6.829 Final Project: A comparison of bit-rate selection algorithms

Colleen Josephson
MIT

Pavel Panchekha
MIT

Abstract

In this paper we compare the performance SampleRate and Minstrel, two popular bit rate selection algorithms that have widespread real-world usage on the Linux operating system. We use a novel trace-based approach that allows us avoid in-depth kernel programming and write simple Python scripts instead. We also introduce improvements to the Minstrel algorithm that allow for significant gains in throughput.

1 Introduction

One of the key ways that wireless networks differ from wired is that wireless networks have varying link rates. Link conditions vary with time due to interference from other devices, changing network geometry, and mobile clients. An optimal rate at some time t may be different from the optimal rate just 30 seconds earlier. A good bit rate selection algorithm has to detect and adapt to these conditions. If the chosen bit rate is too slow, then the throughput will be unnecessarily low. If the rate is too high, then failures will be very frequent and throughput will again suffer.

There are three main classes of bit rate adaptiation protocols: frame-based, SNR-based, and cross-layer protocols. Frame-based protocols measure the fraction of sucessfully received protocols. SNR protocols make decisions based on the estimated Signal to Noise ratio. Cross-layer protocols use SoftPHY data from the physical layer. The most commonly implemented protocols on today's networks are frame based, because SNR protocols perform poorly [CITATION HERE], and cross-layer protocols cannot be deployed on current networks because they violate the network layering abstration.

The two most popular fram-based protocols are SampleRate and Minstrel. Both were implemented in the MadWifi drivers for Linux wirless, and today Minstrel is the default bit-rate selection algorithm for all wireless drivers on Linux. Our research used a trace-based approach to analyze the performance of these two algorithms. We also created a modified version of Minstrel that provides significant throughput gains over the vanilla Minstrel implementation.

This paper will provide a general overview of SampleRate and Minstrel. We then discuss our testing methodology. We compare the performance of the two algorithms, and then discuss modifications we made to Minstrel and compare it's performance to the current kernel implementation.

2 SampleRate

Sample all the rates [1, 3, 2].

3 Minstrel

Derek Smithies, Ph-fucking-D.

4 Methodology

We did some things using some methods.

4.1 Motivation

MadWifi is old and sucks plus we'd need to buy hardware off e-bay. The current Atheros drivers use the linux default, Minstrel. There is no kernel implementation of SampleRate and porting it woile have been extremely annoying (possibly impossible) given the time constraints. Etc. Etc. Blah Blah.

4.2 Trace Collection

Yeee haw.

4.3 Testing framework

Harness.py, etc.

5 Analysis

Well, it's getting boring isn't it. This is the last subsection before we wrap it up.

5.1 Improvements to Minstrel

Minproved.

6 Future Work

802.11n, SampleRate with EWMA, take data using broadcast approach, submitting minstrel changes as kernel patch?

7 Conclusion

The end.

References

[1] Bicket, J. Bit-rate selection in wireless networks. Master's thesis, MIT, 2005.

[2] Smithies, D. Minstrel rate control algorithm for mac80211.

[3] Winstein, K., Sivaraman, A., and Balakrishnan, H. Stochastic forecasts achieve high throughput and low delay over cellular networks. In *Proceedings of the Tenth USENIX Symposium on Networked Systems Design and Implementation (NSDI 2013)* (Lombard, Ill, April 2013).

8 Acknowledgements

Jonathan Perry, Hari Balikrishnan, and Derek Smithies.