Network Working Group
Request for Comments: 3927
Category: Standards Track

S. Cheshire
Apple Computer
B. Aboba
Microsoft Corporation
E. Guttman
Sun Microsystems

May 2005

Dynamic Configuration of IPv4 Link-Local Addresses

Status of This Memo

Copyright Notice

Abstract

To participate in wide-area IP networking, a host needs to be configured with IP addresses for its interfaces, either manually by the user or automatically from a source on the network such as a Dynamic Host Configuration Protocol (DHCP) server. Unfortunately, such address configuration information may not always be available. It is therefore beneficial for a host to be able to depend on a useful subset of IP networking functions even when no address configuration is available. This document describes how a host may automatically configure an interface with an IPv4 address within the 169.254/16 prefix that is valid for communication with other devices connected to the same physical (or logical) link.

IPv4 Link-Local addresses are not suitable for communication with devices not directly connected to the same physical (or logical) link, and are only used where stable, routable addresses are not available (such as on ad hoc or isolated networks). This document does not recommend that IPv4 Link-Local addresses and routable addresses be configured simultaneously on the same interface.

Table of Contents

1. Introduction

As the Internet Protocol continues to grow in popularity, it becomes increasingly valuable to be able to use familiar IP tools such as FTP not only for global communication, but for local communication as well. For example, two people with laptop computers supporting IEEE 802.11 Wireless LANs [802.11] may meet and wish to exchange files. It is desirable for these people to be able to use IP application software without the inconvenience of having to manually configure static IP addresses or set up a DHCP server [RFC2131].

This document describes a method by which a host may automatically configure an interface with an IPv4 address in the 169.254/16 prefix that is valid for Link-Local communication on that interface. This is especially valuable in environments where no other configuration mechanism is available. The IPv4 prefix 169.254/16 is registered with the IANA for this purpose. Allocation of IPv6 Link-Local addresses is described in "IPv6 Stateless Address Autoconfiguration" [RFC2462].

Link-Local communication using IPv4 Link-Local addresses is only suitable for communication with other devices connected to the same physical (or logical) link. Link-Local communication using IPv4 Link-Local addresses is not suitable for communication with devices not directly connected to the same physical (or logical) link.

Microsoft Windows 98 (and later) and Mac OS 8.5 (and later) already support this capability. This document standardizes usage, prescribing rules for how IPv4 Link-Local addresses are to be treated by hosts and routers. In particular, it describes how routers are to behave when receiving packets with IPv4 Link-Local addresses in the source or destination address. With respect to hosts, it discusses claiming and defending addresses, maintaining Link-Local and routable IPv4 addresses on the same interface, and multi-homing issues.

1.1. Requirements

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 "Key words for use in RFCs" [RFC2119].

1.2. Terminology

This document describes Link-Local addressing, for IPv4 communication between two hosts on a single link. A set of hosts is considered to be "on the same link", if:

- when any host A from that set sends a packet to any other host B in that set, using unicast, multicast, or broadcast, the entire link-layer packet payload arrives unmodified, and

- a broadcast sent over that link by any host from that set of hosts can be received by every other host in that set

The link-layer *header* may be modified, such as in Token Ring Source Routing [802.5], but not the link-layer *payload*. In particular, if any device forwarding a packet modifies any part of the IP header or IP payload then the packet is no longer considered to be on the same link. This means that the packet may pass through devices such as repeaters, bridges, hubs or switches and still be considered to be on the same link for the purpose of this document, but not through a device such as an IP router that decrements the TTL or otherwise modifies the IP header.

This document uses the term "routable address" to refer to all valid unicast IPv4 addresses outside the 169.254/16 prefix that may be forwarded via routers. This includes all global IP addresses and private addresses such as Net 10/8 [RFC1918], but not loopback addresses such as 127.0.0.1.

Wherever this document uses the term "host" when describing use of IPv4 Link-Local addresses, the text applies equally to routers when they are the source of or intended destination of packets containing IPv4 Link-Local source or destination addresses.

Wherever this document uses the term "sender IP address" or "target IP address" in the context of an ARP packet, it is referring to the fields of the ARP packet identified in the ARP specification [RFC826] as "ar$spa" (Sender Protocol Address) and "ar$tpa" (Target Protocol Address) respectively. For the usage of ARP described in this document, each of these fields always contains an IP address.

In this document, the term "ARP Probe" is used to refer to an ARP Request packet, broadcast on the local link, with an all-zero 'sender IP address'. The 'sender hardware address' MUST contain the hardware address of the interface sending the packet. The 'target hardware address' field is ignored and SHOULD be set to all zeroes. The 'target IP address' field MUST be set to the address being probed.

In this document, the term "ARP Announcement" is used to refer to an ARP Request packet, broadcast on the local link, identical to the ARP Probe described above, except that both the sender and target IP address fields contain the IP address being announced.

Constants are introduced in all capital letters. Their values are given in Section 9.

1.3. Applicability

This specification applies to all IEEE 802 Local Area Networks (LANs) [802], including Ethernet [802.3], Token-Ring [802.5] and IEEE 802.11 wireless LANs [802.11], as well as to other link-layer technologies that operate at data rates of at least 1 Mbps, have a round-trip latency of at most one second, and support ARP [RFC826]. Wherever this document uses the term "IEEE 802", the text applies equally to any of these network technologies.

Link-layer technologies that support ARP but operate at rates below 1 Mbps or latencies above one second may need to specify different values for the following parameters:

(a) the number of, and interval between, ARP probes, see PROBE_NUM, PROBE_MIN, PROBE_MAX defined in Section 2.2.1

(b) the number of, and interval between, ARP announcements, see ANNOUNCE_NUM and ANNOUNCE_INTERVAL defined in Section 2.4

(c) the maximum rate at which address claiming may be attempted, see RATE_LIMIT_INTERVAL and MAX_CONFLICTS defined in Section 2.2.1

(d) the time interval between conflicting ARPs below which a host MUST reconfigure instead of attempting to defend its address, see DEFEND_INTERVAL defined in Section 2.5

Link-layer technologies that do not support ARP may be able to use other techniques for determining whether a particular IP address is currently in use. However, the application of claim-and-defend mechanisms to such networks is outside the scope of this document.

This specification is intended for use with small ad hoc networks -- a single link containing only a few hosts. Although 65024 IPv4 Link-Local addresses are available in principle, attempting to use all those addresses on a single link would result in a high probability of address conflicts, requiring a host to take an inordinate amount of time to find an available address.

Network operators with more than 1300 hosts on a single link may want to consider dividing that single link into two or more subnets. A host connecting to a link that already has 1300 hosts, selecting an IPv4 Link-Local address at random, has a 98% chance of selecting an unused IPv4 Link-Local address on the first try. A host has a 99.96%

chance of selecting an unused IPv4 Link-Local address within two tries. The probability that it will have to try more than ten times is about 1 in 10^17.

1.4. Application Layer Protocol Considerations

IPv4 Link-Local addresses and their dynamic configuration have profound implications upon applications which use them. This is discussed in Section 6. Many applications fundamentally assume that addresses of communicating peers are routable, relatively unchanging and unique. These assumptions no longer hold with IPv4 Link-Local addresses, or a mixture of Link-Local and routable IPv4 addresses.

Therefore while many applications will work properly with IPv4 Link-Local addresses, or a mixture of Link-Local and routable IPv4 addresses, others may do so only after modification, or will exhibit reduced or partial functionality.

In some cases it may be infeasible for the application to be modified to operate under such conditions.

IPv4 Link-Local addresses should therefore only be used where stable, routable addresses are not available (such as on ad hoc or isolated networks) or in controlled situations where these limitations and their impact on applications are understood and accepted. This document does not recommend that IPv4 Link-Local addresses and routable addresses be configured simultaneously on the same interface.

Use of IPv4 Link-Local addresses in off-link communication is likely to cause application failures. This can occur within any application that includes embedded addresses, if an IPv4 Link-Local address is embedded when communicating with a host that is not on the link. Examples of applications that embed addresses include IPsec, Kerberos 4/5, FTP, RSVP, SMTP, SIP, X-Windows/Xterm/Telnet, Real Audio, H.323, and SNMP [RFC3027].

To preclude use of IPv4 Link-Local addresses in off-link communication, the following cautionary measures are advised:

a. IPv4 Link-Local addresses MUST NOT be configured in the DNS. Mapping from IPv4 addresses to host names is conventionally done by issuing DNS queries for names of the form, "x.x.x.x.in-addr.arpa." When used for link-local addresses, which have significance only on the local link, it is inappropriate to send such DNS queries beyond the local link. DNS clients MUST NOT send DNS queries for any name that falls within the "254.169.in-addr.arpa." domain.

DNS recursive name servers receiving queries from non-compliant clients for names within the "254.169.in-addr.arpa." domain MUST by default return RCODE 3, authoritatively asserting that no such name exists in the Domain Name System.

b. Names that are globally resolvable to routable addresses should be used within applications whenever they are available. Names that are resolvable only on the local link (such as through use of protocols such as Link Local Multicast Name Resolution [LLMNR]) MUST NOT be used in off-link communication. IPv4 addresses and names that can only be resolved on the local link SHOULD NOT be forwarded beyond the local link. IPv4 Link-Local addresses SHOULD only be sent when a Link-Local address is used as the source and/or destination address. This strong advice should hinder limited scope addresses and names from leaving the context in which they apply.

c. If names resolvable to globally routable addresses are not available, but the globally routable addresses are, they should be used instead of IPv4 Link-Local addresses.

1.5. Autoconfiguration Issues

Implementations of IPv4 Link-Local address autoconfiguration MUST expect address conflicts, and MUST be prepared to handle them gracefully by automatically selecting a new address whenever a conflict is detected, as described in Section 2. This requirement to detect and handle address conflicts applies during the entire period that a host is using a 169.254/16 IPv4 Link-Local address, not just during initial interface configuration. For example, address conflicts can occur well after a host has completed booting if two previously separate networks are joined, as described in Section 4.

1.6. Alternate Use Prohibition

Note that addresses in the 169.254/16 prefix SHOULD NOT be configured manually or by a DHCP server. Manual or DHCP configuration may cause a host to use an address in the 169.254/16 prefix without following the special rules regarding duplicate detection and automatic configuration that pertain to addresses in this prefix. While the DHCP specification [RFC2131] indicates that a DHCP client SHOULD probe a newly received address with ARP, this is not mandatory. Similarly, while the DHCP specification recommends that a DHCP server SHOULD probe an address using an ICMP Echo Request before allocating it, this is also not mandatory, and even if the server does this, IPv4 Link-Local addresses are not routable, so a DHCP server not directly connected to a link cannot detect whether a host on that link is already using the desired IPv4 Link-Local address.

Administrators wishing to configure their own local addresses (using manual configuration, a DHCP server, or any other mechanism not described in this document) should use one of the existing private address prefixes [RFC1918], not the 169.254/16 prefix.

1.7. Multiple Interfaces

Additional considerations apply to hosts that support more than one active interface where one or more of these interfaces support IPv4 Link-Local address configuration. These considerations are discussed in Section 3.

1.8. Communication with Routable Addresses

There will be cases when devices with a configured Link-Local address will need to communicate with a device with a routable address configured on the same physical link, and vice versa. The rules in Section 2.6 allow this communication.

This allows, for example, a laptop computer with only a routable address to communicate with web servers world-wide using its globally-routable address while at the same time printing those web pages on a local printer that has only an IPv4 Link-Local address.

1.9. When to configure an IPv4 Link-Local address

Having addresses of multiple different scopes assigned to an interface, with no adequate way to determine in what circumstances each address should be used, leads to complexity for applications and confusion for users. A host with an address on a link can communicate with all other devices on that link, whether those devices use Link-Local addresses, or routable addresses. For these reasons, a host SHOULD NOT have both an operable routable address and an IPv4 Link-Local address configured on the same interface. The term "operable address" is used to mean an address which works effectively for communication in the current network context (see below). When an operable routable address is available on an interface, the host SHOULD NOT also assign an IPv4 Link-Local address on that interface. However, during the transition (in either direction) between using routable and IPv4 Link-Local addresses both MAY be in use at once subject to these rules:

1. The assignment of an IPv4 Link-Local address on an interface is based solely on the state of the interface, and is independent of any other protocols such as DHCP. A host MUST NOT alter its behavior and use of other protocols such as DHCP because the host has assigned an IPv4 Link-Local address to an interface.

2. If a host finds that an interface that was previously configured with an IPv4 Link-Local address now has an operable routable address available, the host MUST use the routable address when initiating new communications, and MUST cease advertising the availability of the IPv4 Link-Local address through whatever mechanisms that address had been made known to others. The host SHOULD continue to use the IPv4 Link-Local address for communications already underway, and MAY continue to accept new communications addressed to the IPv4 Link-Local address. Ways in which an operable routable address might become available on an interface include:

   * Manual configuration
   * Address assignment through DHCP
   * Roaming of the host to a network on which a previously assigned address becomes operable

3. If a host finds that an interface no longer has an operable routable address available, the host MAY identify a usable IPv4 Link-Local address (as described in section 2) and assign that address to the interface. Ways in which an operable routable address might cease to be available on an interface include:

   * Removal of the address from the interface through manual configuration
   * Expiration of the lease on the address assigned through DHCP
   * Roaming of the host to a new network on which the address is no longer operable.

The determination by the system of whether an address is "operable" is not clear cut and many changes in the system context (e.g., router changes) may affect the operability of an address. In particular roaming of a host from one network to another is likely -- but not certain -- to change the operability of a configured address but detecting such a move is not always trivial.

"Detection of Network Attachment (DNA) in IPv4" [DNAv4] provides further discussion of address assignment and operability determination.

2. Address Selection, Defense and Delivery

The following section explains the IPv4 Link-Local address selection algorithm, how IPv4 Link-Local addresses are defended, and how IPv4 packets with IPv4 Link-Local addresses are delivered.

Windows and Mac OS hosts that already implement Link-Local IPv4 address auto-configuration are compatible with the rules presented in this section. However, should any interoperability problem be discovered, this document, not any prior implementation, defines the standard.

2.1. Link-Local Address Selection

When a host wishes to configure an IPv4 Link-Local address, it selects an address using a pseudo-random number generator with a uniform distribution in the range from 169.254.1.0 to 169.254.254.255 inclusive.

The IPv4 prefix 169.254/16 is registered with the IANA for this purpose. The first 256 and last 256 addresses in the 169.254/16 prefix are reserved for future use and MUST NOT be selected by a host using this dynamic configuration mechanism.

The pseudo-random number generation algorithm MUST be chosen so that different hosts do not generate the same sequence of numbers. If the host has access to persistent information that is different for each host, such as its IEEE 802 MAC address, then the pseudo-random number generator SHOULD be seeded using a value derived from this information. This means that even without using any other persistent storage, a host will usually select the same IPv4 Link-Local address each time it is booted, which can be convenient for debugging and other operational reasons. Seeding the pseudo-random number generator using the real-time clock or any other information which is (or may be) identical in every host is NOT suitable for this purpose, because a group of hosts that are all powered on at the same time might then all generate the same sequence, resulting in a never-ending series of conflicts as the hosts move in lock-step through exactly the same pseudo-random sequence, conflicting on every address they probe.

Hosts that are equipped with persistent storage MAY, for each interface, record the IPv4 address they have selected. On booting, hosts with a previously recorded address SHOULD use that address as their first candidate when probing. This increases the stability of addresses. For example, if a group of hosts are powered off at night, then when they are powered on the next morning they will all resume using the same addresses, instead of picking different addresses and potentially having to resolve conflicts that arise.

2.2. Claiming a Link-Local Address

After it has selected an IPv4 Link-Local address, a host MUST test to see if the IPv4 Link-Local address is already in use before beginning to use it. When a network interface transitions from an inactive to an active state, the host does not have knowledge of what IPv4 Link-Local addresses may currently be in use on that link, since the point of attachment may have changed or the network interface may have been inactive when a conflicting address was claimed.

Were the host to immediately begin using an IPv4 Link-Local address which is already in use by another host, this would be disruptive to that other host. Since it is possible that the host has changed its point of attachment, a routable address may be obtainable on the new network, and therefore it cannot be assumed that an IPv4 Link-Local address is to be preferred.

Before using the IPv4 Link-Local address (e.g., using it as the source address in an IPv4 packet, or as the Sender IPv4 address in an ARP packet) a host MUST perform the probing test described below to achieve better confidence that using the IPv4 Link-Local address will not cause disruption.

Examples of events that involve an interface becoming active include:

- Reboot/startup
- Wake from sleep (if network interface was inactive during sleep)
- Bringing up previously inactive network interface
- IEEE 802 hardware link-state change (appropriate for the media type and security mechanisms which apply) indicates that an interface has become active.
- Association with a wireless base station or ad hoc network.

A host MUST NOT perform this check periodically as a matter of course. This would be a waste of network bandwidth, and is unnecessary due to the ability of hosts to passively discover conflicts, as described in Section 2.5.

2.2.1. Probe details

On a link-layer such as IEEE 802 that supports ARP, conflict detection is done using ARP probes. On link-layer technologies that do not support ARP other techniques may be available for determining whether a particular IPv4 address is currently in use. However, the application of claim-and-defend mechanisms to such networks is outside the scope of this document.

A host probes to see if an address is already in use by broadcasting an ARP Request for the desired address. The client MUST fill in the 'sender hardware address' field of the ARP Request with the hardware address of the interface through which it is sending the packet. The 'sender IP address' field MUST be set to all zeroes, to avoid polluting ARP caches in other hosts on the same link in the case where the address turns out to be already in use by another host. The 'target hardware address' field is ignored and SHOULD be set to all zeroes. The 'target IP address' field MUST be set to the address being probed. An ARP Request constructed this way with an all-zero 'sender IP address' is referred to as an "ARP Probe".

When ready to begin probing, the host should then wait for a random time interval selected uniformly in the range zero to PROBE_WAIT seconds, and should then send PROBE_NUM probe packets, each of these probe packets spaced randomly, PROBE_MIN to PROBE_MAX seconds apart. If during this period, from the beginning of the probing process until ANNOUNCE_WAIT seconds after the last probe packet is sent, the host receives any ARP packet (Request *or* Reply) on the interface where the probe is being performed where the packet's 'sender IP address' is the address being probed for, then the host MUST treat this address as being in use by some other host, and MUST select a new pseudo-random address and repeat the process. In addition, if during this period the host receives any ARP Probe where the packet's 'target IP address' is the address being probed for, and the packet's 'sender hardware address' is not the hardware address of the interface the host is attempting to configure, then the host MUST similarly treat this as an address conflict and select a new address as above. This can occur if two (or more) hosts attempt to configure the same IPv4 Link-Local address at the same time.

A host should maintain a counter of the number of address conflicts it has experienced in the process of trying to acquire an address, and if the number of conflicts exceeds MAX_CONFLICTS then the host MUST limit the rate at which it probes for new addresses to no more than one new address per RATE_LIMIT_INTERVAL. This is to prevent catastrophic ARP storms in pathological failure cases, such as a rogue host that answers all ARP probes, causing legitimate hosts to go into an infinite loop attempting to select a usable address.

If, by ANNOUNCE_WAIT seconds after the transmission of the last ARP Probe no conflicting ARP Reply or ARP Probe has been received, then the host has successfully claimed the desired IPv4 Link-Local address.

2.3. Shorter Timeouts

Network technologies may emerge for which shorter delays are appropriate than those required by this document. A subsequent IETF publication may be produced providing guidelines for different values for PROBE_WAIT, PROBE_NUM, PROBE_MIN and PROBE_MAX on those technologies.

2.4. Announcing an Address

Having probed to determine a unique address to use, the host MUST then announce its claimed address by broadcasting ANNOUNCE_NUM ARP announcements, spaced ANNOUNCE_INTERVAL seconds apart. An ARP announcement is identical to the ARP Probe described above, except that now the sender and target IP addresses are both set to the host's newly selected IPv4 address. The purpose of these ARP announcements is to make sure that other hosts on the link do not have stale ARP cache entries left over from some other host that may previously have been using the same address.

2.5. Conflict Detection and Defense

Address conflict detection is not limited to the address selection phase, when a host is sending ARP probes. Address conflict detection is an ongoing process that is in effect for as long as a host is using an IPv4 Link-Local address. At any time, if a host receives an ARP packet (request *or* reply) on an interface where the 'sender IP address' is the IP address the host has configured for that interface, but the 'sender hardware address' does not match the hardware address of that interface, then this is a conflicting ARP packet, indicating an address conflict.

A host MUST respond to a conflicting ARP packet as described in either (a) or (b) below:

(a) Upon receiving a conflicting ARP packet, a host MAY elect to immediately configure a new IPv4 Link-Local address as described above, or

(b) If a host currently has active TCP connections or other reasons to prefer to keep the same IPv4 address, and it has not seen any other conflicting ARP packets within the last DEFEND_INTERVAL seconds, then it MAY elect to attempt to defend its address by recording the time that the conflicting ARP packet was received, and then broadcasting one single ARP announcement, giving its own IP and hardware addresses as the sender addresses of the ARP. Having done this, the host can then continue to use the address normally without any further special action. However, if this is not the first

conflicting ARP packet the host has seen, and the time recorded for the previous conflicting ARP packet is recent, within DEFEND_INTERVAL seconds, then the host MUST immediately cease using this address and configure a new IPv4 Link-Local address as described above. This is necessary to ensure that two hosts do not get stuck in an endless loop with both hosts trying to defend the same address.

A host MUST respond to conflicting ARP packets as described in either (a) or (b) above. A host MUST NOT ignore conflicting ARP packets.

Forced address reconfiguration may be disruptive, causing TCP connections to be broken. However, it is expected that such disruptions will be rare, and if inadvertent address duplication happens, then disruption of communication is inevitable, no matter how the addresses were assigned. It is not possible for two different hosts using the same IP address on the same network to operate reliably.

Before abandoning an address due to a conflict, hosts SHOULD actively attempt to reset any existing connections using that address. This mitigates some security threats posed by address reconfiguration, as discussed in Section 5.

Immediately configuring a new address as soon as the conflict is detected is the best way to restore useful communication as quickly as possible. The mechanism described above of broadcasting a single ARP announcement to defend the address mitigates the problem somewhat, by helping to improve the chance that one of the two conflicting hosts may be able to retain its address.

All ARP packets (*replies* as well as requests) that contain a Link-Local 'sender IP address' MUST be sent using link-layer broadcast instead of link-layer unicast. This aids timely detection of duplicate addresses. An example illustrating how this helps is given in Section 4.

2.6. Address Usage and Forwarding Rules

A host implementing this specification has additional rules to conform to, whether or not it has an interface configured with an IPv4 Link-Local address.

2.6.1. Source Address Usage

Since each interface on a host may have an IPv4 Link-Local address in addition to zero or more other addresses configured by other means (e.g., manually or via a DHCP server), a host may have to make a

choice about what source address to use when it sends a packet or initiates a TCP connection.

Where both an IPv4 Link-Local and a routable address are available on the same interface, the routable address should be preferred as the source address for new communications, but packets sent from or to the IPv4 Link-Local address are still delivered as expected. The IPv4 Link-Local address may continue to be used as a source address in communications where switching to a preferred address would cause communications failure because of the requirements of an upper-layer protocol (e.g., an existing TCP connection). For more details, see Section 1.7.

A multi-homed host needs to select an outgoing interface whether or not the destination is an IPv4 Link-Local address. Details of that process are beyond the scope of this specification. After selecting an interface, the multi-homed host should send packets involving IPv4 Link-Local addresses as specified in this document, as if the selected interface were the host's only interface. See Section 3 for further discussion of multi-homed hosts.

2.6.2. Forwarding Rules

Whichever interface is used, if the destination address is in the 169.254/16 prefix (excluding the address 169.254.255.255, which is the broadcast address for the Link-Local prefix), then the sender MUST ARP for the destination address and then send its packet directly to the destination on the same physical link. This MUST be done whether the interface is configured with a Link-Local or a routable IPv4 address.

In many network stacks, achieving this functionality may be as simple as adding a routing table entry indicating that 169.254/16 is directly reachable on the local link. This approach will not work for routers or multi-homed hosts. Refer to section 3 for more discussion of multi-homed hosts.

The host MUST NOT send a packet with an IPv4 Link-Local destination address to any router for forwarding.

If the destination address is a unicast address outside the 169.254/16 prefix, then the host SHOULD use an appropriate routable IPv4 source address, if it can. If for any reason the host chooses to send the packet with an IPv4 Link-Local source address (e.g., no routable address is available on the selected interface), then it MUST ARP for the destination address and then send its packet, with

an IPv4 Link-Local source address and a routable destination IPv4 address, directly to its destination on the same physical link. The host MUST NOT send the packet to any router for forwarding.

In the case of a device with a single interface and only an Link-Local IPv4 address, this requirement can be paraphrased as "ARP for everything".

In many network stacks, achieving this "ARP for everything" behavior may be as simple as having no primary IP router configured, having the primary IP router address configured to 0.0.0.0, or having the primary IP router address set to be the same as the host's own Link-Local IPv4 address. For suggested behavior in multi-homed hosts, see Section 3.

2.7. Link-Local Packets Are Not Forwarded

A sensible default for applications which are sending from an IPv4 Link-Local address is to explicitly set the IPv4 TTL to 1. This is not appropriate in all cases as some applications may require that the IPv4 TTL be set to other values.

An IPv4 packet whose source and/or destination address is in the 169.254/16 prefix MUST NOT be sent to any router for forwarding, and any network device receiving such a packet MUST NOT forward it, regardless of the TTL in the IPv4 header. Similarly, a router or other host MUST NOT indiscriminately answer all ARP Requests for addresses in the 169.254/16 prefix. A router may of course answer ARP Requests for one or more IPv4 Link-Local address(es) that it has legitimately claimed for its own use according to the claim-and-defend protocol described in this document.

This restriction also applies to multicast packets. IPv4 packets with a Link-Local source address MUST NOT be forwarded outside the local link even if they have a multicast destination address.

2.8. Link-Local Packets are Local

The non-forwarding rule means that hosts may assume that all 169.254/16 destination addresses are "on-link" and directly reachable. The 169.254/16 address prefix MUST NOT be subnetted. This specification utilizes ARP-based address conflict detection, which functions by broadcasting on the local subnet. Since such broadcasts are not forwarded, were subnetting to be allowed then address conflicts could remain undetected.

This does not mean that Link-Local devices are forbidden from any communication outside the local link. IP hosts that implement both Link-Local and conventional routable IPv4 addresses may still use their routable addresses without restriction as they do today.

2.9. Higher-Layer Protocol Considerations

Similar considerations apply at layers above IP.

For example, designers of Web pages (including automatically generated web pages) SHOULD NOT contain links with embedded IPv4 Link-Local addresses if those pages are viewable from hosts outside the local link where the addresses are valid.

As IPv4 Link-Local addresses may change at any time and have limited scope, IPv4 Link-Local addresses MUST NOT be stored in the DNS.

2.10. Privacy Concerns

Another reason to restrict leakage of IPv4 Link-Local addresses outside the local link is privacy concerns. If IPv4 Link-Local addresses are derived from a hash of the MAC address, some argue that they could be indirectly associated with an individual, and thereby used to track that individual's activities. Within the local link the hardware addresses in the packets are all directly observable, so as long as IPv4 Link-Local addresses don't leave the local link they provide no more information to an intruder than could be gained by direct observation of hardware addresses.

2.11. Interaction between DHCPv4 client and IPv4 Link-Local State Machines

As documented in Appendix A, early implementations of IPv4 Link-Local have modified the DHCP state machine. Field experience shows that these modifications reduce the reliability of the DHCP service.

A device that implements both IPv4 Link-Local and a DHCPv4 client should not alter the behavior of the DHCPv4 client to accommodate IPv4 Link-Local configuration. In particular configuration of an IPv4 Link-Local address, whether or not a DHCP server is currently responding, is not sufficient reason to unconfigure a valid DHCP lease, to stop the DHCP client from attempting to acquire a new IP address, to change DHCP timeouts or to change the behavior of the DHCP state machine in any other way.

Further discussion of this issue is provided in "Detection of Network Attachment (DNA) in IPv4" [DNAv4].

3. Considerations for Multiple Interfaces

The considerations outlined here also apply whenever a host has multiple IP addresses, whether or not it has multiple physical interfaces. Other examples of multiple interfaces include different logical endpoints (tunnels, virtual private networks etc.) and multiple logical networks on the same physical medium. This is often referred to as "multi-homing".

Hosts which have more than one active interface and elect to implement dynamic configuration of IPv4 Link-Local addresses on one or more of those interfaces will face various problems. This section lists these problems but does no more than indicate how one might solve them. At the time of this writing, there is no silver bullet which solves these problems in all cases, in a general way. Implementors must think through these issues before implementing the protocol specified in this document on a system which may have more than one active interface as part of a TCP/IP stack capable of multi-homing.

3.1. Scoped Addresses

A host may be attached to more than one network at the same time. It would be nice if there was a single address space used in every network, but this is not the case. Addresses used in one network, be it a network behind a NAT or a link on which IPv4 Link-Local addresses are used, cannot be used in another network and have the same effect.

It would also be nice if addresses were not exposed to applications, but they are. Most software using TCP/IP which await messages receives from any interface at a particular port number, for a particular transport protocol. Applications are generally only aware (and care) that they have received a message. The application knows the address of the sender to which the application will reply.

The first scoped address problem is source address selection. A multi-homed host has more than one address. Which address should be used as the source address when sending to a particular destination? This question is usually answered by referring to a routing table, which expresses on which interface (with which address) to send, and how to send (should one forward to a router, or send directly). The choice is made complicated by scoped addresses because the address range in which the destination lies may be ambiguous. The table may not be able to yield a good answer. This problem is bound up with next-hop selection, which is discussed in Section 3.2.

The second scoped address problem arises from scoped parameters leaking outside their scope. This is discussed in Section 7.

It is possible to overcome these problems. One way is to expose scope information to applications such that they are always aware of what scope a peer is in. This way, the correct interface could be selected, and a safe procedure could be followed with respect to forwarding addresses and other scoped parameters. There are other possible approaches. None of these methods have been standardized for IPv4 nor are they specified in this document. A good API design could mitigate the problems, either by exposing address scopes to 'scoped-address aware' applications or by cleverly encapsulating the scoping information and logic so that applications do the right thing without being aware of address scoping.

An implementer could undertake to solve these problems, but cannot simply ignore them. With sufficient experience, it is hoped that specifications will emerge explaining how to overcome scoped address multi-homing problems.

3.2. Address Ambiguity

This is a core problem with respect to IPv4 Link-Local destination addresses being reachable on more than one interface. What should a host do when it needs to send to Link-Local destination L and L can be resolved using ARP on more than one link?

Even if a Link-Local address can be resolved on only one link at a given moment, there is no guarantee that it will remain unambiguous in the future. Additional hosts on other interfaces may claim the address L as well.

One possibility is to support this only in the case where the application specifically expresses which interface to send from.

There is no standard or obvious solution to this problem. Existing application software written for the IPv4 protocol suite is largely incapable of dealing with address ambiguity. This does not preclude an implementer from finding a solution, writing applications which are able to use it, and providing a host which can support dynamic configuration of IPv4 Link-Local addresses on more than one interface. This solution will almost surely not be generally applicable to existing software and transparent to higher layers, however.

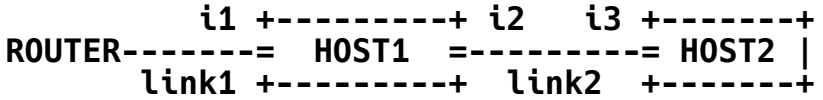
Given that the IP stack must have the outbound interface associated with a packet that needs to be sent to a Link-Local destination address, interface selection must occur. The outbound interface

cannot be derived from the packet's header parameters such as source or destination address (e.g., by using the forwarding table lookup). Therefore, outbound interface association must be done explicitly through other means. The specification does not stipulate those means.

3.3. Interaction with Hosts with Routable Addresses

Attention is paid in this specification to transition from the use of IPv4 Link-Local addresses to routable addresses (see Section 1.5). The intention is to allow a host with a single interface to first support Link-Local configuration then gracefully transition to the use of a routable address. Since the host transitioning to the use of a routable address may temporarily have more than one address active, the scoped address issues described in Section 3.1 will apply. When a host acquires a routable address, it does not need to retain its Link-Local address for the purpose of communicating with other devices on the link that are themselves using only Link-Local addresses: any host conforming to this specification knows that regardless of source address an IPv4 Link-Local destination must be reached by forwarding directly to the destination, not via a router; it is not necessary for that host to have a Link-Local source address in order to send to a Link-Local destination address.

A host with an IPv4 Link-Local address may send to a destination which does not have an IPv4 Link-Local address. If the host is not multi-homed, the procedure is simple and unambiguous: Using ARP and forwarding directly to on-link destinations is the default route. If the host is multi-homed, however, the routing policy is more complex, especially if one of the interfaces is configured with a routable address and the default route is (sensibly) directed at a router accessible through that interface. The following example illustrates this problem and provides a common solution to it.



In the figure above, HOST1 is connected to link1 and link2. Interface i1 is configured with a routable address, while i2 is an IPv4 Link-Local address. HOST1 has its default route set to ROUTER's address, through i1. HOST1 will route to destinations in 169.254/16 to i2, sending directly to the destination.

HOST2 has a configured (non-Link-Local) IPv4 address assigned to i3.

Using a name resolution or service discovery protocol HOST1 can discover HOST2's address. Since HOST2's address is not in 169.254/16, HOST1's routing policy will send datagrams to HOST2 via i1, to the ROUTER. Unless there is a route from ROUTER to HOST2, the datagrams sent from HOST1 to HOST2 will not reach it.

One solution to this problem is for a host to attempt to reach any host locally (using ARP) for which it receives an unreachable ICMP error message (ICMP message codes 0, 1, 6 or 7 [RFC792]). The host tries all its attached links in a round robin fashion. This has been implemented successfully for some IPv6 hosts, to circumvent exactly this problem. In terms of this example, HOST1 upon failing to reach HOST2 via the ROUTER, will attempt to forward to HOST2 via i2 and succeed.

It may also be possible to overcome this problem using techniques described in section 3.2, or other means not discussed here. This specification does not provide a standard solution, nor does it preclude implementers from supporting multi-homed configurations, provided that they address the concerns in this section for the applications which will be supported on the host.

3.4. Unintentional Autoimmune Response

Care must be taken if a multi-homed host can support more than one interface on the same link, all of which support IPv4 Link-Local autoconfiguration. If these interfaces attempt to allocate the same address, they will defend the host against itself -- causing the claiming algorithm to fail. The simplest solution to this problem is to run the algorithm independently on each interface configured with IPv4 Link-Local addresses.

In particular, ARP packets which appear to claim an address which is assigned to a specific interface, indicate conflict only if they are received on that interface and their hardware address is of some other interface.

If a host has two interfaces on the same link, then claiming and defending on those interfaces must ensure that they end up with different addresses just as if they were on different hosts. Note that some of the ways a host may find itself with two interfaces on the same link may be unexpected and non-obvious, such as when a host has Ethernet and 802.11 wireless, but those two links are (possibly even without the knowledge of the host's user) bridged together.

4. Healing of Network Partitions

Hosts on disjoint network links may configure the same IPv4 Link-Local address. If these separate network links are later joined or bridged together, then there may be two hosts which are now on the same link, trying to use the same address. When either host attempts to communicate with any other host on the network, it will at some point broadcast an ARP packet which will enable the hosts in question to detect that there is an address conflict.

When these address conflicts are detected, the subsequent forced reconfiguration may be disruptive, causing TCP connections to be broken. However, it is expected that such disruptions will be rare. It should be relatively uncommon for networks to be joined while hosts on those networks are active. Also, 65024 addresses are available for IPv4 Link-Local use, so even when two small networks are joined, the chance of conflict for any given host is fairly small.

When joining two large networks (defined as networks with a substantial number of hosts per segment) there is a greater chance of conflict. In such networks, it is likely that the joining of previously separated segments will result in one or more hosts needing to change their IPv4 Link-Local address, with subsequent loss of TCP connections. In cases where separation and re-joining is frequent, as in remotely bridged networks, this could prove disruptive. However, unless the number of hosts on the joined segments is very large, the traffic resulting from the join and subsequent address conflict resolution will be small.

Sending ARP replies that have IPv4 Link-Local sender addresses via broadcast instead of unicast ensures that these conflicts can be detected as soon as they become potential problems, but no sooner. For example, if two disjoint network links are joined, where hosts A and B have both configured the same Link-Local address, X, they can remain in this state until A, B or some other host attempts to initiate communication. If some other host C now sends an ARP request for address X, and hosts A and B were to both reply with conventional unicast ARP replies, then host C might be confused, but A and B still wouldn't know there is a problem because neither would have seen the other's packet. Sending these replies via broadcast allows A and B to see each other's conflicting ARP packets and respond accordingly.

Note that sending periodic gratuitous ARPs in an attempt to detect these conflicts sooner is not necessary, wastes network bandwidth, and may actually be detrimental. For example, if the network links were joined only briefly, and were separated again before any new

communication involving A or B were initiated, then the temporary conflict would have been benign and no forced reconfiguration would have been required. Triggering an unnecessary forced reconfiguration in this case would not serve any useful purpose. Hosts SHOULD NOT send periodic gratuitous ARPs.

5. Security Considerations

The use of IPv4 Link-Local Addresses may open a network host to new attacks. In particular, a host that previously did not have an IP address, and no IP stack running, was not susceptible to IP-based attacks. By configuring a working address, the host may now be vulnerable to IP-based attacks.

The ARP protocol [RFC826] is insecure. A malicious host may send fraudulent ARP packets on the network, interfering with the correct operation of other hosts. For example, it is easy for a host to answer all ARP requests with replies giving its own hardware address, thereby claiming ownership of every address on the network.

NOTE: There are certain kinds of local links, such as wireless LANs, that provide no physical security. Because of the existence of these links it would be very unwise for an implementer to assume that when a device is communicating only on the local link it can dispense with normal security precautions. Failure to implement appropriate security measures could expose users to considerable risks.

A host implementing IPv4 Link-Local configuration has an additional vulnerability to selective reconfiguration and disruption. It is possible for an on-link attacker to issue ARP packets which would cause a host to break all its connections by switching to a new address. The attacker could force the host implementing IPv4 Link-Local configuration to select certain addresses, or prevent it from ever completing address selection. This is a distinct threat from that posed by spoofed ARPs, described in the preceding paragraph.

Implementations and users should also note that a node that gives up an address and reconfigures, as required by section 2.5, allows the possibility that another node can easily and successfully hijack existing TCP connections.

Implementers are advised that the Internet Protocol architecture expects every networked device or host must implement security which is adequate to protect the resources to which the device or host has access, including the network itself, against known or credible threats. Even though use of IPv4 Link-Local addresses may reduce the

number of threats to which a device is exposed, implementers of devices supporting the Internet Protocol must not assume that a customer's local network is free from security risks.

While there may be particular kinds of devices, or particular environments, for which the security provided by the network is adequate to protect the resources that are accessible by the device, it would be misleading to make a general statement to the effect that the requirement to provide security is reduced for devices using IPv4 Link-Local addresses as a sole means of access.

In all cases, whether or not IPv4 Link-Local addresses are used, it is necessary for implementers of devices supporting the Internet Protocol to analyze the known and credible threats to which a specific host or device might be subjected, and to the extent that it is feasible, to provide security mechanisms which ameliorate or reduce the risks associated with such threats.

6. Application Programming Considerations

Use of IPv4 Link-Local autoconfigured addresses presents additional challenges to writers of applications and may result in existing application software failing.

6.1. Address Changes, Failure and Recovery

IPv4 Link-Local addresses used by an application may change over time. Some application software encountering an address change will fail. For example, existing client TCP connections will be aborted, servers whose addresses change will have to be rediscovered, blocked reads and writes will exit with an error condition, and so on.

Vendors producing application software which will be used on IP implementations supporting IPv4 Link-Local address configuration SHOULD detect and cope with address change events. Vendors producing IPv4 implementations supporting IPv4 Link-Local address configuration SHOULD expose address change events to applications.

6.2. Limited Forwarding of Locators

IPv4 Link-Local addresses MUST NOT be forwarded via an application protocol (for example in a URL), to a destination that is not on the same link. This is discussed further in Sections 2.9 and 3.

Existing distributed application software that forwards address information may fail. For example, FTP [RFC959] (when not using passive mode) transmits the IP address of the client. Suppose a client starts up and obtains its IPv4 configuration at a time when it

has only a Link-Local address. Later, the host gets a global IP address, and the client contacts an FTP server outside the local link. If the FTP client transmits its old Link-Local address instead of its new global IP address in the FTP "port" command, then the FTP server will be unable to open a data connection back to the client, and the FTP operation will fail.

6.3. Address Ambiguity

Application software run on a multi-homed host that supports IPv4 Link-Local address configuration on more than one interface may fail.

This is because application software assumes that an IPv4 address is unambiguous, that it can refer to only one host. IPv4 Link-Local addresses are unique only on a single link. A host attached to multiple links can easily encounter a situation where the same address is present on more than one interface, or first on one interface, later on another; in any case associated with more than one host. Most existing software is not prepared for this ambiguity. In the future, application programming interfaces could be developed to prevent this problem. This issue is discussed in Section 3.

7. Router Considerations

A router MUST NOT forward a packet with an IPv4 Link-Local source or destination address, irrespective of the router's default route configuration or routes obtained from dynamic routing protocols.

A router which receives a packet with an IPv4 Link-Local source or destination address MUST NOT forward the packet. This prevents forwarding of packets back onto the network segment from which they originated, or to any other segment.

8. IANA Considerations

The IANA has allocated the prefix 169.254/16 for the use described in this document. The first and last 256 addresses in this range (169.254.0.x and 169.254.255.x) are allocated by Standards Action, as defined in "Guidelines for Writing an IANA" (BCP 26) [RFC2434]. No other IANA services are required by this document.

9. Constants

The following timing constants are used in this protocol; they are not intended to be user configurable.

PROBE_WAIT	1 second	(initial random delay)
PROBE_NUM	3	(number of probe packets)
PROBE_MIN	1 second	(minimum delay till repeated probe)
PROBE_MAX	2 seconds	(maximum delay till repeated probe)
ANNOUNCE_WAIT	2 seconds	(delay before announcing)
ANNOUNCE_NUM	2	(number of announcement packets)
ANNOUNCE_INTERVAL	2 seconds	(time between announcement packets)
MAX_CONFLICTS	10	(max conflicts before rate limiting)
RATE_LIMIT_INTERVAL	60 seconds	(delay between successive attempts)
DEFEND_INTERVAL	10 seconds	(minimum interval between defensive ARPs).

10. References

10.1. Normative References

[RFC792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.

[RFC826] 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.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

10.2. Informative References

[802] IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture, ANSI/IEEE Std 802, 1990.

[802.3] ISO/IEC 8802-3 Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Common specifications - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, (also ANSI/IEEE Std 802.3- 1996), 1996.

[802.5] ISO/IEC 8802-5 Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Common specifications - Part 5: Token ring access method and physical layer specifications, (also ANSI/IEEE Std 802.5-1998), 1998.

[802.11] Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std. 802.11-1999, 1999.

[RFC959] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, October 1985.

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC2462] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.

[RFC3027] Holdrege, M. and P. Srisuresh, "Protocol Complications with the IP Network Address Translator", RFC 3027, January 2001.

[DNAv4] Aboba, B., "Detection of Network Attachment (DNA) in IPv4", Work in Progress, July 2004.

[LLMNR] Esibov, L., Aboba, B. and D. Thaler, "Linklocal Multicast Name Resolution (LLMNR)", Work in Progress, June 2004.

Acknowledgments

We would like to thank (in alphabetical order) Jim Busse, Pavani Diwanji, Donald Eastlake 3rd, Robert Elz, Peter Ford, Spencer Giacalone, Josh Graessley, Brad Hards, Myron Hattig, Hugh Holbrook, Christian Huitema, Richard Johnson, Kim Yong-Woon, Mika Liljeberg, Rod Lopez, Keith Moore, Satish Mundra, Thomas Narten, Erik Nordmark, Philip Nye, Howard Ridenour, Daniel Senie, Dieter Siegmund, Valery Smyslov, and Ryan Troll for their contributions.

Appendix A - Prior Implementations

A.1. Apple Mac OS 8.x and 9.x.

Mac OS chooses the IP address on a pseudo-random basis. The selected address is saved in persistent storage for continued use after reboot, when possible.

Mac OS sends nine DHCPDISCOVER packets, with an interval of two seconds between packets. If no response is received from any of these requests (18 seconds), it will autoconfigure.

Upon finding that a selected address is in use, Mac OS will select a new random address and try again, at a rate limited to no more than one attempt every two seconds.

Autoconfigured Mac OS systems check for the presence of a DHCP server every five minutes. If a DHCP server is found but Mac OS is not successful in obtaining a new lease, it keeps the existing autoconfigured IP address. If Mac OS is successful at obtaining a new lease, it drops all existing connections without warning. This may cause users to lose sessions in progress. Once a new lease is obtained, Mac OS will not allocate further connections using the autoconfigured IP address.

Mac OS systems do not send packets addressed to a Link-Local address to the default gateway if one is present; these addresses are always resolved on the local segment.

Mac OS systems by default send all outgoing unicast packets with a TTL of 255. All multicast and broadcast packets are also sent with a TTL of 255 if they have a source address in the 169.254/16 prefix.

Mac OS implements media sense where the hardware (and driver software) supports this. As soon as network connectivity is detected, a DHCPDISCOVER will be sent on the interface. This means that systems will immediately transition out of autoconfigured mode as soon as connectivity is restored.

A.2. Apple Mac OS X Version 10.2

Mac OS X chooses the IP address on a pseudo-random basis. The selected address is saved in memory so that it can be re-used during subsequent autoconfiguration attempts during a single boot of the system.

Autoconfiguration of a Link-Local address depends on the results of the DHCP process. DHCP sends two packets, with timeouts of one and two seconds. If no response is received (three seconds), it begins autoconfiguration. DHCP continues sending packets in parallel for a total time of 60 seconds.

At the start of autoconfiguration, it generates 10 unique random IP addresses, and probes each one in turn for 2 seconds. It stops probing after finding an address that is not in use, or the list of addresses is exhausted.

If DHCP is not successful, it waits five minutes before starting over again. Once DHCP is successful, the autoconfigured Link-Local address is given up. The Link-Local subnet, however, remains configured.

Autoconfiguration is only attempted on a single interface at any given moment in time.

Mac OS X ensures that the connected interface with the highest priority is associated with the Link-Local subnet. Packets addressed to a Link-Local address are never sent to the default gateway, if one is present. Link-local addresses are always resolved on the local segment.

Mac OS X implements media sense where the hardware and driver support it. When the network media indicates that it has been connected, the autoconfiguration process begins again, and attempts to re-use the previously assigned Link-Local address. When the network media indicates that it has been disconnected, the system waits four seconds before de-configuring the Link-Local address and subnet. If the connection is restored before that time, the autoconfiguration process begins again. If the connection is not restored before that time, the system chooses another interface to autoconfigure.

Mac OS X by default sends all outgoing unicast packets with a TTL of 255. All multicast and broadcast packets are also sent with a TTL of 255 if they have a source address in the 169.254/16 prefix.

A.3. Microsoft Windows 98/98SE

Windows 98/98SE systems choose their IPv4 Link-Local address on a pseudo-random basis. The address selection algorithm is based on computing a hash on the interface's MAC address, so that a large collection of hosts should obey the uniform probability distribution in choosing addresses within the 169.254/16 address space. Deriving

the initial IPv4 Link-Local address from the interface's MAC address also ensures that systems rebooting will obtain the same autoconfigured address, unless a conflict is detected.

When in INIT state, the Windows 98/98SE DHCP Client sends out a total of 4 DHCPDISCOVERs, with an inter-packet interval of 6 seconds. When no response is received after all 4 packets (24 seconds), it will autoconfigure an address.

The autoconfigure retry count for Windows 98/98SE systems is 10. After trying 10 autoconfigured IPv4 addresses, and finding all are taken, the host will boot without an IPv4 address.

Autoconfigured Windows 98/98SE systems check for the presence of a DHCP server every five minutes. If a DHCP server is found but Windows 98 is not successful in obtaining a new lease, it keeps the existing autoconfigured IPv4 Link-Local address. If Windows 98/98SE is successful at obtaining a new lease, it drops all existing connections without warning. This may cause users to lose sessions in progress. Once a new lease is obtained, Windows 98/98SE will not allocate further connections using the autoconfigured IPv4 Link-Local address.

Windows 98/98SE systems with an IPv4 Link-Local address do not send packets addressed to an IPv4 Link-Local address to the default gateway if one is present; these addresses are always resolved on the local segment.

Windows 98/98SE systems by default send all outgoing unicast packets with a TTL of 128. TTL configuration is performed by setting the Windows Registry Key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services:\Tcpip\ Parameters\DefaultTTL of type REG_DWORD to the appropriate value. However, this default TTL will apply to all packets. While this facility could be used to set the default TTL to 255, it cannot be used to set the default TTL of IPv4 Link-Local packets to one (1), while allowing other packets to be sent with a TTL larger than one.

Windows 98/98SE systems do not implement media sense. This means that network connectivity issues (such as a loose cable) may prevent a system from contacting the DHCP server, thereby causing it to auto-configure. When the connectivity problem is fixed (such as when the cable is re-connected) the situation will not immediately correct itself. Since the system will not sense the re-connection, it will remain in autoconfigured mode until an attempt is made to reach the DHCP server.

The DHCP server included with Windows 98SE Internet Connection Sharing (ICS) (a NAT implementation) allocates out of the 192.168/16 private address space by default.

However, it is possible to change the allocation prefix via a registry key, and no checks are made to prevent allocation out of the IPv4 Link-Local prefix. When configured to do so, Windows 98SE ICS will rewrite packets from the IPv4 Link-Local prefix and forward them beyond the local link. Windows 98SE ICS does not automatically route for the IPv4 Link-Local prefix, so that hosts obtaining addresses via DHCP cannot communicate with autoconfigured-only devices.

Other home gateways exist that allocate addresses out of the IPv4 Link-Local prefix by default. Windows 98/98SE systems can use a 169.254/16 IPv4 Link-Local address as the source address when communicating with non-Link-Local hosts. Windows 98/98SE does not support router solicitation/advertisement. Windows 98/98SE systems will not automatically discover a default gateway when in autoconfigured mode.

A.4. Windows XP, 2000, and ME

The autoconfiguration behavior of Windows XP, Windows 2000, and Windows ME systems is identical to Windows 98/98SE except in the following respects:

Media Sense
Router Discovery
Silent RIP

Windows XP, 2000, and ME implement media sense. As soon as network connectivity is detected, a DHCPREQUEST or DHCPDISCOVER will be sent on the interface. This means that systems will immediately transition out of autoconfigured mode as soon as connectivity is restored.

Windows XP, 2000, and ME also support router discovery, although it is turned off by default. Windows XP and 2000 also support a RIP listener. This means that they may inadvertently discover a default gateway while in autoconfigured mode.

ICS on Windows XP/2000/ME behaves identically to Windows 98SE with respect to address allocation and NATing of Link-Local prefixes.

Authors' Addresses

Stuart Cheshire
Apple Computer, Inc.
1 Infinite Loop
Cupertino
California 95014, USA

Phone: +1 408 974 3207
EMail: rfc@stuartcheshire.org

Bernard Aboba
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Phone: +1 425 818 4011
EMail: bernarda@microsoft.com

Erik Guttman
Sun Microsystems
Eichhoelzelstr. 7
74915 Waibstadt Germany

Phone: +49 7263 911 701
EMail: erik@spybeam.org

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.