High Availability

SWITCH Module 9

Agenda

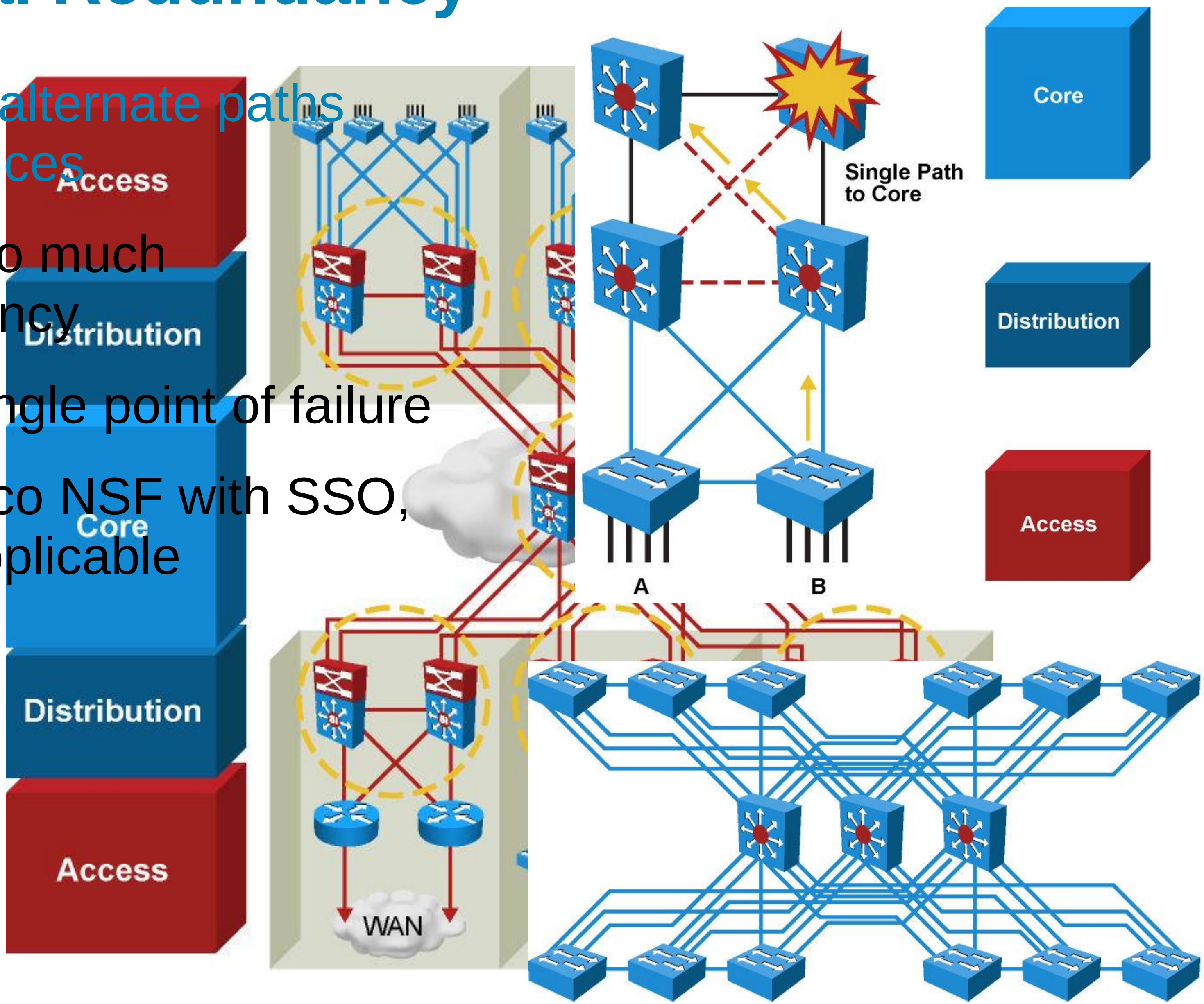
- **StackWise**
- **Virtual System Switching**
- **Redundant Processor Supervisor**
- **Server Load Balancing**
- **BiDirectional Forwarding**

Resiliency for High Availability

- High availability is implemented with the following components
  - **Network-level resiliency**
    - Redundant links
    - Redundant devices
    - Power redundancy
    - Fast convergence
  - **System-level resiliency**
    - Integrated hardware resiliency
    - Redundant power supply
    - Stackable switches
  - **Management and monitoring**
    - Detection of failure

Optimal Redundancy

- Provide alternate paths and devices
- Avoid too much redundancy
- Avoid single point of failure
- Use Cisco NSF with SSO, when applicable



StackWise

What Is StackWise?

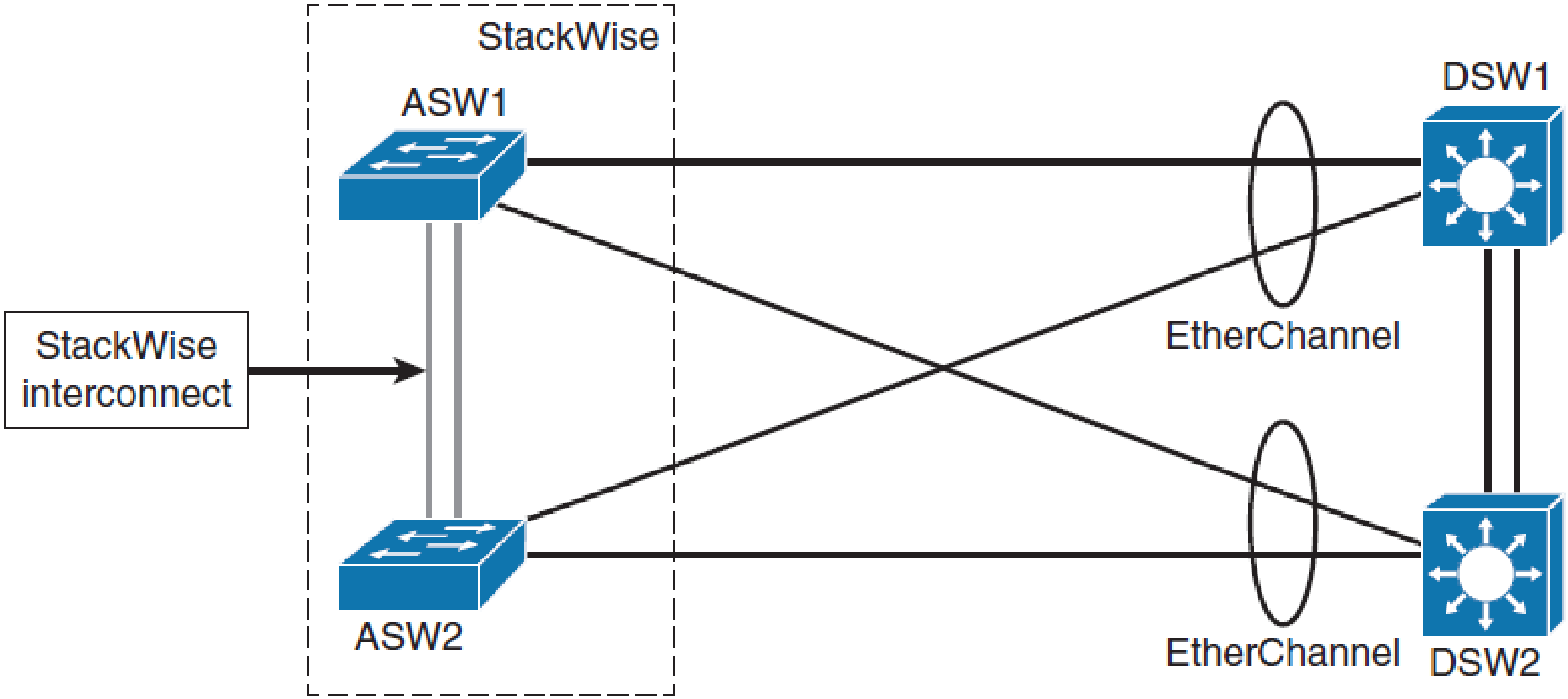
- Cisco StackWise technology provides a method for collectively utilizing the capabilities of a stack of switches.
- Configuration and routing information is shared by every switch in the stack, creating a single switching unit.
- Switches can be added to and deleted from a working stack without affecting performance.

- The stack is managed as a single unit by a master switch, which is elected from one of the stack member switches.

StackWise Details

- Each stack of switches has a single IP address and is managed as a single object.
- This allows each switch in the stack to share the same network topology, MAC address, and routing information.
- Catalyst 3750-E, 3750-X, and 3850 series switches support StackWise and StackWise Plus.
- StackWise Plus is an evolution of StackWise. StackWise Plus supports local switching, so locally destined packets need not traverse the stack ring.
- Catalyst 3850 series supports StackWise-480 with improved 480-Gbps stacking. Catalyst 2960-S series supports FlexStack, aStackWise-based feature tailored for Layer 2 switches. FlexStack is limited to four stacked switches.

StackWise Benefits



Verifying StackWise

```
Switch1# show switch
Switch/Stack Mac Address: 0013.6075.7280

Switch#  Role      Mac Address      Priority H/W   Version  Current State
*1       Master    0013.6075.7280       1               0      Ready
2        Member    0013.60e1.1800       1               0      Ready
```

```
Switch1# show switch stack-ports
Switch #         Port 1              Port 2
---------        ------              ------
1                 Ok                  Ok
2                 Ok                  Ok
```

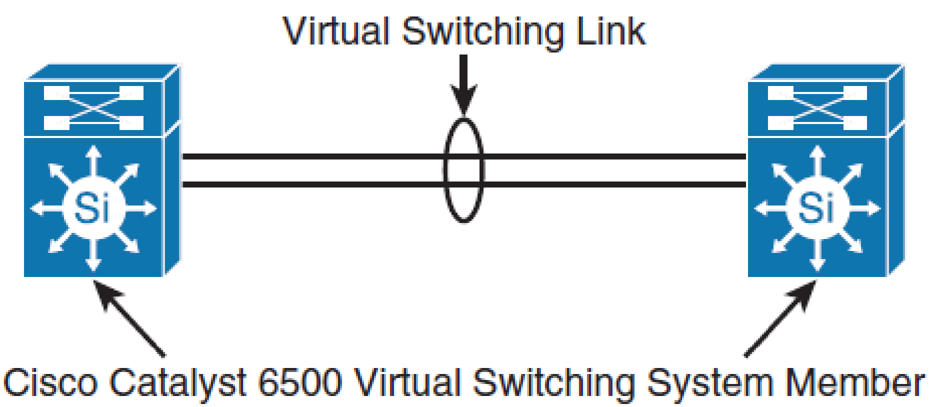
Virtual Switching System

What Is VSS?

- Virtual Switching System (VSS) is a network system virtualization technology that combines a pair of Catalyst 4500 or 6500 series switches into one virtual switch, increasing the operational efficiency, boosting nonstop communications, and scaling the system bandwidth capacity.
- The VSS simplifies network configuration and operation by reducing the number of Layer 3 routing neighbors and by providing a loop-free Layer 2 topology.

What Is VSS?

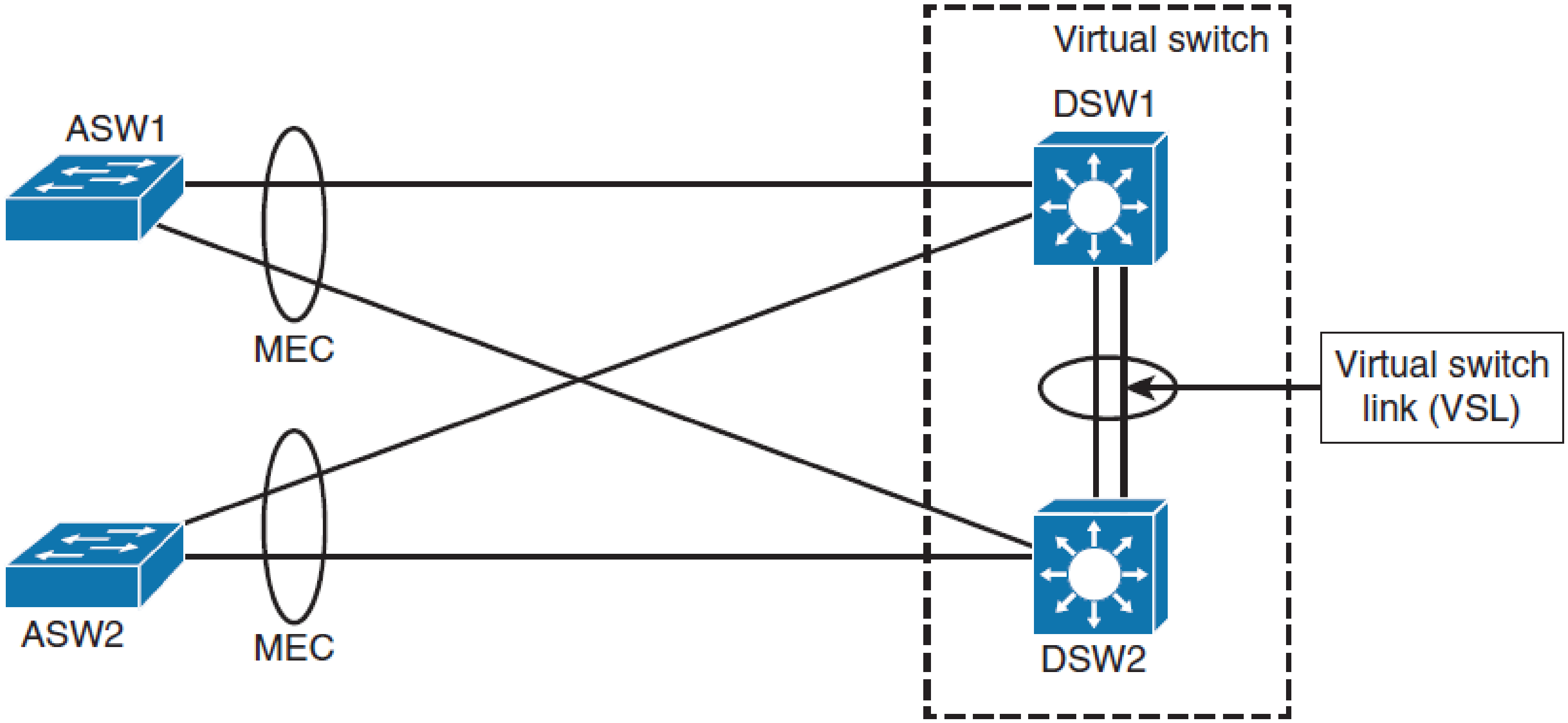
- The VSL is made of up to eight 10 Gigabit Ethernet connections bundled into an EtherChannel.
- VSL carries the control plane communication between the two VSS members, in addition to regular data traffic.
- Once the VSS is formed, only the control plane of one of the members is active. The data plane and switch fabric of both members are active.
- Both chassis are kept in sync with the interchassis SSO mechanism, along with NSF to provide nonstop communication even in the event of failure of one is.



VSS Benefits

- VSS increases operational efficiency by reducing switch management overhead and simplifying the network.
- It provides a single point of management, IP address, and routing instance.
- Neighbors see the VSS as a single Layer 2 switching or Layer 3 routing node, thus reducing the control protocol traffic.
- VSS provides a single VLAN gateway IP address, removing the need for the first-hop redundancy protocol (HSRP, VRRP, GLBP),
- Multichannel EtherChannel (MEC) allows you to bundle links to two physical switches in VSS, creating a loop-free redundant topology without the need for STP.
- Interchassis stateful failover results in no disruption to applications that rely on network state information.
- VSS eliminates Layer 2 / Layer 3 protocol reconvergence if a virtual switch member fails, resulting in deterministic subsecond virtual switch recovery.

VSS Benefits



Verifying VSS

To verify the status of VSS configuration, use the following commands:

- **`show switch virtual`**
- **`show switch virtual link`**
- **`show switch virtual role`**
- **`show switch virtual link port-channel`**

```
Switch1# show switch virtual
Switch mode                        : Virtual Switch
Virtual switch domain number       : 1
Local switch number                : 1
Local switch operational role      : Virtual Switch Active
Peer switch number                 : 2
Peer switch operational role       : Virtual Switch Standby
```

Verifying VSL

```
Switch1# show switch virtual link
VSL Status : UP
VLS Uptime : 7 weeks, 4 days, 31 minutes
VSL SCP Ping : Pass
VSL ICC Ping : Pass
VSL Control Link : Tel/5/5
VSL Encryption : Configured Mode - Off, Operational Mode - Off
```

```
Switch1# show switch virtual link port-channel
Flags:  D - down          P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3        S - Layer2
        U - in use        N - not in use, no aggregation
        f - failed to allocate aggregator

        M - not in use, no aggregation due to minimum links not met
        m - not in use, port not aggregated due to minimum links not met
        u - unsuitable for bundling
        d - default port

        w - waiting to be aggregated

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-------------------
2      Po2(RU)         -         Te1/5/4(P)     Te1/5/5(P)
3      Po3(RU)         -         Te2/5/4(P)     Te2/5/5(P)
```

Redundant Switch Supervisors

Redundancy Features

- Redundancy of Supervise Engines
  - **Route Processor Redundancy**
  - **Route Processor Redundancy+**
  - **Stateful SwitchOver**
  - **Non-Stop Forwarding** with SSO
- Available ONLY
  on Catalyst 4500/6500

Supervisor Redundancy Modes

Redundancy Mode	Behavior When Active Module Fails	Failover Time
RPR	The standby module reloads every other module, initializes all supervisor functions.	> 2 minutes
RPR+	The standby module finishes initializing without reloading other modules.	> 30 seconds
SSO	The standby module is already initialized.	> 1 second

- Redundant supervisor modules can be configured in several modes.
- Redundancy mode limits the standby supervisor's state of readiness.
- SSO allows for NSF.

Route Processor Redundancy (RPR)

- With **RPR**, any of the following events triggers a switchover from the active to the standby Supervisor Engine
  - Route Processor (RP) or Switch Processor (SP) crash on the active Supervisor Engine
  - A manual switchover from the CLI
  - Removal of the active Supervisor Engine
  - Clock synchronization failure between Supervisor Engines
- **RPR+** enhances Supervisor redundancy compared to RPR
  - Reduced switchover time (in the range of 30 seconds to 60 seconds)
  - No reloading of installed modules (Because both the startup configuration and the running configuration stay continually synchronized)
- *RPR is not preferred any longer!*

Configuring and Verifying RPR

- To use RPR and change its mode RPR+ issue following:

```
Router(config)# redundancy
Router(config-red)# mode rpr-plus
```

- Type following command to verify RPR status:

```
Switch# show redundancy states
         my state = 13 -ACTIVE
       peer state = 1 -DISABLED
             Mode = Simplex
             Unit = Primary
          Unit ID = 1
Redundancy Mode (Operational) = Route Processor Redundancy Plus
Redundancy Mode (Configured) = Route Processor Redundancy Plus
       Split Mode = Disabled
     Manual Swact = Disabled Reason: Simplex mode
   Communications = Down Reason: Simplex mode
```

Stateful Switchover (SSO)

- Provides minimal Layer 2 traffic disruption during Supervisor switchover
- Redundant Supervisor starts up in fully initialized state and synchronizes with startup configuration and running configuration of active Supervisor
- Standby Supervisor in SSO mode keeps in sync with active Supervisor for all changes in hardware and software states for features supported via SSO
- *Preferred solution replacing RPR!*

Features Supported by SSO

- On Cat6500 switchover is between 1 to 3 seconds, on Cat4500 it is subsecond
- Protocols that are maintained synchronized by SSO
  - 802.3x (Flow Control)
  - 802.3ad (LACP) and PAgP
  - 802.1X (Authentication) and Port security
  - 802.3af (Inline power)
  - VTP
  - Dynamic ARP Inspection/DHCP snooping/IP source guard
  - IGMP snooping (versions 1 and 2)
  - DTP (802.1Q and ISL)
  - MST/PVST+/Rapid-PVST
  - PortFast/UplinkFast/BackboneFast /BPDU Guard and filtering
  - Voice VLAN
  - Unicast MAC filtering
  - ACL (VLAN ACLs, Port ACLs, Router ACLs)
  - QOS (DBL)
  - Multicast storm control/broadcast storm control
- *Observe that mostly L2 remains synchronized, what about L3?*

Configuring and Verifying SSO

- To use SSO issue following:

```
Router(config)# redundancy
Router(config-red)# mode sso
```

- IF mode is changed THEN standby is reset
- Same command as for RPR could be used to verify SSO:

```
Switch# show redundancy states
          my state = 13 -ACTIVE
        peer state = 8 -STANDBY HOT
              Mode = Duplex
              Unit = Primary
           Unit ID = 2
Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured) = Stateful Switchover
        Split Mode = Disabled
      Manual Swact = Enabled
    Communications = Up
```

Non-Stop Forwarding (NSF) with SSO

- Minimizes time that L3 network is by continuing to forward IP packets using CEF entries built from the old active SE
  - Zero or near zero packet loss
  - Supports BGP, EIGRP, OSPF, and IS-IS
  - Prevents route flapping
- *How is it done?*
  - Adjacencies must not be reset when switchover is complete; otherwise, protocol state is not maintained
  - FIB must remain unchanged during switchover
  - Current routes are marked as stale during restart and routes are refreshed after Cisco NSF convergence is complete
  - Switchover must be completed before dead or hold timer expires; otherwise, peers will reset the adjacency and reroute the traffic
  - Cisco NSF-capable routers are also aware about Cisco NSF-capable neigbours
- *The most preferred solution replacing SSO!*

Configuring NSF

- NSF is an additional configuration option when SSO is enabled
- To configure NSF for OSPF, EIGRP, and IS-IS, use the:

```
Router(config-router)# nsf router-level
```

- To configure BGP for NSF support, use the:

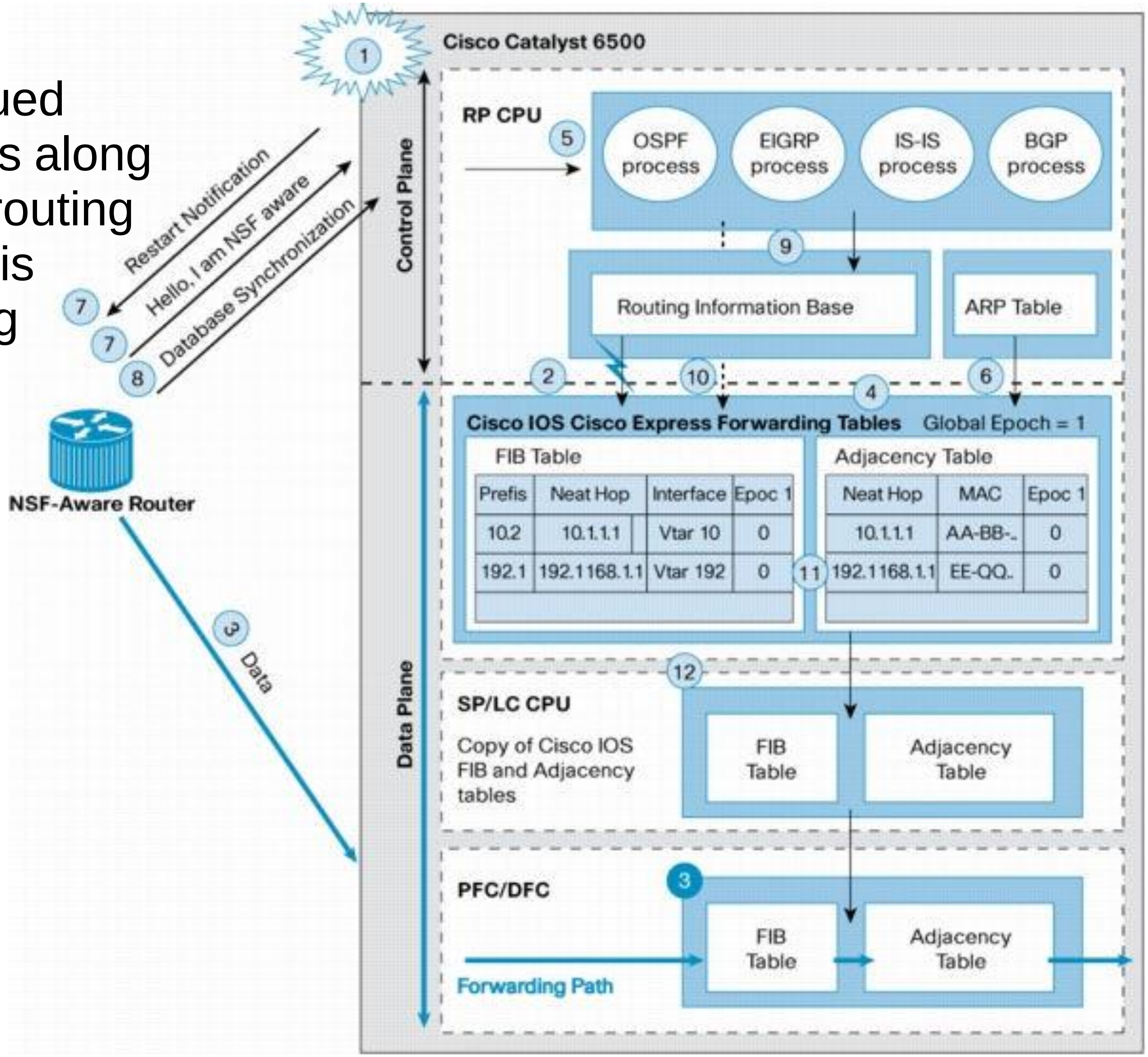
```
Router(config-router)#
   bgp graceful-restart router-level
```

Verifying NSF

```
Switch# show ip bgp neighbors 192.168.200.1
BGP neighbor is 192.168.200.1, remote AS 200, external link
BGP version 4, remote router ID 192.168.200.1
BGP state = Established, up for 00:01:23
Last read 00:00:17, hold time is 180, keepalive interval is 60 seconds
Neighbor capabilities:
   Route refresh:advertised and received(new)
   Address family IPv4 Unicast:advertised and received
   Address family IPv4 Multicast:advertised and received
   Graceful Restart Capability:advertised and received
   Remote Restart timer is 120 seconds
…
Switch# show ip ospf
Routing Process "ospf 200" with ID 192.168.20.1 and Domain ID 0.0.0.1
Supports only single TOS(TOS0) routes
Supports opaque LSA
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 0x0
Number of opaque AS LSA 0. Checksum Sum 0x0
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
External flood list length 0
Non-Stop Forwarding enabled, last NSF restart 00:02:36 ago (took 34 secs)
Area BACKBONE(0)
Number of interfaces in this area is 1 (0 loopback)
```

Routing Protocols and NSF

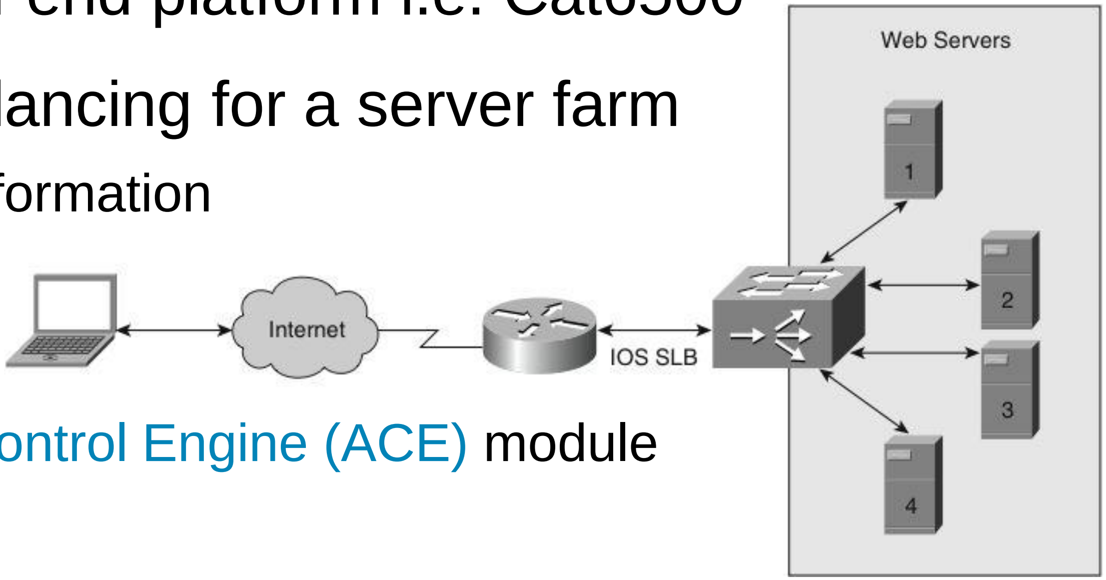
- NSF enables continued forwarding of packets along known routes while routing protocol information is being restored during switchover



IOS Server Load Balancing

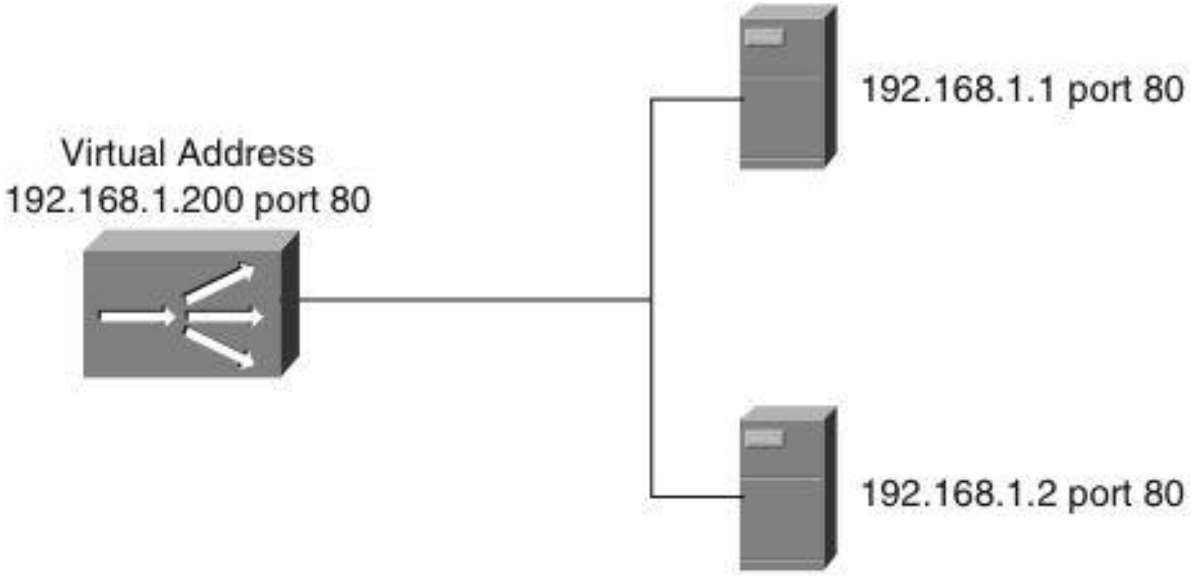
Server Load Balancing

- Available only on high-end platform i.e. Cat6500
- SLB provides load balancing for a server farm
  - According to L4 - L7 information
  - SW
  - HW
    - Cisco Application Control Engine (ACE) module
- Advantages
  - Reducing server load
  - Increased security – real IP address is not visible
  - Reducing downtime (switch detects down servers)



Virtual Server and Server Farm

- Cisco IOS SLB enables users to represent a group of network servers (a server farm in a data center) as a single server instance so called **virtual server**
  - Balance the traffic and limit it to individual servers
  - Any request to virtual server is served by **real servers**



Cisco IOS SLB modes

- **Dispatched mode**
  - Each of the real servers is configured with the virtual server address as a loopback address or secondary IP address
  - Packets are redirected to the real servers at the MAC layer
    - Packet targeted to the virtual IP address is encapsulated into the frame with MAC address corresponding to the real server IP address
  - Servers must be in same network (Layer2 adjacent)
- **Directed mode**
  - Each of the real servers has own real IP address
  - Server does not known virtual IP address of a server farm
  - Packets are translated using NAT

Configuring the Server Farm with Real Servers

1) Define the server farm:

```
Switch(config)# ip slb serverfarm SERVERFARM-NAME
```

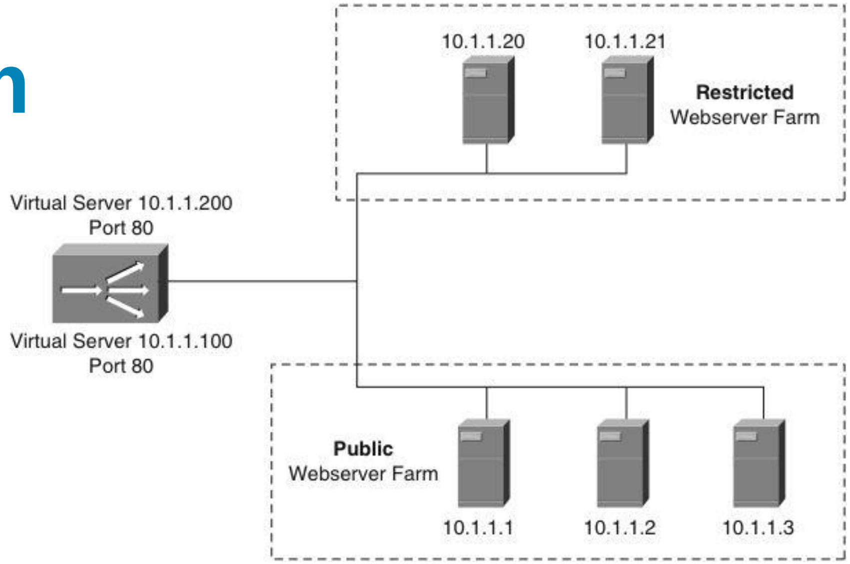
2) Associate the real server with the server farm:

```
Switch(config-slb-sfarm)# real A.B.C.D
```

3) Enable the real server in a server farm:

```
Switch(config-slb-real)# inservice
```

Example: Server Farm

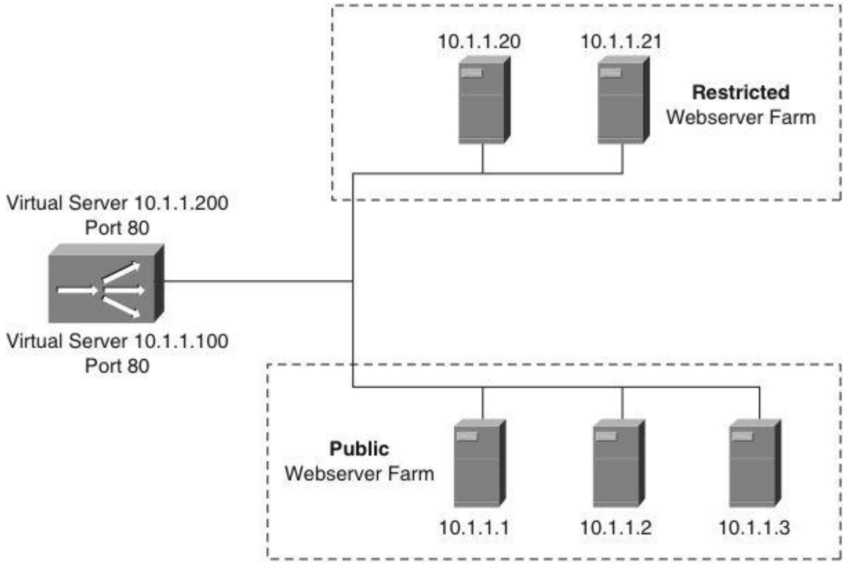


- Two server farms in a data center, PUBLIC and RESTRICTED
- PUBLIC: three real servers: 10.1.1.1, 10.1.1.2 a 10.1.1.3
- RESTRICTED: two real servers: 10.1.1.20 a 10.1.1.21

```
Switch(config)# ip slb serverfarm PUBLIC
Switch(config-slb-sfarm)# nat server ! Directed Mode
Switch(config-slb-sfarm)# real 10.1.1.1
Switch(config-slb-real)# inservice
Switch(config-slb-real)# real 10.1.1.2
Switch(config-slb-real)# inservice
Switch(config-slb-real)# real 10.1.1.3
Switch(config-slb-real)# inservice
!
Switch(config)# ip slb serverfarm RESTRICTED
Switch(config-slb-sfarm)# nat server ! Directed Mode
Switch(config-slb-sfarm)# real 10.1.1.20
Switch(config-slb-real)# inservice
Switch(config-slb-real)# real 10.1.1.21
Switch(config-slb-real)# inservice
```

SLB Verification

- Displaying the status of the server farms
  - Associated servers
  - State of real servers
  - Load balancing mode




```
Switch# show ip slb real
real            farm name       weight  state           cons
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
10.1.1.1        PUBLIC          8       OPERATIONAL     0
10.1.1.2        PUBLIC          8       OPERATIONAL     0
10.1.1.3        PUBLIC          8       OPERATIONAL     0
10.1.1.20       RESTRICTED      8       OPERATIONAL     0
10.1.1.21       RESTRICTED      8       OPERATIONAL     0

Switch# show ip slb serverfarm
server farm     predictor       nat     reals   bind id
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
PUBLIC          ROUNDROBIN      none    3       0
RESTRICTED      ROUNDROBIN      none    2       0
```

Configuring Virtual Servers

1) Define the virtual server:

```
Switch(config)# ip slb vserver vserver-name
```

2) Configure the IP address of the virtual server:

```
Switch(config-slb-vserver)# virtual ip-address [network-mask]
{tcp | udp} [port-number | wsp | wsp-wtp | wsp-wtls | wsp-wtp-wtls]
[service service-name]
```

3) Associate the primary and secondary server farm to the virtual server:

```
Switch(config-slb-vserver)# serverfarm primary-servfarm-name
[backup backup-serverfarm-name [sticky]]
```

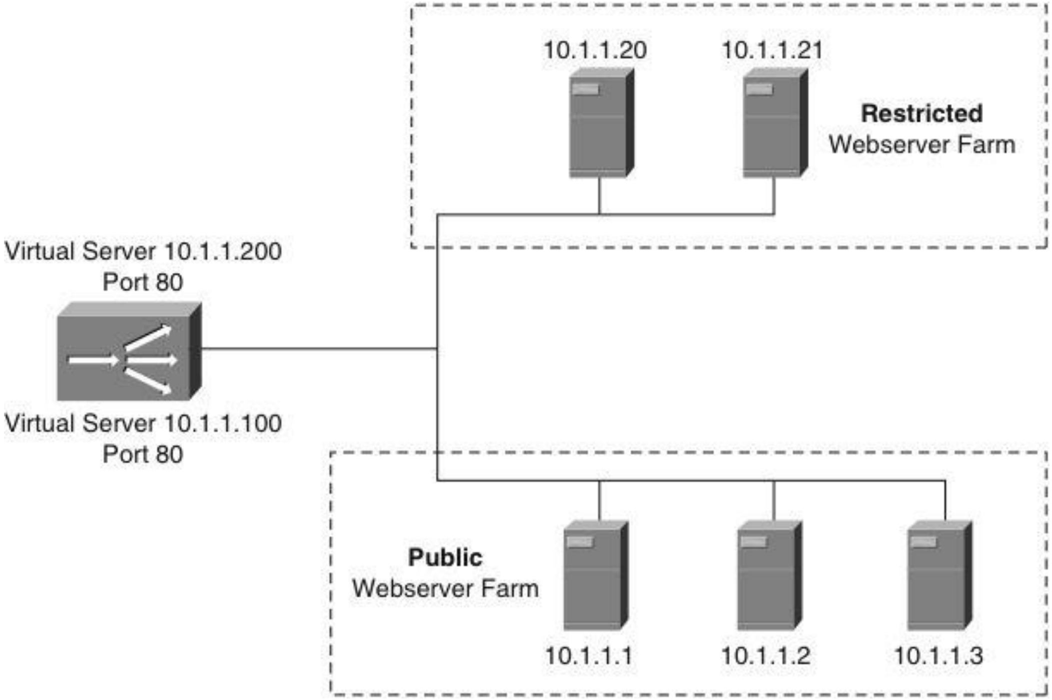
4) Enable the virtual server:

```
Switch(config-slb-vserver)# inservice
```

5) Specify the clients allowed to access the virtual server:

```
Switch(config-slb-vserver)# client ip-address network-mask
```

Example: Virtual Servers




```
Switch(config)# ip slb vserver PUBLIC_HTTP
Switch(config-slb-vserver)# virtual 10.1.1.100 tcp www
Switch(config-slb-vserver)# serverfarm PUBLIC
Switch(config-slb-vserver)# inservice
Switch(config)# ip slb vserver RESTRICTED_HTTP
Switch(config-slb-vserver)# virtual 10.1.1.200 tcp www
Switch(config-slb-vserver)# client 10.4.4.0 255.255.255.0
Switch(config-slb-vserver)# serverfarm RESTRICTED
Switch(config-slb-vserver)# inservice
```

Virtual Server Verification

```
Switch# show ip slb vserver
slb vserver     prot    virtual                 state           cons
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
PUBLIC_HTTP      TCP    10.1.1.100:80   OPERATIONAL    0
RESTRICTED_HTTP TCP     10.1.1.200:80   OPERATIONAL    0


! Check the connections
Switch# show ip slb connections
vserver              prot         client            real            state     nat
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
RESTRICTED_HTTP     TCP          10.4.4.0:80      10.1.1.20       CLOSING    none
```

Troubleshooting

- Display detailed info Information for an SLB Client

show ip slb connections client

- Display the statistics

show ip slb stats

```
Switch# show ip slb connections client 10.4.4.0 detail
VSTEST_UDP, client = 10.4.4.0:80
state = CLOSING, real = 10.1.1.20, nat = none
v_ip = 10.1.1.200:80, TCP, service = NONE
client_syns = 0, sticky = FALSE, flows attached = 0

Switch# show ip slb stats
Pkts via normal switching: 0
Pkts via special switching: 6
Connections Created: 1
Connections Established: 1
Connections Destroyed: 0
Connections Reassigned: 0
Zombie Count: 0
Connections Reused: 0
```

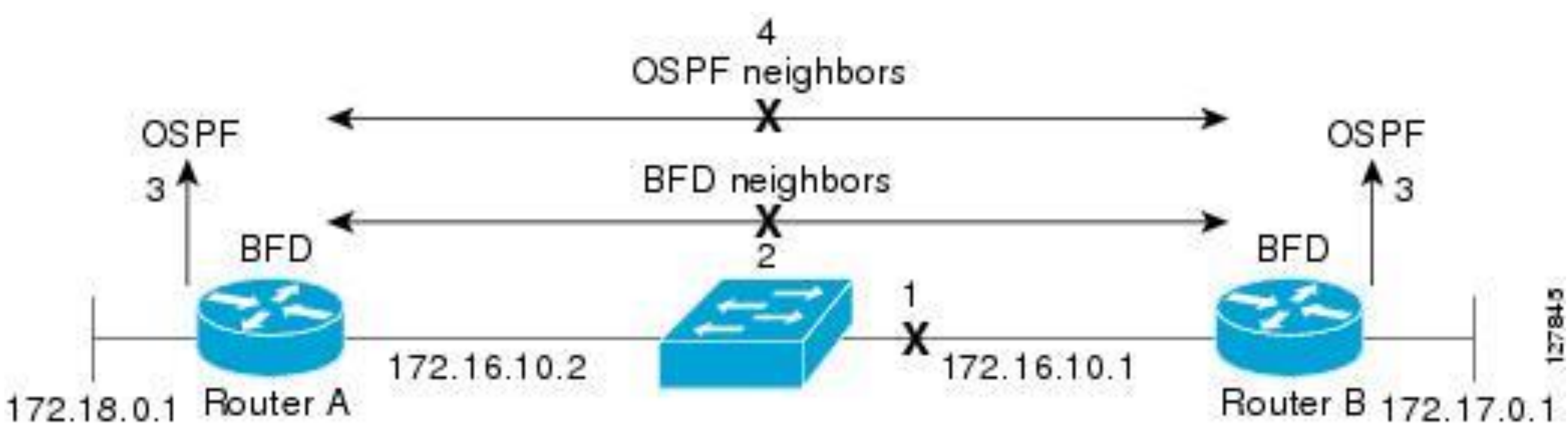
Bidirectional Forward Detection

Bidirectional Forwarding Detection

- RFC 5880
- **Bidirectional Forwarding Detection (BFD)** provides a low-overhead, short-duration method of detecting failures in the forwarding path between two adjacent routers
- Once a BFD session has been established and timer negations are complete, BFD peers send BFD control packets that act in the same manner as an IGP hello protocol to detect liveliness
- BFDv0 and BFDv1 do exist, both supported on Cisco boxes
- Prerequisites
  - CEF and IP routing enabled on all BFD neighbors
  - Each routing protocol MUST be configured to benefit from BFD
- “Bidirectional Forwarding Detection”, Cisco IOS Release 12.2SR

Features

- BFD detects a failure, but the IGP/BGP/FHRP must take action to bypass a failed peer
- BFD can provide failure detection in less than one second
  - Reducing the IGP/BGP/FHRP timers can result in minimum detection timer of one to two seconds
- BFD can be used as a generic and consistent failure detection mechanism
- BFD can be less CPU-intensive
  - Some parts of BFD can be distributed to the data plane
  - Reduced IGP/BGP/FHRP timers exist wholly at the control plane



Configuration

- On interface issue following command:

```
Router(config-if)# bfd interval send-timer
   min_rx receive-timer multiplier interval-multiplier
```

- **`interval`**: period between two consecutive BFD control messages
- **`min_rx`**: minimum interval between packets accepted from BFD peers
- **`multiplier`**: specifies the minimum number of consecutive packets that can be missed before a BFD session is declared down and neighbor dead (default is 3)

Supported Protocols

```
(conf-router)# bfd all-interfaces
```

- **IGP**
  - EIGRP
    ```
    (conf-router)# bfd interface
    ```
  - OSPF
    ```
    (conf-if)# ip ospf bfd [disable]
    ```
  - IS-IS
    ```
    (conf-if)# isis bfd [disable]
    ```
- **EGP**
  - BGP
    ```
    (conf-router)# neighbor ip-address fall-over bfd
    ```
- **FHRP**
  - HSRP
    ```
    (conf-if)# standby bfd
    ```
  - VRRP
    ```
    (conf-if)# vrrp bfd
    ```
- PIM

Verifying

show ip bfd neighbors [detail]

```
R1# show bfd neighbor
OurAddr    NeighAddr LD/RD RH/RS   Holddown(mult) State   Int
10.1.3.1   10.1.3.3  1/2     Up         0     (3 )  Up      Fa0/1
```

Example

```
R1# show ip ospf neighbor

NeighborID   Pri   State       Dead Time   Address    Interface
2.2.2.2      1     FULL/DR     00:00:37    10.1.2.2   FastEthernet0/0
3.3.3.3      1     FULL/BDR    00:00:37    10.1.3.3   FastEthernet0/1

R1(config)# int fa 0/0
R1(config-if)# sh

19:52:13.115: %LINK-5-CHANGED: Interface FastEthernet0/0, changed state to
administratively down

R2#
19:52:42.643: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on FastEthernet0/0 from
FULL to DOWN, Neighbor Down: Dead timer expired

…

R1(config)#int fa 0/1
R1(config-if)#shut

20:04:10.204: %OSPF-5-ADJCHG: Process 1, Nbr on FastEthernet0/1 from FULL to
DOWN, Neighbor Down: Interface down or detached

20:04:12.202: %LINK-5-CHANGED: Interface FastEthernet0/1, changed state to
administratively down

R3#

20:04:10.511: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on FastEthernet0/1 from
FULL to DOWN, Neighbor Down: BFD node down
```

Slides adapted by Matěj Grégr and tuned by Vladimír Veselý
partially from official course materials
but the most of the credit goes to CCIE#23527 Ing. Peter Palúch, Ph.D.

The last update: 2016-11-02