This article provides examples of server load balancing configurations. For details about configuring server load balancing on the ACE, see the *Cisco Application Control Engine Module Server Load-Balancing Configuration Guide*.

To return to the main article, click here.

Contents

Example of a UDP Probe Load-Balancing Configuration

The following example shows a running configuration that load balances DNS traffic across multiple real servers, and transmits and receives UDP data that spans multiple packets. The configuration uses a UDP health probe. The UDP probe configuration eements appears in bold in the example.

```
access-list ACL1 line 10 extended permit ip any any

probe udp UDP
  interval 5
  passdetect interval 10
  description THIS PROBE IS INTENDED FOR LOAD BALANCING DNS TRAFFIC
  port 53
  send-data UDP_TEST

rserver host SERVER1
  ip address 192.168.252.245
  inservice
rserver host SERVER2
  ip address 192.168.252.246
  inservice
rserver host SERVER3
  ip address 192.168.252.247
  inservice

serverfarm host SFARM1
  probe UDP
  rserver SERVER1
    inservice
  rserver SERVER2
    inservice
  rserver SERVER3
    inservice

class-map match-all L4UDP-VIP_114:UDP_CLASS
   2 match virtual-address 192.168.120.114 udp eq 53
policy-map type loadbalance first-match L7PLBSF_UDP_POLICY
  class class-default
    serverfarm SFARM1
policy-map multi-match L4SH-Gold-VIPs_POLICY
    class L4UDP-VIP_114:UDP_CLASS
    loadbalance vip inservice
    loadbalance policy L7PLBSF_UDP_POLICY
    loadbalance vip icmp-reply
    nat dynamic 1 vlan 120
    connection advanced-options 1SECOND-IDLE
interface vlan 120
  description Upstream VLAN_120 - Clients and VIPs
  ip address 192.168.120.1 255.255.255.0
  fragment chain 20
  fragment min-mtu 68
  access-group input ACL1
  nat-pool 1 192.168.120.70 192.168.120.70 netmask 255.255.255.0 pat
  service-policy input L4SH-Gold-VIPs_POLICY
  no shutdown

ip route 10.1.0.0 255.255.255.0 192.168.120.254
```

Examples of Real Server Configurations

The following examples illustrate a running configuration that creates multiple real servers. These configurations show how to specify a host as the real server and how to specify a real server to redirect traffic to a different location. The real server configuration elements appear in bold in the following two examples:

- Real Server that Hosts Content
- Real Server that Redirects Client Requests

Real Server that Hosts Content

The following example creates three real servers, places them in service, and adds each real server to a server farm:

```
access-list ACL1 line 10 extended permit ip any any

rserver host SERVER1
  ip address 192.168.252.245
  inservice
rserver host SERVER2
  ip address 192.168.252.246
  inservice
rserver host SERVER3
  ip address 192.168.252.247
  inservice
serverfarm host SFARM1
  probe HTTP_PROBE
  predictor roundrobin
  rserver SERVER1
     weight 10
     inservice
  rserver SERVER2
     weight 20
     inservice
  rserver SERVER3
     weight 30
     inservice
```

Real Server that Redirects Client Requests

The following example specifies a series of real servers that redirect all incoming connections that match URLs ending in /redirect-100k.html, /redirect-10k.html, and /redirect-1k.html to the appropriate server farm:

```
access-list ACL1 line 10 extended permit ip any any

rserver redirect SERVER1
  webhost-redirection http://192.168.120.132/redirect-100k.html 301
  inservice
rserver redirect SERVER2
  webhost-redirection http://192.168.120.133/redirect-10k.html 301
  inservice
rserver redirect SERVER3
  webhost-redirection http://192.168.120.134/redirect-1k.html 301
  inservice
serverfarm redirect SFARM1
    rserver SERVER1
    inservice
serverfarm redirect SFARM2
    rserver SERVER2
    inservice
serverfarm redirect SFARM3
    rserver SERVER3
    inservice
```

Example of a Server Load-Balancing Policy Configuration

The following example shows a running configuration that includes multiple class maps and policy maps that define a traffic policy for SLB. The class map, policy map, and parameter map configuration elements appear

in bold in the example. In this configuration, when a server farm is chosen for a connection, the connection is sent to a real server based on one of several load-balancing predictors. The leastconns predictor method load balances connections to the server that has the lowest number of open connections.

```
access-list ACL1 line 10 extended permit ip any any

probe tcp TCP
  interval 5
  faildetect 2
  passdetect interval 10
  open 3

rserver host SERVER1
  ip address 172.168.12.10
    inservice
rserver host SERVER2
  ip address 192.168.12.11
    inservice
rserver host SERVER3
  ip address 192.168.12.12
    inservice
rserver host SERVER4
  ip address 192.168.12.13
    inservice
rserver host SERVER5
  ip address 192.168.12.14
    inservice
rserver host SERVER6
  ip address 192.168.12.15
    inservice
rserver host SERVER7
  ip address 192.168.12.16
    inservice
rserver host SERVER8
  ip address 192.168.12.17
    inservice

serverfarm host PRED-CONNS
  predictor leastconns
  rserver SERVER1
    inservice
  rserver SERVER2
    inservice
  rserver SERVER3
    inservice
  rserver SERVER4
    inservice
  rserver SERVER5
    inservice
  rserver SERVER6
    inservice
  rserver SERVER7
    inservice
  rserver SERVER8
    inservice
serverfarm host PRED-CONNS-UDP
  failaction purge
  predictor leastconns
  rserver SERVER1
    inservice
  rserver SERVER2
    inservice
  rserver SERVER3
```

```
    probe ICMP
    inservice
  rserver SERVER5
    inservice
  rserver SERVER6
    inservice
  rserver SERVER7
    inservice
serverfarm host PREDICTOR
  probe TCP
  rserver SERVER1
    inservice
  rserver SERVER2
    inservice
  rserver SERVER6
    inservice
  rserver SERVER7
    inservice

parameter-map type http PERSIST-REBALANCE
  persistence-rebalance
parameter-map type connection PRED-CONNS-UDP_CONN
  set timeout inactivity 300

sticky http-cookie COOKIE_TEST STKY-GRP-43
  cookie offset 1 length 999
  timeout 30
  replicate sticky
  serverfarm PREDICTOR

class-map match-all L4PRED-CONNS-UDP-VIP_128:2222_CLASS
  2 match virtual-address 192.168.120.128 udp eq 0
class-map match-all L4PRED-CONNS-VIP_128:80_CLASS
  2 match virtual-address 192.168.120.128 tcp eq www
class-map match-all L4PREDICTOR_117:80_CLASS
  2 match virtual-address 192.168.120.117 tcp eq www
policy-map type loadbalance first-match L7PLBSF_PRED-CONNS_POLICY
  class class-default
    serverfarm PRED-CONNS
policy-map type loadbalance first-match L7PLBSF_PRED-CONNS-UDP_POLICY
  class class-default
    serverfarm PRED-CONNS-UDP
policy-map type loadbalance first-match L7PLBSF_PREDICTOR_POLICY
  class class-default
    sticky-serverfarm STKY-GRP-43
policy-map multi-match L4SH-Gold-VIPs_POLICY
  class L4PREDICTOR_117:80_CLASS
    loadbalance vip inservice
    loadbalance policy L7PLBSF_PREDICTOR_POLICY
    loadbalance vip icmp-reply active
    nat dynamic 1 vlan 120
    appl-parameter http advanced-options PERSIST-REBALANCE
  class L4PRED-CONNS-VIP_128:80_CLASS
    loadbalance vip inservice
    loadbalance policy L7PLBSF_PRED-CONNS_POLICY
    loadbalance vip icmp-reply active
    nat dynamic 1 vlan 120
    appl-parameter http advanced-options PERSIST-REBALANCE
  class L4PRED-CONNS-UDP-VIP_128:2222_CLASS
    loadbalance vip inservice
    loadbalance policy L7PLBSF_PRED-CONNS-UDP_POLICY
    loadbalance vip icmp-reply active
    nat dynamic 1 vlan 120
    appl-parameter http advanced-options PERSIST-REBALANCE
```

```
    connection advanced-options PRED-CONNS-UDP_CONN

interface vlan 120
  description Upstream VLAN_120 - Clients and VIPs
  ip address 192.168.120.1 255.255.255.0
  fragment chain 20
  fragment min-mtu 68
  access-group input ACL1
  nat-pool 1 192.168.120.70 192.168.120.70 netmask 255.255.255.0 pat
  service-policy input L4SH-Gold-VIPs_POLICY
  no shutdown

ip route 10.1.0.0 255.255.255.0 192.168.120.254
```

Example of an RDP Load-Balancing Configuration

The following example provides a running configuration for RDP load balancing. The RDP-specific load balancing configuration elements appear in bold text.

```
access-list ACL1 line 10 extended permit ip any any

rserver host RS1
  ip address 10.6.252.245
  inservice
rserver host RS2
  ip address 10.6.252.246
  inservice

serverfarm host SF1
  rserver RS1
    inservice
  rserver RS2
    inservice

class-map match-any RDP_L4_CLASS
  2 match virtual-address 10.6.252.19 tcp eq rdp

policy-map type loadbalance rdp first-match RDP_L7_POLICY
  class class-default
    serverfarm SF1

policy-map multi-match RDP_L4_POLICY
  class RDP_L4_CLASS
    loadbalance vip inservice
    loadbalance RDP_L7_POLICY

interface vlan 10
  ip address 10.6.252.12 255.255.255.0
  access-group input ACL1
  service-policy input RDP_L4_POLICY
  no shutdown

ip route 0.0.0.0 0.0.0.0 10.6.252.0
```

Examples of RADIUS Load-Balancing Configurations

The following configuration examples provide RADIUS load-balancing configurations with and without a Layer 7 RADIUS class map. The RADIUS load-balancing configuration elments are shown in bold text.

- Without a Layer 7 RADIUS Class Map

Without a Layer 7 RADIUS Class Map

The following configuration example shows the running configuration for RADIUS load-balancing without a Layer 7 RADIUS class map:

```
access-list ACL1 line 10 extended permit ip any any

rserver host RS1
  ip address 10.6.252.245
  inservice
rserver host RS2
  ip address 10.6.252.246
  inservice

serverfarm host SF1
  rserver RS1
    inservice
  rserver RS2
    inservice

sticky radius framed-ip calling-station-id RADIUS_GROUP
  serverfarm SF1
class-map match-any RADIUS_L4_CLASS
  2 match virtual-address 12.1.1.11 udp range 1812 1813

policy-map type loadbalance radius first-match RADIUS_L7_POLICY
  class class-default
    sticky-serverfarm RADIUS_GROUP

policy-map multi-match RADIUS_L4_POLICY
  class RADIUS_L4_CLASS
    loadbalance vip inservice
    loadbalance RADIUS_L7_POLICY

interface vlan 10
  ip address 192.168.12.13 255.255.255.0
  service-policy input RADIUS_DATA_L4_POLICY
  no shutdown

ip route 0.0.0.0 0.0.0.0 10.6.252.1
```

With a Layer 7 RADIUS Class Map

The following configuration example shows the running configuration for RADIUS load-balancing with a Layer 7 RADIUS class map. For Layer 7 Radius load balancing, an ACL is required to deny radius traffic that is initiated from the rservers. This protects against a problem where late responses are interpreted as server-initiated connections, and then impact the forwarding of subsequent Radius requests received on the same client connection. (This is not required for L4 RLB).

```
access-list ACL1 line 10 extended permit ip any any
access-list deny_srvr_init_radius line 10 extended deny udp any range 1812 1813 any
access-list deny_srvr_init_radius line 20 extended permit ip any any

 rserver host RS1
  ip address 10.6.252.245
  inservice
```

```
rserver host RS2
  ip address 10.6.252.246
  inservice
rserver host RS3
  ip address 10.6.252.247
  inservice
rserver host RS4
  ip address 10.6.252.248
  inservice

serverfarm host SF1
  rserver RS1
    inservice
  rserver RS2
    inservice

serverfarm host SF2
  rserver RS3
    inservice
  rserver RS4
    inservice

sticky radius framed-ip calling-station-id RADIUS_GROUP1
  serverfarm SF1

sticky radius framed-ip calling-station-id RADIUS_GROUP2
  serverfarm SF2

class-map match-any RADIUS_L4_CLASS
  2 match virtual-address 192.168.12.15 udp range 1812 1813

class-map type radius loadbalance match-any RADIUS_L7_CLASS1
   2 match radius attribute calling-station-id 122*
   3 match radius attribute calling-station-id 133*

class-map type radius loadbalance match-any RADIUS_L7_CLASS2
   2 match radius attribute calling-station-id 144*

policy-map type loadbalance radius first-match RADIUS_L7_POLICY
  class RADIUS_L7_CLASS1
    sticky-serverfarm RADIUS_GROUP1
  class RADIUS_L7_CLASS2
    sticky-serverfarm RADIUS_GROUP2

policy-map multi-match RADIUS_L4_POLICY
  class RADIUS_L4_CLASS
    loadbalance vip inservice
    loadbalance RADIUS_L7_POLICY

interface vlan 10
  ip address 192.168.12.12 255.255.255.0
  service-policy input RADIUS_L4_POLICY
  no shutdown

 interface vlan 20
   description server-side
   ip address 10.6.252.250 255.255.255.0
   access-group input deny_srvr_init_radius
   nat-pool 1 10.6.252.10 10.6.252.17 netmask 255.255.255.0 pat
   no shutdown

ip route 0.0.0.0 0.0.0.0 10.6.252.1
```

End User Data Forwarding Policy

The following configuration example provides the commands necessary to instruct the ACE to forward non-RADIUS data packets to the same RADIUS server where the ACE sent the first RADIUS packet.

```
access-list ACL1 line 10 extended permit ip any any

rserver host RS1
  ip address 10.6.252.245
  inservice
rserver host RS2
  ip address 10.6.252.246
  inservice

serverfarm host SF1
  rserver RS1
    inservice
  rserver RS2
    inservice

sticky radius framed-ip RADIUS_GROUP1
  serverfarm SF1

class-map type radius loadbalance match-any RADIUS_L7_CLASS
  2 match radius attribute calling-station-id 133*

class-map match-any RADIUS_L4_CLASS
  3 match virtual-address 192.168.12.15 udp range 1812 1813

class-map match-any LAYER4_CLASS
  4 match virtual-address 0.0.0.0 0.0.0.0 any

policy-map type loadbalance radius first-match RADIUS_L7_POLICY
  class RADIUS_L7_CLASS
    sticky-serverfarm RADIUS_GROUP1

policy-map type loadbalance first-match DATA_FORWARD_L7POLICY
  class class-default
    sticky-serverfarm RADIUS_GROUP1

policy-map multi-match RADIUS_L4_POLICY
  class RADIUS_L4_CLASS
    loadbalance vip inservice
    loadbalance policy RADIUS_L7_POLICY
    loadbalance vip icmp-reply
    loadbalance vip advertise
  class LAYER4_CLASS
    loadbalance vip inservice
    loadbalance policy LAYER7_POLICY

interface vlan 10
  ip address 192.168.12.12 255.255.255.0
  service-policy input RADIUS_L4_POLICY
  no shutdown

ip route 0.0.0.0 0.0.0.0 10.6.252.1
```

Example of an RTSP Load-Balancing Configuration

The following is a sample RTSP configuration. The sticky group and RTSP inspection portions are optional. If a client uses different connections for multiple requests in one session, you must configure the sticky

group. If you use RTP for data traffic that runs on separate connections, an inspection is needed to open the proper pinholes. The RTSP-specific configuration elements appear in bold in the example.

```
access-list ACL1 line 10 extended permit ip any any

rserver host RS1
  ip address 10.6.252.245
    inservice
rserver host RS2
  ip address 10.6.252.246
    inservice

serverfarm host SF4
  rserver RS1
    inservice
  rserver RS2
    inservice

sticky rtsp-header Session RTSP_GROUP
  serverfarm SF4

class-map type rtsp loadbalance match-any RTSP_L7_CLASS
  match rtsp url rtsp://cisco.com/movie
  match rtsp header Accept header-value application/sdp

class-map match-all RTSP_L4_CLASS
  match virtual-address 192.168.12.15 tcp eq rtsp

policy-map type loadbalance rtsp first-match RTSP_L7_POLICY
  class RTSP_L7_CLASS
    sticky-serverfarm RTSP_GROUP

policy-map multi-match RTSP_L4_POLICY
  class RTSP_L4_CLASS
    loadbalance vip inservice
    loadbalance policy RTSP_L7_POLICY
    inspect rtsp

interface vlan 10
  ip address 192.168.12.12 255.255.255.0
  service-policy input RTSP_L4_POLICY
  no shutdown

ip route 0.0.0.0 0.0.0.0 10.6.252.1
```

Examples of SIP Load-Balancing Configurations

The following examples are from a sample SIP configuration:

- SIP Load Balancing Without Match Criteria
- SIP Load Balancing Based on SIP headers and SIP Inspection

The sticky group and SIP inspection are optional. If you do not configure class-map match criteria, the ACE performs the action specified under the class class-default command. The class-default class map contains an implicit match any statement that enables it to match all traffic. Additionally, the ACE ensures that messages with the same Call-ID are load balanced to the same server. The SIP load-balancing configuration elements appear in bold in the example.

SIP Load Balancing Without Match Criteria

The following configuration example shows the running configuration for SIP load-balancing with match criteria. The parts of the configuration that pertain strictly to the SIP load-balancing configuration are shown in bold text.

```
access-list ACL1 line 10 extended permit ip any any

rserver host RS1
  ip address 10.6.252.245
  inservice
rserver host RS2
  ip address 10.6.252.246
  inservice

serverfarm host SF1
  rserver RS1
    inservice
  rserver RS2
    inservice

sticky sip-header Call-ID SIP_GROUP
  serverfarm SF3

class-map match-all SIP_L4_CLASS
  match virtual-address 192.168.12.15 udp eq sip

policy-map type loadbalance sip first-match SIP_L7_POLICY
  class class-default
    sticky-serverfarm SIP_GROUP
policy-map multi-match SIP_L4_POLICY
  class SIP_L4_CLASS
    loadbalance vip inservice
    loadbalance policy SIP_L7_POLICY
    inspect sip

interface vlan 10
  ip address 192.168.12.12 255.255.255.0
  service-policy input SIP_L4_POLICY
  no shutdown

ip route 0.0.0.0 0.0.0.0 10.6.252.1
```

SIP Load Balancing Based on SIP headers and SIP Inspection

The following configuration example shows the running configuration for SIP load-balancing on SIP headers and SIP inspection.

```
access-list ACL1 line 10 extended permit ip any any

rserver host RS1
  ip address 10.6.252.245
  inservice
rserver host RS2
  ip address 10.6.252.246
  inservice

serverfarm host SF3
  rserver RS1
    inservice
  rserver RS2
```

```
    inservice

sticky sip-header Call-ID SIP_GROUP
  serverfarm SF3

class-map type sip loadbalance match-any SIP_L7_CLASS
  match sip header Call_ID header-value sip:

class-map match-all SIP_L4_CLASS
  match virtual-address 192.168.12.15 tcp eq sip

policy-map type loadbalance sip first-match SIP_L7_POLICY
  class SIP_L7_CLASS
    sticky-serverfarm SIP_GROUP

policy-map multi-match SIP_L4_POLICY
  class SIP_L4_CLASS
    loadbalance vip inservice
    loadbalance policy SIP_L7_POLICY
    inspect sip

interface vlan 10
  ip address 192.168.12.12 255.255.255.0
  service-policy input SIP_L4_POLICY
  no shutdown

ip route 0.0.0.0 0.0.0.0 10.6.252.1
```

Examples of Sticky Configurations

The following examples show sample configurations for HTTP-header stickiness and SSL Session ID stickiness.

Example of an HTTP-Header Sticky Configuration

This following example provides a sample HTTP-header sticky configuration. The HTTP-header sticky configuration elements appear in bold in the example.

```
resource-class RC1
  limit-resource all minimum 0.00 maximum unlimited
  limit-resource sticky minimum 10.00 maximum unlimited

context Admin
  member RC1

access-list ACL1 extended permit tcp any any eq http

rserver SERVER1
  address 192.168.12.15
  probe PROBE1
  inservice
rserver SERVER2
  address 192.168.12.16
  probe PROBE2
  inservice

serverfarm SFARM1
  rserver SERVER1
    inservice
  rserver SERVER2
    inservice
```

```
sticky http-header Host GROUP4
  serverfarm SFARM1
  timeout 720
  timeout activeconns
  replicate sticky
  header offset 3000 length 1000
  static header Host rserver SERVER1 4000

class-map match-any L4CLASS
  10 match virtual-address 192.168.12.15 netmask 255.255.255.0 tcp port eq 80

class-map type http loadbalance match-any L7CLASS
  10 match http header Host header-value .*cisco.com

policy-map multi-match L4POLICY
  sequence interval 10
  class L4CLASS
    loadbalance policy L7POLICY

policy-map type loadbalance first-match L7POLICY
  class L7CLASS
    sticky-serverfarm GROUP4
    insert-http Host header-value *.cisco.com

interface vlan 193
  ip address 192.168.3.13 255.255.255.0
  service-policy input L4POLICY
  no shutdown

context Admin
  member RC1

ip route 0.0.0.0 0.0.0.0 192.168.12.1
```

Example of an SSL Session ID Sticky Configuration

```
resource-class RC1
  limit-resource sticky minimum 10.00 maximum equal-to-min

access-list ACL1 line 10 extended permit ip any any

parameter-map type generic SSLID_PARAMMAP
  set max-parse-length 70

rserver SSL_SERVER1
  ip address 192.168.12.2
  inservice
rserver SSL_SERVER2
  ip address 192.168.12.3
  inservice

serverfarm SSL_SFARM1
  rserver SSL_SERVER1
    inservice
  rserver SSL_SERVER2
    inservice

sticky layer4-payload SSL_GROUP
  timeout 600
  serverfarm SSL_SFARM1
  response sticky
  layer4-payload offset 43 length 32 begin-pattern "\x20|\x00\xST)"
```

```
class-map match-any L4_CLASS
  match virtual-address 172.27.16.2 tcp eq any

policy-map type loadbalance generic first-match SSLID_32_POLICY
  class class-default
  sticky-serverfarm SSL_GROUP

policy-map multi-match L4_POLICY
  class L4_CLASS
    loadbalance vip advertise active
    loadbalance vip inservice
    loadbalance vip icmp-reply active
    loadbalance policy SSLID_32_POLICY
    appl-parameter generic advanced-options SSLID-PARAMMAP

interface vlan 200
  ip address 172.27.16.1 255.255.255.0
  access-group input ACL1
  service-policy input L4_POLICY

interface vlan 300
  ip address 192.168.12.1 255.255.255.0

context Admin
  member RC1

ip route 0.0.0.0 0.0.0.0 192.168.12.1
```

{Note} For SSL Session-ID stickiness for different lengths of Session IDs, you can configure as many class maps as necessary.

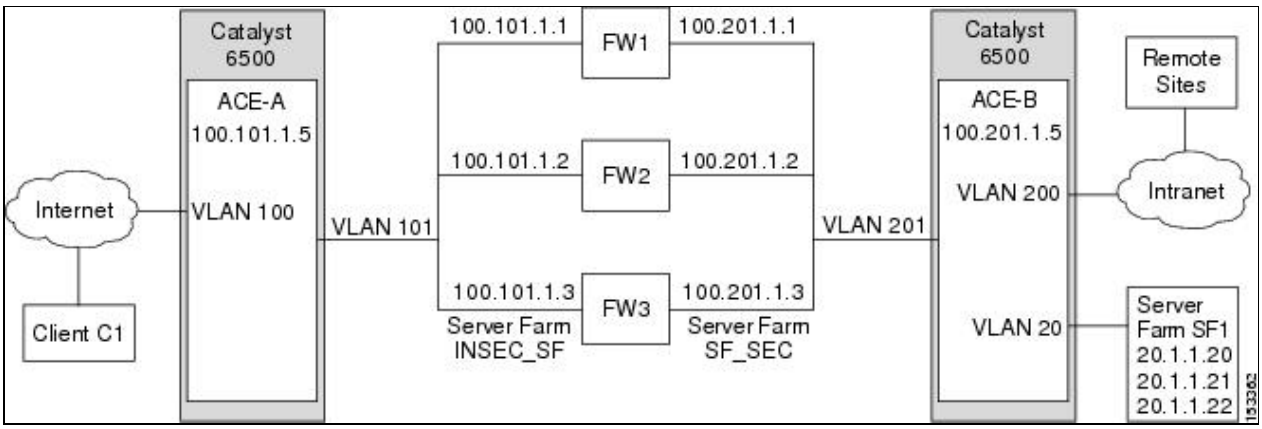
Examples of Firewall Load-Balancing Configurations

This section provides examples of both standard and stealth FWLB configurations. It contains the following topics:

Example of a Standard Firewall Load-Balancing Configuration

The following example shows those portions of the running configuration that pertain to standard FWLB. The configuration is based on two ACE modules each in a separate Catalyst 6500 series switch with the firewalls situated between them (see Figure 1). You can also configure standard FWLB using a single ACE.

Figure 1. Standard Firewall Load Balancing



This section contains the following configuration examples:

- ACE A Configuration?Standard Firewall Load Balancing
- ACE B Configuration?Standard Firewall Load Balancing

ACE A Configuration?Standard Firewall Load Balancing

```
access-list ACL1 line 10 extended permit ip any any

rserver host FW_INSEC_1
  ip address 100.101.1.1
  inservice
rserver host FW_INSEC_2
  ip address 100.101.1.2
  inservice
rserver host FW_INSEC_3
  ip address 100.101.1.3
  inservice

serverfarm INSEC_SF
  transparent
  predictor hash address source 255.255.255.255
  rserver FW_INSEC_1
    inservice
  rserver FW_INSEC_2
    inservice
  rserver FW_INSEC_3
    inservice

class-map match-any FW_VIP
  10 match virtual-address 200.1.1.1 255.255.0.0 any
policy-map type loadbalance first-match LB_FW_INSEC
  class class-default
    serverfarm INSEC_SF
policy-map multi-match POL_INSEC
  class FW_VIP
    loadbalance vip inservice
    loadbalance policy LB_FW_INSEC

interface vlan 100
  ip addr 100.100.1.100 255.255.0.0
  access-group input ACL1
  service-policy input POL_INSEC
  no shutdown
interface vlan 101
  ip addr 100.101.1.101 255.255.0.0
```

```
  access-group input ACL1
  mac-sticky enable
  service-policy input POL_INSEC
  no shutdown
```

ACE B Configuration?Standard Firewall Load Balancing

```
access-list ACL1 line 10 extended permit ip any any

rserver FW_SEC_1
  ip address 100.201.1.1
  inservice
rserver FW_SEC_2
  ip address 100.201.1.2
  inservice
rserver FW_SEC_3
  ip address  100.201.1.3
  inservice

rserver REAL1
  ip address 20.1.1.1
  inservice
rserver REAL2
  ip address 20.1.1.2
  inservice
rserver REAL3
  ip address 20.1.1.3
  inservice

serverfarm SEC_SF
  predictor hash address destination 255.255.255.255
  transparent
  rserver FW_SEC_1
    inservice
  rserver FW_SEC_2
    inservice
  rserver FW_SEC_3
    inservice

serverfarm SEC_20_SF
  rserver REAL1
    inservice
  rserver REAL2
    inservice
  rserver REAL3
    inservice

class-map match-any SEC_20_VS
  10 match virtual-address 200.1.1.1 255.255.0.0 any
class-map match any FW_SEC_VIP
  10 match virtual-address 0.0.0.0 0.0.0.0 any

policy-map type loadbalance first-match SEC_20_LB
  class class-default
    serverfarm SEC_20_SF
policy-map multi-match POL_SEC_20
  class SEC_20_VS
    loadbalance vip inservice
    loadbalance policy SEC_20_LB

policy-map type loadbalance first-match LB_FW_SEC
  class class-default
    serverfarm SEC_SF
```

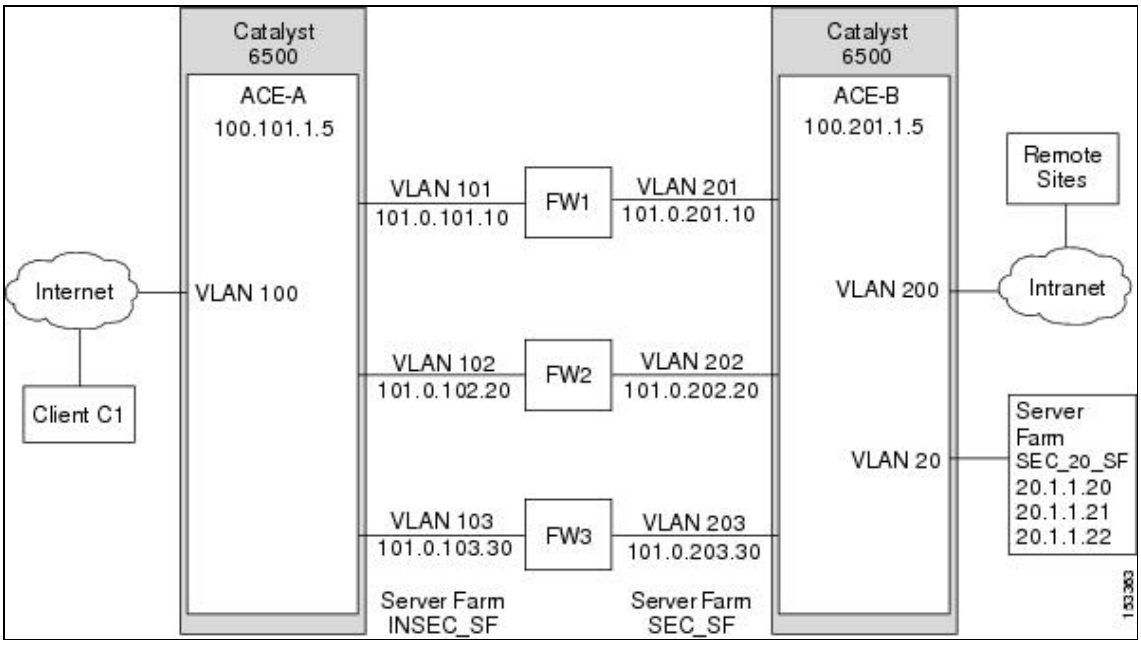
```
policy-map multi-match POL_SEC
  class FW_SEC_VIP
    loadbalance vip inservice
    loadbalance policy LB_FW_SEC

interface vlan 201
  ip address 100.201.1.201 255.255.0.0
  access-group input ACL1
  mac-sticky enable
  service-policy input POL_SEC_20
  no shutdown
interface vlan 20
  ip address 20.1.1.20 255.255.255.0
  access-group input ACL1
  service-policy input POL_SEC
  no shutdown
interface vlan 200
  ip address 200.1.1.200 255.255.255.0
  access-group input ACL1
  service-policy input POL_SEC
  no shutdown
```

Example of a Stealth Firewall Configuration

The following example shows those portions of the running configuration that pertain to stealth FWLB. This configuration requires two ACE modules each residing in a different Catalyst 6500 series switch. See Figure 2.

Figure 2. Stealth Firewall Load Balancing



This section contains the following configuration examples:

- ACE A Configuration?Stealth Firewall Load Balancing
- ACE B Configuration?Stealth Firewall Load Balancing

ACE A Configuration?Stealth Firewall Load Balancing

```
access-list ACL1 line 10 extended permit ip any any

rserver FW_INSEC_1
  ip address 101.0.201.100
  inservice
rserver FW_INSEC_2
  ip address 101.0.202.100
  inservice
rserver FW_INSEC_3
  ip address 101.0.203.100
  inservice

serverfarm INSEC_SF
  transparent
  predictor hash address source 255.255.255.255
  rserver FW_INSEC_1
    inservice
  rserver FW_INSEC_2
    inservice
  rserver FW_INSEC_3
    inservice

class-map match-any FORWARD-VIP
  10 match virtual-address 0.0.0.0 0.0.0.0 any
class-map match-any FW_VIP
  10 match virtual-address 200.1.1.1 255.255.0.0 any
policy-map type loadbalance first-match FORWARD_FW_INSEC
  class class-default
    forward
policy-map type loadbalance first-match LB_FW_INSEC
  class class-default
    serverfarm INSEC_SF
policy-map multi-match FORWARD_INSEC
  class FORWARD_VIP
    loadbalance vip inservice
    loadbalance policy FORWARD_FW_INSEC
policy-map multi-match POL_INSEC
  class FW_VIP
    loadbalance vip inservice
    loadbalance policy LB_FW_INSEC

interface vlan 100
  ip address 100.100.1.10 255.255.0.0
  access-group input ACL1
  service-policy input POL_INSEC
  no shutdown
interface vlan 101
  ip address 101.0.101.10 255.255.255.0
  alias 101.0.101.100 255.255.255.0
  access-group input ACL1
  service-policy input FORWARD_INSEC
  no shutdown
interface vlan 102
  ip address 101.0.102.20 255.255.255.0
  alias 101.0.102.100 255.255.255.0
  access-group input ACL1
  service-policy input FORWARD_INSEC
  no shutdown
interface vlan 103
  ip address 101.0.103.30 255.255.0.0
  alias 101.0.103.100 255.255.255.0
```

```
  access-group input ACL1
  service-policy input FORWARD_INSEC
  no shutdown
```

ACE B Configuration?Stealth Firewall Load Balancing

```
access-list ACL1 line 10 extended permit ip any any

rserver host REAL1
  ip address 20.1.1.1
  inservice
rserver host REAL2
  ip address 20.1.1.2
  inservice
rserver host REAL3
  ip address 20.1.1.3
  inservice

rserver host FW_SEC_1
  ip address 101.0.101.100
  inservice
rserver host FW_SEC_2
  ip address 101.0.102.100
  inservice
rserver host FW_SEC_3
  ip address 101.0.103.100
  inservice

serverfarm SEC_20_SF
  rserver REAL1
    inservice
  rserver REAL2
    inservice
  rserver REAL3
    inservice
serverfarm SEC_SF
  transparent
  predictor hash address destination 255.255.255.255
  rserver FW_SEC_1
    inservice
  rserver FW_SEC_2
    inservice
  rserver FW_SEC_3
    inservice

class-map match-any SEC_20_VS
  10 match virtual-address 200.1.1.1 255.255.0.0 any
class-map match-any FW_SEC_VIP
  10 match virtual-address 0.0.0.0 0.0.0.0 any

policy-map type loadbalance first-match SEC_20_LB
  class class-default
    serverfarm SEC_20_SF
policy-map type loadbalance first-match LB_FW_SEC
  class class-default
    serverfarm SEC_SF
policy-map multi-match POL_SEC_20
  class SEC_20_VS
    loadbalance vip inservice
    loadbalance policy SEC_20_LB
policy-map multi-match POL_SEC
  class FW_SEC_VIP
```

```
    loadbalance vip inservice
    loadbalance policy LB_FW_SEC

interface vlan 201
  ip address 101.0.201.10 255.255.255.0
  alias 101.0.201.100 255.255.255.0
  access-group input ACL1
  mac-sticky enable
  service-policy input POL_SEC_20
  no shutdown
interface vlan 202
  ip address 101.0.202.20 255.255.255.0
  alias 101.0.202.100 255.255.255.0
  access-group input ACL1
  mac-sticky enable
  service-policy input POL_SEC_20
  no shutdown
interface vlan 203
  ip address 101.0.203.30 255.255.0.0
  alias 101.0.203.100 255.255.255.0
  access-group input ACL1
  mac-sticky enable
  service-policy input POL_SEC_20
  no shutdown
interface vlan 20
  ip address 20.100.1.100 255.255.0.0
  access-group input ACL1
  service-policy input POL_SEC
  no shutdown
interface vlan 200
  ip address 200.1.1.200 255.255.255.0
  access-group input ACL1
  service-policy input POL_SEC
  no shutdown
```

Example of How to Write a Scripted Probe

This example shows how to write a script to probe an HTTP server using a health script:

```
# get the IP address of the real server from a predefined global array
# scriptprobe_env
set ip $scriptprobe_env(realIP)
set port 80
set url "GET /index.html HTTP/1.0\n\n"

# Open a socket to the server. This creates a TCP connection to the
# real server
set exit_msg "opening socket"
set sock [socket $ip $port]
fconfigure $sock -buffering none -eofchar {}

# Send the get request as defined
puts -nonewline $sock $url;

# Wait for the response from the server and read that in variable line
set exit_msg "receiving response"
set line [ read $sock ]

# Parse the response
if { [ regexp "HTTP/1.. (\[0-9\]+) " $line match status ] }

# Close the socket. Application MUST close the socket once the
```

```
# request/response is over.
# This allows other applications and tcl scripts to make
# a good use of socket resource. Health monitoring is allowed to open
# only 200 sockets simultaneously.
set exit_msg "closing socket"
close $sock

# decide the exit code to return to control module.
# If the status code is OK then script MUST do exit 30001
# to signal successful completion of a script probe.
# In this example any other status code means failure.
# User must do exit 30002 when a probe has failed.
if { $status == 200 } {
    set exit_msg "probe success"
    exit 30001
} else {
    set exit_msg "probe fail : can't find status code"
    exit 30002
```