Wi-Fi for Control: experimental network assessment

Matthias Pezzutto

***Abstract*— This technical report is to describe the attached Wi-Fi communication data. In the following, we provide a detailed description of the experimental setup and we accurately outline how data are saved. The attached data can be used to better understand the behavior of Wi-Fi networks for control applications and to validate control algorithms over unreliable communication links.**

I. Introduction and motivation

Wi-Fi is an attractive solution for future control applications over wireless because of its high bit-rates, which allow low sampling periods, and its large spreading, which ensures a large availability of off-the-shelf devices and technical support. Unfortunately, Wi-Fi is prone to packet losses and random delays due to the intrinsic unreliability of wireless channels. This behavior is emphasized because Wi-Fi is not specifically conceived for control applications but it favors the overall throughput over the timeliness of the communications. The overall behavior is difficult to be accurately modeled because it depends on the complex communication systems adopted by Wi-Fi (e.g. stochastic back-off time, multi-antenna communications, frequency slot allocation) and on the uncontrolled channel conditions (e.g. channel access with competing devices, external interference). To overcome the lack of suitable models, we provide Wi-Fi communication data collected through accurate experiments. This data can be used to better understand the behavior of Wi-Fi when used in a standard control setup and to validate control strategies with realistic channel evolutions. In this technical report we provide the necessary information to use the attached data.

II. Experimental setup

In the following, we provide a detailed description of the experimental setup. The idealized setup is reported in Fig. 1.

A. Devices

The experiment comprises a remote PC (Intel Core i5-6400, 2.70 GHz, 16GB RAM, Ubuntu 17.04, with wireless Qualcomm Atheros AR9227 chip and PC-link interface) and a local board (Raspberry Pi 3 mod. B with Broadcom BCM2837 Wi-Fi module). The two devices are connected by a Wi-Fi network (IEEE 802.11n standard) in which the remote PC is set as Access Point and the local board is the only node. The two devices are $5\ \mathrm{m}$ apart.

For any clarification, question, request, do not hesitate to contact the author at matthiaspezzutto@gmail.com. Current version: 2022/04/14.

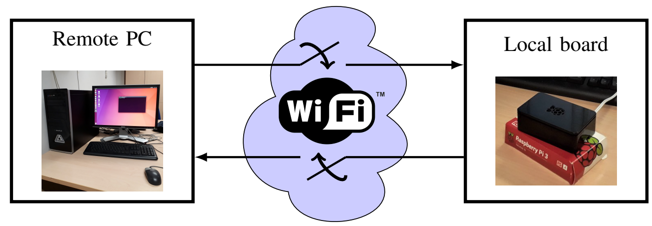

Fig. 1. Setup. Devices and network involved in the experiments.

B. Network

The network has been set up by exploiting the Wi-Fi interfaces of the devices and leveraging the common Internet protocol suite stack of the Linux kernel. Further technical details of the implementation can be found in [1].

At the Physical layer, MCS index is set equal to HT7 (64-QAM modulation, 5/6 coding rate) providing a rough bit-rate of 64Mbit/s. This choice is intended to allow for the transmission of possibly large packets while achieving good robustness against moderate noise and interference.

At the Data Link layer, the number of MAC transmission retries is set equal to 1 with the aim of decreasing the traffic due to old information. Moreover, power-safe mode is disabled to avoid uncontrolled resting periods of the devices.

At the Transport layer, UDP protocol is selected to further decrease the number of transmitted packets. This is important to avoid additional packet losses when the channel capacity is approaching the limit but the network is busy due to re-transmissions.

At the Application layer, we implement suitable routines to perform periodic transmissions between the two devices. A complete description is given in the following subsection.

C. Testing application

Suitable testing applications are coded to transmit time-stamped packets from the remote PC to the local board and the other way around, with a fixed time-span T between two following transmissions from each device. Transmission instants from the two devices are separated approximately of a period $T/2$. Based on the time-span T, we can obtain a sequence of time instants indexed by $t \in \mathbb{Z}$ so that t refers to the time instant $Tt \in \mathbb{R}$ for a suitable initial instant. For clarity of exposition, we may think that the transmission from the local board occurs at time Tt, while transmission from the remote PC occurs at time $Tt + T/2$. A pictorial representation is reported in Fig 2. This setup mimics the communication at sampling instants between a plant and a remote control unit. In particular, the temporal shift between

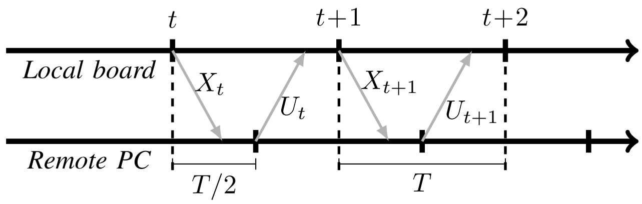

Fig. 2. Configuration. An arrow ideally represents a transmission of a packet, indicated by the label. The tail of the arrow indicates the starting instant of the transmission and the transmitter side, while the head indicates the end of the transmission and the receiver side. We can see that transmissions from the two sides are shifted of $T/2$. By convention, the time sequence is determined by the transmission from the local board.

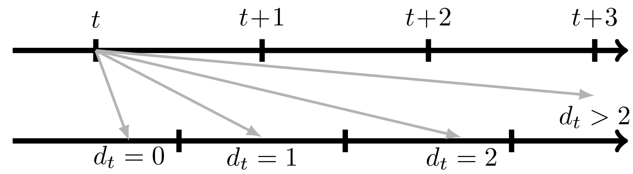

Fig. 3. Possible delays. Note that a null delay is possible due to the particular definition of the arrival process and the temporal shift between the transmissions from the two sides.

the two sides allows the control input at time t to exploit the information on the system state at time $t-1$.

Let X_t denote the packet transmitted by the local board during the period $(t, t+1)$, and U_t denote the packet transmitted by the remote PC during the period $(t, t+1)$. The arrival process on the link from the board to the PC, referred to as downlink, is represented by the binary variables

$$\gamma_{t-i}^{t} = \begin{cases} 1 \text{ if } X_{t-i} \text{ is available at the PC before sending } U_t \\ 0 \text{ otherwise} \end{cases}$$

Similarly, the arrival process on the link from the PC to the board, referred to as uplink, is represented by the binary variables

$$\theta_{t-i}^{t} = \begin{cases} 1 \text{ if } U_{t-i} \text{ is available at the board before sending } X_{t+1} \\ 0 \text{ otherwise} \end{cases}$$

Note that, if a packet is available at a time instant k, then it will be available also for all future time instants $t > k$. Through a suitable algorithm, we compute and store the values of γ_{t-i}^{t} and θ_{t-i}^{t} occurred during the experiments.

D. Environment and external interference

The test is carried out in the laboratories of the University of Padova. The room resembles the typical office environment where several Wi-Fi networks and several users are present. More specifically, the Wi-Fi network used for the experiments is set at channel 7 (2442 Mhz), while co-existing Wi-Fi networks are present at channels 5, 9, and 12. The interference level mainly depends on the traffic load on the co-existing networks and may range substantially. We collect data on different days and in different noise conditions. In particular, in order to test the limit behavior of the network, we manually add constant download traffic on the network operating at channel 9, with a limited bit-rate up to 40Mbit/s. The end-user of this traffic is located close ($\simeq 1$ m) to the remote PC. This setup aims to resemble a realistic environment where control systems with wireless communications can take place. This can be the case of future mobile autonomous systems, which will operate in environments where Wi-Fi networks might be present to provide Internet connection to other users, or control systems in factories, where Wi-Fi networks might be present to serve the higher levels of the automation pyramid.

III. Description of the Results

A. Useful quantities

From the arrival processes, several important channel parameters can be computed. In the following, we define some quantities relevant from the control point of view. For simplicity, the definitions are given for the downlink, but the same quantities can be introduced for the uplink.

We define the delay d_t with which the packet transmitted at time t is received as

$$d_t = \min d \quad \text{s.t.} \quad \gamma_t^{t+d} = 1 \tag{1}$$

Possible values of d_t are reported in Fig. 3. According to the proposed setup, a null delay is possible. Let the age of information a_t at time t be defined as

$$a_t = \min a \quad \text{s.t.} \quad \gamma_{t-a}^{t} = 1 \tag{2}$$

which can be seen as how old is the last received packet at time t. Let the blackout length b_t at time t be defined as

$$b_t = \min b \quad \text{s.t.} \quad \gamma_{t-b}^{t-b} = 1 \tag{3}$$

which represents the number of consecutive packets at time t that are not arrived before the following transmission instant, usually regarded as packet losses in the literature. Note that b_t can be easily computed recursively based on γ_t^t.

B. General overview of the dataset

The dataset contains 18 experimental realizations of the arrival sequences obtained with different sampling periods ($T \in \{1\,\mathrm{ms}, 5\,\mathrm{ms}, 10\,\mathrm{ms}, 20\,\mathrm{ms}\}$) and network conditions (with and without manually-added traffic on a co-existing network with a fixed bit-rate of 40Mbit/s). Each experimental realization is $30\,\mathrm{s}$ long and contains both the uplink and downlink evolutions. For sake of simplicity, packets with a delay longer than 10 steps are automatically discarded.

Each experimental realization is stored in a Matlab `.mat` file and it is conveniently represented through 5 different data structures. First, the vector `pktDelay` stores the delays of each packet, i.e. the t-th entry of the vector contains d_t. The entries associated to lost packets, whose delay would be infinity, are set equal to -1. In a similar way, we define the vector `blackout` storing the current blackout, i.e. the t-th entry of the vector contains b_t, and the vector `ageInformation` storing the current age of information, i.e. the t-th entry of the vector contains a_t. Then, the matrix `receivedId` stores the time-stamps of the packets received at the current time instant, i.e. the t-th row of the matrix is a vector including the time stamps of new packets arrived during the previous sampling period. Along a row, a more recent packet precedes less recent packets. The matrix is padded with -1 when no more new packets have arrived. Finally, the vector `pktLossStatus` store the loss sequence, i.e. the t-th entry of the vector contains γ_t^t. To simplify the readability of the file, we also include the vector `pktDelayDistr` that stores the empirical probability density function of delays, i.e. the i-th entry contains $P(d_t = i)$.

Provided data structures are redundant but allow to simplify the access to the experimental arrival sequences. Each file can be easily uploaded to Matlab workspace (using `load`) and the arrival sequences can be used to test the evolutions of simulated systems with novel control algorithms.

C. Content list of the dataset with details

In the following we provide a brief characterization for each attached experimental realization. In particular, we indicate $p_0 = P(d_t > 0)$, $p_{10} = P(d_t > 10)$, $b_{\max} = \max_t b_t$, and $a_{\max} = \max_t a_t$.

The following realizations are obtained with sampling period $T = 1$ ms with and without additional interference. Due to the very low sampling period, the application is challenging also for small interference level. For this reason, the manually-added traffic is limited to 10Mbit/s.

`ws_1ms_1.mat`

Uplink	$p_0 = 0.32$	$p_{10} < 0.01$	$a_{\max} = 14$	$b_{\max} = 22$
Downlink	$p_0 = 0.11$	$p_{10} < 0.01$	$a_{\max} = 31$	$b_{\max} = 40$

`ws_1ms_noise.mat`

Uplink	$p_0 = 0.16$	$p_{10} = 0.01$	$a_{\max} = 50$	$b_{\max} = 99$
Downlink	$p_0 = 0.54$	$p_{10} = 0.03$	$a_{\max} = 67$	$b_{\max} = 215$

`ws_1ms_high_noise.mat`

Uplink	$p_0 = 0.50$	$p_{10} = 0.27$	$a_{\max} = 199$	$b_{\max} = 1696$
Downlink	$p_0 = 0.60$	$p_{10} = 0.31$	$a_{\max} = 349$	$b_{\max} = 1995$

The following realizations are obtained with sampling period $T = 5$ ms with and without additional interference.

`ws_5ms_1.mat`

Uplink	$p_0 < 0.01$	$p_{10} < 0.01$	$a_{\max} = 1$	$b_{\max} = 1$
Downlink	$p_0 = 0.01$	$p_{10} < 0.01$	$a_{\max} = 1$	$b_{\max} = 2$

`ws_5ms_2.mat`

Uplink	$p_0 = 0.03$	$p_{10} < 0.01$	$a_{\max} = 1$	$b_{\max} = 2$
Downlink	$p_0 = 0.05$	$p_{10} < 0.01$	$a_{\max} = 3$	$b_{\max} = 4$

`ws_5ms_3.mat`

Uplink	$p_0 = 0.08$	$p_{10} < 0.01$	$a_{\max} = 4$	$b_{\max} = 4$
Downlink	$p_0 = 0.10$	$p_{10} = 0.03$	$a_{\max} = 5$	$b_{\max} = 7$

`ws_5ms_noise_1.mat`

Uplink	$p_0 = 0.45$	$p_{10} < 0.01$	$a_{\max} = 8$	$b_{\max} = 15$
Downlink	$p_0 = 0.56$	$p_{10} = 0.03$	$a_{\max} = 11$	$b_{\max} = 29$

`ws_5ms_noise_2.mat`

Uplink	$p_0 = 0.75$	$p_{10} < 0.01$	$a_{\max} = 9$	$b_{\max} = 31$
Downlink	$p_0 = 0.87$	$p_{10} = 0.03$	$a_{\max} = 20$	$b_{\max} = 68$

`ws_5ms_noise_3.mat`

Uplink	$p_0 = 0.62$	$p_{10} < 0.01$	$a_{\max} = 10$	$b_{\max} = 19$
Downlink	$p_0 = 0.74$	$p_{10} = 0.03$	$a_{\max} = 12$	$b_{\max} = 46$

`ws_5ms_noise_4.mat`

Uplink	$p_0 = 0.66$	$p_{10} < 0.01$	$a_{\max} = 10$	$b_{\max} = 22$
Downlink	$p_0 = 0.75$	$p_{10} = 0.06$	$a_{\max} = 17$	$b_{\max} = 35$

`ws_5ms_noise_5.mat`

Uplink	$p_0 = 0.67$	$p_{10} < 0.01$	$a_{\max} = 11$	$b_{\max} = 33$
Downlink	$p_0 = 0.76$	$p_{10} = 0.05$	$a_{\max} = 15$	$b_{\max} = 37$

`ws_5ms_high_noise.mat`

Uplink	$p_0 = 0.81$	$p_{10} = 0.04$	$a_{\max} = 24$	$b_{\max} = 88$
Downlink	$p_0 = 0.90$	$p_{10} = 0.23$	$a_{\max} = 28$	$b_{\max} = 120$

The following realizations are obtained with sampling period $T = 10$ ms with and without additional interference.

`ws_10ms_1.mat`

Uplink	$p_0 < 0.01$	$p_{10} < 0.01$	$a_{\max} = 1$	$b_{\max} = 1$
Downlink	$p_0 < 0.01$	$p_{10} < 0.01$	$a_{\max} = 1$	$b_{\max} = 1$

`ws_10ms_2.mat`

Uplink	$p_0 < 0.01$	$p_{10} < 0.01$	$a_{\max} = 2$	$b_{\max} = 2$
Downlink	$p_0 = 0.15$	$p_{10} < 0.01$	$a_{\max} = 7$	$b_{\max} = 7$

`ws_10ms_noise_1.mat`

Uplink	$p_0 = 0.15$	$p_{10} < 0.01$	$a_{\max} = 3$	$b_{\max} = 5$
Downlink	$p_0 = 0.53$	$p_{10} = 0.07$	$a_{\max} = 5$	$b_{\max} = 12$

`ws_10ms_noise_2.mat`

Uplink	$p_0 = 0.20$	$p_{10} < 0.01$	$a_{\max} = 5$	$b_{\max} = 6$
Downlink	$p_0 = 0.46$	$p_{10} = 0.06$	$a_{\max} = 5$	$b_{\max} = 9$

`ws_10ms_high_noise.mat`

Uplink	$p_0 = 0.46$	$p_{10} = 0.04$	$a_{\max} = 23$	$b_{\max} = 63$
Downlink	$p_0 = 0.43$	$p_{10} = 0.12$	$a_{\max} = 13$	$b_{\max} = 24$

The following realization is obtained with sampling period $T = 20$ ms with additional interference. The case without additional interference is not included because it presents only sporadic packet losses.

`ws_20ms_noise.mat`

Uplink	$p_0 = 0.28$	$p_{10} < 0.01$	$a_{\max} = 6$	$b_{\max} = 6$
Downlink	$p_0 = 0.18$	$p_{10} = 0.09$	$a_{\max} = 5$	$b_{\max} = 8$

References

[1] F. Branz, R. Antonello, F. Tramarin, S. Vitturi, and L. Schenato, "Time-critical wireless networked embedded systems: Feasibility and experimental assessment," *IEEE Transactions on Industrial Informatics*, 2020.