DHPv6 Bake-off Report

RIPE-NCC, Amsterdam,
March 14-16 2007

Alain Durand, Comcast

Bake-off Objectives

- Lots of new DHCPv6 code available in the last year or so.
- Initial lab test suggested some interoperability issues.
- Bake-off organized to verify interoperability, operation impact and usability with a larger set of implementations.
- We expected to find a small number of issues where implementers might have read the spec differently.

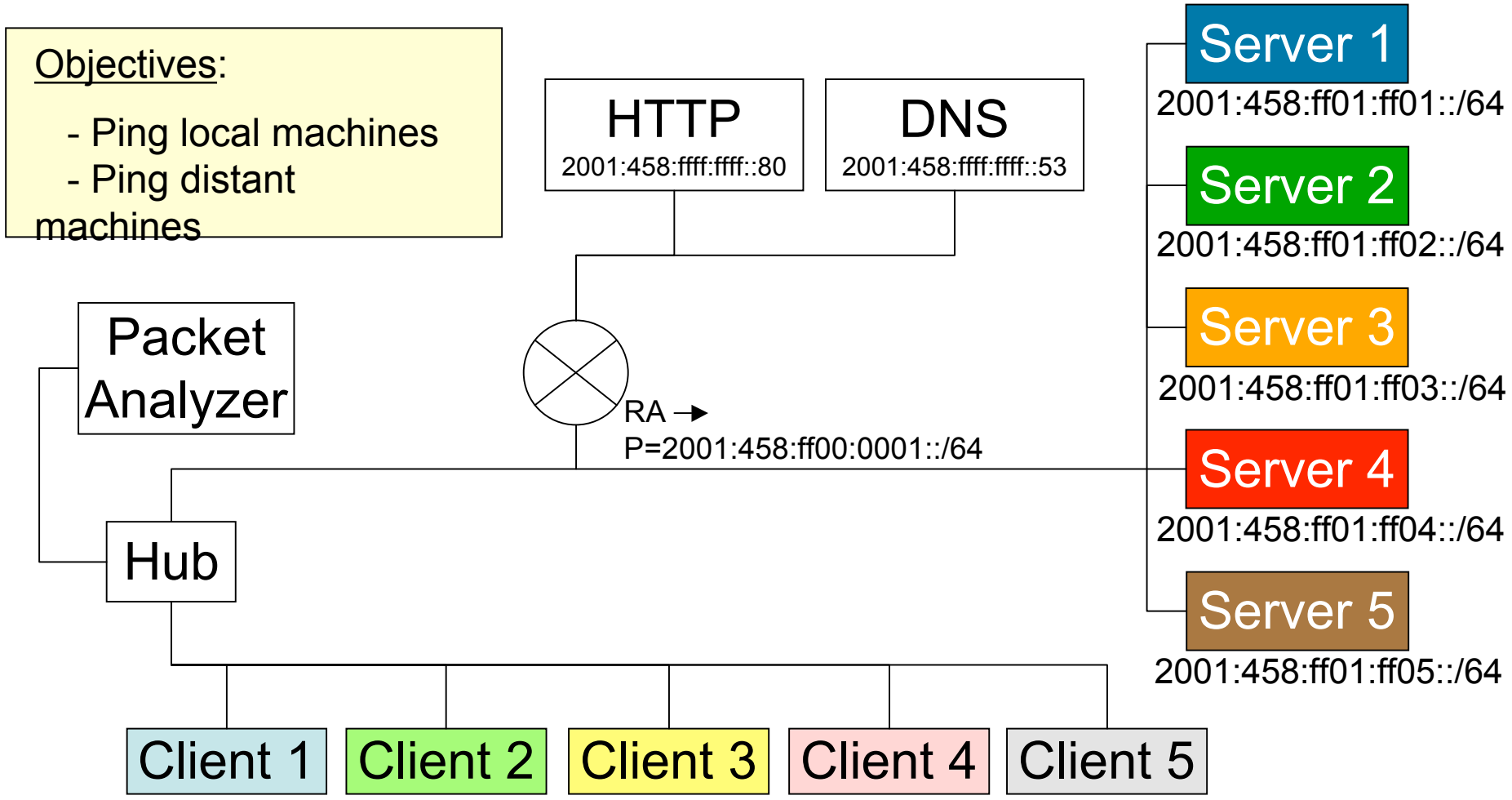
Who Was There?

- 7 vendors and/or open source providers
- 14 participants (one remote)
- 13 Implementations
  - 5 Clients
  - 5 Servers
  - 3 Relays

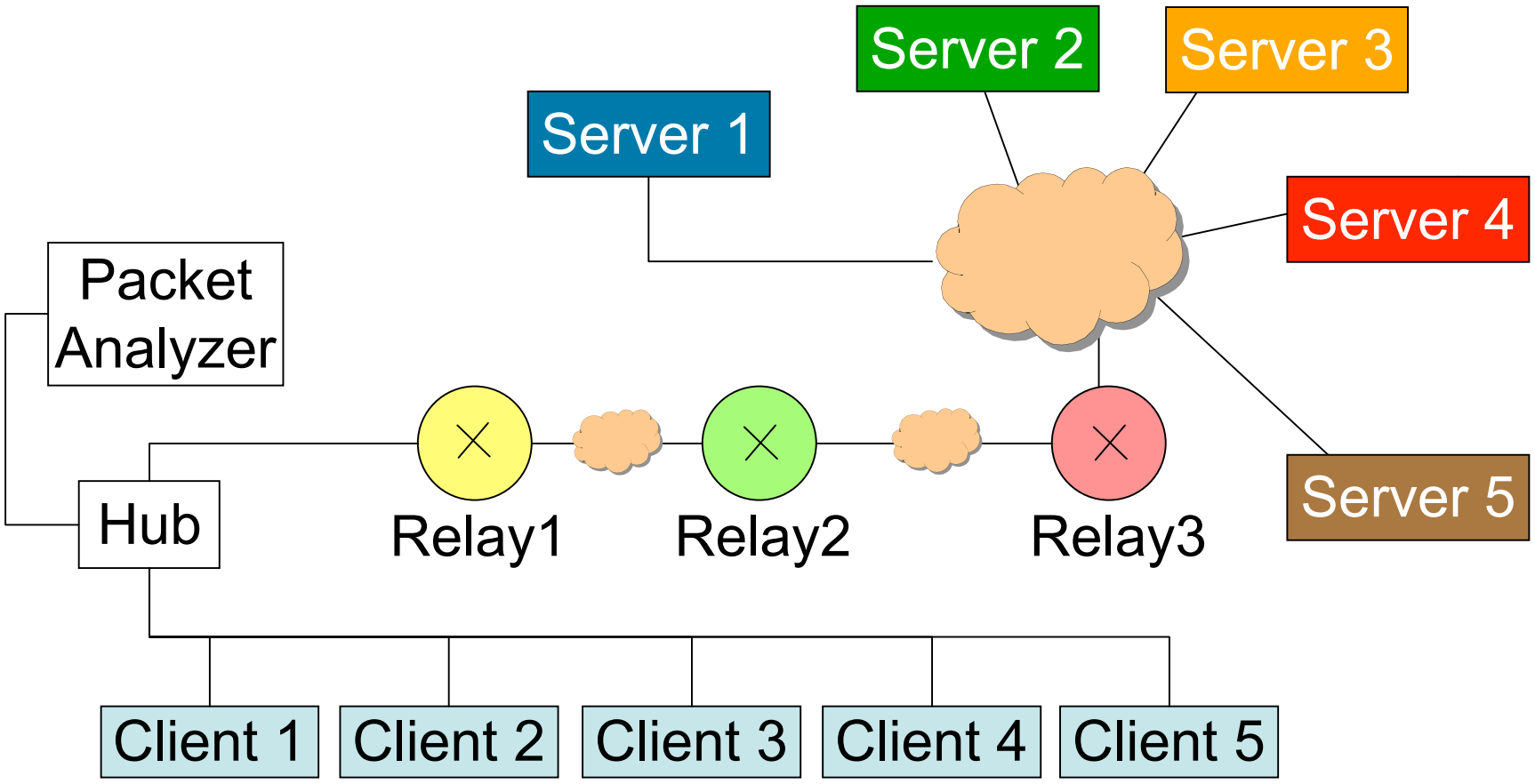
Special Thank You

- ISC for organizing the test plan
- RIPE-NCC for organizing the network
- Comcast crew who help run the test
- All the participants that I cannot name who came from 3 continents

Client / Server Test Topology

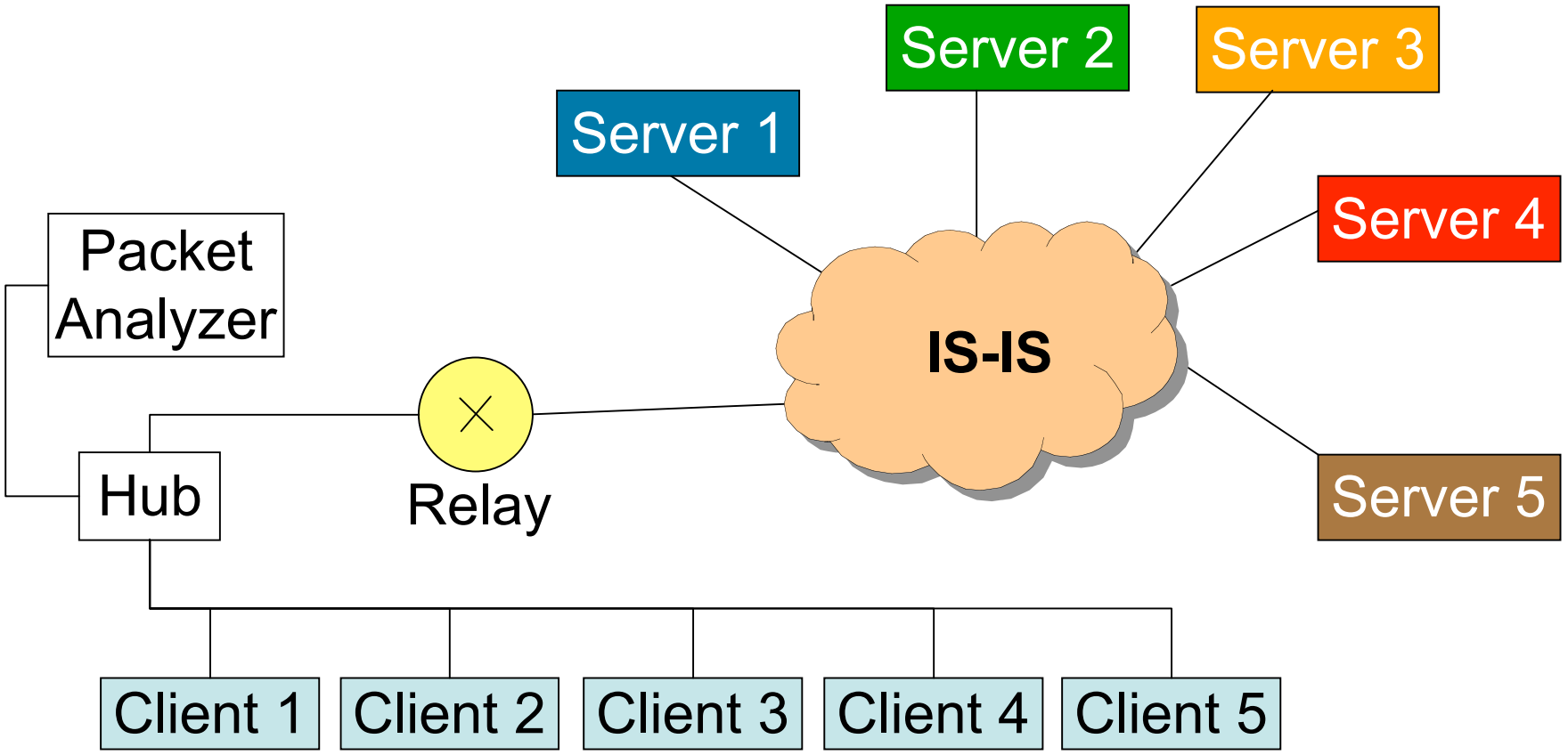


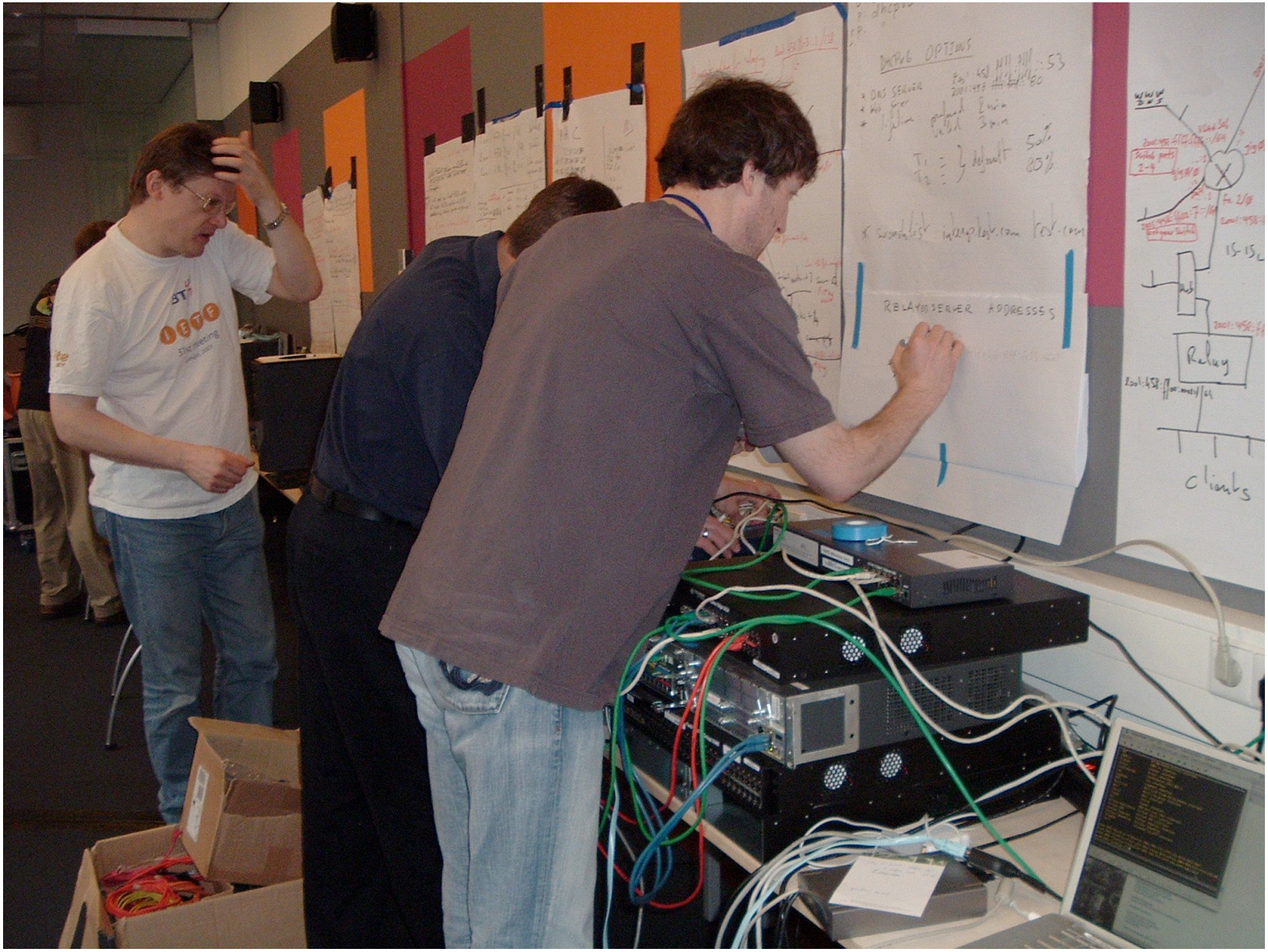
Unicast Relay Test Topology



Anycast Relay Test Topology

Each DHCPv6 server advertises 2001:458:ff03::1/128 to the IGP.
The IGP route requests to the ‘nearest’ one.
When one server fails, clients fall back to the next one.





The Crew

Bake-off Findings

- Most things ***worked***. Totally ***independent*** implementations could inter-operate well.
- We found ***16 operational or implementation choices issues*** that requires either clarifications or definition of new options in the spec.
- We will have another bake-off before Vancouver IETF!

Major Issues for Discussion

Issue 1

- Issue
  - Clients do not know how to route the local subnet associated with the addresses assigned by DHCPv6. Some assume prefix length is /64, some it is /128
- Suggested fix
  - Create new DHCPv6 server option to carry prefix length
- Work around
  - Manually add relevant routes on local router and rely on ICMPv6 redirect

Issue 6

- Issue
  - Client sends FQDN option to server to update the DNS.
    - How can the server notifies that the operation is ACKed or NACKed by DNS?
    - What should the server do if the name is already taken?
- Suggested fix
  - None
- Work around
  - Clients polls the DNS until something change…

Issue 4

- Issue
  - Client issues request including IA_ option. What should server do with IA_ADDR?
- Suggested fix
  - Client SHOULD include IA_ADDR from previous transaction
  - If IA_ADDR empty, server SHOULD generate a new address
  - If IA_ADDR not empty and the server is unwilling to lease the address , there are 2 options:
    - Error
    - Provide a different address
- Work around
  - none

Issue 16

- Issue
  - How to validate IA_ADDR field in IA_NA (or IA_TA)?
- Suggested fix
  - Define jointly with 3041bis an IANA registry to list restricted addresses.
  - A server should not lease an address in the reserved range unless configured to do so.
  - Client behavior in this case requires more discussion.
- Work around
  - none

Issue 2

- Issue
  - Server sets T1/T2=0. Client is allowed to renew whenever it wants. At least one client waited for the lease to expire before renewing. Interface went down and up and sometimes got a different address…
- Suggested fix
  - If T1/T2=0 and client don't know better, they SHOULD use default derived values
- Work around
  - none

Relay Related Issues

Issue 8

- Issue
  - Some servers use the link addr field of the relay agent to restrict the range of addresses to lease
- Suggested fix
  - Link addr field in relay agent is only a hint. Servers MUST be able to assign addresses outside of that scope
- Work around
  - none

Issue 9

- Issue
  - How should a relay choose the link-addr?
- Suggested fix
  - Link-addr must be the global unicast address of the interface from which the packet was received or is set to 0 if no value is available. It MUST not be a link-local address.
- Work around
  - none

Issue 10

- Issue
  - With multiple relays, which link-addr should the server use?
- Suggested fix
  - Servers should use the first non-zero link address in the chain of relays starting with the relay closest to client. If all link addresses in relay chain are zero, server may drop the packet.
- Work around
  - none

Issue 11

- Issue
  - Some relay implementers were confused about link-addr/peer-addr & relay-forward construction
- Suggested fix
  - The relay part of the spec need clarifications.
- Work around
  - none

Issue 13

- Issue
  - At least one relay implementation assumed that it needed to be a router and forward every packet
- Suggested fix
  - A relay agent is not required to be a router and forward all packets .
- Work around
  - none

Issue 14

- Issue
  - RFC3315 reserved multicast addresses are not useable for inter-relay multicasting:
    - The link local “all relays and servers” multicast address cannot be used between relays that are not on the same link.
    - The site local “all servers” multicast address can not be used between relays
- Suggested fix
  - Clarify the spec about this.
- Work around
  - none

Issue 15

- Issue
  - Potential of routing loop when using multicast for inter-relay communication when more than two levels of relays are in place.
- Suggested fix
  - Document the risk of multicast loop
  - Recommendation:
    “Use inter-relay multicast at your own risks”
- Work around
  - none

Issue 12

- Issue
  - What is the maximum number of relays? 4, 32, 256?
- Suggested fix
  - servers should be configurable, and default to the published value in the specification (32). Recommend servers should check the number of relay headers.
- Work around
  - none

Other Issues
Requiring Clarifications

Issue 3

- Issue
  - What should a server do when it receives a new request from the same client before the current lease expires?
- Suggested fix
  - The server SHOULD assign the same address again
- Work around
  - none

Issue 5

- Issue
  - Client sends ORO with FQDN sub-option but does not include a client FQDN option, how should server respond?
- Suggested fix
  - Server SHOULD ignore ORO FQDN request
- Work around
  - none

Issue 7

- Issue
  - Some clients use IA_ADDR with all zero to request a specific lifetime
- Suggested fix
  - “legitimize” this behavior
- Work around
  - none