Chapter 4 – Configuring Global Load Balancing

This chapter describes the methods used in the global application switching schemes for global traffic management. This chapter includes the following sections:

- Global Traffic Management, page 315
- Proximity, page 319
- Configuring Local Report Protocol, page 326
- Redirection, page 341

Global Traffic Management

This section introduces global traffic management methods and devices and includes these topics:

- IP Traffic Management, page 315
- Global Solution Configuration Guidelines, page 318

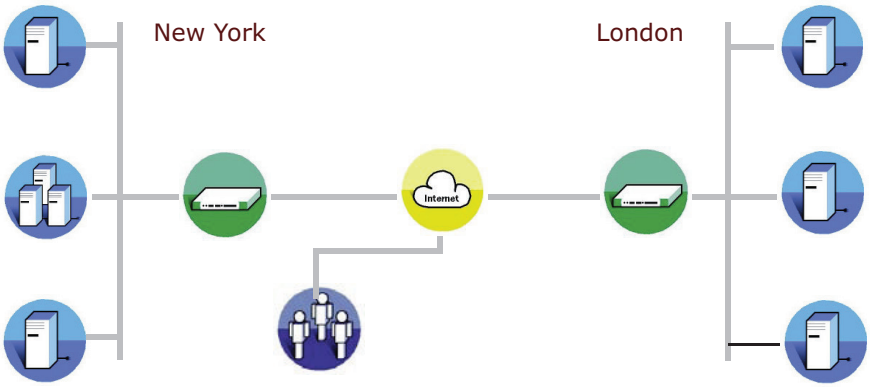
IP Traffic Management

The global IP traffic management solution is intended for companies with server sites in multiple locations. Distribution of server sites at different locations ensures high availability while maintaining multiple level redundancy. If there is damage to a single server, farm, or site, business continuity is maintained since switching from one server site to another is transparent to the users.

Globalization of business requires global server sites that ensure availability and efficiency over great geographical distances. Organizations can increase productivity through resource sharing among different branches placed in various locations.

For example, in a company, that has multiple data centers located all over the world, each data center may act as an independent business unit. Global traffic management leads to better administration and provides all employees, business partners and customers with critical resources, 24/7 availability, and optimal content delivery.

This figure illustrates an example of a global load balancing scheme.



AppDirector provides site optimization and availability over geographic distances in a way that is entirely transparent to the user. Various corporate resources are treated as a single entity. The entire corporate data resource can be represented by a single logical address that corresponds to entities at multiple physical locations.

Transactional Flow

Before considering a global solution it is essential to understand the flow of transactions over the web and the challenges posed when multiple sites are used.

- **DNS resolution.** To access Internet services the first basic step regardless of the application is host name resolution; finding the IP address for the service name, using the DNS (Domain Name System) protocol.
- **Application Transaction**. Once the client receives the IP address of the target host name the application transaction starts. A common client transaction involves multiple steps and often multiple TCP connections. During the transaction a client may perform name resolution a few times, work behind proxies or experience disconnections and resets. Therefore, the IP address of the client may not be consistent.

AppDirector Global solution follows the transaction flow to provide availability and continuity throughout the entire transaction life:

- **DNS Redirection** - AppDirector functions as authoritative domain name server for the services it load balances. When an AppDirector device receives a DNS A record query it selects the best site for the service and answers with that site VIP.
- **Application Redirection** - is required in the following cases:
  - During the transaction life, the site selected by the DNS redirection mechanism becomes unavailable or overloaded.
  - During an HTTP transaction life, a new TCP connection arrives at a different site than the one where the transaction started.
  - Protocols that do not use DNS as a first step.

AppDirector supports specific protocol redirection mechanisms (HTTP/S, RTSP and SIP) and generic mechanisms (Proxy and Radware patented Global Triangulation).

Site Selection

AppDirector selects the best site based on availability, load and network proximity.

- **Load & Availability.** Any AppDirector with site selection privileges must know the condition of every other site to make the appropriate decisions, based on the real-time dynamic load. This is achieved via Radware proprietary protocol called LRP (Load Report Protocol).
- **Proximity.** Network proximity indicates the network distance or time distance between a user and a data resource. For example, if a user is geographically closer to the New York site than to the Chicago site, yet can access the Chicago site faster when the network path to the New York site is overloaded. To measure network proximity AppDirector devices perform proximity checks to the client subnet. In addition any AppDirector with site selection privileges must know the network proximity of the client to every other site to make the appropriate decisions. This is achieved via Radware proprietary protocol called PRP (Proximity Report Protocol).

Caution: All devices in a Global configuration must run the same software version to ensure that LRP and PRP algorithms can run between the sites.

Load Balancing Scheme

AppDirector-Global balances traffic between multiple distributed sites according to the load present at each site. However the AppDirector Global device selects itself as the best site as long as the load on its local servers is lower than the Distribution Threshold configured for the specific farm and local servers are available. An AppDirector Global device starts distributing immediately in the following cases:

- All local servers are inactive, either operationally or administratively (disabled or in backup mode).
- The Distribution Threshold is set to 0.

The maximum number of users that an AppDirector device can receive from other AppDirectors is determined by the configured Farm Capacity Threshold. Once reached, the AppDirector device uses LRP to inform all other AppDirectors sending traffic to this farm that it can no longer handle directed traffic.

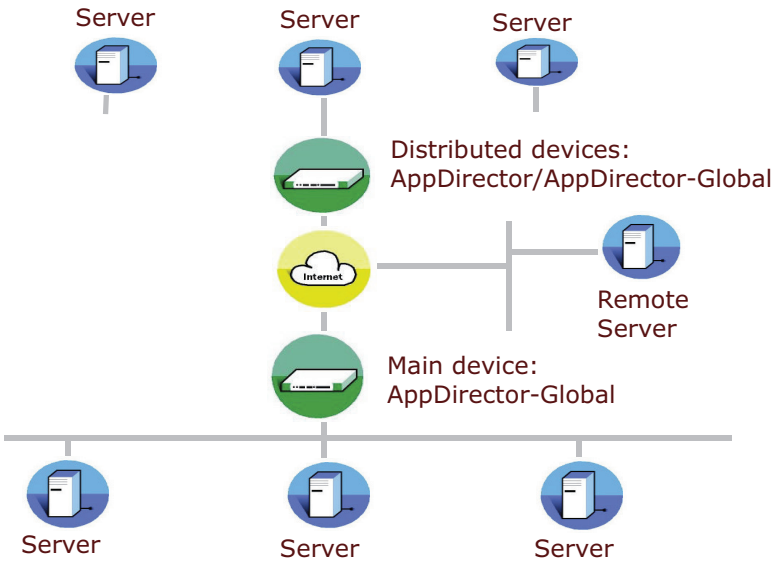
You can define the *Distribution Threshold* parameters and the *Capacity Threshold* parameters per farm within AppDirector-Global. These parameters are measured in number of *Client Table* entries for this farm. You can also define the Capacity Threshold for the farms of AppDirector devices.

Radware Devices Used in Global Solution

To implement the global traffic management solution, you need to work with an AppDirector-Global device - an AppDirector device where a Global Traffic Management license is installed. An AppDirector Global device supports both local traffic management and global traffic management - best site selection. A distributed site can be a remote server or another AppDirector site. The table below describes the differences between AppDirector and AppDirector global solutions for functionality.

Table 3: AppDirector: Regular versus Global Functionality

Functionality	AppDirector	AppDirector Global
Server types supported	Regular Local Triangulation Local Farm	Regular Local Triangulation Local Farm Remote Server AppDirector
LRP	Send	Send and Receive
PRP	Answer PRP queries	Answer and Initiate PRP queries
Redirects traffic to other locations	No	Yes
Redirection Type Supported	DNS, HTTP & RTSP for Local Redirection	DNS, HTTP & RTSP & Global Triangulation.
DNS Server Functionality (DNS Resolution)	Yes	Yes
VIP Anycast advertisement	Yes	Yes
BWM and IPS and DoS functionality	With proper license	With proper license
Cookie Persistency (Requires Special License)	Yes*	Yes*



Global Solution Configuration Guidelines

To achieve a globally distributed solution, the following steps should be performed:

On a device without site selection privileges (AppDirector or AppDirector Global):

- Configure provided services as for local load balancing (farms including local servers, Layer 4 policies, host names, etc.)
- Configure an LRP entry for each combination of a farm that is part of a distributed service and a site with distribution privileges.
- Configure the proximity checks that you want the device to perform.

On a device with site selection privileges (must be AppDirector Global) you need to:

- Define a farm for each distributed service. Each farm includes local servers plus Distributed AppDirector servers for all distributed sites (the server address is the VIP address of the service on the distributed AppDirector or a public NAT address of that VIP) and/or remote servers.

Note: If a distributed site is connected via multiple WAN links (multi-homing), it is displayed in the distributing site farm as multiple servers - one for each WAN link (each server address is the public NAT address for the distributed site VIP via a different WAN link).

- For each farm the redirection methods must be configured.
- Configure the rest of the service configuration as for local load balancing (Layer 4 policies, host names, etc.)
- Configure proximity, including static proximity if required.
- Configure DNS persistency, if required.

When the solution includes multiple sites with distribution privileges, LRP entries must be configured to report local load to the other distributing sites.

Caution: All devices in a Global configuration must run the same software version to ensure that LRP and PRP algorithms can run between the sites.

Proximity

This section provides information about the methods AppDirector uses to measure proximity to redirect traffic and includes the following topics:

- Proximity Parameters, page 319
- Proximity DNS Address, page 320
- Proximity Checks, page 321
- Proximity Report Protocol (PRP), page 322
- Static Proximity Database, page 323

AppDirector-Global maintains two proximity databases that hold information about a specific subnet of IP addresses and lists the best three servers for this range. The servers are presented in the list according to proximity considerations, the closest server appearing first. The server is either a Virtual IP address on a Distributed AppDirector device (bound to a cluster of physical servers) or a standalone remote server. If the top priority server is unavailable or loaded, AppDirector-Global sends clients to the next best server/site. If multiple application instances (farm servers) are defined on the top priority server, AppDirector-Global distributes the clients between the instances in a weighted cyclic manner. The following databases are kept:

- Static database, user-defined
- Dynamic database, built dynamically by AppDirector-Global

Proximity Parameters

Before you configure proximity checks, you need to set up the Proximity Parameters.

To configure Proximity Parameters

1. From the AppDirector menu, select **Proximity > Parameters > General.** The *Proximity Parameters* window is displayed.
2. Set the parameters.

Parameter	Description
Proximity Mode	You can determine the proximity mode as either: • No Proximity (default): No proximity is operated. • Static Proximity: AppDirector will only use redirections according to static proximity table. Dynamic auto learning mechanism is off. • Full Proximity: AppDirector will redirect according to the static redirections, and will use auto learning for subnets which are not defined as static entries.
Proximity Aging Period	Period of time, in minutes, in which a dynamic entry is kept in the database. When this time is about to expire and a new request is received from a client IP within this range, AppDirector -Global refreshes the information of that entry by re-executing the proximity checks. Values: 0 - 10080 minutes (a week) Default: 2880 minutes (2 days).
Hops Weight	Emphasis to put on the number of hops between client and farms when determining proximity. The number of hops affects the load balancing decision based on proximity considerations. Values: 1 (default) - 100

Parameter	Description
Latency Weight	Emphasis to put on the time between client and farms when determining proximity. The number effects the load balancing decision based on proximity considerations. Values: 1 (default) - 100
Load Weight	Emphasis to put on the load of remote server farm between client and farms when determining proximity. The number effects the load balancing decision based on proximity considerations. Values: 1 (default) - 100
Proximity Table Cleanup	Frequency of the Proximity Table cleanup. Default: 0
Proximity IPv6 Client Grouping Prefix Length	Sets the prefix length for grouping of ipv6 clients in the dynamic proximity table. One dynamic proximity entry will be maintained for all clients within the same subnet as defined by this prefix.

3. Click **Set.** The proximity parameters are recorded.

Proximity DNS Address

You must define local DNS servers (DNS servers located near the AppDirector) to avoid unnecessary proximity calculations for traffic coming from these servers. DNS requests received from these DNS servers are resolved using load and availability considerations only. Up to two servers can be configured.

To configure Proximity Parameters

1. From the AppDirector menu, select **Proximity > Parameters > Local DNS Servers.** The *Proximity DNS Address* window is displayed.
2. Set the parameters.

Parameter	Description
DNS Server Address	IP address of the local primary DNS server. AppDirector avoids unnecessary proximity calculations in case the DNS server is located near AppDirector. DNS requests that are received from this DNS server are replied using load considerations only.

3. Click **Set.** The proximity parameters are recorded.

Proximity Checks

AppDirector has a sophisticated mechanism to detect network proximity using a dynamic database of clients and their proximate sites. This is constantly updated by an auto-learning mechanism.

To get accurate network proximity results, the checking method should be configured to cross all obstacles en route to the client. AppDirector uses several methods to detect the number of hops and the latency from the client to each of the sites. These methods ensure that the proximity check will go through any router and firewall with maximum accuracy.

When a client approaches AppDirector, a proximity check is performed by each site and the results are communicated using the Proximity Report Protocol (PRP). Now AppDirector can redirect the client to the closest site. When another client from the same network approaches AppDirector, later, the nearest site is now known, and the client is immediately redirected there.

To configure Proximity checks

1. From the AppDirector menu, select **Proximity > Parameters > Proximity Checks.** The *Proximity Checks* window is displayed.
2. Set the parameters.

Parameter	Description
Proximity Checks	Sets whether the AppDirector device is allowed to perform proximity checks for AppDirector-Global. The proximity probes themselves are a combination of IP,TCP and application layer probes (Including TCP ACKs and ICMP Echo requests) to ensure accurate measurements. You can determine one of the following options: **Enabled (Default):** AppDirector/AppDirector-Global can serve as a Distributed server for other AppDirector-Global devices and can perform proximity checks for them. These AppDirector devices appear in the Distributed Sites definitions. **Disabled:** No proximity checks are done for other AppDirector devices.
Check Retries	If another AppDirector does not answer consecutive PRP requests, AppDirector-Global assumes that it cannot answer and ignores that particular AppDirector for this client. Default: 2
Check Interval	An interval in seconds during which AppDirector -Global sends a PRP request packet to another AppDirector. If no answer is received within this period, AppDirector-Global resends the PRP request packet. Default: 5
Application UDP Traceroute Check	Using the traceroute tool for the proximity check in the example above, AppDirector can measure latency and the number of hops to the last router. The traceroute proximity check is implemented using the UDP protocol to port 37853. Default: Enabled.
Failure Notification	Enables or disables the application aware proximity check (TCP).

3. Click **Set.** The proximity checks are recorded.

Proximity Report Protocol (PRP)

AppDirector-Global can redirect traffic to distributed locations over the Internet, if the distributed site provides better service to the client (in terms of availability, load and proximity). These distributed sites can have a standalone server or a server farm managed by another AppDirector. Information on the proximity of these distributed sites to the client enables AppDirector-Global to make such decisions. When a distributed site is equipped with an AppDirector that manages a server farm, a proprietary inter-AppDirector protocol, called PRP (Proximity Reporting Protocol), is used by AppDirector-Global to query other remote AppDirectors about their proximity to a client (can be client or DNS server). PRP is a simple UDP-based protocol that uses port UDP port 2091.

To select the closest site for a specific client, AppDirector-Global finds out how "far" this client is from all the AppDirectors in the system. To do this, AppDirector-Global calculates the number of router hops and latency between itself and the client. Then, AppDirector-Global requests all other AppDirectors to do the same and receives a report from each indicating router hops and latency between each of them and the client.

To "ask" other AppDirectors about the proximity information, AppDirector-Global uses the Proximity Report Protocol (PRP), which is a UDP-based protocol. When AppDirector-Global needs to gather proximity information about a client, it sends PRP requests to all AppDirectors in the system. Each AppDirector then calculates router hops and latency between itself and the client and reports back to AppDirector-Global, using a PRP response packet. PRP uses UDP port 2091.

Notes:

- **>>** AppDirector can also send PRP reports. Only AppDirector-Global can send PRP requests for proximity information. In addition, AppDirector-Global receives and uses the PRP responses to distribute traffic globally according to proximity considerations.
- **>>** An AppDirector which receives the request from the client and is initiating the PRP queries must always be an AppDirector-Global.
- **>>** An AppDirector device that receives PRP queries and responds can be either AppDirector or AppDirector-Global and proximity checks must be configured.

When a client request arrives from a class C network for which there is no proximity data, AppDirector gathers proximity information for the class C network of the client. To gather this data the Initiator AppDirector performs the following:

- Sends proximity checks to this new class C network.
- Sends PRP queries to all distributed servers (Distributed AppDirector type servers) in the farm servicing this client request, asking them to perform proximity checks to this class C network.
- An AppDirector that initiates PRP queries saves both its own proximity results and those received from AppDirectors receiving PRP queries in its Dynamic Proximity table for future requests from this class C network.

PRP in a Multi-Homed Environment

When an AppDirector is installed behind a NAT device that load balances inbound and outbound traffic between multiple WAN links it is operating in a multi-homed environment. Here, an AppDirector service (VIP) is accessible via a number of public IP addresses - one for each WAN link load balanced by the multi-homing device.

Static Proximity Database

The *Static Proximity Table* is user-defined. Each row in the table defines the farm that it applies to and a range of IP addresses. This range can include only one IP address or an entire IP address range. For the predefined range, you can list up to three IP addresses in order of priority. The priority defines which IP should server the client request. This is used when redirecting a client in a Global solution, either in the DNS stage or later (HTTP, RTSP, etc.).

Each server can be:

- An IP of a Distributed AppDirector type server in the relevant farm
- An IP of a Remote Server type server in the relevant farm
- A VIP of a Layer 4 policy associated with the relevant farm

Notes:

>> You need to enter the VIP associated with the farm in the static proximity so that the VIP is always associated with the farm.

>> The number of proximity subnets is configurable per farm. The default number of entries is 500, but you can select any value between 1 and 5000.

>> After setting the new values, the device must be rebooted.

>> You can configure for a known range of clients, the three nearest sites that will provide the service. Only if all the configured sites are down or overloaded will the next most convenient site be selected to provide the service.

You use the Static Proximity Table window to configure this feature. This window manages the ranges of static proximity redirections. You can configure ranges of IP addresses of clients for each farm address with a list of the three preferred sites for this range. If the range should be handled in the local site, the local site farm name should be entered.

For the predefined range, you can list up to three IP addresses in order of priority. The priority defines which IP should server the client request. This is used when redirecting client in a Global solution, either in the DNS stage or later (HTTP, RTSP, etc.). The Static Proximity window allows you to insert a new static client.

To define the static proximity parameters

1. From the AppDirector menu, select **Proximity > Static Proximity.** The *Static Proximity* window is displayed.
2. Click **Create/Update**. The *Static Proximity Create/Update* window is displayed.
3. Set the parameters.

Parameter	Description
Farm Name	Name of the farm to which the entry is applied. Default: None
From Address	IP address of the first client IP in the range.
To Address	IP address of the last client IP in the range.

Parameter	Description
Server 1	• IP of a Distributed AppDirector type server in the relevant farm OR • IP of a Remote Server type server in the relevant farm OR • VIP of a Layer 4 policy which is associated with the relevant farm • Local Service: The first priority server that this range of clients will be redirected to.
Server 2	• IP of a Distributed AppDirector type server in the relevant farm OR • IP of a Remote Server type server in the relevant farm OR • VIP of a Layer 4 policy which is associated with the relevant farm • Local Service: The second priority server that this range of clients will be redirected to.
Server 3	• IP of a Distributed AppDirector type server in the relevant farm OR • IP of a Remote Server type server in the relevant farm OR • VIP of a Layer 4 policy which is associated with the relevant farm • Local Service: The third priority server that this range of clients will be redirected to.

4. Click **Set.** The client is added to the list.

Default Redirection

Default Redirection is applicable only for remote or distributed servers. When proximity is used and a client for whom no proximity settings are defined approaches AppDirector, a server is selected for that client based on load considerations only. When no proximity information is available for a client, Default Redirection enables you to define which servers to use and in which order of priority. The table below presents the parameters you need to set for each farm in which you want to employ Default Redirection.

To configure Default Redirection

1. From the AppDirector menu, select **Distributed System > Default Redirection**. The *Default Redirection* window is displayed.
2. Click **Create**. The *Default Redirection Create* window is displayed.
3. Set the parameters.

Parameter	Description
Farm Name	Name of farm to use Default Redirection.
Priority	Order in which servers are used, where 0 indicates the highest priority. Default: 0
Server Address	Default server IP address used when no proximity information available for client approaching this farm. Values: Remote or distributed servers configured for this farm. No default.
Server Port	Zero (0) if no application port has been specified.
Admin Status	Enables (default) or disables default redirection for the farm.

4. Click **Set**. Your configuration is set.

Configuring Local Report Protocol

This section describes methods used to provide an AppDirector-Global device with load and availability information for other AppDirector sites to redirect traffic according to load and availability. The following topics are included:

- Introducing the Load Report Protocol (LRP), page 326
- Local Load Report Protocol (Local LRP), page 329

Introducing the Load Report Protocol (LRP)

AppDirector-Global can redirect traffic to distributed locations over the Internet, if the distributed site provides better service to the client (in terms of availability, load and proximity). These distributed sites can have a standalone server or a server farm managed by another AppDirector.

AppDirector-Global needs information on the load and availability of these distributed sites to be able to make such decisions. When a distributed site is equipped with an AppDirector that manages a server farm, a proprietary inter-AppDirector protocol, called LRP (Load Report Protocol), is used by distributed AppDirectors to report the availability and load of their farms to other AppDirectors. LRP is a simple UDP-based protocol using UDP port 2090.

An AppDirector running this version will be able to communicate via LRP only with AppDirector devices running version 2.14 or higher or devices running AppDirector 1.07 versions from 1.07.15 and higher.For better clarity we are going to call the device which sends load reports for its farms the "Local AppDirector" and the device that receives the reports, the "Remote AppDirector”.

A Local AppDirector can be either an AppDirector or an AppDirector-Global. A Remote AppDirector must always be an AppDirector-Global.

If more than one global site is a primary site and makes redirection decisions, an AppDirector-Global device can function as both a Local and as a Remote device - it will send load reports to other primary sites and receive load reports from all other sites.

The Local AppDirector reports the load of all of its farms that appear as distributed servers (Distributed AppDirector server type) in Remote AppDirectors.

The frequency (in seconds) with which LRP reports are sent is configurable via the Load Report Interval parameter on a Local AppDirector.

A Remote AppDirector can receive reports only for servers that appear in any of its farms as Distributed AppDirector servers.

The time (in seconds) a Remote AppDirector waits to receive load reports from the Local AppDirector is configured via the Load Report Timeout parameter in the Remote AppDirector. After this timeout, the Remote AppDirector considers the Local AppDirector as non-responding and the status of the Distributed Server that represents this Local AppDirector farm in the Remote AppDirector farm is changed to Not In Service.

The Local AppDirector must be configured with all the reports it needs to send to Remote AppDirectors. The Report Table is used for this purpose.

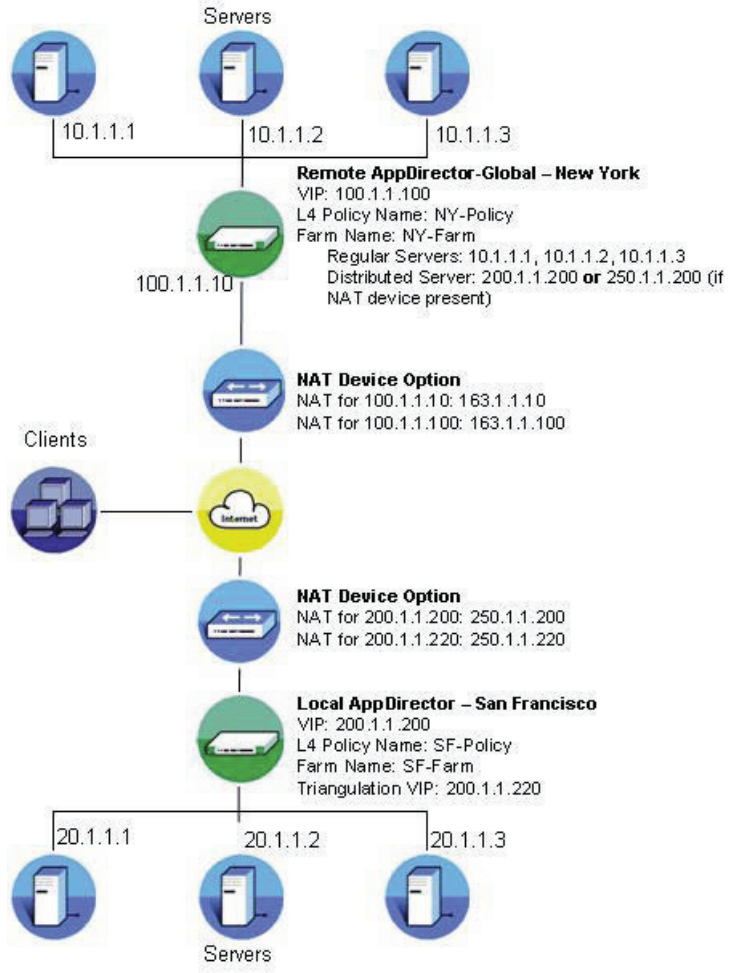
A Load Report entry must be configured for each combination of a local farm that is displayed as a distributed server in other sites and a Remote AppDirector to which its load must be reported. For example, if the Local AppDirector has 3 farms that appear as distributed servers in 2 remote AppDirectors, 6 entries are to be configured in the Report Table.

The configuration of a Report Table entry depends on the environment at the Local AppDirector site. The factors that influence it are:

- Whether the Global Triangulation method can be used to redirect traffic to this Local AppDirector farm.
- Whether any NAT device is installed in front of the Local AppDirector device and/or Remote AppDirector device.
- Whether any multi-homing NAT device (such as LinkProof) is installed in front of the Local AppDirector device

The following figure describes a configuration in which SF-Farm from San Francisco (Local AppDirector) is displayed as a distributed server in NY-Farm from New York (Remote AppDirector) meaning that the AppDirector in San Francisco (Local) must send reports to the AppDirector-Global in New York (Remote).

Figure 13: Load Reporting



To configure Load Report for Local AppDirector

1. From the AppDirector menu, select **Distributed System > Report Configuration**. The *Load Report* window is displayed.
2. Click **Create.** The *Load Report Create* window is displayed.

3. Set the parameters.

Parameter	Description
Distributed Farm Name	Name of farm in the Remote device that includes the Local AppDirector farm as a distributed server - in the example above, NY-Farm.
Distributed Server	Server address for the distributed server in the Remote AppDirector - the value of this parameter depends on whether a NAT device is installed before the Local AppDirector. • **No NAT device:** Local AppDirector Layer 4 policy VIP that is connected to the farm whose load we are reporting - in the example above, 200.1.1.200. • **NAT device:** NAT address of the Local AppDirector Layer 4 policy VIP that is connected to the farm whose load we are reporting - in the example above, 250.1.1.200.
Farm Name	Name of the local farm whose load we are reporting - in the example above, SF-Farm. This is required if the Layer 4 policy configured for this entry points to a Layer 7 policy (otherwise it s automatically set to the farm name used in the Layer 4 policy). **Note:** If no farm name configured, no report is sent.
L4 Policy Name	Layer 4 policy configured in the Local device, that points to the farm whose load we are reporting - in the example above, SF-Policy.
Triangulation VIP	Virtual IP address for global triangulation on the Local AppDirector - in the example above, 200.1.1.220. This parameter is relevant only when Global Triangulation is used.
Triangulation VIP NAT	Public IP address for Triangulation VIP, required only when Global Triangulation with NAT is used - in the example above 250.1.1.220.
Destination Address	IP interface of the Remote AppDirector device, or NAT IP for this interface, to which load and availability reports for local farms are sent - in the example above 100.1.1.10, or 163.1.1.10 if NAT device is present.
Redundant Destination Address	IP interface of a backup Remote AppDirector device, or NAT IP for this interface.
Health Monitoring ID	An identifier for this report that allows it to be associated with health monitoring checks, required only when a multi-homing device is installed in front of Local AppDirector. For more details see LRP in multi-homed environments.
Operation Status	Values: Active /Inactive. (Read Only) in Update mode.
Original VIP	Original VIP (on the Remote AppDirector) to which the client sent the request, required when Global Triangulation is used- in the example above 100.1.1.100, or 163.1.1.100 if NAT device is present. This is the address that the Local AppDirector device uses as source IP for triangulated traffic.

4. Click **Set**. Your configuration is set.
5. Adjust Load Report Interval and Load Report Timeout fields in the Global Configuration window, as described in the Global Configuration section.

Local Load Report Protocol (Local LRP)

Large companies may have large networks, including multiple sites and broad internal networks. Handling these networks might require a layer of devices, consisting of global, local, or centralized AppDirectors. This is emphasized when security devices such as firewalls are connected in front of the internal LAN. Since firewalls usually perform NAT and policy management on internal clients, connecting one AppDirector on the external side of the firewall entails complicated IP management. Connecting one AppDirector on the internal side of the firewall does not solve the problem if there are global sites to be handled; the solution is a two-tiered AppDirector setup.

Global AppDirectors handle traffic redirection between two sites, while Local AppDirectors handle local servers. From the Global AppDirector view, the farm of the Local AppDirector is a local server handling all traffic to the internal network.

The solution is to use LRP messages between the two AppDirector devices using the Local AppDirector server type. Traffic is forwarded to the Local AppDirector devices and not redirected, as in the global solution with Distributed AppDirectors. In a Global Triangulation scheme, the local AppDirector is configured as a farm of the Global AppDirector, causing the Global AppDirector to load balance traffic to the local AppDirector as if it was a regular server.

To configure Local LRP

1. On the global AppDirector, define the local AppDirector as a local AppDirector server in the relevant farm's *Remote Server Table*.
2. On the local AppDirector, define the relevant LRP reporting entries to accommodate the relationship between the global and local AppDirectors.

To configure AppDirector Local LRP

1. From the AppDirector menu, select **Global > Global Parameters.** The *Global Parameters* window is displayed.
2. Set the parameters.

Parameter	Description
Open New Entry When Source Port Different	• Enable: Each session that a client opens, is recorded in the Client Table separately. • Disable (default): All client sessions are considered a single session, providing better performance.
Select New Server When Source Port Different	• Enable: Different sessions opened by a client's application are served by different servers, according to the load balancing algorithms. • Disable (default) **Note:** This option provides a more accurate minimum-user load balancing, but may hinder some applications that depend on the same server. It also may overload AppDirector`s internal tables. This option overrides the New Entry On Source Port option.

Parameter	Description
Load Report Interval	Interval (in seconds) for sending dynamic updates of the local load to other AppDirector's devices that are served by this AppDirector. Also see Configuring Local Report Protocol, page 326.
Load Report Timeout	The time (in seconds) a Remote AppDirector waits to receive load reports from the Local AppDirector is configured via the Load Report Timeout parameter in the Remote AppDirector. Also see Configuring Local Report Protocol, page 326.

3. Click **Set.** Your configuration is set.

LRP in Multi-Homed Environments

When an AppDirector is installed behind a NAT device that load balances inbound and outbound traffic between multiple WAN links it means it operates in a multi-homed environment. The most obvious example of such a NAT device is Radware's LinkProof.

In such an environment an AppDirector service (VIP) is accessible via a number of public IP addresses - one for each WAN link load balanced by the multi-homing device.

When a Local AppDirector site is multi-homed, it affects the LRP in the following way:

1. The Local AppDirector must send multiple reports for each "local farm-Remote AppDirector" combination - one report for each WAN link managed by the multi-homing device. For example, if the site has 3 WAN links, 3 Report Table entries are configured for each "local farm-Remote AppDirector" combination.
2. The Remote AppDirector farm must also have multiple distributed servers representing the same Local AppDirector farm. In the above example, 3 distributed servers that represent Local AppDirector must be configured in the Remote AppDirector farm.
3. The Local AppDirector needs to split the actual load of the local farm between the multiple reports it sends to the same Remote AppDirector. In the above example, if the current load is 1500 users, each of the three reports Local AppDirector is sending will report 500 users.
4. The reports Local AppDirector is sending must also take into consideration the availability of the WAN links - if one of the WAN links is unavailable, Local AppDirector cannot send the relevant report for this entry. For the example in figure 2, if link A is down, Local AppDirector should not send a report for distributed server 250.1.1.200 - the Remote AppDirector will understand that this server is unavailable.

To allow Local AppDirector to check WAN links availability and send LRP messages accordingly, health checks must be configured for each WAN link on Local AppDirector. The health check of each link must be bound to the respective Load Report entry. In the Health Monitoring Bind table (page ….) the Load Report entry is identified by its Health Monitoring ID field (a string identifier must be configured in this field when the Load Report entry is defined).

To ensure that each health check configured on Local AppDirector will indeed check the required link there are two options:

- The destination IP of the health check is the internal interface of the access router for that link - in this case LinkProof routes the health check request to the proper router.
- The destination IP of the health check is an external address from the respective link provider (DNS server, etc). The address used for each health check must be different. The LinkProof must be configured to send each health check through the appropriate link by using traffic routing policies (grouping or flow policies according to LinkProof version) to ensure the correct WAN link is checked - for details please see LinkProof user guide.

Example

The following is an example of the configuration required on a Local AppDirector that is installed behind a LinkProof (Figure 2).

Step 1.

Configure the appropriate entries in the Report Table.

Parameter	Entry 1	Entry 2	Entry 3
Distributed Farm	NY-Farm	NY-Farm	NY-Farm
Distributed Server	250.1.1.200	192.1.1.200	176.1.1.200
L4 Policy	SF-Policy	SF-Policy	SF-Policy
Farm Name	SF-Farm	SF-Farm	SF-Farm
Triangulation VIP	200.1.1.220	200.1.1.220	200.1.1.220
Triangulation VIP NAT	250.1.1.220	192.1.1.220	176.1.1.220
Report Destination Address	100.1.1.10	100.1.1.10	100.1.1.10
Backup Report Destination Address			
Health Monitoring ID	SF-link1	SF-link2	SF-link3
Origin IP Address	100.1.1.100	100.1.1.100	100.1.1.100

Step 2.

Configure health checks for each WAN link. This example shows configuration of simple health checks, but a health check group can also be bound to a Load Report entry.

Health Check Name	WAN1	WAN2	WAN3
Method/Arguments	As required	As required	As required
Destination Host	250.1.1.11	192.1.1.31	176.1.1.53
Next Hop Router	200.1.1.10	200.1.1.10	200.1.1.10

Step 3.

Bind each health check with the corresponding Load Report entry.

Health Check Name	WAN1	WAN2	WAN3
Server/NHR/Report	Report-SF-link1	Report-SF-link2	Report-SF-link3

Domain Name System (DNS)

This section discusses host names and persistency and contains the following topics:

- Host Names, page 332
- DNS Server Parameters, page 336
- Static DNS Persistency, page 339
- DNS Statistics, page 341

The Domain Name System (DNS) allows Internet hosts to use names rather than IP addresses when accessing an Internet resource. DNS translates easy-to-remember names, such as www.radware.com, to IP addresses. When a user instructs a Web browser to go to a URL such as www.radware.com, DNS equates that name with an IP address allowing the user's machine to communicate through IP with the machine that hosts the website of www.radware.com.

The DNS server consists of two main components:

- **The resolver:** Component responsible for asking a DNS question about the IP address, associated with the URL representing this address.
- **The name server:** Component responsible for answering a DNS query. This is the agent present in DNS servers. When asked *"What is the IP address of www.company.com?"*, the name server answers to the best of its ability.

All basic Internet hosts and TCP/IP stacks contain the resolver, while DNS servers contain both components: resolver and name server. The resolver is necessary if there is a question that the DNS server cannot answer.

There are two kinds of DNS questions, or DNS "queries," that can be asked:

- **Iterative:** An iterative query CAN be answered with an absolute answer (IP address) or a referral.
- **Recursive:** A recursive query MUST be answered with an absolute answer (IP address).

Client resolvers cannot handle referrals and therefore, can only ask recursive questions. Server resolvers, on the other hand, can handle referrals and can ask recursive or iterative questions. Although it is more common for server resolvers to make iterative queries, they may at times make recursive queries. When a name server is asked a recursive question, it must answer the question. If it does not know the answer, it finds it. When a name server is asked an iterative question, it answers the question to the best of its ability. If a name server knows the answer, the response is the requested IP address; if a name server does not know the answer, the response is a referral answer that includes the DNS and IP address of one or more name server(s) that may know the answer.In the DNS, the IP world is divided into domains. Each domain contains hosts. For example, a host known as www.radware.com is a host in the radware.com domain. Radware.com is a sub-domain of the.com domain. Each domain in the Internet community has one or more "authoritative" name servers. An authoritative name server for a domain is responsible for all sub-domains and hosts within the domain or any of the sub-domains.

Host Names

This DNS table is used to define the relationships between hostnames and farms. You configure the Host Names Table by defining the IP of the farm which handles the URL.

You can define wildcard host names using the RegExp Host Name Table. In addition to the definition of explicit URLs in the Host Names Table. This allows you to set a single definition for many similar URLs that are hosted on the same farm. When a DNS request arrives AppDirector first looks for an exact match in the Host Name Table. If such a match is not found, AppDirector looks for a match in the RegExp Host Name Table according to the configured order of entries. When no match is found, AppDirector discards the DNS request.

To set/update the parameters of the Host Names Table

1. From the *AppDirector* menu, select **DNS > Host Names.** The *Host Names Table* window is displayed.
2. Set the parameters.

Parameter	Description
Host Name (Read-Only Parameter for Update mode)	Verifies that the farm name is bound to a Layer 7 policy.
IPv4 Entry	
Layer 4 Policy Name IPv4	Layer 4 Policy Name associated with Host Name entry provides information about the relevant farm, and Virtual IP to be used in reply. The farm is used to consider the servers' load and can optionally use Remote Servers or Distributed AppDirector in the farm for the DNS resolution process. If the Layer 4 policy selected for host name entry is connected to a Layer 7 policy, then you need to select the appropriate farm in the Farm Name field. If no farm is selected, DNS queries to this host name will not be answered and the Farm Name from the Layer 4 Policy is automatically selected. **Note:** System administrators need to configure the appropriate farm, on whose availability the decision for DNS resolution of this host name is made.
Preferred Resolve IP IPv4	Chooses how to resolve for the best available IP. • 0.0.0.0 (default): Here the host name is resolved to the best available IP without taking Operation Mode into account. • The VIP of the Layer 4 policy defined for this host name (default): Here the host name is resolved to the best available IP while taking Operation Mode into account. If a local server in Regular Operation Mode is available and the Farm distribution threshold was not reached, the device answers with the Layer 4 policy VIP, if not it selects the IP of one of the remote and distributed server's IPs according to availability, load and proximity. • The IP of a Distributed AppDirector server or a Remote server in the farm attached to the Layer 4 policy defined for this host name. While the specified server is available, the host name is resolved to its IP.
Farm Name IPv4 (Read-Only Parameter for Update mode)	Farm that you want to include in this policy, for example, Main Farm. Notes: • When a host name entry is created, if the Layer 4 Policy defined for this host name entry has the Farm Name field configured (does not include a Layer 7 policy), that farm name is displayed in the host name entry Farm Name field by default. • System administrators need to configure the appropriate farm, whose availability the decision for host name DNS resolution is made.
External NAT Address IPv4	Required when AppDirector is located behind a NAT device that NATs the VIP address for the host name. AppDirector must use External NAT Address in its DNS reply for DNS queries for host names.

Parameter	Description
IPv6 Entry	
Layer 4 Policy Name IPv6	Layer 4 Policy Name associated with Host Name entry provides information about the relevant farm, and Virtual IP to be used in reply. The farm is used to consider the servers' load and can optionally use Remote Servers or Distributed AppDirector in the farm for the DNS resolution process. If the Layer 4 policy selected for host name entry is connected to a Layer 7 policy, then you need to select the appropriate farm in the Farm Name field. If no farm is selected, DNS queries to this host name will not be answered and the Farm Name from the Layer 4 Policy is automatically selected. **Note:** System administrators need to configure the appropriate farm, on whose availability the decision for DNS resolution of this host name is made.
Preferred Resolve IP IPv6	Chooses how to resolve for the best available IP. • 0.0.0.0(default): Here the host name is resolved to the best available IP without taking Operation Mode into account. • The VIP of the Layer 4 policy defined for this host name (default): Here the host name is resolved to the best available IP while taking Operation Mode into account. If a local server in Regular Operation Mode is available and the Farm distribution threshold was not reached, the device answers with the Layer 4 policy VIP, if not it selects the IP of one of the remote and distributed server's IPs according to availability, load and proximity. • The IP of a Distributed AppDirector server or a Remote server in the farm attached to the Layer 4 policy defined for this host name. While the specified server is available, the host name is resolved to its IP.
Farm Name IPv6 (Read-Only Parameter for Update mode)	Farm that you want to include in this policy, for example, Main Farm. Notes: • When a host name entry is created, if the Layer 4 Policy defined for this host name entry has the Farm Name field configured (does not include a Layer 7 policy), that farm name is displayed in the host name entry Farm Name field by default. • System administrators need to configure the appropriate farm, whose availability the decision for host name DNS resolution is made.
External NAT Address IPv6	Required when AppDirector is located behind a NAT device that NATs the VIP address for the host name. AppDirector must use External NAT Address in its DNS reply for DNS queries for host names.

3. When creating, in the *Host Names Table* window, click **Create.** The *Host Names Table Create* window is displayed, which contains the above parameters:
4. When updating, in the *Host Names Table* window, select the desired **Host Name.** The *Host Names Table Update* window is displayed.
5. Set the parameters.
6. Click **Set.** Your configuration is set.

To update the parameters of the RegExp Host Name Table

1. From the *AppDirector* menu select **DNS > Host Names**. The *Host Names* window is displayed.
2. Select the desired **Host Name**. The *RegExp Host Names Update* window is displayed.
3. Set the parameters.

Parameter	Description
RegExp Host Name	Displays the Regexp Host Name for the Farm table. Enter in the URL type. For example any URL that begins with the string "www.abc.", followed by any text using letters and then followed by ".com" is resolved to the IP of the related farm or to the Layer 4 policy, when defined.
Index	Regular expression evaluation order. Default: 0
Layer 4 Policy Name	The Layer 4 Policy Name associated with a Host Name entry provides information about the relevant farm, and Virtual IP to be used in reply. The farm is used to consider the servers' load and can optionally use Remote Servers or Distributed AppDirector in the farm for the DNS resolution process. If the Layer 4 policy selected is connected to a Layer 7 policy, then you need to select the appropriate farm in the Farm Name field. If no farm is selected, DNS queries to this host name will not be answered and the Farm Name from the Layer 4 Policy is automatically selected. **Note:** System administrators need to configure the appropriate farm, whose availability the decision for host name DNS resolution is made.
Farm Name (Read-Only Parameter for Update mode)	Farm that you want to include in this policy, for example, Main Farm. Default: None **Notes:** • When a host name entry is created, if the Layer 4 Policy defined for this has the Farm Name field configured (without a Layer 7 policy), the farm name is displayed in the host name entry Farm Name field by default. • System administrators need to configure the appropriate farm, whose availability is decided for host name DNS resolution.
External NAT Address	Required when AppDirector is located behind a NAT device that NATs the Virtual IP address for the host name. Here, AppDirector must use the External NAT Address in DNS reply for DNS queries for host names.

Parameter	Description
Preferred Resolve IP	Chooses how to resolve for the best available IP. • 0.0.0.0 (default): The host name is resolved to the best available IP (either local VIP or VIP of a distributed site as part of local farm). AppDirector selects a Non-Backup server local or distributed (if the farm reaches the threshold) and there is no Non-Backup, it will choose a Backup server but it will treat the Backup Distributed server as a regular distributed server. • The Local VIP - The VIP of the Layer 4 policy defined for this host name. If a local server is available, the device answers with the Layer 4 policy VIP. it selects the IP of one of the remote and distributed server's IPs according to availability, load and proximity. AppDirector selects a Non-Backup server local or distributed (if the farm reaches the threshold), if there is no Non-Backup it will choose a Backup server. • Remote VIP - IP of a Distributed AppDirector server or a Remote server (not recommended, but possible) in the farm attached to the Layer 4 policy defined for this host name. If the distributed server is active it will ALWAYS be chosen regardless of the other servers.

4. Click **Set.** Your configuration is set.

Notes:

>> Host Name field (Host Name and RegExp Host Name Table) is case insensitive.

>> Total number of entries in Host Name Table and RegExp Host Name Table is determined by the value of Host Names parameter in Tuning settings.

>> A "." In a regular expression means any single character, to indicate a ".", a"\" must be used before the ".".

DNS Server Parameters

You can provide multiple a Virtual IP Interface address that can be backed up by the redundant device. If the main device fails, DNS requests are handled seamlessly and transparently by the redundant device.

To configure a Virtual IP Interface address to be backed up

1. From the *AppDirector* menu**,** select **DNS > Server.** The *DNS Server Parameters* window is displayed.
2. Set the parameters.

Parameter	Description
DNS Service	Enable or Disable (Default) the DNS service.
Two Records in DNS Reply	Enable or Disable (Default) return of two A records in the DNS response. Enable returns two A records, disable returns one.

3. Click **Set.** Your configuration is set.

DNS Persistency

AppDirector maintains persistency for consecutive DNS queries received from the same DNS client IP address. For example, if a DNS server honors the low TTL that the AppDirector assigned to DNS replies, it sends queries for the same farm every few seconds. If the client does not cache the replies, or caches them for a short period, consecutive connections for the same session may end up on two different sites.

When AppDirector receives a DNS request for a host name, for example www.a.com, it first searches for the host name in the *Host Name Table*; and if the host name is not found, AppDirector searches the *RegExp Host Name Table*. Once an entry is matched, the relevant Layer 4 Policy that serves this host name is determined.

If DNS Persistency is configured for this Layer 4 Policy, AppDirector searches the static and dynamic tables to determine the IP address used in the DNS reply. When AppDirector devices are used in multiple sites to provide a global solution, Hashing can be used to provide consistent DNS replies from different AppDirector devices to the same DNS client IP.

DNS Persistency can be configured for each farm using:

- **DNS Persistency:** When AppDirector answers a DNS request, it creates an entry in the *DNS Persistency Table*, indicating the requester's IP address and the VIP that was sent as a response. AppDirector provides the same reply to that requester as long as there is a record in the table.
- **Static DNS Persistency:** You can statically set VIPs to be used for a range of DNS IP addresses.

You can tune the *DNS Persistency* and *Static DNS Persistency* Tables.

Note: When using redundant AppDirector devices, mirroring can be used for the *DNS Persistency Table*. Similar to other mirrored tables, you can enable or disable DNS Persistency Table Mirroring, and set the Update Time and Mirroring Percentage (see Stateful Failover (Mirroring), page 158).

Working with DNS Persistency

DNS Persistency can be maintained using load balancing or Hashing. When DNS Persistency is maintained using load balancing, AppDirector dynamically selects the VIP to be used for the DNS reply based on load and proximity information. Subsequent requests from the same IP are replied to using the same VIP. You can also set Static DNS Persistency. Each range of DNS client IPs can be associated with a predefined VIP (see Static DNS Persistency, page 339).

Use of Hashing in the Global DNS Persistency Solution

In the global DNS Persistency solution, ensure that multiple AppDirector devices located at different sites use the same DNS reply for the same client IP. Then, DNS Persistency is maintained using the Hash function. To select the VIP for the DNS reply, AppDirector uses Hash on the DNS client IP address. You need to consider the following when using Hash for Global DNS Persistency:

- AppDirector devices must be configured so that the same VIPs are available for DNS replies.
- When Hash is used for DNS Persistency, server weights are taken into account. To ensure that different AppDirectors choose the same server, you must set servers with the same respective weights.
- If the VIP selected by Hashing is not available, AppDirector selects the next available VIP. This behavior is consistent among different AppDirectors.
- Entries in the DNS table should be created ONLY if the selected by the hash VIP option is NOT available at the moment and because of that a different VIP must be selected.
- When Hashing is used for DNS Persistency, load and proximity information are not used in the DNS reply process.

- When Hashing is used for DNS Persistency, Capacity Threshold and Distribution Threshold are ignored.

DNS Grouping Mask

DNS Persistency can be maintained for groups of DNS client IP addresses, both when using Load Balancing or Hash. For example, when the DNS Grouping Mask is set to 255.255.255.0, the same DNS replies are sent to IP address 1.1.1.1 and 1.1.1.66.

To define DNS persistency

1. From the AppDirector menu, select **Farms > DNS Persistency Parameters**. The *DNS Persistency Parameters Table* window is displayed.
2. Select the farm where you want to define DNS persistency. The *DNS Persistency Parameters Table Update* window is displayed.
3. Set the parameters.

Parameter	Description
Farm Name (Read Only)	Name of the farm for which DNS persistency is used.
Status	• Disabled (Default): Disables DNS Persistency. • Enabled: Enables DNS Persistency.
Mode	AppDirector selects VIP used in DNS replies for the farm using these modes: • Load Balancing (Default): Load Balancing algorithms are used, considering load and proximity information. • Hash: Hash on client IP address. This can be used when AppDirector devices are in a multiple site and global DNS persistency is required.
Static Mode	• Disabled (Default): Disables Static DNS Persistency. • Enabled: Enables Static DNS Persistency.
Aging Mode	Entries in DNS Persistency Table can be aged based on these modes: • Fixed: Entry is removed from the table after the period of time defined by the Aging Time parameter. • Inactivity (Default): Entry is removed from table if during the period defined by the Aging Time parameter no additional DNS queries are received from the same DNS client IP address.
Aging Time	Defines how many seconds an entry remains in the DNS Persistency Table. Values: 1 - 4,234,967,295 seconds (136 years). Default: 60 seconds
Grouping Suffix Length	The suffix length for a group of DNS client IP addresses. This is the IPv6 mask suffix length that defines a group of IPv6 clients for which the same farm server should be selected.
Grouping Suffix Type	The suffix type for a group of DNS client IP addresses. Values: IPv4, IPv6

4. Click **Set.** The *DNS Persistency Parameters Table Update* window closes.

Aging Mode

Entries in the *DNS Persistency Table* can be aged by the *Fixed* or *Inactivity* modes. When *Fixed* mode is used, the entry is removed from the table after the period of time defined by the *Aging Time* parameter. When *Inactivity* mode is used, the entry is removed from the table if no additional DNS queries are received from the same DNS client IP address during the period defined by the *Aging Time* parameter.

Static DNS Persistency

Static DNS Persistency operates at the farm level based on Client IP Range. This means that requests from client IP addresses in the same range to different hostnames that are mapped to the same farm are replied using the same VIP. This VIP is called Preferred VIP and can be defined using the Static DNS Persistency table.

You can also set an Alternate VIP to be used when the Preferred VIP is not available or overloaded. When the Preferred VIP is not available and the Alternate VIP is set to Any, then AppDirector dynamically selects the VIP for the DNS reply.

Note: To use static DNS, you need to first set the tuning on the device.

To define static DNS persistency

1. From the *AppDirector* menu, select **DNS > Persistency Table.** The *Static DNS Persistency Table* window is displayed.
2. Click **Create.** The *Static DNS Persistency Table Create* window is displayed.

Caution: Before adding entries to the Static DNS Persistency table, you must enable DNS Persistency and Static DNS Persistency.

3. Set the parameters.

Parameter	Description
Farm Name	Select the name to describe the AppDirector farm.
From Client IP Address	First IP address in client IP range for static DNS resolution.
To Client IP Address	Last IP address in client IP range for static DNS resolution.
Preferred VIP IPv4	IP address to be used for DNS replies for host names managed by this farm. This address is usually set to the Virtual IP address associated to Layer 4 Policy, or one of that farm's servers, of type Remote Server, Distributed Server, or Local Farm. **Note:** AppDirector uses this IP address only if the corresponding farm server is available.

Parameter	Description
Alternate VIP IPv4	Alternate VIP can be set to one of the following values: • **None (default):** AppDirector does not reply to DNS query if the preferred server is not available. • **Any:** When the preferred server is not available, AppDirector selects a VIP according to the configuration of the DNS Persistency Mode parameter using Load Balancing algorithm or using Hash. **Note:** AppDirector uses the IP address configured as Alternate VIP for DNS replies only when the corresponding farm or server is available.
Preferred VIP IPv6	IP address to be used for DNS replies for host names managed by this farm. This address is usually set to the Virtual IP address associated to Layer 4 Policy, or one of that farm's servers, of type Remote Server, Distributed Server, or Local Farm. **Note:** AppDirector uses this IP address only if the corresponding farm server is available.
Alternate VIP IPv6	Alternate VIP can be set to one of the following values: • **None (default):** AppDirector does not reply to DNS query if the preferred server is not available. • **Any:** When the preferred server is not available, AppDirector selects a VIP according to the configuration of the DNS Persistency Mode parameter using Load Balancing algorithm or using Hash. **Note:** AppDirector uses the IP address configured as Alternate VIP for DNS replies only when the corresponding farm or server is available.

4. Click **Set.** The *DNS Persistency Parameters Table Update* window closes and the new parameters appear in the *DNS Persistency Parameters Table* window.

DNS Statistics

You can generate statistics regarding DNS requests.

To view the DNS Statistics

From the AppDirector menu, select **DNS > Statistics.** The *DNS Statistics* window is displayed, which contains the following read-only statistics:

Parameter	Description
DNS Requests Last Second	Number of DNS requests in the last second. Default: 0
DNS Replies Last Second	Number of DNS replies in the last second. Default: 0
Failed DNS Requests Last Second	Number of DNS failed requests in the last second. Default: 0

Redirection

This section describes methods used to redirect traffic in the global traffic management solution and includes the following topics:

- DNS Redirection, page 342
- HTTP Redirection, page 342
- HTTP to HTTPS Protocol Redirection, page 343
- RTSP Redirection, page 345
- SIP Redirection, page 345
- Proxy Redirection, page 346
- Global Triangulation Redirection, page 346
- Setting Redirection Parameters, page 348
- Anycast Advertise, page 350

When using the global traffic management solution, several Redirection methods can help to define how service requests are redirected.

When a client sends a new request for service, AppDirector selects the best available server. If the required server is at the local site, AppDirector forwards the service request to that server. If the required server is at a remote site, AppDirector redirects the service request using one of the methods.

Multiple redirection modes can be enabled per farm. Exceptions include Global Triangulation and Proxy (Client NAT) which cannot be enabled simultaneously.

When an application-specific redirection process (HTTP, RTSP, SIP) and a Global Triangulation or Proxy mode are enabled in a farm, traffic belonging to an application for which a specific redirection mode is enabled (HTTP, RTSP or SIP) is redirected accordingly, while other applications are redirected using the Triangulation or Proxy methods.

DNS Redirection

The DNS Redirection method is based on the DNS process (see DNS Persistency, page 337). When a client sends a DNS query to find the IP address of the host name of the requested service, AppDirector operates as a DNS server. When a DNS query is made, AppDirector responds with the IP address of the best site for the client. If the local AppDirector decides that the current site is best suited for handling the client, it sends the query to its own VIP address. Otherwise, AppDirector resolves the DNS query using the IP address of a remote farm or server. Redirection is only performed during the DNS query/answer stage. Therefore, if DNS Redirection is enabled on a farm, any packet destined to the Virtual IP address is handled by the local servers of this farm.

The DNS Redirection process involves the following steps:

1. The DNS request reaches the AppDirector-Global physical IP Interface or Virtual IP Interface from a DNS server. The request is to resolve a host name to an IP address.
2. No search of the *Client Table* is made. AppDirector-Global searches the *Static Proximity Table* for a range fitting the asking DNS server. If a match is made, the top priority server from the active AND not overloaded servers is selected. AppDirector-Global resolves the name to the IP address of the chosen server, which can be a local Layer 4 VIP or a VIP configured on a remote AppDirector.
3. If there is no match in the *Static Proximity Table*, the *Dynamic Proximity Table* is searched. If there is a match, AppDirector-Global resolves the request to the Layer 4 VIP address of the highest priority site (currently active and not overloaded), accounting for the hops weight, latency weight, and the load weight variables.
4. If there is no match, AppDirector-Global resolves the request to the IP address of the least loaded site, while calculating proximity information for the querying DNS server (if proximity is enabled). Then AppDirector-Global sends PRP requests to other AppDirectors to do the same.
5. AppDirector-Global resolves the query to the IP address of the least loaded site.

Notes:

- **>>** DNS answers are made with a DNS TTL of 0,(default) to reduce Internet caching and to keep the system dynamic.
- **>>** You can set DNS TTL to a higher value and you can set different DNS TTL values for different farms.

Using AppDirector-Global, DNS Redirection works best if DNS servers from all over the Internet make queries to AppDirector-Global. If the DNS servers local to AppDirector-Global or responsible for the "super-domain" make queries to AppDirector-Global, their proximity calculations result in inaccurate data. AppDirector-Global allows you to configure up to two DNS servers with requests that are resolved to the least loaded site; no proximity calculations are made if a request comes from either of these two DNS servers.

HTTP Redirection

The HTTP redirection method is used to redirect the HTTP traffic as follows:

1. The client sends the GET request using the HTTP protocol to a VIP address of AppDirector.
2. AppDirector receives the request and selects the best server for the task.
3. If AppDirector decides that a distributed site or remote server is the most appropriate, then it issues an HTTP redirect to the user indicating that the user has been redirected. Here, AppDirector replies to the HTTP request using an HTTP code redirection code (Moved Temporarily or Moved Permanently) and redirects the client to the relevant server.

HTTP redirection can be done by IP address or by name. HTTP redirection by IP redirects the request for service to the IP of the remote server or the VIP of the Distributed AppDirector. When using HTTP redirection by name, AppDirector redirects the client to another URL. The URL used for redirection is configured using the *Redirect To URL* parameter in the server to which redirection is performed.

Note: When redirect by name is enabled and the redirect to URL field is empty, AppDirector uses server name for redirection.

HTTP to HTTPS Protocol Redirection

HTTP to HTTPS protocol redirection is based on the location field in HTTP headers where AppDirector handles SSL for backend servers, and the backend server performs a redirect using HTTP headers specifying URL with HTTP:// instead of HTTPS://. AppDirector can redirect HTTP traffic to HTTPS. Using this method, you can configure AppDirector to redirect clients to secure access to the site. You can set AppDirector to indicate to a client to access the site using HTTPS rather than HTTP when redirecting a client using HTTP redirection.

HTTP Persistency in Global Environment

In order to ensure transaction stickiness at all times in a global environment it is recommended to use the Insert cookie capability, together with HTTP Redirection, for all farms involved.

AppDirector can identify the cookie used by client belongs to a different site (was inserted at the beginning of the transaction by an AppDirector at that site) and employ HTTP redirection to transfer the new request to the original site.

To maintain site and server persistency in a global environment:

- Ensure that the automatically generated Set-Cookie method uses the same cookie key in all sites; this can be achieved either by using the same Farm Name in all sites, or by editing the automatically generate cookie key.
- If HTTP redirection is used:
  - Redirection must be by name
  - Hostnames for all sites must belong to the same domain (for example www.a.com for the main service, www.site1.a.com for site 1 and www.site2.a.com for site 2).
  - Set the domain value of the automatically generated Set-Cookie method to the service domain (in the example above a.com)
- Site and server persistency based on cookie can be supported also when application redirection is performed using proxy or global triangulation methods instead of HTTP redirection.

When AppDirector performs SSL for Back End servers the servers receives the requests in HTTP (clear), therefore, when servers perform redirect to another page/site (using HTTP headers in response with location field), it can now also use HTTP. If the clients receive this header they will initiate the new connection over HTTP instead of over HTTPS and is dropped. Therefore the AppDirector, that sends the response back to the clients must change the server's redirection location URL appearing in the HTTP header from HTTP:// to HTTPS://.

This modifies the location header of any HTTP header in server's responses from HTTP:// to HTTPS://, and the target port. The modification is performed where the host name in the client's request matches the host name in the server's response, or where matching criteria are met. Matching criteria can consist of one Regex that represent hostnames.

Notes:

- **>>** For HTTPS redirection - if the Layer 4 policy port is different than port 443, the Layer 4 policy port is appended after a colon to the URL.
- **>>** For HTTP redirection - If the Layer 4 policy port is different than port 80, the Layer 4 policy port is appended after a colon to the URL
- **>>** When Backend SSL is working, you do not need to use HTTP to HTTPS protocol redirection and you cannot configure them together.

Protocol Redirection

If a user requests the www.ab.com/base_redirect.html page, the page is redirected to www.bb.com/Redirect/Path/redirect_page.html.

If the redirect was from ab.com to ab.com/some-other-path, no regular expression

is needed since this is the same host.

In this example, the redirect was from ab.com to bb.com and this works only when the regular expression matches the host (the new host). Thus, when the regular expression is www.ab.com, it does not match bb.com (which is the real location of the page).

The redirection works as follows:

- AppDirector changes http to https for the following regular expressions:

  `"www.bb.com"`

  `"www.b*.com"`

  `"/*"`

  `"bb.com"`

  `"bb"`

  `"www"`

  `".com"`

  `"."`

  `".c"`

  `"bb.com/Redirect/"`

  `"www.bb.com/Redirect/Path/redirect_page.html"`

  `"/"`

- AppDirector does not change to https for the following regular expressions:

  `"www.ab.com"`

  `"www.bb.com/main"`

  `""`

  `" "`

  `"ab.com"`

AppDirector can redirect HTTP traffic to HTTPS. Using this method, you can configure AppDirector to redirect clients to secure access to the site. You can set AppDirector to indicate to a client to access the site using HTTPS rather than HTTP when redirecting a client using HTTP redirection.

RTSP Redirection

Real Time Streaming Protocol (RTSP) is used for audio/video streaming applications such as news broadcasts, radio stations and live shows over the Internet. Using RTSP redirection, AppDirector can redirect RTSP sessions to a remote site.

AppDirector forwards a RTSP request to a remote server or AppDirector. During the redirection, AppDirector responds with a standard RTSP redirection message causing the client sending the request to establish a new connection to a remote site to view/hear the streaming information. RTSP redirection is used for requests to TCP port 554 and is enabled by the RTSP Redirection.

The RTSP redirection process involves the following steps:

1. The client uses the RTSP protocol to send the Options, Describe, or Setup (request for a file) commands to the VIP address of AppDirector.
2. AppDirector receives the request for service and selects the best server.
3. If AppDirector decides that a distributed site, rather than a local server, is the most appropriate, then AppDirector issues a RTSP redirect to the user, redirecting the user to one of the distributed sites.

The RTSP redirection and the HTTP redirection methods work similarly as shown here maintaining persistency in a global environment.

To maintain RTSP site and server persistency in a global environment:

- Ensure that the automatically generated Set-Cookie method uses the same cookie key in all sites; this can be achieved either by using the same Farm Name in all sites, or by editing the automatically generate cookie key.
- Site and server persistency based on cookie can be supported also when application redirection is performed using proxy or global triangulation methods instead of RTSP redirection.

RTSP redirection can be done by an IP address or by name. The name is taken from the *Redirect To URL* parameter of the server. If the *Redirect To URL* parameter is not configured, the *Server Name* is used instead (see HTTP Redirection, page 342). RTSP redirection can be used globally, redirecting clients to remote farms or servers, or locally, using *Redirect to Self* (see Using Redirect to Self in Multi-Homed Environments, page 414).

Note: RTSP redirection preserves the client-server persistency of RTSP sessions when the *Client Table* mode *Select Server When Source Port is Different* is used.

SIP Redirection

The Session Initiation Protocol (SIP) is an IETF standard for a signaling protocol used for establishing sessions between two or more end users. The SIP redirection method is used to redirect SIP session invitations to external domains, such as a distributed or remote server.

The SIP redirection process involves the following steps:

1. The client sends the SIP request to an AppDirector's VIP address.
2. AppDirector receives the request and selects the best server for the task.
3. AppDirector chooses the distributed site or remote server, replies to the SIP request using the SIP code 302 (Moved Temporarily) and redirects the client to the relevant server.
4. SIP redirection can be done by an IP address or by name. SIP redirection by an IP address redirects the request for service to the IP of the remote server or the Distributed AppDirector's VIP. When using SIP redirection by name, AppDirector redirects the client to another URL.
5. SIP redirection by name is configured using the *Redirect To URL* parameter for the server.

Proxy Redirection

This method uses Client NAT to redirect traffic. AppDirector acts as a proxy at the IP level, retrieving content and then responding to the user. Before selecting this method, the Client NAT must be enabled on the device and the Client NAT range must be configured for the farm. When traffic is forwarded to another site, AppDirector replaces the original Source IP of the request with a predefined NAT IP address and dynamically selected ports. Client NAT enables AppDirector to hide the IP addresses of clients when forwarding traffic to servers in farms. Using Proxy redirection, the server does not see the original client IP address. In certain applications, the application server needs to know the client's identity and therefore AppDirector has a service entry point where the client can insert the *X-Forwarded-For Header* in the traffic it redirects.

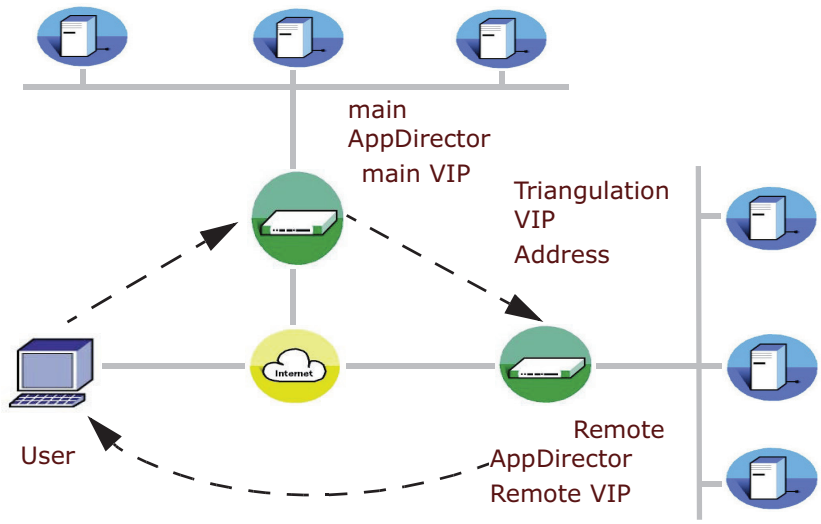
Global Triangulation Redirection

To handle the distribution of IP protocols (for example, TCP or UDP), and when other redirection methods cannot be used, use Global Triangulation redirection. The following figure illustrates an example where a user needs to receive FTP services from *ftp.company.com* and approaches the main AppDirector's VIP. If the main AppDirector has reached its Distribution Threshold and decides to send this user to a Remote AppDirector, a simple delivery of the user's packets to the Remote AppDirector's VIP will not succeed. Since the user attempted to open the FTP session with main AppDirector, if the reply comes from Remote AppDirector, the session fails. For a successful FTP session, the reply to the user must be sent using the main AppDirector VIP as the Source IP address.

Notes:

>> You cannot use Global Triangulation when any Acceleration Engine related capability (i.e. SSL, Cache, Compression, and Authentication) is used.

>> The Triangulation redirection method is not applicable in a global solution configuration that uses remote servers.



To overcome this issue, AppDirector utilizes a process called VIP Mapping, which is used at the receiving AppDirector's end (in this example, Remote AppDirector). The process consists of the mapping of three parameters:

Parameter	Description
Distributed Server	VIP address of the Remote AppDirector. The Remote VIP address is configured as a distributed server in the farm on the main AppDirector.
Triangulation VIP Address	Virtual address on the Remote AppDirector associated with the main AppDirector farm. This indicates that Remote AppDirector must send the reply to the user using the main AppDirector VIP as the source address.
Origin VIP Address	VIP address of the main AppDirector farm that directed the user to this AppDirector. The Origin VIP Address in the example is the virtual address of the main VIP.

This mappingis achieved via the LRP mechanism.

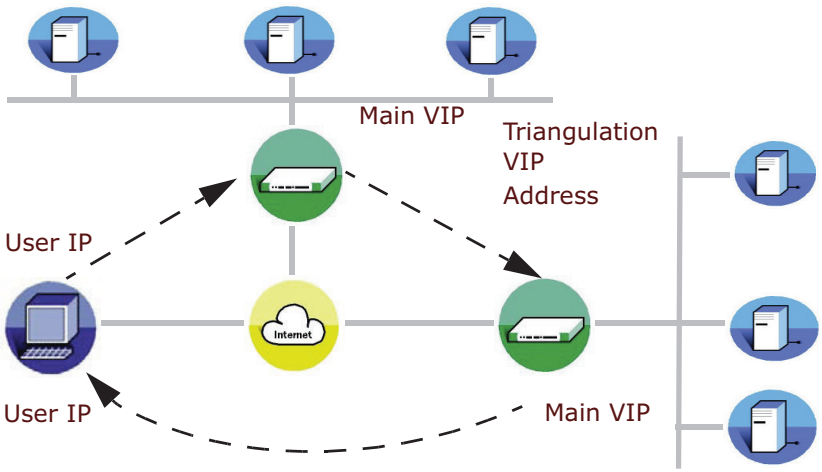
When Remote AppDirector receives packets destined for the Triangulation VIP address, it selects a server in the relevant farm and forwards the request to it. The difference is that for the reply from the server to the user, Remote AppDirector replaces the source address of the packet with the Origin VIP Address, in this example, the main VIP. This way, the user receives replies from the same address with which it tried to open the IP session.

Notes:

- **>>** Global Triangulation does not work with persistent Layer 7 switching.
- **>>** Layer 7 with Global Triangulation assumes all sites have the same configuration.
- **>>** In Layer 7 with Global Triangulation, the main device sends the request to the distributed site with no TCP handshake, just the GET.

The Global Triangulation redirection process involves the following stages:

1. User sends a new service request to VIP of main AppDirector (main VIP).
2. Main AppDirector receives the request for service and selects the best server for the task in the relevant farm.
3. Main AppDirector decides to send the user to a distributed site. The request for service is sent using the Triangulation VIP address associated with the Remote AppDirector VIP.
4. Remote AppDirector sends the packet to one of its local servers. The reply to the user is sent using the Origin VIP Address as the source address.



AppDirector uses Layer 4 policies internally to manage Triangulation VIP addresses for Global Triangulation. The Triangulation VIP addresses are defined in the *Distributed Sites* table.

AppDirector automatically creates, updates, and deletes the corresponding Layer 4 entries. These entries appear in *Layer 4 policies*. They are:

- *Layer 4 Policy Name* parameters is Auto_Triangulation.
- *Policy Defined By* parameter is System, for internally managed Layer 4 policies.

Extended LRP Security

When the Triangulation redirection method is used (see Global Triangulation Redirection, page 346), the Triangulation VIP address must be on the same subnet as the Virtual IP address, for security reasons. To ensure that the Triangulation VIP address configured for a farm is on the same subnet as that farm, the Extended LRP Security option must be enabled.

Note: The Triangulation VIP address and the Layer 4 Virtual IP address cannot be configured on different subnets when Extended LRP Security is enabled. Extended LRP Security is defined globally for each AppDirector and enabled by default. To place them on different subnets, you must disable this option.

Setting Redirection Parameters

When using the global traffic management solution, several Redirection methods are available to define how service requests are redirected to their required destinations. When a client sends a new request for service, AppDirector selects the best available server for the task. If the required server is placed at the local site, AppDirector forwards the request for service to that server. If the required server is located at the remote site, AppDirector redirects the request for service using one of the redirection methods. The redirection methods are configured at farm level. Multiple redirection methods can be configured for each farm.

To set Redirection Parameters

1. From the AppDirector menu, select **Farms > Redirection.** The *Redirection Table* window is displayed.
2. Select the farm where you want to define redirection. The *Redirection Table Update* window is displayed.
3. Set the parameters.

Parameter	Description
Farm Name	Name of farm for which redirection is configured. Read-only parameter.
DNS Redirection	Enables the DNS redirection method. Based on the DNS mechanism. When a client sends a DNS query to find an IP address of requested service host name, AppDirector operates as a DNS server.
DNS Response TLL	Number of seconds in which DNS responses are cached. Default: 0

Parameter	Description
HTTP Redirection	Enables HTTP redirection method and is used to redirect HTTP traffic. Values: • 302 Moved Temporarily • Disabled (default) • Moved Permanently • RFC Moved Temporarily
Redirect To HTTPS	Enables the HTTPS redirection method. Values: • Disabled • Enabled (default)- both HTTP and HTTPS traffic that arrives at this farm will be redirected to HTTPS, in case redirection is necessary. • HTTPS only - only HTTPS is redirected (HTTP is not redirected to HTTPS).
RTSP Redirection	Enables the RTSP redirection method where AppDirector can redirect RTSP sessions to a remote site. Default: Disabled
SIP Redirection	Enables the SIP redirection method. Default: Disabled AppDirector can redirect Session Initiation Protocol (SIP) sessions (UDP) using the redirection option (302) within the SIP protocol.
Global Triangulation	Enables the Global Triangulation method. Default: Disabled Global Triangulation uses a Radware proprietary scheme to redirect traffic to a remote AppDirector.
Proxy Redirection (Client NAT)	Enables the Proxy redirection method. Default: Disabled This method uses Client NAT to redirect traffic to another server or site. When selected, the Client NAT ranges must also be configured.
Redirect By Name	Server's Redirect To parameter is used as the host name to which redirect is performed (for HTTP/HTTPS/RTSP/SIP). Or you can use the server IP.
Farm Distribution Threshold	Local load balancing is performed between local servers until the farm reaches the Distribution Threshold limit. Then the distribution algorithm allows AppDirector to redirect users to other servers. Default: 2,500
Farm Capacity Threshold	Maximum number of connections that farm's local servers may accept. When this threshold is reached, LRP reports from the farm inform you that no more redirects can be accepted from distributed AppDirectors. Default: 5,000

Parameter	Description
Static Proximity Entries	Number of entries in Static Proximity Table. If you configure more entries than memory allows, AppDirector prints a message to the terminal and writes it to the log file. User defined Static Proximity table displays proximity subnets per farm. Each table row displays the accompanying farm and a range of IP addresses. Values: 1 - 5000 Default: 500
Application Redirection Mode	Defines whether DNS redirection is: • **only** redirection method for this farm • **primary** method: backup redirection methods can be configured where the application request (after DNS resolution) reaches a site where there are no servers available • **one** of the farm redirection methods. The following values are available: • **Disabled (Default):** Farm can only use DNS redirection. • **Enabled:** Application redirection is performed using redirection methods enabled for this farm. • **DNS Fallback Redirection:** Application redirection is used only if DNS Redirection is enabled and availability problems exist. The method used depends on methods enabled for this farm.

4. Click **Set.** The *Redirection Table Update* window closes and the new parameters appear in the *Redirection Table* window.

Anycast Advertise

Anycast is the process that allows a single IP address to be announced from multiple locations. It's a simulation of a situation where a routing domain may have multiple routes that lead to a certain destination. Such an application is useful if a service is required globally, and there are multiple service points that should be totally transparent to the user.

Once a packet has the Anycast address as a destination, the routing domain will control the flow of that packet towards one of the destinations.

A global system may use the Anycast to announce multiple service points. There are two types of Anycast service:

- **Global Anycast:** The global Anycast addresses are equally announced from multiple locations allowing users to connect to these points concurrently. The routing logic creates the load distribution between the different service points. This type of Anycast is suitable for stateless applications only, such as DNS. This type of Anycast can be used to select the DNS resolver in a global solution.
- **Local Anycast**: The local Anycast addresses are not concurrently active in more than a single primary site. Secondary sites use these addresses only to provide a transparent backup to the primary site. For the routing logic there should be a single location that serves each IP address. This type of Anycast is suitable for all protocols – as long as the primary site is active persistency is maintained.

Note: Anycast currently only supports IPv4 format IP addresses.

To create an entry in the Anycast Advertise Table

1. From the *AppDirector* menu, select **Distributed System > Anycast Advertise Table**. The *Anycast Advertise Table* window is displayed.
2. Click **Create**. The *Anycast Advertise Table Create* window is displayed.
3. Set the parameters

Parameter	Description
Layer 4 Policy Name	Name that provides AppDirector with information on the IP that must be advertised (Layer 4 Policy VIP). The value that is displayed in this field depends on the type of the Layer 4 policy, whether this is a Virtual IP Interface (VIPI) or not and if not what is the farm on whose status the advertisement depends: **Note:** The advertisement does not depend on farm status (it depends on device status up/down)
Host Route Metric	VIP Advertisement via Dynamic Routing is used to advertise a Farm's VIP or any other configured IP address via RIP or OSPF. You can configure the route metric to be used when adding the route to the routing table. This can be used when two devices are used on different sites for site redundancy. When the main site is available, the device adds the VIP to the routing table with the configured metric, then removes the entry from the routing table when the servers are unavailable. When the farms in both sites are available, clients are routed to the selected device based on the route metric. Enter the required metric value. Default: 1
Advertised VIP Farm Name	Name of farm on whose availability this advertisement is made. Layer 4 Policy VIP is only advertised if this farm is available. If Layer 4 policy Application is set to Virtual IP Interface, this field is obsolete.
External NAT Address	If AppDirector is installed behind a NAT device, the advertised IP must be not the VIP, but rather its public (NAT) address. When a NAT address is defined in the relevant entry, AppDirector responds with the External NAT IP Address.

4. Click **Set.** Your configuration is set.

Chapter 5 – Configuring Health Monitoring

This chapter describes the Health Monitoring module, a component of the Radware APSolute OS architecture and includes the following sections:

- Health Checks, page 355
- AppDirector Farm Connectivity Checks, page 370

The Health Monitoring module implemented on Radware IAS (Intelligent Application Switching) products is responsible for checking the health of network elements like servers, firewalls, and Next Hop Routers (NHRs) that are managed by the IAS. It determines which network elements are available for service, enabling the IAS to load balance traffic among the available resources.

Traffic management decisions are based mainly on the availability of the load-balanced elements and other resources on the data path. The module provides flexible configuration for health monitoring of the load-balanced elements. The module supports various predefined and user-defined checks, and enables the creation of dependencies between Health Checks of different elements.

The Health Monitoring module enables users to track the round-trip time of Health Checks. The AppDirector keeps a Response Level indicator for each check. The Response Level is the average ratio between the response time and the configured Timeout. The average is calculated based on the number of samples defined in the Response Level Samples parameters in the *Global* window. Setting the Response Level Samples to 0 disables the parameters; any other value between 1-9 defines the number of samples.

Health Monitoring Workflow

1. Set **Health monitoring >Global Parameters >Health Monitoring Status** - to Enable.
2. Set **Health monitoring >Global Parameters > Response Level Samples** - to non zero value.
3. Create **Health Monitoring > Check Table** and set **Measure Response Time** - to Enable for All servers in the farm.
4. Create **Health Monitoring > Binding Table** for All servers in the farm.

Checked Element

A Checked Element is a network element that is managed and load balanced by the AppDirector. For example, AppDirector-checked elements include the Farm Servers, NHRs, and LRP and PRP reports.

The health of a Checked Element may depend on a network element that the IAS does not load balance. For example, the health of a server managed by AppDirector may depend on the health of a database server, or other application servers, which are not load balanced by the AppDirector, or the health of a Next Hop Router managed by LinkProof may depend on the availability of the service provider.

Health Check

A Health Check defines how to test the health of any network element (not necessarily a Checked Element). A check configuration includes such parameters as the Check Method, the TCP/UDP port to which the test is sent, the time interval for the test, its timeout, the number of retries, and more. These parameters are explained in detail in the Health Checks section. A network element can be tested using one or more Health Checks.

Health Monitoring Global Parameters

The Health Monitoring module enables extensive health monitoring of all network devices, for example, server farms. The module enables the AppDirector to optimize load balancing by making sure which network elements are available and active. The Health Monitoring menu's Global Parameters window allows you to configure the Health Monitoring mode.

To configure Health Monitoring Global Parameters

1. From the Health Monitoring menu select **Global Parameters**. The *Health Monitoring Global Parameters* window is displayed.
2. Set the parameters.

Parameter	Description
Health Monitoring	Determines whether to use the Health Monitoring module or the AppDirector's connectivity checks. Default: Disabled.
Response Level Samples	Number of Response Level samples serving as basis for calculating average Response Level. AppDirector keeps a Response Level indicator for each check. This is the average ratio between response time and configured Timeout. Default: 0 **Notes:** • A value of 0 disables this parameter and no sample is taken. • Any other value between 1 - 9 defines the number of samples.
SSL Certificate Entry Name	This is used by the AppDirector when the Web server requires a client certificate during the SSL handshake. Default: Client certificate generated by the AppDirector.

3. Click **Set.** Your configuration is set.

Note: SSL certificate file and SSL private keys are not exported as part of the AppDirector configuration export.