

# CSS Technology White Paper

**Issue**      01  
**Date**        2013-04-09

HUAWEI TECHNOLOGIES CO., LTD.



**Copyright © Huawei Technologies Co., Ltd. 2013. All rights reserved.**

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

## **Trademarks and Permissions**



and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

## **Notice**

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

## **Huawei Technologies Co., Ltd.**

Address: Huawei Industrial Base  
Bantian, Longgang  
Shenzhen 518129  
People's Republic of China

Website: <http://enterprise.huawei.com>

Email: [ChinaEnterprise\\_TAC@huawei.com](mailto:ChinaEnterprise_TAC@huawei.com)

---

# Contents

---

<b>1 CSS</b> .....	<b>1</b>
1.1 CSS Overview .....	1
1.2 Principles.....	3
1.2.1 Concepts.....	3
1.2.2 Principles .....	5
1.3 Applications.....	10
1.4 Example for Configuring the CSS Function .....	11
<b>A A Terms, Acronyms, and Abbreviations</b> .....	<b>16</b>

# 1 CSS

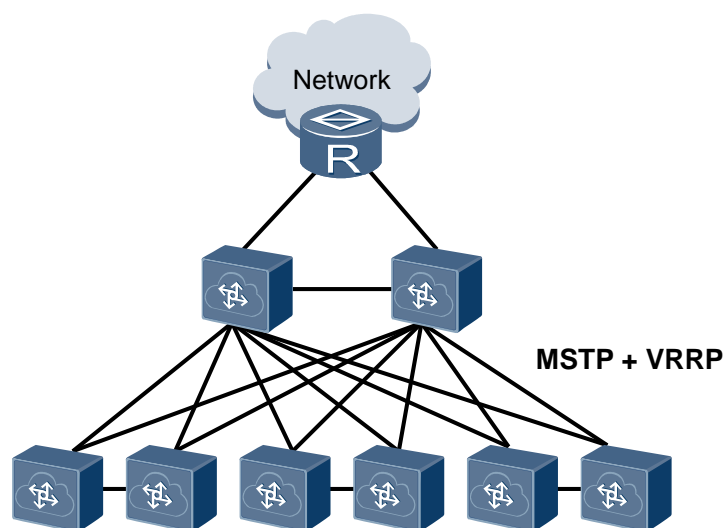
## About This Chapter

- 1.1 CSS Overview
- 1.2 Principles
- 1.3 Applications
- 1.4 Example for Configuring the CSS Function

## 1.1 CSS Overview

As shown in [Figure 1-1](#), dual-node redundancy design is used at the core layer and the aggregation layer to improve network reliability.

**Figure 1-1** Redundant network structure

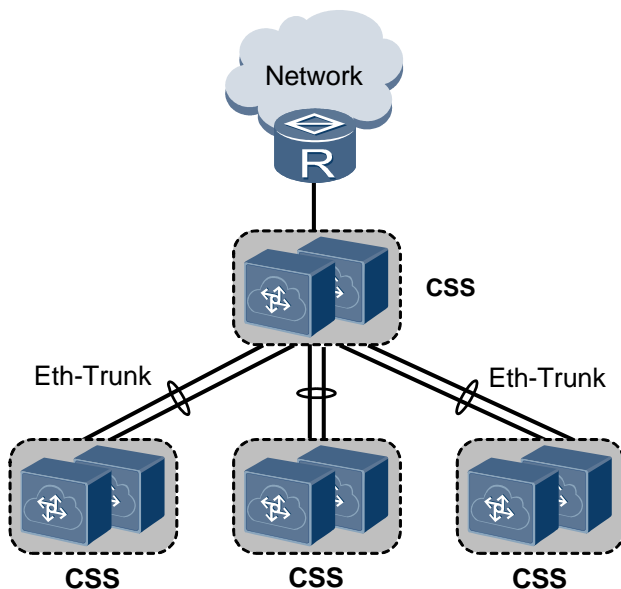


Dual-node redundancy design improves network reliability, but complicates network structure and connections. Loop-prevention protocols, such as MSTP, often need to be deployed to

prevent loops, and VRRP needs to run to support node redundancy backup. These requirements complicate network protocol deployment.

A Cluster Switch System (CSS), also called a stack, virtualizes multiple switches supporting the stacking function into one logical device as shown in [Figure 1-2](#).

**Figure 1-2** CSS networking

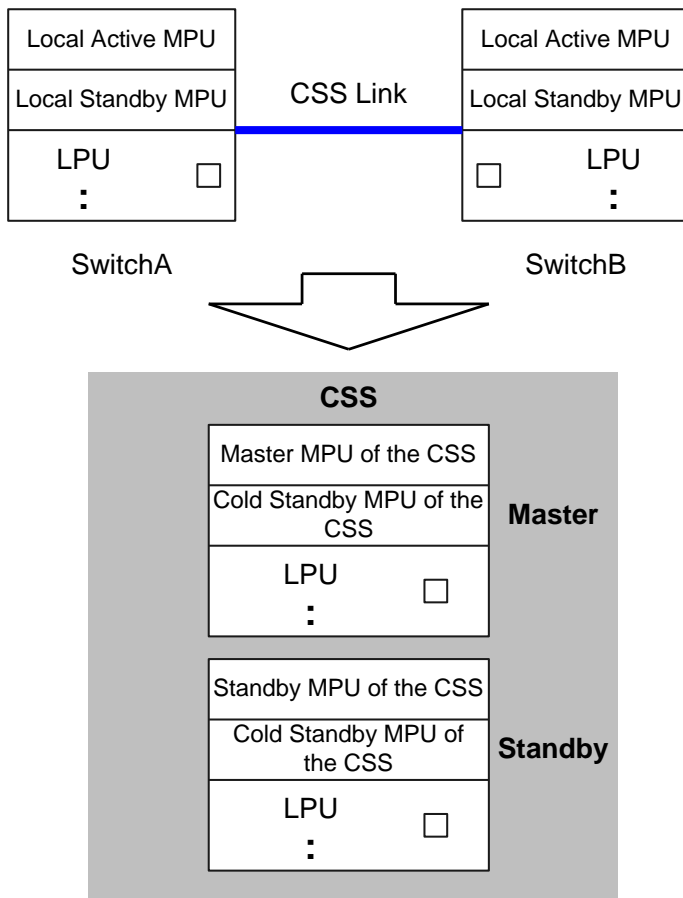


A CSS combines control planes and forwarding planes of two switches together, and has the following advantages:

- High reliability  
Redundancy backup is implemented between member switches in a CSS. In addition, a CSS supports inter-chassis link aggregation to implement inter-chassis link backup.
- Simplified network structure and protocol deployment  
The CSS technology simplifies a complex network topology into a simple and hierarchical network structure, connects network layers through link aggregation, and prevents loops without requiring protocols like MSTP and VRRP to be deployed.
- Simplified configuration and management  
After a CSS is set up, multiple physical devices are virtualized into one logical device. You can log in to the CSS and manage all the member devices in a unified manner.

After a CSS is set up, a master switch and a standby switch will be elected. On the control plane, the active MPU of the master switch becomes the master MPU of the CSS and plays a major role in CSS management. The active MPU of the standby switch becomes the standby MPU of the CSS and plays a standby role in CSS management. The standby MPUs of the master switch and the standby switch are become cold standby MPUs of the CSS.

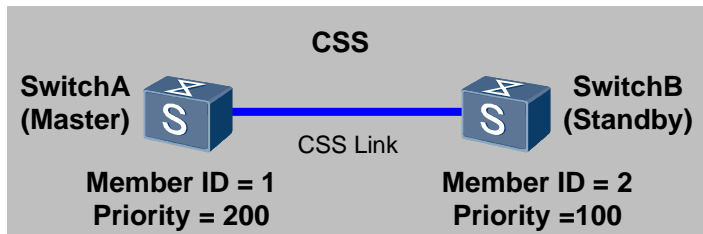
**Figure 1-3** Master competition



## 1.2 Principles

### 1.2.1 Concepts

**Figure 1-4** CSS networking diagram



- **Switch roles**  
 Each switch in a CSS is a member switch. Member switches are classified into the following roles:
  - Master switch

A master switch manages the CSS. A CSS has only one master switch.

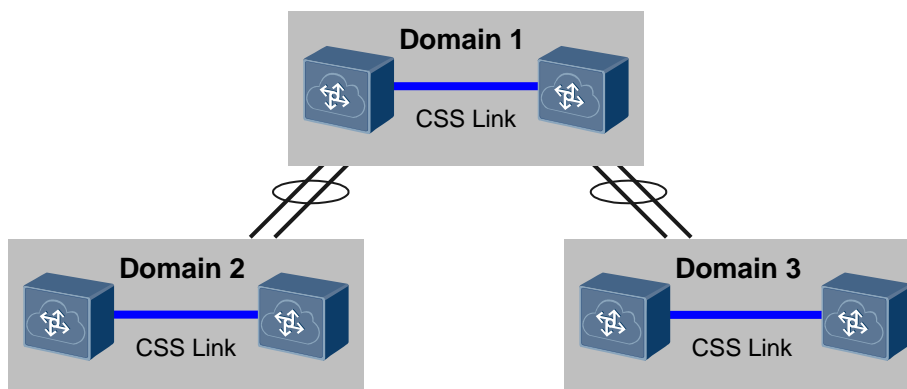
- **Standby switch**

The standby switch is the backup to the master switch. When the master switch becomes faulty, the standby switch takes over the master role. A CSS has only one standby switch.

- **Stack domain**

Switches that connect to each other using stack links to form a CSS belong to a stack domain. To meet various networking requirements, you can configure multiple CSS on a network and use stack domain IDs to identify these CSSs as shown in [Figure 1-5](#).

**Figure 1-5** Multiple stack domains



- **Stack ID**

A stack ID, also called a member ID, is used to identify and manage member switches in a CSS. All member switches in a CSS have a unique stack ID.

- **Stack priority**

The stack priority is an attribute of member switches, which helps determine the role of member switches in role election. A larger priority value indicates a higher priority. The member switch with a higher stack priority has a higher probability of becoming the master switch.

- **Physical member interface**

Switches connect to each other to form a CSS using physical member interfaces. Physical member interfaces forward inter-chassis service packets and stack protocol packets between member switches.

- **Stack interface**

A stack interface is a logical interface that is bound to physical member interfaces to implement the stacking function of service interface connection. Each member switch has one stack interface, which is named Stack-Portn/1. *n* specifies the stack ID of the member switch.

## 1.2.2 Principles

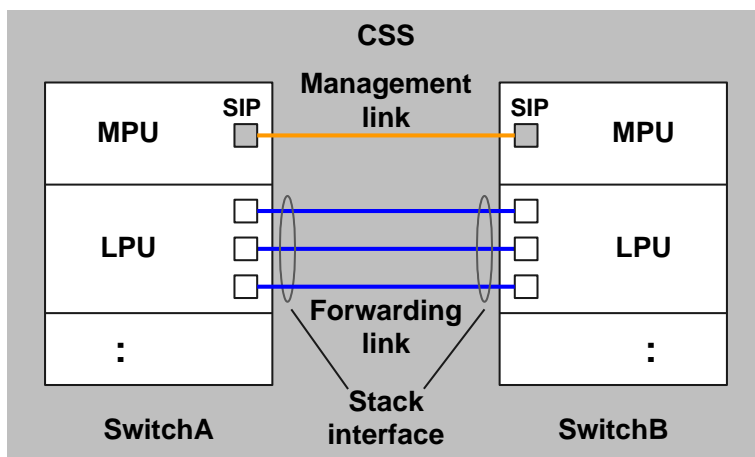
### Physical Connection in a CSS

To use multiple switches to form a CSS, connect physical member interfaces bound to the local stack interface to those bound to the neighbor stack interface. A stack interface can be bound to multiple physical member interfaces to improve stack link bandwidth and reliability.

A stack link contains the management link and forwarding link. The management link forwards stack management and maintenance packets, and the forwarding link forwards service data packets between member switches. Two connection modes are available for the stack management link:

- Direct connection between MPUs: The stack management link is established using dedicated system inter-connect ports (SIPs) of MPUs as shown in [Figure 1-6](#). This connection mode improves link reliability and saves service resources.

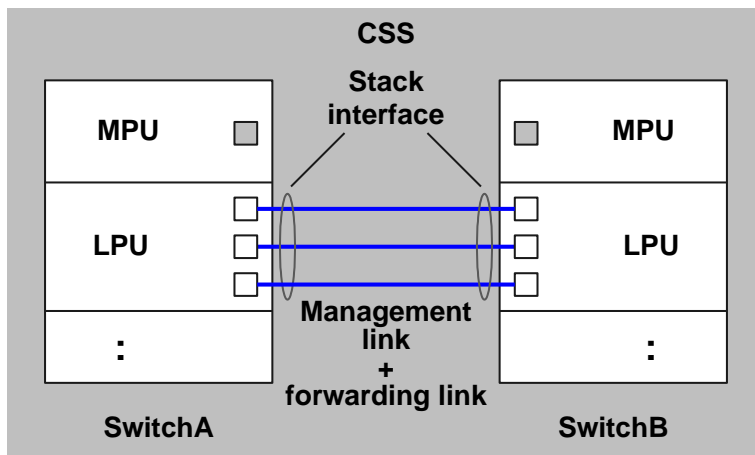
**Figure 1-6** Direct connection between MPUs



**Direct connection between main control boards**

- Direct connection between LPUs: Like the forwarding link, the management link is established using service ports of line cards as shown in [Figure 1-7](#). This connection mode simplifies network deployment and maintenance.



**Figure 1-7** Direct connection between line cards**Direct connection between service boards**

## Setting Up a CSS

After a CSS is set up, member switches send competition packets to each other to elect the master switch. A switch is elected as the master switch to manage the CSS. The other switch becomes the standby switch.

The rules for electing the master switch are as follows. Start from the first rule until the master switch is elected.

1. The switch that has started is preferred over the switch that is starting.
2. The switch with higher stack priority is preferred.
3. The switch with a later software version is preferred. The V, R, C, and SPC versions are compared in sequence.
4. The switch that has two MPUs is preferred over the switch that has one MPU.
5. The switch with a smaller MAC address is preferred.

Before a CSS is set up, each switch is an independent entity and has its own IP address. You need to manage the switches separately. After the CSS is set up, the switches in the CSS form a logical entity, and you can use a single IP address to manage and maintain the switches uniformly. The IP address and MAC address of the CSS is the IP address and MAC address of the master switch when the CSS is set up for the first time. When the master and standby switches use different software versions, standby switches synchronize the software version with the master switch, restarts, and then joins the CSS.

## Managing and Maintaining a CSS

After a CSS is set up, all member switches function as one logical switch on the network, and the master switch manages the resources of all member switches. You can log in to the CSS through the management network interface or serial interface of any member switch to manage and maintain the CSS.

On a single switch that does not join any CSS, the interface number is in the format of slot ID/subcard ID/interface sequence number. After the switch joins a CSS, the interface number is in the format of stack ID/slot ID/subcard ID/interface sequence number. For example, when

a switch does not join a CSS, the interface number is 10GE1/0/1; when the switch joins a CSS, the interface number is 10GE2/1/0/1 if the stack ID is 2.

## Active/Standby Switchover

After the active/standby switchover is performed on MPUs on the master switch or standby switch, the roles of the switch and MPU changes as follows:

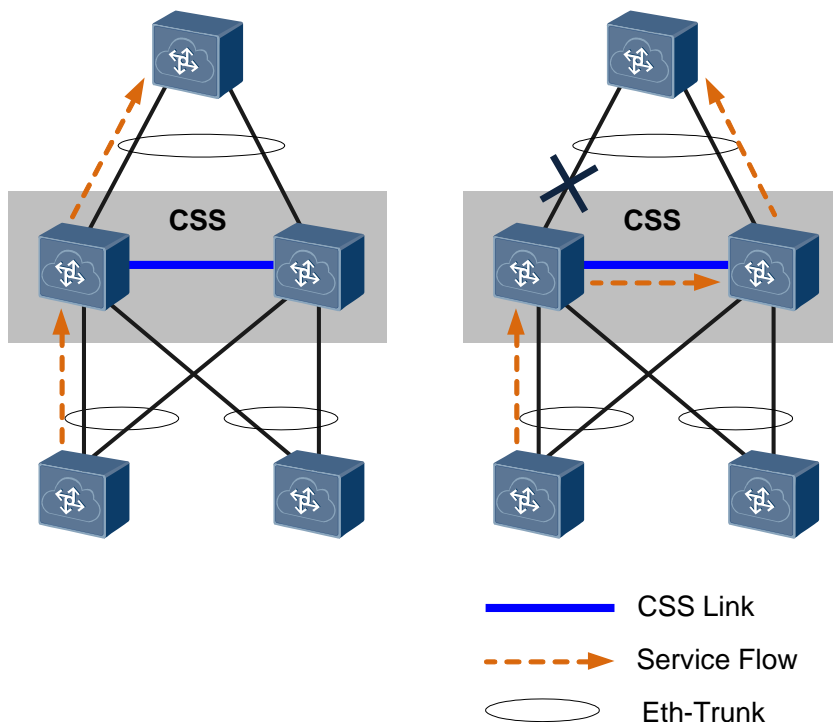
- When an active/standby switchover is performed on the two MPUs of the master switch:
  - The standby switch becomes the master switch, and the previous standby MPU of the CSS becomes the new master MPU of the CSS.
  - The master switch becomes the standby switch.
  - The previous master MPU of the CSS restarts. The cold standby MPU of the previous master switch becomes the standby MPU of the CSS and synchronizes data from the new master MPU of the CSS.
- When an active/standby switchover is performed on the two MPUs of the standby switch:
  - The master switch and the standby switch do not change their roles.
  - The active MPU of the standby switch (the previous standby MPU of the CSS) restarts. The cold standby MPU of the CSS becomes the standby MPU of the CSS and synchronizes data from the active MPU of the CSS.

## Preferential Forwarding of Local Traffic

As shown in [Figure 1-8](#), two switches form a CSS. To ensure reliable traffic transmission, an inter-chassis Eth-Trunk interface is configured as the outbound interface of traffic. Data from a downstream switch enters the CSS from the Eth-Trunk interface and is forwarded preferentially through the local upstream Eth-Trunk to an upstream switch.

When a member interface in the Eth-Trunk fails, traffic can be transmitted between devices through stack links. This ensures reliable data transmission.

Figure 1-8 Preferential forwarding local traffic

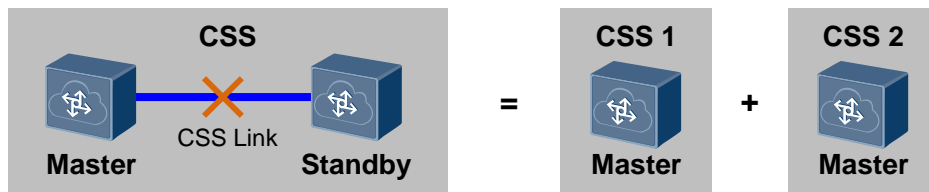


### CSS Split

After a CSS is set up, the master switch and the standby switch periodically send heartbeat packets to each other to maintain the CSS status. If a stack cable is faulty, communication between the two switches may fail. In this case, the heartbeat packets between the two switches time out and the CSS splits into two independent switches as shown in Figure 1-9.

If the two switches run normally after the CSS splits, they use the same IP address and same MAC address to communicate with other devices on the network because their global configurations are the same. This causes conflicts of IP addresses and MAC addresses and faults on the entire network.

Figure 1-9 CSS split



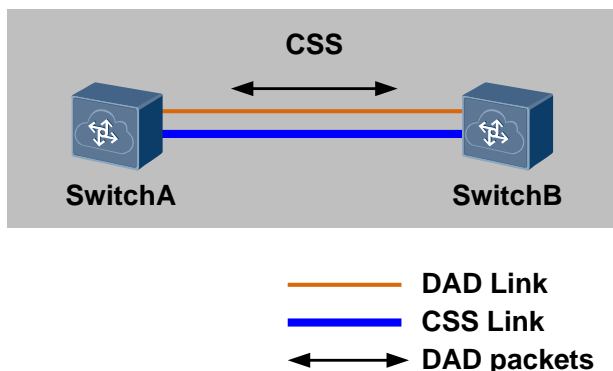
### Dual-active Detection

Dual-active detection (DAD) is a method to detect a dual-active scenario and take recovery action, ensuring network stability.

DAD has two modes:

- DAD in direct mode  
As shown in Figure 1-10, DAD is performed between member switches in a CSS using a dedicated direct link.

Figure 1-10 DAD in direct mode

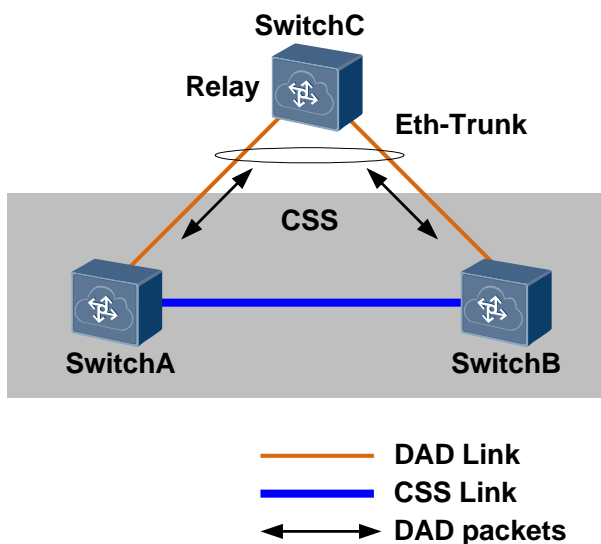


- DAD in relay mode  
As shown in Figure 1-11, DAD is configured on the inter-chassis Eth-Trunk in a CSS, and DAD in relay mode is configured on the proxy device.

**NOTE**

To enable DAD packets to be forwarded between Eth-Trunk member links, the proxy device must support DAD in relay mode. Huawei CloudEngine series switches support DAD in relay mode.

Figure 1-11 DAD in relay mode



After a DAD link is configured, CSSs exchange DAD packets on the DAD link. After a CSS splits into multiple CSSs, the CSSs compare information in received DAD packet with local information. If the switch in a CSS is elected as the master switch, the switch remains Active and continues forwarding service packets. If the switch in a CSS is elected as the standby switch, the switch shuts down all its service interfaces except those excluded from shutdown, enters the Recovery state, and stops forwarding service packets.

The rules for electing the master switch are as follows:

1. The switch with higher stack priority is preferred.
2. The switch with two MPUs is preferred over the switch with only one MPU.
3. The switch with a smaller MAC address is preferred.

The election results are compared one by one. In the case of the same election result, the next rule is used until the master switch is elected.

After the stack link recovers, the switch in Recovery state restarts and restores all the blocked service interfaces.

## Fast CSS Upgrade

Fast CSS upgrade is a mechanism that upgrades the software versions of member switches in a CSS without interrupting service forwarding. This mechanism reduces the impact of device upgrade on services.

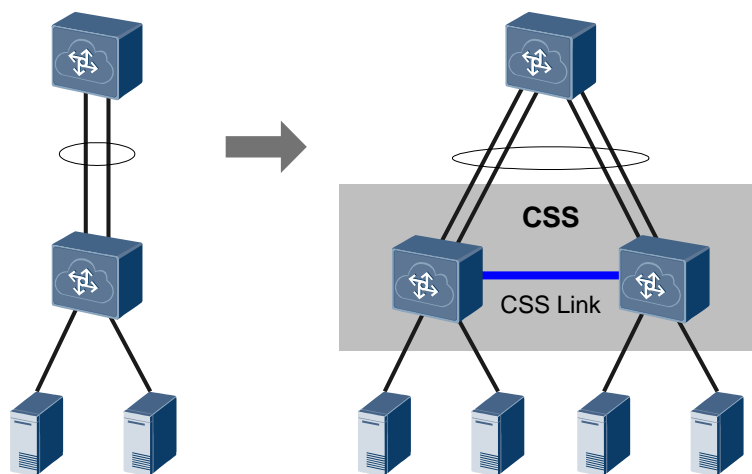
During fast CSS upgrade, the standby switch restarts using the new version, and the master switch forwards data traffic. If the upgrade fails, the standby switch restarts and rolls back to the previous version. After the standby switch is upgraded, it becomes the master switch and forwards data traffic. The previous master switch restarts using the new version. After the upgrade, the switch becomes the standby switch.

## 1.3 Applications

### Increasing Bandwidth

As shown in [Figure 1-12](#), when the uplink bandwidth of a switch increases, you can enable this switch to work with another switch to form a CSS, and configure multiple physical links of the two member switches into a link aggregation group to increase the uplink bandwidth of the switch.

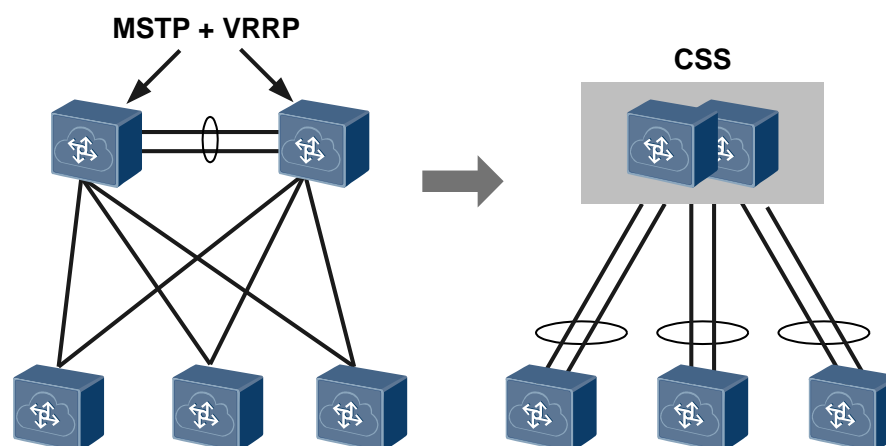
**Figure 1-12** Increasing bandwidth



## Simplifying Networking

As shown in [Figure 1-13](#), multiple devices on the device form a CSS and are virtualized into a single logical device. The simplified networking does not require MSTP or VRRP, simplifying the network configuration. In addition, inter-chassis link aggregation implements fast convergence and improves network reliability.

**Figure 1-13** Simplifying networking



## 1.4 Example for Configuring the CSS Function

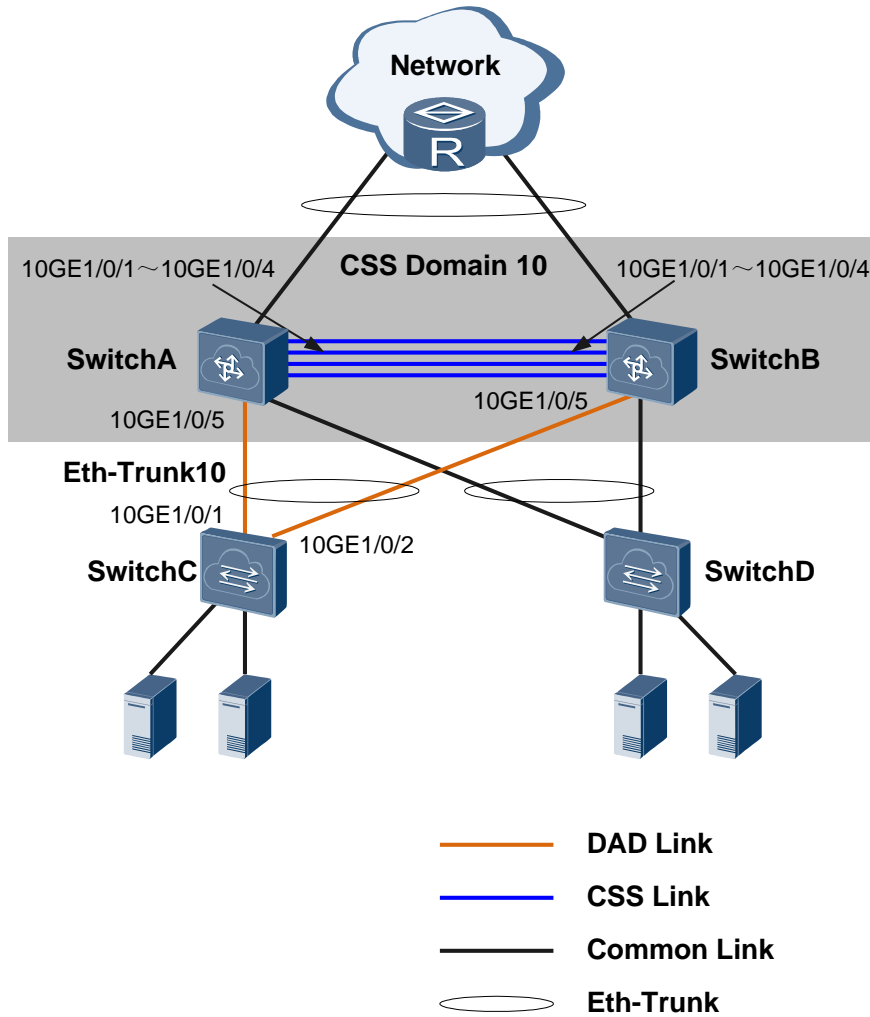
### Networking Requirements

As the network expands rapidly, the network forwarding capability needs to be increased, redundancy backup is required between devices to ensure high network reliability, and the network must be easy to manage and maintain. However, a single core switch cannot meet these requirements.

As shown in [Figure 1-14](#), SwitchA and SwitchB form a CSS, and service interfaces 10GE1/0/1 through 10GE1/0/4 are added to a stack interface.

SwitchA and SwitchB connect to SwitchC through Eth-Trunk10. Dual-active detection (DAD) needs to be configured on Eth-Trunk10.

**Figure 1-14** Networking diagram for CSS configuration



## Procedure

- Step 1** Set the domain ID of SwitchA and SwitchB to 10 and set the CSS connection mode to direct MPU connection. Set the stack priority of SwitchA to 10, and set the stack ID of SwitchB to 2.

```
<HUAWEI> system-view
[~HUAWEI] sysname SwitchA
[~HUAWEI] commit
[~SwitchA] stack
[~SwitchA-stack] stack domain 10
[~SwitchA-stack] stack priority 10
[~SwitchA-stack] stack link-type mainboard-direct
[~SwitchA-stack] commit

<HUAWEI> system-view
[~HUAWEI] sysname SwitchB
[~HUAWEI] commit
[~SwitchB] stack
```

```
[~SwitchB-stack] stack domain 10
[~SwitchB-stack] stack member 2
[~SwitchB-stack] stack link-type mainboard-direct
[~SwitchB-stack] commit
```

**Step 2** Configure service interfaces 10GE1/0/1 to 10GE1/0/4 on SwitchA and SwitchB as physical member interfaces and add them to a stack interface.

```
[~SwitchA] stack
[~SwitchA-stack] port mode stack interface 10ge 1/0/1 to 1/0/4
[~SwitchA-stack] commit
[~SwitchA-stack] quit
[~SwitchA] interface stack-port 1
[~SwitchA-Stack-Port1] port member-group interface 10ge 1/0/1 to 1/0/4
[~SwitchA-Stack-Port1] quit
[~SwitchA] commit
```

The configuration of SwitchB is the same as that of SwitchA.

**Step 3** Connect a stack link and enable the CSS function on SwitchA and SwitchB. Restart SwitchA and SwitchB.

```
[~SwitchA] stack
[~SwitchA-stack] stack enable
Warning: Make sure to configure one or more dual-active detection methods once t
he conversion is complete and the switches have come
up in stack mode.
Current configuration will be converted to the next startup saved-configuration
file of stack mode.
System will reboot! Continue? [Y/N]: y
```

The configuration of SwitchB is the same as that of SwitchA.

**Step 4** After the CSS is set up, check CSS information. The following information shows that SwitchA is the master switch of the CSS.

```
<SwitchA> display stack
-----
MemberID Role      Mac                Priority  Device Type
-----
1         Master  006d-8835-2b00    10       CE12804
2         Standby 006d-8835-2c00    100      CE12804
-----
```

**Step 5** Configure DAD in relay mode.

```
<SwitchA> system-view
[~SwitchA] interface eth-trunk 10
[~SwitchA-Eth-Trunk10] trunkport 10ge 1/1/0/5
[~SwitchA-Eth-Trunk10] trunkport 10ge 2/1/0/5
[~SwitchA-Eth-Trunk10] dual-active detect mode relay
[~SwitchA-Eth-Trunk10] commit

<HUAWEI> system-view
[~HUAWEI] sysname SwitchC
```



```
[~HUAWEI] commit
[~SwitchC] interface eth-trunk 10
[~SwitchC-Eth-Trunk10] trunkport 10ge 1/0/1
[~SwitchC-Eth-Trunk10] trunkport 10ge 1/0/2
[~SwitchC-Eth-Trunk10] dual-active proxy
[~SwitchC-Eth-Trunk10] commit
```

----End

## Configuration Files

- Configuration file of the CSS

```
#
sysname SwitchA
#
interface 10GE1/1/0/1
port mode stack
stack-port 1/1
#
interface 10GE1/1/0/2
port mode stack
stack-port 1/1
#
interface 10GE1/1/0/3
port mode stack
stack-port 1/1
#
interface 10GE1/1/0/4
port mode stack
stack-port 1/1
#
interface 10GE1/1/0/5
eth-trunk 10
#
interface 10GE2/1/0/1
port mode stack
stack-port 2/1
#
interface 10GE2/1/0/2
port mode stack
stack-port 2/1
#
interface 10GE2/1/0/3
port mode stack
stack-port 2/1
#
interface 10GE2/1/0/4
port mode stack
stack-port 2/1
#
interface 10GE2/1/0/5
eth-trunk 10
#
interface Eth-Trunk10
```

```
    dual-active detect mode relay
#
return
```

- Configuration file of SwitchC

```
#
sysname SwitchC
#
interface 10GE1/0/1
 eth-trunk 10
#
interface 10GE1/0/2
 eth-trunk 10
#
interface Eth-Trunk10
 dual-active proxy
#
return
```

# A

## A Terms, Acronyms, and Abbreviations

Term/Acronym/Abbreviation	Full Name	Description
CSS	Cluster Switch System	A cluster switching system, also called a stack
	Dual-Active Detection	A function that can detect the problem of double masters in a stack
Eth-Trunk	-	A technology of binding multiple physical interfaces into a logical interface to increase bandwidth, also called link aggregation.
LPU	Line Processing Unit	Interface card
MPU	Main Processing Unit	Main control unit of a device
SIP	System Inter-Connect Port	A port that connects to the stack management link on the MPU