NOKIA

Nokia VitalQIP

DHCP failover for a secure and stable network

White paper

To ensure their networks are stable and secure, operators depend on the Dynamic Host Configuration Protocol (DHCP). DHCP provides the dynamic address capability for network devices. With this capability, DHCP delivers the redundancy necessary to maximize availability to all network clients. This paper discusses the Nokia VitalQIP failover solution to common issues in achieving DHCP redundancy, including steps in transitioning from the primary to the failover DHCP server, as well as enabling primary server recovery. It also covers VitalQIP's many-to-one failover guidelines, failover configuration parameters, as well as how to configure a DHCP failover.

NOKIA

Contents

Introduction	3
Basic DHCP redundancy	3
VitalQIP® failover solution	5
VitalQIP failover guidelines	8
VitalQIP failover configuration parameters	9
Configuring VitalQIP DHCP failover	11
Conclusion	12

NOKIA

Introduction

The Dynamic Host Configuration Protocol (DHCP) is a network protocol that enables a server to automatically assign an IP address to a computer from a defined range of addresses, which usually consists of a subnet or scope configured for a given network. DHCP is the successor to the previous protocol called Bootstrap Protocol (BootP). DHCP operation is very similar to BootP, except BootP requires prior knowledge of the MAC address, which makes automated assignments difficult.

DHCP availability is mandatory in today's networks. That's because today's networks rely on DHCP to provide dynamic addressing for network devices. For its part, DHCP redundancy is required for maximum DHCP availability to all clients on the network. Consequently, IP administrators must implement some form of redundant DHCP server configuration.

In the event that the primary server becomes unreachable, "DHCP failover" is the phrase used to configure a secondary DHCP server that accepts the responsibility of issuing dynamic IP addresses. The operation of a DHCP failover server and the configuration of it are detailed in this document.

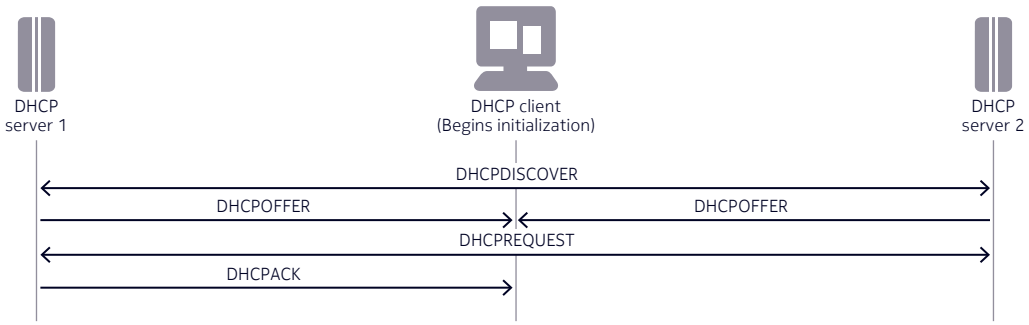
Basic DHCP redundancy

DHCP was designed from the outset to provide redundancy. Using the client's broadcast capability and the Bootp relay support established already, DHCP messages could be seen by multiple servers without clients having any foreknowledge of DHCP server addresses.

Figure 1 depicts the generic version of DHCP redundancy, using the split scope approach. The DHCPDISCOVER message is seen by two servers that both respond with an offered IP address. Using the DHCPREQUEST and DHCPACK message exchange, the DHCP client and the selected DHCP server create a BINDING of the IP address to the DHCP client's MAC address or Client Identifier (CI). The CI is a client selected string that uniquely identifies the subnet device. It can be used as an alternative to the MAC address.

Which server a client selects is strictly up to the client IP stack implementation. Also, the DHCP RFCs do not dictate the server's behavior, so how it reserves, manages, and issues the IP addresses is strictly up to each implementation. Should two servers choose to share IP addresses, it is up to those servers to keep themselves synchronized.

Figure 1. DHCP redundancy



The DHCP split scope redundancy mechanism takes a fairly simple approach that requires no communication between the servers. One drawback is that all DHCP messages are User Datagram Protocol (UDP) packets and there is no guarantee that all the servers will see the DHCPREQUEST broadcast that goes from the client back to all the servers accepting the offered address from one of the servers.

It is recommended that no two servers manage the same individual address or range of IP addresses. Consequently, installations must oversubscribe to IP address ranges on each server to handle all requests if another server should fail. For IPv4, this can be a significant disadvantage, since IPv4 addresses are in short supply.

Servers are not obligated to track addresses offered by other servers. Few, if any, DHCP servers will respond when a client is in the REBINDING state, unless it is the server that initially offered the address. Should a server fail, the clients using that server will eventually have their leases expire — forcing them back to the INIT state. This prompts the broadcast of a DHCPDISCOVER message to all available DHCP servers. If the original DHCP server remains down, the DHCPDISCOVER is handled by a different server, distributing a different IP address. Changing IP addresses while workstations are running may not sit well with the applications running on that workstation at the time, especially if they use TCP sessions.

If a company cannot tolerate the IP address change problem or does not have the IP addresses to oversubscribe address ranges on different DHCP servers, this redundancy scheme will probably not be appropriate. For this reason, a number of vendors, including Nokia, provide other DHCP backup schemes that do not have these limitations. Nokia VitalQIP supports split scopes, as well as its own implementation of DHCP failover.

VitalQIP® failover solution

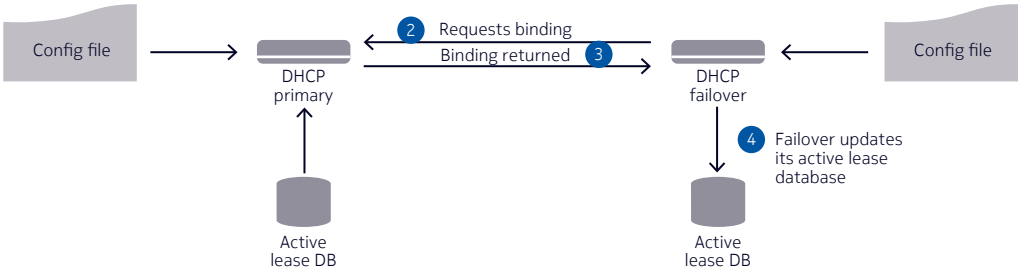
Nokia's solution to the standard DHCP redundancy issues described above is its failover server, which has the following characteristics:

- DHCP servers operate in pairs: a primary and a failover. These two servers directly communicate with each other using a software heartbeat signal and messages to transfer data on IP address leases.
- A failover cannot act as a primary while the primary is running.
- A single failover can service multiple primary servers.
- The failover operates in sleep mode when the primary is active; it only awakens if the primary fails.
- VitalQIP completely manages the configuration files on both the primary and the failover.

When the primary regains control after a failure, the lease information from the failover is synchronized with the primary per subnet. This minimizes DHCP non-availability during the sync. While a subnet is being synchronized, no leases from that subnet will be granted by either server. The time to sync a subnet varies, but it is generally measured in seconds.

Before IP address leases are granted, the initialization process shown in Figure 2 is performed.

Figure 2. Initialization process

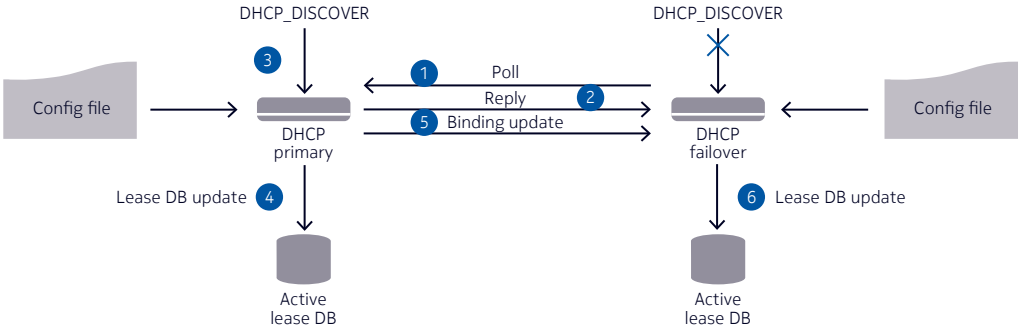


The steps completed during the initialization process are:

1. The primary DHCP server is brought up first.
2. The failover DHCP server then requests binding of the active lease file.
3. The data is sent to the failover server.
4. The failover server updates its active lease file from the primary.

Figure 3 illustrates normal DHCP operation:

Figure 3. Normal DHCP operation



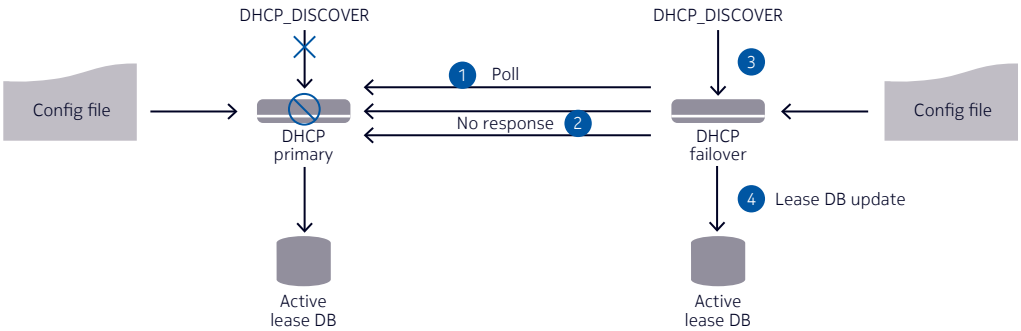
The steps shown Figure 3 are as follows:

1. Failover server polls the primary using the heartbeat process.
2. Primary replies with an alive message.
3. Primary server is healthy and accepting client messages. Failover is not accepting client messages.
4. Primary continues to update its active lease DB.
5. Primary sends updated lease information to the failover.
6. The failover updates the failover lease DB.

In summary, the primary server accepts DHCP requests and the failover server ignores those requests, as designed.

Figure 4 shows what happens when a primary VitalQIP DHCP server is unreachable.

Figure 4. Primary server failure



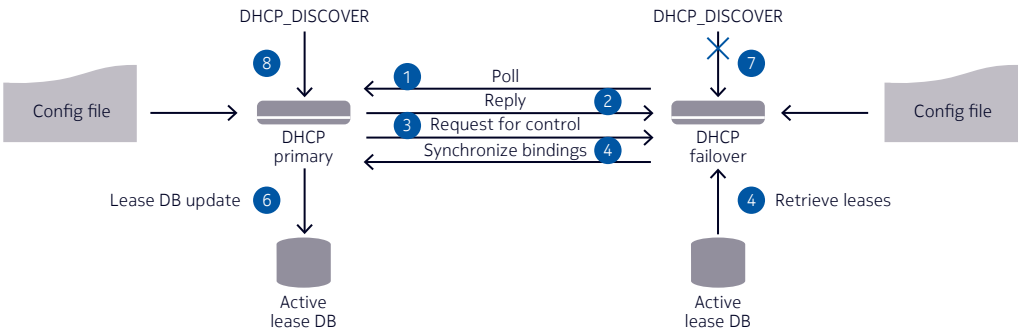
The following steps occur in this case:

1. Failover server polls the primary using the heartbeat process.
2. Primary is down and does not reply within the timeframe specified by configured failover parameters.
3. Failover server detects that the primary is down and starts accepting client messages.
4. Failover updates its lease data base (DB) with new lease information.

When the failover DHCP server fails to receive an acknowledgement message during heartbeat polling, within the timeframe specified by the failover configuration parameters, the failover process takes effect. Consequently, the failover starts accepting Discover and Request messages while the primary server is down. Also, the failover continues to update its active lease file until the primary DHCP server comes back up and requests control.

Figure 5 outlines the steps when the primary DHCP server is brought back into service:

Figure 5. Primary server recovery



The steps during the recovery process are:

1. Failover server continues to poll the primary using the heartbeat process.
2. Primary now replies with an alive message.
3. Primary server is alive and asks for control.
4. Failover retrieves lease information from its DB.
5. Failover sends updated lease information to the primary.
6. The primary updates its active lease DB.
7. The failover returns to ignoring Discover messages.

In summary, after the primary recovers, heartbeat polling resumes, and the primary DHCP server syncs its active lease file with the failover server. Then the primary server requests control and returns to running as the primary DHCP server.

It does not matter where the servers are located, as long as the routers are properly configured to send the broadcast DHCP traffic to both the primary and failover servers. This means that the primary and failover servers can be on separate subnets, Virtual Local Area Networks (VLANs) and so on. The heartbeat message also needs to communicate between the primary and the secondary server.

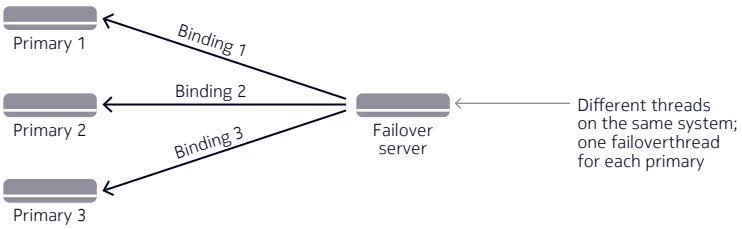
The heartbeat should always be configured on the same path as the DHCP client data so if the network on which the heartbeat travels is interrupted, the DHCP client requests are also interrupted. This ensures that if the heartbeat is broken due to network problems, the failover won't receive the DHCP client requests and give out leases.

If a network is unstable, the ping should also be used before the assign feature so if heartbeat is not detected but both primary and failover are up, no duplicate IP address can be given.

VitalQIP failover guidelines

Although VitalQIP does not limit the number of primary DHCP servers assigned to a single failover DHCP server, there are practical limits for the many-to-one configuration. From the outset, it has always been recommended that the ratio not exceed one failover to five primaries.

Figure 6. Many-to-one failover



The one-to-five ratio is recommended for several reasons. In the event of catastrophic failure — if, for example, all primary servers are down, or, if the failover server cannot communicate with the primary servers — there may not be enough processing power for the failover server to reliably replace all primary servers.

For each primary server backed up by a failover server, a separate thread is created to manage communication between servers. The failover server also manages a configurable number of threads to process DHCP client messages (default = 15). Additional threads for each failover channel cause additional context switching by the operating system, so performance can be severely hampered. In some cases, massive context switching can degrade the server to inoperable levels.

Primary servers can be backed up with DHCP many-to-one failover server support, and failover policies can be specified for each primary. This allows the failover server to "poll" the primaries at different intervals and maintain different options for retries, synchronization, and so on.

VitalQIP failover configuration parameters

VitalQIP failover parameters for the primary server are:

- CtlReqRetryMax — If no CtlRet message is received from the failover server in the time specified by WaitCtlRetSecs, attempts are made to contact the failover by resending the CtlReq message up to a specified number of times. If this maximum is reached without receiving a CtlRet response, the primary assumes that the failover is inoperable, and continues to operate as a DHCP server. It continues to send all binding changes to the server identified in the SecondaryIpAddr parameter, even though that server may not be up. The recommended maximum value is 10.
- PollDelay — The amount of time between each poll/reply sequence, specified in seconds. Upon startup, after synchronization (if specified), the failover sends a Poll message to the primary server. It waits for the amount of time specified by the WaitPollRplSecs parameter for a reply. If a reply is received, the failover server "sleeps" for the period of time specified by this parameter before sending another Poll message to the primary server. The maximum value is 86,400 (1 day).
- SyncBindRetryMax — The number of times the server should resend a binding update, if an Ack is not received within WaitSyncBindAckSecs.
- SyncBindingBufSize — Size of the "options" area for synchronizing the binding information between the primary and secondary servers. Note: This value must be the same on both servers. The minimum value is 64. The maximum value is 4096.
- Use Failover Server — If set to True, the Failover Server parameter appears, where an administrator can select the fully qualified host name of the failover server to be used by the primary server. If set to False, no failover server is associated with the primary server.
- WaitCtlRetSecs — The number of seconds that this primary server should wait for a response to the CtlReq (request for control) message sent to the failover server at start-up. The maximum number of seconds is 3,600 (1 hour). If a value of zero is supplied, the default is used.
- WaitSyncBindAckSecs — The number of seconds the server waits for an Ack to a binding update packet when operating on the sending side of synchronizing with the other server.
- WaitSyncBindUpdateSecs — The number of seconds that the server should wait for subsequent binding update packets when operating on the receiving side of synchronizing with the other server.

VitalQIP failover parameters for secondary server are as follows:

- PollRetryMax — If no reply to the Poll message is received from the primary, the failover retries sending the Poll message and waiting for a reply for this number of times before assuming that the primary has crashed, and before becoming active in handling DHCP client requests. The maximum is 10.
- SyncBindingBufSize — Size of the “options” area for synchronizing the binding information between the primary and secondary servers. Note: This value must be the same on both servers. The minimum value is 64. The maximum value is 4096.
- SyncBindings — If this parameter is True, this failover server requests all current binding information from the primary at startup.
- SyncBindRetryMax — The number of times the server should resend a binding update if an Ack is not received within WaitSyncBindAckSecs.
- SyncFailCritical — If True, the failover server terminates upon failure to synchronize bindings with the primary server. Note that this parameter only applies if the SyncBindings parameter is True.
- SyncReqRetryMax — If no SyncStart message is received from the primary server in the time specified by WaitSyncStartSecs, an attempt is made to contact the primary by resending the SyncReq message a specified number of times. If this maximum is reached without receiving a SyncStart response, this failover checks the value of the SyncFailCritical parameter. If that parameter is True, this secondary server terminates. Otherwise, this secondary server initiates polling of the primary server. The maximum value is 10.
- WaitPollRplSecs — The number of seconds that the failover should wait for a Poll reply from the primary. Note that if the failover receives a binding update from the primary during this period, the primary is assumed operational, and the binding update message is taken as a response to the poll. The maximum is 3,600 (1 hour).
- WaitSyncBindAckSecs — The number of seconds the server waits for an Ack to a binding update packet when operating on the sending side of synchronizing with the other server.
- WaitSyncBindUpdateSecs — The number of seconds that the server should wait for subsequent binding update packets when operating on the receiving side of synchronizing with the other server.
- WaitSyncStartSecs — The number of seconds that the failover server should wait for a response to the SyncReq (request for bindings) message sent to the primary server at start-up. A value of zero causes the server to wait indefinitely. The maximum value is 3,600 (1 hour).

Configuring VitalQIP DHCP failover

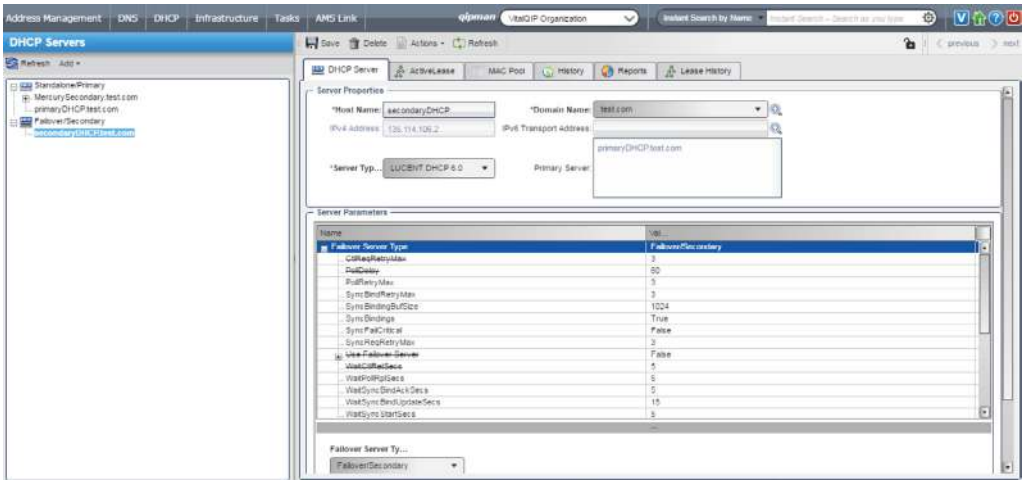
To configure DHCP failover, always review the current Administrator Guide for the release of VitalQIP you are using. Below are the basic steps for configuring DHCP failover in VitalQIP 8.1.

In the VitalQIP GUI:

1. Click on the DHCP tab and select IPv4>DHCP Servers, then **Add Server** from the **Add** button.
2. Enter the required data (host name, domain name, server type) and then configure the parameters for the DHCP server on the default DHCP server tab.
3. Click on the **Failover Server Type** and then choose **Failover/Secondary** from the list at the bottom. At this time the system adds strikethrough lines through all parameters associated with a primary DHCP server to show they are not applicable.
4. Click on each applicable parameter one by one and set the correct value, or keep the default value.
5. When done, click on the **Submit** button.

Figure 7 is a screenshot of the GUI that shows the parameters for a failover DHCP server.

Figure 7. Parameters for a failover DHCP server



NOKIA

Conclusion

Operators appreciate that one of the key ingredients in maintaining a secure and stable network is a robust DHCP solution — one that assures the redundancy necessary for maximizing availability for all network clients. The Nokia VitalQIP solution meets this challenge by addressing the most common challenges to achieving superior DHCP redundancy in networks today.

For more information on DHCP failover and Nokia VitalQIP, please visit: https://networks.nokia.com/products/vitalqip-ip-address-management

Nokia Oyj
Karaportti 3
FI-02610 Espoo
Finland
Tel. +358 (0) 10 44 88 000

Product code: PR1607021453EN (August)