ADRIN: An Adaptive and Distributed Routing Mechanism for Challenged IoT Networks

Shayantan Halder
Department of CST
IIEST, Shibpur, India
2021csm011.shayantan@students.iiests.ac.in

Satyaki Roy
Department of Math. Sc.
UAH, AL, USA
sr0215@uah.edu

Preetam Ghosh
Department of CS
VCU, VA, USA
pghosh@vcu.edu

Nirnay Ghosh
Department of CST
IIEST, Shibpur, India
nirnay@cs.iiests.ac.in

2024 16th International Conference on COMmunication Systems & NETworkS (COMSNETS) | 979-8-3503-8311-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/COMSNETS59351.2024.10427101

***Abstract*—Data and communication networks in disaster-hit areas are often crippled due to partial or complete outages of the networking infrastructures. In such challenging post-disaster scenarios, message sharing is indispensable for a timely recovery. Typically, it is done by building mobile ad-hoc networks by devices in possession of users. However, challenged networks are often constrained by intermittent connectivity, delay, and energy constraints, necessitating routing strategies that offer seamless communication in the face of node failure and mobility. In this paper, we present an adaptive and distributed routing mechanism termed *ADRIN*. It employs backtracking-based mobility tracing, inspired by the backward algorithm of Hidden Markov Models, to infer periodicity in mobility to determine stable routes and then constructs a spatiotemporal ad-hoc network to relay data in a multi-hop fashion to the base station. Preliminary experiments show that *ADRIN* approximates the underlying mobility distribution and performs data forwarding despite failures.**

***Index Terms*—Internet of Things, Challenged networks, Disaster response, Distributed routing, Periodicity, Mobile computing**

I. Introduction

The Internet of Things (IoT) encompasses an ecosystem of billions of interconnected intelligent devices capable of sensing, computing, and actuating. Such devices include smartphones, tablet computers, wearables, smart home devices, and healthcare devices. They support a variety of communication protocols, such as WiFi, Bluetooth Low Energy (BLE), Zigbee, WiFi direct, 3G/4G/5G LTE, etc. These devices may or may not always be mobile but require a seamless Internet connection. Various real-time and interactive applications run on them, which demands Quality of Services (QoS), such as low response time, energy saving, etc. With traditional centralized cloud-centric architecture, where servers hosted in remote data centers process user requests, it is impossible to conform to the QoS demands of such a massive-scale ubiquitous paradigm.

In recent years, we witnessed the emergence of a distributed multi-access edge computing (MEC) paradigm to enhance real-time applications' Quality of Service (QoS) goals [1] [2]. It addresses latency concerns by placing processing at the network periphery. MEC servers, co-located with cellular base stations, handle location-aware and low-latency computations for users within their radio access network (RAN). The edge layer, administered by mobile service operators, comprises networked nodes offering dynamic computing, networking, and storage services to IoT devices, reducing reliance on back-end cloud/CDN servers. However, in the aftermath of a disaster, edge nodes may be damaged, impacting network services. To address this, survivors and rescue workers create mobile ad-hoc networks using their hand-held IoT devices/wearables with intermittent connectivity and node failures, thus necessitating robustness. Our prior work leveraged machine learning to understand human mobility periodicity, proposing centralized techniques for improved data forwarding [3], [4]. Yet, realistic routing in challenged IoT networks requires distributed and adaptive routing, accommodating low latency and energy-efficient data delivery even in disaster-impacted environments.

A. Contributions

This paper presents a preliminary version of an adaptive and distributed approach, acronymed *ADRIN*, that achieves data forwarding in a challenged (disaster-affected) environment characterized by a complete outage of networking infrastructure, node mobility, and failures. *ADRIN* exploits the underlying patterns or *periodicity* in node mobility in a dynamic environment to determine stable routes to the base station (BS). Given that the IoT nodes follow a stochastic transition matrix of a Markov chain to migrate from one zone to another periodically, *ADRIN* leverages a backtracking approach based on local mobility traces to infer the likely locations of other nodes. Subsequently, each node constructs a local spatiotemporal network of zone visits of peer IoT nodes over time to select the next hops for communication with the BS. We make the following contributions to this work:

1) We introduce *ADRIN*, an adaptive and distributed IoT routing mechanism for challenged network scenarios.
2) We propose a backtracking-based mobility tracing algorithm inspired by the backward algorithm in the Hidden Markov Model that enables IoT nodes to adapt their routing decisions based on the partial location information of peer nodes in a dynamic environment. We elucidate how *ADRIN* differs from existing centralized machine learning (ML) based strategies that use a sliding window approach to infer periodicity in mobility.
3) We experimentally demonstrate that *ADRIN* facilitates each IoT node to utilize local network information to

approximate the temporal mobility trends and carry out data forwarding even under scenarios of node failures.

The rest of the paper is organized as follows. We present the system model in Section II. Section III elucidates the proposed distributed and adaptive routing mechanism. Performance analysis and validation of the proposed mechanism are done in Section IV. Finally, in Section V, we conclude and identify the future research directions.

II. System Model

Our system model comprises an urban environment (assumed to be susceptible to regular natural disasters) in which multiple edge nodes $\mathcal{E}$ and IoT devices $\mathcal{D}$ coalesce to form a network. The system model consists of several IoT devices (e.g., smartphones, wearables, etc.) in possession of the users and a base station responsible for forwarding requests and responses between the users and the cloud or CDN servers through the wired backhaul. Fig. 1 depicts the system model under a post-disaster scenario where all edge nodes have been impaired. In this scenario, the IoT devices collaborate to build multiple ad-hoc networks and transmit data using the suggested distributed routing protocol, *ADRIN*. This system model differs from our earlier work [5] wherein we proposed a motif-centrality-based distributed routing to facilitate forwarding messages in a network with a partially depleted infrastructure. The primary elements of the system model are adopted from [5] and are as follows:

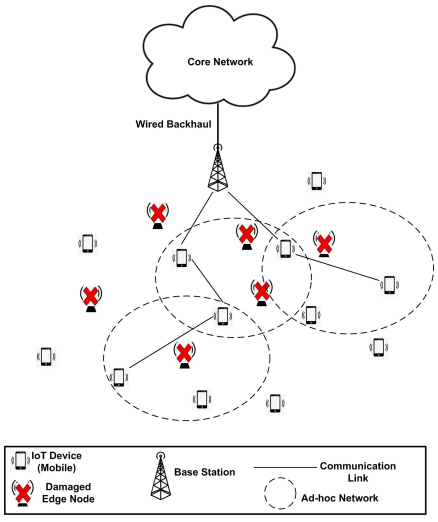

Fig. 1: System Model without Edge Layer

1. IoT device: An IoT device $u \in \mathcal{D}$ is a user-owned, energy-limited mobile device with a low communication range (R_d), limited storage, and computing capabilities. These mobile and position-altering devices provide localized support for distributed routing during disasters, using D2D communication protocols and IPv4/IPv6 for far-off base stations.

2. Base Station (BS): A base transceiver station operated by a mobile operator, receives messages directly from users post-disaster and transfers them to cloud/CDN servers through the backhaul. BS has a higher communication range (R_b), is not energy-limited, and remains operational after disasters.

3. Events: Incidents affecting survivors' lives, given by tuples: $\{$*event ID*, *location coordinates*, *time of occurrence*$\}$.

4. Data message: Data messages containing event data are sent from one IoT device to another. The goal is to transfer the event data to BS in a multi-hop fashion, where the information can be processed, and necessary actions can be taken.

5. Control message: Control messages enable nodes to self-organize themselves into varying topologies, thus imparting an adaptive nature to our routing mechanism. Certain essential control messages include the node's position, buffer status, etc.

6. Mobility of users: Random Waypoint Mobility Model simulates device owners' movement, pausing in locations for defined periods and selecting destinations with varying speeds.

7. Time Epoch: Time epoch, T, encompasses the systematic processes in each routing cycle. This parameter can be configured, and its duration is contingent upon the data collection frequency through mutual interactions among the devices. Utilizing a quasi-static assumption, it is assumed that the network situation remains constant during an epoch.

8. Anchor points: Designated zones, e.g., residences or shelters, where IoT devices periodically reappear, are known from the start and subject to change.

III. Approach

In this section, we discuss the workings of the proposed distributed routing mechanism, *ADRIN*. Each mobile node visits a sequence of zones (or waypoints) $z \in Z$. Given a current time variable $t = 1, 2, \cdots, T$ and a time window $W < t$, let the mobility trace of a node u be $[\cdots, z_{t-W}(u), z_{t-W+1}(u), z_{t-W+2}(u), \cdots, z_t(u)]$, where $z_t(u)$ denotes its zone or waypoint ID visited at time t.

A. Centralized Routing

Centralized routing (proposed in [4]) assumes that there exists a controller that possesses a global view of the current network topology and employs supervised machine learning (ML) to predict the next location of the nodes [4]. We define a feature dataset ($\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots]$) and label ($\mathbf{y} = [y_1, y_2, \cdots]$). We employ the concept of *sliding window* to learn the nodes' periodicity vector (π) and predict a node's future location. The sliding window of length W enables the learning of $\mathbf{X}$ and $\mathbf{y}$ vectors on the most recent temporal mobility patterns of the nodes, given by the node's location in the last W time-points. A feature-length parameter f is given as:

$$(\mathbf{x}_1, y_1) \equiv ([z_{t-W}(u), z_{t-W+1}(u), \cdots, z_{t-W+f}(u)], z_{t-W+f+1}(u)) \quad (1)$$

$$(\mathbf{x}_2, y_2) \equiv ([z_{t-W+1}(u), z_{t-W+2}(u), \cdots, z_{t-W+f+1}(u)], z_{t-W+f+2}(u)) \quad (2)$$

The length of the sliding window for model training (W) is a configurable parameter. Also, note that both centralized routing protocols retrain themselves if the prediction of node locations drops below a predefined threshold.

B. ADRIN: Distributed Routing

The centralized routing mechanism relies on the global knowledge of IoT topology, making it unsuitable for mobile communication in a challenged environment. To address this, we propose a distributed routing mechanism *ADRIN*.

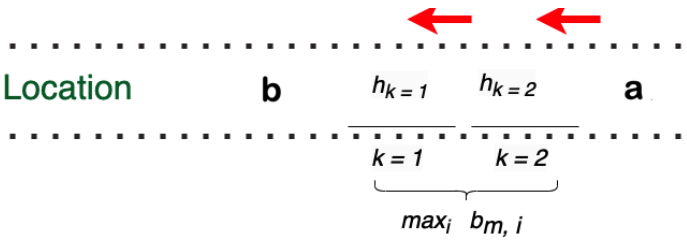

Fig. 2: Mobility tracing through backtracking: an example of an IoT node starting location $z_1 = b$ and migrating to $z_2 = a$ ($\kappa = 3$ hops away). The intermediate location traces ($h_{k=1}$ and $h_{k=2}$) are missing and determined through mobility tracing.

1) Generation of transition matrix: A mobile node, say u, in *ADRIN* maintains a local frequency matrix $\mathbf{F}^u_{|N|\times T}$, which contains information on the zone visited by all the mobile nodes over time, i.e., $\mathbf{F}^u_{v,t}$ gives the ID of the zone, where node $v \in N$ was located at time $t \in T$. All mobile nodes maintain similar ordering along rows (for peer nodes) and columns (for time) in the local frequency matrix. Recall from our discussion in Sec. II that all mobile nodes visit a predetermined set of anchor IDs at specific time points. At the start, each node u initializes its frequency matrix $\mathbf{F}^u$, pre-filled with all its anchor visits and those by other mobile nodes that are its immediate neighbors. Subsequently, two nodes, say u, v, upon establishing contact, exchange their $\mathbf{F}$ matrices (i.e., $\mathbf{F}^u$ and $\mathbf{F}^v$) and augment their matrices with new information received from the peer, using the following update rule:

$$\mathbf{F}^u \leftarrow \mathbf{F}^u + \mathbf{F}^v \tag{3}$$

$$\mathbf{F}^v \leftarrow \mathbf{F}^u + \mathbf{F}^v \tag{4}$$

Mobile node u leverages the locations on its updated frequency matrix $\mathbf{F}^u$ to make routing decisions by creating a location transition matrix $\mathbf{B}^u_{|Z|\times|Z|}$, where Z is a set of zones. Each element $\mathbf{b}_{m,n} \in \mathbf{B}^u$ is initialized to 0 and subsequently updated to hold a normalized measure of the number of transitions in $\mathbf{F}^u$ made by u itself or its peers between zones m and n. For instance, if node u receives information (through message-passing with peer IoT nodes) that another node v has visited locations m and n in consequent timepoints t and $t+1$, respectively, node u updates its local frequency table (i.e., $\mathbf{F}^u_{v,t} = m$ and $\mathbf{F}^u_{v,t+1} = n$). Finally, transitions in the $\mathbf{F}^u$ matrix are processed to create $\mathbf{B}^u$, and $\mathbf{B}^u$ is row-normalized to act as a local transition matrix for u.

2) Mobility tracing through backtracking: To determine the next hop to forward the event data, a mobile IoT node employs a backward algorithm in the Hidden Markov Model [6] to find neighbor nodes likely to interact with the base station. As discussed earlier, in a dynamic environment, a node only has partial knowledge of the location of other nodes. *ADRIN* enables nodes to infer the unknown location visits of other nodes. Each node u maintains a local version of the transition matrix $\mathbf{b}_{m,n} \in \mathbf{B}^u$, based on prior interaction with peers.

Given the starting and ending zones (denoted by z_1 and z_2, respectively), the distributed algorithm backtracks from location z_2 towards z_1 to find the missing location information. Fig. 2 depicts a case where the last known starting zone location $z_1 = b$ and the ending location $z_2 = a$ is $\kappa = 3$ hops away. The intermediate location traces ($h_{k=1}$ and $h_{k=2}$) are missing. The estimation of the last missing zone ($h_{k=2}$) can be broken into the combination of the probability of (1) traveling from z_1 to $h_{k=2}$ in two hops and (2) moving from $h_{k=2}$ to z_2 in one hop. Once $h_{k=2}$ is known, the same principle can be applied to infer $h_{k=1}$. Overall, we generalize this tracing back to predict the missing location κ hops away from the transition matrix $\mathbf{b}_{m,n} \in \mathbf{B}$, capturing a node's location knowledge of moving from zone m to n, as:

$$h_k \leftarrow \max_i \mathbf{b}^k_{z_1,i} \times \mathbf{b}_{i,h_{k+1}} \tag{5}$$

3) Data forwarding: A node selects a subset of nodes from among its neighbors to forward event data. Using the mobility information collected through the *distributed mobility tracing* (refer to Sec. III-B2), a node constructs a spatiotemporal network comprising the zones and links tracing visits to and from the zones over time. The *data forwarding step* is invoked to push the data towards the base station BS with the least possible delay. Specifically, each node selects K nodes likely to visit the BS soon and forwards a copy of the event data to each of them. For instance, for $K = 2$, node 0 with neighbors $1, 4, 3$ at any given time t, chooses two nodes, say 4 followed by 1 in the order in which they are expected to visit BS.

IV. Results

We create a customized simulation environment using the Python SimPy library [7] to capture the interaction among the nodes and base station. Simulations on 50 nodes are done for 200 minutes. A node visits zones (each representing one of five boroughs of New York City) based on a probability distribution learned from the mobility dataset (refer to Sec. IV-A for the details). A node returns to a preassigned (but not necessarily the same) anchor location, given by one of the five zones, every 40 minutes. During the simulation, the mobile IoT nodes sense and forward event data to the BS in a multi-hop fashion via peer IoT nodes (refer to Sec. II).

A. Estimating Transmission Probability

We consider the five boroughs in New York, namely, Bronx (1), Brooklyn (2), Manhattan (3), Queens (4), and Staten Island (5). We source the mobility data of NYC traffic from NYCOpenData [8], a public data repository for fields on city government, education, environment, health, public safety, recreation, social services, and transportation. We use the GeoPy python library [9] to link mobility traces' (latitude, longitude) coordinates to the boroughs and estimate the probability of moving from one borough to another.

We consider 20 nodes moving *daily* over 200 days. Each node employs the proposed distributed approach to estimate the inter-zonal transition probabilities (refer to Sec. III-B2). In this experiment, we analyze whether the local transition matrix ($b_{m,n} \in \mathbf{B}$) approximates the true transition probabilities. Figs. 3a, 3b, 3c, and 3d show the estimated local transition probabilities for source destination pairs $(2,2)$, $(2,4)$, $(3,3)$ and $(3,4)$, respectively. For the 20 nodes, the local transition probabilities over time are represented by different lines. We demonstrate that the mean of the local transition probability (shown as a black curve) converges to the true transition probability (shown in red), suggesting that the distributed mobility tracing approach approximates the true mobility trends.

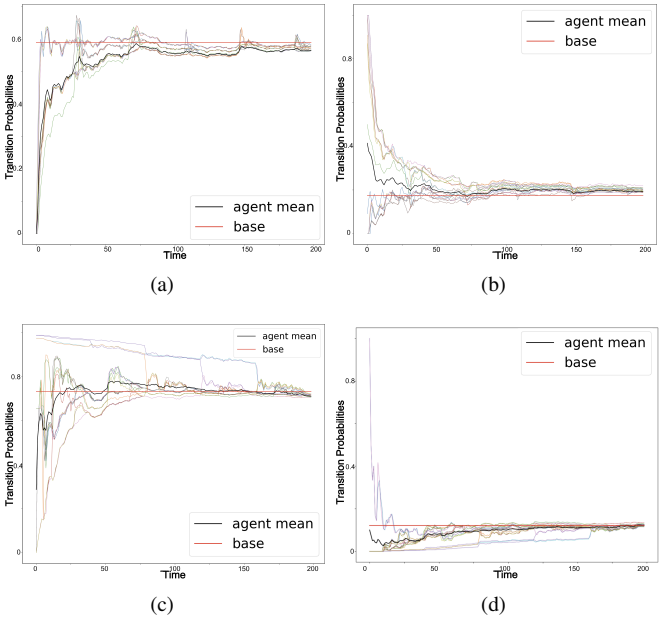

Fig. 3: Comparison of true transition probability (shown in red) against predicted transition probability of 20 mobile nodes (with mean shown as a black curve) for source destination pairs (a) $(2,2)$, (b) $(2,4)$, (c) $(3,3)$ and (d) $(3,4)$.

B. Node Failure

We study the robustness of *ADRIN* when mobile nodes fail randomly. We consider three sets having node (1) $fail\% = 80$; (2) $fail\% = 40$; (3) $fail\% = 0$. In set-1, we randomly choose 20% of total alive nodes at every 40-minute interval and deactivate them. Similarly, we randomly select 10% of the active nodes every 40 minutes and make them dead (set-2). Other parameter values are set as $mobile\# = 20$, $range = 80.46$ meters, a buffer size of 100 messages, and $K = 3$ (where K, as discussed in Sec. III-B3, denotes the number of neighbors a node forwards its messages to). Node failure affects the packet delivery rate (refer Fig. 4(a)), overall delay for messages (refer Fig. 4(b)), and rate of the energy consumption (refer Fig. 4(c)). Failure of nodes increases packet drop, leading to delayed data delivery to the base station. As packet loss is high, the number of transmissions and receipts as well as energy usage are low.

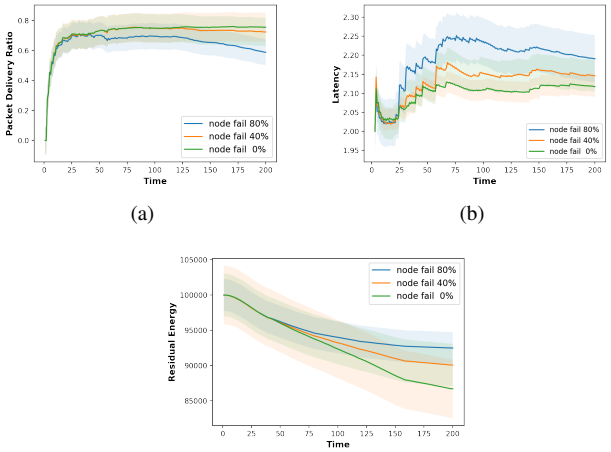

Fig. 4: Temporal effect of node failure on PDR, latency, and residual energy: Varying (a) PDR, (b) latency, (c) energy

V. Conclusions

In this paper, we introduced *ADRIN*, an adaptive and distributed routing mechanism designed for challenged IoT networks. Operating in infrastructure-limited environments where mobile nodes establish ad-hoc networks for message exchange, *ADRIN* utilizes a backtracking strategy. This strategy involves learning the periodicity in the mobility of neighboring nodes, leading to the construction of a local spatiotemporal network of neighbor locations. The results indicate that *ADRIN* achieves a favorable balance between data delivery and energy conservation, minimizing data forwarding delays. Future investigations will involve testing *ADRIN*'s performance under various failure scenarios typical of post-disaster environments. We shall explore its efficacy in scenarios where privacy concerns restrict mobile nodes from freely sharing their locations.

References

[1] L. Vaquero et al. Finding your way in the fog: Towards a comprehensive definition of fog computing. *ACM SIGCOMM comp. comm. Rev.*, 44(5):27–32, 2014.

[2] F. Liu et al. A survey on edge computing systems and tools. *Proceedings of the IEEE*, 107(8):1537–1562, 2019.

[3] 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.

[4] S. Roy, B. Chandra, A. Anand, P. Ghosh, and N. Ghosh. Learning temporal mobility patterns to improve qos in mobile wireless communications. In *Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2022*, pages 355–365. Springer, 2022.

[5] U. Roy et al. Mcr: A motif centrality-based distributed message routing for disaster area networks. *IEEE Internet of Things Journal*, 9(24):25337–25349, 2022.

[6] S. Yu and H. Kobayashi. An efficient forward-backward algorithm for an explicit-duration hidden markov model. *IEEE signal processing letters*, 10(1):11–14, 2003.

[7] N Matloff. Introduction to discrete-event simulation and the simpy language. *Dept. of CS, Univ. of California at Davis*, 2(2009), 2008.

[8] Nycopendata. https://data.cityofnewyork.us/Transportation/Traffic-Volume-Counts-2012-2013-/p424-amsu, 2020.

[9] Geopy: Geocoding library for python. https://github.com/geopy/geopy, 2020.