ADDRCONF Working Group
INTERNET-DRAFT
<draft-ietf-addrconf-ipv6-auto-01.txt>

Susan Thomson (Bellcore)

March 24, 1995

IPv6 Stateless Address Autoconfiguration

Status of this Memo

This document is a submission to the ADDRCONF Working Group of the Internet Engineering Task Force (IETF). Comments should be submitted to the addrconf@cisco.com mailing list.

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are draft documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a "working draft" or "work in progress."

To learn the current status of any Internet Draft. please check the 1id-abstracts.txt listing contained in the Internet Drafts Shadow Directories on ds.internic.net, nic.nordu.net, ftp.nisc.sri.com or munnari.oz.au.

Abstract

This document specifies stateless address autoconfiguration. A host can form a link-local address autonomously based on information local to the host. A host can form an inter-link scope address in two ways: either autonomously, based on prefixes advertised by routers, or by using the IPv6 version of the Dynamic Host Configuration Protocol(DHCPv6). All mechanisms rely on a host having a token that is unique at least per link. This document specifies how a host forms addresses autonomously. DHCPv6 is described elsewhere.

1. INTRODUCTION

An IPv6 host may have multiple addresses per interface. The addresses can have one of three scopes:

1. a link-local address,

2. a site-local address, and

3. a global address.

All three address scopes can be autoconfigured. A host can autoconfigure a link-local address autonomously. A host can autoconfigure a site-local or global address only when a router or a DHCPv6 server is present on the link.

There is only one way to form a link-local address. On initialization of an interface, a host forms such an address by concatenating a well-known link-local prefix[1] to a token that is unique per link. The definition of the tokens used are link-dependent. For example, in the case of a host attached to an link that uses IEEE 802 addresses, the token is the IEEE 802 address of the interface.

There are two ways to form a site-local or global address. In the first mechanism, a host forms an inter-link scope address by concatenating a network prefix advertised in a Router Advertisement[2,3] to a token that is unique per link. Like the link-local address, the token is defined to be link-layer dependent. This mechanism for forming a site-local or global address is suitable for environments where no administrative control is desired. It is a simple protocol designed for a very specific purpose: to make stateless address configuration very straightforward to use and implement.

The other mechanism available to hosts is to use the IPv6 version of the Dynamic Host Configuration Protocol (DHCPv6). DHCPv6 is a more complex protocol allowing for very flexible address assignment under the control of a system administrator. This protocol typically requires significant system management, including server and database configuration.

The choice of mechanism to use in forming an inter-link scope address is advertised by a router, if present, and this choice is configurable by a system administrator.

This document describes how a host forms a link-local address and one or more site-local or global addresses autonomously. It also specifies how a host determines which mechanism to use to form an inter-link scope address: the autonomous (stateless) approach or DHCPv6. Section 2 describes the router's role in address autoconfiguration and Section 3 the host's role.

2. ROUTER BEHAVIOR

The stateless address autoconfiguration mechanism relies on the router discovery mechanism specified in [2,3] for forming addresses with site-local or global scope. If configured to do so, routers advertise prefix information in periodic Router Advertisements. The prefixes are contained in Prefix-Information extensions of a Router Advertisement. Each Prefix-Information extension indicates whether the prefix can be used for autonomous address autoconfiguration and, if so, for how long the resulting address is valid. Note that the lifetime of the address is defined separately from that of the Router Advertisement itself (other information is advertised in the advertisement which has different lifetime requirements). The extension also explicitly indicates to hosts whether DHCPv6 is required to be used since it is possible that system administrators would like to have hosts autoconfigure addresses autonomously, but also use DHCPv6 to acquire other configuration information besides the address.

Router Advertisement and Solicitation processing is specified in [2] and the message formats in [3].

DISCUSSION: An alternative approach is to advertise address configuration information in a separate advertisement entirely. This would be somewhat cleaner since the lifetime of the advertisement would then be that of the information advertised. On the other hand, having two types of router advertisements would mean that prefix information is advertised redundantly, and in particular, would double traffic on initialisation and on router solicitations.

2.1. Router Configuration Variables

In addition to the configuration variables specified in [2,3], routers MUST also be configurable as follows.

For each of the IPv6 unicast addresses per interface:

Autonomous Flag

If and only if TRUE, the prefix length is to be advertised for the purposes of autonomous address configuration.

Default: TRUE

For each interface:

Administered Flag

If and only if TRUE, the host must autoconfigure other configuration information using DHCPv6. Only valid in extensions with the Autonomous Flag set to TRUE.

Default: FALSE

Address_Advertisement_Interval

The time allowed between sending unsolicited Address Advertisements from the interface, in seconds. The value must not be less than Maximum_Advertisement_Interval of Router Advertisements.

Default: XX

Address_Lifetime

The value to be placed in the Lifetime field of the Prefix_Information extension sent from the interface in seconds. The value must not be less than Address_Advertisement_Interval. This value indicates how long an address formed from the prefix advertised is valid. Only valid in extensions with the Autonomous flag set to TRUE.

Default: 3 * Address_Advertisement_Interval

All routers advertising a given prefix on a link MUST be configured to advertise the same autoconfiguration mode to hosts.

2.2. Processing

A router MUST advertise address autoconfiguration information in a Prefix Information Extension of a Router Advertisement. The values of the Autonomous and Administered flags are advertised along with Address_Lifetime. The address configuration information need not be advertised in each Router Advertisement. It must be sent (almost) periodically in a Router Advertisement at an interval of approximately Address_Advertisement_Interval.

Address configuration information must also be sent in the first few Router Advertisements after startup or enabling of an interface (up to MAX_INITIAL_ADVERTISEMENTS) and in a Router Advertisement that is sent in response to a Router Solicitation.

Address configuration information may also be sent in a Router Advertisement due to actions taken by system management, in particular, when the Address_Lifetime of a prefix is set to zero, e.g. because the link is to be renumbered. In this case, a Prefix-Information extension must be transmitted in a Router Advertisement advertising the appropriate address prefix with the Autonomous Flag set to TRUE and Address_Lifetime set to zero.

3. HOST ADDRESS AUTOCONFIGURATION PROCESSING

3.1. Host Configuration Variables

A host maintains a list of addresses per interface. At a minimum, the list includes the link-local address, which the host can form autonomously whenever an interface is initialised. If a router is attached to the link or DHCPv6 server is available, the list may also include site-local or global addresses formed either from subnet prefixes advertised in Router Advertisements or by making requests using DHCPv6. Addresses may also be manually configured. Note there may be multiple site-local or global addresses autoconfigured per interface.

A host must maintain a list of the following configurable variables per interface:

Address

A valid IPv6 unicast address for this interface

Default: None

Prefix Length

The length of the prefix in bits. Prefix length semantics are specified in [2].

A host must also allow the following variable to be configured per interface:

Perform_Auto_Config

If and only if TRUE, the host must perform address autoconfiguration processing.

Default: TRUE

3.2. Host Initialization Behavior

A host must perform the following autoconfiguration procedure whenever an interface needs to be initialised:

When a host has no address for an interface with Perform_Auto_Config flag set to TRUE, e.g. when a host boots or when an interface is enabled after being disabled, the host forms an address of link-local scope. Appendix A specifies how a host that is attached to a link that uses IEEE 802 addresses forms a link-local address.

Before adding the link-local address as a valid address to the list of addresses for the interface, the host SHOULD verify that the address is indeed unique. The procedure for validating an address is described in Section X. A host SHOULD also validate any manually configured addresses this way too.

The host solicits a Router Advertisement using one or more Router Solicitations, if no Router Advertisements have been heard in the interface. The procedure for sending Router Solicitations is specified in [2].

If no Router Advertisement is heard after sending MAX_SOLICITATIONS and waiting Router_Solicitation_Interval after each as specified in Sending Router Solicitations in [2], the host must determine whether a DHCPv6 server is present and whether any configuration information needs to be acquired. This is to cater for a routerless topology which has a DHCPv6 server. Once a router is added to the network, however, a host MUST use Router Advertisements to determine the autoconfiguration mode in use as described in the section on Router Advertisement Processing.

3.3. Router Advertisement Processing

Router Advertisement processing is specified in [2] and the message format in [3]. In addition to this processing, the host MUST perform the following address configuration processing when a solicited or unsolicited Router Advertisement is received over an interface:

For each Prefix-Information extension in the Router Advertisement: (The format of the Prefix-Information extension as amended by this draft for autoconfiguration purposes is specified in Appendix C):

> The host silently ignores the extension for the purposes of auto-configuration if the Perform_Auto_Config flag for the interface is FALSE.
>
> Otherwise, the host checks the autoconfiguration mode bits.
>
> If only the Autonomous flag is set in the Prefix-Information extension, the host forms or verifies a site-local or global address as specified below.
>
> If both the Autonomous and Administered flags are set in the Prefix-Information extension, the host forms or verifies a site-local or global address as specified below and uses or continues using DHCPv6 for other autoconfiguration.
>
> Otherwise, the host silently ignores the extension for the purposes of autonomous autoconfiguration.

If none of the prefixes advertised in extensions of the Router Advertisement have the Autonomous flag set, then the host uses or continues using DHCPv6 for autoconfiguration.

Note that the above procedure should continue to operate when a system administrator decides to change the autoconfiguration mode from the autonomous mode to DHCPv6, and vice versa. The host should keep track of the current autoconfiguration mode, so that it can detect when there is a change. The system administrator must ensure that, during a changeover, a DHCPv6 server is configured to hand out addresses that are unique per link, particularly with respect to addresses that hosts have configured autonomously and which may not

yet be invalidated. To avoid problems during a changeover, it is recommended that a DHCP server should use exactly the same algorithm to form a host address as that used in the autonomous mode when the prefix is the same. It is also important to ensure that a DHCPv6 server is configured to hand out addresses only to those hosts that it should, since, if a DHCPv6 server is present on a link, hosts may request the server for addresses (even if the network is configured to be in autonomous mode) when Router Advertisements are not heard because the router is down.

For each Prefix-Information extension received over an autoconfigurable interface, the host updates the address list as follows when the Autonomous flag is set:

a) If the prefix advertised matches the prefix of an autoconfigured address already in the list, then set a timer to that of the lifetime specified in the extension. Note there is no timer associated with a link-local address or manually configured address.

b) If the prefix advertised does not match the prefix of an address already in the list, then form an address using this network prefix. Appendix A specifies how to form an address for hosts that have IEEE 802 tokens. The extension is ignored if the prefix is not the right length for forming an address as specified in Appendix A.

   Add this address to the list with a timer set to that of the lifetime specified in the extension.

3.3.1. Address Deprecation and Invalidation

An address is valid until the timer expires.

When the lifetime of an address expires, an address is said to be deprecated. In general, a deprecated address should no longer be used in new communications, but may be used in existing communications.

In particular, the internetworking layer should not select a

deprecated address as a source address in new communications. However, a deprecated address should be allowed to be used as a source address if it is in use by the transport layer in existing communications or it is explicitly requested by an application.

The internetworking layer must continue to accept datagrams destined to a deprecated address. The transport layer may refuse to accept new communications requests to a deprecated address, however.

In addition, a host may send a Remote Redirect[2,3] to correspondents when the source address used in communications is deprecated as long as the host has a valid alternative address. Also, a deprecated address should be removed from the Domain Name System (DNS). This may be done by the host itself, given a DNS dynamic update protocol and sufficient authority, or it may be done on the host's behalf.

The time at which a deprecated address becomes invalid (removed from the list of addresses per interface) is dependent on the storage available for the address list and is thus implementation-dependent. A host should keep a deprecated address until it is no longer in use, i.e. it is no longer being used in current communications such as an existing TCP connection, and it is no longer stored or cached in the Domain Name System. After this point, a deprecated address may be removed from the address list.

If Router Advertisements stop being heard and all autoconfigured inter-link scope addresses become deprecated, then the host must determine whether a DHCPv6 server is available for address autoconfiguration. The host follows the same procedure as described in the initialisation procedure in this case.

3.4. Detecting Duplicate IPv6 Addresses

One of the basic assumptions of the autoconfiguration schemes outlined in this document is that the host token is at least unique per link. Tokens are defined to be link-layer dependent, and the token is the link layer address in most cases. In practice, two hosts on the same link may have the same link layer address because of an assignment error, in which case the resulting IPv6 addresses will not be unique. For this reason, it is prudent to check that the addresses are indeed unique. In IPv6, it is only necessary to check that one of the autoconfigured addresses is unique since the same token is

used to form all addresses and the prefixes used to form the addresses are all unique (the autoconfiguration procedure should ensure this). It is recommended that the link-local address be the address checked since it is formed once and first, on initialisation.

The procedures use General Solicitations and Advertisements specified in [2,3] as modified below. To validate an address, the node sends a General Solicitation with the source and destination set to that of the address to be checked. The node should specify an appropriate Media-Access extension.

On receiving a General Solicitation with a source address that is the same as the destination address and apparently from itself, a host must respond with a General Advertisement. The General Advertisement is sent to the All-Nodes Multicast Address with intra-link scope. The Media-Access extension from the General Solicitation MUST NOT be retained.

After sending a General Solicitation, the node waits for a period of General_Solicitation_Interval. If a General Advertisement is not received in response to the General Solicitation within the interval, the address is considered to be validated. If a General Advertisement is received with a source address the same as the address being validated, it must cease operation on that interface and indicate an appropriate error.

Note that this mechanism is not completely reliable, and so it is possible that duplicate addresses will still exist. If a duplicate address is discovered, the host will need to be configured with a new token, or if this is not possible, the IPv6 addresses will need to be manually configured.

DISCUSSION: There is a problem with a race condition when two (or more) nodes boot up at the same time. Both will send out a General Solicitation, receive no advertisement and assume all is well. A fix may be to have a node process General Solicitations during the validation stage and flag an error if it sees more than one General Solicitation for an address it is in the process of validating.

DISCUSSION: Should the solicitations be dithered? Note that randomising based on the token (link-layer address) only does not help if the token is not unique.

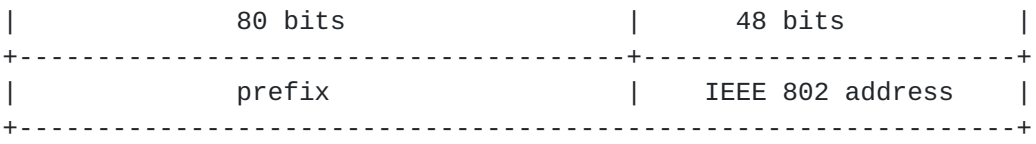
4. SECURITY CONSIDERATIONS

To be completed.

5. APPENDIX A: FORMING AN IPv6 ADDRESS FOR IEEE 802 LINKS

The token for an interface on an IEEE 802 link or any link that uses IEEE 802 addressing, such as FDDI, is the 48-bit IEEE 802 address in canonical format, i.e. the Individual/Group bit is the low-order bit of the furst byte.

A host forms an IPv6 address per link by concatenating an 80-bit prefix with the IEEE 802 address as follows:



In the case of a link-local prefix, the prefix is well-defined[1].

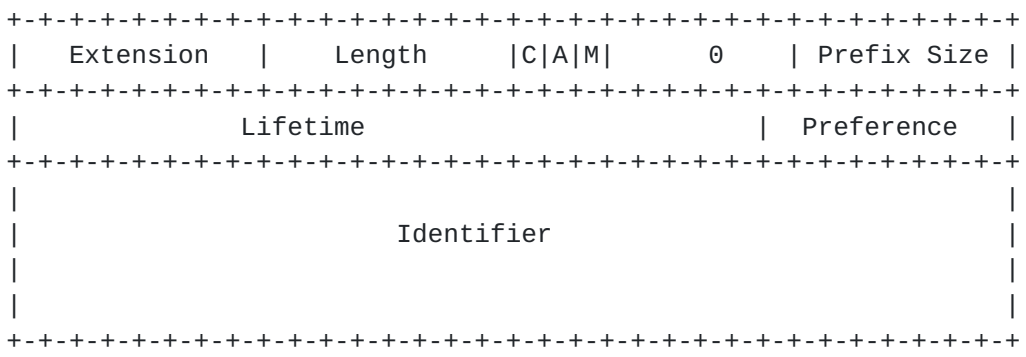
The prefixes of other types of addresses need to be configured.

6. APPENDIX B: UNIQUENESS OF HOST TOKENS

As has been mentioned, one of the basic assumptions of the autoconfiguration scheme outlined in this document is that the host token is at least unique per link, but that tokens may not always be unique, in practice. A host should check that an address is unique using the scheme proposed in this document. Since this is not completely reliable, system administrators may also use DNS to help detect when such a problem occurs since two different hosts will register the same IPv6 address.

Duplicate IPv6 addresses may occur as a result of non-unique tokens in any particular network topology. One particular scenario deserves further mention though. Consider a topology consisting of two links with singly-homed hosts attached to each, a multi-homed host attached to both and no router. In this case, because no router is present, hosts will form link-local addresses only on all interfaces. It is imperative that hosts have interface tokens that are unique across both links. However, this may not be true for two reasons: the links may be of different types and thus the tokens used may not be unique. Also, the token may not be unique if it is defined to be a link layer address and the link-layer address is only defined to be unique per link as is true in some cases. Strictly speaking, we require that host tokens are globally unique to ensure correct operation in these topologies. In practice, link layer addresses are frequently globally unique and so the uniqueness problems in this scenario should not be appreciably worse than in more traditional topologies as described above. If two link-local scope addresses are detected to be the same in this scenario, there are two solutions: one is to make the multihomed host a router, the other is to manually configure the link-local address of an offending host.

7. APPENDIX C: Prefix-Information Extension



Extension	As in [3]
Length	As in [3]
C	As in [3]
A	Autonomous Mode Form an address using prefix of advertised identifier.
M	Administered Mode Use DHCPv6 to autoconfigure other information.
Prefix Size	Number of bits of identifier defining the routing prefix for this link
Preference	As in [3]
Identifier	One of IPv6 unicast addresses of this interface

This extension is found in Router Advertisements.

8. REFERENCES

[1] R. Hinden, "Internet Protocol Version (IPv6) Specification", Internet Draft, March 1995, <draft-ietf-ipngwg-ipv6-addr-arch-01.txt>

[2] W. A. Simpson, "IPv6 Neighbor Discovery -- Processing", Internet Draft, January 1995, <draft-simpson-ipv6-discov-process-02.txt>

[3] W. A. Simpson, "IPv6 Neighbor Discovery -- ICMP Message Formats", Internet Draft, January 1995, <draft-simpson-ipv6-discov-formats-02.txt>

Acknowledgements

The author would like to thank the members of both the IPNG and ADDRCONF working groups for their input. In particular, thanks to Jim Bound, Steve Deering and Bill Simpson.

Author's Addresses

Susan Thomson
Bellcore
445 South Street
Morristown, NJ 07960
U.S.A.

Phone: +1 201-829-4514
Email: set@thumper.bellcore.com