

ND remaining problems

Erik Nordmark

Multicast RAs

- Modifying timers, unicast solicited RA, and changing standard/hosts to refresh using unicast RS
 - Would remove almost all multicast RAs
 - Router/interface up would multicast three RAs
 - Option to multicast RA when new information (additional prefixes, reduced lifetimes, etc)
- Mixed mode – how do routers know that all hosts perform RS refresh? [REMAINS]
 - A flag/option to RS so routers can know “new”?
 - Difficult for router to know “legacy”
 - Need some low-rate of multicast RAs for legacy hosts?

Multicast RS?

- Not a significant problem
 - sent to all-routers MC address
 - can be filtered and only sent towards routers
[DOCUMENT]
- Require that refresh RS be unicast
- Multicast RS when host moves (DNA)
 - DNA [RFC 6059] recommends multicast RS and unicast NS in parallel – better to defer RS in case response to NS [STD]
- Multicast RS when host powers on
 - But if hosts remember last state and do DNA that might help

Multicast Address Resolution

- Advertise prefixes with L=0 plus Redirect
 - Host will send to router
 - Router can redirect to subsequent packets go direct
- Should work with existing host implementations
 - Note that routers need to multicast NS to find each other and the hosts
 - What is the impact when a router loses state? Potentially multicast NS for every IP address on the link [STUDY]
- Does not apply to link-local addresses
 - Those are always assumed on-link
 - Do we need to improve link-locals? Under what constraints? [STUDY]

Multicast Address Resolution

- Router would multicast NS
 - Note that current proxy-dad approach helps little
 - Can not use DAD to reliably maintain NCEs
 - Potential to learn new source addresses from data packets (and unicast NS to get NA from host)
 - But implementation concerns? [SAVI-FCFS]
 - Need to share state across multiple routers on the link?
- Router state loss study would help discover worst case [STUDY]

Multicast for DAD? [REMAINS]

- Even if $L=0$ the DAD probes are multicast
- A DAD proxy can potentially help
 - But DAD proxy assumes always on, and
 - DAD proxy assumes reliable single DAD probe
 - Useful in constrained environments – not good enough for general deployment
- DAD proxy would assume filtering
 - no solicited-node MC reflected downstream by switches/APs
 - Solicited-node MC from routers sent downstream

Link-locals? [STUDY]

- Which protocols care about link-locals?
Multicast vs. unicast? Which also check hop limit=255?
 - ND
 - mDNS no 255 check
 - VRRP and other router-router protocols
- Do we need to improve this?
- Split-horizon not likely to be acceptable for general deployment

Sleepy nodes [REMAINS]

- DAD is a big concern.
- Possible approaches
 - Remove DAD from the IPv6 architecture? No really.
 - Host wakes up to defend (whether multicast DAD probe, or unicast NUD in dad-proxy approach)
 - Host performs DAD each time it wakes up
 - Eats battery partly due to required 1 second timeout
 - Proxy defends address based on implicit or explicit registration

Registrations?

- Proxy-dad draft performs implicit registrations using DAD probes
 - Assumes/requires that those are reliably delivered (which is not the case)
 - Using NUD to determine when an entry should be deleted (incorrect for sleepy nodes)
 - No choice since no lifetime and no “leave” message
 - But the best that can be done without host changes
 - Data path trigger (packets from unknown source) would address reliability but not support sleepy nodes

Explicit registrations

- Would allow sleepy nodes – a lifetime and host-driven renewal of the registrations
- Already defined and implemented - ARO in RFC 6775
 - Extensions in efficient-nd draft to handle router crash and VRRP
- Possible to define L3/L2 registrations using DHCP
 - Need to support host-generated addresses and L3->L2 bindings robustly
 - Implies updates to DHCP RFCs and code (DHCP client code, DHCP relay standard and code, define distribution from relay/server to defending routers, change approach for DNS server-server protocol?)

Next steps?

- Does the WG want to limit which parts of the problem should be solved?
 - Not handling sleepy nodes might simplify, but lack of reliable DAD will show up even without them.
- Design team to continue work?
 - More studies to complete – add to problem statement
 - Flush out tuning/deployment and implementation recommendations
 - Protocol improvements
- Testing issues? Rate limiting of RAs from Andrew's tests?