Troubleshooting WiFi in Dense Scenarios

Muhammad Inamullah
Department of CSE
IIT Bombay
Mumbai, India

Bhaskaran Raman
Department of CSE
IIT Bombay
Mumbai, India

***Abstract*— "How to provide a better download experience in WiFi in Large Audience Environments (LAEs)?" To enhance the download experience in LAEs, we present our current work and a proposal to design a troubleshooter that captures the picture of happenings in the wireless system from AP-side, and client-side when possible, and tries to indicate any wired side problems.**

I. Introduction

Higher speed and better connectivity, possibly at lower cost, are what we want when we download. And we eagerly seek such a "better download experience" when we connect to a WiFi network. However, that experience becomes bitter when there are problems with WiFi.

We propose to provide a better download experience in WiFi in Large Audience Environments (LAEs) – a rather general target. Specifically, we need to be able to troubleshoot the WiFi network whenever some problem occurs. The WiFi networks suffer from various issues including interference from WiFi and other sources, low signal strength, high overhead, and local contention. In addition, in LAEs, each end-device generates uplink traffic plus management and control data that contend with download traffic. These issues not only degrade performance, but due to being distributed across network, also make it hard to troubleshoot the WiFi.

There have been prior works on troubleshooting, including, Antler, WiFiProfiler, Client-Conduit, Wise, and Witals. These address a subset of the problems the WiFi networks face. Some are AP centric but lack in a client-side view of the problem, while others depend on client-side modifications. Moreover, many don't say anything when problems are on the wired-side.

In order to troubleshoot effectively, we need to identify the root cause of the problem, and for that, we need a solution that gives a comprehensive picture from AP, and when possible from clients as well, of happenings in the wireless system, and that goes further decidedly telling about wired side problems.

In the following we describe issues we witnessed in WiFi, our trace analysis and results, and our future action-plan.

II. The Experiments and Analysis

Smart-phone based quizzes, where students in classrooms downloaded, took and submitted the quiz from their smart phones over WiFi, showed several problems: Students were unable to connect to WiFi, connections slowed down, quiz file downloading took excessively long time, and submissions failed etc. To investigate further, we captured traces for these sessions.

We analyzed the WiFi-traces with several aspects like finding airtime-shares of different frame types, determining traffic share of each AP, classifying the traffic according to MAC, IP and other protocols, and obtaining number of retries and various delays like DHCP allocation delay etc.

III. The Results

Some of the problems we identified through above analysis are: multiple APs operating in the same channel, faulty APs, clients sending unnecessarily large number of probe requests to particular AP, a large number (more than 80%) of mobiles operating in 802.11b mode, and some APs sending large numbers of RTS/CTS frames. We also found some wired side issues like gateway unreachable and clients not getting IP address through DHCP.

An interesting anomaly we observed is that when we tried to firewall "outside" traffic passing through our AP, uplink traffic share for the outside traffic increased. Again, an in-depth trace analysis revealed that it was due to increased TCP SYN and SYN retries, when many clients frantically wanted to resume their TCP connections they had lost to the firewall.

IV. Our Action Plan

- The open-source ath9k driver gives ability to get values out of hardware registers on Atheros-based chips. These registers give values of time the AP is busy receiving, time the AP transmits, and the time the channel is sensed idle. We plan to utilize these values for our troubleshooting purpose.
- Rate adaptation algorithms lower the rate on every packet loss, which is a bad choice when the cause is collision. Thus we also need to inculcate differentiation of the cause of packet losses in our troubleshooting mechanism. Some prior works like MiRA and BLMon discerned the cause of losses just by getting patterns out of blockACKs. We plan to utilize hardware register values on the AP to enhance the loss differentiation of above works.
- On the client-side we propose to collect the channel conditions, like Channel Busy Time. The above mentioned registers can help collect these values on the clients too. We plan to send the collected channel condition to the AP (or to a centralized controller), possibly through cellular network in the case of client disconnection from WiFi.
- As the third and final component of our troubleshooter, we plan to detect possible wired side issues. These include DHCP and AP's IP address problems, gateway reachability, DNS issues, and Proxy settings.