Exploring Wi-Fi Network Diversity for Vehicle-to-Infrastructure Communication

Rui Meireles
Vassar College
Poughkeepsie, NY, USA

António Rodrigues
Instituto de Telecomunicações
Porto, Portugal

Andrei Stanciu
Vassar College
Poughkeepsie, NY, USA

Ana Aguiar
University of Porto
Instituto de Telecomunicações
Porto, Portugal

Peter Steenkiste
Carnegie Mellon University
Pittsburgh, PA, USA

***Abstract*—Due to their ubiquity and low use-cost, the opportunistic use of Wi-Fi networks to offload data from moving vehicles is enticing. However, due to their limited coverage and variable performance, choosing what Access Points (APs) to use in order to maximize the amount of data that can be offloaded is challenging. This difficulty is exacerbated by the heterogeneity created by the introduction of new Wi-Fi standards such as 802.11ad, which renders heuristics designed for homogeneous environments, e.g., signal quality, ineffective. In this work we test the hypothesis that historical network performance, indexed by vehicular mobility information, can be used to effectively forecast future network performance, and consequently help select APs for data offloading in a heterogeneous Wi-Fi environment. Our approach was to perform a trace-based analysis on experimental data collected in a realistic vehicular environment. Our results show that a practical algorithm based on data rate forecasting from mobility information was able to transfer at least 80% of the optimal amount of data, under the tested scenarios.**

***Index Terms*—wireless communication, vehicular communication, data rate forecasting, AP selection**

I. Introduction

Vehicle-to-Infrastructure (V2I) communication today relies mostly on cellular connectivity. However, while 4G and 5G cellular provide fast and, in urban areas, pervasive service, they can be expensive for clients. For operators, network capacity can be an issue, especially as bandwidth requirements increase with the deployment of new driver-assistance and autonomous driving systems. Wi-Fi is a less expensive, and sometimes free, alternative, and urban area coverage is extensive. For example, a large-scale data collection campaign done in Porto, Portugal [1], found the mean number of visible APs per street traversal to always be larger than zero, as shown in Fig. 1. Moreover, 80 % of streets contained at least one public AP, e.g., part of a commercial hotspot network such as Fon [2].

This work is a result of projects MobiWise (SAICTPAC/0011/2015, POCI-01-0145-FEDER-016426), S2MovingCity (CMUP-ERI/TIC/0010/2014), and FCT/MCTES UIDB/50008/2020, all funded by Portuguese and EU funds: Fundo Europeu de Desenvolvimento Regional (FEDER) through Programa Operacional Competitividade e Internacionalização (POCI), and Fundação para a Ciência e Tecnologia (FCT). Additional support was provided by FCT through the Carnegie Mellon Portugal Program under grant SFRH/BD/52161/2013.

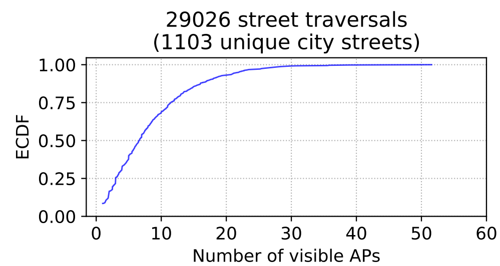


Fig. 1: ECDF of the mean number of visible APs per street traversal observed in Porto, Portugal (2016-2019).

Wi-Fi networks are already used for communication with parked vehicles, e.g., for software updates as done by Tesla [3]. While a number of research projects have explored the use of public Wi-Fi for moving vehicles (§V), it has yet to see actual deployment, because it is challenging. First, due to their limited radio range and required association process, the amount of time during which a Wi-Fi network can be used by a moving vehicle is often very short. Second, due to the unsupervised nature of deployments, performance can differ dramatically across networks, and is often poor. Finally, Wi-Fi security may further slow down association, and potentially even prevent clients from using a large portion of networks.

Many of the studies on the use of Wi-Fi for vehicular networking are more than 10 years old (§V). Since then, new Wi-Fi standards have emerged, offering dramatic improvements in throughput: 802.11ac, ad, and more recently ax and ay. For these standards, even short periods of connection may enable the transmission of hundreds of megabytes of data. However, these developments create new challenges for vehicular communication, namely increased diversity. While the older Wi-Fi standards - b/a/g/n - are relatively similar, albeit with distinct bit rates, newer versions - ac/ad/ax/ay - are very different. Besides differences in bit rate, they use very distinct frequencies (2.4, 5, and 60 GHz, impacting range), different radiation patterns (impacting range and requiring training), and require more detailed channel state information (introducing overhead). Previous work often used signal strength as the main heuristic to guide AP selection, but this is unlikely to be effective given the current diversity in technologies.

This paper looks at the question of AP selection for V2I communication in an heterogeneous Wi-Fi environment. Our goal is to determine which AP a vehicular client should be connected to at each point in time, in order to maximize the total amount of data it can offload over Wi-Fi. To do so, we follow a two-step approach: first, we predict the data rate that can be offered by the different APs on the vehicle's path; then, we use these predictions to determine, at each point in time, whether the client should remain connected to the current AP, or initiate a switch to a different one.

We hypothesize that past data rate observations, indexed by the client's current mobility, are a good predictor of future data rates. And that these are good predictors for throughput. The intuition is this data-driven approach captures the cumulative effects of all factors affecting network performance, such as buildings and terrain. Note that prior work has explored the use of historical data for AP selection [4], [5], but relied solely on signal quality as a performance indicator.

Our study is based on a set of traces with detailed information on V2I communication using different Wi-Fi standards – n/ac/ad. The traces were collected by a vehicle with three Wi-Fi interfaces (one per Wi-Fi type) communicating in parallel with infrastructure APs. The vehicle was driven along a circuit to collect multiple measurements for each road segment.

In summary, we make the following contributions:

- Collect spatially-indexed performance measurements in a Wi-Fi-diverse vehicular environment (§II).
- Propose and evaluate multiple Wi-Fi data rate forecasting algorithms that leverage both real-time and historical mobility and performance information (§III).
- Propose and evaluate multiple data rate forecasting-based Wi-Fi AP selection algorithms (§IV).

II. Experimental data collection

Experimental setup: We used an intersection in a residential area of Gaia, Portugal as the stage for our experiments, as per Fig. 2a. To focus on Wi-Fi diversity, we co-located 3 APs at the intersection's southwest corner, each supporting a different Wi-Fi standard: 802.11n, ac, and ad. The three technologies operate in different frequency bands, 2.4, 5, and $60\,\mathrm{GHz}$, as shown in Tab. I. All APs were powered by, and mounted on the roof of, a parked vehicle. The client equipment was mounted on the roof of a second vehicle – Fig. 2b. We had three pairs of Wi-Fi interfaces, one for each network. Out of each pair, one interface was used for communication. The other was a monitor, running `tcpdump` to capture 802.11 frames.

Pseudo-random data was sent from each AP to its respective client, at a rate capable of saturating the network. To avoid potential periods of inactivity from flow and congestion control, packets were exchanged over UDP. During the experiments, all devices were controlled and monitored through a separate 802.11n control network, operating on a non-overlapping $2.4\,\mathrm{GHz}$ channel. This was also used to synchronize clocks over NTP, allowing for time-based fusion of the collected data.

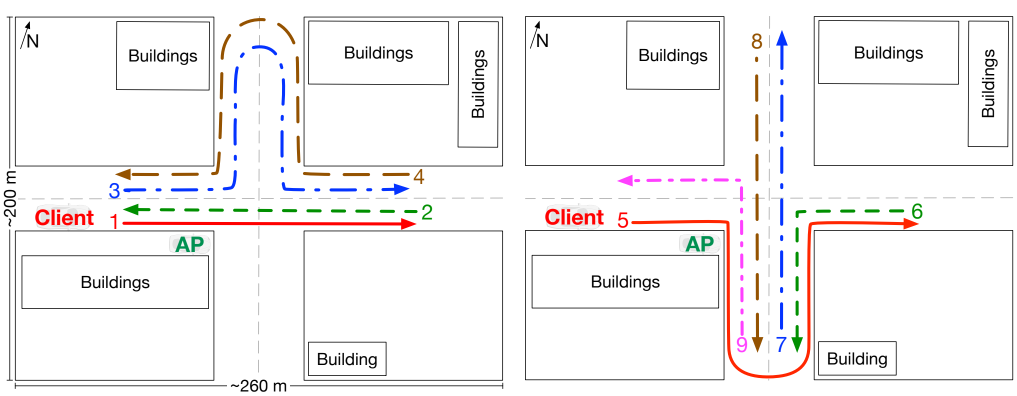
To maximize trajectory diversity, the client vehicle drove laps around the intersection using the circuit pattern depicted in Fig. 2a, for a total of around $1\,\mathrm{h}$ and $45\,\mathrm{min}$. We performed this experiment twice, once on a Tuesday, and once on a Thursday. The latter experiment differed from the former in two ways: (i) the AP vehicle was $\sim 30\,\mathrm{cm}$ taller, and (ii) we spent more time at slower speeds and closer to the APs, in an attempt to better understand 802.11ad connectivity. This is reflected in the client speed and distance ECDFs of Figs. 3a and 3b. However, the range of these variables was the same on both days: between 0 and $50\,\mathrm{km/h}$, and 0 and $170\,\mathrm{m}$.

Dataset: From the frames captured via the monitor interfaces we created a time-indexed, $1\,\mathrm{Hz}$-resolution, table containing performance statistics for each Wi-Fi network, including:

- The mean data rate the AP sent data frames at.

802.11 standard	Channel #	Center freq. (GHz)	Bandwidth (MHz)
n	6	2.437	20
ac (wave 1)	40	5.2	40
ad	1	60.48	2160

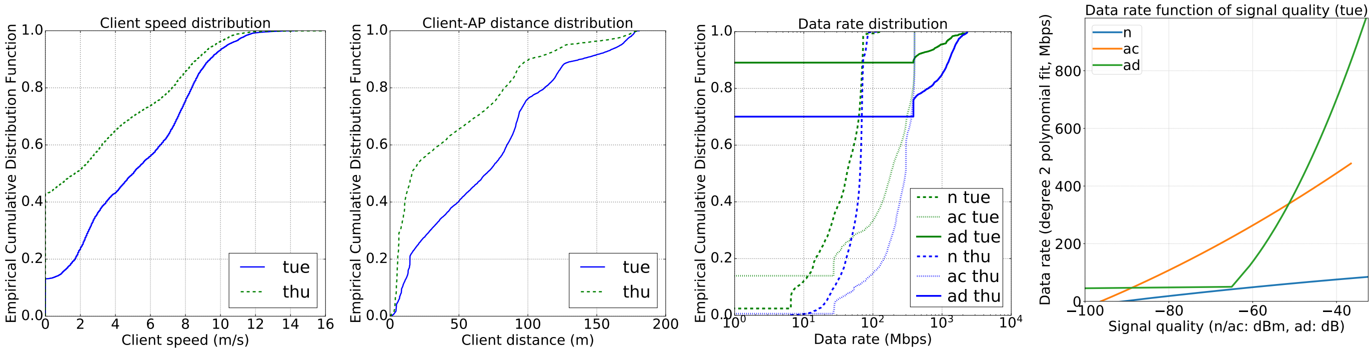
TABLE I: Wi-Fi channels used during the experiments.



(a) Experimental environment, Access Points (APs) location, and mobile client's driving circuit.

(b) Wi-Fi hardware installed on the mobile client's roof.

Fig. 2: Experimental data collection setup. Location's GPS coordinates: 41.111935, -8.631083.



(a) Client vehicle speed ECDF.

(b) Client-AP distance ECDF.

(c) Data rate ECDF.

(d) Data rate (quadratic fit) function of signal quality.

Fig. 3: Experimental datasets summary.

- The mean signal quality observed by the client. For 802.11n and ac, this is the Received Signal Strength Indicator (RSSI) from both data frames and beacons. For ad, it is the Signal-to-Noise Ratio (SNR) extracted from Sector Level Sweep (SLS) feedback frames [6].
- The client vehicle's GPS coordinates and speed, as captured by a Google Nexus 5 smartphone.

This dataset is available online [7].

Fig. 3c summarizes the observed data rates. On Tuesday, 802.11ad was connected only $\sim$10 % of the time, versus 85 % for 802.11ac, and 100 % for 802.11n. However, when connected, ad achieved data rates up to 5 times higher than ac, and up to 17 times higher than n. The AP selection algorithms need to manage this tradeoff between range and throughput. On Thursday, we had 802.11ad connectivity around 30 % of the time, due to the client being close to the AP for longer.

For use in rate forecasting, we converted the client's mobility information into a tuple $(road, dir, toAp, stopped)$, where:

$road$**:** A unique identifier of the road the vehicle is currently on, assigned as the closest to its GPS coordinates. There are only 2 roads in our experiment, one running roughly north-south and another roughly east-west.

dir**:** The vehicle's direction of movement, as one of the 4 cardinal directions. This is determined by looking at the change in latitude and longitude over the last 3 s.

$toAp$**:** A real number whose magnitude represents the distance, in meters (10 m resolution), between client and AP along the road the former is on. The number's sign is positive if the vehicle is north or east of the AP, and negative otherwise (i.e., when it is south or west).

$stopped$**:** A binary variable that takes the value 1 if the vehicle's speed is under 1 m/s (i.e., stopped), and 1 otherwise (i.e., moving).

This conversion serves three purposes: (i) minimize the effect of GPS errors (by bounding vehicles to roads); (ii) let us understand the angle between transmission and reception antennas (by combining position and direction of movement); and (iii) avoid data sparsity by limiting granularity.

III. Data Rate Forecasting

As a first step, we look at the problem of forecasting data rates for the different Wi-Fi networks.

A. Problem formulation

We treat data rate as a discrete time series with a 1 Hz resolution. Given a set of Wi-Fi networks and a time t, our goal is to predict, for each network, the data rate at future time $t+i$, for all integer i in interval $[1, win_f]$. win_f is the size of the forecasting window. We assume vehicles can collect the following information, to be used in forecasting:

- Signal quality: SNR for 80211.ad, RSSI for other types;
- Data rate, per network;
- Mobility: $(road, dir, toAp, stopped)$ as described in §II.

B. Algorithms

General description: Our goal is to predict the data rate for network net, i seconds into the future, i.e., at time t we predict the data rate at time $t + i$. All of our forecasting algorithms share a common structure. We predict data rate as the arithmetic mean of historical data rates observed over a window of win_h seconds ending at time t ($win_h \gg i$). The past observations used for prediction are filtered using a set of conditions K, e.g., location, RSSI, etc., and the proposed algorithms differ only in the set K they use. More formally:

$$Drate'_{net,K}[t+i] = \frac{1}{win_h - i + 1} \sum_{j=i}^{win_h} Drate_{net,K}[t+i-j]$$

where $Drate'$ represents the predicted data rate, and $Drate$ a previously observed rate for network net under conditions K. We have also experimented with Exponentially Weighted Moving Average (EWMA)-based aggregation, but found arithmetic mean to yield better forecasting results. Note that data rate samples are not shared among APs. This circumscribes AP-specific relationships to their proper contexts, e.g., the different signal quality vs. data rate relationships observed for 802.11n, ac and ad, depicted in Fig. 3d.

Algorithm variants: Our algorithms identify applicable past data rate samples by matching the vehicle's current state against a set of conditions K. We consider the following variations of matched conditions K, chosen by their Kendall-τ correlation with data rate in our dataset:

SQ: Signal quality level.

MRD mobility + SQ: The vehicle's road (R) and direction of movement (D) combined with signal quality.

MRDP mobility: The vehicle's road (R), direction of movement (D), and position (P) relative to the AP.

MRDP mobility + SQ: MRDP mobility and signal quality combined.

MRDPS mobility: MRDP mobility and whether the vehicle's stopped or moving (S).

For example, if $K = \{\text{road}\}$, all samples from the road the vehicle is currently on are used. We store past data rate samples in a hash table indexed by key K for efficient lookup. If no samples match current conditions K, the set of matched conditions is gradually reduced, as a fallback. E.g., in MRDPS, if we can't match the *stopped* variable, we match only the road, direction of movement, and position relative to the AP.

Besides the algorithms described above, our evaluation also considers a simple CDR (Constant Data Rate) one that predicts future data rates to be the same as the current one.

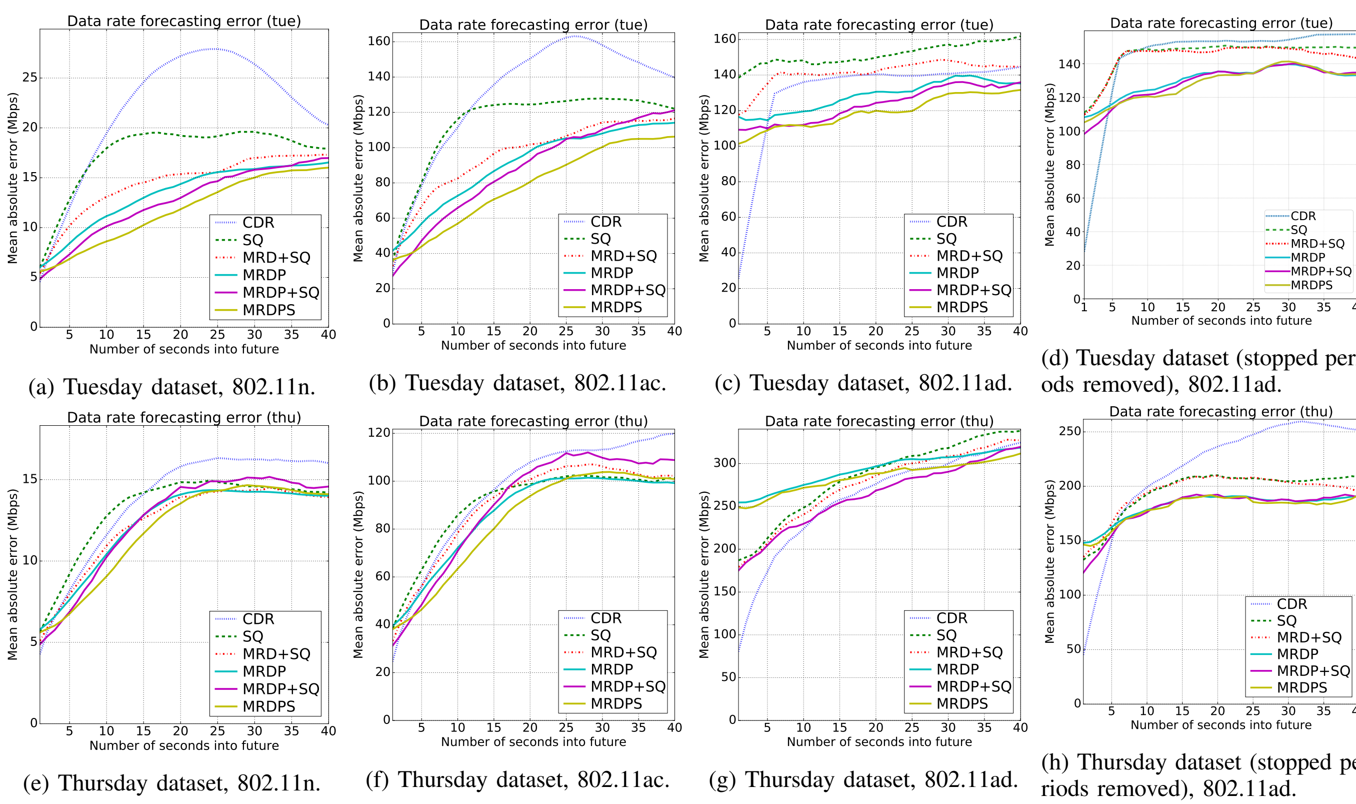
C. Evaluation

1) Setup: Starting with the dataset from §II, we sorted it in chronological order and divided it into two halves. The first was used to initialize the historical data rate tables for all (net, K) pairs, while the second was used for evaluation. Traversing the evaluation set in chronological order, for each combination of time t, network net, and algorithm $algo$, we:

1) **Prediction:** Used $algo$ to predict data rate for all times in the interval $[t+1, t+win_f]$. The forecasting window length parameter win_f was set to $40\,\text{s}$.
2) **Error computation:** For each prediction, calculated the error ae as the absolute delta between the predicted and real rates: $ae[t+i] = |Drate'_{net}[t+i] - Drate_{net}[t+i]|$.
3) **Online learning:** Added the real rate observed at time t, $Drate_{net}[t]$, to the algorithm's historical data table.

2) Results and discussion: Fig. 4 summarizes our results. The Mean Absolute Error (MAE) is shown as a function of how far in advance the prediction was made (variable i), for each experiment date and network type combination. Two general trends can be observed across the different networks. First, unsurprisingly, distant-future predictions tended to exhibit higher error. Second, the difference between algorithms was larger on the Tuesday dataset. We believe this is a result of the differences in mobility patterns shown in Fig. 3.

Let us first focus on the Tuesday results for 802.11n and ac. For $i = 1\,\text{s}$ all algorithms performed similarly, but as the prediction horizon increases, so do the differences in the plots. CDR performed the worst, with the delta to the best-performing algorithm peaking at 14.5 Mbps for 802.11n ($10\,\%$ of its maximum data rate) and 70 Mbps for ac ($17.5\,\%$ of its



(a) Tuesday dataset, 802.11n.
(b) Tuesday dataset, 802.11ac.
(c) Tuesday dataset, 802.11ad.
(d) Tuesday dataset (stopped periods removed), 802.11ad.
(e) Thursday dataset, 802.11n.
(f) Thursday dataset, 802.11ac.
(g) Thursday dataset, 802.11ad.
(h) Thursday dataset (stopped periods removed), 802.11ad.

Fig. 4: Data rate forecast evaluation results.

maximum), when $i = 25\,\mathrm{s}$.

Still on Tuesday, the SQ method, the closest scheme to those proposed in prior work, performed the second worst, in-between CDR and the mobility-based algorithms, which performed the best. From the latter, MRD+SQ performed the worst, and MRDPS the best.

On Tuesday, MRDPS's MAE peaked at 16 Mbps for 802.11n (11.4 % of its maximum data rate) and 105 Mbps for ac (26.25 % of its maximum), when $i = 40\,\mathrm{s}$ (the largest tested). From this we conclude that 802.11ac's data rate is harder to predict than 802.11n's.

801.11n and ac results for the Thursday dataset differed. MRDPS still performed the best and the MAE's magnitude was similar. But the delta between mobility and non-mobility-based methods was smaller, and MAE leveled off for $i > 20\,\mathrm{s}$.

Results for 802.11ad were peculiar. First, CDR performed competitively, even outperforming every other algorithm for near-future predictions. We believe this to be caused by ad's bimodal behavior – data rate tends to be either zero or very high. In this situation, averages can lead to substantial errors. Second, CDR excluded, the error was not as well correlated with how far in advance the prediction was made. On the Tuesday dataset, relative ad performance was similar to that observed for other networks, with the SQ method performing the worst and MRDPS the best. The error delta between them was about 35 Mbps. MRDPS achieved MAEs under 135 Mbps, or 6 % of the maximum observed rate of 2310 Mbps, for all i. In relative terms, this is less than the MAE observed for other networks.

Thursday's ad results were interesting. First, MAEs were generally larger, which is the result of having a higher number of non-zero rate samples, as per Fig. 3c. The largest MAE for the best-performing algorithm was 310 Mbps, or 13.4 % of the maximum rate – still lower than that observed for 802.11ac.

Second, and unlike other results, for predictions up to 15 s into the future, algorithms that only rely on mobility performed worse than those using signal quality. We theorize this is due to us having spent more time stopped near the APs on Thursday (Figs. 3a and 3b), giving the client time to establish a connection and beamform. This broke the correlation between location and data rate: both very low (zero) and very high data rates could be observed for a given location, depending on whether a connection had been established or not. 802.11n and ac were less affected due to their longer range. To test our hypothesis, we removed periods of immobility $\geq 20\,\mathrm{s}$ from the dataset and reran the evaluation. The results in Fig. 4h confirm our theory: both MRDP and MRDPS worked very well, being the best performers for values of $i > 5\,\mathrm{s}$.

IV. ACCESS POINT SELECTION

A. Problem formulation

The goal of an AP selection algorithm is to select an AP, or, for simplicity, network, to use at each time in an interval $[t_0, t_n]$, so that the total amount of offloadable data is maximized. Using data rate as a proxy, the goal becomes finding a schedule (i.e., a sequence of APs) yielding $max \sum_{i=0}^{n} Drate[t_i]$, where $Drate[t_i]$ is the data rate at time t_i. We assume the client can use at most one network at a time, and that a network switch implies a communication outage – i.e., a cost – of "network switch time", nst, seconds.

B. Algorithms

General description: At a given time t_i, our forecasting-based AP selection algorithms estimate the amount of data the vehicle would be able to offload over a window of size win_f seconds into the future, were it to remain on the current network, or switch to a different one. This indicates whether a switch is worthwhile. More concretely, we use the following algorithm to decide whether a switch should be initiated at time t_i:

1) Estimate the amount of offloadable data assuming the vehicle remains on the current network $cnet$. This is calculated as $\sum_{k=1}^{win_f} Drate'_{cnet}[t_{i+k}]$, where $Drate'_{cnet}[t_{i+k}]$ is the predicted data rate for network $cnet$ at time t_{i+k}.
2) Estimate the amount of offloadable data from every other possible network $onet$ as $\sum_{k=nst+1}^{win_f} Drate'_{onet}[t_{i+k}]$. The outage caused by a network switch is modeled by discarding the interval $[t_{i+1}, t_{i+nst}]$.
3) If the network that maximizes the amount of offloadable data is not the current one, initiate a switch.

Algorithm variants: Taking into account the results of §III-C2, we implemented three variants of forecasting-based AP selection, using the MRDP, MRD+SQ and MRDP+SQ rate forecasting algorithms (see §III-C1) as subroutines. We compared the performance of these methods against: (a) an optimal algorithm, and (b) a pair of naïve algorithms.

a) Optimal algorithm:: The optimal algorithm finds the schedule that maximizes the total amount of offloadable data by leveraging knowledge of the future data rates of each network for the entire trip (times in $[t_0, t_n]$), i.e., it has access to an oracle. It uses a dynamic programming algorithm based on the computation of $Data_{net}[t_i]$, which represents the maximum amount of data that can be offloaded in the interval $[t_i, t_n]$ if we're connected to network net at time t_i, assuming optimal choices are made from that point onwards. The algorithm starts by computing $Data_{net}[t_n]$ and works backwards until it reaches t_0. More formally:

$$Data_{net}[t_i] = Drate_{net}[t_i] + \max_{\forall onet} fd(net, onet, t_i).$$

$Data_{net}[t_i]$ is equal to net's data rate at time t_i, plus the maximum amount of data that can be offloaded in the future. $fd(net, onet, t_i)$ represents the maximum amount of data that can be offloaded in the future if we go from network net to network $onet$ at time t_i. It is defined as:

$$fd(net, onet, t_i) = \begin{cases} Data_{onet}[t_{i+nst+1}] & \text{if } net \neq onet \\ Data_{net}[t_{i+1}] & \text{if } net = onet \end{cases}.$$

fd models the cost associated with network switching by skipping nst seconds ahead. The computation of $Data_{net}[t_i]$ always picks the network that maximizes fd as the one to be used next, call it $Next_{net}[t_i] = \arg\max_{onet} fd(net, onet, t_i)$.

Once the network that maximizes the amount of offloadable data from t_0 is found – i.e., $inet = \arg\max_{net} Data_{net}[t_0]$ – we can reconstruct the optimal schedule by following the trail of networks stemming from $Next_{inet}[t_0]$.

b) Naïve algorithms: We consider two trivial algorithms to establish a performance lower bound:

Greedy Switch-on-Zero (SoZ): If data rate drops to zero (or at the very start), connect to the network with the highest signal quality. Otherwise stay on the current network.

Greedy Switch-on-Signal Quality (SoSQ): This algorithm is based on the observation that 802.11n, ac, and ad yield very distinct data rates. Therefore, we want to use the highest-performing network standard for which the signal is good enough. From the relationships between data rate and signal quality observed in our dataset (Fig. 3d), we arrived at the following simple rule: pick 802.11ad if its SNR $\geq -50\,\mathrm{dB}$; else pick 802.11ac if its RSSI $\geq -81\,\mathrm{dBm}$; otherwise pick 802.11n as a fallback.

C. Evaluation

1) Setup: The overall setup was akin to that used in §III-C1. The dataset was sorted chronologically and split into two halves: the first was used to initialize the historical data tables, the second for algorithm evaluation. Let the evaluation period contain timestamps $[t_0, t_n]$. For each algorithm variant, starting with $i = 0$, and while $i \leq n$, we:

1) Let the chosen algorithm determine whether a network switch should be initiated at time t_i. If so, said switch was added to the schedule.
2) If the chosen algorithm uses historical data, the data rate at time t_i for the network the client is currently connected to was added to that history (data rates of other networks were unobservable and hence not added), for future use in data rate forecasting, i.e., we do online learning.
3) If a switch was initiated, we reran the analysis with $i = i + nst + 1$, to account for the switching cost; otherwise we reran it with $i = i + 1$.

Finally, we compared each variant's schedule with the optimal one, according to multiple metrics, including the total amount of offloadable data, and time allocated to each network.

The entire process was repeated for multiple values of the network switching time and forecast window parameters.

2) Results and discussion: Fig. 5 shows the total amount of data offloaded when using each algorithm, relative to the optimal. Fig. 6 displays the percentage of time spent on each network, as well as how much data was transferred over each of them, for the particular case of $nts = 5$ s and $win_f = 40$ s.

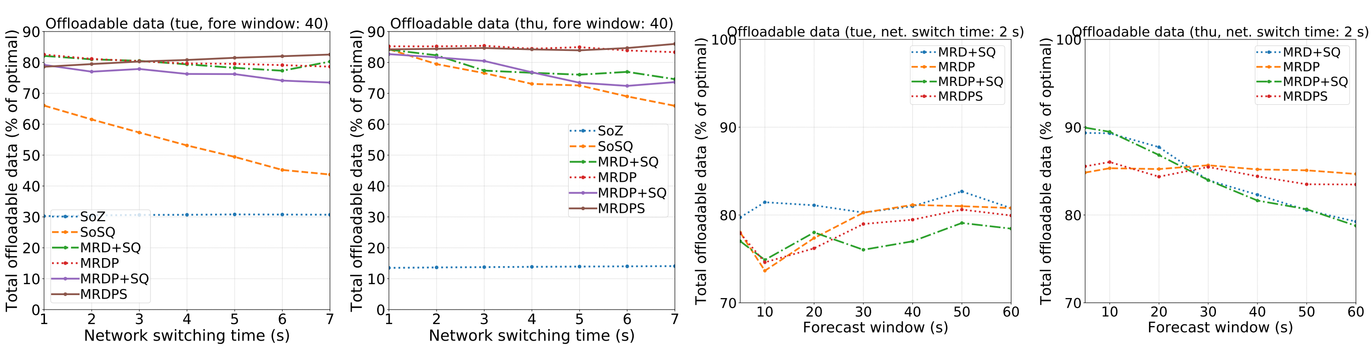
Fig. 5a shows the data transferred for the Tuesday dataset. The forecasting window win_f was fixed at $40\,\mathrm{s}$, while the network switching time nts was varied between 1 and 7 s.

The greedy SoZ algorithm performed the worst, transferring $\sim 30\,\%$ of the optimal regardless of the network switching time. Fig. 6a shows it mostly relied on 802.11n, whose data rate rarely dropped to zero due to its long range. This explains SoZ's insensitivity to the network switching time parameter.

The greedy SoSQ algorithm performed quite well for $nts = 1$ s, transferring over $65\,\%$ of the optimal amount. However, since its switching decisions do not account for the associated outage periods, its performance worsened for larger nts values. In Fig. 6a ($nts = 5$ s), SoSQ is seen spending over $30\,\%$ of its time switching between networks.

All forecasting-based algorithms performed significantly better than the greedy ones, offloading around $80\,\%$ of the optimal amount, and thus justifying the use of mobility information. MRDP+SQ and MRDP performed the best for small nts, and MRDPS for large nts. The tendency is for performance to decrease slightly with increased nts, due to the larger penalty incurred for each bad switching decision. However, because they take outages into account, all of the forecasting-based schemes were able to mitigate this well.

Fig. 6a highlights the differences between the forecasting and optimal algorithms. The optimal schedule uses all three networks, with a strong preference for 802.11ac. The forecasting algorithms rely even more on ac, in detriment of ad, and do not use n at all. This is explained by n's low performance and ad's unpredictability. Being able to accurately predict 802.11ad performance is the optimal algorithm's big advantage.



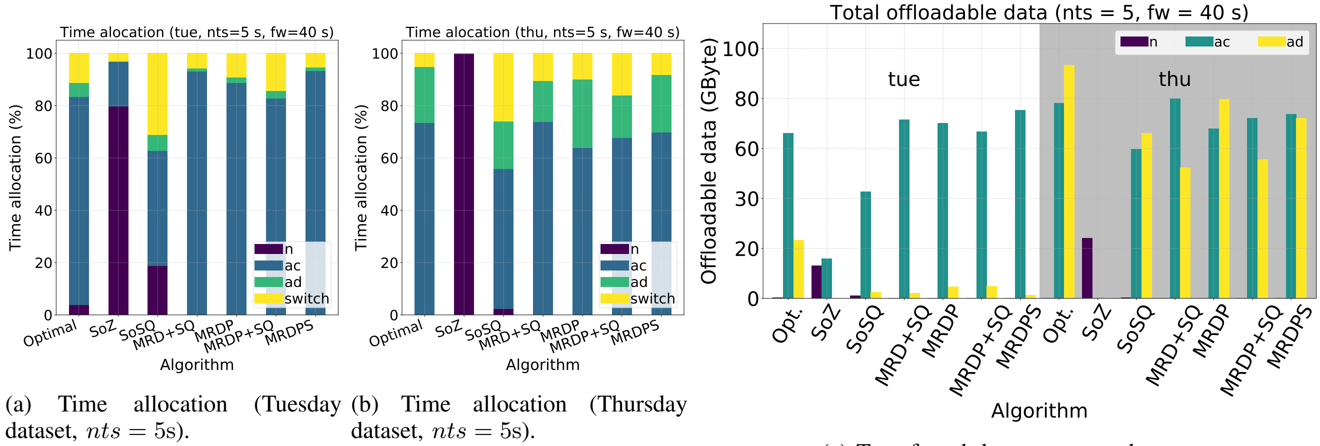
(a) Data transferred as function of nts (Tuesday dataset).

(b) Data transferred as function of nts (Thursday dataset).

(c) Data transferred as function of forecasting window (Tuesday).

(d) Data transferred as function of forecasting window (Thursday).

Fig. 5: AP selection evaluation results: amount of transferred data relative to optimal.



(a) Time allocation (Tuesday dataset, $nts = 5$s).

(b) Time allocation (Thursday dataset, $nts = 5$s).

(c) Transferred data per network type.

Fig. 6: AP selection evaluation results: schedule details.

Fig. 5c shows the effect of the forecasting window win_f parameter on scheduling performance. Overall, the sensitivity to win_f is low, with the amount of offloadable data hovering around $80\,\%$ of the optimal. By varying win_f we were also able to discover an inverse correlation between it and both the number of network switches and 802.11ad's usage.

Figs. 5b and 6b show the amount of transferred data and time allocation for the Thursday dataset. Compared with Tuesday's results, 802.11ad was used more often across the board, a reflection of the fact that more time was spent near the APs, resulting in more frequent ad connectivity.

Greedy SoZ performed even worse than on Tuesday, achieving less than $15\,\%$ of the optimal, and only using the 802.11n network. Surprisingly, SoSQ performed very well, matching the forecasting-based algorithms for $nts = 1$ s, at around $85\,\%$ of optimal. However, like on Tuesday, its performance fell off for larger nts. As shown in Fig. 6b, when $nts = 5$ s, SoSQ spent significantly more time switching networks than optimal.

All forecasting algorithms performed well, but MRDP and MRDPS, which rely solely on mobility information, distinguished themselves, always offloading more than $80\,\%$ of the optimal amount. Fig. 6b shows that both MRDP and MRDPS used 802.11ad more frequently than the other forecasting-based algorithms, and switched networks less often.

Fig. 6c shows that even though the 802.11ad network was only used ∼20% of the time on Thursday, it accounted for more than half of the total amount of offloaded data for the better-performing algorithms. This result highlights the added value of leveraging Wi-Fi diversity, which is likely to increase even further with the upcoming 802.11ay standard.

MRD+SQ and MRDP+SQ fared worse on the Thursday dataset, both starting above $80\,\%$ for $nts = 1$ s, but then falling gradually, and eventually hitting $\sim 75\,\%$ for $nts = 7$ s. Even though both algorithms account for switching costs, we found the use of signal quality to be detrimental in two ways: (i) for smaller nts, signal fluctuations trigger too many network switches; and (ii) the periods of 802.11ad connectivity become shorter as the switching cost increases.

Taking the results from both datasets into account, MRDP and MRDPS performed the best. Both were consistently good across different datasets and switching costs. If collecting historical data is deemed too onerous, the greedy SoSQ algorithm may be a reasonable alternative. However, we note its sensitivity to switching costs and the fact that it really only applies when choosing between networks with very distinct performance characteristics. MRDP and MRDPS, on the other hand, are general, and hence free from that assumption.

V. Related work

The opportunistic use of Wi-Fi networks by moving vehicles has been studied extensively in the past. Authors have mostly focused on three subproblems: (i) minimizing time spent on Wi-Fi handoffs [8]–[12], (ii) performance forecasting [13]–[16], and (iii) optimizing network selection [4], [17]–[19]. Since the first topic is largely orthogonal to our work, we focus on the last two.

Also, to our knowledge, few have used 802.11ad for vehicular data offloading. 802.11ad vehicular research has, so far, focused on feasibility and beam training optimization [20].

Network performance forecasting: Early work on network performance forecasting aimed at TCP throughput prediction, using three techniques: TCP-modeling based [13], history based [13]–[16], and machine-learning based [14]–[16].

He *et al.* [13] showed that history-based schemes are often superior to model-based ones. Mirza *et al.* [14] showed that a machine-learning technique based on Support Vector Regression (SVR) is superior to history-based methods in a wired environment. But, when applied to a wireless vehicular network [15], SVR-based prediction was outperformed by history-based techniques. However, the paper did not consider wireless network properties and used fairly homogeneous networks.

Network selection: Previous work has mostly used older Wi-Fi versions, up to 802.11n, which resulted in fairly homogeneous deployments. One exception is recent work on using 802.11ad for vehicular data offloading [20], but this focused on feasibility and beam training optimization. In addition, prior work that uses historical data for Wi-Fi network selection in vehicular environments has largely relied on signal quality alone as a performance indicator [5], [17]–[19]. This assumption breaks in diverse Wi-Fi environments, where more recent Wi-Fi versions may perform better, despite having a weaker signal, as exemplified by Fig. 3d. Work by Giannoulis *et al.* [4] improved on traditional RSSI-based algorithms by using long-term differences in IP performance, in addition to RSSI. However, it used fairly homogeneous networks and did not consider recent throughput samples.

The use of spatial location as a channel-quality estimator in vehicular networks was previously explored by Meireles *et al.* in the context of multi-hop routing [21].

VI. Conclusions

Our results show that AP selection based on historical network performance, indexed by mobility features, is a promising approach for data offloading in our Wi-Fi-diverse V2I scenario. Mobility, in the form of location and direction of movement, is able to capture the average speed at which a vehicle is moving towards or away from an AP, the angle between the client's and AP's antennas, and the fading environment around both nodes. This gives it good predictive power, letting our algorithms exploit the tradeoff between the extended range provided by legacy 802.11n, and the higher data rates provided by 802.11ac and 802.11ad. We found that AP schedules that make good use of 802.11ad are capable of offloading up to 80 % more data than schedules that only use 802.11n networks. As 802.11ax and ay are developed, the benefits of leveraging diversity are likely to increase.

The results in this paper are just a first step towards building a system that can opportunistically use a heterogeneous Wi-Fi infrastructure for vehicular access. Further studies are needed that use more extensive datasets and cover more diverse traffic scenarios. For example, our work can be extended to capture the effect of multiple users on throughput by adding features such as channel busy time, or AP load – extracted from the Wi-Fi radio or AP beacons. Research is also needed on how deal with factors such as unpredictable network switching times, competing traffic, etc.

Acknowledgment

The authors would like to thank Bruno Maurício, from the University of Porto, for help with the experiments of §II.

References

[1] J. G. P. Rodrigues, A. Aguiar, and J. Barros, "Sensemycity: Crowdsourcing an urban sensor," 2014.

[2] FON Wi-Fi network. [Online]. https://fon.com, visited on 11/25/2020.

[3] A. K. Lyyra and K. M. Koskinen, "The ambivalent characteristics of connected, digitised products: Case tesla model s," in *Nordic Contributions in IS Research*, U. Lundh Snis, Ed. Cham: Springer International Publishing, 2016.

[4] A. Giannoulis, M. Fiore, and E. W. Knightly, "Supporting Vehicular Mobility in Urban Multi-hop Wireless Networks," in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '08. New York, NY, USA: ACM, 2008.

[5] P. Deshpande, A. Kashyap, C. Sung, and S. R. Das, "Predictive Methods for Improved Vehicular WiFi Access," in *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '09. New York, NY, USA: ACM, 2009.

[6] "IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Am," *IEEE Std 802.11ad-2012*, dec 2012.

[7] (2020) Vehicular Wi-Fi experiment data set. [Online]. Available: https://cs.vassar.edu/~rpachecomeireles/research/wifi/

[8] M. Shin, A. Mishra, and W. A. Arbaugh, "Improving the Latency of 802.11 Hand-offs Using Neighbor Graphs," in *Proceedings of the 2Nd International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '04. New York, NY, USA: ACM, 2004.

[9] S. Shin, A. G. Forte, A. S. Rawat, and H. Schulzrinne, "Reducing MAC Layer Handoff Latency in IEEE 802.11 Wireless LANs," in *Proceedings of the Second International Workshop on Mobility Management &Amp; Wireless Access Protocols*, ser. MobiWac '04. New York, NY, USA: ACM, 2004.

[10] V. Brik, A. Mishra, and S. Banerjee, "Eliminating Handoff Latencies in 802.11 WLANs Using Multiple Radios: Applications, Experience, and Evaluation," in *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '05. Berkeley, CA, USA: USENIX Association, 2005.

[11] J. Eriksson, H. Balakrishnan, and S. Madden, "Cabernet: Vehicular Content Delivery Using WiFi," in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, ser. MobiCom '08. New York, NY, USA: ACM, 2008.

[12] Y. S. Chen, M. C. Chuang, and C. K. Chen, "DeuceScan: Deuce-Based Fast Handoff Scheme in IEEE 802.11 Wireless Networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 2, mar 2008.

[13] Q. He, C. Dovrolis, and M. Ammar, "On the Predictability of Large Transfer TCP Throughput," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, aug 2005.

[14] M. Mirza, J. Sommers, P. Barford, and X. Zhu, "A Machine Learning Approach to TCP Throughput Prediction," in *Proceedings of the 2007 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '07. New York, NY, USA: Association for Computing Machinery, 2007.

[15] M. Mirza, K. Springborn, S. Banerjee, P. Barford, M. Blodgett, and X. Zhu, "On The Accuracy of TCP Throughput Prediction for Opportunistic Wireless Networks," in *2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, jun 2009.

[16] Y. Liu and J. Y. B. Lee, "An Empirical Study of Throughput Prediction in Mobile Data Networks," in *2015 IEEE Global Communications Conference (GLOBECOM)*, dec 2015.

[17] Z. Song, L. Shangguan, and K. Jamieson, "Wi-Fi Goes to Town: Rapid Picocell Switching for Wireless Transit Networks," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, ser. SIGCOMM '17. New York, NY, USA: ACM, 2017.

[18] H. Lee, J. Flinn, and B. Tonshal, "RAVEN: Improving Interactive Latency for the Connected Car," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '18. New York, NY, USA: ACM, 2018.

[19] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 Packet Delivery from Wireless Channel Measurements," in *Proceedings of the ACM SIGCOMM 2010 Conference*, ser. SIGCOMM '10. New York, NY, USA: ACM, 2010.

[20] A. Loch, A. Asadi, G. H. Sim, J. Widmer, and M. Hollick, "mm-Wave on wheels: Practical 60 GHz vehicular communication without beam training," in *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, jan 2017.

[21] R. Meireles, P. Steenkiste, J. Barros, and D. C. Moura, "Lasp: Look-ahead spatial protocol for vehicular multi-hop communication," in *2016 IEEE Vehicular Networking Conference (VNC)*, 2016.