DHCPv6 Failover Update IETF83

Tomek Mrugalski <tomasz@isc.org>
2012-03-29

DHCPv6 Failover Grand Plan

- Step 0: Redundancy considerations
  - IESG Last Call in progress

- Step 1: Requirements document (info)
  - 1 comment received since Taipei (-00 WG)
  - Plan to publish -01 soon
  - Next step: ask for WGLC?

- **Step 2: Design document (info/std)**
  - Initial version + -01 published early March
  - Next step: Working towards adoption

- Step 3: Protocol document (std)
  - TBD

- Possible extension drafts

DHCPv6 Redundancy Considerations

Received comments from Applications Area Directorate

- Distinction between Service Provider and Enterprise Provider models should be clarified
- Multiple prefixes look like special case of split prefixes. Clarify differences.
- Must => MUST (???, not appropriate for BCP)
- Minor editorial comments

Advised to wait for AD direction before publishing -03

Overall status: Almost finished

No WG actions requested

DHCPv6 Failover Requirements

Received 5 comments from Prasad Gaitonde (thanks!)
All but one belong to design document rather than requirements

1. Configuration synchronization. Proposal:
   Allow secondary configuration, but overwrite it when connecting

Overall status: Minor update (-01) planned
More comments from WG requested

DHCPv6 Failover Design

- Initial submission followed by -01 in early March
- Based on v4 failover draft, but simplified
    - Hot standby (Active-passive only)
    - Failover relationship = 2 failover endpoints
    - No load balancing
    - Plan to remove CONFLICT-DONE and PAUSE states
- Contents:
    - Failover Endpoint State Machine
    - Connection management
    - 2 Resource Allocation Algorithms: Proportional and Independent
    - Time skew
    - Lazy updates + MCLT
- TODO:
    - Lease reservation
    - DDNS considerations
    - Many other smaller things

draft-mrugalski-dhc-dhcpv6-failover-design-01

Failover Design :: Connection Management

1. Communication over TCP
2. Reusing bulk leasequery framing, but with different message types
3. TLS usage optional
4. Failover endpoint – unique per role per partner per relationship. (referred to as “a partner”)
5. CONNECT, CONNECTACK, DISCONNECT
6. State notifications
7. Lease updates (BNDUPD, BNDUPDALL, BNDACK, UPDDONE)
8. Pool requests (POOLREQ, POOLRESP)
9. CONTACT (keep-alive)

Failover Design :: Resource Allocation

In both: a subpool of available resources I delegate to secondary.

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)

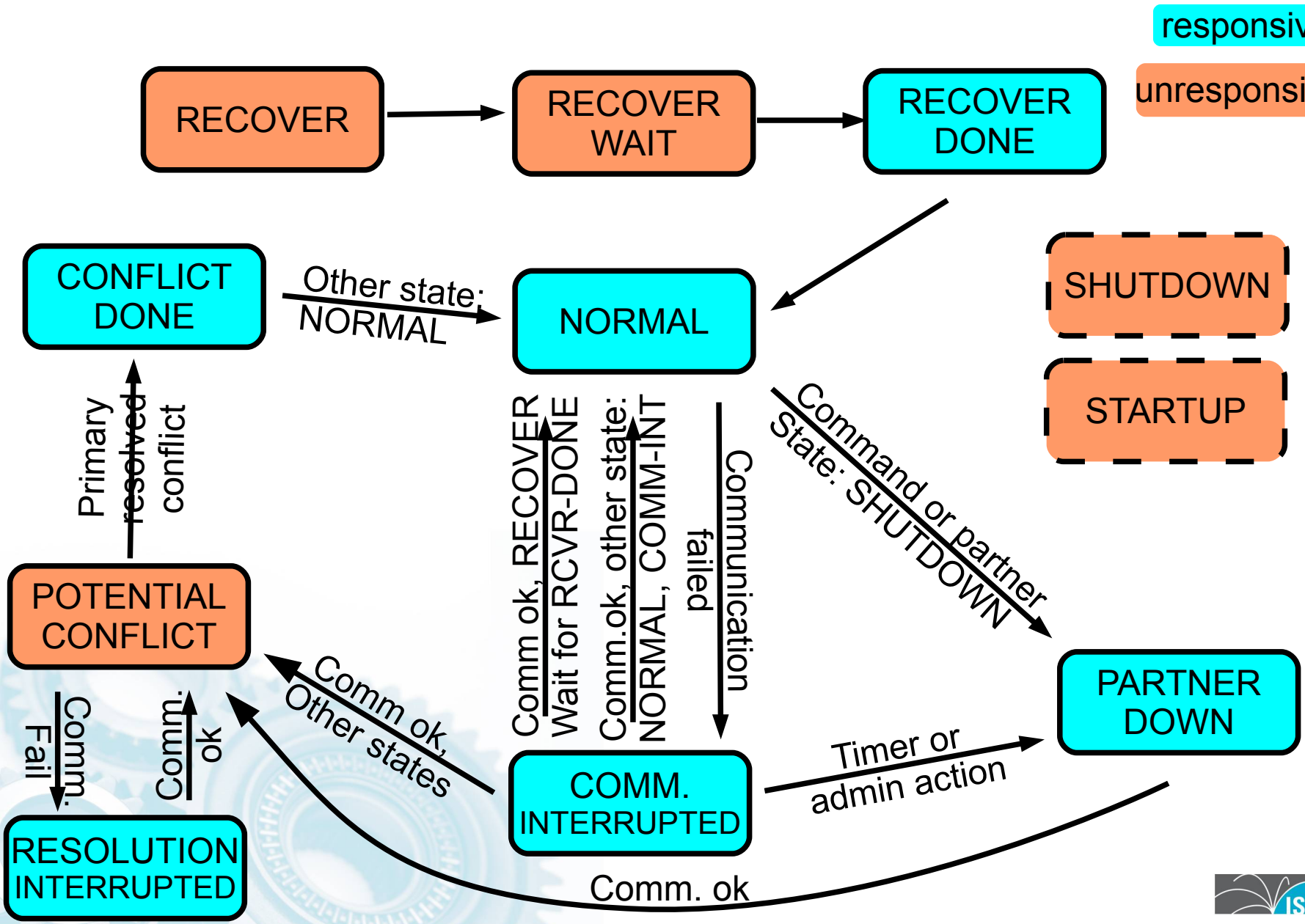
    Problem: failure between 1. and 2.
2. Maximum Client Lead Time
    - The maximum difference between lease time known by a client and acknowledge by its partner.
3. Useful in communications-interrupted
    - Server does not know if its partner extended any lease;
    - It knows that its parter could extend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

MCLT example

Valid lifetime = 3 days, MCLT = 1 hour

1. Client asks for an address.
2. Partner ack'd lease time is 0. Client gets 0+MCLT = 1 hour
3. Server updates its partner with 3 days + ½ hour.
4. Partner acks.
5. 30 minutes passes and client renews.
6. Partner's ack'd time is 3 days now.
7. Client receives renewed lease with valid lifetime 3 days.
8. Server updates its partner with expected renewal time (½*3 days) + desired potential valid lifetime (3 days) = 4,5 days.
9. Partner acks. Ack'd lease time is 4,5 days.
10. Client renews in 1,5 days and steps 6-9 repeat.

Failover Endpoint (partner) State Machine



DHCPv6 Failover Design :: Next steps

1. Comments are more than welcome.
2. This draft is a Standards-Track. Ok or change to INFO?
3. Are there any requirements regarding draft maturity before requesting adoption?
4. Other suggestions?

Thank you