ADDRCONF Working Group Susan Thomson (Bellcore)
INTERNET-DRAFT July 7, 1995
<draft-ietf-addrconf-ipv6-auto-03.txt>

IPv6 Stateless Address Autoconfiguration

Status of this Memo

Abstract

This document specifies how a node configures a link-local address per interface, and how a host configures a list of global or site-local addresses per interface, without any manual configuration. Stateless address autoconfiguration is only one mechanism available to hosts; stateful address configuration is also available. While stateful address configuration is outside the scope of this document, it is specified how hosts determine which configuration method must be used. The document also specifies a protocol for detecting whether an address is a duplicate when it is initially configured.

1. INTRODUCTION

In IPv6, a host has two mechanisms available to form a global or site-local address that require no manual configuration: the stateless method and the stateful method. In the stateless method, no external state is maintained for the purpose of indicating to a host the list of addresses to use for an interface. Rather, a host forms a list of addresses algorithmically by concatenating each of the network prefixes of the attached link to an interface token unique per link. The interface token is defined to be link-dependent and may be the hardware address, for example. In contrast, state is maintained in the stateful address configuration, typically in a server. For example, the IPv6 Dynamic Host Configuration Protocol is an example of stateful address autoconfiguration.

Stateless autoconfiguration is designed to make address configuration very simple to use and implement, and is suitable for environments with simple topologies, e.g. routerless networks, and for environments where system administrative control is not desired, e.g. plug-and-play environments. In contrast, stateful address configuration is designed to support flexible address assignment and is suitable for more sophisticated topologies and for environments where system administrative control is desired, e.g. corporate networks.

Any node can use stateless address autoconfiguration to form a link-local address per interface. A host can use either stateless or stateful autoconfiguration (or both) to configure global or site-local addresses for an interface. The choice of mechanism for configuring global or site-local addresses is itself configurable, and requires no manual configuration per host.

One of the basic assumptions of stateless autoconfiguration is that the token used to form addresses per interface is at least unique per link. However, whatever the type of tokens used, interface tokens are not guaranteed to always be unique in practice because of errors in assignment. Thus, it is possible that IPv6 addresses formed using stateless autoconfiguration are not unique among all nodes on a link. Since duplicate IPv6 addresses are very difficult to track down when they occur, this document also specifies a procedure for detecting duplicate addresses. Note that the algorithm does not only apply to addresses autoconfigured using the stateless mode. It should be used to verify the uniqueness of any address, for example, addresses configured using the stateful mode or manually configured addresses.

This document describes how a node configures a link-local address per interface using stateless address autoconfiguration, and how a host configures global or site-local unicast addresses per interface using stateless address autoconfiguration. Stateful address autoconfiguration is outside the scope of this document. However, the document does specify how a host determines whether to use the stateless mechanism or the stateful mechanism for configuring global or site-local addresses.

The document also describes the algorithm used by a node to detect if an address is a duplicate when initially configured.

2. TERMINOLOGY

node - a device that implements IPv6.

router - a node that forwards IPv6 packets not explicitly addressed to itself.

host - any node that is not a router.

upper layer - a protocol layer immediately above IPv6. Examples are transport protocols such as TCP and UDP, control protocols such as ICMP, routing protocols such as OSPF, and internet or lower-layer protocols being "tunneled" over (i.e., encapsulated in) IPv6 such as IPX, AppleTalk, or IPv6 itself.

link - a communication facility or medium over which nodes can communicate at the link layer, i.e., the layer immediately below IPv6. Examples are Ethernets (simple or bridged); PPP links; X.25, Frame Relay, or ATM networks; and internet (or higher) layer "tunnels", such as tunnels over IPv4 or IPv6 itself.

neighbors - nodes attached to the same link.

interface - a node's attachment to a link.

packet - an IPv6 header plus payload.

link MTU - the maximum transmission unit, i.e., maximum packet size in octets, that can be conveyed in one piece over a link.

target - a node about which address resolution information is sought, or a node which is the new first-hop when being redirected.

address - an IPv6-layer identifier for an interface or a set of interfaces.

unicast address - an identifier for a single interface. A packet sent to a unicast address is delivered to the interface

identified by that address.

anycast address
- an identifier for a set of interfaces (typically belonging to different nodes). A packet sent to an anycast address is delivered to one of the interfaces identified by that address (the "nearest" one, according to the routing protocols' measure of distance).

multicast address
- an identifier for a set of interfaces (typically belonging to different nodes). A packet sent to a multicast address is delivered to all interfaces identified by that address.

link-layer address
- a link-layer identifier for an interface. Examples are IEEE 802 addresses for Ethernet links, E.164 addresses for ISDN links.

link-local address
- an address with a scope that is limited to the locally attached link.

site-local address
- an address with a scope that is limited to the local site.

global address
- an address with unlimited scope.

communication
- any packet exchange between nodes that requires or recommends that the address of each node used in the exchange remain the same for the duration of the packet exchange. Examples are a TCP connection or a UDP request-response.

deprecation lifetime
- indicates the time at which an address should no longer be used as a source address in new communications. The deprecation lifetime must be less than or equal to the invalidation lifetime

of the address.

invalidation lifetime
- indicates the time at which an address no longer identifies an interface or set of interfaces. The invalidation lifetime must be greater than or equal to the deprecation lifetime of the address.

valid address
- an address whose deprecation lifetime has not yet expired.

deprecated address
- an address whose deprecation lifetime has expired, but whose invalidation lifetime has not.

invalid address
- an address whose invalidation lifetime has expired.

interface token
- a link-dependent identifier for an interface that is (at least) unique per link. An example is an IEEE 802 address.

2.1. Requirements

Throughout this document, the words that are used to define the significance of the particular requirements are capitalized. These words are:

MUST
This word or the adjective "REQUIRED" means that the item is an absolute requirement of this specification.

MUST NOT
This phrase means the item is an absolute prohibition of this specification.

SHOULD

This word or the adjective "RECOMMENDED" means that there may exist valid reasons in particular circumstances to ignore this item, but the full impliciations should be understood and the case carefully weighed before choosing a different course.

SHOULD NOT

This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighted before implementing any behavior described with this label.

MAY

This word or the adjective "OPTIONAL" means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example, another vendor may omit the same item.

3. CONCEPTUAL MODEL OF HOST BEHAVIOR

Conceptually, a host maintains three data structures: a list of addresses per interface and two flags. One flag, called the "address mode" flag, indicates whether the stateful protocol is to be used for address configuration (possibly in addition to the stateless protocol). The other flag, called the "other configuration mode" flag, indicates whether the stateful protocol is to be used for configuration of other information besides addresses.

The address list always includes at least one address, the host's link-local address, which is an address that can only be used in communications between nodes attached to the link. In addition, the address list includes any global or site-local addresses that have been manually or automatically configured.

Note that stateless address autoconfiguration applies only to the formation of unicast addresses. A node may, however, have anycast and multicast addresses associated with an interface. In particular, a host must join the all-nodes multicast address on any multicast interface, and the solicited-node multicast address corresponding to each unicast address on any multicast interface.

3.1. Address Configuration

3.1.1. Link-Local Address

On initialization of an interface, a host must form a link-local address by concatenating a well-known link-local prefix[1] to an interface 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 a link that uses IEEE 802 addresses, the token is the 48-bit IEEE 802 link-layer address of the interface. Tokens for a specific link are defined in link-specific IPv6 documents and are outside the scope of this document.

Note that a host is able to autoconfigure a link-local address

completely autonomously. In particular, a host can form a link-local address without a router present on the link.

3.1.2. Global/Site-Local Address

A host forms a new global/site-local address whenever a new prefix is advertised by a router and stateless address configuration is enabled. The address is formed by concatenating the network prefix to the interface token that is unique per link in the same way as a link-local address described above.

The mechanism used to advertise network prefixes for the purposes of address configuration is the same as that used in Neighbor Discovery for the purposes of prefix discovery. A router advertises prefix information periodically and a host uses this information to configure and reconfigure addresses.

3.2. Address Reconfiguration

One of the goals of IPv6 address autoconfiguration is not only to be able to autoconfigure a list of addresses on initialization of an interface, but to be able to change the address list dynamically. Note that the link-local address never changes (except possibly on interface re-initialization). Host addresses may need to change for a number of reasons. For example, if the address assignment scheme is provider-based, hosts may need to change addresses when hosts change provider. Hosts may also need to change addresses when they are disconnected from one link and connected to another. Reconfiguration must not only allow a host to acquire a new address, but must also allow hosts to time-out an old address.

Current networking protocols have not been designed to maintain existing communications during an address change. For example, a TCP connection will no longer work if one of the hosts changes its address in the middle of a connection. Even in a UDP exchange, a host is expected to be able to identify itself using the same address for the length of the exchange.

To minimise disruption caused by an address change, an address is configured with two lifetimes : a deprecation lifetime and an invalidation lifetime. The deprecation lifetime must be less than or equal to the invalidation lifetime. Before expiry of the deprecation lifetime, the address is valid and may be used as the source and destination in any communications. At and after expiry of the deprecation lifetime, but before the invalidation lifetime has expired, the address is considered to be deprecated. A deprecated address can still be used as the source and destination in packets legitimately, but the deprecated address should not be used as a source address in new communications if other valid addresses exist and these addresses have sufficient scope. If no valid addresses of sufficient scope exist, then the deprecated address should still be used. (Note that there will always be at least one valid address in the address list, the link-local address (see below), but this address is only useful for communications on the local link, and thus cannot be used in place of a deprecated address for non-local link communications).

An address becomes invalid when the invalidation lifetime expires. Such an address must not be used as a source address in outgoing communications or accepted as a destination address in incoming communications. An invalid address is removed from the address list of the interface.

Note that the deprecation lifetimes and invalidation lifetimes of the link-local address are set to infinity. Thus, the link-local address is never deprecated.

The intention of the two lifetimes per address is to allow system administrators to specify a graceful transition period during renumbering. The purpose of the deprecation time is to indicate to the host to start using another (presumably longer lasting) address in new communications to minimise the risk of breaking communications when the old one times out. A system administrator should set the deprecation period long enough so that most, if not all, communications have switched over to using the new address by the time the old one becomes invalid. The length of the deprecation period will be environment-dependent as it depends on the expected length of communications.

The fact that addresses have a deprecation lifetime does not need to affect the implementation of applications, i.e. an application is not expected to react when an address it is using becomes deprecated. The purpose of the deprecation lifetime is to help applications or networking software to select a sufficiently long-lasting source

address at the beginning of a new communication. The IP layer is expected to provide a means for upper layers or applications to select the most appropriate source address given a particular destination. An application may choose to select the source address itself before starting a new communication or may leave the address unspecified, in which case the upper networking layers will use the mechanism provided by the IP layer to choose a suitable address on the application's behalf.

4. ROUTER SPECIFICATION

The stateless address autoconfiguration mechanism relies on the prefix discovery mechanism specified in [2] for advertising network prefixes required for the formation of addresses with site-local or global scope.

A prefix is advertised in a Prefix Information option of a Router Advertisement. Prefix Information includes

1. the prefix itself

2. the prefix length

3. a flag indicating whether the prefix is to be used for prefix discovery[2].

4. a flag indicating whether the prefix is to be used for stateless address autoconfiguration

5. the deprecation lifetime of the prefix in seconds for the purpose of address deprecation

6 the invalidation lifetime of the prefix for the purpose of address invalidation. This field is also used by prefix discovery[2].

Router Advertisement processing is specified completely in [2] along with the message formats and configuration variables.

5. HOST SPECIFICATION

This section specifies host address autoconfiguration behavior on receiving a Router Advertisement.

5.1. Host Configuration Variables

A host SHOULD allow the following variable to be configured per multicast interface:

Perform_Addr_Config

If set, the host MUST use either stateless or stateful mechanisms to configure global or site-local addresses and to acquire other configuration information as specified in this document.

Default: TRUE

An interface that has the Perform_Addr_Config flag set is called an "autoconfigurable interface".

5.2. Message Validation

A host MUST silently discard any Router Advertisement that does not specify the validity checks as specified in [2]. An advertisement that passes these validity checks is called a valid advertisement.

5.3. Router Advertisement Processing

To receive a Router Advertisement, a host joins the all-nodes multicast address over all multicast-capable interfaces.

A host performs the following address configuration processing when a valid solicited or unsolicited Router Advertisement is received over an autoconfigurable interface:

For each valid Router Advertisement:

The host stores the current value of the Address Mode and then sets the Address Mode to the value of the Managed flag (M bit). If the new value is set, the host MUST use stateful address configuration to configure and maintain a list of site-local or global addresses per interface.

Note that this does not mean that the stateful protocol is necessarily invoked each time a Router Advertisement arrives with the M bit set. The host uses the flag only to indicate whether the stateful protocol is to be used to configure addresses. The stateful protocol is enabled as soon as the Address Mode changes from FALSE to TRUE. The protocol is disabled as soon as the flag changes from TRUE to FALSE. The actual times at which the protocol is invoked, for example, to request a list of addresses or renew a list of addresses, are specified by the protocol itself.

The host stores the current value of the Other Configuration flag and then sets the Other Configuration Mode flag to that in the Router Advertisement (O bit). If the new value is set, the host MUST use stateful autoconfiguration to acquire other configuration information besides the address.

The above disclaimer applies here as well. The O bit indicates whether the host must use the stateful mode to acquire other configuration information. The stateful protocol is enabled for this purpose as soon as the Other Configuration Mode changes from FALSE to TRUE. The protocol is disabled as soon as the mode changes from TRUE to FALSE. The mode does not indicate the timing of frequency of acquiring that information. This is defined by the stateful protocol itself.

For each Prefix-Information extension in the Router Advertisement that has the Autonomous flag set:

- If the prefix advertised matches the prefix of an

autoconfigured address already in the list, then set the deprecation timer to that of the deprecation lifetime specified in the extension and the invalidation timer to that of the invalidation lifetime.

Note that this processing MUST NOT be applied to the link-local address. That is, if the well-known link-local prefix is advertised for some reason (probably a configuration error), then the prefix should be ignored and a system management error logged.

- If the prefix advertised does not match the prefix of an address already in the list, then form an address using this network prefix and the interface token. Definitions of the interface token to be used on a specific link are documented elsewhere.

  If the prefix advertised is too short or too long to form a 128-bit address, given the interface token, the prefix is ignored and an error is logged.

  Add this address to the list with the deprecation timer set to that of the deprecation lifetime and the invalidation timer to that of the invalidation lifetime.

  Note: The address list should be variable-length. Hosts should be able to store the link-local address as well as all addresses configured using both the stateless and stateful modes. If the implementation cannot store all addresses, the host should log a system management error.

Note that if the deprecation lifetime is zero, the address with that prefix is immediately deprecated. Similarly, if the invalidation lifetime is zero, the address with that prefix is immediately made invalid. (The invalidation lifetime is defined to be no less than the deprecation lifetime.) If the deprecation lifetime is infinity, the address is never deprecated. Similarly, if the invalidation lifetime is infinity, the address is never invalidated. The value of infinity is defined in [2].

An address is valid until the deprecation timer expires. A valid address can be used as a source address in all outgoing

communications and is accepted as a destination address in all incoming communications.

When the deprecation lifetime of an address expires, the address SHOULD continue to be used as a source address if it is in use in existing communications, but SHOULD NOT be used in new communications if a valid address is available and it has sufficient scope. The IP layer MUST continue to accept datagrams destined to a deprecated address since a deprecated address is still defined to identify the interface.

An address remains deprecated until its invalidation timer expires at which point the address becomes invalid and is removed from the address list. An invalid address can no longer be used as a source address in outgoing communications and is not recognised as a valid destination address in incoming communications since the address is defined to no longer identify the interface.

On initialisation of an interface, if a host determines that there are no IPv6 routers on a link, a host MUST attempt to use stateful autoconfiguration to acquire addresses and other configuration information. This is needed to support networks with no IPv6 routers. The host determines that there are no routers on the link after startup if no Router Advertisements are heard in the time that it would take to send MAX_ROUTER_SOLICITATIONs and wait for a response[2]. If a router comes up on the link and Router Advertisements begin to be received, a host MUST start to use Router Advertisements in the normal way, and, in particular, use advertisements to determine whether stateless or stateful address configuration should be in use.

Note that it is possible for hosts to get address information using both stateless and stateful protocols since both may be enabled at the same time. Even if only stateless address autoconfiguration mode is enabled, it is still possible for hosts to get information from multiple sources since multiple routers may be advertising prefix information. The rules for handling this is as follows: hosts accept the union of all information received using the stateless and stateful protocols. If different sources configure the same address, then the address is updated with the most recently advertised lifetime.

It is also possible for hosts to contain address information from different sources, when changing from one mechanism to the other, i.e. when changing from stateless mode to stateful mode, and vice

versa. In this case, the rules are no different from above. If the newly enabled mode is configured to hand out different addresses than the mode just disabled, then the host contains the union of addresses from both sources until the addresses configured using the old protocol timeout. If both the old and new modes are configured to hand out the same address, then the address is updated with the most recently advertised lifetime. NOTE: The above assumes that the stateless and stateful modes have the same lifetime semantics.

6. NODE SPECIFICATION

This section specifies the rules for forming a link-local address. It also specifies the protocol to be used for duplicate address detection.

6.1. Forming a Link-Local Address

A node MUST have a link-local address. A node forms a link-local address whenever an interface is initialised. The method for forming a link-local address is link-dependent and is outside the scope of this document.

An interface may be initialised or become autoconfigurable after any of the following events:

- The interface is initialized at system startup time.
- The interface is reinitialized after a temporary interface failure or after being temporarily disabled by system management.
- The node is re-attached to a link after being detached for some time.

The link-local address is highly likely to need to be one of the first events in the interface initialisation process. Clearly, it must be formed before any duplicate detection processing is performed for this address (Section 6.2). In hosts, a link-local address is also required before a sends out a Router Solicitation, assuming the node chooses to do this[2]. This is because a solicitation is only sent if an advertisement has not yet been heard (and hence no non-link local address can be formed), and the unspecified address cannot be used as a source address in a Router Solicitation.

6.2. Detecting Duplicate Addresses

The procedure to detect a duplicate address MUST be enabled by default in nodes and SHOULD be used.

In principle, the duplicate address detection procedure is applied to each new address configured for an interface, whether it be manually configured or configured automatically using either the stateless or stateful mode. However, if the addresses belonging to an interface are formed using the same interface token (as is the case in stateless autoconfiguration and may be the case in other forms of configuration), then it is only necessary to check that one of the addresses is unique on the link. In the stateless mechanism, it is recommended that the link-local address be the address checked for two reasons. First, it makes the implementation simpler, since the link-local address is guaranteed to always exist in all nodes, whereas global and site-local addresses are not. Second, in hosts, there will be less delay in performing duplicate address detection. Address validation can be done as soon as a link-local address is formed (this can be done immediately on initialisation of an interface), whereas checking a global or site-local address involves waiting until a host hears a Router Advertisement containing address prefixes, and there is the possibility that no advertisement will be heard at all.

The procedure for duplicate address detection uses Neighbor Solicitation and Advertisement messages specified in [2] to validate an address as specified below. Note that before carrying out this procedure, a node joins the all-nodes multicast address. Also, this mechanism is not completely reliable, and so it is possible that duplicate addresses will still exist. If a duplicate address is discovered after carrying out this procedure, the node will need to be configured with a new token, or if this is not possible, the IPv6 addresses will need to be manually configured.

6.2.1. Soliciting Node Behavior

An address is checked for uniqueness only once, when the address is initially configured.

Once the address is configured, the node sends a Neighbor Solicitation with a target address containing the address to be validated. The source is set to the unspecified address and the destination is set to the solicited-node multicast address. The Source Link Layer Address extension SHOULD NOT be sent (as it will be ignored by the

destination node[2]). Loopback of the Neighbor Solicitation MUST NOT be disabled.

NOTE: If the Neighbor Solicitation is the first message to be sent from an interface on interface initialisation, the node should delay a random amount of time between 0 and MAX_INITIALIZATION_DELAY seconds before sending the solicitation. This serves to alleviate congestion when many nodes start up on the link at the same time, such as after a power failure, and helps to avoid race conditions when more than one node is trying to solicit for the same address at the same time. (It is recommended that nodes include some unique value in the seed used to initialise their pseudo-random number generators. Note that using only the node token as a unique value is not sufficient to avoid race conditions since the token cannot be relied upon to be unique. Although the randomization range is specified in units of seconds, the actual randomly-chosen value should not be in units of whole seconds, but rather in units of the highest available timer resolution.)

If after sending a solicitation, no Neighbor Advertisement is received from the target, the node SHOULD retransmit the solicitation at most every DUPL_ADDR_RETRANS_TIMER seconds until either an advertisement is received or the solicitation has been retransmitted MAX_DUPL_ADDR_RETRANS times. If an advertisement is received with a target address the same as the address being validated in the time it takes to send and wait for MAX_DUPL_ADDR_RETRANS solicitations, it must disable that interface and log a system management error. If no such advertisement is received within the time specified, the address is considered to be valid.

6.2.2. Solicited Node Behavior

A node is in the process of validating an address when a Neighbor Solicitation has been sent for the address and no Neighbor Advertisement advertising that address has been received in the time it takes to send out MAX_DUPL_ADDR_RETRANS solicitations and wait for an advertisement (DUPL_ADDR_IGNORE_TIMER seconds).

A node must follow special rules when it receives a Neighbor Solicitation for an address that it is in the process of validating. This

is done to help avoid the race condition where more than one node is attempting to validate the address at the same time, i.e. each node sends out a Neighbor Solicitation and waits to hear a Neighbor Advertisement for the address, but no node actually sends an advertisement since the address has not yet been validated.

The special rules are as follows: When a node is in the process of validating an address and receives a Neighbor Solicitation for that address, it MUST NOT send any advertisement in response to a solicitation for that address. Rather, the node silently discards the solicitation unless the source address is the unspecified address. In the latter case, the first such solicitation received is noted, but otherwise silently discarded (see NOTE below). Any subsequent such solicitations cause the node to disable the interface and log a system management error.

NOTE: The above behavior is required because nodes send out Neighbor Solicitations for their own address with loopback enabled. Thus, a node will always receive at least one solicitation for its own address. However, there is no way for the node to determine, in general, whether the solicitation comes from itself or another node since the source of the packet is the unspecified address. Hence, the above rules specify that a duplicate address is detected only if the node receives more than one solicitation for the address.

Once a node has validated its address, it responds to a Neighbor Solicitation with a Neighbor Advertisement as specified in [2].

6.2.3. Constants

MAX_INITIALIZATION_DELAY	3 seconds
DUPL_ADDR_RETRANS_TIMER	3 seconds
MAX_DUPL_ADDR_RETRANS	1 transmission

7. SECURITY CONSIDERATIONS

To be completed.

8. REFERENCES

[1] R. Hinden and S. Deering, "Internet Protocol Version (IPv6) Addressing Architecture", Internet Draft, May 1995, draft-ietf-ipngwg-addr-arch-02.txt

[2] T. Narten, E. Nordmark and W. A. Simpson, "IPv6 Neighbor Discovery", Internet Draft, July 1995, <draft-ietf-ipngwg-discovery-01.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, Erik Nordmark 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