Demo: WiChoose – Practical Network Selection for Wi-Fi Vehicle-to-Infrastructure Communication

Rui Meireles
Department of Computer Science
Vassar College
Poughkeepsie, NY, USA

Vinícius Abrunhosa
Instituto de Telecomunicações
University of Porto
Porto, Portugal

Ana Aguiar
Instituto de Telecomunicações
University of Porto
Porto, Portugal

***Abstract*—Opportunistic Wi-Fi access can be a useful complement to other Vehicle-to-Infrastructure (V2I) communication technologies. We focus on the problem of choosing between available Wi-Fi networks, to maximize the amount of transferable data. Our solution, WiChoose, uses passively observable variables such as signal strength to estimate throughput for each network. It then uses mobility information such as position and heading to forecast how throughput will evolve. Finally, these forecasts are used to determine the optimal network switching schedule. WiChoose was experimentally evaluated in a realistic scenario featuring IEEE 802.11n (long range, low throughput) and ad (short range, high throughput) networks. As the mobile client moved, WiChoose successfully alternated between the networks, transferring 95 % of what an oracle scheme with perfect knowledge of the future would have. We propose to demonstrate WiChoose's operation using the same two contrasting networks. The mobile client will be placed on a wheeled cart. Moving it back and forth will affect connectivity, leading WiChoose to alternate between the two networks to maximize data transfer.**

***Index Terms*—C.2.1.k Wireless communication, C.2.8.c Mobile communication systems**

I. Introduction

The pervasiveness of Wi-Fi in urban areas opens up the opportunity for inexpensive data offloading. In denser areas, multiple networks can be available, requiring a choice. We focus on this problem from the point of view of a delay-tolerant application that is trying to maximize total data transfer, i.e., the sum of throughput over time.

Our proposal, WiChoose [2], splits the problem into:

1) Estimating current throughput for each network. Using only passively-observable variables, to avoid overhead.
2) Forecasting throughput evolution. Mobility is a good proxy for throughput, so we use the mean of previously-seen throughput values given current mobility for this.
3) Periodically picking the best network, by applying a provably-optimal optimization algorithm to the forecasts.

WiChoose was experimentally proven capable of hopping between two Wi-Fi networks with complementary range and throughput, IEEE 802.11n and 802.11ad, to maximize total data transfer under a realistic vehicular mobility scenario [2]. We now propose a live demonstration of that capability.

This work was supported by: i) the European Union/Next Generation EU, through Programa de Recuperação e Resiliência (PRR) [Project Nr. 29: Route 25 (02/C05-i01.01/2022.PC645463824-00000063)]; ii) Fundação para a Ciência e Tecnologia, I.P. (FCT) under Project UIDB/50008/2020 [1], funded by FEDER through COMPETE.

II. System operation (more detail in [2])

*a) **Throughput estimation**:* To avoid introducing probe traffic into the network, instead of measuring throughput we estimate it based on received signal strength (RSSI), vehicle speed (s), and the number of active users (N_{users}). The following formulas for 802.11n and 802.11ad were obtained by applying symbolic regression to an experimental dataset:

$$\begin{aligned} \hat{tput}_n &= 0.7111 * RSSI - 2.479 * N_{users} \\ &\quad + 11.88 * e^{-N_{users}} + 62.02 \end{aligned} \tag{1}$$

$$\begin{aligned} \hat{tput}_{ad} &= 0.7334 * RSSI + 47.74 * \sin(s * RSSI) \\ &\quad - 112.6 * \tanh(s)^{1/4} \\ &\quad - 115.8 * \tanh(\cos(s)) * \log(N_{users})^2 + 387.9. \end{aligned} \tag{2}$$

*b) **Throughput forecasting**:* Signal strength predicts achievable throughput. Position predicts signal strength. Finally, current mobility predicts future position. Thus, we use it to indirectly forecast future throughput. We encode mobility as the combination of: road identifier (R), direction of movement (D) in degrees from north, position relative to the AP (P) in meters, and binary speed indicator (S), i.e., *stopped* or *moving*.

Assume that at time t, the vehicle has mobility m. Throughput at time $t + i$ can be forecast as the average of throughput values observed i seconds after the vehicle had the same mobility m in the past. Any average function can be used, but currently we employ a simple exponential moving average.

*c) **Network selection decision**:* Let t_0 be the current time. We want to maximize the total data transferable between t_0 and $t_{win_f - 1}$, where win_f is the forecasting window size. The maximum offloadable data from time t_i on ($i \geq 0$), given a current network $cnet$, is:

$$Data_{cnet}[t_i] = \begin{cases} Tput_{cnet}[t_i] \\ + \\ \max\limits_{\forall onet} fd(cnet, onet, t_i) & \text{if } i < win_f \\ \\ 0 & \text{if } i \geq win_f \end{cases} . \tag{3}$$

$Data_{cnet}[t_i]$ is the sum of the current throughput for $cnet$, $Tput_{cnet}[t_i]$, with the maximum amount of data that can be

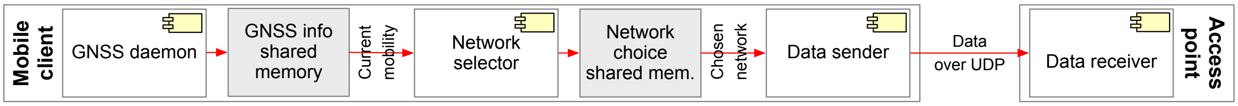

Fig. 1: Summarized view of WiChoose's logical and physical system architecture.

transferred in the future. The latter is given by maximizing function $fd(cnet, onet, t_i)$, which represents the future data if a switch from current network $cnet$ to network $onet$ is started at time t_i, over the available networks. It is defined as:

$$fd(cnet, onet, t_i) = \begin{cases} Data_{onet}[t_{i+1+ot}] & \text{if } onet \neq cnet \\ Data_{cnet}[t_{i+1}] & \text{if } onet = cnet \end{cases}. \quad (4)$$

The first case represents an actual network switch, which implies an outage period of ot seconds, hence why the same amount of time is skipped in the computation. The second case corresponds to staying on the current network, in which case data transfer continues uninterrupted.

Each passing second, WiChoose uses an optimal dynamic-programming algorithm to compute $Data_{cnet}[t_0]$ and pick the network yielding that maximum transferable data amount.

d) ***Prototype design and implementation:*** Fig. 1 depicts WiChoose's architecture. The key component is the network selector, deployed on the client node. It takes in mobility information from a Global Navigation Satellite System (GNSS) daemon, which it then combines with throughput estimates and forecasts to select the best network to use, using the logic described earlier. The decision is written to a shared memory region that data sending applications can read from.

We designed WiChoose to not just select networks, but also allow us to evaluate the quality of those selections. For this reason we collect ground truth throughput information for all available networks. For that purpose, we employ devices with multiple radio interfaces, each connected to a different network, and constantly stream data over each of them. Additionally, WiChoose can run multiple network selection strategies in parallel, for comparison purposes.

WiChoose was written in C++17 and has only two dependencies (libpthread and librt), making it easy to run on commodity Linux-based embedded systems. WiChoose is open source and available for download on GitLab [3].

III. Capability demonstration

WiChoose's key feature is its ability to hop between networks to maximize total data transfer, in arbitrarily complex environments. However, for logistical reasons, we demo it under the simplified scenario of Fig. 2. In it, a single client, mounted on a wheeled cart, moves back and forth while picking between two contrasting network options: a high-bandwidth, short-range 802.11ad network, and a low-bandwidth, long-range 802.11n one.

For simplicity, we use TP-Link Talon AD7200 routers [4] for both Access Point (AP) and client. They feature multiple Wi-Fi radios, letting us run multiple APs on a single device.

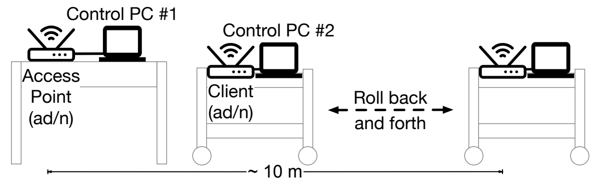

Fig. 2: Demonstration setup featuring wheeled cart.

To be able to run WiChoose, we replaced the original non-programmable operating system with LEDE-AD7200 [5], a Talon-tailored derivative of OpenWrt GNU/Linux [6]. This setup yields the following indoors performance:

Network	Communication range	Throughput
802.11ad	$<$ 10 m (line of sight)	$>$ 400 Mbit/s
802.11n	$>$ 20 m (non-line of sight)	$<$ 200 Mbit/s

When both networks are in range, 11ad is clearly preferable. However, moving the cart 10 m away from the AP should be enough to make 11ad throughput drop to zero, making 11n the best choice. Thus, rolling the cart back and forth should make the client hop between the two networks. If the demo space is small, cart movement can be supplemented by line-of-sight obstruction to achieve the necessary signal attenuation.

One issue with an indoor demonstration of WiChoose is its reliance on satellite-based positioning. To overcome this we have developed a simple GNSS daemon simulator. It simulates mobility according to a pre-programmed list of mobility data points, with user-triggered transitions. By providing a list that alternates between two distinct locations we can mimic the effect of wheeling the cart between those two locations.

In order to control the demonstration's execution, and display WiChoose's network selection in real time, both client and AP routers are connected to a laptop through Ethernet.

References

[1] F. para a Ciência e Tecnologia I.P. (FCT), "Project UIDB/50008/2020," 2020. [Online]. Available: https://doi.org/10.54499/UIDB/50008/2020

[2] R. Meireles, V. Abrunhosa, and A. Aguiar, "WiChoose: Practical Network Selection for Vehicle-to-Infrastructure Communication," in *2025 20th Wireless On-Demand Network Systems and Services Conference (WONS)*, 2025.

[3] R. Meireles, "WiChoose: Wi-Fi network selection for V2I Communication," https://gitlab.com/rui-meireles/wichoose, 2024.

[4] TP-Link, "AD7200 Multi-Band Wi-Fi Router Datasheet," 2018, (accessed on 2024-07-26). [Online]. Available: https://www.tp-link.com/en/home-networking/wifi-router/ad7200

[5] D. Steinmetzer, D. Wegemer, and M. Hollick, "Talon Tools: The Framework for Practical IEEE 802.11ad Research," 2017, (accessed on 2024-07-26). [Online]. Available: https://seemoo.de/talon-tools

[6] "OpenWrt - Open Wireless Router," 2024, (accessed on 2024-06-02). [Online]. Available: https://openwrt.org