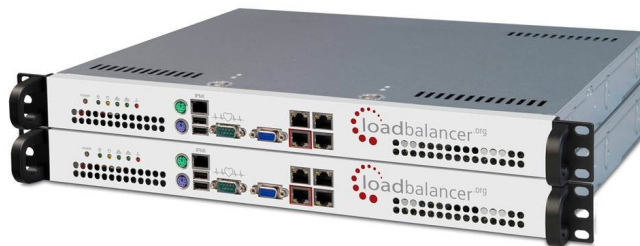




Appliance Administration Manual

v7.4



This document covers all required administration information for Loadbalancer.org appliances

Copyright © 2002 – 2013 Loadbalancer.org, Inc.

Table of Contents

Section A – Introduction.....	8
Appliance Details.....	9
Version 7.....	9
Initial Configuration.....	9
Additional Information.....	9
Deployment Guides.....	10
Section B – Load Balancing Concepts.....	11
Load Balancing Algorithms.....	12
Weighted Round Robin.....	12
Weighted Least Connection.....	12
Destination Hashing.....	12
Real Server Agent.....	12
Layer 4 vs Layer 7.....	13
Our Recommendation.....	13
Section C – Quick Start Guide.....	14
Loadbalancer.org Terminology.....	15
What is a Virtual IP Address?.....	15
What is a Floating IP Address?.....	15
What are Your Objectives?.....	16
What is the Difference Between a One-Arm and a Two-Arm Configuration?.....	17
What Load Balancing Methods are Supported?.....	17
Direct Routing (DR).....	19
Network Address Translation (NAT).....	20
Source Network Address Translation (SNAT)	21
High-Availability Configuration of two Loadbalancer.org Appliances.....	22
Clustered Pair Configuration Methods.....	22
Using the Wizard.....	22
Manual Configuration.....	22
Unpacking and Connecting the Loadbalancer.org Appliance.....	23
Initial Network Interface Configuration.....	24
Using the Network Setup Wizard.....	24
Using Linux Commands.....	25
Accessing the Web User Interface (WUI).....	26
Configuring the Loadbalancer.org Appliance Using the Web Based Wizard.....	26
Example Answers Using the Wizard for a Two-Arm NAT Configuration (Single Unit).....	27
Appliance Configuration Using the Web User Interface.....	28
Adding Virtual Servers.....	29
Adding Real Servers.....	30
Configuring the Real Servers.....	31
Configuring the Real Servers for Layer 4 NAT Mode.....	31
Configuring the Real Servers for Layer 4 DR Mode (Linux).....	31
Detecting the ARP Problem.....	31
Solving for Linux – Method 1 (using iptables).....	31
Solving for Linux – Method 2 (using arp_ignore sysctl values).....	32
Configuring the Real Servers for Layer 4 DR Mode (Windows).....	33
Configuring IIS to Respond to Both the RIP and VIP.....	33
Resolving ARP issues for Windows server 2000 (applies to DR mode only).....	34
Step 1 – Install the Microsoft loopback adapter.....	34
Step 2 – Configure the loopback adapter.....	35
Resolving ARP issues for Windows server 2003 (applies to DR mode only).....	37
Step 1 – Install the Microsoft loopback adapter.....	37
Step 2 – Configure the loopback adapter.....	38
Resolving ARP issues for Windows server 2008 (applies to DR mode only).....	40
Step 1 – Install the Microsoft loopback adapter.....	40
Step 2 – Configure the loopback adapter.....	40
Step 3 – Configure the strong / weak host behavior.....	42
Resolving ARP issues for Windows server 2012 (applies to DR mode only).....	43
Step 1 – Install the Microsoft loopback adapter.....	43
Step 2 – Configure the loopback adapter.....	43
Step 3 – Configure the strong / weak host behavior.....	45

Verifying netsh Settings for Windows 2008 & 2012.....	46
Configuring the Real Server for Layer 7 SNAT Mode.....	47
IPv6 Support.....	47
Testing the Load Balancer Configuration.....	48
Connection Error Diagnosis.....	48
Health Check Diagnosis.....	49
Appliance Log Files.....	49
Testing High-Availability for a Loadbalancer.org HA-Pair.....	50
Does Your Application Cluster Correctly Handle its Own State?.....	51
Replication Solutions for Shared Data.....	51
Solutions for Session Data.....	51
Persistence.....	51
What do You do if Your Application is Not Stateless?.....	52
Loadbalancer.org Persistence Methods.....	52
Section D – Typical Deployment Examples.....	53
Example 1 – One-Arm DR Mode (Single Appliance).....	54
Initial Network Interface Configuration.....	54
Accessing the Web User Interface (WUI).....	54
Configuring the Load Balancer (using the WUI).....	54
Configuration Overview.....	54
Network Settings.....	55
Virtual Server (VIP).....	56
Real Servers (RIP).....	56
Real Server Changes – Solve the ARP Problem.....	57
Basic Testing & Verification.....	57
Example 2 – Two-Arm NAT Mode (Clustered Pair).....	58
Initial Network Interface Configuration.....	58
Accessing the Web User Interface (WUI).....	58
Configuring the Load Balancer (using the WUI).....	58
Configuration Overview.....	58
Slave Unit – Network Settings.....	59
Master Unit – Network Settings.....	60
Master & Slave – Heartbeat Settings.....	62
Virtual Server (VIP).....	62
Real Servers (RIP).....	63
Real Server Changes – Set the Default Gateway.....	63
Verify the Slave Configuration.....	64
Restart Heartbeat.....	64
Basic Testing & Verification.....	64
Example 3 – One-Arm SNAT Mode With SSL – HAProxy & Pound (Single Unit).....	65
Initial Network Interface Configuration.....	65
Accessing the Web User Interface (WUI).....	65
Configuring the Load Balancer (using the WUI).....	65
Configuration Overview.....	65
Network Settings.....	66
Virtual Server (VIP).....	67
Real Servers (RIP).....	67
SSL Termination.....	68
Basic Testing & Verification.....	69
Section E – Detailed Configuration Information.....	70
Appliance Configuration Methods.....	71
Console Access.....	71
Console Access via a Serial Cable.....	71
Remote Configuration Methods.....	72
Full Root Access.....	74
Network Configuration.....	75
IP Addresses.....	75
Setting IP Addresses.....	75
Setting Multiple Addresses.....	76
Physical Interfaces.....	76
Configuring Bonding.....	77
Bonding Configuration Modes.....	78
Example 1: Bonding for Bandwidth.....	78
Example 2: Bonding for High-Availability (the Default Mode).....	78

Example 3: Bonding for High-Availability & Bandwidth.....	78
Configuring VLANs.....	79
Default Gateway & Static Routes.....	80
Hostname & DNS Configuration.....	81
Advanced DR Considerations.....	82
What Is the ARP Problem?.....	82
Detecting the ARP Problem.....	82
Solving for Linux – Method 1 (using iptables).....	82
Solving for Linux – Method 2 (using arp_ignore sysctl values).....	83
Solving for Solaris.....	84
Solving for Mac OS X or BSD.....	84
Resolving ARP issues for Windows server 2000 (applies to DR mode only).....	85
Step 1 – Install the Microsoft loopback adapter.....	85
Step 2 – Configure the loopback adapter.....	86
Resolving ARP issues for Windows server 2003 (applies to DR mode only).....	88
Step 1 – Install the Microsoft loopback adapter.....	88
Step 2 – Configure the loopback adapter.....	89
Resolving ARP issues for Windows server 2008 (applies to DR mode only).....	91
Step 1 – Install the Microsoft loopback adapter.....	91
Step 2 – Configure the loopback adapter.....	91
Step 3 – Configure the strong / weak host behavior.....	93
Resolving ARP issues for Windows server 2012 (applies to DR mode only).....	94
Step 1 – Install the Microsoft loopback adapter.....	94
Step 2 – Configure the loopback adapter.....	94
Step 3 – Configure the strong / weak host behavior.....	96
Verifying netsh Settings for Windows 2008 & 2012.....	97
Configuring IIS to Respond to Both the RIP and VIP.....	98
Windows 2000 / 2003.....	98
Windows 2008 / 2012.....	99
Firewall Settings.....	100
Windows 2003 SP1+.....	100
Windows 2008 R1 Firewall Settings.....	100
Windows 2008 R2 Firewall Settings.....	101
Windows 2012 Firewall Settings.....	101
Advanced NAT Considerations.....	102
Explaining the RIP & VIP in NAT Mode.....	104
Route Configuration for Windows Server with One-Arm NAT Mode.....	105
Route Configuration for Linux with One-Arm NAT Mode.....	105
Advanced Layer 7 Considerations.....	106
Load balancing Based on URL Match with HAProxy.....	107
HTTP to HTTPS Redirect using HAProxy & Pound (SSL Termination on the Load Balancer).....	108
HTTP to HTTPS Redirect using HAProxy (SSL Termination on the Real Server).....	109
HAProxy Error Codes.....	110
Configuring VIPs & RIPs via Command Line / Script.....	111
Layer 4.....	111
Layer 7.....	112
SSL Termination.....	113
Certificate on the Real Servers.....	113
Certificate on the Load Balancer.....	113
Creating a New Certificate Using a CSR.....	114
Using an Existing Certificate.....	115
Creating a PEM File.....	115
Adding an Intermediate Certificate.....	116
Windows Servers.....	117
Import a Certificate Exported from Windows Server.....	118
Converting an Encrypted Private Key to an Unencrypted Key.....	118
Limiting Ciphers.....	118
Using Tproxy.....	119
Health Monitoring.....	121
Load balancer Health (Clustered Pair).....	121
Heartbeat Communication Method.....	121
Serial Cable.....	121
Unicast (ucast).....	122
Broadcast (bcast) - Deprecated.....	122
Ping Node.....	122

Auto-Failback.....	122
Real Server Health.....	123
Layer 4.....	123
Layer 7.....	128
Simulating Health-Check Failures.....	129
Fallback Server Settings.....	130
Advanced Firewall Considerations.....	132
Firewall Marks (Layer 4).....	132
Firewall Marks – Auto Configuration.....	133
Firewall Marks – Manual Configuration.....	134
Layer 4 Persistence Considerations.....	139
Persistence State Table Replication.....	139
Server Maintenance when using Persistence.....	139
SNMP Reporting.....	140
SNMP for Layer 4 Based Services.....	140
SNMP for Layer 7 Based Services.....	140
Server Feedback Agent.....	141
Installing the Windows Agent.....	141
Using the Windows Agent.....	143
Installing the Linux/Unix Agent.....	144
Custom HTTP Agent.....	144
Changing the Local Date, Time & Time Zone.....	145
NTP Configuration.....	145
Restoring Manufacturer's Settings.....	146
From the WUI.....	146
From the Console.....	146
Force Master/Slave Take-Over In a Clustered Pair.....	146
Force the Slave to Become Active & Master Passive.....	146
Force the Master to Become Active & Slave Passive.....	146
Application Specific Settings.....	147
FTP.....	147
Layer 4 Virtual Servers for FTP.....	147
FTP Layer 4 Negotiate Health Check.....	147
FTP Recommended Persistence Settings.....	148
Layer 7 Virtual Servers for FTP.....	148
Active Mode.....	148
Passive Mode.....	149
Limiting Passive FTP Ports.....	150
For Windows 2008.....	150
For Windows 2003.....	151
For Windows 2000.....	151
For Linux.....	151
Terminal Services & RDP.....	152
Layer 4 – IP Persistence.....	152
Layer 7 – RDP Cookies.....	152
Layer 7 – Microsoft Connection Broker / Session Directory.....	153
Appliance Software Updates.....	154
Checking the Current Software Version & Revision.....	154
Online Update.....	154
Offline Update.....	155
Updating a Clustered Pair.....	155
Appliance Security.....	156
Firewall.....	156
Passwords.....	156
Appliance Lockdown Script.....	156
PCI Compliance.....	156
Adding a Slave Unit After the Master Has Been Configured.....	157
Verifying Master / Slave Replication & Testing Failover.....	158
IPMI Configuration.....	160
Section F – Disaster Recovery.....	164
Being Prepared.....	165
Backing Up to a Remote Location.....	165
Using wget to Copy the Files.....	165
Backing Up to the Load Balancer.....	166
Appliance Recovery using a USB Memory Stick.....	166

Disaster Recovery After Master Failure.....	167
Disaster Recovery After Slave Failure.....	169
Option 1 – Using the XML Backup.....	170
Option 2 – Synchronizing From the Master.....	171
Section G – Web User Interface Reference.....	172
System Overview.....	173
View Configuration.....	173
XML.....	173
Layer 4.....	173
Layer 7.....	173
SSL Termination.....	173
Heartbeat Configuration.....	173
Heartbeat Resources.....	173
Network Configuration.....	173
Routing Table.....	173
Firewall Rules.....	174
Edit Configuration.....	174
Layer 4 – Virtual Servers.....	174
Adding a Virtual Server.....	175
Modifying a Virtual Server.....	176
Layer 4 – Real Servers.....	178
Adding / Modifying a Real Server.....	178
Layer 4 – Advanced Configuration.....	180
Layer 7 – Virtual Servers.....	182
Adding a Virtual Server.....	183
Modifying a Virtual Server.....	184
Layer 7 – Real Servers.....	185
Adding / Modifying a Real Server.....	186
Layer 7 – Advanced Configuration.....	187
SSL Termination.....	190
Layer 7 Backend VIP.....	190
Layer 4 Backend VIP.....	190
Adding / modifying an SSL Virtual Server.....	191
Cipher Settings and the BEAST Attack.....	192
SSL – Advanced Configuration.....	193
Heartbeat Configuration.....	194
Floating IPs	196
Hostname & DNS.....	197
Network Interface Configuration.....	198
Routing.....	200
System Date & Time.....	201
Physical – Advanced Configuration.....	202
Setup Wizard.....	203
Upgrade Appliance.....	203
Execute Shell Command.....	203
Maintenance.....	204
Backup & Restore.....	204
Restart Services.....	206
System Control.....	207
Software Update.....	208
Fallback Page.....	209
Firewall Script.....	210
Firewall Lock Down Wizard.....	211
Initialize Graphs.....	212
Passwords.....	212
Reports.....	214
Layer 4 Status.....	214
Layer 4 Traffic Rate.....	214
Layer 4 traffic Counters.....	214
Layer 4 Current Connections.....	214
Layer 4 Current Connections (resolve hostnames).....	214
Layer 7 Status.....	215
Layer 7 Stick Table.....	215
Graphing.....	216
Reset Packet Counters.....	216

Logs.....	217
Load Balancer.....	217
Layer 4.....	217
Layer 7.....	217
SSL Termination.....	217
Heartbeat.....	217
Support.....	218
Contact Us.....	218
Technical Support Download.....	218
Section H – Appendix.....	219
Front & Rear Panel Layouts.....	220

Section A – Introduction

Appliance Details

The Loadbalancer.org appliance is an Intel based server running the GNU/Linux operating system with a custom kernel configured for load balancing. Loadbalancer.org strongly recommends that appliances should always be deployed in a fail-over (clustered pair) configuration for high availability and resilience.

The core software is based on customized versions of: Centos 6 / RHEL 6, Linux 2.6, LVS, HA-Linux, HAProxy, Pound & Ldirectord. Full root access is provided which enables complete control of all settings.

Version 7

The latest version delivers a completely revamped user interface, several new features as well as many improvements to others. A quick summary is shown below:

- Brand new Web User Interface
- Full IPv6 support
- Improved System Overview with real-time graphs showing key appliance statistics
- New master / slave role status display
- Multiple ports per VIP at layer 4
- SIP Call ID Persistence
- New Technical Support Download option that automatically bundles all logs & data to assist Loadbalancer.org staff
- Enhanced WUI data validation checks
- Code optimized throughout

Initial Configuration

Initial network configuration must be carried out on the console – either by logging in as the 'setup' user which will launch the network setup wizard, or by logging in as 'root' and running standard Linux network setup commands. When using a clustered pair – i.e. a master & slave, the network must first be configured on each appliance.

Once the network is configured, the Web User Interface (WUI) can be used for the remainder of the setup. The WUI is accessible using HTTP on port 9080 and HTTPS on port 9443. It's also possible to configure the load balancer at the console using the text based Links browser if preferred.

For a clustered pair the slave device must also be defined on the master. Once this is done, all configuration must be carried out on the master, this configuration is then automatically replicated to the slave.

Additional Information

This manual should provide you with enough information to be very productive with your Loadbalancer.org appliance. However, if there are aspects of the appliance that have not been covered, or you have any questions, then please contact our support team at : support@loadbalancer.org.

Deployment Guides

Deployment guides have also been written that focus on load balancing specific applications. Links to these are included on the Solutions page of our website : <http://www.loadbalancer.org/solutions.php>

At the time of writing, the following deployment / quick-reference guides are available:

- [Load Balancing Microsoft IIS Web Servers](#)
- [Load Balancing Microsoft Terminal Services](#)
- [Load Balancing Microsoft Exchange 2010](#)
- [Load Balancing Microsoft Sharepoint 2010](#)
- [Load Balancing VMware View](#)
- [Load Balancing Microsoft OCS 2007 R2](#)
- [Load Balancing Microsoft Lync 2010](#)
- [Load Balancing Web Proxies / Filters](#)

Section B – Load Balancing Concepts

Load Balancing Algorithms

The Loadbalancer.org appliance supports several different load balancing algorithms. Each one has its advantages and disadvantages and it depends on the specific application which is the most appropriate to use. Usually the default method *Weighted Least Connection* is a good solution which works well in most situations. The following sections summarize each method supported.

Weighted Round Robin

With this method incoming requests are distributed to Real Servers proportionally to the Real Servers weight. Servers with higher weights receive new jobs first and get more jobs than servers with lower weights. Servers with equal weights get an equal distribution of new jobs. This method addresses the weakness of the simple round robin method. Weightings are relative, so it makes no difference if Real Server #1 and #2 have weightings of 50 and 100 respectively or 5 and 10 respectively.

Weighted Least Connection

This method distributes incoming requests based on the number of current connections and also the weighting of each server. Again, weightings are relative, so it makes no difference if Real Server #1 and #2 have weightings of 50 and 100 respectively or 5 and 10 respectively.

This is the default method for new VIPs.

Destination Hashing

This algorithm assign jobs to servers through looking up a statically assigned hash table by their destination IP addresses.

Real Server Agent

To compliment the methods above, Loadbalancer.org appliances also support Real Server (i.e back-end server) agents. This permits the load balancing algorithm to be dynamically modified based on each Real Servers running characteristics. For example, one Real Server could have a run-away process that is consuming excessive CPU resources. Without the agent, the load balancer would have no way of knowing this and would continue to send requests to the overloaded server based on the algorithm selected. With the agent installed on the Real Server, feedback is provided to the load balancer and the algorithm is then adjusted to reduce requests that are sent to that server. For more details on using the agent please refer to page 141.

Layer 4 vs Layer 7

A fundamental choice when setting up the load balancer is whether to configure the services at layer 4 or layer 7.

The Basics

At layer 4 the primary protocols used are TCP and UDP. These protocols are not aware of upper level protocols such as FTP, HTTP, DNS, RDP etc. Therefore the load balancer can only make load balancing decisions based on details available at layers 4 and below such as port numbers and IP addresses. At layer 7, the load balancer effectively has more information to make load balancing related decisions since more information about upper levels protocols is available.

Layer 7 load balancing uses a proxy at the application layer (HAProxy). HTTP requests are terminated on the load balancer, and the proxy generates a new request which is passed to the chosen Real Server.

Performance

Due to the increased amount of information at layer 7, performance is not as fast as at layer 4. If raw throughput is a primary concern, then layer 4 is probably the better choice.

Persistence

Persistence (aka affinity or sticky connections) is the ability to ensure that a specific client connects back to the same server within a specific time limit. It is normally required when the session state is stored locally on the web server rather than in a separate database. At Layer 4, Source IP persistence is the only option. At layer 7, additional methods are available such as HTTP cookie persistence where the load balancer sets a cookie to identify the same session and RDP cookie persistence which is used to ensure RDP Terminal Server clients are reconnected to their existing sessions.

Real Server Changes

At Layer 4, either the ARP problem (please refer to section C page 31-46 or section E page 82-101 for more details) has to be solved (required when using Layer4 DR mode) or the default gateway on the Real Servers must be set to point at the load balancer (required when using Layer 4 NAT mode). At Layer 7, the connection is fully proxied and therefore the Real Servers do not need to be changed.

Transparency

Transparency refers to the ability to see the originating IP address of the client. Connections at Layer 4 are always transparent where as at layer 7 the IP address of the load balancer is recorded as the source address unless additional configuration steps are taken (such as using Tproxy or utilizing the X-Forwarded-For headers, please see page 188 for more details).

Our Recommendation

Where possible we recommend that Layer 4 Direct Routing (DR) mode is used. This offers the best possible performance since replies go direct from the Real Servers to the client, not via the load balancer. It's also relatively simple to implement.

Ultimately, the final choice does depend on your specific requirements and infrastructure.

Section C – Quick Start Guide

(Also available as a separate download)

Loadbalancer.org Terminology

Acronym	Terminology
Load Balancer	An IP based traffic manager for clusters
VIP	The Virtual IP address that a cluster is contactable on (Virtual Server)
RIP	The Real IP address of a back-end server in the cluster (Real Server)
GW	The Default Gateway for a back-end server in the cluster
WUI	Web User Interface
Floating IP	An IP address shared by the master & slave load balancer when in a high-availability configuration (shared IP)
Layer 4	Part of the seven layer OSI model, descriptive term for a network device that can route packets based on TCP/IP header information
Layer 7	Part of the seven layer OSI model, descriptive term for a network device that can read and write the entire TCP/IP header and payload information at the application layer
DR	Direct Routing is a standard load balancing technique that distributes packets by altering only the destination MAC address of the packet
NAT	Network Address Translation – Standard load balancing technique that changes the destination of packets to and from the VIP (external subnet to internal cluster subnet)
SNAT (HAProxy)	Source Network Address Translation – Load balancer acts as a proxy for all incoming & outgoing traffic
SSL Termination (Pound)	The SSL certificate is installed on the load balancer in order to decrypt HTTPS traffic on behalf of the cluster
MASQUERADE	Descriptive term for standard firewall technique where internal servers are represented as an external public IP address. Sometimes referred to as a combination of SNAT & DNAT rules
One Arm	The load balancer has one physical network card connected to one subnet
Two Arm	The load balancer has two network interfaces connected to two subnets – this may be achieved by using two physical network cards or by assigning two addresses to one physical network card
Eth0	Usually the internal interface also known as Gb0
Eth1	Usually the external interface also known as Gb1

What is a Virtual IP Address?

Most load balancer vendors use the term virtual IP address (VIP) to describe the address that the cluster is accessed from. It is important to understand that the virtual IP (VIP) refers both to the physical IP address and also to the logical load balancer configuration. Likewise the real IP (RIP) address refers both to the real servers physical IP address and its representation in the logical load balancer configuration.

What is a Floating IP Address?

The floating IP address is shared by the master and slave load balancer when in a high-availability configuration. The network knows that the master controls the floating IP address and all traffic will be sent to this address. The logical VIP matches this address and is used to load balance the traffic to the application cluster. If the master has a hardware failure then the slave will take over the floating IP address and seamlessly handle the load balancing for the cluster. In scenarios that only have a master load balancer there can still be a floating IP address, but in this case it would remain active on the master unit only.

What are Your Objectives?

It's important to have a clear focus on your objectives and the required outcome for the successful implementation of your load balancing solution. If the objective is clear and measurable, you know when you have achieved the goal.

Load balancers have a number of flexible features and benefits for your technical infrastructure and applications. The first question to ask is:

Are you looking for increased performance, reliability, ease of maintenance or all three?

Performance	A load balancer can increase performance by allowing you to utilize several commodity servers to handle the workload of one application
Reliability	Running an application on one server gives you a single point of failure. Utilizing a load balancer moves the point of failure to the load balancer. At Loadbalancer.org we advise that you only deploy load balancers as clustered pairs to remove this single point of failure
Maintenance	Using the appliance, you can easily bring servers on and off line to perform maintenance tasks, without disrupting your users



In order to achieve all three objectives of performance, reliability & maintenance in a web based application, your application must handle persistence correctly (see page 51 for more details).

What is the Difference Between a One-Arm and a Two-Arm Configuration?

The number of 'arms' is normally a descriptive term for how many physical connections (Ethernet interfaces) are used to connect a device to a network. It's very common for a load balancer that uses a routing method (NAT) to have a two-arm configuration. Proxy based load balancers (SNAT) commonly use a one-arm configuration.

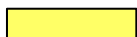
One-Arm	The load balancer has one physical network card connected to one subnet
Two-Arm	The load balancer has two network interfaces connected to two subnets – this can be achieved by using two physical network cards or by assigning two addresses to one physical network card

What Load Balancing Methods are Supported?

The Loadbalancer.org appliance is one of the most flexible load balancers on the market. The design of the appliance allows different load balancing modules to utilize the core high availability framework of the appliance. Multiple load balancing methods can be used at the same time or in combination with each other.

Layer 4	DR (Direct Routing)	Ultra-fast local server based load balancing <i>Requires handling the ARP issue on the real servers</i>	1 ARM
Layer 4	NAT (Network Address Translation)	Fast Layer 4 load balancing, the appliance becomes the default gateway for the real servers	2 ARM
Layer 4	TUN	Similar to DR but works across IP encapsulated tunnels	1 ARM
Layer 7	SSL Termination (Pound)	Usually required in order to process cookie persistence in HTTPS streams on the load balancer <i>Processor intensive</i>	1 or 2 ARM
Layer 7	SNAT (Source Network Address Translation: HAProxy)	Layer 7 allows great flexibility including full SNAT and WAN load balancing, cookie insertion and URL switching <i>Not as fast as Layer 4</i>	1 or 2 ARM

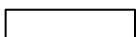
Key:



Recommended for high performance fully transparent and scalable solutions



Recommended if HTTP cookie persistence is required, also used for numerous Microsoft applications such as Terminal Services (RDP cookie persistence) and Exchange, that require SNAT mode



Only required for Direct Routing implementation across routed networks (rarely used)

Loadbalancer.org Recommendation:

Where feasible, one-arm direct routing (DR) mode is our recommended method because it's a very high performance solution with little change to your existing infrastructure.



Sometimes it's not possible to use DR mode. The two most common reasons being: if the application cannot bind to the RIP & VIP at the same time; or if the host operating system cannot be modified to handle the ARP problem (see page 31-46 for more details).

A second option is Network Address Translation (NAT) mode. This is a fairly high performance solution but it requires the implementation of a two-arm infrastructure with an internal and external subnet to carry out the translation (the same way a firewall works). Network engineers with experience of hardware load balancers will have often used this method.

The third option is Source Network Address Translation (SNAT) mode using HAProxy. If your application requires that the load balancer handles cookie insertion, RDP cookies, Session Broker integration or SSL termination then this option is appropriate. This can be deployed in one-arm or two-arm mode and does not require any changes to the application servers. HAProxy is a high-performance solution that operates as a full proxy, but due to this it cannot perform as fast as the layer 4 solutions.



If your application doesn't maintain its own state information then you may need to use cookie insertion to maintain server persistence (affinity).

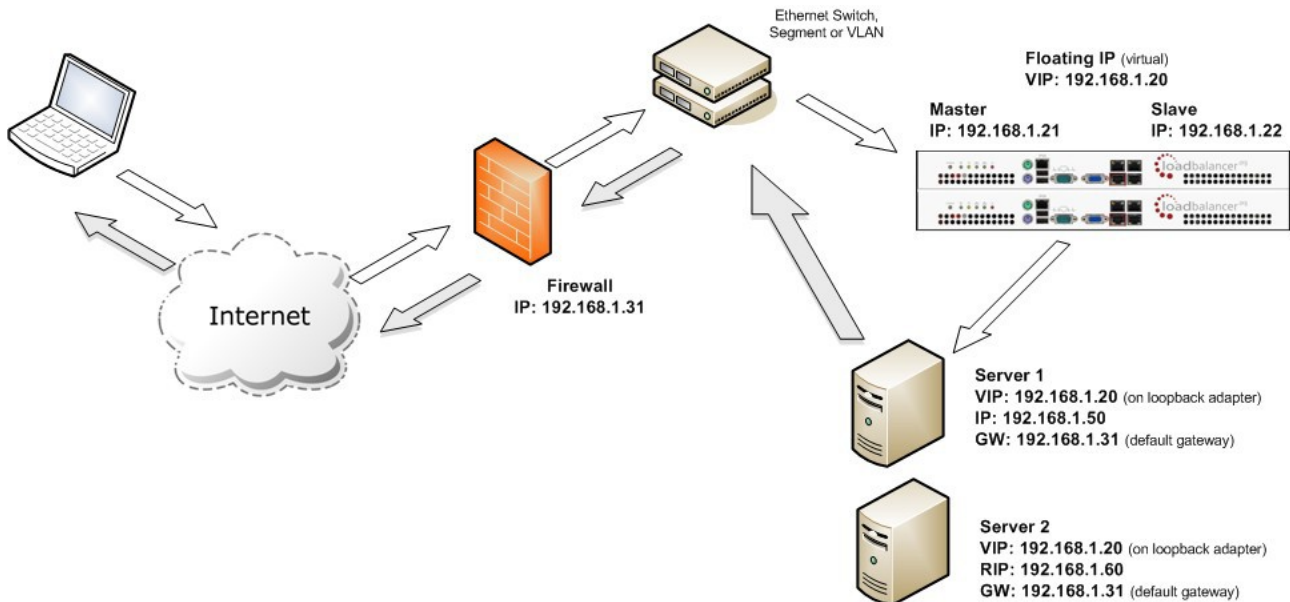
The following sections describe these configurations in more details.



IMPORTANT NOTE – If you are using Microsoft Windows real servers (i.e. back-end servers) make sure that Windows NLB (Network Load Balancing) is completely disabled to ensure that this does not interfere with the operation of the load balancer.

Direct Routing (DR)

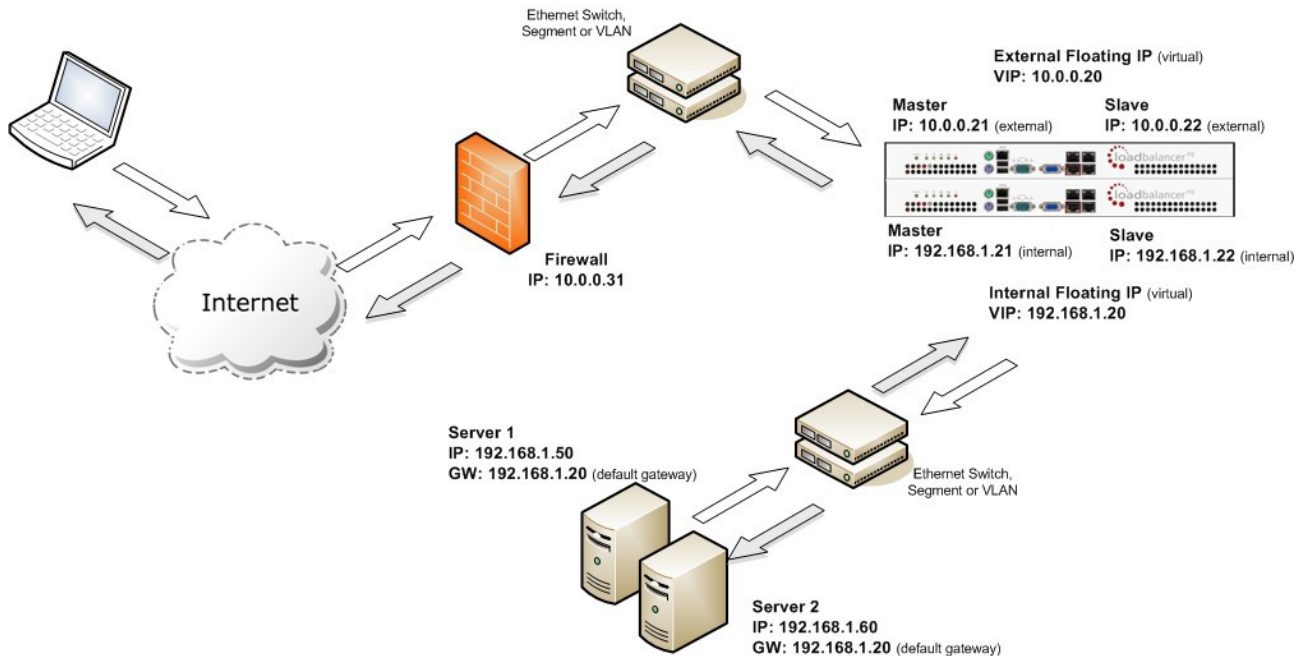
The one-arm direct routing (DR) mode is the recommended mode because it's a very high performance solution with little change to your existing infrastructure. *NB. Foundry networks call this Direct Server Return and F5 call it N-Path.*



- Direct routing works by changing the destination MAC address of the incoming packet on the fly which is very fast
- However, this means that when the packet reaches the real server it expects it to own the VIP. This means you need to make sure the real server responds to both its own IP and the VIP, but does not respond to ARP requests for the VIP. Please refer to page 31-46 for more details on resolving the ARP problem
- On average, DR mode is 8 times quicker than NAT for HTTP, 50 times quicker for Terminal Services and much, much faster for streaming media or FTP
- Load balanced services can be configured directly on the interface (normally eth0) with no additional IP address. However, when using a clustered pair, all load balanced virtual services **must** be configured on a floating IP to enable failover & failback between master & slave
- The virtual server and real servers must be in the same switch fabric / logical network. They can be on different subnets, provided there are no router hops between them. If multiple subnets are used, an IP address in each subnet must be defined on the load balancer
- Port translation is not possible in DR mode i.e. having a different RIP port than the VIP port
- DR mode is transparent, i.e. the real server will see the source IP address of the client
- Administration of the load balancer is via any active IP address (on port 9080)

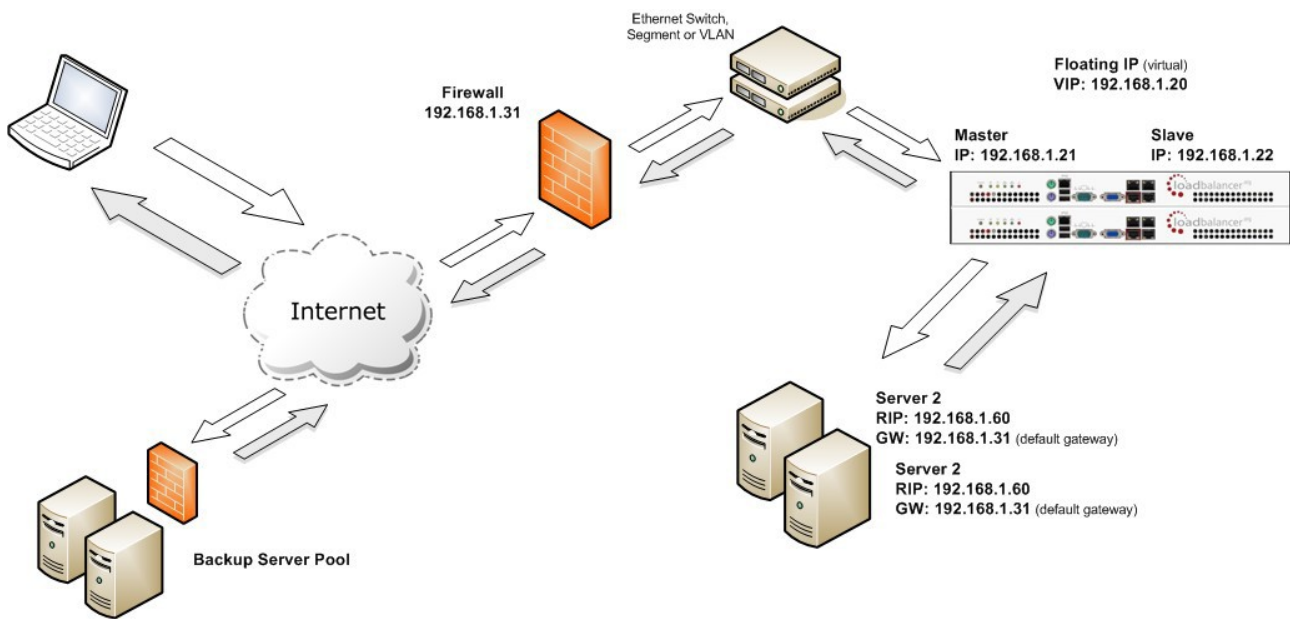
Network Address Translation (NAT)

Sometimes it's not possible to use DR mode. The two most common reasons being: if the application cannot bind to RIP & VIP at the same time; or if the host operating system cannot be modified to handle the ARP problem. The second choice is Network Address Translation (NAT) mode. This is also a fairly high performance solution but it requires the implementation of a two arm infrastructure with an internal and external subnet to carry out the translation (the same way a firewall works).



- In two-arm NAT mode the load balancer translates all requests from the external virtual server to the internal real servers
- It's a good idea to use *eth1* as your external network and *eth0* as internal, otherwise you will need to change the *autonat* setting in *Edit Configuration > Layer 4 – Advanced Configuration*
- The real servers must have their default gateway configured to point at the load balancer. When master & slave units are used, a floating IP must be used to enable failover
- Real servers are automatically given access to the Internet through the load balancer (via *autonat*)
- Load balanced services can be configured directly on the interface (normally *eth0*) with no additional IP address. However, when using a clustered pair all load balanced virtual services must be configured on a floating IP to enable failover & failback between master & slave
- Normally the virtual server and real servers should be located on different subnets within the same logical network (i.e. no router hops) and the load balancer should have an IP address in each subnet. *NB. It is possible to have real and virtual servers in the same subnet – please refer to the Advanced NAT topic in Section F of the administration manual. NB. It is possible to have the real servers located on routed subnets, but this would require a customized routing configuration on the real servers and is not recommended*
- If you want real servers to be accessible on their own IP address for non-load balanced services, e.g. SMTP or RDP, you will need to setup individual SNAT and DNAT firewall script rules for each real server. Please refer to the Advanced NAT Considerations section in the administration manual for more details
- You can also configure the load balancers in one-arm NAT mode, but in order to make the servers accessible from the local network you need to change the routing configuration on the real servers. Please refer to the Advanced NAT Considerations section in the administration manual for more details.
- NAT mode is transparent , i.e. the real server will see the source IP address of the client
- Administration of the load balancer is via any active IP address (on port 9080)

Source Network Address Translation (SNAT)



If your application requires that the load balancer handles cookie insertion then you need to use the SNAT configuration. This mode is also used with numerous Microsoft applications such as Terminal Services (using RDP cookies or Integrated with Connection Broker) and Exchange that require SNAT mode.

This mode also has the advantage of a one arm configuration and does not require any changes to the application servers. However, as the load balancer is acting as a full proxy it doesn't have the same raw throughput as the routing based methods.

The network diagram for the Layer 7 HAProxy SNAT mode is very similar to the Direct Routing example except that no re-configuration of the real servers is required. The load balancer proxies the application traffic to the servers so that the source of all traffic becomes the load balancer.

- As with other modes a single unit does not require a Floating IP, although it is recommended to make adding a slave unit easier
- SNAT is a full proxy and therefore load balanced real servers do not need to be changed in any way
- Because SNAT is a full proxy any server in the cluster can be on any accessible subnet including across the Internet or WAN
- SNAT is not transparent by default, i.e. the real servers will not see the source IP address of the client, they will see the load balancers IP address. If required, this can be solved by either enabling TPROXY on the loadbalancer, or leveraging the X-forwarded-For header. See the administration manual for more details.



For detailed configuration examples, please refer to section D in the administration manual.

High-Availability Configuration of two Loadbalancer.org Appliances

Loadbalancer.org's recommended configuration is to use a clustered pair of load balancers to provide a highly available and resilient load balancing solution. In this configuration, the pair communicates via a heartbeat to determine if the master node is active. Should the master node suffer a failure, the slave will immediately take over any resources hosted on the shared floating IP addresses.



Using a single load balancer introduces a single point of failure for your infrastructure so it is strongly recommended to use two appliances in a clustered pair.

Clustered Pair Configuration Methods

There are two ways to configure a clustered pair; either by using the wizard or configuring the units manually.

Using the Wizard

If the wizard is used, the slave is configured first and then the master. This ensures that both units can first communicate via the selected link (via a serial cable – the default, or over the network), and also that settings that are configured on the master and correctly replicated to the slave.



For more details on using the wizard and an example, please refer to pages 26-27.

Manual Configuration

If the master is configured first without using the wizard and the slave is added later, the following points should be considered:

- The hostname of the unit to be used as the slave must be set to 'lbslave' using *Edit Configuration > Hostname & DNS* in the WUI
- The IP address of the slave must be defined on the master using *Edit Configuration > Hostname & DNS* in the WUI
- The **Force full slave sync** option in *Edit Configuration > Hostname & DNS* should be checked prior to clicking **Update** – this will ensure that all configured services are correctly replicated over to the slave unit
- Once the IP address is set and synchronization has occurred, its important to restart heartbeat on both units to ensure heartbeat starts cleanly. This can be done using *Maintenance > Restart Services* in the WUI



For more details please refer to the configuration examples in section D of the administration manual.

Unpacking and Connecting the Loadbalancer.org Appliance

- Remove all packaging
- Rack mount the appliance if required
- The power supply is an auto sensing unit (100v to 240v)
- Connect the power lead from the power socket to the mains or UPS
- Connect your network cable from your switch or hub to the internal network port (*eth0*)
- If using a two-armed configuration connect a second network cable to the external port (*eth1*)

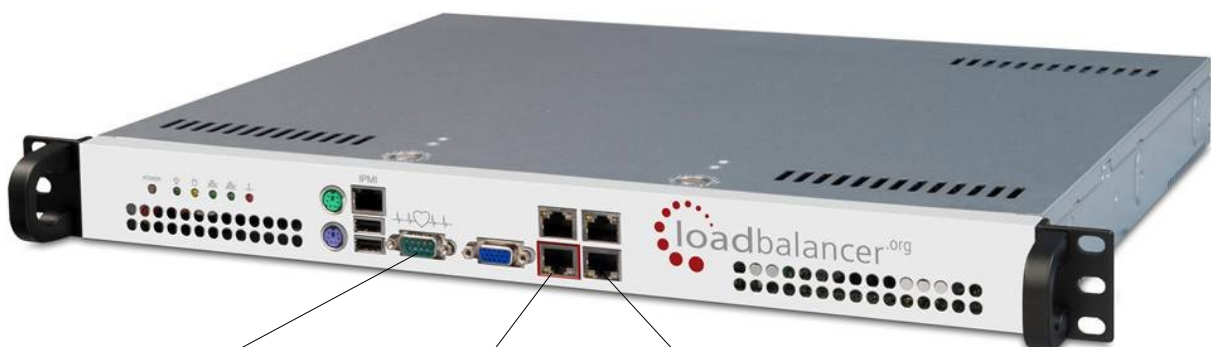


If two load balancers (recommended) are being used, connect a null modem cable (one cable is supplied with each appliance) between the two serial ports, then configure the slave first.

- Attach a monitor to the VGA port and keyboard to the USB or PS/2 port
- Check mains power is on and press the power switch to start the appliance (the fans should start & front panel LEDs should light)
- Allow a minute for booting

The next few pages of this document cover the following steps:

- Initial Network Interface Configuration
- Accessing the WUI
- Configuring the appliance using the web based wizard
- Appliance configuration using the WUI
- Testing the load balancer configuration



Serial connection
for the fail-over
(heartbeat) cable

eth0 is usually the
internal network

eth1 is usually the
external network

Initial Network Interface Configuration

By default the load balancer is pre-configured with the following IP address & mask:

192.168.2.21 / 255.255.255.0

This default address can be changed in two ways:

- Using the built-in Network Setup Wizard
- Using traditional Linux commands

Using the Network Setup Wizard

To run the wizard, login to the console of the appliance as the 'setup' user. This is explained in the initial console start-up message as shown below:

```
Welcome to the Loadbalancer.org appliance.

To perform initial network configuration, log in to the console as
Username: setup
Password: setup

To access the web interface and wizard, point your browser at
http://192.168.2.21:9080/
or
https://192.168.2.21:9443/

lbmaster login: _
```

- login to the console:
Username: setup
Password: setup
- Once logged in, enter the IP address, mask and default gateway at the prompts as shown below:

```
Loadbalancer.org basic network set up

Static IP address (eg. 192.168.0.26)      : 192.168.70.175
Interface netmask (eg. 24)                : 18
Default gateway (eg. 192.168.0.1)         : 192.168.64.1
```

After the required settings have been entered, a summary will be presented along with details of how to access the WUI as shown below:

Summary of settings

Static IP address: 192.168.70.175/18
Default gateway: 192.168.64.1

You may now connect the eth0 network interface to your switch, and continue configuration through the web interface on:

`http://192.168.70.175:9080/lbadmin/`

Press any key... _

The IP address is now configured for interface eth0.

IP addresses for the other interfaces can now be configured via the WUI or using the Linux commands covered in the next section.

Using Linux Commands

To set the IP address, login to the console or an SSH session as root:

Username: root
Password: loadbalancer

Now set the IP address using the following command:

```
ip addr add <IP address>/<mask> dev eth0
```

e.g.

```
ip addr add 192.168.1.100/24 dev eth0
```

NB. Setting the IP address in this way is temporary, the IP address MUST be set via the WUI to make this permanent

Accessing the Web User Interface (WUI)

- Using a web browser, access the WUI using the following URL:

http://192.168.2.21:9080/lbadmin/

(replace 192.168.2.21 with your IP address if its been changed)

NB. If you prefer you can use the HTTPS administration address:

https://192.168.2.21:9443/lbadmin/

(replace 192.168.2.21 with your IP address if its been changed)

- Login to the WUI:

Username: loadbalancer

Password: loadbalancer

- Once logged in, you'll be asked if you want to run the web based setup wizard. The wizard asks a series of questions in order to setup the appliance with an initial basic configuration. If you prefer to configure the appliance manually, simple select 'no' to the question.

EDIT CONFIGURATION > SETUP WIZARD

The Loadbalancer.org Setup Wizard has not been run yet. You can run it now or anytime later with Edit Configuration > Setup Wizard

Do you want to run it now?

☐ yes ☐ no

Configuring the Loadbalancer.org Appliance Using the Web Based Wizard

The wizard can be used to setup a single layer 4 DR mode or NAT mode Virtual Server with a single real server. The wizard can be used for both single unit deployments and clustered pair deployments.

Outline steps – Single unit deployments:

- Set the IP address using the methods described earlier
- Now start the WUI and run the wizard (*Edit Configuration > Setup Wizard*)

Outline steps – Clustered pair deployments:

- Set the IP address on both units as described earlier
- Connect the serial cable (*NB. it's also possible to use the network for heartbeat comms if preferred*)
- Start the WUI on the slave unit and run the wizard (*Edit Configuration > Setup Wizard*)
- Now run the wizard on the master unit to complete the process

Example Answers Using the Wizard for a Two-Arm NAT Configuration (Single Unit)

The following example covers setting up a layer 4 NAT mode virtual server with one real server. Additional Virtual Servers (VIPs) and Real Servers (RIPs) can then be added using the WUI.

EDIT CONFIGURATION > SETUP WIZARD

Is this unit part of an HA-pair? ☐ yes ☒ no

Will the load balancer form part of a one armed set-up (i.e. same subnet as servers)? ☐ yes ☒ no

Then the load balancer will form part of a two-armed set-up. (See Quickstart guide for further explanation.)

We will now configure the load balancer's network interfaces:

Enter the IP address for the INTERNAL interface eth0 (CIDR format):

Enter the IP address for the EXTERNAL interface eth1 (CIDR format):

Now we will configure the DNS and gateway settings for the load balancer.

Enter the IP address of the default gateway IP v4:

Enter the IP address of the default gateway IP v6:

Enter the IP address of the nameserver:

Enter the IP address of the second nameserver:

Now we will configure the first Virtual Service.

Enter the port number for the Virtual Service:

Enter the IP address of the first Real Server (backend):

Please check that all your settings are correct!

Check that your settings are correct and click **Submit**. Once the wizard is complete the load balancer is configured and ready to use.

For NAT mode – as used in this example, you must also configure the real server to ensure that it uses the internal IP address of the load balancer as its default gateway. Once this is done you can test the virtual server from the external network. By default, the wizard uses the IP address of the external interface for the first virtual server, 10.0.0.120 in this example.

You can now use the *Edit Configuration* menu in the WUI to easily add more virtual or real servers to your configuration.



To restore manufacturer's settings – at the console use the command **lbrestore** or in the WUI goto *Maintenance > Backup & Restore > Restore Manufacturer's Defaults*. This will reset the address to 192.168.2.21.

Appliance Configuration Using the Web User Interface



When using a Clustered Pair, all configuration must be done via the master unit, the slave unit will then be synchronized automatically.

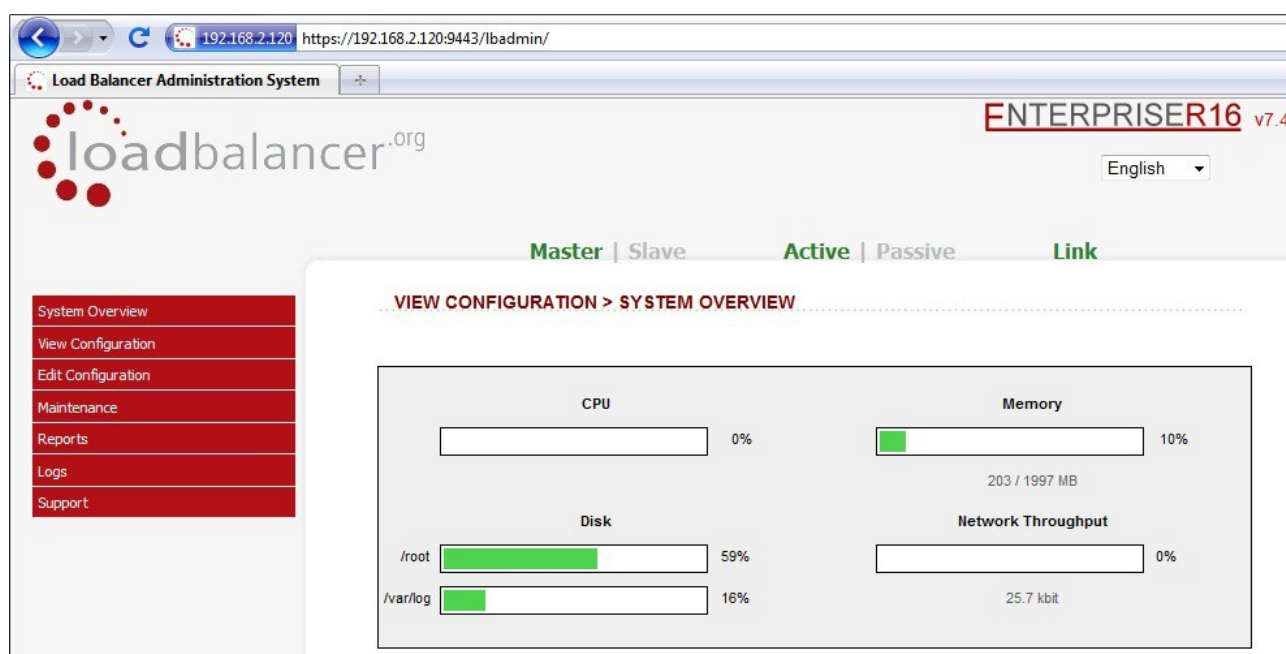
If you have already used the web based wizard, then you will already be using the WUI. From here all administration tasks can be carried out. If not, access the WUI as follows:

With a web browser access the web interface: ***http://192.168.2.21:9080/lbadmin/***

(replace 192.168.2.21 with the correct IP address)

log in to the WUI: ***Username:*** loadbalancer
Password: loadbalancer

*NB. If you prefer you can use the HTTPS administration address: ***https://192.168.2.21:9443/lbadmin/****



All administration tasks can be carried out through the web interface. If root access to the appliance is required for any reason via the console or a ssh session, the following default credentials should be used:

root credentials: ***Username:*** root
Password: loadbalancer

Adding Virtual Servers

If used, the wizard sets up a single Virtual Server (VIP). Extra VIPs can be added using the WUI.

- Select *Edit Configuration > Layer 4 Configuration > Virtual Servers*

NB. If the wizard was used, you'll see the VIP that was created by the wizard

EDIT CONFIGURATION > VIRTUAL SERVERS

[Add a new Virtual Server]

VIP1	192.168.69.35	Port 80 - tcp	Direct Routing	[Modify]	[Delete]
------	---------------	---------------	----------------	------------	------------

- Click [Add a new Virtual Server]

EDIT CONFIGURATION > ADD A NEW VIRTUAL SERVER

Label	<input type="text" value="VIP Name"/>	?
Virtual Server IP address	<input type="text" value="10.0.0.20"/>	?
Virtual Server Ports	<input type="text" value="80"/>	?
Forwarding Method	<input type="text" value="Direct Routing"/>	?
Persistent	<input type="text" value="no"/>	?
Protocol	<input type="text" value="TCP"/>	?
<input type="button" value="Update"/>		

- Define the required settings for the new VIP :
 - Enter the Label, IP address and port(s) for the VIP
 - Select the required forwarding method
 - Enable persistence if required
 - Set the protocol (normally TCP)

Adding Real Servers

If used, the wizard sets up a single Real Server (RIP). Extra RIPs can be added using the WUI.

- Select *Edit Configuration > Layer 4 Configuration > Real Servers*

NB. If the wizard was used, you'll see the RIP that was created by the wizard

EDIT CONFIGURATION > REAL SERVERS

VIP1	192.168.69.35	Port 80	Direct Routing	[Add a new Real Server]
RIP1	192.168.68.41		Weight 1	[Modify] [Delete]

- Click [Add a new Real Server]

EDIT CONFIGURATION > ADD A NEW REAL SERVER

Label	<input type="text" value="RIP Name"/>	?
Real Server IP Address	<input type="text" value="IPAddress"/>	?
Weight	<input type="text" value="1"/>	?
Minimum Connections	<input type="text" value="0"/>	?
Maximum Connections	<input type="text" value="0"/>	?
<input type="button" value="Update"/>		

- Define the required settings for the new RIP :
 - Enter the Label, IP address and port for the RIP
 - Set the weight – this defaults to 1. If real servers have different performance specifications, then the weight can be adjusted – a higher number means more traffic is sent to that server
 - Leave the Minimum & Maximum Connections set to 0 for unrestricted

Configuring the Real Servers

Depending on the deployment method (DR, NAT or SNAT) used, the actual physical servers may need additional configuration to allow the load balancer to operate correctly. The following sections define what is needed for the various modes.

Configuring the Real Servers for Layer 4 NAT Mode

If you are using a two-arm NAT load balancing method, the real server configuration is a simple case of configuring the load balancer as the default gateway. Normally, a floating IP address is added using *Edit Configuration > Floating IPs*. This is important when a master / slave configuration is used to allow failover & fallback of the default gateway address.



Failure to correctly configure the real servers default gateway is the most common mistake when using NAT mode.

Configuring the Real Servers for Layer 4 DR Mode (Linux)

If you are using the one-arm DR load balancing method, each real server requires the ARP problem to be solved. All real servers must be configured to respond to the VIP address as well as the RIP address. This is because in DR mode load balanced traffic arrives on the VIP address, whilst other traffic such as health-checks, administration traffic etc. use the real server's IP address.

Detecting the ARP Problem

You can use *Reports > Layer 4 Current Connections* to check whether the ARP problem has been solved. If not, the connection state will be SYN_RECV as shown below when a client connection to the VIP is attempted:

REPORTS > LAYER 4 CURRENT CONNECTIONS

```
IPVS connection entries
pro expire state      source          virtual         destination
TCP 00:51 SYN_RECV    192.168.2.7:64763 192.168.2.109:80 192.168.2.99:80
```

Solving for Linux – Method 1 (using iptables)

You can use iptables (netfilter) on each real server to re-direct incoming packets destined for the virtual server IP address. To make this permanent, simply add the command to an appropriate start-up script such as */etc/rc.local*. If the real server is serving multiple VIPs, add additional iptables rules for each VIP.

```
iptables -t nat -A PREROUTING -p tcp -d <VIP> -j REDIRECT
```

e.g.

```
iptables -t nat -A PREROUTING -p tcp -d 10.0.0.21 -j REDIRECT
```

this means redirect any incoming packets destined for 10.0.0.21 (the virtual server) locally.



Method 1 does not work with IPv6 Virtual Servers, use method 2 below instead.

Solving for Linux – Method 2 (using arp_ignore sysctl values)

This is the preferred method as it supports both IPv4 and IPv6. Each real server needs the loopback adapter to be configured with the Virtual Servers IP address. This address must not respond to ARP requests and the web server also needs to be configured to respond to this address. To set this up follow steps 1-3 below.

Step 1 : re-configure ARP on the real servers (this step can be skipped for IPv6 virtual servers)

To do this add the following lines to /etc/sysctl.conf:

```
net.ipv4.conf.all.arp_ignore=1
net.ipv4.conf.eth0.arp_ignore=1
net.ipv4.conf.eth1.arp_ignore=1
net.ipv4.conf.all.arp_announce=2
net.ipv4.conf.eth0.arp_announce=2
net.ipv4.conf.eth1.arp_announce=2
```

Step 2 : apply these settings

Either reboot the real server or run the following command to apply these settings:

```
/sbin/sysctl -p
```

Step 3 : add the virtual servers' IP address to the loopback adapter

run the following command for each VIP. To make this permanent, simply add the command to an appropriate startup script such as /etc/rc.local.

```
ip addr add dev lo <IPv4-VIP>/32
```

for IPv6 addresses use:

```
ip addr add dev lo <IPv6-VIP>/128
```

Alternatively, modify the appropriate interface script to add the additional IP address(es).



Failure to correctly configure the real servers to handle the ARP problem is the most common mistake in DR mode configurations.

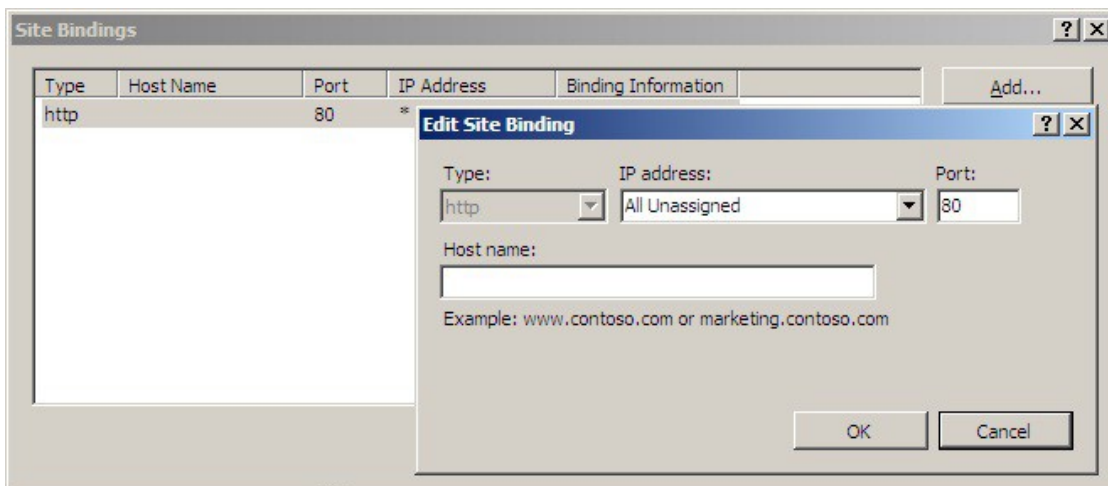
Configuring the Real Servers for Layer 4 DR Mode (Windows)

If you're using a one-arm DR mode load balancing method, each web server requires the ARP problem to be handled:

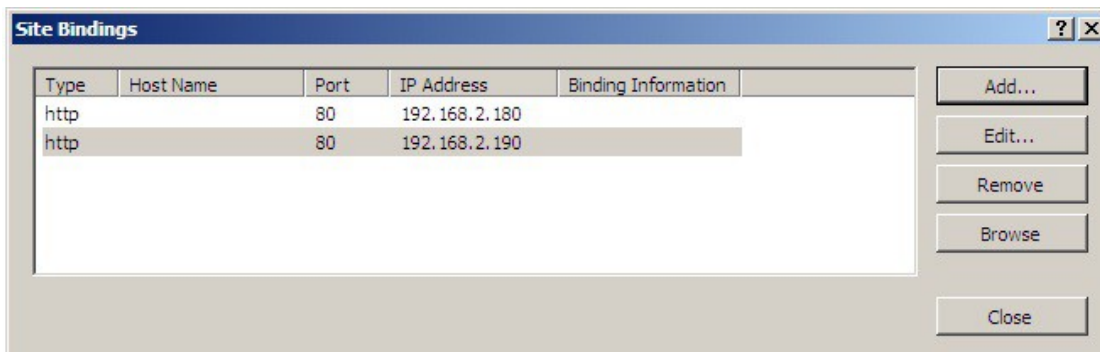
- For all real servers in Direct Routing mode the load balanced application must respond to both the Virtual IP as well as the servers Real IP. With Windows IIS the IP address must either be set to 'All Unassigned' or use the Advanced tab to add a second IP address as shown below
- Each real server must have the Microsoft loopback adapter installed and configured
- The Microsoft loopback adapter must be configured to deal with the ARP problem
- For Windows 2008 / 2012 a series of three netsh commands must also be run on each server to configure the weak / strong host behavior

Configuring IIS to Respond to Both the RIP and VIP

By default, IIS listens on all configured IP addresses, this is shown in the example below. As can be seen the IP address field is set to 'All Unassigned'.



If the default configuration is left, no further IIS configuration is required. If you do change the IP address in the bindings from 'All Unassigned' to a specific IP address, then you need to make sure that you also add a binding for the Virtual Server IP address (VIP) as shown below:

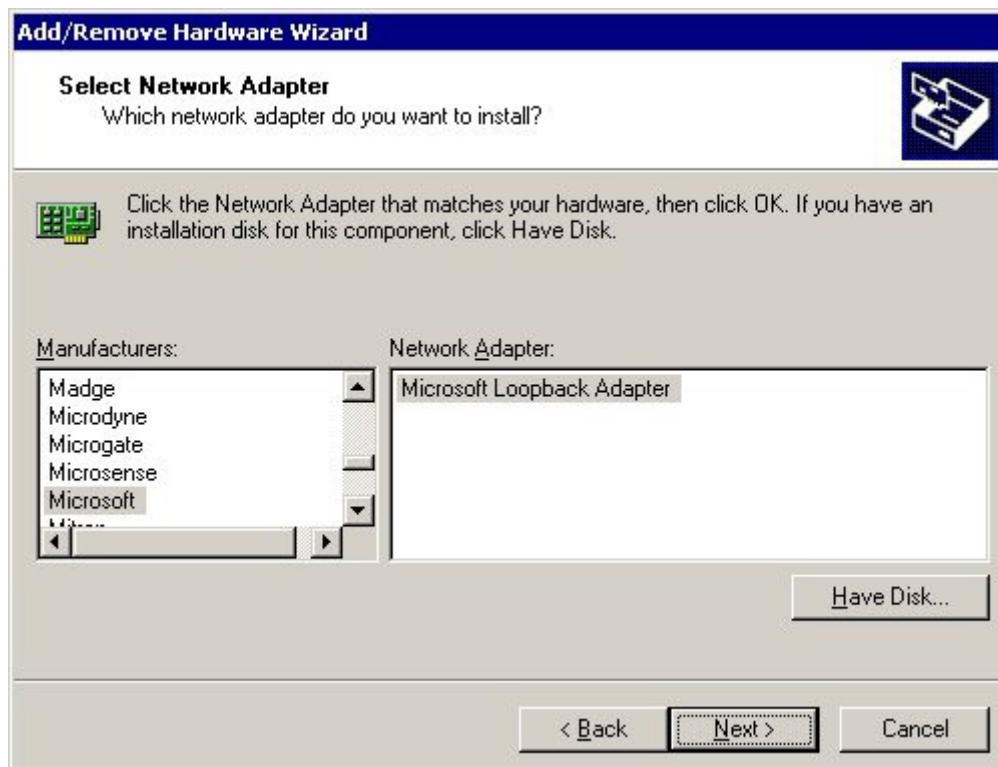


Resolving ARP issues for Windows server 2000 (applies to DR mode only)

Windows 2000 Server supports the direct routing (DR) method through the use of the MS loopback adapter to handle the traffic. The IP address on the loopback adapter must be set to be the same as the Virtual Servers IP address (VIP). If the real server is included in multiple VIPs, you can add additional IP addresses to the loopback adapter that correspond to each VIP.

Step 1 – Install the Microsoft loopback adapter

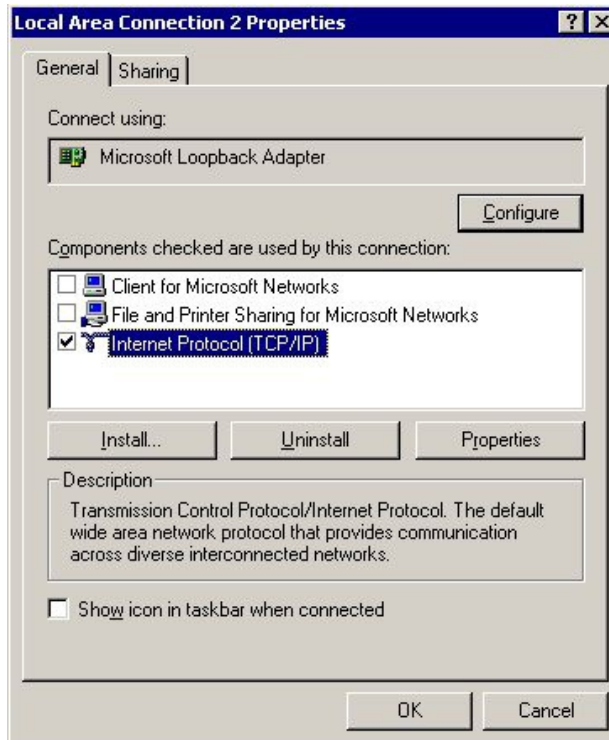
1. Open the Control Panel and double-click **Add/Remove Hardware**
2. Once the Hardware Wizard opens, click **Next**
3. Select **Add/Troubleshoot a device**, click **Next**
4. Once the device list appears, select **Add a new device** at the top of the list, click **Next**
5. Select **No, I want to select the hardware from a list**, click **Next**
6. Scroll down the list and select **Network Adapters**, click **Next**
7. Select **Microsoft & Microsoft Loopback Adapter**, click **Next** as shown below



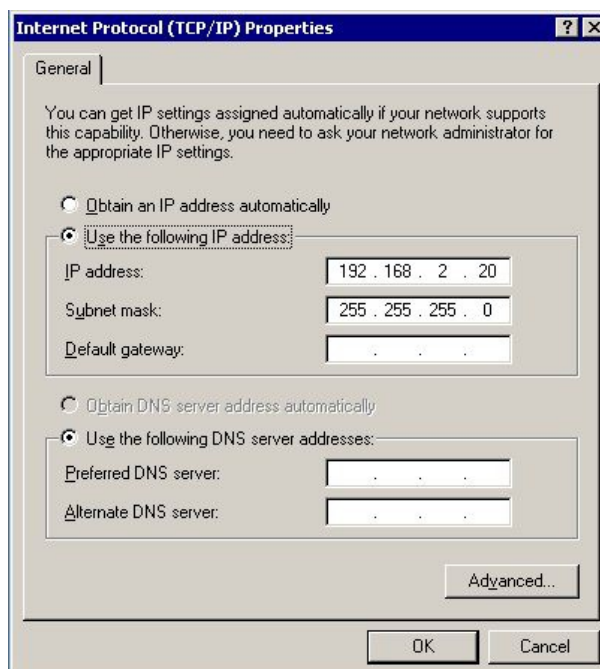
8. Click **Next** to start the installation, when complete click **Finish**

Step 2 – Configure the loopback adapter

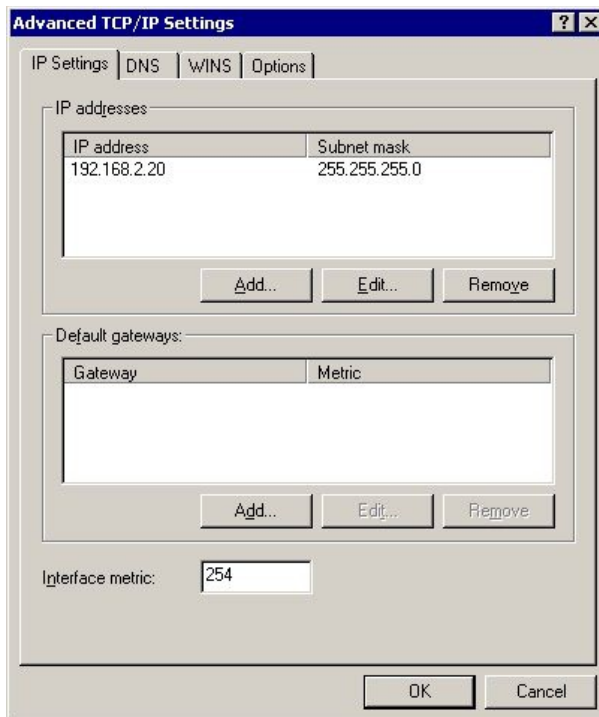
1. Open the Control Panel and double-click **Network and Dial-up Connections**
2. Right-click the new loopback adapter and select **Properties**
3. Un-check all items except **Internet Protocol (TCP/IP)** as shown below



4. Select **Internet Protocol (TCP/IP)**, click **Properties** and configure the IP address and mask to be the same as the Virtual Server IP address (VIP), e.g. 192.168.2.20/24 as shown below



5. Click **Advanced** and change the **Interface metric** to 254 as shown below, this prevents the adapter responding to ARP requests for the VIP address



6. Click **OK** on Advanced Settings, TCP/IP Properties and Connection Properties to save and apply the new settings
7. Repeat the above steps for all other Windows 2000 real servers

Resolving ARP issues for Windows server 2003 (applies to DR mode only)

Windows server supports the direct routing (DR) method through the use of the MS loopback adapter to handle the traffic. The IP address on the loopback adapter must be set to be the same as the Virtual Servers IP address (VIP). If the real server is included in multiple VIPs, you can add additional IP addresses to the loopback adapter that correspond to each VIP.

Step 1 – Install the Microsoft loopback adapter

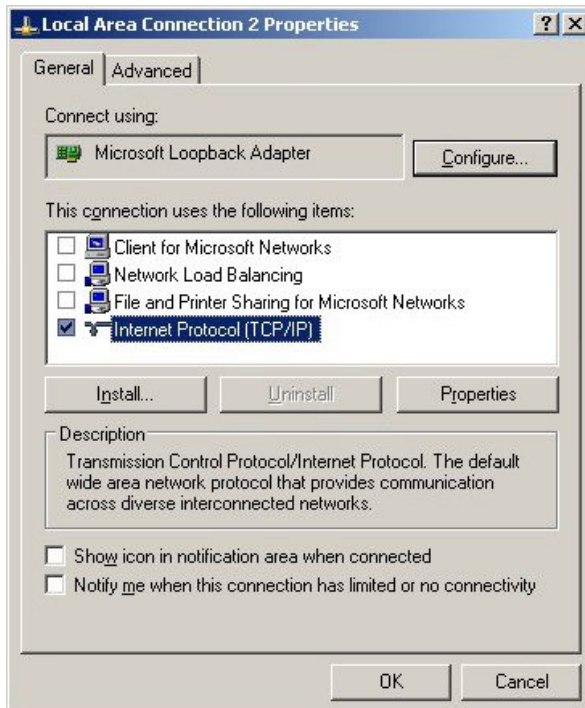
1. Open the Control Panel and double-click **Add Hardware**
2. Once the Hardware Wizard opens, click **Next**
3. Select **Yes, I have already connected the hardware**, click **Next**
4. Scroll to the bottom of the list, select **Add a new hardware device**, click **Next**
5. Select **Install the hardware that I manually select from a list (Advanced)**, click **Next**
6. Select **Network adapters**, click **Next**
7. Select **Microsoft & Microsoft Loopback Adapter**, click **Next** as shown below



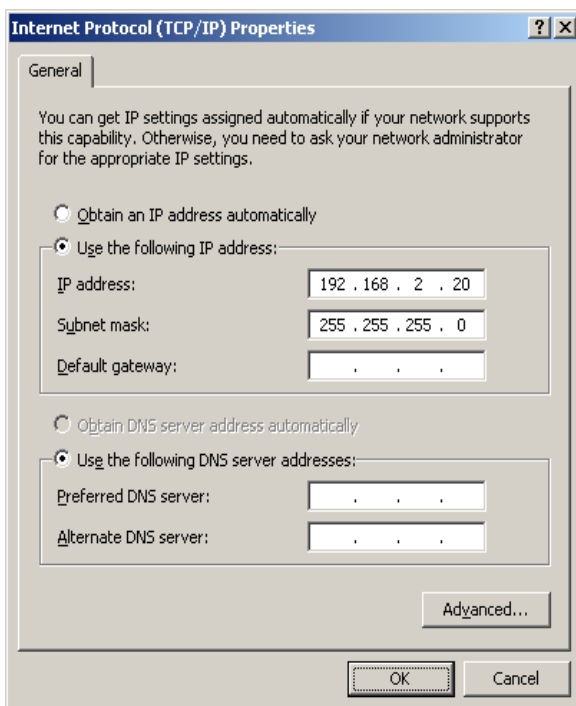
8. Click **Next** to start the installation, when complete click **Finish**

Step 2 – Configure the loopback adapter

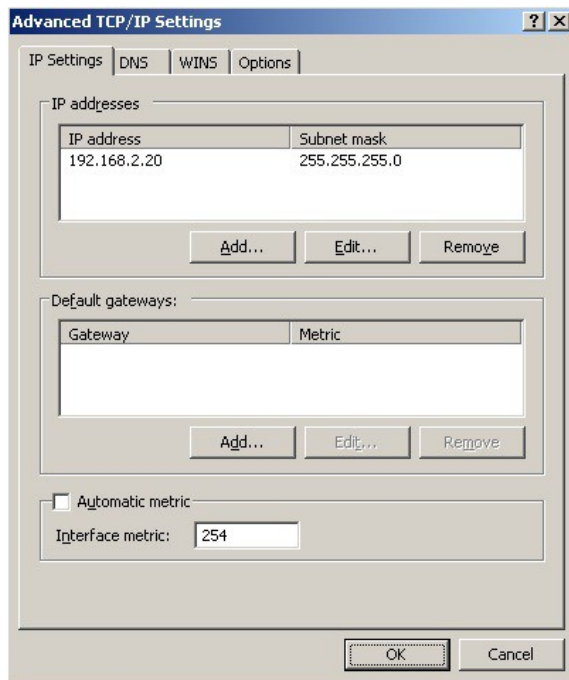
1. Open the Control Panel and double-click **Network Connections**
2. Right-click the new loopback adapter and select **Properties**
3. Un-check all items except **Internet Protocol (TCP/IP)** as shown below



4. Select **Internet Protocol (TCP/IP)**, click **Properties** and configure the IP address and mask to be the same as the Virtual Server (VIP), e.g. 192.168.2.20/24 as shown below



- Click **Advanced**, un-check **Automatic metric** and change **Interface metric** to 254 as shown below, this prevents the adapter responding to ARP requests for the VIP address



- Click **OK** on Advanced Settings & TCP/IP Properties, then click **Close** on Connection Properties to save and apply the new settings
- Now repeat the above process for all other Windows 2003 real servers



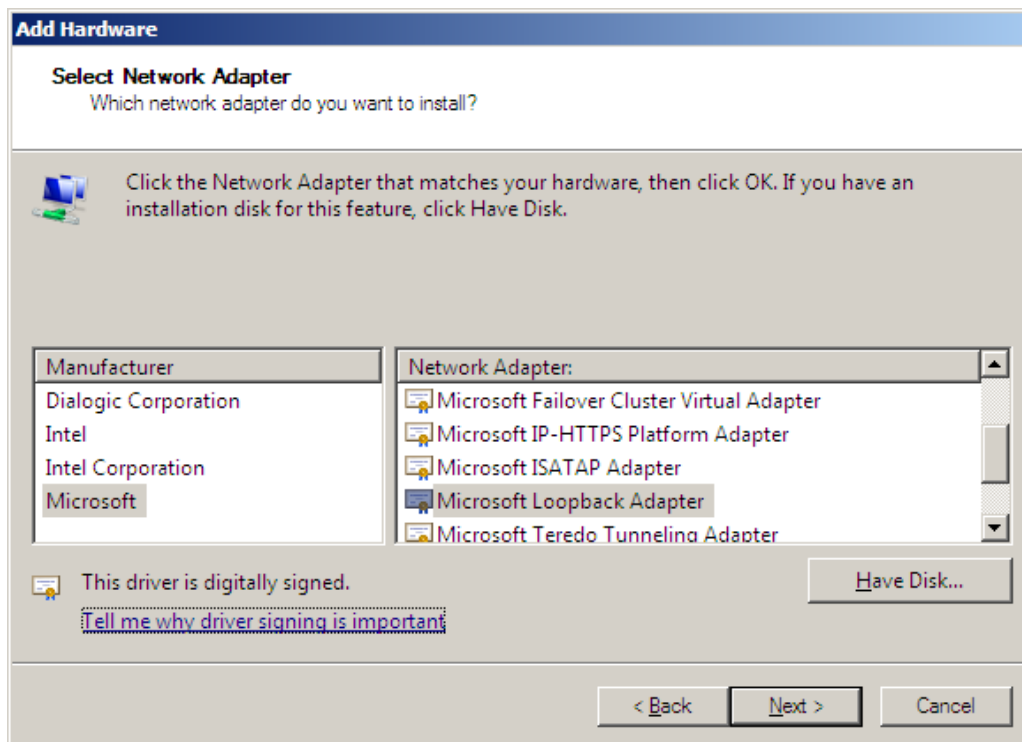
For Windows server 2003 SP1 & later, if you have enabled the built-in firewall, you will need to create an exception to enable access to the web server. This exception by default will allow traffic on both the network and loopback adapters.

Resolving ARP issues for Windows server 2008 (applies to DR mode only)

The basic concept is the same as for Windows 2000 / 2003. However, additional steps are required to set the strong / weak host behavior. This is used to either block or allow interfaces receiving packets destined for a different interface on the same server. As with Windows 2000 / 2003, If the real server is included in multiple VIPs, you can add additional IP addresses to the loopback adapter that correspond to each VIP.

Step 1 – Install the Microsoft loopback adapter

1. Click **Start**, then run **hdwwiz** to start the Hardware Installation Wizard
2. When the Wizard has started, click **Next**
3. Select **Install the hardware that I manually select from a list (Advanced)**, click **Next**
4. Select **Network adapters**, click **Next**
5. Select **Microsoft & Microsoft Loopback Adapter**, click **Next**

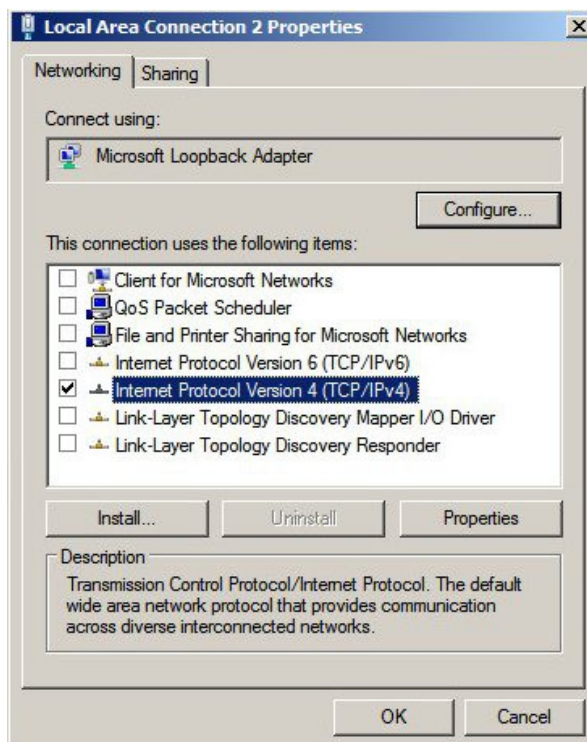


6. Click **Next** to start the installation, when complete click **Finish**

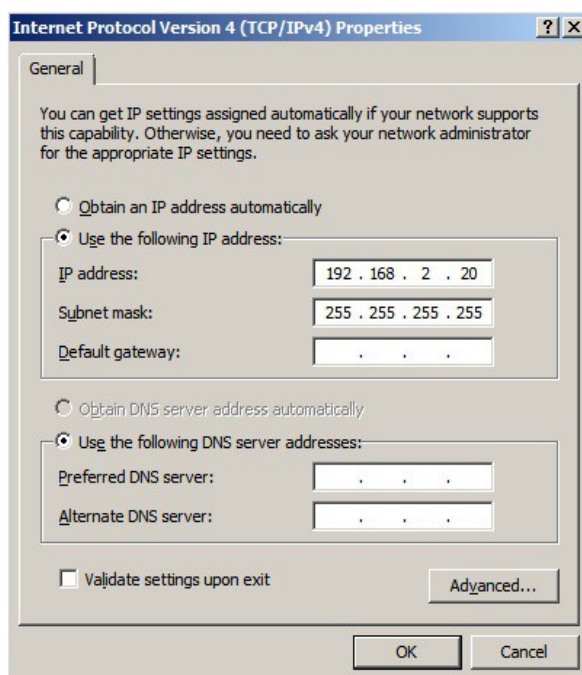
Step 2 – Configure the loopback adapter

1. Open Control Panel and click **View Network status and tasks** under **Network and internet**
2. Click **Change adapter settings**
3. Right-click the new Loopback adapter and select **Properties**

4. Un-check all items except **Internet Protocol Version 4 (TCP/IPv4)** as shown below



5. Select **Internet Protocol Version (TCP/IPv4)**, click **Properties** and configure the IP address to be the same as the Virtual Server (VIP) with a full subnet mask, e.g. 192.168.2.20/32 as shown below



6. Click **OK** on TCP/IP Properties, then click **Close** on Connection Properties to save and apply the new settings
7. Now repeat the above process on the other Windows 2008 real servers

N.B. For Windows 2008, it's not necessary to modify the interface metric on the advanced tab and should be left set to Automatic

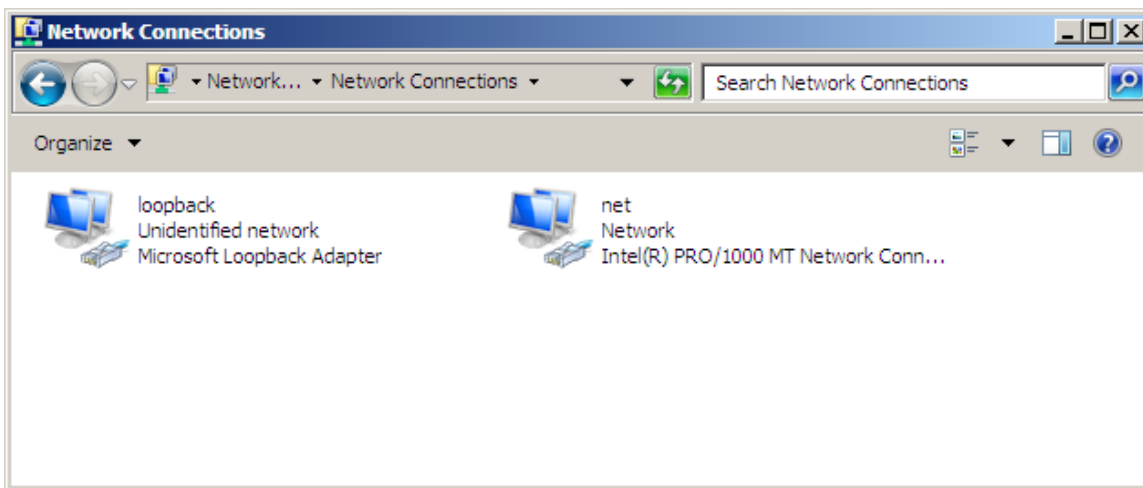
Step 3 – Configure the strong / weak host behavior

Windows Server 2000 and Windows Server 2003 use the weak host model for sending and receiving for all IPv4 interfaces and the strong host model for sending and receiving for all IPv6 interfaces. You cannot configure this behavior. The Next Generation TCP/IP stack in Windows 2008 and later supports strong host sends and receives for both IPv4 and IPv6 by default. To ensure that the Windows 2008 is running in the correct mode to be able to respond to the VIP, the following commands must be run on each real server:

```
netsh interface ipv4 set interface "net" weakhostreceive=enabled
netsh interface ipv4 set interface "loopback" weakhostreceive=enabled
netsh interface ipv4 set interface "loopback" weakhostsend=enabled
```

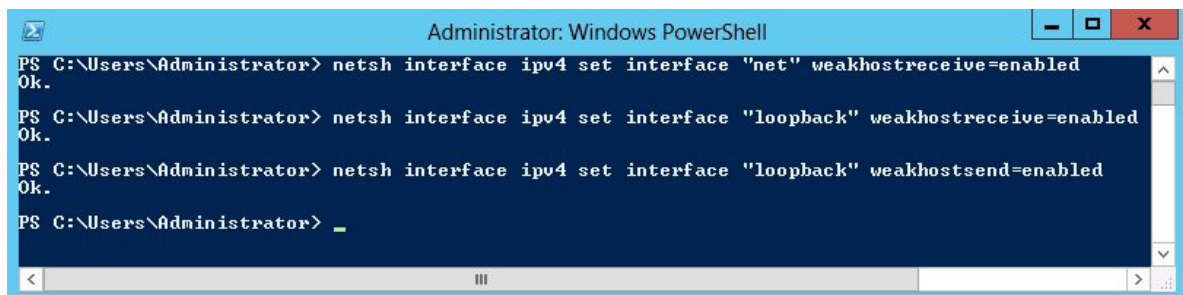
For these commands to work, the LAN connection NIC must be named “net” and the loopback NIC must be named “loopback” as shown below. If you prefer to leave your current NIC names, then the commands above must be modified accordingly. For example, if your network adapters are named “LAN” and “LOOPBACK”, the commands required would be:

```
netsh interface ipv4 set interface "LAN" weakhostreceive=enabled
netsh interface ipv4 set interface "LOOPBACK" weakhostreceive=enabled
netsh interface ipv4 set interface "LOOPBACK" weakhostsend=enabled
```



N.B. The names for the NICs are case sensitive, so make sure that the name used for the interface and the name used in the commands match exactly.

1. Start Powershell or use a command Window to run the 3 netsh commands as shown below



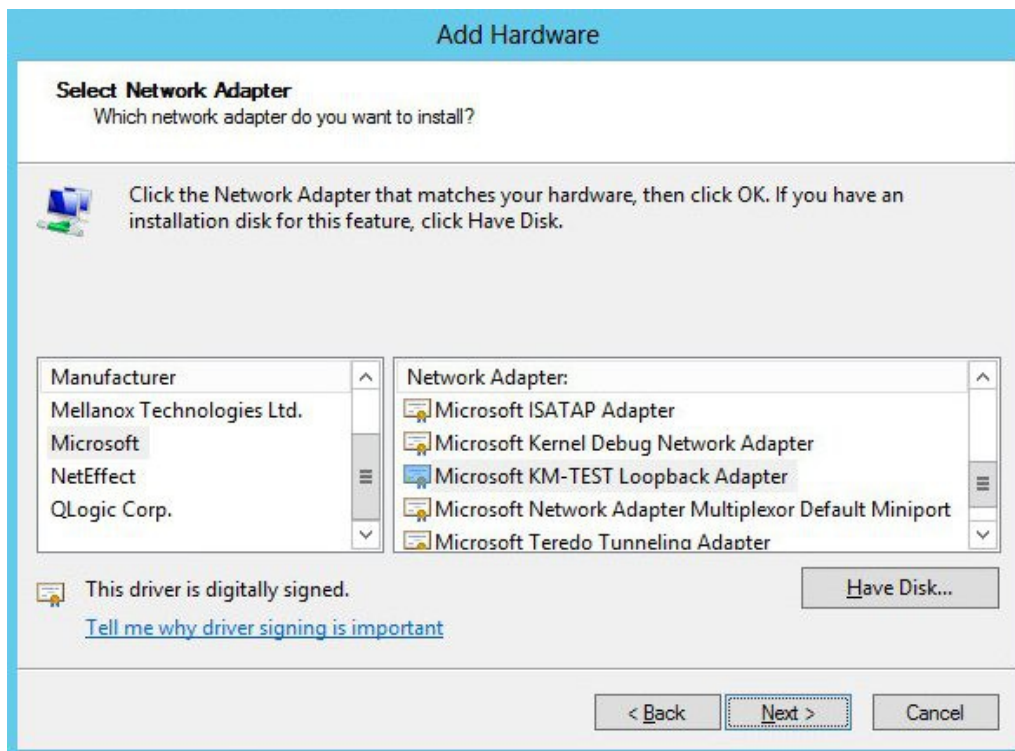
2. Now repeat these 3 commands on the other Windows 2008 real servers

Resolving ARP issues for Windows server 2012 (applies to DR mode only)

The basic concept is the same as for Windows 2000 / 2003. However, additional steps are required to set the strong / weak host behavior. This is used to either block or allow interfaces receiving packets destined for a different interface on the same server. As with Windows 2000 / 2003 / 2008, If the real server is included in multiple VIPs, you can add additional IP addresses to the loopback adapter that correspond to each VIP.

Step 1 – Install the Microsoft loopback adapter

1. Click **Start**, then run **hdwwiz** to start the Hardware Installation Wizard
2. When the Wizard has started, click **Next**
3. Select **Install the hardware that I manually select from a list (Advanced)**, click **Next**
4. Select **Network adapters**, click **Next**
5. Select **Microsoft & Microsoft KM-Test Loopback Adapter**, click **Next**

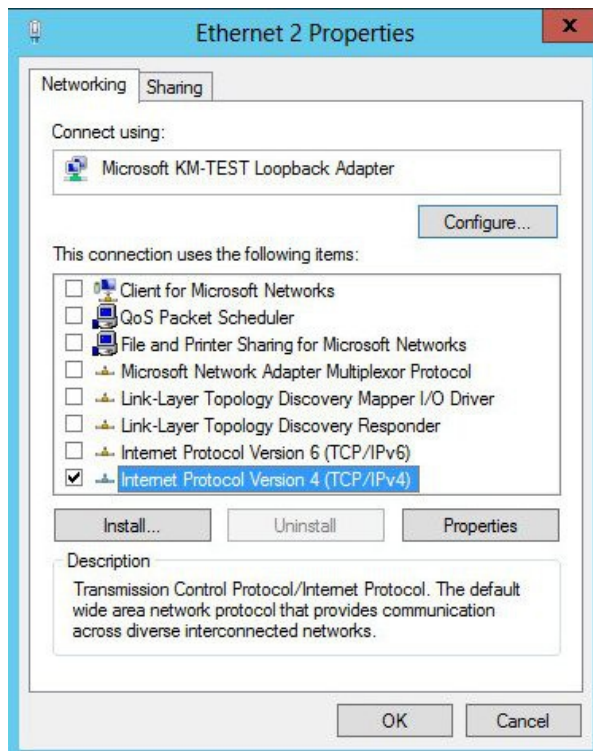


6. Click **Next** to start the installation, when complete click **Finish**

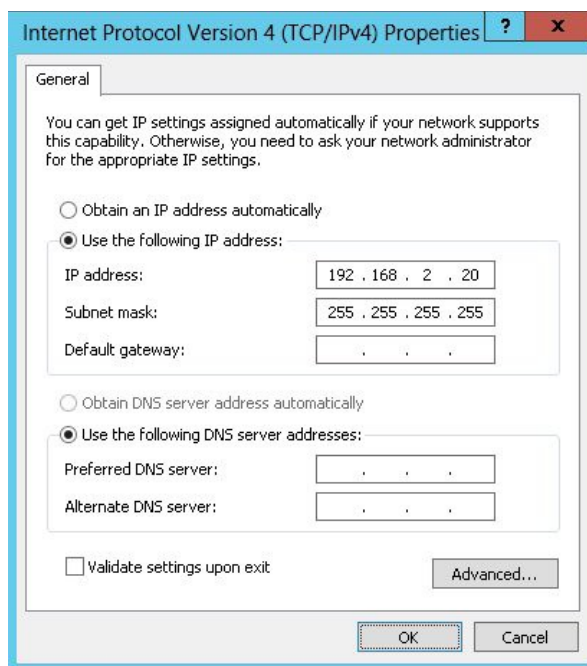
Step 2 – Configure the loopback adapter

1. Open Control Panel and click **Network and Sharing Center**
2. Click **Change adapter settings**
3. Right-click the new Loopback adapter and select **Properties**

- Un-check all items except **Internet Protocol Version 4 (TCP/IPv4)** as shown below



- Select **Internet Protocol Version (TCP/IPv4)**, click **Properties** and configure the IP address to be the same as the Virtual Server (VIP), with a full subnet mask e.g. 192.168.2.20/32 as shown below



- Click **OK** on TCP/IP Properties, then click **Close** on Ethernet Properties to save and apply the new settings
- Now repeat the above process on the other Windows 2012 real servers

N.B. For Windows 2012, it's not necessary to modify the interface metric on the advanced tab and should be left set to Automatic

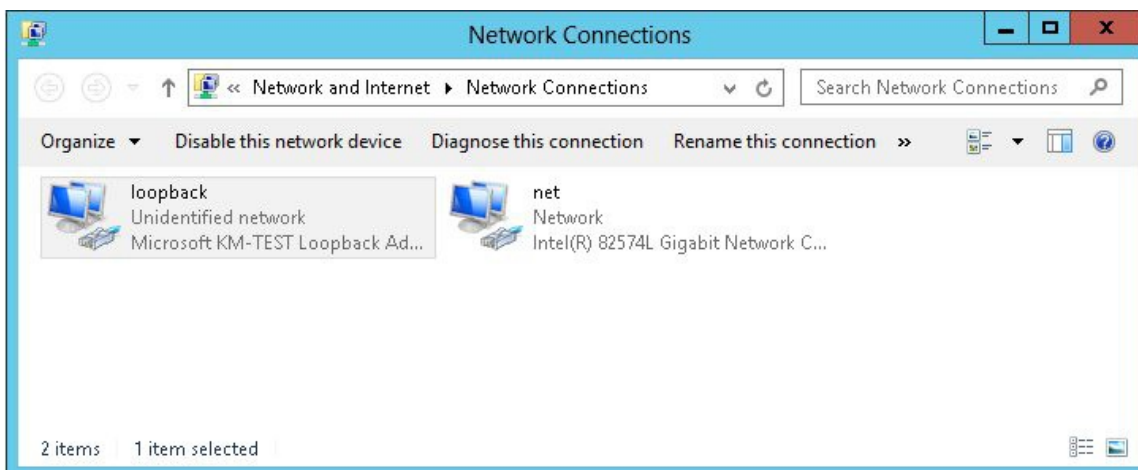
Step 3 – Configure the strong / weak host behavior

Windows Server 2000 and Windows Server 2003 use the weak host model for sending and receiving for all IPv4 interfaces and the strong host model for sending and receiving for all IPv6 interfaces. You cannot configure this behavior. The Next Generation TCP/IP stack in Windows 2008 and later supports strong host sends and receives for both IPv4 and IPv6 by default. To ensure that the Windows 2008 is running in the correct mode to be able to respond to the VIP, the following commands must be run on each real server:

```
netsh interface ipv4 set interface "net" weakhostreceive=enabled
netsh interface ipv4 set interface "loopback" weakhostreceive=enabled
netsh interface ipv4 set interface "loopback" weakhostsend=enabled
```

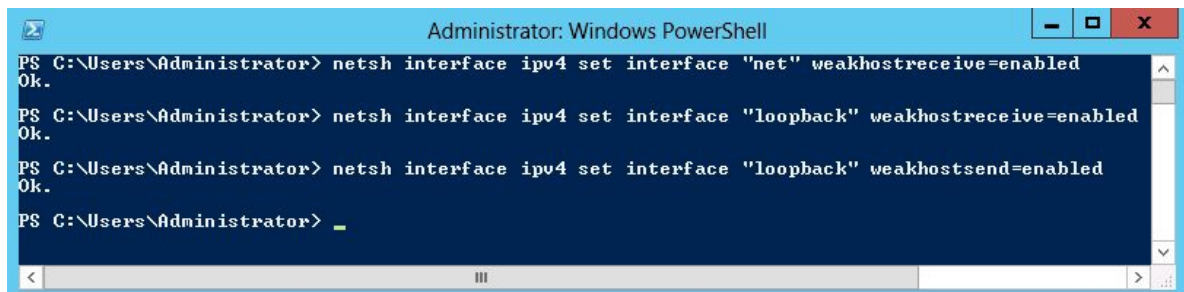
For these commands to work, the LAN connection NIC must be named “net” and the loopback NIC must be named “loopback” as shown below. If you prefer to leave your current NIC names, then the commands above must be modified accordingly. For example, if your network adapters are named “LAN” and “LOOPBACK”, the commands required would be:

```
netsh interface ipv4 set interface "LAN" weakhostreceive=enabled
netsh interface ipv4 set interface "LOOPBACK" weakhostreceive=enabled
netsh interface ipv4 set interface "LOOPBACK" weakhostsend=enabled
```



N.B. The names for the NICs are case sensitive, so make sure that the name used for the interface and the name used in the commands match exactly.

1. Start Powershell or use a command Window to run the 3 netsh commands as shown below



2. Now repeat these 3 commands on the other Windows 2012 real servers

Verifying netsh Settings for Windows 2008 & 2012

To verify that settings have been configured correctly, run the following command on each real server to clearly list the settings that have been applied to the interface:

```
netsh interface ipv4 show interface <interface name>
```

i.e. for the 'loopback' adapter run :netsh interface ipv4 show interface loopback

i.e. for the 'net' adapter run :netsh interface ipv4 show interface net

e.g.

```
C:\Users\Administrator>netsh interface ipv4 show interface loopback
```

```
Interface loopback Parameters
```

```
-----
IfLuid                : ethernet_9
IfIndex               : 15
State                 : connected
Metric                : 30
Link MTU              : 1500 bytes
Reachable Time        : 28500 ms
Base Reachable Time   : 30000 ms
Retransmission Interval : 1000 ms
DAD Transmits         : 3
Site Prefix Length    : 64
Site Id               : 1
Forwarding             : disabled
Advertising           : disabled
Neighbor Discovery     : enabled
Neighbor Unreachability Detection : enabled
Router Discovery       : dhcp
Managed Address Configuration : enabled
Other Stateful Configuration : enabled
Weak Host Sends        : enabled
Weak Host Receives     : enabled
Use Automatic Metric   : enabled
Ignore Default Routes  : disabled
Advertised Router Lifetime : 1800 seconds
Advertise Default Route : disabled
Current Hop Limit      : 0
Force ARPND wake up patterns : disabled
Directed MAC wake up patterns : disabled
```

```
C:\Users\Administrator>
```

This shows that the settings have been applied correctly.



For Windows server 2008 / 2012, if you want to leave the built-in firewall enabled, you'll either need to enable the relevant default firewall exceptions or create your own to enable access to the web server. By default these exceptions will allow traffic on both the network and loopback adapters.



Failure to correctly configure the real servers to handle the ARP problem is the most common problem in DR configurations.

Configuring the Real Server for Layer 7 SNAT Mode

When using Layer7 (HAProxy) Virtual Servers, no changes are required to the real servers.

IPv6 Support

New to v7.x is full IPv6 support. This allows Virtual Servers to be configured using IPv6 addresses. Its also possible to mix IPv4 and IPv6 addresses on a single appliance as illustrated below:

EDIT CONFIGURATION > NETWORK INTERFACE CONFIGURATION

Bonding	
Bond eth0 & eth1 as bond0:	<input type="checkbox"/> ?
Bond eth2 & eth3 as bond1:	<input type="checkbox"/> ?
Bond Interfaces	
VLAN	
Interface:	eth0 ? Add VLAN
VLAN ID:	1 ?
IP Address Assignment	
eth0	192.168.2.135/24 fde6:d14c:3089:1::382/120
eth1	10.12.1.135/24 fde6:d14c:3089:1::384/120
eth2	
eth3	
Configure Interfaces	

Testing the Load Balancer Configuration

For testing, add a page to each real web servers root directory e.g. test.html and put the server name on this page for easy identification during your tests.

Now you need a couple of clients to do the testing. Open up a web browser on two different clients and enter the URL for the VIP i.e. <http://192.168.1.20/>.

Each client should see a different server name because of the load balancing algorithm in use i.e. they are being load balanced across the cluster.

Why test using two clients? If you use a single client it will most likely keep on hitting the same server for multiple requests. This is to do with the way that the load balancing algorithms are optimized.



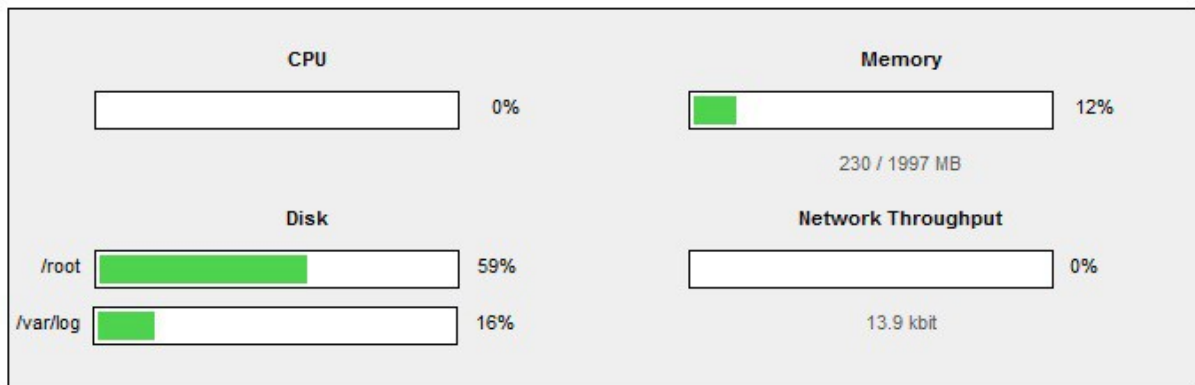
When using a two-arm NAT load balancing method the test client must be in the external subnet.

Connection Error Diagnosis

If you get a connection error when trying to access the VIP then:

1. Check *View Configuration > Network Configuration* and make sure that the VIP is active on the load balancer, if not check *Logs > Heartbeat* for errors
2. Check *System Overview* and make sure none of your VIPs are highlighted in red. If they are, your cluster is down. Green indicates a healthy cluster, yellow indicates that your cluster may need attention (one real server may be down), and blue indicates a real server has been deliberately taken offline

VIEW CONFIGURATION > SYSTEM OVERVIEW



Key cluster healthy cluster may need attention cluster is down real server deliberately offline

+	HTTP_Cluster - 192.168.2.182 Ports 80 Protocol TCP	Connections - Active: 0 Inactive: 0
+	FTP_Cluster - 192.168.2.184 Ports 21 Protocol TCP	Connections - Active: 0 Inactive: 0
+	SMTP_Cluster - 192.168.2.186 Ports 25 Protocol TCP	Connections - Active: 0 Inactive: 0

- If the VIP is still not working then check *Reports > Current Connections* to see the current traffic in detail, any packets marked SYN_RECV imply incorrect real server configuration. Check that any NAT mode servers have the correct default gateway and any DR mode servers are responding to the VIP as well as their own IP.

Health Check Diagnosis

Go to the Maintenance > System Overview section of the web interface and check that when you use 'take offline' the connections are redirected to the rest of the cluster as expected.

Pull the network cable out of one of the web servers, wait a few seconds (for the load balancer to detect the change) and then refresh the browsers on both clients. They should now both switch to the same server (as one has been removed from the load balancing list).

Put the network cable back in to the web server, wait a few seconds and then refresh the browsers again. They should now show different web servers again.

Key cluster healthy cluster may need attention cluster is down real server deliberately offline

HTTP_Cluster - 192.168.2.182 Ports 80 Protocol TCP Connections - Active: 0 Inactive: 0									
Label	IP	Method	Weight	Active conns	Inactive conns				
alpha_server	192.168.2.178	DR	1	0	0	Drain	Halt	↑	
bravo_server	192.168.2.190	DR	0	0	0	Bring Online		⚙	
charlie_server	192.168.2.191	DR	0	0	0	Drain	Halt	↓	

'**alpha_server**' is green which indicates that the server is operating normally.

'**bravo_server**' is blue, this indicates that it is deliberately in maintenance mode. You can use 'Bring Online' to make it active.

'**charlie_server**' is down (red). This implies that the real server has failed a health check; you can investigate this using *Logs > Layer 4*. If you know the real server should be active, you may need to increase the health check time-out *Edit Configuration > Layer 4 – Advanced Settings* or *Layer 7 – Advanced Settings*.

Appliance Log Files

The appliance has a number of log files that are very useful when diagnosing problems. These can be viewed using the WUI under the *Logs* main menu option.

Any errors that occur can prevent services being brought up, so if you're experiencing issues, reviewing the logs for any obvious issues is often a good place to start.

Testing High-Availability for a Loadbalancer.org HA-Pair

To test fail-over of a clustered pair, once fully configured power down the master and check that the slave unit takes over all the floating IP(s). If fail-over to the slave unit does not occur correctly, check *Logs > Heartbeat* on both nodes for any errors.



It's very important to verify that master / slave failover occurs correctly before going live. This proves the resilience of the cluster and makes you aware of the failover / failback process. Please refer to the administration manual for details of the `hb_takeover` command which can be used to force a failover / failback.



When testing load balancer fail-over, don't just pull the serial cable and network cable out. This will not cause a fail-over but will cause a split brain (i.e. both units active) to occur. You can configure fail-over on network failure but this is not enabled by default. To enable this, a ping node must be configured under *Edit Configuration > Heartbeat Configuration*.

New to v7.x is the role status at the top of each screen. For a working pair, the normal view is shown below:



This shows that the master unit is active and that the heartbeat link is up between master & slave.

Other states:

Master Slave	Active Passive	Link	this is a master unit, it's active, no slave unit has been defined
Master Slave	Active Passive	Link	this is a master unit, it's active, a slave has been defined but the link to the slave is down. Action: <i>check & verify the heartbeat configuration</i>
Master Slave	Active Passive	Link	this is a slave unit, it's active (a failover from the master has occurred) and the heartbeat link to the master has been established
Master Slave	Active Passive	Link	this is a master unit, a slave unit has been defined, but the link is down (e.g. serial cable unplugged) so the state cannot be determined. In this case the floating IP's may be active on both units. Action: <i>check & verify the heartbeat configuration, check the serial cable (if applicable), check heartbeat logs & if required restart heartbeat on both units</i>
Master Slave	Active Passive	Link	this is the master unit, a slave unit has been defined on the master, but the link is down (e.g. serial cable unplugged) so the state cannot be determined. In this case the heartbeat service has probably stopped on both units. Action: <i>check & verify the heartbeat configuration, check the serial cable (if applicable), check heartbeat logs & if required restart heartbeat on both units</i>

NB. Restarting heartbeat will cause a temporary outage of all load balanced services

Does Your Application Cluster Correctly Handle its Own State?



Load balancers work most effectively if the application servers are completely stateless. This means that if a web server fails and is automatically taken out of the cluster; then all the current user sessions will be transferred to other servers in the cluster without the users needing to re-login to the application again. ***If your application doesn't have a persistent data store then you can't have seamless fail over for your back-end servers.***

Do your web servers store persistent information on local drives?

- Images (jpeg, png, gif etc.)
- Files (html, php, asp etc.)

If so these files either need to be on shared storage such as an NFS/CIFS mount, or they need to be replicated to all of the nodes in the cluster.

Replication Solutions for Shared Data

On UNIX you can use the RSYNC command to replicate files, on Windows Server you can use RSYNC as well but you may prefer ROBOCOPY from the Windows Server Resource Kit. Usually you will upload your content to one master server and then replicate it to the other servers in the cluster.

Solutions for Session Data

Standard ASP and PHP session data is stored locally by default, leaving your session data in a local store will prevent you from implementing seamless application server fail-over in your cluster. If an application server fails, all of the local session data will be lost and your user will need to re-log in and possibly lose shopping baskets etc.

This problem is easily resolvable by implementing a shared persistent data store for the cluster. This is usually either done with a shared back-end database or a shared memory solution.

Persistence

Persistence is a feature that is required by many web applications. Once a user has interacted with a particular server all subsequent requests are sent to the same server thus persisting to that particular server. It is normally required when the session state is stored locally to the web server as opposed to a database.

What do You do if Your Application is Not Stateless?

Some applications require state to be maintained such as:

- Terminal Services
- SSH
- FTP (upload)
- SMTP (incoming)

You may also find that you are unable to modify your HTTP/HTTPS based application to handle shared session data.

For these cases, you can use persistence based on source IP address. You lose the ability to have transparent fail-over, but you do still get increased capacity and manageability. This persistence problem occurs with all load balancers and all vendors use standard methods and technologies to mitigate the issue.

Loadbalancer.org Persistence Methods

- Source IP (subnet)
- Cookie (Active or Passive)

The standard Layer 4 persistence method is source IP persistence, you can handle millions of persistent connections at Layer 4. Just modify your virtual server to be persistent if you require source IP persistence.

Cookies are a Layer 7 based persistence method that can offer more even traffic distribution and also handle any clients where the source IP address may change during the session (e.g. mega proxies).

Section D – Typical Deployment Examples

Example 1 – One-Arm DR Mode (Single Appliance)

This DR (Direct Return) mode example has one Virtual Server (VIP) with two Real Servers (RIPs). It's a straight forward deployment mode and can be used in many situations. It also offers the highest performance because return traffic passes directly from the Real Servers to the client (i.e. not via the load balancer).

Initial Network Interface Configuration

The default IP address is 192.168.2.21/24. To change this, at the console:

either login as user 'setup' and run through the network setup wizard (the wizard starts automatically):

Username: setup
Password: setup

or login as user 'root' and set the IP address & mask manually:

Username: root
Password: loadbalancer

now set the IP address and mask using the following command:

```
ip addr add <IP address>/<mask> dev eth0          e.g. ip addr add 192.168.2.1/24 dev eth0
```

N.B. If the IP address is set manually (i.e. not using the network setup wizard) then the IP address MUST be set via the web interface to make this permanent, otherwise it will be lost after a reboot

Accessing the Web User Interface (WUI)

Using HTTP : ***http://192.168.2.21:9080/lbadmin/*** (replace with your IP address)

Using HTTPS : ***https://192.168.2.21:9443/lbadmin/*** (replace with your IP address)

now login to the WUI using the following credentials:

Username: loadbalancer
Password: loadbalancer

Configuring the Load Balancer (using the WUI)

All configuration is performed via the Web User Interface.

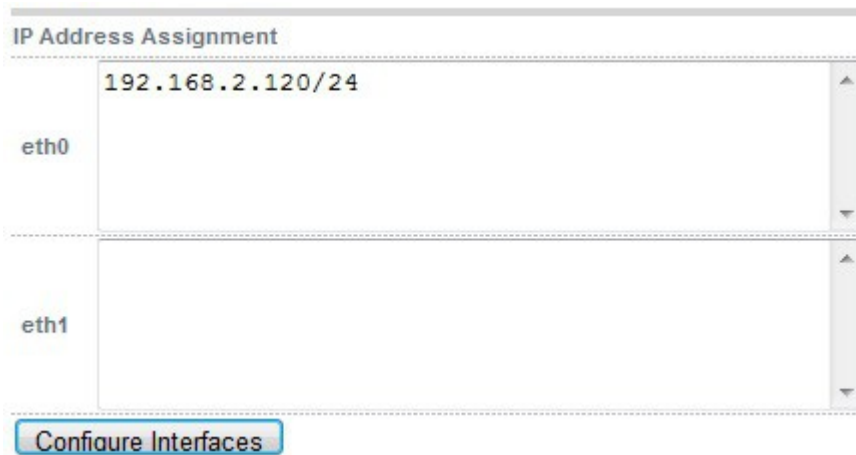
Configuration Overview

- **Configure Network Settings (WUI)** – A single Interface is needed, eth0 is normally used
- **Define the Virtual Server (WUI)** – All real (back-end) servers are accessed via this IP address
- **Define the Real Servers (WUI)** – Define the Real Servers
- **Implement the required changes to the Real Servers** – In DR mode, the ARP issue must be solved

Network Settings

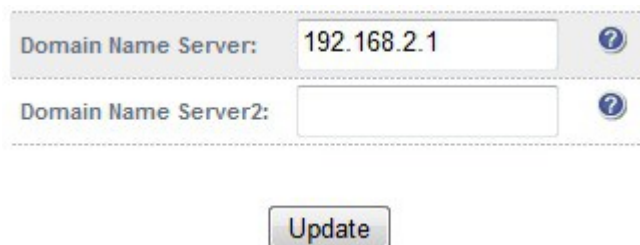
Configure the various network settings as outlined below:

- Open *Edit Configuration > Network Interface Configuration*
N.B. this step can be skipped if you used the network setup wizard



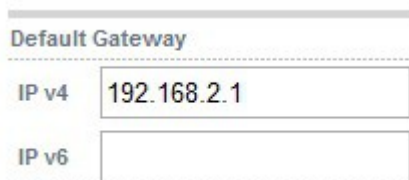
The screenshot shows a window titled "IP Address Assignment". It contains two sections for network interfaces. The first section is for "eth0" and has a text input field containing "192.168.2.120/24". The second section is for "eth1" and has an empty text input field. At the bottom of the window is a button labeled "Configure Interfaces".

- Specify the IP address & mask for eth0 (normally eth0 is used for single-arm configurations although this is not mandatory) , e.g. 192.168.2.120/24
- Click **Configure Interfaces**
- Open *Edit Configuration > DNS & Hostname*
- Specify the DNS server(s)



The screenshot shows a window for DNS and Hostname configuration. It has two input fields for "Domain Name Server". The first field contains "192.168.2.1" and the second field is empty. Both fields have a question mark icon to their right. Below the input fields is a button labeled "Update".

- Click **Update**
- Open *Edit Configuration > Routing*
N.B. this step can be skipped if you used the network setup wizard



The screenshot shows a window titled "Default Gateway". It contains two input fields. The first field is labeled "IP v4" and contains "192.168.2.1". The second field is labeled "IP v6" and is empty.

- Specify the default gateway
- Click **Configure Routing**

Virtual Server (VIP)

Next, configure the Virtual Server. This is the IP address that is presented to clients. Any packet arriving at the load balancer with that IP address and port number will be handled by the Real Servers associated with the Virtual Server.

- Use *Edit Configuration > Layer 4 Virtual Servers > Add a new Virtual Server*

Label	<input type="text" value="VIP Name"/>	?
Virtual Server IP address	<input type="text" value="192.168.2.130"/>	?
Virtual Server Ports	<input type="text" value="80"/>	?
Forwarding Method	<input type="text" value="Direct Routing"/>	?
Persistent	<input type="text" value="no"/>	?

- Enter a suitable Label (name) for the VIP
- Enter a valid IP address , e.g. 192.168.2.130
- Enter a valid port , e.g. 80
- Ensure that the Forwarding Method is set to 'Direct Routing' (*N.B. this is the default*)

Real Servers (RIP)

Each Virtual Server requires a cluster of Real Servers (back-end servers) to forward the traffic to.

- Use *Edit Configuration > Layer 4 Real Servers > Add a new Real Server*
- Next to the relevant Virtual Server, click *Add a new Real Server*

Label	<input type="text" value="RIP1"/>	?
Real Server IP Address	<input type="text" value="192.168.2.150"/>	?
Weight	<input type="text" value="1"/>	?
Minimum Connections	<input type="text" value="0"/>	?
Maximum Connections	<input type="text" value="0"/>	?

- Enter a suitable Label (name) for the RIP
- Enter a valid IP address , e.g. 192.168.2.150

N.B. A port does not need to be specified since port redirection is not possible in DR mode, therefore the port used will be the same as that configured for the VIP

- The weight defaults to 1 making Real Servers active immediately
- Leave the Minimum & Maximum Connections as 0 which means unrestricted
- Repeat for remaining Real Servers

Real Server Changes – Solve the ARP Problem

Since this example uses the one-arm DR mode load balancing method each web server requires the ARP problem to be handled:

- Each server must be configured to respond to the VIP address as well as the RIP address
- Each Windows server must have the MS Loopback Adapter installed and configured
- The MS Loopback Adapter must be configured to deal with the ARP problem



Failure to correctly configure the Real Servers to handle the ARP problem is the most common problem in DR configurations. Please refer to pages 31-46 & 82-101 for more details.

Basic Testing & Verification

A few quick checks can be performed to verify the configuration:

- Using *System Overview* , check that the VIP & RIP are shown as active (green)
- Using a browser, navigate to the VIP address , i.e. <http://192.168.2.130> to verify that you can reach the Real Servers via the Virtual Server
- Check *Reports > Layer 4 Current Connections* to ensure you client connections are reported in state 'ESTABLISHED'. if connections are in state 'SYN_RECEIVED' , this normally means that the ARP issue on the Real Servers has not been solved

Example 2 – Two-Arm NAT Mode (Clustered Pair)

This example covers the process of configuring two load balancers (as a clustered pair) in NAT mode. In this scenario, the slave's network settings must be configured first, followed by the master. This allows the master to successfully communicate with the slave and replicate settings as they are configured.

Using two appliances configured as a clustered pair is Loadbalancer.org's recommended configuration and ensures that no single point of failure is introduced.



When using two-arm NAT mode each web server has to be in the same subnet as the internal interface of the load balancer and the Real Servers' default gateway must point at an IP address on the load balancer.



By default the hardware appliance uses the serial interfaces to transmit / receive heartbeat information, so make sure that you connect the serial cable (one is included with each unit) between the master & slave, by default the Virtual Appliance is configured to use ucast for heartbeat when configured as a pair.

Initial Network Interface Configuration

Please refer to example 1.

Accessing the Web User Interface (WUI)

Please refer to example 1.

Configuring the Load Balancer (using the WUI)

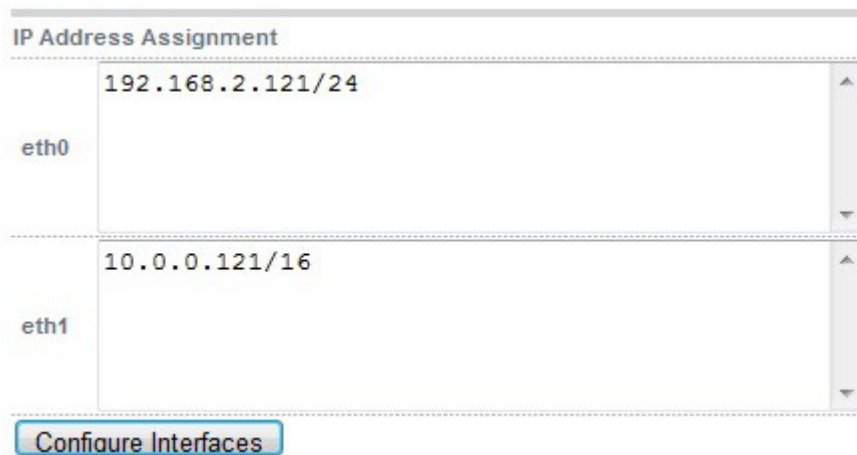
Configuration Overview

- **Configure the Slave's Network Settings** – Two Interfaces are needed, this can be either two physical interfaces such as eth0 and eth1, or one physical interface and an alias/secondary interface such as eth0:0
- **Configure the Master's Network Settings** – Two Interfaces are needed, this can be either two physical interfaces such as eth0 and eth1, or one physical interface and an alias/secondary interface such as eth0:0
- **Configure the Master & Slave Heartbeat Settings** – Set the heartbeat comms method
- **Define the Virtual Server (via the master)** – All IIS servers are accessed via this IP address
- **Define the Real Servers (via the master)** – Define the servers that make up the IIS cluster
- **Implement the required changes to the Real Servers** – In NAT mode, the IIS servers default gateway must be the load balancer

Slave Unit – Network Settings

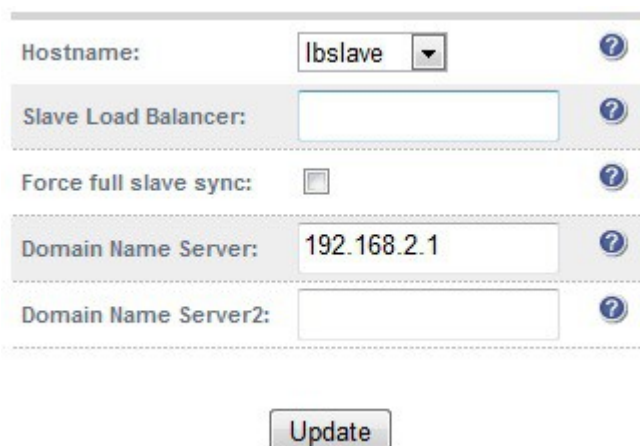
Configure the various network settings as outlined below:

- Open *Edit Configuration > Network Interface Configuration*
N.B. This step can be skipped if you used the network setup wizard



The screenshot shows a configuration window titled "IP Address Assignment". It contains two rows of configuration for network interfaces. The first row is for "eth0" and shows the IP address "192.168.2.121/24". The second row is for "eth1" and shows the IP address "10.0.0.121/16". At the bottom of the window is a button labeled "Configure Interfaces".

- Specify the IP address & mask for eth0 (normally eth0 is configured as the internal interface, although this is not mandatory) , e.g. 192.168.2.121/24
- Specify the IP address & mask for eth1 (normally eth1 is configured as the external interface, although this is not mandatory) , e.g. 10.0.0.121/16
- Click **Configure Interfaces**
- Open *Edit Configuration > Hostname & DNS*



The screenshot shows a configuration window titled "Hostname & DNS". It contains several fields for configuration. The "Hostname:" field has a dropdown menu with "lbslave" selected. The "Slave Load Balancer:" field is empty. The "Force full slave sync:" field has an unchecked checkbox. The "Domain Name Server:" field has the value "192.168.2.1". The "Domain Name Server2:" field is empty. At the bottom of the window is a button labeled "Update".

- Set the hostname drop-down to '**lbslave**'
- Specify the DNS server(s) , e.g. 192.168.2.1
- Click **Update**
- Open *Edit Configuration > Routing*
N.B. This step can be skipped if you used the network setup wizard

Default Gateway	
IP v4	192.168.2.1
IP v6	

- Specify the default gateway , e.g. 192.168.2.1
- Click **Configure Routing**

Master Unit – Network Settings

Once the slave is configured, continue with the master unit.

- On the master, open *Edit Configuration > Network Interface Configuration*
N.B. This step can be skipped if you used the network setup wizard

IP Address Assignment	
eth0	192.168.2.120/24
eth1	10.0.0.120/16

Configure Interfaces

- Specify the IP address & mask for eth0 (normally eth0 is configured as the internal interface, although this is not mandatory) , e.g. 192.168.2.120/24
- Specify the IP address & mask for eth1 (normally eth1 is configured as the external interface, although this is not mandatory) , e.g. 10.0.0.120/16
- Click **Configure Interfaces**
- Open *Edit Configuration > Hostname & DNS*

Hostname:	lbmaster ▼	?
Slave Load Balancer:	192.168.2.121	?
Force full slave sync:	<input type="checkbox"/>	?
Domain Name Server:	192.168.2.1	?
Domain Name Server2:		?

Update

- Using the hostname drop-down, ensure that the hostname is set to '**lbmaster**'
- Specify the Slave Load balancer's IP address , e.g. 192.168.2.121
- Specify the DNS server(s) , e.g. 192.168.2.1
- Click **Update**
- Open *Edit Configuration > Routing*
N.B. This step can be skipped if you used the network setup wizard
- Specify the default gateway










Default Gateway	
IP v4	192.168.2.1
IP v6	

- Click **Configure Routing**

Master & Slave – Heartbeat Settings

- Open *Edit Configuration > Modify Heartbeat Configuration*

EDIT CONFIGURATION > MODIFY HEARTBEAT CONFIGURATION

Serial	<input checked="" type="checkbox"/>	
Unicast	<input type="checkbox"/>	
Broadcast	None ▾	
UDP Port for broadcast & unicast	6694	
Keepalive	3	
Deadtime	10	
Warntime	5	
Ping node		
Automatic Fail-back	<input checked="" type="checkbox"/>	






Modify Heartbeat configuration

- Set the heartbeat communications method as required. For a hardware load balancer the default is serial, for a VA the default is unicast (i.e. via the network)

Virtual Server (VIP)

Next, configure the Virtual Server. This is the IP address that is presented to clients. Any packet arriving at the load balancer with that IP address / port number will be handled by the Real Servers associated with the Virtual Server.

- Use *Edit Configuration > Layer 4 Virtual Servers > Add a new Virtual Server*

Label	VIP Name	
Virtual Server IP address	192.168.2.130	
Virtual Server Ports	80	
Forwarding Method	NAT ▾	
Persistent	no ▾	

- Enter a suitable label (name) for the VIP
- Enter a valid IP address , e.g. 192.168.2.130
- Enter a valid port, e.g. 80
- Ensure that the Forwarding Method is set to 'NAT'

Real Servers (RIP)

Each Virtual Server requires a cluster of Real Servers (back-end servers) to forward the traffic to.

- Use *Edit Configuration > Layer 4 Real Servers > Add a new Real Server*
- Next to the relevant Virtual Server, click *Add a new Real Server*

Label	<input type="text" value="RIP1"/>	?
Real Server IP Address	<input type="text" value="192.168.2.150"/>	?
Weight	<input type="text" value="1"/>	?
Minimum Connections	<input type="text" value="0"/>	?
Maximum Connections	<input type="text" value="0"/>	?

- Enter a valid IP address , e.g. 192.168.2.150
- Enter a valid port , e.g. 80
- The weight defaults to 1 making Real Servers active immediately
- Leave the Minimum & Maximum connections as 0 which means unrestricted
- Repeat for the remaining Real Servers

Real Server Changes – Set the Default Gateway

As we are using NAT mode, each web servers' default gateway must be changed to be the load balancer. When using a clustered pair, you must define an additional floating IP for this purpose. Then, if failover is required, the same IP will also be brought up on the slave.

To add a floating IP, use *Edit Configuration > Floating IP's* enter the IP address that you'd like to use for the default gateway, then click **update**.

EDIT CONFIGURATION > ADD NEW FLOATING IP

<input type="text" value="192.168.2.254"/>
<input type="button" value="Update"/>

Verify the Slave Configuration

To verify that the new VIP & RIP have been replicated correctly, open the WUI on the slave and goto *Edit Configuration > Layer 4 Virtual Servers & Edit Configuration > Layer 4 Real Servers* and check that your configuration appears there also.

If not, double check that both units are configured correctly and that the IP address for the slave defined on the master is correct. Then on the master open *Edit Configuration > Hostname & DNS*, check 'Force full slave sync' and click **update**, this will force all setting to be copied from the master to the slave, then check again.

Restart Heartbeat

Now restart heartbeat on both units using *Maintenance > Restart Services > Restart Heartbeat*. This ensures that heartbeat starts cleanly and is communicating between the two devices correctly. Once the restart is complete, the status of each appliance should be as follows:

On the Master unit:



On the Slave unit:



Also, on the active unit (the master) you should see the floating IP for the corresponding VIP displayed as follows under *View Configuration > Network Configuration*

```
inet 192.168.2.120/24 brd 192.168.2.255 scope global eth0
inet 10.0.0.120/16 brd 10.0.255.255 scope global eth1
inet 192.168.2.130/24 brd 192.168.2.255 scope global secondary eth0
```

- the first two lines show the interface IP addresses (eth0 & eth1)
- the last line shows the active floating IP address (VIP)

Basic Testing & Verification

A few quick checks can be performed to verify the configuration:

- Using *System Overview*, check that the VIP & RIPs are shown as active (green)
- Using a browser, navigate to the VIP address, i.e. <http://192.168.2.130> to verify that you can reach the Real Servers via the Virtual Server
- Check *Reports > Layer 4 Current Connections* to ensure you client connections are reported in state 'ESTABLISHED'. If not, double-check that you have set the default gateway on all Real Servers to be an IP address on the load balancer

Example 3 – One-Arm SNAT Mode With SSL – HAProxy & Pound (Single Unit)

This example uses HAProxy and Pound at layer 7. Pound is used to terminate the SSL connection on the load balancer. Pound then passes traffic to an HAProxy VIP / RIP cluster. HAProxy does not offer the raw throughput of layer 4 , but is still a high performance solution that is appropriate in many situations.

In this example it's assumed that the Real Server application has not been designed to track & share session details between Real Servers. Therefore, persistence will be enabled on the load balancer to ensure that clients connect to the same real (back-end) sever on each subsequent connection (within the persistence timeout window). If persistence is not configured then new connections may get distributed to a different Real Server which in this case would result in failure of the application.



Because HAProxy is a full proxy, any server in the cluster can be on any accessible subnet including across the Internet or WAN.



In this mode, no changes are required to the real (back-end) servers.



We generally recommend that SSL is terminated on the real serves rather than on the load balancer. This ensures that the SSL load is distributed and also ensures scalability.

Initial Network Interface Configuration

Please refer to example 1.

Accessing the Web User Interface (WUI)

Please refer to example 1.

Configuring the Load Balancer (using the WUI)

Configuration Overview

- **Configure Network Settings (WUI)** – A single Interface is needed, eth0 is normally used
- **Define the Virtual Server (WUI)** – All real (back-end) servers are accessed via this IP address
- **Define the Real Servers (WUI)** – Define the Real Servers
- **Configure SSL Termination** – Configure Pound provide SSL

Network Settings

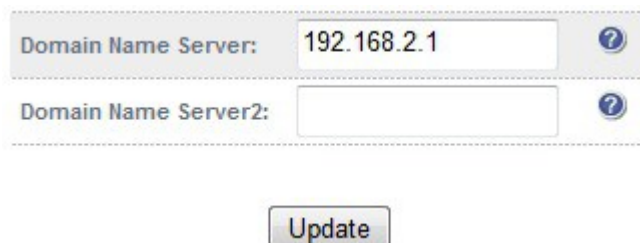
Configure the various network settings as outlined below:

- Open *Edit Configuration > Network Interface Configuration*
N.B. This step can be skipped if you used the network setup wizard



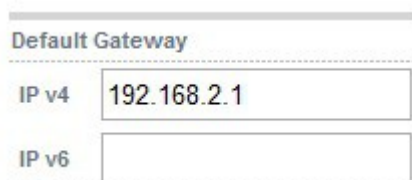
The screenshot shows a window titled "IP Address Assignment". It contains two sections for network interfaces. The first section is for "eth0" and has a text input field containing "192.168.2.120/24". The second section is for "eth1" and has an empty text input field. At the bottom of the window is a button labeled "Configure Interfaces".

- Specify the IP address & mask for eth0 (normally eth0 is used for single-arm configurations although this is not mandatory) , e.g. 192.168.2.120/24
- Click **Configure Interfaces**
- Open *Edit Configuration > DNS & Hostname*
- Specify the DNS server(s)



The screenshot shows a window titled "DNS & Hostname". It contains two input fields for domain name servers. The first field is labeled "Domain Name Server:" and contains the value "192.168.2.1". The second field is labeled "Domain Name Server2:" and is empty. Below these fields is a button labeled "Update".

- Click **Update**
- Open *Edit Configuration > Routing*
N.B. This step can be skipped if you used the network setup wizard



The screenshot shows a window titled "Default Gateway". It contains two input fields. The first field is labeled "IP v4" and contains the value "192.168.2.1". The second field is labeled "IP v6" and is empty.

- Specify the default gateway
- Click **Configure Routing**

Virtual Server (VIP)

Next, configure the Virtual Server. This is the IP address that is presented to clients. Any packet arriving at the load balancer with that IP address and port number will be handled by the Real Servers associated with the Virtual Server.

- Use *Edit Configuration > Layer 7 Virtual Servers > Add a new Virtual Server*

Label	<input type="text" value="VIP Name"/>	?
Virtual Server IP address	<input type="text" value="192.168.2.130"/>	?
Virtual Server Ports	<input type="text" value="80"/>	?
Persistence mode	<input type="text" value="HTTP Cookie"/>	?
Fallback Server	<input type="text" value="127.0.0.1:9081"/>	?
<input type="button" value="Update"/>		

- Enter a suitable Label (name) for the VIP
- Enter a valid IP address , e.g. 192.168.2.130
- Enter a valid port , e.g. 80
- Set Persistence mode to '**HTTP Cookie**'
- Restart HAProxy to apply the new settings using the link provided in the yellow box

Real Servers (RIP)

Each Virtual Server requires a cluster of Real Servers (back-end servers) to forward the traffic to.

- Use *Edit Configuration > Layer 4 Real Servers > Add a new Real Server*
- Next to the relevant Virtual Server, click *Add a new Real Server*

Label	<input type="text" value="RIP1"/>	?
Real Server IP Address	<input type="text" value="192.168.2.150"/>	?
Real Server Port	<input type="text" value="80"/>	?
Weight	<input type="text" value="1"/>	?
<input type="button" value="Update"/>		

- Enter a suitable Label (name) for the RIP
- Enter a valid IP address , e.g. 192.168.2.150

N.B. In this mode it's possible to have a different port for the RIP than was configured for the VIP, in this example both are the same

- Enter a valid port , e.g. 80
- The weight defaults to 1 making Real Servers active immediately
- Repeat for the remaining Real Servers
- Reload HAProxy to apply the new settings using the link provided in the yellow box



The label set for the VIP is used as the name for the HTTP session cookie that is set for use with cookie persistence.

SSL Termination

Typically, a Pound VIP is configured on port 443 using the same IP address as the Layer 7 VIP created previously. This allows a single IP address to be used.

- Use *Edit Configuration > SSL Termination Virtual Servers*
- Click *Add a New Virtual Server*

Label	<input type="text" value="VIP Name"/>	?
Virtual Server IP address	<input type="text" value="192.168.2.130"/>	?
Virtual Server Port	<input type="text" value="443"/>	?
Backend Virtual Server IP Address	<input type="text" value="192.168.2.130"/>	?
Backend Virtual Server Port	<input type="text" value="80"/>	?
Ciphers to use	<input type="text"/>	?

- Enter a suitable Label (name) for the VIP
- Enter the same IP address used for the layer 7 VIP , i.e. 192.168.2.130
- Set the port to 443
- Now define the backend server, as the layer 7 VIP , i.e. 192.168.2.130 , port 80
- Click Update
- Restart Pound to apply the new settings using the link provided in the yellow box

When creating the SSL Virtual Server, a default self-signed certificate is used. This is ideal for testing but needs to be replaced for live deployments.



For more detailed information on SSL termination, Pound configuration and using Certificates please refer to the SSL Certificates & Pound topic on page 113.

Basic Testing & Verification

A few quick checks can be performed to verify the configuration:

- Using *System Overview* , check that the VIP & RIP are shown as active (green)
- Using a browser, navigate to the VIP address , i.e. **http://192.168.2.130** to verify that you can reach the Real Servers via the Virtual Server using HTTP
- Using a browser, navigate to the Pound SSL VIP address , i.e. **https://192.168.2.130** to verify that you can reach the Real Servers via the Virtual Server using HTTPS
- check / verify the certificate details

Section E – Detailed Configuration Information

Appliance Configuration Methods

The load balancer is normally configured using a browser and the built-in WUI. However it's also possible to setup the appliance using the text based Links browser from the console or a terminal session as outlined below.

Console Access

To access the console, connect a monitor and keyboard to the load balancer, power up and you'll be presented with a login prompt.

Log in to the console:

Username: root
Password: loadbalancer

For configuration at the console using links, type:

```
links 127.0.0.1:9080/lbadmin
```

This will start Links on the local machine and bring up the text based administration interface.

Log in to links:

Username: loadbalancer
Password: loadbalancer

Use the 'down' cursor key to select a link and the 'right' cursor key to follow a link.



It is recommended to change the default password. To do this type passwd at the console or a terminal window to change the default root password. You can also run the 'lbsecure' lockdown script that also requests new passwords to be defined for user 'root' and the WUI user 'loadbalancer'. For more details please refer to page 156.

Console Access via a Serial Cable

By default the hardware is shipped with the serial port configured for heartbeat and therefore can't be used for a serial console connection. However if this is your preferred access method then simply go to *Edit Configuration > Heartbeat Configuration* and change the heartbeat to use the network (i.e. ucast or bcast) rather than the serial option. This will automatically activate a console on the serial port.

Keyboard Layout

to change the keyboard locale edit /etc/sysconfig/keyboard

e.g. to change from a UK to a USA layout replace KEYTABLE="uk" with KEYTABLE="us" , then re-boot.

Remote Configuration Methods

Remote configuration is recommended in most cases, but be very cautious if you are changing the network configuration. Make sure you have access to the console in case you make a mistake. You can access each load balancer, lbmaster & lbslave remotely via their own IP address using the following tools:

- | | |
|---------------------|--------------------------|
| • HTTP or HTTPS | Web based Administration |
| • OpenSSH or PuTTY | Secure Shell Access |
| • OpenSCP or WinSCP | Secure File Transfer |

The default IP address is 192.168.2.21/24. To change this, at the console use:

```
ip addr add <IP address>/<mask> dev eth0
```

N.B. This is temporary, the IP address MUST be set via the WUI to make this permanent

For SSH and SCP access use the following credentials:

Username: root
Password: loadbalancer

The WUI uses a different set of user accounts and passwords based on the .htaccess files. For HTTP & HTTPS access to the WUI, use the following URLs:

Using HTTP : ***http://192.168.2.21:9080/lbadmin/*** (replace with your IP address)

Using HTTPS : ***https://192.168.2.21:9443/lbadmin/*** (replace with your IP address)

access the WUI using the following credentials:

Username: loadbalancer
Password: loadbalancer



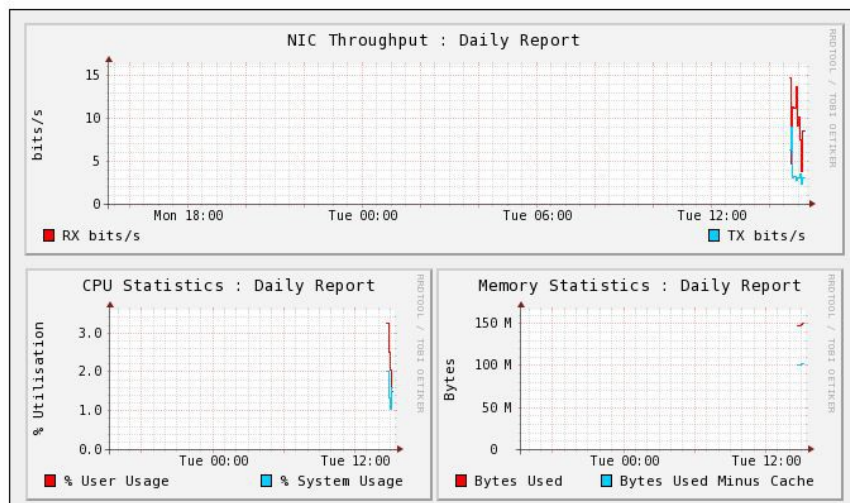
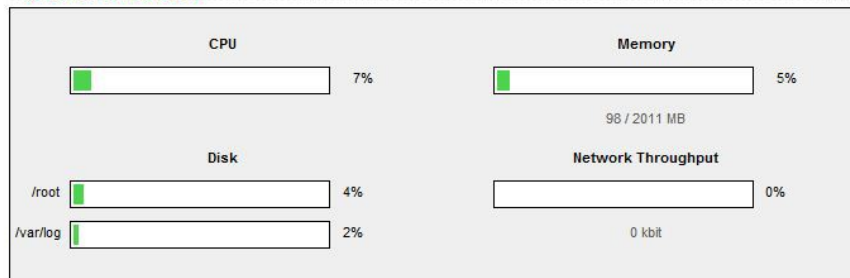
A number of interoperability issues have been found with various versions of IE. The WUI has been tested and verified using both Chrome and Firefox.

Once logged in, you will be presented with the following screen:

Master | Slave Active | Passive Link

- System Overview
- View Configuration
- Edit Configuration
- Maintenance
- Reports
- Logs
- Support

SYSTEM OVERVIEW



You can then select an option from one of the main menus. The menu options are as follows:

- **System Overview** : Quickly view system resources, configured VIPs & RIPs and throughput stats
- **View Configuration** : View the network & load balancer configuration
- **Edit Configuration** : Set up or modify the physical and virtual configuration
- **Maintenance** : Take servers offline or bring them back online
- **Reports**: View the actual live status of the load balancer or historical statistics
- **Logs**: View Ldirectord, Lbadmin, Heartbeat, HAProxy and Pound (SSL)
- **Support** : Create a support download bundle and contact loadbalancer.org support



The first time you access the web interface you will be prompted to run the configuration wizard. If you prefer to configure manually, simply cancel the wizard.

Full Root Access

One of the great advantages of the Loadbalancer.org appliance is that you have full root access and a complete development environment with all of the usual tools you would expect for customizing the installation for your environment.

The following configuration files may be useful:

Physical configuration:	/etc/sysconfig/network-scripts/ifcfg-eth*
Firewall configuration:	/etc/rc.d/rc.firewall
Layer 4 configuration:	/etc/ha.d/conf/loadbalancer.cf
Layer 7 HAProxy configuration	/etc/haproxy/haproxy.cfg
Pound SSL configuration	/etc/pound/pound.cfg
SSL Certificates	/etc/pound/certs
Fail-over (heartbeat) configuration:	/etc/ha.d/ha.cf

Network Configuration

IP Addresses

New to version 7.x is full IPv6 support. This allows IPv6 services to be configured in the same way as IPv4 services.

Depending on the type of appliance you are using you may have either 2 or 4 network ports. You can manually change the physical IP addresses on the load balancer using *Edit Configuration > Network Interface configuration*.

Normally *eth0* is used as the internal interface and *eth1* is used as the external interface. However, unlike other appliances on the market you can use any interface for any purpose.

In a standard one-arm configuration you would just need to configure *eth0*, the netmask and the default gateway.

Setting IP Addresses

To set the IP address, in the WUI open *Edit Configuration > Network Interface Configuration* as shown below:

EDIT CONFIGURATION > NETWORK INTERFACE CONFIGURATION

Bonding

Bond eth0 & eth1 as bond0: ☐ [?](#) [Bond Interfaces](#)

VLAN

Interface: [?](#) [Add VLAN](#)

VLAN ID: [?](#)

IP Address Assignment

eth0	<input type="text" value="192.168.2.21/24"/>
eth1	<input type="text"/>

[Configure Interfaces](#)



NOTE: If you already have Virtual Servers defined when making changes to the network configuration, you should verify that your Virtual Servers are still up and working correctly after making the changes.

Setting Multiple Addresses

Multiple addresses can be configured per interface as shown below:

EDIT CONFIGURATION > NETWORK INTERFACE CONFIGURATION

Bonding		
Bond eth0 & eth1 as bond0:	<input type="checkbox"/>	?
Bond eth2 & eth3 as bond1:	<input type="checkbox"/>	?
Bond Interfaces		
VLAN		
Interface:	<input type="text" value="eth0"/>	?
VLAN ID:	<input type="text" value="1"/>	?
Add VLAN		
IP Address Assignment		
eth0	<div>192.168.2.120/24 192.168.8.120/24</div>	
eth1	<div>10.20.1.1/16 fde6:d14c:3089:1::360/64</div>	
Configure Interfaces		

Physical Interfaces

The Enterprise R16, and Enterprise models have 2 physical interfaces and the MAX and 10G models have 4 physical interfaces. If multiple logical interfaces are required, these can be added as described in the section 'Setting Multiple Addresses' above. If multiple cables must be connected, an external switch can be used.

Typically, the only reason for using all 4 interfaces is when bonding (e.g. 802.3ad) is required in a 2-arm SNAT mode (layer 7) or 2-arm NAT mode (layer 4) configuration.

Configuring Bonding

- In the WUI, open *Edit Configuration > Network Configuration*
- If you want to bond eth0 and eth1, check the box named **Bond eth0 & eth1 as bond0**
- Click **Bond Interfaces**
- The eth0 and eth1 fields will be replaced with bond0
- Enter the IP address for bond0 and click **Configure Interfaces**

The screenshot displays a network configuration interface with three main sections:

- Bonding:** Contains two rows. The first row, "Bond eth0 & eth1 as bond0:", has a checked checkbox and a help icon. The second row, "Bond eth2 & eth3 as bond1:", has an unchecked checkbox and a help icon. A "Bond Interfaces" button is located to the right of these rows.
- VLAN:** Contains two rows. The first row, "Interface:", has a dropdown menu set to "bond0" and a help icon. The second row, "VLAN ID:", has a text input field containing "1" and a help icon. An "Add VLAN" button is located to the right of these rows.
- IP Address Assignment:** A table with three rows. The first row is for "bond0" and contains the IP address "192.168.2.74/24". The second row is for "eth2" and is empty. The third row is for "eth3" and is empty. A "Configure Interfaces" button is located at the bottom of the table.

By default, the bond is configured for high-availability. This can be changed by editing `/etc/modprobe.conf` as described in the following section.



NOTE: If you have a master and slave configured as a high availability pair, make sure you configure bonding in the same way on both units. Failure to do this may result in heartbeat related issues.

Bonding Configuration Modes

Ideally you want to remove any single point of failure in your network. You can achieve this with a cross-wired switch environment. Every single server including the load balancers is cross wired into two switch stacks. Then, if a network switch fails the servers & load balancers will activate the connection to the second switch.

Loadbalancer.org appliances support this using the standard Linux bonding driver. Once you have set up the load balancer using a single network card and are happy with the configuration then you can set up bonding using *Edit Configuration > Network Interface Configuration*.

If required, you can change the bonding mode in the */etc/modprobe.conf* file:

Example 1: Bonding for Bandwidth

```
alias bond0 bonding
options bond0 miimon=100 mode=0
```

Are you really doing 1Gb/s+?

Example 2: Bonding for High-Availability (the Default Mode)

```
alias bond0 bonding
options bond0 miimon=100 mode=1
```

This works with any switch.

Example 3: Bonding for High-Availability & Bandwidth

```
alias bond0 bonding
options bond0 miimon=100 mode=4
```

This requires the ports on the switch to be configured as a TRUNK with 802.3ad support.



If your Real Servers, ESX hosts etc. support network bonding using Broadcom's SLB (Smart Load Balancing), this can cause issues in Layer 4 DR mode if older drivers are used. We have successfully tested SLB (Auto Fallback Disable) with driver version 15.2.0.5. Therefore at least this version is recommended.

Configuring VLANs

Native 8021q VLAN support can be enabled to load balance clusters on multiple VLANs.

In access mode, the switch port is dedicated to one VLAN. The switch handles all the tagging and detagging of frames – the station connected to the port does not need to be configured for the VLAN at all. In trunk mode, the switch passes on the raw VLAN frames, and the station must be configured to handle them. Trunk mode is usually used to connect two VLAN-carrying switches, or to connect a server or router to a switch.

If the load balancer is connected to an access mode switch port there is no VLAN configuration needed. If the load balancer is connected to a trunk port, then all the required VLANs will need to be configured under Network Config.

To configure a VLAN:

- In the WUI, open *Edit Configuration > Network Configuration*
- In the VLAN section select the required interface (e.g. eth0)
- Enter the VLAN ID (e.g. 100)
- Click **Add VLAN**
- An extra IP Address Assignment field named eth0.100 will be created as shown below. The required IP address should be entered in this field

IP Address Assignment	
eth0	192.168.1.1/24
eth0.100	192.168.100.1/24
eth1	

Delete eth0.100

Configure Interfaces

- Click **Configure Interfaces**

To delete the VLAN definition, click the appropriate **Delete** button



If you have a clustered pair, don't forget to configure any VLANs on the slave as these will not be replicated / created automatically.

Default Gateway & Static Routes

To set the default gateway for IPv4 and IPv6:

- In the WUI, open *Edit Configuration > Routing*
- In the Default Gateway section configure the IP addresses as shown below:

To configure static routes:

- In the WUI, open *Edit Configuration > Routing*
- In the Static Routes section configure the subnets / gateways as shown below:

EDIT CONFIGURATION > ROUTING

Default Gateway

IP v4	<input type="text" value="192.168.64.1"/>
IP v6	<input type="text"/>

Static Routes

Subnet	<input type="text" value="10.10.0.0/16"/>	via gateway	<input type="text" value="192.168.110.252"/>
Subnet	<input type="text"/>	via gateway	<input type="text"/>
Subnet	<input type="text"/>	via gateway	<input type="text"/>
Subnet	<input type="text"/>	via gateway	<input type="text"/>

Configure Routing

- Click **Configure Routing**

Hostname & DNS Configuration

For a single appliance, use the default setting of 'lbmaster'. For a clustered pair, the hostname of the slave unit should be changed to 'lbslave'. To change this, use the Hostname drop-down as shown below.

To set the Hostname & DNS servers:

- In the WUI, open *Edit Configuration > Hostname & DNS*
- Use the drop-down to select the appropriate hostname
- Specify the DNS servers using the **Domain Name Server** and **Domain Name Server2** fields.
- Click **Update**

EDIT CONFIGURATION > HOSTNAME & DNS

Hostname:	lbmaster ▼	?
Slave Load Balancer:	192.168.2.121	?
Force full slave sync:	<input type="checkbox"/>	?
Domain Name Server:	192.168.2.1	?
Domain Name Server2:		?

Update

N.B. The 'Slave Load Balancer' field is used to specify the IP address of the slave unit in a clustered pair. The Force full sync check-box enables the master units config to be copied to the slave – this occurs automatically for a correctly configured pair.

Advanced DR Considerations

The most important consideration with DR mode is how to handle the ARP problem.



The ARP problem only effects layer 4 DR (Direct Return) mode VIPs and therefore it is only necessary to implement the changes to the Real Servers described below when using this mode.

What Is the ARP Problem?

DR mode works by changing the MAC address of the inbound packets to match the real server selected by the load balancing algorithm. The destination IP address remains unchanged and therefore each real server must respond to both its own IP address and also the Virtual Server IP address (VIP). It's important that Real Servers do not 'fight' with the load balancer for control of the shared VIP. If they do then requests could be sent directly to the real servers rather than hitting the load balancer VIP as intended.

- You only need to resolve the ARP issue on the Real Servers when you are using the default DR (Direct Routing) load balancing method or IPIP (TUN or IP encapsulation)
- Real servers must not respond to ARP requests for the VIP
- The application running on each real sever must respond to both RIP address and the VIP address

Detecting the ARP Problem

You can use *Reports > Layer 4 Current Connections* to check whether the ARP problem has been solved. If not, the connection state will be SYN_RECV as shown below when a client connection to the VIP is attempted:

REPORTS > LAYER 4 CURRENT CONNECTIONS

```
IPVS connection entries
pro expire state      source          virtual         destination
TCP 00:51 SYN_RECV    192.168.2.7:64763 192.168.2.109:80 192.168.2.99:80
```

Solving for Linux – Method 1 (using iptables)

You can use iptables (netfilter) on each Real Server to re-direct incoming packets destined for the Virtual Server IP address. To make this permanent, simply add the command to an appropriate start-up script such as */etc/rc.local*. If the Real Server is serving multiple VIPs, add additional iptables rules for each VIP.

```
iptables -t nat -A PREROUTING -p tcp -d <VIP> -j REDIRECT
```

e.g.

```
iptables -t nat -A PREROUTING -p tcp -d 10.0.0.21 -j REDIRECT
```

(Change the IP address to be the same as your Virtual Server)

This means redirect any incoming packets destined for 10.0.0.21 (the Virtual Server) locally, i.e. to the primary address of the incoming interface on the Real Server.



Method 1 may not always be appropriate if you're using IP-based virtual hosting on your web server. This is because the iptables rule above redirects incoming packets to the primary address of the incoming interface on the web server rather than any of the virtual hosts that are configured. Where this is an issue, use method 2 below instead.



Method 1 does not work with IPv6 Virtual Servers, use method 2 below instead.

Solving for Linux – Method 2 (using arp_ignore sysctl values)

This is the preferred method as it supports both IPv4 and IPv6. Each Real Server needs the loopback adapter to be configured with the Virtual Servers IP address. This address must not respond to ARP requests and the web server also needs to be configured to respond to this address. To set this up follow steps 1-3 below.

Step 1 : re-configure ARP on the Real Servers (this step can be skipped for IPv6 Virtual Servers)

To do this add the following lines to /etc/sysctl.conf:

```
net.ipv4.conf.all.arp_ignore=1
net.ipv4.conf.eth0.arp_ignore=1
net.ipv4.conf.eth1.arp_ignore=1
net.ipv4.conf.all.arp_announce=2
net.ipv4.conf.eth0.arp_announce=2
net.ipv4.conf.eth1.arp_announce=2
```

Step 2 : apply these settings

Either reboot the Real Server or run the following command to apply these settings:

```
/sbin/sysctl -p
```

Step 3 : add the Virtual Servers IP address to the loopback adapter

Run the following command for each VIP. To make this permanent, simply add the command to an appropriate startup script such as /etc/rc.local.

```
ip addr add dev lo <IPv4-VIP>/32
```

for IPv6 addresses use:

```
ip addr add dev lo <IPv6-VIP>/128
```

N.B. Steps 1 & 2 can be replaced by writing directly to the required files (temporary until the next reboot) :

```
echo 1 > /proc/sys/net/ipv4/conf/all/arp_ignore
echo 1 > /proc/sys/net/ipv4/conf/eth0/arp_ignore
echo 1 > /proc/sys/net/ipv4/conf/eth1/arp_ignore
echo 2 > /proc/sys/net/ipv4/conf/all/arp_announce
echo 2 > /proc/sys/net/ipv4/conf/eth0/arp_announce
echo 2 > /proc/sys/net/ipv4/conf/eth1/arp_announce
```

Solving for Solaris

With Solaris the loopback interface does not respond to ARP requests so you just add your VIPs to it.

```
ifconfig lo0:1 plumb  
ifconfig lo0:1 VIP netmask 255.255.255.255 up
```

You will need add this to your start up scripts for your server.

Solving for Mac OS X or BSD

OS X is BSDish, so you need to use BSDish syntax:

```
ifconfig lo0 alias VIP netmask 255.255.255.255 -arp up
```

You will need add this to your start up scripts for your server.



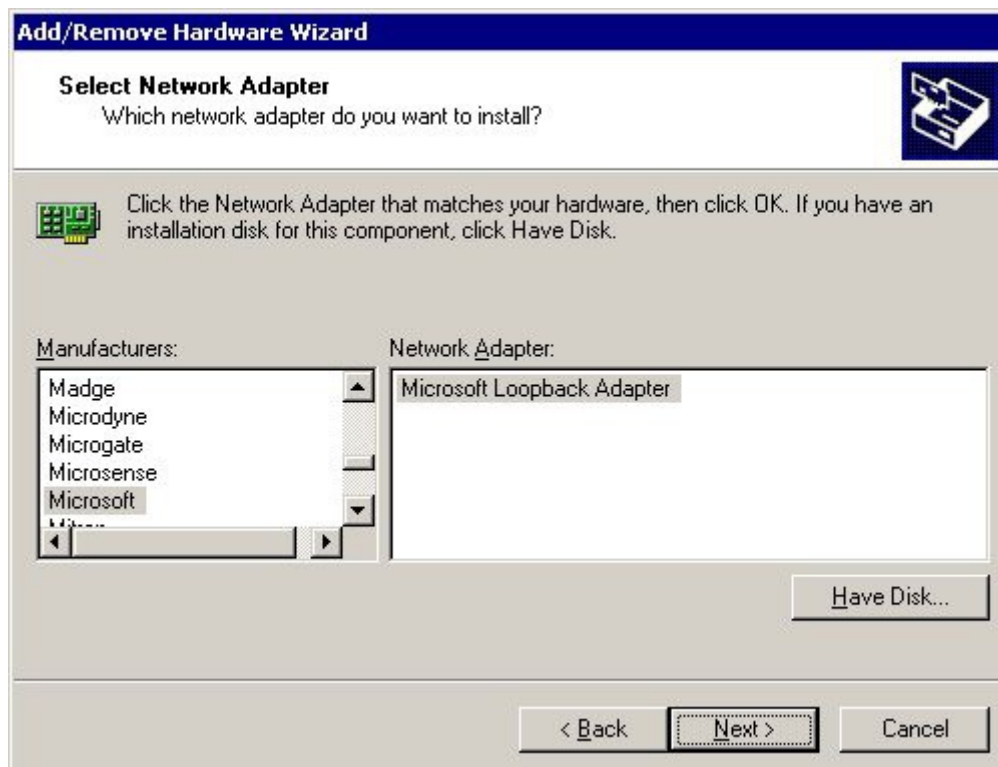
Failure to correctly configure the Real Servers to handle the ARP problem is the most common mistake in DR mode configurations.

Resolving ARP issues for Windows server 2000 (applies to DR mode only)

Windows 2000 Server supports the direct routing (DR) method through the use of the MS loopback adapter to handle the traffic. The IP address on the loopback adapter must be set to be the same as the Virtual Servers IP address (VIP). If the real server is included in multiple VIPs, you can add additional IP addresses to the loopback adapter that correspond to each VIP.

Step 1 – Install the Microsoft loopback adapter

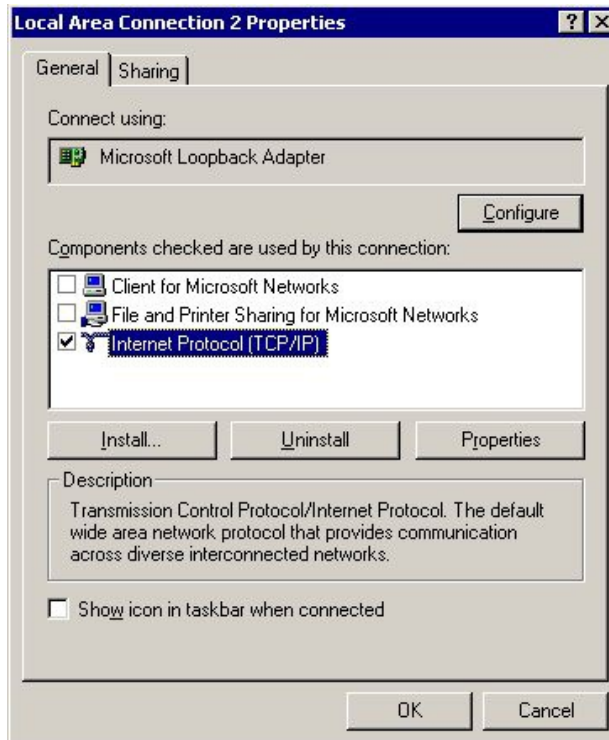
1. Open the Control Panel and double-click **Add/Remove Hardware**
2. Once the Hardware Wizard opens, click **Next**
3. Select **Add/Troubleshoot a device**, click **Next**
4. Once the device list appears, select **Add a new device** at the top of the list, click **Next**
5. Select **No, I want to select the hardware from a list**, click **Next**
6. Scroll down the list and select **Network Adapters**, click **Next**
7. Select **Microsoft & Microsoft Loopback Adapter**, click **Next** as shown below



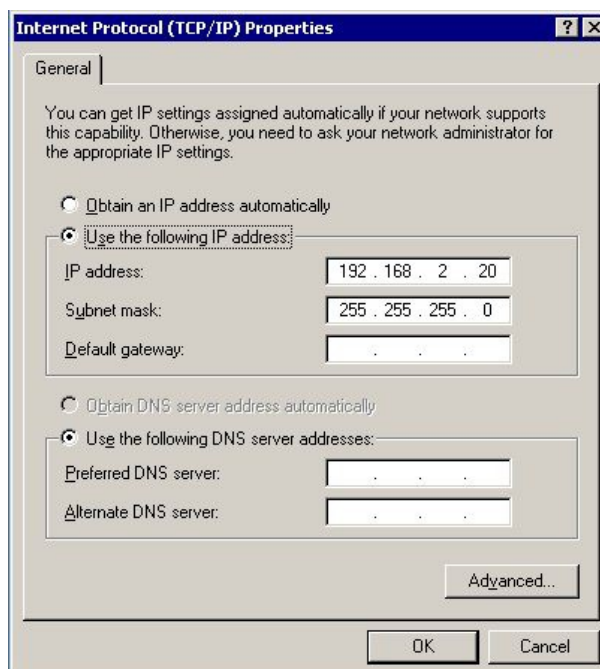
8. Click **Next** to start the installation, when complete click **Finish**

Step 2 – Configure the loopback adapter

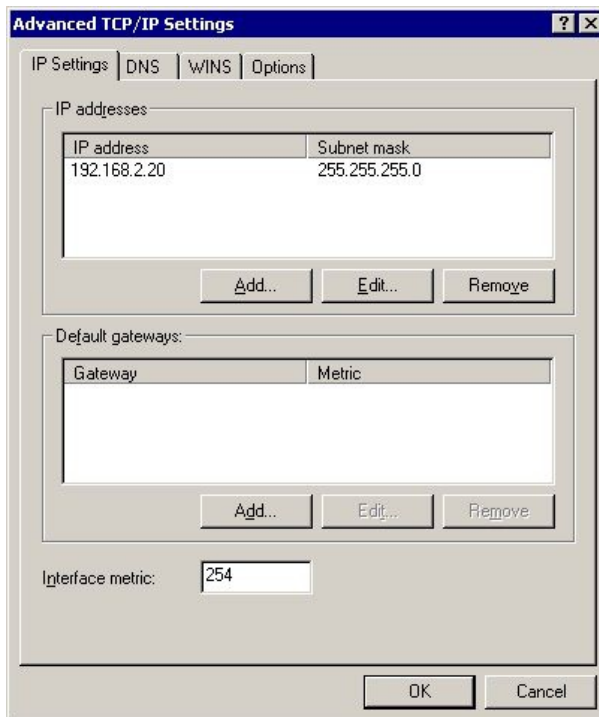
1. Open the Control Panel and double-click **Network and Dial-up Connections**
2. Right-click the new loopback adapter and select **Properties**
3. Un-check all items except **Internet Protocol (TCP/IP)** as shown below



4. Select **Internet Protocol (TCP/IP)**, click **Properties** and configure the IP address and mask to be the same as the Virtual Server IP address (VIP), e.g. 192.168.2.20/24 as shown below



- Click **Advanced** and change the **Interface metric** to 254 as shown below, this prevents the adapter responding to ARP requests for the VIP address



- Click **OK** on Advanced Settings, TCP/IP Properties and Connection Properties to save and apply the new settings
- Repeat the above steps for all other Windows 2000 real servers

Resolving ARP issues for Windows server 2003 (applies to DR mode only)

Windows server supports the direct routing (DR) method through the use of the MS loopback adapter to handle the traffic. The IP address on the loopback adapter must be set to be the same as the Virtual Servers IP address (VIP). If the real server is included in multiple VIPs, you can add additional IP addresses to the loopback adapter that correspond to each VIP.

Step 1 – Install the Microsoft loopback adapter

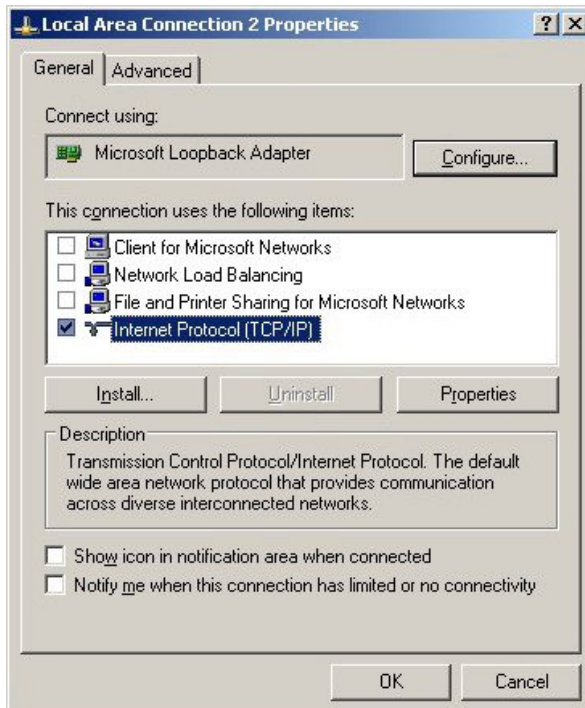
1. Open the Control Panel and double-click **Add Hardware**
2. Once the Hardware Wizard opens, click **Next**
3. Select **Yes, I have already connected the hardware**, click **Next**
4. Scroll to the bottom of the list, select **Add a new hardware device**, click **Next**
5. Select **Install the hardware that I manually select from a list (Advanced)**, click **Next**
6. Select **Network adapters**, click **Next**
7. Select **Microsoft & Microsoft Loopback Adapter**, click **Next** as shown below



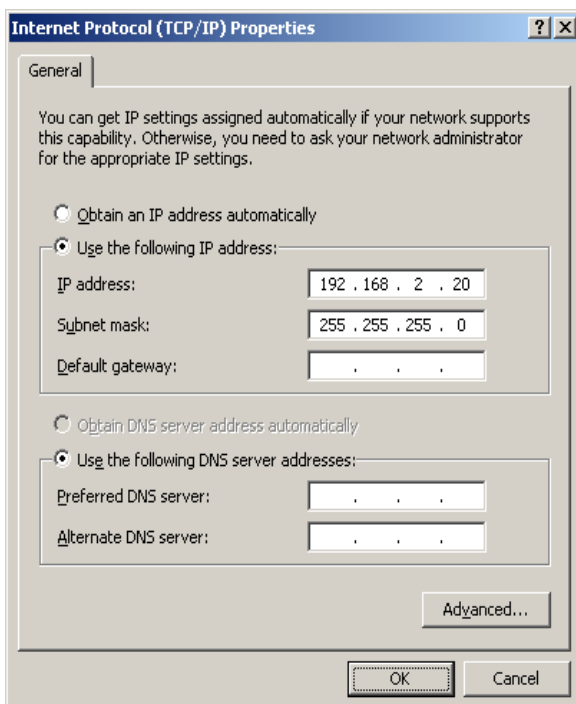
8. Click **Next** to start the installation, when complete click **Finish**

Step 2 – Configure the loopback adapter

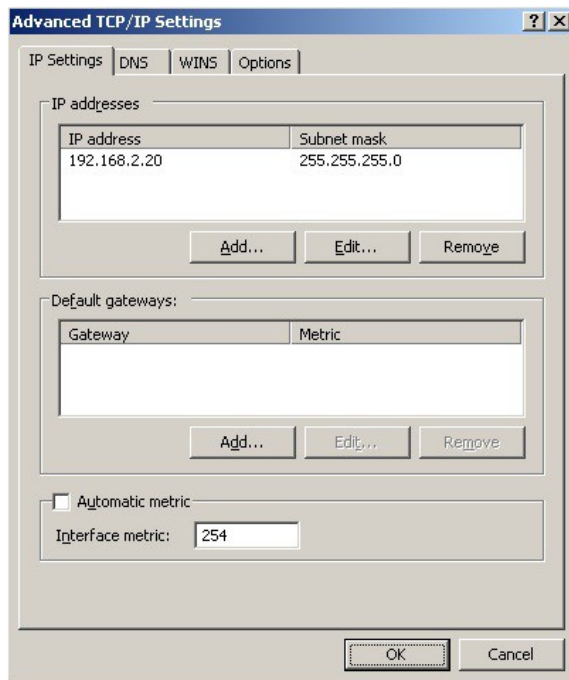
1. Open the Control Panel and double-click **Network Connections**
2. Right-click the new loopback adapter and select **Properties**
3. Un-check all items except **Internet Protocol (TCP/IP)** as shown below



4. Select **Internet Protocol (TCP/IP)**, click **Properties** and configure the IP address and mask to be the same as the Virtual Server (VIP), e.g. 192.168.2.20/24 as shown below



- Click **Advanced**, un-check **Automatic metric** and change **Interface metric** to 254 as shown below, this prevents the adapter responding to ARP requests for the VIP address



- Click **OK** on Advanced Settings & TCP/IP Properties, then click **Close** on Connection Properties to save and apply the new settings
- Now repeat the above process for all other Windows 2003 real servers



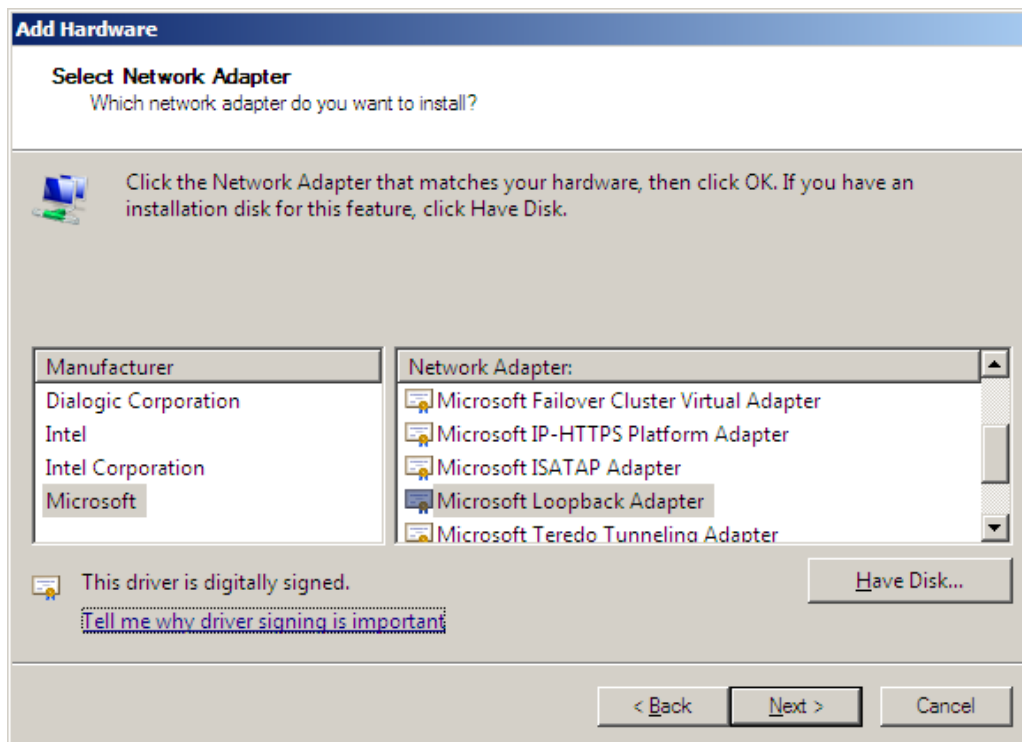
For Windows server 2003 SP1 & later, if you have enabled the built-in firewall, you will need to create an exception to enable access to the web server. This exception by default will allow traffic on both the network and loopback adapters.

Resolving ARP issues for Windows server 2008 (applies to DR mode only)

The basic concept is the same as for Windows 2000 / 2003. However, additional steps are required to set the strong / weak host behavior. This is used to either block or allow interfaces receiving packets destined for a different interface on the same server. As with Windows 2000 / 2003, If the real server is included in multiple VIPs, you can add additional IP addresses to the loopback adapter that correspond to each VIP.

Step 1 – Install the Microsoft loopback adapter

1. Click **Start**, then run **hdwwiz** to start the Hardware Installation Wizard
2. When the Wizard has started, click **Next**
3. Select **Install the hardware that I manually select from a list (Advanced)**, click **Next**
4. Select **Network adapters**, click **Next**
5. Select **Microsoft & Microsoft Loopback Adapter**, click **Next**

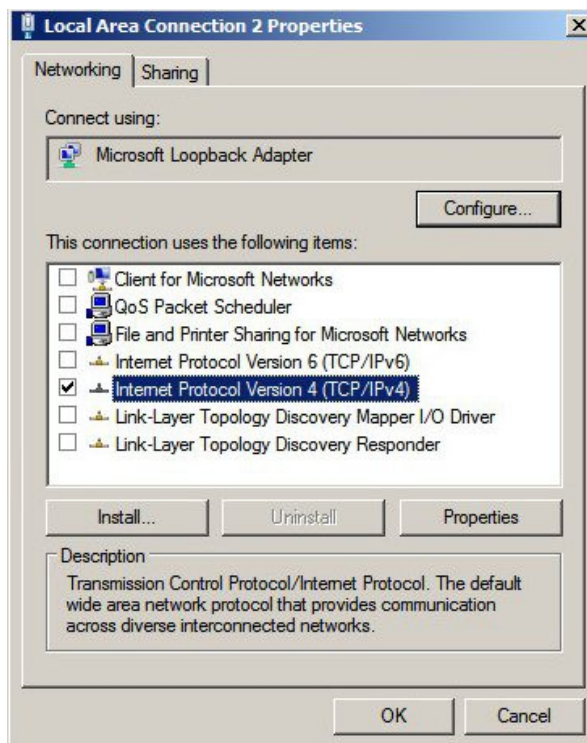


6. Click **Next** to start the installation, when complete click **Finish**

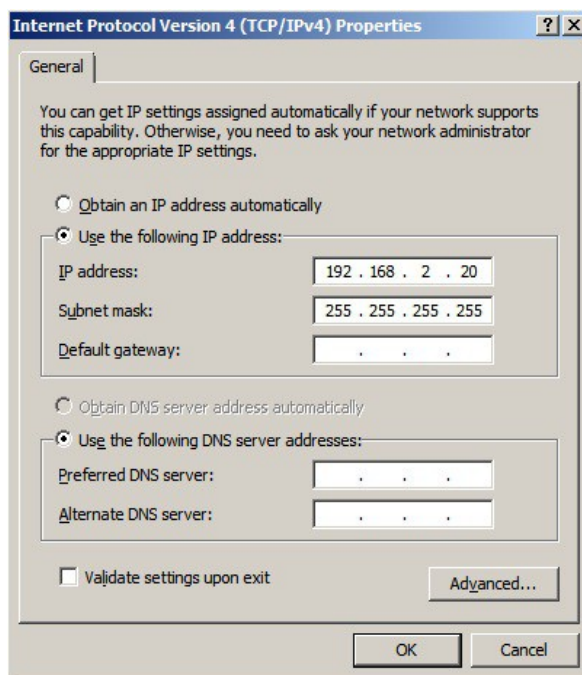
Step 2 – Configure the loopback adapter

1. Open Control Panel and click **View Network status and tasks** under **Network and internet**
2. Click **Change adapter settings**
3. Right-click the new Loopback adapter and select **Properties**

4. Un-check all items except **Internet Protocol Version 4 (TCP/IPv4)** as shown below



5. Select **Internet Protocol Version (TCP/IPv4)**, click **Properties** and configure the IP address to be the same as the Virtual Server (VIP) with a full subnet mask, e.g. 192.168.2.20/32 as shown below



6. Click **OK** on TCP/IP Properties, then click **Close** on Connection Properties to save and apply the new settings
7. Now repeat the above process on the other Windows 2008 real servers

N.B. For Windows 2008, it's not necessary to modify the interface metric on the advanced tab and should be left set to Automatic

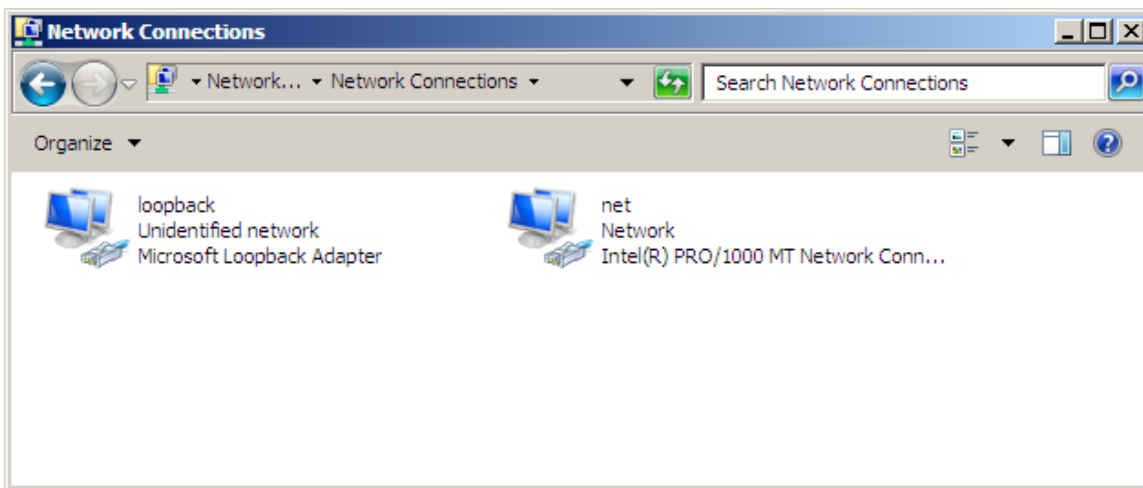
Step 3 – Configure the strong / weak host behavior

Windows Server 2000 and Windows Server 2003 use the weak host model for sending and receiving for all IPv4 interfaces and the strong host model for sending and receiving for all IPv6 interfaces. You cannot configure this behavior. The Next Generation TCP/IP stack in Windows 2008 and later supports strong host sends and receives for both IPv4 and IPv6 by default. To ensure that the Windows 2008 is running in the correct mode to be able to respond to the VIP, the following commands must be run on each real server:

```
netsh interface ipv4 set interface "net" weakhostreceive=enabled
netsh interface ipv4 set interface "loopback" weakhostreceive=enabled
netsh interface ipv4 set interface "loopback" weakhostsend=enabled
```

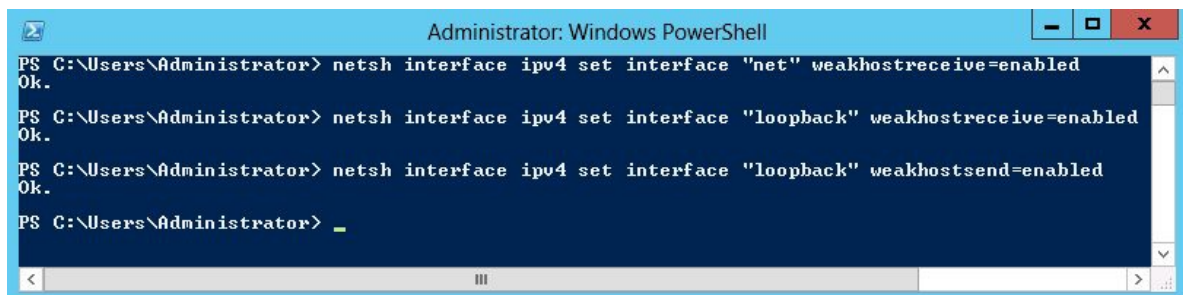
For these commands to work, the LAN connection NIC must be named “net” and the loopback NIC must be named “loopback” as shown below. If you prefer to leave your current NIC names, then the commands above must be modified accordingly. For example, if your network adapters are named “LAN” and “LOOPBACK”, the commands required would be:

```
netsh interface ipv4 set interface "LAN" weakhostreceive=enabled
netsh interface ipv4 set interface "LOOPBACK" weakhostreceive=enabled
netsh interface ipv4 set interface "LOOPBACK" weakhostsend=enabled
```



N.B. The names for the NICs are case sensitive, so make sure that the name used for the interface and the name used in the commands match exactly.

1. Start Powershell or use a command Window to run the 3 netsh commands as shown below



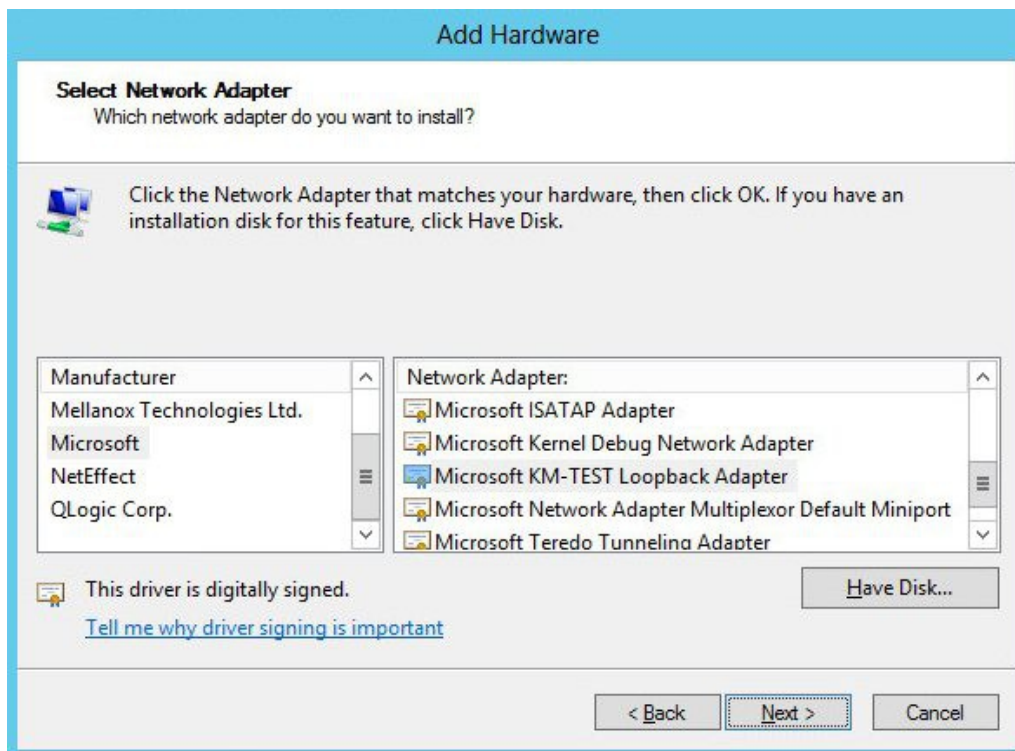
2. Now repeat these 3 commands on the other Windows 2008 real servers

Resolving ARP issues for Windows server 2012 (applies to DR mode only)

The basic concept is the same as for Windows 2000 / 2003. However, additional steps are required to set the strong / weak host behavior. This is used to either block or allow interfaces receiving packets destined for a different interface on the same server. As with Windows 2000 / 2003 / 2008, If the real server is included in multiple VIPs, you can add additional IP addresses to the loopback adapter that correspond to each VIP.

Step 1 – Install the Microsoft loopback adapter

1. Click **Start**, then run **hdwwiz** to start the Hardware Installation Wizard
2. When the Wizard has started, click **Next**
3. Select **Install the hardware that I manually select from a list (Advanced)**, click **Next**
4. Select **Network adapters**, click **Next**
5. Select **Microsoft & Microsoft KM-Test Loopback Adapter**, click **Next**

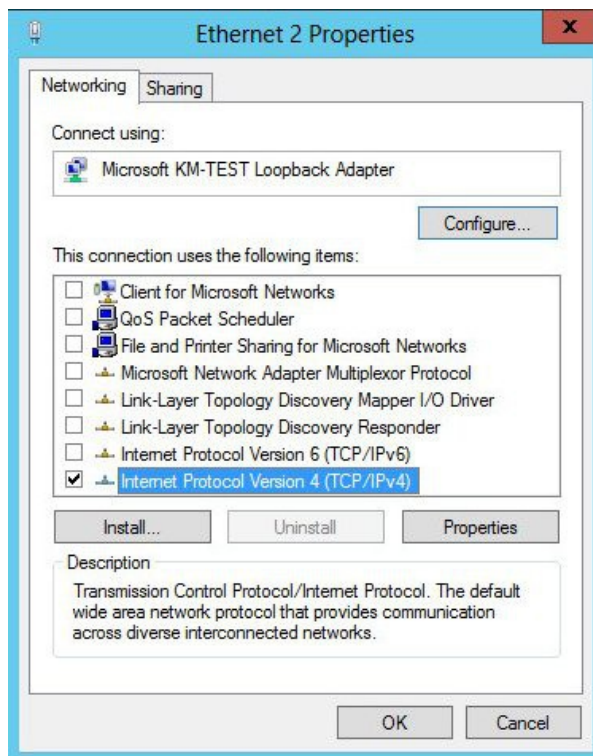


6. Click **Next** to start the installation, when complete click **Finish**

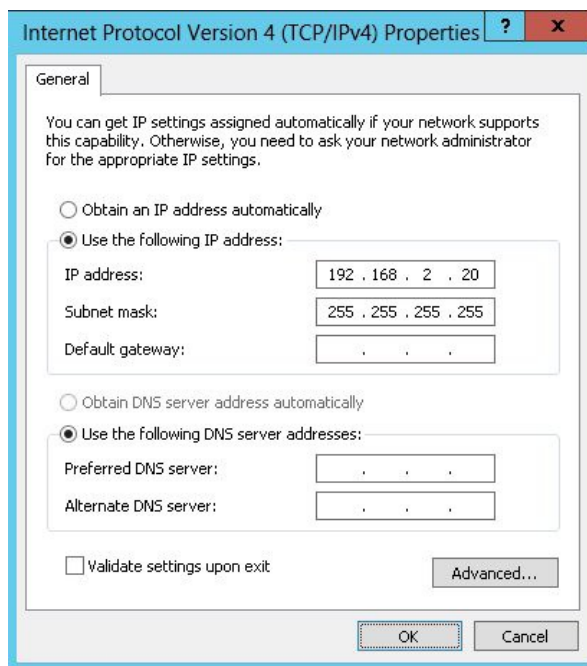
Step 2 – Configure the loopback adapter

1. Open Control Panel and click **Network and Sharing Center**
2. Click **Change adapter settings**
3. Right-click the new Loopback adapter and select **Properties**

- Un-check all items except **Internet Protocol Version 4 (TCP/IPv4)** as shown below



- Select **Internet Protocol Version (TCP/IPv4)**, click **Properties** and configure the IP address to be the same as the Virtual Server (VIP), with a full subnet mask e.g. 192.168.2.20/32 as shown below



- Click **OK** on TCP/IP Properties, then click **Close** on Ethernet Properties to save and apply the new settings
- Now repeat the above process on the other Windows 2012 real servers

N.B. For Windows 2012, it's not necessary to modify the interface metric on the advanced tab and should be left set to Automatic

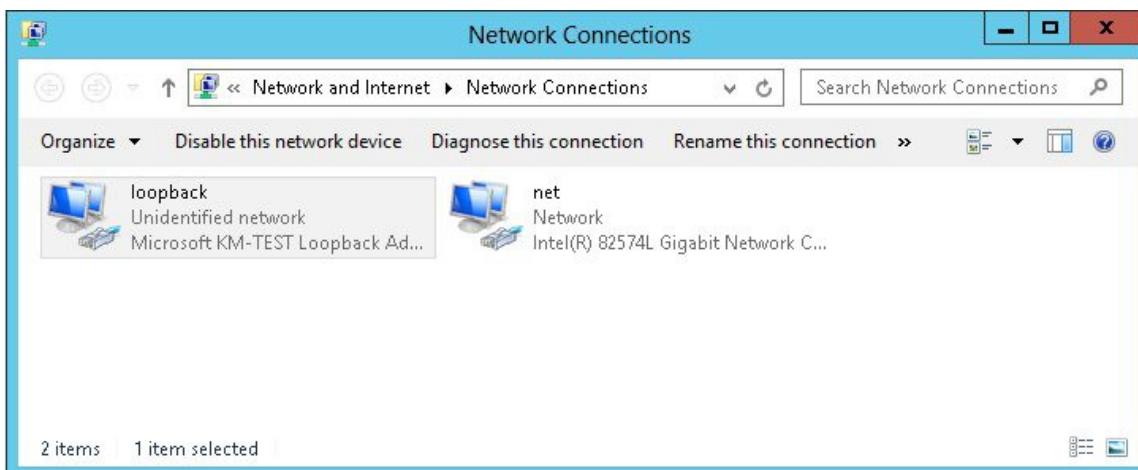
Step 3 – Configure the strong / weak host behavior

Windows Server 2000 and Windows Server 2003 use the weak host model for sending and receiving for all IPv4 interfaces and the strong host model for sending and receiving for all IPv6 interfaces. You cannot configure this behavior. The Next Generation TCP/IP stack in Windows 2008 and later supports strong host sends and receives for both IPv4 and IPv6 by default. To ensure that the Windows 2008 is running in the correct mode to be able to respond to the VIP, the following commands must be run on each real server:

```
netsh interface ipv4 set interface "net" weakhostreceive=enabled
netsh interface ipv4 set interface "loopback" weakhostreceive=enabled
netsh interface ipv4 set interface "loopback" weakhostsend=enabled
```

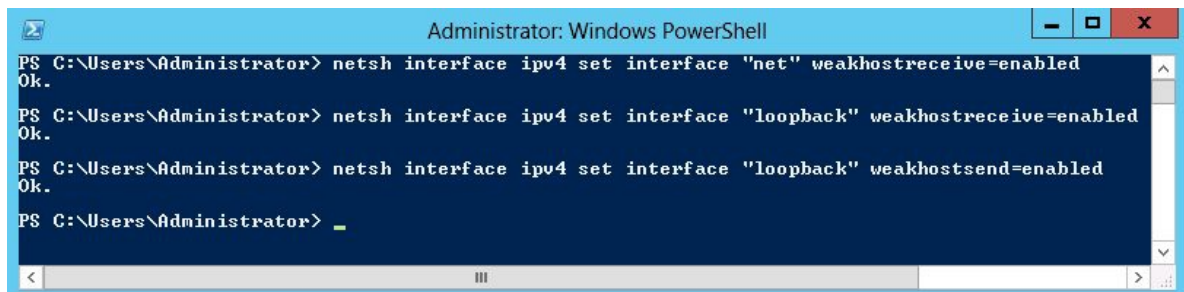
For these commands to work, the LAN connection NIC must be named “net” and the loopback NIC must be named “loopback” as shown below. If you prefer to leave your current NIC names, then the commands above must be modified accordingly. For example, if your network adapters are named “LAN” and “LOOPBACK”, the commands required would be:

```
netsh interface ipv4 set interface "LAN" weakhostreceive=enabled
netsh interface ipv4 set interface "LOOPBACK" weakhostreceive=enabled
netsh interface ipv4 set interface "LOOPBACK" weakhostsend=enabled
```



N.B. The names for the NICs are case sensitive, so make sure that the name used for the interface and the name used in the commands match exactly.

1. Start Powershell or use a command Window to run the 3 netsh commands as shown below



2. Now repeat these 3 commands on the other Windows 2012 real servers

Verifying netsh Settings for Windows 2008 & 2012

To verify that settings have been configured correctly, run the following command on each real server to clearly list the settings that have been applied to the interface:

```
netsh interface ipv4 show interface <interface name>
```

i.e. for the 'loopback' adapter run :netsh interface ipv4 show interface loopback

i.e. for the 'net' adapter run :netsh interface ipv4 show interface net

e.g.

```
C:\Users\Administrator>netsh interface ipv4 show interface loopback
```

```
Interface loopback Parameters
```

```
-----
IfLuid                : ethernet_9
IfIndex               : 15
State                 : connected
Metric                : 30
Link MTU              : 1500 bytes
Reachable Time        : 28500 ms
Base Reachable Time   : 30000 ms
Retransmission Interval : 1000 ms
DAD Transmits         : 3
Site Prefix Length    : 64
Site Id               : 1
Forwarding            : disabled
Advertising           : disabled
Neighbor Discovery    : enabled
Neighbor Unreachability Detection : enabled
Router Discovery      : dhcp
Managed Address Configuration : enabled
Other Stateful Configuration : enabled
Weak Host Sends       : enabled
Weak Host Receives    : enabled
Use Automatic Metric  : enabled
Ignore Default Routes : disabled
Advertised Router Lifetime : 1800 seconds
Advertise Default Route : disabled
Current Hop Limit     : 0
Force ARPND wake up patterns : disabled
Directed MAC wake up patterns : disabled
```

```
C:\Users\Administrator>
```

This shows that the settings have been applied correctly.



For Windows server 2008 / 2012, if you want to leave the built-in firewall enabled, you'll either need to enable the relevant default firewall exceptions or create your own to enable access to the web server. By default these exceptions will allow traffic on both the network and loopback adapters.



Failure to correctly configure the real servers to handle the ARP problem is the most common problem in DR configurations.

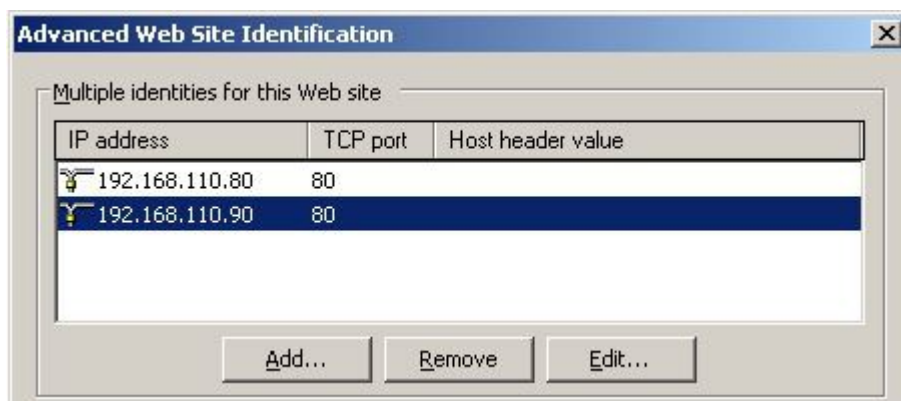
Configuring IIS to Respond to Both the RIP and VIP

Windows 2000 / 2003

For DR mode, it's also important to make sure that IIS responds to both the VIP and RIP. By default, IIS listens on all configured IP addresses, this is shown in the example below (shows Windows 2003 example). As can be seen the IP address field is set to 'All Unassigned'.

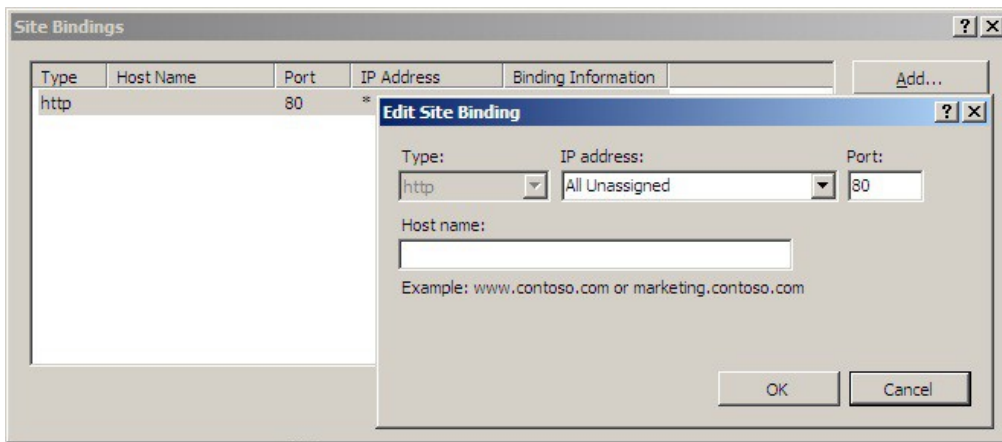


If the default configuration is left, no further IIS configuration is required. If you do change the IP address in the bindings from 'All Unassigned' to a specific IP address, then you need to make sure that you also add a binding for the Virtual Server IP address (VIP) as shown in the example below:

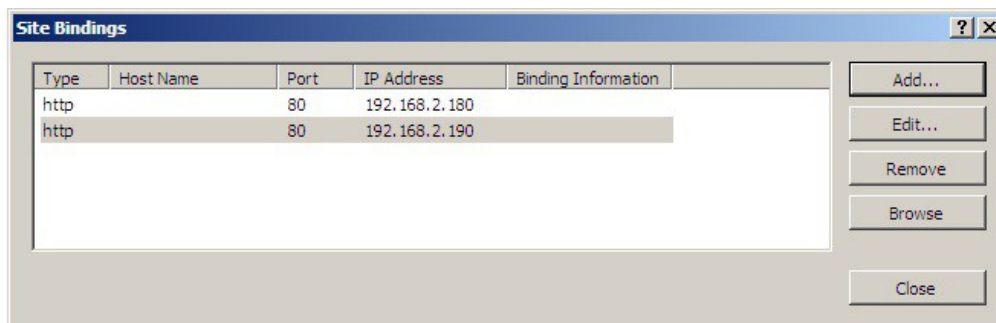


Windows 2008 / 2012

For DR mode, it's also important to make sure that IIS responds to both the VIP and RIP. By default, IIS listens on all configured IP addresses, this is shown in the example below (shows Windows 2008 example). As can be seen the IP address field is set to "All Unassigned".



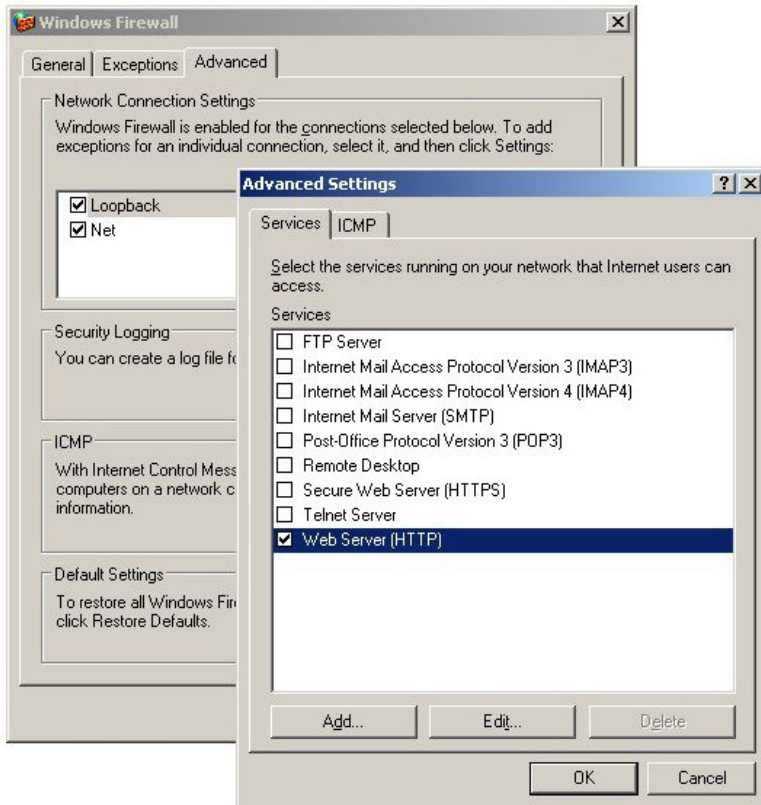
If the default configuration is left, no further IIS configuration is required. If you do change the IP address in the bindings from "All Unassigned" to a specific IP address, then you need to make sure that you also add a binding for the Virtual Server IP address (VIP) as shown in the example below:



Firewall Settings

Windows 2003 SP1+

For Windows Server 2003 SP1 & later, if you have enabled the built-in firewall, you will need to enable the Web Server (HTTP) exception to permit access to the web server. This exception is created automatically when IIS is installed and when enabled allows traffic on both the network and loopback adapters.



Windows 2008 R1 Firewall Settings

For Windows 2008 R1 the firewall configuration is very similar to windows 2003 R2. Again, an exception is created automatically that must be enabled to permit port 80 HTTP traffic. You just need to enable the firewall for both interfaces then ensure that the WWW service check-box is ticked as shown below:



Windows 2008 R2 Firewall Settings

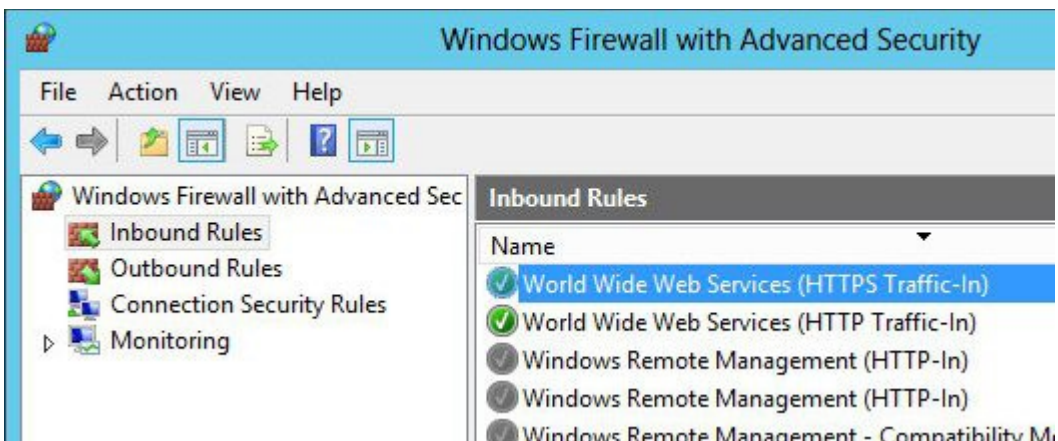
Windows 2008 automatically creates several default firewall rules for both inbound and outbound traffic. By default, all outbound traffic is allowed and all inbound traffic is blocked except where a rule allows it. Outbound rules can also be enabled if necessary. There are 3 firewall policies and interfaces can be associated with one of these 3 policies (domain, private and public) although the loopback adapter automatically gets associated with the public profile and this cannot be changed.

For a web server listening on port 80 the following default HTTP rules need to be enabled as shown below:



Windows 2012 Firewall Settings

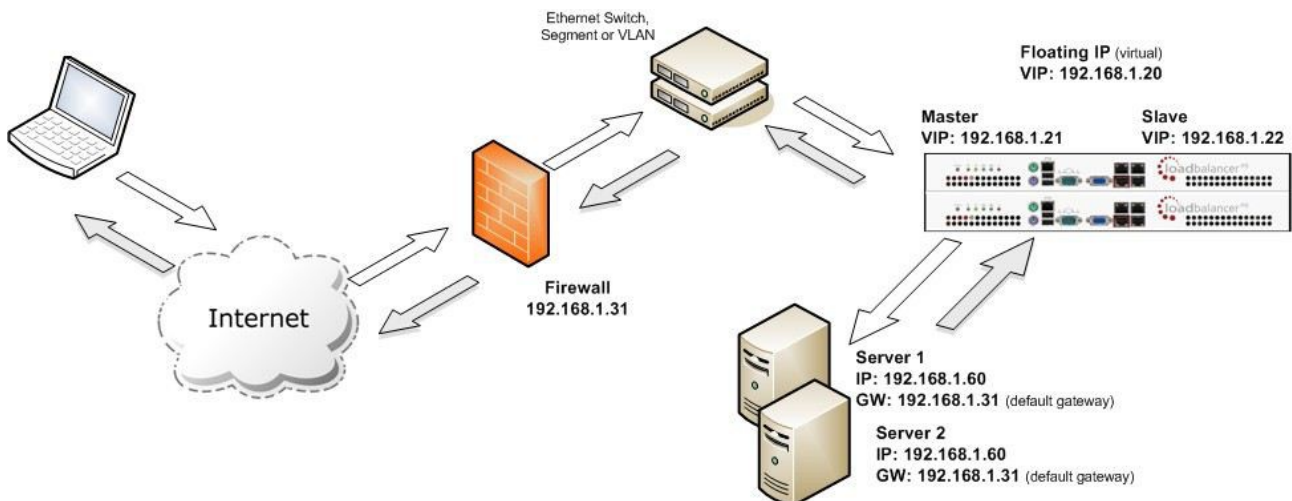
Windows 2012 is very similar to Windows 2008 R2 as shown below.



Advanced NAT Considerations

The NAT style of load balancing does have the advantage that the only change to the Real Servers is to modify the default gateway, IP address and subnet. You can also utilize the added security of having your Real Servers hidden in a subnet behind the load balancer. However, in our honest opinion, we think it is not wise to use your load balancer as a firewall. It adds complexity, and while the Loadbalancer.org appliance can be configured to be rock solid secure, *you should at least be fully aware of what you are doing if it is going to be your bastion host.*

Installing a pair of Loadbalancer.org appliances in NAT mode behind your own firewall solution as shown in the diagram below is a common implementation:



Please also refer to the NAT mode notes on page 20.

POTENTIAL ISSUES:

1. Your Real Servers won't be able to access the Internet through the new default gateway (except when replying to requests made through the external VIP).
2. Non-load balanced services on the Real Servers (e.g. RDP for management access to Windows servers) will not be accessible since these have not been exposed via the load balancer.

To Solve Issue #1

When NAT mode is selected in the setup wizard, the AutoNAT feature will be automatically enabled. If you have not used the wizard, you'll need to configured AutoNAT manually.

To enable AutoNAT manually:

- In the WUI, open *Edit Configuration > Layer 4 – Advanced Configuration*
- Change AutoNAT from 'off' to the external interface being used – normally eth1
- Click **Update**

This activates the rc.nat script that forces external network traffic to be MASQUERADED to and from the external network. The iptables masquerade rule used for this is shown below:

```
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

To Solve Issue #2

If you want any specific services to be exposed for your Real Servers you have two choices:

- Set up a specific Virtual Server with a single Real Server for the service e.g. just one Real Server in an FTP Real Server cluster

or

- Set up individual public IPs for the services required with individual SNAT and DNAT rules for each service as shown in the example below. These lines should be added to the firewall script using the WUI option *Maintenance > Firewall Script*

```
INT_ADDR="10.12.1.1"
EXT_ADDR="234.23.45.236"
EXT_IFACE="eth1"

iptables -t nat -A POSTROUTING -o $EXT_IFACE -p tcp -s $INT_ADDR -j SNAT --to-source $EXT_ADDR
iptables -t nat -A PREROUTING -i $EXT_IFACE -p tcp -d $EXT_ADDR -j DNAT --to-destination $INT_ADDR
```

If AutoNAT was enabled and you have also configured the above DNAT/SNAT rule, the following firewall entries under *View Configuration > Firewall Rules* would be shown:

```
chain PREROUTING (policy ACCEPT 53 packets, 7388 bytes)
pkts bytes target      prot opt in     out     source      destination
  0    0 DNAT          tcp  --  eth1    *       0.0.0.0/0   192.168.110.124  to:10.12.1.1

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in     out     source      destination
  0    0 SNAT          tcp  --  *       eth1    10.12.1.1   0.0.0.0/0      to:234.23.45.236
  0    0 MASQUERADE    all  --  *       eth1    0.0.0.0/0   0.0.0.0/0
```



Don't hesitate to contact support@loadbalancer.org to discuss any specific requirements you may have.

Explaining the RIP & VIP in NAT Mode

RIP is the Real IP address of a back-end server and VIP is the Virtual IP address of the cluster. You can have as many VIPs as you like but for this example we are only using one.

N.B. NAT mode routing is a common and very effective standard routing technique used in firewalls.

The following table illustrates the rules specified for the load balancer in NAT mode:

Protocol	VIP	Port	RIP	Port
TCP	10.0.0.20	80	192.168.1.50	80

All traffic destined for IP address 10.0.0.20 on port 80 is load-balanced to the real IP address 192.168.1.50 on port 80. Packet rewriting works as follows:

The incoming packet for the web service has source and destination addresses as:

```
SOURCE    x.x.x.x:3456          DEST    10.0.0.20:80
```

The packet would be rewritten and forwarded to the back-end server as:

```
SOURCE    x.x.x.x:3456          DEST      192.168.1.50:80
```

Replies get back to the load balancer as:

SOURCE 192.168.1.50:80 DEST x.x.x.x:3456

The packets would be written back to the VIP address and returned to the client as:

```
SOURCE    10.0.0.20:80          DEST      x.x.x.x:3456
```

Notes:

- In NAT mode the source IP address is preserved i.e. back-end server logs client IP address
- The back-end server RIP must have its default gateway pointing at the load balancer
- The back-end server must be on the internal subnet
- Servers on the internal subnet cannot access the external VIP
- NAT mode allows you to do port translation i.e. have a different RIP port than the VIP port

Route Configuration for Windows Server with One-Arm NAT Mode

When a client on the same subnet as the Real Server tries to access the Virtual Server on the load balancer the request will fail. The Real Server will try to use the local network to get back to the client rather than going through the load balancer and getting the correct network translation for the connection.

To rectify this issue we need to add a route to each Windows server that takes priority over the default Windows routing rules.

This is a simple case of adding a permanent route as shown below:

```
route add -p 192.168.1.0 mask 255.255.255.0 metric 1
```

N.B. Replace 192.168.1.0 with your local subnet address.

The default route to the local network has a metric of 10, so this new route overrides all local traffic and forces it to go through the load balancer as required.

Any local traffic (same subnet) is handled by this route and any external traffic is handled by the default route (which also points at the load balancer).

Route Configuration for Linux with One-Arm NAT Mode

When a client on the same subnet as the Real Server tries to access the Virtual Server on the load balancer the request will fail. The Real Server will try to use the local network to get back to the client rather than going through the load balancer and getting the correct network translation for the connection.

To rectify this issue we need to modify the local network route to a higher metric:

```
route del -net 192.168.1.0 netmask 255.255.255.0 dev eth0  
route add -net 192.168.1.0 netmask 255.255.255.0 metric 2000 dev eth0
```

N.B. Replace 192.168.1.0 with your local subnet address.

Then we need to make sure that local network access uses the load balancer as its default route:

```
route add -net 192.168.1.0 netmask 255.255.255.0 gateway 192.168.1.21 metric 0 dev eth0
```

N.B. Replace 192.168.1.21 with your load balancer gateway

Any local traffic (same subnet) is handled by this manual route and any external traffic is handled by the default route (which also points at the load balancer).

Advanced Layer 7 Considerations

It's possible to manually modify the HAProxy configuration file (/etc/haproxy/haproxy.cfg) to enable the appliance to support custom Layer 7 configurations. It's important to note that these manual changes can be overwritten under various circumstances:

- When changes are made to ANY Layer 7 VIP or RIP
- When a Layer 7 VIP is taken off/on line using System Overview

To prevent the configuration file being overwritten in these cases:

- In the WUI, open *Edit Configuration > Layer 7 – Advanced Configuration*
- Change **Disable HAProxy Config Write** to 'Yes'
- Click **Update**
- The message shown below will be displayed:

Warning: Disable HAProxy Config Write is active.
Configuration changes are not permitted.

EDIT CONFIGURATION > ADVANCED CONFIGURATION (HAPROXY)

Layer 7 (HAProxy):	
Disable HAProxy Config Write	on ▼ ?

Based on the above points, aim to configure Layer 7 services in the following order:

- Configure all standard Layer 7 services via the WUI
- Now change **Disable HAProxy Config Write** to 'on'
- Then make the required custom changes to the HAProxy configuration file using your preferred editor



If you do manually add an additional Virtual Server, don't forget to also add a Floating IP on the same address as the new VIP using *Edit Configuration > Floating IPs*

Load balancing Based on URL Match with HAProxy

To support URL matched load balancing the structure of the HAProxy config file must be changed to use the front-end / back-end model as shown in the example below. This requires that the HAProxy file be modified manually which does have various implications as described at the start of this section.

```
# HAProxy configuration file generated by load balancer appliance
global
uid 99
gid 99
daemon
stats socket /var/run/haproxy.stat mode 600
maxconn 40000
ulimit-n 65536
pidfile /var/run/haproxy.pid
defaults
mode http
contimeout 4000
clitimeout 42000
srvtimeout 43000
balance roundrobin

frontend f1
bind 192.168.2.112:80
acl test_acl1 path_beg /test1
acl test_acl2 path_beg /test2
use_backend b1 if test_acl1
use_backend b2 if test_acl2
default_backend b2
option httpclose

backend b1
cookie SERVERID insert nocache indirect
server s1 192.168.2.99:80 weight 1 cookie s1 check
server s2 192.168.2.10:80 weight 1 cookie s2 check

backend b2
cookie SERVERID insert nocache indirect
server s3 192.168.2.6:80 weight 1 cookie s3 check
```

As shown in the above example, instead of the usual 'listen' directive (which groups the Virtual Server and its real backends together), we now have separate frontend and backend sections.

In this example:

'test_acl1' ← this is the name / label of the ACL

'path_beg' ← this means match the beginning of the path to a certain value, in this case '/test1'

and similarly for test_acl2

There are numerous matching options available. For more details and examples, please refer to:

<http://haproxy.1wt.eu/download/1.5/doc/configuration.txt>

then search that page for "Matching at Layer 7"



Don't hesitate to contact support@loadbalancer.org to discuss any specific requirements you may have.

HTTP to HTTPS Redirect using HAProxy & Pound (SSL Termination on the Load Balancer)

In this example the redirect directive is used to redirect connections on port 80 to the Pound SSL VIP listening port 443. This can be achieved by manually adding the 2 lines shown below in bold to the HAProxy configuration file:

```
listen VIP1
    bind 192.168.110.142:80
    mode http
    balance leastconn
    acl ACL-A src 192.168.110.142           ← see note 1
    redirect prefix https://192.168.110.142 if !ACL-A ← see note 2
    cookie SERVERID insert nocache indirect
    server backup 127.0.0.1:9081 backup non-stick
    option httpclose
    option forwardfor
    option redispatch
    option abortonclose
    maxconn 40000
    server rip1 192.168.70.195:80 weight 1 cookie rip1 check inter 2000 rise 2 fall 3
minconn 0 maxconn 0 on-marked-down shutdown-sessions
```

Steps:

1. Create a standard Pound / HAProxy VIP for SSL termination (for an example see page 65)
2. Using an editor such as vi, add the two lines as shown in bold above (substituting the correct IP)

N.B. Using this method, a floating IP address is automatically added when the VIP is created using the WUI. If you modify the HAProxy config file directly without using the WUI, make sure you also add a corresponding floating IP using : Edit Configuration > Floating IP's

Note 1

This line configures an acl named 'ACL-A', where the criteria for a match is that the source IP address must be 192.168.110.142.

Note 2

This line causes the redirect to https://192.168.110.142 to occur when the acl is not matched, i.e. for all traffic that is NOT coming from the pound VIP.

This can also be a domain name entry such as:

```
redirect prefix https://www.loadbalancer.org if !ACL-A
```



Don't forget that any manual changes can be overwritten in various circumstances as explained in the start of this section.

HTTP to HTTPS Redirect using HAProxy (SSL Termination on the Real Server)

In this example a simple VIP is added which redirects inbound requests to another VIP that is listening on port 443.

```
listen VIP-80 192.168.110.178:80          ← see note 1
    redirect location https://192.168.110.178:443    ← see note 2
listen VIP-443
    bind 192.168.110.178:443
    mode tcp
    balance leastconn
    server backup 127.0.0.1:9081 backup non-stick
    option redispatch
    option abortonclose
    maxconn 40000
    server rip1 192.168.101.2:443 weight 1 check inter 2000 rise 2 fall 3 minconn 0 maxconn
0 on-marked-down shutdown-sessions
```

Steps:

1. Create a standard VIP with associated RIPs that listens on port 443 (VIP-443 in the above example)
2. Using an editor such as vi, add the two lines as shown in bold above (substituting the correct IP)

N.B. Using this method, a floating IP address is automatically added when the VIP is created using the WUI. If you modify the HAProxy config file directly without using the WUI, make sure you also add a corresponding floating IP using : Edit Configuration > Floating IP's

Note 1

An additional VIP (VIP-80) is added that listens on port 80.

Note 2

A redirect is implemented to redirect to the second VIP (VIP-443) on port 443. HTTPS traffic is then passed on to the Real Server.

This can also be a domain name entry such as:

```
redirect location https://www.loadbalancer.org
```



Don't forget that any manual changes can be overwritten in various circumstances as explained in the start of this section.

HAProxy Error Codes

For reference, the layer 7 HAProxy error codes are as follows:

Code	When / Reason
200	access to stats , and when replying to monitoring requests
301	when performing a redirection, depending on the configured code
302	when performing a redirection, depending on the configured code
303	when performing a redirection, depending on the configured code
400	for an invalid or too large request
401	when an authentication is required to perform the action (when accessing the stats page)
403	when a request is forbidden by a "block" ACL or "reqdeny" filter
408	when the request timeout strikes before the request is complete
500	when HAProxy encounters an unrecoverable internal error, such as a memory allocation failure, which should never happen
502	when the server returns an empty, invalid or incomplete response, or when an "rspdeny" filter blocks the response
503	when no server was available to handle the request, or in response to monitoring requests which match the "monitor fail" condition
504	when the response timeout strikes before the server responds

Complete detailed information for HAProxy configuration is available here:

<http://haproxy.1wt.eu/download/1.5/doc/configuration.txt>

Configuring VIPs & RIPv via Command Line / Script

If required it is possible to add, remove and edit Virtual / Real Servers via the command line. This enables loadbalancer configuration changes to be made via script rather than using the WUI. This is not a full API, but it does allow basis manipulation of load balanced services.

Layer 4

For layer 4, the ipvsadm command is used. Several examples are provided below.

Add a TCP based Virtual Server , use round robin scheduling:

```
ipvsadm -A -t 192.168.65.192:80 -s rr
```

Add a TCP based Real Server in DR mode:

```
ipvsadm -a -t 192.168.65.192:80 -g -r 192.168.70.196:80
```

Add a TCP based Real Server in NAT mode:

```
ipvsadm -a -t 192.168.65.192:80 -m -r 192.168.70.196:80
```

Add a UDP based Virtual Server , use least connection scheduling:

```
ipvsadm -A -u 192.168.65.192:80 -s lc
```

Add a UDP based Real Server in DR mode:

```
ipvsadm -a -u 192.168.65.192:80 -g -r 192.168.70.196:80
```

Delete a TCP based Virtual Server:

```
ipvsadm -D -t 192.168.65.180:80
```

Delete a TCP based Real Server:

```
ipvsadm -d -t 192.168.65.122:80 -r 192.168.70.134:80
```

View the current running config:

```
ipvsadm -ln
```

```
IP Virtual Server version 1.2.1 (size=4096)
```

```
Prot LocalAddress:Port Scheduler Flags
```

```
-> RemoteAddress:Port          Forward Weight ActiveConn InActConn
```

```
TCP 192.168.65.120:80 rr
```

```
-> 192.168.70.130:80           Route    1      0          0
```

```
-> 192.168.70.131:80           Route    1      0          0
```

```
TCP 192.168.65.122:80 rr
```

```
-> 192.168.70.132:80           Mass     1      0          0
```

```
-> 192.168.70.133:80           Mass     1      0          0
```

Layer 7

For layer 7 HAProxy VIPs, the socat socket command can be used as shown in the examples below.

To take a server offline:

```
echo "disable server VIP_Name/RIP_Name" | socat unix-connect:/var/run/haproxy.stat stdio
```

To bring a server online:

```
echo "enable server VIP_Name/RIP_Name" | socat unix-connect:/var/run/haproxy.stat stdio
```

To set the weight of a Real Server:

```
echo "set weight VIP_Name/RIP_Name 0" | socat unix-connect:/var/run/haproxy.stat stdio
```

To view HAProxy's running configuration:

```
echo "show info" | socat unix-connect:/var/run/haproxy.stat stdio
```

N.B. Other examples can be found by searching for "Unix Socket Commands" at the following link:

<http://haproxy.1wt.eu/download/1.5/doc/configuration.txt>



Please note that since these changes are being made directly to the running configuration, the services that are displayed in the System Overview will not match the running configuration.



For additional assistance don't hesitate to contact : support@loadbalancer.org.

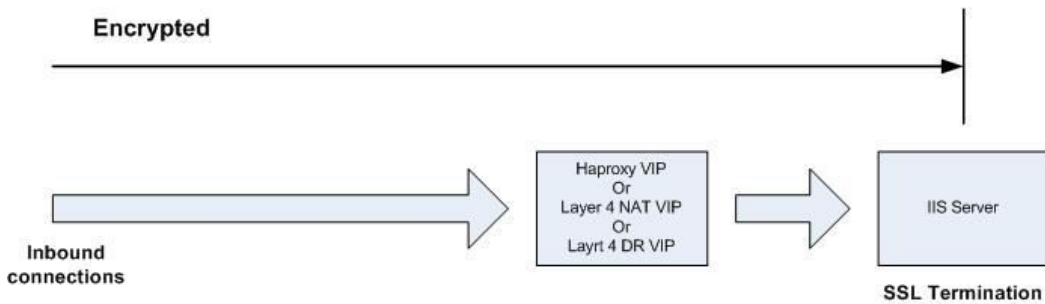
SSL Termination

When SSL termination is required, the certificate can either be installed on the Real Servers (e.g. IIS) or directly on the load balancer.



NOTE: SSL termination on the load balancer can be very CPU intensive. In most cases, for a scalable solution, terminating SSL on the Real Servers is the best option.

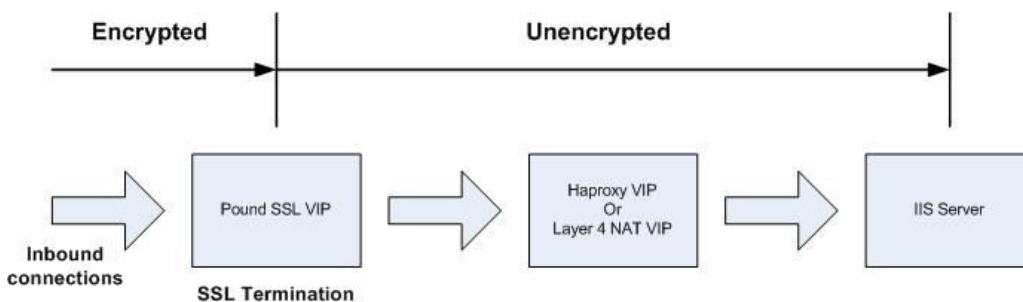
Certificate on the Real Servers



Using this Method:

- Data is encrypted from client to server. This provides full end-to-end data encryption as shown in the diagram below
- It's not possible to use HTTP cookie persistence since the packet is encrypted and therefore the cookie cannot be read. If persistence via the load balancer is required, IP persistence must be used

Certificate on the Load Balancer



Using this Method:

- Since SSL is terminated on the load balancer, data from the load balancer to the IIS servers is not encrypted as shown in the diagram above. This may or may not be an issue depending on the network structure between the load balancer and IIS servers and your security requirements
- It's possible to use HTTP cookie based persistence
- A Pound SSL Virtual Server is used to terminate SSL. The backend for this Virtual Server can be either a Layer 4 NAT mode Virtual Server or a Layer 7 HAProxy Virtual Server



DR mode cannot be used as the back-end VIP since Pound acts as a proxy, and the Real Servers see requests with a source IP address of the Virtual Server. Since the Real Servers believe that they own the Virtual IP (due to the loopback adapter configured to handle to ARP problem) they are therefore unable to reply to Pound.

Creating a New Certificate Using a CSR

By default, when creating the SSL virtual service a self-signed certificate is used. This is ideal for testing but needs to be replaced for live deployments.

In order to obtain a valid signed certificate from a certificate authority such as Verisign or Thawte you'll need to generate a certificate request (CSR).

To customize the certificate configuration, go to *Edit Configuration > SSL Termination*, then click **[Certificate]** next to the relevant Virtual Server.

To generate a CSR, fill in the required details and click **Generate SSL Certificate Request**

Country code (C)	GB	?
State or Province (ST)	Hampshire	?
City (L)	Portsmouth	?
Organisation (O)	Loadbalancer.org	?
Organisation unit (OU)	Support	?
Domain (CN)	www.loadbalancer.org	?
Email address	support@loadbalancer.org	?

Generate SSL Certificate Request

Then copy the resulting Certificate Signing Request from the top pane and send this to your chosen Certificate Authority.

Certificate Signing Request	<pre>-----BEGIN CERTIFICATE REQUEST----- MIICuDCCAaACAQAwwczELMAkGA1UEBhMCVUcxDDAKBgNVBAgTA3NhZjENMA5GA1UE BxMEYXNnZjENMA5GA1UEChMEYXNkZzENMA5GA1UECjMEYXNkZzENMA5GA1UEAxME YXNkZzEaMBGCSqGSIb3DQEJARYLYXNkZzEaMBGCSqGSIb3DQEB AQUAA4IBDwAwggEKAoIBAQCd/aIT7ZjB/1K7TCUvBoByrbhuVrY0/kQtSYU5j4MX 60B3X0x4+OhJvV1SjZ8Nsjsx7m+2rfE+NP2mz/5y0rolyEuHgHT03P/gcOGp5O2N klz3qKgvsDEpIVjCs+ALz+9arh4hsKf8CmpczFnKf1+/LCA97kkoC6zxLiKSoZ+n ab8Gd7rGfMAf95iqmgwUYUd6oqKNjjBb1AT8x10DBPGRJ+ntFhQbZKDPnYRhRYIf -----</pre>
Signed Certificate from CA	

Upload Signed Certificate

Once you receive your signed certificate from the CA, copy this into the lower pane and click **Upload Signed Certificate**.



If you need to add intermediate certificates to the chain, this can be done by appending these certificates at the end of the certificate from your CA in the lower pane.

Using an Existing Certificate

To use an existing certificate, you must first ensure that your certificate and associated files are in PEM format. The file should contain the private key (*without a password*), the signed certificate issued by a Certificate Authority (CA) and also any additional validation / intermediate certificates that may be required by the CA.

Creating a PEM File

Using a text editor such as vi or vim under Linux or Notepad under Windows create an empty file called pem.txt for example. Then copy / paste the Certificate and Private Key into the file as follows (shows truncated versions):

```
-----BEGIN CERTIFICATE-----
MIICsDCCAhmGAWIbAgIJAL98jhEiUm3iMA0GCSqGSIb3DQEBBQUAMEUxCzAJBgNV
BAYTAkFVMRMwEQYDQIExpTb211LVN0YXRIMSEwHwYDVQQKEWhJbnRlcm5ldCBX
kU6DJupvN6U6PRI7+zcKqd8wUIY8+3CyYKHtJmkL5pSPoG8ASp4QnsVa01n+EDKj
E89UJCG2nMW5JVBnkyHYbQTvU8MeR3ilhe2fw+qVE2pgxWYyWaGm8QwTsxQKgbxiG
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbyaAEbcSVympQJdgs6W6ajjLS
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCcPYkYHm8gYwlm3HyoVxjrymusOeIFgZlWYuaebIrreCplo+iy
pSxEruhpmqmdj2tYlpFwp9Q6wEW7OR/E+3ar8HdpHjxYOs/MWBMYPPLAfmh88bS7fh
rPCcmp1kj5oGE2+GZQJBAM2dPXwggR2NWKZJfJRgAuUfNmngRUQPiLosSmUCZ
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbyaAEbcSVympQJdgs6W6ajjLS
YwCZCE0HwquomN/q4ctnhgeN+kugDxlgCTVYd3eo/Dv/KZ16p4HUIrTqwES4Lun
-----END RSA PRIVATE KEY-----
```

Save the file, then using *Edit Configuration > Manage SSL Certificate > Upload prepared PEM file > Browse* , select this file and click **Upload PEM File**

Upload prepared PEM file

Now restart Pound using *Maintenance > Restart Services > Restart Pound*



If your master & slave are correctly configured as a clustered pair, when you upload the PEM file to the master, the file will be automatically copied over to the slave unit.



It's very important to backup all of these files. This can be done via the WUI from *Maintenance > Backup & Restore > Download SSL Certificates*.



If you have already generated the CSR on your Web Server, you will need to create a PEM file using the Certificate and Private Key, then upload this using the interface – see the following sections.

Adding an Intermediate Certificate

Certificate authorities may require that an intermediate CA certificate is installed in your server farm. This can be done by manually pasting the intermediate CA onto the end of your signed server PEM file and then uploading it to the appliance via the upload facility.

N.B. Your current signed key is stored in /usr/local/etc/certs/<vip-name>.pem

Select the text in the top pane and paste it into a text editor such as notepad (not Word or Wordpad):

```
-----BEGIN CERTIFICATE-----
MIICsDCCAhmGAWIbAgJAL98jhEiUm3iMA0GCSqGSIb3DQEBBQUAMEUx CzAJBgNV
BAYTAkFVMRMwEQYD VQI EwpTb21ILVN0YXRIMSEwHwYDVQKExhJbnRlcm5ldCBX
kU6DJupvN6U6PRI7+zcKqd8wUiY8+3CyYKHtJmkL5pSPoG8ASp4QnsVa01n+EDKj
E89UJCG2nMW5JVBKnyHYbQTvU8MeR3ilhe2fw+qVE2pgxWYyWaGm8QWtsxQKgbxiG
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbyaAEbcSVympQJdgs6W6ajjLS
-----END CERTIFICATE-----

-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCcPYkYHm8gYwlm3HyoVxjrymusOelFgZIWyuaeblrrCplo+iy
pSxEruhpmqmdj2tYlpFwp9Q6wEW7OR/E+3ar8HdpHjxYOs/MWBMYPALfah88bS7fh
rPCcmp1kj5oGE2+GZQJBAM2dPXwggR2NWKZJfJRgAuUFnmgrUQPILosSmUCZ
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbyaAEbcSVympQJdgs6W6ajjLS
YwC2ZCE0HwquomN/q4ctnhgeN+kugDxlgCTVYd3eo/Dv/KZ16p4HUIrTqwES4Lun
-----END RSA PRIVATE KEY-----
```

Then paste the intermediate CA certificate from your provider onto the end of the file so you get something similar to, but much longer than the following shortened example:

```
-----BEGIN CERTIFICATE-----
MIICsDCCAhmGAWIbAgJAL98jhEiUm3iMA0GCSqGSIb3DQEBBQUAMEUx CzAJBgNV
BAYTAkFVMRMwEQYD VQI EwpTb21ILVN0YXRIMSEwHwYDVQKExhJbnRlcm5ldCBX
kU6DJupvN6U6PRI7+zcKqd8wUiY8+3CyYKHtJmkL5pSPoG8ASp4QnsVa01n+EDKj
E89UJCG2nMW5JVBKnyHYbQTvU8MeR3ilhe2fw+qVE2pgxWYyWaGm8QWtsxQKgbxiG
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbyaAEbcSVympQJdgs6W6ajjLS
-----END CERTIFICATE-----

-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCcPYkYHm8gYwlm3HyoVxjrymusOelFgZIWyuaeblrrCplo+iy
pSxEruhpmqmdj2tYlpFwp9Q6wEW7OR/E+3ar8HdpHjxYOs/MWBMYPALfah88bS7fh
rPCcmp1kj5oGE2+GZQJBAM2dPXwggR2NWKZJfJRgAuUFnmgrUQPILosSmUCZ
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbyaAEbcSVympQJdgs6W6ajjLS
YwC2ZCE0HwquomN/q4ctnhgeN+kugDxlgCTVYd3eo/Dv/KZ16p4HUIrTqwES4Lun
-----END RSA PRIVATE KEY-----

-----BEGIN CERTIFICATE-----
rPCcmp1kj5oGE2+GZQJBAM2dPXwggR2NWKZJfJRgAuUFnmgrUQPILosSmUCZ
2bShC2AVC+ZDMNu6bvCdvfySi6EypUclvEwao7ZbyaAEbcSVympQJdgs6W6ajjLS
YwC2ZCE0HwquomN/q4ctnhgeN+kugDxlgCTVYd3eo/Dv/KZ16p4HUIrTqwES4Lun
rPCcmp1kj5oGE2+GZQJBAM2dPXwggR2NWKZJfJRgAuUFnmgrUQPILosSmUCZ
-----END CERTIFICATE-----
```

Save this text file and then use the **Upload PEM file** button as to assign this certificate to your Pound Virtual Server. Once the file is uploaded you will need to restart Pound.

Windows Servers

A fundamental requirement of importing a certificate into Pound is that the certificate file and the private key file must be in PEM format.

Windows Server is only able to export a private key file in .pfx format. Therefore the .pfx file must be converted to PEM format. This can be done using the program 'OpenSSL'.

The conversion can be done either on a Windows server or on any UNIX-like Operating System, such as the load balancer itself.

Using Windows:

OpenSSL is available as a binary package for Windows at the following location:

http://www.slproweb.com/download/Win32OpenSSL-1_0_0d.exe

This should be download and installed on a PC where you'd like to run the conversion process. There are no special instructions for this. You will now have an OpenSSL directory located on your filesystem. Click **START, RUN** then type `cmd.exe`. You need to navigate to the path where you installed your OpenSSL binaries. Within this directory chdir to `bin`

To convert .PFX to .PEM

```
openssl pkcs12 -in <drive:\path\to\cert>.pfx -nodes -out <drive:\path\to\new\cert>.pem
```

To convert .CER file to .PEM format:

```
openssl x509 -in <drive:\path\to\cert>.cer -inform DER -out <drive:\path\to\cert>.pem -outform PEM
```

Using UNIX / Linux:

Once OpenSSL has been installed, you can now use the below command to convert your private key into a format ZXTM can correctly decipher.

To convert .PFX to .PEM

```
openssl pkcs12 -in <path/to/exported/cert>.pfx -nodes -out <path/to/new/cert>.pem
```

To convert .CER file to .PEM format:

```
openssl x509 -in </pat/to/cert>.cer -inform DER -out </path/to/cert>.pem -outform PEM
```

This method can also be used from the Loadbalancer.org appliance console if required.

Import a Certificate Exported from Windows Server

For Windows, it's often easiest to get the certificate working on the server first. The certificate can then be exported from Windows in .pfx format, then converted to .pem format and finally loaded into the relevant Pound Virtual Server on the load balancer. The steps are:

- 1) Once the certificate is working correctly on your Windows server, export the certificate from Windows. The format will be .pfx.
- 2) Download openssl from : http://www.slproweb.com/download/Win32OpenSSL-1_0_0d.exe and install this on your PC.
- 3) Using openssl on your PC, convert the pfx file to a pem file. The command to use is:

```
openssl pkcs12 -in drive:\path to cert\cert.pfx -nodes -out drive:\path to cert\cert.pem
```

(You will be prompted for the password used to create the pfx file)

- 4) Now upload the PEM file using *Edit Configuration > SSL Termination > Certificate > Upload prepared PEM file*

- 5) Finally restart Pound (*Maintenance > Restart Pound-ISL*)

Converting an Encrypted Private Key to an Unencrypted Key

If a password has been included in the private key, this should be removed before it is used with your pem file. This can be done using the following method:

```
openssl rsa -in server.key -out server.key.unencrypted
```

(This can be done either on the load balancer or another machine with openssl installed)

Limiting Ciphers

To limit the Ciphers that Pound will respond to, simply enter the cipher string in the Ciphers field. For example, to limit to SSL v3, enter SSLv3 and click update. Multiple Ciphers can be entered separated by commas.

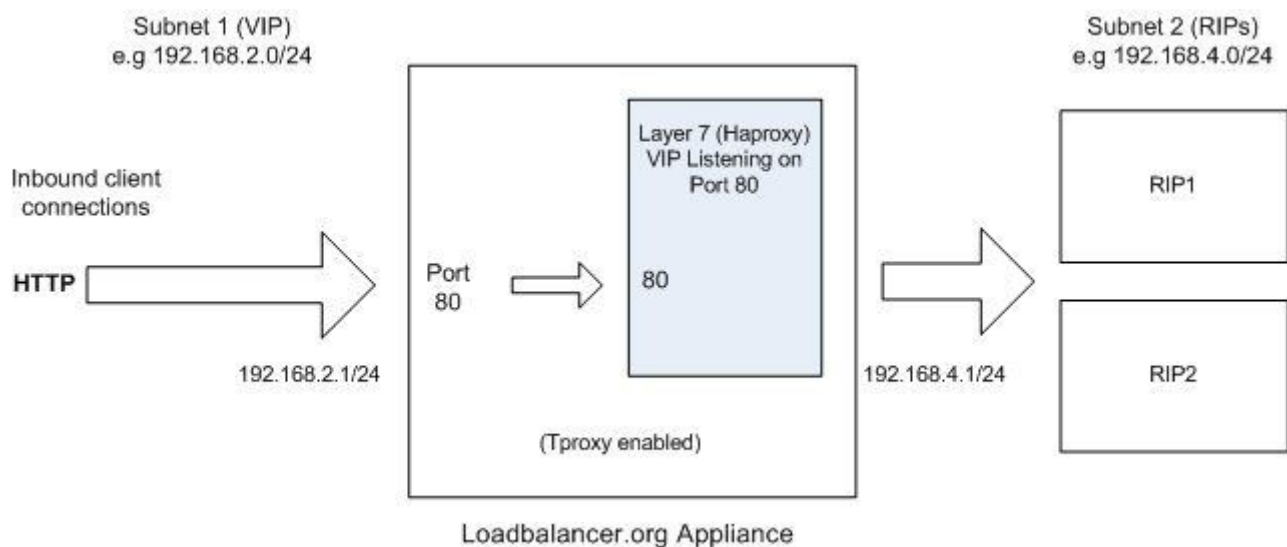
Label	SSL	?
Virtual Server IP address	192.168.110.100	?
Virtual Server Port	443	?
Backend Virtual Server IP Address	192.168.110.100	?
Backend Virtual Server Port	80	?
Ciphers to use	SSLv3	?
Enable WebDAV Verbs	<input type="checkbox"/>	?
Rewrite HTTP Redirects	<input checked="" type="checkbox"/>	?
<input type="button" value="Update"/>		

Using Tproxy

Tproxy can be used with HAProxy and Pound to maintain the actual source IP address of the client rather than the IP address of the proxy itself. When enabling Tproxy, it's important to be aware of the topology requirements for Tproxy to work correctly. This is explained in the examples below.

Example 1 – Layer 7 (HAProxy) with Tproxy Enabled:

In this example, Tproxy is enabled with a layer 7 Virtual Server. The setup is shown in the following example.

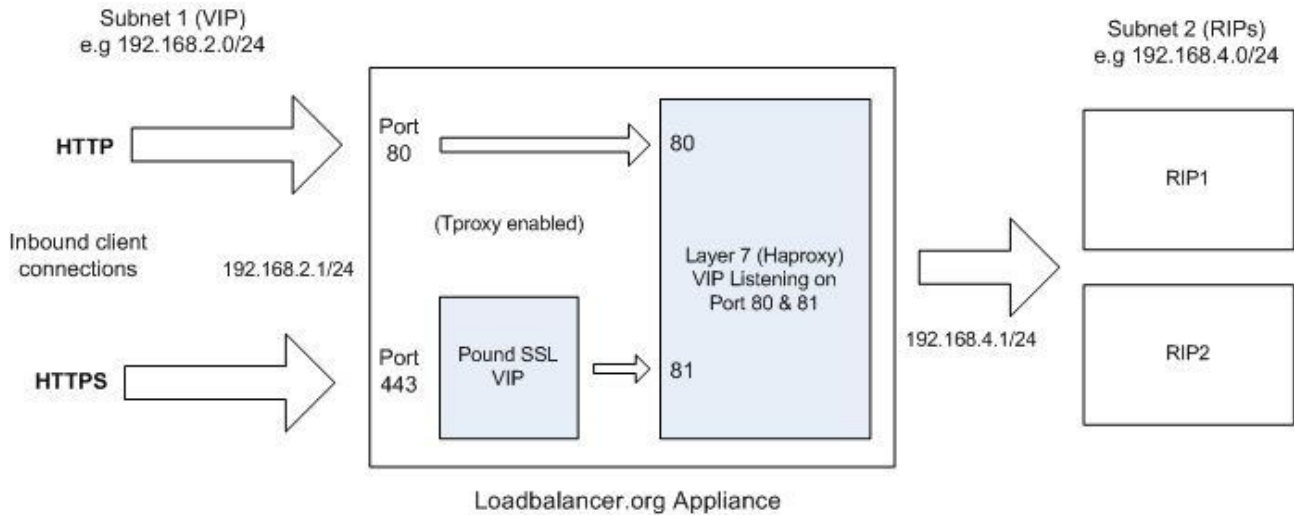


Topology Requirements / Notes

- The RIPs must be on a different subnet to the VIP – this can be achieved by using 2 IP addresses assigned to a single interface, or two separate interfaces (eth0 & eth1)
- Tproxy must be enabled using the WUI : open *Edit Configuration > Layer 7 – Advanced Configuration* and set **Transparent Proxy** to 'On'
- On the Real Servers, the default gateway must be configured to be an IP address on the load balancer. It's best to add an additional floating IP for this to allow failover to the slave

Example 2 – SSL (Pound) Termination with Tproxy Enabled:

In this example, Pound is also used to terminate SSL. Pound passes the decrypted traffic to a layer 7 backend VIP where the Real Servers are configured. This setup is shown in the following diagram.



Topology Requirements / Notes

- The RIPs must be on a different subnet to the VIP – this can be achieved by using 2 IP addresses assigned to a single interface, or two separate interfaces (eth0 & eth1)
- Configure the Layer 7 VIP to listen on 2 ports – e.g. 80 & 81, then use port 81 for the Pound backend and port 80 for client connections as shown above
- Tproxy must be enabled using the WUI : open *Edit Configuration > SSL – Advanced Configuration* and set **Transparent Proxy** to 'On' *N.B. This will also automatically enable Tproxy for HAProxy*
- On the Real Servers, the default gateway must be configured to be an IP address on the load balancer. It's best to add an additional floating IP for this to allow failover to the slave

Health Monitoring

The loadbalancer.org appliance supports both Real Server (back-end) and load balancer health checks.

Load balancer Health (Clustered Pair)

When a clustered pair is deployed rather than a single appliance, the load balancers are configured by default to use a serial connection to check the health of each other (for the VA, this defaults to ucast over the network when a clustered pair is configured). This permits failover to the slave unit if the master unit fails. Multiple checks can be configured between the appliances using the serial cable and network cables, as well as checks to a common node such as the default gateway. This allows a number of checks to be configured to ensure that failover only occurs when needed and 'split brain' (i.e. master and slave are both active) scenarios are avoided.

Heartbeat Communication Method

EDIT CONFIGURATION > MODIFY HEARTBEAT CONFIGURATION

Serial	<input checked="" type="checkbox"/>	?
Unicast	<input type="checkbox"/>	?
Broadcast (Deprecated)	Off ▼	?
UDP Port for broadcast & unicast	6694	?
Keepalive	3	?
Deadtime	10	?
Warntime	5	?
Ping node		?
Automatic Fail-back	<input checked="" type="checkbox"/>	?

Modify Heartbeat configuration

N.B. The screen shot above shows the configuration screen for the hardware appliance, for the VA the serial option is not available.

Serial Cable

This method requires a null modem cable (1 cable is supplied with each appliance) to be connected between the two load balancers in the cluster. This enables heartbeat checks to utilize the serial port (ttyS0 / ttyS1). This is the only method which is active by default, other methods must be enabled manually.



When the wizard is used to configure a VMware based clustered pair, heartbeat is automatically configured to use the network (ucast) for heartbeat.

Unicast (ucast)

This method of heartbeat communication uses unicast UDP between master and slave, with a destination port given by the UDP Port for broadcast & unicast parameter. When unicast is enabled, the load balancer determines the correct interface and IP addresses to use based upon the configured slave IP address. Make sure that the correct slave IP has been entered on the DNS & Hostname page before enabling unicast. Unicast is the preferred communication method if serial cannot be used.

Broadcast (bcast) - *Deprecated*

This method of heartbeat communication uses broadcast UDP between master and slave, with a destination port given by the UDP Port for broadcast & unicast parameter. Care must be taken when using broadcast on multiple pairs of load balancers in the same network. Each high-availability pair must operate on a different UDP port if they are not to interfere with each other. If heartbeat communication over the network is required, it is recommended that unicast be used in preference to broadcast.

Ping Node

Specify a mutually accessible IP address to test network availability. A good ping node to specify is the IP address of a router that both the master and slave can access (e.g. the default gateway). If the active appliance loses access to the ping node, but the passive appliance still has access, then a failover will occur. However, if both nodes lose access failure will not occur.

Auto-Failback

When the master returns to service after a failure do we transfer resources back to it. Sometimes it is useful to always fall back to the master. If you prefer to manually control this process, un-check this option.



If the master unit is configured first and a slave unit is added later, heartbeat will need to be restarted on both units for synchronization to occur. This can be done using : *Maintenance > Restart Services*. For more details on adding a slave unit refer to page 157.

Real Server Health

The appliance supports a range of health-check options for the real serves. These range from simple ping check to more complex negotiate options to determine that the underlying daemon / service is running. The options available do differ between Layer 4 and Layer 7 and are covered in the following sections.

Layer 4

At layer 4, Real Server health checking is provided by Ldirectord. This is integrated into Loadbalancer.org appliances and allows a full range of options to check that Real Servers are operational.

Using the WUI open : *Edit Configuration > Virtual Servers > Modify*

Check Type	Connect to port	?
Negotiate Check Service	HTTP	?
Check Port		?
Check Command		?
Virtual Host		?
Login		?
Password		?
Secret		?
Protocol	TCP	?
Granularity		?
Request to send	check.txt	?
Response expected	OK	?

Check Types

Negotiate connection – Sends a request and looks for a specific response (see service to check below)

Connect to port – Just do a simple connect to the specified port/service & verify that it's able to accept a connection

Ping server – Sends an ICMP echo request packet to the Real Server

External check – Use a custom file for the health check. Specify the filepath in the 'Check Command' field.

No checks, always Off – All Real Servers are off

No checks, always On – All Real Servers are on (no checking)

5 Connects, 1 Negotiate – Do 5 connect checks and then 1 negotiate check

10 Connects, 1 Negotiate – Do 10 connect checks and then 1 negotiate check

Negotiate Check Service

If negotiate is selected as the check type, the following methods are valid:

HTTP – use HTTP as the negotiate protocol (also requires filename, path + text expected)

HTTPS – use HTTPS as the negotiate protocol (also requires filename, path + text expected)

HTTP Proxy – Use an HTTP proxy check

FTP – use FTP as the negotiate protocol (also requires login/password, filename in the default folder)

IMAP (IPv4 only) – use IMAP as the negotiate protocol (requires login/password)

IMAPS (IPv4 only) - use IMAPs as the negotiate protocol (requires login/password)

POP – use POP as the negotiate protocol (also requires login/password)

POPS – use POPs as the negotiate protocol (also requires login/password)

LDAP (IPv4 only) – use LDAP as the negotiate protocol (also requires username/password)

SMTP – use SMTP as the negotiate protocol

NNTP (IPv4 only) – use NNTP as the negotiate protocol

DNS – use DNS as the negotiate protocol

MySQL (IPv4 only) – use MySQL as the negotiate protocol (also requires username/password)

SIP – use SIP as the negotiate protocol (also requires username/password)

Simple TCP – Sends a request string to the server and checks the response

RADIUS (IPv4 only) – use RADIUS as the negotiate protocol (also requires username/password)

none

Check Port

This can be used if the port to check is non standard, e.g., the service to check is HTTPS, but the port used is 4443 instead of the standard 443.

Check Command

The custom check script, used with the external check type. The script should be placed in /var/lib/loadbalancer.org, and given world read and execute permissions.

The following example illustrates how scripts can be constructed. This script uses the Linux command 'links' to connect to the real server, then uses the Linux command 'grep' to look for the text 'OK' in the file 'check.txt'. The variable 'EXIT_CODE' which indicates a pass or fail is then returned to Ldirectord to control whether the server should be left online or removed.

```
# Set Variables
REALIP="$3"
PORT="$2"
REQUEST="check.txt"
RESPONSE="OK"

# Check the page
links -dump https://$REALIP:$PORT/$REQUEST |grep -e $RESPONSE
if [ "$?" -eq "0" ]; then
EXIT_CODE="0"
else
EXIT_CODE="1"
fi

# Exit with result
exit $EXIT_CODE
```

NOTE:

\$2 and \$3 are Ldirectord variables that are passed to the script. The following Ldirectord variables are available and can be used as required:

- \$1 – the VIP address
- \$2 – the VIP port
- \$3 – the RIP address

Virtual Host

If the Real Server will only respond to a URL or 'virtualhost' rather than an ip address. You can specify the virtual host to request here.

Login

The login name to use with negotiate checks where authentication is required.

Password

The password to use with negotiate checks where authentication is required.

Secret

The secret to use with Radius servers.

Request to Send

This is used with negotiate checks and specifies the request to send to the server. The use of this parameter varies with the protocol selected in *Service to Check*. With protocols such as HTTP and FTP, this should be the object to request from the server. Bare filenames will be requested from the web or FTP root. With DNS, this should be either a name to look up in an A record, or an IP address to look up in a PTR record. With databases, this should be an SQL query. With LDAP, this should be the search base for the query. The load balancer will perform an (ObjectClass=*) search relative to this base. With Simple TCP, this should be a string to send verbatim to the server.

Response Expected

This is the response that must be received for the negotiate to be a success. The negotiate check succeed if the specified text (response) is found anywhere in the response from the web server when the file specified in the File to Check field is requested.

For example, a file called 'check.txt' could be placed in the default folder of the web server, this text file could just have the text **OK** in the file, then when the negotiate check runs, it would look for a file called 'check.txt' containing **OK**. If found, the test would succeed, if not found it would fail and no new sessions will be sent to that server.


Email Alerts

Specify the email alert address. This can also be configured at a global level to apply to all Layer 4 Virtual Servers in the WUI using *Edit Configuration > Layer 4 Advanced Configuration*.

Additional Health Check Settings

Using the WUI open : *Edit Configuration > Layer 4 – Advanced Configuration*

EDIT CONFIGURATION > ADVANCED CONFIGURATION

Layer 4		
Check Interval	<input type="text" value="6"/>	
Check Timeout	<input type="text" value="3"/>	
Negotiate Timeout	<input type="text" value="5"/>	
Failure Count	<input type="text" value="1"/>	
Quiescent	<input type="text" value="no"/>	
Email Alerts	<input type="text"/>	
Auto NAT	<input type="text" value="off"/>	
Multi-threaded	<input type="text" value="yes"/>	
Fallback	<input type="text" value="yes"/>	
Disable Write	<input type="text" value="off"/>	

Update

Check Interval

Layer 4 (Ldirectord) health check interval in seconds. If this setting is too low, you may induce un-expected Real Server downtime. For slower servers, this may need to be increased.

Check Timeout

Layer 4 (Ldirectord) health check timeout in seconds. If this setting is too low, you may induce un-expected Real Server downtime. For slower servers, this may need to be increased.

Negotiate Timeout

Layer 4 (Ldirectord) negotiate health check timeout in seconds. The negotiate checks may take longer to process as they involve more server side processing than a simple TCP socket connect check. If this setting is too low, you may induce unexpected Real Server downtime. For slower servers, this may need to be increased.

Failure Count

Layer 4 (Ldirectord) number of times a check has to fail before taking server offline. The time to detect a failure and take down a server will be (check interval + check timeout) * failure count.

Quiescent

When Quiescent is set to 'yes', on a health check failure the Real Server is not removed from the load balancing table, but the weight is set to 0. Persistent connections will continue to be routed to the failed server, but no new connections will be accepted.

When Quiescent is set to 'no', the server is completely removed from the load balancing table on a health check failure. Persistent connections will be broken and sent to a different Real Server.

Quiescent only applies to health checks – it has no effect on taking Real Servers offline in System Overview. To manually force a Real Server to be removed from the table, set Quiescent to no and arrange for the server to fail its health check. This may be done, for example, by shutting down the daemon or service, changing the negotiate check value, or shutting down the server.

Email Alerts

This is the default global setting for email alerts and is used for all Layer 4 Virtual Servers if no other address is specified at the individual VIP level.

Multi-Threaded

Perform health checks with multiple threads. Using multiple-threads for health checks will increase performance when you have a large number of Virtual Servers.

Fallback

Local Fallback server on / off switch. Configure whether the local (nginx) fallback server is active or not, sometimes you may want the local fallback server switched off so that it doesn't change the SNMP results table when activated. You may also want to disable it for security purposes.

Layer 7

At layer 7, Real Server health checking is handled by HAProxy. This is integrated into Loadbalancer.org appliances and allows a range of options to check that Real Servers are operational.

N.B. The default health-check for a new VIP is a port connect.

Using the WUI open : *Edit Configuration > Virtual Servers (HA Proxy) > Modify*

Check Port	<input type="text"/>	?
Request to send	<input type="text"/>	?
Response expected	<input type="text"/>	?

Check Port

Specify a different port for health checks. If specified this setting overrides the default checkport, useful when you are balancing multiple ports.

Request to Send

Specify a specific file for the health check. Open the specified file and check for the response expected, useful for checking a server sided script to check the health of the back-end application.

Response Expected

The content expected for a valid health check on the specified file. The response expected can be any valid regular expression statement.

Example:

If the server has a virtual directory called /customers, with a default page that contained the word 'welcome' the required setup would be as follows:

Request to send: **customers**
Response expected: **welcome**

These settings would configure the following check directives in the HAProxy configuration file:




```
option httpchk GET /customers HTTP/1.0
http-check expect rstring welcome
```

(N.B. the back-slash character before 'customers' is added automatically)

Provided that the load balancer can access the page and see the text 'welcome', the health-check would pass.

Additional Health Check Settings

Using the WUI open : *Edit Configuration > Layer 7 – Advanced Configuration*

Interval	<input type="text" value="2000"/>	
Rise	<input type="text" value="2"/>	
Fall	<input type="text" value="3"/>	

Interval

Interval between health checks. This is the time interval between Real Server health checks in milliseconds.

Rise

Number of health checks to Rise. The number of positive health checks required before re-activating a Real Server.

Fall

Number of health checks to Fall. The number of negative health checks required before de-activating a Real Server.

Simulating Health-Check Failures

It may not always be possible to take a server offline to check that health-checks are working correctly. In these cases, firewall rules can be used. The following rules can be configured at the console, using SSH or via the WUI under *Edit Configuration > Execute a Shell Command*

to disable a Real Server:

```
iptables -A OUTPUT -p tcp --dport <Check Port> -d <REAL-SERVER-IP> -j DROP
```

e.g. `iptables -A OUTPUT -p tcp --dport 80 -d 192.168.65.60 -j DROP`

to re-enable a Real Server:

```
iptables -D OUTPUT -p tcp --dport <Check Port> -d <REAL-SERVER-IP> -j DROP
```

e.g. `iptables -D OUTPUT -p tcp --dport 80 -d 192.168.65.60 -j DROP`

N.B. Make sure these rule are cleared after testing & verification is complete!

Fallback Server Settings

The appliances uses a NGINX for the local fallback server. The fallback server is activated under the following conditions for Layer 4 & Layer 7 Virtual Servers:

Layer 4

The fallback page is displayed when all Real Servers fail. The fallback page is NOT displayed when servers are taken offline manually via the WUI.

At layer 4, to cause the fallback page to be displayed when Real Servers are taken offline, you need to force all Real Servers to fail their health check by for example disabling the relevant service on each Real Server.

Layer 7

For layer 7 VIPs the fallback page is displayed when all Real Servers are unavailable *AND* when all are taken offline via the WUI. The page can be hosted on the load balancer or on an external server. Set the Fallback Server option of the VIP accordingly.

The local fallback page is modified using the WUI : *Maintenance > Fallback Page*

MAINTENANCE > FALLBACK PAGE

```
<html>
<head>
<title>The page is temporarily unavailable</title>
<style>
body { font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body bgcolor="white" text="black">
<table width="100%" height="100%">
<tr>
<td align="center" valign="middle">
The page you are looking for is temporarily unavailable.<br/>
Please try again later.<br/>
(port reminder 9080)
</td>
</tr>
```

Update

- The local fallback server is an NGINX instance that by default listens on ports 80 & 9081
- When a layer 7 HAProxy VIP is added, NGINX is automatically configured to listen on port 9081 only
- You can use any valid HTML for the default page, simply copy and paste the required HTML into the Fallback Page using the Maintenance menu

N.B. If you have a master and slave load balancer then you must change this on both servers. The fallback server on the load balancer is an implementation of Nginx.

Fallback page on the load balancer :



If you are using the load balancer for your holding page and your web servers are offline then the local Nginx server is exposed to hacking attempts, if you are concerned about this you can change the fallback server to one of your internal servers.

Fallback page on an external dedicated server :



For DR mode the fallback server must be listening on the same port as the VIP. Also, don't forget to solve the ARP problem for the dedicated fallback server.



For NAT mode don't forget to set the default gateway of the fallback server to the internal IP of the load balancer or when you have 2 appliances in a cluster, to a floating IP.

Advanced Firewall Considerations



Whilst the load balancer is capable of supporting complex firewall rules, we do not recommend using the load balancer as your main bastion host. We recommend that the load balancer is deployed behind your external firewall.

If you want a quick and simple firewall script then use the firewall lock down wizard. However be very wary of locking yourself out of the system if you are accessing the unit remotely.

If you want to set up firewall rules some points to consider are:

1. All Virtual Server connections are dealt with on the INPUT chain not the FORWARD chain
2. The WUI runs on HTTP port 9080 and HTTPS port 9443
3. SSH on the load balancer listens on the standard port (22)
4. SNAT & DNAT is handled automatically for all layer 4 NAT mode (LVS) and layer 7 (HAProxy) based Virtual/Real load balanced services
5. You can use the standard Linux filters against spoofing attacks and syn floods
6. LVS has built in DOS attack filters that can be implemented
7. Plenty of extra information is available on the Internet relating to Linux Netfilter and LVS (*Linux Virtual Server*)

Firewall Marks (Layer 4)

Using firewall marks enables multiple ports to be combined into a single virtual service. A common use of this feature is to aggregate port 80 (http) and port 443 (https) so that when a client fills their shopping cart on via HTTP, then move to HTTPS to give their credit card information, they will stay on the same Real Server.

Firewall Marks – Auto Configuration

For example, to configure an HTTP/HTTPS Virtual Server, simply specify port 80 & 443 separated by a comma in the 'Virtual Server Ports' field. This will automatically configure the load balancer for firewall marks.

Label	<input type="text" value="HTTP_Cluster"/>	
Virtual Server IP address	<input type="text" value="192.168.50.1"/>	
Virtual Server Ports	<input type="text" value="80,443"/>	
Forwarding Method	<input type="text" value="NAT"/>	
Persistent	<input type="text" value="yes"/>	
Protocol	<input type="text" value="TCP"/>	
<input type="button" value="Update"/>		

For NAT mode VIPs, leave the Real Server port blank as shown below.

Label	<input type="text" value="HTTP1"/>	
Real Server IP Address	<input type="text" value="192.168.50.2"/>	
Real Server Port	<input type="text"/>	
Weight	<input type="text" value="1"/>	
Minimum Connections	<input type="text" value="0"/>	
Maximum Connections	<input type="text" value="0"/>	
<input type="button" value="Update"/>		

For Layer 4 DR mode VIPs, there is no Real Server Port field since port translation is not possible in this mode.



HEALTH CHECK PORT: For DR mode the check port is automatically set to be the first port in the list. For example, if ports 80 & 443 are defined for the VIP, the check port is automatically set to port 80. When using NAT mode, the check port must be set manually.

Firewall Marks – Manual Configuration

Firewall Marks can also be configured manually if required. The basic concept is to create a firewall rule that matches incoming packets to a particular IP address / port(s) and mark them with an arbitrary integer. A Virtual Server is also configured specifying this firewall mark instead of the IP address.

EXAMPLE 1 – Setup a new DR Mode Firewall Mark when no Initial VIP has been Created

Step 1 – Create the New VIP

- Using the WUI, go to *Edit Configuration > Layer 4 – Virtual Servers*
- Click **[Add a new Virtual Server]**
- Instead of entering an IP address, enter a numeric value representing the 'mark' as shown below

Label	<input type="text" value="Server_Cluster"/>	?
Virtual Server IP address	<input type="text" value="1"/>	?
Virtual Server Ports	<input type="text"/>	?
Forwarding Method	<input type="text" value="Direct Routing"/>	?
Persistent	<input type="text" value="yes"/>	?
Protocol	<input type="text" value="Firewall Marks"/>	?
<input type="button" value="Update"/>		






- Leave the *Virtual Server Ports* field blank (the ports will be defined in the firewall script in step 5)
N.B. Some earlier versions may require a port to be entered, in this case simply enter '0' in this field
- Set the *Forwarding Method* to Direct Routing
- Set *Persistence* to Yes
- Set *Protocol* to Firewall Marks
- Click **Update**

Step 2 – Define a Health-Check Port

- Using the WUI, go to *Edit Configuration > Layer 4 – Virtual Servers*
- Click **[Modify]** next to the new Virtual Server
- Enter the appropriate value in the *Check Port* field
- Click **Update**

Step 3 – Add the Real Servers

- Using the WUI, go to *Edit Configuration > Layer 4 – Real Servers*
- Click **[Add a new Real Server]**
- Enter the required details as shown below

Label	<input type="text" value="Server1"/>	
Real Server IP Address	<input type="text" value="192.168.100.10"/>	
Weight	<input type="text" value="1"/>	
Minimum Connections	<input type="text" value="0"/>	
Maximum Connections	<input type="text" value="0"/>	
<input type="button" value="Update"/>		

- Click **Update**

Step 4 – Add the Associated Floating IP Address for the VIP

- Using the WUI, go to *Edit Configuration > Floating IPs*
- Add a floating IP that corresponds to the required VIP, in this example 192.168.100.1

EDIT CONFIGURATION > ADD NEW FLOATING IP

<input type="text" value="192.168.100.1"/>
<input type="button" value="Update"/>

- Click **Update**

Step 5 – Modify the Firewall Script

- Using the WUI, go to *Maintenance > Firewall Script*
- Uncomment / modify the example firewall marks section as shown below

MAINTENANCE > FIREWALL SCRIPT

```
##### FIREWALL MARKS #####

# Now setup any Firewall marks that are required
# Firewall marks allows you to associate multiple ports with one VIP
# This is useful if you need to keep HTTP & HTTPS persistent
# This example marks HTTP & HTTPS connections only

VIP1="192.168.100.1"
iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 80 -j MARK --set-mark 1
iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 443 -j MARK --set-mark 1

# If you then add a virtual service with an address of '1' rather
# than 'IP:port' it will balance HTTPS & HTTP,
# this would usually be set as persistent...
```

Update

- Click **Update**



NOTE: For a clustered pair, the firewall script changes mentioned above must also be completed on the slave unit.




EXAMPLE 2 – Setup a Firewall Mark by Modifying an Existing VIP

In the case, the floating IP address will already exist so does not need to be created manually.

Step 1 – Modify the Existing Virtual Server

- Using the WUI, go to *Edit Configuration > Layer 4 – Virtual Servers*
- Click **[Modify]** next to the relevant VIP
- Change the IP address to the chosen 'mark' value
- Clear the *Virtual Server Ports* field

N.B. Some earlier versions may require a port to be entered, in this case simply enter '0' in this field

Label	<input type="text" value="Server_Cluster"/>	
Virtual Server IP address	<input type="text" value="1"/>	
Virtual Server Ports	<input type="text"/>	

- Set the *Protocol* field to Firewall Marks
- Click **Update**

Step 2 – Define a Health-Check Port

- Using the WUI, go to *Edit Configuration > Layer 4 – Virtual Servers*
- Click **[Modify]** next to the new Virtual Server
- Enter the appropriate value in the *Check Port* field
- Click **Update**

Step 3 – Modify the Firewall Script

- Using the WUI, go to *Maintenance > Firewall Script*
- Uncomment / modify the example firewall marks section as shown in the following example. Additional ports can be added as required by adding additional iptables entries and specifying the appropriate port / protocol.

MAINTENANCE > FIREWALL SCRIPT

```
##### FIREWALL MARKS #####

# Now setup any Firewall marks that are required
# Firewall marks allows you to associate multiple ports with one VIP
# This is useful if you need to keep HTTP & HTTPS persistent
# This example marks HTTP & HTTPS connections only

VIP1="192.168.100.1"
iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 80 -j MARK --set-mark 1
iptables -t mangle -A PREROUTING -p tcp -d $VIP1 --dport 443 -j MARK --set-mark 1

# If you then add a virtual service with an address of '1' rather
# than 'IP:port' it will balance HTTPS & HTTP,
# this would usually be set as persistent...
```

Update

- Click **Update**



NOTE: For a clustered pair, the firewall script changes mentioned above must also be completed on the slave unit.

Firewall Mark Notes:

- When using firewall marks the load balancer forwards traffic to the selected Real Server without changing the destination port. So, incoming traffic to port 80 on the Virtual IP will be forwarded to port 80 on one of the Real Servers. Likewise, incoming traffic to port 443 will be forwarded to port 443 on the same Real Server.
- You can only have one health check port assigned, so if you are grouping port 80 and 443 traffic together you can only check one of these ports, typically this would be port 80.
- You can specify a range of ports rather than a single port as shown below:

```
iptables -t mangle -A PREROUTING -p tcp -d 10.141.12.34 -dport 1024:5000 -j MARK --set-mark 1
```

this specifies destination ports from 1024 to 5000

- You can leave the upper limit blank to use the default upper limit as shown below:

```
iptables -t mangle -A PREROUTING -p tcp -d 10.141.12.34 -dport 1024: -j MARK --set-mark 1
```

this specifies destination ports from 1024 to 65536

- You can specify a range of IP addresses as shown below:

```
iptables -t mangle -A PREROUTING -p tcp -m iprange -dst-range 10.141.12.34-10.141.12.40 --dport 80 -j MARK --set-mark 1
```

this specifies the destination IP address as a range from 10.141.12.34 to 10.141.12.40

Layer 4 Persistence Considerations

Persistence State Table Replication

If you want the current persistent connection table to work when the master load balancer swaps over to the slave then you can start the synchronization daemons on each load balancer to replicate the data in real time.

First login to lbmaster using SSH or the console, then as root run the following command:

```
ipvsadm --start-daemon master  
ipvsadm --start-daemon backup
```

Then login to lbslave using SSH or the console, then as root run the following command:

```
ipvsadm --start-daemon master  
ipvsadm --start-daemon backup
```

N.B. To ensure that these sync daemons are started on each reboot put these commands in the rc.firewall. This can be done via the WUI using Maintenance > Firewall Script. Make sure that the full path is specified in the firewall script , i.e.

```
/usr/local/sbin/ipvsadm --start-daemon master  
/usr/local/sbin/ipvsadm --start-daemon backup
```

After a few seconds you can confirm that it is working by seeing the output from:

```
ipvsadm -Lnc
```

N.B. This is the same command that the 'Layer 4 Current Connections' report is based on.

This should give the same output as running the same command on lbmaster i.e. The state table is being replicated.



Setting this option can generate a high level of connection state synchronization data between the master and slave load balancers.

Server Maintenance when using Persistence

A protocol with a long session & persistence enabled such as Terminal Server RDP maintenance can become problematic because clients that disconnect and re-connect will still go to the same server for the length of the persistence timeout. This behavior has already been modified on the Loadbalancer.org appliances (from v6.5) so that when a client disconnects the persistence template is cleared forcing them to re-connect to a different server.

In the unlikely event that you wish to disable this feature globally use the following commands from the console:

```
echo 0 > /proc/sys/net/ipv4/vs/expire_quiescent_template  
echo 0 > /proc/sys/net/ipv4/vs/expire_nodead_conn
```

N.B. This can be made a permanent setting on both load balancers by adding it to the /etc/sysctl.conf file.

If you are using negotiate checks you may also want to use the quiescent=no global option to ensure that if a server fails a negotiate check but is still technically working the connections are forced to fail over rather than being drained gradually.

SNMP Reporting

Native SNMP support can be enabled on the appliance. This is a simple case of enabling the SNMP service:

```
service snmpd start
chkconfig snmpd on
```

('chkconfig snmpd on' forces snmp to start on appliance reboot)

SNMP for Layer 4 Based Services

The root OID for Layer 4 based services is: 1.3.6.1.4.1.8225.4711

You can test if everything works by invoking:

```
shell> snmpwalk -c public -v 2c -m LVS-MIB localhost 1.3.6.1.4.1.8225.4711

LVS-MIB::lvsVersion.0 = STRING: "1.2.0"
LVS-MIB::lvsNumServices.0 = INTEGER: 2
LVS-MIB::lvsHashTableSize.0 = INTEGER: 4096
LVS-MIB::lvsTcpTimeOut.0 = INTEGER: 900
LVS-MIB::lvsTcpFinTimeOut.0 = INTEGER: 120
LVS-MIB::lvsUdpTimeOut.0 = INTEGER: 300
LVS-MIB::lvsDaemonState.0 = INTEGER: none(0)
...
etc.
```

N.B. LVS-MIB.txt can be downloaded from : <http://www.loadbalancer.org/download/SNMP/>

You can also use all the usual MIB2 counters and gauges such as network and CPU etc.

SNMP for Layer 7 Based Services

The root OID for Layer 7 front-end services is: 1.3.6.1.4.1.29385.106.1.0

The root OID for Layer 7 back-end services is: 1.3.6.1.4.1.29385.106.1.1

Front end stats are returned by invoking:

```
[root@lbmaster ~]# snmpwalk -c public -v2c 127.0.0.1 1.3.6.1.4.1.29385.106.1.0

SNMPv2-SMI::enterprises.29385.106.1.0.0.1.0 = STRING: "stats"
SNMPv2-SMI::enterprises.29385.106.1.0.1.1.0 = STRING: "FRONTEND"
SNMPv2-SMI::enterprises.29385.106.1.0.2.1.0 = ""
SNMPv2-SMI::enterprises.29385.106.1.0.3.1.0 = ""
SNMPv2-SMI::enterprises.29385.106.1.0.4.1.0 = STRING: "0"
SNMPv2-SMI::enterprises.29385.106.1.0.5.1.0 = STRING: "0"
SNMPv2-SMI::enterprises.29385.106.1.0.6.1.0 = STRING: "2000"
...
etc.
```

Back end stats are returned by invoking:

```
[root@lbmaster ~]# snmpwalk -c public -v2c 127.0.0.1 1.3.6.1.4.1.29385.106.1.1

SNMPv2-SMI::enterprises.29385.106.1.1.0.1.0 = STRING: "stats"
SNMPv2-SMI::enterprises.29385.106.1.1.1.1.0 = STRING: "BACKEND"
SNMPv2-SMI::enterprises.29385.106.1.1.2.1.0 = STRING: "0"
SNMPv2-SMI::enterprises.29385.106.1.1.3.1.0 = STRING: "0"
SNMPv2-SMI::enterprises.29385.106.1.1.4.1.0 = STRING: "0"
SNMPv2-SMI::enterprises.29385.106.1.1.5.1.0 = STRING: "0"
SNMPv2-SMI::enterprises.29385.106.1.1.6.1.0 = STRING: "2000"
SNMPv2-SMI::enterprises.29385.106.1.1.7.1.0 = STRING: "0"
SNMPv2-SMI::enterprises.29385.106.1.1.8.1.0 = STRING: "0"
...
etc.
```

Server Feedback Agent

The load balancer can modify the weight (amount of traffic) of each server by gathering data from either a custom agent or an HTTP server. Just set the Virtual Servers feedback method to agent or http as required. A telnet to port 3333 on a Real Server with the agent installed will return the current CPU idle as an integer value in the range 0 – 100.

The load balancer typically expects a 0-99 integer response from the agent which relates to the CPU idle state, i.e. a response of 92 would imply that the Real Servers CPU is 92% idle. The load balancer will then use the formula $(92/10 * \text{requested_weight})$ to find the new optimized weight. Using this method an idle Real Server will get 10 times as many new connections as an overloaded server.

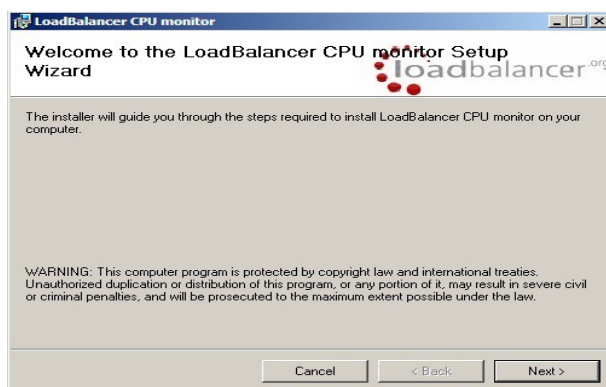
N.B. The feedback agent will never offline a server, only the standard health check can take a server offline.

Installing the Windows Agent

The feedback agent can be downloaded from:

<http://www.loadbalancer.org/download/agent/Windows/LBCPUMonInstallation.msi>

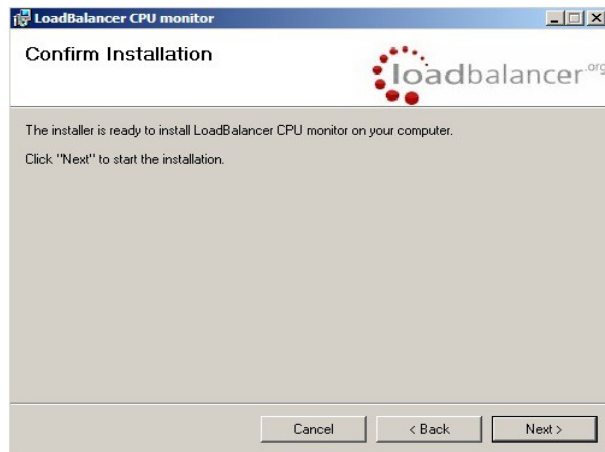
run LBCPUMonInstallation.msi



click next



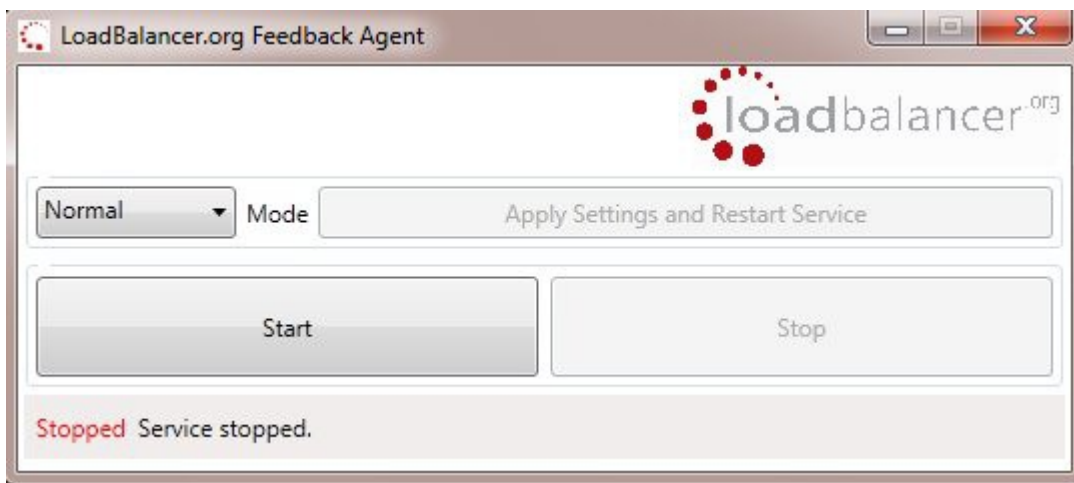
select the installation folder and click next



click next to start the installation

Starting the Agent

Once the installation has completed, you'll need to start the service on the real servers. The service is controlled by the Feedback Agent Monitor program that is also installed along with the Agent. The monitor can be accessed on the Windows server from : *All Programs > Loadbalancer.org > Monitor*. It's also possible to start the service using the services snap-in – the service is called 'Loadbalancer CPU monitor'.




- To start the service, click **Start**
- To stop the service, click **Stop**

N.B. The agent should be installed on all real servers in the cluster

Using the Windows Agent

The Feedback Method for the Virtual Server must be changed as follows:

- Go to *Edit Configuration > Virtual Servers*
- Click [**Modify**] next to the Virtual Server
- Change the Feedback Method to 'agent'



Feedback Method: agent [?] [Update]

- Click Update

Prior to installing & activating the agent, *View Configuration > System Overview* would look similar to the following (weights set to the default of 1) :

Key cluster healthy cluster may need attention cluster is down real server deliberately offline

TS-FARM1 - 192.168.23.165:3389 total connections - active: 0 inactive: 0						
Label	IP	Method	Weight	Active conns	Inactive conns	
TS1	192.168.23.20:3389	DR	1	0	0	take offline
TS2	192.168.23.21:3389	DR	1	0	0	take offline

Once the agents are installed on the Real Server and the feedback method is changed, the weights for the Real Servers are updated :

Key cluster healthy cluster may need attention cluster is down real server deliberately offline

L4TS - 192.168.23.165:3389 total connections - active: 2 inactive: 0						
Label	IP	Method	Weight	Active conns	Inactive conns	
TS1	192.168.23.20:3389	DR	10	1	0	take offline
TS2	192.168.23.21:3389	DR	10	1	0	take offline

If one of the Real Servers is heavily loaded, the weighting is adjusted accordingly – a lower weight causes less sessions for that server. Here, CPU utilization on TS1 is high so the weight has been reduced to 1 :

Key cluster healthy cluster may need attention cluster is down real server deliberately offline

L4TS - 192.168.23.165:3389 total connections - active: 1 inactive: 0						
Label	IP	Method	Weight	Active conns	Inactive conns	
TS1	192.168.23.20:3389	DR	1	0	0	take offline
TS2	192.168.23.21:3389	DR	10	1	0	take offline

Installing the Linux/Unix Agent

Download the agent from <http://www.loadbalancer.org/download/agent/Linux/>

N.B. The agent must be installed on all backend servers, not the load balancer.

```
apt-get install xinetd (if not already installed)

Insert this line into /etc/services
lb-feedback      3333/tcp                                # Loadbalancer.org feedback daemon

Then:
cp lb-feedback.sh /usr/bin/lb-feedback.sh
chmod +x /usr/bin/lb-feedback.sh
cp lb-feedback /etc/xinetd.d/lb-feedback
chmod 644 /etc/xinetd.d/lb-feedback

/etc/init.d/xinetd restart

Testing:
telnet 127.0.0.1 3333

Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
95
Connection closed by foreign host.
Connection closed by foreign host.
```

Custom HTTP Agent

You can use any HTTP server responding on port 3333 to give feedback information to the load balancer. The format of this information must be an integer number of 0-100 without any header information. Using this method you can generate a custom response based on your applications requirements i.e. a mixture of memory usage, IO, CPU etc.

Changing the Local Date, Time & Time Zone

You can change the time & time zone from the web interface using:

Edit Configuration > System date & time

To set the date and time at the console use the following commands:

```
date --set 2012-03-10 (yyyy-mm-dd)
date --set 21:08:00 (hh:mm:ss)
```

To set the hardware clock to the system time:

```
hwclock --systohc
```

NTP Configuration

If the load balancer has ntp access to the Internet you can force an update:

```
ntpdate time.nist.gov
```

N.B. This is already in the root cron job in /etc/crontab.

The load balancers local clock is updated once a day using ntp, this requires that your default gateway and DNS are set correctly.

Timezone can be Coordinated Universal Time (UTC) or GMT based like GMT, GMT+1 hour, GMT-1 hour, and so on. Please consider that the GMT+/-X format as it is returned by the system differs from the GMT +/- X hours format. The GMT+/-X based statement follows the POSIX standard which means that GMT+X is X hours west of Greenwich. GMT-X means X hours east of Greenwich. So GMT+X means GMT-X hours and vice-verse.



When using a clustered pair (i.e. master & slave) manual time & time zone changes on the master will not be automatically replicated to the slave, therefore the slave must also be set manually.

Restoring Manufacturer's Settings

The load balancer settings can be reset to factory default values in two ways:

From the WUI

Maintenance > Backup & Restore > Restore Manufacturer's Defaults

Running this will remove all custom configuration from the load balancer. All VIPs and RIPs will be removed and the IP address configured for eth0 will be set to 192.168.2.21 provided that no other device has this address, if it does, then the current IP address will remain.

From the Console

```
lbrestore
```

Force Master/Slave Take-Over In a Clustered Pair

Force the Slave to Become Active & Master Passive

On the slave:

```
/usr/local/sbin/hb_takeover.php all
```

Force the Master to Become Active & Slave Passive

On the master:

```
/usr/local/sbin/hb_takeover.php all
```

N.B. these commands can either be run on the console, at a terminal session or via the WUI using : Edit Configuration > Execute Shell Command

Application Specific Settings

FTP

FTP is a multi-port service in both active and passive modes:

active 20,21

passive 21,high_port

Layer 4 Virtual Servers for FTP

When configuring a Virtual Server at layer 4 for FTP, simply setup a layer 4 VIP in the normal way and set the Virtual Server / Real Server port field to port 21. Where Firewall Marks are required to handle other FTP ports, these will be configured automatically. This applies to both active and passive mode. In NAT mode, the `ip_vs_ftp` module is used to ensure that the client connects back via the load balancer rather than attempting to connect directly to the Real Server.

FTP Layer 4 Negotiate Health Check

You can modify the layer 4 Virtual Server so that rather than doing a simple socket connect check, it will actually attempt to log into the FTP server and read a file for a specific response:

Check Type	Negotiate connection	?
Negotiate Check Service	FTP	?
Check Port	21	?
..		
..		
Request to send	check.txt	?
Response expected	OK	?

Key Points:

- Change the *check type* to Negotiate Connection
- Make sure the *Negotiate Check Service* is set to FTP
- Specify a suitable *login* and *password* for the FTP server
- Specify the file to check using the *Request to send* field (defaults to the root directory)
- The file is parsed for the *Response expected* that you specify

FTP Recommended Persistence Settings

When you start using multiple FTP servers in a cluster you need to be aware of the effects of a client switching server. For sites that are download only, you generally don't need any special settings on the load balancer as the connection will usually stay on the same server for the length of the connection. You may however wish to force persistence to something sensible like 15mins (If you go higher remember to change the global TCP timeouts).

If you are using the FTP servers for upload it is recommended to use a single FTP server for uploads and then replicate the data to the read only cluster for downloads (or use a clustered file system). For upload it is especially important to use persistence.

Automatically resuming a broken download is no problem even if you switch servers in a cluster on re-connect. This is because the FTP resume functionality is client based and does not need any server session information.

Layer 7 Virtual Servers for FTP

Active Mode

In active mode, the FTP server connects back to the client, so it must be aware of the clients IP address. At layer 7, Tproxy can be used to enable transparency. For this to work two subnets must be used – the VIP in one subnet, the RIPs in another (for more details on Tproxy, please refer to pages 119-120).

Also, to ensure that the client receives a connection from the same address that it established the control connection to, an iptables SNAT rule must be defined in the firewall script. The format of the required rule is as follows:

```
iptables -t nat -A POSTROUTING -p tcp -s <FTP-Server-IP> -j SNAT --to-source <FTP-VIP-IP>
```

e.g.

```
iptables -t nat -A POSTROUTING -p tcp -s 10.20.1.1 -j SNAT --to-source 192.168.20.1
```

N.B. This rule can be added to the firewall script using the WUI : Maintenance > Firewall Script

Active Mode – Key Points:

- Use separate subnets for the VIP & RIPs
- Enable Tproxy
- Set the default gateway on the real servers to be the internal interface of the load balancer
- Setup a layer 7 VIP listening on port 21 & configure the RIPs also to listen on port 21
- Ensure the Layer 7 Protocol is set to 'Other TCP'
- Add the SNAT firewall rule

Passive Mode

In passive mode, all connections are initiated by the client. The server passes the client a port to use for the inbound data connection. By default, ftp servers can use a wide range of ports for the inbound connection and it is often useful to limit this range. The next section “Limiting Passive FTP ports” on page 150 covers this for a range of OS / ftp servers.

Passive Mode – Key Points:

- It's sensible to use a controlled passive port range
- Configure the VIP to listen on port 21 and also the passive range selected, e.g. 50000-50100
- Configure the RIPv without specifying a port
- Ensure the Layer 7 Protocol is set to 'Other TCP'
- To ensure the correct address is passed back to the client, specify the external address for the FTP server:

e.g.

- for Windows 2008 use the 'External IP address of Firewall' field

- for Linux vsftpd, use the directive : `pasv_address=xxx.xxx.xxx.xxx`

Windows 2008 Example:

Select 'FTP Firewall Support' for the specific ftp site, then specify the servers external address – this may be the VIP address or the external firewall address depending on how your system is configured.



FTP Firewall Support

The settings on this page let you configure your FTP server to accept passive connections from an external firewall.

Data Channel Port Range:

50000-50100

Example: 5000-6000

External IP Address of Firewall:

192.168.110.250

Example: 10.0.0.1

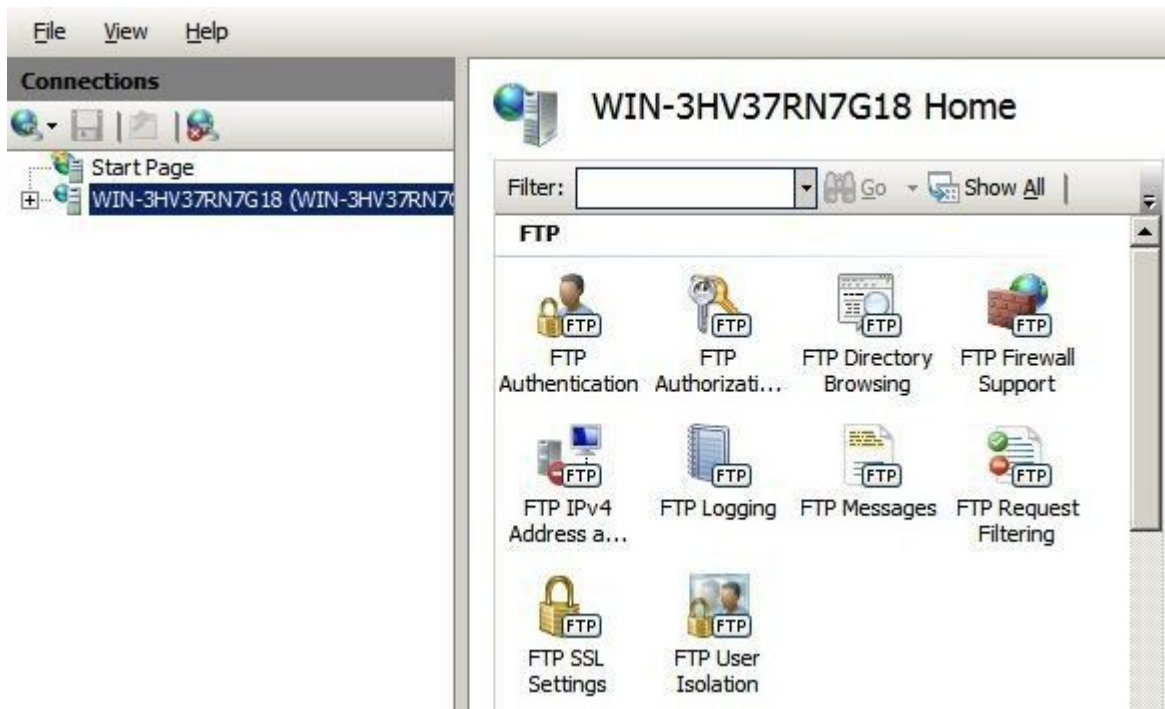
in the 'External IP Address of Firewall' field specify the Virtual Servers IP address (VIP)

Limiting Passive FTP Ports

Limiting passive ports allows your firewall to be more tightly locked down. The following sections show how this is achieved for a range of Operating Systems / FTP servers.

For Windows 2008

Open the IIS Management console, highlight the server node, then double-click the FTP Firewall Support icon.



The following screen will be displayed:



Specify a suitable range, in the example above this is 50000-50100

After any changes have been made, make sure that you restart the Microsoft FTP service.

For Windows 2003

a) Enable Direct Metabase Edit

1. Open the IIS Management Console
2. Right-click on the Local Computer node
3. Select **Properties**
4. Make sure the **Enable Direct Metabase Edit** checkbox is checked

b) Configure PassivePortRange via ADSUTIL script

1. Click **Start**, click **Run**, type cmd, and then click **OK**
2. Type cd Inetpub\AdminScripts and then press ENTER
3. Type the following command from a command prompt
adsutil.vbs set /MSFTPSVC/**PassivePortRange** "50000-50100"
4. Restart the FTP service

For Windows 2000

Configure PassivePortRange via the Registry Editor

1. Start Registry Editor (Regedt32.exe)
2. Locate the following registry key:
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Msftpsvc\Parameters
3. Add a value named "PassivePortRange" (without the quotation marks) of type REG_SZ
4. Close Registry Editor
5. Restart the FTP service

(SP4 or higher must be installed for this to work)

N.B. The range that FTP will validate is from 5001 to 65535

For Linux

in vsftpd, the following line can be added to the vsftpd.conf file to limit the port range:

pasv_max_port – max is 65535
pasv_min_port – min is 1024

in proftpd, the following line can be added to the proftpd.conf file to limit the port range:

PassivePorts 50000 – 50100

in pureftpd, the following startup switch can be used:

-p --passiveportrange <min port:max port>

Terminal Services & RDP

Layer 4 – IP Persistence

RDP is a TCP based service usually on port 3389. Because of the nature of a Terminal Server you'll want the clients to reconnect to the same server so that you maintain the session. The common setting to use with Terminal Server is *persistence=3600* (1 hour). This means that when a client reconnects within this time, they will be sent to the same terminal server. If a client is idle for more than 1 hour, then the load balancer will treat the next connection as a new connection and possibly take them to a different server.

Label	<input type="text" value="RDP_Cluster1"/>	?
Virtual Server IP address	<input type="text" value="192.168.2.165"/>	?
Virtual Server Ports	<input type="text" value="3389"/>	?
Persistent	<input type="text" value="Yes"/>	?
Persistence Timeout	<input type="text" value="3600"/>	?
Scheduler	<input type="text" value="Weighted Least Connection"/>	?

Layer 7 – RDP Cookies

In some instances source IP persistence can result in uneven load balancing. This would normally happen if you have a large number of users coming through a corporate firewall or proxy where network address translation is occurring, in this case all associated users would typically have the same source IP address and would therefore be directed at the same real (back-end) server.







If you have this issue, the Load balancer also supports persistence based on RDP cookies. This method utilizes the cookie sent from the client in the initial Connection Request PDU (msthash). This cookie is created when the username is entered at the first client login prompt (mstsc.exe). Note that if the username is not entered here, the cookie is not created. An associated persistence entry is also created in a stick table on the load balancer for each connection.

Label	<input type="text" value="RDP_Cluster2"/>	?
Virtual Server IP address	<input type="text" value="192.168.2.166"/>	?
Virtual Server Ports	<input type="text" value="3389"/>	?
Layer 7 Protocol	<input type="text" value="Other TCP"/>	?
Persistence mode	<input type="text" value="RDP Cookie"/>	?
Balance mode	<input type="text" value="Round Robin"/>	?

Initial connections are distributed to the Real Servers based on the Balance mode selected. Client re-connects utilize the stick table to return the client to the same server first connected to. This enables clients to reconnect to their disconnected sessions.

Layer 7 – Microsoft Connection Broker / Session Directory

It's also possible to configure the load balancer to interact with Session Directory / Connection Broker by enabling Routing Token Redirection mode. This mode allows the reconnection of disconnected sessions by utilizing a routing token to enable the load balancer to re-connect the client to the correct terminal server.

Label	<input type="text" value="RDP_Cluster3"/>	
Virtual Server IP address	<input type="text" value="192.168.2.167"/>	
Virtual Server Ports	<input type="text" value="3389"/>	
Layer 7 Protocol	<input type="text" value="Other TCP"/>	
Persistence mode	<input type="text" value="MS Connection Broker"/>	
Balance mode	<input type="text" value="Least Connections"/>	



Make sure that the Security Layer setting of the RDP connection properties are set to RDP Security Layer, otherwise the RDP Cookie may be encrypted and will not be readable causing persistence to break.



For additional information, please refer to our RDP Deployment Guide available here:
http://www.loadbalancer.org/pdffiles/Microsoft_Terminal_Services_Deployment_Guide.pdf.

Appliance Software Updates

Loadbalancer.org continually develop and add new & improved features to the appliance. To ensure that customers can benefit from this development and can also receive bug and security updates, Loadbalancer.org have an online and an offline update facility that allows customers who have a valid maintenance and support contract to keep their appliance fully up to date.



Since services can be restarted during the update process we recommend performing the update during a maintenance window.

Checking the Current Software Version & Revision

The current software version and revision can be checked at the console or via an SSH session using the following command:

```
cat /etc/loadbalancer.org/version.txt
```

Online Update

This option is used to update the appliance via the Internet.

MAINTENANCE > SOFTWARE UPDATE

Online Update

Online updates are only available if your organisation has a valid authorisation key.

An authorisation key may be obtained from [Loadbalancer.org support](#).

Before starting the online update we recommend that you backup the XML configuration file, firewall script, and any manual changes that have been made.

[\[Download XML Configuration File \]](#)

[\[Download Firewall Script \]](#)

Update from v7.4 to v7.4.1

Warning: Updates should only be installed during a maintenance window.

Authorisation
Key

Online Update

To use this, simply enter your Auth-Key and click **Online Update**

Notes:

- As indicated in the WUI, we recommend that you backup your XML configuration and firewall script using the links provided before running the update
- Make sure that the load balancer is able to access the Internet – if you have a proxy server, this can be defined using *Edit Configuration > Physical Advanced Configuration*
- Make sure that the default gateway is set correctly (*Edit Configuration > Routing*)
- Make sure that the DNS server are set correctly (*Edit Configuration > Hostname & DNS*)

If after configuring these options you continue to experience update failure messages such as:

MAINTENANCE > SOFTWARE UPDATE

Online Update

Error: Failed to fetch online update information.

Then you'll need to restart Apache on the load balancer. To do this, run the following command at the console or via an SSH session:

```
service httpd restart
```



The auth code is included in your technical support document supplied to you when the appliance was initially purchased or when your support contract was renewed. If you do not have a valid contract, please contact : sales@loadbalancer.org.

Offline Update

If the load balancer does not have access to the Internet, Offline Update can be used.

Please contact support@loadbalancer.org to obtain the download required for your appliance.

Updating a Clustered Pair

To update a clustered pair (i.e. a master unit and a slave unit) , follow the steps below:

1. First perform the update to the slave unit using the online or offline update method described previously. Take care to follow any on-screen instructions that are displayed (e.g. service restarts)
2. Now update the master unit in the same way



For a clustered pair, we strongly recommend fully testing & validating the master / slave failover process before going live. If testing was not carried out before go-live, we recommend scheduling a maintenance window to do this. For detailed steps, please refer to page 158.

Appliance Security

Firewall

Whilst the appliance includes a highly capable Linux based on board firewall, Loadbalancer.org believes that configuring complex access rules directly on the load balancer often over complicates the configuration. The key focus of the appliance is to load balance inbound connections and therefore Loadbalancer.org recommends that the appliance is installed behind a dedicated firewall.

Passwords

At least the following passwords should be changed from their default values:

- the password for the 'loadbalancer' Web User Interface account. To change this use the WUI option:
Maintenance > Passwords
- the password for the Linux 'root' account. To change this at the console or via an SSH session run:

```
passwd
```

N.B. These passwords can also be changed by running the console command 'lbsecure' as explained in the following section.

Appliance Lockdown Script

The security lockdown script can be used to quickly lockdown the appliance. The script helps to lock down the following :

- the password for the 'loadbalancer' Web User Interface account
- the password for the Linux 'root' account
- which subnet / host is permitted access to the load balancer

To run the script, at the console or via an SSH session run:

```
lbsecure
```

PCI Compliance

For a discussion on this topic, please refer to [Loadbalancer.org's PCI blog](#)

Adding a Slave Unit After the Master Has Been Configured

To add a second slave unit to an existing master unit to create a highly available clustered pair, follow the steps below:

1) On the Slave Unit:

- Set the slave appliance's IP address using either the network setup wizard or by running standard Linux commands as follows:

```
ip addr add dev eth0 <IP address>/<mask>
```

e.g.

```
ip addr add dev eth0 192.168.2.100/24
```
- Connect to the WUI using a browser : `http://<IP address>:9080`
- If the IP address was not configured using the network setup wizard, then this setting will be temporary and will not survive a reboot. To set this permanently use the WUI option : *Edit Configuration > Network Interface Configuration* and enter an appropriate IP address in the relevant interface field(s)
- Once configured, click **Configure Interfaces**
- Using the WUI option : *Edit Configuration > Hostname & DNS* change the hostname to 'lbslave' also ensure that a valid DNS server is configured
- Click **Update**
- Using the WUI option : *Edit Configuration > Routing* set the default gateway
- Click **Configure Routing**

2) Master / Slave Interface:

- Connect the serial cable between the master & slave if you intend to run heartbeat over the serial link (this is the default heartbeat comms method), unicast is the other option that can be used which runs over the network and uses port 6694 by default

3) On the Master Unit:

- Using the WUI option : *Edit Configuration > Hostname & DNS* enter the IP address of the slave unit in the 'Slave Load Balancer' field
- Check the box 'Force Full Slave Sync'
- Click **Update**



IMPORTANT! The remaining steps must be done during a maintenance window since all services will be restarted causing potential end-user disruption.

- Using the WUI option : *Maintenance > Restart Services* , click **Restart Heartbeat**

4) On the Slave Unit:

- Using the WUI option : *Maintenance > Restart Services* , click **Restart Heartbeat**

Allow time for heartbeat to restart and synchronize (approx 1min) , then continue to the next section to verify failover to the slave and failback to the master.

Verifying Master / Slave Replication & Testing Failover

- 1) On the master click *System Overview* and verify that the system status appears as follows:



- 2) Also check the system status in the same way on the slave unit:



- 3) Verify that settings been replicated to the slave unit, this can be done by using either the *View Configuration* or *Edit Configuration* menus to validate that the same Virtual & Real Servers exist on the slave as on the master
- 4) During a maintenance window, verify failover to the slave and failback to the master by following the steps below:

1. On the slave using the WUI option : *Edit Configuration > Execute a Shell Command* run the command:

/usr/local/sbin/hb_takeover.php all

now verify that the slave's status has changed to Active as follows:



and the master has changed to Passive:



Also, using the WUI option : *View Configuration > Network Configuration* verify that the floating IPs associated with the VIPs have been brought up on the slave unit and brought down on the master

2. On the master using the WUI option : *Edit Configuration > Execute a Shell Command* run the command:

/usr/local/sbin/hb_takeover.php all

now verify that the master's status has changed to Active as follows:



3. and the slave has changed to Passive:

Master	Slave	Active	Passive	Link
--------	-------	--------	---------	------

Also, using the WUI option : *View Configuration > Network Configuration* verify that the floating IPs associated with the VIPs have been brought up on the master unit and brought down on the slave

- 5) During a maintenance window, verify failover to the slave and failback to the master by following the steps below:

1. Power down the master using the WUI option : *Maintenance > System Control > Halt Server*

now verify that the slave's status has changed to Active as follows:

Master	Slave	Active	Passive	Link
--------	-------	--------	---------	------

Also, on the slave using the WUI option : *View Configuration > Network Configuration* verify that the floating IPs associated with the VIPs have been brought up

2. Power up the master

now verify that the master's status has changed to Active as follows:

Master	Slave	Active	Passive	Link
--------	-------	--------	---------	------

and the slave has changed to Passive:

Master	Slave	Active	Passive	Link
--------	-------	--------	---------	------

Also, using the WUI option : *View Configuration > Network Configuration* verify that the floating IPs associated with the VIPs have been brought up on the master unit and brought down on the slave

*N.B. This assumes that the option 'Automatic Fail-back' is enabled. This is enabled by default and can be set using the WUI option : *Edit Configuration > Heartbeat Configuration*. If not enabled, the services will not auto failback to the master when it becomes available, but will remain active on the slave. In this case, services must be moved back to the master manually using the following command on the master:*

`/usr/local/sbin/hb_takeover.php all`

IPMI Configuration

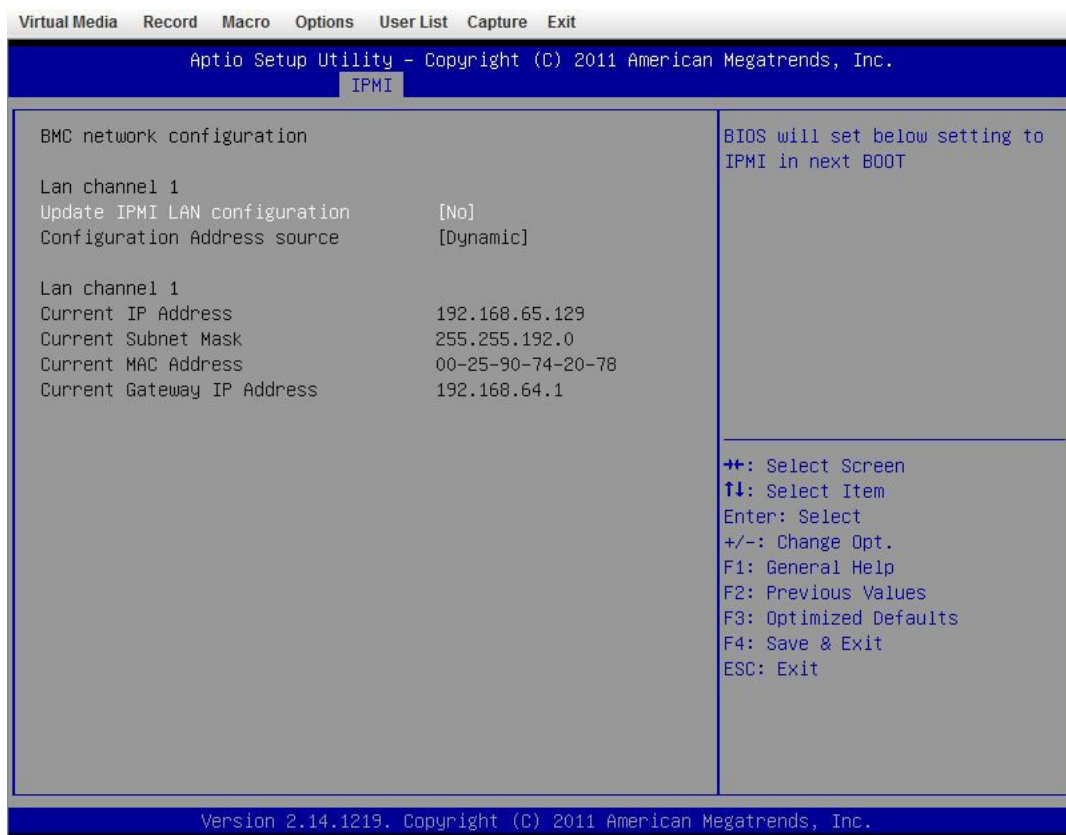
The Supermicro units include an IPMI module to allow remote control & management. This can either be accessed via the dedicated IPMI interface (not included on the Enterprise & Enterprise R16) or via one of the standard Ethernet interfaces in bridged mode.

To use the dedicated IPMI interface, ensure that a network cable is plugged into the interface before powering up the appliance.

Configuring the IP Address

By default the IP address is set using DHCP. The address allocated is displayed in the IPMI sub-menu in system setup. If preferred, a static IP address can also be set using the same menu. To access system setup, hit as directed at boot time.

IPMI BIOS Menu – Enterprise MAX :



To set the address

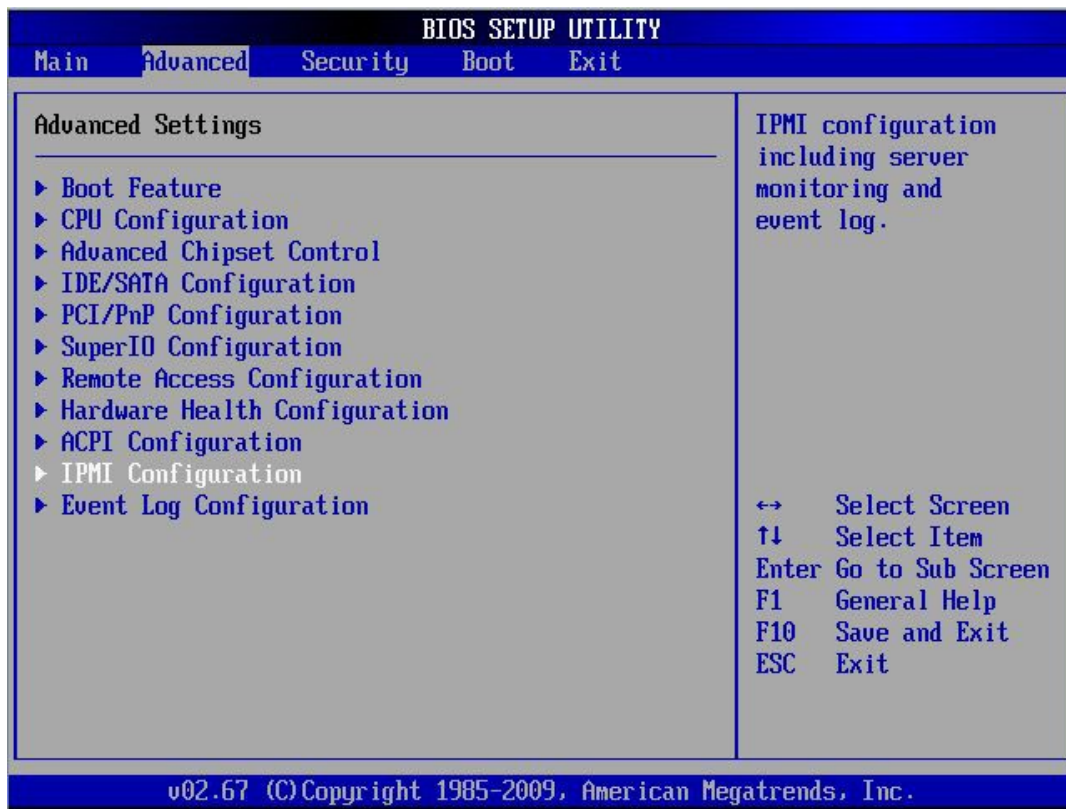
change **Update IPMI LAN configuration** to 'Yes'

change **Configuration Address Source** to 'Static'

now set the IP address, mask etc. as required.

```
Lan channel 1
Update IPMI LAN configuration      [Yes]
Configuration Address source     [Static]
Station IP address                0.0.0.0
Subnet mask                      0.0.0.0
Station MAC address              00-00-00-00-00-00
Gateway IP address               0.0.0.0
```

IPMI BIOS Menu – Enterprise & Enterprise R16 :



To set the address

select **Set LAN Configuration**

change **IP Address Source** to 'Static'

now set the IP address, mask etc. as required.

```
Channel Number          [01]
Channel Number Status:Channel number is OK
IP Address Source       [Static]
IP Address              [192.168.075.111]
Subnet Mask             [255.255.192.000]
Gateway Address         [192.168.064.001]
MAC Address             [00.25.90.6F.39.DA]
```

Accessing the login page

Using a browser, connect to `http://<ip address>` , the following login prompt is displayed:

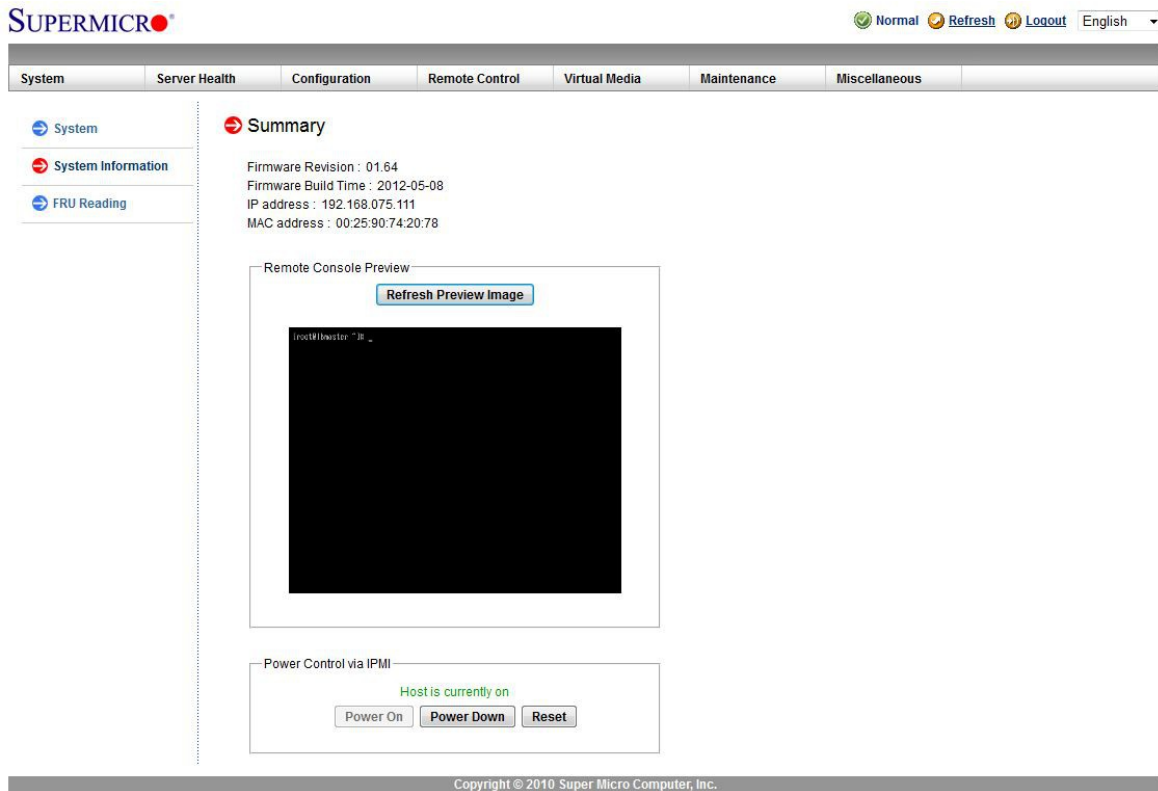



The login form is titled "Please Login" and contains two input fields: "Username" and "Password". Below the fields is a "login" button.

username : ADMIN

default password : ADMIN

Once logged in, the following screen is displayed:



IPMI Interface

As mentioned above IPMI can be accessed via the dedicated interface or via one of the standard on-board NICs. This can be configured in the IPMI interface using : *Configuration > Network > LAN Interface*

Dedicate – use the dedicated interface only

Share – run in bridge mode using one of the standard NICs

Failover – allows either connection method to be used (the default)

N.B. This option is not available on the Enterprise & Enterprise R16 since there is no dedicated IPMI port.

Remote Control

To access the systems console, simply click on the Remote Console Preview. A new window will open with access to the console of the appliance.

Section F – Disaster Recovery

Being Prepared

To be able to quickly recover your appliance when a disaster occurs it's important that you create a backup of the XML configuration file and keep it stored in a safe location off the load balancer. Ideally you should keep a backup of both the master and slave configurations. This can easily be done by following the steps below:

Backing Up to a Remote Location

Login to the web interface:

Username: loadbalancer

Password: loadbalancer

Backup the XML configuration file:

- Select *Maintenance > Backup & Restore > Download XML configuration file*
- Save the file in a secure location

Backup the firewall configuration:

- Select *Maintenance > Backup & Restore > Download Firewall Script*
- Save the file in a secure location

If you're terminating SSL on the load balancer, backup your certificates as well:

- Select *Maintenance > Backup & Restore > Download SSL Certificates*
- Save the file in a secure location

Using wget to Copy the Files

It's also possible to use wget from a Linux session on a remote machine to pull the XML configuration file and firewall script:

```
wget --user=loadbalancer --password=loadbalancer http://<IP>:9080/lbadmin/config/getxmlconfig.php  
-O lb_config.xml
```

```
wget --user=loadbalancer --password=loadbalancer http://<IP>:9080/lbadmin/config/getfirewall.php  
-O rc.firewall
```

N.B. Replace the password 'loadbalancer' with your password

Backing Up to the Load Balancer

To create a backup that is stored on the load balancer itself, follow these steps:

Log in to the web interface:

Username: loadbalancer

Password: loadbalancer

- Select *Maintenance > Backup & Restore > Make local XML backup*
- Select *Maintenance > Backup & Restore > Make local Firewall Script backup*
- A copy of both files will be stored in `/etc/loadbalancer.org/userbkup`

Appliance Recovery using a USB Memory Stick

The following instructions detail how to recover a Loadbalancer.org appliance to the latest version using a USB stick (1GB or more in capacity).



This will only work on 64Bit hardware. From version 6.0 onwards, all appliances are 64Bit. If you are running an older version, this may or may not be possible depending on the hardware.

If you are running v5.x and wish to determine whether your appliance is 64Bit, then enter the following command:

```
grep flags /proc/cpuinfo
```

If **lm** (long mode) is present in the output then the CPU is 64Bit and you can proceed. If not then your appliance is 32Bit and you are limited to the latest v5 software.

The latest images require a standard disk (Dell hardware) or a high speed IDE DOM / SATA SSD (Supermicro hardware). If you are already running v6 then you will already have this and should be able to simply re-image your current drive / disk module. If you are upgrading from v5 you will need to purchase a suitable device and then use the following procedure to build it from the USB stick.



It's not currently possible to import an XML file from a v6.x appliance to a v7.x appliance.

The latest version of the ISO file is available at <http://www.loadbalancer.org/download/>

You can use [UNetBootin](#) (Windows or Linux) to transfer the ISO onto the USB stick.

Make sure you change the server BIOS to boot from the USB first (the stick must be plugged in at that stage to allow it to be selected as a boot device).

When it boots choose:

Default image

Once the **root@lbmaster:/ #** prompt appears, enter the following commands:

```
# cd /etc/recovery
# ./clone-dsk.sh
```

at the first prompt, press <ENTER>

at the second prompt, select option 1
at the third prompt, select option 1
at "Is the disk /dev/hda already formatted (manually) [Y/N]?" type 'N'
at "do you want to reformat /dev/hda [Y/N]?" type 'Y'
then Yes to all other prompts

The image is then transferred onto the disk / module. This takes around 5 mins to complete depending on the hardware specification of the system.

Once complete, remove the USB stick and reboot the appliance

****** You now have a fully functioning appliance ******

Now continue with the relevant slave / master recovery steps to configure the device with your particular configuration

Disaster Recovery After Master Failure

For a correctly configured clustered pair, if the master fails, the slave will take over automatically. To restore the master load balancer's configuration, a backup copy of the lb_config.xml file is used. This backup should be created using the steps on page 165



NOTE: If a backup copy from the master is not available, It's possible to use the lb_config.xml from the slave instead. If there is no current backup of this, then use the WUI option: *Maintenance > Backup & Restore > Download XML Configuration file* to create the file. A couple of changes need to be made so the file represents the master unit rather than the slave as shown below.

Steps (with example IP addresses) to modify a copy of the configuration file from the slave, for use on the master:

find & Change:

```
<network>
  <hostname>lslave</hostname>
  <slave></slave>
```

To:

```
<network>
  <hostname>lbmaster</hostname>
  <slave>192.168.2.165</slave>
```

(i.e. change the hostname to 'lbmaster')
(i.e. specify the IP address of your slave unit)

Find & Change:

```
<rip>
  <eth0>192.168.2.165/24</eth0>
  <eth1>192.168.4.165/24</eth1>
```

To:

```
<rip>
  <eth0>192.168.2.164/24</eth0>
  <eth1>192.168.4.164/24</eth1>
```

(i.e. change to the IP address of your master unit)
(i.e. change to the IP address of your master unit)

N.B. for the MAX & 10G you may also need to change eth2 & eth3 in the same way

To Perform the Recovery

- Locate your copy of *lb_config.xml* (either the backup from the master, or the modified slave copy)
- If the failed master is still on, power it down
- Disconnect all cables
- Repair the problems you are having with the master
- Connect the power lead, mouse, monitor and keyboard
- Restore the master from the Load balancer ISO image using a USB stick by following the steps on page 166.
- Log onto the console of the master appliance as:
Username: root
Password: loadbalancer
- At the console configure the IP address (replace with your IP/mask) :

```
# ip addr add dev eth0 192.168.2.164/24
```



IMPORTANT – The following steps must be done during a maintenance window since all services will be restarted.

- On the console of the repaired master unit run the following command to stop heartbeat:

```
# service heartbeat stop
```
- On the console of the slave unit run the following command to stop heartbeat:

```
# service heartbeat stop
```
- Once heartbeat has successfully stopped on both units, re-connect the network cable to the repaired master and the serial cable (if used for heartbeat) between the two units
- Open the WUI of the repaired master (replace with your IP address) using:

```
http://192.168.2.164:9080
```
- Login to the WUI:
Username: loadbalancer
Password: loadbalancer
- Restore your XML file using *Maintenance > Backup & Restore > Upload XML file & Restore*
- Check that all settings have been restored as expected
- To Synchronize heartbeat between master & slave restart Heartbeat on ***both units*** using *Maintenance > Restart Services > Restart Heartbeat*
- If auto-failback is set to 'on' then the repaired master should now take over all services. If it's set to 'off', then you can manually failback to the repaired master using the following command on the master:

```
# /usr/local/sbin/hb_takeover.php all
```

To Verify the Status

After a minute or so your cluster should be restored with the repaired master unit as the active appliance and the slave as the passive appliance.

To verify this, the master units current status is displayed at the top of the WUI as shown below:



This shows that the unit is the **Master**, its currently **Active** and that the **Link** to the slave has been successfully established.

Disaster Recovery After Slave Failure

If the slave unit has failed, the master will continue to provide load balancing functions as normal. However it is important to recover the slave unit as soon as possible to restore the clustered pair to normal. To restore the slave there are two options:

OPTION 1 – Repair the unit, then restore the slave's XML backup file

or

OPTION 2 – Repair the unit, then use the 'Force full Sync' option on the master to re-synchronize the slave

Option 1 – Using the XML Backup

- Locate your up-to-date copy of the lb_config.xml slave backup file
- If the failed slave is still on, power it down
- Disconnect all cables
- Repair the problems you are having with the slave
- Connect the power lead, mouse, monitor and keyboard
- Restore the slave from the Load balancer ISO image using a USB stick by following the steps on page 166.
- Log onto the console of the slave appliance as:

Username: root
Password: loadbalancer

- At the console configure the IP address (replace with your IP/mask) :

```
# ip addr add dev eth0 192.168.2.164/24
```



IMPORTANT – The following steps must be done during a maintenance window since all services will be restarted.

- On the console of the repaired slave unit run the following command to stop heartbeat:

```
# service heartbeat stop
```
- On the console of the master unit run the following command to stop heartbeat:

```
# service heartbeat stop
```
- Once heartbeat has successfully stopped on both units, re-connect the network cable to the repaired slave and the serial cable (if used for heartbeat) between the two units
- Open the WUI of the repaired slave (replace with your IP address) using:

```
http://192.168.2.164:9080
```
- Login to the WUI:
Username: loadbalancer
Password: loadbalancer
- Restore your XML file using *Maintenance > Backup & Restore > Upload XML file & Restore*
- Verify the configuration to ensure all settings have been restored as expected
- To Synchronize heartbeat between master & slave restart Heartbeat on **both units** using *Maintenance > Restart Services > Restart Heartbeat*

To Verify the Status

After a minute or so your cluster should be restored with the master unit as the active appliance and the repaired slave as the passive appliance.

To verify this, the slave units current status is displayed at the top of the WUI as shown below:



This shows that the unit is the **Slave**, its currently **Passive** and that the **Link** to the master has been successfully established.

Option 2 – Synchronizing From the Master

- If the failed slave is still on, power it down
- Disconnect all cables
- Repair the problems you are having with the slave
- Connect the power lead, mouse, monitor and keyboard and power on the unit

Now follow the steps in the section '*Adding a Slave Unit after the master has been configured*' on page 157.

Section G – Web User Interface Reference

System Overview

Displays various system resource statistics and an overview of the Virtual & Real Servers. For each Real Server links are available to control their state, the options available are:

- **Drain** – This option allows existing connections to close gracefully and prevents new connections
- **Halt** – This options prevents new connections and drops all existing connections immediately without waiting

N.B. If you request to drain or halt all the Real Servers at layer 4, the fallback server will NOT be activated – the fallback server only comes into effect when all servers fail their health-check. At layer 7, the fallback page is displayed when either all servers fail or when all servers are taken offline.

View Configuration

XML

View the lb_config.xml configuration file. This details the main configuration for the appliance.

Layer 4

View the layer 4 configuration file.

Layer 7

View the haproxy.cfg configuration file.

SSL Termination

View the pound.cfg configuration file.

Heartbeat Configuration

View the ha.cfg configuration file.

Heartbeat Resources

Displays the contents of the /etc/ha.d/conf/haresources file.

Network Configuration

View the running configuration of the network of the load balancer.

Routing Table

View the routing table of the appliance.

Firewall Rules

View all firewall rules configured on the appliance.

Edit Configuration

Set up or modify the physical and virtual configuration of the load balancer appliance.

Layer 4 – Virtual Servers

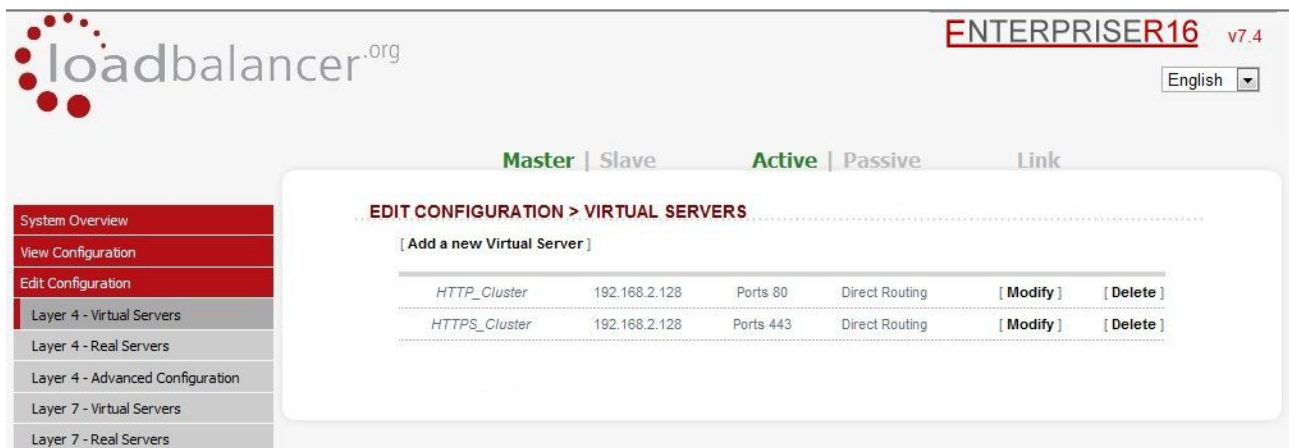
This menu option allows you to add, remove or modify Virtual Servers. Each Virtual Server can have an unlimited number of Real Servers (except the Enterprise R16 which is limited to 4 x VIPs each with up to 4 RIPs).

Typically you'll need one Virtual Server for each distinct cluster *AND* protocol that you wish to load balance. For example, if you want to serve both HTTP and HTTPS then you will need two Virtual Servers:

192.168.2.128:80

and

192.168.2.128:443



The screenshot shows the web interface of the loadbalancer.org Enterprise R16 v7.4. The top navigation bar includes the logo, version information, and a language dropdown set to English. Below the navigation bar, there are tabs for Master, Slave, Active, Passive, and Link. The main content area is titled 'EDIT CONFIGURATION > VIRTUAL SERVERS' and includes a link to 'Add a new Virtual Server'. A table lists the existing virtual servers:

Label	IP Address	Ports	Routing	Actions
HTTP_Cluster	192.168.2.128	Ports 80	Direct Routing	[Modify] [Delete]
HTTPS_Cluster	192.168.2.128	Ports 443	Direct Routing	[Modify] [Delete]

The left sidebar contains a menu with options: System Overview, View Configuration, Edit Configuration, Layer 4 - Virtual Servers (selected), Layer 4 - Real Servers, Layer 4 - Advanced Configuration, Layer 7 - Virtual Servers, and Layer 7 - Real Servers.

N.B. Firewall marks can be used to configure Virtual Servers with more than 1 port. For example, this can be useful if you have an ecommerce site where you want users to connect to the same backend server for both HTTP and HTTPS. This can be achieved using firewall marks. For more details please refer to page 132-138.

Adding a Virtual Server is a simple case of specifying the Label (name), IP address & port. Other settings can be left at default values which are appropriate in many cases. If you require the client connections to be persistent (i.e. stick to the first Real Server they hit), then change persistence to 'yes'. This is recommended for HTTPS to stop clients repeatedly re-negotiating SSL keys as they move between different Real Servers.

Layer 4 persistence is based on source IP address & destination port. The time out value is in seconds and each time the client makes a connection the timer is reset, so even a 5 minute persistence setting could last for hours if the client is active and regularly refreshes their connection.

The load balancer will automatically add the Virtual Server to the pool of Floating IP(s). The Floating IP should activate instantly. Just check *View Configuration > Network Configuration* to ensure that the Floating IP address has been activated correctly. They will show up as aliases, i.e. eth0:0, eth0:1 etc.

Adding a Virtual Server

EDIT CONFIGURATION > ADD A NEW VIRTUAL SERVER

Label	<input type="text" value="VIP Name"/>	?
Virtual Server IP address	<input type="text" value="10.0.0.20"/>	?
Virtual Server Ports	<input type="text" value="80"/>	?
Forwarding Method	<input type="text" value="Direct Routing"/>	?
Persistent	<input type="text" value="no"/>	?
Protocol	<input type="text" value="TCP"/>	?

Label – Set the name for the Virtual Server

Virtual Server IP address – Set the IP address of the Virtual Server

Virtual Server Ports – Configure the ports for the Virtual Server. For multiple ports separate them by commas, for a range use a dash, for all ports use an asterix. e.g. 80,81,90-95.



The following ports are used by the appliance and therefore cannot be used for Virtual Servers: 22 (SSH), 9080 (WUI – http), 9443 (WUI – https), 7777 (HAProxy statistics page), 9081 (nginx fallback page).

Forwarding Method – The routing method that the load balancer uses to forward packets to the Real Servers:

Direct Routing (DR) - This is the default in One-Arm mode (Direct Routing). Direct Routing is recommended because it's easy to understand and implement with two load balancers in failover mode (our recommended configuration). It only requires one external Floating IP address on the same subnet as your web server cluster and only one network card.

A separate firewall is required, as is a NAT gateway if the network uses private IP addresses.

Direct Routing changes the destination MAC address of the Ethernet frame to redirect it to a server in the cluster without modifying the IP packet. Each Real Server must therefore be configured to respond to the Virtual IP, but must not respond to ARP requests for that IP. This is known as 'solving the ARP problem'.

The other advantage of Direct Routing is that each web server can reply through its own default gateway at gigabit speeds without needing the packets to return through the loadbalancer.

NAT – This is the default in 2 Arm mode (Network Address Translation). This has the advantage that you can load balance any device without having to deal with the ARP problem. The Real Servers need their default gateway changed to be the internal floating VIP of the load balancer. Because the load balancer handles the return packet you will get more detailed statistics but slower speed than DR or TUN. NAT can also be implemented with a single NIC – just use the firewall script to set up an alias on the eth0 interface.

Tunneling – This is for WAN links (Tunneling). Tunneling has somewhat limited use as it requires an ip tunnel between the load balancer and the Real Server as the VIP is the target address many routers will drop the packet assuming that it has been spoofed. However it is useful for private networks with Real Servers on multiple subnets.

Persistent – Enable persistence for this Virtual Server, by Source IP or SIP call-ID. Sticky or persistent connections are required for some protocols such as FTP and SIP. It is also kind to clients when using SSL, and unfortunately sometimes required with HTTP if your web application cannot keep state between Real Servers.

N.B. If your Real Servers cannot keep session state persistence themselves, then you will obtain performance benefits from a load balancer, but may not obtain reliability benefits.

Protocol – Select the protocol to load balance (usually TCP):

TCP – Transmission Control Protocol (STD0007, RFC0793). The default and most common option.

UDP – User Datagram Protocol (STD0006, RF0768). Used for DNS, SIP, etc.

One Packet Scheduling - for UDP SIP connections.

Firewall Marks – For use when traffic has been tagged in the firewall script using the MARK target.

Modifying a Virtual Server

When first adding a Virtual Server, only certain values can be configured, others are set at their default setting. These values can be changed after the Virtual Server has been created by clicking **[Modify]** next to the relevant Virtual Server. Settings that can be changed are:

Persistence Timeout – How long do you want connections to be sticky? The persistence time is in seconds and is reset on every connection; i.e. 5 minutes persistence will last for ever if the client clicks on a link within that period.

Balance Mode – Configure how connections are distributed to the Real Servers:

Weighted Least-Connection – assign more jobs to servers with fewer jobs, relative to the Real Servers' weight.

Weighted Round Robin – assign jobs to Real Servers proportionally to the Real Servers' weight. Servers with higher weights receive new jobs first and get more jobs than servers with lower weights. Servers with equal weights get an equal distribution of new jobs. This is the default.

Destination Hashing – assign jobs to servers through looking up a statically assigned hash table by their destination IP addresses.

Fallback Server – The server to route to if all of the Real Servers in the group fail the health check. The local nginx fallback server is configured for the ports 80 and 9081 (configured to always show the indexation page). You can also configure the fallback server to be a 'Hot Spare' if required. For example you have one server in the cluster and one fallback they will act as a master / slave pair.

Fallback Server Port – Set the port for the fallback server. In DR mode, since port redirection is not possible, the port is automatically set to be the same as the Virtual Server.

Check Type – Specify the type of health-check for the Real Servers.

Negotiate Check Service – Specify the protocol to use when check-type is set to negotiate.

Check Port – If you want the Service to check to be say HTTPS but not on the default port (443) then you can specify that here.

Check Command – The custom check script, used with the external check type. The script should be placed in `/var/lib/loadbalancer.org`, and given world read and execute permissions.

Virtual Host - If the Real Server will only respond to a URL or 'virtual' rather than an ip address for its health checks, you can specify the virtualhost to request here.

Login – The login name to use with negotiate checks where authentication is required.

Password – The password to use with negotiate checks where authentication is required.

Secret – Configure the RADIUS secret string for the RADIUS negotiate check.

Granularity – Specify a whole subnet to use instead of source ip for persistence. Some large Sips use clustered proxies this means that the clients source ip address may keep changing. If you require persistence of HTTP and this is causing a problem then you can set a larger masq on the source ip address match for persistence i.e. `255.255.255.0` for a whole class C subnet. *N.B. Single IP `255.255.255.255` is the default.*

Request to Send – Used when Check Type is set to Negotiate. This specifies the request to send to the server. The use of this parameter varies with the protocol selected in Service to Check.

Response Expected – This string will be matched against the response to a negotiate check. If the string matches anywhere in the response data, the negotiate check is considered a success.

Email Alerts – Specify an email address for server health alerts. Email alerts can be specific to one Virtual Server or they can be a global setting. The global setting can be configured using *Edit Configuration > Layer 4 – Advanced Configuration*.

Feedback Method – The method the load balancer uses to measure the performance of the Real Servers:

Agent – A simple telnet to port 3333 on the Real Server

HTTP – A simple HTTP GET to port 3333 on the Real Server

None – No feedback (default setting)

The loadbalancer expects a 0-99 integer response from the agent, usually relating to the CPU idle; i.e. a response of 92 would imply that the Real Servers CPU is 92% idle. The load balancer will then use the formula $((92 / 10) * \text{requested_weight})$ to find the new weight. Using this method an idle Real Server will get 10 times as many new connections as an overloaded server.



For more details on configuring health checks, please refer to page 121.

Layer 4 – Real Servers

This option allows you to add, remove or modify Real Servers. You can add an unlimited number of Real Servers to each Virtual Server (except the Enterprise R16 which is limited to 4 x VIPs each with up to 4 RIPs).

In DR mode, since port redirection is not possible the Real Server port field is not available and the port is automatically set to be the same as the Virtual Server, whilst for a NAT mode Real Server, it's possible to configure the port to be the same or different to the Virtual Server's port.

The screenshot shows the Loadbalancer.org web interface. The top navigation bar includes the logo, version 'ENTERPRISER16 v7.4', and a language dropdown set to 'English'. A secondary navigation bar has tabs for 'Master', 'Slave', 'Active', 'Passive', and 'Link'. The left sidebar contains a menu with options like 'System Overview', 'View Configuration', 'Edit Configuration', and various 'Layer 4' and 'Layer 7' settings. The main content area is titled 'EDIT CONFIGURATION > REAL SERVERS' and displays a table of real servers. The table has columns for the virtual server name, real server IP, port, and forwarding method. Two real servers are listed: 'real_server1' and 'real_server2', both with IP 192.168.2.150 and 192.168.2.160 respectively, and a weight of 1. Action links for 'Add a new Real Server', 'Modify', and 'Delete' are provided for each. A copyright notice at the bottom reads 'Copyright © Loadbalancer.org Limited 2002 – 2011'.

Virtual Server	Real Server IP	Port	Forwarding Method	Weight	Actions
HTTP_Cluster1	192.168.2.100	80	Direct Routing		[Add a new Real Server]
real_server1	192.168.2.150			1	[Modify] [Delete]
real_server2	192.168.2.160			1	[Modify] [Delete]
HTTP_Cluster2	192.168.2.110	80	Direct Routing		[Add a new Real Server]

Adding / modifying a new Real Server is a simple case of specifying IP address, port number and weight. Other settings can be left at default values which are appropriate in many cases.

The forwarding method defaults to that defined for the Virtual Server and you will normally leave this as DR. NAT can be used when you have two Floating Virtual IP(s) set up (one internal and one external) and TUN can be used to route through a tunnel across the Internet or WAN.

Adding / Modifying a Real Server

The screenshot shows the 'EDIT CONFIGURATION > ADD A NEW REAL SERVER' form. It contains several input fields with labels and help icons (question marks). The fields are: 'Label' (containing 'RIP Name'), 'Real Server IP Address' (containing 'IPAddress'), 'Real Server Port' (empty), 'Weight' (containing '1'), 'Minimum Connections' (containing '0'), and 'Maximum Connections' (containing '0'). An 'Update' button is located at the bottom of the form.

Label	RIP Name	?
Real Server IP Address	IPAddress	?
Real Server Port		?
Weight	1	?
Minimum Connections	0	?
Maximum Connections	0	?

Update

Label – Set the name for the Virtual Server.

Real Server IP address – Set the IP address of the Real Server.

Real Server Port – Configure the port for the Real Server (NAT mode only).

Weight – Weight is an integer specifying the capacity of a server relative to the others in the pool. The valid values of weight are 0 through to 65535. The default is 1.

Why would you change the weight of a Real Server? If you had a 4 core Xeon web server and a single core Celeron web server, you could increase the weight of the Xeon based server so that it handled more of the load.

Minimum Connections – An integer specifying the lower connection threshold of a server. The valid values are 0 through to 65535. The default is 0, which means the lower connection threshold is not set.

If Minimum Connections is set with other values, the server will receive new connections when the number of its connections drops below its lower connection threshold. If Minimum Connections is not set but Maximum Connections is set, the server will receive new connections when the number of its connections drops below three quarters of its upper connection threshold.

Maximum Connections – An integer specifying the upper connection threshold of a server. The valid values of Maximum Connections are 0 through to 65535. The default is 0, which means the upper connection threshold is not set.

If Maximum Connections is set with other values, no new connections will be sent to the server when the number of its connections exceeds its upper connection threshold.

Layer 4 – Advanced Configuration

This section allows you to configure the layer 4 global timeouts and logging options for the load balancer.

EDIT CONFIGURATION > ADVANCED CONFIGURATION

Layer 4		
Check Interval	<input type="text" value="6"/>	?
Check Timeout	<input type="text" value="3"/>	?
Negotiate Timeout	<input type="text" value="5"/>	?
Failure Count	<input type="text" value="1"/>	?
Quiescent	<input type="text" value="no"/>	?
Email Alerts	<input type="text"/>	?
Auto NAT	<input type="text" value="off"/>	?
Multi-threaded	<input type="text" value="yes"/>	?
Fallback	<input type="text" value="yes"/>	?
Disable Write	<input type="text" value="off"/>	?

Check Interval – Layer 4 (Ldirectord) health check interval in seconds. If this setting is too low, you may experience unexpected Real Server downtime.

Check Timeout – Layer 4 (Ldirectord) health check timeout in seconds. If this setting is too low, you may induce un-expected Real Server downtime.

Negotiate Timeout – Layer 4 (Ldirectord) negotiate health check timeout in seconds. The negotiate checks may take longer to process as they involve more server side processing than a simple TCP socket connect check. If this setting is too low, you may induce unexpected Real Server downtime.

Failure Count – Layer 4 (Ldirectord) number of times a check has to fail before taking server offline. The time to detect a failure and take down a server will be (check interval + check timeout) * failure count.

Quiescent – When a Real Server fails a health check, do we kill all connections?

When Quiescent is *yes*, on a health check failure the Real Server is not removed from the load balancing table, but the weight is set to 0. Persistent connections will continue to be routed to the failed server, but no new connections will be accepted.

When Quiescent is *no*, the server is completely removed from the load balancing table on a health check failure. Persistent connections will be broken and sent to a different Real Server.

N.B. Quiescent only applies to health checks – it has no effect on taking Real Servers offline in System Overview. To manually force a Real Server to be removed from the table, set Quiescent to no and arrange for the server to fail its health check. This may be done, for example, by shutting down the daemon or service, changing the negotiate check value, or shutting down the server.

Email Alerts – Specify the global email alert address. The global email alert address is used to send notifications of Real Server health check failures. This can also be configured on a Virtual Server level.

Auto NAT – Automatically NAT outbound network connections from internal servers. By default servers behind the load balancer in a NAT configuration will not have access to the outside network. However clients on the outside will be able to access load balanced services. By enabling Auto NAT the internal servers will have their requests automatically mapped to the load balancers external IP address. The default configuration is to map all requests originating from internal network eth0 to the external IP on eth1. If you are using a different interface for external traffic you can select it here. Manual SNAT and DNAT configurations for individual servers can also be configured in the firewall script.

Multi-threaded – Perform health checks with multiple threads. Using multiple-threads for health checks will increase performance when you have a large number of Virtual Servers.

Fallback – Local Fallback server on / off switch . Configure whether the local fallback server is active or not, sometimes you may want the local fallback server switched off so that it doesn't change the SNMP results table when activated. You may also want to disable it for security purposes.

Disable Write – Disable writing to the layer 4 configuration file. When enabled (on) configuration changes via the WUI are not permitted, an on-screen message as shown below is displayed:



This is useful if you want to make manual changes to the configuration file. When disabled (off), changes are permitted via the WUI.



NOTE: If manual changes are made to configuration files, then Disable Write is switched off and changes are made via the WUI, the manual changes will be overwritten.

Layer 7 – Virtual Servers

This menu option allows you to add, remove or modify Virtual Servers. Each Virtual Server can have an unlimited number of Real Servers (except the Enterprise R16 which is limited to 4 x VIPs each with up to 4 RIPs).

Layer 7 Virtual Servers support a variety of additional persistence modes including HTTP cookie and RDP cookie whilst still supporting IP address based persistence.

The Layer 7 Virtual Servers are configured separately from Layer 4 servers because they use HAProxy rather than the LVS (Linux Virtual Server) engine.



The screenshot shows the Loadbalancer.org web interface. The top navigation bar includes the logo, version 'ENTERPRISER16 v7.4', and a language dropdown set to 'English'. The main content area is titled 'EDIT CONFIGURATION > VIRTUAL SERVERS (HAProxy)'. It features a table with two columns: 'cluster' and 'ports'. The table lists two clusters: 'cluster1' with IP '192.168.2.150' and 'Ports 80', and 'cluster2' with IP '192.168.2.160' and 'Ports 80'. Each row has 'Modify' and 'Delete' buttons. A sidebar on the left contains a menu with options like 'System Overview', 'View Configuration', and 'Edit Configuration', with 'Layer 7 - Virtual Servers' highlighted. The footer indicates 'Copyright © Loadbalancer.org Limited 2002 – 2011'.

cluster	IP	Ports	Modify	Delete
cluster1	192.168.2.150	Ports 80	[Modify]	[Delete]
cluster2	192.168.2.160	Ports 80	[Modify]	[Delete]

Layer 7 Virtual Servers are created in the same way as Layer 4 Virtual Servers, but by using a different option in the menu.



As with layer 4 VIPs the following ports are used by the appliance and therefore cannot be used for Virtual Servers: 22 (SSH), 9080 (WUI – http), 9443 (WUI – https), 7777 (HAProxy statistics page), 9081 (nginx fallback page).

When HTTP Cookie persistence mode is used, the inserted cookie name is set to be the same as the Real Server Label (name).

With Layer 7, port re-direction is possible, i.e. VIP:80 → RIP:800 is possible

N.B. Any changes to the Layer 7 configuration requires a reload of the HAProxy service.

Adding a Virtual Server

EDIT CONFIGURATION > ADD A NEW VIRTUAL SERVER (HAProxy)

Label	<input type="text" value="VIP Name"/>	?
Virtual Server IP address	<input type="text" value="10.0.0.20"/>	?
Virtual Server Ports	<input type="text" value="80"/>	?
Persistence mode	<input type="text" value="None"/>	?
Fallback Server	<input type="text" value="127.0.0.1"/>	?
Fallback Server Port	<input type="text" value="9081"/>	?
<input type="button" value="Update"/>		

Label – Designate a recognizable label for this Virtual Server. The Label is used as the cookie so it must be different for each server.

Virtual Service IP Address – Specify the virtual service's IP address.

Virtual Service Ports – Specify the ports on which the virtual service should accept connections. Individual port numbers should be separated by commas, and ranges may be specified using a dash. To specify all ports, use a single asterisk. For example: 81, 443 – 447, 103, 104, 105

Persistence Mode – Set the persistence mode as required:

HTTP Cookie – Use an HTTP cookie to ensure a client always hits the same server.

MS Connection Broker – The load balancer is able to interact with Session Directory / Connection Broker by enabling Routing Token Redirection mode. This mode allows the reconnection of disconnected sessions by utilizing a routing token to enable the load balancer to re-connect the client to the correct terminal server.

RDP Cookie – This method utilizes the cookie sent from the client in the Connection Request PDU to make sure a Terminal Server user always uses the same server. This cookie is created when the username is entered at the first client login prompt (mstsc.exe). Note that if the username is not entered here, the cookie is not created.

Source IP – Make sure the same source IP always hits the same server.

Source Hash – This is now deprecated, please use an alternative.

None – No persistence, users will use the servers in Round Robin mode.

Fallback Server – Set the Fallback server as required. This is where requests go if all servers in the cluster are down. The default address is the load balancer (127.0.0.1).

Fallback Server Port – Set the fallback servers port. This can be different than the port used for the VIP / RIPv.

Modifying a Virtual Server

When first adding a Virtual Server, only certain values can be configured, others are set at their default setting. These values can be changed after the Virtual Server has been created by clicking **[Modify]** next to the relevant Virtual Server. Settings that can be changed are:

Layer 7 Protocol – Select the Layer 7 protocol to be handled by this Virtual Service, either HTTP or any other TCP-based protocol. If this Virtual Service will handle only HTTP traffic, selecting that option here allows more flexibility in the processing of connections. The HTTP Cookie and HTTP application cookie modes, and the X-Forwarded-For header all require HTTP to be selected here. In addition, the HAProxy logs will show more information on the client requests and Real Server responses.

Balance Mode – The scheduler used to specify server rotation. Specify the scheduler to utilize when deciding the backend server to use for the next new connection.

Timeout – The time-out period before an idle connection is removed from the connection table. The source ip will be removed from memory when it has been idle for longer than the persistence timeout. The default units are minutes. Only applies when IP address persistence is selected.

Table Size – The size of the persistence connection table in KB. The size of the connection table (approx 50 bytes per entry) where connection information is stored to allow a session to return to the same server within the timeout period. The default units are in KB. Only applies when IP address persistence is selected.

Fallback Server Persistence – During a health-check failure users can be forward to a fallback server. Setting this to on will make this server persistent so that when the Real Servers are put back in the pool, they will remain on the fallback server until their persistence times out. Setting this to off will move users to a Real Server as soon as one is available.

Check Port – Specify the port to use for health checking. If not specified here, the check port will be the same as the Virtual Server port. Useful when you are balancing multiple ports.

Request to Send – Specify a specific file for the health check. Open the specified file and check for the response expected, useful for checking a server sided script to check the health of the back-end application. For example, if **index.html** was specified in this field, the following check directive would be automatically created in the HAProxy configuration file:

```
option httpchk GET /index.html HTTP/1.0
```

(N.B. the back-slash character before 'index.html' is added automatically)

Response Expected – The content expected for a valid health check on the specified file. The response expected can be any valid regular expression statement. Continuing the example above, if the file index.html contained the word 'Copyright' response expected would be set to **Copyright**. The following check directive would then be automatically created in the HAProxy configuration file:

```
http-check expect rstring Copyright
```

Maximum Connections – Specifies the maximal number of concurrent connections that will be sent to this server. If the number of incoming concurrent requests goes higher than this value, they will be queued, waiting for a connection to be released

Application Cookie Name – Used to configure session stickiness on an existing application cookie. Set the name of the cookie here.

Application Cookie Length – Set the max number of characters that will be memorized and checked in each cookie value.

Application Cookie Hold Time – Set the time in milliseconds after which the cookie will be removed from memory if unused. If no unit is specified, this time is in milliseconds.

Set X-Forwarded-For Header – Instruct HAProxy to add an X-Forwarded-For header to all requests, showing the client's IP Address. If HTTP is selected under Layer 7 Protocol, HAProxy is able to process the header of incoming requests. With this option enabled, it will append a new X-Forwarded-For header containing the client's IP Address. This information may be extracted by the Real Server for use in web applications or logging.



For more details on configuring health checks, please refer to page 121.

Layer 7 – Real Servers

This menu option allows you to add, remove and modify Real Servers. You can add an unlimited number of Real Servers to each Virtual Server (except the Enterprise R16 which is limited to 4 x VIPs each with up to 4 RIPs).

The screenshot shows the Loadbalancer.org web interface. The top header includes the logo, the text "loadbalancer.org", and "ENTERPRISER16 v7.4". A language dropdown menu is set to "English". Below the header, there are tabs for "Master", "Slave", "Active", "Passive", and "Link". The main content area is titled "EDIT CONFIGURATION > REAL SERVERS (HAProxy)". On the left, there is a sidebar menu with options: "System Overview", "View Configuration", "Edit Configuration", "Layer 4 - Virtual Servers", "Layer 4 - Real Servers", "Layer 4 - Advanced Configuration", "Layer 7 - Virtual Servers", "Layer 7 - Real Servers" (which is highlighted), and "Layer 7 - Advanced Configuration". The main content area displays a table of real servers:

HTTP_Cluster	192.168.2.177	Ports 80	[Add a new Real Server]
Server1	192.168.2.99	Port 80	Weight 1 [Modify] [Delete]
Server2	192.168.2.111	Port 80	Weight 1 [Modify] [Delete]





At the bottom of the page, there is a copyright notice: "Copyright © Loadbalancer.org Limited 2002 – 2011".

Adding / modifying a new Real Server is a simple case of specifying IP address, port number and weight. Other settings can be left at default values which are appropriate in many cases.

The Real Servers in a Layer 7 configuration can be on any subnet in any network as long as they are accessible from the load balancer.

Adding / Modifying a Real Server

EDIT CONFIGURATION > ADD A NEW REAL SERVER (HAPROXY)

Label	<input type="text" value="RIP Name"/>	
Real Server IP Address	<input type="text"/>	
Real Server Port	<input type="text"/>	
Weight	<input type="text" value="1"/>	
<input type="button" value="Update"/>		

Label – Designate a recognizable label for this Real Server

Real Server IP Address – The IP address for the appropriate service on your Real Server.

Real Server Port – The port for the appropriate service on your Real Server.

Weight – Weight is an integer specifying the capacity of a server relative to the others in the pool. The valid values of weight are 0 through to 65535. The default is 1.

Layer 7 – Advanced Configuration

This section allows you to configure the global timeouts and logging options for the load balancer.

EDIT CONFIGURATION > ADVANCED CONFIGURATION (HAPROXY)

Layer 7 (HAProxy):		
Logging	<input type="text" value="off"/>	?
Log Only Errors	<input type="text" value="off"/>	?
Redispatch	<input type="text" value="on"/>	?
Connection Timeout	<input type="text" value="4000"/>	?
Client Timeout	<input type="text" value="42000"/>	?
Real Server Timeout	<input type="text" value="43000"/>	?
Maximum Connections	<input type="text" value="40000"/>	?
Ulimit	<input type="text"/>	?
Abort on Close	<input type="text" value="on"/>	?
Transparent Proxy	<input type="text" value="off"/>	?
Interval	<input type="text" value="2000"/>	?
Rise	<input type="text" value="2"/>	?
Fall	<input type="text" value="3"/>	?
Statistics Password	<input type="text"/>	?
Statistics Port	<input type="text"/>	?
Health check buffer length	<input type="text"/>	?
Persistence Table Replication	<input type="text" value="off"/>	?
Persistence Table Replication port	<input type="text"/>	?
Disable HAProxy Config Write	<input type="text" value="off"/>	?

Logging – Activate detailed logging of the Layer 7 HAProxy service. When activated the HAProxy log is written to /var/log/haproxy.

Log Only Errors – Do not log operational connection details, only log errors.

Redispatch – Allows HAProxy to break persistence and redistribute to working servers should failure occur. This setting should not require changing.

Connection Timeout – HAProxy connection timeout in milliseconds. This setting should normally not require changing.

Client Timeout – HAProxy client timeout in milliseconds. This setting should normally not require changing.

Real Server Timeout – HAProxy Real Server timeout in milliseconds. This setting should not require changing.

Maximum Connections – HAProxy maximum concurrent connections. This setting should not require changing, unless you are running a high volume site. See also Maximum Connections for a Virtual Server (HAProxy).

Ulimit – The maximum number of file descriptors used for layer 7 load balancing. This value is optional. If no value is given then a default value will be used internally. For simple configurations where each Virtual Server only listens to one address/port a reasonable value is the sum of:

- * 2 times the number of maximum connections (Global Settings Layer 7)
- * Number of Virtual Servers on layer 7 (HAProxy)
- * Number of Real Servers
- * plus 1 for logging purpose

In a more sophisticated environment you should use the number of address/port/proxy tuples instead of the number of Virtual Servers.

Abort on Close – Abort connections when users close their connection. Recommended as the probability for a closed input channel to represent a user hitting the 'STOP' button is close to 100%

Transparent Proxy – Enable Tproxy support for Layer 7 HAProxy. Tproxy support is required in order for the Real Servers behind a layer 7 HAProxy configuration to see the client source IP address. The load balancer must be in a NAT configuration (internal and external sub nets) with the Real Servers using the load balancers internal Floating IP address as their default gateway.

N.B. all Layer 4 methods are transparent by default



For more details on using Tproxy, refer to pages 119-120.

N.B. Since the load balancer must be in a NAT configuration (i.e. VIPs & RIPs in different subnets) to utilize TPROXY, it is not always an appropriate solution. In situations such as this, it's possible to use the X-Forwarded-For header with layer 7 Virtual Servers. Most web servers can then be configured to record the X-Forwarded-For IP address in the log files.

For details on how to enable X-Forwarded-For support, please refer to page 185.

For details on how to enable X-Forwarded-For support with Apache and IIS, please refer to the following Loadbalancer.org blog links:

Apache : <http://blog.loadbalancer.org/apache-and-x-forwarded-for-headers/>

IIS : <http://blog.loadbalancer.org/iis-and-x-forwarded-for-header/>

Interval – Interval between health checks. This is the time interval between Real Server health checks in milliseconds.

Rise – Number of health checks to Rise. The number of positive health checks required before re-activating a Real Server.

Fall – Number of health checks to Fall. The number of negative health checks required before de-activating a Real Server.

Statistics Password – Set the password used to access *Reports > Layer 7 Status*.

Statistics Port – Change the listening port for the HAProxy web based statistics report from the default 7777.

Health-check Buffer Length – Set the health check buffer length in bytes.

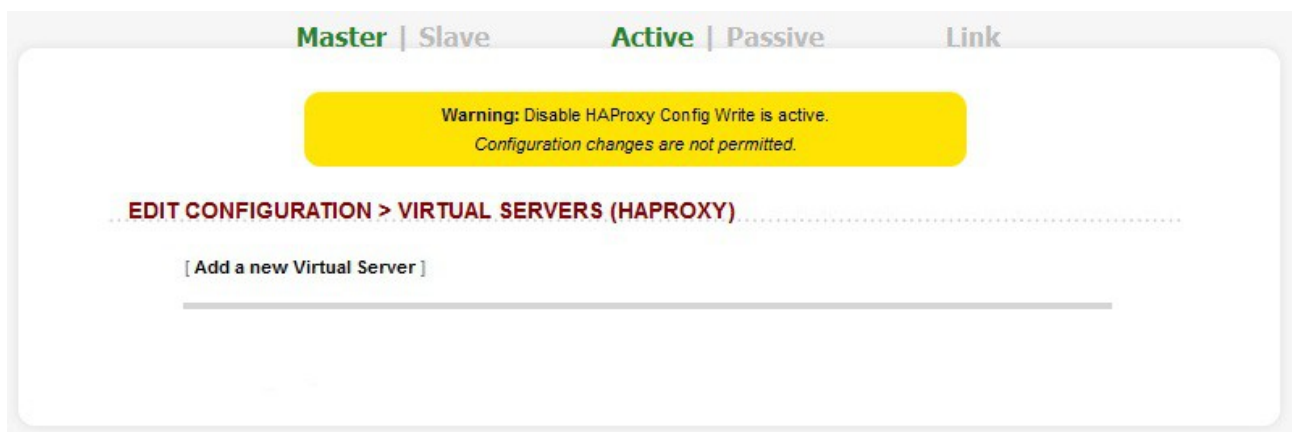
N.B. Changing this value will effect the performance of HAProxy. DO NOT make changes unless you know exactly what you are doing.

Lower values allow more sessions to coexist in the same amount of RAM, and higher values allow some applications with very large cookies to work. The default value is 16384 bytes. It is strongly recommended not to change this from the default value, as very low values will break some services such as statistics, and values larger than the default size will increase memory usage, possibly causing the system to run out of memory. At least the global maxconn parameter should be decreased by the same factor as this one is increased.

Persistence Table Replication – When enabled, HAProxy's persistence tables are replicated to the slave device.

Persistence Table Replication Port – Set the TCP port to use for persistence table replication.

Disable HAProxy Config Write – Disable writing to the layer 7 configuration file. When enabled (on) configuration changes via the WUI are not permitted, an on screen message as shown below is displayed:



This option is useful if you want to make manual changes to the configuration file. When disabled (off), changes are permitted via the WUI.



NOTE: If manual changes are made to configuration files, then Disable Write is switched off and changes are made via the WUI, the manual changes will be overwritten.

SSL Termination

If required, SSL can be offloaded to the load balancer. Pound is used to terminate SSL sessions and requires that the SSL certificate be deployed directly on the load balancer. HTTP traffic will then be passed unencrypted to the Real Servers.



NOTE: SSL termination on the load balancer can be very CPU intensive. In most cases, for a scalable solution, terminating SSL on the Real Servers is the best option.

In order to set up a proxy for the SSL traffic go to *Edit Configuration > SSL Termination*

EDIT CONFIGURATION > SSL TERMINATION ADD A NEW VIRTUAL SERVER

Label	<input type="text" value="VIP Name"/>	?
Virtual Server IP address	<input type="text" value="10.0.0.20"/>	?
Virtual Server Port	<input type="text" value="443"/>	?
Backend Virtual Server IP Address	<input type="text" value="10.0.0.20"/>	?
Backend Virtual Server Port	<input type="text" value="80"/>	?
Ciphers to use	<input type="text"/>	?
Enable WebDAV Verbs	<input type="checkbox"/>	?
Rewrite HTTP Redirects	<input checked="" type="checkbox"/>	?
Honor Cipher Order	<input type="checkbox"/>	?
Allow Client Renegotiation	<input type="text" value="No Client Renegotiation"/>	?

Layer 7 Backend VIP

SSL traffic is terminated by Pound on port 443 and then re-directed to port 80 of a layer 7 VIP for HAProxy to pick it up, insert cookies and load balance.

Layer 4 Backend VIP

For layer 4, Pound must be configured in a similar way, but instead of forwarding requests to HAProxy, requests are forwarded to a Layer 4 Virtual Server configured to operate in NAT mode.

DR mode **cannot be used** since Pound acts as a proxy, and the Real Servers see requests with a source IP address of the Virtual Server. However since the Real Servers believe that they own the Virtual IP (due to the loopback adapter configured to handle the ARP problem) they are unable to reply to Pound.

By default a self generated SSL certificate is associated with the new Virtual Server. it's also possible to upload your current certificate provided that it's in PEM format. Certificates can also be exported from Windows servers, converted to PEM format, then uploaded to the load balancer.

Adding / modifying an SSL Virtual Server

Label	<input type="text" value="VIP Name"/>	?
Virtual Server IP address	<input type="text" value="10.0.0.20"/>	?
Virtual Server Port	<input type="text" value="443"/>	?
Backend Virtual Server IP Address	<input type="text" value="10.0.0.20"/>	?
Backend Virtual Server Port	<input type="text" value="80"/>	?
Ciphers to use	<input type="text"/>	?
Enable WebDAV Verbs	<input type="checkbox"/>	?
Rewrite HTTP Redirects	<input checked="" type="checkbox"/>	?
Honor Cipher Order	<input type="checkbox"/>	?
Allow Client Renegotiation	<input type="text" value="No Client Renegotiation"/>	?

Label – Set the name for the Virtual Server

Virtual Server IP address – Set the IP address for the Virtual Server.

Virtual Server Port – Set the IP address for the Virtual Server. Normally will be port 443.

Backend Virtual Server IP address – Set the IP address for the Backend Virtual Server. This is normally the same IP address as the Virtual Server IP address but can be any valid IP. The IP selected must correspond to a Layer 4 NAT mode VIP or a Layer 7 HAProxy VIP which is where the unencrypted traffic will be sent for load balancing.

Backend Virtual Server Port – Set the port number for the Backend Virtual Server.

Ciphers to use – SSL Ciphers to use. List the SSL ciphers that Pound should accept. Leave blank for the default of any cipher. If you wish to restrict the ciphers that Pound should negotiate with the client, they may be specified here. If the field is left blank, Pound will use the default cipher list. The ciphers should be specified in OpenSSL cipher list format, and may include individual ciphers or groups. Some examples of valid cipher lists are shown below:

- * SSLv3
- * TLSv1
- * SSLv3:HIGH
- * AES128-SHA:DES-CBC3-SHA:RC4-SHA:RC4-MD5:!SSLv2

Enable WebDAV Verbs – When enabled extends which HTTP/WebDAV verbs are accepted

Rewrite HTTP Redirects – Enable Pound to change the Location: and Content-location: headers in responses. If they point to the back-end itself or to the listener (but with the wrong protocol) the response will be changed to show the virtual host in the request. N.B. If you're not sure what this means, then leave this set to the default value (enabled).

Honor Cipher Order – When choosing a cipher during a handshake, normally the client's preference is used. If this directive is enabled, the server's preference will be used instead.

Allow Client Renegotiation – Sets whether the client is allowed to renegotiate the cipher order. The options are:

- **No Client Renegotiation** – no client renegotiation will be honored
- **Secure Renegotiation** – secure renegotiation will be honored
- **Insecure Renegotiation** – insecure renegotiation will be honored

Cipher Settings and the BEAST Attack

The following options should be set to mitigate the BEAST attack :

Ciphers to use – a minimum cipher list of 'RC4:HIGH:!MD5:!aNULL' is required to mitigate the BEAST attack.

Honor Cipher Order – this option should be enabled to mitigate the BEAST attack.

Allow Client Renegotiation – this option should be set to 'No Client Renegotiation' to mitigate the BEAST attack.

If these options are set as shown, this should prevent the BEAST attack, and should also help to mitigate DoS attacks and MITM Attacks.

SSL – Advanced Configuration

EDIT CONFIGURATION > SSL TERMINATION

SSL Termination		
Logging	off ▼	?
Client Timeout	30	?
Global Server Timeout	60	?
Ulimit		?
Transparent Proxy	off ▼	?
Disable Write	off ▼	?

Update

Logging – Activate detailed logging of the Pound SSL termination service. When activated the Pound log is written to /var/log/poundssl.

Client Timeout – Configure the global client response timeout in seconds. This setting should not require changing.

Global Server Timeout – Configure the global Real Server response timeout in seconds. This setting should not require changing.

Ulimit – Set Ulimit value for pound process. This setting will change the maximum number of file descriptors available to the pound process. The default is 81000.

Transparent Proxy – Enable TPROXY support in Pound SSL. The combination of Pound, TPROXY, and HAProxy allows SSL termination on the load balancer whilst passing the client's IP address to the Real Servers. This option only enables TPROXY in Pound – you will also need to enable TPROXY for HAProxy below, and add appropriate rules to the firewall.



One consequence of using transparent proxy with both Pound and HAProxy is that you can no longer access the HAProxy virtual service directly. With transparency turned on HAProxy will only accept traffic from Pound. The way around this is to create two HAProxy virtual services. One will listen on port 80, and be your standard HTTP service. The other will listen on a different port ; 81 for example – and will be the destination for traffic from Pound.

Disable Write – Disable Writing to Configuration File. When enabled it will stop the user interface overwriting the configuration files so manual changes can be made.



For more details on SSL & Pound configuration steps, please refer to page 113.



For more details on using Tproxy, please refer to pages 119-120.

Heartbeat Configuration

EDIT CONFIGURATION > MODIFY HEARTBEAT CONFIGURATION

Serial	<input checked="" type="checkbox"/>	?
Unicast	<input type="checkbox"/>	?
Broadcast (Deprecated)	Off v	?
UDP Port for broadcast & unicast	<input type="text" value="6694"/>	?
Keepalive	<input type="text" value="3"/>	?
Deadtime	<input type="text" value="10"/>	?
Warntime	<input type="text" value="5"/>	?
Ping node	<input type="text"/>	?
Automatic Fail-back	<input checked="" type="checkbox"/>	?

Modify Heartbeat configuration

Serial – Enable or disable heartbeat master/slave communication over the serial port. Serial communication is the preferred method for load balancer pairs located in close proximity.

N.B. Disabling serial communication will automatically activate console access via the serial port.

Unicast – Enable unicast heartbeat master/slave communication. This method of heartbeat communication uses unicast UDP between master and slave, with a destination port given by the UDP Port for broadcast & unicast parameter.

When unicast is enabled, the load balancer determines the correct interface and IP addresses to use based upon the configured slave IP address. Please ensure that the correct slave IP has been entered on the DNS & Hostname page before enabling unicast.

Unicast is the preferred communication method if serial cannot be used.

Broadcast – Enable broadcast heartbeat master/slave communication, and choose the interface. This method of heartbeat communication uses broadcast UDP between master and slave, with a destination port given by the UDP Port for broadcast & unicast parameter.

Care must be taken when using broadcast on multiple pairs of load balancers in the same network. Each high-availability pair must operate on a different UDP port if they are not to interfere with each other.

If heartbeat communication over the network is required, it is recommended that unicast be used in preference to broadcast.

UDP Port for unicast & broadcast – The UDP port number used by heartbeat for network communication over unicast or broadcast. By default, heartbeat uses port 694/udp for unicast or broadcast communication. If you have multiple load balancer pairs on the same subnet, and wish to use broadcast, you will need to set each pair to a different UDP port.

Keepalive – Specify the number of seconds between keepalive pings. The Keepalive setting must be less than the warntime and deadtime.

Deadtime – The number of seconds communication can fail before a fail over is performed. A very low setting of deadtime could cause un-expected fail overs.

Warntime – If communication fails for this length of time write a warning to the logs. This is useful for tuning your deadtime without causing failovers in production.

Ping node – Specify a mutually accessible IP address to test network availability. A good ping node to specify is the IP address of a router that both the master and slave node can access. If one node loses access to the ping node then a failover will occur. However if both nodes lose access nothing will change.

Automatic Fail-back – When the master returns to service after a failure, should it become active again? This option controls the cluster behavior when the master returns to service after a failure. With Automatic Fail-back enabled, the master will automatically return to active status, taking back the floating IP addresses from the slave. With Automatic Fail-back disabled, the slave will remain active and will retain the floating IP addresses. Fail-over back to the master may then be controlled manually.



For more details on heartbeat, please refer to page 121.

Floating IPs

In order for the load balancer to function, the unit must physically own the Virtual IP address that the clients are accessing before they get re-directed to a Real Server in the cluster. The Floating IP(s) are controlled by heartbeat to ensure that only one of the load balancers (normally the master) owns the Floating IP(s). The floating IP(s) are added automatically when new Virtual Servers are created. It's also possible to manually define the Floating IP(s) if required, this is normally only required when in layer 4 NAT mode where it's recommended to use a floating IP address for the default gateway for the Real Servers.

EDIT CONFIGURATION > EDIT FLOATING IP

192.168.2.122	[Delete]
192.168.2.123	[Delete]

EDIT CONFIGURATION > ADD NEW FLOATING IP

Update

To add an IP address simple type the address into the field and click update. The IP address must be on a valid subnet for the load balancer.



NOTE: Floating IPs are not deleted automatically when Virtual Servers are removed or modified, this must be done manually.

Hostname & DNS

By default, all appliances are configured as master units. This is controlled via a drop down on the Hostname & DNS screen. The self explanatory options are *lbmaster* and *lbslave*.

When the wizard is used to configure a master/slave clustered pair, the slave unit is configured first and then the master unit. When configuring manually, it's common to setup the master first, then add the slave later. If this is done the *Force full slave sync* option should be used to force all settings from the master to be replicated to the slave unit.

EDIT CONFIGURATION > HOSTNAME & DNS

Hostname:	lbmaster ▼	?
Slave Load Balancer:	192.168.2.121	?
Force full slave sync:	<input type="checkbox"/>	?
Domain Name Server:	192.168.2.1	?
Domain Name Server2:		?

Update

Hostname – Is this unit the master or slave? The hostname must be correct for heartbeat and replication to work as expected.

Slave load balancer – Specify the slave load balancers IP address. The slave load balancers IP address is required to activate replication of configuration data.

Force full slave sync – Force all current configuration files to the slave unit. If the slave has been disconnected from the network and changes have been made to the master you can force all changes across in one go using this option.

Domain Name Server – Specify the IP address of a Domain Name Server. This is required for the online feature and security updates to work, it also enables the reverse look up of IP Address information in reports.

Entering a DNS address will allow any reports that need to carry out a reverse lookup to work correctly and will also allow on-line updates via the Loadbalancer.org web site.

Domain Name Server2 – Specify the IP address of a second Domain Name Server.

Network Interface Configuration

Depending on the type of appliance you are using you may have either 2 or 4 network ports. For units with two interface cards *eth0* is normally used as the internal interface and *eth1* for the external interface. However, unlike other appliances on the market you can use any interface for any purpose giving flexibility to configure the unit as required.

In a standard one-arm configuration you would just need to configure *eth0*, the netmask and the default gateway.

Typical configurations:

For layer 4 DR mode, only one interface is used – typically *eth0*

For layer 4 NAT mode, two interfaces are normally required, *eth0* for internal, *eth1* for external

For layer 7 (HAProxy), either one or two interfaces can be used depending on your requirements

EDIT CONFIGURATION > NETWORK INTERFACE CONFIGURATION

Bonding

Bond eth0 & eth1 as bond0: ☐ ? Bond Interfaces

VLAN

Interface: eth0 ▼ ? Add VLAN

VLAN ID: 1 ?

IP Address Assignment

eth0

192.168.2.21/24

eth1

Configure Interfaces

Bonding

Bond eth0 & eth1 as bond0 – Create a bonded interface. This combines eth0 and eth1 as bond0. For units with 4 interfaces, an additional option to bond eth2 and eth3 is shown.

N.B. If you are using heartbeat over Ethernet you should modify the heartbeat configuration to use the new interface.

VLAN

802.1q VLANs can be defined here. This is typically required if your Real Servers are connected to specific VLANs. The exact requirements depend on your infrastructure. Native 8021q VLAN support can be enabled to load balance clusters on multiple VLAN.

IP Address Assignment

Add single or multiple IP addresses to the interfaces:

IP Address Assignment

eth0	10.10.1.10/16
eth1	192.168.2.120/24 192.168.8.120/24

Configure Interfaces

To apply new settings, click **Configure Interfaces**.



WARNING: Obviously it's best to modify network settings whilst the unit is available locally!



For more details on configuring the network, please refer to page 75.

Routing

Used to configure the default gateway and any static routes for the load balancer.

EDIT CONFIGURATION > ROUTING

Default Gateway			
IP v4	<input type="text" value="192.168.64.1"/>		
IP v6	<input type="text"/>		

Static Routes			
Subnet	<input type="text"/>	via gateway	<input type="text"/>
Subnet	<input type="text"/>	via gateway	<input type="text"/>
Subnet	<input type="text"/>	via gateway	<input type="text"/>

Default Gateway

IPv4 – set IPv4 default gateway.

IPv6 – set IPv6 default gateway.

Static Routes

Specify any required static routes.

To apply new settings, click **Configure Routing**.

System Date & Time

The load balancer's local clock is updated once a day using ntp, this requires that your default gateway and DNS are set correctly and that the load balancer has access to the ntp servers. By default, the ntpdate command is set to run in /etc/crontab on a daily basis and contacts time.nist.gov for updates.

EDIT CONFIGURATION > CHANGE THE LOCAL TIME ZONE

The current Date, Time & Time Zone is:

Wed May 30 12:41:59 BST 2012 change ?

Please select a time zone ...

Update

NB. The internal clock is updated once a day using NTP.

Change – displays update fields for date and time

May 30 2012 12:43:20

Select a time zone – Timezone can be Coordinated Universal Time (UTC) or GMT based like GMT, GMT+1 hour, GMT-1 hour, and so on. Please consider that the GMT+/-X format as it is returned by the system differs from the GMT +/- X hours format. The GMT+/-X based statement follows the **POSIX** standard which means that GMT+X is X hours west of Greenwich. GMT-X means X hours east of Greenwich. So GMT+X means GMT-X hours and vice-verse.

Physical – Advanced Configuration

Used to configure various load balancer settings.

EDIT CONFIGURATION > PHYSICAL - ADVANCED CONFIGURATION

Internet Access:		
Proxy IP Address	<input type="text"/>	?
Proxy Port	<input type="text"/>	?
Firewall:		
Connection Tracking table size	<input type="text"/>	?
SMTP Relay:		
Smart Host	<input type="text"/>	?
Syslog Server:		
IP or Hostname	<input type="text"/>	?
<input type="button" value="Update"/>		

Internet Access

Proxy IP Address – Set the IP address of the Proxy Server.

Proxy Port – Set the port of the Proxy Server.

Firewall

Connection tracking table size – Set the size of the firewall connection tracking table in number of connections, each connection entry uses approximately 300 bytes of memory. The default table size is 524,288 connections. High traffic load balancers using NAT mode, or using connection tracking in the firewall script, may see the connection tracking table fill up. Systems experiencing this problem will report the following in the kernel log:

```
ip_conntrack: table full, dropping packet.
```

SMTP Relay

Set an SMTP smart host to receive all mail messages generated by the load balancer. By default, email alerts will be sent to the mail server defined in the destination domain's DNS MX record. Configuring a hostname or IP address here will, instead, direct all outgoing mail to the specified relay.

Syslog Server

Configure a Remote Syslog Server. When enabled, all log messages will be transmitted to the remote server. The server may be specified by IP address or hostname.

N.B. If you use a hostname, make sure DNS is correctly configured on the loadbalancer.

Setup Wizard

Starts the setup wizard.

EDIT CONFIGURATION > SETUP WIZARD

Is this unit part of an HA-pair? ☐ yes ☐ no

N.B. Currently the setup wizard can only be used to configure Layer 4 DR mode or NAT mode services. layer 7 services must be configured manually.

Upgrade Appliance

This option allows a license key to be entered to unlock the R16 restrictions. The key is provided when an Enterprise license is purchased.

EDIT CONFIGURATION > UPGRADE LICENSE

This unit is currently limited to 4 Virtual Services, each with 4 Real Servers. Please enter your license key to upgrade your unit and remove this limit.

If you do not have a license key, please contact sales@loadbalancer.org

Install License Key

Enter the license key provided and click **Install License Key**. To purchase an upgrade key please email sales@loadbalancer.org

Execute Shell Command

Allows OS level commands to be run via the WUI.

EDIT CONFIGURATION > EXECUTE SHELL COMMAND

Execute shell command

This allows you to execute a shell command as user root. The output of the command will be displayed on screen.

WARNING: You should know what you are doing if you use this function.

Maintenance

Backup & Restore

MAINTENANCE > BACKUP & RESTORE

Backup

- Download XML configuration file
- Download Firewall script
- Download SSL Certificates
- Make local XML backup
- Make local Firewall script backup

Restore

- Upload XML file & Restore:
- Restore from the last local XML backup
- Restore Manufacturer's defaults

Backup

Download XML configuration file – download and save the load balancer's XML configuration file.

Download Firewall script – download and save load balancer's firewall script.

Download SSL Certificates – download and save the load balancer's SSL certificates.

Make local XML Backup – creates a backup of the current XML file in /etc/loadbalancer.org/userbkup.

Make local Firewall Script Backup – creates a backup of the current rc.firewall in /etc/loadbalancer.org/userbkup

Restore

Upload XML file & Restore – upload an XML file and restore load balancer settings. Once the restore completes the following messages are displayed:

MAINTENANCE > BACKUP & RESTORE > RESTORE

Restoring network interfaces...

If the restored configuration removes the IP address that you are using to connect to the web interface, you will need to reconnect to the load balancer on one of its new IP addresses.

Restoring heartbeat configuration...

Restoring Layer 4 configuration...

Restoring HAProxy configuration...

Restoring Pound configuration...

Restoring Graph configuration...

Resetting System Overview Graphs...

Restoring Syslog configuration....

Information: Restored configuration from uploaded file.

Warning: Please note that heartbeat has been stopped to prevent interference with a running peer. When the configuration of this node is correct, heartbeat **must be restarted**.

Restore from the last local XML backup – Restore the last local backup created with the 'Make local XML Backup' option.

Restore Manufacturer's defaults – Restore system settings to default values.

N.B. Currently the upload facility is not backward compatible with previous major versions of the software, i.e. it is not possible to restore a V6.x XML file to a v7.x appliance.

Restart Services

Enables system services to be restarted and reloaded.

MAINTENANCE > RESTART SERVICES



Restart Ldirectord

Restart Layer 4 Services. Restarting Ldirectord will result in a loss of layer 4 services during the restart. This causes the related process to be stopped and a new instance started. Generally only needed if Ldirectord has failed for some reason and needs to be started again from scratch.

Reload Ldirectord

Reload Layer 4 Services. The Ldirectord configuration is re-read and re-applied. Note that a reload occurs automatically whenever a layer 4 VIP or RIP is added, deleted or modified.

Restart HAProxy

Restart Layer 7 Services. Restarting HAProxy will result in a loss of layer 7 services during the restart. Restarting HAProxy will cause any persistence tables to be dropped and all connections to be closed, it's a complete restart and reload of the HAProxy configuration.

Reload HAProxy

Reload Layer 7 Services. HAProxy will start a new process (leaving the old one) with the new configuration. New connections will be passed onto this process, the old process will maintain existing connections and eventually terminate when there are no more connections accessing it. If you are using stick tables for persistence the entries will be copied between processes.

N.B. If you have long lasting tcp connections it can take quite some time for the old process to terminate, leaving those users running the old configuration. If this is taking too long – See Restart HAProxy.

Clear HAProxy Stick Table

Clears All HAProxy persistence tables. If you are using a Layer 7 persistence mode that relies on stick-tables (IP persistence or RDP cookie persistence), this option will clear all entries from these tables. Once cleared, clients may be directed to a different server upon re-connection.

Restart Pound

Restart SSL Termination Services. Restarting SSL Termination will result in a loss of SSL termination services during the restart.

Restart Heartbeat

Restart Heartbeat Services. Restarting Heartbeat will result in a loss of service during the restart. Restarting heartbeat will cause a temporary loss of all layer 4, layer 7 and SSL services.

Reload Heartbeat

Reload Heartbeat Services. If the configuration has not changed then nothing will happen. If the config has changed, a restart will occur. Restarting Heartbeat will result in a loss of service during the restart. Restarting heartbeat will cause a temporary loss of all layer 4, layer 7 and SSL services.

Restart Iptables

Restarts iptables. This will clear then re-read and re-apply the firewall rules.

Restart Syslogd

Restart the syslog services.

Reload Syslogd

Reload the syslog services.

System Control

MAINTENANCE > SYSTEM CONTROL

- Restart server
- Halt server

Restart Server – Shutdown and restart the appliance.

Halt Server – Shutdown and halt the appliance.

Software Update

This option can be used to access Loadbalancer.org's online update facility. If no newer updates are available, the following message is displayed:

MAINTENANCE > SOFTWARE UPDATE

Online Update

Information: Version v7.4 is the current release. No updates are available.

If valid updates are available, a screen similar to the following is displayed (this example shows 7.4 to 7.4.1) -

MAINTENANCE > SOFTWARE UPDATE

Online Update

Online updates are only available if your organisation has a valid authorisation key.

An authorisation key may be obtained from [Loadbalancer.org support](#).

Before starting the online update, we recommend that you backup the XML configuration file, firewall script, and any manual changes that have been made.

- Download XML Configuration File
- Download Firewall Script

Update from v7.4 to v7.4.1

Changes in this release:

- Update graphing to support new Virtual Service options.
- Modify the Pound SSL Proxy to mitigate the BEAST attack.
- Modify heartbeat configuration to support large numbers of Floating IP addresses.
- Make configuration upload independent of peer device.
- Fix bug in configuring VLAN interfaces where the underlying interface does not have an allocated IP address.
- Ensure that old VLAN interfaces are removed when the configuration is reset, or a configuration file is uploaded.
- Ensure that multiple-port Layer 4 Virtual Services are not affected by changes in the user firewall script.
- Fix synchronisation of global options to peer device.
- Fix HA pair status display when heartbeat is in broadcast mode.

Warning: Updates should only be installed during a maintenance window.

Note: When the online update is started, some web browsers will remain in page loading state for an extended period. Once the update archive has been downloaded, the page display will update.

Authorisation Key

Online Update

If you have a current maintenance agreement for your appliance you can use this form to check for new online updates and install them. For the update to succeed, you will need:

- You will need a valid authorization key.
- You will need your default gateway & DNS correctly configured.
- You will need HTTP access to www.loadbalancer.org enabled through your firewall.

Updates are also available as a complete downloadable ISO software image if preferred.

N.B. If you have a clustered pair, a separate code is provided for each appliance. See page 154-155 for details on how to update a clustered pair.

Fallback Page

This section allows you to view and modify the local holding page on the load balancer. If you have a master and slave load balancer then you must change this on both servers. The fallback server on the load balancer is an implementation of NGINX.

MAINTENANCE > FALLBACK PAGE

```
<html>
<head>
<title>The page is temporarily unavailable</title>
<style>
body { font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body bgcolor="white" text="black">
<table width="100%" height="100%">
<tr>
<td align="center" valign="middle">
The page you are looking for is temporarily unavailable.<br/>
Please try again later.<br/>
(port reminder 9080)
</td>
```

Update

Fallback Page for Layer 4 Services

The fallback page is displayed when all Real Servers fail. The fallback page is NOT displayed when servers are taken offline manually via the WUI. At layer 4, to cause the fallback page to be displayed when Real Servers are taken offline, you would also need to force the Real Server to fail its health check by for example disabling the relevant service on the Real Server.

Fallback Page for Layer 7 Services

For layer 7 VIPs the fallback page is displayed when all Real Servers are unavailable *AND* when all are taken offline via the WUI. The page can be hosted on the load balancer or on an external server. Set the Fallback Server option of the VIP accordingly.



For more details on Fallback server settings , please refer to page 130.

Firewall Script

This form allows you to directly edit /etc/rc.d/rc.firewall.

MAINTENANCE > FIREWALL SCRIPT

```
#!/bin/sh
# $Id: rc.firewall 151 2009-07-15 13:54:37Z thorsten $
# Firewall script for Load Balancer

##### SETUP #####
# Remove any existing rules from all chains
/etc/rc.d/rc.flush-iptables

# Allow unlimited traffic on the loopback interface
#iptables -A INPUT -i lo -j ACCEPT
#iptables -A OUTPUT -o lo -j ACCEPT

##### END OF SETUP #####

# Set up some useful variables

##### NAT MODE #####

# Packet forwarding enabled by default in sysctl.conf

# For one-arm NAT you will also need to disable re-directs
# Load balancer is the default gateway for real servers
# so turn OFF icmp redirects (1 on, 0 off)
#echo "0" >/proc/sys/net/ipv4/conf/all/send_redirects
#echo "0" >/proc/sys/net/ipv4/conf/default/send_redirects
```

Update



WARNING: Be careful! - make a backup before changing this script so that you know you can roll everything back if you cause a problem.

This can either be used for belt & braces security; for example to replicate your normal firewall settings onto the load balancer as well for double security. What kind of settings? Normally you don't want any customers to be able to access the administration IP address on the load balancers, you only want them to have access to the Virtual Services – typically ports 80 & 443.

You can also use the firewall script to group ports together using Firewall Marks (see page 132).

If you are planning to use NAT mode you may also want to use the load balancer as your main firewall which is fine, but we think it is a lot simpler to keep your firewall separate from your load balancer. Especially if you want to set up VPNs etc.

If you wish to clear the firewall tables completely use the following command from the console:

```
/etc/rc.d/rc.flush-iptables
```

Firewall Lock Down Wizard

The firewall lock down wizard automatically configures the load balancer to allow access to the various admin ports from one specific IP address or subnet. The wizard automatically detects the IP of the client running the WUI and inserts this into the Admin IP field. The default mask is set to 255.255.255.0. If you need to specify an administration network, change the mask as required.

The lock down wizard will allow full access to all the defined VIPs and reply traffic from the defined Real Servers.

The generated script is stored here: `/etc/rc.d/rc.lockdownwizard`

This script is activated at the end of the `/etc/rc.d/rc.firewall` script.

Any changes that you have already made to the `/etc/rc.d/rc.firewall` script are kept in place.

MAINTENANCE > FIREWALL LOCK DOWN WIZARD

Warning: This will block all access to the load balancer unless it matches the Admin IP or a Virtual Service IP

Administration subnet:

Firewall Lock Down Wizard

[[Modify the firewall lockdown wizard script](#)]

Enabling the lockdown script

Using the **Administration subnet** field, define the IP address of your management computer or subnet. This is auto-configured to be the IP address of the computer used to run the WUI. To apply the setting, click the **Firewall Lock Down Wizard** button.

N.B. Make sure that the subnet mask is correct – by default a /24 mask is displayed.

Clearing the lockdown script

Click on the **[Modify the firewall lockdown wizard script]** link, select and delete all text, then click the **Firewall Lock Down Wizard** button.

N.B. If you accidentally block your own access to the appliance you will need to clear the current firewall rules and try again. to clear the firewall tables completely use the following command from the console:

```
/etc/rc.d/rc.flush-iptables
```



The firewall lockdown wizard should only be run after the load balancer is fully configured and tested. If changes are made later to the load balanced services, the wizard should be re-run to ensure these changes are reflected in the lockdown script.

Initialize Graphs

This option will construct a series of RRDTool databases and relevant cron jobs to update those databases using the output from LVSGSP. More cron jobs are then used to generate the daily, weekly, monthly and yearly charts accessible from the reports section.

This option should be run after you have configured your Virtual & Real Servers. If you later add additional VIPs or RIPv you will need to re-run this again.



WARNING: All current statistics will be lost when this function is used.

Disable graphs can be used to stop graphing.

Passwords

This section is used to manage user accounts that have access to the WUI.

MAINTENANCE > PASSWORDS

loadbalancer	[Modify]	
reportuser	[Modify]	[Delete]
maintuser	[Modify]	[Delete]

MAINTENANCE > ADD NEW USER

Username *	<input type="text"/>
Password *	<input type="password"/>
<input type="button" value="Add New User"/>	

The default usernames and passwords, their default group membership and their primary use are as follows:

Username	Default Password	Default Group	Use (for full permission details see the table below)
loadbalancer	loadbalancer	config *	appliance administration account
reportuser	reportuser	report	viewing the appliance configuration, reports & logs
maintuser	maintuser	maint	same as reportuser, can also take servers on/off line & create the support download archive file

* It's not possible to change the default group for user 'loadbalancer'

N.B. These are simple Apache .htaccess style accounts and are not related to the local Linux accounts.

The permissions for each group are shown below:

Group	Menu / Access						
	System Overview	View configuration	Edit Configuration	Maintenance	Reports	Logs	Support
config	Full	View	Full	Full	Full	View	Full
report	View	View	None	None	Full	View	View
maint	Full	View	None	None	Full	View	Full

Resetting Passwords

It's possible to reset passwords via the command line if required. To do this you'll need to run root access to the console or terminal session. e.g. to change the password for user 'loadbalancer' use the following command :

```
htpasswd -b /etc/loadbalancer.org/passwords loadbalancer <new password>
```



SECURITY: Don't forget to change your root password from the console using the passwd command! Please also refer to the security section on page 156

Adding New Users

New users can be added using the **Add New User** option:

MAINTENANCE > ADD NEW USER

Username *

Password *

Add New User

- Simply type in the required Username & Password and click **Add New User**
- By default, new users will be added to the report group (least privilege). To change this, click **[Modify]** next to the user, select the required group and click **Edit User**

Reports

Used to display a variety of reports for monitoring the system.

Layer 4 Status

This report shows the current weight and number of active & inactive connections for each Real Server. If a Real Server has failed a health check, it will not be listed.

Use the *Logs > Layer 4* option to view the *ldirectord* log file if expected servers are not listed.

Layer 4 Traffic Rate

This report shows the current connections per second and bytes per second to each Real Server. If a Real Server has failed a health check, it will not be listed.

Layer 4 traffic Counters

This report shows the volume of traffic to each Real Server since the counters were last re-set. If a Real Server has failed a health check, it will not be listed.

Layer 4 Current Connections

The current connections report is very useful for diagnosing issues with routing or ARP related problems. In the example below, the state is shown as *SYN_RECV*, this is normally a good indication that the ARP problem has not been solved. In NAT mode, this is a good indication that the Real Servers default gateway has not been configured to be the load balancer and therefore return traffic is not routed correctly.

REPORTS > LAYER 4 CURRENT CONNECTIONS

```
IPVS connection entries
pro expire state      source          virtual         destination
TCP 00:51 SYN_RECV    192.168.2.7:64763 192.168.2.109:80 192.168.2.99:80
```

Layer 4 Current Connections (resolve hostnames)

This is the same as the current connections report but is slower as it looks up the DNS name of each IP address.

N.B. These reports are generated in real time. Direct Routing is the default load balancing method and you will not see any stats for return packets (as they do not pass through the load balancer). You would however see them if using NAT mode.

Layer 7 Status

This report is provided by the stats instance of HAProxy. This web page contains the current live status of all of the configured layer 7 HAProxy virtual and Real Servers.

Log in using: **Username:** loadbalancer
Password: loadbalancer

N.B. This password can be changed using Edit Configuration > Layer 7 – Advanced Configuration

HAProxy version 1.5-dev7-lb1, released 2011/09/23

Statistics Report for pid 8836

> General process information

pid = 8836 (process #1, nbproc = 1)
uptime = 0d 0h01m05s
system limits: memmax = unlimited; ulimit-n = 81000
maxsock = 80024; maxconn = 40000; maxpipes = 0
current conns = 146; current pipes = 0/0; conn rate = 419/sec
Running tasks: 2/150; idle = 87 %

active UP
active UP, going down
active DOWN, going up
active or backup DOWN
active or backup DOWN for maintenance (MAINT)
backup UP
backup UP, going down
backup DOWN, going up
not checked

Note: UP with load-balancing disabled is reported as "NOLB".

Display option:

- [Hide 'DOWN' servers](#)
- [Refresh now](#)
- [CSV export](#)

External resources:

- [Primary site](#)
- [Updates \(v1.5\)](#)
- [Online manual](#)

HTTP		Queue		Session rate		Sessions				Bytes		Denied		Errors		Warnings		Server										
		Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
Frontend					415	587	-	144	182	40 000	7 731			449 518	10 985 451	0	0	0										
backup	0	0	-	0	0	0		0	0		0	0	0	0	0	0	0	0	OPEN						1	-	Y	
rip1	0	0	-	366	573		73	100	-	7 660	7 660	449 518	10 985 451	0	0	0	0	0	1m5s UP	L4OK in 0ms	1	Y	-	0	0	0s	-	
Backend	0	0		366	573		73	100	4 000	7 660	7 660	449 518	10 985 451	0	0	0	0	0	1m5s UP		1	1	1		0	0s		

	stats																														
	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle		
Frontend				3	4	-	2	2	2 000	26		13 152	265 191	0	0	0					OPEN										
Backend	0	0		0	0		0	0	200	0	0	13 152	265 191	0	0		0	0	0	0	1m5s UP		0	0	0		0				

Layer 7 Stick Table

Displays the layer 7 stick tables. For example, if a layer 7 VIP is created using RDP cookie persistence, a stick table will be used. The related VIP is then available in the drop-down as shown below:

REPORTS > STICK TABLE (HAPROXY)

RDP



1 Entries Returned (Max Entries Returned 1000)

ID	Key	Use	Expires ms	Server
0x2262404:	Rob	use=0	1790927	rip1

Graphing

When first run, you'll be prompted to Initialize Graphs. This creates the required database files that are used to store the data that is used to produce the graphs. Once initialized, the following options are available:

GRAPHING

Interface Throughput Reports

Show Throughput ☒ Hide Throughput

Daily Reports

Show Daily ☒ Hide Daily

Weekly Reports

Show Weekly ☒ Hide Weekly

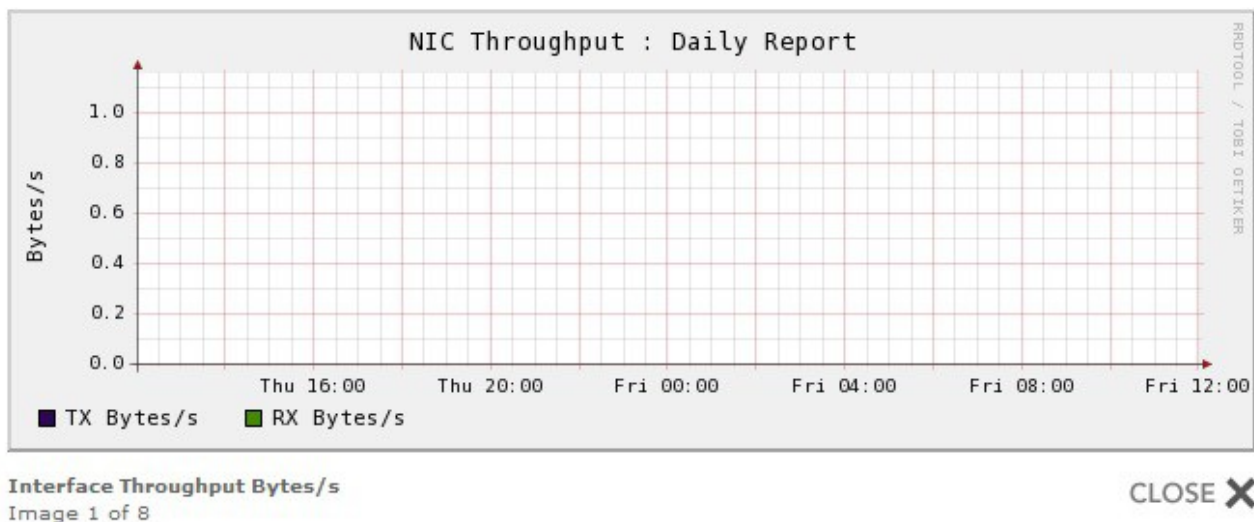
Monthly Reports

Show Monthly ☒ Hide Monthly

Yearly Reports

Show Yearly ☒ Hide Yearly

Once graph data starts to accumulate, the graph links will change from a Loadbalancer.org logo icon to a graph icon. This can then be clicked to open that particular graph.



N.B. If you add Virtual & Real Servers after initializing the graphs, you'll need to initialize them again to include these. Each time this is done, all historical data will be cleared.

Reset Packet Counters

Resets the packet counters to zero for the load balancer reports.

Logs

This menu option is used to display various Appliance log files.

Load Balancer

The Lbadmin log shows all changes made via the admin system. This is very useful for tracking all changes made to the configuration.

Layer 4

The ldirectord log shows the output from the health checking daemon. This is useful for checking how healthy your Real Servers are or pinning down any configuration errors. The logging here can be quite verbose but it clearly shows what the health checking process is doing.

Layer 7

If activated via *Edit Configuration > Layer 7 – Advanced Configuration*, this will show the contents of `/var/log/haproxy.log`. This is a very detailed log of all HAProxy transactions.

SSL Termination

If activated via *Edit Configuration > SSL – Advanced Configuration*, this will show the contents of `/var/log/poundssl.log`. This is a very detailed log of all Pound SSL transactions.

Heartbeat

The heartbeat log shows the status of the heartbeat daemons. Heartbeat is used whether configured as a single device or as a clustered pair. The log provides a detailed real-time status of heartbeat.

Note that heartbeat is used to control all load balanced services, even when deployed as a single appliance.

Support

Contact Us

This option provides details on how to contact Loadbalancer.org, how to report any issues and what information we'll need to resolve issues as quickly as we can. The Loadbalancer.org support team can be contacted using the email address: support@loadbalancer.org

Sending an email to this address creates a ticket in our help desk system and enables all technical support staff to view the case. This is the most efficient way to contact support and guarantees that any reported issues will be acted upon and addressed as quickly and efficiently as possible.

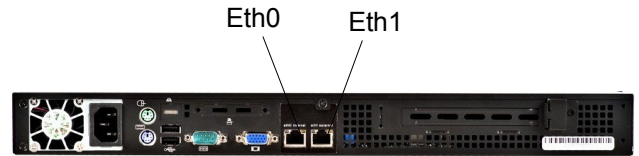
Technical Support Download

This option enables the Support Download to be created. The download is a compressed archive containing all log files and configuration files from the appliance and should be attached to your email.

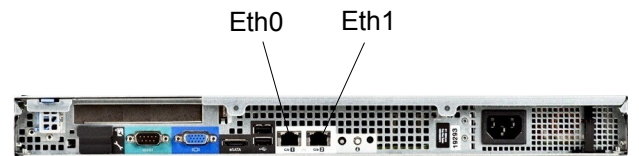
Section H – Appendix

Front & Rear Panel Layouts

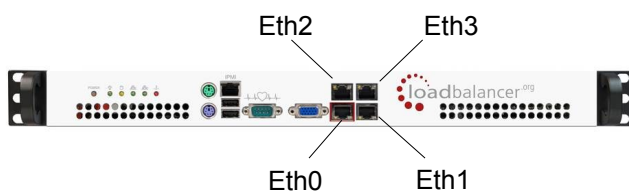
Enterprise / Enterprise R16 – Supermicro



Enterprise – Dell



Enterprise Max – Supermicro



Enterprise Max / 10G – Dell

