

Shncpd
The Simple, Stupid, Slow HNCP Daemon
An HNCP Implementation Report

Juliusz Chroboczek
University of Paris-Diderot

22 July 2015

How it happened

In November 2007, I decided to explain **routing** to myself:

- I started implementing **RIPng**;
- two months later, I had designed and implemented **Babel version 0**.

How it happened

In November 2007, I decided to explain **routing** to myself:

- I started implementing **RIPng**;
- two months later, I had designed and implemented **Babel version 0**.

In June 2015, I decided to explain **DNCP** to myself:

- I started implementing **DNCP** and just enough **HNCP** to get *hnetd* to speak to me;
- three weeks later, I had implemented a **useful subset of HNCP(+DNCP+PA)+RA+DHCPv4**.

It was **a lot of fun**, and I have **plenty of things to say**.
But I only have **10 minutes**.
Please ask **questions at the end** (if any).

Hnetd

Hnetd is the standard implementation of HNCP. Hnetd is highly polished and well integrated into OpenWRT. It is indistinguishable from magic.

Quick howto:

- install OpenWRT;
- `opkg install hnet-full`;
- edit `/etc/config/network`;
- `/etc/init.d/network restart`.

Highly polished and modular:

- depends on `odhcpd`, `udhcpd`, `pcpproxy`, etc.
- all of this stuff interacts over a combination of `ubus` (JSON over Unix sockets) and shell scripts;
- highly modular code written by professionals.

SHNCPD (1)

Shncpd is the **simple, stupid and slow** implementation of HNCP. Quick howto:

```
$ (cd babeld && make && sudo make install)
$ (cd shncpd && make && sudo make install)
# babeld -C 'redistribute proto 43 allow' \
          eth0 wlan0 &
# shncpd eth0 wlan0 &
```

Shncpd implements **HNCP** (DNCP, PA), **RA** server, **DHCPv4** server in a single process:

- **single event loop** (no threads);
- **simple and stupid data structures** (linear search);
- **no fancy callbacks** (no attempt at modularity).

SHNCPD (2)

Unlike hnetd, `shncpd` implements as many protocols as reasonable within a **single binary**, in a **single event loop**.

The **only dependency** is `babeld`. No changes were required to `babeld` — the normal **redistribution** mechanisms are good enough.

Important lesson: just because they are **separate protocols** doesn't mean that you need to implement them in **separate daemons**. **Single process** with **shared data structures** also works.

Shncpd status

Status:

- interoperates with *hnetd*;
- complies with *dncp-06* with one exception;
- should comply with *pa-07* (needs checking);
- currently no claims about HNCIP compliance.

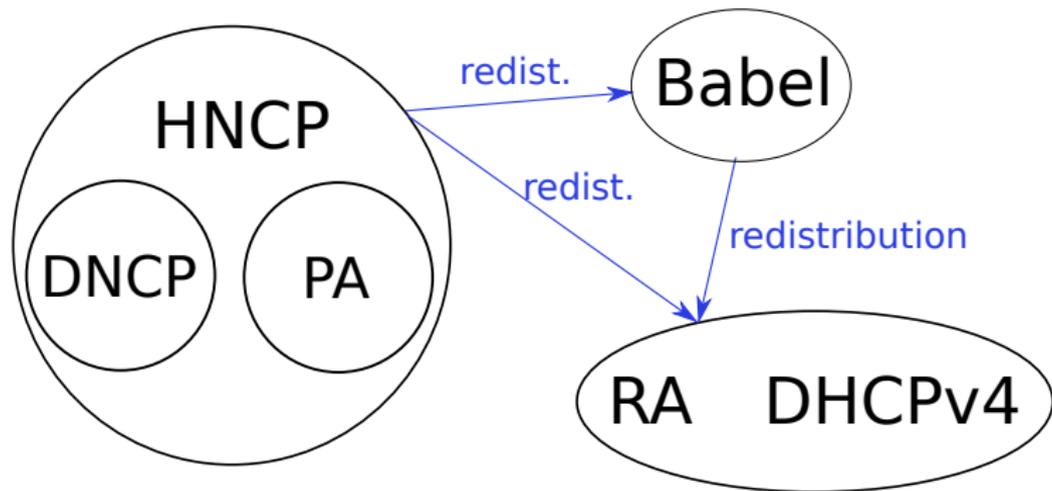
	Lines of code	Binary kB
shncpd	4 000	40
babeld	10 000	100
total	14 000	140

Shncpd does not implement:

- edge detection (MUST with loophole);
- DHCPv6-PD client;
- DNS-SD proxying.

Protocol walkthrough

How the pieces fit together



- DNCP and PA are triggered by various events;
- interaction between the protocols is redistribution;
- redistribution into DHCPv4 feels weird.

DNCP

DNCP contains both exciting and boring bits:

- flooding algorithm (look, Ma, no timeouts!);
- TLV format.

The protocol became clear once I understood the following:

- DNCP's Trickle is not what I think of as Trickle (much more subtle);
- DNCP's keepalives are not what I think of as keepalives (somewhat more subtle).

Prefix Assignment

PA describes how the set of prefixes assigned to a link are negotiated.

Very general algorithm:

- ability to veto a prefix;
- ability to use different prefix lengths (important for IPv4);
- ability to statically configure a prefix for a link.

The proof of convergence has not been published.

The algorithm became clear once I understood the following:

- “best prefix” is a function, but “assigned prefix” is hard state (a variable).

HNCP

HNCP contains the “boring bits”:

- a lot of different TLVs;
- a lot of MUST.

The protocol became clear once I understood the following:

- MUST means “We are Homenet, and we require that you do that”, it doesn’t mean “If you don’t, the protocol will break”.

Conclusion (1)

Shncpd is a **from scratch** reimplementation of a **useful subset** of the Homenet protocol stack.

A useful subset of the Homenet stack can be implemented from scratch in finite time:

- Markus Stenberg implemented **Babel in 2 nights**;
- Juliusz Chroboczek implemented **HNCP(+DNCP+PA)+RA+DHCPv4 in 3 (long) week-ends**.

This was possible because these are **well-defined**, mostly **self-contained** specifications.

Conclusion (2)

I am **very happy** with the subset of HNCP+DNCP+PA that shncpd implements:

- elegant and **correct algorithms**;
- well-defined, **accurate protocol specifications**;
- I only requested **minor protocol changes** (**Markus and Steven agreed**);
- while some clarifications and loosening would be welcome, **the specifications appear to be complete**.

(I hold no opinion on the bits that I haven't implemented.)

Let's move forward!