The Latest Developments in DHCPv6

RIPE66, Dublin, Ireland
May 2013

Tomek Mrugalski <tomasz@isc.org>

Agenda

1. About presenter and ISC
2. Client MAC address option
3. Load Balancing
4. DHCPv6 Failover
5. Provisioning IPv4 hosts in IPv6-only network
6. RADIUS option

Who is Tomek?

- M.Sc., Ph.D from Gdansk University of Technology
- Primary author of Dibbler
  - Portable DHCPv6 implementation (srv, cli, relay)
  - Supports Win 2k-Win8, Linux, BSD, Solaris
  - Confirmed use in 34 countries
- 7 years at Intel (Network Quality Labs, chipsets group)
- 2 years at ISC
  - Lead Developer of BIND10 DHCP (Kea)
  - Occasional contributor to ISC-DHCP
- Active IETF participant since 2009
  - DHC WG co-chair
  - 2 RFCs, 15+ drafts

What is ISC?

Internet Systems Consortium, Inc. (ISC) is a non-profit 501(c)(3) public benefit corporation dedicated to supporting the infrastructure of the universal connected self-organizing Internet - and the autonomy of its participants - by developing and maintaining core production quality software, protocols, and operations.

DHCP in IETF

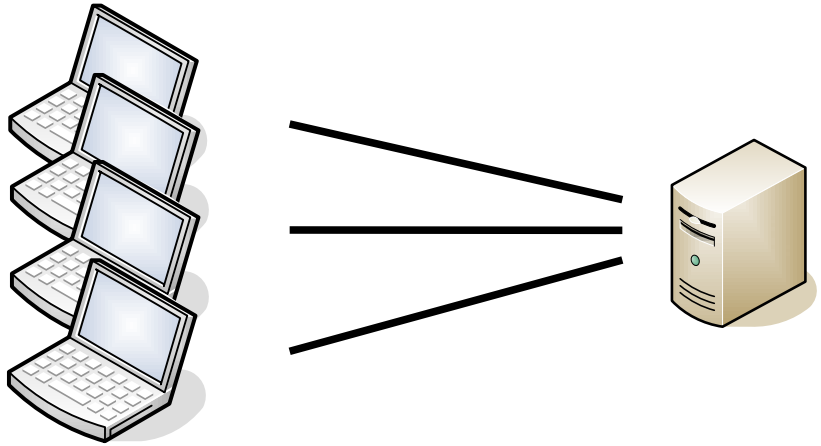
Active work in DHCPv4

MAC and DHCPv6 :: Overview

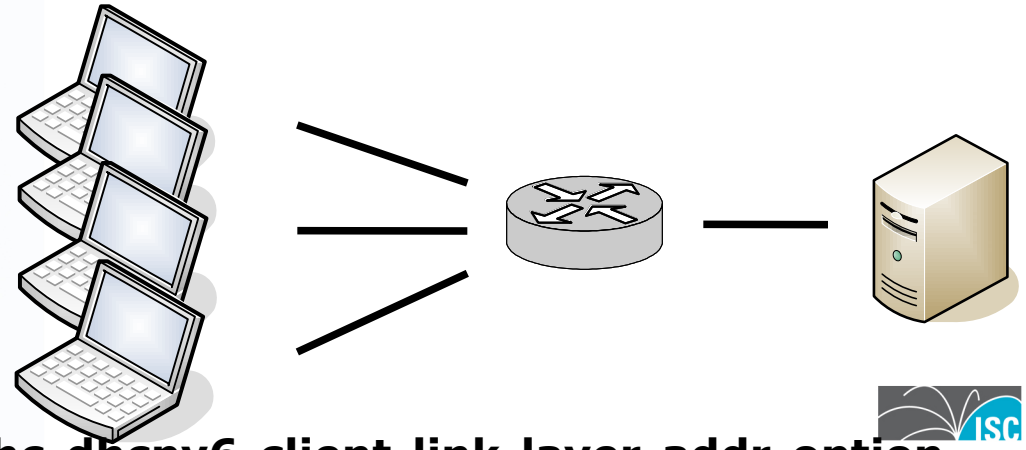
- **DHCPv4 had natural IPv4:MAC address mapping**
- **DHCPv6 is based on DUID concept**
  - Generated once, stored => more stable
  - 4 types: link-local+time, link-local, entreprise, uuid
- **DUID solves some issues...**
  - Change NICs => new DHCPv4 client
  - Some devices don't have fixed MACs
  - Cheap NICs can have the same MACs (or so they say)
- **DUID introduces new ones...**
  - Dual boot: Linux and Windows use different DUIDs
  - Reinstall OS: => new DUID
  - VM cloning => the same DUIDs
- **MAC was not used directly in DHCPv6 (until now)**

MAC and DHCPv6 :: Problem and solution

- **Problems:**
  - MAC not always available (clients behind relays)
  - Legal requirement to log IP:MAC mapping over time
- **Solution:**
  - Directly connected clients are easy to solve



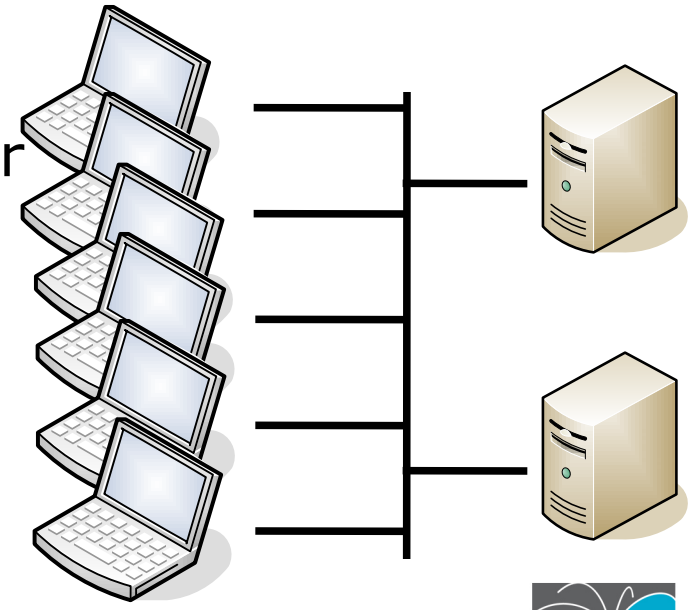
  - Relays insert client link-layer address option



RFC 6939 (RFC-Ed) draft-ietf-dhc-dhcpv6-client-link-layer-addr-option

DHCPv6 Load Balancing :: Overview

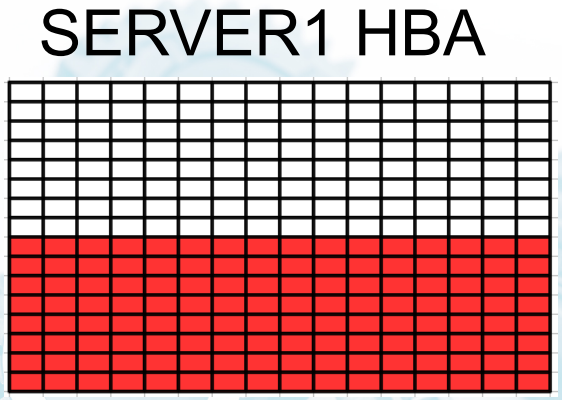
- **Problem space:**
  - So you want to have more than one server?
  - Your server is not beefy enough?
- **Preference option**
  - Different preference: Clients will always pick up the server with greater preference => 100% of traffic to one server
  - Equal preference: both servers must respond to SOLICT, client will discard one and pick the other => double* server load
- Solution: Load Balancing

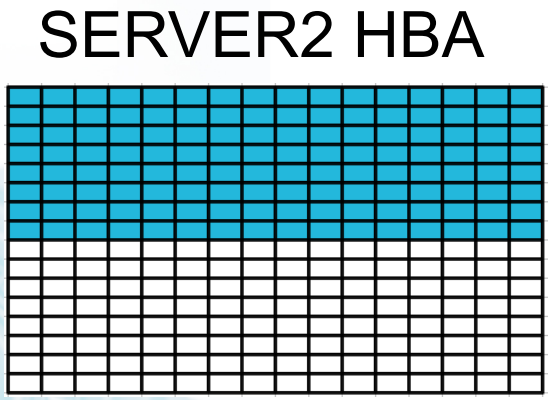


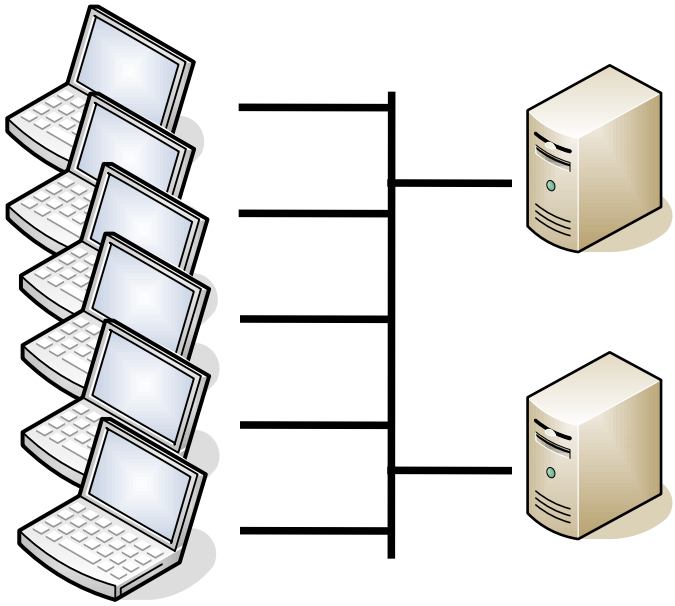
RFC 6939 (RFC-Ed) draft-ietf-dhc-dhcpv6-client-link-layer-addr-option

DHCPv6 Load Balancing :: Solution

- **Hash Bucket Assignments**
  - Server calculates hash, assigns packet to one of 256 buckets
  - If this server is configured to handle specific bucket, then processes packet; otherwise drops it
- **Good for:**
  - Load Balancing (2 or more servers)
- **Not suitable for:**
  - High Availability
  - Lease Stability
  - Redundancy







Myths about DHCPv6 Failover

- **I don't need it, there's so many IPv6 addresses**
  - Server change => address change => Lease stability?
  - Prefix Delegation – really that many?
- **Multi-master database will save me**
  - Get a subnet with 1 lease in it
  - Network split: 2 servers, each connected to its own DB master
  - Client A comes to server1, gets lease X
  - Client B comes to server2, gets lease X
  - Repeat this with 1000000 instead of 1 will only decrease probability
  - repeat with PD (added bonus: routing issues)
- **DAD will save me**
  - Will not work for Prefix Delegation
  - Will not work in non-broadcast environments

draft-ietf-dhc-dhcpv6-failover-requirements-04

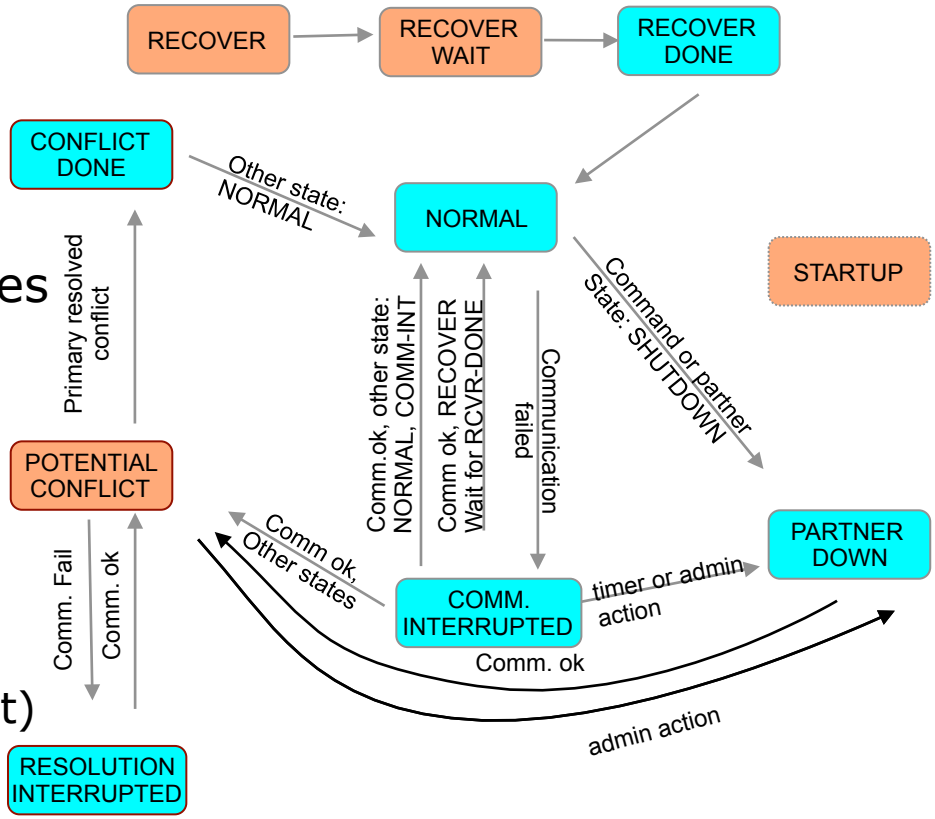
DHCPv6 Failover Grand Plan

- **Step 0:** Redundancy considerations
  - Published as RFC6853 (Feb. 2013)
- **Step 1:** Requirements document (info)
  - WGLC done, to be published soon
  - Comments welcome
- **Step 2:** Design document (std)
  - WG item, published -02
  - Text complete (no major missing parts)
  - Comments welcome
- **Step 3:** Protocol document (std)
  - TBD
- Possible extension drafts

draft-ietf-dhc-dhcpv6-failover-{requirements|design}

DHCPv6 Failover :: Overview

- **Based on v4 failover draft**, but simplified
- **Hot standby** (Active-passive only)
- **No load balancing** in design spec (likely extension)
- Recovery from:
  - **Server crash**
  - **Network partition**
- Major comcepts:
  - MCLT concept, Lazy Updates
  - state machines
  - Binding updates + conflict resolution
  - Connection management
  - 2 Allocation Algorithms (Proportional and Independent)
  - DDNS considerations
  - Lease reservation



draft-ietf-dhc-dhcpv6-failover-{requirements|design}

IPv4 provisioning in IPv6-only network

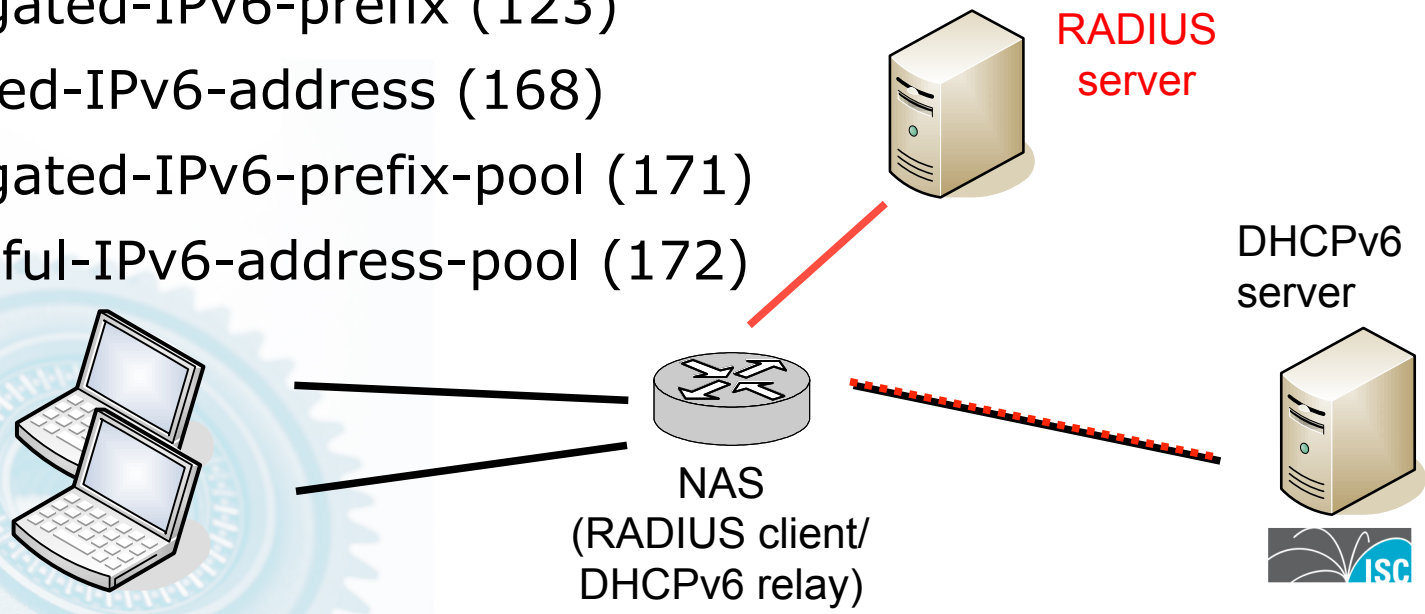
- **MAP** (Mapping Address and Port, DS-Lite successor)
  - Fully stateless (does not require per-session or per-subscriber state)
  - draft-ietf-softwire-map-dhcp
- **Lightweight 4over6**
  - draft-softwire-lw4over6
- **DHCPv4-over-DHCPv6**
  - draft-ietf-dhc-dhcpv4-over-dhcpv6

Attempts to unify/clarify:

- draft-dhc-dhc-v4configuration,
- draft-ietf-softwire-unified-cpe

RADIUS + DHCPv6

- **Very similar to DHCPv4 counterpart**
- **Access control done on NAS**
  - NAS asks RADIUS
  - RADIUS server responds
  - Negative: NAS drops the DHCP client request
  - Positive: NAS forward DHCP client request with include RADIUS attributes
  - Server may use RADIUS attributes
    - Delegated-IPv6-prefix (123)
    - Framed-IPv6-address (168)
    - Delegated-IPv6-prefix-pool (171)
    - Stateful-IPv6-address-pool (172)



DHCPv6 in IETF :: Other work

- **DHCPv6 Stateful Issues**
  - draft-ietf-dhc-dhcpv6-stateful-issues
  - RFC3315bis planned
- **Multiple Provisioning domains**
  - Whole Homenet WG
- **Routing configuation over DHCPv6**
  - draft-ietf-mif-dhcpv6-route-option
  - dying slowly...

What to do?

- **Speaking now is good**
- **Posting comments to DHC mailing list is better**
  - Subscribe https://www.ietf.org/mailman/listinfo/dhcwg
  - Post to dhcwg@ietf.org
  - Unsubscribe (optional)
- **Consider going to IETF meeting**
  - **IETF87: Berlin, July 28 – August 2, 2013**
  - **Helpful, but not required**

DHC working group homepage:

https://datatracker.ietf.org/wg/dhc/

Questions? Comments?

Thank you

isc.org

backup

Failover Design :: Communication

1. Communication over TCP
2. Reusing bulk leasequery framing, but with new FO-specific message types
3. TLS usage (optional)
4. Connection management (CONNECT, CONNECTACK, DISCONNECT)
5. State notifications
6. Lease updates (BNDUPD, BNDUPDALL, BNDACK, UPDDONE)
7. Pool requests (POOLREQ, POOLRESP)
8. Keep alive (CONTACT)

Failover Design :: Resource Allocation

1. Proportional allocation (“IPv4 failover-style”)
    1. Useful for limited resources (e.g. prefixes)
    2. Pool may need to be rebalanced.
    3. Only unleased resources are owned by specific server.
    4. Released/expired resources return to primary

2. Independent allocation (“simple split”)
    1. Useful for vast resources (e.g. /64 address pool)
    2. All resources are owned by specific server.
    3. Pools are never rebalanced.
    4. Released/expired resources return to its owner.
    5. Simpler, but MCLT restrictions still apply.

Failover Design :: MCLT concept & Lazy update

1. Lazy Update:
    1. Server assigns a lease and responds to a client
    2. Server updates its partner at a later time (lockstep would introduce too much delay)

    Problem: failure between 1. and 2.
2. Maximum Client Lead Time
    - The maximum difference between lease time known by a client and acknowledged by its partner.
3. Useful in communications-interrupted
    - Server does not know if its partner extended any lease;
    - It knows that its parter could extend by at most MCLT;
    - To be on the safe side, server assumes that ALL leases were extended by MCLT.