

Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 22, 2011

W. Dec
Cisco Systems
June 20, 2011

IPv6 Router Solicitation Driven Access Considered Harmful
draft-dec-6man-rs-access-harmful-00

Abstract

This document presents issues regarding the reliance of IPv6 Router Solicitation messages for creating or initializing router state necessary to enable IPv6 users' connectivity, particularly in situations where such users have bridged ethernet connectivity with the router. A number of alternative solution approaches are also presented and discussed.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 December 22, 2011.

Copyright Notice

Copyright (c) 2011 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.

Table of Contents

1. Introduction	3
2. Problem Overview	4
2.1. RS Sending Proxy	6
3. Discussion of possible solutions	8
3.1. Modifying RFC4861	9
3.2. Modifying RS-proxy and router behaviour	9
3.3. Ethernet Connectivity Fault Monitoring	10
3.4. Access-Node based DHCPv6 Proxy Client	10
3.5. DHCPv6 client on end hosts	11
3.6. ANCP	12
3.7. Other	12
4. Conclusions	12
5. IANA Considerations	13
6. Security Considerations	13
7. Contributors and Acknowledgements	13
8. References	14
8.1. Normative References	14
8.2. Informative References	14
Author's Address	14

1. Introduction

Recent proposals for including subscriber line identifiers alongside host sourced Router Solicitation (RS) messages ([I-D.ietf-6man-lineid]) in an environment where the host has no direct link layer adjacency with the router (eg when using Ethernet bridging), have highlighted the intent of using these RS messages on the receiving router as a trigger for specific functions & processes. Without the execution of these processes, such as host or line authorization, the host will not receive Router Advertisements (RAs) that allow the establishment of full IPv6 connectivity. Similar RS triggered processes, although without line identifiers, are proposed in specifications concerning WiFi access and appear to share the same pitfalls.

In analyzing the impact of these proposals it is useful to refer to the basics of the IPv6 Neighbour Discovery protocol as defined in [RFC4861], which defines the Router Solicitation (RS) message type. This message is intended to be used by hosts to request routers to generate Router Advertisements sooner than at their next scheduled time. The Router Solicitation mechanism is intended to be used in a very specific set of cases, or not at all, and a regular IPv6 network can work fully without any RS message ever being sent. In general, as per Section 6.3.7 of [RFC4861], Router Solicitations may be sent by a host after any of the following events:

- o The interface is initialized at system startup time.
- o The interface is reinitialized after a temporary interface failure or after being temporarily disabled by system management.
- o The system changes from being a router to being a host, by having its IP forwarding capability turned off by system management.
- o The host attaches to a link for the first time.
- o The host re-attaches to a link after being detached for some time.

Notably in the above a host is at no stage required to periodically send RS messages, nor to send RS messages after a period of not receiving any RAs.

Furthermore [RFC4861] states that once a host "receives a valid Router Advertisement with a non-zero Router Lifetime, the host MUST desist from sending additional solicitations on that interface, until the next time one of the above events occurs." This effectively signifies that following the reception of any given RA message, sent by any device, a host will not issue RS messages until it is

reattached or re-initialized.

The following text from [RFC4861] also illustrates another aspect relating to the rule governing a host's ceasing of RS sending.

"If a host sends MAX_RTR_SOLICITATIONS solicitations, and receives no Router Advertisements after having waited MAX_RTR_SOLICITATION_DELAY seconds after sending the last solicitation, the host concludes that there are no routers on the link"

Experimental evidence conducted on a number of IPv6 implementations confirms that the above behaviour is indeed currently the norm, with specific implementations differing in terms of the default timers (eg MAX_RTR_SOLICITATION_DELAY) used. One implementation has been found to send RS messages at evenly spaced 4 second intervals for up to 12 seconds after the link event. Another implementation has been found to exponentially increase the sending interval for successive messages and stopping RS sending after 90 seconds.

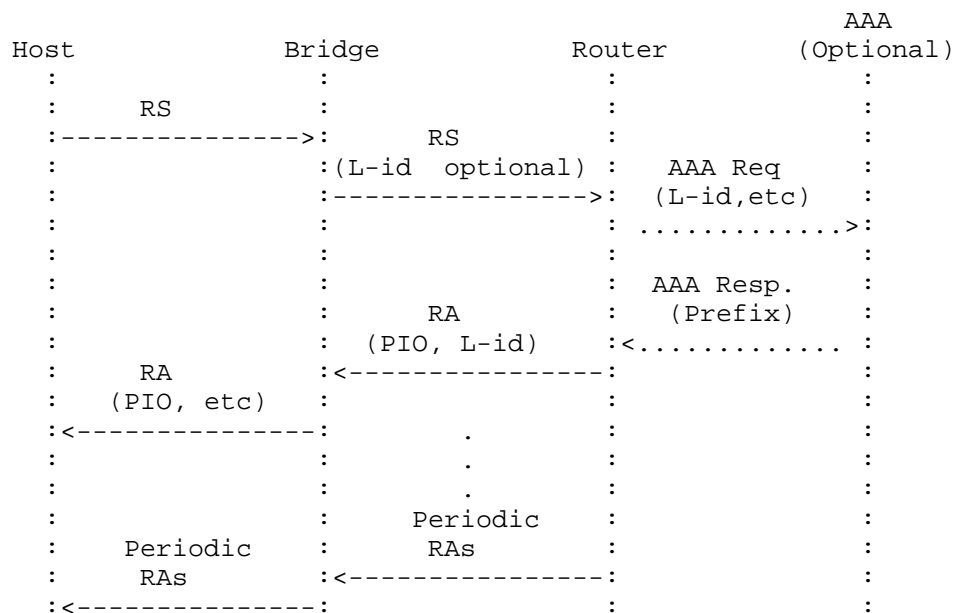
The RS sending mechanism was thus clearly not designed nor is implemented to be periodic, nor reliable, nor expected to be sent by a host that has timed out or received an RA. Any mechanism that presupposes any of these RS sending characteristics, or requires them to work reliably, requires a thorough review.

2. Problem Overview

The main intent of the [I-D.ietf-6man-lineid] proposal is to convey from an Ethernet bridging Access-Node to an upstream IPv6 router, the subscriber-line-id information indicating the origin of downstream host sourced RS messages. All this is envisaged to be done by tunneling such RS messages using IPinIP tunneling between the Access-Node and the Router, with the access node inserting the subscriber-line-id for each tunnelled RS. The reception by the router of such RS messages with the subscriber-line-id is expected to be the trigger for authorizing and allowing the subscriber's connectivity to the network. It is crucial to note that only after successful authorization will the router send RA messages that contain IPv6 Prefix Information Option (PIO) that allow the host to configure a global IPv6 address. A direct example of this usage goal can be found in Section 6.5 and Appendix A of [TR-177].

In generic terms, the principle of such mechanism is shown in Figure 1, and the goal is to create a dynamic user driven IPv6 access system that is in conductive to:

- a. Triggering by means of subscriber sourced ND (RS) messages, processes on the IP edge router which serve to provide and setup hosts/subscribers with IPv6 connectivity.
- b. Deriving from the received messages host identifiers and/or information regarding where the host is connected to in the Layer 2 network (eg based on MAC address and/or subscriber line id) and using that information in performing access and/or address authorization prior to granting connectivity.
- c. Being used in an environment where the host/subscriber has no directly link layer adjacency with the router, but rather indirect connectivity (eg via a bridged Ethernet RG/CPE, and/or a bridging DSLAM).
- d. Being used in an environment where IPv6 hosts implement *only* [RFC4861] as the control protocol, and without any further host changes or client protocols (eg DHCPv6)



A number of deployment contexts that seek to realize such a system will result in the end user having no IPv6 connectivity, and being left without any automated means of recovery it, all very detrimental to the success of the IPv6 deployment.

One such deployment context is the residential broadband N:1 VLAN environment, as described by [I-D.ietf-6man-lineid]. This features hosts indirectly connected to the edge router over a bridged Layer 2 VLAN set-up (aka an N:1 VLAN). End subscriber hosts connect to Ethernet bridging devices, such as an RG/modem and an Access-Node/DSLAM, which provide indirect link connectivity for the host with the router. From each end hosts perspective, its local LAN link state is as presented by the RG/modem's LAN interface, eg Ethernet or WiFi. This state is decoupled from the RG/modem's uplink interface state, or that of the DSLAM links, or that of the IP edge router interface(s). Hence, each host's interface is expected to be "up" even when no DSL WAN link synchronisation has been established, or when the WAN link is being established following a modem reboot (an event lasting 2 minutes or more is not uncommon), etc. Given this, and in consideration of the RS sending characteristics described in Section 1, it is near certain that following a bridge/modem reload, or a DSLAM reload, any and all RS messages sent by hosts will never arrive at the intended IP edge router within the time hosts send RS messages. Since the reception of such RS messages by the edge router is required to trigger the announcement of RAs containing the chosen user address prefix option (PIO) towards the hosts, the host will be left without any addressing information and thus no IPv6 connectivity. The only recourse a user has is manual intervention on the host's interface.

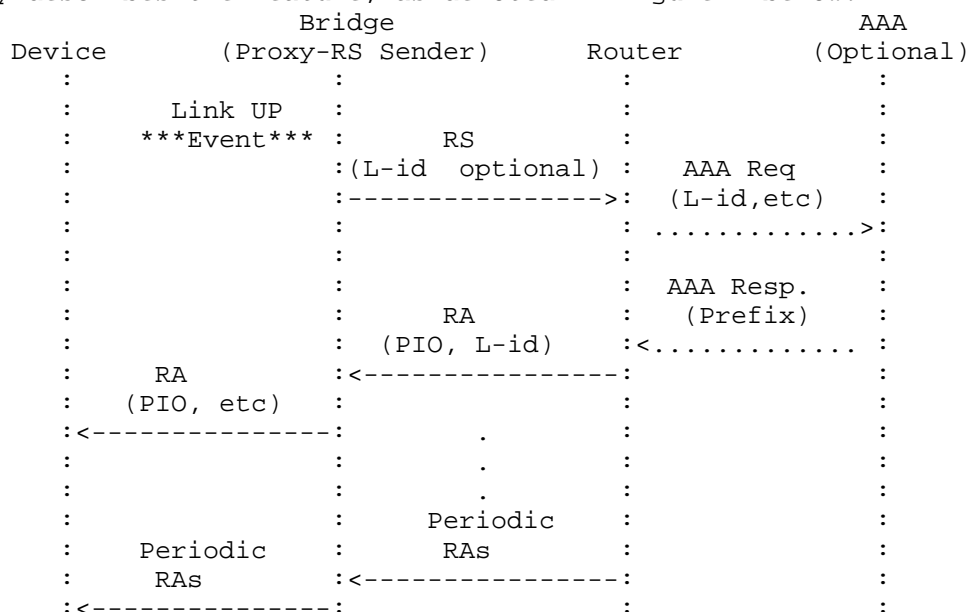
Note: The example of DSL is used above, but the case applies to other media, eg cable modems, that exhibit similar "modem reload" events. Moreover, the same problem appears to apply to each deployment that seeks to realize the mentioned goals and features hosts that have no direct link layer adjacency with a router, eg IEEE 802.11 WiFi architectures.

It's significant to note that the [I-D.ietf-6man-lineid] mechanism implicitly assumes behaviour which by itself will result in the system failing non-deterministically. As exemplified by its usage described in [TR-177], "empty RAs" (ie Router-Advertisement messages that contain no addressing/prefix information) are to be multicast to all subscribers and hosts from the router, in parallel to any specific RAs containing prefix information and the line option. Again, following the cited rules of [RFC4861], should a subscriber host receive such an empty RA prior to issuing an RS, that host will never send an RS and thus never trigger the authorization process necessary to get global IPv6 addressing & connectivity.

2.1. RS Sending Proxy

An update to the [I-D.ietf-6man-lineid] draft proposal has somewhat recognized the critical flaw described in Section 2. It also

attempted a remedy in the form of introducing an Access-Node feature, as described in Section 5.3 of [I-D.ietf-6man-lineid]. This feature, consists in the Access Node issuing RS messages towards the Router driven by subscriber link activation (and only activation) state (ie when the link is "brought up"). The term "proxy-RS sender" rather aptly describes the feature, as denoted in Figure 2 below.



[I-D.ietf-6man-lineid] indicates that a finite number of RS messages are to be sent and that sending should stop after the Access Node receives an RA with a matching subscriber line information option back from the edge router. This remedy, in the context of the overall solution, is not only insufficient, but introduces further problems, consisting of:

1. Unreliability of RS messaging: There is no assurance that the RS messages sent by the proxy will reach the edge router. Eg it is not uncommon for spanning tree protocol events take place on the Ethernet segments, or other similar events, which result in loss of connectivity with the edge router ranging from a couple of seconds to a couple of minutes - this is often the case during access-node activation. Any RS messages sent by the RS-proxy, on behalf of bridged subscribers connected to this access node, would be lost and all the relevant subscribers left without IPv6 connectivity.

2. Lack of subscriber host identifiers: In many of today's broadband deployments end host identifiers are required for the purpose of authorization besides intermediate identifiers such as subscriber line-id. For example, it is quite common to identify and authorize devices like WiFi smart phones or TV set-top-boxes by their unique MAC address. With the RS-proxy mechanism, these identifiers are not be available, and effectively do not meet goal b) of the system
3. No ability to clean up state/recover: Each "active" subscriber link is intended to induce IPv6 subscriber state in the router. Short of manual intervention by the operator there is no mechanism on the router to remove such state should a link ever become "inactive". In other words, there is no equivalent of a "link down" message, nor does the ND protocol provide for such extensibility, and the router and operator are likely to be burdened with a large amount of stale state, besides inefficient use of resources.
4. In ability to recover from node failures: Given that an RS-proxy eventually stops sending RSes, should the edge router loose for any reason any or all of the RS induced state, including the route to the subscriber, the system will fall into a state of unrecoverable connectivity loss for end users, even as they continue to have a valid IPv6 address. Basically, a host that received a previous RA from an Edge Router will following rfc4862 NOT send an further RS messages, while a router without the necessary state will NOT forward traffic to the subscriber. Similarly, neither will the RS-proxy send RS messages as long as the line is still "active".

Given the above issues, while the introduction of the RS-sending-proxy was intended to fix a critical flaw with the original proposal, if not only left the issue in place, but it introduced further issues undermining its overall purpose and compromising the usability and scalability of the system.

3. Discussion of possible solutions

Its readily apparent that any solution based on proxy functionality that is driven by link state changes cannot meet all of the system goals as presented in Section 2 (eg goals a, b and c), while satisfying the constraint of no changes to end hosts (goal e) and within the context of a bridged/indirect-link host-router set-up (goal d). At best compromises to the goals or combinations of solutions need to be adopted. The solutions below indicate such compromises:

3.1. Modifying RFC4861

One possible, solution, that would solve a handful of issues, would be to modify [RFC4861] in such a way as to give the protocol a semblance of reliability and persistence. For example, it could be stipulated that host RS sending behaviour needs to be periodic and continue irrespective of RA messages being received. Router behaviour would need to be modified to detect periods of RS inactivity. All this would be a substantial change to the original protocol specification, and would naturally require changes to any existing IPv6 ND implementations to be useful, falling short of goal e). Besides this, it would also significantly increase the RS processing load on any router.

3.2. Modifying RS-proxy and router behaviour

Modifying the RS-proxy mechanism to issue periodic RS messages driven subscriber link state, or doing so whenever no RA is received for a given subscriber line over a certain period of time could be seen as a possible solution to some, but not all, of the problems identified. In essence this modification transforms RS/RA messaging into link-state notification messages. Unfortunately it also introduces several other flaws, besides not meeting the Section 2 goals a), b) and possibly c):

- o Unknown timers: For the mechanism to function, the behaviour of both the RS-proxy and the edge router need to be modified in terms of RS processing and RA sending, around a timer driven state machine, where both the Access-Node and Router share the timers. Defining for this purpose a new timer negotiation protocol appears a major ND or IPinIP protocol change, while relying on "well known" timers (ie hard set) is highly inflexibility not conducive to automated, reliable and inter operable deployments.
- o Increased load on AAA system: Following the intent of the system, for each RS message for which no authorization state exists on the edge router, authorization from an AAA server is to be requested. With RS messages being periodic, this will place additional burden on any AAA infrastructure, besides being analogous to issuing AAA requests for each link keepalive received.
- o Subscriber management: One of the main premises of an architecture that features a Layer 2 Access Node and an upstream aggregating IP Edge Router is the notion of subscriber management on the IP Edge Router. Operators deploying this architecture seek to use the IP Edge Router as the node on which subscriber related configuration and control is applied - hence the desire to perform dynamic subscriber authorization at/by the router. Introducing into this

architecture a mechanism where periodic RS messages sent by a proxy could lead to similarly periodic denial of authorizations at the edge router, eg for subscriber lines that are not authorized to use the service, with the only way of disabling such RS sending is by maintaining on the Access-Node subscriber configuration information, is counter to the premise of the architecture itself.

- o ND customization: One of the design goals for using the IPinIP tunneling mechanism was to avoid changes to the ND protocol or implementations. Unfortunately, the processing of custom tunneled RS messages as well as generation of custom tunneled RA messages, in effect requires a highly customized ND implementation, the likes of which diverges from typically ND implementations.

Given the above, modifying the RS-proxy mechanism to be periodic would not only require a fairly major extension to the proposal, including the definition of timers covering message sending periodicity discovery and/or negotiation, but also result in more issues to the overall system. Above all, such a modification would in the end only mimic a link-state signalling/keepalive protocol, without actually resolving all of the identified problems, and without actually being one.

3.3. Ethernet Connectivity Fault Monitoring

A core issue in the a system driven by host sourced RS, is the end hosts inability to detect when an indirect link has failed, translating into the hosts inability to re-send RS messages. On links such as PPP, which offer link state keepalives, the issue does not come up, but neither does the need of driving router authorization events via RS messages due to the link layer negotiation stage of PPP. Over Ethernet, a link state keepalive mechanisms could fill in part of that gap. The closest equivalent can be found in Ethernet Connectivity Fault Monitoring that is a component of the IEEE 802.1ag Ethernet OAM specification [802.1ag]. The implementation of such extensions on hosts and routers would allow the regular [RFC4861] RA sending rules to respond appropriately to connectivity or device failures. Unfortunately, there is no known end host implementation of 802.1ag today, which translates that this solution does not meet goal e) (no end host modifications). Nevertheless, it appears like a valid approach, whose realization however does not appear to be within the IETF's specification direct sphere of influence.

3.4. Access-Node based DHCPv6 Proxy Client

An alternative solution to some of the problems identified in relation to periodic RA sending, would be to define an RS/

RA-DHCPv6-proxy function, whose role would be to transform host sourced RS messages into DHCPv6 Solicit/etc messages towards the edge router. The access-node would thus be a multi DUID DHCPv6 client as seen by the rest of the operator's network. Regular mechanisms of DHCPv6 relaying by the edge router and prefix delegation would be used to assign /64 prefixes for each subscriber line. The RS/RA-DHCPv6 proxy would also be responsible for announcing the DHCPv6 derived prefixes in regular RA messages to downstream hosts. An additional bonus of this solution is the fact that the existing DHCPv6 specification allows for the subscriber line-id to be included in the DHCPv6 messages [RFC3315], [RFC6221]. Hence, no additional RS subscriber line id or IPinIP tunnel header extensions would be required, effectively obviating all of the [I-D.ietf-6man-lineid] protocol extension requirements. Similarly, none of the upstream devices, would appear to be affected in supporting this solution.

Though this solution solves the problem of error recovery, state deletion and timer discovery/negotiation, besides removing the need to define any protocol extensions to convey line-id information, in its RS triggered form it remains prone to the critical flaws described in Section 2. Hence, a more reliable version of this solution would see the DHCPv6 proxy client be invoked by line-state changes. Unfortunately, this variant again does not meet goals a), b) and possibly c). Nevertheless, with these usability caveats clearly recognized, it appears that this solution is still superior to what is currently found in [I-D.ietf-6man-lineid], and does not require protocol extensions.

3.5. DHCPv6 client on end hosts

A solution that would see most of the goals realized, without the need to define any new protocol extensions, would be to rely on DHCPv6 [rfc3315] client functionality in the end host. DHCPv6 was designed to offer the degree of reliability sought for, as well as periodic retransmissions of messages, along with client identifiers. The compromise in this solution would be that it does not appear to fit goal e), at least when looked from a universal current host implementation perspective, namely that some end hosts would be required to implement a DHCPv6 client.

Given the relation of the problem being addressed to the bridged connectivity model, a non technical variant of this solution at the service level is to stipulate in the user's terms and conditions it is supported only with DHCPv6 clients. This approach has been effectively assumed by the Cablelabs specifications for bridged media connectivity [MULPI], as well as put into practice by several Ethernet FTTx network operators.

3.6. ANCP

The Access Node Control Protocol (ANCP) [I-D.ietf-ancp-protocol] defines a suite of mechanisms for conveying information pertaining to the state of a subscriber access line between a Layer 2 access node physically terminating the subscriber access line and a separate Layer 3 router. One of the key capabilities of the protocol is that to signal line state changes from the access-node to the router, as well as to apply dynamic configuration on access-lines retrieved from the router. In the combination, these two capabilities offer another alternative solution, at least in so far as a line-state driven mechanism can provide.

The basic premise of the solution would see the Access-Node use existing ANCP "Port-UP" or "Port-Down" messages, which also convey line-id, to signal line state changes to the edge router. These could be considered as the trigger events to drive the edge router to send to the Access-Node either "Line Configuration" messages with IPv6 parameters, or define a new "Raw data" message type which would ferry a raw RA to be sent on the access-line.

As with any of the other Access-Node line state driven solutions, meeting goals a) and b) would not be possible. Despite that, ANCP offers a robust and reliable (TCP based) line-state communication mechanism between an Access-Node and Edge Router, which does not need re-inventing.

3.7. Other

The solution proposed by [I-D.ietf-6man-lineid], consisting in adding a subscriber-line-id parameter as part of an IPinIP encapsulation header, can be realized practically by various other tunneling protocols. Specifically, L2TPv3 already defines AVPs for subscriber-line-id information. As with other solutions that rely only on tunneling host sourced RAs, this will be prone to host connectivity impediments.

4. Conclusions

Due to the inherent design and implementation characteristics of the ND protocol, mechanisms that gate IPv6 user connectivity based on the reception of an RS message are likely to lead to serious IPv6 connectivity failures for end users, and leave both users and operators with no automated means of recovering from the situation. The issues are particularly severe in cases when the end users do not have a direct link adjacency to the router, as is often the case in bridged Ethernet or WiFi based broadband access networks. Moreover,

such a mechanism appears not to meet the expected more general usage goals as presented in Section 2. As such, the definition and deployment of such mechanisms is considered to be harmful to the success of IPv6 usage, and thus should be discouraged in favour of alternative solutions.

Two alternative solutions presented in Sections 3.4 and 3.6, can comprehensively meet the majority of the Section 2 goals. The solution presented in Section 3.5, which has proven to meet the requirements of many operators, indicating the imposed host constraints might not be universally applicable, remains a valid approach which requires no protocol extensions.

Solution variants seek to redress the lack of direct link state adjacency by using an intermediate link state driven messaging proxy function incur a shortcoming. This consist in their inability to be able to provide the to the authorization system information such as the end host MAC address. Thus, any such solution carries usage constraints, that should be clarified.

The solution variant proposed by [I-D.ietf-6man-lineid] introduces itself numerous issues of reliability and deployability, whose resolution is not trivial without major ND protocol extensions, if not other protocol work. Alternatives, as presented in Section 3.4, 3.6 and 3.7 all offer more robust and deployable mechanisms that in most cases leverage already defined protocols and mechanisms hence appear to offer a much more viable solution path.

5. IANA Considerations

This document does not raise any IANA considerations.

6. Security Considerations

The security of the solutions outlined needs to be evaluated in specific solution documents.

7. Contributors and Acknowledgements

The author would like to thank Erik Nordmark, Ole Troan, and Sean Cavanaugh for reviewing this document.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

8.2. Informative References

- [I-D.ietf-6man-lineid]
Krishnan, S., Kavanagh, A., Varga, B., Ooghe, S., and E. Nordmark, "The Line Identification Destination Option", draft-ietf-6man-lineid-01 (work in progress), March 2011.
- [I-D.ietf-ancp-protocol]
Wadhwa, S., Moisand, J., Haag, T., Voigt, N., and T. Taylor, "Protocol for Access Node Control Mechanism in Broadband Networks", draft-ietf-ancp-protocol-17 (work in progress), April 2011.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC5851] Ooghe, S., Voigt, N., Platnic, M., Haag, T., and S. Wadhwa, "Framework and Requirements for an Access Node Control Mechanism in Broadband Multi-Service Networks", RFC 5851, May 2010.
- [RFC6221] Miles, D., Ooghe, S., Dec, W., Krishnan, S., and A. Kavanagh, "Lightweight DHCPv6 Relay Agent", RFC 6221, May 2011.
- [TR-177] - Broadband Forum, <<http://www.broadband-forum.org/technical/download/TR-177.pdf>>
- [IEEE802.1ag] - IEEE, <<http://www.ieee802.org/1/pages/802.1ag.html>>

Author's Address

Wojciech Dec
Cisco Systems
Haarlerbergweg 13-19
1101 CH Amsterdam
The Netherlands

Email: wdec@cisco.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 31, 2011

W. Kumari
Google
I. Gashinsky
Yahoo!
J. Jaeggli
Zynga
June 29, 2011

Operational Neighbor Discovery Problems and Enhancements.
draft-gashinsky-v6nd-enhance-00

Abstract

In IPv4, subnets are generally small, made just large enough to cover the actual number of machines on the subnet. In contrast, the default IPv6 subnet size is a /64, a number so large it covers trillions of addresses, the overwhelming number of which will be unassigned. Consequently, simplistic implementations of Neighbor Discovery can be vulnerable to denial of service attacks whereby they attempt to perform address resolution for large numbers of unassigned addresses. Such denial of attacks can be launched intentionally (by an attacker), or result from legitimate operational tools that scan networks for inventory and other purposes. As a result of these vulnerabilities, new devices may not be able to "join" a network, it may be impossible to establish new IPv6 flows, and existing ipv6 transported flows may be interrupted.

This document describes the problem in detail and suggests possible implementation improvements as well as operational mitigation techniques that can in some cases to protect against such attacks. It also discusses possible modifications to the traditional [RFC4861] neighbor discovery protocol itself.

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 December 31, 2011.

Copyright Notice

Copyright (c) 2011 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.

Table of Contents

1. Introduction	4
1.1. Applicability	4
2. The Problem	4
3. Terminology	5
4. Background	6
5. Neighbor Discovery Overview	7
6. Operational Mitigation Options	7
6.1. Filtering of unused address space.	8
6.2. Appropriate Subnet Sizing.	8
6.3. Routing Mitigation.	8
6.4. Tuning of the NDP Queue Rate Limit.	9
7. Recommendations for Implementors.	9
7.1. Prioritize NDP Activities	10
7.2. Queue Tuning.	11
7.3. NDP Protocol Gratuitous NA	11
7.4. ND cache priming and refresh	12
8. IANA Considerations	13
9. Security Considerations	13
10. Acknowledgements	13
11. References	14
11.1. Normative References	14
11.2. Informative References	14
Appendix A. Text goes here.	14
Authors' Addresses	14

1. Introduction

This document describes implementation issues with IPv6's Neighbor Discovery protocol that can result in vulnerabilities when a network is scanned, either by an intruder or through the use of scanning tools that perform network inventory, security audits, etc. (e.g., "nmap").

This document describes the problem in detail and suggests possible implementation improvements as well as operational mitigation techniques that can in some cases protect against such attacks. It also discusses possible modifications to the traditional [RFC4861] neighbor discovery protocol itself.

The RFC series documents generally describe on-the-wire behavior of protocols, that is, "what" is to be done by a protocol, but not exactly "how" it is to be implemented. The exact details of how best to implement a protocol will depend on the overall hardware and software architecture of a particular device. The actual "how" decisions are (correctly) left in the hands of implementers, so long as implementations produce proper on-the-wire behavior.

While reading this document, it is important to keep in mind that discussions of how things have been implemented beyond basic compliance with the specification is not in the scope of the neighbor discovery RFCs.

1.1. Applicability

This document is primarily intended for operators of IPV6 networks and implementors of [RFC4861]. The Document provides some operational consideration as well as recommendations to increase the resilience of the Neighbor Discovery protocol.

2. The Problem

In IPv4, subnets are generally small, made just large enough to cover the actual number of machines on the subnet. For example, an IPv4 /20 contains only 4096 addresses. In contrast, the default IPv6 subnet size is a /64, a number so large it covers literally billions of billions of addresses, the overwhelming number of which will be unassigned. Consequently, simplistic implementations of Neighbor Discovery can be vulnerable to denial of service attacks whereby they perform address resolution for large numbers of unassigned addresses. Such denial of attacks can be launched intentionally (by an attacker), or result from legitimate operational tools that scan networks for inventory and other purposes. As a result of these

vulnerabilities, new devices may not be able to "join" a network, it may be impossible to establish new IPv6 flows, and existing ipv6 transport flows may be interrupted.

Network scans attempt to find and probe devices on a network. Typically, scans are performed on a range of target addresses, or all the addresses on a particular subnet. When such probes are directed via a router, and the target addresses are on a directly attached network, the router will attempt to perform address resolution on a large number of destinations (i.e., some fraction of the 2^{64} addresses on the subnet). The process of testing for the (non)existence of neighbors can induce a denial of service condition, where the number of Neighbor Discovery requests overwhelms the implementation's capacity to process them, exhausts available memory, replaces existing in-use mappings with incomplete entries that will never be completed, etc. The result can be network disruption, where existing traffic may be impacted, and devices that join the net find that address resolutions fails.

In order to alleviate risk associated with this DOS threat, some router implementations have taken steps to rate-limit the processing rate of Neighbor Solicitations (NS). While these mitigations do help, they do not fully address the issue and may introduce their own set of potential liabilities to the neighbor discovery process.

3. Terminology

Address Resolution Address resolution is the process through which a node determines the link-layer address of a neighbor given only its IP address. In IPv6, address resolution is performed as part of Neighbor Discovery [RFC4861], p60

Forwarding Plane That part of a router responsible for forwarding packets. In higher-end routers, the forwarding plane is typically implemented in specialized hardware optimized for performance. Forwarding steps include determining the correct outgoing interface for a packet, decrementing its Time To Live (TTL), verifying and updating the checksum, placing the correct link-layer header on the packet, and forwarding it.

Control Plane That part of the router implementation that maintains the data structures that determine where packets should be forwarded. The control plane is typically implemented as a "slower" software process running on a general purpose processor and is responsible for such functions as the routing protocols, performing management and resolving the correct link-layer address for adjacent neighbors. The control plane "controls" the

forwarding plane by programming it with the information needed for packet forwarding.

Neighbor Cache As described in [RFC4861], the data structure that holds the cache of (amongst other things) IP address to link-layer address mappings for connected nodes. The forwarding plane accesses the Neighbor Cache on every forwarded packet. Thus it is usually implemented in an ASIC .

Neighbor Discovery Process The Neighbor Discovery Process (NDP) is that part of the control plane that implements the Neighbor Discovery protocol. NDP is responsible for performing address resolution and maintaining the Neighbor Cache. When forwarding packets, the forwarding plane accesses entries within the Neighbor Cache. Whenever the forwarding plane processes a packet for which the corresponding Neighbor Cache Entry is missing or incomplete, it notifies NDP to take appropriate action (typically via a shared queue). NDP picks up requests from the shared queue and performs any necessary actions. In many implementations it is also responsible for responding to router solicitation messages, Neighbor Unreachability Detection (NUD), etc.

4. Background

Modern router architectures separate the forwarding of packets (forwarding plane) from the decisions needed to decide where the packets should go (control plane). In order to deal with the high number of packets per second the forwarding plane is generally implemented in hardware and is highly optimized for the task of forwarding packets. In contrast, the NDP control plane is mostly implemented in software processes running on a general purpose processor.

When a router needs to forward an IP packet, the forwarding plane logic performs the longest match lookup to determine where to send the packet and what outgoing interface to use. To deliver the packet to an adjacent node, It encapsulates the packet in a link-layer frame (which contains a header with the link-layer destination address). The forwarding plane logic checks the Neighbor Cache to see if it already has a suitable link-layer destination, and if not, places the request for the required information into a queue, and signals the control plane (i.e., NDP) that it needs the link-layer address resolved.

In order to protect NDP specifically and the control plane generally from being overwhelmed with these requests, appropriate steps must be taken. For example, the size and rate of the queue might be limited.

NDP running in the control plane of the router dequeues requests and performs the address resolution function (by performing a neighbor solicitation and listening for a neighbor advertisement). This process is usually also responsible for other activities needed to maintain link-layer information, such as Neighbor Unreachability Detection (NUD).

An attacker sending the appropriate packets to addresses on a given subnet can cause the router to queue attempts to resolve so many addresses that it crowds out attempts to resolve "legitimate" addresses (and in many cases becomes unable to perform maintenance of existing entries in the neighbor cache, and unable to answer Neighbor Solicitation). This condition can result in the inability to resolve new neighbors and loss of reachability to neighbors with existing ND-Cache entries. During testing it was concluded that 4 simultaneous nmap sessions from a low-end computer was sufficient to make a router's neighbor discovery process unhappy and therefore forwarding unusable.

This behavior has been observed across multiple platforms and implementations.

5. Neighbor Discovery Overview

When a packet arrives at (or is generated by) a router for a destination on an attached link, the router needs to determine the correct link-layer address to send the packet to. The router checks the Neighbor Cache for an existing Neighbor Cache Entry for the neighbor, and if none exists, invokes the address resolution portions of the IPv6 Neighbor Discovery [RFC4861] protocol to determine the link-layer address.

RFC4861 Section 5.2 (Conceptual Sending Algorithm) outlines how this process works. A very high level summary is that the device creates a new Neighbor Cache Entry for the neighbor, sets the state to INCOMPLETE, queues the packet and initiates the actual address resolution process. The device then sends out one or more Neighbor Solicitations, and when it receives a corresponding Neighbor Advertisement, completes the Neighbor Cache Entry and sends the queued packet.

6. Operational Mitigation Options

This section provides some feasible mitigation options that can be employed today by network operators in order to protect network availability while vendors implement more effective protection

measures. It can be stipulated that some of these options are "kludges", and are operationally difficult to manage. They are presented, as they represent options we currently have. It is each operator's responsibility to evaluate and understand the impact of changes to their network due to these measures.

6.1. Filtering of unused address space.

The DOS condition is induced by making a router try to resolve addresses on the subnet at a high rate. By carefully addressing machines into a small portion of a subnet (such as the lowest numbered addresses), it is possible to filter access to addresses not in that portion. This will prevent the attacker from making the router attempt to resolve unused addresses. For example if there are only 50 hosts connected to an interface, you may be able to filter any address above the first 64 addresses of that subnet by nullrouting the subnet carrying a more specific /122 route.

As mentioned at the beginning of this section, it is fully understood that this is ugly (and difficult to manage); but failing other options, it may be a useful technique especially when responding to an attack.

This solution requires that the hosts be statically or statefully addressed (as is often done in a datacenter) and may not interact well with networks using [RFC4862]

6.2. Appropriate Subnet Sizing.

By sizing subnets to reflect the number of addresses actually in use, the problem can be avoided. For example [RFC6164] recommends sizing the subnet for inter-router links to only have 2 addresses. It is worth noting that this practice is common in IPv4 networks, partly to protect against the harmful effects of ARP flooding attacks.

6.3. Routing Mitigation.

One very effective technique is to route the subnet to a discard interface (most modern router platforms can discard traffic in hardware / the forwarding plane) and then have individual hosts announce routes for their IP addresses into the network (or use some method to inject much more specific addresses into the local routing domain). For example the network 2001:db8:1:2:3::/64 could be routed to a discard interface on "border" routers, and then individual hosts could announce 2001:db8:1:2:3::10/128, 2001:db8:1:2:3::66/128 into the IGP. This is typically done by having the IP address bound to a virtual interface on the host (for example the loopback interface), enabling IP forwarding on the host and having it run a routing

daemon. For obvious reasons, host participation in the IGP makes many operators uncomfortable, but can be a very powerful technique if used in a disciplined and controlled manner.

6.4. Tuning of the NDP Queue Rate Limit.

Many implementations provide a means to control the rate of resolution of unknown addresses. By tuning this rate, it may be possible to ameliorate the issue, although, as with most tuning knobs (especially those that deal with rate limiting), you may be "completing the attack". By excessively lowering this rate you may negatively impact how long the device takes to learn new addresses under normal conditions (for example, after clearing the neighbor cache or when the router first boots) and, under attack conditions you may be unable to resolve "legitimate" addresses sooner than if you had just the the knob alone.

It is worth noting that this technique is only worth investigation if the device has separate queue for resolution of unknown addresses versus maintenance of existing entries.

7. Recommendations for Implementors.

The section provides some recommendations to implementors of IPv4 Neighbor Discovery.

At a high-level, implementors should program defensively. That is, they should assume that intruders will attempt to exploit implementation weaknesses, and should ensure that implementations are robust to various attacks. In the case of Neighbor Discovery, the following general considerations apply:

Manage Resources Explicitly - Resources such as processor cycles, memory, etc. are never infinite, yet with IPv6's large subnets it is easy to cause NDP to generate large numbers of address resolution requests for non-existent destinations. Implementations need to limit resources devoted to processing Neighbor Discovery requests in a thoughtful manner.

Prioritize - Some NDP requests are more important than others. For example, when resources are limited, responding to Neighbor Solicitations for one's own address is more important than initiating address resolution requests that create new entries. Likewise, performing Neighbor Unreachability Detection, which by definition is only invoked on destinations that are actively being used, is more important than creating new entries for possibly non-existent neighbors.

7.1. Prioritize NDP Activities

Not all Neighbor Discovery activities are equally important. Specifically, requests to perform large numbers of address resolutions on non-existent Neighbor Cache Entries should not come at the expense of servicing requests related to keeping existing, in-use entries properly up-to-date. Thus, implementations should divide work activities into categories having different priorities. The following gives examples of different activities and their importance in rough priority order.

1. It is critical to respond to Neighbor Solicitations for one's own address, especially when a router. Whether for address resolution or Neighbor Unreachability Detection, failure to respond to Neighbor Solicitations results in immediate problems. Failure to respond to NS requests that are part of NUD can cause neighbors to delete the NCE for that address, and will result in followup NS messages using multicast. Once an entry has been flushed, existing traffic for destinations using that entry can no longer be forwarded until address resolution completes successfully. In other words, not responding to NS messages further increases the NDP load, and causes on-going communication to fail.

2. It is critical to revalidate one's own existing NCEs in need of refresh. As part of NUD, ND is required to frequently revalidate existing, in-use entries. Failure to do so can result in the entry being discarded. For in-use entries, discarding the entry will almost certainly result in a subsequent request to perform address resolution on the entry, but this time using multicast. As above, once the entry has been flushed, existing traffic for destinations using that entry can no longer be forwarded until address resolution completes successfully.

3. To maintain the stability of the control plane, Neighbor Discovery activity related to traffic sourced by the router (as opposed to traffic being forwarded by the router) should be given high priority. Whenever network problems occur, debugging and making other operational changes requires being able to query and access the router. In addition, routing protocols may begin to react (negatively) to perceived connectivity problems, causing additional undesirable ripple effects.

4. Activities related to the sending and receiving of Router Advertisements also impact address resolutions. [XXX say more?]

5. Traffic to unknown addresses should be given lowest priority. Indeed, it may be useful to distinguish between "never seen" addresses and those that have been seen before, but that do not have

a corresponding NCE. Specifically, the conceptual processing algorithm in IPv6 Neighbor Discovery [RFC4861] calls for deleting NCEs under certain conditions. Rather than delete them completely, however, it might be useful to at least keep track of the fact that an entry at one time existed, in order to prioritize address resolution requests for such neighbors compared with neighbors that have never been seen before.

7.2. Queue Tuning.

On implementations in which requests to NDP are submitted via a single queue, router vendors SHOULD provide operators with means to control both the rate of link-layer address resolution requests placed into the queue and the size of the queue. This will allow operators to tune Neighbour Discovery for their specific environment. The ability to set or have per interface or subnet queue limits at a rate below that of the global queue limit might limit the damage to the neighbor discovery process to the taret network.

Setting those values must be a very careful balancing act - the lower the rate of entry into the queue, the less load there will be on the ND process, however, it also means that it will take the router longer to learn legitimate destinations. In a datacenter with 6,000 hosts attached to a single router, setting that value to be under 1000 would mean that resolving all of the addresses from an initial state (or something that invalidates the address cache, such as a STP TCN) may take over 6 seconds. Similarly, the lower the size of the queue, the higher the likelihood of an attack being able to knock out legitimate traffic (but less memory utilization on the router).

7.3. NDP Protocol Gratuitous NA

Per RFC 4861, section 7.2.5 and 7.2.6 [RFC4861] requires that unsolicited neighbor advertisements result in the receiver setting it's neighbor cache entry to STALE, kicking off the resolution of the neighbor using neighbor solicitation. If the link layer address in an unsolicited neighbor advertisement matches that of the existing ND cache entry, routers SHOULD retain the existing entry updating it's status with regards to LRU retention policy.

Hosts MAY be configured to send unsolicited Neighbor advertisement at a rate set at the discretion of the operators. The rate SHOULD be appropriate to the sizing of ND cache parameters and the host count on the subnet. An unsolicited NA rate parameter MUST NOT be enabled by default. The unsolicted rate interval as interpreted by hosts must jitter the value for the interval between transmissions. Hosts receiving a neighbor solicitation requests from a router following each of three subsequent gratuitous NA intervals MUST revert to RFC

4861 behavior.

Implementation of new behavior for unsolicited neighbor advertisement would make it possible under appropriate circumstances to greatly reduce the dependence on the neighbor solicitation process for retaining existing ND cache entries.

This may impact the detection of one-way reachability.

It is understood that this section may need to be moved into a separate document -- it is (currently) provided here for discussion purposes.

7.4. ND cache priming and refresh

With all of the above recommendations implemented, it should be possible to survive a "scan attack" with very little impact to the network, however, adding new hosts to the network (and the sending of traffic to them) may still be negatively impacted. Traffic to those new hosts would have to go through the unknown Neighbor Resolution queue, which is where the attack traffic would end up as well. A solution to this would be that any new host that joins the network would "announce" itself, and be added to the cache, therefore not requiring packets destined to it to go through the unknown NDP queue. This could be done by sending a ping packet to the all-routers multicast address, which would then trigger the router's own neighbor resolution process, which should be in a different queue than other packets.

All attempts should be made to keep these addresses in cache, since any eviction of legitimate hosts from the cache could potentially place resolutions for them into the same queue as the attack traffic. At present, [RFC4861] states that there should be MAX_UNICAST_SOLICIT (3) attempts, RETRANS_TIMER 1 second apart, so if there is an interruption in the network or control plane processing for longer than 3 seconds during the refresh, the entry would be evicted from the ND Cache. Any network event which takes longer than 3 seconds to converge (UDLD, STP, etc may take 30+ seconds) while under an attack, would result in ND cache eviction. If an entry is evicted during a scan, connectivity could be lost for an extended period of time.

NDP refresh timers could be revised as suggested in draft-nordmark-6man-impatient-nud-00 [1] and SHOULD have a configurable value for MAX_UNICAST_SOLICIT and RETRANS_TIMER, and include capabilities for binary/exponential backoff.

A suggested algorithm, which retains backward compatibility with [RFC4861] is: operator configurable values for MAX_UNICAST_SOLICIT,

RETRANS_TIMER, and a way to set adaptive back-off multiple, similar to ipv4 -- call it BACKOFF_MULTIPLE), so that we could implement:

```
next_retrans =  
($BACKOFF_MULTIPLE^$solicit_attempt_num)*$RETRANS_TIMER + jittered  
value.
```

The recommended behavior is to have 5 attempts, with timing spacing of 0 (initial request), 1 second later, 3 seconds later, then 9, and then 27, which represents:

```
MAX_UNICAST_SOLICIT=5
```

```
RETRANS_TIMER=1 (default)
```

```
BACKOFF_MULTIPLE=3
```

If BACKOFF_MULTIPLE=1 (which should be the default value), and MAX_UNICAST_SOLICIT=3, you would get the backwards-compatible RFC behavior, but operators should be able to adjust the values as necessary to insure that they are sufficiently aggressive about retaining ND entries in cache.

An Implementation following this algorithm would if the request was not answered at first due for example to a transitory condition, retry immediately, and then back off for progressively longer periods. This would allow for a reasonably fast resolution time when the transitory condition clears.

8. IANA Considerations

No IANA resources or consideration are requested in this draft.

9. Security Considerations

This document outlines mitigation options that operators can use to protect themselves from Denial of Service attacks. Implementation advice to router vendors aimed at ameliorating known problems carries the risk of previously unforeseen consequences. It is not believed that these techniques create additional security or DOS exposure

10. Acknowledgements

The authors would like to thank Ron Bonica, Troy Bonin, John Jason Brzozowski, Randy Bush, Vint Cerf, Jason Fesler Erik Kline, Jared

Mauch, Chris Morrow and Suran De Silva. Special thanks to Thomas Narten for detailed review and (even more so) for providing text!

Apologies for anyone we may have missed; it was not intentional.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4398] Josefsson, S., "Storing Certificates in the Domain Name System (DNS)", RFC 4398, March 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [RFC6164] Kohno, M., Nitzan, B., Bush, R., Matsuzaki, Y., Colitti, L., and T. Narten, "Using 127-Bit IPv6 Prefixes on Inter-Router Links", RFC 6164, April 2011.

11.2. Informative References

- [RFC4255] Schlyter, J. and W. Griffin, "Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints", RFC 4255, January 2006.

URIs

- [1] <<http://tools.ietf.org/html/draft-nordmark-6man-impatient-nud-00>>

Appendix A. Text goes here.

TBD

Authors' Addresses

Warren Kumari
Google

Email: warren@kumari.net

Igor
Yahoo!
45 W 18th St
New York, NY
USA

Email: igor@yahoo-inc.com

Joel
Zynga
111 Evelyn
Sunnyvale, CA
USA

Email: jjaeggli@zynga.com

6man Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 30, 2011

A. Matsumoto
T. Fujisaki
J. Kato
NTT
T. Chown
University of Southampton
June 28, 2011

Distributing Address Selection Policy using DHCPv6
draft-ietf-6man-addr-select-opt-01.txt

Abstract

RFC 3484 defines default address selection mechanisms for IPv6 that allow nodes to select appropriate address when faced with multiple source and/or destination addresses to choose between. The RFC allowed for the future definition of methods to administratively configure the address selection policy information. This document defines a new DHCPv6 option for such configuration, allowing a site administrator to distribute address selection policy, and thus control the address selection behavior of nodes in their site. While RFC 3484 is in the process of being updated, with a revised default policy table, that table may not suit every scenario, and thus the DHCPv6 option defined in this text may be used to override that policy where desired.

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 December 30, 2011.

Copyright Notice

Copyright (c) 2011 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

1. Introduction

RFC 3484 [RFC3484] describes default algorithms for selecting an address when a node has multiple destination and/or source addresses to choose between by using an address selection policy. In Section 2 of RFC 3484, it is suggested that the default policy table may be administratively configured to suit the specific needs of a site. This text defines a new DHCPv6 option for such configuration.

Some problems have been identified with the default address selection policy detailed in RFC 3484 [RFC5220], and as a result the RFC is in the process of being updated, as per [I-D.ietf-6man-rfc3484-revise]. While this update provides a better default address selection policy, it is unlikely that such a default will suit all scenarios, and thus mechanisms to control the source address selection policy will be necessary. Requirements for those mechanisms are described in [RFC5221], while solutions are discussed in [I-D.ietf-6man-addr-select-sol] and [I-D.ietf-6man-addr-select-considerations]. Those documents have helped shape the improvements in [I-D.ietf-6man-rfc3484-revise] as well as the DHCPv6 option defined here.

1.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.2. Terminology

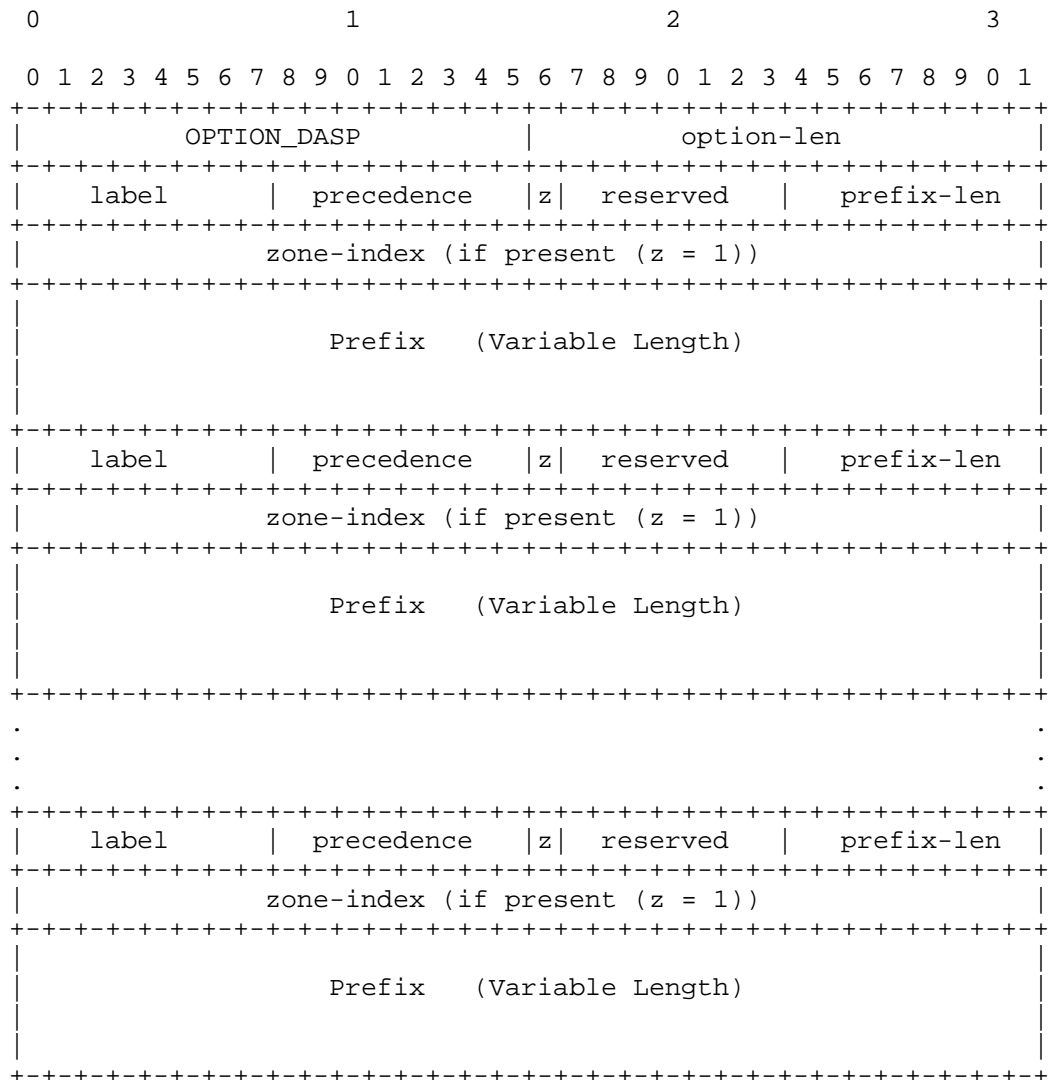
This document uses the terminology defined in [RFC2460] and the DHCPv6 specification defined in [RFC3315]

2. Address Selection Policy Option

The Address Selection Policy Option provides the policy table for address selection rules as described in RFC 3484 and updated in [I-D.ietf-6man-rfc3484-revise].

Each end node is expected to configure its policy table, as described in RFC 3484, using the Address Selection Policy option information as described in the section below on processing the option.

The format of the Address Selection Policy option is given below:



[Fig. 1]

Fields:

option-code: OPTION_DASP (TBD)

option-len: The total length of the label fields, precedence fields, zone-index fields, prefix-len fields, and prefix fields in octets.

label: An 8-bit unsigned integer; this value is used to make a combination of source address prefixes and destination address prefixes.

precedence: An 8-bit unsigned integer; this value is used for sorting destination addresses.

z bit: 'zone-index' bit. If z bit is set to 1, 32 bit zone-index value is included right after the "prefix-len" field, and "Prefix" value continues after the "zone-index" field. If z bit is 0, "Prefix" value continues right after the "prefix-len" value.

reserved: 6-bit reserved field. Initialized to zero by sender, and ignored by receiver.

zone-index: If the z-bit is set to 1, this field is inserted between "prefix-len" field and "Prefix" field. The zone-index field is an 32-bit unsigned integer and used to specify zones for scoped addresses. This bit length is defined in RFC3493 [RFC3493] as 'scope ID'.

prefix-len: An 8-bit unsigned integer; the number of leading bits in the prefix that are valid. The value ranges from 0 to 128. The Prefix field is 0, 4, 8, 12, or 16 octets, depending on the length.

Prefix: A variable-length field containing an IP address or the prefix of an IP address. An IPv4-mapped address [RFC4291] must be used to represent an IPv4 address as a prefix value.

3. Appearance of this Option

The Address Selection Policy option MUST NOT appear in any messages other than the following ones: Solicit, Advertise, Request, Renew, Rebind, Information-Request, and Reply.

4. Processing the Address Selection Policy Option

This section describes how to process received Address Selection Policy Options at the DHCPv6 client.

This option's concept is to serve as a hint for a node about how to behave in the network. So, basically, it should be up to the node's administrator how to make use of or even ignore the received policy information.

However, we need to define the default behavior of the receiving node in order to reduce operational complexity.

4.1. Handling the local policy table

RFC3484 defines the default policy for the policy table. Also, a user is usually able to configure the policy table to satisfy his requirement.

The client node SHOULD provide the following choices:

- a) It receives distributed policy table, and replaces the existing policy tables with that.
- b) It preserves the default policy table, or manually configured policy.

4.2. Processing multiple received policy tables

The policy table is node-global information by its nature. So, the node cannot use multiple received policy tables at the same time.

It should be noted that adopting a received policy table as the node-global information can cause security problems, such as DOS attack, and leak of privacy information.

Moreover, it also should be noted that, when a node is single-homed and has only one upstream line, adopting a received policy table does not degrade the security level.

Under the above assumptions, we specify how to handle multiple received policy tables below.

A node MAY use OPTION_DASP in any of the following two cases:

- 1: The address selection option is delivered across a secure, trusted channel.
- 2: The address selection option is not secured, but the node is single-homed.

In other cases the node **MUST NOT** use `OPTION_DASP` unless the node is specifically configured to do so.

5. Implementation Considerations

- o The value 'label' is passed as an unsigned integer, but there is no special meaning for the value, that is whether it is a large or small number. It is used to select a preferred source address prefix corresponding to a destination address prefix by matching the same label value within the DHCP message. DHCPv6 clients need to convert this label to a representation specified by each implementation (e.g., string).
- o Currently, the label and precedence values are defined as 8-bit unsigned integers. In almost all cases, this value will be enough.
- o The maximum number of address selection rules that may be conveyed in one DHCPv6 message depends on the prefix length of each rule and the maximum DHCPv6 message size defined in RFC 3315. It is possible to carry over 3,000 rules in one DHCPv6 message (maximum UDP message size), but the usual number would be much smaller, e.g. the default policy table defined in RFC 3484 contains 5 rules.
- o Since the number of selection rules could be large, an administrator configuring the policy to be distributed should consider the resulting DHCPv6 message size.

6. Security Considerations

A rogue DHCPv6 server could issue bogus address selection policies to a client. This might lead to incorrect address selection by the client, and the affected packets might be blocked at an outgoing ISP because of ingress filtering. Alternatively, an IPv6 transition mechanism might be preferred over native IPv6, even if it is available.

To guard against such attacks, both DHCP clients and servers **SHOULD** use DHCP authentication, as described in section 21 of RFC 3315,

"Authentication of DHCP messages."

7. IANA Considerations

IANA is requested to assign option codes to OPTION_DASP from the option-code space as defined in section "DHCPv6 Options" of RFC 3315.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

8.2. Informative References

- [I-D.ietf-6man-addr-select-considerations]
Chown, T., "Considerations for IPv6 Address Selection Policy Changes",
draft-ietf-6man-addr-select-considerations-03 (work in progress), March 2011.
- [I-D.ietf-6man-addr-select-sol]
Matsumoto, A., Fujisaki, T., and R. Hiromi, "Solution approaches for address-selection problems",
draft-ietf-6man-addr-select-sol-03 (work in progress), March 2010.
- [I-D.ietf-6man-rfc3484-revise]
Matsumoto, A., Kato, J., and T. Fujisaki, "Update to RFC 3484 Default Address Selection for IPv6",
draft-ietf-6man-rfc3484-revise-03 (work in progress), June 2011.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC3493] Gilligan, R., Thomson, S., Bound, J., McCann, J., and W. Stevens, "Basic Socket Interface Extensions for IPv6",

RFC 3493, February 2003.

- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.
- [RFC5220] Matsumoto, A., Fujisaki, T., Hiromi, R., and K. Kanayama, "Problem Statement for Default Address Selection in Multi-Prefix Environments: Operational Issues of RFC 3484 Default Rules", RFC 5220, July 2008.
- [RFC5221] Matsumoto, A., Fujisaki, T., Hiromi, R., and K. Kanayama, "Requirements for Address Selection Mechanisms", RFC 5221, July 2008.

Appendix A. Past Discussion

- o The 'zone index' value is used to specify a particular zone for scoped addresses. This can be used effectively to control address selection in the site scope (e.g., to tell a node to use a specified source address corresponding to a site-scoped multicast address). However, in some cases such as a link-local scope address, the value specifying one zone is only meaningful locally within that node. There might be some cases where the administrator knows which clients are on the network and wants specific interfaces to be used though. However, in general case, it is hard to use this value.
- o Since we got a comment that some implementations use 32-bit integers for zone index value, we extended the bit length of the 'zone index' field. However, as described above, there might be few cases to specify 'zone index' in policy distribution, we defined this field as optional, controlled by a flag.
- o There may be some demands to control the use of special address types such as the temporary addresses described in RFC4941 [RFC4941], address assigned by DHCPv6 and so on. (e.g., informing not to use a temporary address when it communicate within the an organization's network). It is possible to indicate the type of addresses using reserved field value.

Authors' Addresses

Arifumi Matsumoto
NTT SI Lab
3-9-11 Midori-Cho
Musashino-shi, Tokyo 180-8585
Japan

Phone: +81 422 59 3334
Email: arifumi@nttv6.net

Tomohiro Fujisaki
NTT PF Lab
3-9-11 Midori-Cho
Musashino-shi, Tokyo 180-8585
Japan

Phone: +81 422 59 7351
Email: fujisaki@nttv6.net

Jun-ya Kato
NTT SI Lab
3-9-11 Midori-Cho
Musashino-shi, Tokyo 180-8585
Japan

Phone: +81 422 59 2939
Email: kato@syce.net

Tim Chown
University of Southampton
Southampton, Hampshire SO17 1BJ
United Kingdom

Email: tjc@ecs.soton.ac.uk

6man Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 22, 2011

F. Costa
J-M. Combes
X. Pournard
France Telecom Orange
H. Li
Huawei Technologies
June 20, 2011

Duplicate Address Detection Proxy
draft-ietf-6man-dad-proxy-01

Abstract

The document describes a mechanism allowing the use of Duplicate Address Detection (DAD) by IPv6 nodes in a point-to-multipoint architecture with "split-horizon" forwarding scheme. Based on the DAD signalling, the first hop router stores in a Binding Table all known IPv6 addresses used on a point-to-multipoint domain (e.g. VLAN). When a node performs DAD for an address already used by another node, the first hop router replies instead of this last one.

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 December 22, 2011.

Copyright Notice

Copyright (c) 2011 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.

Table of Contents

1. Introduction	3
1.1. Requirements Language	3
2. Background	3
3. Why existing IETF solutions are not sufficient?	4
3.1. Duplicate Address Detection	5
3.2. Neighbor Discovery Proxy	5
3.3. 6LoWPAN Neighbor Discovery	5
3.4. IPv6 Mobility Manager	6
4. Duplicate Address Detection Proxy (DAD-Proxy) specifications	6
4.1. DAD-Proxy Data structure	6
4.2. DAD-Proxy mechanism	6
4.2.1. No entry exists for the tentative address	7
4.2.2. An entry already exists for the tentative address	7
4.2.3. Confirmation of reachability to check the validity of the conflict	8
5. IANA Considerations	10
6. Security Considerations	10
6.1. Interoperability with SEND	10
6.2. IP source address spoofing protection	11
7. Acknowledgments	11
8. References	11
8.1. Normative References	11
8.2. Informative References	11
Appendix A. Open issues	12
Authors' Addresses	12

1. Introduction

This document explains why Duplicate Address Detection (DAD) mechanism [RFC4862] cannot be used in a point-to-multipoint architecture with "split-horizon" forwarding scheme. One of the main reasons is that, because of this forwarding scheme, IPv6 nodes on the same point-to-multipoint domain cannot have direct communication: any communication between them must go through the first hop router of the same domain.

This document also specifies a function called DAD proxy allowing the use of DAD by the nodes on the same point-to-multipoint domain with "split-horizon" forwarding scheme. It only impacts the first hop router and it doesn't need modifications on the other IPv6 nodes. This mechanism is fully effective if all the nodes of a point-to-multipoint domain (except the DAD proxy itself) perform DAD. However, if it is necessary to cover the scenarios where this assumption is not met, additional solutions could be defined in the future that work in conjunction with the mechanism described here.

It is assumed in this document that Link-layer addresses on a point-to-multipoint domain are unique from the first hop router's point of view (e.g. in an untrusted Ethernet architecture this assumption can be guaranteed thanks to mechanisms such as "MAC Address Translation" performed by an aggregation device between IPv6 nodes and the first hop router).

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Background

Terminology in this document follows that in Neighbor Discovery for IP version 6 (IPv6) document [RFC4861] and IPv6 Stateless Address Autoconfiguration document [RFC4862]. In addition, this section defines additional terms related to DSL and Fiber access architectures, which are an important case where the solution described in this document can be used:

Customer Premises Equipment (CPE)

The first IPv6 node in a customer's network.

Access Node (AN)

The first aggregation point in the public access network. It is considered as a L2 bridge in this document.

Broadband Network Gateway (BNG)

The first hop router from the CPE's point of view.

VLAN N:1 architecture

A point-to-multipoint architecture where many CPEs are connected to the same VLAN. The CPEs may be connected on the same or different Access Nodes.

split-horizon model

A forwarding scheme where CPEs cannot have direct layer 2 communications between them (i.e. IP flows must be forwarded through the BNG via routing).

The following figure shows where are the different entities defined above.

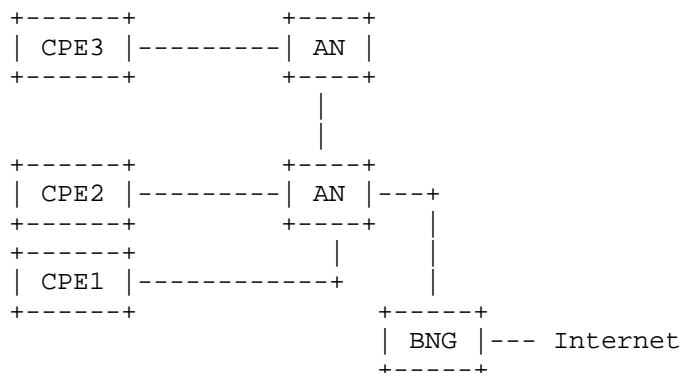


Figure 1: DSL and Fiber access Architecture

3. Why existing IETF solutions are not sufficient?

In a DSL or Fiber access architecture depicted in Figure 1, CPE1,2,3 and the BNG are IPv6 nodes, while AN is a L2 bridge providing connectivity between the BNG and each CPE. The AN enforces a split-horizon model so that CPEs can only send and receive frames (e.g. Ethernet frames) to and from the BNG but not to each other. That said, the BNG is on a same link with all CPE, but one CPE is not on a same link with any other CPE.

3.1. Duplicate Address Detection

Duplicate Address Detection (DAD) [RFC4862] is performed when an IPv6 node verifies the uniqueness of a tentative IPv6 address. This node sends a Neighbor Solicitation (NS) message with the IP destination set to solicited-node multicast address of the tentative address. This NS message is multicasted to other nodes on a same link. When the tentative address is already used on the link by another node, this last one replies with a Neighbor Advertisement (NA) message to inform the first node. So when performing DAD, a node expects the NS messages are received by other nodes.

However, in a point-to-multipoint network with split-horizon forwarding scheme implemented in the AN, the CPEs are prevented from talking to each other directly. All packets sent out from a CPE would be forwarded by AN only to the BNG but not to any other CPE. That said, NS messages sent by a certain CPE will be received only by the BNG and will not reach other CPEs. So, other CPEs have no idea that a certain IPv6 address is used by another CPE. That means, in a network with split-horizon, DAD per [RFC4862] can't work properly without an additional helper.

3.2. Neighbor Discovery Proxy

Neighbor Discovery (ND) Proxy [RFC4389] is designed for forwarding ND messages between different IP links where the subnet prefix is the same. A ND Proxy function on a bridge ensures that packets between nodes on different segments can be received by this function and have the correct link-layer address type on each segment. When the ND proxy receives a multicast ND message, it forwards it to all other interfaces on a same link.

In DSL or Fiber networks, when AN, acting as a ND Proxy, receives a ND message from a CPE, it will forward it to the BNG but none of other CPEs, as only the BNG is on the same link with the CPE. Hence, implementing ND Proxy on AN would not help a CPE acknowledge link-local addresses used by other CPEs.

As the BNG must not forward link-local scoped messages sent from a CPE to other CPEs, ND Proxy cannot be implemented in the BNG.

3.3. 6LoWPAN Neighbor Discovery

[I-D.ietf-6lowpan-nd] defines an optional modification of DAD for a 6LoWPAN. When a 6LoWPAN node wants to configure an IPv6 address, it registers that address with one or more of its default router using the Address Registration option (ARO). If this address is already owned by another node, the router informs the 6LoWPAN node this

address cannot be configured.

A problem for this mechanism is that it requires modifications in hosts in order to support the Address Registration option.

3.4. IPv6 Mobility Manager

According to [RFC3775], a home agent acts as a proxy for mobile nodes when these last ones are away from the home network: the home agent defends an mobile node's home address by replying to NS messages with NA messages.

There is a problem for this mechanism if it is applied in a DSL or Fiber public access network. Operators of such networks require a NA message is only received by the sender of the corresponding NS message, for security and scalability reasons. However, the home agent per [RFC3775] multicasts NA messages on the home link and all nodes on this link will receive these NA messages. This shortcoming prevents this mechanism being deployed in DSL or Fiber access networks directly.

4. Duplicate Address Detection Proxy (DAD-Proxy) specifications

4.1. DAD-Proxy Data structure

A BNG needs to store in a Binding Table information related to the IPv6 addresses generated by any CPE. This must be done per point to multipoint domain (e.g. per Ethernet VLAN). Each entry in this Binding Table MUST contain the following fields:

- o IPv6 Address
- o Link-layer Address

For security or performances reasons, it must be possible to limit the number of IPv6 Addresses per Link-layer Address (possibly, but not necessarily, to 1).

4.2. DAD-Proxy mechanism

When a CPE performs DAD, as specified in [RFC4862], it sends a Neighbor Solicitation (NS) message, with the unspecified address as source address, in order to check if a tentative address is already in use on the link. The BNG receives this message and MUST perform actions depending on the information in the Binding Table.

4.2.1. No entry exists for the tentative address

When there is no entry for the tentative address, the BNG MUST create one with following information:

- o IPv6 Address Field set to the tentative address in the NS message.
- o Link-layer Address Field set to the Link-layer source address in the Link-layer Header of the NS message.

The BNG MUST NOT reply to the CPE or forward the NS message.

4.2.2. An entry already exists for the tentative address

When there is an entry for the tentative address, the BNG MUST check the following conditions:

- o The address in the Target Address Field in the NS message is equal to the address in the IPv6 Address Field in the entry.
- o The source address of the IPv6 Header in the NS message is equal to the unspecified address.

When these conditions are met and the source address of the Link-Layer Header in the NS message is equal to the address in the Link-Layer Address Field in the entry, that means the CPE is still performing DAD for this address. The BNG MUST NOT reply to the CPE or forward the NS message.

When these conditions are met and the source address of the Link-Layer Header in the NS message is not equal to the address in the Link-Layer Address Field in the entry, that means possibly another CPE performs DAD for an already owned address. The BNG then has to verify whether there is a real conflict by checking if the CPE whose IPv6 address is in the entry is still connected. In the following, we will call IPv6-CPE1 the IPv6 address of the existing entry, Link-layer-CPE1 the Link-layer address of that entry and Link-layer-CPE2 the Link-layer address of the CPE which is performing DAD, which is different from Link-layer-CPE1.

The BNG MUST check if the potential address conflict is real. In particular:

- o If IPv6-CPE1 is in the Neighbor Cache and it is associated with Link-layer-CPE1, the reachability of IPv6-CPE1 MUST be confirmed as explained in Section 4.2.3.

- o If IPv6-CPE1 is in the Neighbor Cache, but it is associated with another Link-layer address than Link-layer-CPE1, that means that there is possibly a conflict with another CPE, but that CPE did not perform DAD. This situation is out of the scope of this document, since one assumption made above is that all the nodes of a point-to-multipoint domain (except the DAD proxy itself) perform DAD. This case could be covered in the future by additional solutions that work in conjunction with the DAD proxy.
- o If IPv6-CPE1 is not in the Neighbor Cache, then the BNG MUST create a new entry based on the information of the entry in the Binding Table. This step is necessary in order to trigger the reachability check as explained in Section 4.2.3. The entry in the Neighbor Cache MUST be created based on the algorithm defined in section 7.3.3 of [RFC4861], in particular by considering the case as if a packet other than a solicited Neighbor Advertisement was received from IPv6-CPE1. That means that the new entry of the Neighbor Cache MUST contain the following information:

- * IPv6 address: IPv6-CPE1
- * Link-layer address: Link-layer-CPE1
- * State: STALE

Then the reachability of IPv6-CPE1 MUST be confirmed as soon as possible following the procedure explained in section 4.2.3.

4.2.3. Confirmation of reachability to check the validity of the conflict

Given that the IPv6-CPE1 is in an entry of the Neighbor Cache, the reachability of IPv6-CPE1 is checked by using the NUD (Neighbor Unreachability Detection) mechanism described in section 7.3.1 of [RFC4861]. This mechanism MUST be triggered as if a packet has to be sent to IPv6-CPE1. Note that in some cases this mechanism does not do anything, for instance if the state of the entry is REACHABLE and a positive confirmation was received recently that the forward path to the IPv6-CPE1 was functioning properly (see RFC 4861 for more details).

Next, the behavior of the BNG depends on the result of the NUD process, as explained in the following sections.

4.2.3.1. The result of the NUD process is negative

If the result of the NUD process is negative (i.e. if this process removes IPv6-CPE1 from the Neighbor Cache), that means that the

potential conflict is not real.

The conflicting entry in the Binding Table (Link-layer-CPE1) is deleted and it is replaced by a new entry with the same IPv6 address, but the Link-layer address of the CPE which is performing DAD (Link-layer-CPE2), as explained in Section 4.2.1.

4.2.3.2. The result of the NUD process is positive

If the result of the NUD process is positive (i.e. if after this process the state of IPv6-CPE1 is REACHABLE), that means that the potential conflict is real.

As shown in Figure 2, the BNG MUST reply to CPE that is performing DAD (CPE2 in Figure 1) with a NA message which has the following format:

Layer 2 Header Fields:

Source Address

The Link-layer address of the interface on which the BNG received the NS message.

Destination Address

The source address in the Layer 2 Header of the NS message received by the BNG (i.e. Link-layer-CPE2)

IPv6 Header Fields:

Source Address

An address assigned to the interface from which the advertisement is sent.

Destination Address

The all-nodes multicast address.

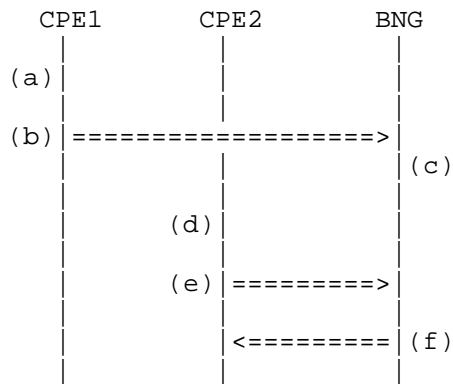
ICMPv6 Fields:

Target Address

The tentative address already used (i.e. IPv6-CPE1).

Target Link-layer address

The Link-layer address of the interface on which the BNG received the NS message.



- (a) CPE1 generated a tentative address
- (b) CPE1 performs DAD for this one
- (c) BNG updates its Binding Table
- (d) CPE2 generates a same tentative address
- (e) CPE2 performs DAD for this one
- (f) BNG informs CPE2 that DAD fails

Figure 2

The BNG and the CPE MUST support the Unicast Transmission on Link-layer of IPv6 Multicast Messages [RFC6085], to be able, respectively, to generate and to process such a packet format.

5. IANA Considerations

No new options or messages are defined in this document.

6. Security Considerations

6.1. Interoperability with SEND

If SEcure Neighbor Discovery (SEND) [RFC3971] is used, the mechanism specified in this document may break the security. Indeed, if an entry already exists and the BNG has to send a reply (cf. Section 4.2.2), the BNG doesn't own the private key(s) associated with to the Cryptographically Generated Addresses (CGA) [RFC3972] to correctly sign the proxied ND messages [RFC5909].

To keep the same level of security, Secure Proxy ND Support for SEND [I-D.ietf-csi-proxy-send] SHOULD be used and implemented on the BNG and the CPEs.

6.2. IP source address spoofing protection

To ensure a protection against IP source address spoofing in data packets, this proposal may be used in combinaison with Source Address Validation Improvement (SAVI) mechanisms [I-D.ietf-savi-fcfs] [I-D.ietf-savi-send].

7. Acknowledgments

TbD

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [RFC6085] Gundavelli, S., Townsley, M., Troan, O., and W. Dec, "Address Mapping of IPv6 Multicast Packets on Ethernet", RFC 6085, January 2011.

8.2. Informative References

- [I-D.ietf-6lowpan-nd] Shelby, Z., Chakrabarti, S., and E. Nordmark, "Neighbor Discovery Optimization for Low Power and Lossy Networks (6LoWPAN)", draft-ietf-6lowpan-nd-17 (work in progress), June 2011.
- [I-D.ietf-csi-proxy-send] Krishnan, S., Laganier, J., Bonola, M., and A. Garcia-Martinez, "Secure Proxy ND Support for SEND", draft-ietf-csi-proxy-send-05 (work in progress), May 2010.
- [I-D.ietf-savi-fcfs] Nordmark, E., Bagnulo, M., and E. Levy-Abegnoli, "FCFS SAVI: First-Come First-Serve Source-Address Validation for Locally Assigned IPv6 Addresses", draft-ietf-savi-fcfs-09

(work in progress), April 2011.

[I-D.ietf-savi-send]

Bagnulo, M. and A. Garcia-Martinez, "SEND-based Source-Address Validation Implementation", draft-ietf-savi-send-05 (work in progress), April 2011.

[RFC3775] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", June 2004.

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.

[RFC4389] Thaler, D., Talwar, M., and C. Patel, "Neighbor Discovery Proxies", RFC 4389, April 2006.

[RFC5909] Combes, J-M., Krishnan, S., and G. Daley, "Securing Neighbor Discovery Proxy: Problem Statement", RFC 5909, July 2010.

Appendix A. Open issues

- o What happens when the BNG receives a NA message with O-bit set to 1 (e.g. the Link-Layer address of the CPE has changed)?

Authors' Addresses

Fabio Costa
France Telecom Orange
38 rue du General Leclerc
92794 Issy-les-Moulineaux Cedex 9
France

Email: fabio.costa@orange-ftgroup.com

Jean-Michel Combes
France Telecom Orange
38 rue du General Leclerc
92794 Issy-les-Moulineaux Cedex 9
France

Email: jeanmichel.combes@orange-ftgroup.com

Xavier Pournard
France Telecom Orange
2 avenue Pierre Marzin
22300 Lannion
France

Email: xavier.pournard@orange-ftgroup.com

Hongyu Li
Huawei Technologies
Huawei Industrial Base
Shenzhen
China

Email: lihy@huawei.com

6MAN WG
Internet-Draft
Expires: January 8, 2012

E. Nordmark
Cisco Systems, Inc.
I. Gashinsky
Yahoo!
July 7, 2011

Neighbor Unreachability Detection is too impatient
draft-nordmark-6man-impatient-nud-01.txt

Abstract

IPv6 Neighbor Discovery includes Neighbor Unreachability Detection. That function is very useful when a host has an alternative, for instance multiple default routers, since it allows the host to switch to the alternative in short time. This time is 3 seconds after the node starts probing. However, if there are no alternatives, this is far too impatient. This document proposes an approach where an implementation can choose the timeout behavior to be different based on whether or not there are alternatives.

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 8, 2012.

Copyright Notice

Copyright (c) 2011 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.

Table of Contents

1. Introduction	3
2. Proposed Remedy	3
3. Acknowledgements	5
4. Security Considerations	6
5. IANA Considerations	6
6. References	6
6.1. Normative References	6
6.2. Informative References	6
Authors' Addresses	6

1. Introduction

IPv6 Neighbor Discovery [RFC4861] includes Neighbor Unreachability Detection, which detects when a neighbor is no longer reachable. The timeouts specified are very short (three transmissions spaced one second apart). That can be appropriate when there are alternative paths the packet can be sent. For example, if a host has multiple default routers in its Default Router List, or if the host has a Neighbor Cache Entry (NCE) created by a Redirect message. The effect of NUD reporting a failure in those cases is that the host will try the alternative; the next router in the Default Router List, or discard the NCE which will also send using a different router.

For that reason the timeouts were chosen to be short; this ensures that if a default router fails the host can use the next router in less than 45 seconds.

However, where there is no alternative there are several benefits in making NUD try probing for a longer time. One of those benefits is to be more robust against transient failures, such as spanning tree reconvergence and other layer 2 issues that can take many seconds to resolve. Marking the NCE as unreachable in that case causes additional multicast on the network. Assuming there are IP packets to send, the lack of an NCE will result in multicast Neighbor Solicitations every second instead of the unicast Neighbor Solicitations that NUD sends.

As a result IPv6 is operationally more brittle than IPv4. For IPv4 there is no mandatory time limit on the retransmission behavior for ARP [RFC0826] which allows implementors to pick more robust schemes.

The following constant values in [RFC4861] seem to have been made part of IPv6 conformance testing: MAX_MULTICAST_SOLICIT, MAX_UNICAST_SOLICIT, RETRANS_TIMER. While such strict conformance testing seems consistent with the specification, it means that we need to update the standard if we want to allow IPv6 Neighbor Discovery to be as operationally robust as ARP.

Additional motivations for making IPv6 Neighbor Discovery as robust as ARP are covered in [I-D.gashinsky-v6nd-enhance].

2. Proposed Remedy

We can clarify that the giving up after three packets spaced one second apart is only REQUIRED when there is an alternative, such as an additional default route or a redirect.

If implementations transmit more than MAX_*CAST_SOLICIT packets they MAY use binary exponential backoff of the retransmit timer. This is so that if we end up with implementations that try for a very long time we don't end up with a steady background level of retransmissions.

However, even if there is no alternative, we still need to be able to handle the case when the link-layer address of the destination has changed. Thus at some point in time we need to switch to multicast Neighbor Solicitations.

A possible way to describe a node behavior which captures all the cases is to introduce a new, optional, UNREACHABLE state in the conceptual model described in [RFC4861]. A NCE in the UNREACHABLE state retains the link-layer address, and IPv6 packets continue to be sent to that link-layer address. But the Neighbor Solicitations are multicast, using a timeout that follows a binary exponential backoff.

In the places where RFC4861 says to discard/delete the NCE after N probes (Section 7.3, 7.3.3 and Appendix C) we will instead transition to the UNREACHABLE state.

If the Neighbor Cache Entry was created by a redirect, a node MAY delete the NCE instead of changing its state to UNREACHABLE. In any case, the node SHOULD NOT use an NCE created by a Redirect to send packets if that NCE is in unreachable state. Packets should be sent following the next-hop selection algorithm in section XXX which disregards NCEs that are not reachable.

The default router selection in section 6.3.6 says to prefer default routers that are "known to be reachable". For the purposes of that section, if the NCE for the router is in UNREACHABLE state, it is not known to be reachable. Thus the particular text in section 6.3.6 which says "in any state other than INCOMPLETE" needs to be extended to say "in any state other than INCOMPLETE or UNREACHABLE".

Apart from the use of multicast NS instead of unicast NS, and the binary exponential backoff of the timer, the UNREACHABLE state works the same as the current PROBE state.

A node MAY garbage collect a Neighbor Cache Entry as any time as specified in RFC 4861. This does not change with the introduction of the UNREACHABLE state in the conceptual model.

The UNREACHABLE state is conceptual and not a required part of this specification. A node merely needs to satisfy the externally observable behavior of this specification.

There is a non-obvious extension to the state machine description in Appendix C in RFC 4861 in the case for "NA, Solicited=1, Override=0. Different link-layer address than cached". There we need to add "UNREACHABLE" to the current list of "STALE, PROBE, Or DELAY". That is, the NCE would be unchanged. Note that there is no corresponding change necessary to the text in section 7.2.5 since it is phrased using "Otherwise" instead of explicitly listing the three states.

The other state transitions described in Appendix C handle the introduction of the UNREACHABLE state without any change, since they are described using "not INCOMPLETE".

There is also the more obvious change already described above. RFC 4861 has this:

PROBE	Retransmit timeout, N or more retransmissions.	Discard entry	-
-------	--	---------------	---

That needs to be replaced by:

PROBE	Retransmit timeout, N or more retransmissions.	Double timeout Send multicast NS	UNREACHABLE
UNREACHABLE	Retransmit timeout	Double timeout Send multicast NS	UNREACHABLE

The binary exponential backoff SHOULD be clamped at some reasonable maximum retransmit timeout, such as 60 seconds. And if there is no IPv6 packets sent using the UNREACHABLE NCE, then it makes sense to stop the retransmits of the multicast NS until either the NCE is garbage collected, or there are IPv6 packets sent using the NCE. In essence the multicast NS and associated binary exponential backoff can be conditioned on the continued use of the NCE to send IPv6 packets to the recorded link-layer address.

A node MAY unicast the first few Neighbor Solicitation messages while in UNREACHABLE state, but it MUST switch to multicast Neighbor Solicitations. Otherwise it would not detect a link-layer address change for the target.

3. Acknowledgements

The comments from Thomas Narten and Philip Homburg have helped improve this draft.

4. Security Considerations

Relaxing the retransmission behavior for NUD has no impact on security. In particular, it doesn't impact applying Secure Neighbor Discovery [RFC3971].

5. IANA Considerations

This are no IANA considerations for this document.

6. References

6.1. Normative References

- [RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

6.2. Informative References

- [I-D.gashinsky-v6nd-enhance]
Kumari, W., "Operational Neighbor Discovery Problems and Enhancements.", draft-gashinsky-v6nd-enhance-00 (work in progress), June 2011.
- [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, November 1982.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

Authors' Addresses

Erik Nordmark
Cisco Systems, Inc.
510 McCarthy Blvd.
Milpitas, CA, 95035
USA

Phone: +1 408 527 6625
Email: nordmark@cisco.com

Igor Gashinsky
Yahoo!
45 W 18th St
New York, NY
USA

Email: igor@yahoo-inc.com

