DHCPv6 Failover Update
IETF85

Kim Kinnear <kkinnear@cisco.com>

Tomek Mrugalski <tomasz@isc.org>

2012-11-08

DHCPv6 Failover Grand Plan

- Step 0: Redundancy considerations
  - Submitted -03 that addresses raised IESG issues (done?)
  - Waiting for IESG to proceed

- Step 1: Requirements document (info)
  - WGLC in progress
  - Received several comments (clarification style)
  - Publish -03

- **Step 2: Design document (std)**
  - WG item, published -02
  - Text complete (no major missing parts)
  - Asking for review/comments

- Step 3: Protocol document (std)
  - Todo

DHCPv6 Failover Requirements

- WGLC announced Oct. 22, end by Nov. 12
- No comments so far

draft-ietf-dhc-dhcpv6-failover-requirements-02

DHCPv6 Failover Design
Overview

- -02 posted on Oct. 22, 2012
- Fulfills all requirements specified in failover-requirements-02
- Based on v4 failover draft, but simplified
- Hot standby (Active-passive only)
- No load balancing in design spec
  - likely extension
  - some provisioning ready
  - common state machine for base and load balancing

draft-ietf-dhc-dhcpv6-failover-design-02

DHCPv6 Failover Design
Major Concepts / Sections

- Lazy Updates for performance -> MCLT
- Failover Endpoint state machine
- Lease state machine additions
- Binding updates + conflict resolution
- TCP Connection management
- 2 Resource Allocation Algorithms
  - Proportional
  - Independent
- DDNS considerations
- Lease reservation

DHCPv6 Failover Design
Communication

- Communication over TCP
- Reuse bulk leasequery framing, with new failover specific message types
- TLS usage (optional)

DHCPv6 Failover Design
Messages

Connection management:
CONNECT, CONNECTACK, DISCONNECT

State notifications: **STATE**

Individual Lease updates: **BNDUPD, BNDACK**

Lease Update Requests:
UPDREQ, UPDREQALL, UPDDONE

Pool requests: **POOLREQ, POOLRESP**

Application level keep alive: **CONTACT**

DHCPv6 Failover Design
Resource Allocation

Two algorithms defined

Proportional allocation ("IPv4 failover-style")

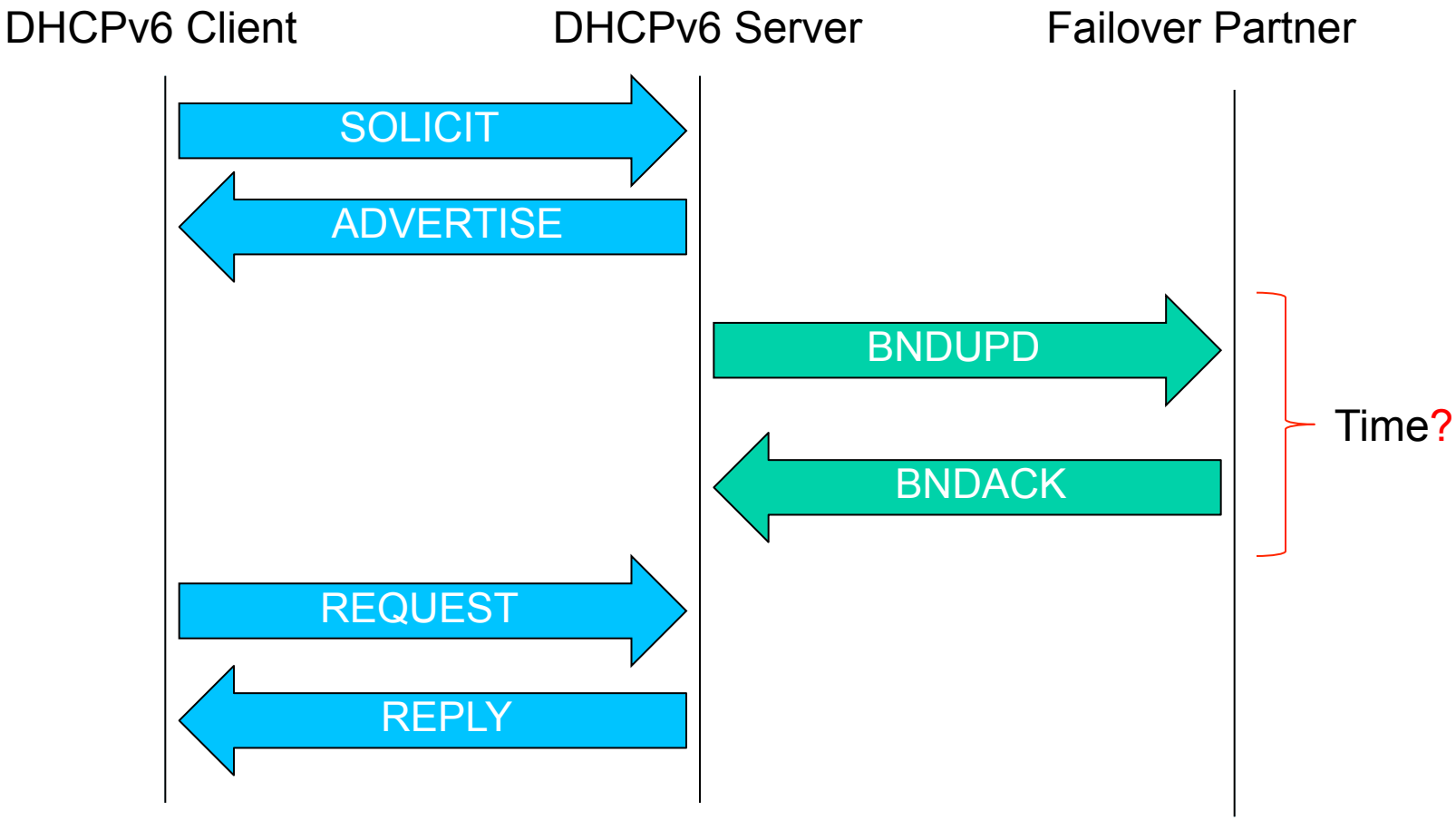
1. Pool may need to be rebalanced.
2. Only unleased resources are owned by specific server.
3. Useful for limited resources (e.g. prefixes)
4. Released/expired resources return to primary

DHCPv6 Failover Design
Resource Allocation

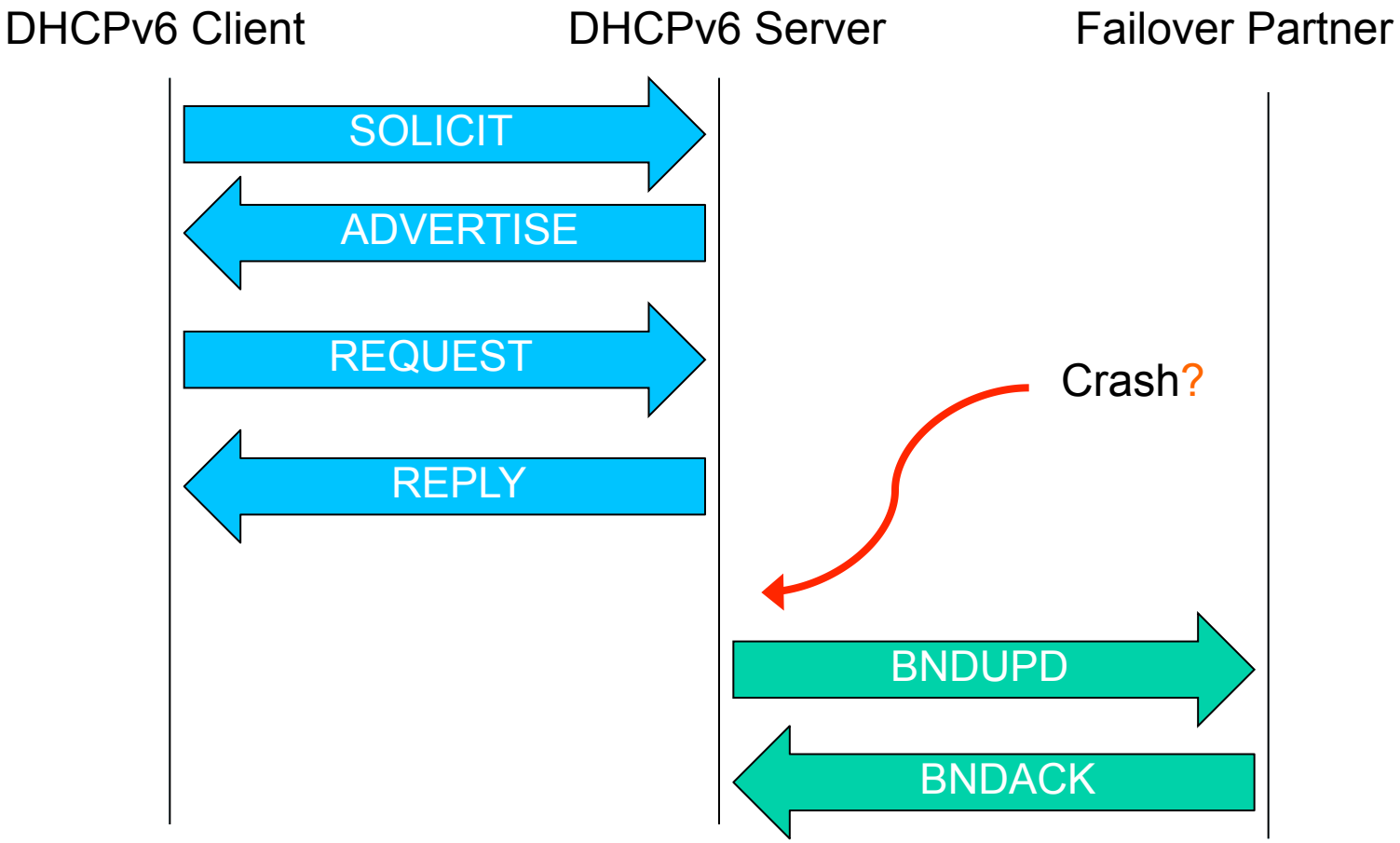
Independent allocation ("simple split")

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.

DHCPv6 Failover Design
Synchronized Update



DHCPv6 Failover Design
Lazy update



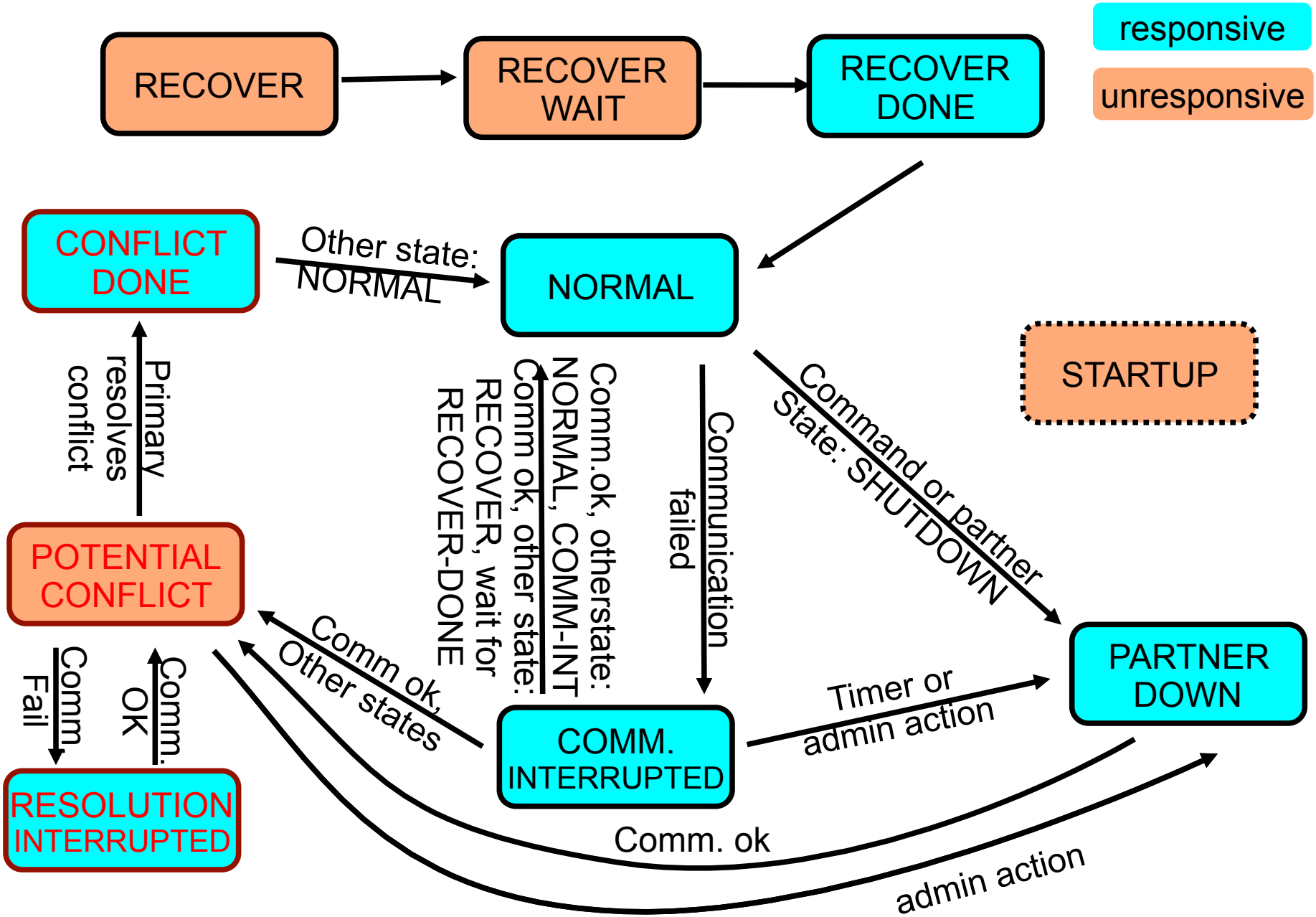
DHCPv6 Failover Design
Maximum Client Lease Time (MCLT)

The maximum difference between lease time known by a client and the lease time acknowledged by the failover partner.

Useful in communications-interrupted

- Server does not know if its partner extended any lease
- It knows that its partner could extend by **at most** MCLT
- To be on the safe side, server assumes that ALL leases were extended by MCLT.

Failover Endpoint (partner) State Machine



DHCPv6 Failover Design
Next steps

1. Comments are more than welcome
2. Working toward WGLC (needs more review)
3. Start work on protocol draft details (messages, options)

Thank you

Backup

MCLT example

Cast: Client, Server, (Failover) Partner

Valid lifetime = 3 days, MCLT = 1 hour

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