WideRate: Reinforcement Learning Rate Adaptation for Mobile Wide Area Networks

Karyn Doke, Elham Sadeghi, Vaasu Taneja, Habib Affinnih, Petko Bogdanov, Mariya Zheleva
Department of Computer Science, University at Albany
{kdoke, esadeghi, vtaneja, haffinnih, pbogdanov, mzheleva}@albany.edu

2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring) | 979-8-3503-8741-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/VTC2024-SPRING62846.2024.10683543

***Abstract*—Mobile wireless networks revolutionize our lives and livelihoods. Yet, rural areas, characterized with sparse populations and rugged terrain, consistently lag behind in mobile connectivity compared to their urban counterparts. As a result, community-owned networks realized through fixed wireless technologies, have become an increasingly viable Internet option for otherwise disconnected areas. Fixed wireless, however, is inherently designed for residential/stationary access and is not readily applicable for the use of mobile agents that might travel through a rural community. In this paper we explore the extension of fixed wireless networks for mobile access. A key factor for continuous mobile access is efficient rate adaptation. To that end, we develop `WideRate`— a reinforcement learning framework that employs signal strength measurements for optimal rate adaptation. We showcase `WideRate` in the context of wide-area Television White Space networks, whereby we design a vehicular mobile unit and carry out an extensive measurement campaign in a real community network. We use the collected traces to motivate the need for rate adaptation and implement a realistic network simulator that aids in our evaluation. We demonstrate that `WideRate` significantly outperforms counterparts from the literature including a reinforcement learning model.**

***Index Terms*—Rate adaptation, reinforcement learning.**

I. Introduction

While mobile Internet access has evolved substantially in both ubiquity and capabilities, rural areas still significantly lag behind in mobile network availability. Based on a 2022 ITU report, 24% of rural residents, and 87% of low-income rural residents, do not have 4G coverage [1]. As a result, some underserved communities have taken charge of their technological progress by deploying and managing their own networks [2] [3] [4]. Fixed wireless technologies, such as Ubiquity Mesh [5] [6] and Television White Spaces [7] [8] are particularly applicable due to their lower infrastructure requirements and relative ease of deployment. While fixed wireless caters to stationary/residential Internet access, it leaves behind essential civic services, such as emergency response, that rely on mobile broadband for efficient operation.

In this paper we explore the applicability of fixed wireless to deliver mobile Internet access using community networks. Commercial omnidirectional technologies such as Television White Spaces (TVWS) are particularly applicable to our problem setting. A key challenge to extend fixed wireless networks for mobile access is their lack of efficient rate adaptation (RA). Because fixed wireless channels maintain a relatively stationary signal, they use manually assigned fixed rates. However, client mobility and rugged terrain compound to high dynamicity of the wireless channel, which in turn calls for adaptive rate selection. This is motivated in Fig. 1, which presents results from a driving campaign in a community TVWS network. Outlined in different colors are the best allowable rates that maximize the achieved throughput with low bit error rate. The optimal rate changes substantially across adjacent locations, which calls for an efficient RA approach.

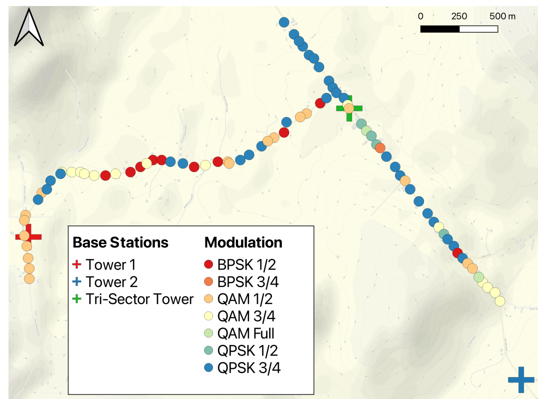

Fig. 1: Optimal downlink modulations at different locations on the two main roads in the deployment TVWS community.

RA in vehicular networks has been heavily researched [9]–[13]. These methods use historical observations of packet loss or signal strength to select an operational rate. However, in vehicular environments the channel history may quickly become irrelevant due to rapidly changing channel conditions, making rate prediction inaccurate. Reinforcement learning (RL) strategies that incorporate rewards based on decision outcomes can equip RA to better adapt to dynamic channel conditions. RL learns an optimal behavior through interactions with the environment and observations of actions taken, making it particularly applicable to rate adaptation, where supervision is not practical. RL rate adaptation has thus far been considered in several works [14]–[18]. However, the focus of these works has been on low-range technologies such as 802.11af with fairly predictable channel dynamics, which makes them inapplicable to wide-area community networks.

To address these problems, we propose `WideRate`, a reinforcement learning rate adaptation framework for wide-area mobile wireless networks. `WideRate` learns from signal strength measurements to dynamically tune the link rate and maximize the throughput in uplink and downlink.

979-8-3503-8741-4/24/$31.00 ©2024 IEEE

WideRate outperforms model- [19] and learning-based [18] counterparts from the literature. WideRate quickly adapts to rapidly changing channel conditions and maintains performance across increasing vehicle speeds.

To evaluate and test WideRate, we design a data mule unit (DMU) used for data collection in a real-world wide-area TVWS deployment. The DMU is carried by vehicles traveling in community TVWS networks. The DMU architecture consists of a TVWS Customer Premise Equipment (CPE), connected via Ethernet to a WiFI access point (AP). Using the CPE, our DMU connects to the wireless backhaul for Internet access. The DMU can be used to transmit and receive data while driving in areas with TVWS coverage.

We partner with the Town of Thurman in upstate NY to explore the off-the-shelf TVWS network technology with CPE mobility. Thurman is a typical rural community, challenged by the lack of commercial mobile and broadband access. The town took charge of their own technological progress and worked as a community to set up a 5-sector TVWS network [20]. During a four month driving campaign, we collect geo-tagged field traces that we incorporate into WideRate's motivation and evaluation.

The contributions of this work are as follows:

- We develop WideRate for rate adaptation in wide-area vehicular networks. The framework can be used to predict up- and down-link rates and is direction agnostic.
- We design a DMU that can be carried by a vehicle while traveling through a TVWS community network. The DMU can be used to transmit data in both uplink and downlink directions using the CPE supported rates.
- We collect in-situ traces from a real-world community TVWS network and develop a realistic simulator do demonstrate WideRate's performance.

II. Related Work

RA for IEEE 802.11-based wireless networks has been widely researched using many different design methodologies. [19], [21]–[24] design RA algorithms based on packet loss. Others employ *physical layer* metrics such as *Signal-to-Noise Ratio (SNR)* to adjust transmission rates based on the perceived channel quality [22], [25], [26]. [27] develops a channel-aware RA algorithm that predicts current channel state based on past observations. All these mechanisms were designed for stationary indoor wireless networks with considerably lower channel variation than that in wide area vehicular settings. In a mobile environment where channel condition are changing constantly, dynamic models are required because they can adapt better to fluctuating channel conditions. While these stationary mechanisms work well in traditional 802.11 indoor networks, they fall short in vehicular networks, which are characterized with high channel dynamicity.

RA in vehicular networks has a long line of research. [9] designs a custom layer RA framework that implements a combination of loss-triggered and SNR-triggered protocols for urban and vehicular environments. The paper found that the loss-triggered mechanisms are unable to adapt to the changing channel conditions in mobile environments. The SNR-triggered protocols are susceptible to over-selection from the ideal rate when the coherence time is low (fast fading). [10] proposes a protocol for vehicular networks that estimates the link quality according to the context information (i.e., vehicle speed and distance from neighbor) and past history. [11] implements an SNR-based RA algorithm which only considers short history to make selection rate decisions. [12], [13] use historical packet loss rate to estimate the channel quality. In vehicular access networks, the constantly changing wireless channel conditions make the channel history quickly irrelevant, and thus feedback from previous transmissions may not accurately predict the channel. Our approach falls within the stream of dynamic models which can be a more effective tool for adapting to changing environments.

In recent years, machine learning methods have been employed in both fixed and mobile wireless networks. A few recent works consider reinforcement learning for link adaptation [14], [15], RA [16] in 802.11ac, and RA in vehicular networks [17], [18]. [17] investigates a rate adaptation scheme for vehicular TVWS access (802.11af) using deep learning classification. Deep learning classification is supervised and requires significant training data. Such models do not adapt and are hard to transfer across environments or vendors. Reinforcement learning, in turn, incorporates rewards based decision outcomes and learns the environment by exploring. [18] proposes a reinforcement learning-based design technique for RA for Internet of Vehicles (IoV) using a WiFi Access Point (802.11ac). This work provides solutions for short range technology, whereas our approach focuses on wide-area networks where the channel dynamics are unpredictable.

III. Methodology

We now present the methodology behind WideRate. We utilize real-world performance data from a 5-sector TVWS network serving a rural community. We collect geo-tagged field traces and employ them to train and validate WideRate. We begin by outlining the data collection campaign followed by a detailed description of the WideRate algorithm.

A. The Data Mule Unit (DMU).

The DMU is a mobile TVWS client we designed to understand current limitations of mobile TVWS networks and evaluate WideRate. We mount a directional antenna on top of a vehicle that connects via coaxial cable to a CPE [20], as shown in Fig. 2b. The CPE is powered using a Power over Ethernet (PoE) device and connects to a client host laptop with an Ethernet cable, as shown in Fig. 2c. With DMU mobility, the CPE will dynamically associate with the various community TVWS base stations to acquire Internet connectivity (Fig. 2a). The CPE-supported rates are BPSK 3/4, QPSK 1/2, QPSK 3/4, 16-QAM 1/2, 16-QAM 3/4, 16-QAM full.

B. Real-World Data Collection

We investigate the performance of off-the-shelf TVWS networks with mobility using the DMU. We set up a driving

(a) Base Station (b) Antenna (c) CPE

Fig. 2: A deployed TVWS base station in Thurman, NY (a), directional antenna mounted on vehicle (b), and a CPE connected to a client laptop via a PoE (c).

campaign to collect field traces including GPS coordinates, current transmission rate, SNR, and achieved throughput. We use iPerf[1] to inject UDP traffic in the TVWS network. We setup a client on the laptop connected to the DMU and a server on a workstation in our lab. As the DMU moves, the client injects a bi-directional 1 $Mbps$ flow. We limit the amount of data generated to 1 $Mbps$ so that our tests are minimally disruptive to residents' Internet service. We use tshark[2] to capture packet traces (PCAP) on both the client and server, and log GPS coordinates every second. For each modulation rate, we performed four driving campaigns for a duration of 45-60 minutes each during 10/2020 – 1/2021.

We organize our data as tuples: $\mathcal{D} = \{(l_i, t_i, s_i, r_i, tr_i)\}$, where i is the sample index, l_i is the sample location (latitude, longitude), t_i is the timestamp, s_i is the measured SNR, r_i is the employed modulation rate and tr_i is the achieved transmission rate (TR). There is an important distinction between TR and throughput. As our $1 Mbps$ client session was insufficient to saturate the TVWS link, there were quiet periods within each second in which no packets were transmitted. Thus, the achieved throughput, which is a measure of how much data is received *per second*, is a less informative measure of link capability than the TR. Intuitively, TR is the achieved rate during periods in which data is actively traveling on the link. Using the PCAP traces, we examine each second interval in small steps (10 ms). For each step, we calculate the number of transmitted bytes and divide by 10 ms to get the TR tr_b. Given the set of TRs for each 10 ms occupied interval, the achieved TR in each second is tr_i=$(1/|TR|)\sum_j tr_{bj}$.

Empirical SNR measurements s_i are collected from the individual base stations in the TVWS community network at a granularity of 120 seconds. To unify the data $\mathcal{D}$, we need additional SNR measurements at regular one second intervals which our empirical measurements did not provide. We interpolate the unknown data points between the empirical SNR measurements using a cubic spline interpolation. This method gives an interpolating polynomial that is smoother and has smaller error than some other interpolating polynomials.

C. Real-World TVWS Connectivity Simulator

Our goal is to design and evaluate approaches for rate adaptation specifically for mobile TVWS clients. A comprehensive evaluation requires observations of varying channel states (SNR) and varying achievable throughput given the network and spatio-temporal context (e.g., terrain, weather, network saturation, etc.). While our driving campaign produced traces with lots of the above characteristics (a total of $|\mathcal{D}| = 105,783$ tuples amounting to over 6GB), it does not capture all possible combinations of conditions and driving trajectories. More importantly, we would also like to quantify the ability of our proposed rate adaptation schemes to adjust to significant changes in the network. To address these experimental goals, we employ our real-world traces for a spatio-temporal network state simulator capable of generating arbitrary traces by sampling realistic SNR and achieved transmission rates. Beyond our evaluation, the simulator will be a useful resource for further research into rate adaptation.

A key assumption in our simulator is that the network state (SNR and TR) is normally distributed in a given spatio-temporal context (time of the day and location). Hence, to sample realistic network states, we estimate locally the mean and variance of observations from our driving data. Specifically, a spatial-temporal context for location l at time of the day t $D_{l,t} \in \mathcal{D}$ is a subset of samples D_j whose locations l_j and time t_j fall within a spatio-temporal radius ρ, i.e. $|l_j - l| \leq \rho_l$ and $|t_j - t| \leq \rho_t$. In our experiments we adopt a spatial radius of $\rho_l = 50m$ and a temporal radius of $\rho_t = 1h$ as these parameters ensured sufficient samples in most locations in our simulation area. For sparsely sampled locations, we progressively increase the radius until at least 20 samples fall in the spatio-temporal context. To generate realistic driving traces we transition between random start and end points and sample SNR and transmission rates for a given modulation at regular intervals.

Algorithm 1 Sample SNR and TR

Require: Location l, time of day t, rate r and samples $\mathcal{D}$
Ensure: Sampled SNR s_i or transmission rate (TR) tr_i

1: Compute context $\mathcal{D}_{lt} = \{\mathcal{D}_j |\ |l_j - l| \leq \rho_l, |t_j - t| \leq \rho_t\}$
2: **if** SNR sampling **then**
3: Estimate $\mu_{lt}^{(SNR)}$ and $\sigma_{lt}^{(SNR)}$ from $\mathcal{D}_{lt}$
4: Return $s \sim \mathcal{N}(\mu_{lt}^{(SNR)}, \sigma_{lt}^{(SNR)})$
5: **else if** TR sampling **then**
6: Compute rate-specific context $\mathcal{D}_{lt|r} = \{\mathcal{D}_{lt} \mid r_j = r\}$
7: Estimate $\mu_{lt|r}^{(TR)}$ and $\sigma_{lt|r}^{(TR)}$ from $\mathcal{D}_{lt|r}$
8: Return $tr \sim \mathcal{N}(\mu_{lt|r}^{(TR)}, \sigma_{lt|r}^{(TR)})$
9: **end if**

The procedures for SNR and TR sampling are both outlined in Alg. 1. The input is a current location l, time of day t, desired (modulation) rate r and real-world samples $\mathcal{D}$. We first compute the context D_{lt} by filtering the data based on the context radius ρ (Step 1). If sampling an SNR we fit a normal distribution $\mathcal{N}(\mu_{lt}^{(SNR)}, \sigma_{lt}^{(SNR)})$ from context SNR tuples and sample a random channel state s (Steps 3,4). When sampling an achieved transmission rate (Steps 5-9), we first subset the context $\mathcal{D}_{lt|r}$ to only tuples with the desired modulation r (Step 6) and then sample from a normal distribution of achieve rates estimated from samples (Steps 7,8). Note that if we do

[1]https://iperf.fr/
[2]https://www.wireshark.org/

not have any sample in the context for the desired modulation r, we use a default TR value from Table I.

TABLE I: CPE Supported Rates

Modulation	Code	SNR (dB)	Rate (Mbps)
BPSK	3/4	5	2
QPSK	1/2	7	5
	3/4	9	6
16-QAM	1/2	11	8
	3/4	15	12
	Full	17	14

D. `WideRate`: Reinforcement Learning for Rate Adaptation

Reinforcement learning (RL) methodologies, adapt their behaviour based on prior decision outcomes and are a natural fit for RA in dynamic channel conditions like our setting. We design an RL framework where an agent at time t_i in a location l_i observes the environment s_i (SNR) and takes an action a_i: transmitting data using a selected modulation. The agent receives a reward r_i based on the outcome (achieved transmission rate tr_i) and moves to a new state s_{i+1}. The transition from s_i to s_{i+1} with action a_i and reward r_i is stored as an event $e_i = (s_i, a_i, r_i, s_{i+1})$ and used for re-training/adapting the model.

State and Action Spaces. The state space S characterizes the channel condition, where the state S_i at a given time i is a time series of the last τ measured SNR values $S_i = (s_{i-\tau}, s_{i-\tau+1}, \dots s_i)$. The action space A are the six supported modulations to choose from (Table I).

Reward Function. The achieved TR is the reward in the model, where action results are three types: optimal rate (the highest achievable rate for a given S_i), sub-optimal rate (any valid rate that is worse than the optimal), and an invalid rate (selected by the model but not supported at the current S_i). The reward r_i for action a_i at state S_i is defined as: $r_j = \mathcal{P} \times tr_i$, where $\mathcal{P}$ is a penalty factor with values 1, -1, or 0 for optimal, sub-optimal and invalid results, and tr_i is the achieved TR.

RL Model. Our RL approach, `WideRate`, is based on the Q-learning algorithm [28] and is implemented using a Deep Q-Network (DQN) [29]. A Q-learning approach determines the optimal action based on its current state S_i, where the DQN architecture is comprised of a Q-network and a target network. We experiment with several architectures (details in §IV-A) and employ a long short-term (LSTM) neural network (NN) which performs best in experiments due to its ability to model both short and long-term dependencies in sequential data such as our time series of SNR states S_i. The Q-network takes the current state S_i and action a_i from each data sample and predicts the Q-value for that particular action (predicted Q-value). The target network takes the next state S_{i+1} from each data sample and predicts the best Q-value out of all actions that can be taken from that state (target Q-value). The model stores each data transition (S_i, a_i, r_i, S_{i+1}) and takes a random batch of these samples to (re-)train the Q-network. After a pre-configured number of time steps, the learned weights from the Q-network are copied over to the target network, which ensures stability of the remaining Q-values.

Algorithm 2 `WideRate`: RL Rate Selection and Adaptation

Require: Exploration rate ϵ, Discount factor γ, Q LSTM$_Q$ and target LSTM' neural networks
1: Measure s_i and form the state $S_i = (s_{i-\tau}, \dots, s_i)$
2: **if** $rand \sim U(0,1) > \epsilon$ **then**
3: $\quad a_j = \arg\max_a LSTM_Q(S_i, a)$
4: **else**
5: $\quad a_j$ = random action
6: **end if**
7: Measure (or sample) tr_i by transmitting with a_i
8: Reward $r_i = \mathcal{P} \times tr_i$
9: Measure (or sample) s_{i+1} and form the state S_{i+1}
10: Store transition (S_i, a_i, r_i, S_{i+1}) in training set M
11: **if** Enough transitions in M **then**
12: $\quad$ Sample a batch of N transitions from M
13: $\quad$ **for** Each transition (S_i, a_i, r_i, S_{i+1}) in batch **do**
14: $\quad\quad y_i = r_i + \gamma \max LSTM'(S_{i+1}, a_i)$
15: $\quad$ **end for**
16: $\quad$ Compute loss $L = \sum_i (y_i - LSTM_Q(S_i, a_i)))^2$
17: $\quad$ Update $LSTM_Q$ by back propagation based on L
18: $\quad$ Every C steps $LSTM'$ = $LSTM_Q$
19: **end if**

Alg. 2 lists the steps of `WideRate` including selecting the current action and performing the necessary updates to the NNs. The input is an exploration rate ϵ, learning rate $gamma$ and the current state of the Q and target NNs whose parameters are initially randomly initialized. Importantly, the exploration rate ϵ decays exponentially as the networks learn (we omit this from the notation for simplicity). The current state S_i is first formed by appending the current SNR s_i to the last $\tau - 1$ measurements (Step 1). Next we "exploit" the action predicted by the currently trained Q network with probability ϵ or take a random exploratory action (Steps 2-6). We then transmit with the selected rate a_i (Step 7), compute the reward r_i (Step 8) and form the next state S_{i+1} (Step 9). State-action-reward transitions are stored in a training set M (Step 10). When sufficient transitions are accumulated, we sample a batch of N transitions and calculate the expected return (Steps 12-15) as the expected reward r_i we get from taking action a_i in state S_i, plus the maximum expected return that can be achieved by taking action a_{i+1} in state s_{i+1} discounted by a factor $\gamma \in [0, 1]$). We next calculate the loss L and update the LSTM$_Q$ based on it (Steps 16-17). Every C steps, we copy the weights from LSTM$_Q$ to those of LSTM' to ensure that the Q-values in the latter remain stable (18).

IV. Experimental Evaluation

We now evaluate `WideRate`. We show that `WideRate` outperforms counterparts from the literature in terms of optimal rate selection. We demonstrate `WideRate`'s ability to maintain consistent and high throughput link in wide area rural mountainous terrains and with increasing vehicle speed.

A. Experimental Setup

Data. We evaluate `WideRate` using trajectories created with our simulator. We initialize the simulator with a start location l_j (set of GPS coordinates) and the desired duration (seconds) of the trajectory. For this paper, we generate 10 trajectories that represent a mix of stable to challenging channel conditions to test our model. In these experiments, we select a single trajectory that represents a challenging channel condition where the SNR fluctuates across the full range of values as outlined in Table I. All trajectories contain 2000 spatial-temporal samples.

TABLE II: NN RL Models & Hyperparameters

Model	Structure & Hyperparameters
LSTM	Input, LSTM Layer (20 hidden), Linear Layer, Argmax
ATTN	Input, Linear Layer, Multihead Layer (10 heads, 20 hidden), Linear Layer, Argmax

`WideRate` implementation and initialization. We evaluate `WideRate` with two NN architectures: a multi-head attention (ATTN) and a long short-term memory (LSTM). While ATTN learns long-term trends in the data, LSTM captures temporal dependencies in sequential data points, which are particularly important for rate adaptation based on perceived channel conditions. Additionally, we explore two learning strategies: a non-decaying epsilon-greedy approach with $\epsilon = 0.1$, and a decaying epsilon-greedy approach with $\epsilon = 1$ decaying exponentially over time. Our structure and hyperparameters are outlined in Table II. Input to our model is a sequence of the 10 most recent SNR values. We set the learning rate (l_r) to 0.01, the discount factor (γ) to 0.9, and the exploration rate (ϵ) to 1 with a decay factor of 0.01. We set the number of transitions N to 128 that we select from memory M of size 10000 and set C, which is the number of steps to update the target network, to 400.

Evaluation Metric. We report Transmission Rate Ratio (TRR), defined as the ratio of achieved to optimal TR for a given channel condition. We determine the optimal TR using Table I where each rate has a minimum SNR requirement.

Baselines. We compare against two counterparts from the literature: AARF [19], and an RL model from a prior paper (PP) [18]. We implement PP using the structure described in [18] and the learning strategies outlined above. We also compare to random modulation selection.

B. Performance Across RA Schemes

We evaluate `WideRate` across the counterpart NN architectures, learning strategies and in comparison with the two baselines. Our results are presented in Fig. 3. First, we observe that `WideRate` with LSTM-Decay outperforms all other counterparts, reaching 80% TRR after 250 observations, whereas the other two RL models remain below 70%. Random achieves an average of 60% TRR across all observations while AARF performs at 40% average. AARF's poor performance is due to continuous packet loss that results from changing channel conditions, which triggers it to fall back to the lowest available modulation.

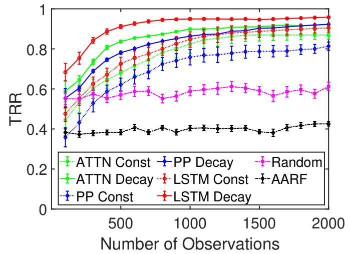

Fig. 3: Performance comparison across RA schemes.

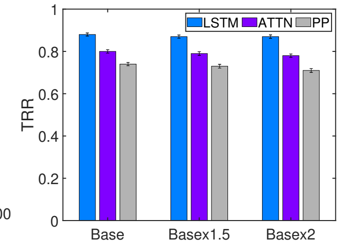

Fig. 4: `WideRate` with increasing vehicle speed.

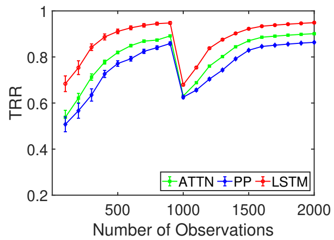

Fig. 5: Adapting to changing channel conditions.

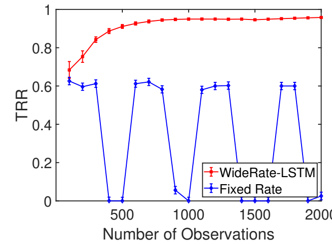

Fig. 6: `WideRate` in a real-world network traversal.

C. Effects of Mobility

We next evaluate the performance of our model with increasing vehicle speed. We compare across the three RL models using the epsilon-greedy decay strategy. We evaluate TRR while increasing the base speed by a factor of 1.5 and then by a factor of 2. Our base speed during the driving campaign averaged 35 mph. We simulate the increase of speed by modifying the input to the models; for x1.5 increase in speed, we skip every third SNR observation whereas for x2 speed we skip and every other observation. Our results are shown in Fig. 4 where we evaluate TRR after 400 observations. We observe that LSTM outperforms the other two models by 8% for ATTN and by 14% for PP. Additionally, LSTM maintains 88% TRR across all velocities, whereas ATTN slightly degrades across all velocities (78% to 76%). PP degrades the most from 75% to 71%. LSTM is able to maintain high and consistent performance despite the increase in speed.

D. Adapting to Dynamic Channel Conditions

We next evaluate `WideRate`'s adaptability to rapidly changing channel conditions. We modify a trajectory by injecting 100 SNR values (σ=18.32) after 900 observations. Fig. 5 presents our results indicating that TRR deteriorates rapidly across all models after 1000 observations. LSTM adapts quicker by taking 100 observations to achieve 75% TRR while ATTN takes 300 observations to achieve 74.5% and PP takes 400 observations to achieve 74.34%. Long-term LSTM outperforms all other models with 95% TRR overall.

E. `WideRate` performance in TVWS networks

We next explore `WideRate`'s benefits over the currently employed fixed modulation scheme in our partner's TVWS network. We select a trajectory, from our driving campaign, that used a fixed rate of `16-QAM 1/2` and we input this into

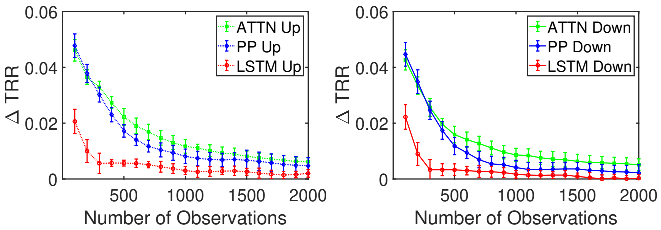

Fig. 7: WideRate with training on uplink and testing on downlink data (left) and vice versa (right).

our model. Fig. 6 presents achieved TRR for the duration of the trip. By selecting a fixed rate, TRR never approaches the optimal rate of 1 and results in a zero TRR where the SNR is low. LSTM achieves 90% TRR after 500 observations and 95% overall. LSTM is able to maintain high and consistent TRR across all observations. This demonstrates that our model is able to effectively adapt to a realistic changing channel.

F. Applicability to up- and down-link adaptation

Lastly, we investigate whether a single model is applicable for both up-link (UP) and down-link (DOWN) rate adaptation. We evaluate the performance when training on UP and testing on DOWN data and vice versa. For each combination we report ΔTRR = TRR$_{DAWA}$ - TRR$_{DA}$, where TRR$_{DAWA}$ is direction-aware training (i.e., both training and testing are performed in DOWN) whereas TRR$_{DA}$ is direction agnostic training. Fig. 7 presents the results for training on UP and testing on DOWN (left) and training on DOWN and testing on UP (right). First, we observe that for both scenarios, ΔTRR are extremely small, ranging from 0.01-0.05. We also observe that ΔTRR for LSTM does not exceed 0.022 for training on UP and testing on DOWN (left) and 0.0266 for training on DOWN and testing on UP which outperforms both PP and ATTN. These results show direction-agnostic training does not significantly deteriorate the performance of our framework and thus a single model can be used for both UP and DOWN RA.

V. Conclusion

We introduced WideRate, an RL RA framework that employs signal strength measurements for optimal rate selection. We showcased WideRate in the context of wide-area TVWS networks, by designing a vehicular mobile unit DMU to carry out an extensive measurement campaign in a 5-sector real world community network. We use the geo-tagged field traces collected to implement a real-world network simulator to evaluate WideRate. We show that WideRate significantly outperforms counterparts from the literature and is able to maintain continuous connectivity. WideRate can adapt quickly to changing channel conditions making it generalizable to unknown TVWS network topographies.

References

[1] International Telecommunication Union, "Measuring digital development: Facts and figures 2022." https://www.itu.int/itu-d/reports/statistics/facts-figures-2022/, 2022. Accessed 2024.

[2] Internet Society, "Connecting to Sovereignty with the Waimanalo Community Network." https://www.internetsociety.org/issues/community-networks/success-stories/waimanalo/, 2021. Accessed 2024.

[3] Internet Society, "Spectrum Access Helps One Tribal Community Weather Emergencies." https://www.internetsociety.org/impact-report/2021/stories-and-highlights/, 2021. Accessed 2024.

[4] N. Metri, "Piloting "White Space" to connect the underserved of rural Tanzania." https://www.internetsociety.org/blog/2018/04/piloting-use-tv-white-space-community-networks-rural-tanzania//, 2018. Accessed 2024.

[5] A. Ajayi, U. Roedig, C. Edwards, and N. Race, "A survey of rural wireless mesh network (wmn) deployments," in *APWiMob*, pp. 119–125, 2014.

[6] J. Ishmael, S. Bury, D. Pezaros, and N. Race, "Deploying rural community wireless mesh networks," *IEEE Internet Computing*, vol. 12, no. 4, pp. 22–29, 2008.

[7] M. Khalil, J. Qadir, O. Onireti, M. A. Imran, and S. Younis, "Feasibility, architecture and cost considerations of using tvws for rural internet access in 5g," in *ICIN*, pp. 23–30, 2017.

[8] M. Oliver and S. Majumder, "Motivation for tv white space: An explorative study on africa for achieving the rural broadband gap," 2019.

[9] J. Camp and E. Knightly, "Modulation rate adaptation in urban and vehicular environments: Cross-layer implementation and experimental evaluation," in *MOBICOM*, pp. 315–326, 2008.

[10] P. Shankar, T. Nadeem, J. Rosca, and L. Iftode, "Cars: Context-aware rate selection for vehicular networks," in *ICNP*, pp. 1–12, 2008.

[11] P. Deshpande and S. R. Das, "Brave: Bit-rate adaptation in vehicular environments," in *VANET*, pp. 33–42, 2012.

[12] T. Zhang, S. Sen, and S. Banerjee, "Enhancing vehicular internet connectivity using whitespaces, heterogeneity, and a scouting radio," in *MobiSys*, pp. 287–300, 2014.

[13] Y. Yao, X. Chen, L. Rao, X. Liu, and X. Zhou, "Lora: Loss differentiation rate adaptation scheme for vehicle-to-vehicle safety communications," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2499–2512, 2016.

[14] S. Yun and C. Caramanis, "Reinforcement learning for link adaptation in mimo-ofdm wireless systems," in *GLOBECOM*, pp. 1–5, 2010.

[15] V. Saxena, H. Tullberg, and J. Jaldén, "Reinforcement learning for efficient and tuning-free link adaptation," *IEEE Transactions on Wireless Communications*, vol. 21, no. 2, pp. 768–780, 2021.

[16] S.-C. Chen, C.-Y. Li, and C.-H. Chiu, "An experience driven design for ieee 802.11 ac rate adaptation based on reinforcement learning," in *INFOCOM*, pp. 1–10, 2021.

[17] W. Xu, C. Ma, S. Guo, and H. Zhou, "Efficient rate adaptation for 802.11af tvws vehicular access via deep learning," in *GLOBECOM*, pp. 1–6, 2019.

[18] W. Xu, S. Guo, S. Ma, H. Zhou, M. Wu, and W. Zhuang, "Augmenting drive-thru internet via reinforcement learning-based rate adaptation," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3114–3123, 2020.

[19] M. Lacage, M. H. Manshaei, and T. Turletti, "Ieee 802.11 rate adaptation: a practical approach," in *MSWIM*, pp. 126–134, 2004.

[20] Carlson, "Carlson Wireless Technologies." https://carlsonwireless.com, 2022. Accessed 2024.

[21] A. Kamerman and L. Monteban, "Wavelan®-ii: a high-performance wireless lan for the unlicensed band," *Bell Labs technical journal*, vol. 2, no. 3, pp. 118–133, 1997.

[22] I. Haratcherev, K. Langendoen, R. Lagendijk, and H. Sips, "Hybrid rate control for ieee 802.11," in *MobiWac*, pp. 10–18, 2004.

[23] J. C. Bicket, *Bit-rate Selection in Wireless Networks*. MIT Master's Thesis, Cambridge, MA, 2005.

[24] J. Kim, S. Kim, S. Choi, and D. Qiao, "Cara: Collision-aware rate adaptation for ieee 802.11 wlans.," in *INFOCOM*, pp. 1–11, 2006.

[25] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive mac protocol for multi-hop wireless networks," in *MOBICOM*, pp. 236–251, 2001.

[26] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media access for multirate ad hoc networks," in *MOBICOM*, pp. 24–35, 2002.

[27] G. Judd, X. Wang, and P. Steenkiste, "Efficient channel-aware rate adaptation in dynamic environments," in *MobiSys*, pp. 118–131, 2008.

[28] C. J. C. H. Watkins, "Learning from delayed rewards," 1989.

[29] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," vol. 518, pp. 529–533, Nature Publishing Group UK London, 2015.