Provisioning IPv4 Configuration Over IPv6 Only Networks
draft-rajtar-dhc-v4configuration-02

Abstract

   As IPv6 becomes more widely adopted, some service providers are
   taking the approach of deploying IPv6 only networks, without dual-
   stack functionality for IPv4. However, access to IPv4 based services
   is still an ongoing requirement and approaches such as IPv4-in-IPv6
   softwire tunnels are being developed to meet this need.

   In order to provision end-user's hosts with the necessary IPv4
   configuration, a number of different mechanisms have been proposed.
   This memo discusses the benefits and drawbacks of each, with the aim
   of recommending a single approach as the basis for future work.

Requirements Language

   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 RFC 2119 [RFC2119].

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

A service provider with an IPv6-only network must also be able to
provide customers with access to the Internet and other services over
IPv4. Softwire based IPv4-in-IPv6 tunneling mechanisms are an
obvious example of this, such as the ones described in:

o [I-D.ietf-softwire-lw4over6]

o [I-D.ietf-softwire-map]

o [I-D.ietf-softwire-unified-cpe]

A general trend here is to relocate NAT44 functionality and IPv4
address sharing from the centralized tunnel concentrator to the CPE
in order to achieve better scalability. This results in the need to
provision a number of configuration parameters to the CPE, such as
the external public IPv4 address and a restricted port-range to use
for NAT.

In order to configure customer's devices for softwire functionality,
a dynamic provisioning mechanism is necessary. In IPv4 only
networks, DHCPv4 has often been used to provide configuration, but in
an IPv6 only network, DHCPv4 messages cannot be transported natively.

Although softwire mechanisms are currently the only use-case for dhcp
based configuration of IPv4 parameters in IPv6 only networks, a
suitable approach must not be limited to only supporting softwire
configuration.

This document compares four different approaches which have been
proposed for resolving this problem.

1.1. Overview of IPv4 Parameter Configuration Approaches

In order to resolve the problem described above, the following
approaches for transporting IPv4 configuration parameters have been
suggested:

1. Adapt DHCPv4 format messages to be transported over IPv6 as
    described in [I-D.ietf-dhc-dhcpv4-over-ipv6]. For brevity, this
    is referred to as DHCPv4o6.

2. Extend DHCPv6 with new options for IPv4 configuration, such as
    [I-D.ietf-softwire-map-dhcp] describes.

3. Use DHCPv6 as above for external IPv4 address and source port
    configuration. Use DHCPv4 over IPv4 messages within an IPv6
    softwire for configuring additional parameters. This is referred
    to as DHCPv4oSW.

4. Use DHCPv4 format messages, transporting them within a new DHCPv6
    message type as described in [I-D.scskf-dhc-dhcpv4-over-dhcpv6].
    This is referred to as DHCPv4oDHCPv6.

At the time of writing, working examples of the first two approaches
have been developed and successfully tested in several different
operators networks. The third and fourth methods are still
theoretical.

The following sections provide more detail for each approach.

1.2. DHCPv4o6 Based Provisioning - Functional Overview

In order to receive IPv4 configuration parameters, IPv4-only clients
initiate and exchange DHCPv4 messages with the DHCPv4 server. In
order adapt this to an IPv6-only network, an existing DHCPv4 client
implements a 'Client Relay' (CRA) function, which takes DHCPv4
messages and puts them into UDPv6 and IPv6.

As the mechanism involves unicast based communications, the IPv6
address of the server must be provisioned to the client. This option
is described in [I-D.mrugalski-softwire-dhcpv4-over-v6-option].

The DHCPv4o6 server must either provide an IPv6 interface to the
client, or an intermediary 'Transport Relay Agent' device can act as
the gateway between the IPv4 and IPv6 domains.

For the dynamic allocation of IPv4 addresses, the DHCPv4o6 server
needs to be extended to support the new functionality, such as
storing the IPv6 address of DHCPv4o6 clients. The CRA6ADDR option
must also be implemented.

This approach currently uses functional elements for ingress and
egress of the IPv6-only transport domain--the CRA on the host and the
TRA or TSV on the server. As a result, this approach has sometimes
been referred to as a tunneling approach. However, relay agent
encapsulation is not a tunnel, since it carries only DHCP traffic; it
would be more accurate to describe it as an encapsulation.

It is worth noting that there is no technical reason for using relay
encapsulation for DHCPv4o6; this approach was taken because the
authors of the draft originally imagined that it might be used to
provide configuration information for an unmodified DHCPv4 client.
However, this turns out not to be a viable approach: in order for
this to work, there would have to be IPv4 routing on the local link
to which the client is connected. In that case, there's no need for
DHCPv4o6.

Given that this is the case, there is no technical reason why
DHCPv4o6 can't simply use the IPv6 transport directly, without any
relay encapsulation. This would greatly simplify the specification
and the implementation, and would still address the requirements
stated in this document.

[I-D.ietf-dhc-dhcpv4-over-ipv6] decribes this solution in detail.

The protocol stack is as follows:

DHCPv4/UDPv6/IPv6

1.3. DHCPv6 Based Provisioning - Functional Overview

In this approach, DHCPv6 would be extended with new DHCPv6 options
for configuring all IPv4 based services and functions. Any DHCPv4
options needed by IPv4 clients connected to the IPV6 network are
updated as new DHCPv6 native options carrying IPv4 configuration
parameters.

At the time of writing, it is not known how many such options would
need to be ported from DHCPv4 to DHCPv6.

An example of this approach is described in
[I-D.ietf-softwire-map-dhcp], where a DHCPv6 message is used to
convey the parameters necessary for IPv4 in IPv6 softwire
configuration.

The protocol stack is as follows:

DHCPv6/UDPv6/IPv6

1.4. DHCPv4oSW Based Provisioning - Functional Overview

In this approach, the configuration of IPv4 address and source ports
(if required) is carried out using DHCPv6 as described in section 1.3
above. Any additional IPv4 configuration parameters which are
required are then provisioned using a DHCPv4 messages transported
within IPv6 in the configured softwire in the same manner as any
other IPv4 based traffic.

On receipt at the tunnel concentrator (e.g. MAP Border Router or a
Lightweight 4over6 lwAFTR), the DHCPv4 message removed from the
softwire and forwarded to the DHCPv4 server in the same way as any
other IPv4 packet is handled.

As the client is already configured with its external IPv4 address
and source ports (using DHCPv6), the messages exchanged between the

DHCPv4 client and server would be strictly DHCPINFORM/DHCPACK
messages, for the configuration of additional IPv4 parameters.
Broadcast based DHCPDISCOVER messages can not be transported as they
are not compatible with the softwire architecture.

For this approach to function, a mechanism for the DHCPv4 client to
learn the IPv4 address of the DHCPv4 server is needed. This could be
done by defining a well-known IPv4 address for the DHCPv4 server,
implementing a DHCPv4 relay function within the tunnel concentrator
or other configuration methods.

From a transport perspective, the key difference between this method
and DHCPv4o6 (described above) is that here, the DHCPv4 message is
put into UDPv4 and IPv4 and then put into the IPv6 softwire, instead
of directly placing the DHCPv4 message into UDPv6 and IPv6.

Currently, this approach is only theoretical and does not have a
corresponding Internet Draft providing more detail.

The protocol stack that would be used for obtaining additional IPv4
configuraion is as follows:

DHCPv4/UDPv4/IPv4/IPv6

1.5. DHCPv4oDHCPv6 Based Provisioning - Functional Overview

[I-D.scskf-dhc-dhcpv4-over-dhcpv6] describes the transport of DHCPv4
messages within two new DHCPv6 messages types: BOOTREQUESTV6 and
BOOTREPLYV6. These messages types must be implemented in both the
DHCPv4oDHCPv6 client and server.

In this approach, the configuration of stateless IPv4 addresses and
source ports (if required) is carried out using DHCPv6 as described
in section 1.3 above. Dynamic IPv4 addressing, and/or any additional
IPv4 configuration, is provided using DHCPv4 messages carried
(without IPv4/UDPv4 headers) within a new OPTION_BOOTP_MSG DHCPv6
option.

OPTION_BOOTP_MSG enables the client and server to send BOOTP/DHCPv4
messages verbatim across the IPv6 network. When a DHCPv4oDHCPv6
server receives a DHCPv6 request containing OPTION_BOOT_MSG within a
BOOTREQUESTV6 message, it passes it to the DHCPv4 server engine.
Likewise, the DHCPv4 server place its DHCPv4 response in the payload
of OPTION_BOOTP_MSG and puts this into a BOOTPRPLYV6 message.

As the DHCPv4 messages are carried within DHCPv6 multicast messages,
using the All_DHCP_Relay_Agents_and_Servers, they can be relayed in
exactly the same way as any other DHCPv6 multicasted message.

Optionally, DHCPv6 relays could be updated so that they forward the
BOOTREQUESTV6 message to a different destination address, allowing
for the separation of DHCPv4 and DHCPv4 provisioning infrastructure.

The protocol stack used for obtaining dynamic v4 addressing or
additional IPv4 configuraion is as follows:

DHCPv4/DHCPv6/UDPv6/IPv6

2. Requirements for the Solution Evaluation

The following requirements have been defined for the evalution of the
different approaches:

1. Minimize the amount of work necessary to implement the solution
    through re-use of existing standards and implementations as much
    as possible.

2. Provide a method of supporting all existing DHCPv4 options so
    that they can be utilised without the need for further
    standardation.

3. Allow for the dynamic leasing of IPv4 addresses to clients. This
    allows for more efficient use of limited IPv4 resources.

4. Enable the separation of IPv4 and IPv6 host configuration.

5. Avoid leaving legacy IPv4 options in DHCPv6.

6. Provide a flexible architecture to give operators the option of
    only deploying the functional elements necessary for their
    specific requirements.

3. Comparison of the Four Approaches

   The table below shows a comparison of the different approaches
   against the solution requirements described above.

```
+----------+----------+--------+----------+--------------+
| Req. No. | DHCPv4o6 | DHCPv6 | DHCPv4oSW | DHCPv4oDHCPv6 |
+----------+----------+--------+----------+--------------+
|    1     |    No    |  Yes   |   Yes    |     Yes      |
|    2     |   Yes    |   No   |   Yes    |     Yes      |
|    3     |   Yes    |   No   |    No    |     Yes      |
|    4     |   Yes    |   No   |   Yes    |     Yes      |
|    5     |   Yes    |   No   |   Yes    |     Yes      |
|    6     |   Yes    |   No   |   Yes    |     Yes      |
+----------+----------+--------+----------+--------------+
```

                     Table 1: Approach Comparison

3.1. Pros and Cons of the Different Approaches

   The following sections of the document provide more details of the
   pros and cons relevant to each of the approaches.

3.1.1. DHCPv4o6 Based Provisioning

3.1.1.1. Pros

   1. Once implemented, all existing DHCPv4 options will be available
       with no further ongoing development work necessary.

   2. IPv4 and IPv6 based provisioning can be separated from each other
       if required, allowing flexibility in network design.

   3. Easy to implement through minor adaptation of existing DHCPv4
       client/server code.

   4. Simple, in that no additional functional elements are necessary
       except the DHCPv4o6 client and server. The Transport Relay Agent
       is completely optional.

   5. Suitable for the provisioning of dynamic IPv4 configuration as
       the existing DHCPv4 leasing mechanism can be used.

   6. Implementations already exist, proving that the approach works.

3.1.1.2. Cons

1. More complex, in that there are more new functional elements
    (CRA, DHCPv4o6 server and optionally TRA) within the architecture
    than are necessary in DHCPv6 based provisioning.

2. A new DHCPv6 option is necessary in order to provision the IPv6
    address of the DHCPv4 server to the end device.

3. For a Host CRA (HCRA), DHCPv4 client host needs to be updated to
    implement the IPv6 encapsulation and decapsulation function.
    Otherwise a physically separate On-Link CRA (LCRA) functional
    element must be deployed.

4. A DHCPv4 server must be deployed and maintained.

5. The DHCPv4 server needs to be updated to implement new DHCPv4o6
    functionality.

3.1.2. DHCPv6 Based Provisioning

3.1.2.1. Pros

1. Simpler, in that no additional functional elements are required
    except the DHCPv6 client and server.

2. A single protocol is used to deliver configuration information
    for IPv4 and IPv6.

3. A single provisioning point for all configuration parameters.

4. Implementations already exist, proving that the approach works.

3.1.2.2. Cons

1. Any required DHCPv4 options must be ported to DHCPv6, which will
    require re-development work for each option. All functional
    elements in the DHCPv6 implementation (clients, servers, relays)
    would need to be updated for each change.

2. Means that DHCPv4 'legacy' options, which will be of decreasing
    relevance in the future will remain in DHCPv6 for the lifetime of
    the protocol.

3. Each time that a DHCPv4 option is ported to DHCPv6, all clients
    and servers would need to be updated to implement the new option.

4. Does not provide an architecture for keeping IPv4 and IPv6
    domains separated.

   5. Does not provide a mechanism for dynamic IPv4 address leasing. A
       DHCPv4 lease lifetime mechanism would need to be added to DHCPv6
       for this.

3.2. DHCPv4oSW Based Provisioning

3.2.1. Pros

   1. Once implemented, all existing DHCPv4 options will be be
       available with no further ongoing development work necessary.

   2. Uses the existing DHCPv4 and DHCPv6 architectures in order to
       provide IPv4 configuration in an IPv6 only environment.

   3. DHCPv4 and DHCPv6 based provisioning can be separated from each
       other if required, allowing flexibility in network design.

3.2.2. Cons

   1. More complex, in that there are more new functional elements
       within the architecture than are necessary in DHCPv6 based
       provisioning.

   2. IPv4 over IPv6 softwire approaches which distribute NAT to the
       CPE and allow for IP address sharing (MAP-E & LW4o6) forbid the
       use of reserved TCP/UDP ports (e.g. 0-1024). Every DHCPv4
       client sharing the same address needs to have a UDP listener
       running on UDP port 68. To resolve this would require
       significant rework to either the softwire mechanisms and/or the
       DHCPv4 client implementation.

   3. From the current specification, DHCPINFORM is not suitable for
       use over a softwire. Additional work, such as the development of
       'shims' would be necessary

   4. The current DHCPINFORM specification has a number of unclear
       points, such as those described in
       [I-D.ietf-dhc-dhcpinform-clarify]. Substantial work would be
       required to resolve this.

   5. Links the deployment of IPv4 configuration over IPv6 to a
       softwire implementation (e.g. requiring a softwire concentrator
       to act as a DHCPv4 relay). Whilst softwires are the only
       application for this functionality at the moment, this may not
       always be the case.

6. A new mechanism must be defined in order to provide the DHCPv4
    client with the IPv4 address of the DHCPv4 server so that unicast
    DHCPINFORM messages can be sent.

7. As only DHCPINFORM/DHCPACK DHCPv4 message types are supported,
    dynamic IPv4 address leasing (using DHCPDISCOVER messages) can
    not be used.

8. The approach is unproven as no existing implementations exist.

3.3. **DHCPv4oDHCPv6 Based Provisioning**

3.3.1. **Pros**

1. Once implemented, all existing DHCPv4 options will be be
    available with no further ongoing development work necessary.

2. Uses the existing DHCPv4 and DHCPv6 architectures in order to
    provide IPv4 configuration in an IPv6 only environment.

3. DHCPv4 and DHCPv6 based provisioning can be separated from each
    other if required, allowing flexibility in network design.

4. Suitable for the provisioning of dynamic IPv4 configuration as
    the existing DHCPv4 leasing mechanism can be used.

3.3.2. **Cons**

1. More complex, in that there are more new functional elements
    within the architecture than are necessary in DHCPv6 based
    provisioning.

2. DHCPv6 clients needs to be updated to implement the new DHCPv6
    message types.

3. The DHCPv6 server needs to be updated to implement new
    DHCPv4oDHCPv6 message types and functionality.

4. If separation of DHCPv4 and DHCPv4 provisioning infrastructure is
    required, DHCPv6 relay agents need to be updated to implement
    dedicated forwarding destinations based on message type.

5. The approach is currently unproven as no existing implementations
    exist.

4. **Conclusion**

Discussion: This chapter will be updated to reflect the consensus of the DHC Working Group.

5. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

6. Security Considerations

7. Acknowledgements

Thanks to Ted Lemon and Tomek Mrugalski for their input and reviews.

8. References

8.1. Normative References

[RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
            Requirement Levels", BCP 14, RFC 2119, March 1997.

8.2. Informative References

[I-D.ietf-dhc-dhcpinform-clarify]
            Hankins, D., "Dynamic Host Configuration Protocol
            DHCPINFORM Message Clarifications", draft-ietf-dhc-
            dhcpinform-clarify-06 (work in progress), October 2011.

[I-D.ietf-dhc-dhcpv4-over-ipv6]
            Cui, Y., Wu, P., Wu, J., and T. Lemon, "DHCPv4 over IPv6
            Transport", draft-ietf-dhc-dhcpv4-over-ipv6-06 (work in
            progress), March 2013.

[I-D.ietf-softwire-lw4over6]
            Cui, Y., Sun, Q., Boucadair, M., Tsou, T., Lee, Y., and I.
            Farrer, "Lightweight 4over6: An Extension to the DS-Lite
            Architecture", draft-ietf-softwire-lw4over6-00 (work in
            progress), April 2013.

[I-D.ietf-softwire-map-dhcp]
            Mrugalski, T., Troan, O., Dec, W., Bao, C.,
            leaf.yeh.sdo@gmail.com, l., and X. Deng, "DHCPv6 Options
            for Mapping of Address and Port", draft-ietf-softwire-map-
            dhcp-03 (work in progress), February 2013.

[I-D.ietf-softwire-map]

            Troan, O., Dec, W., Li, X., Bao, C., Matsushima, S., and
            T. Murakami, "Mapping of Address and Port with
            Encapsulation (MAP)", draft-ietf-softwire-map-05 (work in
            progress), March 2013.

   [I-D.ietf-softwire-unified-cpe]
            Boucadair, M. and I. Farrer, "Unified IPv4-in-IPv6
            Softwire CPE", draft-ietf-softwire-unified-cpe-00 (work in
            progress), March 2013.

   [I-D.mrugalski-softwire-dhcpv4-over-v6-option]
            Mrugalski, T. and P. Wu, "Dynamic Host Configuration
            Protocol for IPv6 (DHCPv6) Option for DHCPv4 over IPv6
            Endpoint", draft-mrugalski-softwire-
            dhcpv4-over-v6-option-01 (work in progress), September
            2012.

   [I-D.scskf-dhc-dhcpv4-over-dhcpv6]
            Sun, Q., Cui, Y., Siodelski, M., Krishnan, S., and I.
            Farrer, "DHCPv4 over DHCPv6 Transport", draft-scskf-dhc-
            dhcpv4-over-dhcpv6-01 (work in progress), April 2013.

Authors' Addresses

   Branimir Rajtar
   Hrvatski Telekom
   Zagreb
   Croatia

   Email: branimir.rajtar@t.ht.hr

   Ian Farrer
   Deutsche Telekom AG
   Bonn
   Germany

   Email: ian.farrer@telekom.de