications

Domestic Conference

1. Fajardo, J.T.B., Yasumoto, K., Shibata, N., Sun, W., and Ito M.: "Aggregation-Based Information Collection in Disaster Areas," *Proc. 2012 Multimedia, Distributed, Cooperative, and Mobile Symposium (DICOMO 2012)*, pp. 1674–1685 (Jul. 2012).

2. Fajardo, J.T.B., Yasumoto, K., Shibata, N., Sun, W., and Ito M.: "Timely In-Network Data Aggregation in Disaster Areas," *IPSJ SIG Technical Reports*, 2011-MBL-59, pp. 1–8 (Aug. 2011).