Continuous DAD implementation in Solaris
draft-varadhan-continuous-dad-00

Abstract

This describes an implementation of IPv6 Duplicate Address Detection (DAD) in Solaris that merges concepts from RFC 5227 to address some of the known issues around DAD robustness and efficiency.

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 January 21, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
1. Introduction

When an IPv6 interface becomes active, DAD is initiated to probe if the address already exists. If no address conflict is detected after DAD probing the node is permitted to keep the address as unique/assigned [RFC4862]. The assumption is that if another node now wishes to configure that address, the new node will send out DAD probes. When the current owner intercepts these probes, it can send a reply to the all-nodes multicast address (Section 7.2.4 of [RFC4861]) asserting its ownership of the address to the new node which will recognize the DAD conflict and relinquish its attempt to configure the address. But that assumption can be invalid in (at least) two cases:

1. DAD probes are lost. Some link types like 802.11a/b/g/n do not pass multicast reliably.
2. A temporarily partitioned network heals, and partitions are merged. E.g., wireless networks that are temporarily out of range, or Ethernet hubs that are not functioning at the time of address configuration.

[RFC4862] does not address what to do if, after a unicast address is declared unique and assigned to an interface, the node receives a NS/NA from/for that matches an assigned address (indicating the presence of a duplicate address that was not detected by DAD). In other words, there is no Address Defense/Announcement mechanism specified in [RFC4862]: exactly one unsolicited NA is sent to the all-nodes mcast address announcing an address after DAD succeeds, and the address is never sent out to the solmcast or all-nodes mcast afterward. As a result duplicate addresses that are not detected during the DAD probing phase will go undetected, but will cause hard-to-diagnose packet delivery issues.

[I-D.yourtchenko-6man-dad-issues] outlines a set of issues around Duplicate Address Detection (DAD) that are analyzed in
The analysis points out that issues around DAD either result in lower efficiency (when there are low-powered devices in the network) or lower robustness (when there is a case of network partition and rejoin).

2. Robustness Solution Approaches

IPv4 ARP robustness against partitions and joins is greatly improved by Address Conflict Detection (ACD) [RFC5227]. One of the conflict detection mechanisms proposed by [RFC5227] is to broadcast both ARP requests and responses on the link. That combination allows every on-link host to quickly detect when some other host provides a different MAC address for an IPv4 address that the host may think it owns uniquely. The host may then use this information coupled with state machines and other logic to determine 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. This may be undesirable in some environments, as it would create an excessive amount of broadcast traffic on the wire that could be triggered by a mere ARP request.

This document describes a version of the address defense mechanism adapted into the Solaris IPv6 implementation that minimizes broadcast noise. The goal is to have some mechanism by which the owner of an IPv6 address can periodically announce its ownership to other onlink nodes, without creating excessive broadcast noise. This is achieved by having each node send a periodic NA announcement for the address it is using, plus the ACD behavior of detecting such an NA with a conflicting address.

3. DAD Probing During Address Configuration

When an address is configured on an interface, it is initially marked tentative, and DAD is initiated using the prescription in [RFC4862]. The node sends out 3 probes, each separated by RetransTimer milliseconds. The RetransTimer is set to a 150 ms for "fast" probing. Fast probes are used when the link layer supports up/down notification, and when the address under consideration is not an IPv4 Link-Local Address. RetransTimer is set to 1500 ms for all other cases. Each probe is a NS that does not include a SLLA. It is sent with the IPv6 source address set to IN6ADDR_ANY, with the desired/tentative IPv6 address as the Target IPv6 Address in the NS. The NS is sent to the solmcast for the desired/tentative address.
4. Ongoing Address Defense

When the required number of DAD probes have been sent out without any conflicts, the sending node begins announcing its newly acquired address through DAD Announcements that are sent out as unsolicited Neighbor Advertisements in IPv6. These should be sent out following the parameters described in Section 7.2.6 of [RFC4861], i.e., it should send out ANNOUNCE_NUM (MAX_NEIGHBOR_ADVERTISEMENT, default 3 transmissions) NAs that are ANNOUNCE_INTERVAL seconds apart.

After the pre-defined number of DAD announcements have been sent, the system switches to the lazier DAD-defense mode, where the IPv6 address is announced at periodic intervals. The defend interval for IPv6 packets defaults MUST be an administratively configurable value. The mechanism for implementing ongoing address-defense thus becomes:

1. Have periodic announce timer set at ANNOUNCE_INTERVAL.
2. If the ANNOUNCE_INTERVAL expires without any intervening unsolicited NA, send one.
3. Unsolicited NA’s should be rate-limited to a maximum of DEFEND_RATE NA’s in any given DEFEND_PERIOD (e.g., 100 NA’s per hour) NA, send one.
4. When replying to a NS, check against the last solmcast announcement. If the last solicited mcast was sent more than DEFEND_INTERVAL seconds before the current time, and we have not yet met the rate limit quota of DEFEND_RATE NAs in the current DEFEND_PERIOD, this NA will be solmcast, and the announce timer is reset. Note that the application of the rate-limiting algorithm and checks against the time of the last solmcast announcement are intended to efficiently manage the link level broadcast/multicast traffic, and are a departure from [RFC5227].

5. Reacting to Address Conflicts in Incoming NS/NA packets

If an incoming NS or NA indicates an address conflict for a tentative address, the tentative address is deleted from the interface to resolve the conflict. If the incoming NS or NA conflicts with a non-tentative address, the receiver MAY send out a NA to announce/defend the address. The conditions under which such an announcement are sent are listed below:

- The elapsed time interval since the last defense is greater than DEFEND_INTERVAL
- If the non-tentative address is an [RFC4941] IPv6 Temporary Address, the owner of the address will send out MAX_TEMP_DEFEND NAs to defend the address. If the non-tentative address is not a [RFC4941] temporary address, MAX_DEFEND NAs will be sent to defend the address. Both MAX_TEMP_DEFEND and MAX_DEFEND parameters are
administratively configurable tunables that default to 1 and 3 respectively.

6. Recovery Upon Conflict

If an address has been defended MAX_DEFEND/MAX_TEMP_DEFEND times within the DEFEND_INTERVAL, but the node continues to receive NAs that indicate that the address is a duplicate, then there is an address conflict in the network.

When an address conflict occurs between two nodes for an address that is not tentative on either of those nodes, one or both of them may have to relinquish ownership of the address. Nodes that lose ownership of an address may have to do some or all of the following.

- delete the address from the system,
- send a TCP reset for any connections using the address as source address, and kill sockets bound to that address,
- log messages to notify the administrator about the detected conflict,
- if the address was acquired via DHCP, send a DHCP DECLINE to the DHCP server

Deleting an address to resolve conflicts is disruptive to existing connections, and the conflict resolution scheme has to be a locally administered policy. Factors that may impact the policy may include the time-interval for which each node has claimed ownership of the address, number of impacted sockets on each conflicting node, and the services offered by each node. As an example, if one of the conflicting nodes is a DHCP server [RFC5227] points out that it may not be possible for the server to re-number that address.

The Solaris implementation choice for addressing address conflict was to disable the address if conflicting NAs continue to be received after the MAX_DEFEND/MAX_TEMP_DEFEND announcements. The rationale behind this choice is that a conflicting IP address is mostly unusable in practice and attempting to ignore this degenerate situation is not productive. However, Solaris will continue to send out DAD probes for the disabled/conflicting address so that if the usurper of the address is removed from the network, the disabled address will be seamlessly re-enabled.

7. References
7.1. Normative References

[I-D.yourtchenko-6man-dad-issues]


7.2. Informative References

[I-D.nordmark-6man-dad-approaches]
Nordmark, E., "Possible approaches to make DAD more robust and/or efficient", draft-nordmark-6man-dad-approaches-02 (work in progress), October 2015.

Author’s Address

Sowmini Varadhan
Oracle Corp
Redwood City, CA
USA

Email: sowmini.varadhan@oracle.com