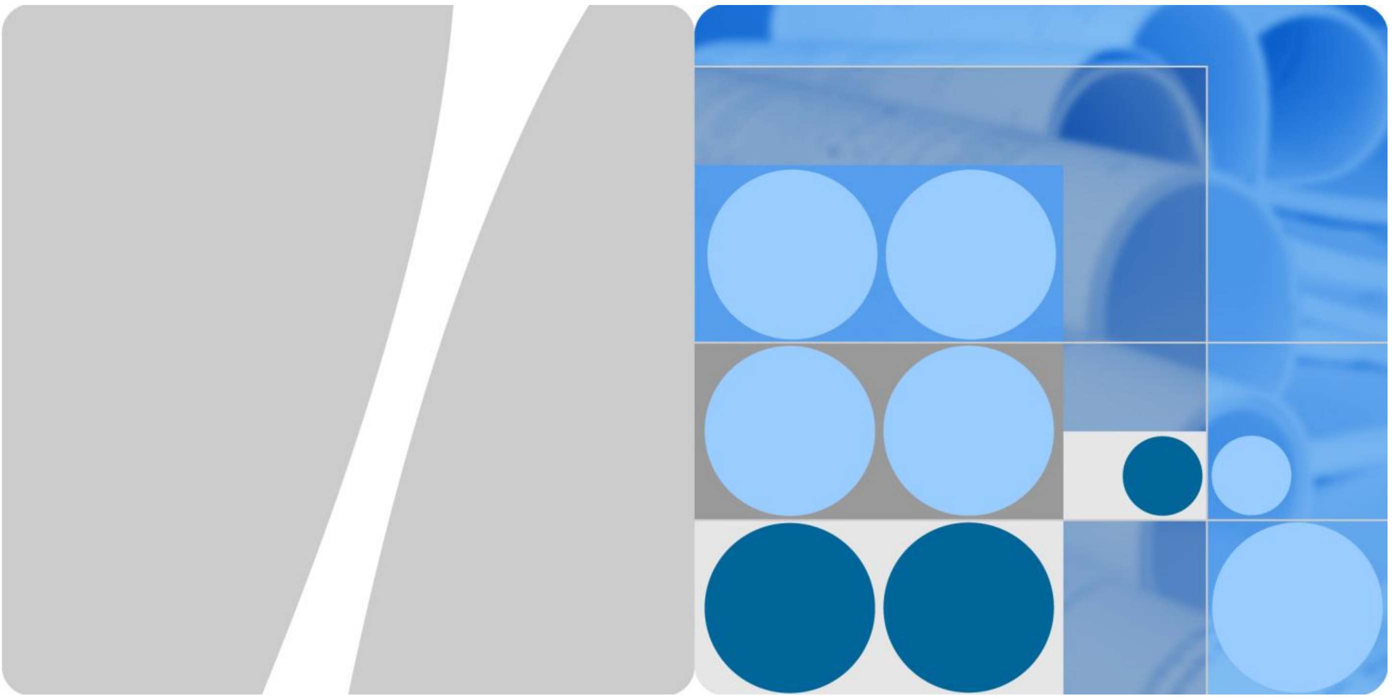


iStack Technology White Paper

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

HUAWEI 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

Contents

Copyright © Huawei Technologies Co., Ltd.

1 iStack

About This Chapter

1.1 iStack Overview

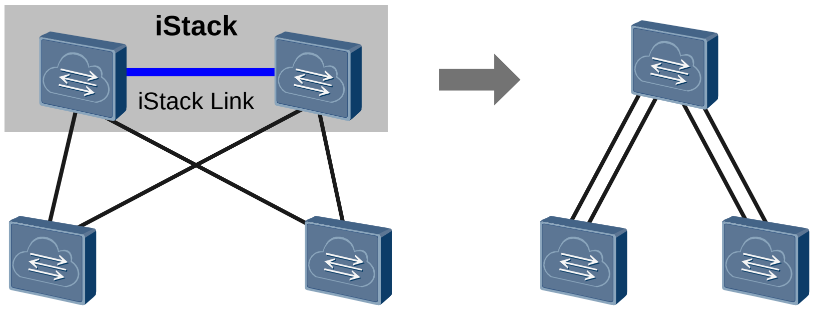
Currently, two types of communication devices are available on the network: box devices and chassis devices.

- Box devices are cost-effective, but do not support high availability and uninterrupted service protection. So box devices cannot be used at the core layer, at the aggregation layer, or in data centers. In complex networking environment, because box devices have low scalability, you have to maintain more network devices and change the original networking structure when adding new devices.
- Chassis devices have advantages such as high availability, high performance, and high port density. Therefore, chassis devices are often used at the core layer, at the aggregation layer, and in data centers. Compared with box devices, chassis devices have disadvantages such as a high initial investment and high single port cost.

The iStack technology combines the advantages of both box devices and chassis devices. The iStack technology virtualizes multiple devices supporting the stacking function into one logical device as shown in Figure 1-1. This virtual device has advantages like cost-effectiveness of box devices and high scalability and reliability of chassis devices.

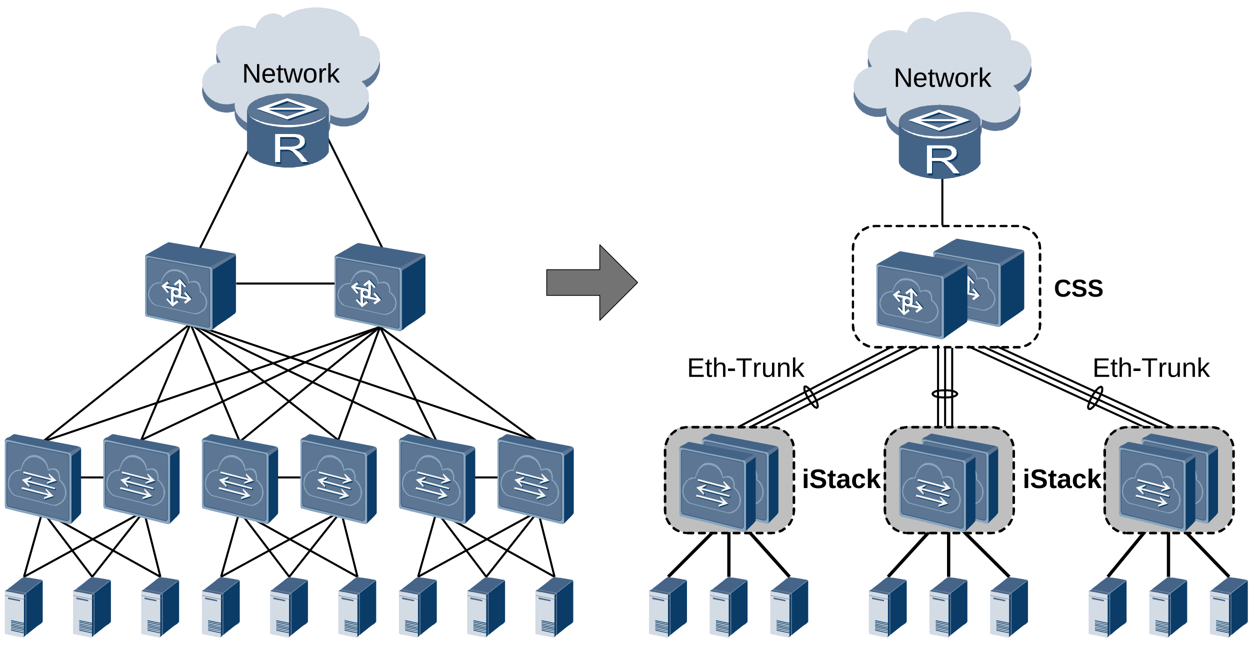
Copyright © Huawei Technologies Co., Ltd.

Figure 1-1 Setting up a stack



iStack is a virtualization technology that virtualizes multiple devices at the same layer into a logical device without changing the network physical topology structure. The iStack technology simplifies network structure and network protocol deployment, and improves network reliability and manageability as shown in Figure 1-2.

Figure 1-2 Network horizontal virtualization



1.2 Advantages of iStack Technology

Simplified Configuration and Management

After a stack is set up, multiple physical devices are virtualized into one logical device. You can log in to the stack through any member device to configure and manage all the member devices.

Copyright © Huawei Technologies Co., Ltd.

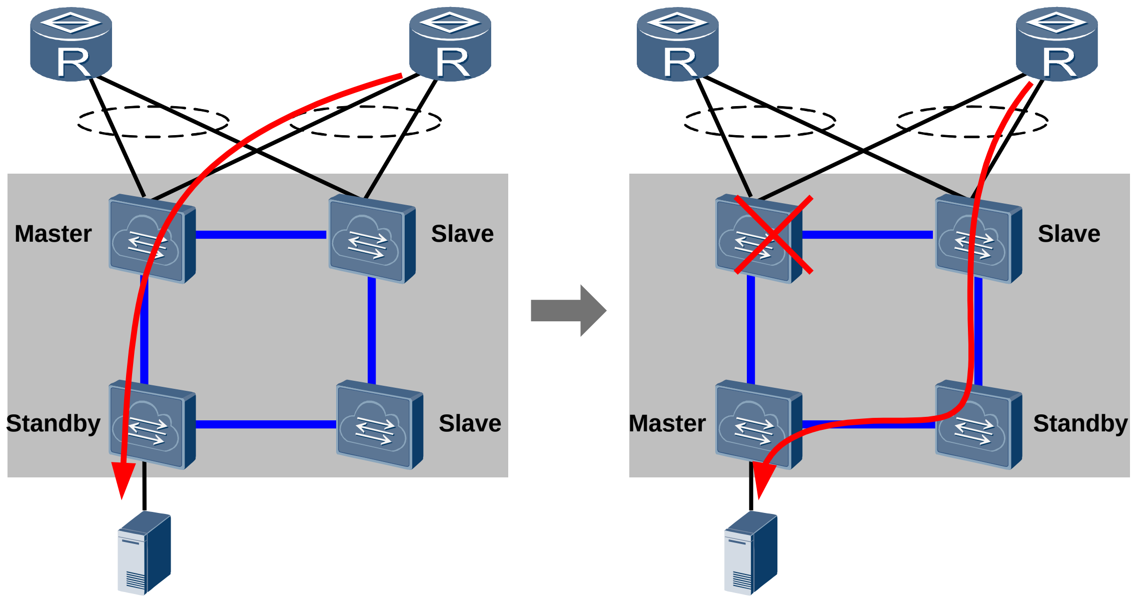
1:1 Redundancy of Control Planes

Chassis switches use the 1:1 redundancy mode. That is, each chassis switch is equipped with two Main Processing Units (MPUs). The master MPU processes services, and the standby MPU functions as a backup of the master MPU and synchronizes information with the master MPU. When the master MPU fails, the standby MPU becomes the new master MPU and starts to process services.

A box switch has only one control plane and cannot implement redundancy. When a box fails, the connected network is interrupted. iStack technology can implement 1:1 redundancy on box switches. In a stack, one master switch processes services, and one standby switch functions as a backup of the master switch and synchronizes information with the master switch. The other switches in the stack are slave switches. When the master switch fails, the standby switch becomes the new master switch, and a new standby switch is selected from the slave switches.

As shown in Figure 1-3, configuration and data on the standby switch is completed synchronized with the master switch. Therefore, when the standby switch becomes the new master switch, it can immediately replace the original master switch to manage other switches in the stack. Multiple slave switches in the stack further improve system reliability.

Figure 1-3 Standby switch becomes the new master switch



Uplink and Downlink Redundancy

iStack can implement redundancy of uplinks and downlinks through inter-device link aggregation. Traditional link aggregation technology combines multiple physical Ethernet interfaces (member interfaces) into one logical interface to provide backup when a link fails. However, this technology cannot provide backup when a device fails.

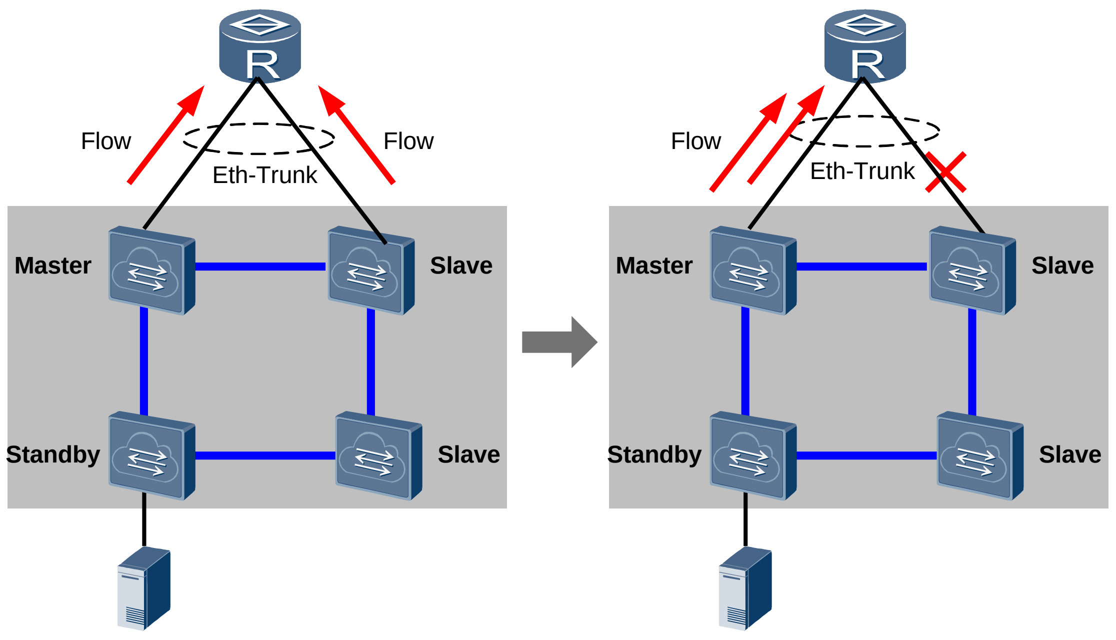
iStack supports inter-device link aggregation, which allows you to aggregate physical Ethernet interfaces on multiple member switches of a stack into one logical interface. When a device some member interfaces fails, the other member switches can manage and maintain the remaining member interfaces so that services are not interrupted. Inter-device link aggregation

Copyright © Huawei Technologies Co., Ltd.

prevents service interruption caused by single-point failures and greatly improves network availability.

As shown in Figure 1-4, traffic sent to core devices of the network is evenly distributed to multiple links in a link aggregation group. When a link fails, traffic on this link is evenly distributed to the other links. This link redundancy mechanism improves network reliability.

Figure 1-4 Data traffic forwarding after a link failure



Redundancy of Stack Links and Stack Interfaces

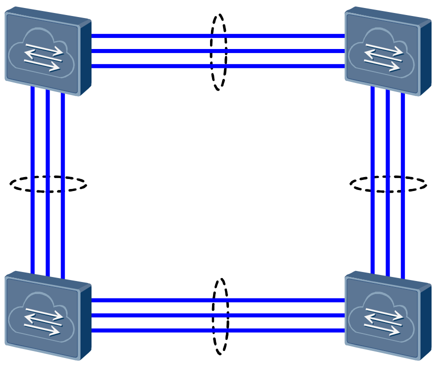
- Redundancy of stack links in a ring topology

  iStack can implement redundancy of stack links in a ring topology. When a link fails, the ring topology changes into a chain topology so that services in the stack are not affected.
- Redundancy of stack interfaces

  iStack uses link aggregation to implement redundancy of stack interfaces. Multiple physical links on stack interfaces can be aggregated to load balance traffic, improve bandwidth and system performance. Additionally, the physical links back up each other so that the failure of one link does not affect services in the stack. This improves device reliability.

Copyright © Huawei Technologies Co., Ltd.

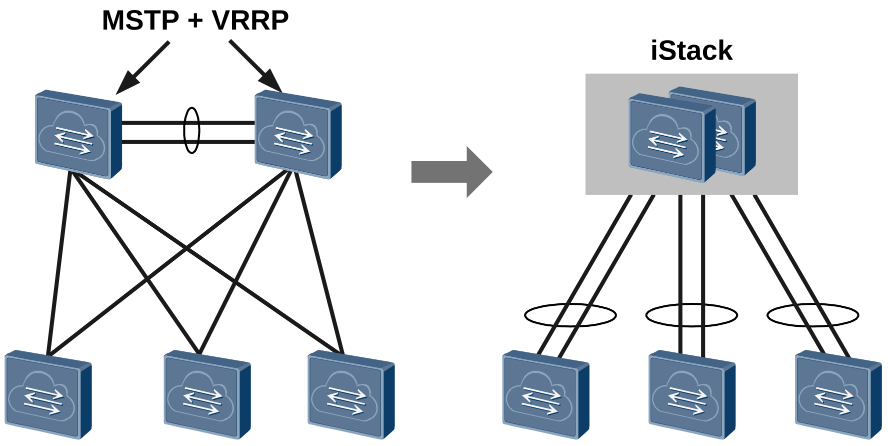
Figure 1-5 Redundancy of stack links and stack interfaces



Simplified Networking

As shown in Figure 1-6, multiple devices on the aggregation layer are virtualized into a logical device through iStack technology. This simplified network does not require the Multiple Spanning Tree Protocol (MSTP) or Virtual Router Redundancy Protocol (VRRP), so network configuration is much simpler. Inter-device link aggregation also speeds up network convergence and improves network reliability.

Figure 1-6 Simplified network



1.3 Principles

1.3.1 Concepts

- **Switch roles**

  Each switch in a stack is a member switch. Member switches are classified into the following roles:

  - Master switch

    The master switch manages the entire stack. A stack has only one master switch.

- Standby switch

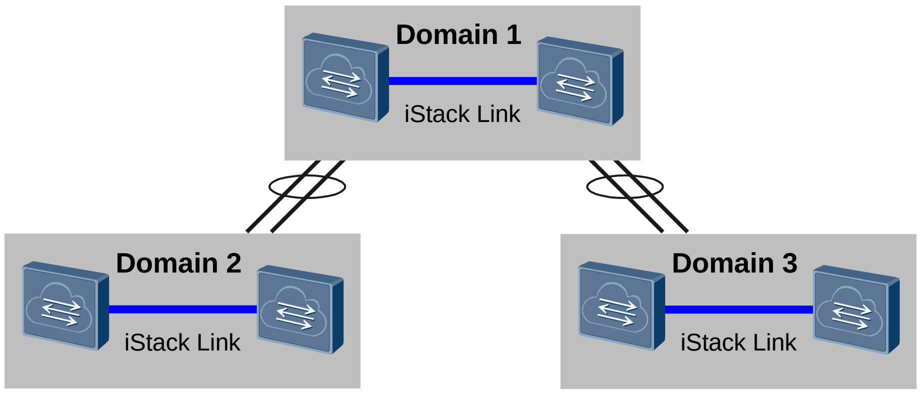
  The standby switch is the backup to the master switch. A stack has only one standby switch.
- Slave switch

  In a stack, all member switches except the master switch are slave switches. The standby switch is also a slave switch.

- **Stack domain**

  Switches that connect to each other using stack links to form a stack belong to a stack domain. To meet various networking requirements, you can configure multiple stacks on a network and use stack domain IDs to identify these stacks as shown in Figure 1-7.

Figure 1-7 Multiple stack domains



- **Stack ID**

  A stack ID, also called a member ID, is used to identify and manage member switches in a stack. All member switches in a stack 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 stack using physical member interfaces. Physical member interfaces forward 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. Each member switch has two stack interfaces, which are named Stack-Port*n*/1 and Stack-Port*n*/2. *n* specifies the stack ID of the member switch.

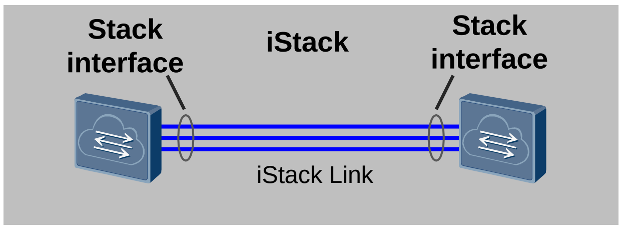
Copyright © Huawei Technologies Co., Ltd.

1.3.2 Principles

Setting Up a Stack

To use multiple switches to form a stack, connect physical member interfaces bound to the local stack interface to those bound to the neighbor stack interface, as shown in Figure 1-8. A stack interface can be bound to multiple physical member interfaces to improve stack link bandwidth and reliability.

Figure 1-8 Stack networking diagram



The stack contains multiple member switches, each of which has a role. During the setup of a stack, member switches exchange packets to elect the master switch that manages the stack, and the other switches become slave switches.

The rules for electing the master switch are as follows:

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.
4. 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.

The master switch collects member information, calculates the stack topology, and synchronizes the stack topology to all the other member switches.

- If slave switches have the same stack ID, the master switch assigns a unique stack ID to each of the member switches.
- When the master and slave switches use different software versions, slave switches synchronize the software version with the master switch, restart, and then join the stack.

The master switch elects a standby switch from slave switches as the standby switch. When the master switch fails, the standby switch takes over all services from the master switch.

The rules for electing the standby switch are as follows:

1. The switch with higher stack priority is preferred.
2. The switch with a smaller MAC address is preferred.

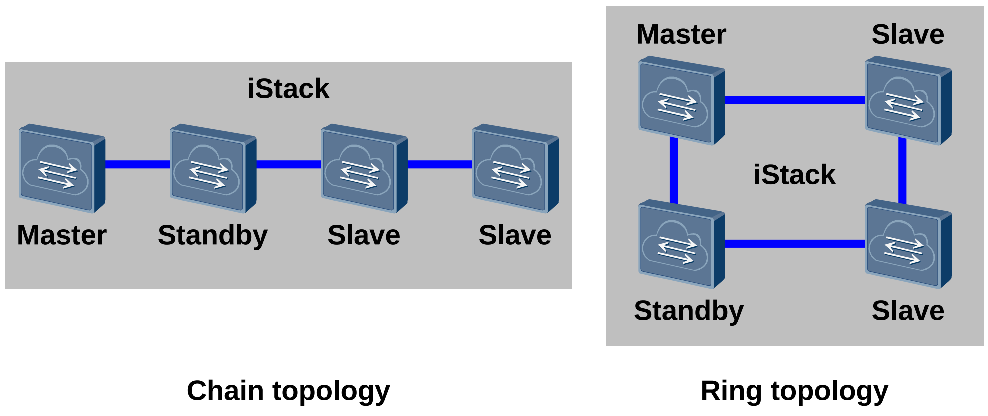
The election results are compared one by one. In the case of the same election result, the next rule is used until the master switch is elected.

Before a stack is set up, each switch is an independent entity and has its own IP address. You need to manage the switches separately. After the stack is set up, the switches in the stack 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 stack is the IP address and MAC address of the master switch when the stack is set up for the first time.

Connection Topology

A stack has two connection topologies: chain topology and ring topology, as shown in Figure 1-9.

Figure 1-9 Stack connection topology



A stack usually uses a ring topology. In a ring topology, iStack blocks a stack link according to the shortest path principle to prevent looping of data packets on stack links. A ring topology is more reliable than a chain topology. In a chain topology, a link failure will cause the stack to split. A ring topology will change into a chain topology when a link fails, so services in the stack are not affected.

Adding a Member Switch to a Stack

During stack maintenance, the master switch collects topology information. When a new switch joins a stack, the master switch is elected based on the following rules:

- If the new switch has not joined any stack, the switch is elected as a slave switch without changing the master and standby roles in the stack. For example, the new switch with the iStack configured is power off and connected to a stack with stack cables. After the new switch is powered on, it becomes a slave switch in the stack.
- If the new switch has already joined a stack (for example, the switch has the iStack function configured and connected to another stack using stack cables), the two stacks merge into a new stack. The master switch of the new stack is elected from the master switches of the two stacks. In the stack with the master switch elected as the new master switch, all member switches retain their roles, and services are not affected. In the other stack, all member switches join the new stack after restarting and synchronizing their configurations with the configuration of the new master switch, and services are interrupted.

Removing a Member Switch from a Stack

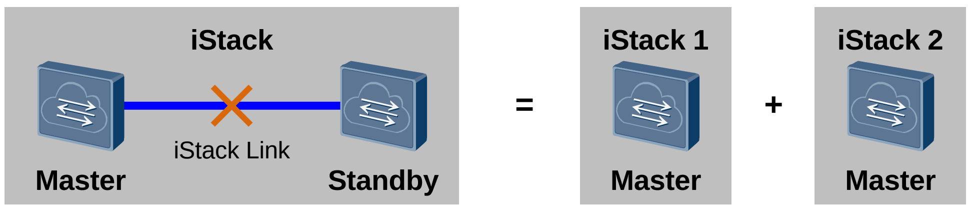
You can remove a member switch from a stack. The stack may be affected depending on the role of the member switch that leaves the stack.

- If the master switch leaves the stack, the standby switch becomes the new master switch, updates the stack topology, and specifies a new standby switch.
- If the standby switch leaves the stack, the master switch updates the stack topology and specifies a new standby switch.
- If a slave switch leaves the stack, the master switch updates the stack topology.
- If both the master and standby switches leave the stack, all slave switches restart and form another stack.

Stack Split

As shown in Figure 1-10, a stack splits into multiple stacks when some member switches are removed from the running stack with power on or when multiple nodes on the stack cable fail. A stack may split into multiple stacks with the same configurations, which causes conflicts of IP addresses and MAC addresses.

Figure 1-10 Stack split



Dual-Active Detection

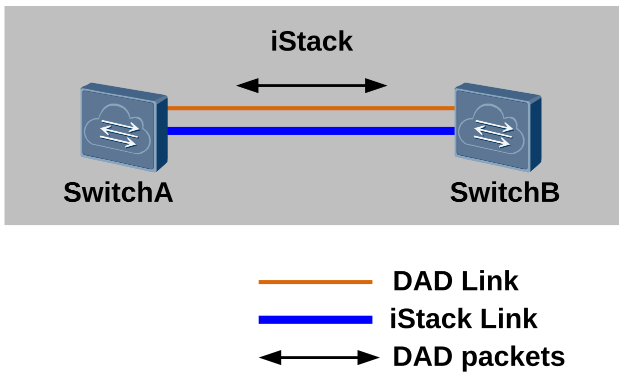
Dual-active detection (DAD) is a method to detect a dual-active scenario and take recovery action, ensuring network stability.

DAD has two modes:

- DAD in direct mode

  As shown in Figure 1-11, DAD is performed between member switches in a stack using a direct link.

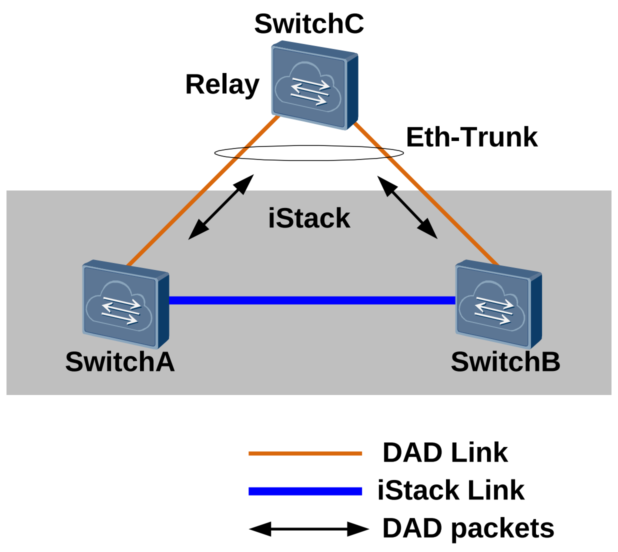
Figure 1-11 DAD in direct mode



- DAD in relay mode

As shown in Figure 1-12, DAD is configured on the inter-chassis Eth-Trunk in a stack, and DAD in relay mode is configured on the proxy device.

Figure 1-12 DAD in relay mode



After a DAD link is configured, stacks exchange DAD packets on the DAD link. After a stack splits into multiple stacks, the stacks compare information in received DAD packet with local information. If the switch in a stack is elected as the master switch, the switch remains Active and continues forwarding service packets. If the switch in a stack 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 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 Stack Upgrade

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

During fast stack 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.

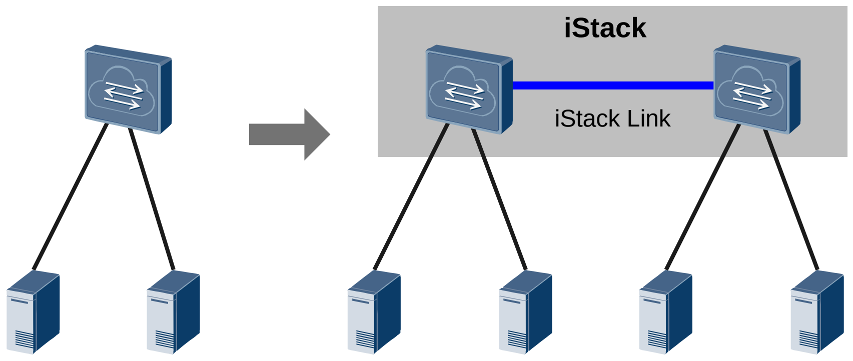
Copyright © Huawei Technologies Co., Ltd.

1.4 Applications

Increasing Ports

As shown in Figure 1-13, when the port density of a stack is insufficient for increasing number of servers, you can add new member switches to the stack to increase ports.

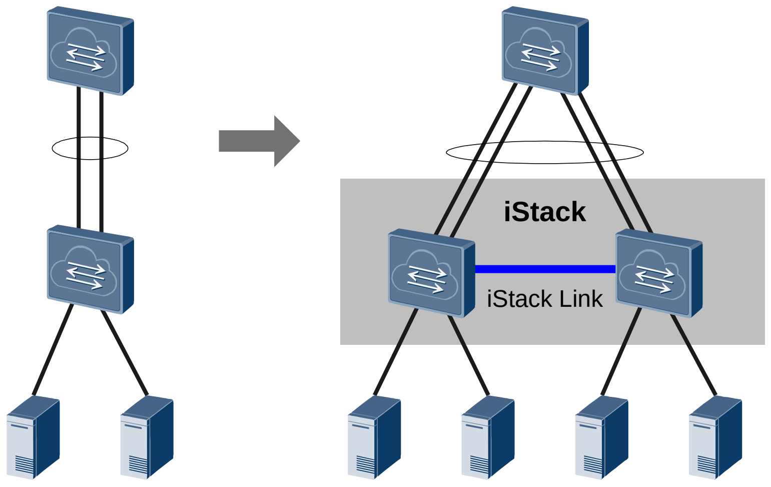
Figure 1-13 Increasing ports



Increasing Bandwidth

As shown in Figure 1-14, when the uplink bandwidth of a switch increases, you can enable this switch to work with another switch to form a stack, and configure multiple physical links of the two member switches into a link aggregation group to increase the uplink bandwidth of the switch.

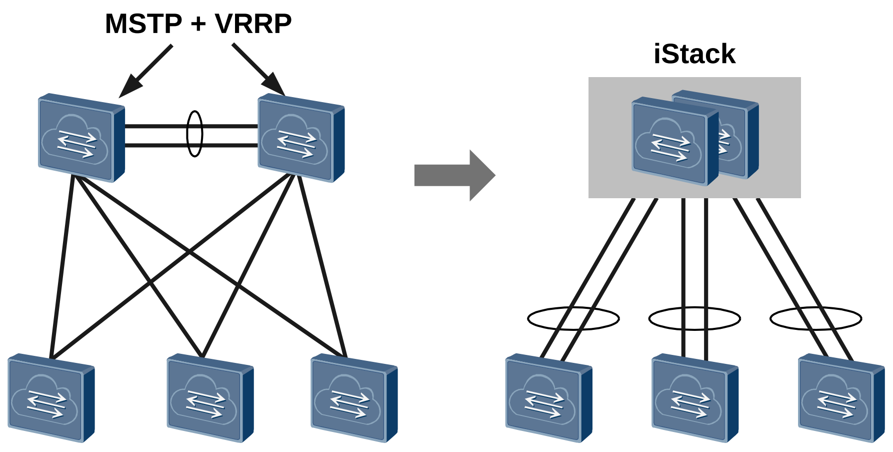
Figure 1-14 Increasing bandwidth



Simplifying Networking

As shown in Figure 1-15, multiple devices on the network form a stack 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 1-15 Simplifying networking



1.5 Example for Configuring the iStack Function

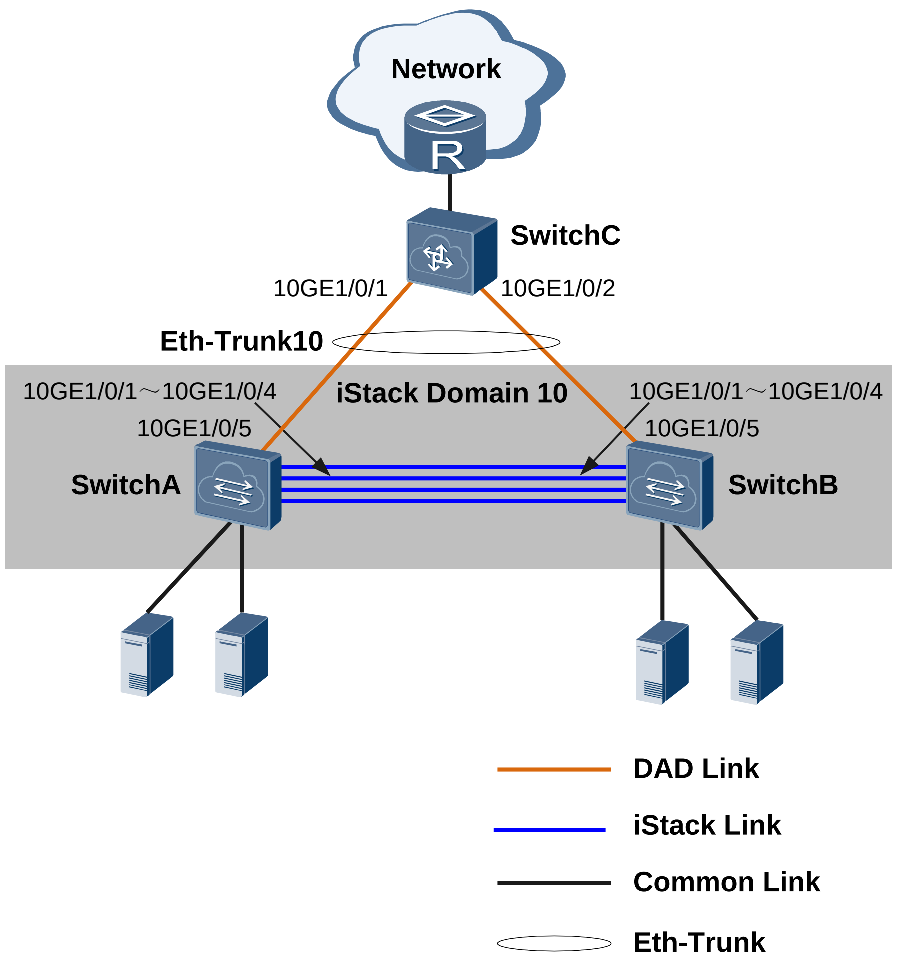
Networking Requirements

As the network size rapidly increases, the number of access interfaces provided by an access switch needs to be increased, and the network must be easy to manage and maintain. However, a single access switch cannot meet these requirements.

As shown in Figure 1-16, SwitchA and SwitchB form a stack, 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 1-16 Configuring a CSS



Procedure

Step 1 Configure stack domain ID 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] commit

<HUAWEI> system-view
[~HUAWEI] sysname SwitchB
[~HUAWEI] commit
[~SwitchB] stack
[~SwitchB-stack] stack domain 10
[~SwitchB-stack] stack renumber 2
[~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.

Copyright © Huawei Technologies Co., Ltd.

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

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

Step 3 Save the configurations of SwitchA and SwitchB, power off the two switches, connect the stack link, and power on the two switches.

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

```
<SwitchA> display stack
------------------------------------------------------------
MemberID Role    Mac            Priority    Device Type
------------------------------------------------------------
1       Master   0004-9f31-d520   100        CE6850-48T4Q-EI
2       StandBy  0004-9f62-1f40   100        CE6850-48T4Q-EI
------------------------------------------------------------
```

Step 5 Configure DAD in relay mode.

```
<SwitchA> system-view
[~SwitchA] interface eth-trunk 10
[~SwitchA-Eth-Trunk10] trunkport 10ge 1/0/5
[~SwitchA-Eth-Trunk10] trunkport 10ge 2/0/5
[~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 stack

```
#
sysname SwitchA
#
interface 10GE1/0/1
 port mode stack
 stack-port 1/1
#
```

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

Terms, Acronyms, and Abbreviations	Full Name	Description
iStack	Intelligent Stack	Stack
Eth-Trunk		A technology of binding multiple physical interfaces into a logical interface to increase bandwidth, also called link aggregation.

Copyright © Huawei Technologies Co., Ltd.