Network Working Group T. Lemon
Internet-Draft Nominum, Inc.
Intended status: Best Current Practice April 13, 2013
Expires: October 15, 2013

Customizing DHCP Configuration on the Basis of Network Topology
draft-lemon-dhc-topo-conf-01

Abstract

DHCP servers have evolved over the years to provide significant functionality beyond that which is 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 makes recommendations as to how they can be used.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 2
2. Locality 2
3. Simple Subnetted Network 5
4. Regional Configuration Example 6
5. Dynamic Lookup 7
6. Acknowledgments 8
7. Security Considerations 8
8. IANA Considerations 8
9. References 8
9.1. Normative References 8
9.2. Informative References 9
Author's Address 9

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 both of operators and of protocol designers who may wish to use DHCP as a means for configuring their own servies, but may not be aware of the capabilities provided by modern DHCP servers.

2. Locality

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

```
           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

This diagram allows 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's important to understand the background of how DHCP works when considering this diagram. DHCP clients are assumed not to 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 configuration, it cannot route packets to an off-link DHCP server, and so some kind of relay mechanism is required.

The details of how this 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. In some cases in DHCPv4, when a DHCP client has a routable IPv4 adddress.

In either case, the DHCP server is able to obtain an IP address that it knows is on-link for the link to which the DHCP client is connected: either the DHCPv4 client's routable IPv4 address, or the relay agent's IP address on the link to which the client is connected.

DHCPv6 also has support for more finely grained link identification, using Lightweight DHCPv6 Relay Agents [RFC6221] (LDRA). In this case, in addition to receiving an IPv6 address that is on-link for the link to which the client is connected, the DHCPv6 server also receives an Interface Identifier option from the relay agent that can be used to more precisely identify the client's location on the network.

What this means in practice is that the DHCP server in all cases has sufficient information to pinpoint, at the very least, the layer 3 link to which the client is connected, and in some cases which layer 2 link the client is connected to, when the layer 3 link is aggregated out of multiple layer 2 links.

In all cases, then, the DHCP server will have a link-identifying IP address, and in some cases it may also have a link-specific identifier. It should be noted that there is no guarantee that the link-specific identifier will be unique outside the scope of the link-identifying IP address.

It is also possible for link-specific identifiers to be nested, so that the actual identifier that identifies the link is an aggregate of two or more link-specific 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 DHCP servers.

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

So let's examine the implications of this in terms of how a DHCP server can deliver targeted supplemental configuration information to DHCP clients.

3. Simple Subnetted Network

Consider Figure 1 in the context of a simple subnetted network. In this network, there are four leaf subnets: links A, B, F and G, on which DHCP clients will be configured. 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 a 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 for configuration. Although we know of no DHCP server that uses this specific syntax, every commercial DHCP server provides similar functionality.

```
{"prefixes":
  {"10.0.0.0/24": {"options": {"routers": ["10.0.0.1"]}
           "on-link": ["a"]}}
   "10.0.1.0/24": {"options": {"routers": ["10.0.1.1"}}
           "on-link": ["b"]}
   "10.0.2.0/24": {"options": {"routers": ["10.0.2.1"}}
           "on-link": ["f"]}
   "10.0.3.0/24": {"options": {"routers": ["10.0.3.1"}}
           "on-link": ["g"]}}
```

Figure 2

In figure 2, 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 10.0.0.0/24 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: 10.0.0.1. A client on link F would have a link-identifying address in the 10.0.2.0/24 prefix, and would receive a routers option containing the IP address 10.0.2.1.

4. Regional Configuration Example

In this 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-\u002Dfor example, the relays might be 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 this example, 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:DB8:0:0::/40":   {"on-link": ["A"]}}
   "2001:DB8:100:0::/40": {"on-link": ["B"]}
   "2001:DB8:200:0::/40": {"on-link": ["F"]}
   "2001:DB8:300:0::/40": {"on-link": ["G"]}}

{"links":
  {"A": {"region": "omashu"},
   "B": {"region": "omashu"},
```

```
      "F": {"region": "gaoling"},
      "G": {"region": "gaoling"}}}

   {"regions":
     {"omashu": {"options": {"sip-servers": ["sip.omashu.example.org"],
                "dns-servers": ["dns1.omashu.example.org",
                        "dns2.omashu.example.org"]}},
      "gaoling": {"options": {"sip-servers": ["sip.gaoling.example.org"],
                 "dns-servers": ["dns1.gaoling.example.org",
                         "dns2.gaoling.example.org"]}}}}
```

Figure 3

In this example, when a request comes in to the DHCP server with a link-identifying IP address in the 2001:DB8:0:0::/40 prefix, it is identified as being on link A. The DHCP 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 DHCP server then looks up omashu in the set of regions, and discovers a list of region-specific options.

The DHCP 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-server option containing the IP addresses returned by the resolver for dns1.omashu.example.org and dns2.omashu.example.org.

Similarly, if the DHCP server receives a request from a DHCP client where the link-identifying IP address is contained by the prefix 2001:DB8:300:0::/40, then the DHCP server identifies the client as being connected to link G. The DHCP server then identifies link G as being in the gaoling region, and returns the sip-servers and dns-servers options specific to that region.

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

5. Dynamic Lookup

In the Regional example, the configuration listed several domain names as values for the sip-servers and dns-servers options. The wire format of both of these options contains one or more IPv6 addresses-\u002Dthere is no way to return a domain name to the client.

This was understood to be an issue when the original DHCP protocol was defined, and historical implementations even from the very early days would accept domain names and resolve them. Some early DHCP implementations, particularly those based on earlier BOOTP implementations, had very limited capacity for reconfiguration.

However, all modern commercial DHCP servers handle name resolution by querying the resolver each time a DHCP packet comes in. This means that if DHCP servers and DNS servers are managed by different administrative entities, there is no need for the administrators of the DHCP servers and DNS servers to communicate when changes are made. When changes are made to the DNS server, these changes are immediately and automatically adopted by the DHCP server. Similarly, when DHCP server configurations change, DNS server administrators need not be aware of this.

6. Acknowledgments

Thanks to Dave Thaler for suggesting that even though "everybody knows" how DHCP servers are deployed in the real world, it might be worthwhile to have an IETF document that explains what everybody knows, because in reality not everybody is an expert in how DHCP servers are administered.

7. Security Considerations

This document explaine 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. This document introduces no new functionality, and hence no new security considerations.

8. IANA Considerations

The IANA is hereby absolved of any requirement to take any action in relation to this document.

9. References

9.1. Normative References

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

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

9.2. Informative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC6221] Miles, D., Ooghe, S., Dec, W., Krishnan, S., and A. Kavanagh, "Lightweight DHCPv6 Relay Agent", RFC 6221, May 2011.

Author's Address

Ted Lemon
Nominum, Inc.
2000 Seaport Blvd
Redwood City, CA 94063
USA

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