M.Sc. Engg. Thesis

Availability-aware Content Replication in Disaster Response Networks

by
Moin Mostakim

Submitted to

Department of Computer Science and Engineering

in partial fulfilment of the requirements for the degree of
Master of Science in Computer Science and Engineering

Department of Computer Science and Engineering

Bangladesh University of Engineering and Technology (BUET)

Dhaka 1000

August 2016

Dedicated to my loving parents

Author's Contact

Moin Mostakim

Department of Computer Science & Engineering

Bangladesh University of Engineering & Technology (BUET).

Email: shisshir11@gmail.com

The thesis titled "Availability-aware Content Replication in Disaster Response Networks", submitted by Moin Mostakim, Roll No. **0412052011P**, Session April 2012, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering and approved as to its style and contents. Examination held on August 14, 2016.

Board of Examiners

1. ______________________________

Dr. Md. Yusuf Sarwar Uddin — Chairman (Supervisor)
Associate Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka.

2. ______________________________

Dr. M. Sohel Rahman — Member (Ex-Officio)
Head and Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka.

3. ______________________________

Dr. Mahmuda Naznin — Member
Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka.

4. ______________________________

Dr. A. B. M. Alim Al Islam — Member
Assistant Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology, Dhaka.

5. ______________________________

Dr. Salekul Islam — Member (External)
Head
Department of Computer Science and Engineering
United International University, Dhaka.

Candidate's Declaration

This is hereby declared that the work titled "Availability-aware Content Replication in Disaster Response Networks" is the outcome of research carried out by me under the supervision of Dr. Md. Yusuf Sarwar Uddin, in the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka 1000. It is also declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Moin Mostakim

Candidate

Acknowledgment

I express my heart-felt gratitude to my supervisor, Dr. Md. Yusuf Sarwar Uddin for his constant supervision of this work. He advise me a lot in every aspect of this work and guided me with proper directions whenever I sought one. His patient hearing of my ideas, critical analysis of my observations and detecting flaws (and amending thereby) in my thinking and writing have made this thesis a success.

I would also want to thank the members of my thesis committee for their valuable suggestions. I thank Dr. Mahmuda Naznin, Dr. A. B. M. Alim Al Islam and specially the external member Dr. Salekul Islam.

In this regard, I remain ever grateful to my beloved parents, who always exist as sources of inspiration behind every success of mine I have ever made.

Abstract

In this thesis, we design and develop a content replication technique for replicating contents in disaster response networks. Disaster response networks evolve after a disaster when usual communication (e.g., the Internet) does not work due to damages or power outage. This kind of networks is a form of Delay-Tolerant Networks (DTNs) that leverage people's mobile phones, devices with rescue workers, volunteers and routers at vehicles or buildings to carry and forward messages around. Replicating important contents (such as phone books, maps of the rescue areas and various logistic information) for higher availability is an important issue in this context where popular replication services such as Dropbox and Google Drive, may not be available during that time. The only option prevails is to copy these contents to some other "nearby" nodes who are more "available" than others. We compute the availability of nodes by modeling the contact process among nodes by using a Markov chain and then analytically derive the expected retrieval time when a content object is replicated onto a set of replica nodes. Our objective is to choose this set so as to minimize the expected retrieval time from at least one such replica. We design a corresponding protocol, named availability-aware content replication (ACR) protocol, that provides replication service in DTNs. We perform evaluation of our replication schemes in post-disaster mobility model (PDM) using the Opportunistic Network Environment (ONE) simulator. Our experimental results demonstrate that our proposed schemes perform better than other techniques such as randomized replication.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Natural disasters break down the normal flow of the human activity and the communication services such as phone and the Internet. Disasters, either natural (e.g., hurricanes and earthquakes) or man-made (e.g., nuclear catastrophe) hinder usual communication experiences of people and therefore, call for different approaches to meet communication needs. Communication becomes more essential in terms of search and rescue operation after the disaster. Different kinds of information such as the locations of affected areas, damaged infrastructures, and the number of injured individuals are needed for initiating rescue mission after the disaster. Furthermore, rescue workers and volunteers need comprehensive, reliable and accurate information in order to supply relief materials (e.g., food, water, shelters) and services to the survivors. Moreover, people staying outside the disaster areas also need information about their closed ones [1]. Therefore, effective communication among rescue workers is essential to collect necessary data of the disaster areas to provide required services to the affected people.

1.1 Communication during Disaster

Communication in the disaster-affected area is very difficult due to the lack of regular communication means. Usually power sources are down and landlines communication are interrupted due to tower damage and other technical difficulties. Sometimes multiple disasters strike at the same time or within a short time difference like it happened in Virginia in 2011. People living in that area had to endure destructive tornadoes, the Tropical Storm Lee, Hurricane Irene, and a 5.8 magnitude earthquake at the same time. Owing to infrastructure damages, it became quite difficult for the inhabitants to contact with the corresponding authorities or to communicate with their region emergency numbers.

It is important for public safety agencies such as law enforcement, emergency medical services, and fire services to be able to provide and maintain effective communication before, during, and after a disaster or an emergency. These rescue agencies provide an office along with communication platforms being capable of moving into the disaster-affected area at a very short notice to meet situational needs ensuring systems interoperability through deployment of various means such as Radio equipment, Satellite back-haul (providing voice, data, and video support), Cross-banding, Line of Sight (moving data in and out of a disaster area). These rescue agencies establish a supporting environment for personnel with proper planning; alert, notification, and deployment monitoring; situational awareness and reporting.

People indeed intend to generate and consume information during emergency moments. Numerous web applications have been used in various crisis moments in near history. There are a few web applications such as "Global Disaster Alerts" (http://www.gdacs.org/alerts/), "Quake Aware" (http://www.quakekare.com/) which propagate information about crisis moments and keep updating people about recent phases. It has been observed that people converge to social networking and on-line media sites, such as Twitter, Facebook and Youtube during crisis moments in order to update their latest status and to know whereabouts of others. People try to share recent updates regarding both natural and human made disaster crisis moment to bring help and inform their safety to their peer ones.

It is quite evident that the Internet-based services are very useful during disaster. However, the Internet connectivity may not be available or may only be available intermittently with limited capacity. This limitation can be caused by infrastructure collapse or power-outage in the affected area. Power outage is recorded to be a commonplace consequence during or after the disaster leading to malfunction of usual communication capabilities [2],[3],[4],[5]. It has been observed that in order to cope with anxiety or due to inherent lack of coordination among users, people in the disaster strike area tend to communicate extensively by sending same messages or pictures of their damaged properties repeatedly.

Therefore, when significant elements of communication are down in the disaster-affected area, people with hand-held wireless devices (e.g., cell phones, PDAs, and laptops), houses with WiFi routers and automobiles with radios require to communicate with each other in an effective way so that they can form good functional communication networks such as Delay Tolerant Networks (DTNs).

1.2 Delay Tolerant Networks (DTNs)

The emergence of new technologies in communication field enhance the capability of modern devices for computing and communicating efficiently. These modern devices (e.g., mobile phones, tablets and other digital gadgets) with a variety of data connectivity provisions such as WiFi, 3G, 4G, LTE, Bluetooth and WiMax make communication experiences quite easy and smoothing, even when people are moving freely. These devices can store various contents holding important information that can be shared among others and can be used effectively in the rescue operation in the disaster-affected areas.

In intermittent connection, there is no end-to-end path between source and destination. This phenomena is called network partition. Communication using traditional TCP/IP protocol does not work. In addition to intermittent connectivity, long propagation delays between hosts contribute to end-to-end path delays that can defeat traditional protocols and applications that rely on quick return of acknowledgment data. Therefore, to solve these problems a method called *store-carry-forward* technique is adopted. In this method, whole messages or fragments of such messages, are forwarded from source to destination via a path created by intermediate hosts. These messages traverse this path until it reaches the destination. The storage places (the hosts in the network who store the messages) can hold messages for indefinite time. They are called persistent storages.

In recent years, the networking research community has proposed specific protocols that can cope with the challenges of intermittent connectivity. This technology is referred as Delay and Disruption-Tolerant Networking (www.dtnrg.org). In DTNs, every communication device acts as both transponder and router. DTNs propose usage of store-carry-forward protocols where data packets can be physically carried and stored in the persistent storages for a long period when no communication link exists. Fig. 1.1 illustrates the store-carry-forward protocol.

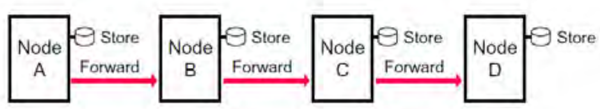


Figure 1.1: Store-carry-forward protocol

For example, if only one person in the disaster-affected area has a mobile device with a data connection, suppose 2G or 3G, others may use this device to relay their messages.

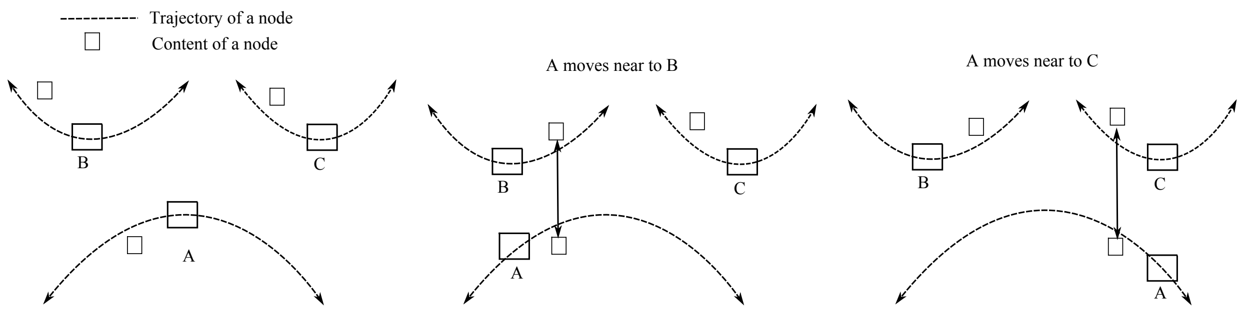
Compared to the Internet, mobile ad hoc networks and WLANs, DTNs have the basic features such as intermittent connection, high delay, low efficiency, high queue delay, limited life time of nodes, limited resources, and heterogeneous connectivity. Mobility and energy of the nodes are limited in DTNs. Therefore, DTNs frequently get disconnected resulting a continuous changes in the topology. The networks keep the status of intermittent connection and partial connection. As a result there is no guarantee to achieve end-to-end route in DTNs. Each hop delay might be very high due to the fact that DTNs intermittent connections keep unreachable in a very long time and thus leading to a lower data rate and showing the asymmetric features in up-down link data rate. In some special circumstances of the restricted networks, each node in the networks can use the battery power in harsh conditions, which will cut the life time of node. When the power is off, the node can not guarantee normal work. It is very possible that the power is off when the message is being transmitted. Therefore functions such as computing and processing abilities, communication abilities, and storage spaces of the nodes in DTNs are weaker compare to that the functions of ordinary computers due to the constraints of price, volume and power. Moreover, the limited storage spaces result in higher packet loss rate in DTNs. DTN is a overlay network for transmission of asynchronous message. It can run on different heterogeneous network protocol stacks and DTN gateway ensures the reliable transmission of interconnection messages.

A disaster response network is one kind of DTN. Where in traditional wireless networks partition are treated as an abnormality, they are quite normal in DTNs; hence end-to-end "always on" connectivity from source to destination never exist. Using redundancy (e.g., replication, coding) and intelligent mobility prediction schemes, data can be transported with single hop or over a sequence of contacts, despite the lack end-to-end communication.

In this thesis, we consider DTNs as primary means of communication in the disaster-affected area.

1.3 Content Replication in DTNs

Many challenging aspects arise in the context of DTN such as resource allocation [6], content placement [7], routing scheme [8] [9], etc. The presence of high dynamic and discontinuous connection between networks made the above mentioned aspects more challenging in DTNs rather than in traditional connected networks. To make the contents more available in the networks, replication is an essential approach.



(a) At initial time T_0 (b) At time T_1 (c) At time T_2, $T_2 > T_1$

Figure 1.2: Content replication process among A, B, and C

Fig. 1.2 illustrates an example of content replication process. Let us consider three nodes A, B, and C placed sparsely in a network. The corresponding trajectories of these nodes are shown in Fig. 1.2(a). At time T_1, node A comes in the communication range of node B. At that time, node A may replicate its contents into node B. Similarly node B can also replicate its content into node A. After certain time, node A moves away from node B and comes near to another node C. Then node A may copy its contents to node C and node C may also replicate its content to node A. Hence, the content of node A can be replicated in both nodes (i.e., B and C). Therefore, the content of node A becomes more available in the network. This will make the network more resilient upon failure of nodes.

Naturally content replication can be done in an arbitrary fashion. Network entities try to make copy of their contents so that they could retrieve them during loss of the original. Replication can be done by gossiping, given that the storage capacity at the network nodes can be considered as unlimited and the content request rate is known in DTNs. However, making populating more contents by frequent replication over the network will flood the network and degrade its performance and efficiency.

To make the DTNs more resilient to failure, important content held in devices (such as a phone book and an evacuation plan map) can be replicated onto a set of "nearby" other nodes opportunistically so that in hour of need one of them will be able to deliver the content back. However, all the existing approaches in the literature focus on local search heuristic mechanisms based on deterministic choices of the next proper candidate or randomized choices for content replication. The performance of these mechanisms are mostly simulation driven as it is hard to develop a suitable analytical solution for them and it is also difficult to make these more generalized because of wide range of mobility scenarios.

Availability of content is ensured by the availability of replicated nodes in the network. Content replication would be considered a success if the replicated content can be collected in hour of need. To make the content more available, the replication should be done in such a manner to the nodes be available over time. So, in time of need the replicated content can be found instantly.

We tried to exploit the recurrent manner of meeting nodes to make a selection choice for replicating contents in DTNs rather then choosing randomly. The nodes meet in recurrent manner is highly available in time of need. So, keeping contents to them would be ensured to be retrieved in hour of need. In disaster response networks, nodes are meeting each other in a recurrent manner. For example, relief workers and volunteers meet the disaster-affected people several times to provide different services. We focused to find such nodes for content replication in the disaster response networks so that the nodes can retrieve the contents in hour of need.

1.4 Contributions of This Thesis

In this thesis, we focus on the problem of selecting suitable neighbors to replicate contents. The contributions of this thesis are as follows:

- We devise an analytical model (e.g., a Markov chain) to entail the nature of meeting process among nodes for computing the availability of nodes.
- We introduce an algorithm for the determination of a replica set of candidate nodes for a given node based on different heuristics approaches that minimizes the expected retrieval time from at least one replica.
- We propose a new protocol entitled "Availability-aware Content Replication Protocol" that selects the nodes for replicating the contents and retrieve the contents minimizing the retrieval time.
- We implement our protocol in the ONE [10] simulator and perform extensive experiments with varied parameters to demonstrate the efficiency of our proposed approaches with other approaches such as random replication approach.

1.5 Organization of the Thesis

The Chapter 2 describes the background and key concepts of our work. Chapter 3 tries to highlight our proposed content replication technique followed by content replication and retrieval protocol stated in Chapter 4. Chapter 5 depicts the simulation scenario and set-up in the ONE simulator [10] for experimental evaluation followed by experimental results. Chapter 6 focuses on the future aspects of this work. Chapter 7 concludes the thesis with closing remarks.

Chapter 2

Background and Preliminaries

We propose a content replication protocol in the post disaster mobility model where after disaster the contents becomes available when it is needed. Due to lack of end-to-end connectivity when the content is in demand it can not be served instantly. Therefore, we focused on a replication technique that minimizes the retrieval time of the content.

We explored variety of recent literatures relevant to our proposal. We briefly discuss works around networking services in DTNs, replication techniques of contents ,content retrieving mechanisms in DTNs followed by some different mobility models for DTNs.

2.1 Networking Services in DTNs

In this section, we focus on various routing protocols of DTNs. Furthermore, we also focus on custody transfer and congestion control mechanisms deployed in DTNs. Finally, we explore diverse applications of DTNs.

2.1.1 Routing Protocol

In computer networks, one fundamental feature is the process of routing data from source to destination. Traditional network protocols implicitly assume that an end-to-end connection exists for any two nodes that exchange data between them. However, the reliable data transmission in DTNs imposes a great challenge on designing routing protocols. Therefore, a new class of protocols is necessary to decide how nodes should transmit messages to other nodes when they meet. We briefly introduce some popular routing protocols for DTNs in this section.

2.1.1.1 Direct Delivery

Jones et. al., proposed a routing protocol entitled Direct Delivery [11]. In this protocol, a source node uses a buffer to store a message. The source sends the message when it connects with a destination node. Hence, only a single copy of the message is present in the network. Besides, the source stores the message in the buffer until the message expires. Since the protocol works when a source node connects with a destination node, it does not consume many network resources. Therefore, this protocol is suitable for applications where the resources impose constraints.

2.1.1.2 Epidemic

Another well known routing protocol for DTNs is Epidemic routing presented in [12]. In epidemic routing, all nodes perform flooding by sending messages to other nodes in the network. The messages are distributed in a random way. Furthermore, the messages are distributed until all the nodes hold them. The protocol relies on transitive distribution of messages through ad hoc networks that ensures the reachability of the messages to their respected destinations. Consequently, each node maintains a buffer for storing not only the messages originated by itself but also the messages received from other nodes. In this protocol, each node maintains an identifier. When two nodes come into communication range of one another, a node with a smaller identifier initiates a session with a node with larger identifier. Moreover, each node maintains a cache of nodes that it has communicated recently to avoid redundant connections.

2.1.1.3 Spray and Wait (SAW)

Spray and Wait protocol proposed in [13, 9] is another well known routing protocol of DTNs. This protocol aims to reduce the number of message transmission. This protocol consists of two phases: Spray phase and Wait phase. In the spray phase, a source node creates the message and sends a defined number of messages to other nodes in the network. However, if the destination node is not found in the spray phase then one of the nodes in the network who received the message performs a direct transmission to the destination node. This protocol combines the speed of Epidemic routing with the simplicity of direct transmission. The spray and wait protocol assumes that a defined number of spread messages in the network is enough to guarantee that at least one will find the destination quickly.

2.1.1.4 PRoPHET

Probabilistic Routing Protocol (PRoPHET) [14] is another popular routing protocol. It makes routing decisions based on encounter probability of nodes in the network. This encounter probability is determined by keeping records of previous encounter of nodes. This protocol assumes that if two nodes in the network meet many times, then the probability of both encountering again is higher. When two nodes meet for the first time, a metric called delivery predictability is initiated. This metric is computed based on the encounter of nodes. PRoPHET protocol utilizes this metric to send messages in the network. Therefore, this protocol utilizes less network bandwidth compared to other state of the art approaches such as Epidemic, Direct delivery that utilizes random messaging. However, the number of copy messages in the network has no limitations. Hence, this protocol consumes more energy compared to other approaches.

Besides routing protocol DTNs also offer other services such as custody transfer and congestion control services.

2.1.2 Custody Transfer Service

A continuous end-to-end connection is not always guaranteed in DTNs. Any point-to-point link may be interrupted at any time. Therefore, DTN protocol does not include an end-to-end confirm mechanism that will initiate prompt correction action at source similar to the conventional TCP protocol. Hence, DTNs introduce a custody transfer service [15, 16]. This service is provided to a content as it is delivered through a DTN. Custody transfer service keeps track of the recent nodes for each content. These nodes are then referred as *custodian.* This custodian is responsible to keep the content safe until another custodian receives it successfully. A Custodian uses a persistent memory for this purpose. When a content is successfully handover from one custodian to another, an acknowledgment is transfered.

2.1.3 Congestion Control Mechanism

Owing to the nature of DTNs congestion control mechanism is relatively difficult compared to other networks. Congestion control mechanism has been explored much less extensively in DTNs, with only a few papers having been published ([17, 18]). If the storage resources of the nodes become scarce due to the presence of too much content a DTN experience congestion. A node experiencing congestion has several options to mitigate the situation such as dropping the expired contents, moving the contents somewhere else, ceasing accepting contents with custody transfer, ceasing accepting regular contents, dropping unexpired contents, and dropping unexpired contents for which the node has custody.

2.1.4 Applications of DTNs

DTN is used in several areas. In September 2003, Cisco router (CLEO) was launched by satellite to monitor disaster in UK, which has done a lot of routing tests in space environment till 2008 with DTN bundle protocol by making full use of the link source to overcome serious asymmetry link conditions. The Zebranet project [19] has installed a global positioning system (GPS) in a zebra collar to study the habits of zebra activities, which is one of the early DTN projects and was started in 2004. There are many rural communication projects in remote villages to provide the access to Internet. Some of which tries to reduce the cost of communications using the way of asynchronous information transmission. Like as a simple one-hop delay network to some remote areas, letting someone to drive a motorcycle with USB storage device to come and go between rural schools and cities with permanent Internet connection (such a round-trip may take several hours of time), so as to realize the connection between the school and the Internet. DTN storage and forwarding mechanism helps the researchers not to chose end-to-end communication mode, but rather using special node (data mule) in the lake for water protection which can exchange information with the gathering nodes accessing Internet when its called back from the lake. Collecting battlefield information or collecting data in depopulated area is actually achieved by the DTN.

2.2 Replication Techniques

Content Replication is an independent process aimed at creating copies of content at the network nodes, regardless of whether they asked for it or not. Given that the storage capacity at the nodes can be considered as unlimited and the content request rate is known, replication can effectively address the scarcity of radio resources and the need for an even traffic load distribution.

For replication gossiping and epidemic dissemination, where the information is forwarded to a randomly selected subset of neighbors. A random replication is an approach where a node hands over its content to a randomly chosen neighbor. Data replication and caching techniques are the important two services in distributed computing networks. It increases data availability by creating local or nearly available copies of popularly used items, by forwarding each query to its nearest copy; the query search latency can be effectively reduced.

In [20] and [21], the authors proposed akin mechanism to random walks to build probabilistic quorum systems for information dissemination and sharing. In [22], location theory is used to model the problem of determining where and how many content replicas to be placed in a dynamic network. Node grouping is exploited in [23], where group of strong links store and share information cooperatively.

Intelligent Replica placement algorithm, classifies the requested content into two classes mainly class I and class II. In class I, most frequently accessed contents are replicated in strong cluster which possesses high weight values and more number of copies. In class II least frequently accessed contents are replicated in weak cluster which is having low weight values and in less number of copies. Routing is performed hierarchically by broadcasting the query only to the strong clusters. Increased user response time and performance are the bottleneck of this approach. In strong clustered network a few replicated contents have more number of copies.

SPIDER (Spatial Indirection for path Diversity for Expedited Replication) proposed in [24], employs two orthogonal components to minimize the maximum time to replicate to all destination sites by creating of multiple dynamic distribution trees using transit nodes. In this algorithm content is not directly pushed to the clients but replicated only to some small set of edge servers. In DRPSCD [25], a balanced dissemination-tree (d-tree) is proposed which reduces the number of replicas deployed by comparing several replica placement algorithms.

Threshold-based content replication mechanism are proposed early. In particular, in [26] it is the original server that decides whether and where to replicate content or not. In [27] nodes have limited storage capabilities; if a node does not have enough free memory, it will replace a previously received content with a new one, only if it is going to access that piece of information more frequently than its neighbors up to h-hops.

2.3 Content Retrieval

There have been works on mobile phone-based data collection and retrieval systems. In [28] and [29], authors proposed that mobile phones or vehicles based DTN uses 3G connectivity to transfer HTML pages against user requests. A distributed image retrieval service is proposed in [30] based on a sensor platform. Works in [31] propose cooperative caching over replication for content retrieval in DTNs.

Information retrieval (IR) community has worked at length on information retrieval system that considers redundancy and novelty of retrieved information [32], [33] and [34].In most cases, these works mainly deal with text documents.

2.4 DTN Mobility Models

Mobility models for mobile wireless networks is an active and challenging research area. The simplest and most widely used mobility models are Random Walk and Random Waypoint (RWP) [35]. In Random Walk, an entity moves continuously, as a piecewise linear function of time. Ending upon one linear segment, the next is obtained through a random choice of new direction, and random choice of speed and distance. However, velocity among segments remain same. There is a introduction of pause time while walking randomly in the Random Waypoint.

In [36], Leavy Walk is proposed which is similar to random walks, except that the flight lengths and pause times are drawn from a power law distribution, which believed that unconstrained human movements follows leavy distribution. Although, it helps to mimic the meeting statistics similar to real world but lacks behind to capture characteristics such as heterogeneity among entities, repetitiveness, group mobility or any relationships between entities.

There are many situations where some model is required to mimic the movement among entities together as a group moving through the environment. For example, people of same family travels together or visit places for common purpose (e.g., eating at restaurants or shopping in malls). To capture their movement some group mobility work is proposed. In [37], Exponential Correlated Random Mobility model is described for introducing the group mobility model in an area.

Authors in [38] discuss community based mobility model, on the idea that entities favor squares with higher social attractivity. The social attractivity is based on how many friends are in the same square. Changing friends depending on the time of a day changing in periodic patterns such as people meeting their work colleagues in the day and their family in the evening. This model lacks group movement and the movement is relatively homogeneous. In [39], entities move to different squares at different times of a day in periodic manner, thereby creating some heterogeneity in both time and space which is introduced as time-variant mobility model.

The everyday life of people who go to work in the morning, spend their day at work, and commute back to their homes at evenings is modelled as a Working Day Movement (WDM) model proposed by Ekman et al. [40].

The authors in [41] describe a role-based and event-driven mobility model for disaster recovery networks, where agents speeds are accelerated when they are in disaster area. The disaster is modeled as events that attract (repel) people with certain roles to (from) the disaster area, as like civilians flee from the disaster area, whereas polices and ambulance frequently enter to the incident area. This model is an extension on random walks, seemingly simulates the chaos at disaster in action and is most suitable for live hazardous events like sudden fire in a building.

Chapter 3

Availability-aware Content Replication

In this chapter, we discuss availability-aware content replication (ACR) scheme for disaster response networks. As we described earlier, disaster response networks evolve as a form of a DTN after a disaster when regular communication means are not available (due to damage or power outage), communication among nodes mainly rely on device-to-device opportunistic forwarding. Disaster response networks usually suffer from frequent disruptions in operation due to link or device failure and high mobility of nodes. To make the network more resilient to failure, important contents should be replicated onto a set of nearby nodes of the network opportunistically using the available DTN communication so that in hour of need one of the replicas becomes available to deliver the content back to the source node.

It can be noted that routing protocols proposed for general DTNs are often not suitable for disaster response networks, as they concern themselves more with other performance objectives. For example, they often achieve a high delivery ratio at the expense of generous message copying and forwarding during significant communication. Almost all newer DTN protocols [42] [43] make their decision based on nodes' utilities towards the schemes' main goals rather then making decision of forwarding/dropping upon nodes characteristics such as, mobility, resources and others. In contrast to them, our proposed replication technique reduces the need for large numbers of copies by discovering recurrent contacts nearby and deliver content objects to the intended one directly (not more than one hop) by observing the mobility among the nodes in consideration. By using the recurrence of devices in the network, the degree of message replication is reduced and the number of message transmissions does also remain low.

We strongly observe and argue that recurrence exists in disaster response networks. This is because movement of entities in disaster scenes is not entirely random, rather some degree of regularity exists. Some observations are the following: police vehicles patrol on given routes, fire trucks originate at fire stations, and volunteers used for relief operation management move in a recurrent manner. Furthermore, a number of fixed points exist that can be used for content handover. For example, the evacuation camps, fire stations, and vehicles stranded on common routes may all serve as stepping stones to deliver content from moving entities. Hence, a core of moving entities exists that revisits an overlapping set of these fixed places repeatedly. Our replication mechanism takes advantage of this core to build a stable routing "base" by observing contact occurrences of network entities over time and identifying recurrent contacts among the network nodes.

3.1 Network Model for Availability-aware Replication

We consider a set of static and mobile nodes extended over a geographic area, where they remain disconnected, except when contacts occur. A *contact* occurs when a node comes near to another node's transmission range. At each contact, some nodes try to put some content to the other. Contents originating at source nodes, can be forwarded to the desired neighbor node to keep the replica of the content in the network. We consider each node maintains a set of neighbors, which are nodes that it contacts recurrently. To replicate the content, we consider no intermediate nodes between the content provider and content receiver.

3.2 Problem Formulation

To address the resilience of failure in the disaster response network, contents contained in a source node are replicated onto other nodes for making them available in hour of need. Since the network is full of disruptions, the replicated contents might not be available instantly as they are requested. We investigate the problem of choosing suitable replica nodes for content replication so that the intended content can be retrieved within a bounded expected retrieval time.

Earlier opportunistic networking algorithms rely on simple, identical node mobility assumptions where nodes meet each other at independent identically distributed (i.i.d) time intervals, which are usually assumed to be exponentially distributed. The meeting rate—measured in terms of number of meetings per unit of time—is considered to be homogeneous, denoted by λ. This rate describes the contacts of every pair of nodes. As a result of this single rate, all nodes are considered equally rather than individually. Different meeting rates per node pair, however, is helpful but introduces higher complexity.

We observe that replicating contents can increase the availability of the content. However, making too much replications over the network floods the network and degrades performance and efficiency. To overcome this, we focus on an optimal content replication technique among nodes over the network. We allow only a small number of replications to happen and thereby choose a set of *replica nodes* for a given source node in a manner so that the source node can retrieve replicated contents back (from at least on the replica nodes) within a lower wait time.

For developing an effective content replication technique, we consider three issues:

- We model the contact process among nodes by a stochastic Markov chain and hence, compute the availability of a node in terms of how likely a node remains in contact with an another given node.
- We then analyze the Markov chain to analytically derive the expected wait time to retrieve contents back from a set of replica nodes.
- Based on the expected retrieval time, we then choose a set of replica nodes for a given source node that minimizes the expected wait time to retrieve the contents back from at least one of the replicas.

3.2.1 Computing Availability of Nodes

Let N be an opportunistic network with $|N| = M$ nodes. We assume that N is a relatively sparse ad-hoc network where node density is insufficient for making and maintaining multi-hop paths [15]. Contents are stored and carried by nodes. Contents are forwarded only to the desired replica nodes which is selected from intermittent contacts by node's mobility. A *contact* is said to occur between a pair of nodes when they arrive within the transmission range of their radios, and then can setup a bidirectional wireless link to transfer messages across them.

Table 3.1 exhibits different notations used in this thesis.

Table 3.1: Important Notations

Symbols	Meaning
N	the network(set of nodes)
i,j,n	nodes in N
L	neighbor set for a node
T_{0j}	disconnection period to j for node n
T_{j0}	connected period to j for node n
$E[HT_s]$	expected holding time in state s
R	replica set of a node
$X(t)$	Markov Chain
λ_{0j}	connecting rate to node j considering $X(t)$ for node n
λ_{j0}	disconnecting rate from node j considering $X(t)$ for node n
P_0	probability to stay in state 0 of $X(t)$
P_j	probability to stay in state j of $X(t)$
k	maximum allowed replication (number of copies)
$E[T_r]$	expected retrieval time

Let n be a node in a network which is assumed to be the *source node* for content replication. Node n tries to find suitable replica nodes where it can be replicate its important contents. It can be seen that node n would not be able to communicate with all other nodes in the network due to lack of direct encounter. As a result, node n can only interact with a smaller number of other nodes to whom it may have contacts with over time. Among them, a few of them may be more frequent whom it meets more frequently than others. This set of nodes constitutes its *neighbor set*, denoted by L. In the following, we use the same symbol L to denote the neighbor set as well as the cardinality of the set. We also assume neighbor nodes are indexed, in that we can express, $L = \{1, 2, \cdots, L\}$.

It is obvious that neighbor nodes in L are able to directly communicate with the source node n. The source node can pass its content to a neighbor node when the source node is in contact with the neighbor node. Apparently, not all neighbors are suitable to be selected as a replica node, instead of a smaller subset of L is identified as the replica set. We denote this set by R. Clearly, $R \subseteq L$. Our task is to find R for a given source node.

3.2.1.1 Modeling Markov Chain

We model the contact process among nodes in the network by a *continuous-time Markov chain.* As a node can be in contact with any of its neighbors at any arbitrary time, we denote the Markov state, $X(t)$, as the neighbor node to which the source node is in contact at time t. In that, $X(t) \in \{1, 2, \cdots, L\}$. The state $X(t)$ can also take '0' to denote that the node is *not* in contact with any of its neighbor at time t, which means the source node is *disconnected.* That means the chain take values from $\{0, 1, 2, \cdots, L\}$ and any non-zero value of $X(t)$ denotes that the source node is *connected* with some of its neighbor nodes (given by $X(t)$).

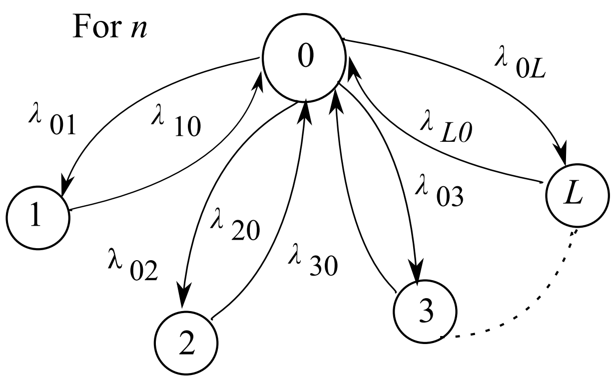


Figure 3.1: Continuous-time Markov chain $X(t)$ of node n

$$X(t) : t \geq 0 \text{ where X(t)= neighbor node in meeting with n at time t.}$$

This chain, $X(t)$, is constructed for every node in the network. State transition rates of the chain denote the transition rates as the source node moves from one contact to another as it gets connected to one of its neighbor at a certain time, remains connected for a while before getting disconnected from it (that is, moving to state '0'), and then connecting again to another node. Thus the chain transits from one state to another over time. As per Markov chain context, these state transition rates capture the timings among these state as time passes.

Every node contains the chain and updates the rates at each contact with other nodes. The states are summarized in Table 3.2.

Table 3.2: State Description

State, $X(t)$	Description
0:	Disconnected state of node n (connected to none)
j:	Node n is connected with node j

3.2.1.2 State Transition Rates of the Chain

We denote the transition rate from state i to state j by λ_{ij}, which means the rate at which the source node being connected with node i moves to connecting node j. As per model, we assume that the source node does not remain connected with two nodes at the same time. That means, all transition starts from state 0 and also ends at 0. Consequently, we have the following rate matrix.

$$\lambda_{ij} = \begin{cases} 0 & \text{if} \quad i \neq 0 \quad \text{and} \quad j \neq 0 \\ \lambda_{i0} & \text{rate of leaving node i} \\ \lambda_{0i} & \text{rate of entering to node i} \end{cases}$$

Recall that rate λ_{ij} indicates that the time gap between being in state i and being in state j is exponentially distributed with rate λ_{ij}, that is, $T_{ij} \sim expo(\lambda_{ij})$, which in turns means the expected transition time $E[T_{ij}] = \frac{1}{\lambda_{ij}}$.

Rate λ_{0j} means the rate of leaving from the disconnecting state 0 towards state j by connecting the source node, n, with neighbor node j. Similarly, rate λ_{j0} represents the leaving rate of node j towards to 0.

3.2.1.3 State Probabilities

Given state transition rates of the Markov chain, we can also derive *state probability* assigned to every state indicating what fraction of time in the long run the chain remains at that state. We use P_j to denote state probability at state j. In our context, P_j means what fraction of time the source node n remains connected with node j ($j > 0$), where P_0 means what fraction of time the source node remains connected to none. In other words, these probability values indicate the stationary chance that the source node would be found remain connected with a given neighbor node (or not connected to anyone either) at any random instance of time. That means, $P_j = P\{X(t) = j\}$, for any time t.

According to the Markov chain rule,the following equation holds (which means the node needs to be any state whatsoever at any time):

$$P_0 + \sum_{j=1}^{L} P_j = 1 \tag{3.1}$$

We can use the *balance equations* from the Markov chain to derive these state probabilities from the transition rates. Balance equations that for any given state the total incoming rate toward the state is equal to the rate of leaving from that state. We have the following equations:

Table 3.3: Balance equations

State	Rate at which the process leaves = Rate at which it enters
0	$\lambda_{01}P_0 + \lambda_{02}P_0 + \cdots + \lambda_{0L}P_0 = \lambda_{10}P_1 + \lambda_{20}P_2 + \cdots + \lambda_{L0}P_L$
j	$\lambda_{j0}P_J = \lambda_{0j}P_0$

From the balance equations, we compute the disconnected and connected state probabilities P_0 and P_j respectively as follows.

$$P_0 = \frac{1}{1 + \sum_{j=1}^{L} \frac{\lambda_{0j}}{\lambda_{j0}}} \tag{3.2}$$

$$P_j = \frac{\lambda_{0j}}{\lambda_{j0}} * \frac{1}{1 + \sum_{j=1}^{L} \frac{\lambda_{0j}}{\lambda_{j0}}} \tag{3.3}$$

These P_j values indicate the *availability* of the corresponding node j in terms of what fraction of time node j remains connected to the source node. Node with higher P_j is likely to be more available to n than nodes with lower P_j. Using these probability values as well as the state transition rates, we select the replica set for a given source node which we describe next.

3.2.2 Replica Node Selection

We derive an analytical expression for computing the expected wait time to retrieve content when replicated to a set of replica nodes. The objective to find the replica set that gives the lowest expected wait time for a given source node. We try to choose k replicas, denoted by R, for a source node n from its neighbor set L, to which the source node replicates its content to. The parameter k designates the *replication factor*.

The expected retrieval of a content is measured from the states of $X(t)$. It will add all the possibilities of being in any state of the chain $X(t)$ to compute the expected retrieval time. We can compute any expectation by conditioning upon some instance. That is,

$$E[T] = \sum_{c} E[T|C]P(C) \tag{3.4}$$

We therefore, use Eq. 3.4 to compute our expected retrieve time conditioning upon the states of chain $X(t)$'s.

$$E[T_r] = \sum_{l=0}^{|L|} E[T|X(t) = l]P_l \tag{3.5}$$

We can consider the neighbor set L as a union of two sets, one is the replica set R and others are considered as in $L - R$. As per chain's concept we can consider a set of states $\{L\} \cup \{0\}$ upon which we could condition the expected retrieve time as follows:

$$E[T|X(t)=j] = \begin{cases} E[T|0] & j = 0 \\ 0 & j \in R \\ \frac{1}{\lambda_{j0}} + E[T|0] & j \in \{L - R\} \end{cases}$$

If node n is in contact to any of the replica nodes then the expected retrieve of the content from that node will be instantaneous. If n is connected with any other node l rather than replicas then it needs to wait till to get in contact to any replica node.

The time to connect from a disconnection state to any replica node of the replica set R is exponentially distributed.

$$T|0 \sim expo(\lambda_1 + \lambda_2 + \cdots + \lambda_R) \tag{3.6}$$

So that, the expected time to be in a contact with atleast one replica is as follows.

$$E[T|0] = \frac{1}{\sum_{j \in R} \lambda_{0j}} \tag{3.7}$$

In general we get the expected retrieval time $E[T_r]$ from Eq. 3.5

$$E[T_r] = \sum_{i \in (L-R)} \left(\frac{1}{\lambda_{i0}} + \frac{1}{\sum_{j \in R} \lambda_{0j}} \right) P_i \tag{3.8}$$

We have to select suitable nodes for R so that the expected retrieval time is minimized.

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in (L-R)} \left(\frac{1}{\lambda_{i0}} + \frac{1}{\sum_{j \in R} \lambda_{0j}} \right) P_i \\ & \text{subject to } \mid R \mid \leq k \end{aligned} \tag{3.9}$$

Let x_i be an indicator variable, denotes a binary value that node $i \; \epsilon \; L$ is chosen for replica by node n. So x_i would be set to 1 if chosen, otherwise 0. This gives—

$$x_i = \begin{cases} 1 & \text{if node } i \text{ is chosen} \\ 0 & \text{otherwise} \end{cases}$$

By introducing the indicator variable x_i in Eq. 3.5

$$E[T_r] = \sum_{i=0}^{L} (1 - x_i) \left(\frac{1}{\lambda_{i0}} + \frac{1}{\sum_{j=1}^{L} x_j \lambda_{0j}} \right) P_i \quad (3.10)$$

We have to find the proper x_i 's for content replication such that they could minimize the retrieving time for content, depicted at Eq. 3.10

$$\begin{aligned} \text{minimize} \quad & \sum_{i=0}^{L} (1 - x_i) \left(\frac{1}{\lambda_{i0}} + \frac{1}{\sum_{j=1}^{L} x_j \lambda_{0j}} \right) P_i \\ \text{subject to} & \sum_{i=1}^{L} x_i \leq k \end{aligned} \quad (3.11)$$

Eq. 3.11 is a hard instance of non linear optimization problem. If we consider leaving rates (λ_{i0}) and state probabilities (P_i) are constants then the expression of Eq. 3.11 is considered to be minimized if $\sum_{j=1}^{L} x_j \lambda_{0j}$ is maximized. To maximize this $\sum_{j=1}^{L} x_j \lambda_{0j}$ it is considered to select the top k neighbors having higher values in λ_{0j}, connecting rate to neighbor. To solve the hard instance we consider these transition rates λ_{0i} as one of the heuristic approaches. Along with λ_{0i} some other heuristic approaches are discussed in the following section to find the suitable replica set.

3.3 Heuristic Approaches

In this section, we propose a few heuristic approaches, to find the suitable replicas that minimizes content retrieval time. Algorithm 1 illustrates the selection process of suitable replica set from the neighbors using different schemes for replicate the content to it.

3.3.1 Lambda Rate (λ)(LRAR)

The *Markov* chain stated in Section 3.2.1.1 have heterogeneous meeting rates noted as λ. We use the chain's connecting and disconnection meeting rates for metric the availability. The rates of connecting of any node n to the other node j by λ_{0j} are stored in the neighbor table. By sorting the neighbor list based upon descending connection rates (λ_{0j}) and picking up first k neighbors as a replica set. The meeting rates are changed upon the contact duration and meeting counting. The neighbor list updates periodically due to change in meeting rate.

3.3.2 State Probability(P_j) (PRAR)

In this section, we propose another heuristic approach for selecting suitable replica set for content replication. We used the state probabilities P_j stated in 3.2.1.1 *Markov* chain $X(t)$. We also sort the chains P_j in ascending order upon the neighbor list and select the first top k neighbors to replicate the content. If there is less then k neighbors this will pick all neighbors maintaining the neighbor property.

3.3.3 State Probability and Lambda rate ($P_j + \lambda$) (PLAR)

In this section, we suggest another heuristic approach of selecting the top k neighboring nodes upon ranking the neighbors with state probability P_j added up with meeting rate to the neighboring node j with rate λ_{0j} for any node n 's chain $X(t)$. We ranked the neighbors with the higher added values of $P_j + \lambda$ ones upright upon the lower ones.

3.3.4 State Probability by Lambda rate (P_j/λ) (PbyLAR)

In this part, we try to make a choice of replica set for n by the ascending order of state probability P_j by the meeting rate λ_{0j} of node j in the *Markov* chain $X(t)$ stated in 3.2.1.1.

Algorithm 1 Selecting replica set from neighbour list based upon different schemes

procedure SELECTREPLICAS(*neighbourlist*,*scheme*)
  for each node $i \epsilon$ *neighbourlist* **do**
    update meeting rates λ_{0i} and λ_{i0}
    compute probability P_i with updated rates of the chain $X(t)$
  end for
  if *scheme* is LRAR **then**
    sort the *neighbourlist* in descending order based on connecting rate of the neighbor λ_{0i}
  else if *scheme* is PRAR **then**
    sort the *neighbourlist* in descending order based on state probability P_i of the *Markov* chain $X(t)$.
  else if *scheme* is PLAR **then**
    sort the *neighbourlist* in decreasing order based upon the value of state probability P_i of the *Markov* chain $X(t)$ added with the connecting rate λ, $(P_i + \lambda_{0i})$.
  else if *scheme* is PbyLAR **then**
    sort the *neighbourlist* in ascending order based on state probability P_i of the *Markov* chain $X(t)$ by the connecting rate λ, (P_i / λ_{0i}).
  end if
  replicaSet $\leftarrow$ select top k neighbors from the sorted *neighbourlist*.
  return *replicaSet*
end procedure

Chapter 4

Availability-aware Replication Protocol

In this chapter we describe our proposed protocol for availability-aware content replication. It describes the models of the contents and queries, communication among nodes, neighbor identification and the selection of neighbors.

4.1 Content and Query Model

Nodes generate contents, and contents have unique identifiers so they can be differentiated. The size of the contents resides as a meta-data in the content. Contents can be originated in different times with different lifespans. Table 4.1 depicts the contents container associated with the nodes. When a source node intends to retrieve a content it invokes queries. These queries also have unique identifier associated with the object identifier and object size as a meta-data reside in it. Such identifiers help the queries to retrieve proper contents. The objects will have an expected bound retrieval time which will be specified upon the contact process of the replica nodes based on nodes movement. Rather than making a broadcast, the queries are asked the contact neighbors that the source node is meeting currently. Queries lifespan over the network is considered until the query is served by one of the suitable replica node. The contents are considered irrelevant after their lifespan gets over. Table 4.2 shows the query container associated with the nodes.

Table 4.1: Content container

Key	Content	Content Size	TTL
O_1	Obj_1	$size_{Obj_1}$	TTL_{Obj_1}
O_2	Obj_2	$size_{Obj_2}$	TTL_{Obj_2}
O_3	Obj_3	$size_{Obj_3}$	TTL_{Obj_3}
O_4	Obj_4	$size_{Obj_4}$	TTL_{Obj_4}
$\vdots$	$\vdots$	$\vdots$	$\vdots$
O_X	Obj_X	$size_{Obj_X}$	TTL_{Obj_X}

Table 4.2: Query container

Query	key	Object to Query	expected retrieve time
Q_1	$MQ1 - O1$	Obj_1	$E[T_{Obj_1}]$
Q_2	$MQ2 - O2$	Obj_2	$E[T_{Obj_2}]$
Q_3	$MQ3 - O3$	Obj_3	$E[T_{Obj_3}]$
Q_4	$MQ4 - O4$	Obj_4	$E[T_{Obj_4}]$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
Q_X	$MQ1 - O1$	Obj_X	$E[T_{Obj_X}]$

4.2 Modeling Communication

We cannot assume we know the source address of a content because multiple nodes can put the same meta information to a message at different times. For this reason, we choose that only sources will push their contents to others. The queries are also generated in a similar manner. Queries are initiated after the object creation. It will not be resolved until the desired content is retrieved. The nodes will form a neighbor set from which they could decide to replicate their contents those are called replica nodes measuring probabilistically an expected time when any replica node can contact with him again so the query generated node can retrieve its content in that contact. We discuss about choosing suitable replica from a neighbor set formed by recurrent contacts to whom the source could place the replica in the following sections.

4.3 Identify Neighbors

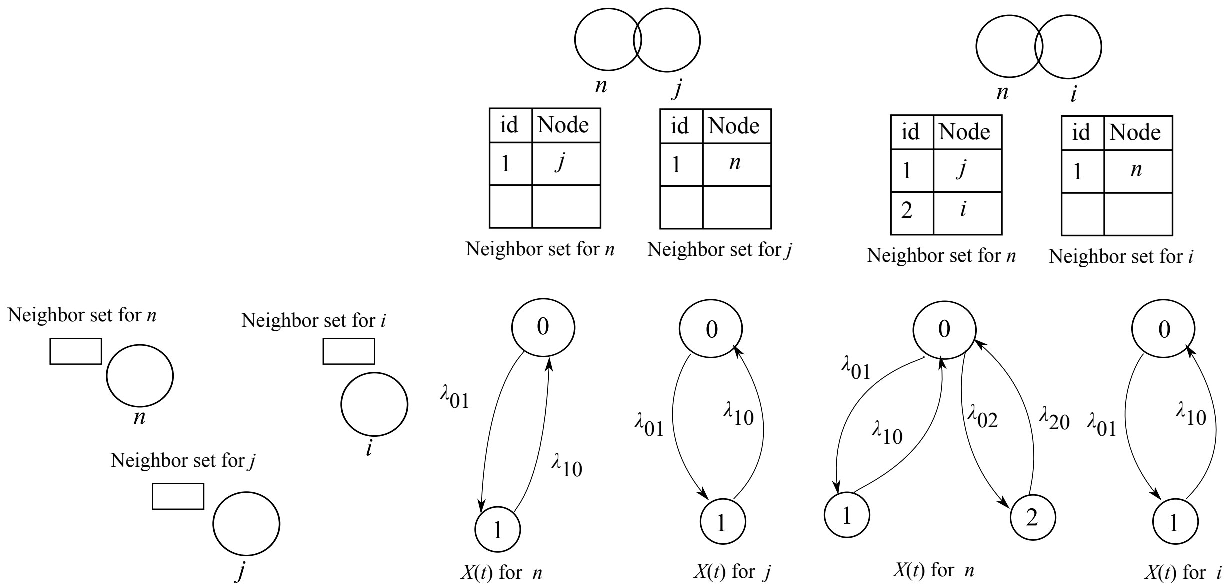
In disaster response networks (DRNs), nodes move in space and pose unavailability in communication links both in time and space. In order to utilize intermittently available links for effective data communication, one needs to understand the availability as well as disruption of links. In DRNs, to route packets, we exploit an overarching node behavior principle; namely, recurrence. In that, the network can be construed as a collection of recurrent contacts (i.e., meeting between a pair of nodes) and the associated time gaps between those contacts. We proposed this recurrent contact process as Markov chain in 3.2.1.1. The mobility model we used follows the recurrent contact process among nodes. For viewing the recurrent contact, consider node i,j,p,l in the communicating area are sparsely distributed. At time t_s node i meets node j. After a while node i meets k and then l. node j meets node i again after some time , this meeting is considered as a recurrent contact between node i and j.

Mobility of node will cause recurrent contacts among the nodes. We used Post Disaster Mobility (PDM) stated in [44] by M.Y.S.Uddin et al., which is capable to capture the recurrence nature in the disaster-affected areas.

Table 4.3: Neighbor table set for node n

Neighbor	Probability	Connecting Rate	Disconnecting Rate
1	P_1	λ_{01}	λ_{10}
2	P_2	λ_{02}	λ_{20}
3	P_3	λ_{03}	λ_{30}
4	P_4	λ_{04}	λ_{40}
$\vdots$	$\vdots$	$\vdots$	$\vdots$
L	P_L	λ_{0L}	λ_{L0}

Based on the recurrent contacts among the nodes, the source node forms a neighbor set. Every node maintains a neighbor table. In the neighbor table the nodes preserve information such as the meeting times, meeting durations, meeting disconnecting times, number of contacts, rate to connection, and rate of disconnection. When two nodes meet, they take into their neighbor set to see whether it is a new node or an already seen one. Based upon the situation, they update their neighbor set by appending the new node and update an existing one. For instance, if any node n meets nodes i, j, l, and m consequently, they are considered as n 's first, second, third and fourth neighbor respectively. If n meets another node u, then it is considered as a new neighbor. Table 4.3 will show the structure of neighbor table for n.



(a) At initial time T_0 (b) At time T_x (c) At time T_y, $T_y > T_x$

Figure 4.1: Construction of neighbor sets for nodes in the network

Let us consider a scenario where n, i, j are sparsely located nodes in the network. At initial time T_0, the corresponding neighbor sets for each node are empty. This phenomena is illustrated in Fig. 4.1(a). At time T_x, node j comes within the transmission range of node n and establish a connection. Next, node n adds an entry for node j in its neighbor set. Similarly node j also adds an entry for node n in its neighbor set (Fig. 4.1(b)). At the same time both the nodes construct their corresponding *Markov* chain $X(t)$. Fig. 4.1(b) exhibits the *Markov* chain for node n and j.

After a certain duration, node j moves out from the transmission range of node n and another node i comes within the transmission range of n. Hence, node n terminates the corresponding connection with node j and establishes a new connection with node i. Then, node n add a new entry for node i in its neighbor set. Similarly, node i also add an entry for node n in its neighbor list (Fig. 4.1(c)). Fig. 4.1(c) demonstrates the corresponding *Markov* chains for node n and node i.

4.4 Computing transition rates λ_{ij}

The transition rates λ_{0j} and λ_{j0} depend upon the expected holding time in the chain's state $X(t)$ for any node n. For n, the expected disconnecting time $E[HT_0]$, is the aggregate period of disconnection from all other neighbor nodes per contact with other neighbor nodes. Node n meets either i, j, l or other neighbor node in a recurrent manner. So the chain $X(t)$ will have i, j, l as states and when n comes to the transmission range of any one of them then they are connected with each other. When node n is connected with a specific neighbor it will remain on the corresponding state of the neighbor in chain $X(t)$. The expected holding period of such node n with any neighbor nodes can be thought of node n's aggregated connection periods to other neighbor nodes per contact. The calculation of the expected holding time is elaborated with an example.

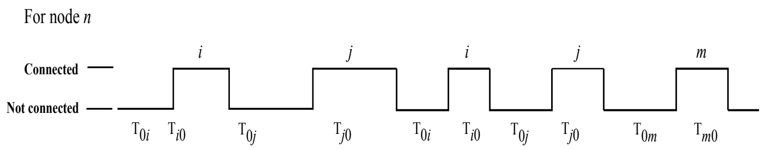


Figure 4.2: Instances of node n connecting with different nodes at different time

Let us consider a scenario with four nodes n, i, j, m in the network. Node n comes with the contact of different nodes at different time interval. Suppose, after T_{0i} node n contacts with node i. This T_{0i} is denoted as the disconnecting period for node n with respect to node i. Node n remains with the contact of node i for the time interval T_{i0}. This time interval is considered as connecting period to node i for the node n. After the time interval T_{0j} node n comes in to the contact of another node j. This T_{0j} time interval is the disconnecting period of node n with respect to node j. Node j remains with the contact of node n for T_{j0} which is the contact period of node n to node j. Next after T_{0i} time interval node i comes with the contact of node n and remains in the contact for T_{i0} time interval. Again after time interval T_{0j} the node j comes with the contact of n and remains in contact with node j for T_{j0} time interval. Similarly node m also connects with node n. This scenario is demonstrated in Fig. 4.2. Now we need to compute the holding time for disconnection and connection period for node n. First we, illustrate the computation of holding time for disconnection period for node n. The expected holding time of disconnection period of node n is shown in the following equation:

$$E[HT_0] = \frac{T_{0i} + T_{0j} + T_{0i} + T_{0j} + T_{0m}}{\text{total number of disconnection occurances} = 5} \tag{4.1}$$

Next we, compute the expected holding time for being in contact of node n with node i is as follows:

$$E[HT_i] = \frac{T_{i0} + T_{i0}}{\text{total number of contacts with } i = 2} \tag{4.2}$$

Similarly, the expected holding time for connection period for node n with node j and m are as follows:

$$E[HT_i] = \frac{T_{j0} + T_{j0}}{\text{total number of contacts with } j = 2} \tag{4.3}$$

$$E[HT_m] = \frac{T_{m0}}{\text{total number of contacts with } m = 1} \tag{4.4}$$

In general the expected holding time for disconnection and connection period of any node is as follows:

$$E[HT_0] = \frac{\text{aggregate amount of disconnection period}}{\text{number of disconnection occurances}} \tag{4.5}$$

$$E[HT_j] = \frac{\text{aggregate amount of connection period to } j}{\text{number of contacts with } j} \tag{4.6}$$

The inverse of this expected holding time will help to compute the rates of the chain $X(t)$. Therefore the transition rates are calculated as follows:

$$\lambda_{0j} = \frac{1}{E[HT_0]} \tag{4.7}$$

$$\lambda_{j0} = \frac{1}{E[HT_j]} \tag{4.8}$$

4.5 Selection of Replica Nodes

After forming the neighbor set the nodes choose the replica nodes to put the contents and retrieve them in a expected retrieval time when needed. Not all the nodes are considered as replica nodes. Minimum three recurrent contacts is considered as a primary criteria for any node for being a replica node. A node not only considering the recurrent contacts but also seeing its contact duration as transfer metric of making replication of a content to choose replica. Instead of solving the optimizing problem of minimizing Eq. 3.10 rigorously, we propose some heuristic approaches to select top k nodes as replicas by the source node. Contents will be pushed to the suitable replica ones when the replica nodes meet the source node again. It is ensured that no more than k instances of the content will be created and spread among the network.

After selecting the replica nodes we need to replicate the contents to the selected replica nodes. We replicate the contents in a single hop transfer. When a query for a specific content is issued int the network then we retrieve the content at least from one of the selected replica nodes within a expected retrieval time. The following section will briefly describe content replication and retrieval mechanisms.

4.6 Content Replication and Retrieval

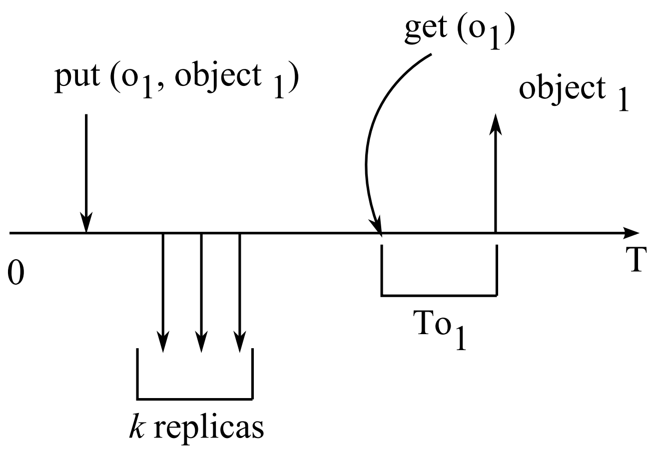


Figure 4.3: Replication and retrieval of a content

Contents are replicated to the suitable replicas. The content originator tries to relocate the contents by associating an unique key to the content while replicating it. So it will be easy to retrieve from the replica node with less effort. The content originator uses put(key,object) method to replicate the content upon the replica nodes and to retrieve the content in intended time it uses get(key) method. Let consider an example for node n using Fig. 4.3. Suppose another node o comes with in the communication range of node n and wants to replicate its content in node n. Hence, node o initiates a put function. This function will contain the content i.e., $object_1$ and an identifier o_1 for node o. Next, the content $object_1$ will be replicated k times. When the node o wants to retrieve its contents from node n, it just initiates a get function comprising ifs identifier o_1. The content will be then supplied to node o within a time interval T_{o_1}. Our goal is to minimize this time interval. These methods i.e., put and get are described elaborately.

4.6.1 put(key,object)

After selecting the replicas the originating node will tries to push the content to the desired replica node that he selects. The nodes are pushed to the replicas by put(key,object) command. In this command the content originating node wraps the content in a message with destination set to the replica ones and put it in to its message queue. When the originating node comes to contact of the replica ones, its try to push the message from the queue to the replica during that contact. If it is not delivered during that contact, the content will remain in the message queue until its lifespan. The algorithm 2, will perform the put operation for any originating node for replicating the content to the replica nodes.

4.6.2 get(key)

Content retrieval will be performed by resolving the queries. Queries will be served by the nodes to the intended node after there is successful content replication done by the content originator node. Queries are time driven. The queries will be preserved by the nodes unless they are served. When two nodes meet each other they share their queries for content to each

Algorithm 2 Replicate the content in to replica nodes'

procedure PUT(key,$object$)
  $replicaSet \leftarrow$ SelectReplica($neighbourlist$,$scheme$)
  $T_{ex} \leftarrow$ compute expected retrieve time for $object$.
  for each node $r \ \epsilon \ replicaSet$ **do**
    $replica_{object} \leftarrow$ make replica of $object$ from content originator.
    enqueue the $replica_{object}$ in the message queue selecting the destination node to r.
    if r is connected to content originator **then**
      if r can take the $replica_{object}$ **then**
        content originator push the $replica_{object}$ into r's object container.
        update status for transferring $replica_{object}$ is successfully done.
        removes the content entry from the message queue.
      end if
    end if
    if r is not connected to the content originator **then**
      content will remain in the message queue.
    end if
  end for
end procedure

other, if content exists in node's content buffer then they started transfer the content. After the successful transfer the query is considered as a successful serve. If it does not contain the object the query will remain unresolved and will be asked to the next meeter unless it was found. The algorithm 3, depicts the way of content retrieving from a node in a contact.

Algorithm 3 Retrieving the content from $contact{:}u \leftrightarrow v$

procedure GET(key)
  if query container is not empty **then**
    for each $q \ \epsilon$ query container of u **do**
      $q_{key} \leftarrow$ query object with associated key.
      if connected node v contains q_{key} object in its container **then**
        u will pass a control message to v for initiating object transfer
        v will transfer object in a message by setting u as destination
        if transfer is done during the contact **then**
          Update the retrieving time of the content.
        end if
      end if
    end for
  end if
end procedure

Various scenarios can occur during query resolve as failure. One could query a content that is not being replicated to the choose replica, means the replica it chooses did not appear again while scheme time ends another one is content is replicated but the replica node took too much time as per his previous contact cycle. Another scene might be the content the node is asking might not yet being replicated to the replica node by the put operation. These queries will remain unresolved and unsuccessful because it is not served to the intended one. The meeting time of two nodes is high in manner so the get command might fell short to retrieve the intent content in expected retrieve time.

Chapter 5

Experimental Results

We simulate a disaster response scenario based on the post-disaster mobility model “PDM” [44] extending the ONE [10]. In the scenario we see some coordination centers and relief camps along with neighborhoods , police stations and fire-stations. Rescue workers along with volunteer to help the people during the disaster. All these entity need to communicate effectively for the surviving in hard situation. After the disaster a relief operation is launched to help survivors (e.g.,supply food and water to people). There are a small number of main coordination centers, and a larger number of evaluation centers. Vehicles move between the main coordination centers and the evacuation centers to supply relief goods. These are some of the vehicles assumed to have DTN devices that buffer and carry communication between isolated coordination and evaluation centers.

5.1 Post-Disaster Mobility Model (PDM)

In this section, we want to highlight the mobility model that is used in our experiment to evaluate our protocol for content replication in the disaster response network. To produce recurrent nature of people, objects and activities, that is not captured in the base mobility models, such as the classical Random Waypoint Model (RWP) and the Map-Based Random Walk Model (MBM) in a post-disaster scenario. In contrast, the mobility model we use, named the post-disaster mobility model (PDM), attempts to produce the mobility of after the disasters happen such as large earthquakes or hurricane. Post-disaster model do not create the disaster mobility to a high degree of realism, but rather create a recognizable abstraction of the disaster mobility, focusing on key aspects whose effects we want to explore. The key aspect is the existence of some recurrent nature activities after the disaster. For example, consider a Hurricane site after the disaster. Coordination centers and relief camps are established nearby. Vehicles may carry supplies between the coordination centers and evacuation camps. A good number of rescue workers and volunteers are engaged to locate survivors and offer help. A few emergency vehicles may run between relief centers and distressed neighborhoods. Police officers may patrol neighborhoods to prevent unwanted activities.

PDM describes different role-based movements, based on a given city map. It models two main groups after a disaster: survivors, and rescue workers that aid survivors. PDM describes movement models for both groups. Hurricanes, tornadoes,earthquakes, and fire different disasters produce different movements. PDM model concentrate on scenarios where a fore-warned population moves in advance of the disaster to evacuation centers. They stay at these centers for a substantial amount of time and then return to their homes.

A number of rescue workers and volunteers are deployed at each evacuation center to dispense relief goods and services to the affected people serves as DTN carriers; other DTN carriers include police officers, who patrol a larger area, e.g. to prevent looting from relief camp. PDM , tries to mimic a couple of response services with the associated agents and their mobility. Instead of realizing the exact rescue operation in much detail, it wanted to do is to identify key mobile agents and to extract their mobility patterns in order to build at simulator accordingly.

PDM, focuses a number of characteristics which are described in the following sections.

5.1.1 Disaster Area and Neighborhoods

There is a map associated in the ONE simulator (a part of Helsinki, the capital of Finland) which is used as a map model of disaster area. The map contains connected road segments onto a 2-D plane where possible movements of people and vehicle can occur. PDM assumes that human lives in clustered neighborhoods and a few such neighborhoods are affected by the disaster.To construct neighborhoods, points are randomly chosen on the map as neighborhood centers that are far away from one another by certain distance (say, 500m). Then, the put the houses randomly around every center within a certain radius (say, 200m) from the center. Houses are located at the intersection of streets. Once the houses are built, people are created and are randomly assigned to particular houses. The house to which a person is assigned to is treated as his/her home. Figure 5.2 shows the city's neighborhood centers and houses.

Figure 5.1: City Map

Figure 5.2: Neighbourhood Map

The various values like the number of total neighborhoods, houses per neighborhoods, total number of people, minimum distance between neighborhoods, the radius of neighborhood are used as parameters to the model PDM.

5.1.2 Evacuation and Return

When the evacuation is announced, each human waits a random period of time, then decides whether or not to move. Those that move (chosen randomly, using a "probability of evacuating") go to the nearest evacuation center, where they wait for another random period of time, and then return home. People that do not evacuate, and those that have returned from evacuation randomly move around within their neighborhood.

5.1.3 Post-Disaster Relief Operation

The relief operation begins after the disaster. A set of centers are declared to participate in the recovery operation. A few centers are established prior (e.g., fire-stations), and a few are prepared in some premises for rescue and relief operations (e.g., medial centers, relief camps). There are three key components in modeling the disaster operation, placements of centers, mobile agents, and their interactions.

Centers PDM, includes a number of different center types, such as relief centers, evaluation camps, medical centers and hospitals, and police stations. These are at static locations in the map and are commonly visited repeatedly by moving agents.

Mobile Agents and Mobility Patterns Rescue workers are the main moving agents in the disaster response, as well as vehicles running between centers, camps, and stations to carry supplies, services, and aid workers. PDM, constrain all movement to take place along the streets on the map, whereas centers can be located in any intersection of streets. PDM, shows four mobility patterns undertaken by the agents which are described as follows.

Center-to-Center : This mobility is observed by vehicles which travel back and forth between a set of designated centers, camps or stations. In each trip starting from its home center, the agent picks a destination center from a set, finds a route, and moves toward the destination. After reaching the destination, it waits for a random duration which is considered as service time. After the service at the destination, it returns back to the home center. The oscillation repeats during the operation with some pause time in between.

Event-driven : This type of movement is made when a specific event is notified to a designated center and the associated agent (i.e., vehicle) visits the incident area. After the service, the agent returns to the base. This movement does not however oscillate, but occurs once as event triggers.

Cyclic route : Some agents take cyclic route from a particular center, visits a few locations of interest and returns to the home center. This is mainly observed by police as a part of their patrol, or any public transport system (bus/tram routes). After visiting each location, agents can optionally take a random walk around the location for a while before heading to the next location.

Convergence-Move : A set of agents with a particular role/duty get back to their reporting center around a certain fixed time, e.g., when all rescue workers are called back to relief camp for some special instruction.The opposite pattern is the divergence-Move.

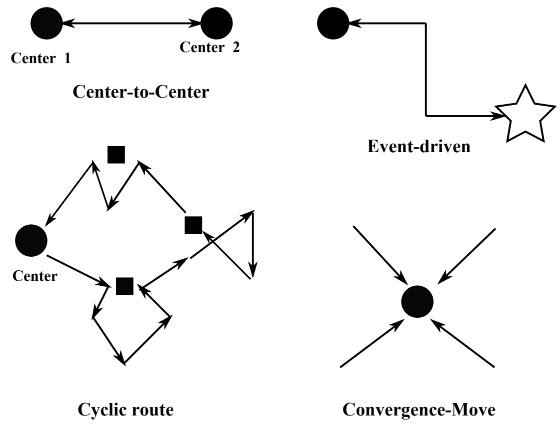


Figure 5.3: Different Mobility Patterns in PDM

PDM includes the following moving agents:

- *Rescue workers* at each neighborhood to help people to evacuate, and later to assist people with relocation. They move from the relief centers to houses and also perform a reverse route.
- *Supply vehicles* to carry relief goods between main centers and evacuation centers. They follow **center-to-center** mobility pattern
- *Ambulances and fire trucks* that respond to emergencies (i.e., event- driven mobility). Emergency events are generated at random locations at random times. An ambulance or fire truck starts from its respective home center to visit the target place. After the on-site service (with random duration) the vehicle returns back to the home center.

- *Police patrol cars* originate from police stations and regularly visit neighborhoods. Obviously, they take the **cyclic route** pattern. At the beginning of a patrol, a set of neighborhoods are chosen randomly. The patrol car starts from the police station and moves to the first neighborhood center. After reaching the neighborhood, it randomly visits a couple of locations (blocks) in that neighborhood with a random wait time at every location. Then, it chooses the next neighborhood and visits a few places there. Finally it returns to the police station. After a while, it picks another patrol.
- *Volunteers* People staying within a neighborhood join with rescue workers and move randomly in the neighborhood. The difference between volunteers and the rescue workers is that volunteers do not report to the relief camps, rather they return to their homes.

5.1.4 Interaction among agents

Moving agents are equipped with communicating devices that are capable of transferring data when they are within the radio range of each other. There are usually short range radio (say, 50m range) with pretty low data transmission rate. Generally, people (rescue workers, volunteers) keep battery-powered hand held devices (e.g., PDA), while for vehicles these devices are carried by the respective drivers. Each center (or camp/station) is also equipped with similar communicating devices which opportunistically relays information among vehicles or people when they pass by. All these devices are called "DTN routers". While moving around and meeting now and then, they form a disrupted but functional store-carry-and-forward type network that provides communication opportunity to the rescuers and survivors.

5.1.5 Implementation of PDM in the ONE

PDM, implemented on top of the ONE [10] simulator, which has four major movement types: inter-center movement (repeated back and forth movement of supply vehicles between relief centers and main coordination centers), rescue worker movement (localized mobility of volun-

Table 5.1: Simulation settings

Parameter	Value	Parameter	Value
Simulation time (hr)	24	Speed of human (km/hr)	4-5
Transmission range (m)	150	Wait time of vehicle (min)	10-30
Transmit speed (kBps)	250	Wait time of human (min)	10-20
Speed of vehicle (km/hr)	50-70	Retrieval deadlines (hr)	0.5-6
Average meeting time (s)	200		

teers in distressed neighborhoods), police patrols (cyclic police patrol movement among neighborhoods), and emergency movement (vehicles attending to an emergency event). There are also several types of fixed nodes including centers, police stations, and hospitals, which act as meeting places of moving nodes.

ONE configures itself based on input. ONE comes with three types of map-based movement models. The Map-Based Movement model is a derivative of the Random Walk model, where entities move to randomly determined directions on the map following the roads. Random pauses occur at the way points. The Shortest Path Map-Based Movement model is a derivative of the Random Waypoint model, where at decision points entities choose a random destination and then follow the map-based shortest path to that destination. It is also possible in ONE to specify in the configuration file deterministic routes for entities to follow.

5.2 Simulation Setup

In ONE, we build the immobile neighborhood by StaticMovement model (which is already part of ONE), and each neighborhood is at least 500 m far from one another. We consider there would be SimpleBroadcastInterface with transmission range of 150 m with speed of 250 kBps. Speed of vehicles are considered to be 50-70 km/hr while human walking speed is 4-5 km/hr in the scenario. Standard settings for vehicles speed, wait time along with Human walking speed and its wait time summarized in Table 5.1.

Table 5.2 shows the disaster response network entities with their movement models used in the ONE simulator for moving around the disaster area.

Table 5.2: Model settings

Type	Number	Movement model
Neighborhoods	10	CenterMovement [44]
Emergency	5	InCenterVehicleMovement [44]
House	100	CenterMovement
Repairer	1	InCenterVehicleMovement
ReleifCenter	9	ReliefCenterMovement [44]
Rescueworkers	40	RescueWorkerMovement [44]
MainReleifCenter	2	CenterMovement
Policepatrol	5	PolicePatrolMovement [44]
Policestation	1	CenterMovement
Publicworks	1	CenterMovement
EvacuationCenters	5	CenterMovement
MedicalCenters	2	CenterMovement
People	25	HumanMovement [44]
Coord. to Releif supply vehicle	10	InCenterVehicleMovement

We simulate for 24 hours in the post disaster mobility scenario.

5.3 Content Objects and Query Model in ONE

ONE can be configured with external events which tries to create, send, transfer and abort messages among the routers based upon event control. For the content replication scheme, we need to enhance the external events to trigger the creation of contents to an originating source. We create the queries for contents after they are created in the environment. Contents are generated after a certain observation over the environment of the network so that nodes can identify the recurrent neighbor nodes and construct a trustworthy availability metric.

5.4 Implementing Availability Router in ONE

In the simulation, we assumed every mobile device carries a communication device that also serves as a DTN router and that every center has a DTN router too. We assume that all communication occurs by the DTN-routing between routers when they come to close enough proximity (within 50 m). We enhanced the DTN routers as Availability routers to simulate our proposed protocol. As in our proposed protocol, nodes need to have storage for contents as well as for queries. Contents and Queries need to pass from originator to the replicas. So, we need to incorporate these storage in Availability router. Availability routers are capable of the computation of the availability for the selection of proper replica nodes using different selection schemes. Contents as well as control messages are passed between these routers incorporating in to messages. We extended the configuration of the ONE simulator to take buffer size as well as the number of replicas for the content in to the router. Extending the Active router class with such capabilities of immense computation and storage we form a new router in the routing package of the ONE simulator as Availability router. This router will follow the availability-aware replication protocol to replicate the contents in the disaster response network. Most of the connectivity in DRN is supplied by the movement of disaster response vehicles between various centers impacting content replication process. The map used is the one supplied with ONE (a portion of Helsinki) simulator.

5.5 Implementing Random Replication Router in ONE

We need to evaluate our proposed protocol for content replication to some state of the art approaches. Therefore, we extended the DTN routers to a random replica selection router named Random Replication Router(RRR). The RRR routers deliver the contents after observing the environment by the nodes for some time, which is configured in the input of ONE configuration. As per Availability Router, RRR also take buffer size and number of replica from the configuration set. For the selection of suitable replica, it just picks a node from its neighbor set randomly.

5.6 Experimental Evaluations

In this section, we narrate out extensive experiments to verify the efficiency of our proposed protocol with various choice schemes of selecting replicas such as LRAR, PRAR, PLAR, and PbyLAR stated in Sections (3.3.1, 3.3.2, 3.3.3, 3.3.4) with respect to the randomized replication scheme. We experimented the Availability Router's performance in the ONE simulator by varying different parameters. We vary different parameters: number of nodes, number of objects, number of replicas (k), and retrieval deadlines. Here retrieval deadlines denote a specified time interval to retrieve contents within it. Table 5.3 summarizes the values used for each parameter in our experiments.

Table 5.3: Simulation parameters

Parameter	Value
Number of replicas (k)	2, 3, 4, 5
Retrieval deadline (hr)	0.5, 1, 2, 4, 6
Number of objects	100, 300, 500, 700, 900
Number of nodes	225, 275, 325, 375, 425, 475, 525, 575

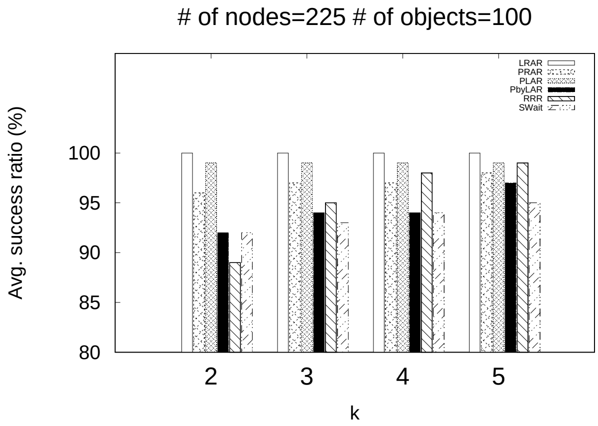
5.6.1 Performance Metrics

We evaluated the performance of our proposed methods using the ONE simulator. Here, we focused on two different performance metrics. These metrics are as follows:

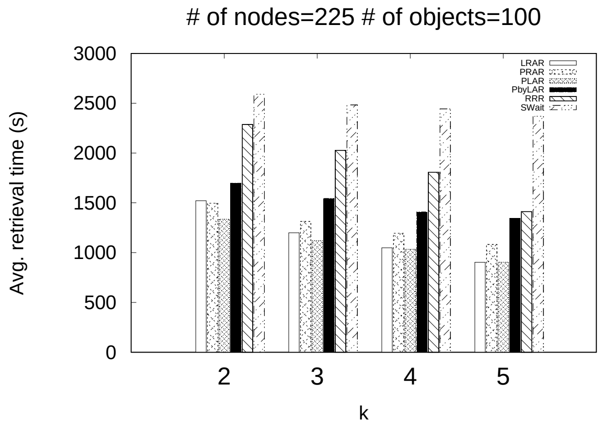
- *Average success ratio* : It measures the number of successful retrieval of contents with respect to the number of queries performed.
- *Average retrieval time (s)*: How long does it take to retrieve the contents by the nodes from at least one replica node?

We implement all our proposed schemes in Java and run the experiments 5 times and take the average of the results. We run our experiments on machines with Core i5 2.20 GHz CPU and 3.8 GB memory.

5.6.2 Impact of Variation of Different Number of Replicas k



(a) Average success ratio



(b) Average retrieval time

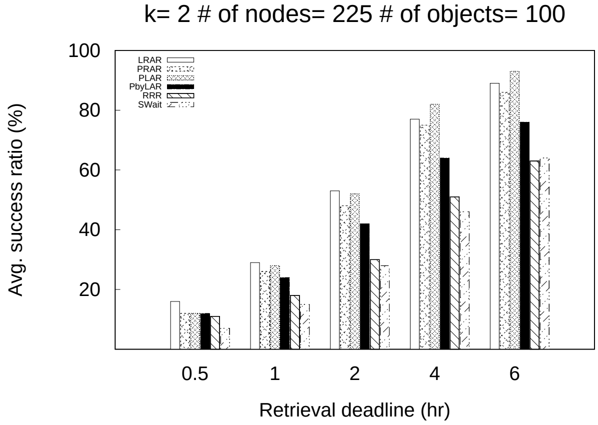
Figure 5.4: Impact of varying # replicas k for 225 nodes and 100 objects

We study the impact of different number of replicas k by varying k size using 2, 3, 4, and 5 and measuring the average success ratio and average retrieval time.

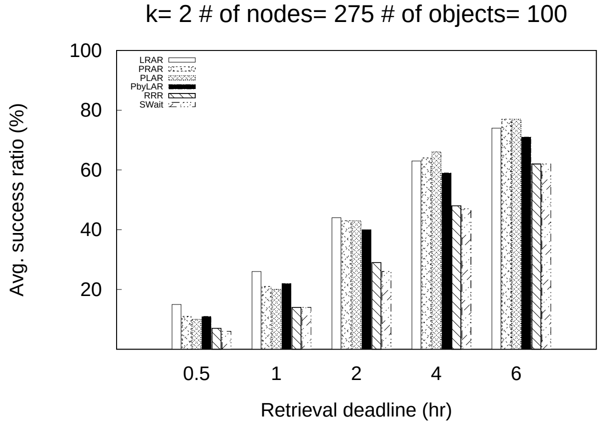
Fig. 5.4(a) depicts the impact of number of replicas on average success ratio. The average success ratio increases with an increase in the number of replicas k. However, the rate of increment diminishes with an increase in k. Fig. 5.4(a) demonstrates the efficacy of our proposed schemes over the state of the art approach RRR and SWait.

Furthermore, Fig. 5.4(b) illustrates the impact of different values of k on average retrieval time. It is evident from the figure that the average retrieval time degrades with an increase in k. For example, an increase of the number of replicas from 2 to 5, the average retrieval time decreases approximately 1.68 times, 1.38 times, 1.47 times, 1.26 times, and 1.62 times for LRAR, PRAR, PLAR, PbyLAR, and RRR respectively for 225 nodes and for 100 objects (5.4(b)). The efficacy of our proposed schemes over RRR and SWait is also evident for this variation.

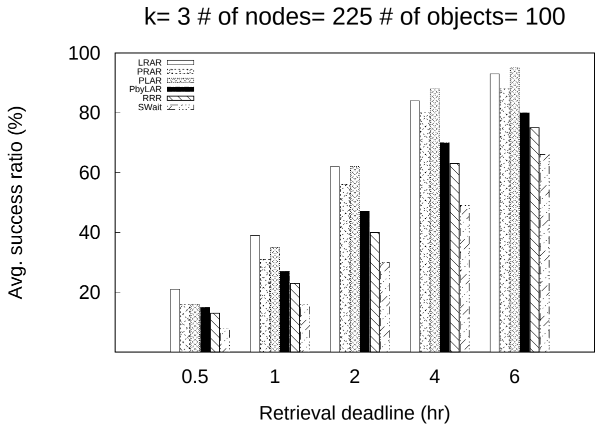
5.6.3 Impact of Variation of Retrieval Deadlines



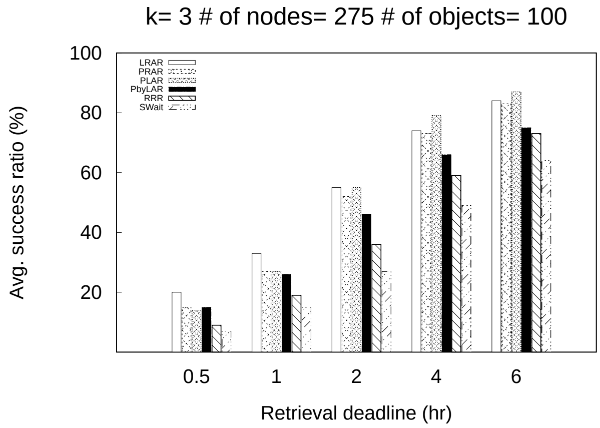
(a) Average success for k=2 and 225 nodes



(b) Average success for k=2 and 275 nodes



(c) Average success for k=3 and 225 nodes



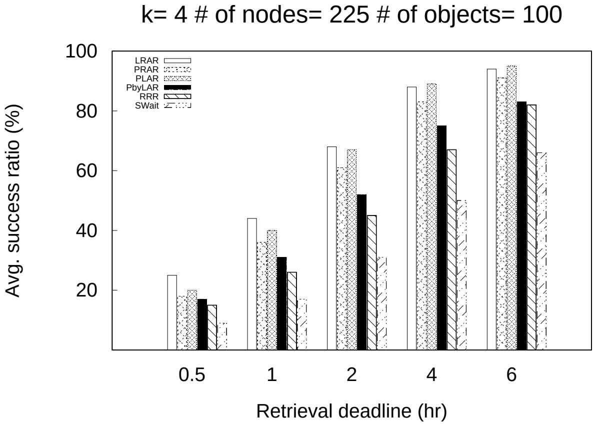
(d) Average success for k=3 and 275 nodes

Figure 5.5: Impact of varying retrieval deadlines on average success for different number of nodes for k=2 and k=3

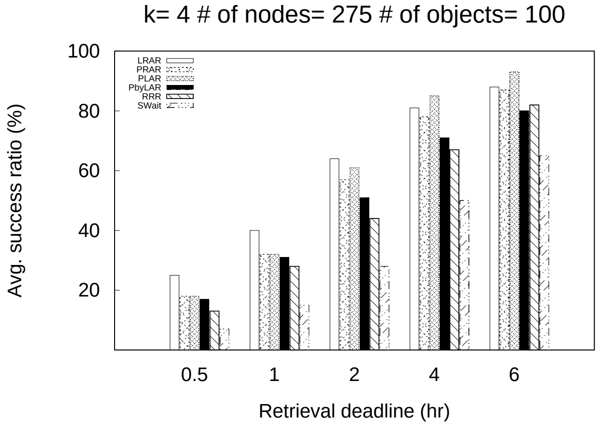
We study the impact of retrieval deadlines using 0.5, 1, 2, 4, and 5 and measuring the average success ratio. We also analyze the impact of variation of number of nodes and number of objects with respect to retrieval deadlines in Fig. 5.5, Fig. 5.6 and Fig. 5.7 respectively.

5.6.3.1 Effect of Different Number of Nodes

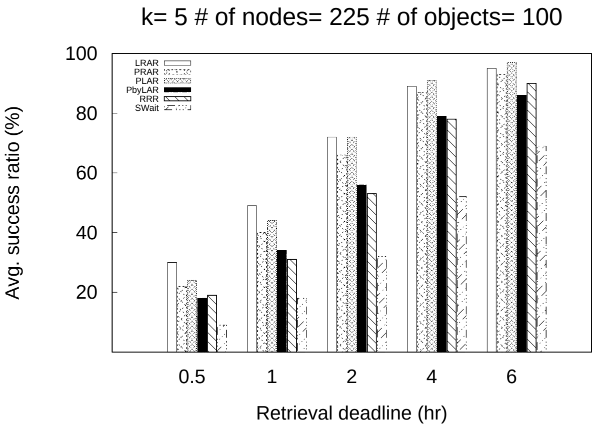
Fig. 5.5(a) depicts the impact of retrieval deadlines for average success for k=2 and 225 nodes. It is evident from the Fig. 5.5(a) that the average success increases with an increase in the retrieval deadlines. Fig. 5.5(c), Fig. 5.6(a), and Fig. 5.6(c) demonstrate the impact of retrieval



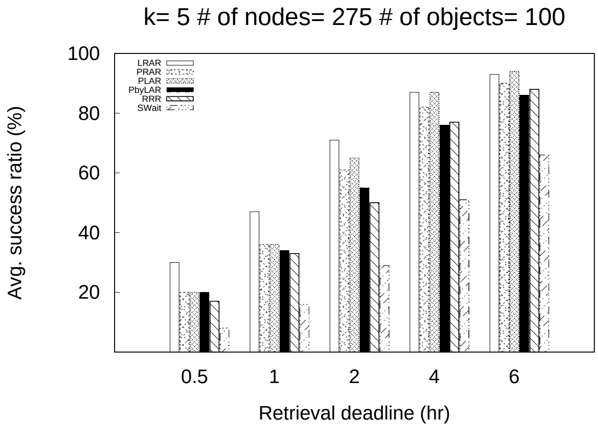
(a) Average success for k=4 and 225 nodes



(b) Average success for k=4 and 275 nodes



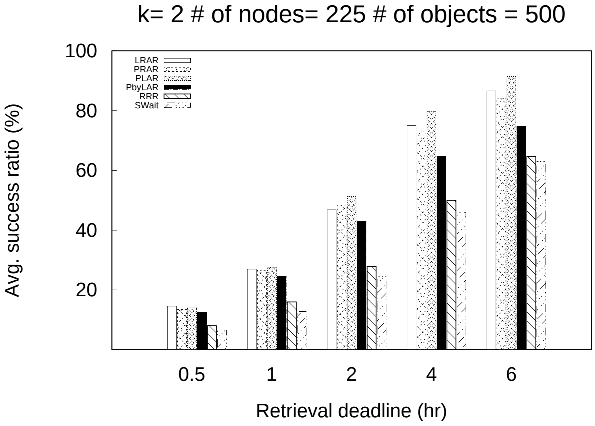
(c) Average success for k=5 and 225 nodes



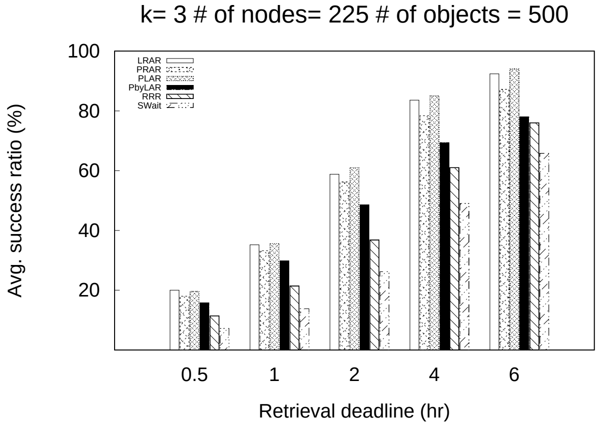
(d) Average success for k=5 and 275 nodes

Figure 5.6: Impact of varying retrieval deadlines on average success for different number of nodes for k=4 and k=5

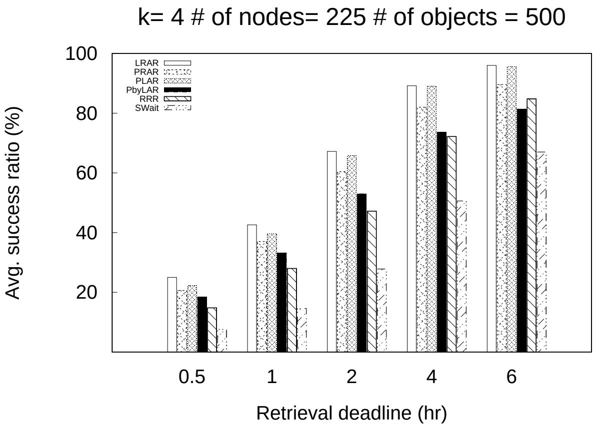
deadline for $k = 3$, $k = 4$, and $k = 5$ respectively for 225 nodes. These figures depict that the average success increases with an increase in the retrieval deadline for different values of replicas k. Besides, Fig. 5.5(b), Fig. 5.5(d), Fig. 5.6(b), and Fig. 5.6(d) illustrate the effect of different number of nodes on average success for different retrieval time. The efficacy of our proposed schemes over the state of the art approach RRR and Swait is evident for all variations.



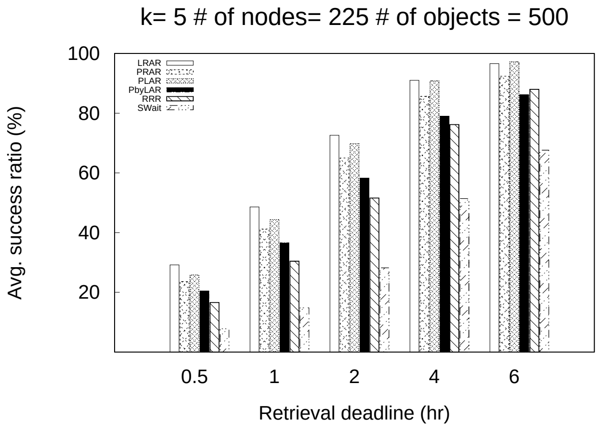
(a) Average success for k=4 and 225 nodes



(b) Average success for k=4 and 225 nodes



(c) Average success for k=4 and 225 nodes



(d) Average success for k=5 and 225 nodes

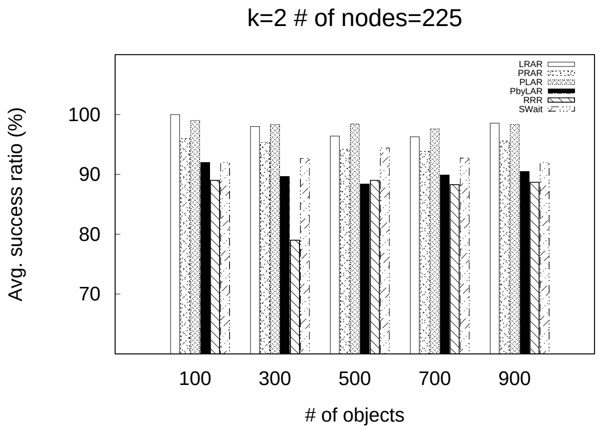
Figure 5.7: Impact of varying retrieval deadlines on average success for different number of objects

5.6.3.2 Effect of Different Number of Objects

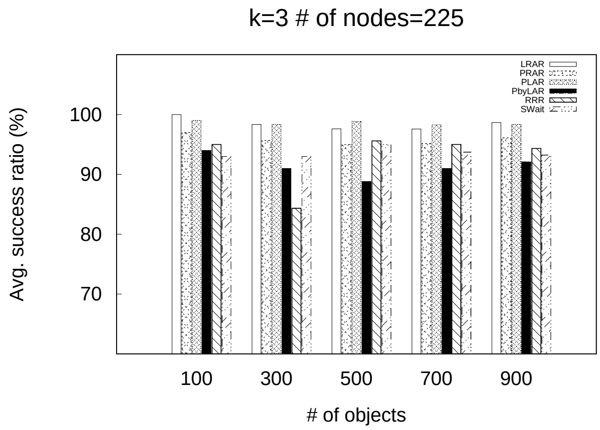
Fig. 5.7 demonstrates the effect of different number of objects for different deadlines. Fig. 5.7(a) depicts that the average success increases with an increase in the retrieval deadline for $k = 2$, 225 nodes, and 500 objects. Fig. 5.7(b), Fig. 5.7(c), and Fig. 5.7(d) depict the similar pattern for $k = 3$, $k = 4$, and $k = 5$ respectively. All these figures exhibit our efficacy.

5.6.4 Impact of Variation of Different Number of Objects

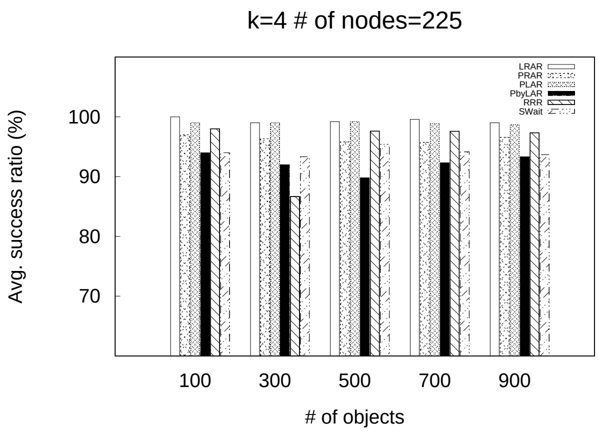
Fig. 5.8 demonstrates the impact of variation of number of objects on average success for different values of replicas k. Fig. 5.8(a), Fig. 5.8(b), Fig. 5.8(c), and Fig. 5.8(d) depict an increase in the average success with an increase in the number of objects. For example, for an increase of the number of objects from 100 to 700, the average success increases approximately 0.148 times, 0.146 times, 0.144 times, 0.146 times, and 0.144 times for LRAR, PRAR, PLAR, PbyLAR, and RRR respectively for replica $k = 2$ and for 225 nodes (5.8(a)).



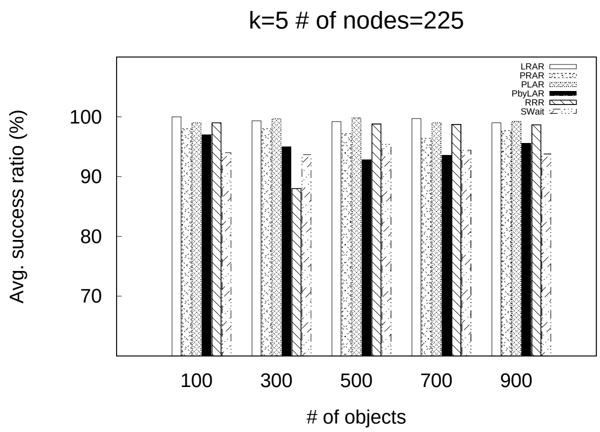
(a) Average success for k=2



(b) Average success for k=3



(c) Average success for k=4

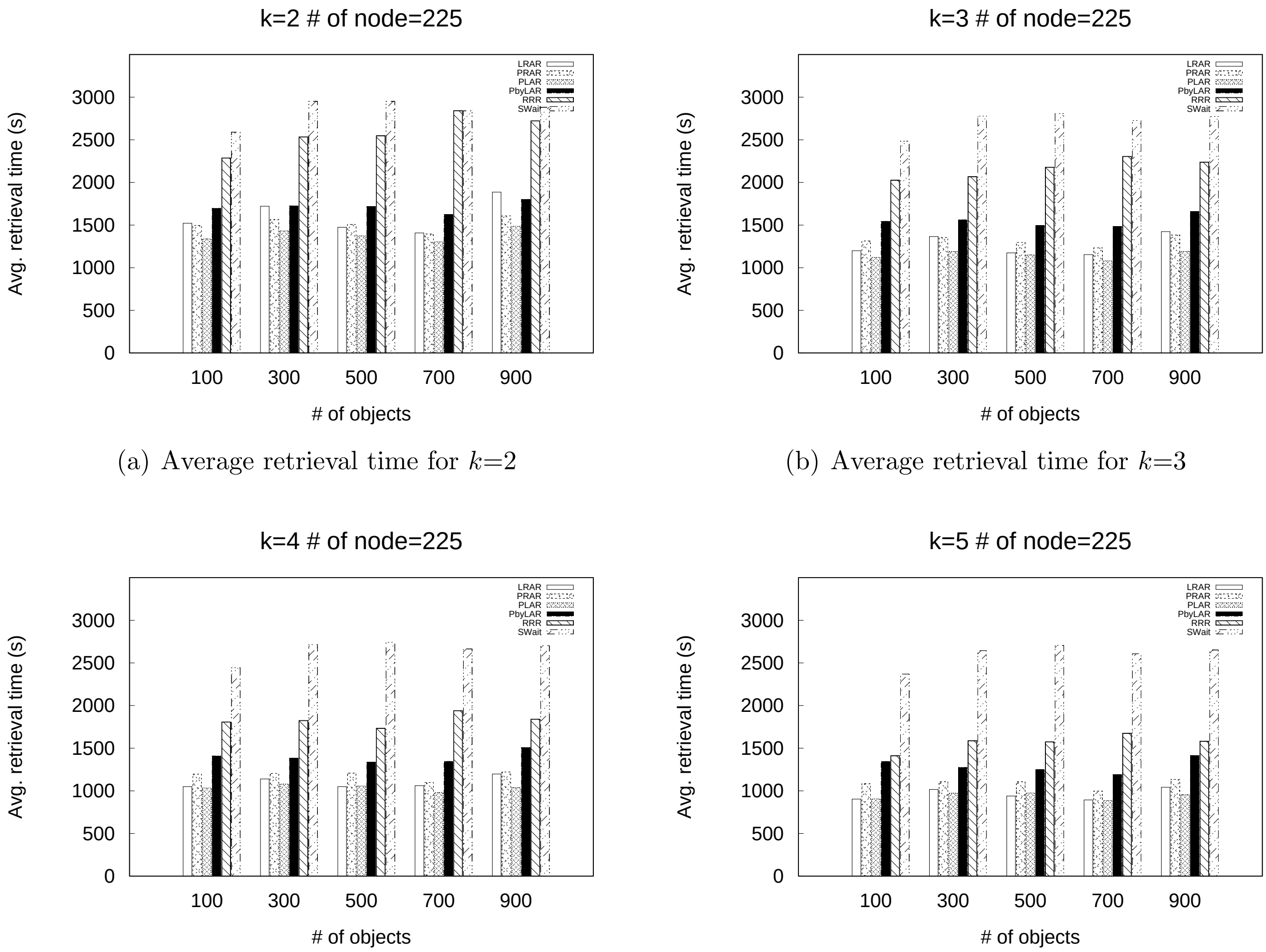


(d) Average success for k=5

Figure 5.8: Impact of varying # objects on average success ratio for different values of k

Fig. 5.9 illustrates the impact of variation of different number of objects on average retrieval time for different number of replicas k and for 225 nodes. We study the impact of number of nodes on the performance of different schemes by varying the number of objects using 100, 300, 500, 700, and 900.

It is evident from Fig. 5.9 the average retrieval time increases with an increase in the number of objects for different replicas for a fixed number of nodes. For example, for an increase of the number of objects from 100 to 500, the average retrieval time increases approximately 1.03 times, 0.99 times, 0.97 times, 0.98 times, and 0.89 times for LRAR, PRAR, PLAR, PbyLAR, and RRR respectively for replica $k = 2$ and for 225 nodes (5.9(a)). Moreover, Fig. 5.9(a) depicts that all our proposed schemes outperformed the state of the art approaches RRR and SWait with an increase in the number of objects.



(a) Average retrieval time for k=2

(b) Average retrieval time for k=3

(c) Average retrieval time for k=4

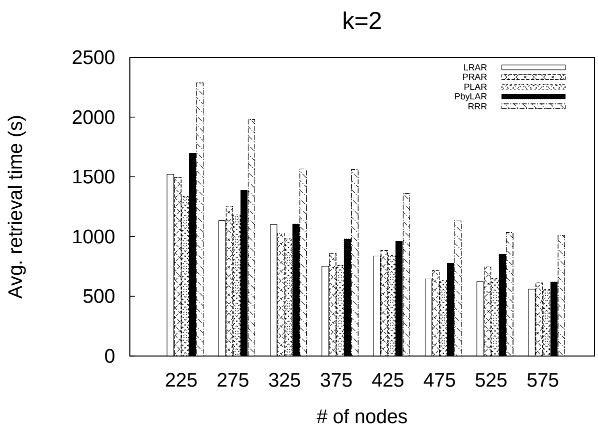
(d) Average retrieval time for k=5

Figure 5.9: Impact of varying # objects on avg. retrieval time for different values of k

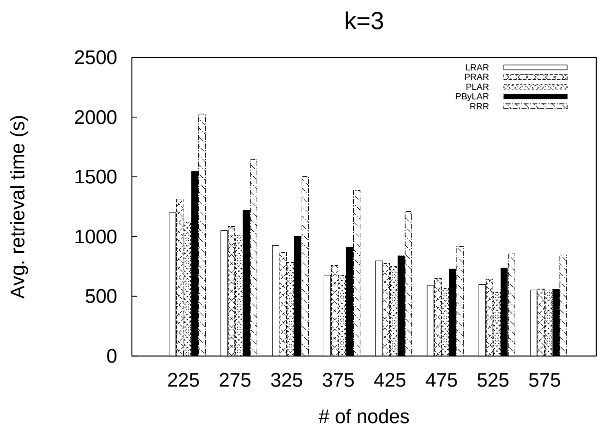
Besides, Fig. 5.9(b), Fig. 5.9(c), and Fig. 5.9(d) demonstrate the impact of different number of objects on retrieval time for $k = 2$, $k = 3$, and $k = 4$ respectively. Consequently, these figures also depict that the average retrieval time increases with an increase in the value of k (the number of replicas). The efficacy of our proposed schemes over the state of the art approach is also evident for different values of k.

It is evident from the results that our proposed schemes outperformed the traditional RRR approach for different values of k.

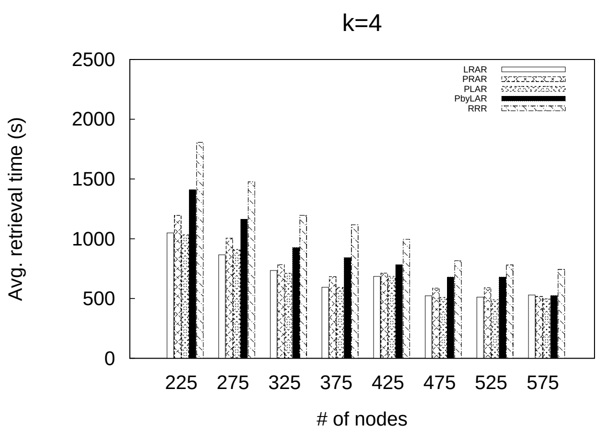
5.6.5 Impact of Variation of Different Number of Nodes



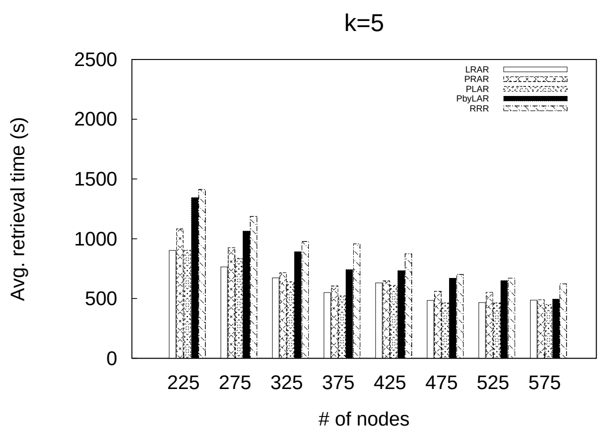
(a) Average retrieval time for k=2



(b) Average retrieval time for k=3



(c) Average retrieval time for k=4



(d) Average retrieval time for k=5

Figure 5.10: Impact of varying # nodes on avg. retrieval time for different values of k

We study the impact of number of nodes on the performance of different schemes by varying the number of nodes using 225, 275, 325, 375, 425, 475, 525, and 575 and measuring the average retrieval time. Fig. 5.10 illustrates the impact of variation of different number of nodes on

Table 5.4: Percentages of improvement for average success ratio

Parameters under variation	**Different values of parameter**	**% improvement of LRAR w.r.t.**		**% improvement of PRAR w.r.t.**		**% improvement of PLAR w.r.t.**		**% improvement of PbyLAR w.r.t.**	
		RRR	**SWait**	**RRR**	**SWait**	**RRR**	**SWait**	**RRR**	**SWait**
Number of replicas k	2	12.3	8.7	7.9	4.3	11.2	7.6	3.37	0
	3	5.2	7.5	2.1	4.3	4.2	6.5	-1.05	1.1
	4	2.04	6.4	-1.02	3.2	1.02	5.3	-4.08	0
	5	1.01	5.3	-1.01	3.2	0	4.2	-2.02	2.1
Number of objects	100	5.3	7.5	2.1	4.3	4.2	6.5	-1.05	1.1
	300	16.6	5.7	13.4	2.9	16.6	5.7	7.9	-2.2
	500	2.1	2.7	-0.6	0	3.3	4.0	7.11	-6.5
	700	2.7	4.1	0.15	1.5	3.5	4.9	-4.2	-2.9
	900	4.6	5.8	1.9	3.1	4.2	5.5	-2.36	-1.2
Retrieval deadline (hr)	0.5	61.5	162.5	23.1	100	23.1	100	15.4	87.5
	1	69.6	143.8	34.8	93.7	52.5	118.7	17.4	68.8
	2	55.0	106.7	40.0	86.7	55.0	106.7	17.5	56.7
	4	33.3	71.4	21.0	63.2	39.7	79.6	11.1	42.9
	6	24.0	40.9	17.3	33.3	26.7	4.4	6.7	21.2

average retrieval time for different number of replicas k using 100 objects.

It is evident from Fig. 5.10 the average retrieval time degrades with an increase in the number of nodes for different replicas. For example, for an increase of the number of nodes from 225 to 375, the average retrieval time decreases approximately 2.02 times, 1.73 times, 1.76 times, 1.73 times, 1.46 times for LRAR, PRAR, PLAR, PbyLAR, and RRR respectively for replica $k = 2$ (5.10(a)). The reason behind this: an increase in the number of nodes increases the meeting rate. Moreover, Fig. 5.10(a) depicts that all our proposed schemes outperformed the state of the art approaches RRR and SWait.

Fig. 5.10(b), Fig. 5.10(c), and Fig. 5.10(d) demonstrates the impact of different number of nodes on retrieval time for $k = 2$, $k = 3$, and $k = 4$ respectively. Consequently, these figures also depict that the average retrieval time decrease with an increase in the value of k (the number of replicas). The efficacy of our proposed schemes over the state of the art approaches is also evident for different values of k.

Finally, we delineate % improvement of our proposed schemes such as LRAR, PRAR, PLAR, and PbyLAR with respect to state of the art approaches RRR and SWait using Table 5.4 and Table 5.5 for average success ratio and average retrieval time respectively.

Table 5.5: Percentages of improvement for average retrieval time

Parameters under variation	**Different values of parameter**	**% improvement of LRAR w.r.t.**		**% improvement of PRAR w.r.t.**		**% improvement of PLAR w.r.t.**		**% improvement of PbyLAR w.r.t.**	
		RRR	**SWait**	**RRR**	**SWait**	**RRR**	**SWait**	**RRR**	**SWait**
Number of replicas k	2	33.5	41.3	34.6	42.2	41.6	48.4	25.7	34.5
	3	40.8	51.7	35.1	47.1	44.7	54.9	23.8	37.8
	4	42.0	57.1	33.8	51.1	42.8	57.7	22.0	42.3
	5	36.0	61.9	23.3	54.3	36.0	61.8	4.7	43.2
Number of objects	100	40.8	51.7	35.1	47.1	44.7	54.9	23.8	37.8
	300	34.0	50.9	34.4	51.2	42.5	57.3	24.4	43.8
	500	46.1	58.1	40.4	53.8	47.2	59.1	31.4	46.8
	700	50.0	57.7	46.5	54.8	53.1	60.4	35.7	45.6
	900	36.3	48.7	38.2	50.1	46.8	57.1	25.9	40.2

Table 5.6: % improvement of average retrieval time for different number of nodes

Parameters under variation	**Different values of parameters**	**% improvement of LRAR w.r.t. RRR**	**% improvement of PRAR w.r.t. RRR**	**% improvement of PLAR w.r.t. RRR**	**% improvement of PbyLAR w.r.t. RRR**
Number of nodes	225	40.8	35.1	44.7	23.8
	275	36.2	34.2	38.5	25.9
	325	38.4	42.3	48.0	33.4
	375	51.1	45.3	51.4	34.1
	425	34.1	35.8	37.9	30.5
	475	35.9	29.4	38.5	20.6
	525	30.0	24.8	37.7	14.0
	575	34.7	33.9	35.6	34.2

Chapter 6

Future Work

Because of the new dimension of this work and research phenomena in opportunistic challenges increasing day by day different aspects could be explored to view the impact of the contexts in the performance of schemes proposed.

In this thesis we focused primarily to see how PDM [44] mobility model effect in our proposed routing scheme for content retrieval. We focused on generic contents and keep all as same. Classification of contents such as popular or general are not considered in the current work. We would like to investigate, popular content retrieval in bounded wait time which will be an interesting phenomena to simulate in our proposed protocol using the recurrent mobility pattern in disaster scenario. Memory is not infinite in the mobile devices. Devices can focus for limited contents, having big size and also sometimes to make room for new contents need to remove some contents. Introducing some content replacement in the device memory to see how it will impact on the content retrieval.

Social context is a great deal in sharing contents. Similar people would like to share common contents. People with same community would like to share contents among themselves. We assume in our work that affected people in disaster strike area will share contents that will not impacting social context. We would focus this socio context in computing the availability definition of the neighbor node in the context.

We primarily focused upon mobility model PDM only. We will enhance our work to different mobility models so we could see the impact of socio-context upon mobility models, which will evaluate the performance in content retrieving in bound wait time.

We will try to implement our proposed solution in a low energy devices to get a hard bound upon our content retrieval from the adverse scenario to evaluate the performance with state approach techniques.

We would like to add different mobility models to our proposed protocol to see the experimented results efficacy over the real scenario. To see the content replication will not produce much redundant data in the network with the different mobility models.

As because the real tracing of disaster movement is hard to capture and simulate, the different protocols are not seeing in real scene. Our future goal is to deploy the proposed scheme in the real environment so that the smart mobile devices with high computability capability can create a resilient network by replicating contents aftermath a disaster strike.

Chapter 7

Conclusion

Delay tolerant networking is an emerging area of study for providing connectivity in scenarios where end-to-end connectivity is not normal. As DTN networks depend heavily on physical movement of devices that can buffer and exchange messages with other DTN devices, the behavior of a DTN would seem to depend on the underlying mobility assumed of the vehicles and people carrying DTN-enable devices. Disaster response networks (DRN) is a special kind of DTN network where in traditional communication means are unavailable due to the damage in infrastructure and power outage in the disaster strike area. To make the network more resilient in a disaster strike area, the communicating devices need to reproduce more and more valuable contents and replicate the contents to others to increase the availability of information. Recognizing this, we have developed a replication protocol for post-disaster scenarios, and extended the capabilities of the most widely used DTN simulator (ONE) to include the scheme.

We have formulated and analyzed the problem of choosing suitable replica for content replication in disaster response networks. We have work around to find the optimal replica set for the contents, which will minimize the delay of retrieving the content in a bound wait time. We considered the mobility model PDM, for the nodes to create recurrent contact among themselves. To model the recurrent contact process we uses stochastic approach *Markov* chain. Our proposed replication and retrieving protocol shows significant performance over the state art of randomized replication technique in the disaster strike area.

References

[1] Fall, K., Iannaccone, G., Kannan, J., Silveira, F., and Taft, N., "A disruption-tolerant architecture for secure and efficient disaster response communications," in *In ISCRAM*. Citeseer, 2010.

[2] Hosenball, M. (2012) "U.S. power, telecoms outages leave Americans vulnerable". http://in.reuters.com/article/usa-weather-storm-response-idinl2e8i2bep20120703. [Online; accessed 3-July-2012].

[3] Lorimer, S. (2016) "Power Cut affects 400 homes around Dundee". https://www.eveningtelegraph.co.uk/2016/07/04/power-cut-affects-400-homes-around-dundee/. [Online; accessed 4-July-2016].

[4] Writer, S. (2016) "Power outage affecting passengers, flights at Lambert Airport". http://fox2now.com/2016/07/03/power-outage-affecting-passengers-flights-at-lambert/. [Online; accessed 3-July-2016].

[5] Petras, T. (2016) "Power returned to more than 1,000 south Columbia residents". http://www.abc17news.com/news/. [Online; accessed 4-July-2016].

[6] Balasubramanian, A., Levine, B. N., and Venkataramani, A., "Replication routing in dtns: a resource allocation approach," *IEEE/ACM Transactions on Networking (TON)*, vol. 18, no. 2, pp. 596–609, 2010.

[7] Picu, A. and Spyropoulos, T., "Distributed stochastic optimization in opportunistic networks: the case of optimal relay selection," in *Proceedings of the 5th ACM workshop on Challenged networks*. ACM, 2010, pp. 21–28.

[8] Vahdat, A., Becker, D. *et al.*, "Epidemic routing for partially connected ad hoc networks," 2000.

[9] Spyropoulos, T., Psounis, K., and Raghavendra, C. S., "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. ACM, 2005, pp. 252–259.

[10] Keränen, A., Ott, J., and Kärkkäinen, T., "The one simulator for dtn protocol evaluation," in *Proceedings of the 2nd international conference on simulation tools and techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, p. 55.

[11] Jones, E. P. and Ward, P. A., "Routing strategies for delay-tolerant networks," *ACM Computer Communication Review (CCR)*, 2008.

[12] Jain, S., Fall, K., and Patra, R., *Routing in a delay tolerant network*. ACM, 2004, vol. 34, no. 4.

[13] Prueksasri, V., Intanagonwiwat, C., and Rojviboonchai, K., "Dnh-saw: the different neighbor-history spray and wait routing scheme for delay tolerant networks," in *Intelligent Signal Processing and Communications Systems (ISPACS), 2011 International Symposium on*. IEEE, 2011, pp. 1–5.

[14] Grasic, S., Davies, E., Lindgren, A., and Doria, A., "The evolution of a dtn routing protocol-prophetv2," in *Proceedings of the 6th ACM workshop on Challenged networks*. ACM, 2011, pp. 27–30.

[15] Fall, K., "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2003, pp. 27–34.

[16] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and Weiss, H., "Delay-tolerant networking architecture," Tech. Rep., 2007.

[17] Seligman, M., Fall, K., and Mundur, P., "Alternative custodians for congestion control in delay tolerant networks," in *Proceedings of the 2006 SIGCOMM workshop on Challenged networks*. ACM, 2006, pp. 229–236.

[18] Wang, R., Wei, Z., Zhang, Q., and Hou, J., "Ltp aggregation of dtn bundles in space communications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 3, pp. 1677–1691, 2013.

[19] Zhang, P., Sadler, C. M., Lyon, S. A., and Martonosi, M., "Hardware design experiences in zebranet," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM, 2004, pp. 227–238.

[20] Friedman, R., Kliot, G., and Avin, C., "Probabilistic quorum systems in wireless ad hoc networks," *ACM Transactions on Computer Systems (TOCS)*, vol. 28, no. 3, p. 7, 2010.

[21] Luo, J., Eugster, P. T., and Hubaux, J.-P., "Pilot: Probabilistic lightweight group communication system for ad hoc networks," *IEEE transactions on mobile computing*, vol. 3, no. 2, pp. 164–179, 2004.

[22] La, C.-A., Michiardi, P., Casetti, C., Chiasserini, C.-F., and Fiore, M., "Content replication in mobile networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 9, pp. 1762–1770, 2012.

[23] Hara, T., Loh, Y.-H., and Nishio, S., "Data replication methods based on the stability of radio links in ad hoc networks," in *Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on*. IEEE, 2003, pp. 969–973.

[24] Ganguly, S., Saxena, A., Bhatnagar, S., Izmailov, R., and Banerjee, S., "Fast replication in content distribution overlays," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 4. IEEE, 2005, pp. 2246–2256.

[25] Chen, Y., Katz, R. H., and Kubiatowicz, J. D., "Dynamic replica placement for scalable content delivery," in *International Workshop on Peer-to-Peer Systems.* Springer, 2002, pp. 306–318.

[26] Thanedar, V., Almeroth, K. C., and Belding-Royer, E. M., "A lightweight content replication scheme for mobile ad hoc environments," in *International Conference on Research in Networking.* Springer, 2004, pp. 125–136.

[27] Shinohara, M., Hayashi, H., Hara, T., and Nishio, S., "Replica allocation considering power consumption in mobile ad hoc networks," in *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06).* IEEE, 2006, pp. 5–pp.

[28] Balasubramanian, A., Levine, B. N., and Venkataramani, A., "Enhancing interactive web applications in hybrid networks," in *Proceedings of the 14th ACM international conference on Mobile computing and networking.* ACM, 2008, pp. 70–80.

[29] Balasubramanian, A., Mahajan, R., and Venkataramani, A., "Augmenting mobile 3g using wifi," in *Proceedings of the 8th international conference on Mobile systems, applications, and services.* ACM, 2010, pp. 209–222.

[30] Yan, T., Ganesan, D., and Manmatha, R., "Distributed image search in camera sensor networks," in *Proceedings of the 6th ACM conference on Embedded network sensor systems.* ACM, 2008, pp. 155–168.

[31] Gao, W., Cao, G., Iyengar, A., and Srivatsa, M., "Supporting cooperative caching in disruption tolerant networks," in *Distributed Computing Systems (ICDCS), 2011 31st International Conference on.* IEEE, 2011, pp. 151–161.

[32] Zhang, Y., Callan, J., and Minka, T., "Novelty and redundancy detection in adaptive filtering," in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 2002, pp. 81–88.

[33] Chen, H. and Karger, D. R., "Less is more: probabilistic models for retrieving fewer relevant documents," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 2006, pp. 429–436.

[34] Clarke, C. L., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Büttcher, S., and MacKinnon, I., "Novelty and diversity in information retrieval evaluation," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 2008, pp. 659–666.

[35] Johnson, D. B. and Maltz, D. A., "Dynamic source routing in ad hoc wireless networks," in *Mobile computing.* Springer, 1996, pp. 153–181.

[36] Rhee, I., Shin, M., Hong, S., Lee, K., Kim, S. J., and Chong, S., "On the levy-walk nature of human mobility," vol. 19, no. 3. IEEE Press, 2011, pp. 630–643.

[37] Hong, X., Gerla, M., Pei, G., and Chiang, C.-C., "A group mobility model for ad hoc wireless networks," in *Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems.* ACM, 1999, pp. 53–60.

[38] Musolesi, M. and Mascolo, C., "A community based mobility model for ad hoc network research," in *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality.* ACM, 2006, pp. 31–38.

[39] Hsu, W.-J., Spyropoulos, T., Psounis, K., and Helmy, A., "Modeling time-variant user mobility in wireless mobile networks," in *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications.* IEEE, 2007, pp. 758–766.

[40] Ekman, F., Keränen, A., Karvo, J., and Ott, J., "Working day movement model," in *Proceedings of the 1st ACM SIGMOBILE workshop on Mobility models.* ACM, 2008, pp. 33–40.

[41] Nelson, S. C., Harris III, A. F., and Kravets, R., "Event-driven, role-based mobility in disaster recovery networks," in *Proceedings of the second ACM workshop on Challenged networks.* ACM, 2007, pp. 27–34.

[42] Hui, P., Crowcroft, J., and Yoneki, E., "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, 2011.

[43] Daly, E. M. and Haahr, M., "Social network analysis for routing in disconnected delay-tolerant manets," in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing.* ACM, 2007, pp. 32–40.

[44] Uddin, M. Y. S., Nicol, D. M., Abdelzaher, T. F., and Kravets, R. H., "A post-disaster mobility model for delay tolerant networking," in *Winter Simulation Conference.* Winter Simulation Conference, 2009, pp. 2785–2796.