Automatically Choosing an IP Address in an Ad-Hoc IPv4 Network

<draft-ietf-dhc-ipv4-autoconfig-05.txt>

Status of this Memo

   This document is an Internet-Draft and is in full conformance with
   all provisions of Section 10 of RFC2026. Internet-Drafts are work-
   ing documents of the Internet Engineering Task Force (IETF), its
   areas, and its working groups. Note that other groups may also dis-
   tribute working documents as Internet-Drafts.

   Internet-Drafts are draft documents valid for a maximum of six
   months and may be updated, replaced, or obsoleted by other documents
   at any time. It is inappropriate to use Internet- Drafts as refer-
   ence material or to cite them other than as "work in progress."

    The list of current Internet-Drafts can be accessed at
    http://www.ietf.org/ietf/1id-abstracts.txt

    The list of Internet-Draft Shadow Directories can be accessed at
    http://www.ietf.org/shadow.html.

   Distribution of this memo is unlimited.

Abstract

   With operating systems appearing in more and more devices, as well
   as computers appearing in more and more aspects of everyday life,
   communication between networked devices is increasingly important.
   The communication mechanism between these devices must be able to
   not only support the office LAN environment, but must also scale to
   larger WANS and the internet.

   This draft describes a method by which a host may automatically give
   itself a link-local IPv4 address, so that it will be able to use IP
   applications in the absence of an IP address management mechanism,
   such as DHCP. This mechanism is in use today by a few operating
   systems, and additional information on those implementations is also
   provided.

1.   Introduction

   Now that networked applications are becoming more prevalent, operat-

ing systems are migrating towards more scalable network protocols
such as IP, allowing them to work in all sizes of environments.

However, there is a price to pay for this migration -- IP requires
configuration that other protocols (IPX, Appletalk) do not require.

Dynamic creation of usable ad-hoc networks is very useful when there
are only a few machines on the entire network. (For example, a
dentist's office may only have a couple of machines.) In order to
allow a site such as this to use IP, the machines must each be con-
figured with an IP address. OS's wish to retain the minimal confi-
guration that was necessary under their non-IP network stacks.

Dynamic configuration protocols such as DHCP [DHCP] allow a site
administrator to take care of the network configuration for a
machine remotely. By requesting network parameters via DHCP, the
site administrator may provide all information necessary without the
host's owner having to do anything. However, not all sites have a
central administrator to take care of this.

To accommodate unmanaged networks, the OS may decide to intelli-
gently choose an IP address for itself. These addresses are only
valid for the local network.

This document describes a method by which an OS may determine
whether or not to autoconfigure itself an IP address, as well as how
to inter-operate cleanly with an existing managed infrastructure,
allowing a host to easily move between managed and unmanaged network
segments.

1.1.   Conventions Used in the Document

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

1.2.   Terminology

     Site Administrator
                    A Site Administrator is the person or organiza-
                    tion responsible for handing out IP addresses to
                    client machines.

DHCP Client   A DHCP Client is an Internet host using DHCP to
                       obtain configuration parameters such as a network
                       address.

        DHCP Server   A DHCP Server is an Internet host that returns
                       configuration parameters to DHCP Clients.

1.3.   Usage Clarification

   This document describes a method by which a host may automatically
   choose an IPv4 address in the absence of a central service to main-
   tain and hand out addresses. This is not designed to replace this
   functionality, but to basicly provide it in small networks.

   This SHOULD NOT be used for large-scale networks. As more and more
   machines begin to use this mechanism on a network, startup times for
   these machines will begin to increase, as the chance of collisions
   will rise.

   Addresses allocated by this mechanism MUST NOT be routed by any net-
   work device. The addresses are designed to be link local addresses.
   Link local address are to be, by definition, restricted to the local
   network segment. Allocation of link-local addresses in an IPv6 net-
   work is described in [IPv6SAC].

   Finally, it should be noted that this functionality is designed to
   work on a host that is running a DHCP Client. Hosts that are not
   running a DHCP Client are not considered in this draft, and MUST NOT
   attempt to use these mechanisms to configure an IP address, as these
   no-dhcp-client hosts will not interoperate with the existing infras-
   tructure correctly. (This mechanism relies on the DHCP Server for
   some information, and if a host can not receive this information, it
   will not be fully informed.)

2.   To Choose or Not To Choose

   The first thing an Internet host should do is attempt to get an
   administratively assigned address using DHCP. Only after the DHCP
   process fails should a host use the methods later for autoconfigura-

tion.

2.1.   Obtaining an Address via DHCP

A host begins by requesting an IP address via DHCP [DHCP]. This is
done by sending out a DHCPDISCOVER message, with various tags set
indicating what options the DHCP Client would like to receive infor-
mation for [DHCPOPT]. The DHCP Client SHOULD also send the DHCP
AutoConfigure option described in [DHCPAC].

According to [DHCP], Section 4.4.1, the amount of time over which a
DHCP Client should listen for DHCPOFFERS is implementation depen-
dant. During this time, if a DHCPOFFER is received, network confi-
guration MUST occur as described in [DHCP] and [DHCPAC].

If, during this time, no valid DHCPOFFERS are received, the DHCP

Client is free to autoconfigure an IP address according to section 3
of this document.

2.2.   Rebinding an Existing IP Address

If the DHCP Client already has an existing IP address, it MUST fol-
low the instructions outlined in [DHCP]. If the client winds up
back in the INIT state, refer to section 2.1 of this document.

3.   Choosing an IP Address

Once a DHCP Client has determined it must auto-configure an IP
address, it chooses an address. The algorithm for choosing an
address is implementation dependant. The address range to use MUST
be "169.254/16", which is registered with the IANA as the LINKLOCAL
net.

If choosing an address in this range, the DHCP Client MUST NOT use
the first 256 or the last 256, as these are reserved for future use.

When an address is chosen, the DHCP Client MUST test to see if the
address is already in use. If the network address appears to be in
use, the client MUST choose another address, and try again. The
client MUST keep choosing addresses until it either finds one, or it
has tried more then the autoconfig-retry count. The autoconfig-

retry count is implementation specific, and should be based on the
algorithm used for choosing an IP address. This retry count is
present to make sure that DHCP Clients auto-configuring on busy
auto-configured network segments do not loop infinitely looking for
an IP address.

3.1.   Determining Whether or Not an Address is in Use

If the client is on a network that supports ARP, the client may
issue an ARP request for the suggested address. When broadcasting
an ARP request for the suggested address, the client MUST fill in
its own hardware address as the sender's hardware address, and all
0s as the sender's IP address, to avoid confusing ARP caches in
other hosts on the same subnet. This ARP request with a sender IP
address of all 0s is referred to as an "ARP probe".

While waiting for a possible response to this request, the client
MUST also listen for other ARP probes for the same address (but not
from its own hardware address). This will occur if two (or more)
hosts are attempting to autoconfigure the exact same address. If
the client receives a response to the ARP request, or sees another
ARP probe for the same address, it MUST consider the address as
being in use, and move on.

Clients SHOULD randomize their IP address selection algorithm, to
help keep clients from becoming stuck in a loop with other clients
while trying to allocate an address. Algorithm designers MUST make
sure that their algorithm will not cause the client IP selection
process to become stuck when interoperating with any other possible
algorithms.

4.   Ongoing Checks for a DHCP Server

When the client originally sent out it's request, there may have
been a network problem stopping the DHCP Server from responding. To
make sure this is not the case, a host with an auto-configured IP
address MUST keep checking for an active DHCP Server. To do this,
the host MUST attempt to fetch an IP address as described in section
1 of this document.

When rechecking, if the host has determined no DHCP Server is
responding, it MUST wait a period of time and try again. This

interval is tunable. The suggested default for Ethernet implementa-
tions is to check every 5 minutes.

If the host receives a response from a DHCP Server, it MUST respond
and attempt to obtain a lease from the server (per the DHCP specifi-
cation). If the client is successful in obtaining a new lease, and
the internet host does not support multiple addresses on the inter-
face being configured, it MUST drop any existing auto-configured IP
address, and all active connections, while moving to the new
address. If the internet host does support multiple addresses on
the interface, it MAY keep the auto-configured address active.

If the DHCP response is an AutoConfigure [DHCPAC] response set to
"DoNOTAutoConfigure", the host MUST drop all connections, give up
any existing auto-configured IP address, and continue checking for a
DHCP server.

WARNING: If the client drops all connections, the user may lose data
being access over the existing connections. The operating system
SHOULD do everything it can to allow existing clients to finish
using all existing connections before closing them. It SHOULD NOT
allocate any further connections using the auto-configured IP
address.

5.   Current Vendor Implementations

As of this writing, Microsoft and Apple have operating systems that
contain this functionality. Descriptions of the implementation
dependant parts are listed below.

5.1.   Microsoft Windows 98

With the initial release of Windows 98, Microsoft introduced auto-
configuration functionality. When developed, the AutoConfig
[DHCPAC] specification did not exist, so the initial release does
not contain this functionality.

The Win98 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 auto-configure an address.

The auto-configure retry count for Windows 98 is 10. After trying
10 auto-configured IP addresses, and finding all are taken, the host
will boot without an IP address.

5.2.   Apple MacOS 8.5

MacOS 8.5 sends three DHCPDISCOVER packets, with timeouts of 4, 8,
and then 16 seconds. When no response is received from all of these
requests (28 seconds), it will auto-configure.

The auto-configure retry count for MacOS 8.5 is 10. After trying 10
auto-configured IP addresses, and finding all are taken, the host
will boot without an IP address.

6.   Security Considerations

The use of this functionality may open a network host to new Denial
Of Service (DOS) attacks. In particular, a host that previously did
not have an IP address, and no IP stack running, was not succeptable
to IP based DOS attacks, as there was no IP stack configured to
interpret these packets.

However, the addition of this functionality to an OS may cause IP
stacks to be capable of receiving and interpreting information that
the host was not previously configured to receive. As this host is
now interpreting IP communications, it is now open to IP based DOS
attacks.

Another security concern is the DOS attack that may be made on the
local subnet which stops all machines from being able to allocate an
IP address. A malicious host on the local wire may listen for ARP
probes, and respond with it's own ARP probe. This will stop the
auto-configuring machine from using that address, and it will move
on to the next one. Eventually, it will run out of addresses to
attempt, and will give up. The use of DHCP removes this attack,
leaving only the concerns described in [DHCP].

Finally, machines that rely on this for communication over a large
network may allocate the same address if the network itself is seg-
mented when the machines boot. If the link between two machines is
down when they boot, they may both auto-configure the same address.

However, when the network link returns, there will be numerous prob-
lems (ARP caches, etc.) There is currently no way to solve this
auto-configuration problem without causing all hosts involved to
re-autoconfigure IP addresses. The use of DHCP to configure hosts
on a subnet will solve this, and hosts that implement this confi-
guration mechanism will behave appropriately on a DHCP managed net-
work in which the DHCP server is not initially available.

7.   Acknowledgments

   I'd like to thank Microsoft and Apple for their help in writing this
   document.

8.   Copyright

   Copyright (C) The Internet Society 1999. All Rights Reserved.

   This document and translations of it may be copied and furnished to
   others, and derivative works that comment on or otherwise explain it
   or assist in its implementation may be prepared, copied, published
   and distributed, in whole or in part, without restriction of any
   kind, provided that the above copyright notice and this paragraph
   are included on all such copies and derivative works. However, this
   document itself may not be modified in any way, such as by removing
   the copyright notice or references to the Internet Society or other
   Internet organizations, except as needed for the purpose of develop-
   ing Internet standards in which case the procedures for copyrights
   defined in the Internet Standards process must be followed, or as
   required to translate it into languages other than English.

   The limited permissions granted above are perpetual and will not be
   revoked by the Internet Society or its successors or assigns.

   This document and the information contained herein is provided on an
   "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING
   TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
   BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION
   HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF
   MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

9.   References

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

        <ftp://ftp.isi.edu/in-notes/rfc2131.txt>

     [DHCPAC] Troll, R., "DHCP Option to Disable Stateless Auto-
          Configuration in IPv4 Clients", RFC 2563, May 1999

        <ftp://ftp.isi.edu/in-notes/rfc2563.txt>

     [DHCPOPT] Alexander, S. and Droms, R., "DHCP Options and BOOTP Ven-
          dor Extension", RFC 2132, March 1997

        <ftp://ftp.isi.edu/in-notes/rfc2132.txt>

     [IPv6SAC] Thomson, S. and Narten, T., "IPv6 Stateless Address Auto-
          configuration", RFC 2462, December 1998

        <ftp://ftp.isi.edu/in-notes/rfc2462.txt>

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

        <ftp://ftp.isi.edu/in-notes/rfc2119.txt>

10.   Author's Address

     Ryan Troll
     Excite@Home Network
     450 Broadway
     Redwood City, CA 94063

     Phone: (650) 556-6031
     EMail: rtroll@excitehome.net