DHC WG
Internet-Draft
Intended status: Informational
Expires: January 2, 2015

B. Rajtar
Hrvatski Telekom
I. Farrer
Deutsche Telekom AG

July 01, 2014

Provisioning IPv4 Configuration Over IPv6 Only Networks
draft-ietf-dhc-v4configuration-06

Abstract

As IPv6 becomes more widely adopted, some service providers are choosing to deploy IPv6 only networks without dual-stack functionality for IPv4. However, as access to IPv4 based services will continue to be a requirement for the foreseeable future, IPv4 over IPv6 mechanisms, such as softwire tunnels are being developed.

In order to provision end-user's hosts with the IPv4 configuration necessary for such mechanisms, a number of different approaches have been proposed. This memo discusses each of the proposals, identifies the benefits and drawbacks and recommends approaches to be used as the basis for future deployment and development.

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 IPv4 Internet and other IPv4-only services. IPv4 over IPv6 tunneling / translation mechanisms are an obvious example of this, such as the ones described in:

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

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

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

In today's home networks, each residential user is allocated a single global IPv4 address which is used for NAT44. Decentralizing NAT44 allows for much better scaling and, when combined with stateless network functions, can simplify redundancy and logging when compared to centralized Carrier Grade NAT architectures. 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. Other parameters may also be necessary, depending on the underlying transport technology that is in use. In IPv4 only networks, DHCPv4 has often been used to provide IPv4 configuration, but in an IPv6 only network, DHCPv4 messages cannot be transported natively without either IPv6 encapsulation or translation.

DHCPv4 messages can be transported, unmodified, over a broadcast capable link-layer, depending on the underlying IPv4 in IPv6 technology, network topology and DHCPv4 client capabilities. A functional description of how unmodified DHCPv4 can be used is provided in Section 5. This approach is recommended for service providers whose network and clients can support this DHCPv4 architecture.

For the most simple IPv4 provisioning case, where the client only needs to receive a static IPv4 address assignment (with no dynamic address leasing or additional IPv4 configuration), a DHCPv6 based approach (e.g. [I-D.ietf-softwire-map-dhcp]) may provide a suitable solution.

This document is concerned with more complex IPv4 configuration scenarios, to bring IPv4 configuration over IPv6-only networks in line with the functionality offered by DHCPv4 in IPv4 native networks. DHCPv4 options may also need to be conveyed to clients for configuring IPv4 based services, e.g., SIP server addresses.

Although IPv4-in-IPv6 softwire tunnel and translation clients are currently the only use-case for DHCP based configuration of IPv4 parameters in IPv6 only networks, a suitable IPv4 provisioning solution should not be limited to only supporting the configuration of softwires, or be bound to specific IPv4 over IPv6 architectures or mechanisms. The solution needs to be flexible enough to support new IPv4 over IPv6 technologies as they are developed.

This document describes and compares four different methods which have been proposed as solutions to this problem.

1.1. Overview of IPv4 Parameter Configuration Approaches

The following approaches for transporting IPv4 configuration parameters over IPv6 only networks 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 to support IPv4 address leasing and other DHCPv4 options.

3. Use DHCPv6 for external IPv4 address and source port configuration (e.g. [I-D.ietf-softwire-map-dhcp]. Use DHCPv4 over IPv4 messages within an IPv6 softwire for configuring additional parameters. This is referred to as DHCPv6 + Stateless DHCPv4oSW.

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

At the time of writing, working examples of all but the third method have been developed and successfully tested in several different operators networks.

The following sections describe each of the approaches in more detail.

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. To adapt this for an IPv6-only network, an existing DHCPv4 client implements a Host Client Relay Agent (HCRA) function, which takes DHCPv4 messages and puts them into UDP and IPv6.

As the mechanism involves unicast IPv6 based communications, the IPv6 address of the server must be provisioned to the client. A DHCPv6 option for provisioning clients with this address is described in [I-D.mrugalski-softwire-dhcpv4-over-v6-option].

The IPv6 Transport Server (TSV) provides an IPv6 interface to the client. This interface may be implemented directly on the server and/or via an intermediary 'Transport Relay Agent' (TRA) device which acts as the gateway between the IPv4 and IPv6 domains.

For the dynamic allocation of IPv4 addresses, the DHCPv4 server function needs to be extended to add DHCPv4o6 TSV capabilities, such as the storing the IPv6 address of DHCPv4o6 clients and implementing the CRA6ADDR option.

This approach currently uses functional elements for ingress and egress of the IPv6-only transport domain - the HCRA on the host and the TRA or TSV on the server. As a result, this 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 based transport.

[I-D.ietf-dhc-dhcpv4-over-ipv6] also defines an On-Link Client Relay Agent (LCRA), which is a Client Relay Agent located on the same link as an unmodified DHCPv4 client. 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] describes this solution in detail.

The protocol stack for provisioning IPv4/IPv6 tunneling and translation mechanisms is as follows:

DHCPv4/UDP/IPv6

1.3. DHCPv6 Based Provisioning - Functional Overview

In this approach, DHCPv6 [RFC3315] would be extended with new DHCPv6 options for configuring all IPv4 based services and functions (i.e. IPv4 address assignment and any necessary DHCPv4 options). DHCPv4 options needed by IPv4 clients connected to the IPv6 network are updated as new DHCPv6 native options carrying IPv4 configuration parameters. IPv4 address leasing would also need to be managed by the DHCPv6 server.

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

The protocol stack for provisioning IPv4/IPv6 tunneling and translation mechanisms is as follows:

DHCPv6/UDP/IPv6

1.4. DHCPv6 + Stateless DHCPv4oSW Based Provisioning - Functional Overview

In this approach, configuration of the IPv4 address and source ports (if required) is carried out using DHCPv6, e.g. using [I-D.ietf-softwire-map-dhcp]. Any additional IPv4 configuration parameters that are required are then provisioned using DHCPv4 messages transported, within IPv6, through the configured softwire in the same manner as any other IPv4 based traffic. Broadcast based DHCPv4 DHCPDISCOVER messages (necessary for IPv4 address assignment) can not be transported as some softwire mechanisms implement NBMA links, where broadcast isn't supported. Additionally, there is a more general issue with the use of fixed L4 ports in A+P [RFC6346] based approaches. Here, a single IPv4 address is shared among multiple users, each using a unique set of ports for differentiation meaning that it is not possible for every client to be allocated a fixed L4 within its unique port set.

On receipt by the tunnel concentrator (e.g. MAP Border Router or a Lightweight 4over6 lwAFTR), the DHCPv4 message is extracted from the IPv6 packet and forwarded to the DHCPv4 server in the same way as any other IPv4 forwarding plane packet is handled.

As the client is already configured with its external IPv4 address and source ports (using DHCPv6 or a well-known IPv4 address for DS-Lite clients), the messages exchanged between the DHCPv4 client and server would be strictly DHCPINFORM/DHCPACK messages. These can be used for conveying additional DHCPv4 based options.

For this approach to function, a mechanism for the DHCPv4 client to learn the IPv4 address of the DHCPv4 server is also required. This could be via a well-known IPv4 address for the DHCPv4 server, a DHCPv4 relay function within the tunnel concentrator or other methods.

From a transport perspective, the key difference between this method and DHCPv4o6 (described above) is the protocol stack. Here the DHCPv4 message is first put into UDP and IPv4 and then into the IPv6 softwire, instead of placing the DHCPv4 message directly into UDP and IPv6.

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

For IPv4/IPv6 tunneling and translation mechanism, the protocol stack used for obtaining an IPv4 address and source ports (if required) is as follows:

DHCPv6/UDP/IPv6

For provisioning IPv4/IPv6 tunneling mechanisms, the protocol stack for obtaining additional IPv4 configuration is:

DHCPv4/UDP/IPv4

NB: The encapsulating IPv6 tunneling header is not shown as it is functionally a layer 2 header.

And for provisioning IPv4/IPv6 translation mechanisms:

DHCPv4/UDP/IPv6

1.5. DHCPv4oDHCPv6 Based Provisioning - Functional Overview

[I-D.ietf-dhc-dhcpv4-over-dhcpv6] describes transporting DHCPv4 messages within two new DHCPv6 messages types: DHCPV4-QUERY and DHCPV4-RESPONSE. These new messages types must be implemented in both the DHCPv4oDHCPv6 client and server.

In this approach, dynamic IPv4 addressing, and/or any additional IPv4 configuration, is provided using DHCPv4 messages carried (without IPv4/UDP headers) within a new OPTION_DHCPV4_MSG DHCPv6 option.

OPTION_DHCPV4_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_DHCPV4_MSG within a DHCPV4-QUERY message, it passes it to the DHCPv4 server engine.

Likewise, the DHCPv4 server place its DHCPv4 response in the payload of OPTION_DHCPV4_MSG and puts this into a DHCPV4-RESPONSE message.

DHCPv4 messages can be carried within DHCPv6 multicast messages, using the All_DHCP_Relay_Agents_and_Servers multicast address. These can be relayed in exactly the same way as any other DHCPv6 multicasted messages.

Optionally, DHCPv6 relays could be updated so that they forward the DHCPV4-QUERY message to a different destination address, allowing for the separation of DHCPv4 and DHCPv6 provisioning infrastructure.

If the DHCPv4oDHCPv6 client is provisioned with a unicast IPv6 address(es) for the server(s), then an entirely unicast message flow between the client and server is also possible without the need for relaying.

For provisioning IPv4/IPv6 tunneling and translation mechanisms, the protocol stack used for obtaining dynamic v4 addressing and/or additional IPv4 configuration is as follows:

DHCPv4/DHCPv6/UDP/IPv6

2. Requirements for the Solution Evaluation

The following requirements have been defined to evaluate 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 DHCPv4 options so that they can be utilized without the need for further standardization.

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 infrastructure, i.e. independent DHCPv4 and DHCPv6 server functions to restrict provisioning domains to the relevant protocol and allow the removal of IPv4 infrastructure in the future.

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.

7. Not be restricted to specific underlying IPv4 over IPv6 transport mechanisms or architectures. The solution needs to be flexible enough to support new IPv4 over IPv6 technologies as they are developed.

3. Comparison of the Four Approaches

The table below provides a comparative evaluation showing how the different approaches meet the solution requirements described above.

Req. No.	DHCPv4o6	DHCPv6	DHCPv6 + Stateless DHCPv4oSW	DHCPv4oDHCPv6
1	No	Yes	No	Yes
2	Yes	No	Yes	Yes
3	Yes	No	No	Yes
4	Yes	No	Yes	Yes
5	Yes	No	Yes	Yes
6	No	No	Yes	Yes
7	Yes	Yes	No	Yes

Table 1: Approach Comparison

The following sections of the document provide more detail on the pros and cons of each of the approaches.

3.1. DHCPv4o6 Based Provisioning

3.1.1. Pros

1. Implementation makes all existing DHCPv4 options 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, relay and server code.

4. Suitable for dynamic IPv4 address leases where the IPv4 address lifetime is not linked to the lifetime of a DHCPv6 lease.

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

3.1.2. Cons

1. More new functional elements required within the architecture (CRA, DHCPv4o6 server and optionally TRA) 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. The DHCPv4 client host needs to be updated to implement the IPv6 encapsulation and decapsulation function (i.e., an HCRA). Otherwise a 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.

3.2. DHCPv6 Based Provisioning

3.2.1. Pros

1. 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. Single provisioning point for all configuration parameters.

3.2.2. Cons

1. Any required DHCPv4 options must be ported to DHCPv6, which will require re-development work for each option.

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, servers and possibly relays would need to be updated to implement the new option.

4. Architecture does not allow for the separation of IPv4 and IPv6 domains.

5. Does not provide a mechanism for dynamic IPv4 address leasing. The lifetime of the IPv4 address is linked to the lifetime of a DHCPv6 address lease (i.e. the IPv4 address can only be changed when a DHCPv6 RENEW/REBIND message is sent). To remove this interdependency, a new DHCPv4 lease management mechanism would need to be added to DHCPv6 (e.g. a new Identity Association solely for IPv4 address leasing).

3.3. DHCPv6 + Stateless DHCPv4oSW Based Provisioning

3.3.1. Pros

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

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

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

3.3.2. Cons

1. More new functional elements required than are necessary with DHCPv6 based provisioning.

2. IPv4 over IPv6 softwire approaches that distribute the NAT44 function 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 be the case in the future, meaning another solution may be required.

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 the DHCPINFORM/DHCPACK DHCPv4 message types are supported, dynamic IPv4 address leasing (using DHCPDISCOVER messages) cannot be used.

8. Restricted to underlying hub-and-spoke IPv4 over IPv6 architectures. The hub is necessary to locate the DHCPv4 relay function, as all traffic must pass through it. An underlying mesh architecture does not have such a location to deploy the relay function.

9. The approach is currently unproven. Although existing implementations may currently exist, the approach has not been demonstrated.

3.4. DHCPv4oDHCPv6 Based Provisioning

3.4.1. Pros

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

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

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

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

3.4.2. Cons

1. More new functional elements within the architecture than are necessary in DHCPv6 based provisioning.

2. DHCPv6 clients need to be updated to implement the new DHCPv6 message types (BOOTPREQUESTv6 and BOOTPREPLYv6).

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

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

4. Conclusion

Whilst all of the approaches described here will require some development work to realize, it is clear from the above analysis that the most sustainable approach capitalizes on existing DHCPv4 implementations and include them as new DHCPv6 message types. The main rationale for this is that it enables all of DHCPv4's existing options to be migrated for use over IPv6 in a single step.

Porting of all necessary DHCPv4 options to DHCPv6 would require ongoing development work, re-implementing existing DHCPv4 functionality in DHCPv6. This will result in having legacy DHCPv4 options in DHCPv6, which will no longer be useful once IPv4 is completely abandoned.

Therefore, the DHCPv6 approach is not appropriate for delivering IPv4 configuration parameters.

The dynamic leasing of IPv4 addresses is fundamental to the efficient use of remaining IPv4 resources. This will become increasingly important in the future, so a mechanism which supports this is necessary. DHCPv6 + Stateless DHCPv4oSW does not provide this function and so is not recommended.

The DHCPv4o6 approach requires a DHCPv4 server (with DHCPv4o6 functionality) for all deployment scenarios, even when DHCPv4 specific functionality (e.g. sending DHCPv4 options) is not required by the operator.

Therefore, this memo recommends DHCPv4oDHCPv6 [I-D.ietf-dhc-dhcpv4-over-dhcpv6] as the best underlying approach for provisioning IPv4 parameters over an IPv6 only network.

5. Transporting Unmodified DHCPv4 Messages over an IPv6 Link Layer

DHCPv4 can be transported across a broadcast capable link layer, such as a softwire. Functionally, a DHCPv4 client operates on the link layer interface (e.g. the softwire tunnel interface). As the link layer must support broadcasts, DHCPDISCOVER and other broadcast DHCPv4 messages can be transported. The DHCPv4 message flow is then the same as described in section 3.1 of [RFC2131].

For an unmodified DHCPv4 client to function over an IPv6 native network, the underlying IPv4 over IPv6 architecture must be based on a point-to-point link between the client and a central point (i.e. a

hub or tunnel concentrator) which all client DHCPv4 broadcast messages will pass through. This hub must function as either the DHCPv4 server or a DHCPv4 relay. The relay forwards broadcast DHCPv4 DHCPDISCOVER/DHCPREQUEST messages to a separate DHCPv4 server.

5.1. Combined Hub and DHCPv4 Relay Required Functionality

When the DHCPv4 relay function is co-located with the IPv4 in IPv6 hub function, there are some implementation considerations and requirements that must be fulfilled. The following list describes these.

1. Depending on the underlying IPv4 over IPv6 mechanism that the hub is based upon, it may be necessary to modify the encapsulation/ decapsulation or IPv6/IPv4 translation packet validation policy so that IPv4 payload packets sourced from the unspecified address (0.0.0.0) are not dropped for broadcast DHCPv4 payload packets.

2. The DHCPv4 relay must use the DHCPv4 Relay Information Option (option 82) Relay-ID sub-option (2) to convey the client's source IPv6 address. This is used by the relay to route DHCPv4 response packets sent by the DHCPv4 server to the correct client.

3. For some IPv4 in IPv6 transition technologies, a client may be configured with an IPv4 address which is shared by other clients. In these cases, clients using a single IPv4 address are differentiated using the combination of the IPv4 address and a range of restricted layer 4 source ports unique to each client (used for NAPT). The DHCPv4 client L4 port (68) must not be provisioned to any client for NAPT use.

4. The DHCPv4 relay must implement the Server Identifier Override Sub-option described in [RFC5107] to direct all DHCPv4 messages through the DHCPv4 relay. As option 82 is being used to identify the destination IPv6 address for messages from the DHCPv4 server to the client, the L4 destination port is not required for the return path lookup process and is left unchanged as port 68.

6. IANA Considerations

This document does not make any request from IANA.

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

7. Security Considerations

This document analyzes various solutions and doesn't introduce any new capabilities necessitating additional security considerations. The following sub-sections provide pointers to the documented security considerations associated with each approach.

7.1. DHCPv4oIPv6

Security considerations associated with this approach are described in Section 8 of [I-D.ietf-dhc-dhcpv4-over-ipv6].

7.2. DHCPv6

Security considerations associated with this approach are described in Section 23 of [RFC3315].

7.3. DHCPv6+DHCPv4oSW

There is currently no document describing this mechanism, so no security considerations have been documented.

7.4. DHCPv4oDHCPv6

Security considerations associated with this approach are described in [I-D.ietf-dhc-dhcpv4-over-dhcpv6].

8. Acknowledgements

Thanks to Ted Lemon, Tomek Mrugalski, Ole Troan, Bernie Volz and Francis Dupont for their input and reviews.

9. 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-dhcpv6]
Sun, Q., Cui, Y., Siodelski, M., Krishnan, S., and I. Farrer, "DHCPv4 over DHCPv6 Transport", draft-ietf-dhc-dhcpv4-over-dhcpv6-09 (work in progress), June 2014.

[I-D.ietf-dhc-dhcpv4-over-ipv6]
Cui, Y., Wu, P., Wu, J., Lemon, T., and Q. Sun, "DHCPv4 over IPv6 Transport", draft-ietf-dhc-dhcpv4-over-ipv6-09 (work in progress), April 2014.

[I-D.ietf-softwire-lw4over6]
Cui, Y., Qiong, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the DS-Lite Architecture", draft-ietf-softwire-lw4over6-10 (work in progress), June 2014.

[I-D.ietf-softwire-map]
Troan, O., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, "Mapping of Address and Port with Encapsulation (MAP)", draft-ietf-softwire-map-10 (work in progress), January 2014.

[I-D.ietf-softwire-map-dhcp]
Mrugalski, T., Troan, O., Farrer, I., Perreault, S., Dec, W., Bao, C., leaf.yeh.sdo@gmail.com, l., and X. Deng, "DHCPv6 Options for configuration of Softwire Address and Port Mapped Clients", draft-ietf-softwire-map-dhcp-07 (work in progress), March 2014.

[I-D.ietf-softwire-map-t]
Li, X., Bao, C., Dec, W., Troan, O., Matsushima, S., and T. Murakami, "Mapping of Address and Port using Translation (MAP-T)", draft-ietf-softwire-map-t-05 (work in progress), February 2014.

[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.

[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.

[RFC5107] Johnson, R., Kumarasamy, J., Kinnear, K., and M. Stapp, "DHCP Server Identifier Override Suboption", RFC 5107, February 2008.

[RFC6346] Bush, R., "The Address plus Port (A+P) Approach to the IPv4 Address Shortage", RFC 6346, August 2011.

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