Deploying Server Load Balancing Technology

Session 2800
Agenda

- Foundational Concepts
- SLB Technology Concepts and Features
- SLB Design Guidelines and Deployment Scenarios

Foundational Concepts Agenda

- IP Connection Overview
- TCP/UDP Connection Overview
- HTTP Connection Overview
- HTTP Persistent Connections
- HTTP Pipelining
### IP Segment Header

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>4-bit version number</td>
</tr>
<tr>
<td>IHL</td>
<td>4-bit Internet Header Length</td>
</tr>
<tr>
<td>Type of Service</td>
<td>8-bit Type of Service</td>
</tr>
<tr>
<td>Total Length</td>
<td>16-bit total length</td>
</tr>
<tr>
<td>Identification</td>
<td>16-bit identification</td>
</tr>
<tr>
<td>Time to Live</td>
<td>8-bit time to live</td>
</tr>
<tr>
<td>Protocol</td>
<td>8-bit protocol number</td>
</tr>
<tr>
<td>Header Checksum</td>
<td>16-bit header checksum</td>
</tr>
<tr>
<td>Source Address</td>
<td>32-bit source address</td>
</tr>
<tr>
<td>Destination Address</td>
<td>32-bit destination address</td>
</tr>
<tr>
<td>Options</td>
<td>0 or more 32-bit words</td>
</tr>
</tbody>
</table>

### TCP/UDP Segment Headers

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Port</td>
<td>16-bit source port number</td>
</tr>
<tr>
<td>Destination Port</td>
<td>16-bit destination port number</td>
</tr>
<tr>
<td>Sequence Number</td>
<td>32-bit sequence number</td>
</tr>
<tr>
<td>Acknowledgement</td>
<td>32-bit acknowledgement</td>
</tr>
<tr>
<td>TCP Header Length</td>
<td>8-bit TCP header length</td>
</tr>
<tr>
<td>Checksum</td>
<td>16-bit checksum</td>
</tr>
<tr>
<td>Urgent Pointer</td>
<td>16-bit urgent pointer</td>
</tr>
<tr>
<td>Options</td>
<td>0 or more 32-bit words</td>
</tr>
<tr>
<td>Window Size</td>
<td>16-bit window size</td>
</tr>
<tr>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>Source Port</td>
<td>16-bit source port number</td>
</tr>
<tr>
<td>Destination Port</td>
<td>16-bit destination port number</td>
</tr>
<tr>
<td>Length</td>
<td>8-bit length</td>
</tr>
<tr>
<td>UDP Checksum</td>
<td>16-bit UDP checksum</td>
</tr>
<tr>
<td>Data</td>
<td></td>
</tr>
</tbody>
</table>
TCP Connection Overview

Three-Way Handshake:
Connection Setup

Sender

1

SYN

SYN/ACK

2

Receiver

Time

3

ACK

Data

N

ACK

N-3

FIN

ACK

N-2

FIN/ACK

ACK

N-1

HTTP Connection Overview

• HTTP is a connectionless protocol—relies on TCP

• HTTP 1.0 (RFC 1945)
  A single TCP connection is required to fetch a single URL or object
  Too much overhead on the network and routers and servers

• HTTP 1.1 (RFC 2616)
  A single TCP connection may be used to fetch multiple URL and objects
  Is the default behavior in HTTP 1.1
HTTP 1.0 Connection

Three-Way Handshake:
Connection Setup

Single URL Is Fetched

Three-Way Handshake:
Connection Tear-Down

HTTP 1.1 Connection

Three-Way Handshake:
Connection Setup

Multiple Objects Are Fetched—Persistent Connection—

Three-Way Handshake:
Connection Tear-Down
HTTP 1.1 Connection: Pipelining

Multiple Requests Are Sent Before Waiting for Each Response

GET HTTP/1.0 www.cisco.com/index.htm
GET HTTP/1.0 www.cisco.com/banner.bmp

HTTP/1.1 200 OK—Data
ACK
ACK
Data
ACK

HTTP/1.1 200 OK—Data
ACK
ACK
Data
ACK

Three-Way Handshake: Connection Setup

Three-Way Handshake: Connection Tear-Down

HTTP 1.1 vs. HTTP 1.0

Performance Test

- High bandwidth, low latency (ethernet 10Mb) RTT < 1 ms
- High bandwidth, high latency (WAN) RTT ~ 90 ms
- HTML page 42KB + 42 inline GIF images (125 KB total)
- 43 total GET requests

<table>
<thead>
<tr>
<th></th>
<th>HTTP/1.0</th>
<th>HTTP/1.1 Persistent</th>
<th>HTTP/1.1 Pipeline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Simultaneous Sockets</td>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Total Number of Sockets</td>
<td>40</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Packets from Client to Server</td>
<td>226</td>
<td>70</td>
<td>25</td>
</tr>
<tr>
<td>Packets from Server to Client</td>
<td>271</td>
<td>58</td>
<td>58</td>
</tr>
<tr>
<td>Total Number of Packets</td>
<td>497</td>
<td>83</td>
<td>83</td>
</tr>
<tr>
<td>Total Elapsed Time Sec</td>
<td>1.85</td>
<td>4.13</td>
<td>3.02</td>
</tr>
</tbody>
</table>

Pipeline Implementation vs. Nagle Interaction

The Server Introduces a Transmission Delay Hoping that More Data becomes Available so that Larger TCP Segments Can Be Sent

Server Buffers its Output before Writing it to the TCP Stack
HTTP 1.1 vs. HTTP 1.0

Performance Test

Same test with pipeline code optimizations and Nagle algorithm turned off

**Socket option TCP_NODELAY**

<table>
<thead>
<tr>
<th>High Bandwidth—Low Latency</th>
<th>Packets</th>
<th>Bytes</th>
<th>Seconds</th>
<th>% Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP/1.0</td>
<td>489.4</td>
<td>215536</td>
<td>0.72</td>
<td>8.3</td>
</tr>
<tr>
<td>HTTP/1.1</td>
<td>244.2</td>
<td>189023</td>
<td>0.81</td>
<td>4.9</td>
</tr>
<tr>
<td>HTTP/1.1 Pipelined</td>
<td>175.8</td>
<td>189607</td>
<td>0.49</td>
<td>3.6</td>
</tr>
<tr>
<td>HTTP/1.1 Pipelined and Compression</td>
<td>139.8</td>
<td>156834</td>
<td>0.41</td>
<td>3.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>High Bandwidth—High Latency</th>
<th>Packets</th>
<th>Bytes</th>
<th>Seconds</th>
<th>% Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP/1.0</td>
<td>559.6</td>
<td>248655.2</td>
<td>4.09</td>
<td>8.3</td>
</tr>
<tr>
<td>HTTP/1.1</td>
<td>309.4</td>
<td>191436.0</td>
<td>6.14</td>
<td>6.1</td>
</tr>
<tr>
<td>HTTP/1.1 Pipelined</td>
<td>221.4</td>
<td>191180.6</td>
<td>2.23</td>
<td>4.4</td>
</tr>
<tr>
<td>HTTP/1.1 Pipelined and Compression</td>
<td>182.0</td>
<td>159170.0</td>
<td>2.11</td>
<td>4.4</td>
</tr>
</tbody>
</table>

www.w3.org/Protocols/HTTP/Performance/Pipeline.html
Nagle Algorithm: rfc896

HTTP Redirection

Client

1

SYN

SYN/ACK

ACK

GET/HTTP/1.0
Host:www.cisco.com

HTTP/1.1 302 Found
Location:server1.cisco.com

GET/HTTP/1.0
Host:server1.cisco.com

N-3

FIN

ACK

N

FIN/ACK

ACK

Server

2

Three-Way Handshake: Connection Setup

Server Redirects Request to Different Host Name Which Triggers a New GET to Redirected Host

Three-Way Handshake: Connection Tear-down
HTTP Return Codes

- "200" ; OK
- "201" ; Created
- "202" ; Accepted
- "204" ; No Content
- "301" ; Moved Permanently
- "302" ; Moved Temporarily
- "304" ; Not Modified
- "400" ; Bad Request
- "401" ; Unauthorized
- "403" ; Forbidden
- "404" ; Not Found
- "500" ; Internal Server Error
- "501" ; Not Implemented
- "502" ; Bad Gateway
- "503" ; Service Unavailable

SLB Technology Concepts
Agenda (1)

- Traditional SLB
- SLB Overview
- SLB Modes of Operation
- SLB Algorithms
- Session Persistence
SLB Technology Concepts
Agenda (2)

- SLB Server Health Management
- Content Verification System
- URL Load Balancing
- SLB Redundancy
- Geographic SLB

Typical Server Load Distribution

Time of Day

Hits
Typical Server Load to Response Time

Hits per Second

Response Time Secs

Inflection Point

Server Capacity

Scaling Vertically vs. Horizontally

Cluster CPUs
More Processors
Faster and Expensive CPUs

Round Robin CPUs (DNS-Based)
More CPUs
Slower and Cheaper CPUs

Users
Round Robin DNS

- Round robin DNS rotates sequentially through small host list
- Offline servers still participate
- All servers receive equal traffic volume regardless of their capacity
- Server load is not considered when directing traffic to each server
- DNS caching issues

Load Balancing Traditional Approach

- Horizontal scaling vs. vertical scaling
- Horizontal scaling techniques
  - DNS round robin
- SLB scaling: Mapping a single IP address to many real servers
SLB Overview

Virtual Server 1 10.10.10.50 http
ftp.testnet.com 10.10.10.50

Real Server
Real Server
Real Server
Real Server

Virtual Server 2
10.10.10.50 ftp

Clients
DNS Server

SLB Modes of Operation

• Basic SLB modes
  On packets destined to the virtual server’s IP address

  Dispatch
  Rewrites the destination MAC address using the selected real server’s MAC addresses

  Directed
  Rewrites the IP and MAC addresses using the selected real server’s IP and MAC addresses
Dispatch Mode

Client
IP 10.10.20.50
MAC 0000.000c.0001

Router
IP 10.10.20.1
MAC 0000.0000.a001

Router
IP 10.10.10.1
MAC 0000.0000.a002

Server1
IP 10.10.20.1
MAC 0000.0000.dd01
eth0 10.10.10.100
eth0:0 10.10.10.50

Server2
MAC 0000.0000.dd02
eth0 10.10.10.101
eth0:0 10.10.10.50

Virtual IP Address
10.10.10.50

DST MAC Address 0000.0000.a001
SRC MAC Address 0000.0000.c001
SRC IP Address 10.10.20.50
DST IP Address 10.10.10.50
Protocol Headers (TCP/UDP)
Data

DST MAC Address 0000.0000.dd01
SRC MAC Address 0000.0000.a002
SRC IP Address 10.10.20.50
DST IP Address 10.10.10.50
Protocol Headers (TCP/UDP)
Data

Directed Mode

Client
IP 10.10.20.50
MAC 0000.0000.c001

Router
IP 10.10.20.1
MAC 0000.0000.a001

Router
IP 10.10.10.1
MAC 0000.0000.a002

Server1
IP 10.10.20.1
MAC 0000.0000.dd01
eth0 10.10.10.2

Server2
MAC 0000.0000.dd02
eth0 10.10.10.3

Virtual IP Address
10.10.10.50

DST MAC Address 0000.0000.1d00
SRC MAC Address 0000.0000.c001
SRC IP Address 10.10.20.50
DST IP Address 10.10.10.50
Protocol Headers (TCP/UDP)
Data

DST MAC Address 0000.0000.dd01
SRC MAC Address 0000.0000.a002
SRC IP Address 10.10.20.50
DST IP Address 10.10.10.2
Protocol Headers (TCP/UDP)
Data
SLB Modes of Operation

Each basic mode operates in two modes

Normal
No translation of the client IP address

Client NAT/source NAT
Client’s IP address is translated to IP address in SLB device’s address pool

SLB Algorithms

- Round robin
- Weighted round robin (loaded)
- Least connections
- Weighted least connections (weighted)
- Fastest
- Maximum connections
SLB Algorithms: RR and WRR

• If weight is equal to all servers
  The next server selected for a new connection is the one that follows the last server chosen from real server list

• If different weights are assigned to servers
  The next server selected for a new connection is chosen $n$ times before the next server in the list is chosen

Round Robin and Weighted Round Robin

Server Farm

<table>
<thead>
<tr>
<th>Server</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
</tr>
</tbody>
</table>

Chooses Next Real Server in the List

Chooses Next Real Server in the List $n$ Times

Internet

Intranet

Clients
SLB Algorithms: LC and WLC

- If weight is equal to all servers
  The next server selected for a new connection is the one with the least number of active connections
- If different weights are assigned to servers
  The next server selected for a new connection is the one with the lowest metric
  Metric = number of connections/server weight

Least Connections and Weighted Least Connections

Server Farm

- Connections=24
  - Weight=3
  - Metric=8

- Connections=9
  - Weight=1
  - Metric=9

- Connections=20
  - Weight=2
  - Metric=10

SLB Switch

Internet

Intranet

Clients

Chooses Real Server with Lowest Connections
Chooses Real Server with Lowest Metric
Metric = Conns/Weight
SLB Algorithms: Fastest

- The next server selected is the one with fastest response time
  It tends to overload servers before moving on
  Server does not follow linear progression in response time to the number of connections
  Server presents a sharp increase in response time at certain loads

Fastest and Maxconns
Sessions Persistence A.K.A. Stickiness

- **Session**
  Loosely defined as series of HTTP and TCP connections to a single site that constitute a client’s visit

- **Persistent or sticky sessions**
  Connections that belong to the same session for which session state is kept
  This creates the need to send all session connections to the same server regardless of the TCP or UDP port

- **Session state**
  Information being stored to help remember client’s relevant pieces of information for the duration of a session
  State information could be kept at the client, server or both

Why Sticky?

Virtual Server 1
  Port 80

Virtual Server 2
  Port 443

Shopping Cart
Database

Client
X

Y

Internet

Client

Common Database Approach

Virtual Server 1
Port 80

Virtual Server 2
Port 443

Client

Internet

Virtual Server 1

Virtual Server 2

Session Persistence or Sticky Modes

- Source IP sticky
  - Client’s IP address
- Cookie sticky
  - Cookie-passive
  - Cookie-insert
- SSL sticky
  - SSL ID v2-v3
- HTTP redirection sticky
Source IP Sticky

- Sticky session based on client’s IP
  
  The mechanism used to direct different requests to the same server is based upon the IP address of the client

  Sticky connection is controlled by the inactivity time (time between different TCP connections)

- Source IP sticky A.K.A
  
  Buddy sticky and sticky groups—Cisco IOS® SLB—

  Sticky generic—LD—

- Recommendation: match the sticky timer with the server timer

Session Persistence Issues: Source IP Sticky

[Diagram showing virtual server routing to different servers and the concept of source IP sticky being turned off.]
A “cookie” is a small piece of information sent by a web server to store on a web browser so it can later be read back from that browser.

- This piece of information could be used to associate session state information with a specific user.
- Cookies are transmitted in the HTTP header from the server in the form of a:

  ```
  Set-Cookie: NAME=VALUE; expires=DATE; path=PATH; domain=DOMAIN_NAME
  ```

- Up to 300 cookies per client
- Each cookie up to 4K
- 20 cookies per server or domain per session
Cookie Sticky

• Sticky connection based upon cookie
  The mechanism used to direct different requests to the same server is based upon the cookie kept by the client’s browser
  Cookie is assigned by the server or the SLB device
  TCP connection is proxied long enough to determine correct server

Cookie Sticky

• Cookie-insert
  The session state is maintained by a cookie created by the SLB device
  Client side of connection is proxied long enough to determine if there a cookie in the HTTP header

• Cookie-passive
  The session state is maintained by a cookie provided by the server
  Client side of connection is proxied long enough to determine if there a cookie in the HTTP header
Cookie Insert

- Cookie-insert without cookie from client

- Cookie-insert with cookie from client
Cookie Passive

- Cookie-passive without cookie from client

Client → SYN → SLB Device → SYN/ACK → Server

Client → ACK → Unproxied Connection Client and Server Talk Directly → HTTP GET w/o Cookie

HTTP Response w/ Set Cookie → Server → SYN → Server → SYN/ACK → Server

HTTP Response w/ Set Cookie → Server → ACK → Server

Unproxied Connection Client and Server Talk Directly

Cookie Passive

- Cookie-passive with cookie from client

Client → SYN → SLB Device → SYN/ACK → Server

Client → ACK → Unproxied Connection Client and Server Talk Directly → HTTP GET w/o Cookie

HTTP Response w/ Set Cookie → Server → SYN → Server → SYN/ACK → Server

HTTP Response w/ Set Cookie → Server → ACK → Server

HTTP GET w/ Cookie → Server → HTTP Response w/ Set Cookie
Session Persistence Issues
Cookie Sticky

Virtual Server

Source IP Sticky
Cookie Sticky

SSL Sticky

- HTTPS
  Normal HTTP wrapped in SSL layer
- SSL handshake
  Authenticate server to client using public-key
  Authenticate client to server
  Establish an encrypted SSL connection
- SSL sticky uses SSL ID
SSL Sticky

- SSL v2.0 encrypts SSL ID
  Can’t be used for persistence
- SSL v3.0 uses clear text for SSL ID
  SSL ID can be used to track SSL session

Session Persistence Issues
SSL Sticky
Session Persistence Issues
SSL Sticky

SSL Sticky: SSL ID Renegotiated
SSL Timer or HTTP Redirect Sticky

Virtual Server

Go to B

Company A

SLB Device

Company C

SSL only

SSLProxy Servers

HTTP Redirection Sticky

• HTTP redirection
  Client’s request is sent back using an HTTP redirect to the appropriate URL
  Bookmark issue: backup

• Server changes
  No absolute URLs
  DNS entries for virtual addresses
HTTP Redirection Sticky

Client

GET/HTTP/1.0
Host:www.cisco.com

HTTP/1.1 302 Found
Location:server2.cisco.com

GET/HTTP/1.0
Host:server2.cisco.com

HTTP/1.1 200 OK
Host:server2.cisco.com

Client talks to server2.cisco.com for the reminder of the session

Server Farm 2-Tier Architecture

Web/Application Servers

Database Server

SLB Device

Company A

Company C

Internet

R1

R2

R3

SLB Device

© 2000, Cisco Systems, Inc.
Session Persistence

<table>
<thead>
<tr>
<th>Session Persistence Method</th>
<th>No Proxy HTTP Only Traffic</th>
<th>No Proxy E-Commerce Traffic</th>
<th>No Proxy SSL Only Traffic</th>
<th>One or More Proxies HTTP Only Traffic</th>
<th>One or More Proxies E-Commerce Traffic</th>
<th>One or More Proxies SSL Only Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source IP Sticky</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cookie Sticky</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>SSL Sticky</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>HTTP Redirection</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Server Farm 3-Tier Architecture

Application Servers

Web Servers

Database Server

SLB Device

Internet

Company A

Company C
### Session Persistence

<table>
<thead>
<tr>
<th>Session Persistence Method</th>
<th>No Proxy HTTP Only Traffic</th>
<th>No Proxy E-Commerce Traffic</th>
<th>No Proxy SSL Only Traffic</th>
<th>One or More Proxies HTTP Only Traffic</th>
<th>One or More Proxies E-Commerce Traffic</th>
<th>One or More Proxies SSL Only Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source IP Sticky</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Cookie Sticky</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SSL Sticky</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>HTTP Redirection</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

---

### SLB Server Health Management (1)

- **Automatic server failure detection**
  - A connection is reassigned to a new server if:
    - A number of consecutive SYNs, controlled by a reassign counter, are not answered
    - A SYN is replied with a RST
  - Server is placed “out of service” once number of reassigned connections exceeds a threshold counter
SLB Server Health Management (2)

- **Auto Unfail**
  No new connections are sent to a “failed” server for the retry amount of time; If next new connection is successful, server is placed “in service”

- **Slow Start**
  It prevents a recent “in service” real server overload by controlling the number of connections directed to it, since it start with zero connections
  
  Slowstart is applicable to least connections and weighted least connections

SLB Server Health Management (3)

- **MAXCONNS**
  To limit the number of active connections on a real server

- **SynGuard**
  It prevents a SYN flood denial of service attack by controlling the number of unanswered SYNids over a period of time doesn’t exceed the syn-count

LD and Cisco IOS-SLB default disable
Content Verification (1)

• HTTP probes

  User defined mechanisms to test application availability
  Uses HTTP methods that provide feedback through HTTP return codes based on the status of the URL
  Methods: Get or post and optional username, password, name of URL
  HTTP return codes may be used to take server “out of service”
  HTTP probes are defined per serverfarm and applied to All Real Servers
  Virtual Servers associated with a serverfarm are probed
  Real(s) failed for a Virtual will be failed for all Virtuals in the serverfarm

HTTP Probes

HTTP PROBE DEFINITION
Expect: 200
GET/HTTP/1.0
Accept: */*
Connection: close
User-Agent:slb-probe/1.0

HTTP/1.0 200 OK

Serverfarm WEB
Vip1 http:80

HTTP PROBE DEFINITION
Expect: 204
POST /cgi/form.pl HTTP/1.0
Accept: */*
Connection: close
User-Agent:slb-probe/1.0
name= john & lastname=smith

HTTP/1.0 204 No Content
Content Verification (2)

- Content Verification System
  - HTTP probes from a management station
  - Probes can be run automatically by a user-specified schedule
  - Probes can run on specified real server(s)
  - Extensive list of conditions on the content verification process
  - Much more flexible than built-in probes

Content Verification (3)

- Content Verification System
  - Probes can test the following:
    - Response time of a server when a request for that URL is made
    - Response of a server when particular cookies are present in the request
    - Response of a server to particular user agents (browser types)
    - Proper operation of any HTTP basic authentication scheme protecting access to that URL
    - Any redirection performed by the server when that URL is requested
    - Length of the content returned by the server
    - Type of content returned by the server
    - Status code returned by the server
    - Presence or absence of particular strings in the content returned by the server
    - Presence and content of any cookies returned by the server
Content Verification System

CVS Station

GET/HTTP/1.0
Host:www.cisco.com

HTTP/1.1 200 OK
Host:server2.cisco.com

CVS Station Keep Track of:
HTTP Return Code
Content Length
User Authentication OK
Response Time: 25ms

SLB Device

GET/HTTP/1.0
Host:www.cisco.com

HTTP/1.1 200 OK
Host:server2.cisco.com

Dynamic Feedback Protocol

- Cisco developed Dynamic Feedback Protocol (DFP) to allow greater network/server integration
- Server and application health metrics can be used in load balancing decision
- Supported on NT, UNIX, and OS/390 platforms (Cisco)
- TCP session-based
- Security via MD5 or DES
- Introduced to IETF for standardization
- Partners:
  - Workload Manager
  - SiteAssure
  - Workload Manager
  - ColdFusion

Content Verification System

Server and application health metrics can be used in load balancing decision.
Dynamic Feedback Protocol

Serverfarm SECURE-WEB
server1.cisco.com
CPU: 70%
Mem: 50%
Session: 80%
server2.cisco.com
CPU: 10%
Mem: 40%
Session: 30%
server3.cisco.com
CPU: 20%
Mem: 30%
Session: 30%

Serverfarm WEB

WRR or WLC

Weight=x

Internet
Intranet

SLB Device

Clients request specific URL
SLB looks to match content based on ‘regular expression’
Virtual are defined per specific URL or regular expression

virtual 1.1.1.1:80:0:*.jpg
virtual 1.1.1.1:80:0:*.html

Bind real servers to each virtual
Regular expression could have a ‘*’ for multiple characters and a ‘?’ for a single character
URL Load Balancing

**Benefits**
- Specific redirection of URL’s to servers
- Segregation of content to specific servers

**Disadvantages**
- Static configuration
- Performance is much lower
- HTML coding provides the same service
SLB Redundancy

- No single point of failure on SLB device
- State-less Failover
  Connection state information is not maintained
- State-full Failover
  Connection state information is maintained
- active-standby & active-active

State-less Failover
Active-Active

- High performance SLB environment
- No session state requirements under failure scenarios
- HSRP and MHSRP or IP routing to upstream switches
- SLB and HSRP on SLB switch
State-full Failover
Active-Standby

- Highly redundant SLB environment with session state redundancy
- HSRP and MHSRP or IP routing to upstream switches
- SLB and optional routing HSRP on SLB switch
- SLB, HSRP and state replication on SLB switches

Geographic Server Load Balancing
GSLB

- Selection of the best server farm or cache cluster
- Best location is selected in relation to the user’s location
- Selection process uses DNS and is based on:
  - RTT
  - Content availability and freshness
  - Current server farm load
  - Fastest triggered response
- User is bound to selected location: TTL
- Distributed Director, Boomerang, Content Routers
DD uses UDP based Director Response Protocol (DRP)

- DRP Server Agents return:
  - BGP and IGP metrics to determine topological proximity
  - DRP external, DRP internal, DRP server and DRP-MED
  - server-to-client RTT metrics
  - TCP probe to determine round-trip times

- DD operates as:
  - DNS Caching nameserver or HTTP session redirector
SLB Design Guidelines and Deployment Scenarios

- Clients and reals must not be in the same VLAN
- LD is a bridge vs. Cisco IOS SLB devices which are routers/switches
- In dispatch mode reals must be L2 adjacent to SLB device
- When multiple virtuals are configured to use the same IP address, they must be on the same VLAN
- In SLB devices control packets are control switched and data packets are hardware switched
SLB Design Guidelines (2)

- When using LD FEC make sure to configure the switch to LB at the MAC layer
- Enable ping_allow if you want to ping LD IP addresses
- Use “secure” to turn off bridging on LD interfaces
- LD alias—equivalent to secondary address on Cisco IOS SLB devices

SLB Design Guidelines (3)

- Cisco IOS SLB server farms
  
  A Cisco IOS SLB server farm is a set of real servers configured to provide some common services
  
  A single SLB engine may support multiple server farms instances
  
  A single server farm may have multiple real servers and multiple virtual servers
  
  A real server may be a member of multiple server farms and multiple virtual servers
  
  Real server specific configuration applies to the server farm where the parameters were defined
  
  Real server failures take effect within the server farm only
Accelerated Server Load Balancing

SLB Design Scenarios

- Design Scenarios
- Case 1
dispatch mode loopback and primary—different subnets
- Case 2
directed mode state-full failover active-standby IOSSLB
- Case 3
dispatch mode state-full failover active-standby ASLB

<table>
<thead>
<tr>
<th>No.</th>
<th>VLAN</th>
<th>MAC-Dst</th>
<th>MAC-Src</th>
<th>IPDA</th>
<th>IPSA</th>
<th>Flags</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>98</td>
<td>MAC-LD</td>
<td>MAC-R</td>
<td>VIP</td>
<td>CIP</td>
<td>SYN</td>
<td>Path 1, Candidate/ACL redirect</td>
</tr>
<tr>
<td>2</td>
<td>99</td>
<td>MAC-Sn</td>
<td>MAC-R</td>
<td>VIP</td>
<td>CIP</td>
<td></td>
<td>Path 2, Enabler</td>
</tr>
<tr>
<td>3-N</td>
<td>98</td>
<td>MAC-LD</td>
<td>MAC-R</td>
<td>VIP</td>
<td>CIP</td>
<td></td>
<td>Path 3, shortcut</td>
</tr>
<tr>
<td>N+1</td>
<td>99</td>
<td>MAC-LD</td>
<td>MAC-R</td>
<td>VIP</td>
<td>CIP</td>
<td>FIN/STR</td>
<td>Path 1, ACL redirect</td>
</tr>
<tr>
<td>N+2</td>
<td>98</td>
<td>MAC-Sn</td>
<td>MAC-R</td>
<td>VIP</td>
<td>CIP</td>
<td>FIN/RST</td>
<td>Path 2</td>
</tr>
</tbody>
</table>

MAC-LD  MAC address of Local Director
MAC-Sn  MAC address of Server n
MAC-R   MAC address of Router
Case 1: Dispatch Mode

SLB Engine

10.10.10.1
http VIP1:10.10.100.50
ftp VIP2:10.10.100.60

10.10.20.1
10.10.30.1

10.10.20.50
10.10.20.51
10.10.20.52
10.10.20.53

R3
R5
R6

R2

http VIP1:10.10.100.50
ftp VIP2:10.10.100.60

Case 2: Directed Mode state-full Failover

http VIP1:10.10.100.50
https VIP2:10.10.100.50

10.10.10.102
10.10.10.103
10.10.10.105
10.10.10.106

Vlan 11
Vlan 21

primary SLB switch
Standby SLB switch

hsrp address 10.10.10.1
hsrp address 10.10.21.1

R2
R3
R5
R6

10.10.10.2
10.10.10.3
10.10.21.2
10.10.21.3
10.10.21.4

10.10.20.1
10.10.20.2
10.10.20.50
10.10.20.51
10.10.20.52
10.10.20.53

Clients

10.10.30.50
10.10.30.51
10.10.30.52
10.10.30.53
Case 3: ASLB Redundant Scenario

Useful Links

- **SSL links**
  - Netscape→www.netscape.com
    - Cookie Specification
    - SSL v2/0 Specification
      [http://home.netscape.com/eng/security/SSL_2.html](http://home.netscape.com/eng/security/SSL_2.html)
    - SSL v3.0 Specification
    - SSL v3.0 current draft

- **More on SSL**
  - Check your browser's support for SSL
    [http://www.fortify.net/sslcheck.html](http://www.fortify.net/sslcheck.html)
  - Microsoft information on SSL
Useful Links

- WWW Consortium — www.w3c.org
  HTTP performance http://www.w3.org/Protocols/HTTP/Performance/Pipeline.html
- More on Cookies — www.cookiecentral.com
- Nagle Algorithm — rfc896.txt
- HTTP Docs
  HTTP 1.0 rfc1945.txt http 1.0
  HTTP 1.1 rfc2616.txt http 1.1

Deploying Server Load Balancing Technology

Session 2800
Please Complete Your Evaluation Form

Session 2800