BIG-IP® TMOS®: IP Routing Administration

Version 11.5.1

Table of Contents

Legal Notices

Publication Date

This document was published on September 10, 2015.

Publication Number

MAN-0412-05

Copyright

Trademarks

Patents

This product may be protected by one or more patents indicated at: *http://www.f5.com/about/guidelines-policies/patents*

Export Regulation Notice

This product may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this product from the United States.

RF Interference Warning

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

FCC Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This unit generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

Canadian Regulatory Compliance

This Class A digital apparatus complies with Canadian ICES-003.

Standards Compliance

This product conforms to the IEC, European Union, ANSI/UL and Canadian CSA standards applicable to Information Technology products at the time of manufacture.

Acknowledgments

This product includes software developed by Bill Paul.

This product includes software developed by Jonathan Stone.

This product includes software developed by Manuel Bouyer.

This product includes software developed by Paul Richards.

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes software developed by the Politecnico di Torino, and its contributors.

This product includes software developed by the Swedish Institute of Computer Science and its contributors.

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by the Computer Systems Engineering Group at the Lawrence Berkeley Laboratory.

This product includes software developed by Christopher G. Demetriou for the NetBSD Project.

This product includes software developed by Adam Glass.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Dean Huxley.

This product includes software developed by John Kohl.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Herb Peyerl.

This product includes software developed by Jochen Pohl for the NetBSD Project.

This product includes software developed by Chris Provenzano.

This product includes software developed by Theo de Raadt.

This product includes software developed by David Muir Sharnoff.

This product includes software developed by SigmaSoft, Th. Lockert.

This product includes software developed for the NetBSD Project by Jason R. Thorpe.

This product includes software developed by Jason R. Thorpe for And Communications, http://www.and.com.

This product includes software developed for the NetBSD Project by Frank Van der Linden.

This product includes software developed for the NetBSD Project by John M. Vinopal.

This product includes software developed by Christos Zoulas.

This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.

This product includes software developed by Balazs Scheidler (bazsi@balabit.hu), which is protected under the GNU Public License.

This product includes software developed by Niels Mueller (nisse@lysator.liu.se), which is protected under the GNU Public License.

In the following statement, *This software* refers to the Mitsumi CD-ROM driver: This software was developed by Holger Veit and Brian Moore for use with 386BSD and similar operating systems. *Similar operating systems* includes mainly non-profit oriented systems for research and education, including but not restricted to NetBSD, FreeBSD, Mach (by CMU).

This product includes software developed by the Apache Group for use in the Apache HTTP server project (http://www.apache.org/).

This product includes software licensed from Richard H. Porter under the GNU Library General Public License (© 1998, Red Hat Software), www.gnu.org/copyleft/lgpl.html.

This product includes the standard version of Perl software licensed under the Perl Artistic License (© 1997, 1998 Tom Christiansen and Nathan Torkington). All rights reserved. You may find the most current standard version of Perl at http://www.perl.com.

This product includes software developed by Jared Minch.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product contains software based on oprofile, which is protected under the GNU Public License.

This product includes software with glib library utility functions, which is protected under the GNU Public License.

This product includes software with grub2 bootloader functions, which is protected under the GNU Public License.

This product includes software with the Intel Gigabit Linux driver, which is protected under the GNU Public License. Copyright ©1999 - 2012 Intel Corporation.

This product includes software with the Intel 10 Gigabit PCI Express Linux driver, which is protected under the GNU Public License. Copyright ©1999 - 2012 Intel Corporation.

This product includes RRDtool software developed by Tobi Oetiker (http://www.rrdtool.com/index.html) and licensed under the GNU General Public License.

This product contains software licensed from Dr. Brian Gladman under the GNU General Public License (GPL).

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

This product includes Hypersonic SQL.

This product contains software developed by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and others.

This product includes software developed by the Internet Software Consortium.

This product includes software developed by Nominum, Inc. (http://www.nominum.com).

This product contains software developed by Broadcom Corporation, which is protected under the GNU Public License.

This product contains software developed by MaxMind LLC, and is protected under the GNU Lesser General Public License, as published by the Free Software Foundation.

This product includes software developed by Andrew Tridgell, which is protected under the GNU Public License, copyright ©1992-2000.

This product includes software developed by Jeremy Allison, which is protected under the GNU Public License, copyright ©1998.

This product includes software developed by Guenther Deschner, which is protected under the GNU Public License, copyright ©2008.

This product includes Intel QuickAssist kernel module, library, and headers software licensed under the GNU General Public License (GPL).

This product includes gd-libgd library software developed by the following in accordance with the following copyrights:

- Portions copyright ©1994, 1995, 1996, 1997, 1998, 2000, 2001, 2002 by Cold Spring Harbor Laboratory. Funded under Grant P41-RR02188 by the National Institutes of Health.
- Portions copyright ©1996, 1997, 1998, 1999, 2000, 2001, 2002 by Boutell.Com, Inc.
- Portions relating to GD2 format copyright ©1999, 2000, 2001, 2002 Philip Warner.
- Portions relating to PNG copyright ©1999, 2000, 2001, 2002 Greg Roelofs.
- Portions relating to gdttf.c copyright ©1999, 2000, 2001, 2002 John Ellson (ellson@lucent.com).
- Portions relating to gdft.c copyright ©2001, 2002 John Ellson (ellson@lucent.com).
- Portions copyright ©2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007 2008 Pierre-Alain Joye (pierre@libgd.org).
- Portions relating to JPEG and to color quantization copyright ©2000, 2001, 2002, Doug Becker and copyright ©1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, Thomas G. Lane. This software is based in part on the work of the Independent JPEG Group.
- Portions relating to WBMP copyright 2000, 2001, 2002 Maurice Szmurlo and Johan Van den Brande. Permission has been granted to copy, distribute and modify gd in any context without fee, including a commercial application, provided that this notice is present in user-accessible supporting documentation.

This product includes software developed by Oracle America, Inc. Copyright ©2012.

1. Java Technology Restrictions. Licensee shall not create, modify, change the behavior of, or authorize licensees of licensee to create, modify, or change the behavior of, classes, interfaces, or subpackages that are in any way identified as "java", "javax", "sun" or similar convention as specified by Oracle in any naming convention designation. In the event that Licensee creates an additional API(s) which: (a) extends the functionality of a Java Environment; and (b) is exposed to third party software developers for the purpose of developing additional software which invokes such additional API, Licensee must promptly publish broadly an accurate specification for such API for free use by all developer.
2. Trademarks and Logos. This License does not authorize an end user licensee to use any Oracle America, Inc. name, trademark, service mark, logo or icon. The end user licensee acknowledges that Oracle owns the Java trademark and all Java-related trademarks, logos and icon including the Coffee Cup and Duke ("Java Marks") and agrees to: (a) comply with the Java Trademark Guidelines at http://www.oraclc.com/html/3party.html; (b) not do anything harmful to or inconsistent with Oracle's rights in the Java Marks; and (c) assist Oracle in protecting those rights, including assigning to Oracle any rights acquired by Licensee in any Java Mark.
3. Source Code. Software may contain source code that, unless expressly licensed for other purposes, is provided solely for reference purposes pursuant to the terms of your license. Source code may not be redistributed unless expressly provided for in the terms of your license.
4. Third Party Code. Additional copyright notices and license terms applicable to portion of the Software are set forth in the THIRDPARTYLICENSEREADME.txt file.
5. Commercial Features. Use of the Commercial Features for any commercial or production purpose requires a separate license from Oracle. "Commercial Features" means those features identified in Table I-I (Commercial Features In Java SE Product Editions) of tile Software documentation accessible at http://www.oracle.com/technetwork/java/javase/documentation/index.html.

This product includes utilities developed by Linus Torvalds for inspecting devices connected to a USB bus.

This product includes perl-PHP-Serialization software, developed by Jesse Brown, copyright ©2003, and distributed under the Perl Development Artistic License (http://dev.perl.org/licenses/artistic.html).

This product includes software developed by members of the CentOS Project under the GNU Public License, copyright ©2004-2011 by the CentOS Project.

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product includes software developed by Douglas Gilbert.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE FREEBSD PROJECT ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FREEBSD PROJECT OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

This product includes software developed as open source software.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)"

a. be disclosed or distributed in source code form; or
b. be licensed by the user to third parties for the purpose of making and/or distributing derivative works; or
c. be redistributable at no charge.

Open Source Software includes, without limitation, software licensed or distributed under any of the following licenses or distribution models, or licenses or distribution models substantially similar to any of the following:

a. GNU's General Public License (GPL) or Lesser/Library GPL (LGPL),
b. the Artistic License (e.g., PERL),
c. the Mozilla Public License,
d. the Netscape Public License,
e. the Sun Community Source License (SCSL),
f. vi) the Sun Industry Source License (SISL),
g. vii) the Apache Software license, and
h. viii) the Common Public License (CPL).

3. OWNERSHIP OF SOFTWARE AND COPYRIGHTS. Title to all copies of the Software remains with Intel or its suppliers. The Software is copyrighted and protected by the laws of the United States and other countries, and international treaty provisions. You may not remove any copyright notices from the Software. Intel may make changes to the Software, or to materials referenced therein, at any time and without notice, but is not obligated to support or update the Software. Except as otherwise expressly provided, Intel grants no express or implied right or license under Intel patents, copyrights, trademarks, or other intellectual property rights.
4. Entire Agreement. This Agreement contains the complete and exclusive statement of the agreement between You and Intel and supersedes all proposals, oral or written, and all other communications relating to the subject matter of this Agreement. Only a written instrument duly executed by authorized representatives of Intel and You may modify this Agreement.
5. LIMITED MEDIA WARRANTY. If the Software has been delivered by Intel on physical media, Intel warrants the media to be free from material physical defects for a period of ninety (90) days after delivery by Intel. If such a defect is found, return the media to Intel for replacement or alternate delivery of the Software as Intel may select.
6. EXCLUSION OF OTHER WARRANTIES. EXCEPT AS PROVIDED ABOVE, THE SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND, INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. Intel does not warrant or assume responsibility for any errors, the accuracy or completeness of any information, text, graphics, links or other materials contained within the Software.
7. LIMITATION OF LIABILITY. IN NO EVENT WILL INTEL OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION OR LOST INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF INTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS PROHIBIT EXCLUSION OR LIMITATION OF LIABILITY FOR IMPLIED WARRANTIES OR CONSEQUENTIAL OR INCIDENTAL DAMAGES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU. YOU MAY ALSO HAVE OTHER LEGAL RIGHTS THAT VARY FROM JURISDICTION TO JURISDICTION.
8. TERMINATION OF THIS AGREEMENT. Intel may terminate this Agreement at any time if You violate its terms. Upon termination, You will immediately destroy the Software or return all copies of the Software to Intel.
9. APPLICABLE LAWS. Claims arising under this Agreement will be governed by the laws of Delaware, excluding its principles of conflict of laws and the United Nations Convention on Contracts for the Sale

of Goods. You may not export the Software in violation of applicable export laws and regulations. Intel is not obligated under any other agreements unless they are in writing and signed by an authorized representative of Intel.

10. GOVERNMENT RESTRICTED RIGHTS. The Software is provided with "RESTRICTED RIGHTS." Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or their successors. Use of the Software by the Government constitutes acknowledgment of Intel's proprietary rights therein. Contractor or Manufacturer is Intel Corporation, 2200 Mission College Blvd., Santa Clara, CA 95054.

Chapter 1

Overview of TMOS Routing

Overview of IP routing administration in TMOS

As a BIG-IP ® system administrator, you typically manage routing on the system by configuring these BIG-IP system features.

Table 1: BIG-IP system features for route configuration

BIG-IP system feature	Benefit
Route domains	You create route domains to segment traffic associated with different applications and to allow devices to have duplicate IP addresses within the same network.
Local IP addresses	Whenever you create virtual addresses and self IP addresses on the BIG-IP system, the system automatically adds routes to the system that pertain to those addresses, as directly-connected routes.
Static routes	For destination IP addresses that are not on the directly-connected network, you can explicitly add static routes. You can add both management (administrative) and TMM static routes to the BIG-IP system.
Advanced routing modules	You can configure the advanced routing modules--a set of dynamic routing protocols and core daemons--to ensure that the BIG-IP system can learn about routes from other routers and advertise BIG-IP system routes. These advertised routes can include BIG-IP virtual addresses.
The ARP cache	You can manage static and dynamic entries in the ARP cache to resolve IP addresses into MAC addresses.

About BIG-IP system routing tables

The BIG-IP system contains two sets of routing tables:

- The Linux routing tables, for routing administrative traffic through the management interface

- A special TMM routing table, for routing application and administrative traffic through the TMM interfaces

As a BIG-IP administrator, you configure the system so that the BIG-IP system can use these routing tables to route both management and application traffic successfully.

About BIG-IP management routes and TMM routes

The BIG-IP system maintains two kinds of routes:

Management routes

Management routes are routes that the BIG-IP system uses to forward traffic through the special management interface. The BIG-IP system stores management routes in the Linux (that is, kernel) routing table.

TMM routes

TMM routes are routes that the BIG-IP system uses to forward traffic through the Traffic Management Microkernel (TMM) interfaces instead of through the management interface. The BIG-IP system stores TMM routes in both the TMM and kernel routing tables.

Viewing routes on the BIG-IP system

You can use the `tmsh` utility to view different kinds of routes on the BIG-IP system.

1. Open a console window, or an SSH session using the management port, on the BIG-IP system.
2. Use your user credentials to log in to the system.
3. Perform one of these actions at the command prompt:
    - To view all routes on the system, type: `tmsh show /net route`
    - To view all configured static routes on the system, type: `tmsh list /net route`

You are now able to view BIG-IP system routes.

Chapter 2

Working with Route Domains

What is a route domain?

A *route domain* is a configuration object that isolates network traffic for a particular application on the network.

Because route domains segment network traffic, you can assign the same IP address or subnet to multiple nodes on a network, provided that each instance of the IP address resides in a separate routing domain.

Note: *Route domains are compatible with both IPv4 and IPv6 address formats.*

Important: *For systems that include both BIG-IP® Local Traffic Manager™ (LTM) and BIG-IP Global Traffic Manager™ (GTM), you can configure route domains on internal interfaces only.*

Benefits of route domains

Using the route domains feature of the BIG-IP® system, you can provide hosting service for multiple customers by isolating each type of application traffic within a defined address space on the network.

With route domains, you can also use duplicate IP addresses on the network, provided that each of the duplicate addresses resides in a separate route domain and is isolated on the network through a separate VLAN. For example, if you are processing traffic for two different customers, you can create two separate route domains. The same node address (such as `10.0.10.1`) can reside in each route domain, in the same pool or in different pools, and you can assign a different monitor to each of the two corresponding pool members.

Sample partitions with route domain objects

This illustration shows two route domain objects on a BIG-IP system, where each route domain corresponds to a separate customer, and thus resides in its own partition. Within each partition, the customer created the network objects and local traffic objects required for that customer's application (`AppA` or `AppB`).

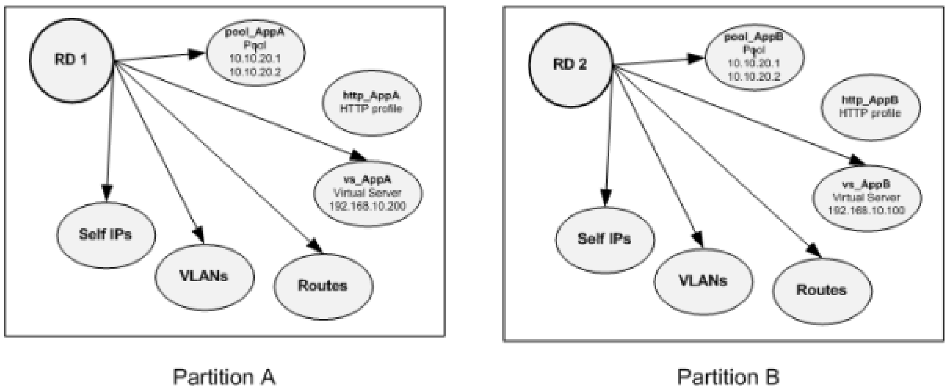


Figure 1: Sample partitions with route domains

Sample route domain deployment

A good example of the use of route domains is a configuration for an ISP that services multiple customers, where each customer deploys a different application. In this case, the BIG-IP system isolates traffic for two different applications into two separate route domains. The routes for each application's traffic cannot cross route domain boundaries because cross-routing restrictions are enabled on the BIG-IP system by default.

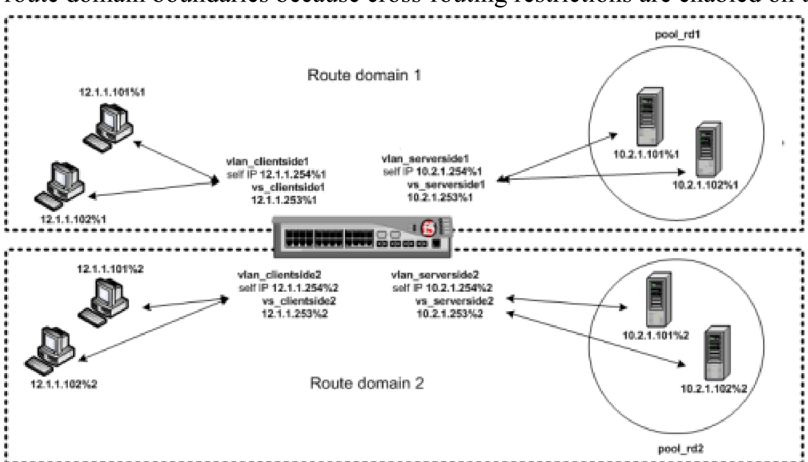


Figure 2: A sample route domain deployment

About route domain IDs

A *route domain ID* is a unique numerical identifier for a route domain. You can assign objects with IP addresses (such as self IP addresses, virtual addresses, pool members, and gateway addresses) to a route domain by appending the `%ID` to the IP address.

The format required for specifying a route domain ID in an object's IP address is `A.B.C.D%ID`, where *`ID`* is the ID of the relevant route domain. For example, both the local traffic node object `10.10.10.30%2` and the pool member `10.10.10.30%2:80` pertain to route domain `2`.

The BIG-IP system includes a default route domain with an ID of `0`. If you do not explicitly create any route domains, all routes on the system pertain to route domain `0`.

Important: *A route domain ID must be unique on the BIG-IP system; that is, no two route domains on the system can have the same ID.*

Traffic forwarding across route domains

You can create a parent-child relationship between two route domains, and configure strict isolation, to control the extent to which the BIG-IP® system can forward traffic from one route domain to another.

About parent IDs

When you create a route domain, you can specify the ID of another route domain as the parent route domain. The *parent ID* identifies another route domain that the system can search to find a route if the system cannot find the route within the child route domain.

For example, using the BIG-IP® Configuration utility, suppose you create route domain `1` and assign it a parent ID of `0`. For traffic pertaining to route domain `1`, the system looks within route domain `1` for a route for the specified destination. If no route is found, the system searches the routes in route domain `0`.

By default, if the system finds no route in the parent route domain, the system searches the parent route domain's parent, and so on, until the system finds either a match or a route domain with no parent. In the latter case, the system refrains from searching any other route domains to find a match, thus preventing the system from using a route from another route domain.

You can disable this behavior on a route domain.

About strict isolation

You can control the forwarding of traffic across route domain boundaries by configuring the *strict isolation* feature of a route domain:

- If strict isolation is enabled, the BIG-IP® system allows traffic forwarding from that route domain to the specified parent route domain only. This is the default behavior. Note that for successful isolation, you must enable the strict isolation feature on both the child and the parent route domains.
- If strict isolation is disabled, the BIG-IP system allows traffic forwarding from that route domain to any route domain on the system, without the need to define a parent-child relationship between route domains.

Note that in this case, for successful forwarding, you must disable the strict isolation feature on both the forwarding route domain and the target route domain (that is, the route domain to which the traffic is being forwarded).

About default route domains for administrative partitions

The route domains feature includes the concept of default route domains, to minimize the need for you to specify the %ID notation. When you designate a route domain as the *default route domain* in a partition, any BIG-IP system objects in that partition that do not include the %ID notation in their IP addresses are automatically associated with the default route domain.

The default route domain for partition Common

The BIG-IP system, by default, includes one route domain, named route domain `0`. Route domain `0` is known as the *default route domain* on the BIG-IP system, and this route domain resides in administrative partition `Common`. If you do not create any other route domains on the system, all traffic automatically pertains to route domain `0`.

If you want to segment traffic into multiple route domains, you can create additional route domains in partition `Common` and then segment application traffic among those route domains. Any BIG-IP addresses that do not include the route domain ID notation are automatically associated with the default route domain.

Note: *Any VLANs that reside in partition* `Common` *are automatically assigned to the default route domain.*

The default route domain for other partitions

For administrative partitions other than `Common`, you can create a route domain and designate it as a *partition default route domain*. A partition can contain one partition default route domain only.

The benefit of having a partition default route domain is that when you create objects such as a virtual server and pool members within that partition, you do not need to specify the ID of that default route domain within the addresses for those objects. For example, if you create a partition default route domain with an ID of `2` in partition `A`, the system automatically assigns any partition `A` object IP addresses without a route domain ID to route domain `2`.

If no partition default route domain exists within the partition, the system associates those addresses with route domain `0` in partition `Common`.

About VLANs and tunnels for a route domain

You can assign one or more VLANs, VLAN groups, or tunnels to a route domain. The VLANs, VLAN groups, or tunnels that you assign to a route domain are those pertaining to the particular traffic that you want to isolate in that route domain. Each VLAN, VLAN group, or tunnel can be a member of one route domain only.

When you assign a VLAN group to a route domain, the BIG-IP system automatically assigns the VLAN group members to the route domain.

Please note the following facts:

- If you delete a VLAN group from the system, the VLAN group members remain assigned to the route domain.

- If a VLAN is assigned to a non-default route domain and you delete that route domain, the BIG-IP system automatically assigns the VLAN to the default route domain for that partition.
- When you create VLANs, VLAN groups, and tunnels, the BIG-IP system automatically assigns them to the default route domain of the current partition. You can change this assignment when you create other route domains in the partition.

About advanced routing modules for a route domain

For each route domain that you configure, you can enable one or more dynamic routing protocols, as well as the network protocol Bidirectional Forwarding Detection (BFD). Use of dynamic routing and BFD for route domain `0` or any other route domain is optional.

About throughput limits on route domain traffic

When you configure more than one route domain on the BIG-IP system, the traffic from one particular route domain can potentially consume an inordinate amount of BIG-IP system resource. To prevent this, you can define the amount of BIG-IP system resource that traffic for each route domain can consume.

You do this by assigning a different throughput limit to each route domain. This throughput limit is defined in a *bandwidth controller policy*. For example, for route domain `1`, you can assign a static bandwidth controller policy that specifies a throughput limit of 10 Gbps, while for route domain `2`, you can assign a static bandwidth controller policy that specifies a throughput limit of 20 Gbps. When you assign a different bandwidth controller policy to each route domain, traffic for one route domain does not cross the boundary into another route domain on the system.

Important: *The BIG-IP system applies a bandwidth controller policy to a route domain's egress traffic only, that is, the traffic that a server within a particular route domain sends back through the BIG-IP system on its way to the client on the public network. A bandwidth controller policy is not applied to traffic coming from the public network to the route domain on the internal network.*

Creating a route domain on the BIG-IP system

Before you create a route domain:

- Ensure that an external and an internal VLAN exist on the BIG-IP® system.
- If you intend to assign a static bandwidth controller policy to the route domain, you must first create the policy. You can do this using the BIG-IP Configuration utility.
- Verify that you have set the current partition on the system to the partition in which you want the route domain to reside.

You can create a route domain on BIG-IP system to segment (isolate) traffic on your network. Route domains are useful for multi-tenant configurations.

1. On the Main tab, click **Network** > **Route Domains**.
   The Route Domain List screen opens.
2. Click **Create**.
   The New Route Domain screen opens.

3. In the **Name** field, type a name for the route domain.

   This name must be unique within the administrative partition in which the route domain resides.

4. In the **ID** field, type an ID number for the route domain.

   This ID must be unique on the BIG-IP system; that is, no other route domain on the system can have this ID.

5. In the **Description** field, type a description of the route domain.
   For example: `This route domain applies to traffic for application MyApp.`
6. For the **Strict Isolation** setting, select the **Enabled** check box to restrict traffic in this route domain from crossing into another route domain.
7. For the **Parent Name** setting, retain the default value.
8. For the **VLANs** setting, from the **Available** list, select a VLAN name and move it to the **Members** list.

   Select the VLAN that processes the application traffic relevant to this route domain.

   Configuring this setting ensures that the BIG-IP system immediately associates any self IP addresses pertaining to the selected VLANs with this route domain.

9. For the **Dynamic Routing Protocols** setting, from the **Available** list, select one or more protocol names and move them to the **Enabled** list.

   You can enable any number of listed protocols for this route domain. This setting is optional.

10. From the **Bandwidth Controller** list, select a static bandwidth control policy to enforce a throughput limit on traffic for this route domain.
11. From the **Partition Default Route Domain** list, select either **Another route domain (0) is the Partition Default Route Domain** or **Make this route domain the Partition Default Route Domain**.

    This setting does not appear if the current administrative partition is partition `Common`.

    When you configure this setting, either route domain `0` or this route domain becomes the default route domain for the current administrative partition.
12. Click **Finished**.
    The system displays a list of route domains on the BIG-IP system.

You now have another route domain on the BIG-IP system.

Chapter 3

Working with Static Routes

Static route management on the BIG-IP system

Part of managing routing on a BIG-IP® system is to add static routes for destinations that are not located on the directly-connected network. If you are using the route domains feature, you can specify a route domain ID as part of each IP address that you include in a static route entry.

Adding a static route

Before adding a route, if the IP addresses in the route pertain to any route domains, verify that the relevant route domains are present on the system.

Perform this task when you want to explicitly add a route for a destination that is not on the directly-connected network. Depending on the settings you choose, the BIG-IP system can forward packets to a specified network device (such as a next-hop router or a destination server), or the system can drop packets altogether.

1. On the Main tab, click **Network** > **Routes**.
2. Click **Add**.
   The New Route screen opens.
3. In the **Name** field, type a unique user name.

   This name can be any combination of alphanumeric characters, including an IP address.
4. In the **Description** field, type a description for this route entry.

   This setting is optional.
5. In the **Destination** field, type either the destination IP address for the route, or IP address `0.0.0.0` for the default route.

   This address can represent either a host or a network. Also, if you are using the route domains and the relevant route domain is the partition default route domain, you do not need to append a route domain ID to this address.
6. In the **Netmask** field, type the network mask for the destination IP address.
7. From the **Resource** list, specify the method through which the system forwards packets:

Option	Description
Use Gateway	Select this option when you want the next hop in the route to be a network IP address. This choice works well when the destination is a pool member on the same internal network as this gateway address.
Use Pool	Select this option when you want the next hop in the route to be a pool of routers instead of a single next-hop router. If you select this option, verify that you have created a pool on the BIG-IP system, with the routers as pool members.
Use VLAN/Tunnel	Select this option when you want the next hop in the route to be a VLAN or tunnel. This option works well when the destination address you specify in the routing entry is a network address. Selecting a VLAN/tunnel name as the resource implies that the specified network is directly connected to the BIG-IP system. In this case, the BIG-IP system can find the destination host simply by sending an ARP request to the hosts in the specified VLAN, thereby obtaining the destination host's MAC address.
Reject	Select this option when you want the BIG-IP system to reject packets sent to the specified destination.

8. In the **MTU** field, specify in bytes a maximum transmission unit (MTU) for this route.
9. At the bottom of the screen, click **Finished**.

After you perform this task, a static route is defined on the BIG-IP system with IP addresses that can pertain to one or more route domains.

You should define a default route for each route domain on the system. Otherwise, certain types of administrative traffic that would normally use a TMM interface might instead use the management interface.

Chapter

4

Working with Dynamic Routing

Dynamic routing on the BIG-IP system

By enabling and configuring any of the BIG-IP® advanced routing modules, you can configure dynamic routing on the BIG-IP system. You enable one or more advanced routing modules, as well as the Bidirectional Forwarding Detection (BFD) protocol, on a per-route-domain basis. Advanced routing module configuration on the BIG-IP system provides these functions:

- Dynamically adds routes to the Traffic Management Microkernel (TMM) and host route tables.
- Advertises and redistributes routes for BIG-IP virtual addresses to other routers.
- When BFD is enabled, detects failing links more quickly than would normally be possible using the dynamic routing protocols' own detection mechanisms.

Note: *On the BIG-IP system, directly-connected and static routes take precedence over dynamically-learned routes.*

Supported protocols for dynamic routing

The BIG-IP® advanced routing modules support these protocols.

Table 2: Dynamic routing protocols

Protocol Name	Description	Daemon	IP version supported
BFD	*Bidirectional Forwarding Detection* is a protocol that detects faults between two forwarding engines connected by a link. On the BIG-IP system, you can enable the BFD protocol for the OSPFv2, BGP4, and IS-IS dynamic routing protocols specifically.	`oamd`	IPv4 and IPv6
BGP4	*Border Gateway Protocol (BGP)* with multi-protocol extension is a dynamic routing protocol for external networks that supports the IPv4 and IPv6 addressing formats.	`bgpd`	IPv4 and IPv6
IS-IS	*Intermediate System-to-Intermediate System (IS-IS)* is a dynamic routing protocol for internal networks, based on a link-state algorithm.	`isisd`	IPv4 and IPv6
OSPFv2	The *Open Shortest Path First (OSPF)* protocol is a dynamic routing protocol for internal networks, based on a link-state algorithm.	`ospfd`	IPv4
OSPFv3	The *OSPFv3* protocol is an enhanced version of OSPFv2.	`ospf6d`	IPv6
RIPv1/RIPv2	*Routing Information Protocol (RIP)* is a dynamic routing protocol for internal networks, based on a distance-vector algorithm (number of hops).	`ripd`	IPv4
RIPng	The *RIPng* protocol is an enhanced version of RIPv2.	`ripngd`	IPv6

About the Bidirectional Forwarding Detection protocol

Bidirectional Forwarding Detection (BFD) is an industry-standard network protocol on the BIG-IP® system that provides a common service to the dynamic routing protocols BGPv4, OSPFv2, and IS-IS. Enabled on a per-route domain basis, BFD identifies changes to the connectivity between two forwarding engines, or endpoints, by transmitting periodic BFD control packets on each path between the two endpoints. When either endpoint fails to receive these control packets for a specific duration of time, the connectivity between the endpoints is considered lost, and BFD notifies the associated dynamic routing protocols. In general, BFD detects connectivity changes more rapidly than the endpoints' standard Hello mechanisms, leading to quicker network convergence, which is highly desirable to data center applications.

BFD operates by establishing a session between two endpoints, sending BFD control packets over the link. If more than one link exists between two endpoints, BFD can establish multiple sessions to monitor each link.

A BFD session can operate in one of two modes, either asynchronous mode or demand mode:

- You configure BFD to operate in *asynchronous mode* when you want both endpoints to verify connectivity by periodically sending Hello packets to each other. This is the most commonly-used mode.
- You configure BFD to operate in *demand mode* when you want the endpoints to use another way to verify connectivity to each other instead of sending Hello packets. For example, the endpoints might verify connectivity at the underlying physical layer. Note, however, that in demand mode, either host can send Hello packets if needed.

Note: *BFD failure detection between two BIG-IP systems does not trigger failover.*

Configuration overview

The first step in configuring the Bidirectional Forwarding Detection (BFD) protocol on the BIG-IP® system is to use the IMI Shell within `tmsh` to configure the protocol for the relevant advanced routing modules (BGP4, OSPFv2, and IS-IS):

- Because BFD does not include a discovery mechanism, you must explicitly configure BFD sessions between endpoints.
- The BFD protocol requires you to commit a nominal amount of additional system resources, in the form of timers, interface bandwidth, and system memory.

After configuring BFD protocol behavior, you enable the protocol on one or more specific route domains.

Important: *You can find detailed documentation on BFD commands in the AskF5™ knowledge base at* `http://support.f5.com`.

Enabling the BFD protocol for a route domain

Before you perform this task, verify that you have configured the **Port Lockdown** setting on all self IP addresses with which routers must communicate. Specifically, you must configure self IP addresses to allow TCP connections on the relevant service port.

You must enable the Bidirectional Forwarding Detection (BFD) network protocol on a per-route domain basis. Use this task to enable BFD on an existing route domain.

1. On the Main tab, click **Network** > **Route Domains**.
   The Route Domain List screen opens.
2. In the Name column, click the name of the relevant route domain.
3. For the **Dynamic Routing Protocols** setting, from the **Available** list, select **BFD** and move it to the **Enabled** list.

   When you enable BFD, the BIG-IP system starts one BFD session for the route domain, and this session supports the BGP4, IS-IS, and OSPFv2 protocols.
4. Click **Update**.
   The system displays the list of route domains on the BIG-IP system.

After you perform this task, the BIG-IP® system starts the daemon `oamd`. Once enabled, the BFD protocol automatically restarts whenever the BIG-IP system is restarted.

Common commands for BFD base configuration

There are two common BFD commands that you can use to perform BFD base configuration. To use these commands, you use the IMI Shell within `tmsh`.

Sample command line sequence	Result
`bigip (config-if)# bfd interval 100 minrx 200 multiplier 4`	Sets desired Min Tx, required Min Rx, and detect Multiplier.
`bigip (config)# bfd slow-timer 2000`	Sets BFD slow timer to two seconds.

Common commands for BFD routing configuration

There are a number of common BFD commands that you can use to perform BFD routing configuration. To use these commands, you use the IMI Shell within `tmsh`.

Protocol	Sample command line sequence	Result
BGP4	`bigip (config-if)# neighbor 1.1.1.1 fallover bfd multihop`	Enables multi-hop bidirectional forwarding detection to BGP neighbor `1.1.1.1`.
OSPFv2	`bigip (config)# bfd all-interfaces`	Enables single-hop bidirectional forwarding detection for all OSPF neighbors.
OSPFv2	`bigip (config)# area 1 virtual-link 3.3.3.3 fallover bfd`	Enables multi-hop bidirectional forwarding detection to OSPF router `3.3.3.3`.
IS-IS	`bigip (config-if)# bfd all-interfaces`	Enables bidirectional forwarding detection for all IS-IS neighbors.

About ECMP routing

Some of the advanced routing modules on the BIG-IP® system include support for Equal Cost Multipath (ECMP) routing. *ECMP* is a forwarding mechanism for routing a traffic flow along multiple paths of equal cost, with the goal of achieving equally-distributed link load sharing. By load balancing traffic over multiple paths, ECMP offers potential increases in bandwidth, as well as some level of fault tolerance when a path on the network becomes unavailable.

Advanced routing modules that support ECMP

The BIG-IP® system deploys Equal Cost Multipath (ECMP) routing with these advanced routing modules:

- BGP4
- IS-IS
- OSPFv2
- OSPFv3
- RIPv1
- RIPv2

The ECMP protocol is enabled by default for all of these advanced routing modules except BGP4. For BGP4, you must explicitly enable the ECMP forwarding mechanism.

Enabling the ECMP protocol for BGP4

You can enable the Equal Cost Multipath (ECMP) forwarding mechanism for the BGP4 advanced routing module, using the Traffic Management Shell (`tmsh`) command line interface. When you enable ECMP for BGP4, the BIG-IP® system provides multiple paths for a traffic flow to choose from, in order to reach the destination.

Important: *For all other advanced routing modules, the ECMP protocol is enabled by default.*

1. Open a console window, or an SSH session using the management port, on a BIG-IP system.
2. Use your user credentials to log in to the system.
3. At the command prompt, type `tmsh`.
   This opens the `tmsh` shell.
4. Type this command: `run /util imish -r ID`.
   The *`ID`* variable represents the route domain ID.
   This command invokes the IMI shell.
5. Type `enable`.
6. Type `configure terminal`.
7. Type this command: `bgp max-paths (ebgp|ibgp|) 2-64`

After you perform this task, the ECMP forwarding mechanism is enabled for the BGP4 advanced routing module.

Viewing routes that use ECMP

You can perform this task to view the dynamic routes on the system that are using the Equal Cost Multipath (ECMP) forwarding mechanism.

1. Open a console window, or an SSH session using the management port, on a BIG-IP® system.
2. Use your user credentials to log in to the system.
3. At the command prompt, type `tmsh show net route`.

The system displays all dynamic routes and indicates the routes that are using ECMP.

Location of startup configuration for advanced routing modules

When you enable advanced routing modules for a route domain, the BIG-IP system creates a dynamic routing startup configuration. Each route domain has its own dynamic routing configuration, located in the folder `/config/zebos/rdn`, where *`n`* is the numeric route domain ID.

Warning: *F5 Networks strongly discourages manual modifications to the startup configuration (such as by using a text editor). Doing so might lead to unexpected results.*

Accessing the IMI Shell

Perform this task when you want to use IMI Shell (`imish`) to configure any of the dynamic routing protocols. Note that if you are using the route domains feature, you must specify the route domain pertaining to the dynamic routing protocol that you want to configure.

1. Open a console window, or an SSH session using the management port, on a BIG-IP device.
2. Use your user credentials to log in to the system.

3. At the command prompt, type `tmsh`.
   This opens the `tmsh` shell.
4. Type this command: `run /util imish -r ID`.

   If the route domain for the protocol you want to configure is the default route domain for the current partition, you do not need to use the `-r` option to specify the route domain ID.

   This command invokes the IMI shell.

You can now use any of the IMI shell commands.

Relationship of advanced routing modules and BFD to route domains

For each route domain on the BIG-IP system (including route domain `0`), you can enable one or more dynamic routing protocols, as well as the network protocol Bidirectional Forwarding Detection (BFD). For example, you can enable BGP4 and OSPFv3 on a specific route domain. Use of dynamic routing protocols for a route domain is optional.

When you enable dynamic routing on a specific route domain, the BIG-IP system creates a dynamic routing instance. This dynamic routing instance is made up of the core dynamic routing daemons (`imi` and `nsm`), as well each relevant dynamic routing protocol daemon. If you enable BFD, the BFD instance is made up of the `oamd` protocol daemon. Thus, each dynamic routing instance for a route domain has a separate configuration. You manage a dynamic routing configuration using the IMI shell (`imish`).

Enabling a protocol for a route domain

Before you perform this task, verify that you have configured the **Port Lockdown** setting on all self IP addresses with which routers must communicate. Specifically, you must configure self IP addresses to allow TCP connections on the relevant service port. For example, for BGP4, you must configure self IP addresses to allow TCP connections for port `179`, the well-known port for BGP4.

The first step in configuring dynamic routing protocols on the BIG-IP system is to enable one or more routing protocols, as well as the optional the Bidirectional Forwarding Detection (BFD) network protocol. A protocol is enabled when at least one instance of the protocol is enabled on a route domain.

Important: *The BIG-IP system does not synchronize enabled protocols at runtime during configuration synchronization in a redundant system configuration. This can adversely affect the OSPFv2 and OSPFv3 protocols. To prevent these effects, always enable the protocol on an active device. Then synchronize the configuration to a standby device.*

1. On the Main tab, click **Network** > **Route Domains**.
   The Route Domain List screen opens.
2. In the Name column, click the name of the relevant route domain.
3. For the **Dynamic Routing Protocols** setting, from the **Available** list, select a protocol name and move it to the **Enabled** list.

   You can enable any number of listed protocols for this route domain.

   Important: *When you enable BFD, the BIG-IP system starts one BFD session for the route domain, and this session supports the BGP4, IS-IS, and OSPFv2 protocols only.*
4. Click **Update**.

The system displays the list of route domains on the BIG-IP system.

After performing this task, the BIG-IP system starts an instance of the specified protocol daemon for the specified route domain, and starts the core daemons `nsm` and `imi`. If BFD is enabled, the system also starts the daemon `oamd`. Once enabled, a protocol automatically restarts whenever the BIG-IP system is restarted.

Disabling a protocol for a route domain

Perform this task to disable an instance of a routing or network protocol that is currently associated with a route domain other than route domain`0`.

Important: *The BIG-IP system does not synchronize disabled protocols at runtime during configuration synchronization in a device service clustering (redundant) configuration. This can adversely affect the OSPFv2 and OSPFv3 protocols. To prevent these effects, always disable the protocol on a standby device. Then synchronize the configuration to an active device.*

---

1. On the Main tab, click **Network** > **Route Domains**.
   The Route Domain List screen opens.
2. In the Name column, click the name of the relevant route domain.
3. For the **Dynamic Routing Protocols** setting, from the **Enabled** list, select a protocol name and move it to the **Available** list.

   You can disable any number of listed protocols for this route domain.
4. Click **Update**.
   The system displays the list of route domains on the BIG-IP system.

After disabling a dynamic routing protocol for a route domain, the BIG-IP system stops the daemon of the specified protocol, resulting in these effects:

- If the specified protocol was the only protocol enabled on the system, the system stops the common daemons `nsm` and `imi`, and possibly the `oamd` daemon. You will no longer see these daemons running on the system.
- The relevant configuration is removed from the runtime configuration, but the configuration is stored on the system until you explicitly save the running configuration.
- If restarted later, the BIG-IP system does not automatically re-enable the protocol. In this case, you must explicitly re-enable the protocol after the system restarts.

Displaying the status of enabled protocols

Perform this task to display the status of instances of any dynamic routing protocols (including the Bidirectional Forwarding Detection (BFD) protocol) that are enabled for a specific route domain.

1. Open a console window, or an SSH session using the management port, on a BIG-IP device.
2. Use your user credentials to log in to the system.
3. At the command prompt, type `tmsh`.
   This opens the `tmsh` shell.
4. Type either `run util zebos check` or `list /net route-domain` *`route_domain_ID`*
   This displays the status and process IDs of any enabled dynamic routing protocols or BFD protocol for the specified route domain.

After performing this task, you can see the status and process IDs of any enabled protocols. The following shows sample output:

```
bgpd    is running [22320]
```

About Route Health Injection

Route Health Injection (RHI) is the system process of advertising the availability of virtual addresses to other routers on the network. You can configure two aspects of RHI: route advertisement and route redistribution.

About route advertisement of virtual addresses

Route advertisement is the function that the BIG-IP® system performs when advertising a route for a virtual address to the Traffic Management Microkernel (TMM) routing table. You must configure route advertisement to ensure that the dynamic routing protocols propagate this route to other routers on the network.

When configuring route advertisement for a virtual address, you can specify the particular condition under which you want the BIG-IP system to advertise the address. The available conditions that you can choose from, and their descriptions, are:

When any relevant virtual server is available
: If the system has multiple virtual servers for that virtual address and at least one of them is available, the system advertises the route for the virtual address.

When all relevant virtual servers are available
: The system only advertises the route for the virtual address when all of the relevant virtual servers are available.

Always
: The system can advertise the route even when all relevant virtual servers are unavailable. For example, the system can advertise the route when the virtual server is disabled but the virtual address is enabled and the assigned pool is available.

After you specify the desired behavior of the system with respect to route advertisement, the `tmrouted` daemon attempts to comply. The daemon only succeeds in advertising the route for the virtual address when the relevant virtual servers, pool, and pool members collectively report their status in specific combinations.

Note: *When you configure RHI in a device group configuration, only devices with active traffic groups attempt to advertise routes to virtual addresses.*

Determination of UP state for a virtual address

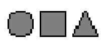
The `tmrouted` daemon within the BIG-IP® system considers a virtual IP address to be in an UP state when any one of the following conditions are true:

- The BIG-IP Configuration utility shows blue, green, or yellow status for the virtual address.
- The virtual address is a member of an active traffic group.
- The virtual address is enabled and is currently being advertised.

Conditions for route advertisement of virtual addresses

This table shows the ways that Local Traffic Manager™ (LTM®) object status affects whether the BIG-IP® system advertises a route to a virtual address. In the table, the colors represent object status shown on the Local Traffic screens within the BIG-IP Configuration utility. The table also summarizes the collective LTM object status that determines route advertisement.

Table 3: Route advertisement for virtual addresses based on LTM object status

	LTM object status				
Route advertised?	**Pool member**	**Pool**	**Virtual server**	**Virtual address**	**Status summary**
Yes					Pool members are monitored and `UP`. The virtual address is `UP`.
Yes					Pool or pool members are unmonitored. The virtual address is enabled.
Yes					Pool members are disabled. Other objects are enabled.
Yes					Virtual server is disabled. Virtual address is enabled.
Yes	N/A				The pool has no members. The virtual address is enabled.
Yes	N/A	N/A			Virtual server has no pool assigned.
No					Pool members are monitored and `DOWN`.
No					Virtual server and virtual address are disabled.
No					Virtual address is disabled. Other objects are enabled.

LTM object status indicators

The BIG-IP® Configuration utility displays various colored icons to report the status of virtual servers, virtual addresses, pools, and pool members.

Green circle 

The object is available in some capacity. The BIG-IP system services traffic destined for this object.

Blue square 

The availability of the object is unknown. Sample causes of this status are when the object is not configured for service checking, the IP address of the object is misconfigured, or the object is disconnected fromthe network.

Yellow triangle 

The object is not currently available but might become available later with no user intervention. For example, an object that has reached its configured connection limit might show yellow status but later switch to green when the number of connections falls below the configured limit.

Red diamond

The object is not available. The BIG-IP system cannot service traffic destined for this object. A sample cause of this status is when a node fails service checking because it has become unavailable. This status requires user intervention to restore the object status to green.

Black circle

A user has actively disabled an available object.

Black diamond

A user has actively disabled an unavailable object.

Gray icons

A parent object disabled the object, or the object is enabled but unavailable because of another disabled object.

Configuring route advertisement on virtual addresses

Before performing this task, verify that you have created the relevant virtual server on the BIG-IP system. Also, the virtual address that you want to advertise must have a status of Up, Unavailable, or Unknown.

Perform this task to specify the criterion that the BIG-IP system uses to advertise routes for virtual addresses. You must perform this task if you want the dynamic routing protocols to propagate this route to other routers on the network.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen displays a list of existing virtual servers.
2. On the menu bar, click **Virtual Address List**.
3. Click the name of the virtual server you want to configure.
4. For the **Advertise Route** setting, select an option:
   - **When any virtual server is available**
   - **When all virtual server(s) are available**
   - **Always**

   Note:

   If the **ICMP Echo** setting for the virtual address is set to **Selective**, then the way that the BIG-IP system manages ICMP echo responses differs depending on how you configure the **Advertise Route** setting:

   - When you select **When any virtual server is available**, the BIG-IP system sends an ICMP echo response for a request sent to the virtual address, if one or more virtual severs associated with the virtual address is in an Up or Unknown state.
   - When you select **When all virtual server(s) are available**, the BIG-IP system always sends an ICMP echo response for a request sent to the virtual address, but only when all virtual servers are available.
   - When you select **Always**, the BIG-IP system always sends an ICMP echo response for a request sent to the virtual address, regardless of the state of any virtual servers associated with the virtual address.

   ---
5. Click **Update**.

After you perform this task, properly-configured dynamic routing protocols can redistribute the advertised route to other routers on the network.

Displaying advertised routes for virtual addresses

Before you perform this task, depending on the dynamic routing protocol, you might need to configure the protocol's router definition to redistribute the kernel.

Perform this task when you want to display routes for virtual addresses that the BIG-IP system has advertised to other routers on the network.

1. Open a console window, or an SSH session using the management port, on a BIG-IP device.
2. Use your user credentials to log in to the system.
3. At the command prompt, type `tmsh`.
   This opens the `tmsh` shell.
4. Type `run /util imish -r ID`

   The variable *`ID`* is the ID of the relevant route domain. This ID must be an integer.

   This opens the IMI shell.
5. At the prompt, type `show ip route kernel`.

After performing this task, you should see the advertised routes for virtual addresses. For example, advertised routes for virtual addresses **10.1.51.80/32** and **10.2.51.81/32** appear as follows:

```
K       10.1.51.80/32 is directly connected, tmm0
K       10.1.51.81/32 is directly connected, tmm0
```

The `/32` netmask indicates that the IP addresses pertain to individual hosts, and the `tmm0` indicator shows that protocols on other routers have learned these routes from the Traffic Management Microkernel (TMM).

Delaying the withdrawal of RHI routes

Perform this task to delay the withdrawal of RHI routes when operation status changes. Delaying route withdrawal prevents short route flaps that might occur due to both the short period during failover when both devices are in a standby state, and the periodic housekeeping processes in routing protocol daemons (specifically `bgpd`).

1. Open a console window, or an SSH session using the management port, on a BIG-IP device.
2. Use your user credentials to log in to the system.
3. At the command prompt, type `tmsh`.
   This opens the `tmsh` shell.
4. Set the `bigdb` variable to the needed delay by typing this command: `modify /sys db tmrouted.rhifailoverdelay value` *`delay_in_seconds`*

The BIG-IP system now delays the withdrawal of RHI routes by the number of seconds that you specified.

Redistribution of routes for BIG-IP virtual addresses

You can explicitly configure each dynamic routing protocol to redistribute routes for advertised virtual addresses, to ensure that other routers on the network learn these routes. For purposes of redistribution, the dynamic routing protocols consider any route generated through Route Health Injection (RHI) to be a host route.

Note: For all dynamic routing protocols, you must configure route redistribution for IPv4 addresses separately from that of IPv6 addresses.

This example shows an entry in the OSPF configuration. When you add this statement to the OSPF configuration, the BIG-IP system redistributes the route for the virtual address.

```
router ospf
   redistribute kernel
```

You can optionally specify a `route-map` reference that specifies the route map to use for filtering routes prior to redistribution. For example:

```
redistribute kernel route-map external-out
```

Route maps provide an extremely flexible mechanism for fine-tuning redistribution of routes using the dynamic routing protocols.

About ICMP echo responses on the BIG-IP system

You can control whether the BIG-IP® system sends responses to Internet Control Message Protocol (ICMP) echo requests, on a per-virtual address basis.

If you disable ICMP echo responses on a virtual address, the BIG-IP system never sends an ICMP echo response for an ICMP request packet sent to the virtual address, regardless of the state of any virtual servers associated with the virtual address. If you enable ICMP echo responses on a virtual address, the BIG-IP system always sends an ICMP echo response for an ICMP request packet sent to the virtual address, regardless of the state of any virtual servers associated with the virtual address.

Alternatively, you can selectively enable ICMP echo responses. Selectively enabling ICMP echo responses causes the BIG-IP system to internally enable or disable ICMP responses for the virtual address, based on which virtual server state you choose for enabling route advertisement. This table shows that for each possible virtual server state that you can specify to enable route advertisement for a virtual address, the system controls ICMP echo responses in a unique way.

Virtual server state for route advertisement	ICMP echo response behavior
When any virtual server for that virtual address is available	The BIG-IP system sends an ICMP echo response for a request sent to the virtual address, if one or more virtual severs associated with the virtual address is in an Up or Unknown state.
When all virtual servers for that virtual address are available	The BIG-IP system always sends an ICMP echo response for a request sent to the virtual address, but only when all virtual servers are available.
When you want the system to always advertise a route to the virtual address	The BIG-IP system always sends an ICMP echo response for a request sent to the virtual address, regardless of the state of any virtual servers associated with the virtual address.

Configuring ICMP echo responses for a virtual address

You perform this task to control the way that the BIG-IP® system controls responses to ICMP echo requests sent to an individual BIG-IP virtual address. Note that the way you configure route advertisement for the virtual address can affect the way that the system controls ICMP echo responses.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen displays a list of existing virtual servers.
2. On the menu bar, click **Virtual Address List**.
3. Click the name of the virtual server you want to configure.
4. For the **ICMP Echo** setting, choose a value:
   - If you choose **Enabled**, the BIG-IP system always sends an ICMP echo response for an ICMP request packet sent to the virtual address, regardless of the state of any virtual servers associated with the virtual address
   - If you choose **Disabled**, the BIG-IP system never sends an ICMP echo response for an ICMP request packet sent to the virtual address, regardless of the state of any virtual servers associated with the virtual address.
   - If you choose **Selective** and route advertisement on a virtual address is set to **When any virtual server is available**, the BIG-IP system sends an ICMP echo response for a request sent to the virtual address, if one or more virtual severs associated with the virtual address is in an Up or Unknown state.
   - If you choose **Selective** and route advertisement on a virtual address is set to **When all virtual server(s) are available**, the BIG-IP system always sends an ICMP echo response for a request sent to the virtual address, but only when all virtual servers are available.
   - If you choose **Selective** and route advertisement on a virtual address is set to**Always**, the BIG-IP system always sends an ICMP echo response for a request sent to the virtual address, regardless of the state of any virtual servers associated with the virtual address.

   Important: *For those choices that depend on virtual server status, you must configure each relevant virtual server to notify the virtual address of its status.*

5. Click **Update**.

After performing this task, the virtual address configuration specifies the behavior that you want the BIG-IP system to exhibit when controlling responses to ICMP echo requests.

Advertisement of next-hop addresses

The BIG-IP system advertises all self IP addresses, including floating self IP addresses, to the dynamic routing protocols. The protocols store floating addresses so that the protocols can prefer a floating address as the advertised next hop. This applies only to protocols that allow explicit next-hop advertisement.

IPv6 next-hop address selection (BGP4 only)

When you are using BGP4 and IPv6 addressing, you can advertise one or two next-hop addresses for each route. The BIG-IP system selects the addresses to advertise based on several factors.

Parameter combinations for next-hop address selection

For BGP-4 only, you can choose from several combinations of configuration parameters to control the selection of next-hop IPv6 addresses.

Table 4: P = Peering, X = Configured

Link-local autoconf. (LL-A)	Link-local (LL)	Link-local floating (LL-F)	Global (G)	Global floating (G-F)	EBGP multihop	Advertised nexthop addresses
P						LL-A
X	P					LL
X	P	X				LL-F
X	P		X			G, LL
X			P			G, LL-A
X	X	X	P		X	G
X	X	X	P			LL-F
X			P	X		G-F
X	P		X	X		GF, LL
X	X		P	X		GF
X	P	X	X	X		LL-F
X	X	X	P	X		GF-F

Visibility of static routes

The dynamic routing protocols view Traffic Management Microkernel (TMM) static routes as kernel routes. (*TMM static routes* are routes that you configure using `tmsh` or the BIG-IP Configuration utility.) Because TMM static routes are viewed as kernel routes, a TMM static route has a higher precedence than a dynamic route (with an identical destination).

Management routes and addresses are not visible to the dynamic routing protocols and cannot be advertised. Routes to the networks reachable through the management interface can be learned by dynamic routing protocols if they are reachable through a VLAN, VLAN group, or tunnel.

About dynamic routing for redundant system configurations

If the BIG-IP system that you are configuring for dynamic routing is part of a redundant system configuration, you should consider these factors:

- You must configure the dynamic routing protocols on each member of the device group.
- For protocols that include the router ID attribute, you should verify that each member of the device group has a unique router ID.

- When you configure Route Health Injection (RHI), only active device group members advertise routes to virtual addresses.

Special considerations for BGP4, RIP, and IS-IS

For the BGP, RIP, RIPng, and IS-IS protocols, you no longer need to specifically configure these protocols to function in active-standby configurations. Each member of the device group automatically advertises the first floating self IP address of the same IP subnet as the next hop for all advertised routes. This applies to both IPv4 and IPv6 addresses.

Advertising a next-hop address that is always serviced by an active device guarantees that all traffic that follows routes advertised by any device in the redundant pair is forwarded based on the active LTM® configuration.

Special considerations for OSPF

For OSPF protocols, the BIG-IP system ensures that standby device group members are the least preferred next-hop routers. The system does this by automatically changing the runtime state as follows:

Table 5: Dynamic routing protocols

Protocol Name	Runtime state change
OSPFv2	The OSPF interface cost is increased on all interfaces to the maximum value (65535) when the status of the device is Standby. Also, all external type 2 Link State Advertisements (LSAs) are aged out.
OSPFv3	The OSPF interface cost is increased on all interfaces to the maximum value.

Displaying OSPF interface status

When you display OSPF interface status, you can see the effect of runtime state changes.

1. Open a console window, or an SSH session using the management port, on a BIG-IP device.
2. Use your user credentials to log in to the system.
3. At the command prompt, type `tmsh`.
4. Type this command: `run /util imish -r` *`ID`*.
5. Type `sh ip ospf interface`.

   The variable *`id`* is the ID of the relevant route domain.

Listing the OSPF link state database

When you list the contents of the OSPF link state database, you can see the effect of runtime state changes.

1. Open a console window, or an SSH session using the management port, on a BIG-IP device.
2. Use your user credentials to log in to the system.

3. At the command prompt, type `tmsh`.
4. Type this command: `run /util imish -r ID`
5. Type `sh ip ospf database external self-originate`.

   The variable *id* is the ID of the relevant route domain.

Dynamic routing on a VIPRION system

If you have a VIPRION® system, it is helpful to understand how the cluster environment affects the dynamic routing functionality.

VIPRION appearance as a single router

On a VIPRION® system, the dynamic routing system behaves as if the cluster were a single router. This means that a cluster always appears as a single router to any peer routers, regardless of the dynamic routing protocol being used.

From a management perspective, the VIPRION system is designed to appear as if you are configuring and managing the routing configuration on a single appliance. When you use the cluster IP address to configure the dynamic routing protocols, you transparently configure the primary blade in the cluster. The cluster synchronization process ensures that those configuration changes are automatically propagated to the other blades in the cluster.

Redundancy for the dynamic routing control plane

The dynamic routing system takes advantage of the redundancy provided by the cluster environment of a VIPRION® chassis, for the purpose of providing redundancy for the dynamic routing control plane. Two key aspects of dynamic routing control plane redundancy are the VIPRION cluster's appearance to the routing modules as a single router, and the operational modes of the enabled dynamic routing protocols.

Operational modes for primary and secondary blades

Enabled dynamic routing protocols run on every blade in a cluster in one of these operational modes: MASTER, STANDBY, or SLAVE.

This table shows the operational modes for primary and secondary blades, on both the active cluster and the standby cluster.

Table 6: Operational modes for dynamic routing protocols per blade type

Blade Type	Active Cluster	Standby Cluster	Notes
Primary	MASTER mode	STANDBY mode	The dynamic routing protocols: • Actively participate in dynamic routing protocol communication with peer routers.

Blade Type	Active Cluster	Standby Cluster	Notes
			• Maintain TMM and host route tables on all blades in the cluster.
Secondary	SLAVE mode	SLAVE mode	The dynamic routing protocols: • Do not transmit any dynamic routing protocol traffic. • Track communication between a module and the peer routers, or wait for transition to MASTER or STANDBY mode.

In MASTER and STANDBY modes, all routes learned by way of dynamic routing protocols on the primary blade are (in real-time) propagated to all secondary blades. The difference between MASTER and STANDBY mode is in the parameters of advertised routes, with the goal to always make the active unit the preferred next hop for all advertised routes.

The transition from SLAVE to MASTER or STANDBY mode takes advantage of standard dynamic routing protocol graceful restart functionality.

Viewing the current operational mode

Perform this task to display the current operational mode (MASTER, STANDBY, or SLAVE) of a blade.

1. Open a console window, or an SSH session using the management port, on a BIG-IP device.
2. Use your user credentials to log in to the system.
3. At the command prompt, type `tmsh`.
4. Type `run /util imish -r` *`ID`*.

   If the route domain for the protocol you want to configure is the default route domain for the current partition, you do not need to use the `-r` option to specify the route domain ID.

   This command invokes the IMI shell.
5. Type `show state`.

The BIG-IP system displays a message such as `Current operational state: MASTER`.

About graceful restart on the VIPRION system

With the *graceful restart* function, the dynamic routing protocol control plane moves from one blade to another without disruption to traffic. Graceful restart is enabled for most supported protocols and address families by default.

To operate successfully, the graceful restart function must be supported and enabled on all peer routers with which the VIPRION® system exchanges routing information. If one or more peer routers does not support graceful restart for one or more enabled dynamic routing protocols, a change in the primary blade causes full dynamic routing reconvergence, and probably traffic disruption. The traffic disruption is caused primarily by peer routers discarding routes advertised by the VIPRION system.

The BIG-IP system always preserves complete forwarding information (TMM and host route tables) on VIPRION systems during primary blade changes, regardless of support for graceful restart on peer routers.

Runtime monitoring of individual blades

The BIG-IP system automatically copies the startup configuration to all secondary blades and loads the new configuration when the running configuration is saved on the primary blade.

You can display information about the runtime state of both the primary and secondary blades. However, some information displayed on secondary blades might differ from the information on the primary blade. For troubleshooting, you should use the information displayed on the primary blade only, because only the primary blade both actively participates in dynamic routing communication and controls route tables on all blades.

Troubleshooting information for dynamic routing

Dynamic route propagation depends on a BIG-IP® system daemon named `tmrouted`. The BIG-IP system starts the `tmrouted` daemon when you enable the first dynamic routing protocol. and restarts the daemon whenever the BIG-IP system restarts.

In the rare case when you need to manage the `tmrouted` daemon due to a system issue, you can perform a number of different tasks to troubleshoot and solve the problem.

Checking the status of the tmrouted daemon

Use this procedure to verify that the `tmrouted` daemon is running. This daemon must be running for the enabled dynamic routing protocols to propagate routes.

1. Open a console window, or an SSH session using the management port, on a BIG-IP device.
2. Type your user credentials to log in to the system.
3. If the system has granted you access to the BASH shell prompt, type `tmsh`. Otherwise, skip this step.
4. Type `show /sys service tmrouted`.

The BIG-IP system displays information about `tmrouted` such as: `tmrouted run (pid 5113) 1 days`

Stopping the tmrouted daemon

Before you can stop an instance of the `tmrouted` daemon, the associated protocol instance must be enabled on the BIG-IP system. Also, the BIG-IP system `mcpd` and `tmm` daemons must be running on the system.

You perform this task to stop an instance of the `tmrouted` daemon.

Important: *Manage the* `tmrouted` *daemon using the* `tmsh` *utility only. Attempting to manage* `tmrouted` *using a Linux command or with invalid parameters might cause the daemon to fail.*

1. Open a console window, or an SSH session using the management port, on the BIG-IP device.
2. Type your user credentials to log in to the system.
3. If the system has granted you access to the BASH shell prompt, type `tmsh`. Otherwise, skip this step.
4. At the `tmsh` shell prompt, type `stop /sys service tmrouted`.

This command stops any instances of `tmrouted` that are running on the system, causing the associated protocol instance to stop propagating routes.

Restarting the tmrouted daemon

Before restarting an instance of the `tmrouted` daemon, verify that the associated protocol instance is enabled on the BIG-IP system. Also, verify that the BIG-IP system `mcpd` and `tmm` daemons are running on the system.

You perform this task to restart an instance of the `tmrouted` daemon. Whenever the BIG-IP system reboots for any reason, the BIG-IP system automatically starts an instance of `tmrouted` for each instance of an enabled dynamic routing protocol.

Important: *Manage the* `tmrouted` *daemon using the* `tmsh` *utility only. Attempting to manage* `tmrouted` *using a Linux command or with invalid parameters might cause the daemon to fail.*

1. Open a console window, or an SSH session using the management port, on the BIG-IP system.
2. Type your user credentials to log in to the system.
3. If the system has granted you access to the BASH shell prompt, type `tmsh`. Otherwise, skip this step.
4. At the `tmsh` shell prompt, type `restart /sys service tmrouted`.

This command restarts any instances of `tmrouted` that are currently stopped. The daemon also communicates with the `nsm` daemon to propagate dynamically-learned routes to other BIG-IP system processes that need to direct application traffic.

Configuring tmrouted recovery actions

Use this task to configure recovery actions when the `tmrouted` daemon restarts.

1. Open a console window, or an SSH session using the management port, on a BIG-IP device.
2. Use your user credentials to log in to the system.
3. At the command prompt, type `tmsh`.
4. Type `modify /sys daemon-ha tmrouted running [enabled|disabled]`
   - If you want to enable the `running-timeout` and `non-running-action` options, type `enabled`.
   - If you want to disable the `running-timeout` and `non-running-action` options, type `disabled`.

   Typing this command with the `enabled` option causes the active BIG-IP device to fail over to another device in the device group whenever the `tmrouted` daemon restarts.
5. Type `modify /sys daemon-ha tmrouted heartbeat [enabled|disabled]`
   - If you want to enable monitoring for the `tmrouted` heartbeat, type `enabled`.
   - If you want to disable monitoring for the `tmrouted` heartbeat, type `disabled`.

   When you type this command with the `enabled` option and the `tmrouted` heartbeat is subsequently lost, the system behaves according to the action specified by the `heartbeat-action` option.

Location and content of log files

For each dynamic routing protocol, the BIG-IP system logs messages to a file that pertains to the route domain in which the protocol is running. An example of the path name to a dynamic routing log file is `/var/log/zebos/rd1/zebos.log` file, where `rd1` is the route domain of the protocol instance.

The system logs additional messages to the files `/var/log/daemon.log` and `/var/log/ltm`. The system logs protocol daemon information for protocol-specific issues, and logs `nsm` and `imi` daemon information for core daemon-related issues.

If a core dynamic routing daemon exits, the system logs an error message similar to the following to the `/var/log/daemon.log` file:

```
Mar  5 22:43:01 mybigip LOGIN: Re-starting tmrouted
```

In addition, the BIG-IP system logs error messages similar to the following to the `/var/log/ltm` file:

```
mcpd[5157]: 01070410:5: Removed subscription with subscriber id bgpd
mcpd[5157]: 01070533:3: evWrite finished with no byte sent to connection 0xa56f9d0 (user
Unknown) - connection deleted
```

Creating a debug log file

Perform this task to create a log file for debugging. With a debug log file, you can more effectively troubleshoot any issues with a dynamic routing protocol.

1. Open a console window, or an SSH session using the management port, on a BIG-IP device.
2. Use your user credentials to log in to the system.
3. At the command prompt, type `tmsh`.
   This opens the `tmsh` shell.
4. At the `tmsh` prompt, type this command: `run /util imish -r` *`ID`*.

   If the route domain for the protocol you want to configure is the default route domain for the current partition, you do not need to specify the route domain ID.

   This command invokes the IMI shell.
5. Type the command `log file /var/log/zebos/rdn/zebos.log`.

   The variable *n* represents the relevant route domain ID. This ID must be an integer.

   The system creates a debug log file.
6. Type `write`.
   This action saves the log file.

Chapter 5

Working with Address Resolution Protocol

Address Resolution Protocol on the BIG-IP system

The BIG-IP® system is a multi-layer network device, and as such, needs to perform routing functions. To do this, the BIG-IP system must be able to find destination MAC addresses on the network, based on known IP addresses. The way that the BIG-IP system does this is by supporting *Address Resolution Protocol (ARP)*, an industry-standard Layer 3 protocol.

What are the states of ARP entries?

When you use the BIG-IP Configuration utility to view the entries in the ARP cache, you can view the state of each entry:

RESOLVED
Indicates that the system has successfully received an ARP response (a MAC address) for the requested IP address within two seconds of initiating the request. An entry in a RESOLVED state remains in the ARP cache until the timeout period has expired.

INCOMPLETE
Indicates that the system has made one or more ARP requests within the maximum number of requests allowed, but has not yet received a response.

DOWN
Indicates that the system has made the maximum number of requests allowed, and still receives no response. In this case, the system discards the packet, and sends an ICMP host unreachable message to the sender. An entry with a DOWN state remains in the ARP cache until the first of these events occurs:

- Twenty seconds elapse.

- The BIG-IP system receives either a resolution response or a gratuitous ARP from the destination host. (A *gratuitous ARP* is an ARP message that a host sends without having been prompted by an ARP request.)
- You explicitly delete the entry from the ARP cache.

About BIG-IP responses to ARP requests from firewall devices

The system does not respond to ARP requests sent from any firewall that uses a multicast IP address as its source address.

About gratuitous ARP messages

When dynamically updating the ARP cache, the BIG-IP system includes not only entries resulting from responses to ARP requests, but also entries resulting from gratuitous ARP messages.

For security reasons, the system does not fully trust gratuitous ARP entries. Consequently, if there is no existing entry in the cache for the IP address/MAC pair, and the BIG-IP system cannot verify the validity of the gratuitous ARP entry within a short period of time, the BIG-IP system deletes the entry.

Management of static ARP entries

You can manage static entries in the ARP cache in various ways.

Task summary

Adding a static ARP entry
Viewing static ARP entries
Deleting static ARP entries

Adding a static ARP entry

Perform this task to add entries to the ARP cache on the BIG-IP system. Adding a static entry for a destination server to the ARP cache saves the BIG-IP system from having to send an ARP broadcast request for that destination server. This can be useful when you want the system to forward packets to a special MAC address, such as a shared MAC address, or you want to ensure that the MAC address never changes for a given IP address.

1. On the Main tab, click **Network** > **ARP** > **Static List**.
2. Click **Create**.
3. In the **Name** field, type a name for the ARP entry.
4. In the **IP Address** field, type the IP address with which you want to associate a MAC address.
5. In the **MAC Address** field, type the MAC address that you want to associate with the specified IP address.
6. Click **Finished**.

When the BIG-IP system must forward packets to the specified IP address, the system checks the ARP cache to find the MAC address. The system then checks the VLAN's Layer 2 forwarding table to determine the appropriate outgoing interface.

Viewing static ARP entries

Perform this task to view static entries in the ARP cache.

1. On the Main tab, click **Network** > **ARP** > **Static List**.
2. View the list of static ARP entries.

You can now see all static entries in the ARP cache.

Deleting static ARP entries

Perform this task to delete a static entry from the ARP cache.

1. On the Main tab, click **Network** > **ARP** > **Static List**.
2. Locate the entry you want to delete, and to the left of the entry, select the check box.
3. Click **Delete**.
   A confirmation message appears.
4. Click **Delete**.

The deleted entry is no longer in the BIG-IP system ARP cache.

Management of dynamic ARP entries

You can manage dynamic entries in the ARP cache in various ways.

Task summary

Viewing dynamic ARP entries
Deleting dynamic ARP entries
Configuring global options for dynamic ARP entries

Viewing dynamic ARP entries

Perform this task to view dynamic entries in the ARP cache.

1. On the Main tab, click **Network** > **ARP** > **Dynamic List**.
2. View the list of dynamic ARP entries.

You can now see the list of dynamic ARP entries.

Deleting dynamic ARP entries

Perform this task to delete a dynamic entry from the ARP cache.

1. On the Main tab, click **Network** > **ARP** > **Dynamic List**.
2. Locate the entry you want to delete and, to the left of the entry, select the check box.
3. Click **Delete**.
   A confirmation message appears.
4. Click **Delete**.

The deleted entry is no longer in the BIG-IP system ARP cache.

Configuring global options for dynamic ARP entries

Perform this task to apply global options to all dynamic ARP entries.

1. On the Main tab, click **Network** > **ARP** > **Options**.
2. In the **Dynamic Timeout** field, specify a value, in seconds.

   The seconds begin to count down toward `0` for any dynamically-added entry. When the value reaches `0`, the BIG-IP system automatically deletes the entry from the cache. If the entry is actively being used as the time approaches `0`, ARP attempts to refresh the entry by sending an ARP request.
3. In the **Maximum Dynamic Entries** field, specify a maximum number of entries.

   Configure a value large enough to maintain entries for all directly-connected hosts with which the BIG-IP system must communicate. If you have more than 2000 hosts that are directly connected to the BIG-IP system, you should specify a value that exceeds the default value of `2048`.

   If the number of dynamic entries in the cache reaches the limit that you specified, you can still add static entries to the cache. This is possible because the system can remove an older dynamic entry prematurely to make space for a new static entry that you add.
4. In the **Request Retries** field, specify the number of times that the system can resend an ARP request before marking the host as unreachable.
5. For the **Reciprocal Update** setting, select or clear the check box to enable or disable the setting.

Option	Description
Enabled	Creates an entry in the ARP cache whenever the system receives who-has packets from another host on the network. When you enable this option, you slightly enhance system performance by eliminating the need for the BIG-IP system to perform an additional ARP exchange later.
Disabled	Prevents a malicious action known as ARP poisoning. *ARP poisoning* occurs when a host is intentionally altered to send an ARP response containing a false MAC address.

6. Click **Update**.

The BIG-IP system now applies these values to all dynamic ARP entries.

Global options for dynamic ARP cache entries

You can configure a set of global options for controlling dynamic ARP cache entries.

Table 7: Global options for dynamic ARP entries

Option	Description
Dynamic Timeout	Specifies the maximum number of seconds that a dynamic entry can remain in the ARP cache before the BIG-IP system automatically removes it.
Maximum Dynamic Entries	Limits the number of dynamic entries that the BIG-IP system can hold in the ARP cache at any given time. This setting has no effect on the number of static entries that the ARP cache can hold.
Request Retries	Specifies the number of times that the BIG-IP system resends an ARP request before finally marking the host as unreachable.
Reciprocal Update	Enables the BIG-IP system to store additional information, which is information that the system learns as a result of other hosts on the network sending ARP broadcast requests to the BIG-IP system.

Index

M

N

O

P

R

S

T

V