

Cisco StackWise architecture on Catalyst 9200 Series Switches

Contents

Overview	3
Introduction to StackWise-160/80	3
Stacking components	4
Stack ports	5
Stack connectors	6
Stacking architecture	6
Ring architecture	6
Stack discovery	8
ACTIVE election.....	8
Stackwise-160/80 architecture	12
Distributed forwarding architecture	12
SSO architecture.....	13
Implementing StackWise-160/80 SSO	14
Stack operations	14
Adding a switch.....	14
Removing a switch.....	16
Stack split.....	17
Stack merge.....	17
Auto-upgrade	17
Licensing.....	19
Smart Licensing.....	19
Stack design	19
Simplified multilayer network design	19
Scalable StackWise-160/80 cross-stack EtherChannel design.....	20
Optimal StackWise-160/80 cross-stack forwarding EtherChannel design	21
Reliable StackWise-160/80 cross-stack forwarding Ether Channel design	23
Summary	24

Overview

The network access model in the enterprise campus has evolved significantly from basic user connectivity to an intelligent, powerful, and high-speed building block. Security, cloud, mobility, and the Internet of Things (IoT) in enterprise networks have been driving the network toward major innovations. Cisco® Catalyst® 9000 software and hardware have been designed to address these current and future demands.

Wireless technology is the industry's new megatrend, and it presents a multidimensional challenge to enterprise IT organizations. This swiftly growing technology, with exponentially increasing numbers of mobile devices with high-performance demands, is rapidly changing the landscape of network infrastructure and its reliability. IT requires a reassessment of traditional network models and assembly of network designs that can respond with a broad set of evolutionary architectures. Stacking provides an opportunity to use a pay-as-you-grow model to meet these demands.

As the wired and wireless Gigabit evolution begins at the access layer, user and application demands will increase to high-speed, low-latency data switching for optimal performance. Cisco has built a system architecture to support such innovations. The new Cisco Catalyst 9000 switching family is the next generation in the legendary family of enterprise LAN access, aggregation, and core switches. Cisco Catalyst 9200 Series Switches extend the power of intent-based networking and hardware and software innovation to a broader set of deployments. Compared to the scale and feature richness of the Cisco Catalyst 9300 Series Switches, the 9200 Series switches focus on offering features for the midmarket and simple branch deployments. With its family pedigree, the Cisco Catalyst 9200 Series offers simplicity without compromise. It is secure, is always on, and provides IT simplicity. This white paper talks in detail about the benefits and architecture of Cisco StackWise®-160/80, which provides the always-on experience.

Introduction to StackWise-160/80

The StackWise-160/80 architecture allows stacking of up to eight switches in a ring topology to achieve a high density of stack bandwidth. The stacking architecture expands the switches' form factor, switching capacity, port density, and redundancy as well as providing a single control plane. It also provides resiliency, scalability, and central management. The latest Cisco Catalyst 9200 Series Switches support StackWise-160, whereas the Cisco Catalyst 9200L models support StackWise-80. This technology is flexible, modular, and evolutionary, and it delivers Cisco IOS® XE feature capabilities with hardware acceleration to every port in the stack.

Cisco Catalyst 9200 Series Switches come in data, Power over Ethernet (PoE), and Multigigabit versions. Each version has 24-port and 48-port variants. The hardware design of each model is cost-effective to support different network capacity load and switching performance.

Up to a maximum of eight switches can be stacked together physically in a ring topology to form a single, unified, virtual stack system. A Cisco Catalyst 9200 Series Switch, when deployed in StackWise-160/80 mode, has been designed to deliver deterministic and nonblocking switching performance to a maximum port density of 448 ports with a distributed data plane, single control plane, and management plane. The switching performance delivers hardware-accelerated, integrated borderless network services such as PoE, PoE+, Quality of Service (QoS), Access Control Lists (ACLs), Flexible NetFlow, Next-Generation Network-Based Application Recognition (NBAR2), streaming telemetry, and many more services on every port.

Depending on the requirements of each switch in the stack, the Cisco Catalyst 9200 Series Switches provide the flexibility for mixed-mode support between different models in a single stack ring. You can mix switches with different model variants (PoE, data, Multigigabit) and different network modules in the stack; however, it is not possible to mix the 9200 and 9200L models due to different stacking bandwidth.

Figure 1 shows the StackWise-160/80 technology when four switches are part of the stack. Figure 2 shows simplified physical and logical views of a stack.

Figure 1. Cisco Catalyst 9200 Series StackWise-160 technology

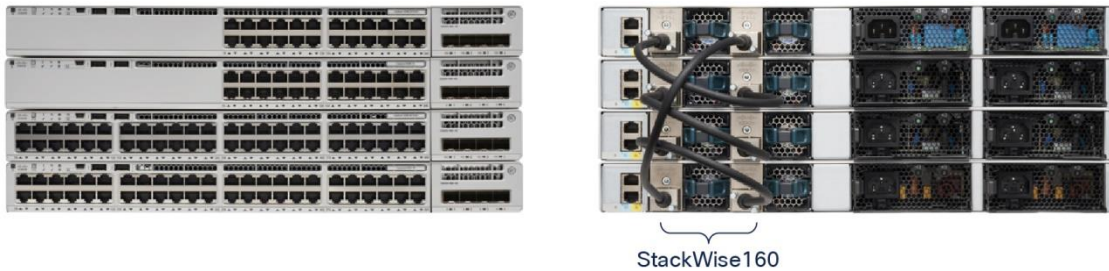
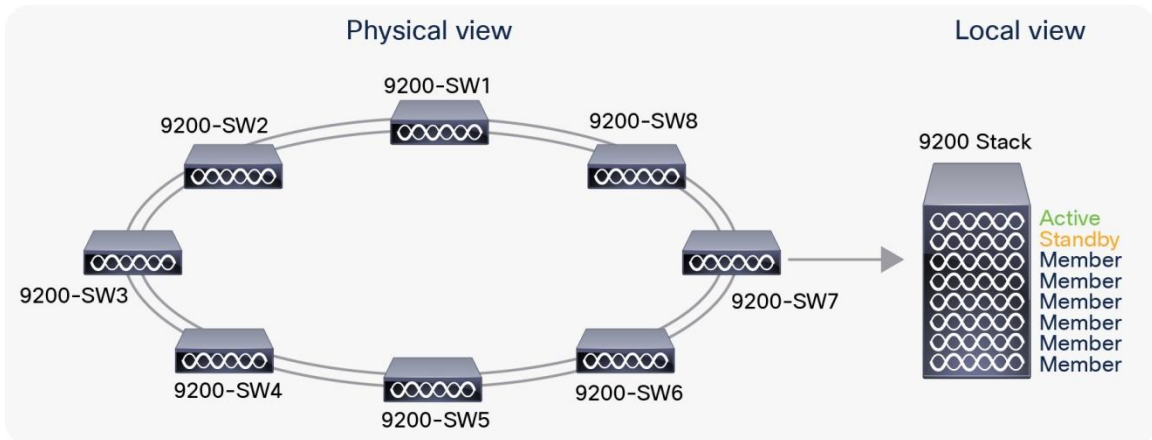


Figure 2. Simplified Cisco Catalyst 9200 Series physical and logical views



Stacking components

Stacking cables are mandatory for stacking architecture. Depending on the physical setup of the infrastructure, different lengths of stacking cable may be needed. Each Cisco Catalyst switch supports a maximum of two stack cables for data stacking. Table 1 lists the stacking cables available. Stacking cables are not backward compatible with the predecessor Cisco Catalyst 2960-X Series Switches. The stacking kit is not included by default when a switch is ordered and should be added separately.

Table 1. Different orderable stack cables

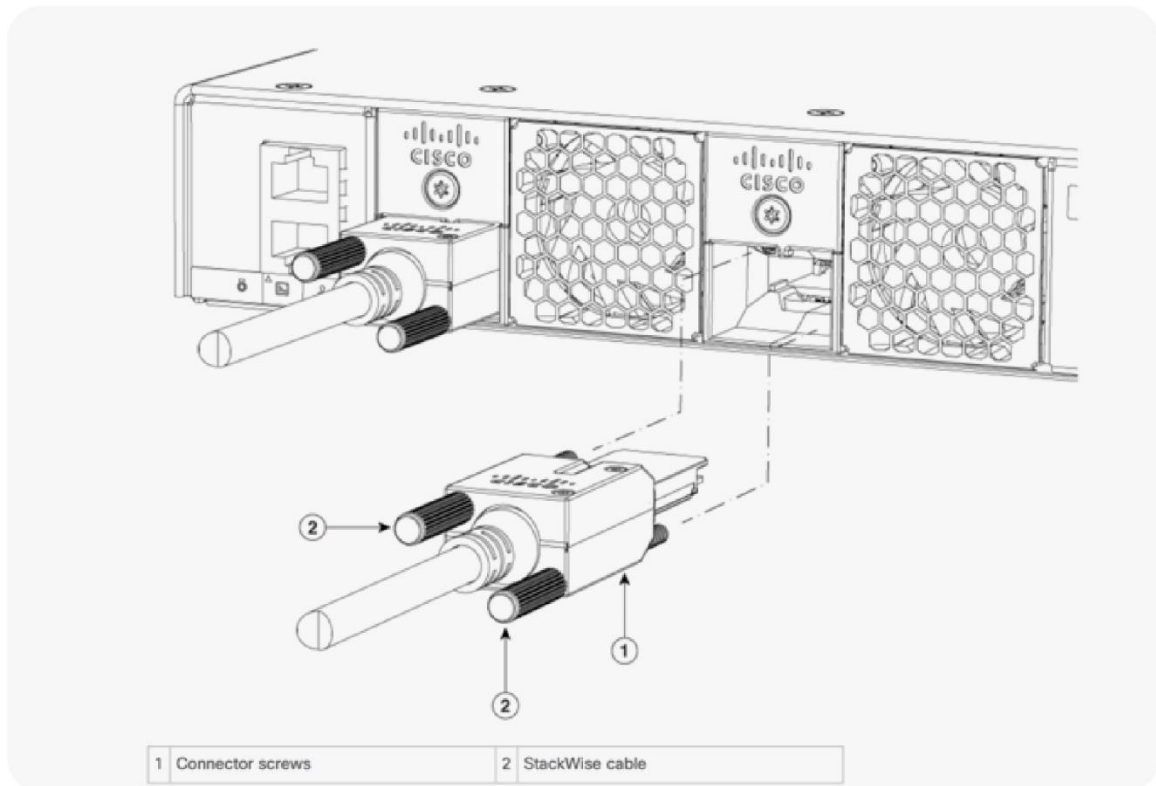
Product ID	Description
C9200-STACK-KIT	Stack kit for C9200 SKUs only: Two data stack adapters and one data stack cable
C9200L-STACK-KIT	Stack kit for C9200L SKUs only: Two data stack adapters and one data stack cable
STACK-T4-50CM	50CM Type 3 Stacking Cable

Product ID	Description
STACK-T4-1M	1M Type 3 Stacking Cable
STACK-T4-3M	3M Type 3 Stacking Cable

Stack ports

Each Cisco Catalyst 9200 Series Switch comes with two stack ports located on the back panel of the switch to support the StackWise-160/80 architecture. Figure 3 shows the stack port location.

Figure 3. Stack cable and stack cable slot on a 9200L model



Stack connectors

Figure 4 shows the stack connector. All stack ports are identical on all models of the Cisco Catalyst 9200 Series. Any side of the stack cable can connect to any stack port. Make sure the screws are completely tightened and the connection is secure.

Figure 4. Cisco Catalyst 9200 Series stacking kit



Stacking architecture

Ring architecture

When the stack is operating in a full ring, it can deliver high-speed throughput of up to 160 Gbps on each Cisco Catalyst 9200 Series Switch and throughput of 80 Gbps on each Cisco Catalyst 9200L switch. This multifold performance improvement is possible by combining the new internal Cisco Unified Access[®] Data Plane Application-Specific Integrated Circuit (UADP ASIC) and dual stack ports.

The high-speed backplane of the Cisco Catalyst 9200 Series stack-ring fabric is constructed by daisy-chaining the stack-member switches with Cisco proprietary cables that connect rear-side stack ports. The stack fabric consists of six unidirectional data transmission rings.

Figure 5 illustrates the 9200 Series StackWise-160 internal forwarding architecture.

Figure 5. Cisco Catalyst 9200 Series StackWise-160 internal forwarding architecture

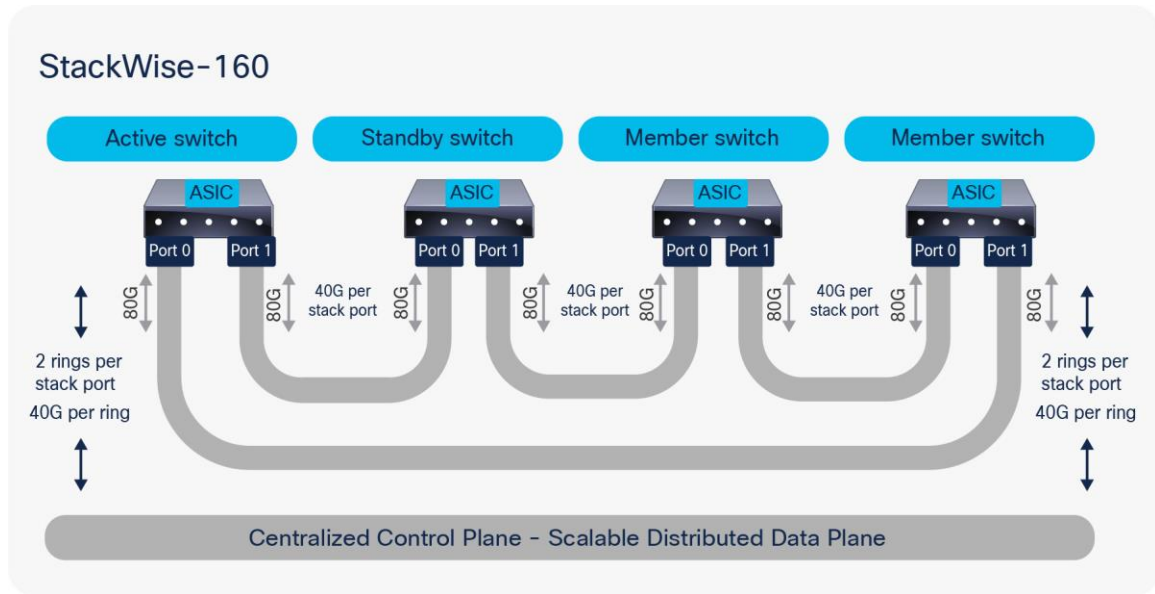
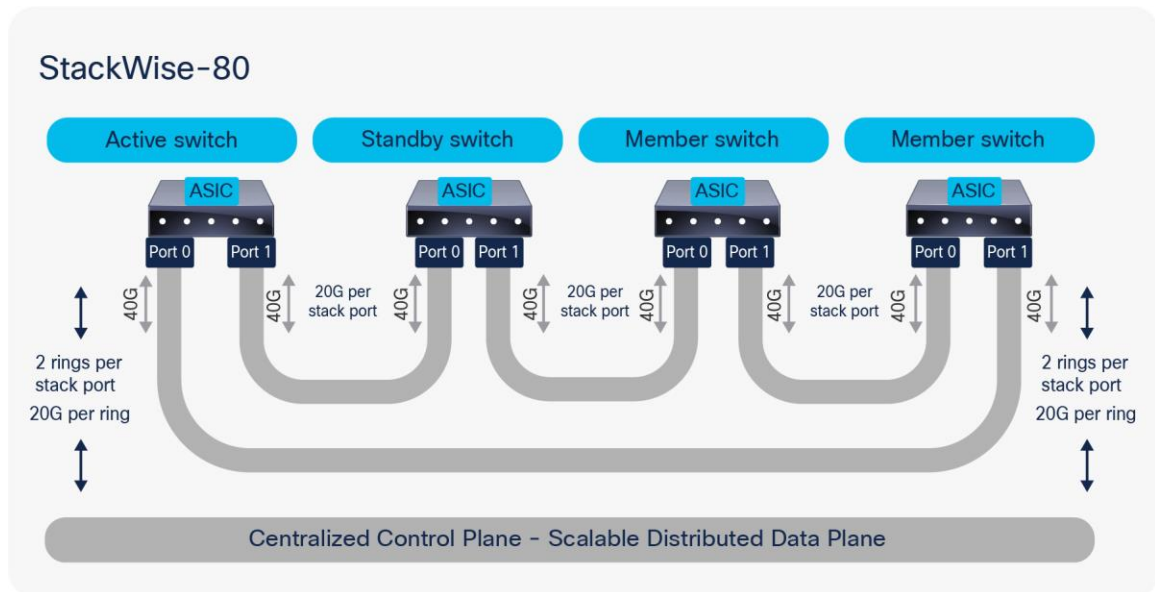


Figure 6 illustrates the StackWise-80 internal forwarding architecture for the 9200L models.

Figure 6. Cisco Catalyst 9200L StackWise-80 internal forwarding architecture



The aggregated throughput that a stack of switches supports is a combination of two primary factors:

- **Total transmission rings:** Each stack connector bundles multiple individual cables that carry data across the stack ring. This cabling structure creates four internal stack rings. This hardware design significantly improves the data transmission performance of each stack port of a Cisco Catalyst 9200 Series Switch.
- **Maximum throughput per ring:** Each stack ring can transmit data at up to 40 Gbps on StackWise-160 and up to 20 Gbps on StackWise-80. These four internal stack rings enable an aggregated throughput of up to 80 Gbps per switch (up to 160 Gbps of unicast with Spatial Reuse Protocol [SRP]).

Table 2 lists the major details of the stacking architecture.

Table 2. Details of the Cisco StackWise architecture

	Cisco Catalyst 9200 StackWise-160	Cisco Catalyst 9200L StackWise-80
Total number of rings	4	4
Throughput per ring	40 Gbps	20 Gbps
Throughput per stack (full ring)	80 Gbps	40 Gbps
Throughput per stack (full ring) with SRP	160 Gbps	80 Gbps

Stack discovery

Once all switches are powered on and the stack interfaces are up, the Stack Discovery Protocol (SDP) discovers the stack topology using broadcasts. Neighbor information is shared with all other switches in the stack. In a full ring, discovery exits after all the members are found. In a half ring, the system waits for two minutes. Once all switches are discovered, switch numbers are determined. After switch number conflicts are resolved, the information is stored in a flash variable block for future use. Active election begins after discovery exits.

The following command can be used to check the status of the stack cable and to identify the neighbor device of the stack cable.

```
C92-Stack#show switch stack-ports summary
```

Sw#/Port#	Port	Status	Neighbor	Cable Length	Link OK	Link Active	Sync OK	#Changes to LinkOK	In Loopback
1/1	OK	4	50cm	Yes	Yes	Yes	1	No	
1/2	OK	3	50cm	Yes	Yes	Yes	1	No	
2/1	OK	3	50cm	Yes	Yes	Yes	1	No	
2/2	OK	4	50cm	Yes	Yes	Yes	1	No	
3/1	OK	1	50cm	Yes	Yes	Yes	1	No	
3/2	OK	2	50cm	Yes	Yes	Yes	1	No	
4/1	OK	2	50cm	Yes	Yes	Yes	1	No	
4/2	OK	1	50cm	Yes	Yes	Yes	1	No	

ACTIVE election

To determine the single ACTIVE and STANDBY switch role during the complete stack reboot process or during the initial boot-up, all switches are required to go through an election process. All member switches participate in the election of an ACTIVE stack switch if they all boot up within the election window (120 seconds).

The following parameters are taken into account in the order listed below for active switch election:

- Highest priority
- Lowest MAC address

The STANDBY is elected by the ACTIVE switch after two minutes to reduce the stress of high-availability sync on the stack.

By default, the priority is 1 for all switches; hence it falls back to the MAC address to determine the ACTIVE switch if a priority is not defined explicitly. The switch with the lowest MAC address will take the role of the ACTIVE switch. The remaining switches join the stack as member switches. Once all the switches in the stack are discovered as members, the ACTIVE switch elects the STANDBY switch. Any switch that you add to the stack after the election process completes is considered a straggler and will not participate in the ACTIVE stack switch election.

To determine which switch is elected as the ACTIVE switch, adjust the switch priority for all stack switches. The switch with the highest priority becomes the ACTIVE switch. This configuration is a one-step process and is typically done at the time of the initial switch boot-up. The switch priorities are set in the ROMMON configuration of each individual switch in the stack ring. The switch priority configuration from ROMMON is parsed during the boot cycle instead of being read from the startup configuration stored in NVRAM. Thus, the switch-priority configuration cannot be verified from startup or the running configuration, since it is programmed into a different configuration component.

The priority of a Cisco Catalyst 9200 Series Switch can be modified using Exec mode commands. A reload is required for the change to take effect. The following commands show how to configure the switch priority and change switch numbers.

Cisco Catalyst 9200 Series switch priority

```
9200>enable
9200#switch <number> priority 15
!Set priority 15 to elect switch in ACTIVE role
9200#switch <number> priority 14
!Set priority 14 to elect switch in STANDBY role
9200#switch <number> priority 13
!Set priority 13 to elect switch in next STANDBY role
9200#switch <number> priority 12
!Set priority 12 to elect switch in next STANDBY role
9200>enable
9200#switch <number> renumber <number>
!Statically renumber switch in stack-ring
```

There are two ways to influence a particular switch in the stack to take over the role of the ACTIVE switch:

- Configure the preferred switch with the highest priority (15) so that it assumes the role of the ACTIVE switch.
- Power the preferred switch on first to have it take on the role of the ACTIVE switch.

Some conditions apply for switches when deployed in StackWise-160/80:

- Valid switch numbers are 1 through 8 for Cisco Catalyst 9200 Series Switches. The port numbers of the member switches begin with the switch number, such as Gig1/0/1, Te1/1/1, or TwentyFiveGigE1/1/1, depending on whether the ports are Gigabit Ethernet (GE), 10 GE, or 25 GE.

Example: Switch 3 would have G3/0/1, Te3/1/1, TwentyFiveGigE3/1/1.

- Switch numbers are persistent, meaning that each switch keeps the same switch number after it reboots as a stack member, even if it is no longer part of the stack.
- The stack ACTIVE switch resolves any switch number conflicts and renumbers the switch.
- Switch numbering does not reflect the physical location of the switch. However, the numbering can be changed to match the physical location using the command **“switch current-stack-member-number renumber new-stack-member-number”** from the enable mode.

Example: “switch 1 renumber 2” renames switch 1 to 2 and changes the port numbers from G1/1/1 and Te1/1/1 to G2/1/1 and Te2/1/1. A reload is required for this to take effect.

When you disconnect a stack member, the switch numbers and port numbers of the remaining stack members do not change, and the stack does not reload.

The system roles in the resilient StackWise-160 architecture can be verified with the commands shown below.

```
C92-Stack#show switch
Switch/Stack Mac Address : dc8c.3777.5b00 - Local Mac Address
Mac persistency wait time: Indefinite
```

Switch#	Role	Mac Address	Priority	H/W Version	Current State
*1	Active	dc8c.3777.5b00	10	V01	Ready
2	Standby	dc8c.3777.6c00	1	V01	Ready
3	Member	dc8c.377f.af00	1	V01	Ready
4	Member	dc8c.37a3.d100	1	V01	Ready

```
C92-Stack#show redundancy
Redundant System Information :
-----
    Available system uptime = 4 days, 21 hours, 27 minutes
Switchovers system experienced = 0
    Standby failures = 0
    Last switchover reason = none

    Hardware Mode = Duplex
Configured Redundancy Mode = sso
Operating Redundancy Mode = sso
    Maintenance Mode = Disabled
    Communications = Up
```

Current Processor Information :

Active Location = slot 1

Current Software state = ACTIVE

Uptime in current state = 4 days, 21 hours, 27 minutes

Image Version = Cisco IOS Software [Gibraltar], Catalyst L3
Switch Software (CAT9K_LITE_IOSXE), Version 16.11.1, RELEASE SOFTWARE (fc3)

Technical Support: <http://www.cisco.com/techsupport>

Copyright (c) 1986-2019 by Cisco Systems, Inc.

Compiled Thu 28-Mar-19 07:19 by mcpre

BOOT = flash:packages.conf;

Configuration register = 0x102

Peer Processor Information :

Standby Location = slot 2

Current Software state = STANDBY HOT

Uptime in current state = 4 days, 21 hours, 24 minutes

Image Version = Cisco IOS Software [Gibraltar], Catalyst L3
Switch Software (CAT9K_LITE_IOSXE), Version 16.11.1, RELEASE SOFTWARE (fc3)

Technical Support: <http://www.cisco.com/techsupport>

Copyright (c) 1986-2019 by Cisco Systems, Inc.

Compiled Thu 28-Mar-19 07:19 by mcpre

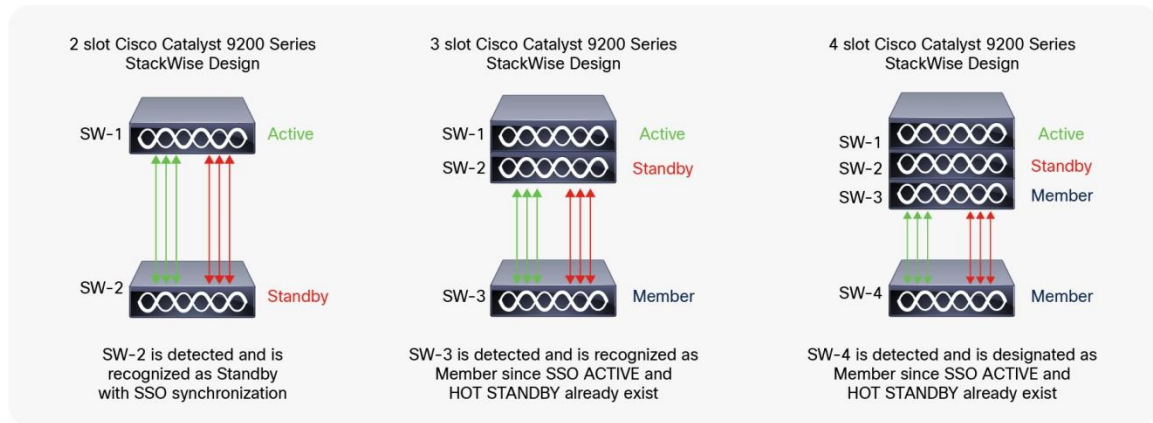
BOOT = flash:packages.conf;

Configuration register = 0x102

The STANDBY switch, which is in HOT-STANDBY mode, will transition into the ACTIVE role upon detecting failure of the primary ACTIVE switch. A new STANDBY switch will be elected from the available member switches, and it will transition to HOT-STANDBY.

Figure 7 shows the roles and operation of switches in a StackWise-160/80 architecture.

Figure 7. Roles and operation of StackWise-160/80



Stackwise-160/80 architecture

The forwarding architecture in the Cisco Catalyst 9200 Series Switches has been designed to provide up to 160G of stack bandwidth. The software architecture uses the credit-based token algorithm. To optimally forward the traffic within the stack ring, the packet-stripping function is performed on the destination switch instead of on the source or ingress switch.

This mechanism is known as spatial-reuse forwarding. It allows multiple flows to coexist to enable a parallel forwarding design. The spatial-reuse capability significantly boosts data-plane switching performance in the stack-ring switching architecture. The broadcast and multicast packets are still required to do source stripping, since the location of the destination device is known and there could be multiple multicast listener devices within the stack ring.

Distributed forwarding architecture

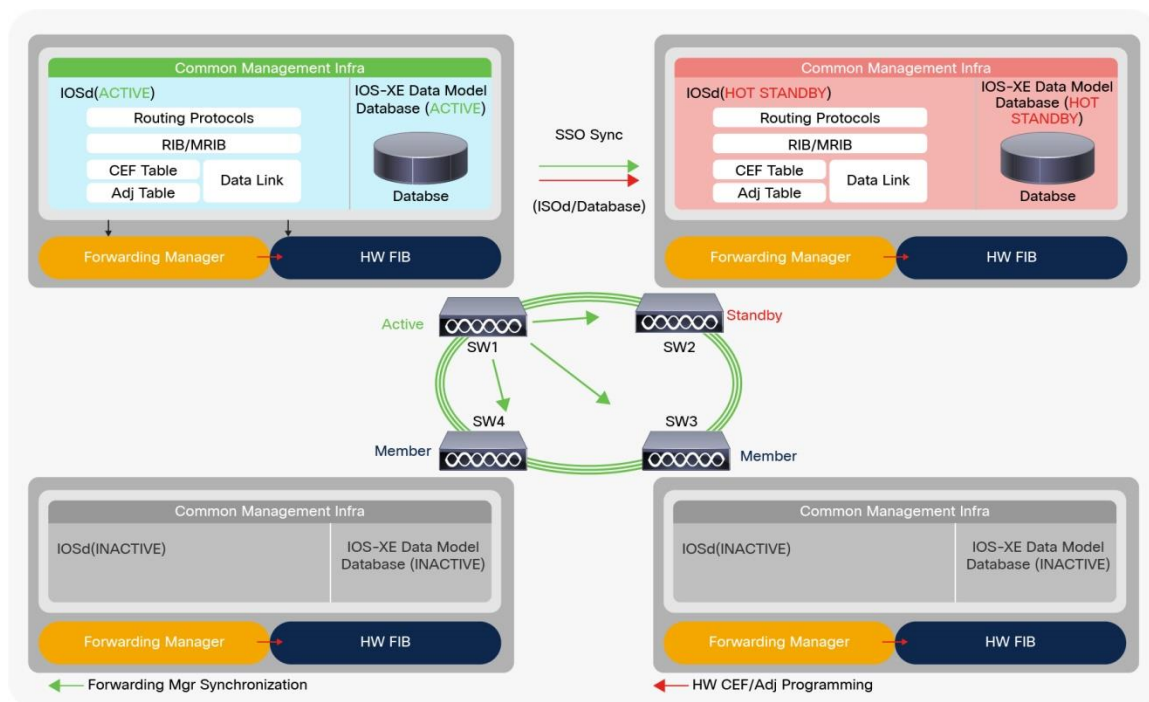
The forwarding architecture is designed to provide distributed switching across all member switches in the stack, as implemented in distributed, modular Cisco platforms. To optimize data-plane performance by using hardware resources from each Cisco Catalyst 9200 Series stack member switch, network services such as QoS, security ACLs, and others are fully distributed and programmed to be enforced locally on network ports. This distributed hardware resource utilization process delivers wire-speed switching performance that increases overall system resource capacity, prevents centralized overload processing on the ACTIVE switch, and optimizes stack-ring bandwidth capacity.

Since Cisco StackWise-160/80 technology is designed to offer modular-class system redundancy in stack design, It requires centralized control and a management plane with a distributed forwarding architecture. To enable the stacked switches to logically appear as a single virtual switch, the Cisco IOS[®] daemon (IOSd) process on the ACTIVE switch centrally manages all management-plane and network-control-plane operations with Layer 2 and Layer 3 protocols, including Spanning Tree Protocol (STP), IP routing, Cisco Express Forwarding, Policy-Based Routing (PBR), and others.

Depending on the implemented network protocols, the ACTIVE switch communicates with the rest of the multilayer or routed access infrastructure to dynamically develop the forwarding tables. The ACTIVE switch also updates all member switches for forwarding information. The distributed forwarding capability provides local switching lookup for the switching decision process. All ingress and egress wired data-plane traffic is fully distributed in a StackWise-160/80-based system design.

Independent of the IOSd operational state, the hardware forwarding information base (FIB) is actively programmed in ASICs across all stack-member switches in a stack ring. Figure 8 shows the Cisco Catalyst 9200 Series system architecture with centralized processing for control and management functions and distributed forwarding.

Figure 8. Cisco Catalyst 9200 Series and centralized processing



SSO architecture

The highly resilient stateful switchover (SSO) technology is a widely deployed solution in mission-critical campus and branch network designs. The key advantage of SSO is that it constantly delivers network availability without compromising performance and scalability during planned or unplanned network outages. The StackWise-160/80 architecture takes advantage of the same technology to maintain state machines and gracefully recover during an ACTIVE switch failure.

StackWise-160 SSO technology expands route processor redundancy (RPR) capabilities to provide transparent failover of several high-availability-aware Layer 2 and 3 protocols and Cisco IOS Software applications when the ACTIVE switchover occurs.

The state machines of non-high-availability-aware protocols and applications are not synchronized from ACTIVE to STANDBY, something the Cisco Catalyst 9200 Series Switches require to rebuild adjacencies and forwarding entries during an ACTIVE switch failure.

Implementing StackWise-160/80 SSO

To increase availability, the SSO capability is enabled by default when Cisco Catalyst 9200 Series Switches are deployed in StackWise-160/80 mode. No additional user intervention is required to enable this capability. The user can verify that SSO is configured and that the operational state is using a consistent CLI as a modular Cisco Catalyst system. The following example shows sample output of SSO redundancy states in the StackWise-160-based network design.

```
C92-Stack#show redundancy states
  my state = 13 -ACTIVE
  peer state = 8  -STANDBY HOT
    Mode = Duplex
    Unit = Primary
    Unit ID = 1

Redundancy Mode (Operational) = sso
Redundancy Mode (Configured)  = sso
Redundancy State               = sso
  Maintenance Mode = Disabled
  Manual Swact = enabled
  Communications = Up

  client count = 86
  client_notification_TMR = 30000 milliseconds
  RF debug mask = 0x0
```

Stack operations

Adding a switch

When adding a new switch to the stack, stack cables have to be connected appropriately before the switch is powered on. The stack will operate at half bandwidth until the newly added switch has been powered on and has been discovered as a member. Once the switch had been discovered and is part of the stack, the stack ring transitions to the “Full” state.

The following output shows that the stack is in the “Half” ring state

```
9200#show switch stack-ring speed

Stack Ring Speed           : 80G
Stack Ring Configuration: Half
Stack Ring Protocol         : StackWise
```

Figure 9 shows the stack cabling when three switches are operating in StackWise-160/80.

Figure 9. Stack with three switches

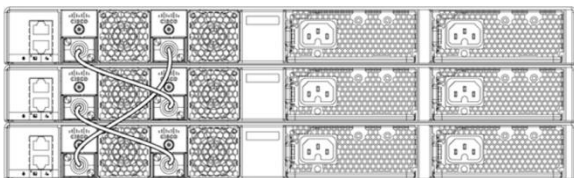
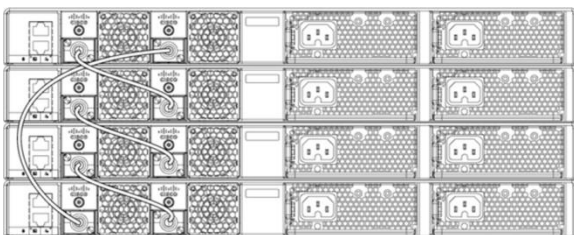


Figure 10 shows the stacking cable structure when a fourth switch has been added to the above stack.

Figure 10. StackWise-160/80 with four switches



The following logs describe the addition of a new switch to the stack.

Newly added member switch logs

```
Initializing Hardware...
```

```
<SNIP>
```

```
Current ROMMON image : Primary
```

```
Last reset cause : PowerOn
```

```
C9200-24P platform with 8388608 Kbytes of main memory
```

```
boot: attempting to boot from [flash:packages.conf]
```

```
boot: reading file packages.conf
```

```
< SNIP >
```

```
Waiting for 120 seconds for other switches to boot
```

```
The switch number is 4
```

```
All switches in the stack have been discovered. Accelerating discovery
```

The stack discovers this new switch and assigns a switch number of 4:

```
9200-STACK#  
*Jun 20 18:44:51.427: %STACKMGR-6-SWITCH_ADDED: Switch 2 R0/0: stack_mgr: Switch  
4 has been added to the stack.  
*Jun 20 18:44:51.428: %STACKMGR-6-SWITCH_ADDED: Switch 1 R0/0: stack_mgr: Switch  
4 has been added to the stack.  
*Jun 20 18:44:51.430: %STACKMGR-6-SWITCH_ADDED: Switch 3 R0/0: stack_mgr: Switch  
4 has been added to the stack.  
  
<SNIP>  
  
*Jun 20 18:44:57.034: %STACKMGR-6-SWITCH_ADDED: Switch 4 R0/0: stack_mgr: Switch  
4 has been added to the stack.  
*Jun 20 18:44:57.034: %STACKMGR-6-SWITCH_ADDED: Switch 4 R0/0: stack_mgr: Switch  
4 has been added to the stack.  
*Jun 20 18:44:57.249: %HMANRP-6-HMAN_IOS_CHANNEL_INFO: HMAN-IOS channel event for  
switch 4: EMP_RELAY: Channel UP!
```

We recommend having the stack cables connected to the newly added switch before it is powered on. If the stack cables are connected after the switch has been powered on, it will result in a reload of the newly added switch. A message similar to the following will be seen during this process on the new switch:

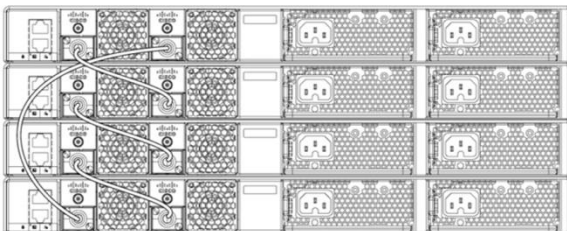
Chassis 1 reloading, reason - stack merge

Removing a switch

To remove a switch from the stack, the switch has to be powered down and the stack cables have to be disconnected from the switch. The stack will operate at half bandwidth during this state until the stack cables are reconnected to form a full ring.

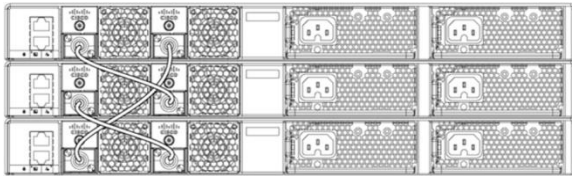
Figures 11 and 12 show stack cable connections before and after the removal of SW-4 from the stack.

Figure 11. Stack cable connections with four switches in the stack



SW-4 has been removed and the stack cable connections have been moved to bring the ring to the "Full" state.

Figure 12. Stack cable connections after SW-4 has been removed from the stack



Stack split

Stack split can occur when the backplane connectivity via the stack cables is lost due to unforeseen reasons, the member switches reload as they lose connectivity to both the ACTIVE and STANDBY switches.

```
Chassis 3 reloading, reason - lost both active and standby
Chassis 4 reloading, reason - lost both active and standby
```

In this scenario above, SW-3 and SW-4 discover each other during the reload process and the ACTIVE switch is elected. The ACTIVE switch elects the STANDBY switch. The stack ring will continue to operate at half bandwidth until the link has been restored.

Stack merge

Once the connectivity is restored on a split stack, stack merge occurs. The stack pair that has the lowest priority on the ACTIVE switch will reload and join the stack pair that has the highest priority. If the priority is the same, the ACTIVE switch in the stack pair with the highest MAC address reloads.

Auto-upgrade

One of the options that gives the StackWise-160/80 architecture greater flexibility is the ability to increase the number of ports in the wiring closet without needing to configure the management plane or upgrade the newly installed switch manually. The newly added switch automatically upgrades to the software that the stack is currently running, without any manual intervention. The hardware and software architecture of the Cisco Catalyst 9200 Series continues to provide backward compatibility to dynamically insert a new 9200 Series switch into the stack ring without a major network disruption. The system and management operation, network configuration, and topologies remain transparent for network upgrades.

As a best practice, the newly joined switch can automatically receive consistent software versions from an ACTIVE switch and bring the system online without any user intervention. To automatically download consistent software versions to newly joined switches, you can use the following command from the global configuration mode. The auto-upgrade feature is not supported in bundled mode.

```
9200-STACK#conf t
Enter configuration commands, one per line. End with CNTL/Z.
9200-STACK(config)#software auto-upgrade enable
```

With the aforementioned command enabled, any member switch that is added to the stack will automatically upgrade to the current stack software.

Stack logs

9200-STACK#

*Jun 20 22:56:20.696: %STACKMGR-6-SWITCH_ADDED: Switch 2 R0/0: stack_mgr: Switch 4 has been added to the stack.

*Jun 20 22:56:20.697: %STACKMGR-6-SWITCH_ADDED: Switch 1 R0/0: stack_mgr: Switch 4 has been added to the stack.

*Jun 20 22:56:20.740: %STACKMGR-6-SWITCH_ADDED: Switch 3 R0/0: stack_mgr: Switch 4 has been added to the stack.

*Jun 20 22:56:21.172: %BOOT-3-BOOTTIME_INCOMPATIBLE_SW_DETECTED: Switch 1 R0/0: issu_stack: Incompatible software detected. Details: Chassis 4 is detected INCOMPATIBLE with software version of Active: FAILED: Version '16.10.01' mismatch with Active's running version '16.11.01' for package: 'guestshell'

***Jun 20 22:56:21.298: %AUTO_UPGRADE-5-AUTO_UPGRADE_START_CHECK: Switch 1 R0/0: auto_upgrade_client: Auto upgrade start checking for incompatible switches.**

*Jun 20 22:56:24.452: %IOSXE_INFRA-6-PROCPATH_CLIENT_HOG: IOS shim client 'chasfs' has taken 3168 msec (runtime: 0 msec) to process a 'stack chasfs fd' message

*Jun 20 22:56:25.476: %AUTO_UPGRADE-5-AUTO_UPGRADE_INITIATED: Switch 1 R0/0: auto_upgrade_client: Auto upgrade initiated for switch 4.

*Jun 20 22:56:25.502: %AUTO_UPGRADE-5-AUTO_UPGRADE_SEARCH: Switch 1 R0/0: auto_upgrade_client: Searching stack for software to upgrade switch 4.

*Jun 20 22:56:25.518: %AUTO_UPGRADE-5-AUTO_UPGRADE_FOUND: Switch 1 R0/0: auto_upgrade_client: Found donor switch 1 to auto upgrade switch 4.

*Jun 20 22:56:25.534: %AUTO_UPGRADE-5-AUTO_UPGRADE_START: Switch 1 R0/0: auto_upgrade_client: Upgrading switch 4 with software from switch 1.

*Jun 20 22:57:05.536: %AUTO_UPGRADE_MODULAR-5-SMU_AUTO_UPGRADE_INITIATING: Switch 1 R0/0: auto_upgrade_client: Initiating SMU autoupgrade for switch 4

***Jun 20 22:57:05.904: %AUTO_UPGRADE-5-AUTO_UPGRADE_FINISH: Switch 1 R0/0: auto_upgrade_client: Finished installing software on switch 4. → upgrade complete**

***Jun 20 22:57:09.625: %AUTO_UPGRADE-5-AUTO_UPGRADE_RELOAD: Switch 1 R0/0: auto_upgrade_client: Reloading switch 4 to complete the auto upgrade. → reload after upgrade**

<SNIP >

*Jun 20 23:00:07.066: %STACKMGR-6-SWITCH_ADDED: Switch 4 R0/0: stack_mgr: Switch 4 has been added to the stack.

*Jun 20 23:00:07.066: %STACKMGR-6-SWITCH_ADDED: Switch 4 R0/0: stack_mgr: Switch 4 has been added to the stack.

*Jun 20 23:00:07.736: %HMANRP-6-HMAN_IOS_CHANNEL_INFO: HMAN-IOS channel event for switch 4: EMP_RELAY: Channel UP!

Licensing

Smart Licensing

With Release 16.9.1, for which a Smart License is mandatory, the stack automatically enables the EVAL license on a newly added switch should a license mismatch be present on it. Data programming will occur once the switch joins the stack.

License usage can be verified using the following command.

```
9200-STACK#sh license usage
License Authorization:
  Status: AUTHORIZED on May 25 22:53:33 2019 UTC

C9300 Network Advantage (C9200-24 Network Advantage):
  Description: C9200-24P Network Advantage
  Count: 4 → Number of switches in stack
  Version: 1.0
  Status: AUTHORIZED

C9200 DNA Advantage (C9200-24 DNA Advantage):
  Description: C9200-24P DNA Advantage
  Count: 4
  Version: 1.0
  Status: AUTHORIZED
```

Stack design

Simplified multilayer network design

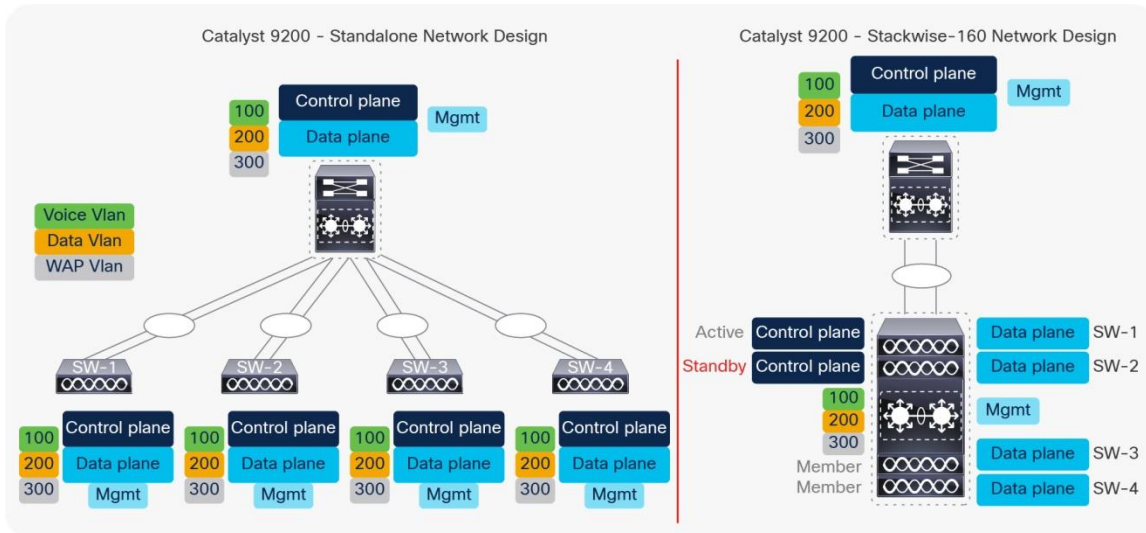
The enterprise customer relies on real-time business applications such as voice, video, and others. System reliability and network availability become core, integrated service requirements to deliver nonstop communication in the network.

When the access layer environment becomes highly dense, StackWise-160/80 pools up to eight physical chassis into a single logical system from a network design perspective. As the access layer network infrastructure expands, the device-pooling capability of the StackWise-160/80 technology significantly simplifies operations and the network architecture itself.

Cisco recommends designing and deploying the multilayer distribution block with four major elements in the overall architecture: reduced fault domain, increased network security, deterministic forwarding paths, and optimal resiliency. Designing and developing a wiring closet with this architecture requires isolated broadcast domains or VLANs for each workgroup category, device, and application type. This network design needs to be applied consistently throughout the wiring closet network. It provides solid network security, stability, and reliability, but depending on the access layer network size, it may increase operational and troubleshooting complexity due to the increased number of VLANs, subnets, neighbor counts, and more.

The Cisco Catalyst 9200 Series StackWise-160/80 device-pooling design retains Cisco's multilayer design principles. It also simplifies operational challenges with a reduced number of VLANs, STP instances, subnets, neighbor counts, etc. at the access and distribution layers. Figure 13 shows the simplified network design and operational data points of 9200 Series switches deployed in standalone mode compared to StackWise-160/80 mode.

Figure 13. Comparison between Cisco Catalyst 9200 Series Switches in standalone mode and a StackWise



Scalable StackWise-160/80 cross-stack EtherChannel design

StackWise-160/80 can make it possible to build a single-uplink EtherChannel interface by bundling up to eight parallel physical links, which could then be distributed evenly across all stack switches. Multiple uplinks from a mission-critical access layer switch are a base requirement for reliable networks to provide high-speed data load sharing and to deliver 1+1 path redundancy upon failure.

However, from a forwarding perspective the Layer 2 network becomes suboptimal when multiple parallel interfaces are deployed between two Layer 2 Ethernet switches.

The cross-stack EtherChannel interface simplifies STP topology over the logical interface and builds loop-free forwarding paths across all bundled uplink ports. If the Cisco Catalyst 9200 Series Switch is deployed in routed access mode, the cross-stack EtherChannel can also be deployed as a Layer 3 uplink interface to simplify unicast or multicast neighbor adjacencies as well as to simplify network topology with one forwarding table. Regardless of the 9200 Series switch's deployed mode (multilayer or routed access), during a member switch or member link failure the EtherChannel does not disrupt network topology tables and delivers deterministic, hardware-driven, subsecond network recovery processes.

Cisco Catalyst 9200 Series Switches support multiple physical uplink ports to connect distribution-layer switches. Typically, up to two physical uplink interfaces are deployed from access layer switches for optimal load balancing and redundancy in the wiring closet.

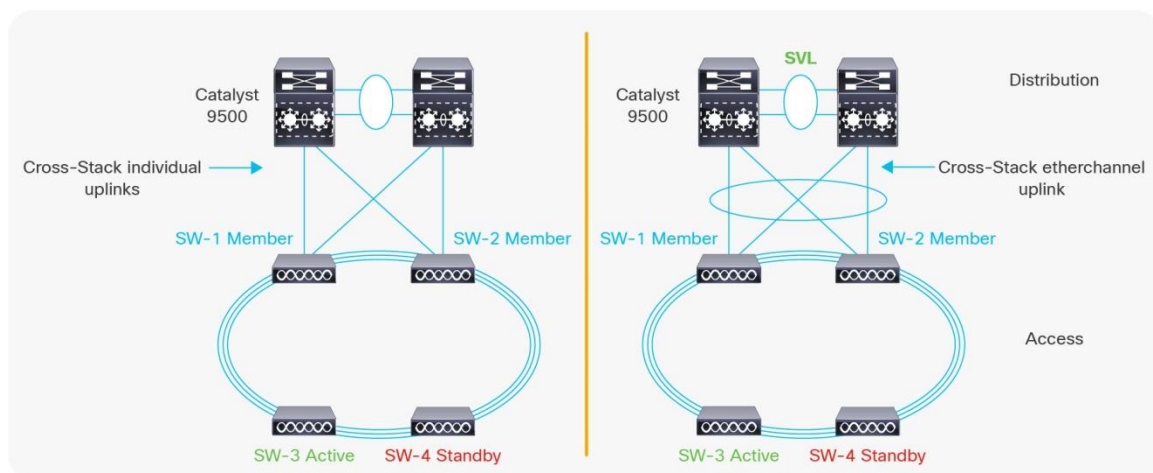
When these switches are deployed in stack configuration mode, we recommend maintaining the same uplink connection design principle as for a dual stack-member system. The recommendation is to use uplinks on the member switches with multichassis EtherChannel (MEC), which helps with spanning multiple uplinks over the member switches. For example, eight Cisco Catalyst 9200 Series Switches deployed in a stack ring would have multiple diversified uplink ports from member switches. The remaining switches, where uplinks are not connected, would forward the data toward the core using a high-speed stack backplane.

This recommended uplink port design offers various benefits, from enhanced application performance to optimal user experience. Some key benefits include:

- Improved application performance by increasing aggregated stack switching capacity with multiple distributed, high-speed 10-Gbps/25-Gbps uplinks between stack-member switches
- Enhanced bidirectional traffic engineering with intelligent network data load sharing within the stack ring and across all distributed uplink physical ports
- Improved system and application performance by using the distributed forwarding architecture advantage of hardware resources: buffers, queues, Ternary Content-Addressable Memory (TCAM), and others
- Protection of the stack and network-level redundancy and reduction in congestion between distributed aggregation systems during a major outage at the access or distribution layer

Figure 14 illustrates a sample uplink network design for a Cisco Catalyst 9200 Series StackWise-160/80 distribution layer system.

Figure 14. Cisco Catalyst 9200 Series StackWise-160/80 uplink design best practices



Optimal StackWise-160/80 cross-stack forwarding EtherChannel design

The egress data load forwarding from StackWise-160/80 is determined based on how the upstream network is designed. The loop-free forwarding topology uses all available paths to switch data traffic based on precomputed Cisco Express Forwarding or EtherChannel hash results. In a distributed forwarding architecture, the Cisco Catalyst 9200 Series stack switches verify the Layer 2 to Layer 3 data variables from incoming traffic to determine the best physical uplink ports before forwarding traffic to the upstream system.

The Cisco Catalyst 9200 Series Switches require more variables in packet tuples to perform granular switching decisions. In a large-scale design, the source MAC address-based EtherChannel load-sharing mode may deliver adequate results to enable the use of all upstream member links. However, in mid- to low-scale networks, the switches may not have enough variable points to compute the best egress uplink path. In this case, to optimize the switching performance with granular packet forwarding decisions across all available cross-stack uplink paths, the default EtherChannel hash computation can be tuned to include Layer 2 to Layer 3 address variables.

The next-generation Cisco Catalyst 9200 Series is designed to support a large number of EtherChannel hash variables to deliver optimal upstream egress forwarding decisions. Table 3 outlines the supported Layer 2 to Layer 4 EtherChannel hash algorithm.

Table 3. Cisco Catalyst 9200 Series EtherChannel hash algorithm

Layer	EtherChannel hash
Non-IP	src-mac (default)
Layer 2	src-mac (default) dst-mac src-dst-mac
Layer 3	src-ip dst-ip src-dst-ip
Layer 4	src-port dst-port src-dst-port
Layer 3 plus Layer 4	src-mixed-ip-port dst-mixed-ip-port src-dst-mixed-ip-port (Recommended)

The network administrator can adjust the default EtherChannel hash algorithm from a global configuration mode, as demonstrated in the sample code that follows.

Verification:

```

9200-STACK#show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
    src-mac

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source MAC address
IPv4: Source MAC address
IPv6: Source MAC address

```

Command to change the default load-balance method:

```
9200-STACK#conf t
9200-STACK(config)#port-channel load-balance src-dst-mixed-ip-port
```

Verification:

```
9200-STACK#sh etherchannel load-balance
EtherChannel Load-Balancing Configuration:
      src-dst-mixed-ip-port

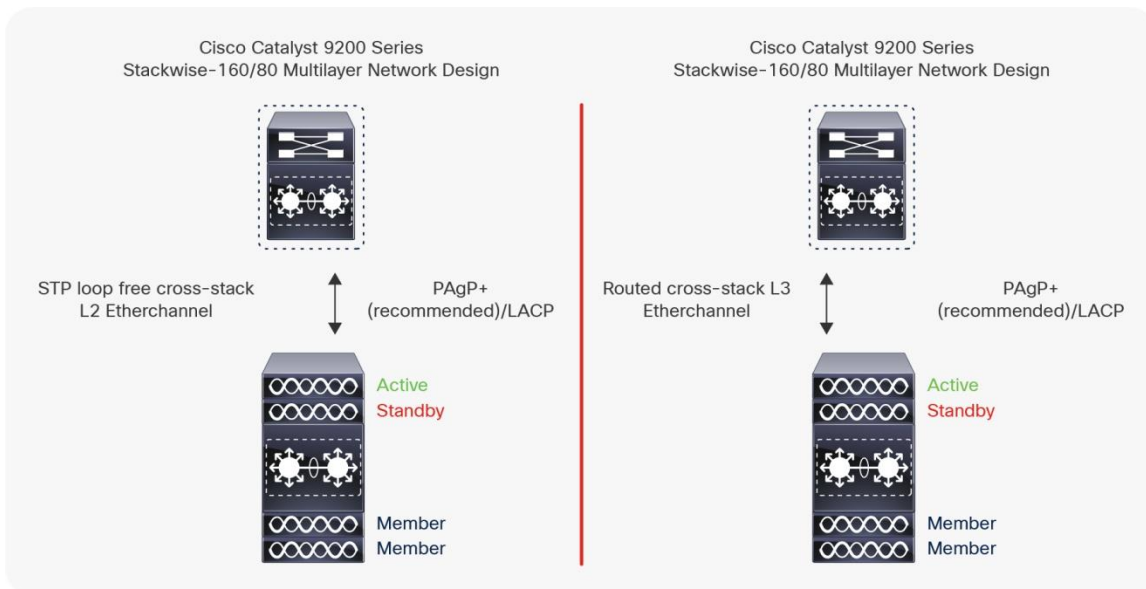
EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source XOR Destination MAC address
IPv4: Source XOR Destination IP address and TCP/UDP (layer-4) port number
IPv6: Source XOR Destination IP address and TCP/UDP (layer-4) port number
```

Reliable StackWise-160/80 cross-stack forwarding Ether Channel design

The link aggregation protocols build stateful, consistent, and reliable EtherChannel communication between two systems. To successfully establish a logical EtherChannel interface between two systems, the link aggregation protocol performs several link parameter checks to assure that each member link is equipped to deliver consistent switching performance and network service in the event of failure. During the EtherChannel startup process, each end of the system verifies the capabilities of each local and remote member link, including attributes such as speed, duplex, protocol dependencies, QoS capabilities, and more.

Cisco recommends bundling the cross-stack EtherChannel interface using link aggregation protocols such as Cisco Port Aggregation Protocol Plus (PAgP+) or Link Aggregation Control Protocol (LACP). Cisco Catalyst 9200 Series Switches deployed in StackWise-160/80 support both of these link aggregation protocols.

Figure 15. Cisco Catalyst 9200 Series cross-stack EtherChannel design



Summary

Next-generation Cisco Catalyst 9200 Series Switches have been designed to meet future demands in wiring closet networks. Stackwise-160/80 provides maximum port density at the access layer, along with platform, software, and network resiliency at the access layer. As more technologies are integrated into the system, the Cisco Catalyst 9200 Series offers operational simplicity, scalability, performance, and adaptability for future protocols. The software architecture of Cisco StackWise-160/80 technology delivers superior performance and best-in-class resiliency along with the flexibility of the UADP ASIC. This document has focused primarily on the StackWise architecture for the Cisco Catalyst 9200 Series Switches.



Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at <https://www.cisco.com/go/offices>.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)