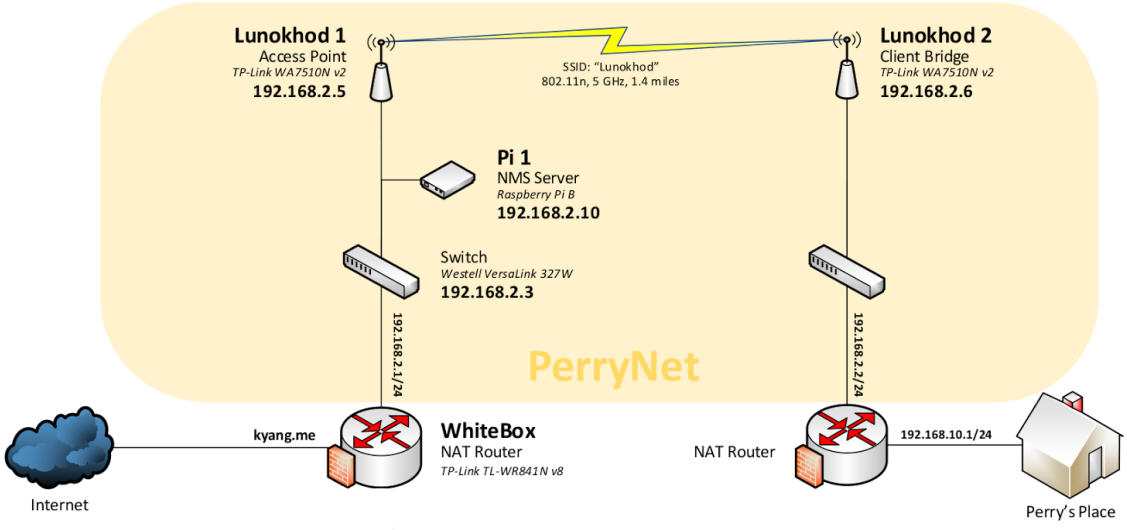
Wireless Backbone Link Monitoring System

Our Test WiFi Parabolic Reflector Antenna

Introduction

The Wireless Backbone Link Monitoring System was made as a project for the UC Riverside Spring 2016 CS183 class. The motivation behind this project is to create a functional system for monitoring the wireless backbone between members Kevin Yang and Kevin Yen's houses.

Tested Network Diagram



Monitored Statistics

Our system monitors several items that impact wireless backbone connections.

- Ping
  - The end-to-end packet transfer latency, usually expressed as milliseconds or thousandths $\left(\frac{1}{1000} \right)$ of seconds.
- Throughput
  - The integral sum of data transferred over time
  - $\int_{start}^{end} (Bits)/(Second)\ d(Second)$
- Total Data Transferred
  - The total data transferred
  - Taken as the last valid time value of the throughput

Group Info

Resources: GITHub

People:
Brandon Lu <b.k.lu@ieee.org>,
Christine Hawley <chawl002@ucr.edu>,
Jay Song <jsong022@ucr.edu>,
Kenneth Chan <kchan049@ucr.edu>,
Kevin Yang <kyang014@ucr.edu>,
Kevin Yen <KY.17364@gmail.com>,
Yohanan Arciniega <yarci001@ucr.edu>

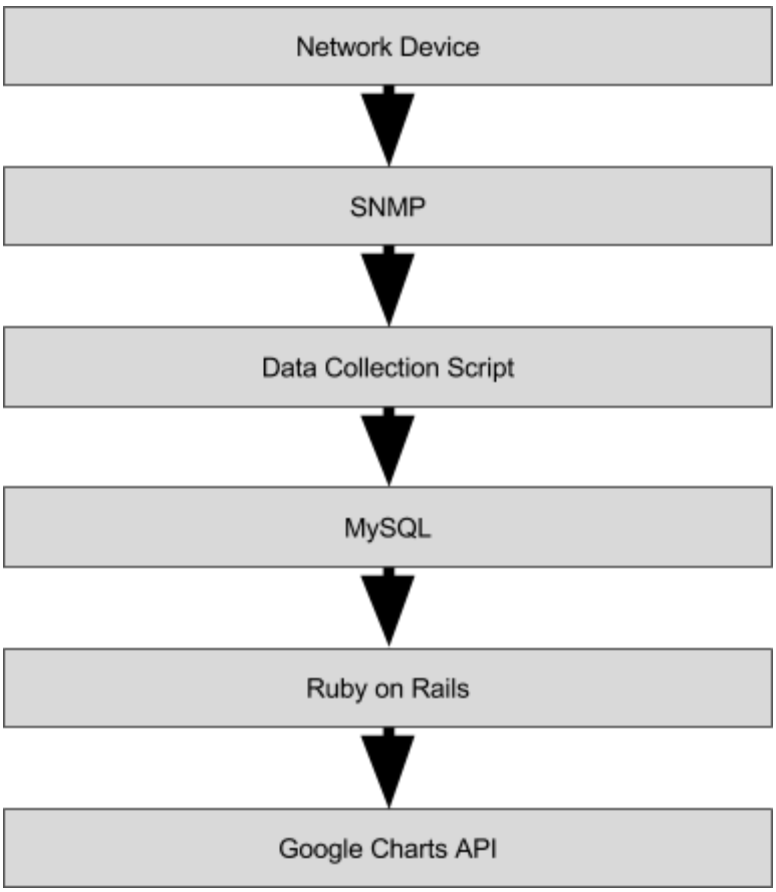
Technology Overview

The Wireless Backbone Link Monitoring System uses standard network protocols and will operate on any modern web browser.

- SNMP - Simple Network Management Protocol
  - Provides network information through known MIBs (management information bases) such as TCP/IP network packets, interface names, system load, *et cetera*.
- MySQL
  - Stores gathered network information accessible by a well-known (Structured Query Language) format to be retrieved and graphed.
- Perl / Putty / BASH
  - Used to put the gathered information from SNMP or other sources into the MySQL database backend.
- Ruby on Rails
  - Provides a simple web interface to view wireless backbone data.
- Google Chart API

- Google's chart generation application program interface used for displaying our graphs.

Information Dataflow



The **network devices** are monitored using the **SNMP** bindings provided by most devices using a **data collection script** which then places the data into the **MySQL** database backend which is polled by **Ruby on Rails** and formatted for the **Google Charts API** used on the **web interface**.