MCR: A Motif Centrality-based Distributed Message Routing for Disaster Area Networks

A Thesis Report

submitted in fulfillment of the requirements

for the award of the degree of

Master of Technology

in

COMPUTER SCIENCE AND ENGINEERING

by

Rajshekhar Khan

(320519003)

Under the supervision of

Dr. Nirnay Ghosh

Assistant Professor (Gr.-II)

Department of Computer Science and Technology

Indian Institute of Engineering Science and Technology, Shibpur

HOWRAH - 711103 WEST BENGAL, (INDIA)

CERTIFICATE

I hereby certify that the work which is being presented in the M.Tech. Thesis report entitled **"MCR: A Motif Centrality-based Distributed Message Routing for Disaster Area Networks",** in fulfillment of the requirements for the award of the **Master of Technology in Computer Science and Engineering** is an authentic record of my own work carried out during a period from August, 2020 to July, 2021 under the supervision of **Dr. Nirnay Ghosh, Assistant Professor (Gr.-II)**, Computer Science and Technology Department, Indian Institute of Engineering Science and Technology, Shibpur.

The matter presented in this thesis has not been submitted for the award of any other degree elsewhere.

Signature of Candidate

Rajshekhar Khan

Enrollment no - 320519003

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Signature of Supervisor

Date:

Dr. Nirnay Ghosh,

Assistant Professor (Gr.-II)

Acknowledgement

I gratefully acknowledge the resourceful guidance, active supervision, and constant encouragement of Dr. Nirnay Ghosh who despite his other commitments could find time to help me in bringing this report to its present shape. I do convey my sincere thanks and gratitude to him. I am also thankful to Prof. Asit Kumar Das, Head of the Dept. of Computer Science and Technology, IIEST, Shibpur for extending all facilities to carry out the present study. I am also thankful to our other faculties of the Dept. of Computer Science and Engineering, Shibpur for their support in all manner in this study. I also thankfully acknowledge the support and guidance received from Prof. Preetam Ghosh, Professor, Department of Computer Science, Virginia Commonwealth University, Richmond, Virginia, USA, Dr. Satyaki Roy, Postdoctoral Research Associate in the Department of Genetics, University of North Carolina, Chapel Hill, USA. I also thankfully acknowledge the assistance I received from my friends and others for their cooperation during the preparation of this report.

Rajshekhar Khan

Enrollment no. - 320519003

Abstract

Internet of Things (IoT) enables the collection of large volumes of data by billions of pervasive intelligent devices and sharing them with remote cloud servers for processing. Such massive volumes of data movement result in increased network congestion and server response times. The advent of edge computing has addressed these challenges by introducing an intermediate edge layer comprising networked fog nodes. Fog nodes provide on-demand compute, caching, and communication services to the applications running on end-devices and satisfy their critical QoS requirements. However, in a challenging environment brought by disaster and aftershocks, the QoS is hampered as several fog nodes in the edge layer are damaged. Nevertheless, the existing network infrastructure should still support the uninterrupted flow of time-critical contextual data between the survivors and rescuers for quick recovery operations. In this work, we envision that the IoT devices and existing fog nodes will collaborate to form ad-hoc networks for emergency message delivery under disaster situations. We present a distributed routing mechanism, termed *motif centrality-based routing (MCR)*, that leverages the concept of network motifs (subgraphs) seen in social and biological networks. Specifically, the proposed mechanism addresses three QoS requirements of an ad-hoc network - robustness against component failures, low latency, and energy efficiency. We analytically show that the MCR-based routing ensures high data delivery, low latency, and efficiency in energy usage. Finally, an extensive simulation-based study shows that MCR outperforms the related benchmarks in terms of the three QoS requirements.

Contents

Chapter 1

Introduction

1.1 motivation

The present-day network architectures thrive on *Information and Communication Technology (ICT)* to facilitate communication among billions of heterogeneous smart devices. Such devices are equipped with sensing, computing, and actuating capabilities and can support a wide range of communication protocols, viz., WiFi, Bluetooth Low Energy (BLE), Zigbee, 3G/4G/5G LTE, and so on. This network ecosystem, comprising devices such as smartphones, tablet computers, wearables, smart building and healthcare devices, etc. is termed the *Internet of Things (IoT)*. A plethora of applications run on top of IoT devices such as computer vision, speech recognition, natural language processing, augmented reality, interactive games, and so on. All these applications are delay-sensitive and energy-hungry. Hence, the quality of service (QoS) requirements for IoT applications are hard to achieve.

Due to the large scale and pervasive nature of the IoT paradigm, the applications running on these devices generate massive volumes of multi-modal data. These data are exchanged back and forth between the smart devices and cloud and content servers hosted in the core network. The massive volumes of data movement

increase the network congestion level, use up the bandwidth in back-haul networks creating bottleneck across the Internet, and also slow down the servers' responses. The exorbitant QoS demands are met by shifting from a centralized cloud-centric architecture to a distributed *edge computing framework*, which improves latency by introducing an edge/fog layer between the end device and cloud layers. Fog nodes are location-aware and provide on-demand computation, communication, and caching services to applications running on IoT devices (see Fig. 1.1). Henceforth, in the edge computing paradigm, end device requests are not directly sent to the back-end cloud/content servers but to the intermediate fog nodes, resulting in better bandwidth utilization for the back-haul networks.

Consider a scenario where a particular region leverages edge computing framework and the edge layer is implemented by several networked fog nodes. If this region is affected by a natural disaster (e.g., earthquake, cyclone, flood, etc.), the edge layer should enable the timely exchange of contextual information between the survivors possessing IoT devices and the control station (or base station). Nevertheless, in the event of disaster, a large fraction of the fog nodes are expected to be damaged, disrupting the exchange of emergency messages. It is evident that such failures in the network infrastructure can potentially cost human lives.

To counter the challenges of message exchange in a disaster-hit network, we envision a scenario where the user-owned devices (IoT devices) collaborate with the existing fog nodes in a distributed manner to build temporary ad-hoc networks. First, such a network must exhibit *robustness*, defined as the ability of the network to maintain seamless communication despite gradual component failures. Second, messages must be delivered to the rescuers with *minimum latency* to initiate timely relief or evacuation operations. Third, the routing mechanism must be *energy-efficient* and guarantee a prolonged network lifetime. Combining them, we need a *distributed and energy-efficient routing mechanism* for robust ad-hoc networks and low-latency data delivery under disaster situations.

We propose a priority-aware routing mechanism, termed *motif centrality-based*

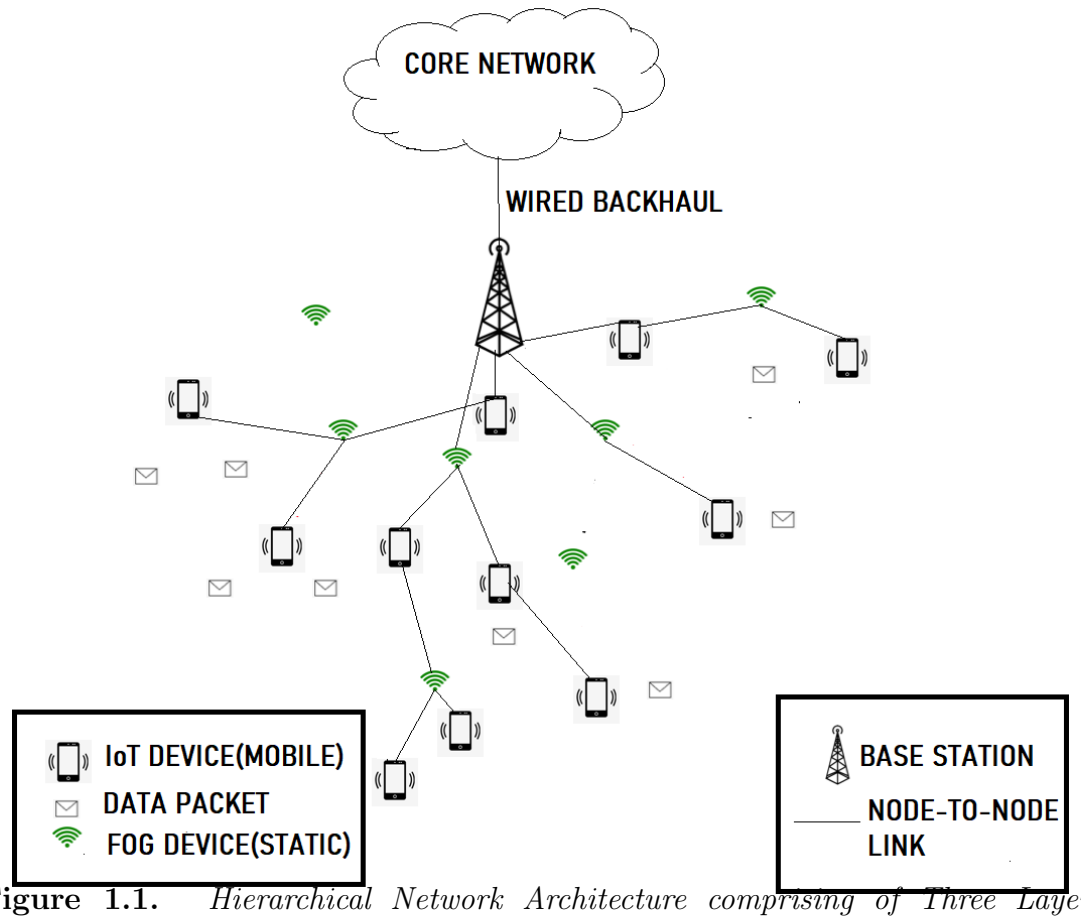


Figure 1.1. *Hierarchical Network Architecture comprising of Three Layers: End/IoT Device Layer, Edge Layer, and Cloud Layer*

routing (MCR) for distributed message delivery over a network infrastructure intermittently depleted by disaster and aftershocks. Our proposed solution addresses the three aforementioned requirements, namely, robustness, latency and energy efficiency and is based on the notion of *network motifs*, which are frequently occurring subgraphs and possess topological properties contributing towards robust information propagation in social and biological networks (refer Sec. 3.1). Each node that participates in building an ad-hoc network identifies motif-rich neighbors in its vicinity through a distributed message-passing protocol (from our earlier work [1]). Following this, the node selects robust and low-latency pathways to transfer an emergency message towards the base station. We prove that MCR, despite operating on 2-hop neighbor information, optimally routes data via the nodes with the highest motif participation to facilitate data delivery of high priority events. We carry out extensive simulation experiments to demonstrate that MCR exhibits ro-

bustness during post disaster situation with high packet delivery rate compared to the state-of-the-art benchmarks. Additionally, MCR outperforms others in terms of average latency for message transfer and overall energy consumption during the simulation window. In the following subsection, we present a brief literature review on two areas relevant to this work: (i) routing protocols in disaster environment and (ii) topological properties of motifs as well as their use in network design.

1.2 Related Works

Routing protocols in disaster area networks

Existing literature on data routing has primarily focused on the energy challenges in disaster area networks since low network lifetime may result in the inability to transfer time-critical information to the base station [2]. Uddin et al. leveraged the recurrent mobility and contact pattern of the rescue workers, volunteers, survivors, etc., to minimize the number of message replications and conserve energy [3]. Mukherjee et al. proposed a multicast routing protocol that achieves energy-efficient self-organizing routing at low communication overhead [4]. Shah et al. employed dynamic clusters managed by well-connected cluster heads to cut off redundant communication [5]. Other works like DRNs (and DTNs) [6–8] attempt to achieve high packet delivery in addition to energy efficiency. Huang et al. and Chen et al. have proposed routing via sparse network connectivity as an efficient approach to combat intermittent connectivity as well as node failure due to energy depletion, hardware faults. etc. [9, 10] .

Motifs in bio-inspired network design

Motifs serve as a basis for the analysis of complex networks like biological networks. For example, four-node motifs contribute to nutrient metabolism and bio-synthesis in biological networks [11, 12] and three-node motifs are filters, pulse generators, and response accelerators [13]. Kashtan et al. and Gorochowski et al. explored the

organizational principle of motifs and how motifs aggregate to form higher order motifs [14, 15]. Roy et al. analyzed the information dissemination potential of motifs and their implications in the design of routing protocols in communication networks [16]. Kosyfaki et al. presented a network centrality metric, termed *motif-based centrality* (discussed in Sec. 3.1), that quantifies the importance of a node in terms of its motif participation [17]. This metric motivated the design of bio-inspired network protocols [18–21], delay tolerant networks [22], and edge computing frameworks [1]. Experimental analysis demonstrated that these networks are both Quality-of-Service (QoS) and energy efficient.

The rest of the paper is organized as follows. In Section 2 we present the system model. Section 3 elucidates the the motif centrality-based routing mechanism. Performance analysis and validation of the proposed mechanism are done in Section 4. Finally, in Section 5, we draw the conclusions and identify the future research directions.

Chapter 2

System Model

We consider an urban space which has hundreds of users moving around with IoT devices (e.g. smartphones, tablets, wearables, etc.), equipped with multiple sensors and have communication capabilities. The user devices constitute an *end device layer*. They communicate with the cloud (for compute service) and content (for content delivery) servers hosted in the core network via an *edge device layer* which comprises of multiple networked fog nodes.

The fog nodes are deployed at the edge and offer computation, communication, and caching services to the IoT devices. Such services allow users to offload their computation tasks to the nearest fog node and also fetch Internet contents with less delay. The edge layer bypasses the access to back-haul networks and saves considerable bandwidth compared to service request/delivery to cloud and/or content servers. Under a disaster scenario (like earthquake, cyclone, flood, etc.) networking infrastructure is expected to be heavily damaged, leading to the disabling of a majority of the fog nodes.

In that situation, the mobile IoT devices build ad-hoc networks with themselves and with the existing fog nodes to relay messages to a nearby base station. Essentially, the disaster survivors sense the environment and generate emergency messages (viz., rescue messages, warning messages, environmental messages, healthcare, transportation and safety messages etc.) using their IoT devices. Within the ad-hoc

network, the messages are to be routed along "optimal" paths to the base station so that rescue operations can be initiated in a timely manner. On most occasion, these paths have multiple hops, where each hop constitute either an IoT-IoT or an IoT-fog message exchange. It is evident that due to random mobility of the survivors and ad-hoc nature of the network, packet losses are inevitable. As a result, emergency messages may not reach the base station and affect the rescue operation.

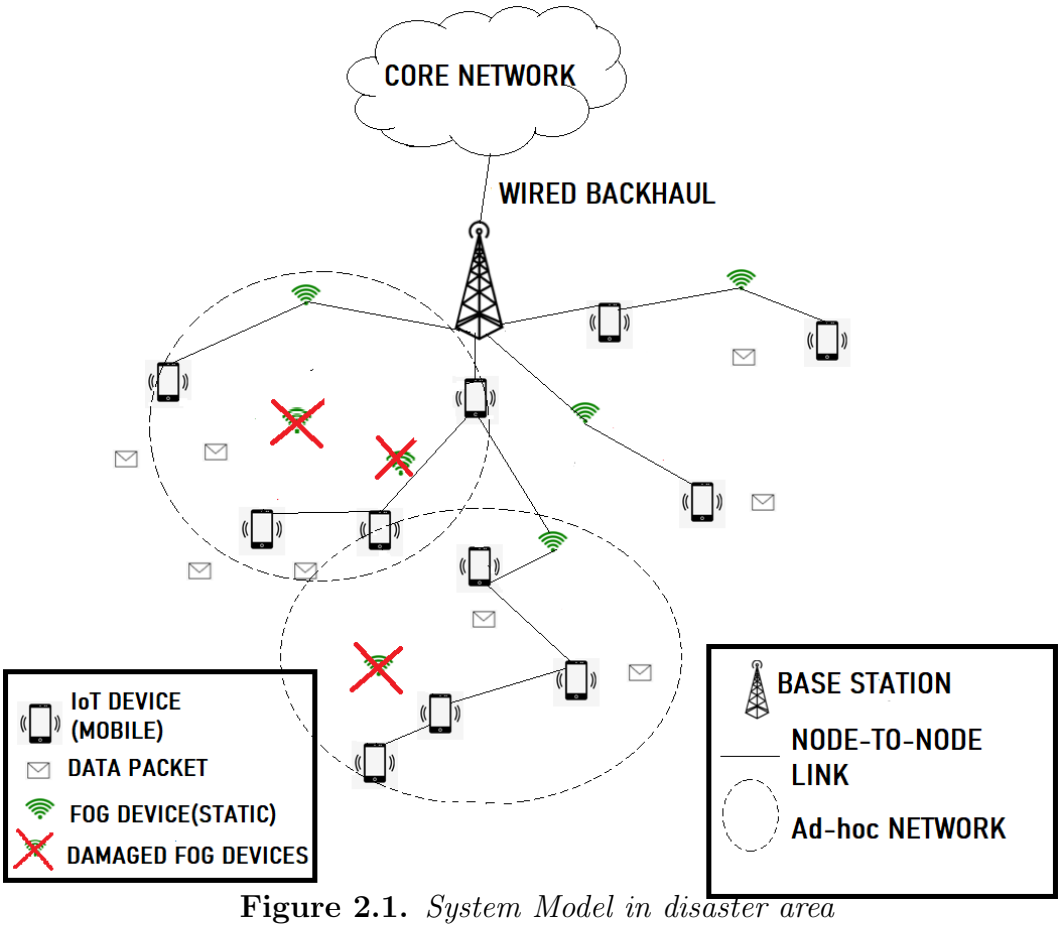


Figure 2.1. *System Model in disaster area*

We propose a scenario where each IoT device or fog node can locally discover its neighbors, and choose the one offering the most robust pathway for message delivery. This ensures that the message always gets routed through a path that consists of only reliable intermediate nodes (IoT device/Fog node) such that its delivery is guaranteed, despite component failures.

Fig. 1.1 shows an urban space (that is prone to frequent natural disasters) where multiple fog nodes $\mathcal{F} = \{f_1, f_2, f_3, \cdots\}$ and IoT devices $\mathcal{D} = \{d_1, d_2, d_3, \cdots\}$ comprise a network. The fog nodes are microservers that can be realized by network

edge devices such as WiFi access points, routers, cloudlets, commodity hardware, and so on. They are also equipped with specialized software to provide on-demand computation and caching services. The IoT devices are owned by users who move around the urban space following certain mobility patterns. The system model also consists of a *base station* which is responsible for delivering the user request to cloud or content servers hosted in the core network via wired back-haul. We assume that both fog node and IoT device have the MCR middleware installed locally to support communication under crisis situation.

Fig. 2.1 illustrates our system model under post-disaster scenario where a significant number of fog nodes are damaged and incapable of providing the services. In this situation, the IoT devices and existing fog nodes build multiple ad-hoc networks and exchange information using the MCR middleware. The major components of the MCR middleware are discussed below.

1. Fog node: We define a fog topology $\mathcal{F}$ that is managed by network or service operators. In this topology, nodes are static and capable of two-way communication with neighboring nodes or IoT devices if they are in range R_f. Each fog node $f_i \in \mathcal{F}$ has routing, processing, and storage capabilities. Besides communication service, they can act as local processing units capable of location-aware and low-latency computing. Fog nodes serve as intermediaries between the IoT devices and base station (BS). We assume that the service provider will install MCR middleware in the fog nodes to facilitate message exchange under disaster situation. The fog nodes use IPv4/IPv6 protocols over Wifi frames to send messages to each other and to the base station.

2. IoT device: An IoT device $d_j \in \mathcal{D}$ is an energy-constrained, handheld device (e.g smartphone, notebook, tablet computers, wearables etc.), has low communication range (R_d) than any fog node ($R_d << R_e$), has limited storage (buffer) and computing, and is owned by a user/survivor. Owing to the users' mobility, these devices can change their positions with time and have installed the MCR middleware application locally. The IoT devices will either sense the environment opportunistically

or may be used by their owners to generate data. Under disaster situation, the IoT devices build ad-hoc networks along with the existing fog nodes to exchange emergency messages. Within the ad-hoc networks, IoT devices leverage device-to-device (D2D) communication protocol (viz., Bluetooth v5.0) to interact with each other and use normal IPv4/IPv6 protocols over Wifi frames to communicate with fog nodes and the base station. The local MCR middleware enables each node to forward message to the 'best' next hop node (fog node or IoT device) such that the data delivery to the base station is guaranteed.

3. Base Station (BS): A base station (BS) can be envisioned as a cellular tower (owned by a mobile operator) whose primary task is to deliver messages back-and-forth between IoT device users and cloud and/or content servers hosted in the core network via the wired back-haul. Its communication range R_b is much higher compared to that of the fog nodes (R_f) and IoT devices(R_d), and it uses the backbone Wireless LAN network (formed by the fog nodes) for bidirectional message exchange with the devices. The base station is not energy-constrained, has unlimited storage capacity, and is expected to remain operational after the disaster.

4. Mobility of users: We model the mobility of the users (owners of IoT devices) using the random waypoint mobility model. A device halts in one location for a brief period, defined as *pause time*. It then chooses a random destination (within the deployment area) as well as a speed v_t ($v_{min} \leq v_t \leq v_{max}$) and travels towards the newly chosen destination at the selected speed. Upon arrival, the device takes a pause before repeating the same step.

5. Message: A message $m_k \in \mathcal{M}$ is a piece of information that an IoT device d_j generates either by sensing the disaster-affected environment (data message) or to communicate with other nodes/devices (control message). Throughout the text we will interchangeably use the terms 'message', 'data', and 'packet' to refer to a piece of information. The two variants of message used in our approach are as follows:

- Data message: Each message is associated with an identifier and a geographical location. In general, it is an alert which gets sensed and generated by a

survivor using the IoT device s/he owns at a location within the disaster area. Few examples of data messages are: intimation about survivors stuck in the disaster, general help request, areas where rescue operations have not started, etc. We divide data messages into three priority classes based on their significance w.r.t the disaster situation: (i) *high*; (ii) *medium*; (iii) *low*. To introduce data freshness and to avoid network congestion, we add the *time-to-live (TTL)* property in messages. Thus, to allow delivery of high priority message we assign it with highest TTL value, while low priority message gets lowest TTL value.

- Control message: These messages are exchanged frequently between IoT devices and existing fog nodes for the management of the ad-hoc networks. Primarily, control messages are used by the devices to determine the current neighbors and use this information to build the network topology. Some of the important control messages are node motif centrality score, location of the node, available buffer capacity etc.

6. Time epoch: We introduce the concept of time epoch, τ, to delineate different processes taking place periodically in each cycle of routing. It is a configurable parameter, whose duration (in terms of time units) depends on the frequency of information collected by IoT devices. We use the quasi-static assumption that the network situation remains unchanged during one time epoch. The mobility of the IoT devices takes place between two time epochs. At each time epoch, the following processes take place: (i) each IoT device senses the environment and generates message; (ii) it creates its neighbor list and collect neighbor information; (iii) it sends the messages to the 'best' next hop neighbor (either an IoT device or a fog node) and this continues till the message is delivered to the base station.

7. Disaster epicenter: In a disaster (earthquake) situation, an epicenter is the source and it is the worst-hit area. Intensity of the disaster decreases as the distance increases from the epicenter. As a result, damages inflicted to environment and public infrastructure are less as the intensity weakens away from the epicenter. In

our context, we ascribe infrastructure to networking facilities (comprising of base station and networked fog nodes) and assume that the degree of damage (disabling of fog nodes) is proportional to the distance from the epicenter. Thus, if $p(f_i)$ is the probability that fog node f_i will be damaged, it is given as $p(f_i) = \frac{\delta(f_i)}{R}$, where $\delta(f_i)$ is the distance of f_i from the epicenter and R is the distance from the epicenter to the farthest fog node in the considered urban space.

8. Middleware: This is one of the most important entities of this system model. We assume that the motif centrality-based routing (MCR) mechanism is packaged as a middleware and implemented locally in all the fog nodes and IoT devices. The motivation for installing the middleware in user-owned IoT devices is to facilitate sending and receiving of emergency/rescue messages during and after any disaster. The MCR middleware enables the devices and nodes to exchange control messages and build ad-hoc networks in scenarios where the networking infrastructure is only partially operational. It helps a node (IoT device or fog node) to prioritize messages in disaster situation, finds its two-hop neighbors, runs a distributed algorithm to select the 'best' next hop neighbor for message transfer, and manages local buffer for store and forwarding of the data.

Chapter 3

Motif Centrality Based Routing Mechanism

As mentioned in Sec. 1, the motifs are recurring subgraphs in complex networks (see Fig. 3.1(a)). They possess a range of functional properties associated with information propagation in complex networks [23]. We focus on one such motif – called the *Feed Forward Loop (FFL)* – an acyclic triangle. As shown in Fig. 3.1(b), it has a direct link between the source node S and the target node T, and an indirect path via I. Besides being the building blocks of complex networks [24], the abundance of FFL motifs enhances the topological robustness of networks in two ways [25]. (1) There are two paths connecting nodes S and T. Since the *minimum number of vertices whose removal disconnects two vertices is equal to the maximum number of pairwise vertex-independent paths between them* [26], the multiplicity of paths between S and T contributes towards network robustness against node and link failures. (2) The elimination of the direct $S \rightarrow T$ link in $FFLs$ increases the path length between S and T by a single hop only, minimizing the communication delay due to failures. In this section, we will use notations u, v, and w to denote a node (which may be either an IoT device or a fog node) in a disaster area network.

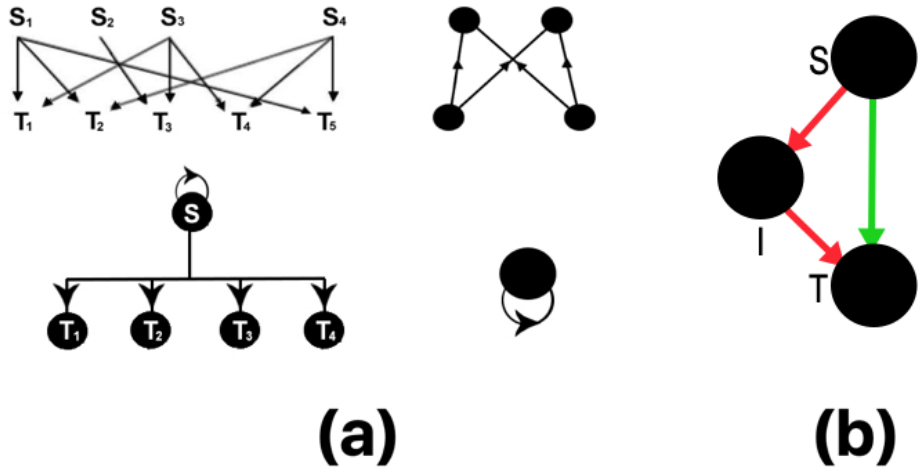


Figure 3.1. *Network motifs: (a) Standard network motifs; (b) Feed Forward Loop motif with direct and indirect path from source S and target T marked green and red, respectively.*

3.1 Node Motif Centrality

The importance of a node in a complex network is gauged in terms of its motif participation. We define *node motif centrality (NMC)* as the number of FFL motifs in which it participates. Given a directed graph $G(V, E)$, the NMC of node u is:

$$\Delta(u) = \sum_{v,w \in V} \xi(u, v, w) + \xi(v, u, w) + \xi(v, w, u) \tag{3.1}$$

Here, $\xi(u, v, w)$, $\xi(v, u, w)$ and $\xi(v, w, u)$ are the indicator variables showing possible existence of an FFL motif between vertices u, v and w, where $\xi(u, v, w) = 1$ if $e(u, v), e(v, w), e(u, w) \in E$, otherwise $\xi(u, v, w) = 0$.

3.2 Distributed Motif Centrality Protocol

The goal of this protocol, introduced in [1], is for each node u to acquire and maintain the information of the neighbor and neighbor-of-neighbor devices and employ this information to calculate the neighbor FFL motif centrality. The 2-hop information is maintained by u in the form of a network H_u.

At the start of each epoch, H_u is initialized with a single node u and updated subsequently. Node u also tracks the residual energy res of the 1-hop neighbors v,

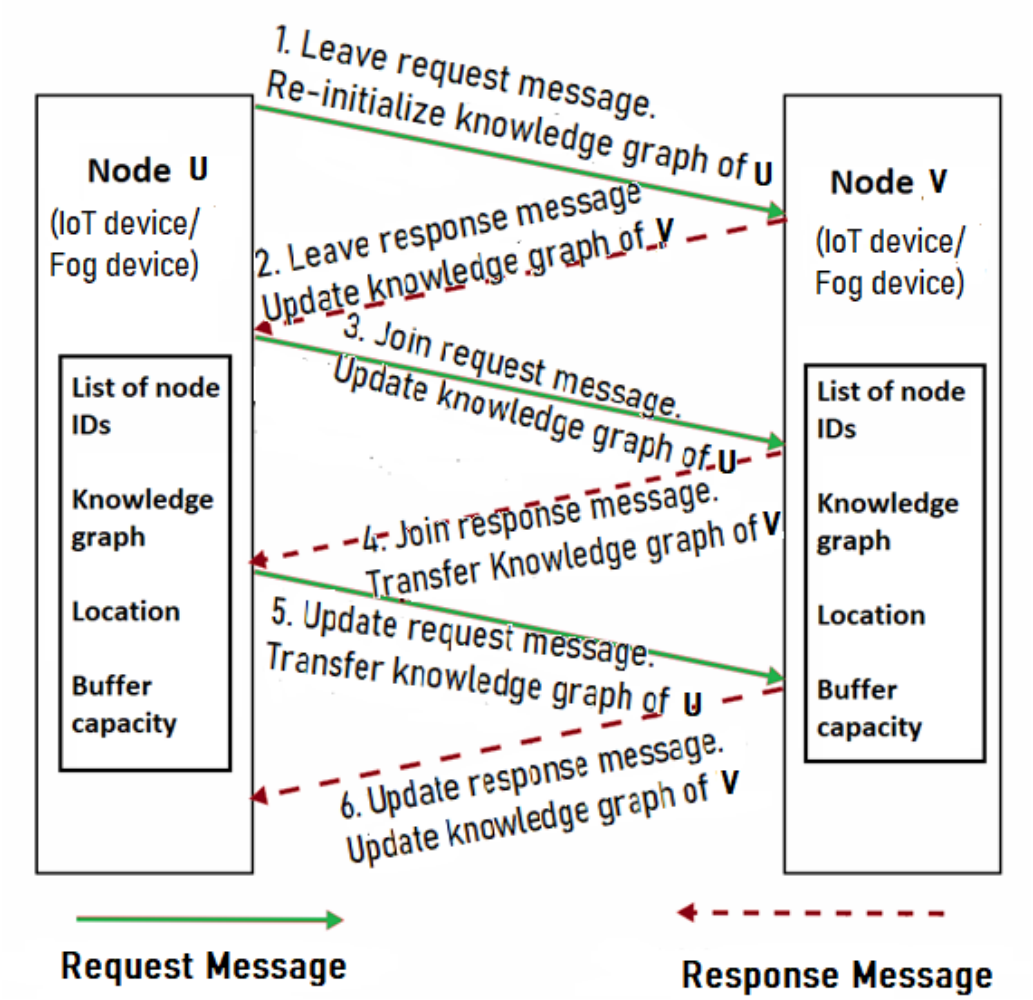


Figure 3.2. *Message Exchanges during NMC*

denoted by $\mathbf{W}(v)$.

1. When u moves to a new position, two steps are observed: (i) Node u drops a *leave message (LReq)* of format $\langle LReq, u\rangle$ to all 1-hop neighbors $v \in V(H_u)$, requesting them to drop u from their neighbor-list; (ii) Node v receives the $LReq$ and removes u and its adjacent links from H_v.

2. When u arrives at a new location, the new neighborhood is created as follows: (i) Node u re-initializes H_u, scans its new neighbors and drops a *join message (JReq)* of format $\langle JReq, u, res_u\rangle$ to the 1-hop neighbor v, while adding an edge (u, v) to H_u, i.e., $E(H_u)) = E(H_u)) \bigcup \{(u, v)\}$; (ii) Node v, on receiving a $JReq$, adds (u, v) to H_v and sends a *join response (JRes)* (of format $\langle JRes, v, res_v, N_v\rangle$, where N_v is the list of 1-hop neighbors in H_v), informing

u of its 1-hop neighborhood; (iii) Node u, upon receipt of the *JRes*, updates the residual energy of v as $\mathbf{W}(v) = res_v$ and adds edge (u, v) in H_u.

3. Each node must periodically update its 2-hop neighborhood information. To this end, (i) node u sends out an *update request (UReq)* of format $\langle URes, u, N_u, res_u \rangle$ to its 1-hop neighbors $v \in N_u$; (ii) Node v, upon receiving *UReq*, updates 1-hop neighborhood information of u, by discarding 1-hop neighbor links of u with prior neighbors $\omega_{old} \in N_u^{old}$ in H_v, i.e., (u, ω_{old}) and adding links with current neighbors $\omega_{cur} \in N_u$, i.e., (u, ω_{cur}).

With the knowledge of the 1- and 2-hop neighbors, each node u employs Eq. (3.1) to calculate the FFL motif centrality of neighbors $v \in N_u$. The motif centrality $\Delta(v)$ $(\forall v \in N_u)$ is utilized by node u to determine the next-hop during the motif centrality-based routing (discussed hereafter).

3.3 Greedy Motif Centrality-based Message Routing

Each node employs the distributed approach (discussed in Sec. 3.2) to calculate the FFL centrality of its neighbor nodes.

Data Forwarding

The motif centrality-based routing (MCR) protocol unfolds in three steps on the basis of the FFL motif centrality in conjunction with (1) proximity to the base station and (2) buffer availability of the neighbors as discussed hereafter.

- *Proximity.* To ensure data message is routed to the base station, at a particular hop, the local MCR middleware leverages the spatial information of the next-hop neighbors obtained before routing. MCR selects only those nodes (IoT device/fog node) as the potential next hop neighbor that are spatially close to

the base station. This results in shorter routes (i.e. less communication delay) and energy saving during the data transmission.

- *Robustness.* MCR uses a greedy strategy and selects the hop with 'highest motif centrality score' as the next hop neighbor from the set of nodes short-listed in the previous step. The node with highest motif centrality score is most robust as it is connected to the maximum number of adjacent nodes giving multiple alternate pathways to reach the base station. Hence, even if a random link failure occurs, the MCR-chosen next hop node can still deliver the message to its neighbor. Thus, MCR ensures guaranteed delivery of the message to the base station.

- *Traffic load balancing.* If two or more nodes have similar motif centrality scores and are also spatially close to the base station, MCR breaks the tie by choosing the node whose current buffer capacity is maximum.

As illustrated in Fig. 3.3, the current node C exploits local information of FFL motif centrality to propel the data along the most reliable path (marked black) towards the base station.

Data and buffer management

At each hop, the message with high priority is sent first, following which the medium and low priority messages are delivered, respectively. If a node (IoT device/fog node) has multiple messages to be forwarded towards the base station, it sorts the messages in descending order of their priorities. Furthermore, if multiple high priority messages have arrived at the same time, MCR selects the one with smallest time-to-live (TTL) value.

Next, the device selects its next hop neighbor based on the MCR protocol discussed above. The node (along with other nodes) sends data to the selected next hop neighbor until the latter's buffer does not overflow. It is to be noted that even if buffer overflow occurs the senders continue to pump-in messages as MCR does

not support buffer capacity feedback in the middle of a time epoch (to ensure low latency in message delivery). During the start of the new time epoch, the next hop node shares its buffer status with the current senders. Following this, the local MCR middleware on the sending nodes will select the node with second highest motif centrality as the next hop for routing. This process continues across multiple time epochs till either all messages are delivered to the base station or no node with available buffer space remains.

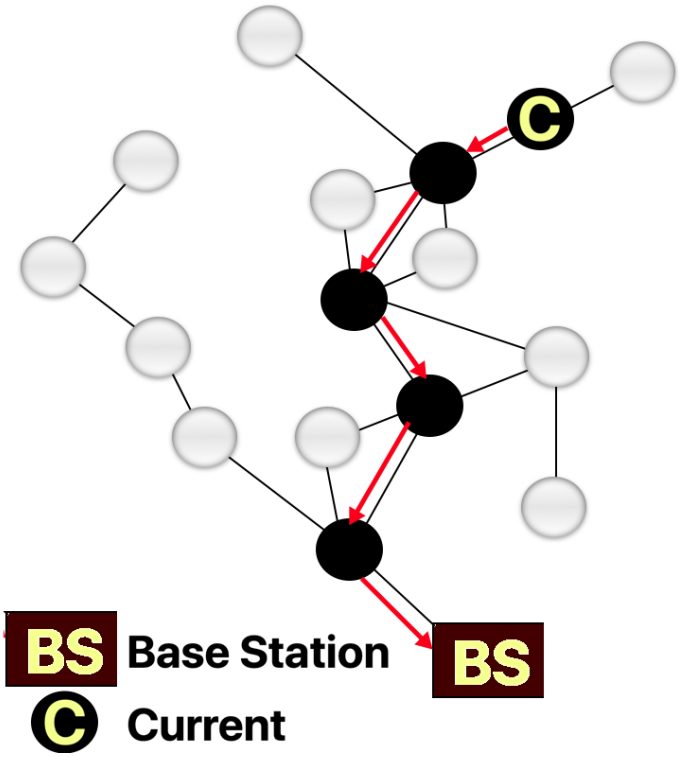


Figure 3.3. *Greedy Motif Centrality-based Message Routing*

Optimizing event data priority. Each node u ranks the messages $m_k \in \mathcal{M}$ in the increasing order of the priority score (i.e, non-increasing order of priority). Each message in the ranked list is transferred to the highest FFL motif central neighbor $v \in \mathcal{N}(u)$ with partially empty buffer.

Let us consider a node $u \in V$ with neighbor set $v \in \mathcal{N}(u)$ and event set $\epsilon \in \mathcal{E}$. Each node u has a FFL motif centrality M_u and each message m_k has a priority score $\mathcal{P}(m_k)$ (low score implies higher message priority). We define a cost function C (Eq. 3.2) that maximizes the transfer of the highest priority message (low priority score)

from each node u to the highest motif central neighbor node v.

$$C = \max \sum_{m_k \in \mathcal{M}} \sum_{v \in \mathcal{N}(u)} \frac{1}{\mathcal{P}(m_k) \times M_v} \tag{3.2}$$

Lemma. The greedy motif centrality-based message routing yields the optimal cost C.

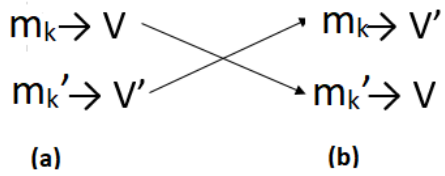


Figure 3.4. *Optimizing message priority: (a) Assigning highest priority message to the highest node motif central node and (b) vice-versa.*

Proof. Let us consider two messages m'_k and m_k in the possession of a node u, such that $\mathcal{P}(m'_k) < \mathcal{P}(m_k)$. Node u has neighbor nodes v' and v, such that $M_{v'} > M_v$. As per the greedy message routing approach, let node u transfer the highest priority message m'_k to the highest motif central neighbor v'; message m_k is sent to v (Fig. 3.4(a)).

To test the optimality, let us flip the neighbor-message order. Contrary to the NMC protocol, node u transfers the highest priority message m'_k to node v and message m_k is sent to v' (Fig. 3.4(b)). The change in cost function due to this flip is:

$$\Delta_C = [-\frac{1}{\mathcal{P}(m'_k)} \times M'_v + \tag{3.3}$$

$$\frac{1}{\mathcal{P}(m'_k)} \times M_v] + [-\frac{1}{\mathcal{P}(m_k)} \times M_v + \frac{1}{\mathcal{P}(m_k)} \times M_{v'}] \tag{3.4}$$

$$\implies \frac{1}{\mathcal{P}(m_k)}[M_{v'} - M_v] - \frac{1}{\mathcal{P}(m'_k)}[M_{v'} - M_v] \tag{3.5}$$

$$\implies [M_{v'} - M_v] \times [\frac{1}{\mathcal{P}(m_k)} - \frac{1}{\mathcal{P}(m'_k)}] < 0 \tag{3.6}$$

Inequality (3.6) holds since $[M_{v'} - M_v] > 0$ and $[\frac{1}{\mathcal{P}(m_k)} - \frac{1}{\mathcal{P}(m'_k)}] < 0$. The change in cost due to a flip cannot be positive. Thus, the greedy solution yields optimal cost C. Hence proved. □

Chapter 4

Results and Validation

We discussed earlier that the IoT devices and fog nodes participate to build an ad-hoc network for sending emergency messages to the base station. Such message exchanges are regulated by the MCR middleware installed locally on the IoT devices and fog nodes. MCR supports multiple communication types: IoT device → IoT device, IoT device fog node, fog node → base station, and IoT device → base station. To implement these communications, we developed a custom simulation environment enabled with real-time visualization based on Python *Simpy* library. For performance analysis of the MCR middleware, we consider the following metrics:

1. Packet Delivery Rate (PDR): It is computed as the ratio of the number of unique data messages delivered at the base station to the total number of unique data messages generated by all IoT devices.

2. Communication latency: It is measured in terms of the time taken by a particular message to travel from the source IoT device to reach the base station.

3. Energy efficiency: It is gauged in terms of the average energy consumption of the IoT devices in the ad-hoc network at the end of the simulation. Below we present the simulation environment for our work.

We consider the following routing benchmarks for comparison:

1. *Flooding* [6]: In this approach, the message is simply shared with every possible neighbor. Thus, data is sent through all possible paths including the best path.

Two disadvantages in this approach are: (i) several clone copies of the same message are being produced by each node, which leads to increased congestion level in the network, and (ii) as the node is sharing data with all possible neighbors, overall network energy consumption is on the higher side.

2. *PROPHET* [27]: Here, with every successful message delivery to the base station, a feedback is reverted back to the intermediate nodes indicating the possible successful path for the data delivery. So when a node has some messages to send towards the base station, if the node has found a successful path, it will select the next node from the successful path-node list to transfer 50% messages, and simply broadcasts the remaining 50% data to all possible neighbors. One disadvantage of this model is that, though we have listed the possible successful paths to base station, but due to mobility of the intermediate IoT devices, it is possible that some of the successful paths do not exist at the time of actual data transfer.

3. *Wireless Routing Protocol (WRP)* [28]: It is based on shortest path routing. In the *WRP* approach, each node maintains a distance table based on the Bellman-Ford algorithm and updates the list of neighbors through periodic update messages, and thereby has global and most recent knowledge of the shortest path to the base station. Each node also has the location and the available buffer information for its neighbors. As the *WRP* represents the 'best case' data routing scenario, our objective is to analyze how MCR model fairs in its comparison and elucidate the obvious benefits of the latter's decentralized nature.

4.1 Simulation Environment

This section is organized as follows. First, we describe a procedure for generating an urban space, bounded by latitude and longitude and uniformly distribute the networking infrastructures (base station, fog nodes, IoT devices) across it. Second, we simulate damages on the infrastructures followed by a disaster and after-shock effects. Third, we simulate message generation events by tuning different system parameters. The default values of the other significant parameters are summarized

in the parameter table (Table 4.1).

Table 4.1: Default Simulation Parameters

Parameter	**Symbol**	**Value (default)**
Simulation region	Lat., Long.	40.5 - 41 N, 73.7 - 74.5 W
Duration of an epoch (time units)	τ	5
Duration of simulation (time units)		180
Time unit at which disaster occurs	-	60
Time unit at which aftershock occurs	-	120
No. of messages per epoch	$n\mathcal{M}$	25
No. of fog nodes	$\|\mathcal{F}\|$	10
No. of IoT devices	$\|\mathcal{D}\|$	40
Mobility range	r	Lat.$\pm$0.01, Long.$\pm$ 0.01
Comm. range of fog and IoT device	R_f, R_d	300, 150 m.
Sensing radius for IoT device	R_s	150 m.
Initial energy for IoT device, fog node	-	500, 1000 J.
Energy baseline	-	10 J.
Power for sensing	-	0.05 W.
Power for scanning fog/IoT device	-	3.68 W.
Power for data receipt	-	0.31 W/packet.
Power for data transfer from IoT device	-	0.137 W
Power for data transfer from fog node	-	0.37 W

4.1.1 Simulation Area Generation

We deploy multiple fog nodes (denoted by green dots) and a base station (denoted by blue dot) on a simulation area (refer to Fig. 4.1) to depict the initial networking architecture in that region. The simulation area also has several active IoT devices owned by users who are mobile. As this captures a pre-disaster scenario, all the fog nodes are currently operational. The latitude and longitude for this area are generated using the Python library for geo-coding services, called *GeoPy* and it captures the map of a portion of the New York City (NYC). In this simulation area, the approximate distance between any two latitudinal and longitudinal points are 69 and 54.6 miles, respectively.

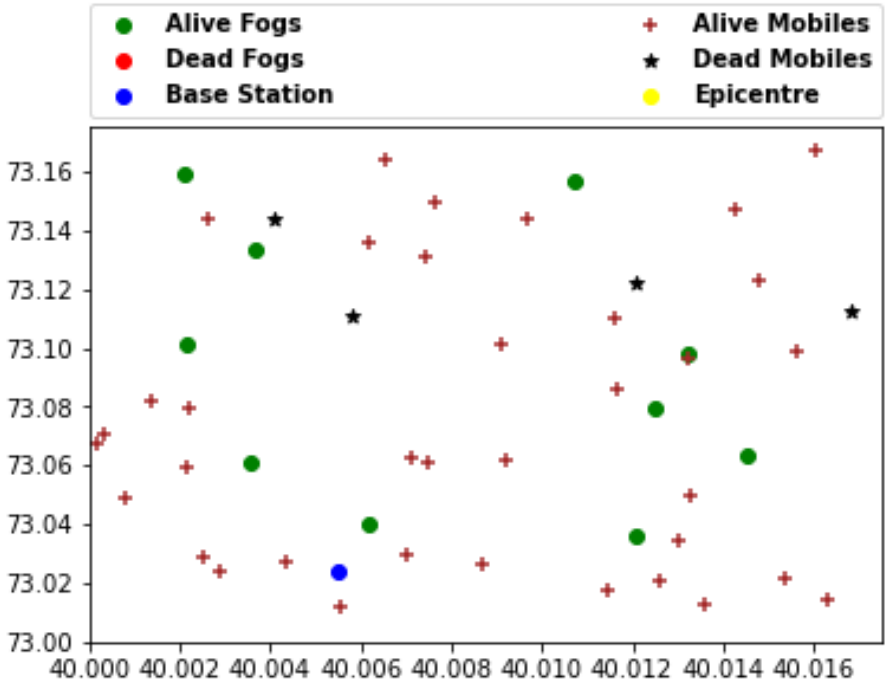
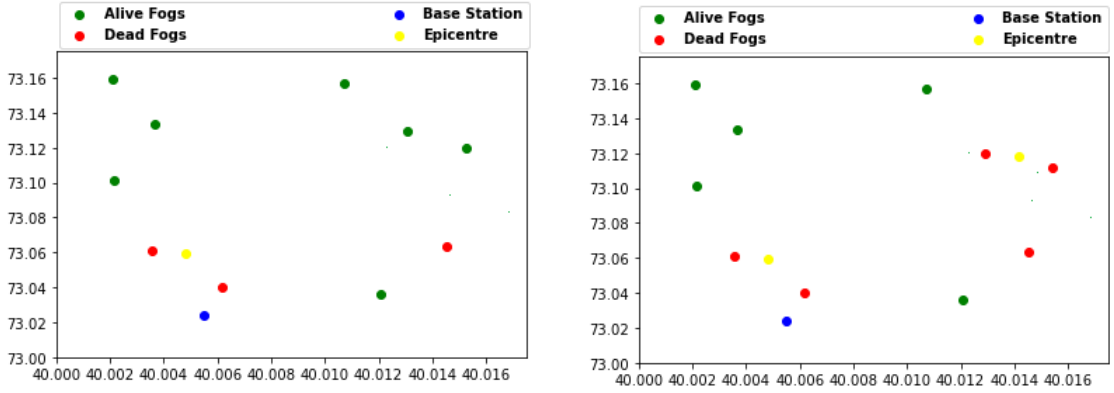


Figure 4.1. *Simulation Area*

4.1.2 Effect of Disaster



(a) (b)

Figure 4.2. *Effects of Disaster on Networking Infrastructure: (a) Post disaster, (b) Post aftershock*

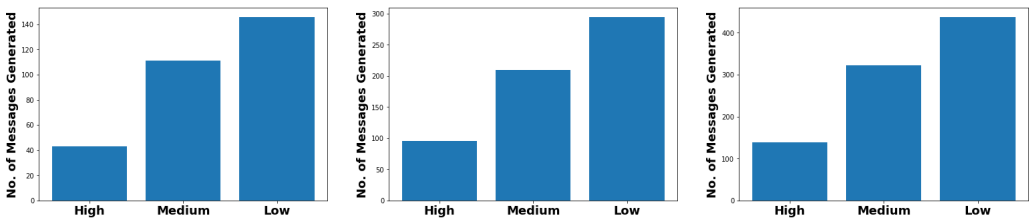
In Fig. 4.2(a), we show the effect of the disaster (e.g., earthquake) on the networking infrastructure. The yellow dot depicts the location of the epicenter of the disaster. The green dots indicate the fog nodes alive (or active) after the disaster, while the red dots indicate the damaged nodes. It can be noted that the base station (blue dot) remains operational. Similarly, Fig. 4.2(b) shows the status of the

fog nodes following the first aftershock, the epicenter of which is shown by a yellow dot. The networking infrastructure is further crippled as few more fog nodes are disabled. At this stage only 40% fog nodes are operational and the challenge is to ensure significant fraction of emergency messages from the IoT devices are delivered to the base station.

It can be observed that the fog nodes in the vicinity of the epicentre are primarily affected, while the intensity of the disaster decreases as the distance from the epicentre increases. Henceforth, we compute a probability score $p(f_i)$ (refer to Section 2) to ascertain the likelihood a fog node f_i situated at a distance $\delta(f_i)$ from the epicentre will be damaged or not.

4.1.3 Message Generation

We generate $n\mathcal{M} = 25$ messages in each time epoch τ, at random points on the simulation area 40.5 - 41 N, 73.7 - 74.5 W. Two principles are followed while generating the simulated messages: (i) messages are generated uniformly within the simulation area; (ii) the power law distribution is applied on message generation mechanism such that only 15% messages are of high priority, 35% are of medium priority, and the remaining 50% are of low priority. This is done to replicate a realistic scenario, where majority of the messages are of low priority. This distribution is maintained across all three stages of the simulation - during normal situation (refer to Fig. 4.3(a)), after the disaster (refer to Fig. 4.3(b), and following the first aftershock (see Fig. 4.3(c))



(a) (b) (c)

Figure 4.3. *Distribution of Generated Messages : (a) Before disaster, (b) Post disaster, (c) Post aftershock*

4.1.4 Message Delivery

Fig. 4.4 illustrates the effect of disaster and aftershock on the message (or packet) delivery rates for different priorities. We simulated the environment and observed message delivery status till 180 time units. As evident, the delivery rate touched 0.9 during normal scenario and dropped considerably after the occurrences of the disaster at 60 time units and the aftershock at 120 time units. The drop in message delivery rate is attributed to the depletion of networking infrastructure by 60% following the disaster and aftershock. However, it is to be noted that even in such a challenged network, the proposed motif centrality-based routing mechanism ensures maximum delivery of the high and medium priority messages (> 0.5) compared to lower ones.

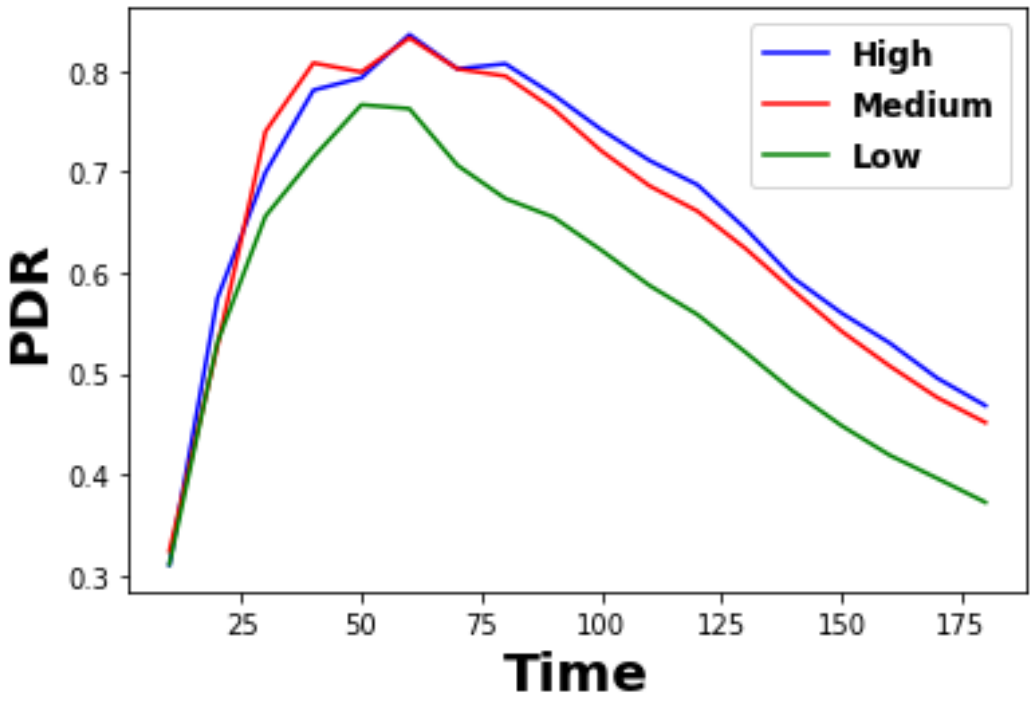


Figure 4.4. *Message Delivery Rate*

4.2 Performance Analysis

In Section 3.3, we discussed the routing protocol followed by the MCR middleware to deliver messages to the base station. In this section, we discuss the effects of

various system parameters on which the efficacy of the routing protocol depends. Some of these parameters are density of the nodes, communication range of the nodes, buffer size of the node, and node failures.

4.2.1 Node Density

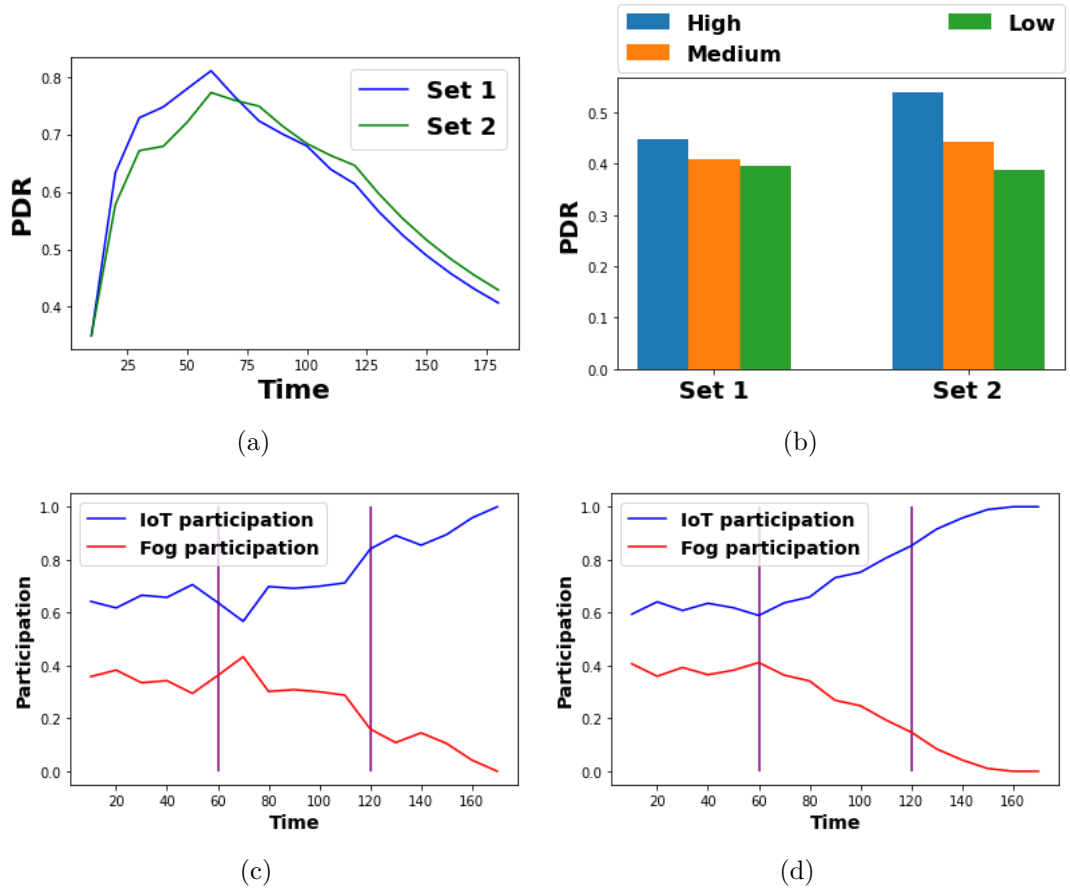


Figure 4.5. *Effect of Node Density on PDR: (a) Varying PDR over Time, (b) Message Priority-wise PDR, (c) Participation Rate under Low Node Density (set-1), (d) Participation Rate under High Node Density (set-2)*

We consider two sets of node densities depending on the numbers of fog nodes and IoT devices in the simulation areas: (a) *set-1*: $fog\# = 10, IoT\# = 40$; (b) *set-2*: $fog\# = 20, IoT\# = 80$. Intuitively, the packet delivery rate under high node density (set-2) should be higher than the rate under low density (set-1), due to the availability of more intermediate hops within the nodes' transmission range.

In post disaster (at simulation time $>$ 60 time units) and aftershock (at simulation time $>$ 120 time units) scenarios, we observe the rate of delivery for messages

(refer Fig. 4.5(a)) steadily drops even for higher node density. This is intuitive as a significant proportion of fog nodes are depleted in the post-disaster scenario which has direct effect on message loss.

Fig. 4.5(b) shows the delivery rates of individual message priority types after the entire simulation duration. It can be observed with both sets of node densities, delivery rates of the high priority messages exceeds that for the medium and low priorities. It is also evident that delivery rates of all message types increase if higher number of alive nodes/devices are available after the disaster.

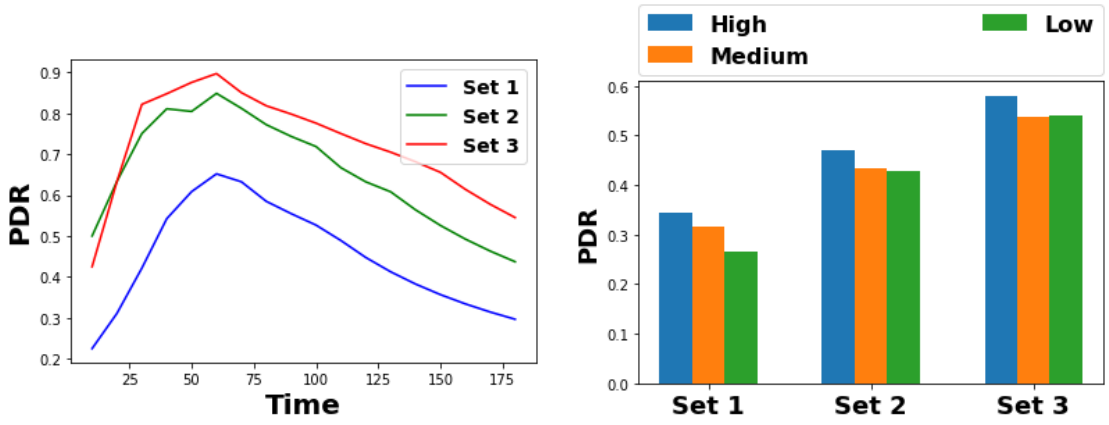
Figs. 4.5(c) and 4.5(d) show the changes in participation for fog nodes and mobile IoT devices in transferring messages to the base station over the entire simulation duration. Irrespective of the node densities (given by set-1 and set-2), during post disaster and aftershock situations (marked by vertical lines at 60 and 120 simulation time units), IoT devices use the MCR middleware and take active participation in building ad-hoc networks and transferring messages. On the contrary, due to depleted networking infrastructure, participation of fog nodes in message delivery gradually decreases.

4.2.2 Communication Range

We consider three sets of communication range (in meters) to determine its effect on packet delivery rate for all types of messages. They are as follows: (a) *set-1*: $fog = 300, IoT = 120$; (b) *set-2*: $fog = 500, IoT = 200m$; (c) *set-3*: $fog = 800, IoT = 320$.

Fig. 4.6(a) depicts that even after the disaster and aftershock, the packet delivery rate is high if the communication ranges of both fog node and IoT devices are on the higher side (set-3). This is an expected result as rise in communication range increases the likelihood of finding a distant next hop neighbor resulting in less hop counts and data drop. Consequently, a high proportion of messages are delivered to the base station.

Similar observation can be made in Fig. 4.6(b) where the delivery rates for all



(a) (b)

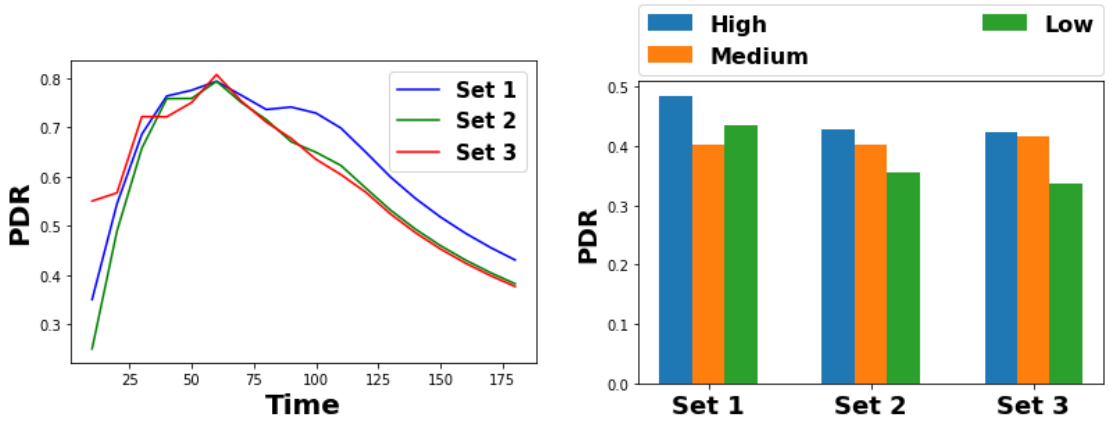
Figure 4.6. *Effect of Communication Range on PDR: (a) Varying PDR over Time (b) Message Priority-wise PDR*

message types increase with the increase in communication ranges of participating nodes/devices. It is to be noted that the delivery rates of high priority message is maximum across different communication range sets. Thus, MCR middleware ensures that even under challenged network environment, majority of the high/medium priority messages are delivered to the base station. This is essential for rescue operations during disaster situation.

4.2.3 Buffer Size

Similar to previous experiments, we design three sets of buffer sizes (in memory units) to illustrate its effects on the packet delivery rates. They are as follows: (a) *set-1*: $fog = 60, IoT = 30$; (b) *set-2*: $fog = 100, IoT = 50$; (c) *set-3*: $fog = 200, IoT = 100$. Intuitively, if the buffer sizes of intermediate nodes increase, packet drop will be reduced and it leads to higher packet delivery rate.

Fig. 4.7(a) reflects a counter-intuitive result where during post disaster message transfers, the delivery rates with large and medium buffer sizes (set-3 and set-2) are noticeably less than that with small buffer size (set-1). This may be attributed to longer queuing time for messages in large and medium buffers. Due to depleted networking infrastructure, the messages remain queued up for longer duration till



(a) (b)

Figure 4.7. *Effect of Buffer Size on PDR: (a) Varying PDR over Time (b) Message Priority-wise PDR*

a suitable next hop neighbor is found. This causes the TTL to expire and the message is dropped. Likewise, in Fig. 4.7(b) the delivery rates for individual message priorities also decrease as the buffer sizes increase. Interestingly, the proportion of high priority message delivered is still higher than both medium and low priorities.

4.2.4 Node Failures

In this experimental study, our objective is to establish the robustness of the proposed MCR mechanism when random nodes and high NMC nodes fail. The random node failure is implemented by choosing an arbitrary node in the network and disable its 30% neighbors (either IoT device or existing fog node) randomly at a time. We repeat the process after ten simulation time units. On the contrary, for high NMC node failure, we again select an arbitrary node from the network, rank its neighbor nodes in descending order of NMC scores, and disable top 30% neighbors at an interval of ten simulation time units.

We design three failure cases to elucidate the robustness property imparted by MCR: (i) *Case-1*: only standard failures occur due to disaster, aftershock, and energy drain-outs; (ii) *Case-2*: standard failures + random node failures; (iii) *Case-3*: standard failures + high NMC node failures.

Fig. 4.8 shows the cumulative packet delivery rates (includes messages of all

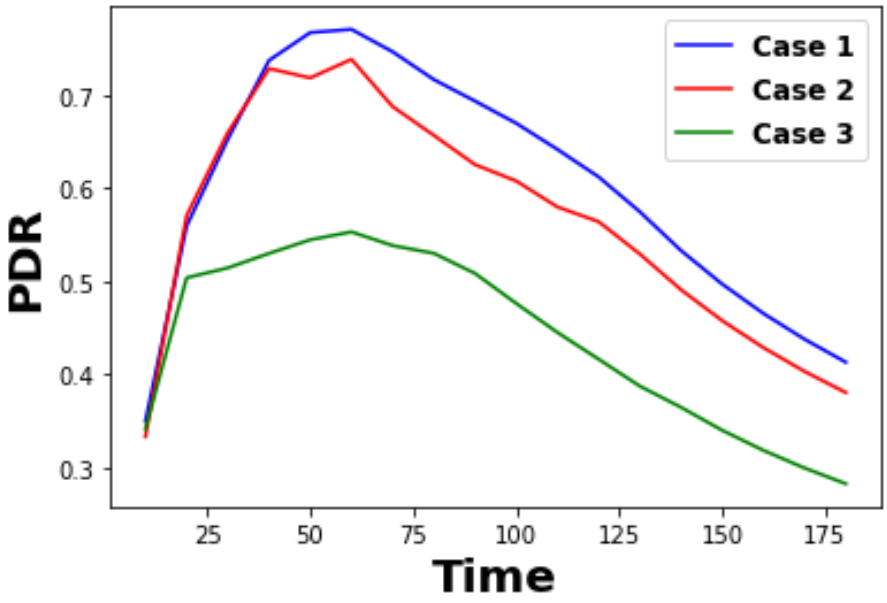


Figure 4.8. *Effect of Node Failures on Performance*

priorities) generated by the MCR scheme under all three cases. As evident, the packet delivery rate is maximum when only standard failures occur, and it attains the lowest value if high NMC nodes fail. However, in the case of random node failures (Case-2), the delivery rate yielded by the proposed routing mechanism is relatively comparable with Case-1. It is due to the ability of the MCR protocol to find a next-hop neighbor that has multiple pathways to its neighbor. If any path becomes disabled due to random node failure, the selected next-hop can choose the alternate path to route the message and exhibit the robustness property of ad-hoc networks. However, Case-3 (i.e., motif-central node failure) causes a significant drop in PDR, since the motif central nodes participate in bulk of the data routing and their removal results in the loss of messages stored in their buffers as well as the current incoming messages. This elucidates the importance of motifs in the network performance.

4.3 Comparison with Benchmarks

We compare the proposed MCR mechanism with three state-of-the-art benchmarks to demonstrate the former's efficacy in terms of message routing over networks which

is partially damaged due to disaster effects. We consider *Flooding*, *PROPHET*, and *Wireless Routing Protocol (WRP)* which is inspired by shortest path-based routing for comparison (discussed at the start of Sec. 4). For comparison, we have taken the three performance matrices into consideration: (i) packet (message) delivery rate (PDR); (ii) communication latency; and (iii) energy efficiency;

4.3.1 Packet Delivery Rate(PDR)

Fig. 4.9 shows the PDR for four mentioned data routing approaches. The two vertical lines at simulation times 60 and 120 time units indicate the occurrence of the disaster and its aftershock, respectively. It is evident, that MCR is the most robust as it clearly outperforms the other three in both normal as well as in disaster scenarios, except towards the fag end of the simulation where its PDR drops below that of *WRP*. The shaded portions depict the upper and lower bounds for MCR's packet delivery rate. We observe the following drawbacks with the three benchmarks.

Flooding increases the congestion level in the network by creating duplicate copies of the same message, which leads to substantial message drop as new data packet arrives.

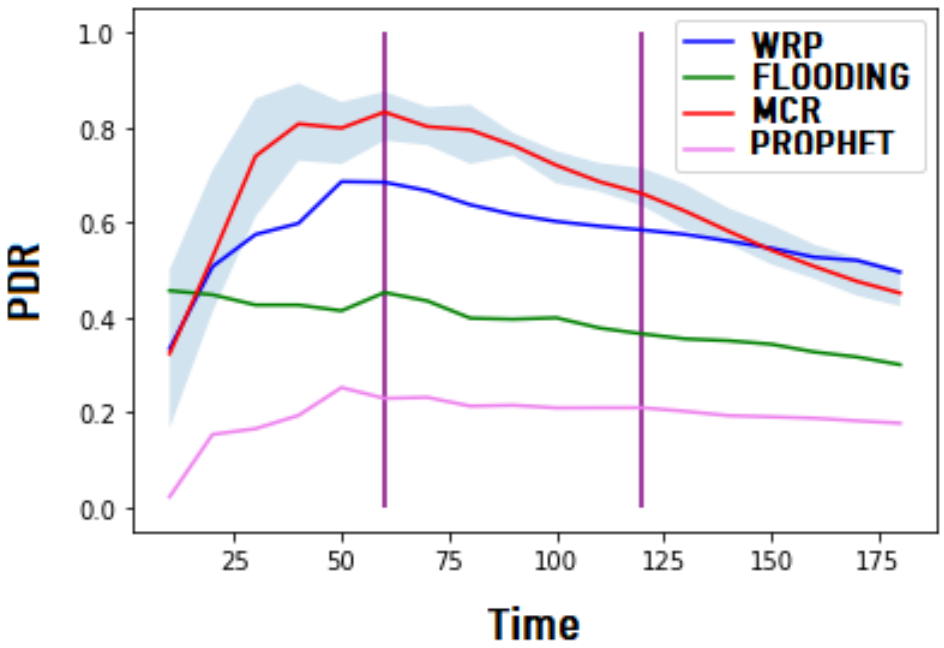


Figure 4.9. *Comparison w.r.t Packet Delivery Rate*

As mentioned, *PROPHET* is a mixture of flooding and routing along the feedback received paths. Due to flooding, the network congestion level rises and the feedback received paths most likely becomes non-existent due to change in location for the intermediate IoT devices, leading to message drops.

In *WRP*, a node sends data through the shortest possible path. Now one node can be a part of many other possible shortest path's intermediate node. So if the node's buffer is full, the sender node has to wait until it is available again. This causes the TTL to exceed the threshold for some of the messages resulting in data drop.

4.3.2 Communication Latency

Fig. 4.10 presents the average routing latency for the four approaches. We express delay in terms of a fraction of the duration of a time epoch. It is understood that under disaster scenario, the message delivery needs to done with minimum delay to initiate quick rescue operations. The line segments above the bars represents the upper and lower latency bounds for each approach.

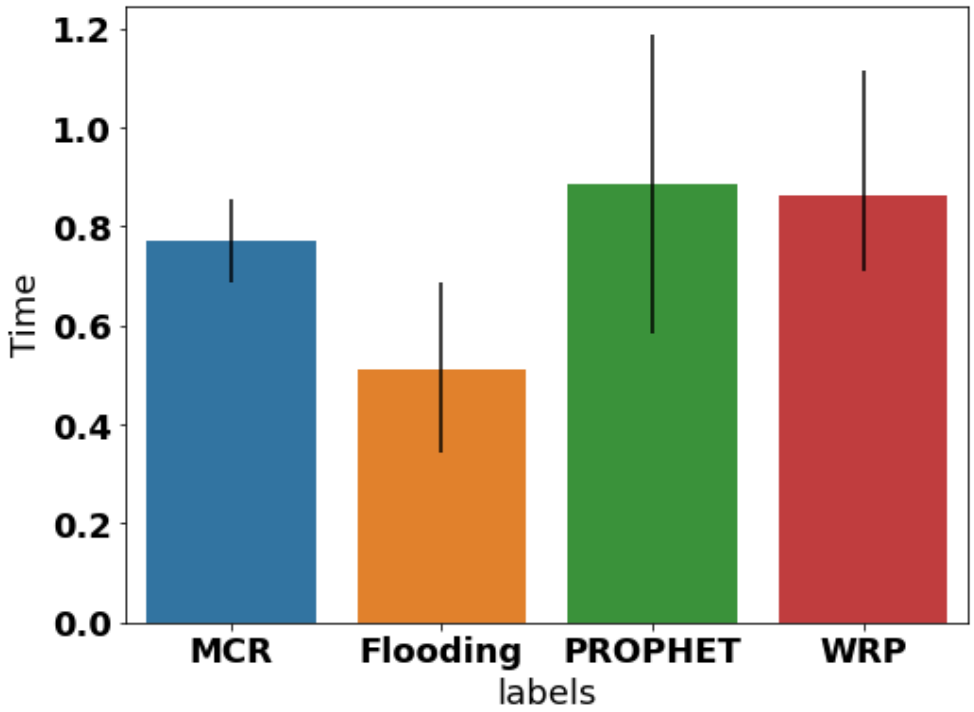


Figure 4.10. *Comparison w.r.t Communication Latency*

It can be observed that *MCR* performs better than both *PROPHET* and *WRP*. In these two approaches, at the current hop, messages need to wait in queue until the chosen next hop neighbor's buffer is free. In contrast, *MCR* takes the next best alternate node for routing if the best next hop neighbor is not available. However, *Flooding* shows minimum latency as a node sends out data in all possible paths, that includes the best path. Hence, the wait time to find the next 'best' node is very less.

4.3.3 Energy Efficiency

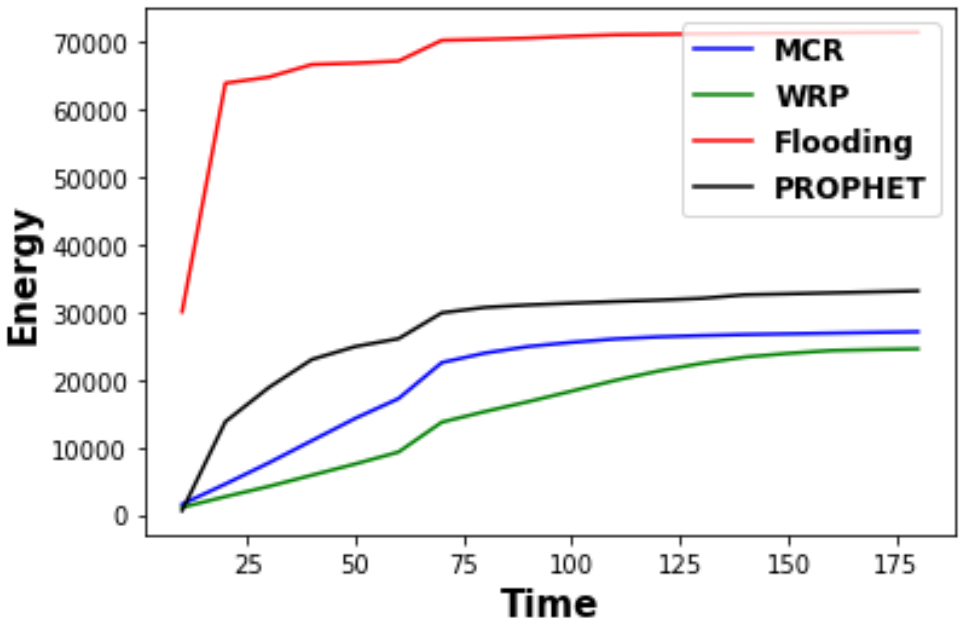


Figure 4.11. *Comparison w.r.t Energy Consumption*

Fig. 4.11 illustrates that *Flooding* is the worst performing mechanism in terms of energy efficiency. During disaster scenarios, it is essential to perform message delivery in an energy efficient way for maximizing network lifetime. In that respect, *MCR* outperforms both *Flooding* and *PROPHET*.

Nevertheless, *WRP* is most energy efficient as only the nodes which form the shortest path participates in the message delivery. *WRP* does not need to spend nodes' energy to find their neighbors and build ad-hoc networks for message delivery as it is a centralized approach and assumes that the global knowledge (to be updated

at periodic intervals) of the network is available. On the contrary, MCR is inherently decentralized in nature and the nodes participate in creating and managing neighbor lists and build ad-hoc networks themselves for message exchanges. As a result, MCR causes higher energy dissipation for the nodes compared to WRP.

Chapter 5

Conclusion

In this work, we presented a distributed, priority-aware routing strategy, called *motif centrality-based routing (MCR)* for disaster-hit regions. MCR leverages the concept of a network motif, called Feed Forward Loop, to find pathways between the mobile IoT devices, static fog nodes and the base station, which are robust to component failures. We mathematically demonstrate that MCR is able to optimize the delivery of high-priority event data and experimentally validate its ability to exhibit robustness against component failures, low latency and energy efficiency, compared to centralized and distributed routing approaches, namely, *PROPHET* and *Wireless Routing Protocol*. In the future, we shall deploy MCR in a simulated disaster environment comprising users carrying mobile handheld devices, wearables, fog nodes, and servers deployed in remote cloud network. It will be interesting to study the variations in the performance of MCR for heterogeneous IoT devices as well as different human mobility models.

Bibliography

[1] S. Roy, N. Ghosh, P. Ghosh, and S. K. Das. biomcs 2.0: A distributed, energy-aware fog-based framework for data forwarding in mobile crowdsensing. *Pervasive and Mobile Computing*, 73:101381, 2021.

[2] Y. Jahir et al. Routing protocols and architecture for disaster area network: A survey. *Ad Hoc Networks*, 82:1–14, 2019.

[3] M. Uddin, H. Ahmadi, T. Abdelzaher, and R. Kravets. A low-energy, multi-copy inter-contact routing protocol for disaster response networks. In *2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 1–9. IEEE, 2009.

[4] T. Mukherjee, G. Varsamopoulos, and S. Gupta. Self-managing energy-efficient multicast support in manets under end-to-end reliability constraints. *Computer Networks*, 53(10):1603–1627, 2009.

[5] V. Shah, S. Roy, S. Silvestri, and S. K Das. Ctr: Cluster based topological routing for disaster response networks. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2017.

[6] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. *Technical Report CS-200006, Duke University*, 2000.

[7] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *IEEE Conference on Computer Communications (INFOCOM)*, 2006.

[8] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In *IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW)*, pages 79–85, 2007.

[9] M. Huang, S. Chen, Y. Zhu, and Y. Wang. Topology control for time-evolving and predictable delay-tolerant networks. *IEEE Transactions on Computers (TOC)*, 62(11):2308–2321, 2013.

[10] F. Li, S. Chen, M. Huang, Z. Yin, C. Zhang, and Y. Wang. Reliable topology design in time-evolving delay-tolerant networks with unreliable links. *IEEE Transactions on Mobile Computing (TMC)*, 14(6):1301–1314, 2015.

[11] U. Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450–461, 2007.

[12] A. Abdelzaher, A. Al-Musawi, P. Ghosh, M. Mayo, and E. Perkins. Transcriptional network growing models using motif-based preferential attachment. *Frontiers in bioengineering and biotechnology*, 3:157, 2015.

[13] S. Mangan and U. Alon. Structure and function of the feed-forward loop network motif. *Proceedings of the National Academy of Sciences*, 100(21):11980–11985, 2003.

[14] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon. Topological generalizations of network motifs. *Physical Review E*, 70(3):031909, 2004.

[15] T. Gorochowski, C. Grierson, and M. di Bernardo. Organization of feed-forward loop motifs reveals architectural principles in natural and engineered networks. *Science advances*, 4(3):eaap9751, 2018.

[16] S. Roy, P. Ghosh, D. Barua, and S. K. Das. Motifs enable communication efficiency and fault-tolerance in transcriptional networks. *Scientific reports*, 10(1):1–15, 2020.

[17] C. Kosyfaki, N. Mamoulis, E. Pitoura, and P. Tsaparas. Flow motifs in interaction networks. *arXiv preprint arXiv:1810.08408*, 2018.

[18] A. Nazi, M. Raj, M. Di Francesco, P. Ghosh, and S. Das. Efficient communications in wireless sensor networks based on biological robustness. In *IEEE Int'l Conf. on Distributed Computing in Sensor Systems (DCOSS)*, pages 161–168, 2016.

[19] S. Roy, V. Shah, and S. Das. Characterization of e. coli gene regulatory network and its topological enhancement by edge rewiring. In *Proceedings of the 9th International Conference on Bio-inspired Information and Communications Technologies*, pages 391–398, 2016.

[20] S. Roy, V. K. Shah, and S. K. Das. Design of robust and efficient topology using enhanced gene regulatory networks. *IEEE Trans. Molecular, Biological and Multi-Scale Communications*, 4(2):73–87, 2018.

[21] S. Roy, R. Dutta, N. Ghosh, and P. Ghosh. Leveraging periodicity to improve quality of service in mobile software defined wireless sensor networks. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–2. IEEE, 2021.

[22] V. Shah, S. Roy, S. Silvestri, and S. Das. Bio-drn: Robust and energy-efficient bio-inspired disaster response networks. In *IEEE Int'l Conf. Mobile Ad Hoc and Sensor Systems (MASS)*, pages 326–334, 2019.

[23] S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature genetics*, 31(1):64–68, 2002.

[24] U. Alon. *An introduction to systems biology: design principles of biological circuits*. CRC press, 2006.

[25] S. Roy, M. Raj, P. Ghosh, and S. K Das. Role of motifs in topological robustness of gene regulatory networks. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.

[26] Mark Newman. *Networks*. Oxford university press, 2018.

[27] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE mobile computing and communications review*, 7(3):19–20, 2003.

[28] S. Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mobile Networks and applications*, 1(2):183–197, 1996.