Contents

Goal

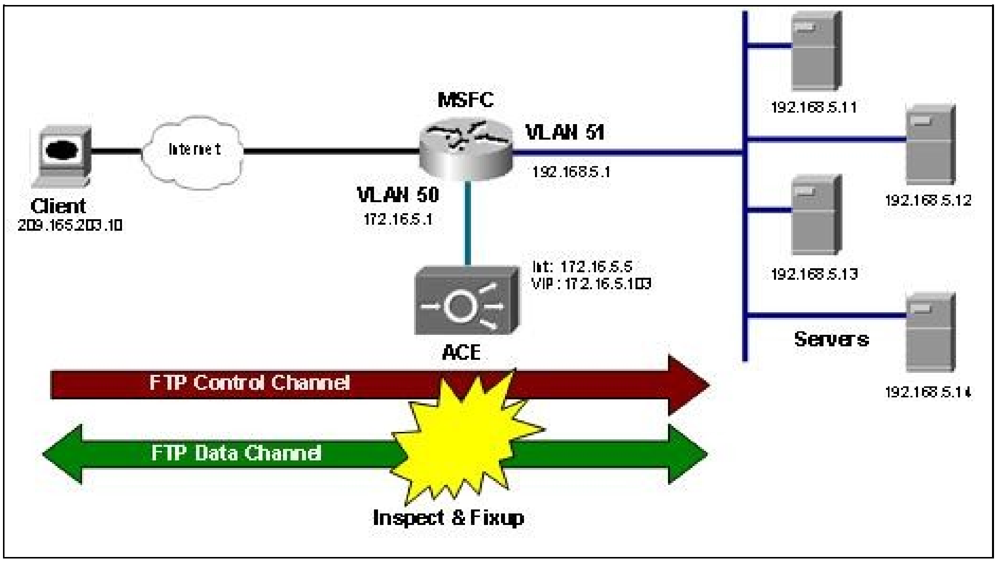
The goal of this document is to configure an ACE Module or ACE 4710 to perform FTP load balancing in a one-arm topology. It will cover the basics of the FTP protocol, and explain why specific configuration elements are necessary to allow FTP to function.

FTP Protocol Basics

FTP is a protocol which allows client PCs running FTP client software, to transfer files to and from a remote FTP server. An FTP session is itself composed of two TCP flows, each with a very specific role. The first flow is created as the user initiates the FTP connection to the server. It is used to pass FTP commands (such as GET, PUT, etc) back and forth between the client and server, and is known as the control channel. This flow is always sourced from the client PC, with the destination being the FTP server (generally TCP port 21), never the other way around. The next TCP flow is known as the data channel, and it is created when data needs to flow between the client and the server (file copies, directory listings, etc). This flow can be established in either direction, depending on whether ACTIVE FTP or PASSIVE FTP is being used. To establish the data connection during an Active FTP session, the server initiates the TCP connection to the client PC. To establish the data connection during a Passive FTP session, the client initiates the TCP connection to the FTP Server. It is important to note that when an FTP session authenticates properly but then hangs on directory listings or file transfers, generally some piece of network equipment is preventing the data channel from being properly established.

Design

Clients will establish an FTP control channel connection with the VIP configured on the ACE. Once established, the ACE will forward the control connection to one of the configured real servers. The ACE will inspect the commands being sent through the control connection, and will take action as necessary to ensure the data connection can also be established between the client and the server in the appropriate direction when necessary. Generally this will involve performing NAT on the IP addresses embedded within the FTP control channel messages, as well as opening any necessary ports in the ACE access-lists. The ACE will also source nat the request as it is passed to the real server. This will ensure that the server response is sent back to the ACE, rather than being sent through the MSFC, bypassing the ACE completely. Only the TCP port for the control channel must be explicitly permitted in the ACE access-list. The TCP port for the data channel is dynamically assigned by the client or server (depending on which FTP mode is used), and ACE will open a pin-hole in its access-list to allow traffic through to the real server on this port.



Configuration

The ACE configuration is performed in a layered fashion, making the order it is built in important. Each configuration step builds upon the previous step; the order this document will follow is outlined below.

- Configure a management policy to allow admin access to the ACE
- Configure access-list to permit traffic into the ACE from the client facing interface
- Define real server addresses (create the rservers)
- Group rservers together (create a serverfarm)
- Define the virtual address (VIP)
- Define how traffic is to be handled once it is received (L7 policy-map)
- Associate traffic handling policy with VIP address (multi-match policy)
- Create VLAN interface and net-pool, then apply service-policy, access-list, and management policy to it
- Add a default route

To begin the configuration, configure a management policy-map to allow all types of management access to the ACE. This policy will be applied to the necessary interface in a later step.

```
ACE-1/onearm(config)# policy-map type management first-match remote-access
ACE-1/onearm(config-pmap-mgmt)# class class-default
ACE-1/onearm(config-pmap-mgmt-c)# permit
```

Next configure an access-list to permit the desired traffic to enter the ACE. Before the traffic can reach any configured virtual servers, it must be permitted by an access-list. Note: While this example shows a ?permit any any?, it is recommend ACLs be used to only permit specific traffic through the ACE.

```
ACE-1/onearm(config)# access-list everyone extended permit ip any any
ACE-1/onearm(config)# access-list everyone extended permit icmp any any
```

The ultimate goal of this configuration is for ACE to distribute FTP connections to a group of real servers. The ACE must have each of these real servers configured as rservers, so that it knows each of their IP addresses. Note that unlike previous SLB products, a TCP/UDP port is NOT specified during this step; it will be defined when the rservers are added to a serverfarm.

```
ACE-1/onearm(config)# rserver host lnx1
ACE-1/onearm(config-rserver-host)# ip address 192.168.5.11
```

```
ACE-1/onearm(config-rserver-host)# inservice
ACE-1/onearm(config-rserver-host)# rserver host lnx2
ACE-1/onearm(config-rserver-host)# ip address 192.168.5.12
ACE-1/onearm(config-rserver-host)# inservice
ACE-1/onearm(config-rserver-host)# rserver host lnx3
ACE-1/onearm(config-rserver-host)# ip address 192.168.5.13
ACE-1/onearm(config-rserver-host)# inservice
ACE-1/onearm(config-rserver-host)# rserver host lnx4
ACE-1/onearm(config-rserver-host)# ip address 192.168.5.14
ACE-1/onearm(config-rserver-host)# inservice
ACE-1/onearm(config-rserver-host)# rserver host lnx5
ACE-1/onearm(config-rserver-host)# ip address 192.168.5.15
ACE-1/onearm(config-rserver-host)# inservice
```

The rservers must be grouped into a serverfarm, accomplishing two things. It allows the whole group of rservers to be attached to any load balancing actions with a single command, and it provides an opportunity to define the port on which the rservers are configured to accept traffic.

Note: In this example, no port is configured on the rservers. This instructs the ACE to inherit the port from the virtual server which will be defined in a later step.

```
ACE-1/onearm(config)# serverfarm host ftp
ACE-1/onearm(config-sfarm-host)# rserver lnx1
ACE-1/onearm(config-sfarm-host-rs)# inservice
ACE-1/onearm(config-sfarm-host-rs)# rserver lnx2
ACE-1/onearm(config-sfarm-host-rs)# inservice
ACE-1/onearm(config-sfarm-host-rs)# rserver lnx3
ACE-1/onearm(config-sfarm-host-rs)# inservice
ACE-1/onearm(config-sfarm-host-rs)# rserver lnx4
ACE-1/onearm(config-sfarm-host-rs)# inservice
ACE-1/onearm(config-sfarm-host-rs)# rserver lnx5
ACE-1/onearm(config-sfarm-host-rs)# inservice
```

In order for the ACE to accept traffic on a virtual server IP, it must be configured using a class-map. In this example the VIP address is configured to accept traffic on TCP port 21, the standard port for FTP control channel connections.

```
ACE-1/onearm(config)# class-map match-all slb-vip
ACE-1/onearm(config-cmap)# 2 match virtual-address 172.16.5.103 tcp eq ftp
```

Once the ACE accepts traffic destined to the virtual address, it must be told how to handle the traffic. This is accomplished by configuring an L7 policy-map. In this example the policy-map is configured to match all traffic (class-default matches anything), and to send it to the ftp serverfarm.

```
ACE-1/onearm(config)# policy-map type loadbalance first-match slb
ACE-1/onearm(config-pmap-lb)# class class-default
ACE-1/onearm(config-pmap-lb-c)# serverfarm ftp
```

The final ?glue? which ties all of the previous steps together, is the multi-match policy. It can contain multiple class references; each would be configured with a different VIP address. In this case only one class is referenced, instructing the ACE to only accept traffic destined to the single VIP address. The class reference also references the L7 policy-map, and a command to put the VIP in service.

Caution: Since this configuration is an example of FTP load balancing, the class reference also contains the ?inspect ftp? command. It instructs the ACE to inspect the FTP control channel commands, and perform any necessary fixups to allow the data channel to establish properly. Without this command, FTP load balancing **WILL NOT WORK!**

```
ACE-1/onearm(config)# policy-map multi-match client-vips
ACE-1/onearm(config-pmap)# class slb-vip
```

```
ACE-1/onearm(config-pmap-c)# loadbalance vip inservice
ACE-1/onearm(config-pmap-c)# loadbalance policy slb
ACE-1/onearm(config-pmap-c)# inspect ftp
```

At this point the traffic handling logic is completely defined within the configuration. The final step is to apply this logic to the interfaces of the ACE. The following steps create the one-arm VLAN interface, and apply the multi-match policy and access-list to it. A NAT pool is also added to the interface, to allow the ACE to source nat the client requests.

```
ACE-1/onearm(config)# interface vlan 50
ACE-1/onearm(config-if)# description ?Client-Server VLAN?
ACE-1/onearm(config-if)# ip address 172.16.5.5 255.255.255.0
ACE-1/onearm(config-if)# nat-pool 5 172.16.5.200 172.16.5.209 netmask 255.255.255.0 pat
ACE-1/onearm(config-if)# access-group input everyone
ACE-1/onearm(config-if)# service-policy input client-vips
ACE-1/onearm(config-if)# service-policy input remote-access
ACE-1/onearm(config-if)# no shutdown
```

Next, apply the nat-pool which was configured. The nat-pool is applied to the class reference within the multi-match policy. This instructs the ACE to source NAT all client requests destined for the VIP address.

```
ACE-1/onearm(config)# policy-map multi-match client-vips
ACE-1/onearm(config-pmap)# class slb-vip
ACE-1/onearm(config-pmap-c)# nat dynamic 5 vlan 50
```

The last step is to add a default route. This allows the ACE to be reachable from remote networks, and allows the ACE to return traffic to distant clients.

```
ACE-1/onearm(config)# ip route 0.0.0.0 0.0.0.0 172.16.5.1
```

Related show Commands

The following command can be used to verify that both the control channel and the data channel were successfully established. In this example conn-id 3 illustrates the control channel being established from the client to the VIP on TCP port 21, and conn-id 19 illustrates the data channel being established from the VIP to the client on TCP port 4726. The directionality of the data channel indicates that this is an active mode FTP session. Note that normally the data channel is torn down immediately after use; in order to observe its behavior a long-running file transfer must be in progress.

```
ACE-1/onearm# sho conn
total current connections : 4
conn-id    np dir proto vlan source                destination           state
----------+--+---+-----+----+---------------------+---------------------+------+
20         1  in  TCP   40   192.168.5.11:20       209.165.201.11:4726   ESTAB
19         1  out TCP   20   209.165.201.11:4726   172.16.5.103:20       ESTAB
3          2  in  TCP   20   209.165.201.11:2045   172.16.5.103:21       ESTAB
18         2  out TCP   40   192.168.5.11:21       209.165.201.11:2045   ESTAB
```

The following command can be used to observe the behavior of the FTP inspection engine. The hit count should increase over time, and dropped connections should not increment.

```
ACE-1/onearm# show service-policy client-vips detail
Status     : ACTIVE
Description: -
-----------------------------------------
Interface: vlan 50
  service-policy: client-vips
    class: slb-vip
```

```
     VIP Address:    Protocol:  Port:
     172.16.5.103    tcp        eq    21
      loadbalance:
        L7 loadbalance policy: slb
        VIP Route Metric     : 77
        VIP Route Advertise  : DISABLED
        VIP ICMP Reply       : DISABLED
        VIP State: INSERVICE
        curr conns       : 0         , hit count        : 8
        dropped conns    : 0
        client pkt count : 169       , client byte count: 7771
        server pkt count : 193       , server byte count: 12506
        conn-rate-limit      : 0         , drop-count : 0
        bandwidth-rate-limit : 0         , drop-count : 0
        L7 Loadbalance policy : slb
          class/match : class-default
            LB action :
               primary serverfarm: ftp
                    state: UP
                backup serverfarm : -
            hit count        : 8
            dropped conns    : 0
      inspect ftp:
        L7 inspect policy : -
        strict ftp: DISABLED
        curr conns       : 0         , hit count        : 8
        dropped conns    : 0
        client pkt count : 169       , client byte count: 7771
        server pkt count : 193       , server byte count: 12506
        conn-rate-limit      : 0         , drop-count : 0
        bandwidth-rate-limit : 0         , drop-count : 0
```

Show running-config

```
access-list everyone line 8 extended permit ip any any
access-list everyone line 16 extended permit icmp any any

rserver host lnx1
  ip address 192.168.5.11
  inservice
rserver host lnx2
  ip address 192.168.5.12
  inservice
rserver host lnx3
  ip address 192.168.5.13
  inservice
rserver host lnx4
  ip address 192.168.5.14
  inservice
rserver host lnx5
  ip address 192.168.5.15
  inservice

serverfarm host ftp
  rserver lnx1
    inservice
  rserver lnx2
    inservice
  rserver lnx3
    inservice
  rserver lnx4
    inservice
  rserver lnx5
```

```
    inservice

class-map match-all slb-vip
  2 match virtual-address 172.16.5.103 tcp eq ftp

policy-map type management first-match remote-access
  class class-default
    permit

policy-map type loadbalance first-match slb
  class class-default
    serverfarm ftp

policy-map multi-match client-vips
  class slb-vip
    loadbalance vip inservice
    loadbalance policy slb
    inspect ftp
    nat dynamic 5 vlan 50

interface vlan 50
  description Client-Server VLAN
  ip address 172.16.5.5 255.255.255.0
  access-group input everyone
  nat-pool 5 172.16.5.200 172.16.5.209 netmask 255.255.255.0 pat
  service-policy input client-vips
  service-policy input remote-access
  no shutdown

ip route 0.0.0.0 0.0.0.0 172.16.5.1
```

Related Information

Technical Support & Documentation - Cisco Systems