# 1 Introduction

FunnelWeb runs on a computer which suts between the internet and a household's computers. It helps manage network usage. The program can enforce per-person daily and monthly usage limits, when the network is available, and (to a degree) even what sites are available. The primary aim, when we first started working on FunnelWeb, was that we'd never utter "Who used all the quota?" ever again. It's worked pretty well.

Since the FunnelWeb must sit between the modem which connects the household to the internet and the rest of the household, it usually requires a computer that can support two simultaneous network connections: one to the household network and one to the internet (see Figure 1). Since it is a program, it needs a computer to run on. ADSL modems often have inbuilt wireless routers; these routers will either need to be configured so that noone can connect to them or set up so that they only connect to the FunnelWeb host computer. All connections to the internet must be routed through the FunnelWeb host so that the program can do its job. Typically, this requires another router of some sort (ethernet or WiFi) connected to the FunnelWeb host which the other devices connect to.

graphic showing both ethernet and WiFi connections

Small wireless routers are readily available through online sellers and consumer electronics stores, and can be found for less than the price of a two movie tickets in Hobart.

The other critical ingredient is a computer running Linux; it acts as the bridge between the two sides and is able to control what goes across the bridge. We could have called the system *Troll*, but that already has a fairly entrenched meaning on the network, and we are just not that brave.

It may *seem* clumsy to insert another computer and wireless modem into the infrastructure, but this really is the only way we can control the flow of network traffic, and it has a few other benefits as well. This strategy makes it easy to add an extra layer of protection for the computers within your household: many network modems have the facility to act like a firewall, but so does the the computer running FunnelWeb; in principle, you could have two quite different layers of those sorts of defences. Since the FunnelWeb host is a computer, it can also do things like block particular sites (like Facebook) at predetermined times (like bedtime) on an individual basis.[1], run code that automatically updates blacklists, and generally provide an extra hurdle for villans trying to sneak into your network.

[1] When our teenagers weren't all teens, the younger ones had the network *open* for a while after school, completely *locked* during dinner and study-time (apart from "school" sites) and then *open* again before bedtime

Figure 1: infrastructure

Though being fluent, or even *familiar*, with LINUX isn't essential for most users, this is a technical document about a program run on a computer running LINUX. When we refer to files, they will be files located in the filesystem of the FUNNELWEB host. We also discuss things like configuration files, *ipaddresses*, *macaddresses*, and IPTABLES. The web-interface hides a lot of these details, but if any of these details are necessary, we describe how to get the information you need easily. We will also refer to Figures, Tables and things of , to makee

# 2 Setting up a household's network

From the point of view of FUNNELWEB, a household consists of *users*and *devices*. The starting point for defining the household's control scheme is to create *user*entries for everyone. Equipment that regularly polls external sites will also need to be treated as having a dedicated *user*. Later we'll discuss *control-groups*, which allow the administrator to control sets of *devices*all at once.

Control groups come later

A *user* specification would typically include an indication of what quotas ought to be allocated to the user (see section 2.1), the devices they use, and any administrative privileges they might have. There are a few important points to be aware of when naming users and devices: spaces and quotation marks *must* be avoided: a machine you might want named `Fezzic's laptop` should really be called something like `Fezzics-laptop`. The reason is that usernames and hostnames with embedded spaces are really hard for computer programs to recognise Spaces and quotation marks in computer names can also really mess up host name handling and webpage addresses.

Networked printers, like some of the printers that support "print-by email", and some other devices [2] may also need a *user* account (and quota) if they regularly use the network to update their status or to retrieve data. Leaving them without an associated *user* may either mean they cannot connect at all (by default) or that they slip through the rules completely unaccounted for.

Usually, the configuration (and, indeed, most of the management) of FUNNELWEB will be done through the web interface and all the changes to the configuration file will be done by the code it calls; to illustrate, an example configuration file which defines a couple of users and the devices they use is

Should we have a program that converts a ".ini" file to a valid config file? The net-control script can look and see which is newer to decide if it needs to run the converter before loading the config file.

```
;;
;; Comments start with semicolons
;;

(default-access reject)
;; (default-access accept)


(user Leslie
  (option quota unrestricted) ;; Leslie's usage is not tracked at all
  (device andy)
```

[2] Networked TV's, networked vending machines, and toasters leap to mind.

```
  (device max)
  (device peecee))

(user Michael
  (option quota dynamic) ;; defaults to one share
  (device monet)
  (device chopin))

(user Alice
  (option quota dynamic) ;; defaults to one share
  (device js)
  (device pdq)
  (device frederic))

(user Bob
  (option quota dynamic) ;; defaults to one share
  (device snare)
  (device bass)
  (device tom))

(user Cassie
  (option quota dynamic 0.25) ;; one quarter of a share of the option quota dynamic allocation
  (option quota static 400M)
  (option quota monthly 2G)
  (device centauri)
  (device sirius)
  (device procyon))

(user Pierre-Auguste         ;; It likes to pick up "email" jobs from the manufacturer's
                             ;; site, so it may use the network
  (option quota dynamic 0.1) ;; only one tenth of a share of a day's option quota dynamic allocation
  (device renoir))

(user guest
  (option quota dynamic)
  ;; devices will be added by login
  )


(group parents Leslie Michael)
(group offspring Alice Bob Cassie)

(group phones andy monet pdq tom  centauri)
(group laptops max chopin js bass snare procyon)
(group bigiron sirius frederic peecee)

;; end of config file
```

The numbers used to describe the dynamic quota allocations are actually the proportions of the total to be allocated. This means that if we look at the

pools for dynamic, static (daily) and monthly allocations, say 15G, 5G and 10G, then the number of shares available for dynamic allocation would be 4.35 and the amount of dynamic quota Cassie would get would be $0.5G*0.35/4.35 = 28.7Mb$. The general rule is that you can use actual quantities of quota or proportions of total available, but do not mix the two!

## 2.1 How allocations work

The starting point for controlling the network traffic is usually to set a monthly "total quota".[3] Unless your ISP provides a service with unlimited quota, this will usually be the same as the quota associated with your allocation cycle (we'll call it a "monthly" quota for convenience), though there are situations where people on monthly cycles may want to control usage for other reasons (such as parents, or people in share houses).

The software allocates a certain quota (megabytes or gigabytes of upload and download) to each individual at regular intervals. This quota can be comprised of fixed amounts or amounts which are dynamically calculated and based on how much of monthly household quota remains.

Allocating a fixed amount every day means that we can work out exactly how much of the total quota we need to set aside for each individual for daily use. We can also set aside a fixed amount for each of the users which is available over the whole month; this lets us accomodate strategies which give each person a normal daily usage plus extra for large downloads or streaming that would otherwise obliterate a day's allocation.

Alternatively, we might consider a strategy where each user is given a proportion of whatever remains of the household quota at the beginning of each day – sharing the remains of last nights dessert, as it were). Usually this would be calculated pro-rata, so that the amount of traffic allowed is more or less even across the month. If a family goes away for the weekend, the quota isn't lost, it just means that the rest of the month has a bigger pool of quota to allocate from than usual.

## 2.2 A share house example

Suppose we consider a share house with three people and a total monthly quota of 120Gb which costs $120. Suppose one of the people is a student who can only afford $20/month, and each of the other two can both cover $50 each. Table 1 give some example allocations

First Scenario

Each persion is allocated a quota which is proportional to their contribution.

[3] This isn't strictly speaking necessary for all the feature of FunnelWeb, but it does make describing quotas easier.

| Scenario 1 | | | | |
|---|---|---|---|---|
| Name | Pays | $Fixed_M$ | $Fixed_D$ | *Dynamic* |
| James | \$50 | 50Gb | 0 | 0 |
| Kath | \$50 | 50GB | 0 | 0 |
| Barry | \$20 | 20Gb | 0 | 0 |

| Scenario 2 | | | | |
|---|---|---|---|---|
| Name | Pays | $Fixed_M$ | $Fixed_D$ | *Dynamic* |
| James | \$50 | 10Gb | 0 | 667Mb |
| Kath | \$50 | 30Gb | 0 | 667Mb |
| Barry | \$20 | 0 | 900Mb | 10Mb |

| Scenario 3 | | | | |
|---|---|---|---|---|
| Name | Pays | $Fixed_M$ | $Fixed_D$ | *Dynamic* |
| James | \$50 | 2Gb | 267Mb | 566Mb |
| Kath | \$50 | 30Gb | 67Mb | 566Mb |
| Barry | \$20 | 0 | 900Mb | 566Mb |

Table 1: Allocation strategies

Second Scenario

Westly goes through many, many papers and books a month (10Gb covers it), and is also happy to work in with Kath for the rest.

Kath has business interests that usually involve lots of data transfers, but is only occasionally interested in other things, and is happy to let the rest be spread amongst the others.

Barry finds that he's run out after the second week, so shifts to a daily fixed quota plus the balance of his contribution as dynamic quota.

Third Scenario

They find that they are consistently under-using their quotas, and that Barry, could really use the extra quota for study.

James ensures that he gets enough quota for his papers by setting a daily fixed quota with the unused portion jackpotting into the dynamic pool

Kath safeguards enough for his daily emails and, like James, she'll jackpot the rest into the dynamic pool

Remember that the total available quota for an individual at any time is the sum of whatever is left of their monthly allocation, whatever is left of their daily fixed allocation and whatever is left of their dynamic allocation.

The way the dynamic allocation works is that relative size of each of the figures in the column is used to calculate how much of the available pool should be allocated to the individual. In the third scenario, the values are all equal so each would get the same amount allocated from the dynamic pool. In the second scenario, however, while Kath and James get the same amount, they get nearly seventy times more allocated than Barry. So in scenario 3, at the crack of dawn on the first day of the month Barry might use 2833Mb during the day, and this would leave him with (nominally) 833Mb/day for the rest of the month. If Barry's housemates didn't use all of their dynamic quota, this number would increase over the course of the month.

FunnelWeb can be configured to either take quota off the fixed daily amount first and then the dynamic daily quota, or the other way around. This makes a difference to where any quota actually goes. FunnelWeb takes quota off the fixed daily quotas first, then any unused portions of the dynamic quotas return to the dynamic pool for reallocation the next day. When the dynamic quotas are used up first, any remaining fixed quotas go into the monthly quota of the user. In either case, a user's monthly quota is only decreased when all of the daily options have been used.

# 3 Controlling FunnelWeb

Users can be given permission to perform various administrative tasks, such as the ability to lock and unlock access to the network, give their quota to other users, add "bonus" quota or to do things like add new users or devices. All of these administrative tasks can be done either through a web interface, or using the command-line tools running on the FunnelWeb host. The web pages and forms are designed to make managing and using the network simple. The command-line tool may be more convenient for someone that uses linux regularly and is more appropriate for using in scripts or atypical tasks. This section will present walk-throughs for several common tasks in both systems

- Adding a new device
- Adding a new user
- Adding a device to a user's profile
- Claiming and releasing a device (like an networked games console).
- Setting/changing a user's quota

### 3.0.1 Web interface

If Funnelweb is running, you should be able to point your browser at the Funnelweb host, let's call it 'Oscar' and connect to the web interface with the address

```
http://oscar
```

or

```
http://oscar:30407
```

if you are running another webserver on the Funnelweb host.

When you connect you ought to be presented with something like:

```
16:30                                          25/07/2016


Our current usage is 154.04Gb out of 250 Gb,
compared with the average of 201.61 Gb


    User: --------------------
    Password: ----------------
```

This page will let you log in and present you with a menu of options. The menu you see depends on what permissions you have been granted when your Funnelweb account was created. One of the common uses will be to associate a "public" device, like a game console, with a particular user.

### 3.0.2 Command-line interface

# 4 Technical overview

FunnelWeb works by constructing a set of chains and rules in the iptables on the the *funnelweb-host*. These rules deal with packets coming into the network from outside (such as web pages, audio or video streams from media sites, or unwanted ssh probes), and outbound packets coming from household computers (such as connections to wikipedia.org, or data sent out from the game server you run). Typically one might choose to DROP or REJECT suspicious incoming ssh packets, for example, while traffic to and from wikipedia.org or from an "unmetered" site might be exempt from normal quota considerations. Quotas are enforced by using *quota rules* which record the volume of traffic going in either direction between the two interfaces (one of which goes to the ADSL modem, the other to the household network). Once the quota has been reached, the traffic is rejected.

The iptables is built so that traffic from each machine may be associated with particular users. Machines which are used by multiple people, like general computers or games consoles, can either be locked out until the user "logs in" in a manner similar to the kiosk access in hotels, or have its own quota. This

Table 2: Significant files, and what they do.

| | |
|---|---|
| /etc/net-control/conf.scm | Defines the users, devices, and sets quotas |
| /var/log/net-control/remainder | records total quota remaining |
| ? | records the user's usage |
| /funnelweb | The source code for the system |

strategy works equally well for computers, ipods, mobile phones, and games consoles.

AS is usual with ADSL modems, the household network is run through the NAT (network address translation) filter in the IPTABLES, so all the traffic will appear to the outside world as coming from either the *funnelweb-host*or more usually the ADSL modem. One consequence is that you can put rather a large number of machines on your network before you have to worry about running out of addresses, but particularly bold users should be aware FUNNELWEB has not been used or tested with IPV6.

# 5 Important files

summary of the files and locations

## 5.1 conf.scm

*format spec, refer to example above*

## 5.2 Saved quota state files

format spec

## 5.3 Source code

description of the source code files and their roles

### 5.3.1 `net-control.py`

detailed discussion

# 6 errata

Proposed syntax

```
;;
;; Comments started with semicolons
;;
```