Bare Demo of IEEEtran.cls for IEEE Conferences

Michael Shell
School of Electrical and
Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332–0250
Email: http://www.michaelshell.org/contact.html

Homer Simpson
Twentieth Century Fox
Springfield, USA
Email: homer@thesimpsons.com

James Kirk
and Montgomery Scott
Starfleet Academy
San Francisco, California 96678–2391
Telephone: (800) 555–1212
Fax: (888) 555–1212

***Abstract*—The abstract goes here.**

I. Introduction

Most of the radio spectrum requires a license to operate on. Devices emitting Radio Frequency (RF) energy without a license are limited to using one of the unlicensed radio bands, known as the Industrial, Scientific and Medical (ISM) radio bands. The most well-known of these bands is the 100MHz wide band centered around 2.45GHz, often called the 2.4GHz band. Many consumer devices such as wireless mice and keyboards, bluetooth headphones and Radio Controlled (RC) cars communicate in the 2.4GHz band. Furthermore most IEEE 802.11 (Wi-Fi) routers operate in the 2.4GHz band, along with the 5GHz band (which actually ranges from 5.725GHz to 5.875GHz). These ISM bands were originally intended and still used for devices emitting RF energy for purposes other than communication. The most common such application is the microwave oven. All this leads to high interference in these bands. In this paper, we investigate how bad this 2.4GHz and 5GHz interference actually is in the city of Antwerp.

To measure this interference we use the City of Things (CoT) network, operated by research institute imec. This network consists of hundreds of sensors and wireless gateways positioned around the city of Antwerp. On TODO (20?) of these gateways, we sampled activity on the 2.4GHz and 5GHz bands every 10 minutes for TODO days. The main motivation of this paper was in fact to investigate whether interference around the gateways is significant enough to pose a problem for the network. Next to the general magnitude of the interference, we look for patterns in daily and weekly variation of the interference.

Some citing [**?**]

mds
August 26, 2015

II. Measuring Interference

All used nodes have a COMPEX WLE900VX-7A network adapter, containing a Qualcomm Atheros QCA9880 wireless chipset. This 802.11ac chipset, along with all other 802.11ac and 802.11n chips by Qualcomm Atheros support a mode called *spectral scan*. In this mode the chip can scan the activity on all frequencies it supports. This activity is not limited to 802.11 traffic, but is influenced by any received signal. In this mode, the chip acts as a simple spectrum analyzer. The Free and Open-Source Software (FOSS) ath9k and ath10k drivers, respectively for the 802.11n and 802.11ac Qualcomm Atheros chipsets, support this mode. Each scan is performed over a period of $4\,\mu s$. Every scanned channel is divided into 64, 128 or 256 equally wide bins. For the full channel the noise floor and Received Signal Strength Indicator (RSSI) are reported, along with a magnitude for each bin. This magnitude incidates how the power within the channel was divided across the bins. We provide an overview of how to interpret these values in TODO.

A. Settings

The ath10k version of spectral scan offers fewer options than the ath9k version. For instance the option to increase the scan time from $4\,\mu s$ to $2044\,\mu s$ is absent in ath10k. The only missing option relevant to our case is the *chanscan* mode of spectral scan. When this mode is triggered the driver gathers a configurable number of samples for every channel it supports. We emulate this behaviour using the *background* mode, which continuously scans the chipset's currently configured channel as long as it is not sending or receiving. We combine this with an *iw scan* command, which listens for access point beacons for a provided list of frequencies. TODO PASSIVE When providing *iw scan* with a list of all supported frequencies, this closely emulates the *chanscan* behaviour.

B. Interpreting spectral scans

Before going into the interpretation of this specific data, we provide a quick overview of the dBm unit, often used to express power ratios. The regular decibel (dB) unit is a dimensionless unit. In contrast dBm expresses absolute power in reference to watt. 0dBm is defined as 1 milliwatt (mW). For each increase of 3dBm, the power in mW doubles, and for each increase of 10dBm, the power is multiplied by 10. Milliwatt-to-dBm conversion is calculated using the formula

$10log_{10}(\frac{x}{1\,\mathrm{mW}})$ with x the power in mW. A negative dBm power simply means the power is less than $1\,\mathrm{mW}$. By using dBm, very small or large power values are often avoided. As power attenuates quadratically with distance, useful power values can vary dramatically.

For each channel scan, the driver reports the noise floor and RSSI value. The noise floor value is hard-coded for every channel and ranges between -95dBm and -106dBm. This value is the expected power of all noise in a channel combined. The chipset can only receive a singal whose power exceeds this noise floor. The RSSI value is a (unitless) integer indicating, as the name implies, the received power in the channel. While the existence of this value is part of the 802.11 specification, its calculation is not. As a result every manufacturer calculates the RSSI differently. Atheros chips follow a rather simple formula: subtracting the noise floor from the current signal strength in dBm results in the RSSI. These two values are enough to estimate the signal strength in a channel: a noise floor of -96dBm and an RSSI of 40 indicate a received signal of -56dBm.

The per-bin magnitudes indicate how power is divided within the channel. To understand these values, some background about radio signaling is needed. Signals in radio communication are usually created using a technique called *IQ modulation*. With this technique, the final signal is a combination of two separate signals (the in-phase or I and the quadrature or Q signals) whose phases are exactly 90 degrees apart. By changing only the amplitude of the I and Q signals, the resulting signal's amplitude and phase can attain any value. The reported bin magnitude $b(i)$ for bin i is the sum of the absolute values of the I and Q signals' magnitudes. This magnitude in turn is the amplitude squared. The per-bin power scales quadratically with this magnitude value: if bin i has twice the magnitude of bin j, bin i's share of the channel's power is four times as large as bin j's. We can compute a coefficient $c(i)$ for every bin $i \in [1, n]$ as follows: $c(i) = \frac{b(i)^2}{\sum\limits_{j=1}^{n} b(j)^2}$. Note that every $c(i) \in [0, 1]$ and $\sum\limits_{c=1}^{n} c(i) = 1$.

As the dBm power values computed from the noise floor and RSSI are not linear but rather logarithmic, we cannot simply multiply by $c(i)$ to get each bin's power value. Instead we first convert $c(i)$ to a logarithmic value: $c_{log}(i) = 10log_{10}(c(i))$. As $log(a * b) = log(a) + log(b)$, we add this coefficient to the total logarithmic power to calculate each share's power: $nf + RSSI + c_{log}(i)$. Noting that $log(a/b) = log(a) - log(b)$, writing the formula using only the parameters returned by the spectral scan results in $nf + RSSI + 10log_{10}(b(i)^2) - 10log_{10}(\sum\limits_{j=1}^{n} b(j)^2)$. The earliest reference to this formula being used with the FOSS Atheros drivers was in an email on the ath9k-devel mailing list by Zefir Kurtisi (https://www.spinics.net/lists/linux-wireless/msg101011.html TODO cite), in reply to the Request For Comments (RFC) on spectral scan support by Simon Wunderlich. As far as we know neither this formula nor the calculation of its parameters were ever confirmed to be correct by Qualcomm Atheros. This remains an educated guess by the community.

C. *Subsection Heading Here*

Subsection text here.

1) Subsubsection Heading Here: Subsubsection text here.

III. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...