



HAL
open science

A probabilistic routing protocol for intermittently connected networks in the rural scenarios

Adriana Arteaga-Arce, Nathalie Mitton

► To cite this version:

Adriana Arteaga-Arce, Nathalie Mitton. A probabilistic routing protocol for intermittently connected networks in the rural scenarios. International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Oct 2024, Paris, France. hal-04684901

HAL Id: hal-04684901

<https://hal.science/hal-04684901>

Submitted on 3 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A probabilistic routing protocol for intermittently connected networks in the rural scenarios

Adriana Arteaga-Arce, Nathalie Mitton
Inria, France - Email: {firstname.lastname}@inria.fr

Abstract—Deploying communication networks in rural scenarios is challenging due to the sparse population density, geographical obstacles, lack of infrastructure, and daily life situations. Inspired by the PROPHET protocol, we present the R-PROPHET protocol to provide an intermittently connected network in a rural context, where the variable mobility of vehicles requires flexibility to model the encounters between nodes. Our protocol incorporates information about the time between encounters in the past to make smooth adjustments in parameters that impact the decision to relay messages. The R-PROPHET protocol implements two forwarding strategies to manage the trade-off between the delivery ratio and the number of messages relayed along the network. Results showed that R-PROPHET increases the delivery ratio by up to 20% and reduces the average latency by up to 16% compared to PROPHET. The overhead produced by R-PROPHET is around two times more than PROPHET, which is part of the trade-off of increasing the average delivery ratio and reducing the average latency.

Index Terms—Delay tolerant networks, intermittently connected networks, rural connectivity, vehicular communications

I. INTRODUCTION

Intermittently connected networks are designed for scenarios where connectivity is sporadic, where nodes can eventually connect to the network and go through long periods of disconnection. One of the scenarios where intermittently connected networks can be proposed is rural environments, given the limited or non-existent communication infrastructures. In these environments, the collection of demographic, environmental and socioeconomic data, among others, allows, on the one hand, to have real information to model different processes such as birth and mortality, the use of natural resources, productivity, health, as well as to provide support for various uses such as healthcare, education, banking and communications. However, the limited or non-existent communication infrastructures makes it difficult for the information to reach its final destination, which may be a service available through the Internet or an application stored on a local server. An alternative to establishing an intermittent network in the rural context is using the vehicles that travel throughout the rural area as relay nodes. In this way, the data may use a communication protocol to decide the hops among the vehicles toward the final destination or to an Internet access point.

One of the most popular communication protocols proposed for intermittently connected networks where some of the nodes are mobile is the Probabilistic Routing Protocol using History of Encounters and Transitivity (PROPHET) [1]. This protocol defines a probabilistic metric named the "Delivery Predictabil-

ity" between node A and node B to predict connectivity patterns between nodes. This protocol can be used in scenarios where mobility and contact patterns are not completely random (e.g., bus routes, repetitive activities like going to work and coming home, etc.) to predict future encounters between two nodes based on past encounters. Every time a node contacts another node and has messages to send, it can decide whether transferring the messages to the contacted node increases the probability of reaching the final destination. Avoiding transferring messages to intermediate nodes with a low probability of reaching the final destination avoids unnecessarily filling the buffers of intermediate nodes.

PROPHET calculates the delivery predictability metric between two nodes using a fixed parameter to represent the typical interval between encounters. However, this "typical" value may change, especially in rural scenarios where the mobility of the vehicles is affected in the mean or long term by the quality of the roads, partial road closures, bad weather, or everyday situations in the rural context. PROPHET is sensitive to a long time between encounters or a drastic change in this parameter. This sensitivity degrades the communication's performance, especially in scenarios where mobile nodes have variable speed along the time.

In this article, we introduce the R-Prophet protocol to be used in rural scenarios where delays or anticipation of encounters are expected. In R-Prophet, we replace the fixed "typical time between encounters" parameter with a new parameter named the "estimated encounter time." This new parameter is a dynamic value adjusted based on the previous time between encounters values and the current one. We propose an additional forwarding strategy to reduce the average delay but it could create a trade-off between the delivery ratio and the among of messages moving along the network. R-PROPHET increases the delivery ratio in up to 20%, and reduces the average latency up to 16% compared to PROPHET. The overhead produced by R-PROPHET is around two times more than PROPHET, which is part of the trade-off of increasing the average delivery rate and reducing the average latency.

The remainder of this paper is organized as follows. Section II presents the state of the art of the PROPHET protocol adaptations. In Section III, the details of the PROPHET protocol are presented. Section IV describes the rural context scenario used to analyze the time between encounters parameter. Section V presents the R-Prophet protocol. Section VI describes the evaluation scenario. Section VII reports the results, before concluding with Conclusions and future work and in Section

VIII.

II. STATE OF THE ART

There are several PROPHET adaptations in the literature. One of the first approaches to increase the dissemination speed was proposed in [2] by employing epidemic dissemination for a message if the number of hops and a forwarding counter is smaller than a threshold. The PROPHET+ protocol proposed in [3] uses a weighted function that evaluates the nodes' buffer size, power, location, popularity and the delivery predictability value from the original PROPHET to generate a "deliverability" value for routing packets. The Distance-based PROPHET protocol proposed in [4] uses a cross-layer distance value retrieval mechanism using the two-ray ground reflection model to determine the distance between neighbors and send the message only to the closest node. This adjustment reduces generating duplicated messages when two or more nodes have the same delivery predictability value for a specific destination. PROPHET-RAID5 was proposed in [5] to manage integrity, confidentiality, and availability in wireless sensors networks. The association rule base PROPHET (ARBP) protocol [6] incorporates the mining technique of machine learning to find the best encountering node to improve the delivery ratio.

The probabilistic approach proposed in the PROPHET protocol has been also to model short-term traffic prediction to determine traffic patterns in vehicular communications in [7] using the Fb-PROPHET forecasting model to predict traffic flow as trends, and the Neural-PROPHET model which extends the probabilistic using an auto regressive neural network for time series forecasting [8].

III. BACKGROUND

PROPHET is a probabilistic routing protocol that uses the Delivery Predictability metric based on node encounters and transitivity to select the best candidate for delivering messages to a specific destination [4]. The Delivery Predictability metric $P(A, B)$ is calculated at every node A for each known destination A. A node with higher predictability for a specific destination is estimated to be a better candidate for delivering the messages; nodes that are often encountered have a high delivery predictability. When nodes A and A meet, they exchange the metric values for the nodes they have encountered (e.g., the neighbors) and calculated their respective delivery predictability metric using [1]

$$P(A, B) = P(A, B)_o + (1 - \delta - P(A, B)_o) * P_E, \quad (1)$$

where $P(A, B)_o$ is the value before meeting the current encounter, δ is an upper bound for $P(A, B)$, and P_E is a scaling factor between 0 and 1 setting the rate at which the predictability increases on encounters after the first. We will explore the P_E parameter in details in Section V.

Finally, PROPHET defines a transitivity function considering that if node A frequently encounters node A, and node A frequently encounters node C, then node C probably is a good node to which to forward messages destined for node A.

The transitivity function to update the $P(A, C)$ value in node A using the metric values received from the encountered node A is [1]

$$P(A, C) = \max(P(A, C)_o, P(A, B) * P(B, C) * \beta), \quad (2)$$

where β is a scaling constant between 0 and 1 that controls how large an impact the transitivity should have on the delivery predictability.

IV. SYSTEM MODEL

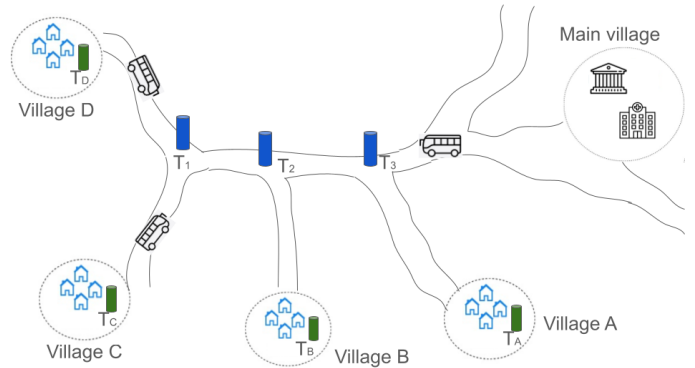


Fig. 1: General scenario of study

We use the scenario shown in Fig. 1 to explain the system model. The figure presents a rural area where there are one main towns (MT) and four small villages (V_A, V_B, V_C, V_D) interconnected by several routes. The main towns have public services facilities such as a hospital, a school, and other public offices, while the small villages bring together small groups of inhabitants, farms, and natural areas. We assume small villages do not have the infrastructure to connect to the Internet. Since it is necessary to collect information in the small village and exchange information between the main town and the rest of the villages, in our system we propose to install fixed connection points called Totems in the villages (T_M, T_A, T_B, T_C, T_D) as a central point to receive messages and get them ready out of the village, or to receive messages coming from outside. Each totem has storage capacity, and multi-technology interfaces to transfer messages using Wi-Fi or Bluetooth, and an energy supply system.

In our system, buses traveling between villages and the main town are relay nodes moving data closer to the destination. Buses have storage and connectivity modules to transfer data between them and the totems. In the studied scenario, we propose to install three totems (T_1, T_2, T_3) in the intersection between some path routes to store messages temporarily and transfer them to bus then messages continue moving closer to the final destination. Regarding to the buses, we consider three routes: bus $B_{M,A}$ traveling between the main town and village A; bus $B_{C,A}$ going between village C and village A; and bus $B_{D,B}$ traveling between village D and village B.

Since this article analyzes the message transfer protocol among buses and totems, we assume each village has mechanisms to bring messages from the applications or devices to the totem in the village and to distribute messages from outside the village to the respective devices and applications inside. For instance, a way to do this is to provide portable devices where users can store data and approach the totem to transfer it. However, we will not give more details about possible approaches since it is outside the scope of this study. Once the data is in the village's totem, the protocol could transfer messages to the bus near the totem or wait for another one. For messages carried by the bus, the protocol could decide whether the messages should remain on the bus, transfer messages to another bus heading towards the final destination, or to another totem along the route.

V. R-PROPHET PROTOCOL

This section presents the proposed R-Prophet protocol, which differs from the original PRoPHET in two aspects. First, we replace the fixed "typical time between encounters" parameter with a new parameter named the "estimated encounter time." As we will explain below, this parameter affects the P_E value in Eq. 1. The second difference is that we include an additional forwarding strategy to favor sending the first messages when the nodes have found a few neighbors.

A. Estimated time between encounters

We focus our study on rural environments, where buses traveling regularly between villages and some fixed nodes located at road intersections establish the intermittently connected network. The mobility of the buses could be affected by the road's state, partial road closures, bad weather, or everyday situations in a rural context. These circumstances cause the buses' speed to vary, affecting schedule compliance and the time between encounters during the trips. We will not consider the variations in the time between encounters caused by failures in the totems located in the villages or at the roads' intersections. Also, we assume messages are transferred correctly whenever the bus reaches a totem or another bus.

In the PRoPHET protocol, the $P(A, B)$ metric is calculated using Eq. 1, which includes the P_E variable. This variable is a function of the time interval since the last encounter between A and A resulted in an update of the $P(A, B)$ and $P(B, A)$ values; that interval corresponds to the time between encounters. The P_E function [1] is as follows :

$$P_E(I) = \begin{cases} P_{E_{max}} * (I/I_T), & \text{if } 0 \leq I \leq I_T \\ P_{E_{max}}, & \text{for } I > I_T, \end{cases} \quad (3)$$

where $P_{E_{max}}$ is used as the upper limit of a scaling factor that increases the delivery predictability for a destination when the destination node is encountered; I_T is the *typical* interval between encounters, which is a static value assigned depending on the mobility. For instance, in a scheduled transport system, I_T is the number of minutes a bus takes to reach the same point again. The I value is the real time interval between

encounters; when vehicles move with constant speed and no delay for external events, I and I_T have the same values.

In rural environments, mobility is not fluid, as mentioned previously, so keeping the I_T static might affect the P_E value in Eq. 3, especially when there are situations that would extend the time between encounters for a long time. Instead of maintaining the I_T value static in Eq. 3, we propose to use a dynamic I_T for the s encounter between node A and node A, (\tilde{I}_T), that is calculated using a weighted mean value of I from the first encounter until the $(s-1)$ -encounter and the I value for the current encounter s . The (\tilde{I}_T) is:

$$\tilde{I}_T = w * \sum_{s=1}^M \frac{I_{s-1}}{M} + (1-w) * I_s, \quad (4)$$

where w is a value between 0 and 1.

B. Forwarding strategies

According to the PRoPHET protocol, nodes A and A have to update their neighbor tables using Eq. 1 and Eq. 2 in each encounter between them. Then, node A checks if it has messages for node A and deliver them directly. If it has messages for a different node, for example node C, node A compares if $P(B, C)$ is greater than $P(A, C)$. In that case, node A will only transfer the message to node A if it has a higher probability of reaching node C.

We implement a first version of the R-Prophet protocol, R-Prophet-vA, for the R-Prophet protocol, which maintains the same forwarding strategy as the original PRoPHET protocol. Additionally, we implement a second version of our protocol, R-Prophet-vB, where if node A has messages to node C, node A will transfer the message to node B if $P(B, C)$ is greater or equal to $P(A, C)$.

Including the equal condition allows initial messages to start moving across the intermittent network from the first encounters when, for instance, no node on the network has information about node C yet. Over time, nodes will have more information about other nodes in the network to become aware of node C, and transferring messages through intermediate nodes will be more contained. This strategy reduces latency in reaching the final destination. Still, it may increase the number of messages in the network since nodes will send more messages during the initial encounters.

VI. EVALUATION

To evaluate R-PRoPHET, we recreate the scenario shown in Fig. 1 using The One Simulator. In the scenario, we placed five villages and one totem that generates packets traffic periodically during the simulation to represent the data generated inside village D going to the main village. The size of the message varies between 500 kB - 1 MB and the traffic load was 1,000 messages in 12 hours. Three totems are placed along the routes between the villages, operating as intermediate nodes. The intermediate totems do not generate data but store data moving through buses; we setup up to three buses traveling between the villages. Each bus follows a

specific path defined in the simulation setup to travel between two villages and pass close to one of two totems in the route, but there is not a bus traveling from village D to the main village directly then messages from village D have to use intermediate totems and buses to reach the final destination. The buffer size in the totems and in the buses is 1GB.

We compare R-PRoPHET-vA, the R-PRoPHET-vB and the PRoPHET protocols when the speed of the buses varies between 15 km/h and 25 km/h. In this study, we evaluate the average delivery ratio, latency, overhead metrics for five specific Time-to-Live (TTL) values going from 60 min to 360 min. We use the TTL parameter as a way to represent the time-frame where the data is still alive; the packet is not delivered if the TTL expires before the packet reaches the final destination. We ran 20 simulations per configuration setup to calculate the average of the performance metrics.

The values of parameters such as δ , β , and $P_{E_{max}}$ for Eq. 1, 2, and 3 were set up using the recommended values in the RFC of the PRoPHET protocol [1].

VII. RESULTS

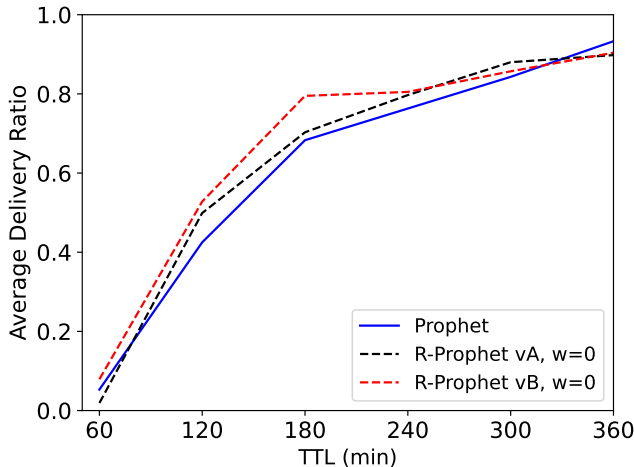


Fig. 2: Average delivery ratio comparison

The average delivery ratio achieved by the protocols is presented in Figure 2. In this first analysis, we do not consider information from previous encounters then both R-PRoPHET-vA and R-PRoPHET-vB protocols use $w = 0$ to initially study the impact of the right side of the Eq. 4 in replacing the typical interval parameter with the dynamic parameter as explained in Section V-A. The results show that the delivery ratio is under 50% for the three protocols when the TTL is less than 60 min. For TTL values between 120 and 180 min, both versions of the R-PRoPHET protocol achieve higher or equal delivery predictability than the PRoPHET protocol, reaching 0.8 for R-PRoPHET-vB, and around 0.7 for R-PRoPHET-vA and PRoPHET when the TTL is 180 min. The average delivery ratio for the three protocols reaches a similar value of around 90% when the TTL becomes longer than 360 min.

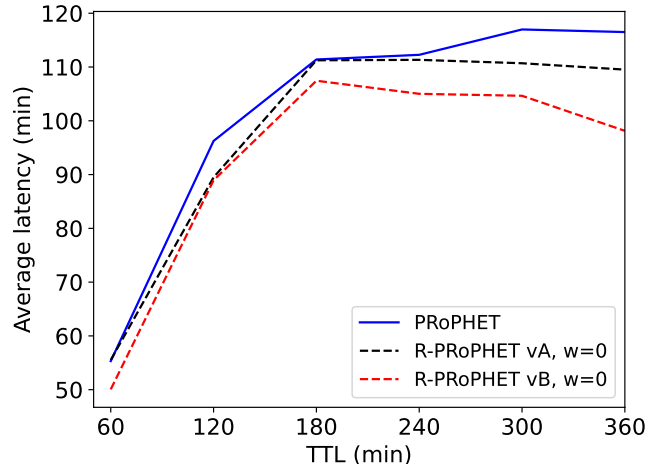


Fig. 3: Average latency comparison

Figure 3 shows the average latency obtained for the three protocols when $w = 0$ for both R-PRoPHET-vA and R-PRoPHET-vB. As explained before, R-PRoPHET-vB reduces the average latency because its forwarding strategy starts transferring messages from the beginning of the simulation, making it possible that more messages reach the destination sooner compared with R-PRoPHET-vA which applies the same forwarding strategy as the PRoPHET protocol. The average latency reduction using R-PRoPHET-vB allows more messages to reach the destination before the TTL expires, explaining why R-PRoPHET-vB has a higher average delivery ratio than R-PRoPHET-vA.

In the following, we analyzed the impact of including previous time between encounters values in calculating the dynamic interval value \tilde{I}_T in the R-PRoPHET protocol. Figure 4 shows the average delivery ratio comparison for both versions of the R-PRoPHET protocol. The w parameter in Eq. 4 is the weight applied to the mean value of the previous time between encounters to regulate the impact of prior information about the current encounter. Results showed that the average delivery ratio increases as the w value increases for all the TTL values in both R-PRoPHET versions. In Fig. 4a, for instance, the average delivery ratio when the TTL is 120 min goes from 0.5 to 0.6 for $w = 0.25$, which means the mean value of the previous time between encounters weights 25% while the time between the last encounter and the current one weights 75%.

As the weight of previous values increases, the value of \tilde{I}_T more accurately represents the variability of time between encounters due to changes in buses' speed. As a chain consequence, making smoother adjustments in the \tilde{I}_T parameter will impact smoothly in the P_E (Eq. 3) and finally in the $P(A, B)$ calculation (Eq. 1). The smooth variation of $P(A, B)$ values avoids the sudden degradation of communication in rural environments where variation in the time between encounters caused by multiple situations is expected.

Regarding the average delivery ratio using R-PRoPHET-

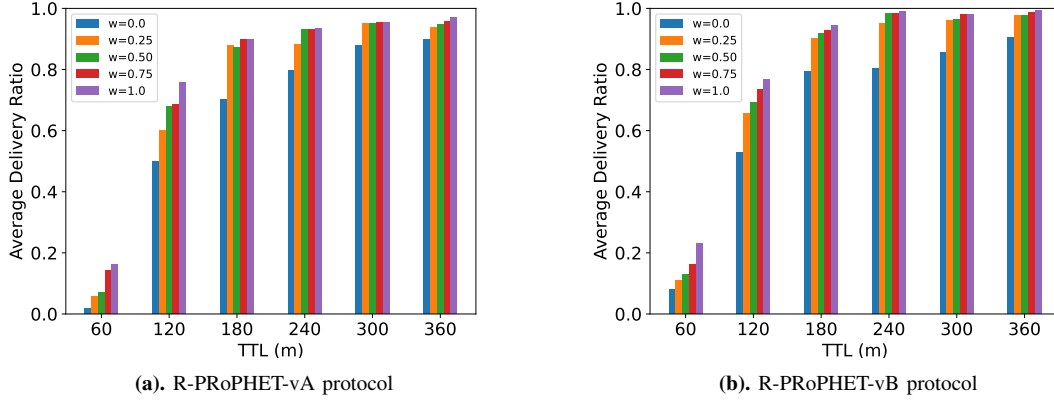


Fig. 4: Average Delivery Ratio varying the w parameter in R-PRoPHET-vA and R-PRoPHET-vB protocols

vB, Fig. 4b shows that both R-PRoPHET versions follow the same tendency. Still, the average delivery ratio is higher using R-PRoPHET-vB because of the average latency reduction explained in Fig. 3.

Finally, Fig. 5 shows the overhead produced by the protocols, particularly for $w = 1.0$ for R-PRoPHET protocols when they achieve the higher average delivery ratio. The overhead is computed as the ratio between the total number of messages relayed during the simulation and how many of those messages were delivered to the final destination. The overhead is maximum when the TTL is 60 min for the three protocols because the number of delivered messages is low since the TTL expires quickly. As the TTL increases, the number of delivered messages increases then the overhead is reduced. Results showed the overhead is reduced by 20 times for the R-PRoPHET-vB protocol and by 40 times for PRoPHET and R-PRoPHET-vA.

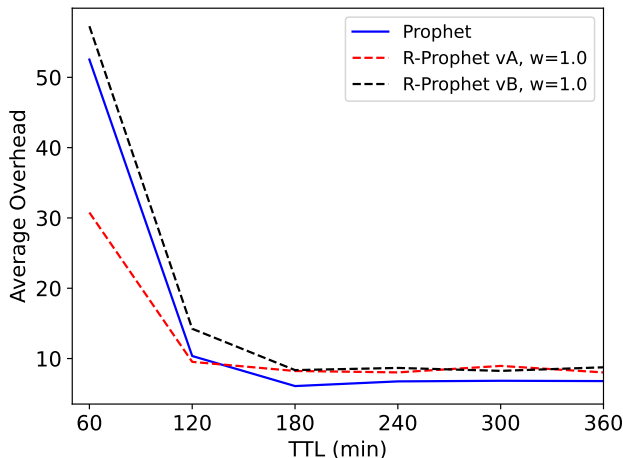


Fig. 5: Average delivery ratio comparison

For the rest of the TTL values, the overhead is around 10 times for all protocols, PRoPHET producing the lowest overhead. The reason is because of the forwarding strategy in

PRoPHET and R-PRoPHET-vA that relays fewer messages at the beginning of the simulation than R-PRoPHET-vB. Regarding the similar overhead amount between R-PRoPHET-vA and R-PRoPHET-vB, even when each version applies a different forwarding strategy, we could say that R-PRoPHET-vA compensates the overhead reducing the number of relayed messages at the beginning of the simulation. In contrast, R-PRoPHET-vB compensates the overhead increasing the number of delivered messages since the average latency is reduced as was explained previously.

VIII. CONCLUSIONS

In this article, we introduced the R-PRoPHET protocol to provide an intermittently connected network in rural environments, where the variable mobility of vehicles requires flexibility to model the encounters between nodes. We evaluated two forwarding strategies to increase the delivery ratio, but it could increase the amount of relayed messages along the network. A comparison between R-PRoPHET and the original PRoPHET showed that R-PRoPHET increases the delivery ratio by up to 20% and reduces the average latency by up to 16% compared to PRoPHET. The overhead produced by R-PRoPHET is around two times more than PRoPHET, which is part of the trade-off of increasing the average delivery ratio and reducing the average latency.

In future work, we will include traffic generated from multiple villages and incorporate a resource allocation strategy to distribute the messages along the network, avoiding overloading the critical relay nodes. We will evaluate the R-PRoPHET performance in the new extended scenario.

REFERENCES

- [1] A. Lindgren, A. Doria, E. B. Davies, and S. Grasic, "Probabilistic Routing Protocol for Intermittently Connected Networks," RFC 6693, Aug. 2012.
- [2] S. D. Han and Y. W. Chung, "An improved PRoPHET routing protocol in delay tolerant network," *ScientificWorldJournal*, p. 623090, Jan. 2015.
- [3] T.-K. Huang, C.-K. Lee, and L.-J. Chen, "Prophet+: An adaptive prophet-based routing protocol for opportunistic network," in *IEEE Inter. Conf. on Advanced Information Networking and Applications*, 2010.

- [4] P. Sok, S. Tan, and K. Kim, "PRoPHET Routing Protocol Based on Neighbor Node Distance Using a Community Mobility Model in Delay Tolerant Networks," in *IEEE International Conference on Embedded and Ubiquitous Computing*, 2013.
- [5] R.-E. Mezouary, A. Houmz, J. Jalil, and M. E. Koutbi, "Prophet-raip5: A new approach to secure routing in wireless sensor networks," in *2015 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 2015, pp. 1–6.
- [6] T. Akhter and M. S. Hossen, "An association rule based prophet (arbp) routing protocol in an opportunistic network," in *Int. Conf. on Recent Progresses in Science, Engineering and Technology (ICRPSET)*, 2022.
- [7] N. ChikkaKrishna, P. Rachakonda, and T. Tallam, "Short - Term Traffic Prediction Using Fb-PROPHET and Neural-PROPHET," in *2022 IEEE Delhi Section Conference (DELCON)*, 2022, pp. 1–4.
- [8] O. Triebe, H. Hewamalage, P. Pilyugina, N. Laptev, C. Bergmeir, and R. Rajagopal, "Neuralprophet: Explainable forecasting at scale," 2021.