Wireless Impairments Detector

Israel Márquez Salinas
Université Pierre & Marie Curie
s.marquez.israel@gmail.com

ABSTRACT

Lorem ipsum

1. INTRODUCTION

In this section we tell the story of why are we working to address the question related to Wireless impairments.

The main goal is to be able to identify Wireless Impairment in the Home Wireless.

Why? - Because most of the times the bottle neck in an end-to-end communication tends to be the Home Wireless.

Users gets frustrated as they do not know how to approach the problem. In fact, even with wireless networks knowledge identifying the root cause can be challenging.

Our work will become part of a tool which strives to identify where the bottleneck in and end-to-end communication is. The tool is already deployed in the wild.

2. BACKGROUND AND RELATED WORK

In this section we present a *high level overview* of what has been done before with regards to Wireless conditions measurements.

We refer to literature and mention the methods and metrics collected from those methods to predict, infer or asses Wireless conditions.

2.1 Wireless Monitoring Metrics

Here we list the metrics found in literature and list them under the two families, active and passive.

Active

- One-Way Delay
- Round Trip Time
- More

Passive

- Rate
- RSSI
- More

2.2 Where do we collect them?

We list the different vantage points from where the metrics have been collected. We can also include the accuracy.

Note - From what I recall most of metrics have been collected from APs, which means we need to have access to the AP.

- Station
- Access Point
- Server

3. WIRELESS BOTTLENECK DETECTOR

In this section we describe the tool we have created.

It is a *custom* version of Ping in *GoLang*. This customer version allow us to define a probing rate, send probes in batches and set an inter-space between probes and batches.

Explain we have used exponential distribution to send batches. We have chosen exponential as Poisson process is related to exponential arrival times. We chose Poisson because sampling a Poisson process results in Poisson process, which allows to keep the same Poisson process even after sampling.

The sampling technique we used is Bernoulli, which is a type of Poisson sampling. In Bernoulli sampling all the observation in the data set have the same probability to become or not to become part of the resulting sampling set.

We varied the probability to be part of the sampling from 10% to 90%. To choose the sample which resembles the most to our original data set we worked with *Two Sample Kolmogorov-Smirnov* Test

Our results are:

Include chart with the results of similarity test.

In the plot we can see that with a probability of 50% we overlap our original data set.

Include Plot of ECDF of original data set and sample

We chose 50% as it results on an overlap with the original data set.

We can also describe that the p-value is close to 1 and the D-Value, which is the KS statistic is low. KS Low value is pursed as it means distance between the two ECDFs is small, meaning they are close to each other, hence more similar.

4. EVALUATION METHOD

Note: Ask on this section, as we might have already described it in the previous section.

4.1 Setup

Here we describe the setup we have in our lab and the test bed we have used in Orbit.

We have worked with two setup, initially our office lab and then Orbit.

In-lab

In our lab we have worked with a Raspberry Pi 3 running Raspbian GNU/Linux 8 (jessie). Wireless Access Point TP-Link AC1750. Dell Laptop Inspiron with Wireless Driver – *Driver Version* List Protocols supported by the Wireless card 802.11 a/b/g/n/ac Laptop running Ubuntu 16.04.4 LTS (Xenial Xerus)

Orbit

We used three nodes with. Atheros 9k and 5k wireless cards.

We configure a node to work as a Wireless station, another as an AP and finally a third one as a wired client from where the pings were issued.

The third node working as a wired client plays a similar role as the Pi in our In-lab setup.

4.2 Evaluation method

Here we explain how we ran the experiments.

In our lab we placed the laptop and the Pi close to each other, a distance smaller than 5 m. We connected to the 5GHz band under 802.11n protocol.

The first set of experiments consisted in progressively adding TCP sessions. The goal was to perceive how was RTT changed with more TCP sessions. We expected to see an increase as more TCP session were added.

Results matched our expectation and saw an increase in average RTT as more TCP session were added.

Include plot in which we have the CDF of RTTs vs TCP Streams

The next set of experiments were ran with the goal of finding a suitable probing rate. The ideal case is to probe frequent enough to have a "good" sense of the network without adding overhead and disrupting the Wireless Network.

We issued pings in sessions of 10 min at a ping rate of 100msec, initially, we call this aggressive scenario. The rate was defined to be 100msec to set our baseline from which we derived our sampling to obtain a suitable probing rate. The main goal is to achieve a rate which is not as aggressive as probing every 100msec.

After completing our sample analysis, we define it to be 200msec and we proceed to run test in Orbit where we can modify parameters as attenuation.

Orbit lab allow to modify attenuation from 0 dB to 30 dB. We perceived an increase in average RTT and loss rate from 27dB to 29dB. (At 30 dB link is unusable).

The results are show in the following plots.

Include Plots with Avg RTTs and Loss Rate results from Orbit

5. RESULTS