Deployment experiences with HNCP (the feel good talk)

Juliusz Chroboczek, joint work with Dorine Chagnon
IRIF
Université Paris-Diderot (Paris 7)

16 July 2016

Babel

Background 1: HNCP

HNCP is the address configuration protocol of the Homenet protocol suite.

HNCP is designed to configure unmanaged, small, stable, prefix-based networks.

Thesis:

- HNCP (the protocol) supports use cases beyond Homenet;
- HNCP must eventually support use cases beyond Homenet (or be extended to do so).

Important note

This talk is about pushing HNCP beyond what it was designed to do.

I will speak about limitations of hnetd. These limitations do not apply to the Homenet use case.

Use case 1: partially managed networks

HNCP was designed for unmanaged networks.

Networks configured by HNCP may turn out to be partially managed :

- a given link must have a specific prefix (think DMZ);
- a given host must have a specific address (think web server);
- a given link must be used for last-resort only (think monetary cost).

While this is out of scope for Homenet, it will be requested.

Use case 2: mesh networks

HNCP was designed for networks composed of stable links that are assigned prefixes (Internet-style).

Parts of networks configured by HNCP may turn out to be wireless meshes:

- flat routing (/128);
- persistently lossy links.

While this is out of scope for Homenet, it will be requested.

Let’s try it out

HNCP most probably supports use cases beyond Homenet.

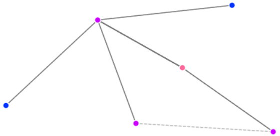
What about the implementations?

> *Ideas are tested by experiment. That is the core of science. Everything else is bookkeeping.*
> *— Zombie Feynman*

Let’s build a partially managed wireless mesh network configured by HNCP.

Background 2: the Babel Towers mesh network

We have a permanent testbed:
a mesh network composed of 6 to
15 routers:



- double-stack;
- links of varying quality (some very lossy);
- routing uses Babel;
- address configuration done by AHCP (don't ask);
- IPv4 edge router uses DHCPv4 + NAT;
- IPv6 edge router uses RIPng.

Plan: retire AHCP, replace it with HNCP. See what breaks.

The good (1)

Dorine Chagnon volunteered to spend a month of her summer holidays upgrading all of our routers and deploying HNCP.

Conclusions:

- give a first year student a router, tell them to install a tftp client on their laptop, and three days later it runs OpenWRT trunk, Babel and hnetd;
- at that point, you see the student running three instances of tcpdump; onlookers are impressed;
- two weeks later, the whole network is running Babel+HNCP.

Note: this study was done with a sample of size 1. We're *computer* scientists after all, not real scientists.

The good (2)

Hnetd works.

The good (2)

Hnetd works.

It has no right to:

- we have some persistently lossy links
  - HNCP is not designed to work over those;
- HNCP synchronises a lot of state;
  - and there is no fragmentation mechanism;
- we only have a single /62 to play with
  - hnetd assigns /80, uses stateful DHCPv6;
  - Android and Blackberry fall back to IPv4 (please implement DHCPv6, or SLAAC in /80).

A lot of state

```
22:22:07.040949 IP6 (hlim 64, next-header UDP (17) payload length: 1388)
    fe80::e046:9aff:fe4e:912e.8231 > fe80::6123:e33b:1dd3:c0be.8231:
    [udp sum ok] hncp (1380)
        Node endpoint (12) NID: 4a:bd:6f:79 EPID: 00000001
        Network state (12) hash: 7ef5adad143cf6fe
        Node state (24) NID: 02:ff:1e:ce seqno: 1844 563.77s hash: 1c9133a594b19865
        Node state (24) NID: 4a:bd:6f:79 seqno: 425 268.65s hash: 0fc682d83bf6c9a1
        Node state (24) NID: 75:05:c4:15 seqno: 67022 8.32s hash: e55b1a59267c9a28
        Node state (24) NID: 7b:ac:d0:4c seqno: 3 168.60s hash: 122f0256087259cf
        Node state (24) NID: 90:d1:74:28 seqno: 5716 575.60s hash: ff114b3525db5096
        Node state (24) NID: 9b:18:b7:26 seqno: 16670 251.40s hash: 33905a9c7c773c8b
        Node state (24) NID: ef:36:04:89 seqno: 34959 169.66s hash: b752659da2bb9ef5
        Node state (1188) NID: 75:05:c4:15 seqno: 67022 8.32s hash: e55b1a59267c9a28
                Peer (x6), HNCP-Version, Assigned-Prefix (x7), Node-Address (x12),
                DNS-Delegated-Zone (x11), Node-Name
```

- This is a unicast exchange;
- there are 7 *short node state* TLVs, one for each router in the network;
- a single *long node state* TLV has been piggybacked onto the same packet (1188 octets!).

(Tcpdump support by Antonin Décimo and Jean-Raphaël Gaglione.)

A lot of state: long node state TLV

```
Node state (1204) NID: ef:36:04:89 seqno: 34955 6.46s hash: a94bfa15b478c76d
    Peer (16) Peer-NID: 02:ff:1e:ce Peer-EPID: 00000005 Local-EPID: 00000003
    Peer (16) Peer-NID: 02:ff:1e:ce Peer-EPID: 00000006 Local-EPID: 00000004
    Peer (16) Peer-NID: 75:05:c4:15 Peer-EPID: 00000002 Local-EPID: 00000003
    Peer (16) Peer-NID: 90:d1:74:28 Peer-EPID: 00000003 Local-EPID: 00000004
    Peer (16) Peer-NID: 90:d1:74:28 Peer-EPID: 00000004 Local-EPID: 00000003
    Peer (16) Peer-NID: 9b:18:b7:26 Peer-EPID: 00000002 Local-EPID: 00000004
    Peer (16) Peer-NID: 9b:18:b7:26 Peer-EPID: 00000003 Local-EPID: 00000003
    HNCP-Version (22) M: 0 P: 4 H: 4 L: 4 User-agent: hnetd/cda52dc
    Assigned-Prefix (18) EPID: 00000001 Rsv: 0 Prty: 2 Prefix: t\0x05
    Assigned-Prefix (18) EPID: 00000002 Rsv: 0 Prty: 2 Prefix: t\0x05
    Assigned-Prefix (18) EPID: 00000004 Rsv: 0 Prty: 2 Prefix: t\0x05
    Assigned-Prefix (20) EPID: 00000001 Rsv: 0 Prty: 2 Prefix: t\0x05
    Assigned-Prefix (20) EPID: 00000002 Rsv: 0 Prty: 3 Prefix: t\0x05
    Assigned-Prefix (25) EPID: 00000001 Rsv: 0 Prty: 2 Prefix: ::/0
    Assigned-Prefix (25) EPID: 00000002 Rsv: 0 Prty: 2 Prefix: ::/0
    Node-Address (24) EPID: 00000001 IP Address: 10.219.152.5
    Node-Address (24) EPID: 00000001 IP Address: 2001:660:3301:9209:1e::5
    Node-Address (24) EPID: 00000001 IP Address: fd1f:f88c:e207:65::5
    Node-Address (24) EPID: 00000002 IP Address: 10.191.218.61
    Node-Address (24) EPID: 00000002 IP Address: 2001:660:3301:920b:9::3d
    Node-Address (24) EPID: 00000002 IP Address: fd1f:f88c:e207:47::3d
    Node-Address (24) EPID: 00000003 IP Address: 10.0.116.30
    Node-Address (24) EPID: 00000003 IP Address: 2001:660:3301:9209:9::1e
    Node-Address (24) EPID: 00000003 IP Address: fd1f:f88c:e207:dba3::1e
    Node-Address (24) EPID: 00000004 IP Address: 10.0.96.5
    Node-Address (24) EPID: 00000004 IP Address: 2001:660:3301:9208:79::5
    Node-Address (24) EPID: 00000004 IP Address: fd1f:f88c:e207:78::5
    DNS-Delegated-Zone (33) IP-Address: 2001:660:3301:9209:1e::5 lb- lan.r.home
    DNS-Delegated-Zone (35) IP-Address: 2001:660:3301:9208:79::5 lb- wlan1.r.home
    DNS-Delegated-Zone (35) IP-Address: 2001:660:3301:920b:9::3d lb- wlan0.r.home
    DNS-Delegated-Zone (36) IP-Address: 2001:660:3301:9209:9::1e --- wlan01.r.home
    DNS-Delegated-Zone (46) IP-Address: 2001:660:3301:9209:1e::5 --- 152.219.10.in-addr.arpa
    ...
```

The bad (1)

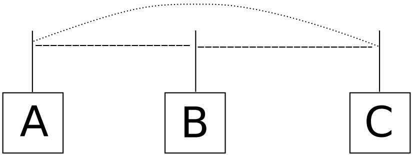
Although we’re pushing HNCP way beyond what it was designed to do,
I have almost nothing bad to say about HNCP.

The bad (1)

Although we're pushing HNCP way beyond what it was designed to do,
I have almost nothing bad to say about HNCP.

We witness refloodings of node state every few dozen seconds. They do not cause renumbering.

Persistently lossy links cause yo-yo neighbour associations (up-down-up-down).



We probably need hysteresis in link sensing.

The bad (2)

Although we're pushing HNCP way beyond what it was designed to do, I have almost nothing bad to say about HNCP.

But hnetd (the implementation) has some limitations:

- bugs;
- undocumented but necessary features;
- missing features.

Bugs

We found a number of minor bugs in hnetd.
The developers are responsive, even when on vacation.

Bugs are a fact of life, but at least under OpenWRT, hnetd is difficult to debug:

- incomprehensible logging;
- scripts fail silently;
- undocumented tools;
- OpenWRT is not friendly to debugging.

We need better tools.

(We're working on it.)

Undocumented features

Our IPv6 edge router speaks RIPng to the outside world.

In order to announce the prefix into HNCP, we had to set up a static route in OpenWRT:

```
config interface 'wan6'
    option ifname 'eth1'
    option proto 'static'
    option ip6addr '2001:660:3301:9202::ac17:248b/64'
    option ip6gw '2001:660:3301:9202::ac17:2ffe'
    option ip6prefix '2001:660:3301:9208::/62'
    option delegate 0
```

Option `delegate` is completely undocumented (and we don't know what it means).

Missing features

Hnetd is missing some important features:

- no way to announce a prefix without setting up a static route with hard-wired gateway address;
- no way to make a static prefix assignment (DMZ link);
- no way to make a static address assignment (for servers).

Supported by HNCP.

These are limitations of hnetd or the way it is integrated into OpenWRT.

Conclusion

HNCP works surprisingly well:

- in lossy mesh networks (but harmless reflooding);
- in partially managed networks.

Hnetd has some limitations, let's fix them.

Further work:

- add Bird to the network (Bird speaks Babel);
- add shncpd to the network;
- build debugging tools:
  - tcpdump (done);
  - wireshark (partly done);
  - visualisation à la Babelweb (in progress).