

25 minutes allocated. 15 minutes for talk, 10 minutes for questions.

## └─What's the problem?

- ⚡ Many people use metered internet connections
- ⚡ Cannot download everything they want to, when they want to
- ⚡ Downloads have to be prioritised

Many people, especially in non-western countries, use metered internet connections where they get charged for the bandwidth they use.

This means they cannot download everything they want to, when they want to; and often have to be quite careful and in-control of what they download each month.

In other words, their downloads have to be prioritised, especially if those downloads are quite large. For example, operating system and app updates.

As they are part of our target market, Endless has been quite interested in making it easier for people with such internet connections to download what they want, without causing huge bandwidth bills.

It's worth noting that what we are discussing here only applies to 'big' downloads, and is a voluntary service for software components to use — this is not a network firewall or enforced queuing system.

## └─What is metering?

- ⚡ Pay per unit bandwidth used
- ⚡ Pricing structures vary a lot
- ⚡ Could have unmetered periods of the day
- ⚡ Could be non-net-neutral

A little more detail on what metered connections are. They are connections where you pay per unit of bandwidth used, rather than paying a fixed price per month for unlimited bandwidth.

However, pricing structures vary a lot. Some tariffs may allow unlimited downloads at a certain time of day (for example, early in the morning, for OS updates); or they may always allow downloads from certain websites for free (this is not net-neutral). Each network provider may have several tariffs, and the tariffs may change regularly (with promotions, for example).

## └ Does it affect me?

- Home/Work internet connection may be unlimited
- Trains, public Wi-Fi, mobile internet often aren't
- Main internet connection may be bandwidth constrained, even if unlimited
- Downloads can be prioritised

This kind of thing affects western countries too, although in less pervasive ways. While it's typical to have an unlimited home or work internet connection, metered connections are encountered on trains (in the UK), public Wi-Fi hotspots (commonly in airports, where you pay for an hour's internet access), or with many mobile internet tariffs.

There are other ways a connection can benefit from prioritising downloads: if the connection is slow, the number of parallel downloads should be limited, and the queue of downloads prioritised to complete important ones first. You might want certain downloads to not be started until you're back home on a fast connection.

## └─ Introducing Mogwai

- Mogwai is a download scheduler
- It does downloads after midnight (hence the name)
  - It's also a daemon, which ties in with the Cantonese meaning
- Applications ask it when to start 'big' downloads
- It schedules all the requests and notifies the applications
- Uses various configuration, including tariff files

So we wrote Mogwai. It's a download scheduler: applications which want to do 'big' downloads tell the scheduler what they want to download (including its size, domain name, etc.), the scheduler schedules all the requests, and signals each application when it's time to start downloading. The scheduler can later pre-empt a download and ask the application to pause downloading for a bit, too.

The scheduling decisions are based on a tariff file for each network connection, which says what period of day that connection is metered, what period of day it's unmetered, and some recurrence rules for when those periods recur. A connection may be metered all the time, or unmetered all the time. The tariff format is extensible, so other properties could be added in future (for example, downloads on a certain domain are unmetered during this period).

## └─ Introducing Mogwai

- Mogwai is a download scheduler
- It does downloads after midnight (hence the name)
  - It's also a daemon, which ties in with the Cantonese meaning
- Applications ask it when to start 'big' downloads
- It schedules all the requests and notifies the applications
- Uses various configuration, including tariff files

Other preferences (stored in NetworkManager on each network connection) are taken into account, such as whether to always allow downloads on that connection.

Mogwai is implemented as a D-Bus service on the system bus, with a client library and some helper utilities provided.

## └─What Mogwai is not

- Not security — it's an opt-in system
- Not currently monitoring bandwidth usage

As mentioned earlier, Mogwai does not intend to provide security. It cannot stop applications doing big downloads whenever they like — the applications have to opt in to querying Mogwai about whether to download, and can ignore what Mogwai tells them.

It doesn't (currently) monitor bandwidth usage either, so its calculations of how much bandwidth has been used during a particular tariff period is entirely blind. If a tariff period is limited to 1GB of downloads, Mogwai has to base its scheduling decisions on the total amount of bandwidth requested by applications, and assume that only a nominal amount of bandwidth is used otherwise. Clearly, improvement can be made here.

## └─ How do I use it?

- # `libmogwai-schedule-client`
- # See source code for `mogwai-schedule-client` utility
- # Or the 'Automatic Updates' panel in `gnome-control-center`

Mogwai is currently used in Endless OS, in our fork of `gnome-software`, and in our OS updater (`eos-updater`). We have changes to `gnome-control-center` to add it to the 'automatic updates' panel. If you want to use it, use the API from `libmogwai-schedule-client`, which is a client for the D-Bus API exported by the Mogwai daemon. There's an example in the code for `mogwai-schedule-client`, which is a simple download client shipped with Mogwai. See the PDF of these slides for a link; they'll be available afterwards.

Essentially, your application needs to create a schedule entry for each download, and tie the state of the download to signals from the schedule entry telling it when to start, pause, and resume downloading.

## └ Future work — available for mentoring

- Monitor bandwidth usage to close the feedback loop
- Provide download statistics to GNOME Usage
- Account for more factors when scheduling downloads

What's left to do? Lots. The next big task on the list is to add a monitoring daemon to Mogwai, which would monitor bandwidth usage per app, and store statistics on usage based on this. Those statistics could be used in GNOME Usage, and could be used by the scheduler to work out when the bandwidth limit for the current tariff period has been reached.

I have a plan for how this can be implemented efficiently, using nftables and the systemd unit hierarchy.

The scheduler can always be made more clever, taking more factors into account from the system, and from the schedule entries provided by applications.

I'm willing to mentor people to implement this functionality in Mogwai, since my time on it at work has been cut.