                Possible approaches to make DAD more robust and/or efficient
                       draft-nordmark-6man-dad-approaches-02

Abstract

   This outlines possible approaches to solve the issues around IPv6
   Duplicate Address Detection robustness and/or efficiency which are
   specified in the "DAD issues" dradft.

Status of this Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at http://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on April 3, 2016.

Table of Contents

1. Introduction

Duplicate Address Detection (DAD) is a procedure in IPv6 performed on
an address before it can be assigned to an interface [RFC4862]. By
default it consists of sending a single multicast Neighbor
Soliciation message and waiting for a response for one second. If no
response is received, the address is declared to not be a duplicate.
Once the address has been tested once, there is no further attempts
to check for duplicates (unless the interface is re-initialized).

The companion document [I-D.yourtchenko-6man-dad-issues] outlines a
set of issues around Duplicate Address Detection (DAD) which either
result in reduced robustness, or result in lower efficiency for
either the hosts wanting to sleep or the network handling more
multicast packets.

The reader is encourage to review the issues in that document. In
summary, the lack of robustness is due to only sending one or a few
DAD probe initially, and not having any positive acknowledgement that
"there are no duplicates". This implies that partioned links that
later heal can result in persistent undetected duplicate IPv6
addresses, including cases of "local partitions" such as the case of
a modem not having connected when the DAD probes are sent. The
inefficiences appears when there are low-powered devices on the link
that wish to sleep a significant amount of time. Such devices must
either be woken up by multicast Neighbor Soliciations sent to one of
their solicited-node multicast addresses, or they need to redo DAD
each time they wake up from sleep. Both drain the battery; the
second one results in sending a DAD probe and then waiting for a
second with the radion receiver enabled to see if a DAD message
indicates a duplicate.

2. Robustness Solution Approaches

IPv4 ARP robustness against partitions and joins is greatly improved

by Address Conflict Detection (ACD) [RFC5227]. That approach is
leverages the fact that ARP requests are broadcast on the link and
also makes the ARP replies be broadcast on the link. That
combination means that a host can immediately detect when some other
host provides a different MAC address for what the host thinks is its
own IPv4 address. That is coupled with state machines and logic for
determining whether to try to reclaim the address or give up and let
the other host have it. When giving up the host will form a new IPv4
address. The ACD approach results in more broadcast traffic than
normal ARP [RFC0826] since the ARP replies are broadcast.

Applying the same approach to IPv6 would require sending the Neighbor

Solicitations and Neighbor Advertisements to the all-nodes multicast
address so that a host can see when a different host is claiming/
using the same source IPv6 address. That would remove the efficiency
that Neighbor Discovery gets from "spreading" the resolution traffic
over 4 million multicast addresses.

One can envision variants on the theme of ACD that fit better with
the use of solicited-node multicast addresses. Suppose we have Host1
with IP1 that hashes to solicited-node multicast address SN1. And we
also have Host2 with IP2 and SN2.The link-layer addresses are MAC1
and MAC2, respectively. In [RFC4861] when Host1 wants to communicate
with Host2 we will see
1. Host1 multicasts a NS from IP1 to SN2. That include a claim for
    IP1->MAC1 using the Source Link-layer Address option.
2. Host2 receives the NA and unicasts a NA from IP2 to IP1. That
    includes a claim for IP2->MAC using the Taget Link-layer Address
    option.

If we want other hosts which might think they own either IP1 or IP2
to see the NA or NS (and we don't want to send the NS and NA to all-
nodes), then we can add additional multicast packets which explicitly
send the claim and send it to the Solicited-node multicast address of
the address that is being claimed. Thus
1. Host1 multicasts a NS from IP1 to SN2. That include a claim for
    IP1->MAC1 using the Source Link-layer Address option.
2. Host1 multicasts a NA from IP1 to SN1 explicitly claiming
    IP1->MAC1 using the TLLAO.
3. Host2 receives the NA and unicasts a NA from IP2 to IP1. That
    includes a claim for IP2->MAC using the Taget Link-layer Address

option.
    4. Host2 multicasts a NA from IP2 to SN2 explicitly claiming
        IP2->MAC2 using the TLLAO.

    The above explicit claims can then trigger the state machine
    described in ACD. The claims can probably be rate limited for any
    given source address since there is no need to repeat the claim just
    because a NS needs to be sent for a new IP3 etc. The impact of such
    rate limitig on the ability to detect duplicates.

    In the worst case the above approach turns one multicast and one
    unicast into three multicasts and one unicast, but all the multicasts
    are sent to solicited-node multicast addresss. Thus a host would not
    need to process the additional multicast packets.

    This ACD-multicast approach assumes that the multicast packets are
    delivered with reasonable reliability, but does not assume perfect
    delivery. If multicast reliability is lower than unicast it will
    result in retransmitted multicast NS in [RFC4861]. However, the

    above rate limiting idea might need care to ensure that claims are
    re-transmitted when the NS is re-transmitted.

    A slightly different approach to on-going DAD is what is implemented
    in Solaris where the node sends a periodic NA annoucement for the
    address it is using, plus the ACD behavior of detecting such an NA
    with a conflicting address. Presumably the NA annoucement can be
    sent to the solicited-node multicast address. It might make sense to
    use the Nonce option used by [I-D.ietf-6man-enhanced-dad] to avoid
    issues where a host would hear its own announcement.

3. Approaches to efficiency

    There exists some form of sleep proxies
    [ECMA-393][http://en.wikipedia.org/wiki/Bonjour_Sleep_Proxy] which
    perform handover of Neighbor Discovery protocol processing. [ECMA-
    393] does not specify the handover mechanism, and there is no know
    dcumentation for the handover mechanism. Even though the details are
    not specified, the approach seems to allow a host to sleep without
    worrying about DAD; the sleep proxy will respond to DAD probes. This
    seems to entail sending multicast NAs to all-nodes to hand-over the

IP address to the proxy's MAC address before going to sleep and then again to hand it back to the host's MAC address when it wakes up.

It is not clear whether such sleep proxies provide protection against Single Points of Failure i.e., whether the host can hand over things to a pair of sleep proxies.

FCFS SAVI [RFC6620] builds up state in devices to be able to detect and prevent when some host is trying to use an IPv6 address already used by another host on the link. This binding is built and checked for DAD packets, but also for data packets to ensure that an attacker can not inject a data packet with somebody elses source address. When FCFS SAVI detects a potential problem it checks whether the IPv6 address has merely moved to a different binding anchor (e.g., port on the switch) by sending a probe to its old anchor. Thus it assumes the host is always awake or can be awoken to answer that probe. Futhermore, implementation of the data triggered aspects can run into hardware limitations since it requires something like an ACL for every IPv6 address which has been validated.

DAD proxies as specified in [RFC6957] was designed to handle split-horizon forwarding which means that a host would never receive a multicast DAD probe sent by another host. This approach maintains a binding cache built up by DAD probes and checked when handling DAD probes. However, just like SAVI in order to handle host mobility and legitimate host MAC address change, it the case of a potential

conflict the proxy ends up verifying whether the IP address is still present at its old port and MAC address. Hence the host can not sleep.

One could explore something along the SAVI and DAD proxy approach that uses timestamps to allow better sleep. In principle would could start some fixed timer each time an IPv6 address is added or updated in the binding cache, and during that time the proxy would respond to DAD probes on behalf of the (potentially sleeping) host. To enable movement between ports/anchors such an approach would have to compare MAC address and assume that if the MAC address is the same it is the same host. (Unclear whether that is a good idea if we end up with random MAC addresses for better privacy.) And if a host would like to change its MAC address it would need to wait for the timeout before it can succeed in doing the change. Thus on one hand one

would want a long time (24 hours?) to facilitate for sleeping hosts,
and on the other hand a short time to allow for MAC address change
and movement.

In essence the above forms an implicit request for the proxy to
handle DAD on behalf of the host, with a fixed time limit. If the
host can instead make that time explicit, then the host can also
remove the proxy behavior (by passing a time of zero). Such a "proxy
for me" request can leverage the ARO option defined for 6LoWPan in
[RFC6775] but use it only for the purposes of DAD offload to the
proxy. That option can also carry an additional identifier which can
be used to distinguish between the same host aka same identifier
changing the MAC address. In the RFC that is an EUI-64 and in
[I-D.chakrabarti-nordmark-energy-aware-nd] in is a more generalized
identifier field. For redundancy the ARO can be sent to more than
one proxy.

4. Security Considerations

   If the working group decides to pursue one of the outlined approaches
   to improve the robustness and/or efficiency of DAD, then the security
   issues for that partiticular approach will need to be studied.

   In general DAD is subject to a Denial of Service attack since a
   malicious host can claim all the IPv6 addresses [RFC4218].

5. Acknowledgements

   Sowmini Varadhan pointed out the Solaris approach to use periodic NA
   annoucements to increase robustness.

6. References

6.1. Normative References

   [I-D.yourtchenko-6man-dad-issues]
              Yourtchenko, A. and E. Nordmark, "A survey of issues
              related to IPv6 Duplicate Address Detection",
              draft-yourtchenko-6man-dad-issues-01 (work in progress),

                   March 2015.

   [RFC0826] Plummer, D., "Ethernet Address Resolution Protocol: Or
              Converting Network Protocol Addresses to 48.bit Ethernet
              Address for Transmission on Ethernet Hardware", STD 37,
              RFC 826, DOI 10.17487/RFC0826, November 1982,
              <http://www.rfc-editor.org/info/rfc826>.

   [RFC4218] Nordmark, E. and T. Li, "Threats Relating to IPv6
              Multihoming Solutions", RFC 4218, DOI 10.17487/RFC4218,
              October 2005, <http://www.rfc-editor.org/info/rfc4218>.

   [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman,
              "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861,
              DOI 10.17487/RFC4861, September 2007,
              <http://www.rfc-editor.org/info/rfc4861>.

   [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless
              Address Autoconfiguration", RFC 4862, DOI 10.17487/
              RFC4862, September 2007,
              <http://www.rfc-editor.org/info/rfc4862>.

   [RFC5227] Cheshire, S., "IPv4 Address Conflict Detection", RFC 5227,
              DOI 10.17487/RFC5227, July 2008,
              <http://www.rfc-editor.org/info/rfc5227>.

   [RFC6620] Nordmark, E., Bagnulo, M., and E. Levy-Abegnoli, "FCFS
              SAVI: First-Come, First-Served Source Address Validation
              Improvement for Locally Assigned IPv6 Addresses",
              RFC 6620, DOI 10.17487/RFC6620, May 2012,
              <http://www.rfc-editor.org/info/rfc6620>.

   [RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C.
              Bormann, "Neighbor Discovery Optimization for IPv6 over
              Low-Power Wireless Personal Area Networks (6LoWPANs)",
              RFC 6775, DOI 10.17487/RFC6775, November 2012,
              <http://www.rfc-editor.org/info/rfc6775>.

   [RFC6957] Costa, F., Combes, J-M., Ed., Pougnard, X., and H. Li,
              "Duplicate Address Detection Proxy", RFC 6957,

              DOI 10.17487/RFC6957, June 2013,

<http://www.rfc-editor.org/info/rfc6957>.

6.2. Informative References

[I-D.chakrabarti-nordmark-energy-aware-nd]
          Chakrabarti, S., Nordmark, E., and M. Wasserman, "Energy
          Aware IPv6 Neighbor Discovery Optimizations",
          draft-chakrabarti-nordmark-energy-aware-nd-02 (work in
          progress), March 2012.

[I-D.ietf-6man-enhanced-dad]
          Asati, R., Singh, H., Beebee, W., Pignataro, C., Dart, E.,
          and W. George, "Enhanced Duplicate Address Detection",
          draft-ietf-6man-enhanced-dad-15 (work in progress),
          March 2015.

Author's Address

   Erik Nordmark
   Arista Networks
   Santa Clara, CA
   USA

   Email: nordmark@arista.com