



# Dynamic selection of relays based on classification of mobility profile in a highly mobile context

Philippe Fabian, Abderrezak Rachedi

## ► To cite this version:

Philippe Fabian, Abderrezak Rachedi. Dynamic selection of relays based on classification of mobility profile in a highly mobile context. ICC 2020 - 2020 IEEE International Conference on Communications (ICC), Jun 2020, Dublin (virtual conference), Ireland. pp.1-6, 10.1109/ICC40277.2020.9149423 . hal-02972895

**HAL Id: hal-02972895**

**<https://hal.science/hal-02972895>**

Submitted on 28 Oct 2020

**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.

# Dynamic selection of relays based on classification of mobility profile in a highly mobile context

Philippe Fabian\*, Abderrezak Rachedi\*

\*University Paris-Est, LIGM (UMR8049), UPEM, F-77454, Marne-la-Vallée, France

Emails: [philippe.fabian][rachedi]@u-pem.fr

**Abstract**—With an increasing amount of connected cars and devices having more and more sensors, the development of smart architectures and algorithms to efficiently transport data is a major concern. The selection of relays to allow users to connect to Internet is an important aspect in networks with high mobility, particularly in low-population areas having poor network coverage. Furthermore, cellular connectivity can be expensive for users. The solution proposed in this paper uses a machine learning based classification algorithm to select the best relays amongst any user based on their mobility profile. Not only this solution can be used on its own to enhance network performance without requiring a dedicated architecture, but it can be coupled with other algorithms as well to increase performance even more. Simulation results will show the proposition is able to scale up to several hundreds of users simultaneously, it improves the delivery rate of packets by up to a factor 2, it increases connectivity, generates less signaling and yields a more stable topology compared to a random selection or the use of static relays.

**Index Terms**—Mobile Relays, Machine Learning, Mobility Classification, Mobile Networks, Internet of Mobile Things

## I. INTRODUCTION

Wireless mobile devices are more and more present in everyday life. All kinds of vehicles are equipped with sensors (cars, trucks, buses, bicycles, scooters, etc.) and are connected to Internet. Many people also carry a smartphone. All these devices generate traffic and the expected traffic from mobile users will reach 20% of the total IP traffic in the near future [1]. Also, most of the mobile traffic will be from cellular connections [1]. Networks must be designed accordingly in order to absorb such a huge quantity of data and to ensure Quality of Service (QoS). Although several users can directly connect to Internet using a 4/5G connection, it is not necessarily possible for everybody. High costs, lack of connectivity due to environmental factors or poorly serviced areas are a few examples where connecting a user can be an issue. One solution is to use relays to forward data from and to those users to allow them to have network access. Selecting which relays to use will have a major impact on performance and in specific contexts it is not always possible to have dedicated static relays (e.g. in a low populated area), as the cost to deploy them is too high compared to the amount of (potential) users. Furthermore, in a context where all users move, like the Internet of Mobile Things (IoMT), the constantly changing topology poses an even greater challenge to the selection process of relays. Indeed, some users can move far enough to leave the coverage area or a traffic jam can cause a section of a road to suddenly have a peak of network traffic due to a sudden increase in the number

of users. In such situations, it can be hard to decide where to place static relays and relaying on Mobile Relays (MR) instead can greatly help in adapting the topology to this very dynamic context, without incurring a higher cost.

The selection of relays is a problem that was extensively studied [2]. Most solutions focus on the selection of relay users connect to (detailed in section II). Relays - mobile or not - are usually dedicated devices as well. Several researchers propose solutions using Artificial Intelligence (AI), more specifically Machine Learning (ML) : such techniques allow to efficiently solve complex problems thanks to their capacity to adapt to dynamic contexts like in the case of mobility in wireless networks [3]. ML is used in networking to address different problems, like routing [4]. We have decided to focus on the process of selection of relays.

In this paper, we propose a novel solution that uses ML to classify users according to their mobility profile. Then, the best profiles amongst users are selected to act as mobile relays. Contrary to the state of the art, our solution does not require a dedicated infrastructure; it is thus able to adapt to a vast amount of different situations. A cellular antenna is still needed, though, as a mean to connect to Internet. Once MRs are selected, any routing protocol can be used to transport data. It is also possible to narrow down the set of selected MRs to improve performance even more by considering connectivity, lowest delay, energy consumption or any other metric seen as relevant. This makes the proposed solution suitable to couple with an existing routing protocol to greatly increase QoS. Furthermore, in this paper we study different mobility profiles together (car and bicycle), interacting with one another at the same time. To the best of our knowledge, other solutions consider motorized vehicles only which makes our approach original.

The remaining of this paper is organized as follows : state of the art is presented in section II, section III introduces the studied architecture and the proposed solution, simulation results are shown and discussed in section IV and the paper is concluded in section V.

## II. RELATED WORK

As we will see in this section, most algorithms focus on the selection of dedicated relays or some information is supposed to be known *a priori* to aid the selection process. The first part (section II-A) is about classical algorithms whereas the second part (II-B) will cover algorithms using machine learning.

### A. Classical algorithms

Classical algorithms are solutions that do not use ML to solve the problem at hand. Some examples include the use of matching theory [5], Markov chains [6] [7] or formulating the question as a maximization problem [8].

Authors in [5] use Matching Game to anticipate future radio conditions of flying drones. The position and trajectory of drones are used to dynamically adjust the transmission mode of each one and to select which drones will forward data. However, in the context of vehicular it is not always possible to know the future trajectory of users.

In [6], the authors propose a relay selection scheme where Mobile Users (MU) select relays according to a cost. The model is composed of a base station, several relays and MUs. The mobility of users is represented by a transition matrix and the problem is modeled using a constrained Markov decision process. Though the mobility of users is accounted for in this proposition, the use of dedicated relays constrains the problem.

The solution proposed in [8] aims to maximize the system throughput by satisfying QoS requirements of users by considering a power constraint. The model is composed of one antenna, several fixed relays and users. The maximization problem allows to find the best relay for each user. However, given relays are dedicated units and no mobility is considered, the solution is not adapted in a highly mobile environment with potential relays moving in and out of the studied area.

In [7], the authors propose to add MRs to help manage high traffic periods of fixed relays in order to reduce signaling overhead and improve user mobility experience. High traffic periods are modeled using Markov chains. Though mobility is considered, MRs are supposed to be mounted on vehicles, such as buses and move at constant speed. This might not be adapted in areas with a low-population density.

Authors in [9] propose a mechanism of relay selection to assist nodes showing high interference. The model is made of a macro cell antenna, a few pico cell antennae and several users. This solution is interesting as any node can potentially relay data to help another node. However, it uses static nodes and is not intended to use in a context with high mobility.

### B. Machine learning based algorithms

ML algorithms use artificial intelligence to solve problems. In the case of supervised ML, the algorithm is first trained on a set of data where the output is known in advance. After this step, the algorithm is able to deduce the output on a set of data having the same nature as the training set.

In [10], the authors propose to use deep learning to select the best relays to enhance dissemination of data. The studied case is made of a road network with static Road Side Units (RSU) near the roads and vehicles moving around on the roads. Although the use of several features is good to make a more relevant choice, relays are static and dedicated units. This makes this solution non-adapted if deployment cost of the infrastructure is too high or if it is not possible to deploy a dedicated architecture at all.

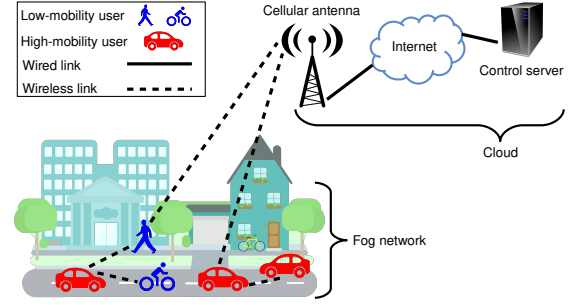


Fig. 1: The proposed architecture with several types of users and the controller running the ML algorithm.

Authors in [11] propose to use deep learning to select the most appropriate relay for nodes suffering blocking of radio waves. A user that has bad radio conditions will consider several metrics for the deep learning algorithm to find a suitable relay. However, the presented architecture does not consider a user can connect to the base station through several hops.

In [12], the authors propose a powerful model using the K-nearest neighbors ML algorithm. They consider a vehicular network composed of several RSUs and many vehicles driving around. The idea is to connect a vehicle directly to a RSU if one is in range, otherwise it tries to find a path through other vehicles to connect to a RSU using several features. Though the model uses many relevant features in the context of mobility, it relies on static relays and only motorized vehicles are considered.

As aforementioned, state of the art solutions either require dedicated relays or make use of some information considered as known beforehand. We are convinced a more general approach is key in improving performance and in adding a great flexibility. This is why we propose a solution that only requires one cellular antenna to connect to Internet and a few users with cellular access to this antenna.

## III. ARCHITECTURE AND CLASSIFICATION ALGORITHM

We first present the studied architecture (section III-A) before introducing the classification algorithm (section III-B).

### A. Proposed architecture

As depicted in figure 1, we are in the context of IoMT. Different types of user are supposed to be present in the environment : soft mobility users such as bicycles or pedestrians and high mobility users like cars, trucks or buses. They can either connect to a nearby cellular antenna or they can connect to other users on the network. These users make the *fog* part of the architecture (see fig. 1). A user directly connected to the antenna can act as a mobile relay. Users that cannot directly connect to the cellular network connect to the MRs or to another user. The resulting topology is a tree where relays act as root of the topology they manage (it is the same kind of topology as in the Routing Protocol for Low-power and

Lossy Networks (RPL), RFC 6550 [13]). The other part of the architecture is the *cloud* part (as shown in fig. 1). It is composed of the cellular antenna used by users in the fog network and a control server (or controller). The antenna is connected to the controller via Internet with a wired link (fiber optic, for instance). The control server gathers users' data and runs the ML algorithm using this data (this is explained below, in section III-B).

Once the prediction is done, the topology can be built : figure 1 shows a network already set up : 2 users (1 car and 1 pedestrian) are directly connected to the antenna and they act as relay. The other users are connected to these relays or intermediate users to reach the antenna. In this paper, we focus on how the controller selects MRs in the fog part of the architecture. Given the topology is based upon RPL, the way a node (user) selects a parent to connect to is left to the developer. We have used a routing algorithm from previous work [14] that we will not detail here, as it is out of the scope of this article. The following section presents the proposed ML algorithm more in detail.

### B. Classification algorithm based on mobility profile

The proposed solution uses supervised ML, more specifically decision trees. Supervised learning for classification works by first training the algorithm on a problem where the classes are already known. A decision tree is a binary tree such that on each node a comparison is made : if the answer is true go to left branch (or right), otherwise go to right branch (or left). In our case, the input of the algorithm is data gathered from users such as speed, acceleration, position (i.e. type of road the user is on), etc. These inputs are the features (see section II-B). The output is the mobility class (car, bicycle, pedestrian). Once this step is done, it is possible to determine which users are the best candidates to serve as relays. A "best candidate" can be a user with soft mobility (a pedestrian, perhaps, or a static node), a device lightly used (e.g. high available resources, low buffer occupancy) or any other target device suited for the current context.

We define a network composed of  $U$  users and  $C$  different classes of mobility. Each user  $u$  stores a vector  $M$  of mobility-related metrics :

$$M_u = [m_0, m_1, \dots, m_\mu] \quad (1)$$

Where  $\mu$  is the number of different metrics (speed, acceleration, and so on). Each user periodically records values for these metrics (recording of speed, position, etc.). Depending on the ML algorithm used, different metrics (features) can yield a different expected accuracy (output). Hence, we introduce the vector of relevant metrics  $M_u^p$  for user  $u$  :

$$M_u^p \subseteq M_u \quad \text{s.t.} \quad |M_u^p| > 0 \quad (2)$$

$M_u^p$  can be composed of all metrics from equation (1). At least one metric must be used otherwise no input will be fed to the prediction algorithm. We suppose the recorded values for the metrics of equation (1) are sent to the controller after a time  $\tau$  that we call the *prediction time*. We define the ML algorithm

that takes as arguments the vector of relevant metrics and the prediction time :

$$c_u^p = A(M_u^p, \tau) \quad (3)$$

Where  $c_u^p \in C$  is the predicted class of user  $u$ . A user can either be correctly classified or wrongly classified. Let  $c_u^r \in C$  be the real mobility class of user  $u$  :

$$c_u^p = \begin{cases} c_u^r & \text{if the prediction is correct} \\ c^x \in (C \setminus c_u^r) & \text{if the prediction failed} \end{cases} \quad (4)$$

From equation (4), we see the predicted class ( $c_u^p$ ) can either be the real class of the user ( $c_u^r$ ) or some other class which is not the right one (depicted as  $c^x$  in eq. (4)). As aforementioned, the resulting accuracy of the considered algorithm depends on the time since a user exists. Indeed, more data is gathered if the user has been on the network for a longer duration and the predicted class usually<sup>1</sup> has a higher probability of being the correct one.

Now, we define  $C_b \in C$  as the best mobility class. Users belonging to this class are determined by equation 4. To act as relay, a user must have access to Internet (for instance via 5G). We define  $I \subseteq U$  the set of users with Internet access. Thus, only users within the correct mobility class and with access to Internet can potentially be selected as relays :

$$R = I \cap C_b \quad (5)$$

Where  $R$  is the set of potential relays. It is now possible to determine the set of selected relays  $R_s$  that will serve as relays in the network :

$$R_s \subseteq R \quad \text{s.t.} \quad |R_s| > 0 \quad (6)$$

From equation 6, we see there can be one relay or as many as the size of  $R_s$ . This number will depend on the context and constraints. If a high connectivity is of paramount importance, more relays might be desirable. However, if the number of users and traffic are low, having only a few relays can be better. Note that  $R_s$  can be small if the number of users with Internet access is small and/or if few users have the best profile as determined by function  $A(t)$  (see eq. 4). If no users are in the set  $R$  (eq. 5), non-optimal users have to be selected. Note that a non-optimal user can end up in  $R$  if the prediction failed (eq. 4). Hence, if the predictive algorithm is not adapted at all to the current context QoS might suffer greatly.

Finally, if we set the gathered metrics of a user as constant, the probability the algorithm from equation (3) yields a correct prediction is proportional to  $\tau$  :

$$P(c_u^p = c_u^r) \propto \tau \quad (\text{with } M_u^p \text{ constant}) \quad (7)$$

Algorithm 1 illustrates the use of the proposed solution. It is called whenever the number of relays is less than the specified amount and a delay of  $\tau$  has elapsed. The algorithm is supposed to be called with two parameters : the maximum

<sup>1</sup>In some cases, for instance in a heavy traffic jam, it might be hard to discern from the different classes of mobility even if a long time  $\tau$  is used by the algorithm.

**Algorithm 1** The machine learning algorithm to classify users depending on their mobility profile.

```

1:  $maxRelays \leftarrow arg[0]$ 
2:  $relevantMetrics[] \leftarrow arg[1]$ 
3:  $relays \leftarrow NULL$ 
4: for all users do
5:    $user.gatherData()$ 
6: end for
7: for all users do
8:   if  $A(user, relevantMetrics) == optimal$  then
9:      $relays.append(user)$ 
10:  end if
11:  if  $relays.size() == maxRelays$  then
12:    break
13:  end if
14: end for
15: return  $relays$ 

```

Delay (s)	2	4	6	8	10
Accuracy (%)	20	27	88	79	86

TABLE I: The accuracy of the predictive algorithm given the delay to gather data.

number of mobile relays and the vector of relevant metrics ( $arg[0]$  and  $arg[1]$  in algorithm 1, respectively). Lines 1 to 3 are the initialization steps, where  $relays$  is the set of selected relays (eq. 6). The users' data are sent to the controller (lines 4 to 6; see section III-A). Lines 7 to 14 is the prediction step : the relevant data of all users are used to predict their mobility class (line 8; eq. 3 and 4). If the current user is predicted as being in the best mobility class, it is added to the set of selected relays (line 9; eq. 6). The algorithm stops when the target amount of mobile relays is reached (line 11 in algorithm 1). If there are more candidates than the amount of required relays, users not connected to Internet (or not in range of a parent able to connect to Internet) have priority to increase connectivity. Note that in the case not enough relays are selected, non-optimal users will have to be selected.

#### IV. PERFORMANCE EVALUATION

We have evaluated the proposed solution using the OMNeT++ simulator. We also used the Simulation of Urban MObility (SUMO) to have more realistic results.

##### A. Simulation setup

The simulation was run using OMNeT++ 5.4.1 and SUMO 0.32.0. The Veins framework was used to interface OMNeT with SUMO. We have imported a real map from OpenStreetMap<sup>2</sup> into SUMO and we have generated urban traffic using a tool from the SUMO package. Veins allows to run an OMNeT simulation using the mobility patterns of users (cars, bicycles) generated from SUMO to create and move nodes in a more realistic way. That is, each user in SUMO

<sup>2</sup>The real map is a sub-area of the *Saint-Laurent* borough, Montreal, Canada.

Parameter	Value
Playground size	1 km <sup>2</sup>
User generation period	1.5 s
Num. of users, total	$\pm 4800$
Num. of users, inst.	10's to 100's
Num. of relays <sup>†</sup>	5
Car:bike ratio	9:1
Simulation duration	8000 s
Traffic model	VBR
Traffic density	10 packets/s
Data packet size	1280 bytes
Max. user throughput	10 Mbps

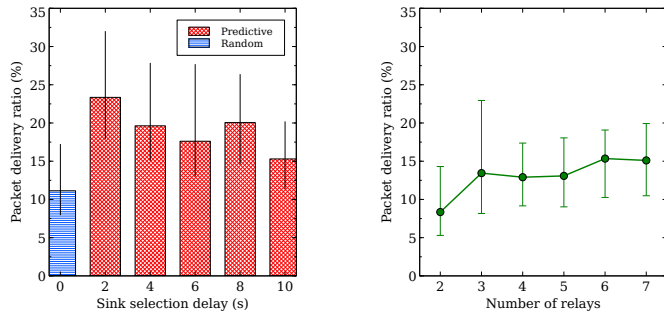
TABLE II: Values for the most relevant parameters of the simulations. <sup>†</sup>Number of relays is set for the predictive case and varies from 2 to 7 in the case of static dedicated relays.

will have a corresponding entity in OMNeT (a “node” in our case). Thus, OMNeT will create one “node” from each user in SUMO and destroy those when they reach their destination. The ML model was trained using a real map different from the the map used to test the model (i.e. run the simulation) : this allows to test the generalization of the studied algorithm. Table I shows the accuracy of the decision tree in correctly predicting the mobility class of users. A good prediction means a car, for instance, is predicted as a car. A wrong prediction means a bicycle, for instance, is predicted as something else than a bicycle. The “Delay” column in table I corresponds to the time data is gathered from users before making the prediction (see section III). When the delay is very short (2s for instance), the error is rather high because the user was not active long enough. For example, a car just starting might stop at a red light right away and it is not possible to differentiate it from a bicycle on such a short notice. However, although a longer delay allows for better accuracy, it also means a longer time before choosing a new sink. During this time, no connectivity will degrade QoS and might not be the right choice. This is why a compromise must be found between accuracy and delay to yield the best QoS possible. The predictive algorithm is compared to a random selection of sinks. The random selection serves as a base case and has the advantage of a zero delay to choose a new sink. The random choice is not represented in table I. We also ran the simulation without prediction using static relays (from 2 to 7) to compare the proposed solution with a solution using a dedicated infrastructure. They are placed randomly on the simulated area.

We suppose all users can connect to Internet (directly via cellular network or not) to send their mobility-related data. Users without cellular connectivity will connect to other users and eventually transmit these data to the controller.

##### B. Result analysis

In the following, bar charts (figs. 2a, 3a, 4a and 5) represent results using dynamic relay selection amongst users. The horizontal axis is the time taken to make the prediction. The leftmost bar is the random selection (selection delay of 0s) and other bars represent the ML algorithm (selection delay of 2s to 10s). The line plots (figs. 2b, 3b and 4b) represent results



(a) The PDR given the allowed prediction time. (b) The PDR in the case of static relays.

Fig. 2: The average PDR of the data packets of users.

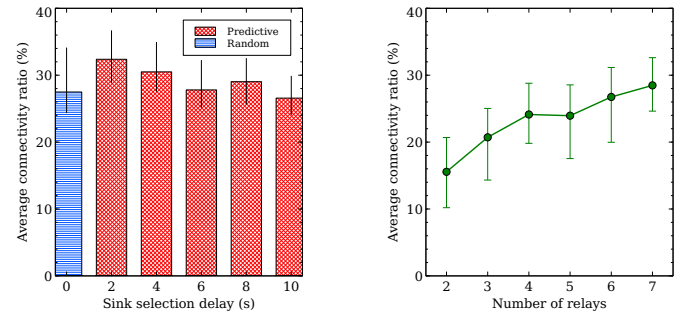
without prediction using static dedicated relays. The horizontal axis is the number of relays placed on the area. We recall the reader simulations with dynamic relay selection are achieved using 5 relays (see table II).

1) *Packet Delivery Ratio*: The PDR is the amount of data packets that reach the destination divided by the amount of data packets sent. This metric allows to understand the reliability of the network. The higher the PDR the better.

Figure 2 shows PDR results of the different simulations. The random selection (fig. 2a) yields a PDR of about 11%. Considering error bars, the proposed solution performs significantly better when the selection delay is 2s (fig. 2a) because the selecting of soft-mobility users as relays allows the topology to be more stable, reducing the amount of packet loss. PDR is potentially higher when using ML and tends to decrease as the selection delay increases because waiting longer to choose a new sink causes data packets to be rerouted after a longer delay, increasing the probability of packet lost. Static relays (fig. 2b) have lower PDR than the predictive solution because users move around and those relays eventually service no users at all. Those results are comparable to the random selection. When 2 relays are used, results from figure 2b show a PDR significantly lower than the proposed solution with a delay of 2s and 4s : static relays have a limited coverage, so only 2 yields low performance. More relays, in this case, help to increase performance, as shown in figure 2b. Note that in figure 2a the lowest PDR value with a selection time of 2s is 17.9865% whereas in figure 2b the highest PDR value with 5 relays is 18.0529%, so there is a slight overlap.

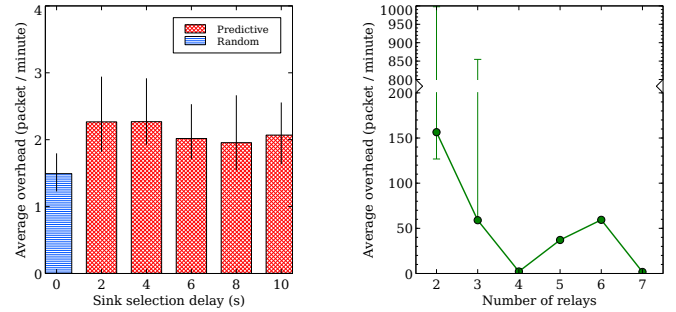
2) *Connectivity ratio*: The connectivity ratio is the percentage of time a user is connected to a sink during her trip. For instance, a connectivity of 25% on a 8000s run means the user was connected during 2000s in total. Connectivity is linked to PDR, as more connectivity increases the chance for packets to reach their destination though it is not necessarily the case, as a non-connected user will not try to send packets. This metric allows to see the coverage of the studied area and if sinks/relays are well placed. The higher the connectivity ratio the better.

The connectivity ratio is depicted in figure 3. The proposed solution is comparable to the random selection (fig. 3a) with a



(a) The connectivity given the allowed prediction time. (b) The connectivity when using static relays.

Fig. 3: The average connectivity of users during their trip.



(a) The amount of signaling given the allowed prediction time. (b) The amount of signaling in the case of static relays. Note the cut on the vertical axis.

Fig. 4: Study of the amount of signaling packets (excluding acknowledgments) in two different scenarii. Note vertical axes have different scales.

connectivity of about 30%. Indeed, mobile relays are located such that there is similar connectivity no matter the selection method. However, with a prediction time of 2s or 4s, the proposed solution performs significantly better when compared to static relays (fig. 3b) when using 2 or 3 relays. This is caused by the fact static relays (fig. 3b) eventually spend some time not being in range of users. A longer selection delay will negatively impact the predictive algorithm because users will spend more time disconnected.

3) *Amount of signaling*: The amount of signaling represents to number of packets sent on the topology excluding data and acknowledgment packets. Signaling is mostly used to set up and maintain the topology. Sending a signaling packet again because it was not acknowledged counts in the amount of signaling. This metric shows how much overhead is generated and how good transmissions are. The lower the signaling the better.

We can see in figure 4 the random selection performs significantly better (about 1.5 packets per minute per user on average) than the proposed solution (around 2.25 packets per minute per user on average) when the selection delay is 2s or 4s (see fig. 4a). The lower overhead of the random selection is due to lower performance in terms of PDR (fig. 2) and



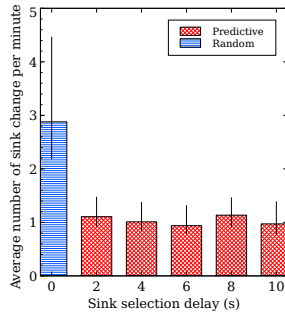


Fig. 5: Stability of the topology in terms of the rate of sink change.

connectivity (fig. 3). Indeed, less signaling will be sent in absolute if users are less connected to one another. In figure 4b, the amount of signaling is very high when there is (are) 1 or 2 relay(s). This is due to low coverage causing long paths (i.e. high number of hops from any user to the sink) with higher probability of retransmissions. Long paths also increase the probability of any user moving away causing the need to repair the topology. Transmission distances are longer as well because relays are static, causing many losses. The proposed solution is significantly better than the case of static relays (fig. 4b) except when the number of relays is 4 or 7, with overheads of about 2.2 packets per minute and 1.7 packets per minute respectively. A high number of overhead does not necessarily means performance is bad (see PDR in fig. 2). However, the scaling will be problematic if overhead is very high, as is the case of 1 and 2 relay(s) (fig. 4b).

4) *Number of sink change*: When dynamic selection of sinks happens, this metric shows how many times sinks are replaced. When an elected sink leaves the area of interest or fails due to low battery, a new one will be chosen. This metric shows the stability of the topology. The lower the number of changes the better.

Figure 5 shows the predictive algorithm performs significantly better in all instances. The random selection process causes more changes of sink. This is due to selecting high mobility users (cars) more often that leave the area of interest quicker, so a change is required. The proposed solution yields more stability as the number of changes is below 1.5 per minute on average. This means that one new sink is chosen approximately every 40s on average, compared to the random case (around 3 changes per minute) where one sink is chosen every 20s on average.

## V. CONCLUSION

The solution proposed in this paper aims at determining the mobility class of users to select the best relays using a machine learning based approach, without requiring a dedicated infrastructure. This solution can be coupled with other algorithms to further improve network performance. Simulation results show the proposed algorithm increases performance in terms of PDR and stability compared to a random selection. The contribution also performs better than an approach with

static relays in terms of PDR, connectivity and overhead and it is able to scale from several tens to several hundreds of users simultaneously in different scenarios. Future works will focus on extending the model with more mobility classes (e.g. pedestrians) and different traffic densities.

## ACKNOWLEDGMENTS

We thank Ozgur Turk who wrote the code to interface data generated by SUMO with the machine learning algorithm, in order to train and test it.

## REFERENCES

- [1] C. V. N. Index, "Global mobile data traffic forecast update, 2016–2021 white paper, accessed on may 2, 2017."
- [2] K. Shamganth and M. J. Sibley, "A survey on relay selection in cooperative device-to-device (d2d) communication for 5g cellular networks," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. IEEE, 2017, pp. 42–46.
- [3] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications Surveys & Tutorials*, 2019.
- [4] J. Jiang, V. Sekar, I. Stoica, and H. Zhang, "Unleashing the potential of data-driven networking," in *International Conference on Communication Systems and Networks*. Springer, 2017, pp. 110–126.
- [5] D. Liu, J. Wang, Y. Xu, Y. Xu, Y. Yang, and Q. Wu, "Opportunistic mobility utilization in flying ad-hoc networks: A dynamic matching approach," *IEEE Communications Letters*, vol. 23, no. 4, pp. 728–731, 2019.
- [6] Y. Li, P. Wang, D. Niyato, and W. Zhuang, "A dynamic relay selection scheme for mobile users in wireless relay networks," in *2011 Proceedings IEEE INFOCOM*. IEEE, 2011, pp. 256–260.
- [7] Z. Liao, J. Liang, and C. Feng, "Mobile relay deployment in multihop relay networks," *Computer Communications*, vol. 112, pp. 14–21, 2017.
- [8] M. S. Alam, J. W. Mark, and X. S. Shen, "Relay selection and resource allocation for multi-user cooperative ofdma networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 5, pp. 2193–2205, 2013.
- [9] N.-N. Dao, M. Park, J. Kim, J. Paek, and S. Cho, "Resource-aware relay selection for inter-cell interference avoidance in 5g heterogeneous network for internet of things systems," *Future Generation Computer Systems*, vol. 93, pp. 877–887, 2019.
- [10] A. Mchergui, T. Moulahi, and S. Nasri, "Relay selection based on deep learning for broadcasting in vanet," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2019, pp. 865–870.
- [11] A. Abdelreheem, O. A. Omer, H. Esmail, and U. S. Mohamed, "Deep learning-based relay selection in d2d millimeter wave communications," in *2019 International Conference on Computer and Information Sciences (ICCIS)*. IEEE, 2019, pp. 1–5.
- [12] Y. Zhou, H. Li, C. Shi, N. Lu, and N. Cheng, "A fuzzy-rule based data delivery scheme in vanets with intelligent speed prediction and relay selection," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.
- [13] T. Winter, P. Thubert, A. Brandt, J. Hui, and R. Kelsey, "Rfc 6550: Rpl: Ipv6 routing protocol for low-power and lossy networks (2012)," Tech. Rep.
- [14] P. Fabian, A. Rachedi, C. Gueguen, and S. Lohier, "Fuzzy-based Objective Function for Routing Protocol in the Internet of Things," in *IEEE Global Communications Conference (GlobeCom)*, Abu-Dhabi, United Arab Emirates, Dec. 2018. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01956385>