     Customizing DHCP Configuration on the Basis of Network Topology

Abstract

   DHCP servers have evolved over the years to provide significant
   functionality beyond that described in the DHCP base specifications.
   One aspect of this functionality is support for context-specific
   configuration information. This memo describes some such features
   and explains their operation.

Status of This Memo

   This document is not an Internet Standards Track specification; it is
   published for informational purposes.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Not all documents
   approved by the IESG are a candidate for any level of Internet
   Standard; see Section 2 of RFC 7841.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc7969.

Table of Contents

1. Introduction

   The DHCPv4 [RFC2131] and DHCPv6 [RFC3315] protocol specifications
   describe how addresses can be allocated to clients based on network
   topology information provided by the DHCP relay infrastructure.
   Address allocation decisions are integral to the allocation of
   addresses and prefixes in DHCP.

   The DHCP protocol also describes mechanisms for provisioning devices
   with additional configuration information, for example, DNS [RFC1034]
   server addresses, default DNS search domains, and similar
   information.

   Although it was the intent of the authors of these specifications
   that DHCP servers would provision devices with configuration
   information appropriate to each device's location on the network,
   this practice was never documented, much less described in detail.

   Existing DHCP server implementations do in fact provide such
   capabilities; the goal of this document is to describe those
   capabilities for the benefit of both operators and protocol designers
   who may wish to use DHCP as a means for configuring their own
   services but may not be aware of the capabilities provided by most
   modern DHCP servers.

2. Terminology

   o CPE device: Customer Premise Equipment device. Typically a router
      belonging to the customer that connects directly to the provider
      link.

   o DHCP, DHCPv4, and DHCPv6: DHCP refers to the Dynamic Host
      Configuration Protocol in general and applies to both DHCPv4 and
      DHCPv6. The terms DHCPv4 and DHCPv6 are used in contexts where it
      is necessary to avoid ambiguity and explain differences.

   o PE router: Provider Edge router. The provider router closest to
      the customer.

   o Routable IP address: An IP address with a scope of use wider than
      the local link.

   o Shared subnet: A case where two or more subnets of the same
      protocol family are available on the same link. 'Shared subnet'
      terminology is typically used in Unix environments. It is
      typically called 'multinet' in the Windows environment. The
      administrative configuration inside a Microsoft DHCP server is
      called 'DHCP Superscope'.

   o Link or local link: A layer 2 network link, as defined in
      Section 1.2 of [RFC3297].

   o Link subset: A portion of a link containing a subset of all the
      connection points on that link, which may be used to finely
      determine the physical location of a set of clients or may be used
      to determine topology to a finer degree of detail than the set of
      all locations at which that particular link is available. The
      smallest link subset is a single link attachment point, for
      example, a port on a layer 2 switch.

3. Identifying Client's Location by DHCP Servers

   Figure 1 illustrates a small hierarchy of network links with Link D
   serving as a backbone to which the DHCP server is attached.

   Figure 2 illustrates a more complex case. Although some of its
   aspects are unlikely to be seen in actual production networks, they
   are beneficial for explaining finer aspects of the DHCP protocols.
   Note that some nodes act as routers (which forward all IP traffic)
   and some are relay agents (i.e., they run DHCP-specific software that
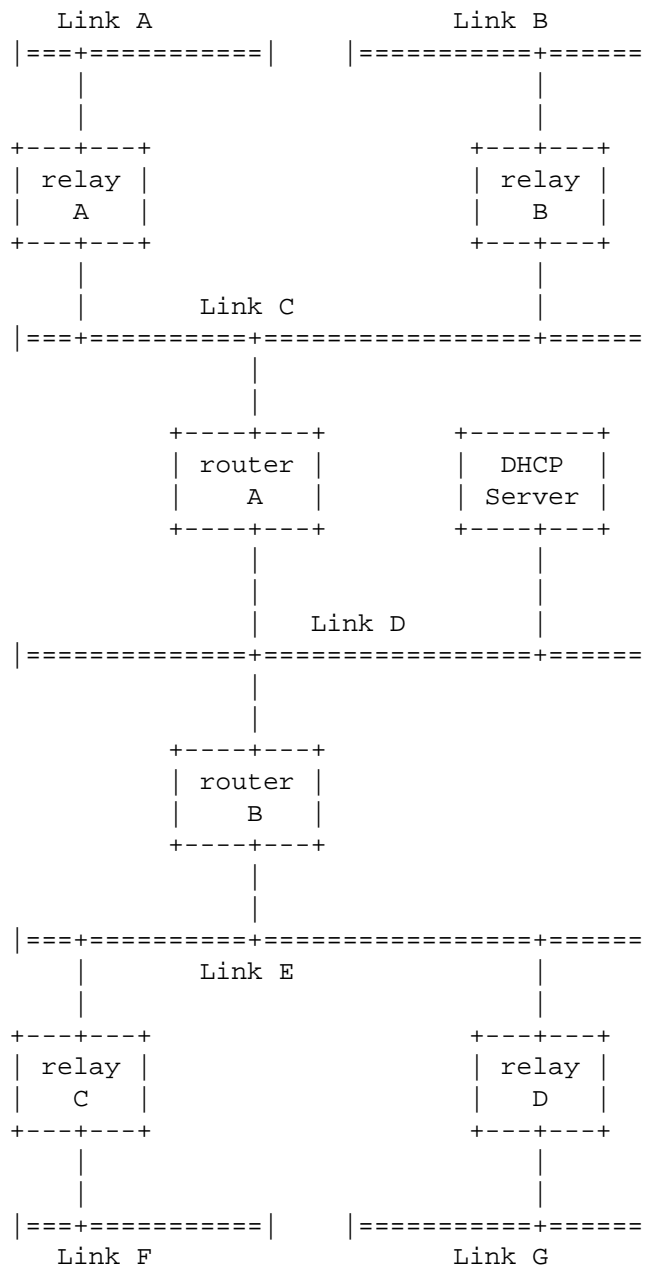   forwards only DHCP traffic).

```
           Link A                       Link B
     |===+==========|     |==========+======|
         |                           |
         |                           |
     +---+---+                   +---+---+
     | relay |                   | relay |
     |   A   |                   |   B   |
     +---+---+                   +---+---+
         |                           |
         |          Link C           |
     |===+==========+================+======|
                    |
                    |
         +----+---+          +--------+
         | router |          |  DHCP  |
         |   A    |          | Server |
         +----+---+          +----+---+
              |                   |
              |                   |
              |       Link D      |
     |============+================+======|
                  |
                  |
         +----+---+
         | router |
         |   B    |
         +----+---+
              |
              |
     |===+==========+================+======|
         |       Link E              |
         |                           |
     +---+---+                   +---+---+
     | relay |                   | relay |
     |   C   |                   |   D   |
     +---+---+                   +---+---+
         |                           |
         |                           |
     |===+==========|     |==========+======|
        Link F                    Link G
```

Figure 1: A Simple Network with a Small Hierarchy of Links

```
        Link A                    Link B                   Link H
    |===+=========|       |=========+=====|      |======+=====|
        |                          |                     |
        |                          |                     |
    +---+---+                  +---+---+             +---+---+
    | relay |                  | relay |             | relay |
    |   A   |                  |   B   |             |   G   |
    +---+---+                  +---+---+             +---+---+
        |                          |                     |
        |        Link C            |                     | Link J
    |===+=========+============+=====|     |======+=====|
        |                          |                     |
        |                          |                     |
    +----+---+              +--------+             +---+---+
    | router |              |  DHCP  |             | relay |
    |   A    |              | Server |             |   F   |
    +----+---+              +----+---+             +---+---+
         |                       |                     |
         |                       |                     |
         |        Link D         |                     |
|=============+========+=======+=============+=====|
         |         |                      
         |         |                      
    +----+---+ +---+---+                
    | router | | relay |                
    |   B    | |   E   |                
    +----+---+ +---+---+                
         |         |                    
         |         |                    
    |===+=========+=========+=======+=====|
         |        Link E                |
         |                              |
    +---+---+                      +---+---+
    | relay |                      | relay |
    |   C   |                      |   D   |
    +---+---+                      +---+---+
         |                              |
         |                              |
    |===+=========|       |==========+=====|
        Link F                   Link G
```
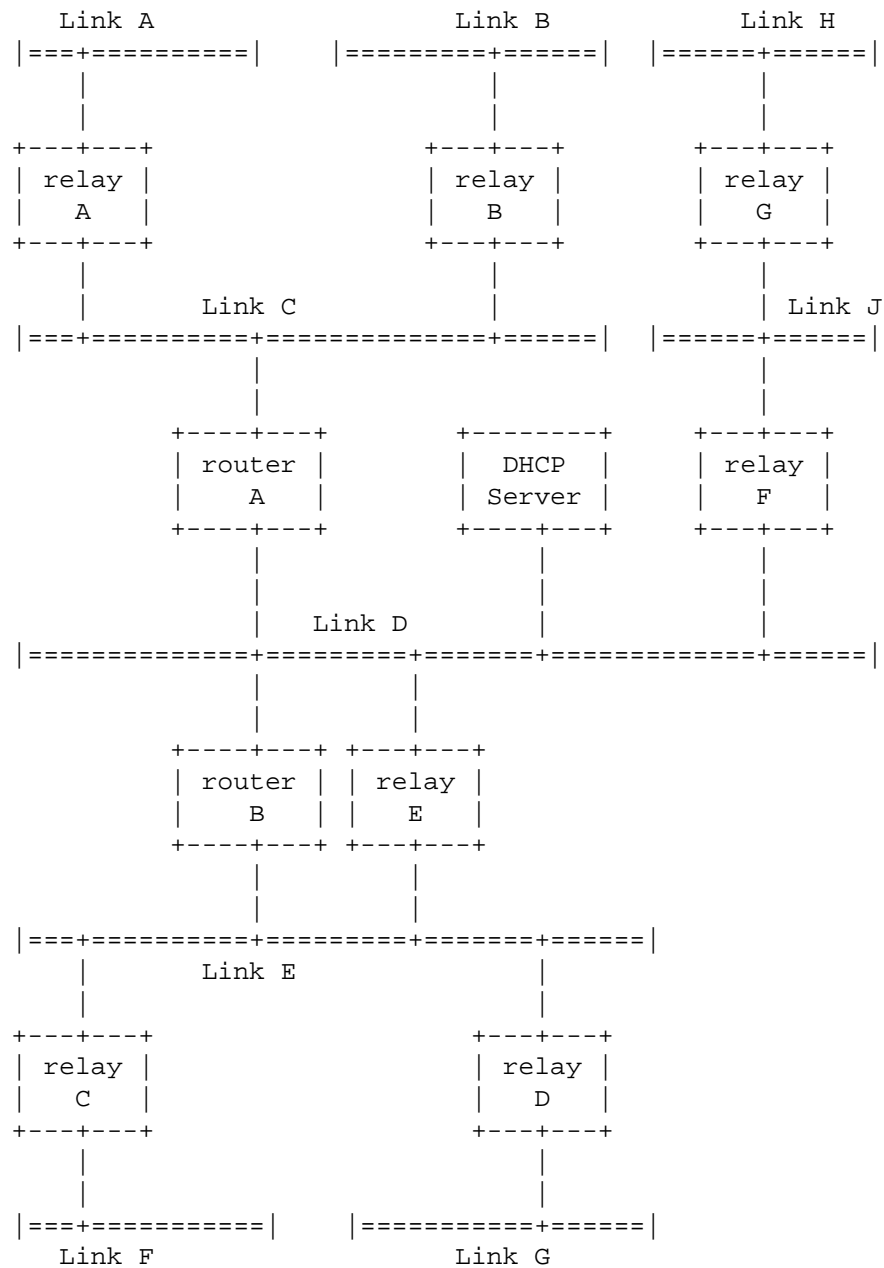
                       Figure 2: Complex Network

   These diagrams allow us to represent a variety of different network
   configurations and illustrate how existing DHCP servers can provide
   configuration information customized to the particular location from
   which a client is making its request.

It is important to understand the background of how DHCP works when
considering those diagrams. It is assumed that the DHCP clients may
not have routable IP addresses when they are attempting to obtain
configuration information.

The reason for making this assumption is that one of the functions of
DHCP is to bootstrap the DHCP client's IP address configuration. If
the client does not yet have an IP address configured, it cannot
route packets to an off-link DHCP server; therefore, some kind of
relay mechanism is required.

The details of how packet delivery between clients and servers works
are different between DHCPv4 and DHCPv6, but the essence is the same:
whether or not the client actually has an IP configuration, it
generally communicates with the DHCP server by sending its requests
to a DHCP relay agent on the local link; this relay agent, which has
a routable IP address, then forwards the DHCP requests to the DHCP
server (directly or via other relays). In later stages of the
configuration, when the client has acquired an address and certain
conditions are met, it is possible for the client to send packets
directly to the server, thus bypassing the relays. The conditions
for such behavior are different for DHCPv4 and DHCPv6 and are
discussed in Sections 3.1 and 3.2.

To determine the client's point of attachment and link-specific
configuration, the server typically uses the client-facing IP address
of the relay agent. In some cases, the server may use the routable
IP address of the client if the client has the routable IP address
assigned to its interface and it is transmitted in the DHCP message.
The server is then able to determine the client's point of attachment
and select the appropriate subnet- or link-specific configuration.

Sometimes it is useful for the relay agents to provide additional
information about the topology. A number of extensions have been
defined for this purpose. The specifics are different, but the core
principle remains the same: the relay agent knows exactly where the
original request came from, so it provides an identifier that will
help the server to choose appropriate address pool and configuration
parameters. Examples of such options are mentioned in the following
sections.

Finally, clients may be connected to the same link as the server, so
no relay agents are required. In such cases, the DHCPv4 server
typically uses the IPv4 address assigned to the network interface
over which the transmission was received to select an appropriate
subnet. This is more complicated for DHCPv6, as the DHCPv6 server is
not required to have any globally unique addresses. In such cases,

additional configuration information may need to be required. Some
servers allow indicating that a given subnet is directly reachable
over a specific local network interface.

3.1. DHCPv4-Specific Behavior

   In some cases in DHCPv4, when a DHCPv4 client has a routable IPv4
   address, the message is unicast to the DHCPv4 server rather than
   going through a relay agent. Examples of such transmissions are
   renewal (DHCPREQUEST) and address release (DHCPRELEASE).

   The relay agent that receives the client's message sets the giaddr
   field to the address of the network interface the message was
   received on. The relay agent may insert a relay agent option
   [RFC3046].

   There are several options defined that are useful for subnet
   selection in DHCPv4. [RFC3527] defines the Link Selection sub-option
   that is inserted by a relay agent. This option is particularly
   useful when the relay agent needs to specify the subnet/link on which
   a DHCPv4 client resides, which is different from an IP address that
   can be used to communicate with the relay agent. The Virtual Subnet
   Selection (VSS) sub-option, specified in [RFC6607], can also be added
   by a relay agent to specify information in a VPN environment. In
   certain cases, it is useful for the client itself to specify the
   Virtual Subnet Selection option, e.g., when there are no relay agents
   involved during the VPN setup process.

   Another option that may influence the subnet selection is the IPv4
   Subnet Selection option, defined in [RFC3011], which allows the
   client to explicitly request allocation from a given subnet.

3.2. DHCPv6-Specific Behavior

   In DHCPv6, unicast communication is possible in cases where the
   server is configured with a Server Unicast option (see Section 22.12
   in [RFC3315]) and clients are able to take advantage of it. In such
   cases, once a client is assigned a (presumably global) address, it is
   able to contact the server directly, bypassing any relays. It should
   be noted that such a mode is completely controllable by
   administrators in DHCPv6. (They may simply choose to not configure
   the Server Unicast option, thus forcing clients to always send their
   messages via relay agents in every case).

   The DHCPv6 protocol [RFC3315] defines two core mechanisms that allow
   a server to distinguish which link the relay agent is connected to.
   The first mechanism is the link-address field in the Relay-forward
   and Relay-reply messages. The link-address field uniquely identifies

the link and should not be mistaken for a link-local address. In
normal circumstances, this is the solution that is easiest to
maintain, as existing address assignments can be used and no
additional administrative actions (like assigning dedicated
identifiers for each relay agent, making sure they are unique, and
maintaining a list of such identifiers) are needed. It requires,
however, for the relay agent to have an address with a scope larger
than link-local configured on its client-facing interface.

The second mechanism uses an Interface-ID option (see Section 22.18
of [RFC3315]) inserted by the relay agent, which identifies the link
that the client is connected to. This mechanism may be used when the
relay agent does not have a globally unique address or Unique Local
Address (ULA) [RFC4193] configured on its client-facing interface,
thus making the first mechanism not feasible. If the interface-id is
unique within an administrative domain, the interface-id value may be
used to select the appropriate subnet. As there is no guarantee for
the uniqueness ([RFC3315] only mandates the interface-id to be unique
within a single relay agent context), it is up to the administrator
to check whether the relay agents deployed use unique interface-id
values. If the interface-id values are not unique, the Interface-ID
option cannot be used to determine the client's point of attachment.

It should be noted that Relay-forward and Relay-reply messages are
exchanged between relays and servers only. Clients are never exposed
to those messages. Also, servers never receive Relay-reply messages.
Relay agents must be able to process both Relay-forward (sending an
already relayed message further towards the server when there is more
than one relay agent in a chain) and Relay-reply (sending back the
response towards the client when there is more than one relay agent
in a chain).

For completeness, we also mention an uncommon but valid case where
relay agents use a link-local address in the link-address field in
relayed Relay-forward messages. This may happen if the relay agent
doesn't have any address with a larger scope on the interface
connected to that specific link. Even though link-local addresses
cannot be automatically used to associate the relay agent with a
given link, with additional configuration information, the server may
still be able to select the proper link.

This requires that the DHCP server has a way of associating a
particular link-local address with a particular link. The network
administrator can then explicitly configure the DHCP server to
recognize that the particular link-address field in a relay message
refers to that link.

There are two ways that this can work. One is that the DHCP server
can provide a mechanism that explicitly associates the link-local
address with a link. In this case, the network administrator simply
determines the link-local address of the relay agent on a particular
link, which we are presuming to be stable, and configures an
association between that address and the link.

If the DHCP server doesn't explicitly provide such a mechanism, it
may still provide a "shared subnet" mechanism (see Section 8). If it
does, the shared subnet mechanism can be used to explicitly associate
a link-local address with a link. To do this, the network
administrator creates a shared subnet association for the link, if
one does not already exist. The network administrator then
configures a /128 subnet that contains just the link-local address of
the relay agent. The administrator then adds this new /128 to the
shared subnet. Now, when a DHCP message comes in with that link-
layer address in the link-address field, the correct shared network
will be selected.

DHCPv6 also has support for more finely grained link identification
using Lightweight DHCPv6 Relay Agents (LDRAs) [RFC6221]. In this
case, the link-address field is set to an unspecified address (::),
but the DHCPv6 server also receives an Interface-ID option from the
relay agent that can be used to more precisely identify the client's
location on the network. It is possible to mix LDRA and regular
relay agents in the same network. See Sections 7.2 and 7.3 in
[RFC6221] for detailed examples.

What this means in practice is that the DHCP server has sufficient
information in all cases to pinpoint the link to which the client is
connected. In some cases, it may additionally be possible to
pinpoint the particular link subset to which the client is connected.

In all cases, then, the DHCPv6 server will have a link-identifying IP
address, and in some cases, it may also have a link-specific
identifier (e.g., the Interface-ID option or the Link Address option
defined in Section 5 of [RFC6977]). It should be noted that the
link-specific identifier is unique only within the scope of the link-
identifying IP address. For example, the link-specific identifier of
"eth0" assigned to a relay agent using IPv6 address 2001:db8::1 is
distinct from an "eth0" identifier used by a different relay agent
with address 2001:db8::2.

It is also possible for link-specific identifiers to be nested so
that the actual identifier that identifies the specific link subset
is an aggregate of two or more identifiers sent by a set of LDRAs in
a chain; in general, this functions exactly as if a single identifier
were received from a single LDRA, so we do not treat it specially in

   the discussion below, but sites that use chained LDRA configurations
   will need to be aware of this when configuring their DHCPv6 servers.

   The Virtual Subnet Selection options, present in DHCPv4, are also
   defined for DHCPv6. The use case is the same as in DHCPv4: the relay
   agent inserts VSS options that can help the server to select the
   appropriate subnet with its address pool and associated configuration
   options. See [RFC6607] for details.

4. Simple Subnetted Network

   Consider Figure 1 in the context of a simple subnetted network. In
   this network, there are four leaf subnets on which DHCP clients will
   be configured: Links A, B, F, and G. Relays A, B, C, and D in this
   example are represented in the diagram as IP routers with an embedded
   relay function, because this is a very typical configuration, but the
   relay function can also be provided in a separate node on each link.

   In a simple network like this, there may be no need for link-specific
   configuration in DHCPv6, since local routing information is delivered
   through router advertisements. However, in IPv4, it is very typical
   to configure the default route using DHCP; in this case, the default
   route will be different on each link. In order to accomplish this,
   the DHCP server will need link-specific configuration for the default
   route.

   To illustrate, we will use an example from a hypothetical DHCP server
   that uses a simple JSON notation [RFC7159] for configuration.
   Although we know of no DHCP server that uses this specific syntax,
   most modern DHCP servers provide similar functionality.

```
{
    "prefixes": {
        "192.0.2.0/26": {
            "options": {
                "routers": ["192.0.2.1"]
            },
            "on-link": ["A"]
        },
        "192.0.2.64/26": {
            "options": {
                "routers": ["192.0.2.65"]
            },
            "on-link": ["B"]
        },
        "192.0.2.128/26": {
            "options": {
                "routers": ["192.0.2.129"]
            },
            "on-link": ["F"]
        },
        "192.0.2.192/26": {
            "options": {
                "routers": ["192.0.2.193"]
            },
            "on-link": ["G"]
        }
    }
}
```

Figure 3: Configuration Example

In Figure 3, we see a configuration example for this scenario: a set
of prefixes, each of which has a set of options and a list of links
for which it is on-link. We have defined one option for each prefix:
a routers option. This option contains a list of values; each list
only has one value, and that value is the IP address of the router
specific to the prefix.

When the DHCP server receives a request, it searches the list of
prefixes for one that encloses the link-identifying IP address
provided by the client or relay agent. The DHCP server then examines
the options list associated with that prefix and returns those
options to the client.

So, for example, a client connected to Link A in the example would
have a link-identifying IP address within the 192.0.2.0/26 prefix, so
the DHCP server would match it to that prefix. Based on the
configuration, the DHCP server would then return a routers option

containing a single IP address: 192.0.2.1. A client on Link F would
have a link-identifying address in the 192.0.2.128/26 prefix and
would receive a routers option containing the IP address 192.0.2.129.

5. Relay Agent Running on a Host

A relay agent is DHCP software that may be run on any IP node.
Although it is typically run on a router, this is by no means
required by the DHCP protocol. The relay agent is simply a service
that operates on a link, receiving link-local multicasts (IPv6) or
broadcasts (IPv4) and relaying them, using IP routing, to a DHCP
server. As long as the relay has an IP address on the link and a
default route or a more specific route through which it can reach a
DHCP server, it need not be a router or even have multiple
interfaces.

A relay agent can be run on a host connected to two links. That case
is presented in Figure 2. There is router B that is connected to
Links D and E. At the same time, there is also a host that is
connected to the same links. The relay agent software is running on
that host. That is uncommon but is a valid configuration.

6. Cascaded Relays

Let's observe another case, shown in Figure 2. Note that in this
configuration, the clients connected to Link G will send their
requests to relay D, which will forward its packets directly to the
DHCP server. That is typical but not the only possible
configuration. It is possible to configure relay agent D to forward
client messages to relay E, which in turn will send them to the DHCP
server. This configuration is sometimes referred to as "cascaded
relay agents".

Note that the relaying mechanism works differently in DHCPv4 and in
DHCPv6. In DHCPv4, only the first relay is able to set the giaddr
field in the DHCPv4 packet. Any following relays that receive that
packet will not change it as the server needs giaddr information from
the first relay (i.e., the closest to the client). The server will
send the response back to the giaddr address, which is the address of
the first relay agent that saw the client's message. That means that
the client messages travel on a different path than the server's
responses. A message from a client connected to Link G will pass
through relay D, then through relay E, to reach the server. A
response message will be sent from the server to relay D via router
B, and relay D will send it to the client on Link G.

Relaying in DHCPv6 is more structured. Each relay agent encapsulates
a packet that is destined to the server and sends it towards the
server. Depending on the configuration, that can be a server's
unicast address, a multicast address, or the next relay agent
address. The next relay repeats the encapsulation process. Although
the resulting packet is more complex (may have up to 32 levels of
encapsulation if the packet traveled through 32 relays), every relay
may insert its own options, and it is clear which relay agent
inserted which option.

7. Regional Configuration Example

In the Figure 2 example, Link C is a regional backbone for an ISP.
Link E is also a regional backbone for that ISP. Relays A, B, C, and
D are PE routers, and Links A, B, F, and G are actually link
aggregators with individual layer 2 circuits to each customer -- for
example, the relays might be Digital Subscriber Line Access
Multiplexers (DSLAMs) or cable head-end systems. At each customer
site, we assume there is a single CPE device attached to the link.

We further assume that Links A, B, F, and G are each addressed by a
single prefix, although it would be equally valid for each CPE device
to be numbered on a separate prefix.

In a real-world deployment, there would likely be many more than two
PE routers connected to each regional backbone; we have kept the
number small for simplicity.

In the example presented in Figure 4, the goal is to configure all
the devices within a region with server addresses local to that
region, so that service traffic does not have to be routed between
regions unnecessarily.

```
{
    "prefixes": {
        "2001:db8::/40": {
            "on-link": ["A"]
        },
        "2001:db8:100::/40": {
            "on-link": ["B"]
        },
        "2001:db8:200::/40": {
            "on-link": ["F"]
        },
        "2001:db8:300::/40": {
            "on-link": ["G"]
        }
    },
    "links": {
        "A": {"region": "omashu"},
        "B": {"region": "omashu"},
        "F": {"region": "gaoling"},
        "G": {"region": "gaoling"}
    },
    "regions": {
        "omashu": {
            "options": {
                "SIP Server": ["sip.omashu.example.org"],
                "DNS Recursive Name Server": ["dns1.omashu.example.org",
                                "dns2.omashu.example.org"]
            }
        },
        "gaoling": {
            "options": {
                "SIP Server": ["sip.gaoling.example.org"],
                "DNS Recursive Name Server": ["dns1.gaoling.example.org",
                                "dns2.gaoling.example.org"]
            }
        }
    }
}
```

                    Figure 4: Regional Configuration Example

   In this example, when a request comes in to the DHCPv6 server with a
   link-identifying IP address in the 2001:db8::/40 prefix, it is
   identified as being on Link A. The DHCPv6 server then looks on the
   list of links to see what region the client is in. Link A is
   identified as being in omashu. The DHCPv6 server then looks up
   omashu in the set of regions and discovers a list of region-specific
   options.

The DHCPv6 server then resolves the domain names listed in the
options and sends a SIP Server option containing the IP addresses
that the resolver returned for sip.omashu.example.org and a DNS
Recursive Name Server option containing the IP addresses returned by
the resolver for dns1.omashu.example.org and dns2.omashu.example.org.
Depending on the server capability and configuration, it may cache
resolved responses for a specific period of time, repeat queries
every time, or even keep the response until reconfiguration or
shutdown. For more detailed discussion, see Section 7 of [RFC7227].

Similarly, if the DHCPv6 server receives a request from a DHCPv6
client where the link-identifying IP address is contained by the
prefix 2001:db8:300::/40, then the DHCPv6 server identifies the
client as being connected to Link G. The DHCPv6 server then
identifies Link G as being in the gaoling region and returns the SIP
Server and DNS Recursive Name Server options specific to that region.

As with the previous example, the exact configuration syntax and
structure shown above does not precisely match what existing DHCPv6
servers do, but the behavior illustrated in this example can be
accomplished with most existing modern DHCPv6 servers.

8. Multiple Subnets on the Same Link

There are scenarios where there is more than one subnet from the same
protocol family (i.e., two or more IPv4 subnets or two or more IPv6
subnets) configured on the same link. Such a configuration is often
referred to as 'shared subnets' in Unix environments or 'multinet' in
Microsoft terminology.

The most frequently mentioned use case is a network renumbering where
some services are migrated to the new addressing scheme, but some
aren't yet.

A second example is expanding the allocation space. In DHCPv4 and
for DHCPv6 Prefix Delegation, there could be cases where multiple
subnets are needed, because a single subnet may be too small to
accommodate the client population.

The third use case covers allocating addresses (or delegation
prefixes) that are not the same as topological information. For
example, the link-address is on prefix X, and the addresses to be
assigned are on prefix Y. This could be based on differentiating
information (i.e., whether the device is a CPE or cable modem in the
Data Over Cable Service Interface Specification (DOCSIS)) or just
because the link-address/giaddr is different from the actual
allocation space.

The fourth use case is a cable network, where cable modems and the
devices connected behind them are connected to the same layer 2 link.
However, operators want the cable modems and user devices to get
addresses from distinct address spaces, so users couldn't easily
access their modems' management interfaces.

To support such a configuration, additional differentiating
information is required. Many DHCP server implementations offer a
feature that is typically called "client classification". The server
segregates incoming packets into one or more classes based on certain
packet characteristics, e.g., the presence or value of certain
options or even a match between existing options. Servers require
additional information to handle such configuration, as they cannot
use the topographical property of the relay addresses alone to
properly choose a subnet. Exact details of such an operation are not
part of the DHCPv4 or DHCPv6 protocols and are implementation
dependent.

9. Security Considerations

This document explains existing practice with respect to the use of
Dynamic Host Configuration Protocol [RFC2131] and Dynamic Host
Configuration Protocol Version 6 [RFC3315]. The security
considerations for these protocols are described in their
specifications and in related documents that extend these protocols.

The mechanisms described in this document could possibly be exploited
by an attacker to misrepresent its point of attachment in the
network. This would cause the server to assign addresses, prefixes,
and other configuration options, which can be considered a leak of
information. In particular, this could be used as a preliminary
stage of an attack when the DHCP server leaks information about
available services in parts of the network the attacker does not have
access to.

There are several ways that such an attack can be prevented. First,
it is a common practice to filter DHCP traffic passing to clients
within a particular administrative domain from outside of that
domain, and also to filter DHCP traffic to clients outside of a
particular administrative domain from within that domain. Second,
the DHCP servers can be configured to not respond to traffic that is
coming from unknown subnets (i.e., those subnets the server is not
configured to serve). Third, some relays provide the ability to
reject messages that do not fit expected characteristics. For
example, the Cable Modem Termination System (CMTS) acting as a DHCP
relay detects if the Media Access Control (MAC) address specified in
chaddr in incoming DHCP messages matches the MAC address of the cable
modem it came from and can alter its behavior accordingly. Also,

relay agents and servers that are connected to clients directly can
reject traffic that looks as if it has passed a relay (this could
indicate the client is attempting to spoof a relay, possibly to
inject forged relay options).

There are a number of general DHCP recommendations that should be
considered in all DHCP deployments. While not strictly related to
the mechanisms described in this document, they may be useful in
certain deployment scenarios. [RFC7819] and [RFC7824] provide an
analysis of privacy problems in DHCPv4 and DHCPv6, respectively. If
those are of concern, [RFC7844] offers mitigation steps.

Current DHCPv4 and DHCPv6 standards lack strong cryptographic
protection. There is an ongoing effort in the DHC working group to
address this. [SECURE-DHCPv6] attempts to provide a mechanism for
strong authentication and encryption between DHCPv6 clients and
servers. [SECURITY-MESSAGES] attempts to improve security of
exchanges between DHCP relay agents and servers.

Another possible attack vector is to set up a rogue DHCP server and
provide clients with false information, either as a denial of service
or to execute a man-in-the-middle type of attack. This can be
mitigated by deploying DHCPv6-Shield [RFC7610].

Finally, there is an ongoing effort to update the DHCPv6
specification, which is currently 13 years old. Sections 21
("Security Considerations") and 22 ("Privacy Considerations") of
[DHCPv6bis] contain more recent analysis of the security and privacy
considerations.

10. References

10.1. Normative References

   [RFC2131] Droms, R., "Dynamic Host Configuration Protocol",
              RFC 2131, DOI 10.17487/RFC2131, March 1997,
              <http://www.rfc-editor.org/info/rfc2131>.

   [RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins,
              C., and M. Carney, "Dynamic Host Configuration Protocol
              for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July
              2003, <http://www.rfc-editor.org/info/rfc3315>.

10.2. Informative References

   [DHCPv6bis]
              Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A.,
              Richardson, M., Jiang, S., Lemon, T., and T. Winters,
              "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
              bis", Work in Progress, draft-ietf-dhc-rfc3315bis-05, June
              2016.

   [RFC1034] Mockapetris, P., "Domain names - concepts and facilities",
              STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987,
              <http://www.rfc-editor.org/info/rfc1034>.

   [RFC3011] Waters, G., "The IPv4 Subnet Selection Option for DHCP",
              RFC 3011, DOI 10.17487/RFC3011, November 2000,
              <http://www.rfc-editor.org/info/rfc3011>.

   [RFC3046] Patrick, M., "DHCP Relay Agent Information Option",
              RFC 3046, DOI 10.17487/RFC3046, January 2001,
              <http://www.rfc-editor.org/info/rfc3046>.

   [RFC3297] Klyne, G., Iwazaki, R., and D. Crocker, "Content
              Negotiation for Messaging Services based on Email",
              RFC 3297, DOI 10.17487/RFC3297, July 2002,
              <http://www.rfc-editor.org/info/rfc3297>.

   [RFC3527] Kinnear, K., Stapp, M., Johnson, R., and J. Kumarasamy,
              "Link Selection sub-option for the Relay Agent Information
              Option for DHCPv4", RFC 3527, DOI 10.17487/RFC3527, April
              2003, <http://www.rfc-editor.org/info/rfc3527>.

   [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast
              Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005,
              <http://www.rfc-editor.org/info/rfc4193>.

   [RFC6221] Miles, D., Ed., Ooghe, S., Dec, W., Krishnan, S., and A.
              Kavanagh, "Lightweight DHCPv6 Relay Agent", RFC 6221,
              DOI 10.17487/RFC6221, May 2011,
              <http://www.rfc-editor.org/info/rfc6221>.

   [RFC6607] Kinnear, K., Johnson, R., and M. Stapp, "Virtual Subnet
              Selection Options for DHCPv4 and DHCPv6", RFC 6607,
              DOI 10.17487/RFC6607, April 2012,
              <http://www.rfc-editor.org/info/rfc6607>.

   [RFC6977] Boucadair, M. and X. Pougnard, "Triggering DHCPv6
              Reconfiguration from Relay Agents", RFC 6977,
              DOI 10.17487/RFC6977, July 2013,
              <http://www.rfc-editor.org/info/rfc6977>.

   [RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data
              Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March
              2014, <http://www.rfc-editor.org/info/rfc7159>.

   [RFC7227] Hankins, D., Mrugalski, T., Siodelski, M., Jiang, S., and
              S. Krishnan, "Guidelines for Creating New DHCPv6 Options",
              BCP 187, RFC 7227, DOI 10.17487/RFC7227, May 2014,
              <http://www.rfc-editor.org/info/rfc7227>.

   [RFC7610] Gont, F., Liu, W., and G. Van de Velde, "DHCPv6-Shield:
              Protecting against Rogue DHCPv6 Servers", BCP 199,
              RFC 7610, DOI 10.17487/RFC7610, August 2015,
              <http://www.rfc-editor.org/info/rfc7610>.

   [RFC7819] Jiang, S., Krishnan, S., and T. Mrugalski, "Privacy
              Considerations for DHCP", RFC 7819, DOI 10.17487/RFC7819,
              April 2016, <http://www.rfc-editor.org/info/rfc7819>.

   [RFC7824] Krishnan, S., Mrugalski, T., and S. Jiang, "Privacy
              Considerations for DHCPv6", RFC 7824,
              DOI 10.17487/RFC7824, May 2016,
              <http://www.rfc-editor.org/info/rfc7824>.

   [RFC7844] Huitema, C., Mrugalski, T., and S. Krishnan, "Anonymity
              Profiles for DHCP Clients", RFC 7844,
              DOI 10.17487/RFC7844, May 2016,
              <http://www.rfc-editor.org/info/rfc7844>.

   [SECURE-DHCPv6]
              Jiang, S., Li, L., Cui, Y., Jinmei, T., Lemon, T., and D.
              Zhang, "Secure DHCPv6", Work in Progress,
              draft-ietf-dhc-sedhcpv6-14, October 2016.

   [SECURITY-MESSAGES]
              Volz, B. and Y. Pal, "Security of Messages Exchanged
              Between Servers and Relay Agents", Work in Progress,
              draft-volz-dhc-relay-server-security-02, September 2016.

Authors' Addresses

   Ted Lemon
   Nominum, Inc.
   800 Bridge Parkway, Suite 100
   Redwood City, CA 94065
   United States of America

   Phone: +1-650-381-6000
   Email: Ted.Lemon@nominum.com

   Tomek Mrugalski
   Internet Systems Consortium, Inc.
   950 Charter Street
   Redwood City, CA 94063
   United States of America

   Phone: +1-650-423-1345
   Email: tomasz.mrugalski@gmail.com