



DCHP Relay User Guide

Version 23.8.1.2, 06 November 2023

Registered Address	Support	Sales
26, Kingston Terrace, Princeton, New Jersey 08540, United States		
		+91 80 4850 5445
http://www.rtbrick.com	support@rtbrick.com	sales@rtbrick.com

©Copyright 2023 RtBrick, Inc. All rights reserved. The information contained herein is subject to change without notice. The trademarks, logos and service marks ("Marks") displayed in this documentation are the property of RtBrick in the United States and other countries. Use of the Marks are subject to RtBrick's Term of Use Policy, available at <https://www.rtbrick.com/privacy>. Use of marks belonging to other parties is for informational purposes only.

Table of Contents

1. Introduction to DHCP	3
1.1. Supported Platforms	3
1.2. DHCP Relay Agent	3
1.3. DHCP Relay in Proxy Mode	4
2. DHCP Relay Configuration	5
2.1. DHCP Server Configuration	5
2.2. DHCP Relay Configuration	6
2.3. DHCP Relay on Unnumbered Interface	7
3. DHCP Relay Operations	9
3.1. Verify DHCP Relay Configuration	9
3.2. Verify DHCP Relay Packet Processing	9
3.3. Verify DHCP Relay Bindings	10

1. Introduction to DHCP

The Dynamic Host Configuration Protocol (DHCP) is a standardized client/server network protocol that dynamically assigns IPv4 addresses and other related configuration information to network devices.

DHCP provides an automated way to distribute and update IPv4 addresses and other configuration information on a network. A DHCP server provides this information to a DHCP client through the exchange of a series of messages, known as the DHCP conversation or the DHCP transaction. If the DHCP server and DHCP clients are located on different subnets, a DHCP relay agent is used to facilitate the conversation.

The RBFS DHCP relay agent handles all DHCP packets received on logical interfaces with the relay agent functionality enabled. This includes all DHCP broadcast and unicast packets to ensure that the whole DHCP communication between client and server can be tracked for operational and security reasons. The DHCP packets received from clients will then be forwarded to one or more DHCP servers to support high availability.

All DHCP packets sent from the relay agent to the configured DHCP servers will be sent with the IP address of the gateway interface (giaddr) or a dedicated source address per server. This allows to better traverse firewalls between relay agent and server.

1.1. Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

1.2. DHCP Relay Agent

The following are the operation steps involved by the DHCP relay agent with multiple clients and servers:

1. DHCP client sends DISCOVER packets onto the LAN.
2. DHCP relay agent receives the DISCOVER packet, creates a client binding and forwards it to the configured server(s) in different subnets.
3. DHCP server replies with OFFER packet.
4. DHCP relay agent on receiving the OFFER, sends it to the client based on the binding.
5. DHCP client sends a REQUEST after receiving the OFFER.
6. DHCP relay agent forwards the REQUEST to the configured server(s).

7. Selected DHCP server replies to the REQUEST with an ACK.
8. DHCP relay agent receives the ACK and forward it to client.
9. Client will start using the IP and configuration after receiving the ACK.
10. DHCP relay agent listen to the unicast communication between the client and server and update the bindings.

1.3. DHCP Relay in Proxy Mode

The proxy mode is an enhancement for the DHCP relay function to hide and protect the actual DHCP servers from the clients. For servers this mode is transparent but for clients it seems that the relay agent is the server and the actual servers are not visible anymore.

The following are the operation steps involved by the DHCP relay agent in proxy mode with multiple clients and servers:

1. DHCP client sends DISCOVER packets onto the LAN.
2. DHCP relay agent receives the DISCOVER packet, creates a client binding and forwards it to the configured server(s) in different subnets.
3. DHCP server replies with OFFER packet.
4. DHCP relay agent on receiving OFFERS from multiple servers, select the OFFER from the first server, replaces the server identifier with relay agent address, update the binding and sends it to the client.
5. DHCP client sends a REQUEST to relay agent address after receiving the OFFER.
6. DHCP relay agent forwards the REQUEST to the configured server(s).
7. Selected DHCP server replies to the REQUEST with an ACK.
8. DHCP relay agent receives the ACK, replaces the server identifier with relay agent address and forward it to client.
9. Client will start using the IP and configuration after receiving the ACK.
10. Client request the relay agent for the lease renewals, which relay agent will forward to the selected server.

In all the steps above, the client remains unaware of the actual DHCP server.

2. DHCP Relay Configuration

DHCP relay requires a pre-provisioned logical interface with at least an local or borrowed (unnumbered) IPv4 address configured which is then used as a gateway IP address (giaddr) by the DHCP relay function.

There are two steps required to enable the DHCP relay functionality:

1. DHCP server configuration
2. DHCP relay configuration

2.1. DHCP Server Configuration

This configuration defines the DHCP servers which are referenced by the actual DHCP relay configuration.

```

supervisor@rtbrick>LEAF01: cfg> set access dhcp-server <server>
<cr>
address                DHCP server address
routing-instance       Instance name from which DHCP server is reachable
source-address         Source address used for DHCP packets

```

The following example shows a typical DHCP server configuration.

```

{
  "rtbrick-config:access": {
    "dhcp-server": [
      {
        "server-name": "server1",
        "address": "198.51.100.101",
        "routing-instance": "default"
      }
    ]
  }
}

```

Attribute	Description
server	Specifies the DHCP server name.
address	Specifies the IPv4 address of the DHCP server.
routing-instance	Routing instance from which DHCP server is reachable. Default: default

Attribute	Description
source-address <source-address>	Specifies the source IPv4 address to be used to reach DHCP server. Default: gateway interfaces address (giaddr)

2.2. DHCP Relay Configuration

This configuration enables the DHCP relay function on the corresponding logical interface (IFL).

```

supervisor@rtbrick>LEAF01: cfg> set access dhcp-relay <interface-name>
<cr>
agent-circuit-id      Add Agent-Circuit-Id (option 82)
agent-remote-id      Add Agent-Remote-ID (option 82)
dhcp-server           DHCP server
proxy-mode           Enable relay proxy mode

```

The following example shows a typical DHCP relay configuration.

```

{
  "rtbrick-config:access": {
    "dhcp-relay": [
      {
        "interface": "ifl-0/0/1/1",
        "dhcp-server": [
          "server1",
          "server2"
        ]
      }
    ]
  }
}

```

Attribute	Description
interface-name	Logical interface name on which client packets are expected.
dhcp-server <dhcp-server>	DHCP server name to which client packets has to be forwarded (multiple servers can be configured).
proxy-mode	Enable the relay proxy mode for this interface.
agent-circuit-id <aci>	Agent-Circuit-Id to be added in DHCP option 82.
agent-remote-id <ari>	Agent-Remote-Id to be added in DHCP option 82.

2.3. DHCP Relay on Unnumbered Interface

DHCP Relay is supported over unnumbered interfaces. The configuration is similar to the regular DHCP relay configuration. The only change is that the logical interface will have a borrowed IP address.

The following example shows a typical DHCP relay configuration over unnumbered interface.

```
{
  "data": {
    "rtbrick-config:interface": [
      {
        "name": "ifp-0/0/1",
        "unit": [
          {
            "unit-id": 1,
            "unnumbered": {
              "interface": "lo-0/0/1/1"
            }
          }
        ]
      },
      {
        "name": "ifp-0/0/2",
        "unit": [
          {
            "unit-id": 1,
            "unnumbered": {
              "interface": "lo-0/0/1/1"
            }
          }
        ]
      },
      {
        "name": "lo-0/0/1",
        "unit": [
          {
            "unit-id": 1,
            "address": {
              "ipv4": [
                {
                  "prefix4": "198.51.100.71/24"
                }
              ]
            }
          }
        ]
      }
    ],
    "rtbrick-config:access": {
      "dhcp-relay": [
        {
          "interface": "if1-0/0/1/1",
          "dhcp-server": [
            "server1",
            "server2"
          ]
        }
      ]
    }
  }
}
```



```
    },  
    {  
      "interface": "if1-0/0/2/1",  
      "dhcp-server": [  
        "server1",  
        "server2"  
      ]  
    }  
  ]  
}  
}
```

3. DHCP Relay Operations

3.1. Verify DHCP Relay Configuration

A good starting point for any troubleshooting is to get an overview about the configuration.

First check the DHCP server configuration and optionally check the reachability using additional commands like `ping`.

```
supervisor@rtbrick>LEAF01: cfg> show config access dhcp-server
{
  "rtbrick-config:dhcp-server": [
    {
      "server-name": "server1",
      "address": "198.51.100.15",
      "source-address": "198.51.100.101",
      "routing-instance": "default"
    }
  ]
}
```

In the next step it should be verified that the DHCP relay function is enabled for the desired logical interfaces. Those interfaces must be configured with a valid local or borrowed (unnumbered) IPv4 address.

```
supervisor@rtbrick>LEAF01: cfg> show config access dhcp-relay
{
  "rtbrick-config:dhcp-relay": [
    {
      "interface": "ifl-0/0/1/1",
      "dhcp-server": [
        "server1",
        "server2"
      ]
    }
  ]
}
```

3.2. Verify DHCP Relay Packet Processing

All DHCP packets processed by RBFS will be handled by the IPoE daemon (`rtbrick-ipoed.1`) which separates between DHCP relay and subscriber packets.

The command `show dhcp statistics` gives some statistics about the DHCP packets send and received by the IPoE daemon.

```
supervisor@rtbrick>LEAF01: op> show dhcp statistics
Packets received           : 0
Decode error               : 0
Relay packets received    : 0
Relay packets sent        : 0
Relay send error          : 0
Subscriber packets received : 0
Subscriber packets sent   : 0
Subscriber send error     : 0
```

Received malformed packets will be counted here as part of the shared DHCP packet infrastructure.

Those statistics can be reset with the command `clear dhcp statistics`.

In the next step, the command `show dhcp relay statistics` provides more detailed statistics for those packets handled by the DHCP relay function.

```
supervisor@rtbrick>LEAF01: op> show dhcp relay statistics
Packet      Received      Sent
DHCP DISCOVER : 0          0
DHCP OFFER   : 0          0
DHCP REQUEST : 0          0
DHCP DECLINE : 0          0
DHCP ACK     : 0          0
DHCP NAK     : 0          0
DHCP RELEASE : 0          0
DHCP INFORM  : 0          0

Errors:

Invalid client packets received : 0
Invalid server packets received : 0
I/O errors                       : 0
Configuration errors             : 0
```

A DHCP packet is considered as invalid if correctly formed but some invalid or unexpected values like a DHCP hop count greater than 16, messages from server without server identifier option, non-matching transaction (XID) and many more.

Those statistics can be reset with the command `clear dhcp relay statistics`.

3.3. Verify DHCP Relay Bindings

The DHCP relay function tracks all active DHCP communications as so called bindings. The command `show dhcp relay binding` allows to verify those bindings with extensive filters to better navigate in scaled environments with thousands of bindings.

```
supervisor@rtbrick>LEAF01: op> show dhcp relay binding ?
<cr>
detail          Detailed binding information
filter          Filter DHCP relay binding
interface       Interface
```

```
supervisor@rtbrick>LEAF01: op> show dhcp relay binding filter ?
address         Client IP address
count          Count matching bindings
detail         Detailed binding information
interface       Interface
interface-regex Interface regex
mac            Hardware address
mac-regex       Hardware address regex
server         Server identifier
```

```
supervisor@rtbrick>LEAF01: op> show dhcp relay binding detail
Interface           : ifp-0/0/1/3
Hardware Address    : 02:00:00:00:00:01
Client IP Address   : 198.51.100.58
Routing Instance   : default
Proxy Mode          : enabled
Server Identifier   : 198.51.100.15
Gateway IP Address  : 198.51.100.35
Lease Time          : 120.0 sec
Lease Start         : Tue Nov 16 22:27:15 GMT +0000 2021
Lease Expires       : 66.0 sec
Last Packet Received : Tue Nov 16 22:27:15 GMT +0000 2021
Last Received Packet Type : ACK
Last Packet Sent    : Tue Nov 16 22:27:15 GMT +0000 2021
Last Sent Packet Type : ACK
```

The most extensive output is provided by selecting a particular binding using the interface and client MAC address as key.

```
supervisor@rtbrick>LEAF01: op> show dhcp relay binding interface ifp-0/0/1/3 mac 02:00:00:00:00:01
Interface                : ifp-0/0/1/3
Hardware Address          : 02:00:00:00:00:01
Client IP Address         : 198.51.100.58
Routing Instance         : default
Proxy Mode                : enabled
Server Identifier         : 198.51.100.15
Gateway IP Address        : 198.51.100.35
Lease Time                : 120.0 sec
Lease Start               : Tue Nov 16 22:31:15 GMT +0000 2021
Lease Expires             : 104.0 sec
Last Packet Received      : Tue Nov 16 22:31:15 GMT +0000 2021
Last Received Packet Type : ACK
Last Packet Sent          : Tue Nov 16 22:31:15 GMT +0000 2021
Last Sent Packet Type     : ACK

Statistics:

Packet      Received      Sent
DHCP DISCOVER : 1          1
DHCP OFFER   : 1          1
DHCP REQUEST : 36         36
DHCP DECLINE : 0          0
DHCP ACK     : 36         36
DHCP NAK     : 0          0
DHCP RELEASE : 0          0
DHCP INFORM  : 0          0

Errors:

Invalid client packets received : 0
Invalid server packets received : 0
I/O errors                       : 0
Configuration errors              : 0
```

This output returns additional statistics per DHCP relay binding.

The command **clear dhcp relay binding** removes either all bindings, all bindings of a corresponding interface or only one particular binding.

```
supervisor@rtbrick>LEAF01: op> clear dhcp relay binding ?
all          Clear all DHCP relay bindings
interface    Interface
```

```
supervisor@rtbrick>LEAF01: op> clear dhcp relay binding all
supervisor@rtbrick>LEAF01: op> clear dhcp relay binding interface ifp-0/0/1/3
supervisor@rtbrick>LEAF01: op> clear dhcp relay binding interface ifp-0/0/1/3 mac
02:00:00:00:00:01
```