Wireless Impairments Detector

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

ABSTRACT

Home WLANs have become an essential element in households nowadays. The preferred method to access Internet from home are WiFi Home WLANs. Home WLAN have brought their benefits and challenges into the home. The variety and complexity of WiFi and non-WiFi devices make Home WLANs keen to experience impairments. These impairments can be complex to identify even by Wireless experts. In this paper we present our tool which main goal is to help to identify Wireless impairments by relying on active and passive measurement techniques. Our tool collect metrics from the Wireless client and AP to get a sense on how each of these devices perceive the environment. Along the Wireless metrics collection, we actively collect bandwidth and RTT metrics. Both metric types are collected during different environment conditions which we have instrumented in the Wireless test bed for this work. Finally we correlate passively collected metrics with active ones to help to identify when a Wireless complication is present.

1. INTRODUCTION

Networks today have evolved significantly, one of the most tangible examples of this evolution are Wireless Network. In this paper we focus on 802.11 WLANs. These WLANs are common in offices, manufacturing plants, homes and many more premises. Home WLANs have brought their benefits and challenges into the home. One of the challenges of Home WLAN comes from the wide diversity of devices inside the home. In one end we have the 802.11 devices, also know as WiFi devices, and in other hand the non-Wifi devices. The WiFi devices variety ranges from handheld devices with limited resources, to complex, resourceful entertainment systems. All of these 802.11 devices will experience a different performance of the home WLAN. Whilst a gaming console can perceive the home WLAN as acceptable, a resource-constrained hand-held might perceive it as poor. This difference is derived from the characteristics of each device, such as Wireless card adapter. The second set of devices are non-Wifi. The non-WiFi devices use the same medium as WiFi and can create conditions leading to poor Home WLAN experience. Examples of such devices are microwave ovens, cordless phones, and bluetooth devices.

Even though Home WLANs are the preferred way to access Internet at home [1] a robust mechanism to detect complications in Home WLAN has not been defined yet. Instrumenting a tool or mechanism to identify Home WLANs complications is challenging due to several reasons. The most critical one is the Wireless medium itself. Wireless by nature is unreliable and variable making hard to accurately "catch" a problem [1]. Interference from non-WiFi devices can be experienced at any time and the length of these episodes can vary. The diversity in terms of WiFi devices characteristics is another factor to consider. A handhled device and a laptop placed at almost the same distance from the can experience Wireless performance differently. This difference, as mentioned before can be derived from the characteristics of each device, battery, Wireless adapter, operating system and more.

A tool to identify home WLAN issues can help to mitigate pain points end users and ISPs suffer today. Previous work has pointed out that end users tend to reach out to their ISPs and hold them responsible for poor home network performance [2]. In most of the cases Home WLAN is the root cause of poor home network performance. Both parties, end users and ISPs have limitations to tell if the Home WLAN is the root cause. From one side end users have visibility from within the home WLAN but most of lack the knowledge to troubleshoot it. On the other side, ISPs have the knowledge to troubleshoot WLANs but lack visibility from within the home WLAN. This is where our motivation to develop a tool to help to identify home WLANs comes from.

In this paper we present our tool which main contribution is to detect Wireless impairments in Home WLANs. Our tool relies on two methodologies to measure network performance, active and passive. From the passive approach we collect 802.11 metrics from the Wireless client and the Access Point. These metrics helps us to get a sense of what are the client and AP perceiving from the home WLAN. From an active ap-

proach we trigger bandwidth measurement and RTT using a Wired client. To measure bandwidth we use *iPerf*. The Wireless client is setup as iPerf client and a wired client connected to the AP plays the role of the iPerf server. In a similar approach we conduct RTT measurements, the wired client pings the Wireless client. The tool we use to issue the pings is a custom tool developed in *GoLang* which allows us to include custom parameters unavailable in the standard ping tool. These metrics have been collected under different environment conditions. Environment conditions have changed using attenuation, congestion and noise. Further details of the design of our tool will be described in section 3.

2. BACKGROUND AND RELATED WORK

In this section we present a *high level overview* of what has been done with regards to Home WiFi problems identification.

We present the two methods for network measurements, active and passive. We describe the main characteristics of each of them and list the pros and cons of each one.

2.1 Wireless Monitoring Metrics

Section to describe active and passive measurements.

Active

Active measurement techniques are mainly characterized by its ability to capture the state of the network in real-time. In other words, active measurement can help to identify a condition when is present in the network. This characteristic is different from passive network which can be considered historical. Active measurements are also characterized by the use of probes. Probe packet are used to measure the state of the network. For example, ping, relies on ICMP request and replies to compute the Round-Trip Time from one host to another. An important consideration to bear in mind when using active measurement techniques are the probes. It is important to define the size of the probes and how frequent are they sent into the network. Probes can add overhead to the network if their size is high compared to the capacity of the path or if they are sent too frequent. If probing is causing overhead to the network, the condition we want to measure will be caused by the same tool we are using to measure. Probing without the proper due diligence can lead to network degradation and/or biased measurements. In the following bullet points we outline the pros and cons of active measurement techniques.

Pros

- Full ownership of the network is not required.
- They do not require large space to store data collected as generally, probe packets are small.
- Privacy concerns are minimal as probe packet used to measure are made of random data which has no sensitive information.
- Useful to get the state of the network in real-time.

Cons

- They add overhead to the network as probe traffic is generated to conduct measurements.
- The very same probe packets being used to measure network conditions can cause network degradation.
- Biased results can be obtained if probing is conducted without due diligence of network conditions.
- They can only capture an instant of the network condition. If problem to be characterized is extended in time, active measurement might not measure it accurately.

Active Metrics

- **Round Trip Time**
  - This metric takes into account the time it takes for a probe to leave the source, reach the destination and come back to the source.
- **Bandwidth**
  - The amount of data that can be sent or received from or by a station will allow to identify how far are we from the PHY data rate. In practice the bandwidth is less than the PHY rate at which the station has connected to in 802.11 protocol PHY rate.
  - In other words this measurement can help to identify how efficiently is the medium being used.
- **Losses**
  - Losses from Ping statistic will allow us to identify the rate loss during the experiment window.
  - *Review Code changes to be made in Go in order to add a second queued.*

Passive

Passive measurement techniques rely on a listening approach. The instrument conducting passive measurements in the network sits in a specific location along the path and records the metrics of interest. The instrument can be a component of then network itself, like a router, or can be a device devoted to measure, like a Wireless sniffer. An important different between

active and passive techniques, is that passive tend to be historical. In an historical sense, passive measurements are more reliable to characterize a network problem which cover an extended time-frame. Active measurements are suitable to pinpoint a problem in the instant it happens, nevertheless they lack accuracy to characterize problems which cover and extended timeframe. Another difference with active measurements is that passive measurements do not trigger probes. Overhead within the network path caused by probe packets is not present with active measurement. When implementing passive measurements it is important to consider the resources of the measuring device. The device might require to have enough storage to store the data being collected. In a similar way, the computing capability of the device an be required to be high depending on the speed of the link being measured. A Gigabit link in a Core Router will produce significantly more data than an 100Mbps Ethernet link of an access switch. In our tool we have passive tools such implemented Wireless Sniffer and WiFi metrics collection from Wireless client and Access Point. Outlined in the following list a high level summary of the Pros and Cons of passive measurement techniques.

Pros

- No extra traffic is generated to collect metrics, risk of causing overhead is minimized.
- They are better suited to accurately characterize network problems which cover an extended time frame.
- In general they are able to collect large dataset leading to increased accuracy of network complications.

Cons

- Data collected by them can be large. Large storage can be required to store data collected.
- Access to devices within the network is required in order to place the passive instrument.
- Measuring device might require to have computational power depending on the link being monitored and the granularity of data pursued.
- They are reactive, findings of network problem can be obtained after data has been analyzed.

Passive Metrics

- **RSSI - Received Signal Strength Indicator**
  - The power at which the signal is being received by the device. Depending on the the type of traffic specific threshold can be defined for it. For example, for VoIP the min value for it is -68 dBm. What is consider a strong signal level is -40 dBm.
- **Busy Time**
  - This metric can tell us how busy was the channel, in other words if the channel the device is working on is close to 100% it can be due to contention by other Wi-Fi devices or interference from non-Wi-Fi sources.
- **PHY Tx Rate**
  - The rate at which without medium access control, error correction or scheduling events the device is expected to operate with. *The PHY rate can be obtained from a radio tap, checking the 802.11 header and check the precise bit rate at which that specific frame was transmitted.*

2.2 Where do we collect them?

We list the different vantage points from where the metrics have been collected in our work. We describe the reason why are collecting them from the specific vantage location we have chosen. For example accuracy, getting RSSI from a client A will be different from client B, even if they are located at a similar distance from AP.

- **Bandwidth or Throughput** - We will measure it from the client as it is from the user that we want to get a sense of his experience. It is know that even with similar Wireless conditions devices can experience different bandwidth and bitrate. [3]. Therefore we will measure bandwidth from the client perspective at not from the AP. Measuring from the AP will give us a sense of the AP perspective and not from the client which is the device from which the users access services. This metric is to be obtained using iPerf with UDP traffic.
- **PHY Tx Rate** - We will extract this metric from WiFi metrics at AP to identify at which rate was the frame being received sent. This correlates to identify the actual bit rate at which each frame is being sent. Metric is to be obtained from AP running Linux-based OS with the command *iwconfig. Note - We can also add a sniffer between AP and wireless client to validate that stats collected from commands match the radiotap headers.*
- **RSSI** - Measure to be collected at the wireless client. Based on the location of the client device, obstacles in the path, contention and interference will impact RSSI. Metric is to be obtained from Linux-based client with the command *iwconfig.*
- **Noise** - A factor contributing to Wireless degradation is Noise, it is the Wireless interference coming

from non-Wi-Fi sources. This can be caused by Microwave ovens, cordless phones and similar devices which "do not speak Wi-Fi language". Noise will be measured at both ends, client and AP as we strive to identify which one experience the more and less noise to pinpoint where the complication might rely. Poor AP or client placement. Command *cat /proc/net/wireless.*

- **Busy Time** - We will measure it from the AP as it is the one servicing clients in a specific channel (channels), meaning that clients will be connecting to that serving channel. If the busy time of the channel is high, means there is interference or other AP and clients using the same channel. A corrective action can be switching to a less busy channel. The busy time is made of two components, WiFi (congestion) and non-Wi-Fi (interference). Command *iw INTERFACE survey dump*

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

Main characteristics of Ping Tool.

- The ping tool being used has been customized to be able to send trains of probes.
- The tool allows to define a probing rate based on a Poisson process, we have chosen a Poisson process as we sample from it. Sampling from a Poisson process leads to another Poisson process.
- Our sampling rate has been defined to be 200 msec based on sampling and similarity test results.

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

Main ideas for Orbit test bed description

1. Orbit is a testbed mostly devoted to Wireless experiments. (Mostly as they also have SDN sandboxes to test SDN technologies)
2. We have been using the Sandbox 4, SB4, which is devoted to Wi-Fi and Wi-Max Experiments.
3. SB4 is made of 9 nodes, each of them runs Linux based systems, Ubuntu 12.04 to be precise.
4. Our main setup is composed by three nodes. One node plays the role of the AP, another the role of Wireless client and the last one is a Wired client.
5. We are using 802.11n in 2.4GHz band to improve the reachability of the AP and the Wireless client.
6. The wired client is the source of the probes and iPerf server.
7. Wireless client plays the role of iPerf client.
8. We can include a diagram of the Orbit SB4 deployment and include the proper references.

Main ideas for the evaluation methods

- **Attenuation**
  - We have been using the embedded manger for attenuation in Orbit.
  - We can instrument attenuation values on the links connecting the nodes, in our case we vary the attenuation values between Wireless client and AP.
  - Attenuation controller allows to define values in the range from 0 - 30 dBm.
  - For our experiment we have been varying the values from 0 to 30 in steps of 3.
  - We vary the attenuation and record the RTTs for pings.
  - We have identified that after 27 dBm of attenuation is when we begin to see an increase in RTTs, each session last 10 min. Probe rate every 200msec.
  - At 30 dBm the connectivity between Wireless client and AP is lost.
  - For bandwidth test we have run iPerf and recorded the bandwidth obtained at the client side.
  - **With 5GHz we identified that after 6dBm the connectivity between client and AP is lost.**
  - We have setup 802.11n using 2.4GHz band to increase the range.
  - The goal is to run iPerf and identify at which attenuation levels does the bitrates drops, record the attenuation values to run ping tests.
  - Once the attenuation values have been identified the next step is to run ping tests using the attenuation values found with iPerf test and record the average RTTs.
- **Interference**
  - Currently looking for a way to create Noise in SB4.
  - Check if for a specific time they can setup a Microwave oven or similar.
- Congestion
  - For this experiment we will deploy a second wireless client connected to the same AP.
  - The 2nd Wireless client will send traffic to the iPerf server located in the wired client.
  - The original client will continue to send pings to the wired client.
  - We will record the results of RTT while other client is sending traffic to the iPerf Server.

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.

We can set a "cost" to our experiments based on overhead at the following points.

- Network
- Device
- Router

We can include the accuracy of our methods depending on where are we setting our Vantage point.ex

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

In the closing section we summarize what we have achieved, similar to what we have discussed at the closing of the previous section, 4.

Based on the tool and the methodology we used we outline the results we obtained.

What are our results telling us?

Can we identify impairments from the chosen metrics?

Which of the two methods, active or passive, can be considered to best suit the detection of Wireless impairments?

Why is the chosen method more suitable?

Future Work can be mentioned to describe the integration of this work with the project with Princeton.

6. REFERENCES

[1] Diego Da Hora, Karel Van Doorselaer, Koen Van Oost, and Renata Teixeira. *Predicting the effect of home Wi-Fi quality on QoE*. PhD thesis, INRIA; Technicolor; Telecom ParisTech, 2018.

[2] Diego Neves da Hora, Karel Van Doorselaer, Koen Van Oost, Renata Teixeira, and Christophe Diot. Passive wi-fi link capacity estimation on commodity access points. In *Traffic Monitoring and Analysis Workshop (TMA) 2016*, 2016.

[3] Srikanth Sundaresan, Nick Feamster, and Renata Teixeira. Measuring the performance of user traffic in home wireless networks. In *International Conference on Passive and Active Network Measurement*, pages 305–317. Springer, 2015.