

QRAAT

March 22, 2013

Abstract

This could eventually be our doc for QRAAT. For now, this just outlines configuration stuff for the prototype system. Build this doc with `pdflatex` (`sudo apt-get install pdflatex`).

Contents

1 System specifications

QRAAT has two main components: the remote field computers, which implement the initial signal detection, and the server, which collects data from the field computers for higher order processing. Here we outline the structure of these two components.

1.1 RMG Remote

Each field computer is configured to run the software defined radio. These systems have 32-bit dual core processors, 2 GB of ram, and 8 GB solid-state persistent memory. They run stock Ubuntu Server 12.04. Along with the SDR implemented in GNU Radio, each is configured to run an ssh server, minicom, ntpdate, and the various other necessities.

The initial signal processing software outputs files corresponding to pulses with the extension `.det`. These are stored on a temporary file system in memory, located at `/tmp/ramdisk/det_files`. The pulse data files are stored in a directory structure according to the the minute in which they were recorded, for example:

```
/tmp/ramdisk/det_files/2013/03/22/15/20/03439743.det
```

The file name gives the second it was created with millisecond precision, i.e. `SSUUUUUU.det`. This is the last step on the field computers; these files are then collected by the server.

Each site has a user called *rmg* whose password—you guessed it—is *rmg*.

The QRAAT software is located in `/home/rmg/QRAAT`. We've set it up as a git repository that pulls from the RMG Server. To update the software, ssh to the site, switch to this directory, and type `'git pull'`. You will be prompted for the server's password.

Lastly, the transmitter file is `/home/rmg/tx.csv`.

1.2 RMG Server

Along with managing the field computers and collecting data, the server is responsible for calibration and triangulation. The `rmg` script allows us to power the computer and RMG module on and off, cycle the RMG module power, start and stop the software defined radio, update the field computer's transmitter file, and collect `.dets`. The server has a file called `sitelist.csv` that stores various parameters and the status of the RMG remotes:

1. *Name* - of the site,
2. *CompIP* - hostname or IP address of the remote computer,
3. *PowerIP* - hostname or IP address of the network addressable power source,
4. *CompOutlet* - outlet number of the computer,
5. *RxOutlet* - outlet number of the RMG receiver module,
6. *PowerType* - power source type, e.g. Netbooter, PingBrother, or WebPowerSwitch, and finally
7. *Status* - is the site up, down, or active (SDR is running).

`rmg` uses RSA encryption for ssh instead of prompting the user for a password each time. This is important since some of `rmg`'s routines, such as `rmg fetch`, make many ssh calls. The RMG remotes store the public part of the key and the server the private part.

We keep a copy of the software repository on the server from which the remote computers pull; the server copy pulls directly from github.

2 Configuration

Here are the configuration steps that need to be taken as of this writing (22 March).

2.1 West campus prototype

We won't have networking for this implementation. As they are the ethernet interfaces of the RMG remotes are configured with a static IP address 10.<site_no>.1.55 and netmask 255.0.0.0. To connect to it, the simplest thing to do is create an interface in network manager. *Network Manager* → *Edit Connections...*, in the *Wired* tab, choose *Add*. In *IPv4 Settings*, choose *Method* = *Manual*. Add an address like 10.253.1.55 with netmask 255.0.0.0. You should be good to go.

2.2 Quail Ridge

There are two configuration steps for each field computer. First, the ethernet interface needs to be configured to use the Quirinet router as its gateway. In `/etc/network/interfaces`, you'll find an interface that is commented out. Uncomment this and comment out the old one.

```
# Default interface in Quirinet
auto eth0
iface eth0 inet static
    address 10.20.1.55
    netmask 255.255.0.0
    gateway 10.20.1.1
```

The second part is to make sure the git repository is pointed to the right place. In `/home/rmg/QRAAT/.git/config`, you'll find a line that reads

```
url = christopher@10.253.1.55:work/QRAAT # change me!
```

Change this to the correct worker and directory and you're set!