This article describes how to diagnose and troubleshoot ACE L7 load-balancing issues.

Guide Contents
Main Article
Overview of ACE Troubleshooting
Understanding the ACE Module Architecture and Traffic Flow
Preliminary ACE Troubleshooting
Troubleshooting ACE Boot Issues
Troubleshooting with ACE Logging
Troubleshooting Connectivity
Troubleshooting ACE Appliance Ethernet Ports
Troubleshooting Remote Access
Troubleshooting Access Control Lists
Troubleshooting Network Address Translation
Troubleshooting ACE Health Monitoring
Troubleshooting Layer 4 Load Balancing
Troubleshooting Layer 7 Load Balancing
Troubleshooting Redundancy
Troubleshooting SSL
Troubleshooting Compression
Troubleshooting Performance Issues
ACE Resource Limits
Managing ACE Resources
Show Counter Reference

Contents

- 1 Overview of ACE Layer 7 Load Balancing
  - 1.1 Load-Balancing Predictors
  - 1.2 Classifying L7 Traffic for Server Load Balancing
- 2 Example of a L7 Load-Balancing Configuration
- 3 Troubleshooting Layer 7 Load Balancing

Overview of ACE Layer 7 Load Balancing

Layer 7 server load balancing (SLB) is the process that the load-balancing device uses to decide which server should fulfill a client request for an L7 service. For example, a client request may consist of a HyperText Transport Protocol (HTTP) GET for a web page or a File Transfer Protocol (FTP) GET to download a file. The job of the load balancer is to select the server that can successfully fulfill the client request and do so in the shortest amount of time without overloading either the server or the server farm as a whole.

The ACE supports the load balancing of the following protocols:

- Generic protocols
- HTTP
- Remote Authentication Dial-In User Service (RADIUS)
- Reliable Datagram Protocol (RDP)
- Real-Time Streaming Protocol (RTSP)
- Session Initiation Protocol (SIP)

Depending on the load-balancing algorithm?or *predictor*?that you configure, the ACE performs a series of checks and calculations to determine which server can best service each client request. The ACE bases server selection on several factors including the source or destination address, cookies, URLs, HTTP headers, or the server with the fewest connections with respect to load.

For detailed information about ACE load balancing, see the *Cisco Application Control Engine Module Server Load Balancing Configuration Guide.*

Load-Balancing Predictors

The ACE uses the following predictors to select the best server to fulfill a client request:

- Application response?Selects the server with the lowest average response time for the specified response-time measurement based on the current connection count and server weight (if configured).
- Hash address?Selects the server using a hash value based on either the source or destination IP address or both. Use these predictors for firewall load balancing (FWLB). For more information about FWLB, see Configuring Firewall Load Balancing in the *Cisco Application Control Engine Module Server Load-Balancing Configuration Guide (Software Version A2(1.0))*.
- Hash content?Selects the server using a hash value based on a content string in the Trusted Third Parties (TTP) packet body.
- Hash cookie?Selects the server using a hash value based on a cookie name.
- Hash header?Selects the server using a hash value based on the HTTP header name.
- Hash URL?Selects the server using a hash value based on the requested URL. You can specify a beginning pattern and an ending pattern to match in the URL. Use this predictor method to load balance cache servers.

- Least bandwidth?Selects the server that processed the least amount of network traffic based on the average bandwidth that the server used over a specified number of samples.
- Least connections?Selects the server with the fewest number of active connections based on the server weight. For the least-connections predictor, you can configure a slow-start mechanism to avoid sending a high rate of new connections to servers that you have just put into service.
- Least loaded?Selects the server with the lowest load based on information obtained from Simple Network Management Protocol (SNMP) probes. To use this predictor, you must associate an SNMP probe with it.
- Round-robin?Selects the next server in the list of real servers based on the server weight (weighted round-robin). Servers with a higher weight value receive a higher percentage of the connections. This is the default predictor.

Note: The hash predictor methods do not recognize the weight value that you configure for real servers. The ACE uses the weight that you assign to real servers only in the least-connections, application-response, and round-robin predictor methods.

Classifying L7 Traffic for Server Load Balancing

You classify inbound network traffic destined to or passing through the ACE based on a series of flow match criteria specified by a class map. Each class map defines a traffic classification, which is network traffic that is of interest to you. A policy map defines a series of actions (functions) that you want applied to a set of classified inbound or outbound traffic.

ACE traffic policies support the following server load-balancing (SLB) traffic attributes:

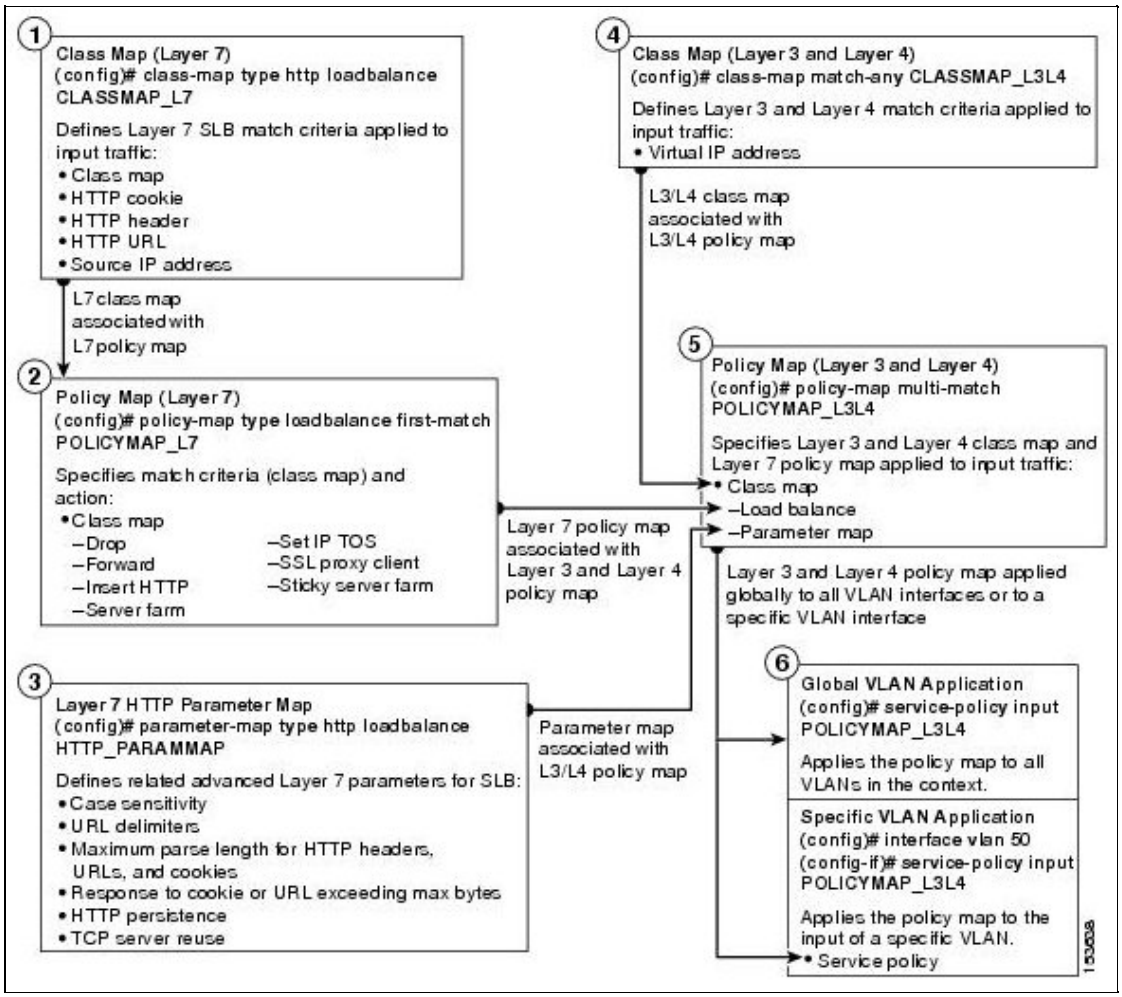
- Layer 3 and Layer 4 connection information?Source or destination IP address, source or destination port, virtual IP address, and IP protocol
- Layer 7 protocol information?Hypertext Transfer Protocol (HTTP) cookie, HTTP URL, HTTP header, Remote Authentication Dial-In User Service (RADIUS), Remote Desktop Protocol (RDP), Real-Time Streaming Protocol (RTSP), Session Initiation Protocol (SIP), and Secure Sockets Layer (SSL)

The three major steps in the traffic classification process are as follows:

1. Create a class map using the **class-map** command and the associated **match** commands, which comprise a set of match criteria related to Layer 3 and Layer 4 traffic classifications or Layer 7 protocol classifications.
2. Create a policy map using the **policy-map** command, which refers to the class maps and identifies a series of actions to perform based on the traffic match criteria.
3. Activate the policy map by associating it with a specific VLAN interface or globally with all VLAN interfaces using the **service-policy** command to filter the traffic received by the ACE.

Figure 1 provides a basic overview of the process required to build and apply the Layer 3, Layer 4, and Layer 7 policies that the ACE uses for SLB. The figure also shows how to associate the various components of the SLB policy configuration with each other.

Figure 1. SLB Flow Diagram



Example of a L7 Load-Balancing Configuration

The following example shows a running configuration that includes multiple class maps and policy maps that define a traffic policy for SLB. 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

parameter-map type http PERSIST-REBALANCE
  persistence-rebalance          <---------------------------- Enabled by default in the ACE applia
parameter-map type connection PRED-CONNS-UDP_CONN
  set timeout inactivity 300

rserver SERVER1
  ip address 10.1.0.2
  inservice
rserver SERVER2
  ip address 10.1.0.3
  inservice
rserver SERVER3
```

```
  ip address 10.1.0.4
  inservice
rserver SERVER4
  ip address 10.1.0.5
  inservice
rserver SERVER5
  ip address 10.1.0.6
  inservice
rserver SERVER6
  ip address 10.1.0.7
  inservice
rserver SERVER7
  ip address 10.1.0.8
  inservice
rserver SERVER8
  ip address 10.1.0.9
  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
```

```
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
```

Troubleshooting Layer 7 Load Balancing

To troubleshoot L7 load-balancing issues, use the following steps:

1. Ensure that your load-balancing configuration is correct and that the following conditions exist:

- ◊ Real servers have valid IP addresses and are in service
- ◊ Servers are associated with server farms of the same type
- ◊ L7 load-balancing policy exists with an associated server farm and that the L7 load-balancing policy is associated with a L4 multimatch policy
- ◊ An L4 class map contains a valid **match virtual-address** command and is associated with the L4 multimatch policy map
- ◊ The L4 policy is applied to the appropriate active interface using a service policy
- ◊ A static route is configured for the server network

Use the following **show** commands to verify your load-balancing configuration:

- ◊ **show running-config rserver**
- ◊ **show running-config serverfarm**
- ◊ **show running-config policy-map**
- ◊ **show running-config class-map**
- ◊ **show running-config interface**
- ◊ **show ip route**

2. Check the ACE connectivity. See the Troubleshooting Connectivity section.

3. Verify that the L7 load-balancing policy is referenced in the L4 policy by entering the following command. Also, check the following fields:

- ◊ VIP address, protocol, and port
- ◊ VIP state
- ◊ Hit count
- ◊ Dropped connections

```
ACE_module5/Admin# show service-policy L4WEB_POLICY detail

Status     : ACTIVE
Description: -
----------------------------------------
Interface: vlan 100
  service-policy: L4WEB_POLICY
    class: L4WEB_CLASS
     VIP Address:      Protocol:   Port:
     192.168.120.112   tcp         eq    80 <------- VIP address, protocol, and port
      loadbalance:
        L7 loadbalance policy: LB_WEB_POLICY <-------L7 load-balancing policy referenced in the L4
        VIP Route Metric     : 77
        VIP Route Advertise  : DISABLED
        VIP ICMP Reply       : ENABLED
        VIP State: INSERVICE <------- VIP state should be INSERVICE
        curr conns       : 0         , hit count        : 56
        dropped conns    : 14 <------- Number of attempted connections to this VIP that the ACE di
        client pkt count : 6297      , client byte count: 1047583
        server pkt count : 1238      , server byte count: 1325495
        L7 Loadbalance policy : LB_WEB_POLICY <------- L7 policy statistics
          class/match : class-default
            LB action :
               serverfarm: SFARM1
            hit count        : 0 <-------|-- Check these counters to see if they are increasing
            dropped conns    : 0 <-------|
```

4. Verify that the L4 policy is applied as a service policy to an active interface by entering the following command:

```
ACE_module5/Admin# show running-config interface
Generating configuration....

interface vlan 100
  ip address 192.168.120.1 255.255.255.0
  access-group input ACL1
  access-group output anyone
  service-policy input L4WEB_POLICY
  no shutdown
.
.
.
```

5. Check the total conn-dropcount field for the primary server farm in the output of the following command. Also, check the IP address, state, and the connection statistics for each real server that is configured in the server farm.

```
ACE_module5/Admin# show serverfarm SFARM1 detail

serverfarm     : SFARM1, type: HOST
total rservers : 3
active rservers: 3
description    : -
state          : ACTIVE <------- Current state of the server farm
predictor      : ROUNDROBIN <------- Load-balancing method
  weight            : -
  autoadjust        : MAXLOAD
failaction     : -
back-inservice    : 40
partial-threshold : 40
num times failover       : 0
num times back inservice : 0
total conn-dropcount : 0 <------- Total number of connection attempts to this server farm that the
--------------------------------
                                                ----------connections-----------
     real                  weight state         current    total      failures
  ---+--------------------+------+------------+----------+----------+---------
  rserver: SERVER1
      192.168.252.245:0     10     INSERVICE    0          0          0 <------- Real server IP ad
        max-conns            : 4000000   , out-of-rotation count : 0
        min-conns            : 4000000
        conn-rate-limit      : -         , out-of-rotation count : -
        bandwidth-rate-limit : -         , out-of-rotation count : -
        retcode out-of-rotation count : -
        load value           : 0

  rserver: SERVER2
      192.168.252.246:0     20     INSERVICE    0          0          0
        max-conns            : 4000000   , out-of-rotation count : 0
        min-conns            : 4000000
        conn-rate-limit      : -         , out-of-rotation count : -
        bandwidth-rate-limit : -         , out-of-rotation count : -
        retcode out-of-rotation count : -
        load value           : 0

  rserver: SERVER3
      192.168.252.247:0     30     INSERVICE    0          0          0
        max-conns            : 4000000   , out-of-rotation count : 0
        min-conns            : 4000000
        conn-rate-limit      : -         , out-of-rotation count : -
        bandwidth-rate-limit : -         , out-of-rotation count : -
        retcode out-of-rotation count : -
        load value           : 0
```

Note: The connection failures counter increments only if the ACE attempts to load balance a connection and the ACE does not receive a SYN-ACK from the real server in response to a SYN or if the real server responds to the SYN with a RST.

6. Check the L7 load-balance statistics by entering the following command:

```
ACE_module5/Admin# show stats loadbalance

+---------------------------------------+
+------- Loadbalance statistics ----------+
+---------------------------------------+
 Total version mismatch              : 0
 Total Layer4 decisions              : 0
 Total Layer4 rejections             : 0
 Total Layer7 decisions              : 0
 Total Layer7 rejections             : 0
 Total Layer4 LB policy misses       : 0
 Total Layer7 LB policy misses       : 0
 Total times rserver was unavailable : 0
 Total ACL denied                    : 0
 Total IDMap Lookup Failures         : 0
```

Note: The ID Map is used to map real servers and server farms between the local and the remote peers in a redundant configuration. The Total IDMap Lookup Failures field increments if the local ACE fails to find the local ACE to peer ACE ID mapping. A failure can occur if the peer ACE did not send a proper remote ID for the local ACE to look up and so the local ACE could not perform a mapping or if the ID Map table was not created.

7. If you are having problems with HTTP, check the HTTP statistics and error counters by entering the following command:

```
ACE_module5/Admin# show stats http

+---------------------------------------+
+-------------- HTTP statistics ----------+
+---------------------------------------+
 LB parse result msgs sent : 0          , TCP data msgs sent       : 0
 Inspect parse result msgs : 0          , SSL data msgs sent       : 0
                      sent
 TCP fin/rst msgs sent     : 0          , Bounced fin/rst msgs sent: 0
 SSL fin/rst msgs sent     : 0          , Unproxy msgs sent        : 0
 Drain msgs sent           : 0          , Particles read           : 0
 Reuse msgs sent           : 0          , HTTP requests            : 0
 Reproxied requests        : 0          , Headers removed          : 0
 Headers inserted          : 0          , HTTP redirects           : 0
 HTTP chunks               : 0          , Pipelined requests       : 0
 HTTP unproxy conns        : 0          , Pipeline flushes         : 0
 Whitespace appends        : 0          , Second pass parsing      : 0
 Response entries recycled : 0          , Analysis errors          : 0
 Header insert errors      : 0          , Max parselen errors      : 0
 Static parse errors       : 0          , Resource errors          : 0
 Invalid path errors       : 0          , Bad HTTP version errors  : 0
 Headers rewritten         : 0          , Header rewrite errors    : 0
```

8. If you suspect a probe issue, for example, a TCP probe, check the probe statistics and error counters by entering the following command:

```
ACE_module5/Admin# show stats probe type tcp

+---------------------------------------+
+----------- Probe statistics ------------+
+---------------------------------------+
```

```
----- tcp probe ----
Total probes sent       : 0           Total send failures   : 0
Total probes passed     : 0           Total probes failed   : 0
Total connect errors    : 0           Total conns refused   : 0
Total RST received      : 0           Total open timeouts   : 0
Total receive timeout   : 0           Total active sockets  : 0
```

9. Check the parameter map statistics for an HTTP parameter map by entering the following command:

```
ACE_module5/Admin# show parameter-map HTTP_PMAP

 Number of parameter-maps : 1


 Parameter-map : HTTP_PMAP
 Type : http
    server-side connection reuse       : disabled
    case-insensitive parsing           : disabled
    persistence-rebalance              : disabled     <---------------------------- Enabled by def
    header modify per-request          : disabled
    header-maxparse-length             : 4096
    content-maxparse-length            : 4096
    parse length-exceed action         : drop
    urlcookie-delimiters               : /&#+
```

10. Clear the L7 load-balancing statistics by entering the following commands:

◊ **clear stats loadbalance** [**radius** | **rdp**]
◊ **clear service-policy** *policy_name*
◊ **clear stats http**
◊ **clear rserver** *server_name*
◊ **clear serverfarm** *serverfarm_name*