Hewlett Packard Enterprise

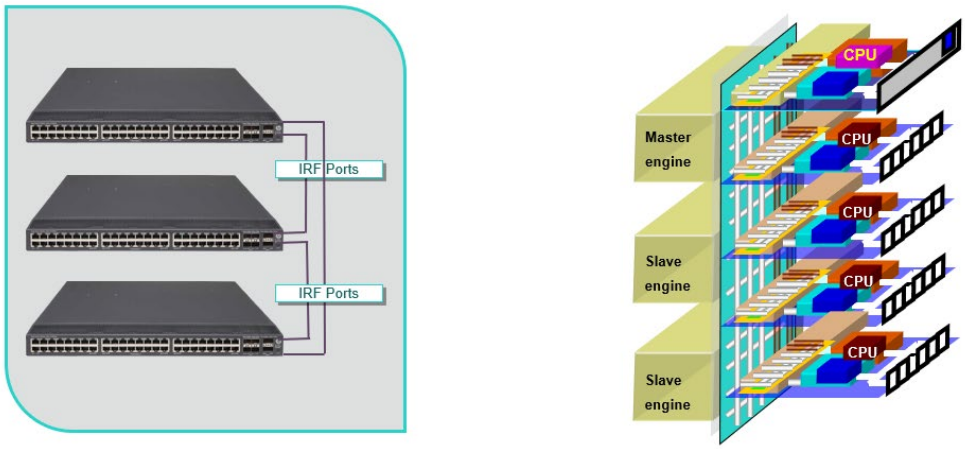
HPE IRF Technology

Technical Whitepaper
November, 2022

1 Overview

Intelligent Resilient Framework (IRF) is a software virtualization technology. It connects multiple network devices through physical ports and performs necessary configurations. These devices are virtualized into a distributed device. IRF technology extends network control over multiple active switches. Management of a group of IRF enabled switches is consolidated around a single management IP address, which vastly simplifies network configuration and operations. The user can combine as many as nine HPE switches to create an ultra-resilient virtual switching fabric comprising hundreds or even thousands of 1GbE or 10GbE switch ports. In different terminologies, we mention IRF as IRF fabric or IRF stack or IRF stacking system which we will be using in this document.

Figure 1 IRF Physical & Logical view for three switches



1.1 Benefits of IRF

IRF Fabric features the following advantages:

Simplified network management and operations

- Consolidates management of multiple, discrete devices into a single, easy-to-manage, virtual switch fabric.
- The user can combine multiple distribution layer switches into one IRF stack. All of the switches have the same routing table and can route packets received from the edge switches. The IRF master will run the routing protocol for the entire virtual device.so IRF Reduces network traffic as network protocols running in IRF stack operate on the logical device.

Higher performance networks

- IRF provides a simple, cost-effective solution to the issues that arise when use population exceeds the available network ports. With IRF deployed, one can add new members to the virtual IRF device, adding port density with minimal configuration of the new switches.
- When the forwarding capability of the core switch cannot satisfy users' needs, one can add a switch to form a Stacking system with the original core switch. If the forwarding capability of one switch is 64 Mbps, the forwarding capability of the whole IRF fabric is 128 Mbps after another switch is added. Note that this increases the forwarding capability of the entire IRF fabric, not a single switch.

Resilient networks with 2-tier architectures

- Flatten legacy 3-tier architectures delivering resilient new data centers that are STP free with faster reconvergence time ensuring interruption free services.
- IRF reduces the requirements of complex technologies STP,VRRP as IRF fabric is considered a single entity.

1.2 Hardware Support

As of now, all products of the HPE FlexFabric & FlexNetwork series support IRF.

2 Technology Implementation

This chapter discusses the implementation of IRF on Comware switches and the technical workflow of IRF.

2.1 Basic Concepts

In an IRF fabric, every single device is a IRF fabric member, and plays one of the following two roles, according to its function:

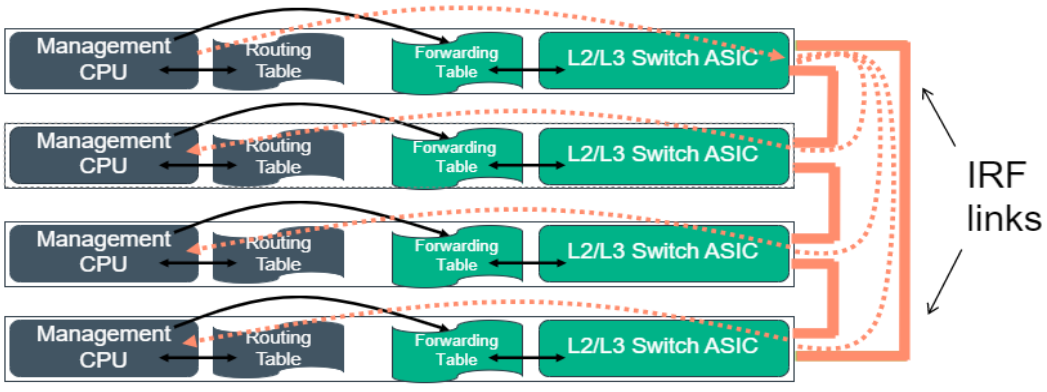
- **Master**: A member device. It is elected to manage the entire IRF fabric. An IRF fabric has only one master.
- **Slave**: A member device. It is managed by the master and operates as a backup of the master. In an IRF fabric, except for the master, all the other devices are slaves.

2.1.1 Software Architecture

. The IRF virtualization module automatically collects IRF fabric topology, performs role election, and virtualizes the all-member switches into a one single virtual device or IRF fabric.

- **Device management**: The management layer of the device and manages the device resources like boards and cards. The word device here includes both the hardware and the logical device.
- **System management and application modules**: This module indicates all management and control programs running on the device, for example, various routing protocol modules, and link layer protocol modules.
- **IRF virtualization modules**: This module can simulate a virtual device, and the device management module manages both the IRF logical device and physical devices and masks their differences. For various kinds of application software running on this system, after the mask of the management layer, they do not care about the physical differences, that is, no matter it is a single physical device or the logical device, they do not need to modify its internal mechanism or interfaces. In the figure below we represent IRF fabric 4 physical switches.

Figure 2 IRF Fabric system logical representation



2.1.2 Physical Connections

To make an IRF fabric operate normally, you need to connect the IRF fabric members physically.

IRF port

An IRF port is a logical interface that connects IRF member devices. Every IRF-capable device has two IRF ports. The IRF ports are named IRF-port n/1 and IRF-port n/2, where n is the member ID of the device. The two IRF ports are referred to as IRF-port 1 and IRF-port 2. A logical IRF port is a logical port dedicated to the internal connection of an IRF virtual device. These ports cannot act as access, trunk or hybrid ports. An IRF port is effective only when it is bound to a physical IRF port.

To use an IRF port, you must bind a minimum of one physical interface to it. The physical interfaces assigned to an IRF port automatically form an aggregate IRF link. An IRF port goes down when all its IRF physical interfaces are down.

IRF physical interface

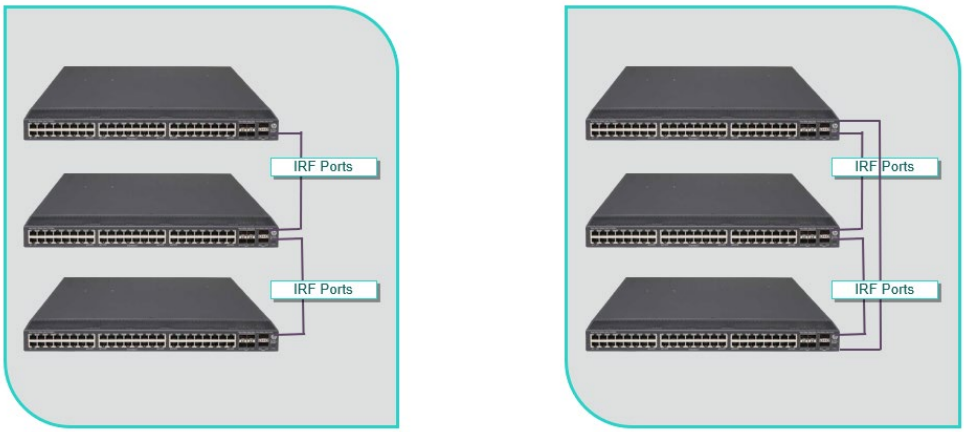
IRF physical interfaces connect IRF member devices and must be bound to an IRF port. They forward traffic between member devices, including IRF protocol packets and data packets that must travel across IRF member devices.

You can connect IRF ports with either dedicated cables or fibers. Dedicated cables provide higher reliability and performance; whereas fibers connect physical devices located very far from each other and provide flexible application.

An IRF Stack typically has a Daisy-chain connection or a ring connection:

- Daisy-chain connection: The port of the first device below is connected to the top port of the second device, and the down port of the second device is connected to the port of the third device.
- Ring connection: The top port of the first device is connected with down port of third device while down port of first and second device is connected to second and third device top port respectively. The failure of one link in a ring connection does not affect the function and performance of the IRF Stack, whereas the failure of one link in a daisy-chain connection causes the split of the IRF Stack.

Figure 3 Physical Connections of an IRF Stack



2.1.3 Topology Collection

Each device in an IRF exchanges hello packets with the directly connected neighbors to collect topology of the entire IRF Stack. The hello packets carry topology information, including IRF port connection states, member IDs, priorities, and bridge MAC addresses.

Each member records its known topology information locally. At the initiation of the collection, the members record their own topology information. When an IRF port of a member becomes up, the member sends its known topology information from this port periodically. Upon receiving the topology information, the directly connected neighbor updates the local topology information.

The topology collection process lasts for a period. When all members have obtained the complete topology information (known as topology convergence), the Stack will enter the next stage: role election.

General recommendations for topology set-up:

- Do not use IRF to interconnect DCs (4 or more devices in an IRF Stack).
- Try to keep Stack s small (4x2 instead of 1x8).
- When using more than 2 devices each connected device should connect to each member (fabric-style).
- Do not overbook IRF links.
- Try to keep traffic from ISL (for example, LACP vs. ARP load-balancing on VMs).
- All IRF member devices must run the same software image version. Make sure the software. auto-update feature is enabled on all member devices.
- For high availability, bind multiple physical interfaces to an IRF port. You can bind a maximum of eight physical interfaces to an IRF port.

2.1.4 Role Election

A Stack is composed of multiple member devices; each member has a role, which is either master or slave. The process of defining the role of Stack members is role election.

Master election occurs each time the IRF fabric topology changes in the following situations:

- The IRF fabric is established.
- The master device fails or is removed.
- The IRF fabric splits.
- Independent IRF fabrics merge.
- Master election does not occur when split IRF fabrics merge.

Master election selects a master in descending order:

1. Current master, even if a new member has higher priority. When an IRF fabric is being formed, all members consider themselves as the master. This rule is skipped.
2. Member with higher priority.
3. Member with the longest system uptime.
   Two members are considered to start up at the same time if the difference between their startup times is equal to or less than 10 minutes. For these members, the next tiebreaker applies.
4. Member with the lowest CPU MAC address.

In this stage, member ID collision, software version loading, and Stack merging are also handled by the master, which are discussed in the later sections.

When a device is booted, it first collects topology information and then participates in the role election. After that, the Stack system can run normally. When the role election is finished, the Stack enters the next stage: Stack maintenance.

For a Stack formed by box-type devices, the logical device is equal to a chassis-type distributed device. The master in this Stack equals the AMB of the logical device, and the slaves equal the SMBs (also the interface boards) of the logical device, as shown in Figure 4 .

For a Stack formed by chassis-type distributed devices, the logical device is also equal to a chassis-type distributed device. The AMB of the master in this Stack equals the AMB of the logical device, and the SMB of the master, the AMB and SMB of the slaves equal the SMBs of the logical device, as shown in Figure 5 .

Figure 4 A Stack formed by box-type devices

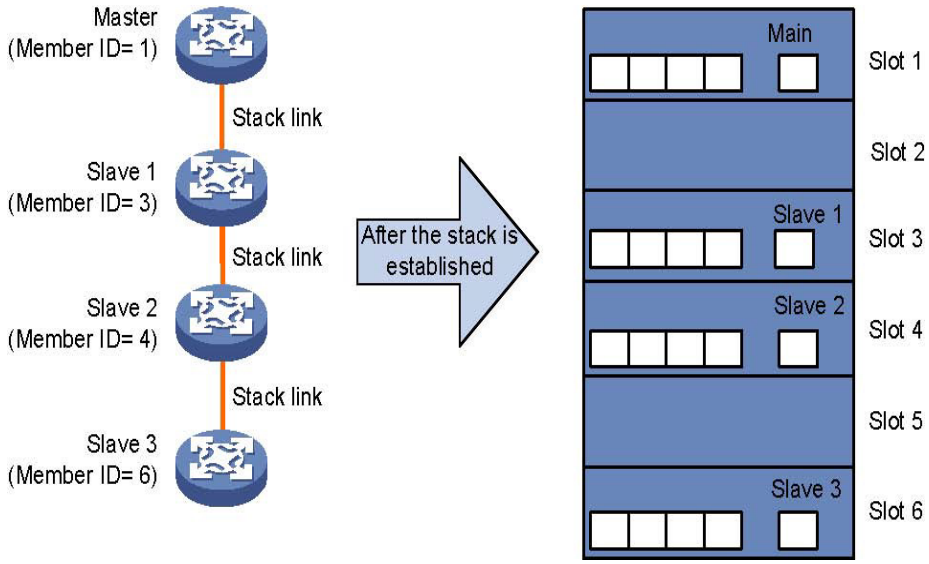
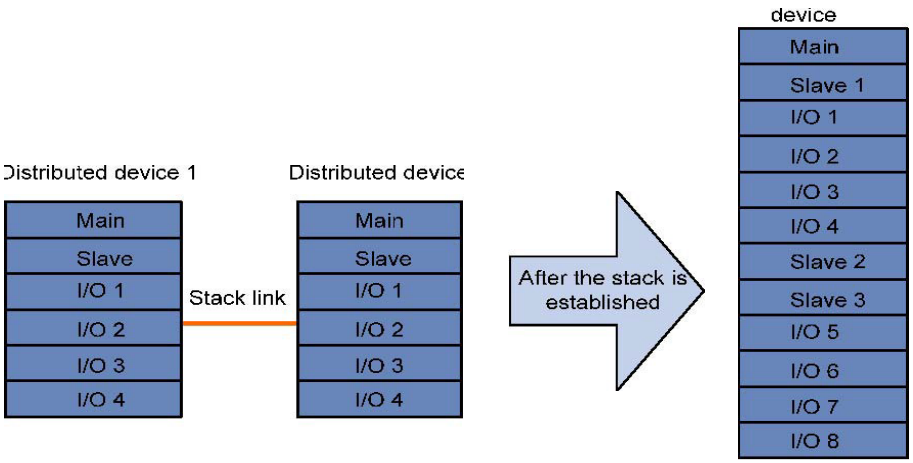


Figure 5 A Stack formed by chassis-type distributed devices



2.2 IRF Stack Management

After an IRF Stack is formed, the logical device can be considered as a single entity, and you can manage and configure the IRF Stack system by telnetting or logging in to a member device through its console port.

As the management center of an IRF Stack system, the master is responsible for replying to the users' login requests, that is, no matter how or through which member to log in to the IRF, the users perform configuration to the master, and the master delivers the configuration to each slave. In this way, the configurations on each device within an IRF Stack will be consistent.

A Stack system uses member IDs to uniquely identify and manage member devices.
For configuration continence and easy identification, member IDs are used in the port IDs, namely, the number of the first dimension of a port ID is the member ID of the device to which the port belongs.

Member IDs are also used in file system management, for example, path slot6#flash:/test.cfg indicates that a file named test.cfg is under the root directory of the flash that is on the member device with member ID of 6.

2.3 IRF Stack Maintenance

The main functions of Stack maintenance are monitoring the joining and leaving of the member devices, collecting new topology information, and maintaining the current topology.

2.3.1 Joining of a Member Device

During the Stack maintenance, topology collection is performed. When a new member joins the Stack, the Stacking system may take one of the following approaches:

The newly added device was not in any other Stack before it joined this Stack. For example, you configure Stack functions on the device, power it off, connect it to the Stack using Stack cables, and power it on. In this situation, the new member will be elected as a slave.

The newly added device was in another Stack before it joined this Stack. For example, you configure Stack functions on the device, and connect it to another Stack. After that, if you want to add the device to this Stack, it is a Stack merge, that is, a process of connecting two existing Stack s together, which is not recommended. During the mergence, Stack election is held, and members of the loser side reboot and join the winner side as slaves.

If a member joins a Stack successfully, it is equal to adding an SMB and interfaces on this board for the Stacking system.

The probable reasons for a member to join a Stack can be:

- The member device is manually added into the Stack.
- The member device is recovered from a system failure or link failure, and it will join the Stack again automatically.

2.3.2 Leaving of a Member Device

During the Stack maintenance, a member device is considered to have left a Stack in one of the following two situations:

- In an IRF Stack, direct neighbors exchange hello packets periodically (the period is 200 ms). Without receiving any hello packet from a direct neighbor for ten periods, a member considers that the hello packets timed out, and the Stack isolates the expired device in the topology.
- When an IRF port of a member becomes down, the member broadcasts the information to all the other members immediately. The members recalculate the current topology before the hello packets time out.

If a slave leaves the Stack, it is equal to losing an SMB and interfaces on this board for the Stacking system; if the master leaves the Stack, a role election will be held, and the elected new master will take over all functions of the original master.
If a single device leaves a Stack, it operates independently; if multiple connected devices leave a Stack, they form another Stack, and the process is Stack split.

The following are the probable reasons for a member to leave a Stack:

- The member device is moved, and the topology is changed manually.
- The member device fails.
- Link failure.

2.3.3 Topology Update

Topology change indicates that the topology changes from a Daisy-chain connection to a ring connection, or vice versa. For example, a ring connection may become a Daisy-chain connection when the link fails; or, when adding new devices to a Stack, you need to first change the ring connection to the Daisy-chain connection, and then connect the new devices. For this kind of simple topology update, the member construction is not changed and only the forwarding path may be automatically changed when necessary, so the device functions properly without being affected.

2.4 Auto Upgrade of Software

The IRF provides the auto loading function. When a new member is added to an Stack, it is not required to have the same software version as that of the logical device, but a compatible version is required. As soon as a device is added into a Stack, the system compares its software version with that of the master. If the versions are not consistent, the device downloads the boot file from the master automatically, reboots with the new boot file, and joins the Stack again. If the device does not support this function, you need to update the software version manually to make the software versions of both the new member and the logical device consistent and then add the new member to the Stack.

3 High Reliability

IRF Stack devices are usually deployed at the access layer, distribution layer and data center, therefore having a high reliability requirement. To shorten the down time of IRF stack devices resulting from daily maintenance and system crash, and to improve the reliability of the Stacking system and application, the IRF adopts a series of backup technologies:

- 1: N backup
- Protocol hot backup
- Up/down link backup
- IRF port backup

3.1 1: N Backup

Common chassis-type distributed devices adopt 1:1 backup, that is, a chassis-type distributed device is installed with two main boards. The active main board (AMB) processes services, and the standby main board (SMB) operates as the backup to keep synchronization with the AMB. When the AMB fails, the SMB takes the responsibility of the AMB immediately.

The IRF adopts 1: N backup, that is, the master processes services, and the slaves operate as the backups of the master. When the master fails, the IRF selects a slave as the master. During the running of the stacking system, strict configuration and data synchronization is performed. Therefore, the new master can take the responsibility of the original master to manage and run the stacking system, without affecting the original network functions and services. Meanwhile, the existence of multiple slaves can improve the reliability of the system.

For a Stack consisting of chassis-type distributed devices, the IRF manages the AMB and SMB of each member device, increasing the reliability of the system.

3.2 Protocol Hot Backup

In a 1: N backup environment, protocol hot backup enables you to back up the configuration data of a protocol and the data supporting the running of the protocol (for example, state machine or session table entries) to all the other member devices. In this way, the stacking system can operate as an independent device to run in the network.

Take routing protocols as an example. As shown in Figure 6, RIP runs in the network on the left of the stacking devices, and OSPF runs in the network on the right. When the master receives the update packets sent from a neighboring router, it updates its routing table, and sends the updated routing table entries and protocol state information to all the other member devices. Upon receiving the entries and protocol state information, the member devices update their local routing tables and protocol states, thus, to ensure consistency of the routing-related information of each physical device in the stacking system. When the slaves receive the update packets sent from their neighbors, they send the packets to the master for processing.

As shown in Figure 7, when the master fails, the newly elected master can take the responsibility of the old master seamlessly. Upon receiving the OSPF packets sent from a neighboring router, the master sends the updated routing table entries and protocol information to all the member devices, without affecting the running of the OSPF protocol. In this way, when a member device fails, the other member devices can operate normally and can quickly take the responsibility of the failed member device. Meanwhile, the intra-domain routing protocol will not be interrupted, and Layer 2 and Layer 3 forwarding traffic and services will not be interrupted, either, thus implementing fault protection and device switching without service interruption.

Figure 6 Protocol Hot Backup (before a member device failure)

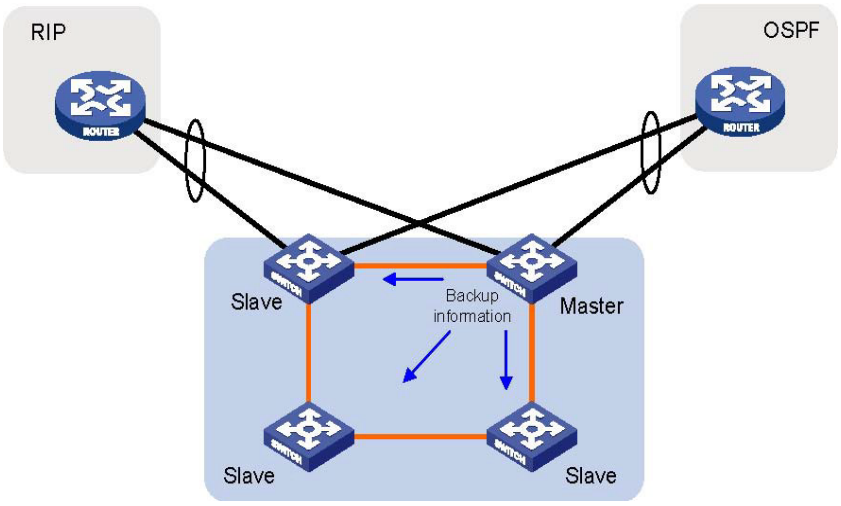
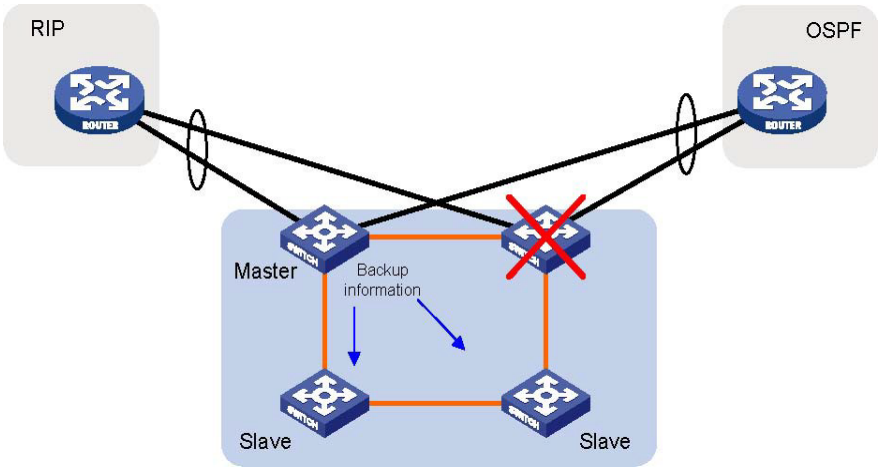


Figure 7 Protocol Hot Backup (after a member device failure)



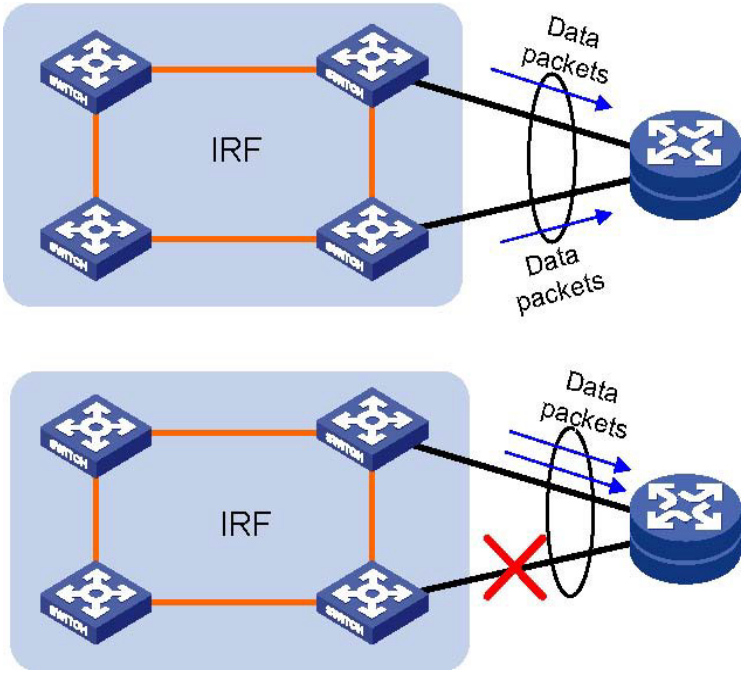
3.3 Uplink/Downlink Backup

IRF uses a distributed aggregation technology to implement uplink/downlink backup. The traditional aggregation technology enables you to aggregate multiple physical Ethernet ports (known as member ports) to implement link backup. However, it does not have a backup mechanism for a single-point failure. The new distributed aggregation technology supported by IRF enables you to add the physical Ethernet ports on different devices to an aggregation port group. In this way, even if the device where some ports reside fails, the aggregation link will not become invalid. Other member devices that work normally will manage and maintain the other aggregation ports. This is of great importance to the network environments with core switching systems and having high-quality service requirements; it not only solves the problem of single-point failure of aggregation devices, but also

increases availability of the entire network.

As shown in Figure 8 , the traffic that goes to the core network is distributed evenly on the aggregation links. When an aggregation link fails, the distributed link aggregation technology can automatically distribute the traffic to other aggregation links to implement link backup and increase network reliability.

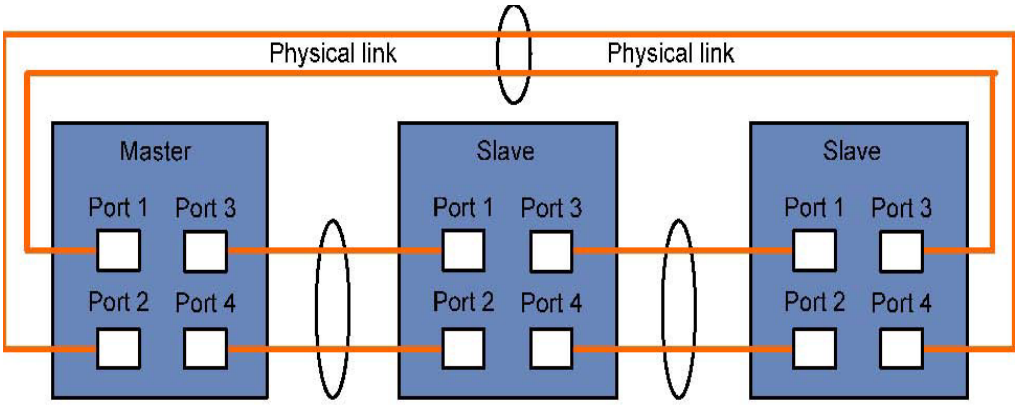
Figure 8 Uplink/Downlink Backup



3.4 IRF Port Backup

IRF uses aggregation technology to implement IRF port backup. As shown in Figure 9 , multiple physical links can be aggregated to share load, which effectively increases bandwidth and performance; in addition, the physical links back up one another, ensuring that even if one link fails, the stacking function is not affected, thus increasing the device reliability.

Figure 9 IRF port backup



For a chassis-type distributed stacking device, the aggregated physical ports can be on the same interface board, or on different interface boards, that is, inter-board aggregation of Stack ports is supported. Even if one interface board fails, the stacking functions will not be affected.

4 MAD Mechanisms

The multi-active handling procedure includes detection, collision handling, and failure recovery. IRF provides MAD mechanisms by extending LACP, BFD, ARP, and IPv6 ND protocols. You can configure a minimum of one MAD mechanism on an IRF fabric for prompt IRF split detection.

- Do not configure LACP MAD together with ARP MAD or ND MAD, because they handle collisions differently.
- Do not configure BFD MAD together with ARP MAD or ND MAD. BFD MAD is mutually exclusive with the spanning tree feature, but ARP MAD and ND MAD require the spanning tree feature. At the same time, BFD MAD handles collisions differently than ARP MAD and ND MAD. The table below compares the MAD mechanisms and their application scenarios.

Table 1 Comparison of MAD mechanisms

MAD Mechanism	Advantages	Disadvantages	Application Scenarios
LACP MAD	The detection speed is fast. Runs on existing aggregate links without requiring MAD-dedicated physical links or Layer 3 interfaces.	Requires an intermediate device that supports extended LACP for MAD.	Link aggregation is used between the IRF fabric and its upstream or downstream device. LACP MAD is general recommendation by HPE.
BFD MAD	The detection speed is fast. Intermediate device, if used, can come from any vendor. BFD MAD offers an IP address that can be used to hop onto the neighbour switch in case of a split. This makes a lot of sense when there's no OoB	Requires MAD dedicated physical links and Layer 3 interfaces, which cannot be used for transmitting user traffic.	No special requirements for network scenarios. If no intermediate device is used, this mechanism is only suitable for IRF fabrics that have only two members that are geographically close to one another.
ARP MAD	No intermediate device is required. Intermediate device, if used, can come from any vendor. Does not require MAD dedicated ports.	Detection speed is slower than BFD MAD and LACP MAD. The spanning tree feature must be enabled if common Ethernet ports are used for ARP MAD links.	Non-link aggregation IPv4 network scenarios. Spanning tree-enabled non-link aggregation IPv4 network scenarios if common Ethernet ports are used.
ND MAD	No intermediate device is required. Intermediate device, if used, can come from any vendor. Does not require MAD dedicated ports.	Detection speed is slower than BFD MAD and LACP MAD. The spanning tree feature must be enabled if	Non-link aggregation IPv6 network scenarios. Spanning tree-enabled non-link aggregation IPv6 network scenarios if common Ethernet ports

		common Ethernet ports are used for ND MAD links.	are used.

4.1 LACP MAD

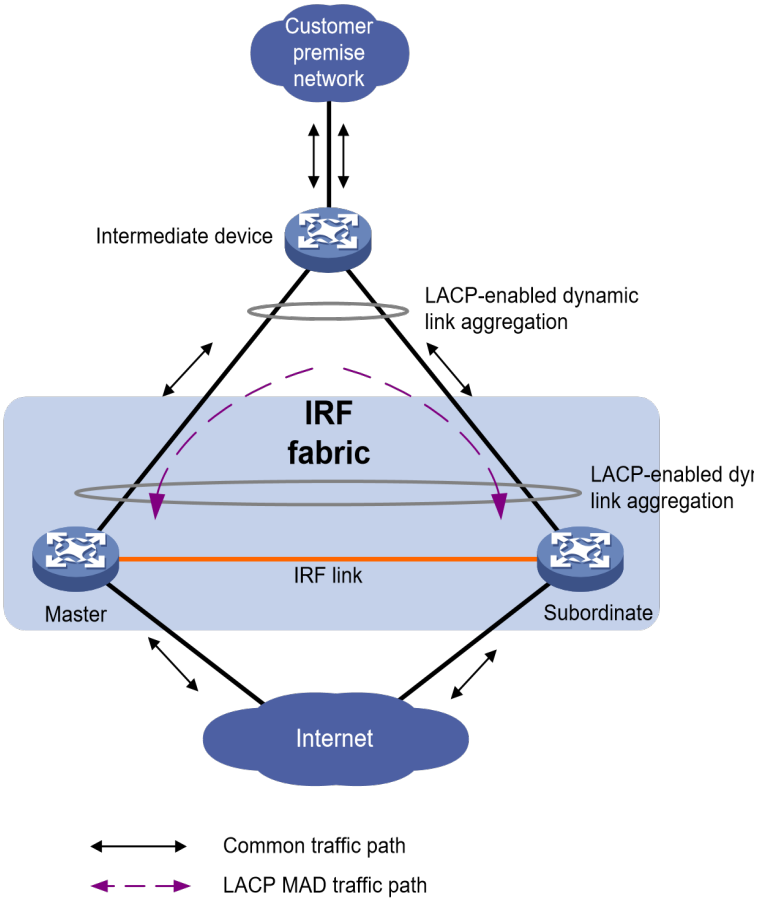
As shown in Figure 10, LACP MAD has the following requirements:

- Every IRF member must have a link with an intermediate device.
- All the links form a dynamic link aggregation group.
- The intermediate device must be a device that supports extended LACP for MAD.

The IRF member devices send extended LACPDUs that convey a domain ID and an active ID (the member ID of the master). The intermediate device transparently forwards the extended LACPDUs received from one member device to all the other member devices.

- If the domain IDs and active IDs sent by all the member devices are the same, the IRF fabric is integrated.
- If the extended LACPDUs convey the same domain ID but different active IDs, a split has occurred. LACP MAD handles this situation as described in "Collision handling."

Figure 10 LACP MAD Mechanism



4.2 BFD MAD

BFD MAD detects multi-active collisions by using BFD.

You can use common or management Ethernet ports for BFD MAD.
If management Ethernet ports are used, BFD MAD has the following requirements:

- An intermediate device is required and each IRF member device must have a BFD MAD link to the intermediate device.
- Each member device is assigned a MAD IP address on the master's management Ethernet port.

If common Ethernet ports are used, BFD MAD has the following requirements:

- If an intermediate device is used, each member device must have a BFD MAD link to the intermediate device.
- If no intermediate device is used, all member devices must have a BFD MAD link to each other.

Ports on BFD MAD links are assigned to the same VLAN or Layer 3 aggregate interface. Each member device is assigned a MAD IP address on the VLAN interface or Layer 3 aggregate interface.

When you use BFD MAD, follow these restrictions and guidelines:

- As a best practice, use an intermediate device to connect IRF member devices if the IRF fabric has more than two member devices. A full mesh of IRF members might cause broadcast loops.
- As a best practice to avoid member device failure from affecting BFD MAD, preferentially use management Ethernet ports for BFD MAD.
- The BFD MAD links, and BFD MAD VLAN (or Layer 3 aggregate interface) must be dedicated. Do not use the BFD MAD links or BFD MAD VLAN (or Layer 3 aggregate interface) for any other purposes.

NOTE:

- The MAD addresses identify the member devices and must belong to the same subnet.
- Of all management Ethernet ports on an IRF fabric, only the master's management Ethernet port is accessible.

Figure 11 shows a typical BFD MAD scenario that uses an intermediate device. On the intermediate device, assign the ports on the BFD MAD links to the same VLAN.

Figure 12 shows a typical BFD MAD scenario that does not use an intermediate device.
With BFD MAD, the master attempts to establish BFD sessions with other member devices by using its MAD IP address as the source IP address.

- If the IRF fabric is integrated, only the MAD IP address of the master takes effect. The master cannot establish a BFD session with any other member. If you execute the **display bfd session** command, the state of the BFD sessions is **Down**.
- When the IRF fabric splits, the IP addresses of the masters in the split IRF fabrics take effect. The masters can establish a BFD session. If you execute the **display bfd session** command, the state of the BFD session between the two devices is **Up**.

Figure 11 BFD MAD scenario with an intermediate device

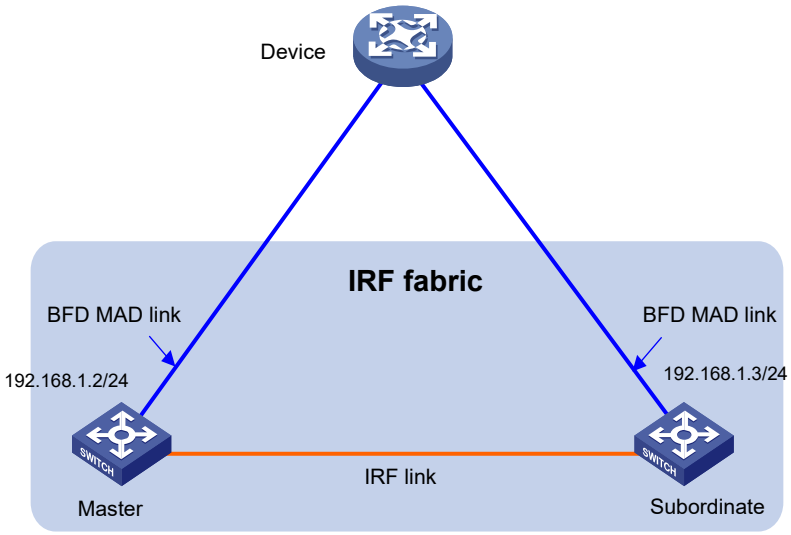
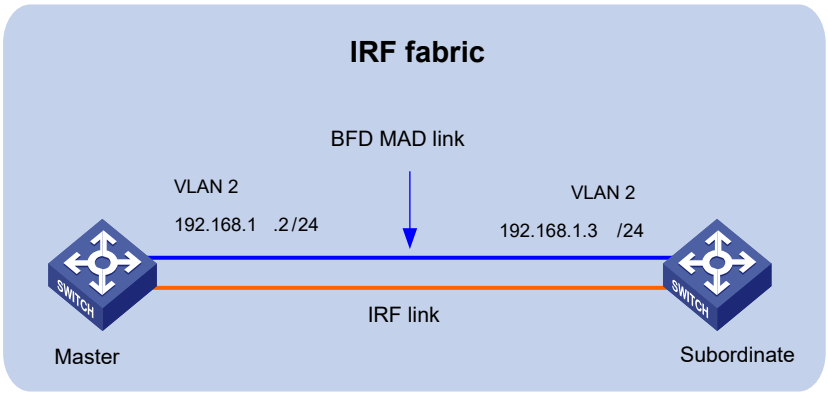


Figure 12 BFD MAD scenario without an intermediate device



4.3 ARP MAD

ARP MAD detects multi-active collisions by using extended ARP packets that convey the IRF domain ID and the active ID (the member ID of the master). You can use common or management Ethernet ports for ARP MAD.

If management Ethernet ports are used, ARP MAD must work with an intermediate device. Make sure the following requirements are met:

- Connect a management Ethernet port on each member device to the intermediate device.
- On the intermediate device, you must assign the ports used for ARP MAD to the same VLAN.

If common Ethernet ports are used, ARP MAD can work with or without an intermediate device. Make sure the following requirements are met:

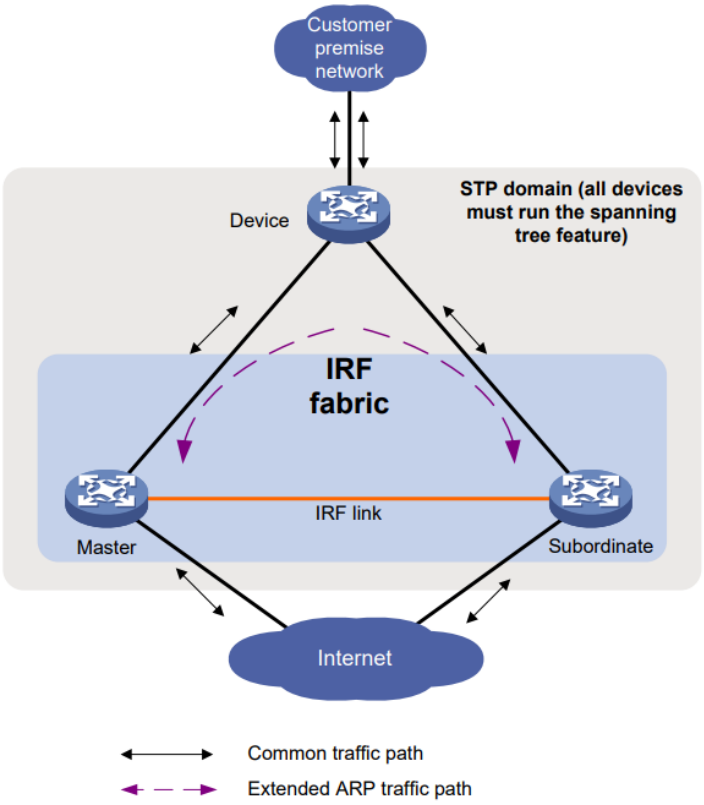
- If an intermediate device is used, connect each IRF member device to the intermediate device, as shown in Figure 13. Run the spanning tree feature between the IRF fabric and the intermediate device. In this situation, data links can be used.

- If no intermediate device is used, connect each IRF member device to all other member devices. In this situation, IRF links cannot be used for ARP MAD.

Each IRF member compares the domain ID and the active ID (the member ID of the master) in incoming extended ARP packets with its domain ID and active ID.

- If the domain IDs are different, the extended ARP packet is from a different IRF fabric. The device does not continue to process the packet with the MAD mechanism.
- If the domain IDs are the same, the device compares the active IDs.
  - If the active IDs are different, the IRF fabric has split.
  - If the active IDs are the same, the IRF fabric is integrated.

Figure 13 ARP MAD scenario (common Ethernet ports)



4.4 Mad Collision Handling

When detecting a multi-active collision, MAD disables all IRF fabrics except one from forwarding data traffic by placing them in Recovery state. The IRF fabrics placed in Recovery state are called inactive IRF fabrics. The IRF fabric that continues to forward traffic is called the active IRF fabric.

1. BFD MAD and LACP MAD use the following process to handle a multi-active collision:
2. Compare the health states of split fabrics.
3. Support for comparing the health states of split fabrics vary by device model.
4. Set all fabrics to the Recovery state except the healthiest one.
5. Compare the number of members in each fabric if all IRF fabrics are in the same health state.
6. Set all fabrics to the Recovery state except the one that has the most members.
7. Compare the member IDs of their masters if all IRF fabrics have the same number of members.
8. Set all fabrics to the Recovery state except the one that has the lowest numbered master.
9. Shut down all common network interfaces in the Recovery-state fabrics except for the following interfaces:
   - Interfaces automatically excluded from being shut down by the system.
   - Interfaces specified by using the **mad exclude interface** command.

ARP MAD and ND MAD use the following process to handle a multi-active collision:

1. Compare the health states of split fabrics.
2. Support for comparing the health states of split fabrics vary by device model.
3. Set all fabrics to the Recovery state except the healthiest one.
4. Compare the member IDs of the masters in the IRF fabrics if all IRF fabrics have the same health state.
5. Set all fabrics to the Recovery state except the one that has the lowest numbered master.
6. Shut down all common network interfaces in the Recovery-state fabrics except for the following interfaces:
    - Interfaces automatically excluded from being shut down by the system.
    - Interfaces specified by using the **mad exclude interface** command.

5 Comparing IRF and DRNI

The Intelligent Resilient Framework (IRF) technology is developed to virtualize multiple physical devices at the same layer into one virtual fabric to provide data center class availability and scalability. IRF virtualization technology offers processing power, interaction, unified management, and uninterrupted maintenance of multiple devices.

Distributed Resilient Network Interconnect (DRNI) virtualizes two physical devices into one system through multi-chassis link aggregation for device redundancy and traffic load sharing.

Table 2 shows the differences between IRF and DRNI. For high availability and short service interruption during software upgrade, use DRNI. You cannot use IRF and DRNI in conjunction on the same device.

Table 2 Comparing IRF and DRNI

Item	IRF	DRNI
Control plane	The IRF member devices have a unified control plane for central management. The IRF member devices synchronize all forwarding entries.	The control plane of the DR member devices is separate. The DR member devices synchronize entries such as MAC, ARP, and ND entries. With DRNI there's no failover for a lot of services as they're totally separated (e.g. BGP).
Device requirements	Hardware: The chips of the IRF member devices must have the same architecture, and typically the IRF member devices are from the same series. Software: The IRF member devices must run the same software version.	Hardware: The DR member devices can be different models. Software: Some device models can run different software versions when they act as DR member devices.
Software upgrade	The IRF member devices are upgraded simultaneously or separately. A separate upgrade is complex. Services are interrupted for about 2 seconds during an upgrade.	The DR member devices are upgrade separately, and the service interruption time is shorter than 1 second during an upgrade. If the software supports graceful insertion and removal (GIR), an upgrade does not interrupt services. For more information about upgrading the DR member devices by using GIR, see the DRNI upgrade guide.
Management	The IRF member devices are configured and managed in a unified manner. Single points of failure might occur when a controller manages the IRF member devices.	The DR member devices are configured separately, and they can perform configuration consistency check for you to remove inconsistencies in the configuration that affects operation of the DR system. You must ensure that service features also have consistent configuration. The DR member devices are managed separately. No single point of failure will occur when a controller manages the DR member devices.
Scaling	IRF can virtualize up to 10 devices	DRNI virtualizes 2 devices

6 IRF with ISSU

The In-Service Software Upgrade (ISSU) feature upgrades software with a minimum amount of downtime. ISSU is implemented based on the following design advantages:

1. Separation of service features from basic functions—Device software is segmented into boot, system, and feature images. The images can be upgraded individually.
2. Independence between service features—Features run independently. One feature can be added or upgraded without affecting the operation of the system or other features.
3. Support for hotfix—Patch images are available to fix system bugs without a system reboot.
4. Hardware redundancy—In an IRF fabric, one member device can be upgraded while other member devices are providing services.

ISSU for a multichassis IRF fabric should be performed as per member in the two steps:

1. First upgrade a subordinate member, and then upgrade the master and the other subordinate members.
2. Before upgrading, use the **display version comp-matrix file** command to verify the compatibility between the new and old images and identify the recommended upgrade methods.

Table 3: IRF with ISSU

Steps	Command	Remarks
1.Enter system view.	**system-view**	N/A
2. (Optional) Set the automatic rollback timer.	**issu rollback-timer** *minutes*	By default, the automatic-rollback interval is 45 minutes. The timer starts when you run the **issu run switchover** command. If you do not execute the **issu accept** or **issu commit** command before this timer expired, the system automatically rolls back to the original software images.
3. Return to user view.	**quit**	N/A
4.Upgrade subordinate members and configure the upgrade images as the main start up software images for the subordinate members.	Method 1: **issu load file** { **boot** *filename* | **system** *filename* | **feature** *filename*&<1-30> } * **chassis** *chassis-number* Method 2: **issu load file ipe** *ipe-filename* **chassis** *chassis-number*&<1-3>	Specify the member IDs of the subordinate members to be upgraded for the **chassis** *chassis-number*&<1-3> option.
5. Perform a master/subordinate switchover.	**issu run switchover**	N/A
6. (Optional) Accept the upgrade and delete the automatic-rollback timer.	**issu accept**	N/A
7. Complete the ISSU process or roll back to the original software images.	To complete the ISSU process, upgrade the subordinate members that have not been upgraded (including the original master) using the following command: **issu commit**	After using the **issu commit** command to upgrade one subordinate member, you must wait for the subordinate member to restart and join the IRF fabric before upgrading

Steps	Command	Remarks
	chassis *chassis-number* To roll back to the original software images:**issu rollback**	another subordinate member. After all members are upgraded, the ISSU process ends and the ISSU status transitions to Init. During this ISSU process, you can use the **issu rollback** command to roll back to the original software images. For more information about rollback, see *Fundamentals Command Reference*.

If a new image is on the Version compatibility list, the new and old images are compatible.

If a new image is not on the Version compatibility list, the new and old images are incompatible.

When you use the issu series commands to install or upgrade the software of MPUs, the device automatically installs or upgrades the software of the interface cards and switching fabric modules as needed. You do not need to install or upgrade the software of the interface cards and switching fabric modules separately.

The ISSU procedure varies by the number of member devices. For an IRF fabric with a single member, the ISSU procedure varies by the number of MPUs.

To perform an ISSU for an incompatible version, execute the following commands in user view:

Table 4: ISSU for an incompatible version

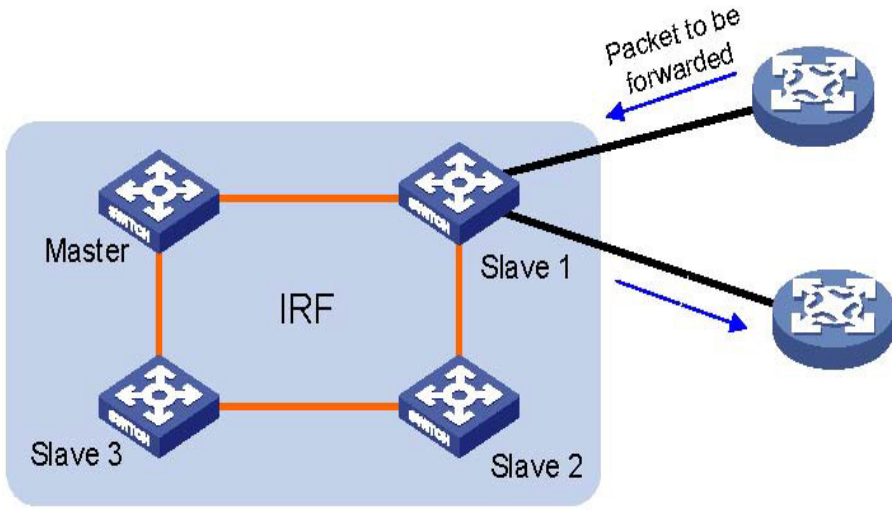
Step	Command	Remarks
1. Upgrade subordinate members and configure the upgrade images as the main start up software images fothe subordinate members.	• Method 1: **issu load file** { **boot** *filename* | **system** *filename* | **feature** *filename*&<1-30> } * **chassis** *chassis-number*&<1-3> • Method 2: **issu load file ipe** *ipe-filename* **chassis** *chassis-number*&<1-3>	Specify the member IDs of the subordinate members to be upgraded for the **chassis** *chassis-number*&<1-3> option. If the member devices of the IRF fabric are connected into a ring topology, HPE recommends that you specify half of the subordinate members for this command to reduce service interruption. Make sure the specified subordinate members are physically connected.
2. Complete the ISSU process or roll back to the original software images.	• To complete the ISSU process**,** perform a master/subordinate switchover to upgrade all members that have not been upgraded: **issu run switchover.** • To roll back to the original software images: **issu rollback**	After all members are upgraded, the ISSU process ends, and the ISSU status transitions to Init. During this ISSU process, automatic rollback is not supported, but you can use the **issu rollback** command to manually roll back to the original software images. For more information about rollback, see *Fundamentals Command Reference*.

7 Packet Forwarding Mechanism

IRF adopts a distributed resilient forwarding technology to implement Layer 2 and Layer 3 packet forwarding, making use of the processing capability of each member device to the maximum extent. Each member device in the stacking system has complete Layer 2 and Layer 3 forwarding capabilities. When a member device receives a Layer 2/3 packet to be forwarded, it finds the outbound interface (and the next hop) of the packet by searching its Layer 2/3 forwarding table, and then forwards the packet from the outbound interface. The outbound interface can be on the local device or on another member device. Forwarding packets from the local device to another member device is unknown to the external, that is, no matter how many member devices the Layer 3 packets traverse, the hop count is increased by one only, that is, the packets traverse one network device only.

As shown in **Figure 14**, the inbound and outbound interfaces of the packet to be forwarded are on the same device. When Slave 1 receives the packet, it searches its forwarding table, and finds that the outbound interface is on itself; then it will forward the packet from the outbound interface.

Figure 14 Intra-device Forwarding



As shown in **Figure 15**, the inbound and outbound interfaces of the packet to be forwarded are not on the same member device. When Slave 1 receives the packet, it searches its forwarding table, and finds that the outbound interface is on the master, and then it forwards the packet to the master according to the optimal path, and the master forwards the packet to the end user through the outbound interface.

Figure 15 Inter-device Forwarding

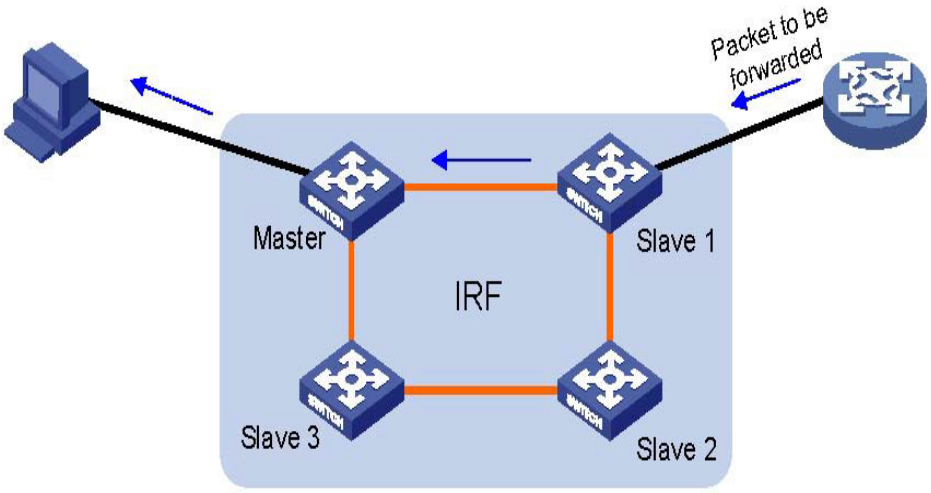
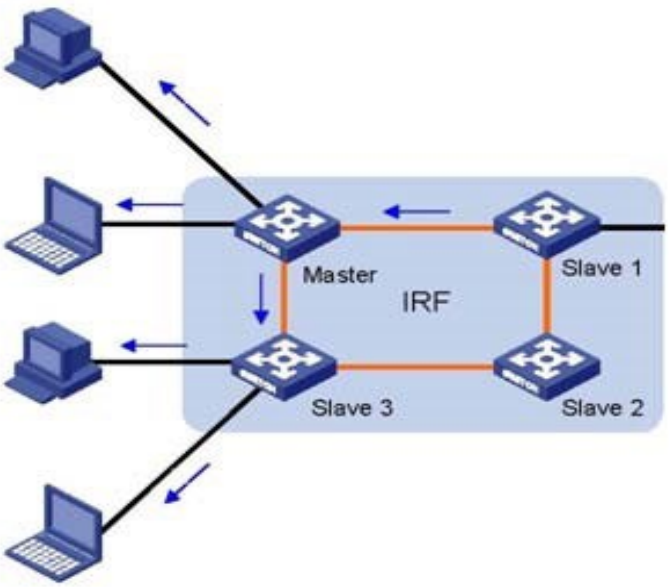


Figure 16 illustrates how IRF processes multicasts. Upon receiving a multicast, Slave 1 searches its multicast forwarding table, and finds that both Master and Slave 3 have connected multicast group members, and the optimal path from Slave 1 to Slave 3 is through Master. Therefore, Slave 1 forwards the multicast to Master, which makes three copies of the multicast, where two of them are forwarded to the multicast group members connected to Master, and the other is forwarded to Slave 3, which then forwards the multicast to other multicast group members.

In this way, each member device only needs to replicate the multicast as needed, ensuring that only one copy of the packet is transmitted among the devices, which saves stacking system resources and increases processing speed of multicasts.

Figure 16 Multicast Forwarding



8 Technical Characteristics

This chapter explains how IRF targets various technical factors in today's switch networking. One of the major benefits of using IRF is the logical switch view with a single management interface, which makes the management and operational tasks very easy.

8.1 Generic Logical Software Architecture

The biggest difference between IRF and other Stacking technologies is that it is not specific to a certain type of product, but it is a generic logical software architecture. With this software architecture, you can Stack devices of the same type to form a single centralized logical device or distributed logical device as needed. For example, you can Stack box-type switches or chassis-type distributed devices so that the consistency of the stacking functions of different types of products can be ensured. On one hand, you can use the function more easily, and on the other hand, the IRF technology will become more and more mature.

In this software architecture, IRF is a relatively independent function. It affects part of the system, rather than the stability of the whole system.

8.2 Mature System Architecture

Different from other stacking technologies, IRF adopts a widely applied system architecture rather than a new architecture.

IRF adopts a generic distributed system architecture. At present, the distributed system architecture has been applied on multiple kinds of HPE devices. A mature architecture has many advantages over a brand-new architecture:

- **Stable system**: Defects of the system developed based on a mature system architecture has been solved, while a brand-new system architecture is bound to bring some problems specific for this architecture.
- Optimal performance to ensure stable, reliable, and effective operation of the stacking system.

8.3 Simplified Chassis-Type Distributed Device

At present, there are few technologies that can Stack chassis-type distributed devices. Even if some technologies can, the logical device formed has limited functions, and supports a few members of devices, for example, some technologies can Stack only two devices. In general design, distributed devices with multiple chassis adopt two-level management. The first level is distribution of chassis, and the second level is distribution of boards on a chassis. Although only one level of switching architecture is added to the present distributed multi-level switching architecture, the implementation complexity increases greatly. Therefore, this scheme delivers high complexity, low performance, low reliability, and is not applicable.

IRF solved this problem by stacking multiple chassis-type distributed devices to form one logical chassis-type distributed device with one AMB, multiple SMBs and multiple interface boards. The only difference between this logical chassis-type distributed device and a common chassis-type distributed device is the number of SMBs and interface boards. The architecture and complexity of the logical device are the same with those of a common chassis-type distributed device.

Therefore, the number of member devices does not depend on the system architecture anymore but depends on the hardware capability.

8.4 Rich and Stable Functions

IRF supports IPv4, IPv6, MPLS, security features, OAA modules, and high availability technologies, and ensures that these functions are effective and stable.
Other stacking technologies adopt a brand-new system architecture, and technologies well applied on other devices need to be supported by each device in a Stack. For example, the high availability technology commonly supported on chassis-type distributed devices is not supported on many stacking technologies, and many functions of the high availability are lost.

However, based on generic software architecture, IRF is an enhancement to the original system functions, without modifying the interfaces and operating mechanism of the original system. Therefore, the stacking system can inherit the functions supported by the original system, ensuring the continuity of the technology and richness of the system functions. Users do not need to know whether different functions are supported in IRF, and how the functions work, facilitating use of the IRF.

8.5 Effective 1:N Backup

Common chassis-type distributed devices adopt 1:1 backup, while IRF adopts 1:N backup, which enhances system reliability because multiple standby main boards are available.

Generally, 1:N backup consumes great bandwidth, and consumes more bandwidth with the increase of the value of N. Other stacking technologies solve the problem in two ways: reduce the supported high availability functions and apply the limited resources to key services; or only back up the manually configured data, and increase service interruption time to reduce the quantity of synchronized data. However, these two methods do not really solve the problem.

IRF solved the problem of O(N) complexity of backed up data by using a multicast group, implementing O(1) algorithm, so that the system resources occupied by multiple SMBs are fixed, and will not change with the increase of the number of SMBs. Therefore, IRF delivers not only high reliability of 1:N backup but also high performance of 1:1 backup.

8.6 Redundancy Protection on a Single Chassis-Type Distributed Device

The stacking technology has the 1:N redundancy protection function, while a chassis type distributed device also has the 1:1 dual-main board redundancy protection function. When stacking chassis-type distributed devices, other stacking technologies make use of only the redundancy function of the stacking technology while give up the backup function of the chassis-type distributed devices.

8.7 Flexible Device Connections

Compared to common stacking technologies, IRF provides more flexible connections. You do not need to connect devices using dedicated cables, and you can specify common Ethernet ports as IRF Stack ports to connect devices. The specified IRF Stack ports can be either electrical ports or optical ports. You can also use optical fibers to connect geographically distributed devices to form a stacking system, making IRF applicable to more networking environments (the requirements on Stack ports depend on the device model). This feature is IRF specific.

9 Application Scenarios

This chapter explains how IRF applies it to highlight distinctive features and trends in the various scenario approaches.

9.1 Increasing Port Numbers

When the number of accessed users increases, and the ports of the switch cannot satisfy users' needs, you can add a switch in the original Stacking system to increase the number of ports.

9.2 Expanding System Processing Capability

When the forwarding capability of the core switch cannot satisfy a user's needs, you can add a switch to form a stacking system with the original core switch. If the forwarding capability of one switch is 64 Mpps, the forwarding capability of the whole Stack system is 128 Mpps after another switch is added. Note that this increases the forwarding capability of the entire stacking system, not a single switch.

9.3 Expanding Bandwidth

You can increase the uplink bandwidth of the edge switch by adding another switch to form a stacking system with the edge switch. You can configure multiple physical links of the member devices as an aggregation group to increase bandwidth of the link to the core switch. To the core switch, the number of edge switches does not change. The original edge switch will back up the current configurations to the newly added switch in batches, which affects the network planning and configuration to the smallest extent.

9.4 Connecting Geographically Distributed Devices

IRF allows you to connect geographically distributed devices using optical fibers to form a stacking system. As shown in Figure 20, users on each floor are connected to the external network through a corridor switch. You can connect the corridor switches to form a stacking system.

In this way, only one access device on each building simplifies the network structure; multiple links to the core network on each floor make the network more Daisy-chain and reliable; configuration of the stacking system rather than multiple corridor switches reduces management and maintenance cost.

9.5 Simplifying Networking

The following is a common networking, which uses MSTP and VRRP to support link backup and gateway backup. This networking is applicable to many environments, and the networking on the distribution and access layers is taken as an example.

With IRF enabled multiple devices on the distribution layer form a single logical device, to which the accessing devices are connected. In this networking, MSTP and VRRP are not needed, simplifying network configuration. Meanwhile, with the inter device link aggregation function, when a member device fails, MSTP and VRRP convergence is no longer needed, thus increasing network reliability.