CHAPTER 3

Medianet Availability Design Considerations

The goal of network availability technologies is to maximize network uptime such that the network is always ready and able to provide needed services to critical applications, such as TelePresence or other critical network video.

Network video has varying availability requirements. At one extreme, if a single packet is lost, the user likely notices an artifact in the video. One the other extreme, video is a unidirectional session; the camera always sends packets and the display always receives packets. When an outage occurs, the camera may not recognize it, and continue to send video packets. Upper layer session control protocols, such as Session Initiation Protocol (SIP) and Real-Time Streaming Protocol (RTSP), are responsible to validate the path. Video applications may respond differently to session disruptions. In all cases, the video on the display initially freezes at the last received frame, and looks to the session control for some resolution. If the packet stream is restored, quite often the video recovers without having to restart the session. TelePresence can recover after a 30-second network outage before SIP terminates the call. Broadcast video may be able to go longer. Availability techniques should be deployed such that the network converges faster than the session control protocol hello interval. The user notices that the video has frozen, but in most cases, the stream recovers without having to restart the media.

Network Availability

Network availability is the cornerstone of network design, on which all other services depend.

The three primary causes of network downtime are as follows:

- Hardware failures, which can include system and sub-component failures, as well as power failures and network link failures
- Software failures, which can include incompatibility issues and bugs
- Operational processes, which mainly include human error; however, poorly-defined management and upgrading processes may also contribute to operational downtime

To offset these types of failures, the network administrator attempts to provision the following types of resiliency:

- Device resiliency—Deploying redundant hardware (including systems, supervisors, line cards, and power-supplies) that can failover in the case of hardware and/or software failure events
- Network resiliency—Tuning network protocols to detect and react to failure events as quickly as possible
- Operational resiliency—Examining and defining processes to maintain and manage the network, leveraging relevant technologies that can reduce downtime, including provisioning for hardware and software upgrades with minimal downtime (or optimally, with no downtime)

Note Because the purpose of this overview of availability technologies is to provide context for the design chapters to follow, this discussion focuses on device and network resiliency, rather than operational resiliency.

Network availability can be quantitatively measured by using the formula shown in Figure 3-1, which correlates the mean time between failures (MTBF) and the mean time to repair (MTTR) such failures.

Figure 3-1 Availability Formula

$$Availability = \frac{MTBF}{MTBF + MRRT}$$

228666

For example, if a network device has an MTFB of 10,000 hours and an MTTR of 4 hours, its availability can be expressed as 99.96 percent[(10,000)/(10,000 + 4), converted to a percentage].

Therefore, from this formula it can be seen that availability can be improved by either increasing the MTBF of a device (or network), or by decreasing the MTTR of the same.

The most effective way to increase the MTBF of a device (or network) is to design with redundancy. This can be mathematically proven by comparing the availability formula of devices connected in serial (without redundancy) with the formula of devices connected in parallel (with redundancy).

The availability of devices connected in series is shown in Figure 3-2.

Figure 3-2 Availability Formula for Devices Connected in Serial

S_1, S_2 - Series Components

System is available when both components are available:

$$A_{series} = A_1 \times A_2$$

223825

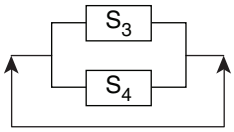
S1 and *S2* represent two separate systems (which may be individual devices or even networks). *A1* and *A2* represent the availability of each of these systems, respectively. *Aseries* represents the overall availability of these systems connected in serial (without redundancy).

For example, if the availability of the first device (S1) is 99.96 percent and the availability of the second device (S2) is 99.98 percent, the overall system availability, with these devices connected serially, is 99.94 percent (99.96% x 99.98%).

Therefore, connecting devices in serial actually reduces the overall availability of the network.

In contrast, consider the availability of devices connected in parallel, as shown in Figure 3-3.

Figure 3-3 Availability Formula for Devices Connected in Parallel



S_3, S_4 - Parallel Components

System is unavailable when both components are unavailable:

$$A_{parallel} = 1 - (1 - A_1) \times (1 - A_2)$$

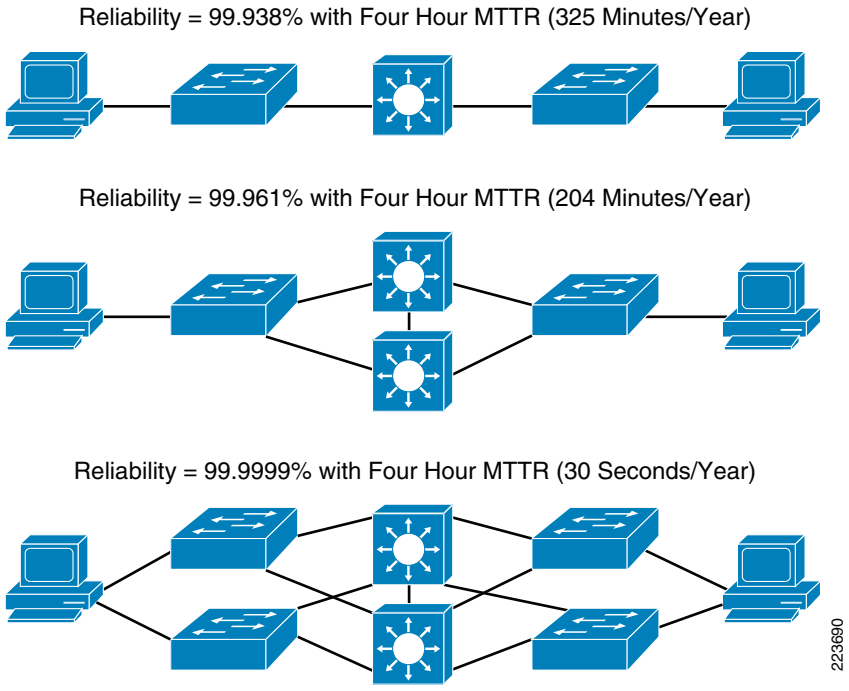
223826

S3 and *S4* represent two separate systems (devices or networks). *A3* and *A4* represent the availability of each of these systems, respectively. *Aparallel* represents the overall availability of these systems connected in parallel (with redundancy).

Continuing the example, using the same availability numbers for each device as before yields an overall system availability, with these devices connected in parallel, of 99.999992 percent [1-(1-99.96%) * (1-99.98%)].

Therefore, connecting devices in parallel significantly increases the overall availability of the combined system. This is a foundational principle of available network design, where individual devices as well as networks are designed to be fully redundant, whenever possible. Figure 3-4 illustrates applying redundancy to network design and its corresponding effect on overall network availability.

Figure 3-4 Impact of Redundant Network Design on Network Availability



A *five nines* network (a network with 99.999 percent availability) has been considered the hallmark of excellent enterprise network design for many years. However, a five nines network allows for only five minutes of downtime per year.

Another commonly used metric for measuring availability is defects per million (DPM). Measuring the probability of failure of a network and establishing the service-level agreement (SLA) that a specific design is able to achieve is a useful tool, but DPM takes a different approach by measuring the impact of defects on the service from the end-user perspective. This is often a better metric for determining the availability of the network because it better reflects the user experience relative to event effects. DPM is calculated based on taking the total affected user minutes for each event, total users affected, and the duration of the event, as compared to the total number of service minutes available during the period in question. The sum of service downtime minutes is divided by the total service minutes and multiplied by 1,000,000, as shown in Figure 3-5.

Figure 3-5 Defects Per Million Calculation

$$DPM = \frac{\sum(\text{number of users affected} * \text{Outage Minutes})}{\text{Total Users} * \text{Total Service Minutes}}$$

228667

For example, if a company of 50 employees suffers two separate outages during the course of a year, with the first outage affecting 12 users for 4 hours and the second outage affecting 25 users for 2 hours, the total DPM is 224 [[[(12 users x 240 min)+(25 users x 120 min)]/(50 users x 525,960 min/year)]x 1,000,000, rounded].

Note The benefit of using a “per-million” scale in a defects calculation is that it allows the final ratio to be more readable, given that this ratio becomes extremely small as availability improves.

DPM is useful because it is a measure of the observed availability and considers the impact to the end user as well as the network itself. Adding this user experience element to the question of network availability is very important to understand, and is becoming a more important part of the question of what makes a highly available network.

Table 3-1 summarizes the availability targets, complete with their DPM and allowable downtime/year.

Table 3-1 Availability, DPM, and Downtime

Availability (Percent)	DPM	Downtime/Year
99.000	10,000	3 days, 15 hours, 36 minutes
99.500	5,000	1 day, 19 hours, 48 minutes
99.900	1,000	8 hours, 46 minutes
99.950	500	4 hours, 23 minutes
99.990	100	53 minutes
99.999	10	5 minutes
99.9999	1	0.5 minutes

Having reviewed these availability principles, metrics, and targets, the next section discusses some of the availability technologies most relevant for systems and networks supporting TelePresence systems.

Device Availability Technologies

Every network design has single points of failure, and the overall availability of the network might depend on the availability of a single device. The access layer of a campus network is a prime example of this. Every access switch represents a single point of failure for all the attached devices (assuming that the endpoints are single-homed; this does not apply to endpoint devices that are dual-homed). Ensuring the availability of the network services often depends on the resiliency of the individual devices.

Device resiliency, as with network resiliency, is achieved through a combination of the appropriate level of physical redundancy, device hardening, and supporting software features. Studies indicate that most common failures in campus networks are associated with Layer 1 failures, from components such as power supplies, fans, and fiber links. The use of diverse fiber paths with redundant links and line cards, combined with fully redundant power supplies and power circuits, are the most critical aspects of device resiliency. The use of redundant power supplies becomes even more critical in access switches with the introduction of power over Ethernet (PoE) devices such as IP phones. Multiple devices now depend on the availability of the access switch and its ability to maintain the necessary level of power for all the attached end devices. After physical failures, the most common cause of device outage is often related to the failure of supervisor hardware or software. The network outages caused by the loss or reset of a device because of supervisor failure can be addressed through the use of supervisor redundancy. Cisco Catalyst switches provide the following mechanisms to achieve this additional level of redundancy:

- Cisco StackWise and Cisco StackWise-Plus
- Cisco non-stop forwarding (NSF) with stateful switchover (SSO)

Both these mechanisms, which are discussed in the following sections, provide for a hot active backup for the switching fabric and control plane, thus ensuring that data forwarding and the network control plane seamlessly recover (with sub-second traffic loss, if any) during any form of software or supervisor hardware crash.

Cisco StackWise and Cisco StackWise Plus

Cisco StackWise and Cisco StackWise Plus technologies are used to create a unified, logical switching architecture through the linkage of multiple, fixed configuration Cisco Catalyst 3750G and/or Cisco Catalyst 3750E switches.

Cisco Catalyst 3750G switches use StackWise technology and Cisco Catalyst 3750E switches can use either StackWise or StackWise Plus. StackWise Plus is used only if all switches within the group are 3750E switches; whereas, if some switches are 3750E and others are 3750G, StackWise technology is used.

Note "StackWise" is used in this section to refer to both Cisco StackWise and Cisco StackWise Plus technologies, with the exception of explicitly pointing out the differences between the two at the end of this section.

Cisco StackWise technology intelligently joins individual switches to create a single switching unit with a 32-Gbps switching stack interconnect. 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 availability.

The switches are united into a single logical unit using special stack interconnect cables that create a bidirectional closed-loop path. This bidirectional path acts as a switch fabric for all the connected switches. Network topology and routing information are updated continuously through the stack interconnect. All stack members have full access to the stack interconnect bandwidth. The stack is managed as a single unit by a master switch, which is elected from one of the stack member switches.

Each switch in the stack has the capability to behave as a master in the hierarchy. The master switch is elected and serves as the control center for the stack. Each switch is assigned a number. Up to nine separate switches can be joined together.

Each stack of Cisco Catalyst 3750 Series Switches has a single IP address and is managed as a single object. This single IP management applies to activities such as fault detection, VLAN creation and modification, security, and quality of service (QoS) controls. Each stack has only one configuration file, which is distributed to each member in the stack. This allows each switch in the stack to share the same network topology, MAC address, and routing information. In addition, this allows for any member to immediately take over as the master, in the event of a master failure.

To efficiently load balance the traffic, packets are allocated between two logical counter-rotating paths. Each counter-rotating path supports 16 Gbps in both directions, yielding a traffic total of 32 Gbps bidirectionally. When a break is detected in a cable, the traffic is immediately wrapped back across the single remaining 16-Gbps path (within microseconds) to continue forwarding.

Switches can be added and deleted to a working stack without affecting stack availability. However, adding additional switches to a stack may have QoS performance implications, as is discussed in more in Chapter 4, "Medianet QoS Design Considerations." Similarly, switches can be removed from a working stack with no operational effect on the remaining switches.

Stacks require no explicit configuration, but are automatically created by StackWise when individual switches are joined together with stacking cables, as shown in Figure 3-6. When the stack ports detect electromechanical activity, each port starts to transmit information about its switch. When the complete set of switches is known, the stack elects one of the members to be the master switch, which becomes responsible for maintaining and updating configuration files, routing information, and other stack information.

Figure 3-6 Cisco Catalyst 3750G StackWise Cabling

Each switch in the stack can serve as a master, creating a 1:N availability scheme for network control. In the unlikely event of a single unit failure, all other units continue to forward traffic and maintain operation. Furthermore, each switch is initialized for routing capability and is ready to be elected as master if the current master fails. Subordinate switches are not reset so that Layer 2 forwarding can continue uninterrupted.

The following are the three main differences between StackWise and StackWise Plus:

- StackWise uses source stripping and StackWise Plus uses destination stripping (for unicast packets). Source stripping means that when a packet is sent on the ring, it is passed to the destination, which copies the packet, and then lets it pass all the way around the ring. When the packet has traveled all

the way around the ring and returns to the source, it is stripped off the ring. This means bandwidth is used up all the way around the ring, even if the packet is destined for a directly attached neighbor. Destination stripping means that when the packet reaches its destination, it is removed from the ring and continues no further. This leaves the rest of the ring bandwidth free to be used. Thus, the throughput performance of the stack is multiplied to a minimum value of 64 Gbps bidirectionally. This ability to free up bandwidth is sometimes referred to as spatial reuse.

Note Even in StackWise Plus, broadcast and multicast packets must use source stripping because the packet may have multiple targets on the stack.

- StackWise Plus can locally switch, whereas StackWise cannot. Furthermore, in StackWise, because there is no local switching and there is source stripping, even locally destined packets must traverse the entire stack ring.
- StackWise Plus supports up to two Ten Gigabit Ethernet ports per Cisco Catalyst 3750-E.

Finally, both StackWise and StackWise Plus can support Layer 3 non-stop forwarding (NSF) when two or more nodes are present in a stack.

Non-Stop Forwarding with Stateful Switch Over

Stateful switchover (SSO) is a redundant route- and/or switch-processor availability feature that significantly reduces MTTR by allowing extremely fast switching between the main and backup processors. SSO is supported on routers (such as the Cisco 7600, 10000, and 12000 Series) and switches (such as the Cisco Catalyst 4500 and 6500 Series).

Before discussing the details of SSO, a few definitions may be helpful. For example, *state* in SSO refers to maintaining between the active and standby processors, among many other elements, the protocol configurations and current status of the following:

- Layer 2 (L2)
- Layer 3 (L3)
- Multicast
- QoS policy
- Access list policy
- Interface

Also, the adjectives *cold, warm*, and *hot* are used to denote the readiness of the system and its components to assume the network services functionality and the job of forwarding packets to their destination. These terms appear in conjunction with Cisco IOS verification command output relating to NSF/SSO, as well as with many high availability feature descriptions. These terms are generally defined as follows:

- Cold—The minimum degree of resiliency that has been traditionally provided by a redundant system. A redundant system is cold when no state information is maintained between the backup or standby system and the system to which it offers protection. Typically, a cold system must complete a boot process before it comes online and is ready to take over from a failed system.
- Warm—A degree of resiliency beyond the cold standby system. In this case, the redundant system has been partially prepared, but does not have all the state information known by the primary system to take over immediately. Additional information must be determined or gleaned from the traffic flow or the peer network devices to handle packet forwarding. A warm system is already booted and needs to learn or generate only the state information before taking over from a failed system.

- Hot—The redundant system is fully capable of handling the traffic of the primary system. Substantial state information has been saved, so the network service is continuous, and the traffic flow is minimally or not affected.

To better understand SSO, it may be helpful to consider its operation in detail within a specific context, such as within a Cisco Catalyst 6500 with two supervisors per chassis.

The supervisor engine that boots first becomes the active supervisor engine. The active supervisor is responsible for control plane and forwarding decisions. The second supervisor is the standby supervisor, which does not participate in the control or data plane decisions. The active supervisor synchronizes configuration and protocol state information to the standby supervisor, which is in a hot standby mode. As a result, the standby supervisor is ready to take over the active supervisor responsibilities if the active supervisor fails. This take-over process from the active supervisor to the standby supervisor is referred to as a *switchover.*

Only one supervisor is active at a time, and supervisor engine redundancy does not provide supervisor engine load balancing. However, the interfaces on a standby supervisor engine are active when the supervisor is up and thus can be used to forward traffic in a redundant configuration.

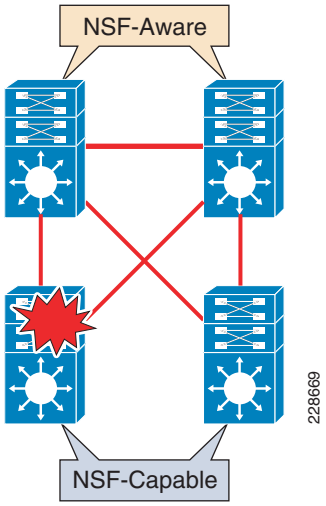
NSF/SSO evolved from a series of progressive enhancements to reduce the impact of MTTR relating to specific supervisor hardware/software network outages. NSF/SSO builds on the earlier work known as Route Processor Redundancy (RPR) and RPR Plus (RPR+). Each of these redundancy modes of operation incrementally improves on the functions of the previous mode.

- RPR-RPR is the first redundancy mode of operation introduced in Cisco IOS Software. In RPR mode, the startup configuration and boot registers are synchronized between the active and standby supervisors, the standby is not fully initialized, and images between the active and standby supervisors do not need to be the same. Upon switchover, the standby supervisor becomes active automatically, but it must complete the boot process. In addition, all line cards are reloaded and the hardware is reprogrammed. Because the standby supervisor is *cold*, the RPR switchover time is two or more minutes.
- RPR+-RPR+ is an enhancement to RPR in which the standby supervisor is completely booted and line cards do not reload upon switchover. The running configuration is synchronized between the active and the standby supervisors. All synchronization activities inherited from RPR are also performed. The synchronization is done before the switchover, and the information synchronized to the standby is used when the standby becomes active to minimize the downtime. No link layer or control plane information is synchronized between the active and the standby supervisors. Interfaces may bounce after switchover, and the hardware contents need to be reprogrammed. Because the standby supervisor is *warm*, the RPR+ switchover time is 30 or more seconds.
- NSF with SSO-NSF works in conjunction with SSO to ensure Layer 3 integrity following a switchover. It allows a router experiencing the failure of an active supervisor to continue forwarding data packets along known routes while the routing protocol information is recovered and validated. This forwarding can continue to occur even though peering arrangements with neighbor routers have been lost on the restarting router. NSF relies on the separation of the control plane and the data plane during supervisor switchover. The data plane continues to forward packets based on pre-switchover Cisco Express Forwarding information. The control plane implements graceful restart routing protocol extensions to signal a supervisor restart to NSF-aware neighbor routers, reform its neighbor adjacencies, and rebuild its routing protocol database (in the background) following a switchover. Because the standby supervisor is *hot*, the NSF/SSO switchover time is 0–3 seconds.

  As previously described, neighbor nodes play a role in NSF function. A node that is capable of continuous packet forwarding during a route processor switchover is NSF-capable. Complementing this functionality, an NSF-aware peer router can enable neighbor recovery without resetting adjacencies, and support routing database re-synchronization to occur in the background. Figure 3-5 illustrates the difference between NSF-capable and NSF-aware routers. To gain the greatest benefit from NSF/SSO deployment, NSF-capable routers should be peered with NSF-aware routers

(although this is not absolutely required for implementation), because only a limited benefit is achieved unless routing peers are aware of the ability of the restarting node to continue packet forwarding and assist in restoring and verifying the integrity of the routing tables after a switchover.

Figure 3-7 NSF-Capable Compared to NSF-Aware Routers



Cisco Nonstop Forwarding and Stateful Switchover are designed to be deployed together. NSF relies on SSO to ensure that links and interfaces remain up during switchover, and that the lower layer protocol state is maintained. However, it is possible to enable SSO with or without NSF, because these are configured separately.

The configuration to enable SSO is very simple, as follows:

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

NSF, on the other hand, is configured within the routing protocol itself, and is supported within Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), Intermediate System to Intermediate System (IS-IS), and (to an extent) Border Gateway Protocol (BGP). Sometimes NSF functionality is also called *graceful-restart*.

To enable NSF for EIGRP, enter the following commands:

```
Router(config)# router eigrp 100
Router(config-router)# nsf
```

Similarly, to enable NSF for OSPF, enter the following commands:

```
Router(config)# router ospf 100
Router(config-router)# nsf
```

Continuing the example, to enable NSF for IS-IS, enter the following commands:

```
Router(config)#router isis level2
Router(config-router)#nsf cisco
```

And finally, to enable NSF/graceful-restart for BGP, enter the following commands:

```
Router(config)#router bgp 100
Router(config-router)#bgp graceful-restart
```

You can see from the example of NSF that the line between device-level availability technologies and network availability technologies is sometimes uncertain. A discussion of more network availability technologies follows.

Network Availability Technologies

Network availability technologies, which include link integrity protocols, link bundling protocols, loop detection protocols, first-hop redundancy protocols (FHRPs) and routing protocols, are used to increase the resiliency of devices connected within a network. Network resiliency relates to how the overall design implements redundant links and topologies, and how the control plane protocols are optimally configured to operate within that design. The use of physical redundancy is a critical part of ensuring the availability of the overall network. In the event of a network device failure, having a path means that the overall network can continue to operate. The control plane capabilities of the network provide the ability to manage the way in which the physical redundancy is leveraged, the network load balances traffic, the network converges, and the network is operated.

The following basic principles can be applied to network availability technologies:

- Wherever possible, leverage the ability of the device hardware to provide the primary detection and recovery mechanism for network failures. This ensures both a faster and a more deterministic failure recovery.
- Implement a defense-in-depth approach to failure detection and recovery mechanisms. Multiple protocols, operating at different network layers, can complement each other in detecting and reacting to network failures.
- Ensure that the design is self-stabilizing. Use a combination of control plane modularization to ensure that any failures are isolated in their impact and that the control plane prevents flooding or thrashing conditions from arising.

These principles are intended to complement the overall structured modular design approach to the network architecture and to re-enforce good resilient network design practices.

Note A complete discussion of all network availability technologies and best practices could easily fill an entire volume. Therefore, this discussion introduces only an overview of the most relevant network availability technologies for TelePresence enterprise network deployments.

The following sections discuss L2 and L3 network availability technologies.

L2 Network Availability Technologies

L2 network availability technologies that particularly relate to TelePresence network design include the following:

- Unidirectional Link Detection (UDLD)
- IEEE 802.1d Spanning Tree Protocol (STP)
- Cisco Spanning Tree Enhancements
- IEEE 802.1w Rapid Spanning Tree Protocol (RSTP)
- Trunks, Cisco Inter-Switch Link, and IEEE 802.1Q
- EtherChannels, Cisco Port Aggregation Protocol, and IEEE 802.3ad

- Cisco Virtual Switching System (VSS)

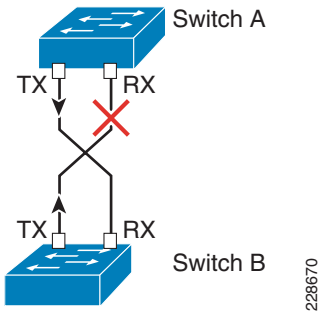
Each of these L2 technologies are discussed in the following sections.

UniDirectional Link Detection

UDLD protocol is a Layer 2 protocol, which uses a keep-alive to test that the switch-to-switch links are connected and operating correctly. Enabling UDLD is a prime example of how a defense-in-depth approach to failure detection and recovery mechanisms can be implemented, because UDLD (an L2 protocol) acts as a backup to the native Layer 1 unidirectional link detection capabilities provided by IEEE 802.3z (Gigabit Ethernet) and 802.3ae (Ten Gigabit Ethernet) standards.

The UDLD protocol allows devices connected through fiber optic or copper Ethernet cables connected to LAN ports to monitor the physical configuration of the cables and detect when a unidirectional link exists. When a unidirectional link is detected, UDLD shuts down the affected LAN port and triggers an alert. Unidirectional links, such as shown in Figure 3-8, can cause a variety of problems, including spanning tree topology loops.

Figure 3-8 Unidirectional Link Failure



You can configure UDLD to be globally enabled on all fiber ports by entering the following command:

```
Switch(config)#udld enable
```

Additionally, you can enable UDLD on individual LAN ports in interface mode, by entering the following commands:

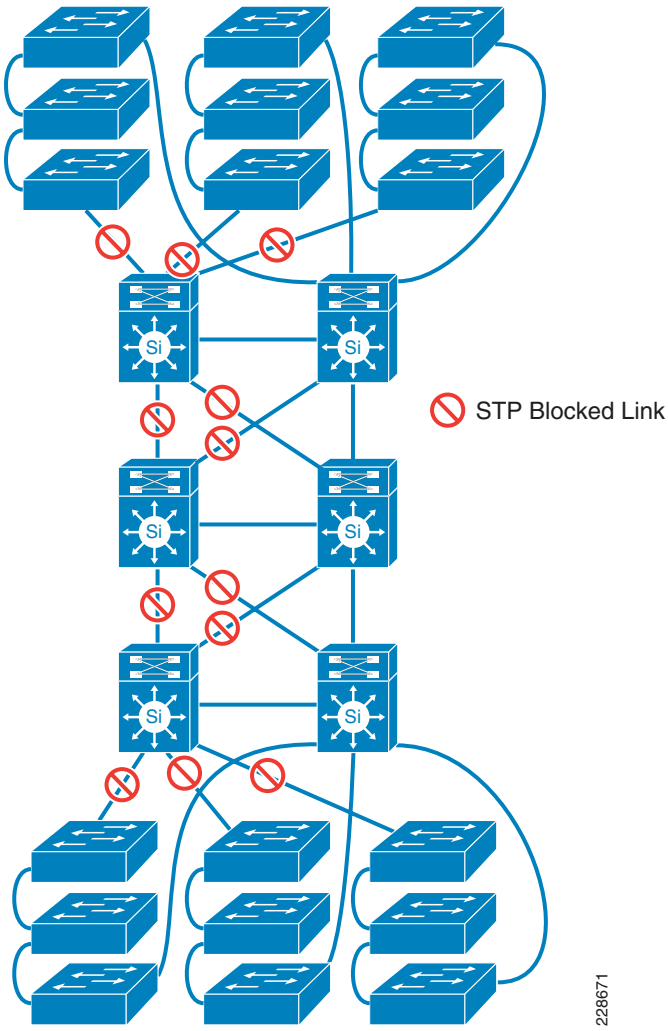
```
Switch(config)#interface GigabitEthernet8/1
Switch(config-if)#udld port
```

Interface configurations override global settings for UDLD.

IEEE 802.1D Spanning Tree Protocol

IEEE 802.1D STP prevents loops from being formed when switches are interconnected via multiple paths. STP implements the spanning tree algorithm by exchanging Bridge Protocol Data Unit (BPDU) messages with other switches to detect loops, and then removes the loop by shutting down selected switch interfaces. This algorithm guarantees that there is only one active path between two network devices, as illustrated in Figure 3-9.

Figure 3-9 STP-Based Redundant Topology

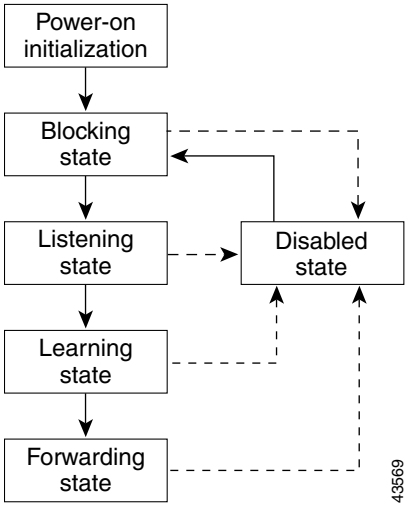


STP prevents a loop in the topology by transitioning all (STP-enabled) ports through four STP states:

- Blocking—The port does not participate in frame forwarding. STP can take up to 20 seconds (by default) to transition a port from blocking to listening.
- Listening—The port transitional state after the blocking state when the spanning tree determines that the interface should participate in frame forwarding. STP takes 15 seconds (by default) to transition between listening and learning.
- Learning—The port prepares to participate in frame forwarding. STP takes 15 seconds (by default) to transition from learning to forwarding (provided such a transition does not cause a loop; otherwise, the port is be set to blocking).
- Forwarding—The port forwards frames.

Figure 3-10 illustrates the STP states, including the disabled state.

Figure 3-10 STP Port States



You can enable STP globally on a per-VLAN basis, using Per-VLAN Spanning-Tree (PVST), by entering the following command:

```
Switch(config)# spanning-tree vlan 100
```

The two main availability limitations for STP are as follows:

- To prevent loops, redundant ports are placed in a blocking state and as such are not used to forward frames/packets. This significantly reduces the advantages of redundant network design, especially with respect to network capacity and load sharing.
- Adding up all the times required for STP port-state transitions shows that STP can take up to 50 seconds to converge on a loop-free topology. Although this may have been acceptable when the protocol was first designed, it is certainly unacceptable today.

Both limitations are addressable using additional technologies. The first limitation can be addressed by using the Cisco Virtual Switching System (VSS), discussed later in this section; and the second limitation can be addressed by various enhancements that Cisco developed for STP, as is discussed next.

Cisco Spanning Tree Enhancements

To improve STP convergence times, Cisco has made a number of enhancements to 802.1D STP, including the following:

- PortFast (with BPDU Guard)
- UplinkFast
- BackboneFast

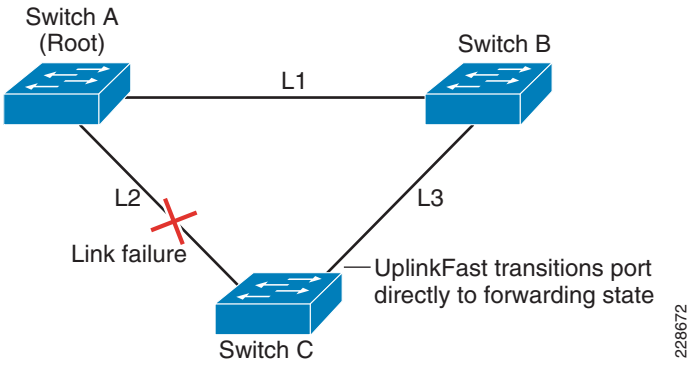
STP PortFast causes a Layer 2 LAN port configured as an access port to enter the forwarding state immediately, bypassing the listening and learning states. PortFast can be used on Layer 2 access ports connected to a single workstation or server to allow those devices to connect to the network immediately, instead of waiting for STP to converge, because interfaces connected to a single workstation or server should not receive BPDUs. Because the purpose of PortFast is to minimize the time that access ports must wait for STP to converge, it should only be used on access ports. Optionally, for an additional level of security, PortFast may be enabled with BPDU Guard, which immediately shuts down a port that has received a BPDU.

You can enable PortFast globally (along with BPDU Guard), or on a per-interface basis, by entering the following commands:

```
Switch(config)# spanning-tree portfast default
Switch(config)# spanning-tree portfast bpduguard default
```

UplinkFast provides fast convergence after a direct link failure and achieves load balancing between redundant Layer 2 links, as shown in Figure 3-11. If a switch detects a link failure on the currently active link (a direct link failure), UplinkFast unblocks the blocked port on the redundant link port and immediately transitions it to the forwarding state without going through the listening and learning states. This switchover takes approximately one to five seconds.

Figure 3-11 UplinkFast Recovery Example After Direct Link Failure

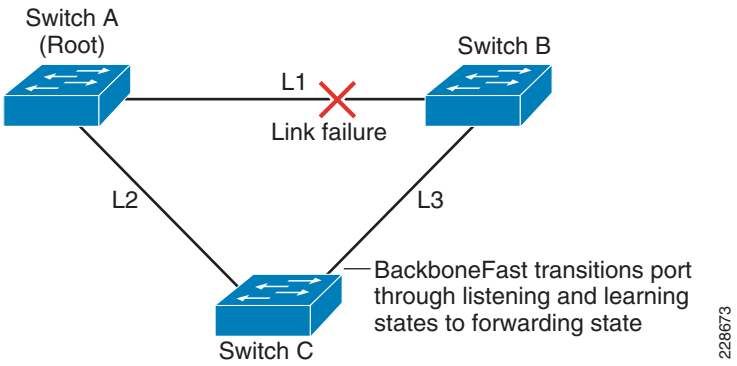


UplinkFast is enabled globally, as follows:

```
Switch(config)# spanning-tree uplinkfast
```

In contrast, BackboneFast provides fast convergence after an indirect link failure, as shown in Figure 3-12. This switchover takes approximately 30 seconds (yet improves on the default STP convergence time by 20 seconds).

Figure 3-12 BackboneFast Recovery Example After Indirect Link Failure



BackboneFast is enabled globally, as follows:

```
Switch(config)# spanning-tree backbonefast
```

These Cisco-proprietary enhancements to 802.1D STP were adapted and adopted into a new standard for STP, IEEE 802.1w or Rapid Spanning-Tree Protocol (RSTP), which is discussed next.

IEEE 802.1w-Rapid Spanning Tree Protocol

RSTP is an evolution of the 802.1D STP standard. RSTP is a Layer 2 loop prevention algorithm like 802.1D; however, RSTP achieves rapid failover and convergence times, because RSTP is not a timer-based spanning tree algorithm (STA) like 802.1D; but rather a handshake-based spanning tree algorithm. Therefore, RSTP offers an improvement of over 30 seconds or more, as compared to 802.1D, in transitioning a link into a forwarding state.

There are the following three port states in RSTP:

- Learning
- Forwarding
- Discarding

The disabled, blocking, and listening states from 802.1D have been merged into a unique 802.1w discarding state.

Rapid transition is the most important feature introduced by 802.1w. The legacy STA passively waited for the network to converge before moving a port into the forwarding state. Achieving faster convergence was a matter of tuning the conservative default timers, often sacrificing the stability of the network.

RSTP is able to actively confirm that a port can safely transition to forwarding without relying on any timer configuration. There is a feedback mechanism that operates between RSTP-compliant bridges. To achieve fast convergence on a port, the RSTP relies on two new variables: edge ports and link type.

The edge port concept basically corresponds to the PortFast feature. The idea is that ports that are directly connected to end stations cannot create bridging loops in the network and can thus directly transition to forwarding (skipping the 802.1D listening and learning states). An edge port does not generate topology changes when its link toggles. Unlike PortFast, however, an edge port that receives a BPDU immediately loses its edge port status and becomes a normal spanning tree port.

RSTP can achieve rapid transition to forwarding only on edge ports and on point-to-point links. The link type is automatically derived from the duplex mode of a port. A port operating in full-duplex is assumed to be point-to-point, while a half-duplex port is considered as a shared port by default. In switched networks today, most links are operating in full-duplex mode and are therefore treated as point-to-point links by RSTP. This makes them candidates for rapid transition to forwarding.

Like STP, you can enable RSTP globally on a per-VLAN basis, also referred to as Rapid-Per-VLAN-Spanning Tree (Rapid-PVST) mode, using the following command:

```
Switch(config)# spanning-tree mode rapid-pvst
```

Beyond STP, there are many other L2 technologies that also play a key role in available network design, such as trunks, which are discussed in the following section.

Trunks, Cisco Inter-Switch Link, and IEEE 802.1Q

A trunk is a point-to-point link between two networking devices (switches and/or routers) capable of carrying traffic from multiple VLANs over a single link. VLAN frames are encapsulated with trunking protocols to preserve logical separation of traffic while transiting the trunk.

There are two trunking encapsulations available to Cisco devices:

- Inter-Switch Link (ISL)—ISL is a Cisco-proprietary trunking encapsulation.
- IEEE 802.1Q—802.1Q is an industry-standard trunking encapsulation. Trunks may be configured on individual links or on EtherChannel bundles (discussed in the following section). ISL encapsulates the original Ethernet frame with both a header and a field check sequence (FCS) trailer,

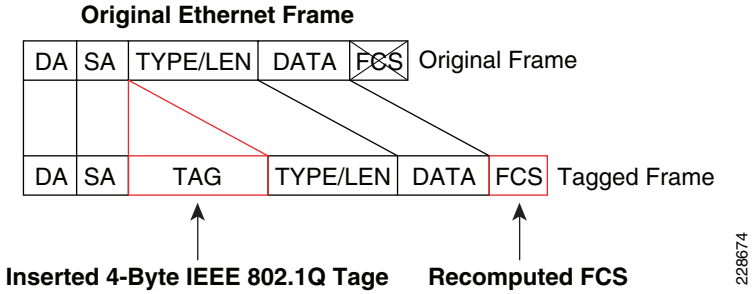
for a total of 30 bytes of encapsulation. ISL trunking can be configured on a switch port interface, as shown in Example 3-1. The trunking mode is set to ISL, and the VLANs permitted to traverse the trunk are explicitly identified; in this example, VLANs 2 and 102 are permitted over the ISL trunk.

Example 3-1 ISL Trunk Example

```
Switch(config)#interface GigabitEthernet8/3
Switch(config-if)# switchport
Switch(config-if)# switchport trunk encapsulation isl
Switch(config-if)# switchport trunk allowed 2, 102
```

In contrast with ISL, 801.1Q does not actually encapsulate the Ethernet frame, but rather inserts a 4-byte tag after the source address field, as well as recomputes a new FCS, as shown in Figure 3-13. This tag not only preserves VLAN information, but also includes a 3-bit field for class of service (CoS) priority (which is discussed in more detail in Chapter 4, "Medianet QoS Design Considerations").

Figure 3-13 IEEE 802.1Q Tagging



IEEE 802.1Q also supports the concept of a native VLAN. Traffic sourced from the native VLAN is not tagged, but is rather simply forwarded over the trunk. As such, only a single native VLAN can be configured for an 802.1Q trunk, to preserve logical separation.

Note Because traffic from the native VLAN is untagged, it is important to ensure that the same native VLAN be specified on both ends of the trunk. Otherwise, this can cause a routing blackhole and potential security vulnerability.

IEEE 802.1Q trunking is likewise configured on a switch port interface, as shown in Example 3-2. The trunking mode is set to 802.1Q, and the VLANs permitted to traverse the trunk are explicitly identified; in this example, VLANs 3 and 103 are permitted over the 802.1Q trunk. Additionally, VLAN 103 is specified as the native VLAN.

Example 3-2 IEEE 802.1Q Trunk Example

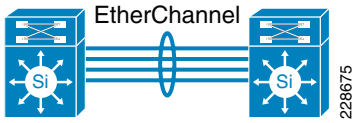
```
Switch(config)# interface GigabitEthernet8/4
Switch(config-if)# switchport
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport trunk allowed 3, 103
Switch(config-if)# switchport trunk native vlan 103
```

Trunks are typically, but not always, configured in conjunction with EtherChannels, which allow for network link redundancy, and are described next.

EtherChannels, Cisco Port Aggregation Protocol, and IEEE 802.3ad

EtherChannel technologies create a single logical link by bundling multiple physical Ethernet-based links (such as Gigabit Ethernet or Ten Gigabit Ethernet links) together, as shown in Figure 3-14. As such, EtherChannel links can provide for increased redundancy, capacity, and load balancing. To optimize the load balancing of traffic over multiple links, Cisco recommends deploying EtherChannels in powers of two (two, four, or eight) physical links. EtherChannel links can operate at either L2 or L3.

Figure 3-14 EtherChannel Bundle



EtherChannel links can be created using Cisco Port Aggregation Protocol (PAgP), which performs a negotiation before forming a channel, to ensure compatibility and administrative policies.

PAgP can be configured in four channeling modes:

- On—This mode forces the LAN port to channel unconditionally. In the On mode, a usable EtherChannel exists only when a LAN port group in the On mode is connected to another LAN port group in the On mode. Ports configured in the On mode do not negotiate to form EtherChannels; they simply do or do not, depending on the configuration of the other port.
- Off—This mode precludes the LAN port from channeling unconditionally.
- Desirable—This PAgP mode places a LAN port into an active negotiating state, in which the port initiates negotiations with other LAN ports to form an EtherChannel by sending PAgP packets. A port in this mode forms an EtherChannel with a peer port that is in either auto or desirable PAgP mode.
- Auto—This (default) PAgP mode places a LAN port into a passive negotiating state, in which the port responds to PAgP packets it receives but does not initiate PAgP negotiation. A port in this mode forms an EtherChannel with a peer port that is in desirable PAgP mode (only).

PAgP, when enabled as an L2 link, is enabled on the physical interface (only). Optionally, you can change the PAgP mode from the default autonegotiation mode, as follows:.

```
Switch(config)# interface GigabitEthernet8/1
Switch(config-if)# channel-protocol pagp
Switch(config-if)# channel-group 15 mode desirable
```

Alternatively, EtherChannels can be negotiated with the IEEE 802.3ad Link Aggregation Control Protocol (LACP). LACP similarly allows a switch to negotiate an automatic bundle by sending LACP packets to the peer. LACP supports two channel negotiation modes:

- Active—This LACP mode places a port into an active negotiating state, in which the port initiates negotiations with other ports by sending LACP packets. A port in this mode forms a bundle with a peer port that is in either active or passive LACP mode.
- Passive—This (default) LACP mode places a port into a passive negotiating state, in which the port responds to LACP packets it receives but does not initiate LACP negotiation. A port in this mode forms a bundle with a peer port that is in active LACP mode (only).

Similar to PAgP, LACP requires only a single command on the physical interface when configured as an L2 link. Optionally, you can change the LACP mode from the default passive negotiation mode, as follows:

```
Switch(config)#interface GigabitEthernet8/2
Switch(config-if)# channel-protocol lacp
```

```
Switch(config-if)# channel-group 16 mode active
```

However, note that PAgP and LACP do not interoperate with each other; ports configured to use PAgP cannot form EtherChannels with ports configured to use LACP, nor can ports configured to use LACP form EtherChannels with ports configured to use PAgP.

EtherChannel plays a critical role in provisioning network link redundancy, especially at the campus distribution and core layers. Furthermore, an evolution of EtherChannel technology plays a key role in Cisco VSS, which is discussed in the following section.

Cisco Virtual Switching System

The Cisco Catalyst 6500 Virtual Switching System (VSS) represents a major leap forward in device and network availability technologies, by combining many of the technologies that have been discussed thus far into a single, integrated system. VSS allows for the combination of two switches into a single, logical network entity from the network control plane and management perspectives. To the neighboring devices, the VSS appears as a single, logical switch or router.

Within the VSS, one chassis is designated as the active virtual switch and the other is designated as the standby virtual switch. All control plane functions, Layer 2 protocols, Layer 3 protocols, and software data path are centrally managed by the active supervisor engine of the active virtual switch chassis. The supervisor engine on the active virtual switch is also responsible for programming the hardware forwarding information onto all the distributed forwarding cards (DFCs) across the entire Cisco VSS as well as the policy feature card (PFC) on the standby virtual switch supervisor engine.

From the data plane and traffic forwarding perspectives, both switches in the VSS actively forward traffic. The PFC on the active virtual switch supervisor engine performs central forwarding lookups for all traffic that ingresses the active virtual switch, whereas the PFC on the standby virtual switch supervisor engine performs central forwarding lookups for all traffic that ingresses the standby virtual switch.

The first step in creating a VSS is to define a new logical entity called the virtual switch domain, which represents both switches as a single unit. Because switches can belong to one or more switch virtual domains, a unique number must be used to define each switch virtual domain, as Example 3-3 demonstrates.

Example 3-3 VSS Virtual Domain Configuration

```
VSS-sw1(config)#switch virtual domain 100

Domain ID 100 config will take effect only
after the exec command `switch convert mode virtual' is issued

VSS-sw1(config-vs-domain)#switch 1
```

Note A corresponding set of commands must be configured on the second switch, with the difference being that *switch 1* becomes *switch 2*. However, the switch virtual domain number must be identical (in this example, 100).

Additionally, to bond the two chassis together into a single, logical node, special signaling and control information must be exchanged between the two chassis in a timely manner. To facilitate this information exchange, a special link is needed to transfer both data and control traffic between the peer chassis. This link is referred to as the virtual switch link (VSL). The VSL, formed as an EtherChannel interface, can comprise links ranging from one to eight physical member ports, as shown by Example 3-4.

Example 3-4 VSL Configuration and VSS Conversion

```
VSS-sw1(config)#interface port-channel 1
VSS-sw1(config-if)#switch virtual link 1
VSS-sw1(config-if)#no shut
VSS-sw1(config-if)#exit
VSS-sw1(config)#interface range tenGigabitEthernet 5/4 - 5
VSS-sw1(config-if-range)#channel-group 1 mode on
VSS-sw1(config-if-range)#no shut
VSS-sw1(config-if-range)#exit
VSS-sw1(config)#exit
VSS-sw1#switch convert mode virtual
```

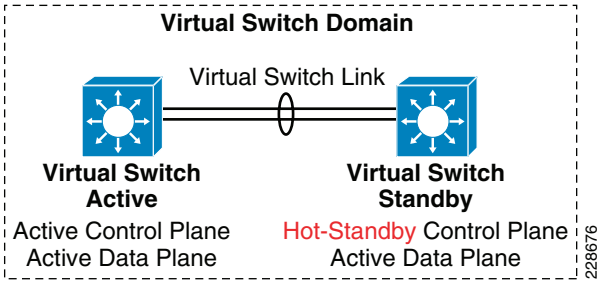
This command converts all interface names to naming convention *interface-type switch-number/slot/port*, saves the running configuration to the startup configuration, and reloads the switch.

```
Do you want to proceed? [yes/no]: yes
Converting interface names
Building configuration...
[OK]
Saving converted configurations to bootflash ...
[OK]
```

Note As previously discussed, a corresponding set of commands must be configured on the second switch, with the difference being that *switch virtual link 1* becomes *switch virtual link 2*. Additionally, *port-channel 1* becomes *port-channel 2*.

VSL links carry two types of traffic: the VSS control traffic and normal data traffic. Figure 3-15 illustrates the virtual switch domain and the VSL.

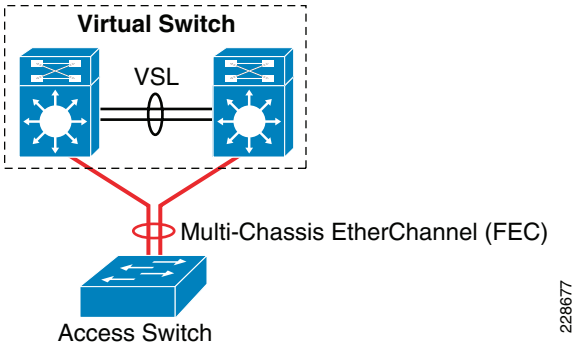
Figure 3-15 Virtual Switch Domain and Virtual Switch Link



Furthermore, VSS allows for an additional addition to EtherChannel technology: multi-chassis EtherChannel (MEC). Before VSS, EtherChannels were restricted to reside within the same physical switch. However, in a VSS environment, the two physical switches form a single logical network entity, and therefore EtherChannels can be extended across the two physical chassis, forming an MEC.

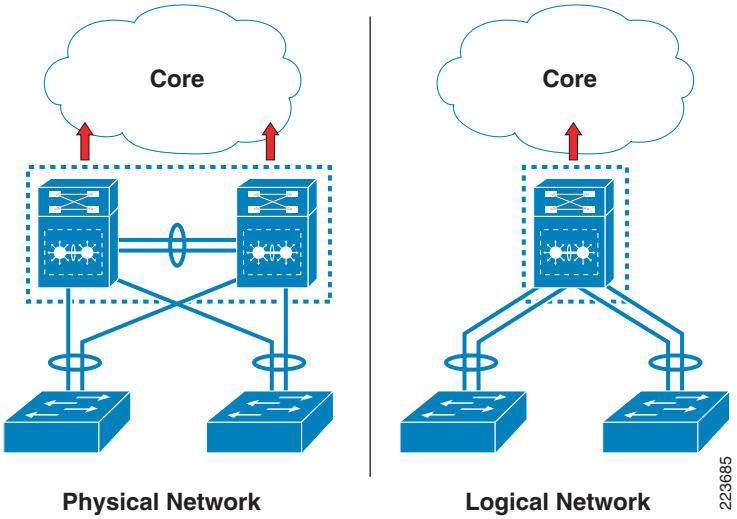
Thus, MEC allows for an EtherChannel bundle to be created across two separate physical chassis (although these two physical chassis are operating as a single, logical entity), as shown in Figure 3-16.

Figure 3-16 Multi-Chassis EtherChannel Topology



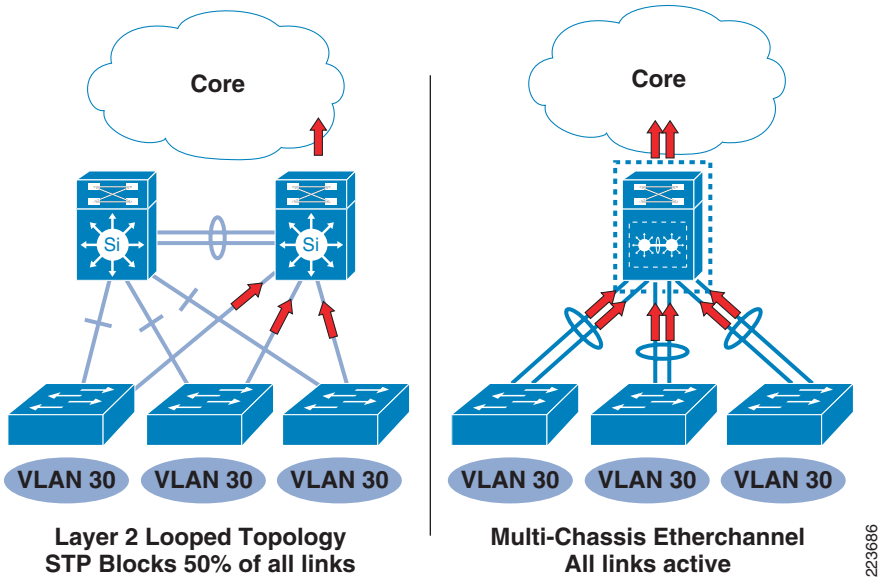
Therefore, MEC allows all the dual-homed connections to and from the upstream and downstream devices to be configured as EtherChannel links, as opposed to individual links. From a configuration standpoint, the commands to form a MEC are the same as a regular EtherChannel; they are simply applied to interfaces that reside on two separate physical switches, as shown in Figure 3-17.

Figure 3-17 MEC--Physical and Logical Campus Network Blocks



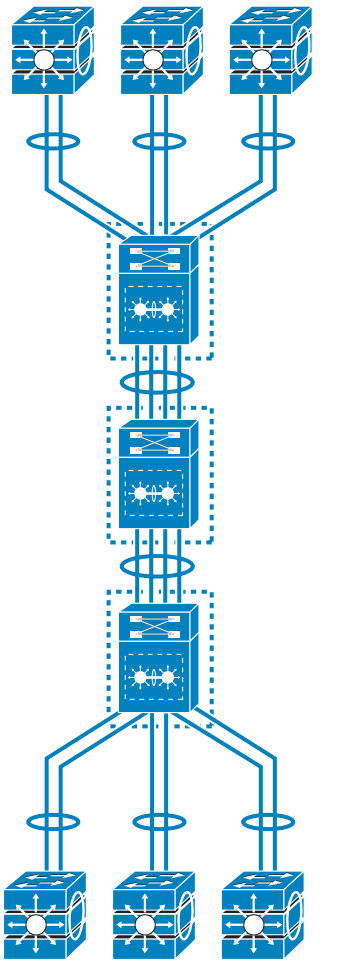
As a result, MEC links allow for implementation of network designs where true Layer 2 multipathing can be implemented without the reliance on Layer 2 redundancy protocols such as STP, as shown in Figure 3-18.

Figure 3-18 STP Topology and VSS Topology



The advantage of VSS over STP is highlighted further by comparing Figure 3-19, which shows a full campus network design using VSS, with Figure 3-9, which shows a similar campus network design using STP.

Figure 3-19 VSS Campus Network Design



The ability to remove physical loops from the topology, and no longer be dependent on spanning tree, is one of the significant advantages of the virtual switch design. However, it is not the only difference. The virtual switch design allows for a number of fundamental changes to be made to the configuration and operation of the distribution block. By simplifying the network topology to use a single virtual distribution switch, many other aspects of the network design are either greatly simplified or, in some cases, no longer necessary.

Furthermore, network designs using VSS can be configured to converge in under 200 ms, which is 250 times faster than STP.

L3 Network Availability Technologies

L3 network availability technologies that particularly relate to TelePresence network design include the following:

- Hot Standby Router Protocol (HSRP)
- Virtual Router Redundancy Protocol (VRRP)

- Gateway Load Balancing Protocol (GLBP)
- IP Event Dampening

Hot Standby Router Protocol

Cisco HSRP is the first of three First Hop Redundancy Protocols (FHRPs) discussed in this chapter (the other two being VRRP and GLBP). A FHRP provides increased availability by allowing for transparent failover of the first-hop IP router, also known as the default gateway (for endpoint devices).

HSRP is used in a group of routers for selecting an active router and a standby router. In a group of router interfaces, the active router is the router of choice for routing packets; the standby router is the router that takes over when the active router fails or when preset conditions are met.

Endpoint devices, or IP hosts, have an IP address of a single router configured as the default gateway. When HSRP is used, the HSRP virtual IP address is configured as the host default gateway instead of the actual IP address of the router.

When HSRP is configured on a network segment, it provides a virtual MAC address and an IP address that is shared among a group of routers running HSRP. The address of this HSRP group is referred to as the virtual IP address. One of these devices is selected by the HSRP to be the active router. The active router receives and routes packets destined for the MAC address of the group.

HSRP detects when the designated active router fails, at which point a selected standby router assumes control of the MAC and IP addresses of the hot standby group. A new standby router is also selected at that time.

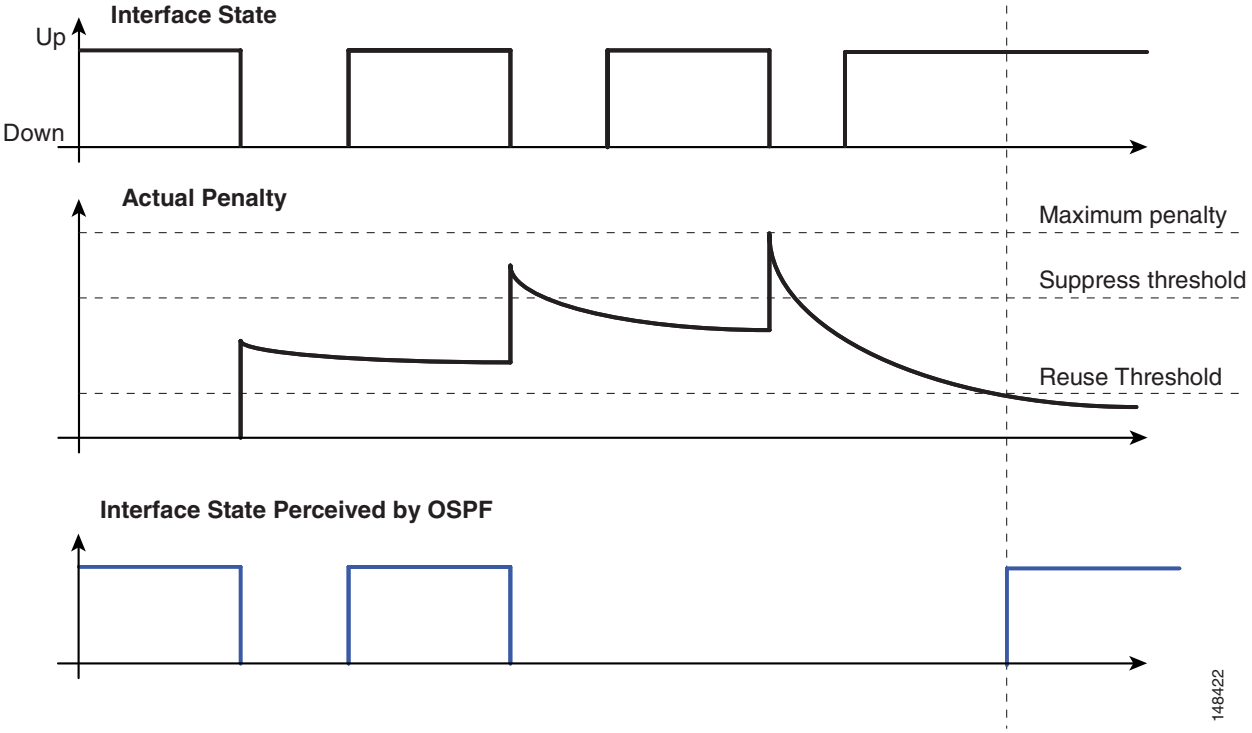
HSRP uses a priority mechanism to determine which HSRP configured router is to be the default active router. To configure a router as the active router, you assign it a priority that is higher than the priority of all the other HSRP-configured routers. The default priority is 100, so if just one router is configured to have a higher priority, that router is the default active router.

Devices that are running HSRP send and receive multicast UDP-based hello messages to detect router failure and to designate active and standby routers. When the active router fails to send a hello message within a configurable period of time, the standby router with the highest priority becomes the active router. The transition of packet forwarding functions between routers is completely transparent to all hosts on the network.

Multiple hot standby groups can be configured on an interface, thereby making fuller use of redundant routers and load sharing.

Figure 3-20 shows a network configured for HSRP. By sharing a virtual MAC address and IP address, two or more routers can act as a single virtual router. The virtual router does not physically exist but represents the common default gateway for routers that are configured to provide backup to each other. All IP hosts are configured with IP address of the virtual router as their default gateway. If the active router fails to send a hello message within the configurable period of time, the standby router takes over and responds to the virtual addresses and becomes the active router, assuming the active router duties.

Figure 3-20 HSRP Topology



HSRP also supports object tracking, such that the HSRP priority of a router can dynamically change when an object that is being tracked goes down. Examples of objects that can be tracked are the line protocol state of an interface or the reachability of an IP route. If the specified object goes down, the HSRP priority is reduced.

Furthermore, HSRP supports SSO awareness, such that HRSP can alter its behavior when a router with redundant route processors (RPs) are configured in SSO redundancy mode. When an RP is active and the other RP is standby, SSO enables the standby RP to take over if the active RP fails.

With this functionality, HSRP SSO information is synchronized to the standby RP, allowing traffic that is sent using the HSRP virtual IP address to be continuously forwarded during a switchover without a loss of data or a path change. Additionally, if both RPs fail on the active HSRP router, the standby HSRP router takes over as the active HSRP router.

Note SSO awareness for HSRP is enabled by default when the redundancy mode of operation of the RP is set to SSO, as was shown in Non-Stop Forwarding with Stateful Switch Over, page 3-7.

Example 3-5 demonstrates the HSRP configuration that can be used on the LAN interface of the active router from Figure 3-20. Each HSRP group on a given subnet requires a unique number; in this example, the HSRP group number is set to 10. The IP address of the virtual router (which is what each IP host on the network uses as a default gateway address) is set to 172.16.128.3. The HRSP priority of this router has been set to 105 and preemption has been enabled on it; preemption allows for the router to immediately take over as the virtual router (provided it has the highest priority on the segment). Finally, object tracking has been configured, such that should the line protocol state of interface Serial0/1 go down (the WAN link for the active router, which is designated as object-number 110), the HSRP priority for this interface dynamically decrements (by a value of 10, by default).

Example 3-5 HSRP Example

```
track 110 interface Serial0/1 line-protocol
!
interface GigabitEthernet0/0
ip address 172.16.128.1 255.255.255.0
standby 10 ip 172.16.128.3
standby 10 priority 105 preempt
standby 10 track 110
!
```

Because HRSP was the first FHRP and because it was invented by Cisco, it is Cisco-proprietary. However, to support multi-vendor interoperability, aspects of HSRP were standardized in the Virtual Router Redundancy Protocol (VRRP), which is discussed next.

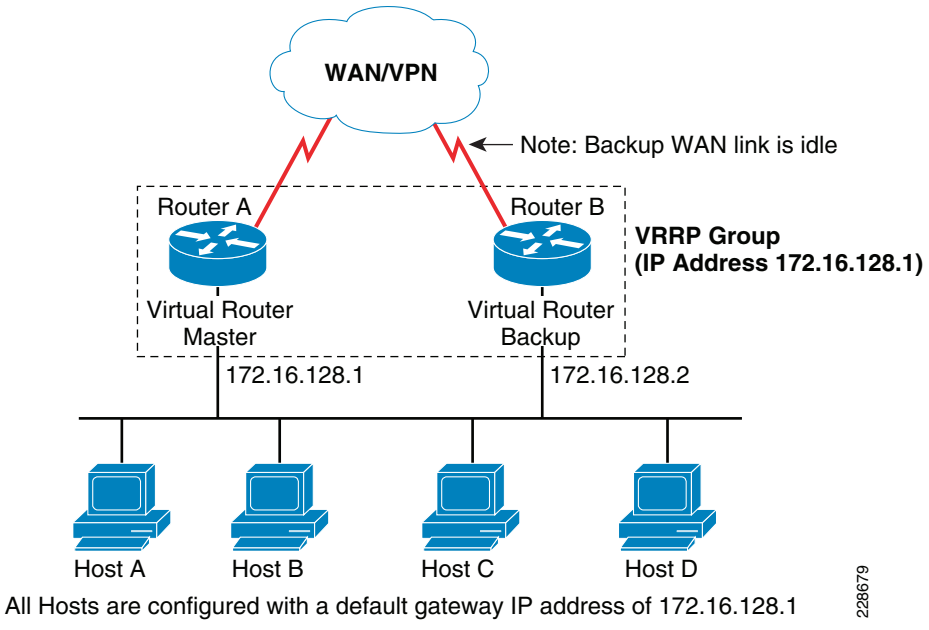
Virtual Router Redundancy Protocol

VRRP, defined in RFC 2338, is an FHRP very similar to HSRP, but is able to support multi-vendor environments. A VRRP router is configured to run the VRRP protocol in conjunction with one or more other routers attached to a LAN. In a VRRP configuration, one router is elected as the virtual router master, with the other routers acting as backups in case the virtual router master fails.

VRRP enables a group of routers to form a single virtual router. The LAN clients can then be configured with the virtual router as their default gateway. The virtual router, representing a group of routers, is also known as a VRRP group.

Figure 3-21 shows a LAN topology in which VRRP is configured. In this example, two VRRP routers (routers running VRRP) comprise a virtual router. However, unlike HSRP, the IP address of the virtual router is the same as that configured for the LAN interface of the virtual router master; in this example, 172.16.128.1.

Figure 3-21 VRRP Topology



Router A assumes the role of the virtual router master and is also known as the IP address owner, because the IP address of the virtual router belongs to it. As the virtual router master, Router A is responsible for forwarding packets sent to this IP address. Each IP host on the subnet is configured with the default gateway IP address of the virtual route master, in this case 172.16.128.1.

Router B, on the other hand, functions as a virtual router backup. If the virtual router master fails, the router configured with the higher priority becomes the virtual router master and provides uninterrupted service for the LAN hosts. When Router A recovers, it becomes the virtual router master again.

Additionally, like HSRP, VRRP supports object tracking, preemption, and SSO awareness.

Note SSO awareness for VRRP is enabled by default when the redundancy mode of operation of the RP is set to SSO, as was shown in Non-Stop Forwarding with Stateful Switch Over, page 3-7.

Example 3-6 shows a VRRP configuration that can be used on the LAN interface of the virtual router master from Figure 3-21. Each VRRP group on a given subnet requires a unique number; in this example, the VRRP group number is set to 10. The virtual IP address is set to the actual LAN interface address, designating this router as the virtual router master. The VRRP priority of this router has been set to 105. Unlike HSRP, preemption for VRRP is enabled by default. Finally, object tracking has been configured, such that should the line protocol state of interface Serial0/1 go down (the WAN link for this router, which is designated as object-number 110), the VRRP priority for this interface dynamically decrements (by a value of 10, by default).

Example 3-6 VRRP Example

```
!
track 110 interface Serial0/1 line-protocol
!
interface GigabitEthernet0/0
ip address 172.16.128.1 255.255.255.0
vrrp 10 ip 172.16.128.1
vrrp 10 priority 105
vrrp 10 track 110
!
```

A drawback to both HSRP and VRRP is that the standby/backup router is not used to forward traffic, and as such wastes both available bandwidth and processing capabilities. This limitation can be worked around by provisioning two complementary HSRP/VRRP groups on each LAN subnet, with one group having the left router as the active/master and the other group having the right router as the active/master router. Then, approximately half of the hosts are configured to use the virtual IP address of one HSRP/VRRP group, and the remaining hosts are configured to use the virtual IP address of the second group. This requires additional operational and management complexity. To improve the efficiency of these FHRP models without such additional complexity, GLBP can be used, which is discussed next.

Gateway Load Balancing Protocol

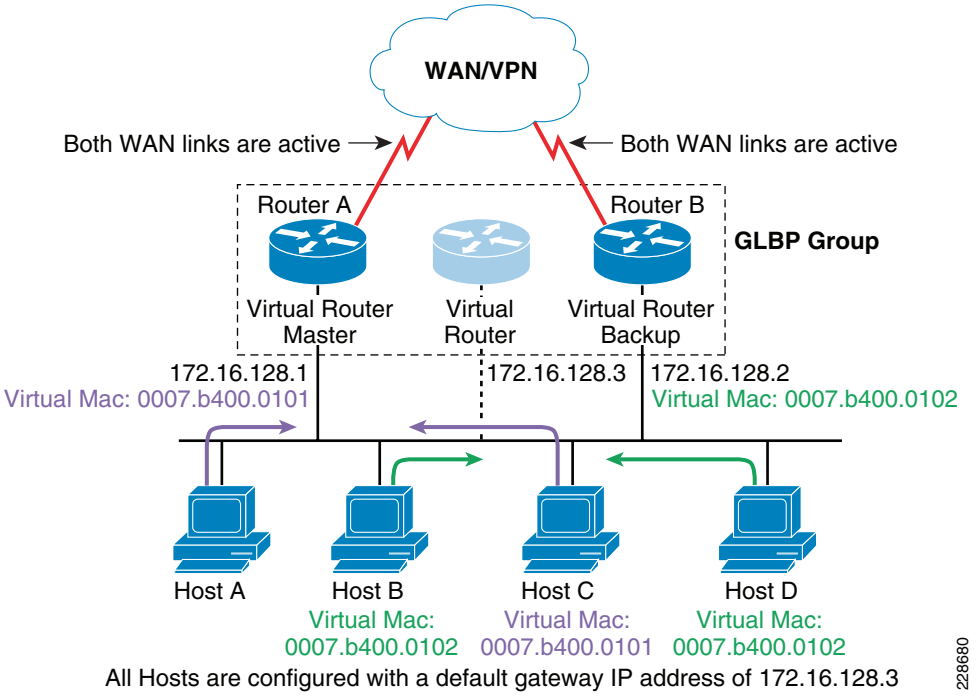
Cisco GLBP improves the efficiency of FHRP protocols by allowing for automatic load balancing of the default gateway. The advantage of GLBP is that it additionally provides load balancing over multiple routers (gateways) using a single virtual IP address and multiple virtual MAC addresses per GLBP group (in contrast, both HRSP and VRRP used only one virtual MAC address per HSRP/VRRP group). The forwarding load is shared among all routers in a GLBP group rather than being handled by a single router while the other routers stand idle. Each host is configured with the same virtual IP address, and all routers in the virtual router group participate in forwarding packets.

Members of a GLBP group elect one gateway to be the active virtual gateway (AVG) for that group. Other group members provide backup for the AVG in the event that the AVG becomes unavailable. The function of the AVG is that it assigns a virtual MAC address to each member of the GLBP group. Each gateway assumes responsibility for forwarding packets sent to the virtual MAC address assigned to it by the AVG. These gateways are known as active virtual forwarders (AVFs) for their virtual MAC address.

The AVG is also responsible for answering Address Resolution Protocol (ARP) requests for the virtual IP address. Load sharing is achieved by the AVG replying to the ARP requests with different virtual MAC addresses (corresponding to each gateway router).

In Figure 3-22, *Router A* is the AVG for a GLBP group, and is primarily responsible for the virtual IP address 172.16.128.3; however, Router A is also an AVF for the virtual MAC address 0007.b400.0101. *Router B* is a member of the same GLBP group and is designated as the AVF for the virtual MAC address 0007.b400.0102. All hosts have their default gateway IP addresses set to the virtual IP address of 172.16.128.3. However, when these use ARP to determine the MAC of this virtual IP address, *Host A* and *Host C* receive a gateway MAC address of 0007.b400.0101 (directing these hosts to use Router A as their default gateway), but *Host B* and *Host D* receive a gateway MAC address 0007.b400.0102 (directing these hosts to use Router B as their default gateway). In this way, the gateway routers automatically load share.

Figure 3-22 GLBP Topology



If Router A becomes unavailable, Hosts A and C do not lose access to the WAN because Router B assumes responsibility for forwarding packets sent to the virtual MAC address of Router A, and for responding to packets sent to its own virtual MAC address. Router B also assumes the role of the AVG for the entire GLBP group. Communication for the GLBP members continues despite the failure of a router in the GLBP group.

Additionally, like HSRP and VRRP, GLBP supports object tracking, preemption, and SSO awareness.

Note SSO awareness for GLBP is enabled by default when the route processor's redundancy mode of operation is set to SSO, as was shown in Non-Stop Forwarding with Stateful Switch Over, page 3-7.

However, unlike the object tracking logic used by HSRP and VRRP, GLBP uses a weighting scheme to determine the forwarding capacity of each router in the GLBP group. The weighting assigned to a router in the GLBP group can be used to determine whether it forwards packets and, if so, the proportion of hosts in the LAN for which it forwards packets. Thresholds can be set to disable forwarding when the weighting for a GLBP group falls below a certain value; when it rises above another threshold, forwarding is automatically re-enabled.

GLBP group weighting can be automatically adjusted by tracking the state of an interface within the router. If a tracked interface goes down, the GLBP group weighting is reduced by a specified value. Different interfaces can be tracked to decrement the GLBP weighting by varying amounts.

Example 3-7 shows a GLBP configuration that can be used on the LAN interface of the AVG from Figure 3-22. Each GLBP group on a given subnet requires a unique number; in this example, the GLBP group number is set to 10. The virtual IP address for the GLBP group is set to 172.16.128.3. The GLBP priority of this interface has been set to 105, and like HSRP, preemption for GLBP must be explicitly enabled (if desired). Finally, object tracking has been configured, such that should the line protocol state of interface Serial0/1 go down (the WAN link for this router, which is designated as object-number 110), the GLBP priority for this interface dynamically decrements (by a value of 10, by default).

Example 3-7 GLBP Example

```
!
track 110 interface Serial0/1 line-protocol
!
interface GigabitEthernet0/0
ip address 172.16.128.1 255.255.255.0
glbp 10 ip 172.16.128.3
glbp 10 priority 105
glbp 10 preempt
glbp 10 weighting track 110
!
```

Having concluded an overview of these FHRPs, a discussion of another type of L3 network availability feature, IP Event Dampening, follows.

IP Event Dampening

Whenever the line protocol of an interface changes state, or flaps, routing protocols are notified of the status of the routes that are affected by the change in state. Every interface state change requires all affected devices in the network to recalculate best paths, install or remove routes from the routing tables, and then advertise valid routes to peer routers. An unstable interface that flaps excessively can cause other devices in the network to consume substantial amounts of system processing resources and cause routing protocols to lose synchronization with the state of the flapping interface.

The IP Event Dampening feature introduces a configurable exponential decay mechanism to suppress the effects of excessive interface flapping events on routing protocols and routing tables in the network. This feature allows the network administrator to configure a router to automatically identify and selectively dampen a local interface that is flapping. Dampening an interface removes the interface from the network until the interface stops flapping and becomes stable.

Configuring the IP Event Dampening feature improves convergence times and stability throughout the network by isolating failures so that disturbances are not propagated, which reduces the use of system processing resources by other devices in the network and improves overall network stability.

IP Event Dampening uses a series of administratively-defined thresholds to identify flapping interfaces, to assign penalties, to suppress state changes (if necessary), and to make stabilized interfaces available to the network. These thresholds are as follows:

- Suppress threshold—The value of the accumulated penalty that triggers the router to dampen a flapping interface. The flapping interface is identified by the router and assigned a penalty for each up and down state change, but the interface is not automatically dampened. The router tracks the penalties that a flapping interface accumulates. When the accumulated penalty reaches the default or preconfigured suppress threshold, the interface is placed in a dampened state. The default suppress threshold value is 2000.
- Half-life period—Determines how fast the accumulated penalty can decay exponentially. When an interface is placed in a dampened state, the router monitors the interface for additional up and down state changes. If the interface continues to accumulate penalties and the interface remains in the suppress threshold range, the interface remains dampened. If the interface stabilizes and stops flapping, the penalty is reduced by half after each half-life period expires. The accumulated penalty is reduced until the penalty drops to the reuse threshold. The default half-life period timer is five seconds.
- Reuse threshold—When the accumulated penalty decreases until the penalty drops to the reuse threshold, the route is unsuppressed and made available to the other devices on the network. The default value is 1000 penalties.
- Maximum suppress time—The maximum suppress time represents the maximum amount of time an interface can remain dampened when a penalty is assigned to an interface. The default maximum penalty timer is 20 seconds.

IP Event Dampening is configured on a per-interface basis (where default values are used for each threshold) as follows:

```
interface FastEthernet0/0
dampening
```

IP Event Dampening can be complemented with the use of route summarization, on a per-routing protocol basis, to further compartmentalize the effects of flapping interfaces and associated routes.

Operational Availability Technologies

As has been shown, the predominant way that availability of a network can be improved is to improve its MTBF by using devices that have redundant components and by engineering the network itself to be as redundant as possible, leveraging many of the technologies discussed in the previous sections.

However, glancing back to the general availability formula from Figure 3-1, another approach to improving availability is to reduce MTTR. Reducing MTTR is primarily a factor of operational resiliency.

MTTR operations can be significantly improved in conjunction with device and network redundant design. Specifically, the ability to make changes, upgrade software, and replace or upgrade hardware in a production network is extensively improved because of the implementation of device and network redundancy. The ability to upgrade individual devices without taking them out of service is based on having internal component redundancy complemented with the system software capabilities. Similarly,

by having dual active paths through redundant network devices designed to converge in sub-second timeframes, it is possible to schedule an outage event on one element of the network and allow it to be upgraded and then brought back into service with minimal or no disruption to the network as a whole.

MTTR can also be improved by reducing the time required to perform any of the following operations:

- Failure detection
- Notification
- Fault diagnosis
- Dispatch/arrival
- Fault repair

Technologies that can help automate and streamline these operations include the following:

- Cisco General Online Diagnostics (GOLD)
- Cisco IOS Embedded Event Manager (EEM)
- Cisco In Service Software Upgrade (ISSU)
- Online Insertion and Removal (OIR)

This section briefly introduces each of these technologies.

Cisco Generic Online Diagnostics

Cisco GOLD defines a common framework for diagnostic operations for Cisco IOS Software-based products. GOLD has the objective of checking the check the health of all hardware components and verifying the proper operation of the system data plane and control plane at boot time, as well as run-time.

GOLD supports the following:

- Bootup tests (includes online insertion)
- Health monitoring tests (background non-disruptive)
- On-demand tests (disruptive and non-disruptive)
- User scheduled tests (disruptive and non-disruptive)
- Command-line interface (CLI) access to data via a management interface

GOLD, in conjunction with several of the technologies previously discussed, can reduce device failure detection time.

Cisco IOS Embedded Event Manager

The Cisco IOS EEM offers the ability to monitor device hardware, software, and operational events and take informational, corrective, or any desired action, including sending an e-mail alert, when the monitored events occur or when a threshold is reached.

EEM can notify a network management server and/or an administrator (via e-mail) when an event of interest occurs. Events that can be monitored include the following:

- Application-specific events
- CLI events
- Counter/interface-counter events

- Object-tracking events
- Online insertion and removal events
- Resource events
- GOLD events
- Redundancy events
- Simple Network Management Protocol (SNMP) events
- Syslog events
- System manager/system monitor events
- IOS watchdog events
- Timer events

Capturing the state of network devices during such situations can be helpful in taking immediate recovery actions and gathering information to perform root-cause analysis, reducing fault detection and diagnosis time. Notification times are reduced by having the device send e-mail alerts to network administrators. Furthermore, availability is also improved if automatic recovery actions are performed without the need to fully reboot the device.

Cisco In Service Software Upgrade

Cisco ISSU provides a mechanism to perform software upgrades and downgrades without taking a switch out of service. ISSU leverages the capabilities of NSF and SSO to allow the switch to forward traffic during supervisor IOS upgrade (or downgrade). With ISSU, the network does not re-route and no active links are taken out of service. ISSU thereby expedites software upgrade operations.

Online Insertion and Removal

OIR allows line cards to be added to a device without affecting the system. Additionally, with OIR, line cards can be exchanged without losing the configuration. OIR thus expedites hardware repair and/or replacement operations.

Summary

Availability was shown to be a factor of two components: the mean time between failures (MTBF) and the mean time to repair (MTTR) such failures. Availability can be improved by increasing MTBF (which is primarily a function of device and network resiliency/redundancy), or by reducing MTTR (which is primarily a function of operational resiliency.

Device availability technologies were discussed, including Cisco Catalyst StackWise/StackWise Plus technologies, which provide 1:N control plane redundancy to Cisco Catalyst 3750G/3750E switches, as well as NSF with SSO, which similarly provides hot standby redundancy to network devices with multiple route processors.

Network availability technologies were also discussed, beginning with Layer 2 technologies, such as spanning tree protocols, trunking protocols, EtherChannel protocols, and Cisco VSS. Additionally, Layer 3 technologies, such as HSRP, VRRP, GLBP, and IP Event dampening were introduced.

Finally, operational availability technologies were introduced to show how availability can be improved by automating and streamlining MTTR operations, including GOLD, EEM, ISSU, and OIR.