



APPLICATION DELIVERY

Scaling Blue Coat ProxySG Secure Web Gateway Using Brocade ServerIron in Service Provider Environments

This document provides best practices for deploying Brocade ServerIron Application Delivery Controllers in Transparent Cache Switching mode with Blue Coat ProxySG Web Gateways. The infrastructure employs features such as client IP spoofing by ProxySG Gateways, Policy-Based Routing, and cache switching by Brocade ServerIrons to handle the large volume of traffic seen in service provider environments.



CONTENTS

| | |
|--|-----------|
| Introduction..... | 3 |
| Best Practice TCS Configuration..... | 4 |
| Cache Load Balancing..... | 5 |
| Load Balancer Scaling..... | 6 |
| IP Spoofing Configuration..... | 7 |
| Step 1..... | 8 |
| Step 2..... | 8 |
| Step 3..... | 9 |
| Step 4..... | 9 |
| Step 5..... | 10 |
| Step 6..... | 10 |
| Step 7..... | 11 |
| Step 8..... | 11 |
| Appendix: ServerIron Command Line Configuration | 12 |
| ServerIron 1 | 12 |
| ServerIron 2 | 17 |

INTRODUCTION

Service providers face great challenges maintaining profitability in a tougher-than-ever environment. To remain competitive, they must

- More effectively manage bandwidth to efficiently monetize their infrastructure and contain costs
- Offer a consistently reliable and interactive user experience to drive customer loyalty
- Stop malware that can compromise user data or disrupt the network
- Meet consumer demand to filter harmful or inappropriate Web content
- Grow their business with high-value managed services

Together, Brocade® and Blue Coat provide a service-oriented platform that lets providers enhance and personalize the end user's Web experience, reduce operational costs, and create new opportunities to monetize network traffic through revenue-generation services.

The use case in this paper is based on real-world service provider scenarios requiring high bandwidth carrier-class caching and filtering. The joint solution uses technologies such as Client IP spoofing by a farm of Blue Coat ProxySG cache servers balanced by a Brocade ServerIron® High Availability (HA) pair connected to a Point of Presence (PoP) router running Policy-Based Routing (PBR).

PBR provides a mechanism for implementing routing of data packets based on the policies defined by the network administrators, and in the use case described in this paper, PBR allows the definition of a flow between the client and the Internet server handling the request to ensure that traffic is directed through the Brocade ServerIron and the Cache servers in both directions. The paper explains a layered approach to implementing policies and thresholds in order to achieve optimal cache load balancing. It also explains the Brocade ServerIron scaling methodology based on specific flows.

In a carrier environment, the origin server frequently performs authentication and accounting based on the client IP address. Cloud services such as Gmail may block sessions with a cache server IP address. In such cases, cache servers such as Blue Coat ProxySG servers need to spoof client IP addresses and send Internet requests using client IP addresses.

Best Practice TCS Configuration

Brocade Serverlrons are placed in a dual-arm design with connections of 10 Gigabit Ethernet (GbE) connections onto the LAN in the PoP. It is assumed that the devices on the LAN in the PoPs can do Policy Based Routing without impacting the performance of the devices. The Serverlrons are configured in an HA pair with session synchronization between them, which ensures that a failure of a unit will not result in session loss.

Brocade ServerIron HA pair

- Dual arm design with 10 GbE links
- Configured with 2 VRRP VRIDs: VRRP-A and VRRP-B
- VRRP-A is the outside VRRP, both PBRs have a next hop of this VRRP
- VRRP-B is the inside VRRP, GW for the cache
- Both Serverlrons perform session synch for HA

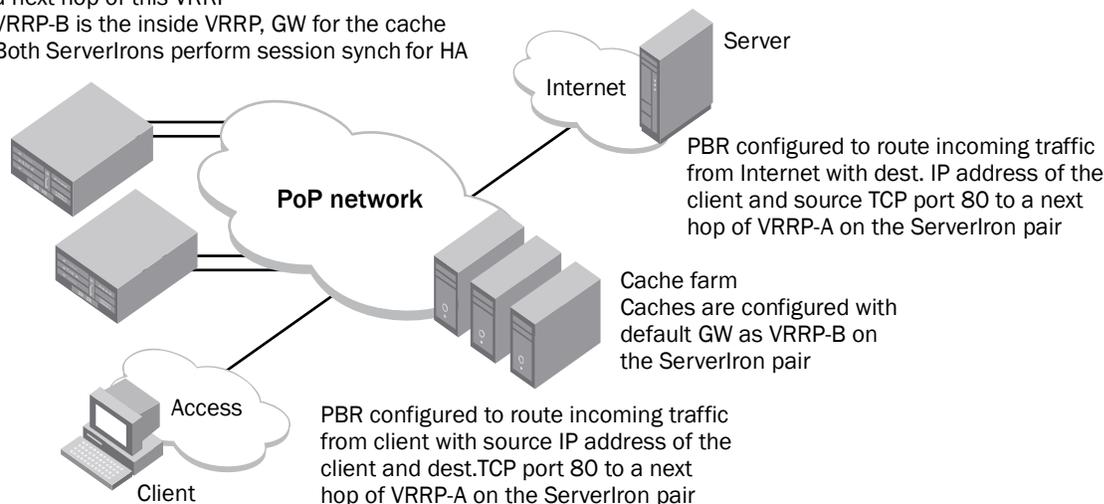


Figure 1. Solution TCS configuration

The Brocade ServerIron pair is also configured with two Virtual Router Redundancy Protocol (VRRP) instances. One is used to send traffic to the ServerIron pair from the outside; the other is used as the default gateway for the caches. The use of VRRP (unlike proprietary implementations) provides a standards-based mechanism for Layer 3 failover between the Serverlrons.

Two Policy-Based Routing policies are configured on a router in the PoP. One is designed to catch outbound user traffic and is configured to route traffic with a source IP address as the client pool and destination TCP port 80 to a next hop as the VRRP address on the ServerIron pair. The other PBR policy is designed to catch return traffic and direct it to the cache setup. This PBR is designed to route traffic that has a destination IP address as client pool and source TCP port 80 to a next hop as the VRRP address on the ServerIron pair. Since the caches are doing IP spoofing and requesting content from the Internet using the client IP addresses, without this PBR, return traffic would bypass the cache setup and go directly to the client, which would then fail.

The default gateway for the caches is configured as a VRRP address on the Brocade ServerIron pair. All requests and the return traffic that goes back to the clients will pass through the Serverlrons.

It is essential for the traffic between the caches and the Internet to pass through the Brocade Serverlrons, since the caches are spoofing client IP addresses. This way the Serverlrons will keep a session entry for this for scaling the infrastructure.

Brocade ServerIrons are carrier class and have the following characteristics:

- They do not rely on an off-the-shelf operating system such as Linux. Rather they use an operating system specifically designed for load balancing switches.
- They do not have any moving parts, such as hard disks, and hence have better Mean Time Between Failure (MTBF) ratings, which makes them more suitable for mission-critical operation.
- They have very fast boot time, which allows the service to start quickly after interruption of service if it should occur.
- The control plane and the data planes are separate, which means that the control plane is not impacted by the load or issues on the data plane. This also implies that you can add resources on the data plane, by adding an Application Switching Mode (ASM), for example, without any changes on the I/O ports, and the same reasoning applies to adding I/O ports without making any changes on the control plane.

Cache Load Balancing

Brocade recommends the use of a layered approach to policies and thresholds in order to achieve optimal load balancing of the caches.

- The *first layer* controls sites with multimedia content, such as youtube.com, which can put a huge strain on the caching setup. It is advisable to segregate caches serving content delivery sites from the rest of the caches. Any traffic destined for such sites is sent to a group of caches, which serve the content. This allows the other caches to serve other types of content without being loaded with the heavy bitstream traffic generated by content delivery sites.
- The next layer is for load balancing, in which load balancing uses IP address-based hashing. The hashing feature uses source hash mask and destination hash mask. The hash mask determines how many of the source and destination IP addresses are used by the hash function. The Brocade ServerIron uses the hash masks to select a cache server. Brocade recommends using destination-IP hash mask, which minimizes duplication of content on the cache servers by ensuring that a particular Web site is always cached on the same cache server.
- The next layer adds more control. Some sites tend to be loaded more heavily than other sites and this impacts some caches more than others, based on the destination IP address hashing. Brocade ServerIron can weight cache load balancing, so a tuning step is required after setups are stable. The caches for the highly loaded sites are given lower weights, and caches that are being sent the less-loaded sites based on the hashing mechanism, will have increased weighting. Overall, this leads to more lightly loaded sites, which leads to overall leveling of the load across the cache servers.
- The next layer of control involves specifying the maximum number of connections that a given cache server can handle. By setting a limit, a condition in which the capacity threshold of a cache server is exceeded can be avoided. If any of the caches reaches its set threshold, the Brocade ServerIron will stop forwarding new requests to the loaded cache server, thereby allowing the number of connections on the cache to decrease and releasing it from load. The requests that were supposed to be handled by the cache that reached its threshold based on the hashing mechanism are load balanced to the other caches in the group. If all the caches reach their limit, as a last resort ServerIron can forward the requests to the Internet to ensure service availability.
- Another layer of control is to set up “reassign thresholds” for the caches on the Brocade ServerIron. The reassign threshold specifies the number of contiguous inbound TCP-SYN packets a cache can fail to respond to before the ServerIron considers it down. The cache usually reaches this state when it is loaded with requests exceeding its designed capacity. When this happens, ServerIron gives the cache some relief by preventing any new connections to the cache and giving it time to recover from the extra load.

Load Balancer Scaling

It is important to clarify how the scaling of load balancing is done. The load balancer has to handle four flows, as shown in Figure 2.

- The client request directed to the cache (Flow A in red)
- The cache request sent to the Web server in the Internet (Flow B in orange)
- The Web server response directed to the cache (Flow C in green)
- The cache response back to the client (Flow D in blue)

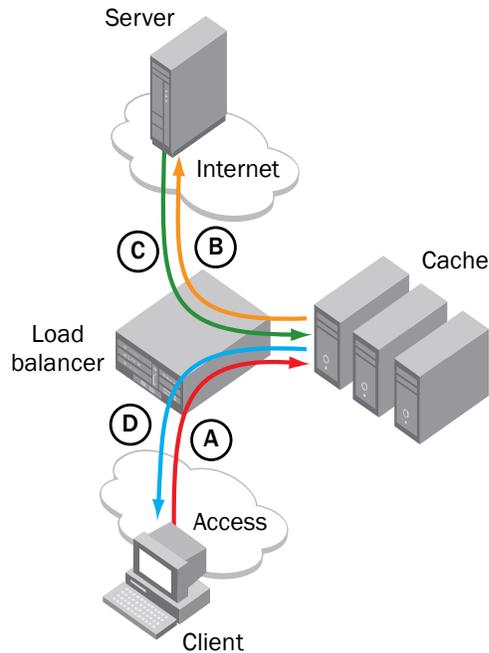


Figure 2. Load balancing flows

For scaling purposes, consider the worst case scenario of having none of the content locally cached so that the caches have to fetch the entire content request from the original Web server. This is a realistic scenario especially when the setup is first put in place.

For this case, Flow D = Flow C.

For scaling the number of requests, Flows A and B were calculated as 40% of the download bandwidth (Flows C and D), which includes the requests as well as any client uploads on HTTP. So the total bandwidth handled by the load balancer = A + B + C + D.

IP Spoofing Configuration

The Brocade ServerIron was designed taking into account IP spoofing, which entails sizing as well as IP spoofing support on the load balancer.

In TCS, when a client makes a request for HTTP content on the Internet, the ServerIron directs the request to a cache server, rather than to the Internet. If the requested content is not on a cache server, it is obtained from an origin Web server on the Internet, stored on a cache server to accommodate future requests, and sent from the cache server back to the requesting client.

When a cache server makes a request for content from the origin server, it can do either of the following:

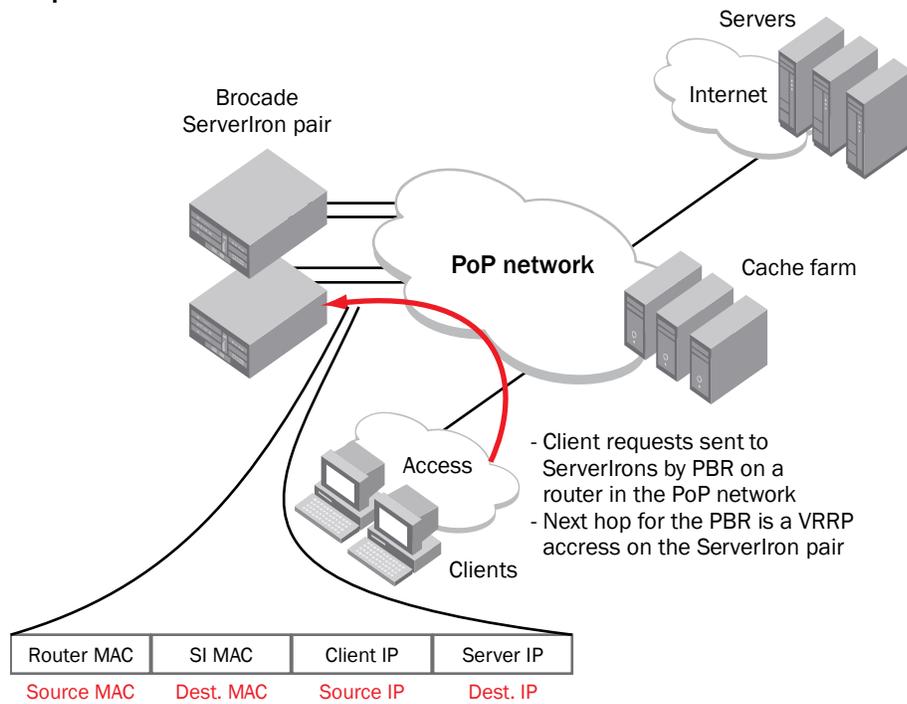
- The cache server replaces the requesting client's IP address with its own before sending the request to the Internet. The origin server then sends the content to the cache server. The cache server stores the content and sends it to the requesting client, changing the source IP address from its own to the origin server's IP address.
- The cache server does not replace the requesting client's IP address with its own. Instead, the cache server sends the request to the Internet using the requesting client's IP address as the source. This allows the origin server to perform authentication and accounting based on the client's IP address, rather than the cache server's IP address. This functionality is known as "cache server IP spoofing."

When cache server spoofing support is enabled, the Brocade ServerIron does the following with requests sent from a cache server to the Internet:

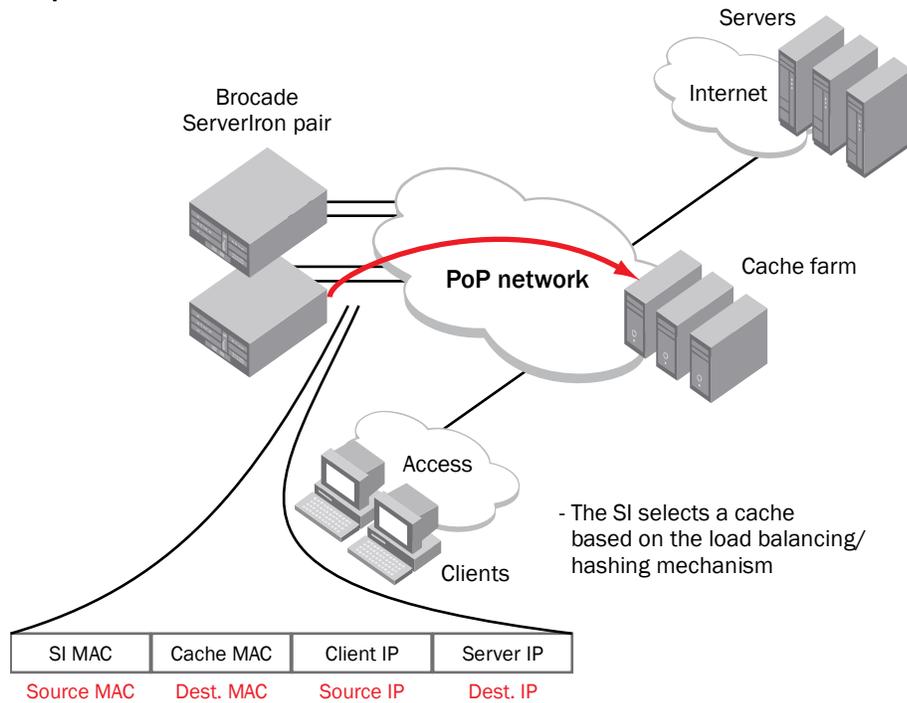
1. The ServerIron looks at the MAC address to see if the packet is from a cache server. The ServerIron uses an ARP request to get the MAC address of each configured cache server.
2. If the MAC address indicates that the packet is from a cache server, the ServerIron checks the source IP address. If the source IP address does not match the cache server's IP address, the ServerIron concludes that this is a spoofed packet.
3. The ServerIron creates a session entry for the source and destination (IP address, port) combination, and then sends the request to the Internet.
4. When the origin server sends the content back, the ServerIron looks for a session entry that matches the packet. If the session entry is found, the ServerIron sends the packet to the appropriate cache server.

The complete traffic flow is illustrated in the following figures.

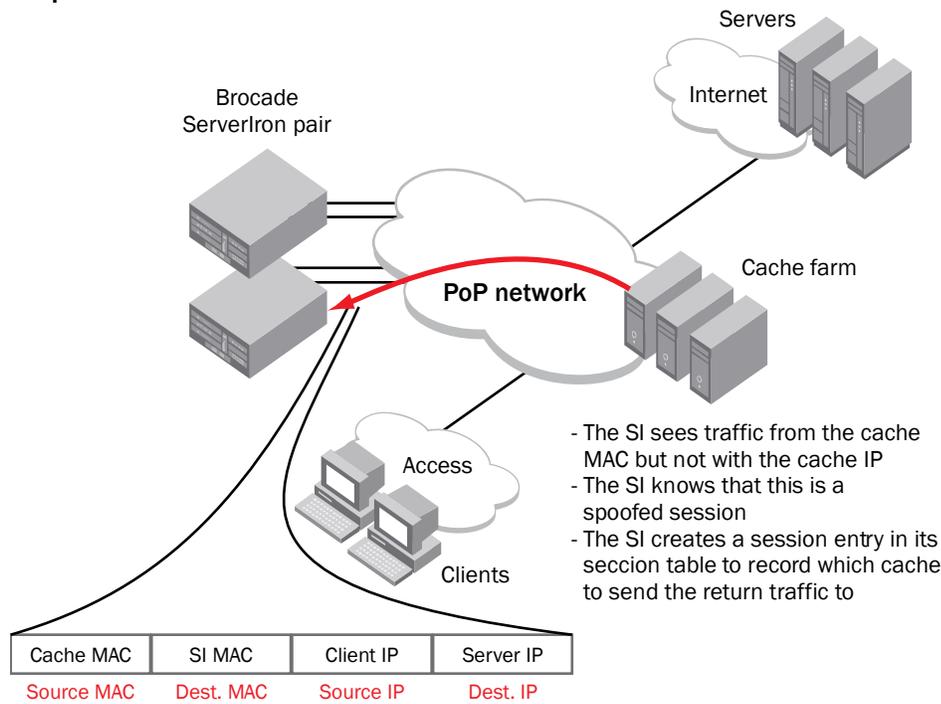
Step 1



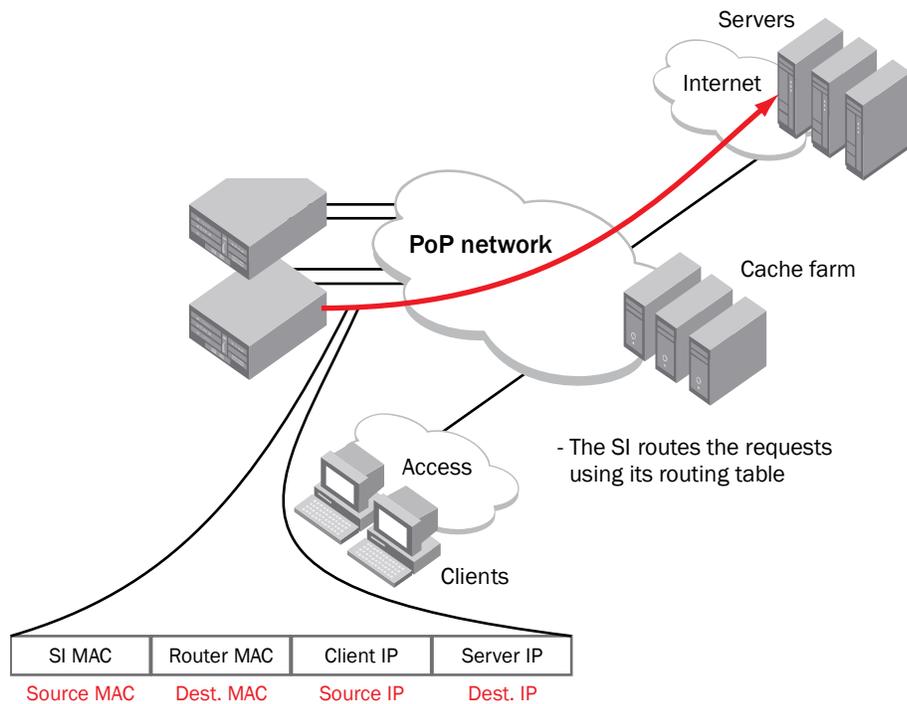
Step 2



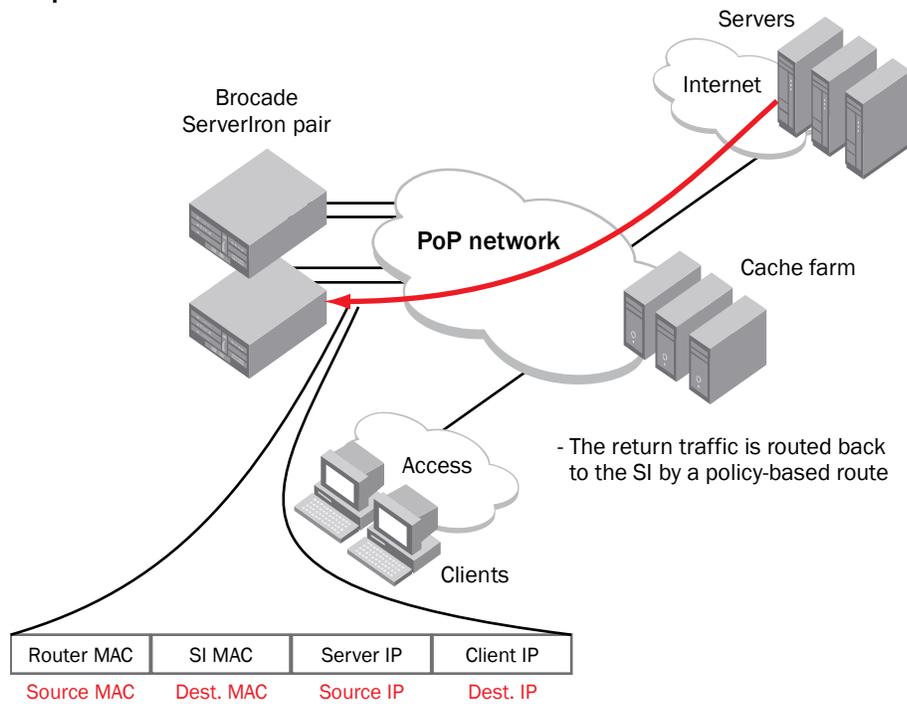
Step 3



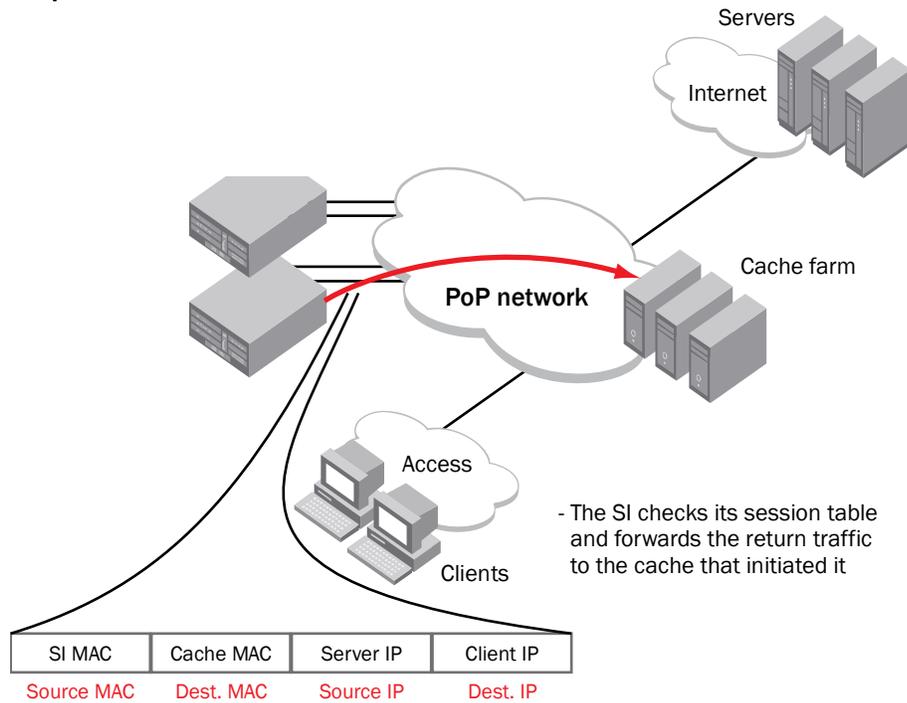
Step 4



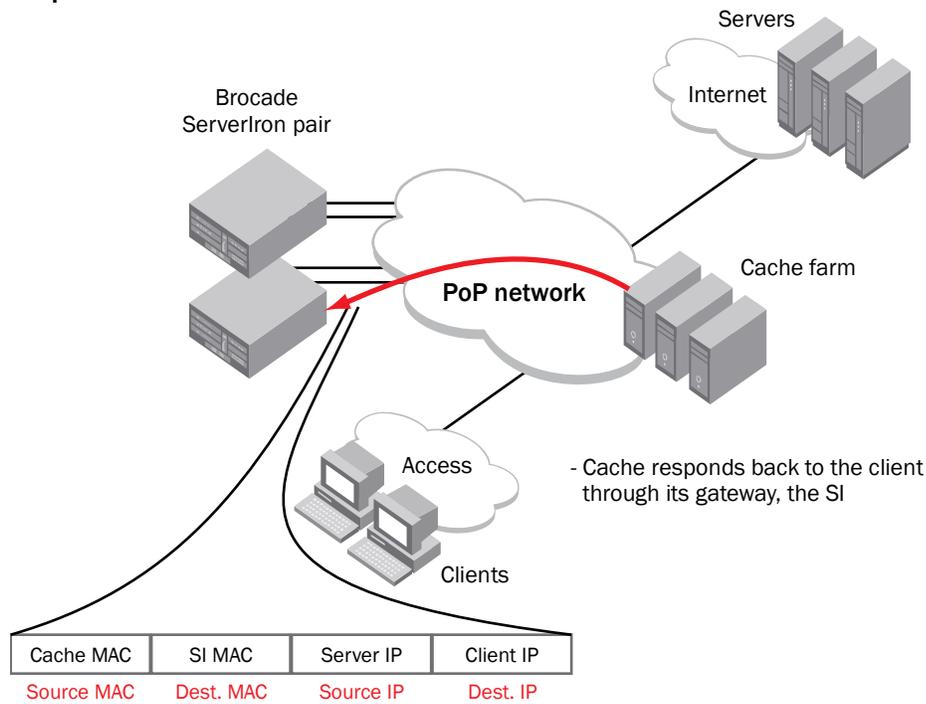
Step 5



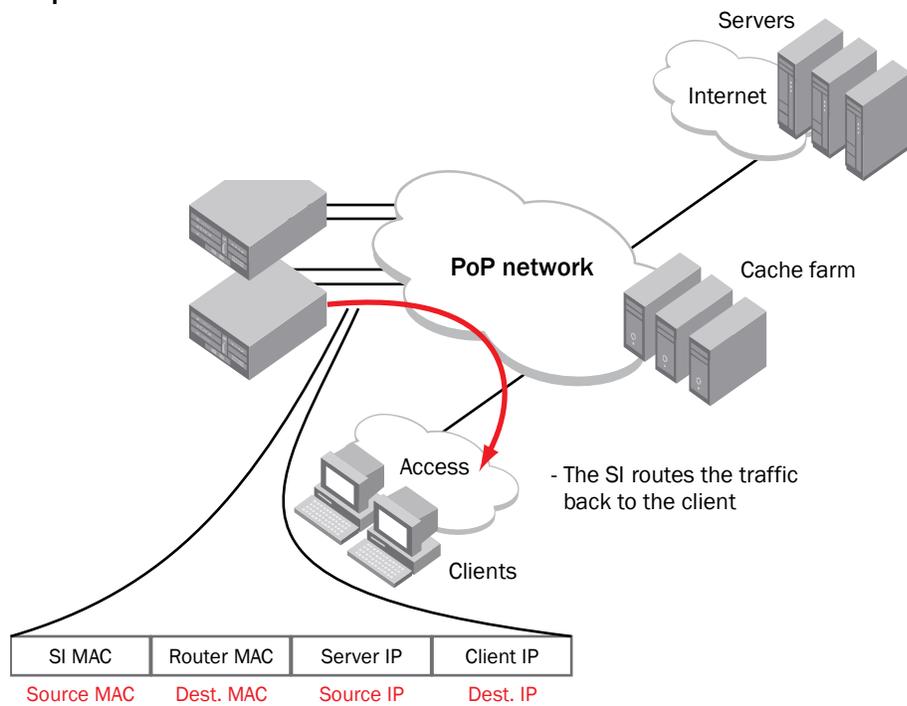
Step 6



Step 7



Step 8



APPENDIX: SERVERIRON COMMAND LINE CONFIGURATION

There are two cache-groups in this configuration:

- One for handling traffic that is totally proxied by the ProxySG (that is, the source IP of the packet going to the Internet belongs to the ProxySG)
- One for handling situations in which the client IP address is preserved as requests are forwarded to the internet (client spoofing)

ProxySG devices SG1 and SG8 were set aside for handling client spoofing and subsequently asymmetrically routed traffic. This technique of clustering ProxySG devices across sites is also known as “IP reflection.” Cache-group 2 forwards traffic to SG1, although an HA active/standby pair of Serverlrons is also supported by this solution. Access list 101 is used to filter traffic that should go to cache-group 1, and access list 102 is reserved for cache-group 2.

ServerIron 1

```

ver 10.2.01eTG4
!
module 1 bi-0-port-wsm7-management-module
module 2 bi-jc-16-port-gig-copper-module
module 3 bi-jc-16-port-gig-copper-module
!
global-stp
global-protocol-vlan
!
trunk server ethe 2/3 to 2/4
  port-name "To_ISG1" ethernet 2/3
trunk server ethe 3/3 to 3/4
  port-name "To_ISG2" ethernet 3/3
!
session sync-update
!
server active-active-port ethe 3/11 vlan-id 200
!
!
server force-cache-rehash
!
server port 80
  session-sync
  tcp
!
context default
!
server cache-name SG2 10.98.1.3
  port http
  port http url "HEAD /"
  port http l4-check-only
  port ssl
  port ssl l4-check-only
!
server cache-name SG3 10.98.1.4
  port http
  port http url "HEAD /"

```

```
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG4 10.98.1.5
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG5 10.98.1.6
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG6 10.98.1.7
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG7 10.98.1.8
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG9 10.98.1.10
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG10 10.98.1.11
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG11 10.98.1.12
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
```

```
server cache-name SG12 10.98.1.13
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG13 10.98.1.14
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG14 10.98.1.15
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG-VIP 10.98.1.100
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG1 10.98.1.2
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG8 10.98.1.9
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-group 1
hash-mask 255.255.255.255 0.0.0.255
filter-acl 101
cache-name SG2
cache-name SG3
cache-name SG4
cache-name SG5
cache-name SG6
cache-name SG7
cache-name SG9
cache-name SG10
```

```
cache-name SG11
cache-name SG12
cache-name SG13
cache-name SG14

server cache-group 2
hash-mask 255.255.255.255 0.0.0.255
filter-acl 102
cache-name SG1
spooof-support

vlan 1 name DEFAULT-VLAN by port
!
vlan 10 by port
untagged ethe 2/5 to 2/6 ethe 3/5 to 3/6
router-interface ve 10
spanning-tree 802-1w
spanning-tree 802-1w priority 7000
!
vlan 100 by port
untagged ethe 2/3 to 2/4 ethe 3/3 to 3/4
router-interface ve 100
spanning-tree 802-1w
spanning-tree 802-1w priority 7000
!
vlan 200 by port
untagged ethe 3/11
static-mac-address 0012.f2a7.bd4a ethernet 3/11
!
vlan 99 by port
untagged ethe 3/13
router-interface ve 99
spanning-tree 802-1w
spanning-tree 802-1w priority 7000
!
default-mtu 9000
aaa authentication web-server default local
aaa authentication enable default local
aaa authentication login default local
aaa authentication login privilege-mode
enable telnet authentication
enable aaa console
hostname SI-1
ip acl-permit-udp-1024
ip l4-policy 1 cache tcp http global
ip route 0.0.0.0 0.0.0.0 10.97.0.1
!
no telnet server
username admin password .....
router vrrp-extended
snmp-server
snmp-server community ..... ro
snmp-server host 10.9.71.90 .....
```

```
no web-management http
web-management https
!
interface ethernet 2/3
  port-name To_ISG1
!
interface ethernet 2/5
  port-name To_LS-top
  link-aggregate configure key 10500
  link-aggregate active
!
interface ethernet 2/6
  link-aggregate configure key 10500
  link-aggregate active
!
interface ethernet 3/1
  port-name Mgmt
  ip address 10.9.71.240 255.255.255.0
!
interface ethernet 3/3
  port-name To_ISG2
!
interface ethernet 3/5
  port-name To_LS-bottom
  link-aggregate configure key 11500
  link-aggregate active
!
interface ethernet 3/6
  link-aggregate configure key 11500
  link-aggregate active
!
interface ethernet 3/13
  disable
!
interface ethernet 3/14
  disable
!
interface ethernet 3/15
  disable
!
interface ethernet 3/16
  disable
!
interface ve 10
  port-name To_SGs
  ip address 10.98.1.254 255.255.255.0
  ip vrrp-extended vrid 2
  backup priority 150
  ip-address 10.98.1.1
  track-port e 2/3 priority 30
  track-trunk-port e 2/3
  track-port e 3/3 priority 30
  track-trunk-port e 3/3
```

```
enable
!
interface ve 99
port-name To_ISG2
ip address 10.99.0.252 255.255.255.0
ip vrrp-extended vrid 3
backup
ip-address 10.99.0.254
track-port e 2/5 priority 30
track-port e 3/5 priority 30
disable
!
interface ve 100
port-name To_ISGs
ip address 10.97.0.252 255.255.255.0
ip vrrp-extended vrid 1
backup priority 150
ip-address 10.97.0.254
track-port e 2/5 priority 30
track-port e 3/5 priority 30
enable
!
access-list 101 deny tcp 10.95.100.0 0.0.0.255 any eq http
access-list 101 deny tcp 10.98.100.0 0.0.0.255 any
access-list 101 permit tcp any any
!
access-list 102 permit tcp 10.95.100.0 0.0.0.255 any eq http
access-list 102 permit tcp 10.95.100.0 0.0.0.255 any eq ssl
!
```

ServerIron 2

```
ver 10.2.01eTG4
!
module 1 bi-0-port-wsm7-management-module
module 2 bi-jc-16-port-gig-copper-module
module 3 bi-jc-16-port-gig-copper-module
!
global-stp
global-protocol-vlan
!
trunk server ethe 2/3 to 2/4
trunk server ethe 3/3 to 3/4
port-name "To_ISG2" ethernet 3/3
!
session sync-update
!
server active-active-port ethe 3/11 vlan-id 200
!
!
server force-cache-rehash

server port 80
session-sync
```

```
tcp
!
context default
!
server cache-name SG-VIP 10.98.1.100
port http
port http url "HEAD /"
port ssl
port ssl l4-check-only
!
server cache-name SG2 10.98.1.3
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG3 10.98.1.4
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG4 10.98.1.5
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG5 10.98.1.6
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG6 10.98.1.7
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG7 10.98.1.8
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG9 10.98.1.10
```

```
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG10 10.98.1.11
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG11 10.98.1.12
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG12 10.98.1.13
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG13 10.98.1.14
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG14 10.98.1.15
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG1 10.98.1.2
port http
port http url "HEAD /"
port http l4-check-only
port ssl
port ssl l4-check-only
!
server cache-name SG8 10.98.1.9
port http
port http url "HEAD /"
port http l4-check-only
port ssl
```

```
port ssl l4-check-only
!
server cache-group 1
hash-mask 255.255.255.255 0.0.0.255
filter-acl 101
cache-name SG2
cache-name SG3
cache-name SG4
cache-name SG5
cache-name SG6
cache-name SG7
cache-name SG9
cache-name SG10
cache-name SG11
cache-name SG12
cache-name SG13
cache-name SG14

server cache-group 2
filter-acl 102
cache-name SG1
spooof-support

vlan 1 name DEFAULT-VLAN by port
!
vlan 100 by port
untagged ethe 2/3 to 2/4 ethe 3/3 to 3/4
router-interface ve 100
spanning-tree 802-1w
!
vlan 10 by port
untagged ethe 2/5 to 2/10 ethe 3/5 to 3/10
router-interface ve 10
spanning-tree 802-1w
!
vlan 200 by port
untagged ethe 3/11
static-mac-address 0012.f2a7.fa4a ethernet 3/11
!
vlan 99 by port
untagged ethe 3/13
router-interface ve 99
spanning-tree 802-1w
spanning-tree 802-1w priority 7000
!
default-mtu 9000
aaa authentication web-server default local
aaa authentication enable default local
aaa authentication login default local
aaa authentication login privilege-mode
enable telnet authentication
enable aaa console
hostname SI-2
```

```
ip acl-permit-udp-1024
ip l4-policy 1 cache tcp http global
ip route 0.0.0.0 0.0.0.0 10.97.0.1
!
no telnet server
username admin password .....
router vrrp-extended
snmp-server
snmp-server community ..... ro
snmp-server host 10.9.71.90 .....
no web-management http
web-management https
!
interface ethernet 2/5
  port-name To_LS-top
  link-aggregate configure key 10500
  link-aggregate active
!
interface ethernet 2/6
  link-aggregate configure key 10500
  link-aggregate active
!
interface ethernet 3/1
  port-name Mgmt
  ip address 10.9.71.241 255.255.255.0
!
interface ethernet 3/3
  port-name To_ISG2
!
interface ethernet 3/5
  port-name To_LS-bottom
  link-aggregate configure key 11500
  link-aggregate active
!
interface ethernet 3/6
  link-aggregate configure key 11500
  link-aggregate active
!
interface ethernet 3/13
  disable
!
interface ethernet 3/14
  disable
!
interface ethernet 3/15
  disable
!
interface ethernet 3/16
  disable
!
interface ve 10
  port-name To_SGs
  ip address 10.98.1.253 255.255.255.0
```

```
ip vrrp-extended vrid 2
  backup
  ip-address 10.98.1.1
  track-port e 2/3 priority 30
  track-trunk-port e 2/3
  track-port e 3/3 priority 30
  track-trunk-port e 3/3
  enable
!
interface ve 99
  port-name To_ISG2
  ip address 10.99.0.253 255.255.255.0
  ip vrrp-extended vrid 3
  backup priority 150
  ip-address 10.99.0.254
  track-port e 2/5 priority 30
  track-port e 3/5 priority 30
  disable
!
interface ve 100
  port-name To_ISGs
  ip address 10.97.0.253 255.255.255.0
  ip vrrp-extended vrid 1
  backup
  ip-address 10.97.0.254
  track-port e 2/5 priority 30
  track-port e 3/5 priority 30
  enable
!
access-list 101 deny tcp 10.95.100.0 0.0.0.255 any eq http
access-list 101 deny tcp 10.98.100.0 0.0.0.255 any
access-list 101 permit tcp any any
!
access-list 102 permit tcp 10.95.100.0 0.0.0.255 any eq http
access-list 102 permit tcp 10.95.100.0 0.0.0.255 any eq ssl
!
```

© 2009 Brocade Communications Systems, Inc. All Rights Reserved. 09/09 GA-SB-218-00

Brocade, the B-wing symbol, BigIron, DCX, Fabric OS, FastIron, IronPoint, IronShield, IronView, IronWare, JetCore, NetIron, SecureIron, ServerIron, StorageX, and Turbolron are registered trademarks, and DCFM, Extraordinary Networks, and SAN Health are trademarks of Brocade Communications Systems, Inc., in the United States and/or in other countries. All other brands, products, or service names are or may be trademarks or service marks of, and are used to identify, products or services of their respective owners.

Notice: This document is for informational purposes only and does not set forth any warranty, expressed or implied, concerning any equipment, equipment feature, or service offered or to be offered by Brocade. Brocade reserves the right to make changes to this document at any time, without notice, and assumes no responsibility for its use. This informational document describes features that may not be currently available. Contact a Brocade sales office for information on feature and product availability. Export of technical data contained in this document may require an export license from the United States government.