

# Home Networking Control Protocol



---

draft-ietf-homenet-hncp-01

Steven Barth & Markus Stenberg

(see <https://github.com/fingon/ietf-drafts> for draft sources & individual diffs)

# Changes from non-WG draft (-00)

Version-TLV: defined as 32-bit number + user-agent string

Node data is distributed regardless of matching version number however only node data with matching version number is actively evaluated.

Opaque User-Agent string for debugging purposes.

Router-Address TLV: added Link-ID value to TLV-data

Link-ID is added to announce the specific link to which a router address is assigned to. This is equivalent to Link-ID for the Assigned Prefix-TLV.

Service Discovery TLVs: changed requirement level to MAY

SD TLV support was "MUST" or "SHOULD" -> SD is now optional

# Refining Border Discovery (-01)

Motivation: reasonable security without cryptography

Prevent malicious nodes from injecting fake uplinks

Prevent compromised ISPs from claiming to be part of the homenet

Solution: interfaces SHOULD be configurable as fixed external or internal

## 1. Fixed External Mode

Never run HNCP or IGP here, always do stateful firewalling

## 2. Fixed Internal Mode

Never accept an uplink here, MAY support more fine grained categories...

# Fine grained internal categories (-01)

## 1. Guest (trust noone, provide internet access)

don't run HNCP / IGP, never accept uplinks, restrict / disable apps like SD  
disallow traffic between guest <-> internal (only allow from/to uplinks)

## 2. Ad-Hoc (allow meshed networks)

run HNCP / IGP, never accept uplinks, modify app behavior like PA  
indicate links on the interface are potentially non-transitive

## 3. Leaf (proposed for -02, trust only clients)

don't run HNCP / IGP, never accept uplinks

make this the default for non-ethernet interfaces (e.g. WiFi)?

# Compatibility with legacy routers (-01)

## Considered Usecases

Legacy router behind homenet

Homenet behind legacy router

## SHOULD provide network access to legacy routers

Offer sub-delegated prefixes via DHCPv6-PD on internal interfaces and announce them via HNCP on the legacy router's behalf

## MAY implement Hybrid Interface Category

fixed-internal, run HNCP / IGP, accept uplinks without changing to external  
allow sharing a link between homenet and trusted legacy routers

# Reference Implementation Updates

Synchronized implementation with draft -01

OpenWrt implementation

Added web interface support

Experimental support for MAP-E/T  
and LW4over6 IPv4-uplinks

Lots of bugfixes...

The screenshot displays the OpenWrt web interface for configuring network interfaces. At the top, there are tabs for 'H1' and 'LAN'. The main heading is 'Interfaces - H1', followed by a descriptive paragraph: 'On this page you can configure the network interfaces. You can bridge several interfaces by ticking the "bridge interfaces" field and enter the names of several network interfaces separated by spaces. You can also use VLAN notation INTERFACE.VLANNR (e.g.: eth0.1).

Below this is the 'Common Configuration' section, which includes sub-tabs for 'General Setup', 'Advanced Settings', 'Physical Settings', and 'Firewall Settings'. The 'Advanced Settings' tab is active, showing the configuration for the 'eth2' interface. The 'Status' section indicates the interface is up, with a 'Uptime' of 0h 0m 0s, a 'MAC-Address' of 08:00:27:C2:30:CB, and traffic statistics: 'RX: 8.46 KB (58 Pkts.)' and 'TX: 124.69 KB (374 Pkts.)'. The 'Protocol' is set to 'Automatic Homenet (HNCP)'. The 'Category' dropdown menu is open, showing options: 'Automatic' (selected), 'External', 'Guest', and 'Ad-Hoc'. At the bottom right, there are three buttons: 'Save & Apply', 'Save', and 'Reset'.

# One More Thing

JSON-Export  
for monitoring

Interactive status  
WebUI page

Demo:

<http://youtu.be/jJ-nPmXcOEM>

The screenshot shows a web browser window displaying the hnet interface. The browser's address bar shows the URL `10.126.132.7/cgi-bin/luci/;stok=84283aa5ec36005de7ec11b5809f002d/z`. The page title is "hnet" and the navigation menu includes "Status", "System", "Network", and "Logout".

The main content area is titled "Click on a node of the graph to view detailed information." and displays a JSON object representing a node's configuration:

```
{
  "face-id": 2,
  "router-id": "71511c1dac8de8344eda5f7970eed9d2",
  "addresses": [
    "10.126.148.17",
    "2001:470:c974:21a2:4c60:deff:feef:b04c"
  ],
  "prefixes": [
    {
      "prefix": "2001:470:c974:21a2::/64",
      "fabricPrefix": false
    }
  ]
}
```

Below the JSON export is a network graph showing five nodes labeled hnet1, hnet2, hnet3, hnet4, and hnet5. Each node is represented by a green circle with four small colored dots (1, 2, 3, 4) around it. The nodes are connected by green lines. Node hnet2 is highlighted in light blue and has a callout box next to it containing the text "2001:470:c974:2100::/56" and "10.0.0.0/8".

# Thank You

Do you have questions or feedback?

We'll be around for more discussions or in case you are interested in trying out the implementation.

Please also visit [www.homewrt.org](http://www.homewrt.org) for source code, binaries and some documentation.