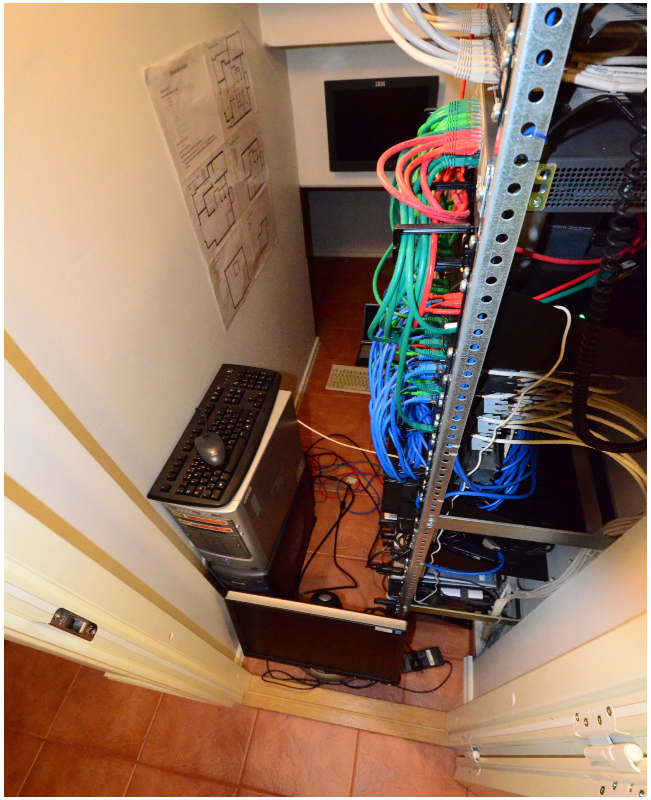
OSPFv3-Based Home Networking

draft-arkko-homenet-prefix-assignment-02.txt

Jari Arkko, Ericsson

Acee Lindem, Ericsson

Benjamin Paterson, Cisco



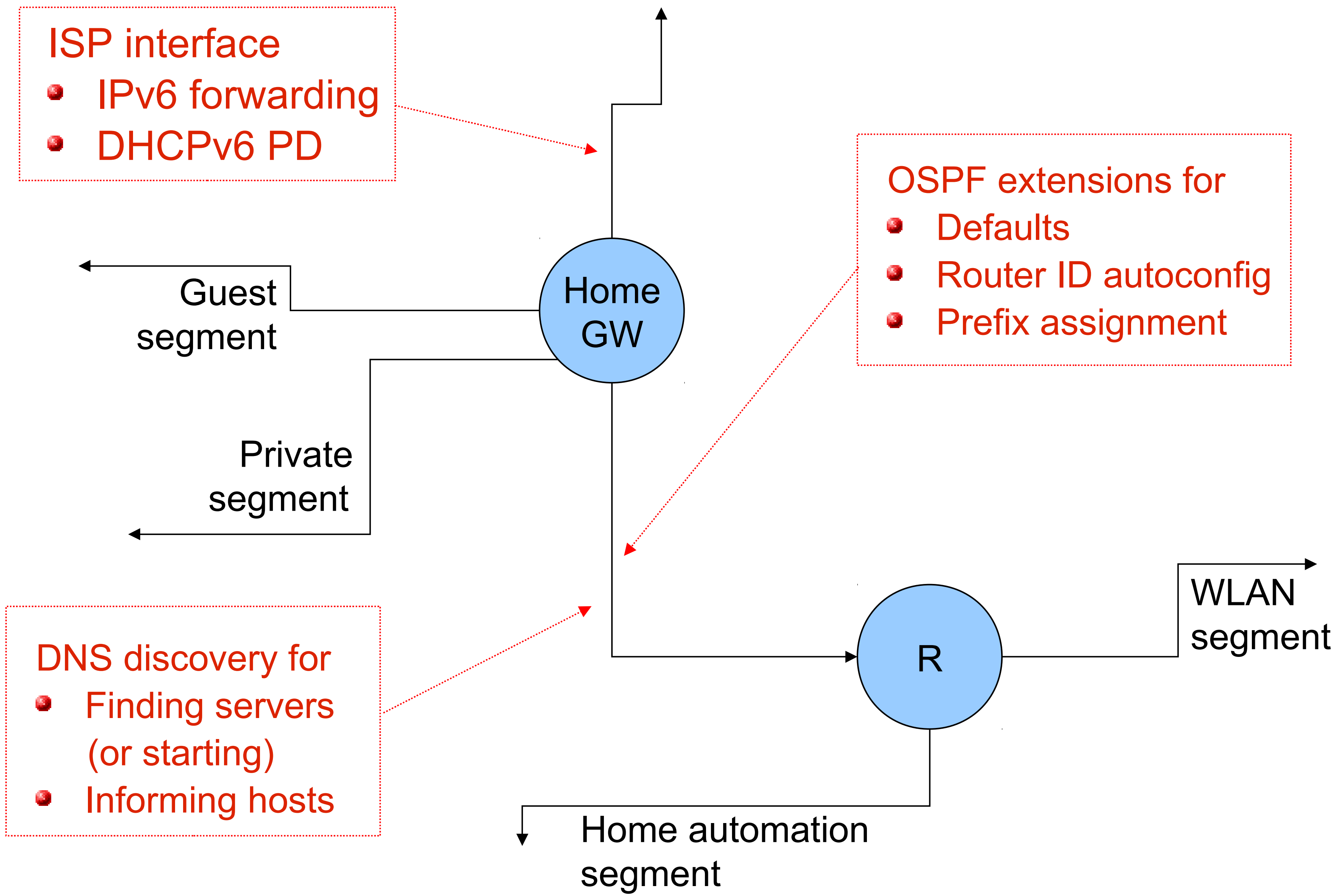
The Dream

No matter how many boxes you have
And how you connect them

- Networks shall have address space
- Routers shall know where to send packets
- Names resolve to addresses
- Human touch is not required

  [Especially by my mother!]

OSPF-Based Home Networking



Implementing and Using HOMENET

Status Report

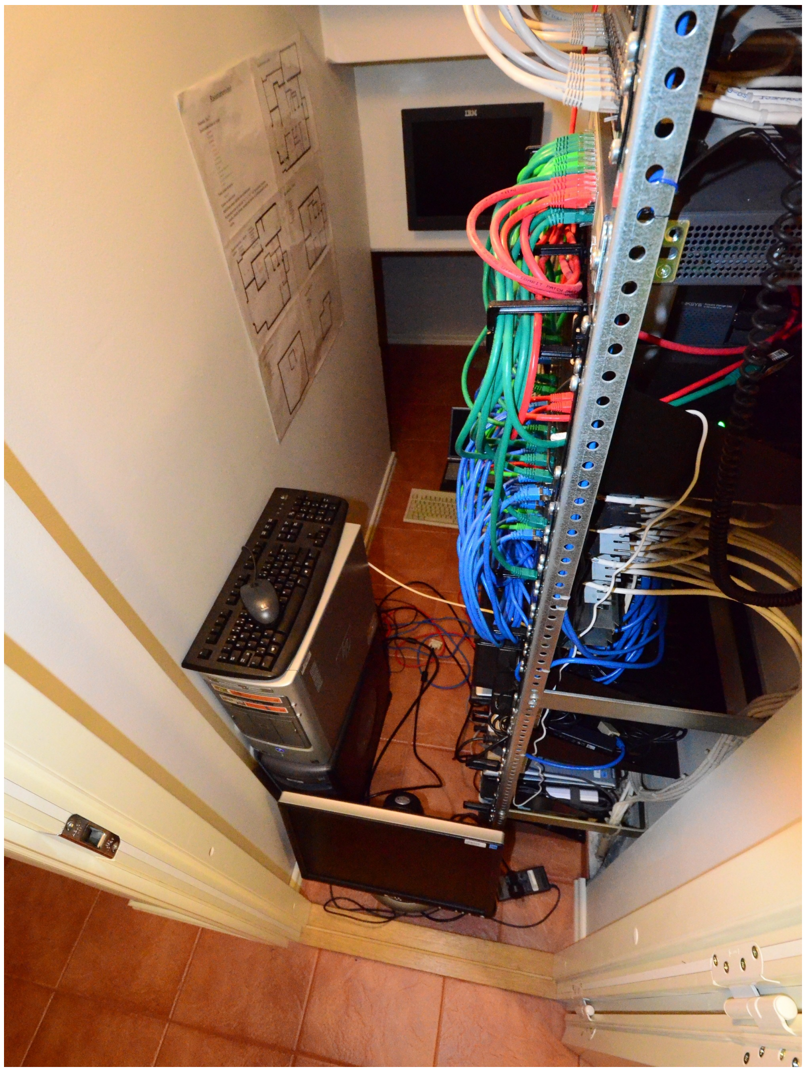
Two implementations up and running!

Generally, seems to work well

But implementations are early & incomplete

The protocol design is still morphing
(but that was the point of these exercises)

A HOMENET Network



Router ID

Prefix

```
hord: debug: 21897, OSPF: Timeout causes a message resend
hord: debug: 21897, RAW: sendto destination fe80::20c:46ff:fe16:9c86
^C
root@newrouter:/tmp# cat /etc/hord/events
Selected own router ID: 16.191.119.86
Selected own hardware fingerprint: 16.191.119.86
Automatically assigned a prefix to an interface on interface eth1: 2001:db8:beef:ddd6::/64
Added a new neighbor on interface eth1: 49.66.233.220
Received a valid DD message from neighbor with sequence number on interface eth1: 49.66.233.220 1954
Neighbor moves to EXSTART state on interface eth1: 49.66.233.220
DD sequence number to a neighbor initialized on interface eth1: 1008170920
Tentatively selecting ourselves as the master for the neighbor on interface eth1: 49.66.233.220
New DD message sent with sequence number, in response to a sequence number on interface eth1: 100817
This router becomes a slave to the following peer on interface eth1: 49.66.233.220
Negotiation done, moving to state EXCHANGE with neighbor on interface eth1: 49.66.233.220
```

root@nat64: /tmp

Tiedosto Muokkaa Näytä Etsi Pääte Ohje

```
root@nat64:/tmp# host -t aaaa www.slashdot.org 2001:14b8:400:f3c:21a:9fff:fe0b:811
Using domain server:
Name: 2001:14b8:400:f3c:21a:9fff:fe0b:811
Address: 2001:14b8:400:f3c:21a:9fff:fe0b:811#53
Aliases:

www.slashdot.org has IPv6 address 2001:14b8:400:f3f::d822:b530
root@nat64:/tmp# cat /etc/nat64.conf
pref64 = 2001:14b8:400:f3f::/64
out_pref46 = 10.70.0.0/24 ;
ports = 40000-60000 ;

interface = ext:eth0:drop_enabled ;
    filter_prefixes = 10.70.0.0/24 ;
interface = int:nat64:drop_enabled ;
    filter_prefixes = 2001:14b8:400:f3f::/64 ;
root@nat64:/tmp# cat /etc/radvd.conf

interface eth3
{
  AdvSendAdvert on;
  MaxRtrAdvInterval 3;
  MinRtrAdvInterval 1;
  AdvIntervalOpt on;
  prefix 2001:14b8:400:f3c::/64
  {
  };
  RDNSS 2001:14b8:400:f3c:21a:9fff:fe0b:811
};
root@nat64:/tmp#
```

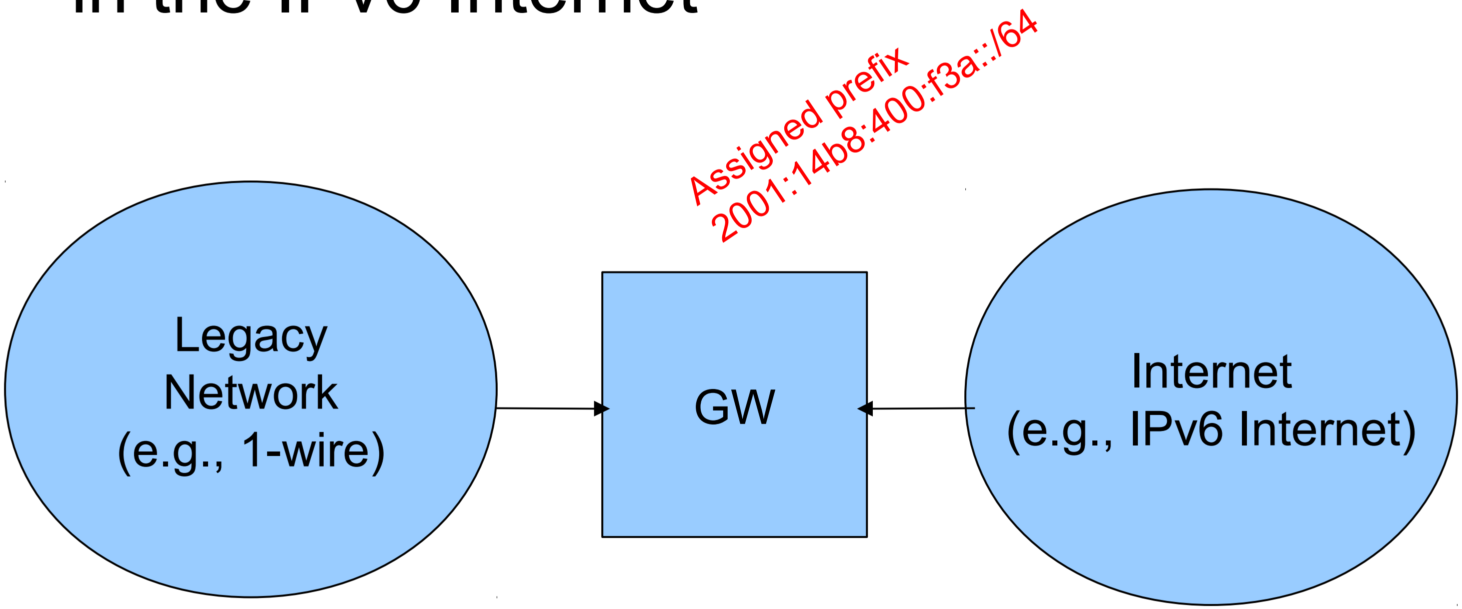
NAT64 config

RA & PIO

DNS discovery

Enabler for Other Things...

- E.g., automatic 1-wire to IPv6 translation GW; sensors are visible as CoAP servers in the IPv6 Internet



Feel free to try your CoAP client to, e.g., toaster.objez.net

activityroom
balcony
bedroommaster
bedroomjanne
bedroomolli
circuitbreakercabinerrk1b
communicationscloset
diningroomeast
diningroomwest
entrance
ethernetswitch
fileserverdisk
kitchen
konehuoneserver
livingroomeast
livingroomtvcorner
owhub
router
secondutilityroom
storageroom
storageroomneargarage
studyroom
technologyshaft2nd
technologyroom
technologyshaft
terrace
toaster
ventilationusedairout
ventilationfreshairtorooms
ventilationfreshairin
ventilationusedairback
weatherserver

Draft-02 Updates

- Added an algorithm to generate ULAs (S. 7)
- Replaced the old algorithm for prefix allocation with a new one (S. 6.3)
- Added an explicit discussion of hysteresis (S. 8)
- Added a requirement to support DNS discovery (S. 4.2)
- Described the design choices (S. 5)
- Added Benjamin as an author
- Various small bug fixes and editorial changes

Prefix Allocation Algorithm in Draft-02

- Assigns /64 prefixes out of an allocated prefix (e.g., /56, the ”usable prefix”)
- One or several routers in the network know the usable prefix(es), all routers co-operate to make the assignments
- The algorithm is triggered by changes in the LSA database or the set of interfaces this router has
- Benjamin's thesis demonstrates some properties relating to the algorithm (convergence, some aspects of correctness, ...)

Overview of the Algorithm

- Routers participate in the autoconfiguration protocol as defined in draft-acee-ospf-ospfv3-autoconfig and calculate their router IDs
- AC LSAs are flooded, with the Usable Prefix TLVs included by those routers who know such prefixes
- The algorithm is run for every pair <usable prefix, interface>
- Assignments are flooded in Assigned Prefix TLVs
- Hysteresis and stable storage applied for stability

Allocations in the Algorithm

- An allocation is made for an interface, unless there is already an allocation from someone else on the same link or when a neighbor has a higher routerID
- Conflicts can occur both on a link and across the network
- Upon detecting a conflict, the higher router ID ”wins” and the other withdraws its allocation

ULA Generation in Draft-02

- This router does NOT need to generate a ULA prefix if any of the following conditions are true:
  - There already is a usable prefix
  - A router with a higher ID is reachable
  - This router has a global address
  - … or can reach the IPv4 Internet
- If not, create a new /48 ULA per RFC 4193
- Apply the usual procedures on NVM, hysteresis, ...
- Some open questions remain

DNS Discovery in Draft-02

- The WG has so far focused mostly on the naming issue as an extension of, say, mDNS to work across subnets
- As I used an autoconfigured network, I realized that while this is useful, it may not be the highest priority task
- How does a host deep in the network resolve www.ietf.org or other name in the Internet?

DNS Discovery in Draft-02

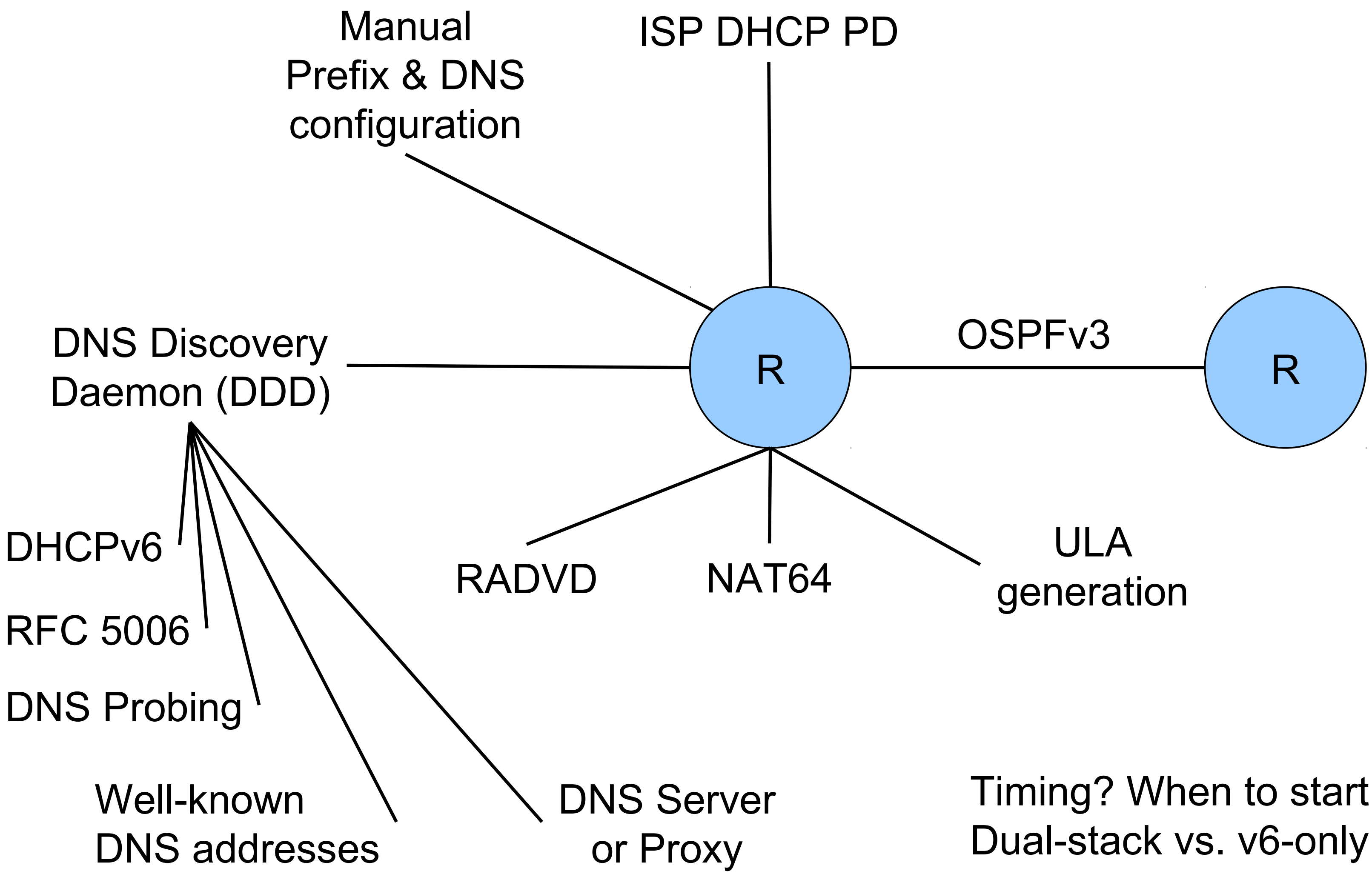
- A router needs to inform hosts within its networks about the addresses of DNS servers
- RECOMMENDED that homenet router supports
  - RA options (RFC 6106)
  - Stateless DHCP (RFC 3736)
- Leaves open where this information comes from
  - My implementation uses DDD to gather information from all possible sources
- But should the routers distribute this info?

Experiences

Here are some experiences:

- The technology seems to work as intended
- Our understanding of the problems developed as the work continued, e.g., on conflicts, naming, interfaces
- Relatively easy to implement, 2-4 KLOC as long as you are sane enough to not implement all of OSPFv3 from scratch...
- It is important to think about interfaces to other systems

Interactions with Other Parts



Timing? When to start a process?
Dual-stack vs. v6-only differences?
Firewalls and border detection?

Topics for Further Discussion 1

- Hysteresis, algorithms, ULA generation probably need more review & experience
- Interactions with other systems need to be described in greater detail
- If a router discovers a DNS server, how does it tell other routers about this find?
- Do we need a priority mechanism to decide allocations when there is not enough space?
  - Or even (shock!) a > 64 bit prefix solution?

Topics for Further Discussion 2

- Alternative designs for LSAs used by the algorithm:
  - Approach 1: TLVs within the AC LSA (draft-02)
  - Approach 2: Just use intra-area-prefix LSAs
  - Approach 3: A provisional assignment LSA followed by actual allocation LSAs (app. 1 or 2)
- There may be use cases for the draft's technology outside the home networks
  - ”RFC 3041 for prefixes”
  - Current thinking is that we should be able to deliver prefixes for any purpose