

实验：BFD

HCIP 分解实验 - BFD



BFD 实验 1：BFD 基本配置

BFD 实验 2：BFD 与 VRRP 配置

=====

BFD 实验 1：BFD 基本配置

为了减小设备故障对网络业务造成的影响，提高网络的可用性，网络设备需要能够尽快检测到与相邻设备之间的通信故障，以便及时采取措施，保证业务尽快恢复正常。

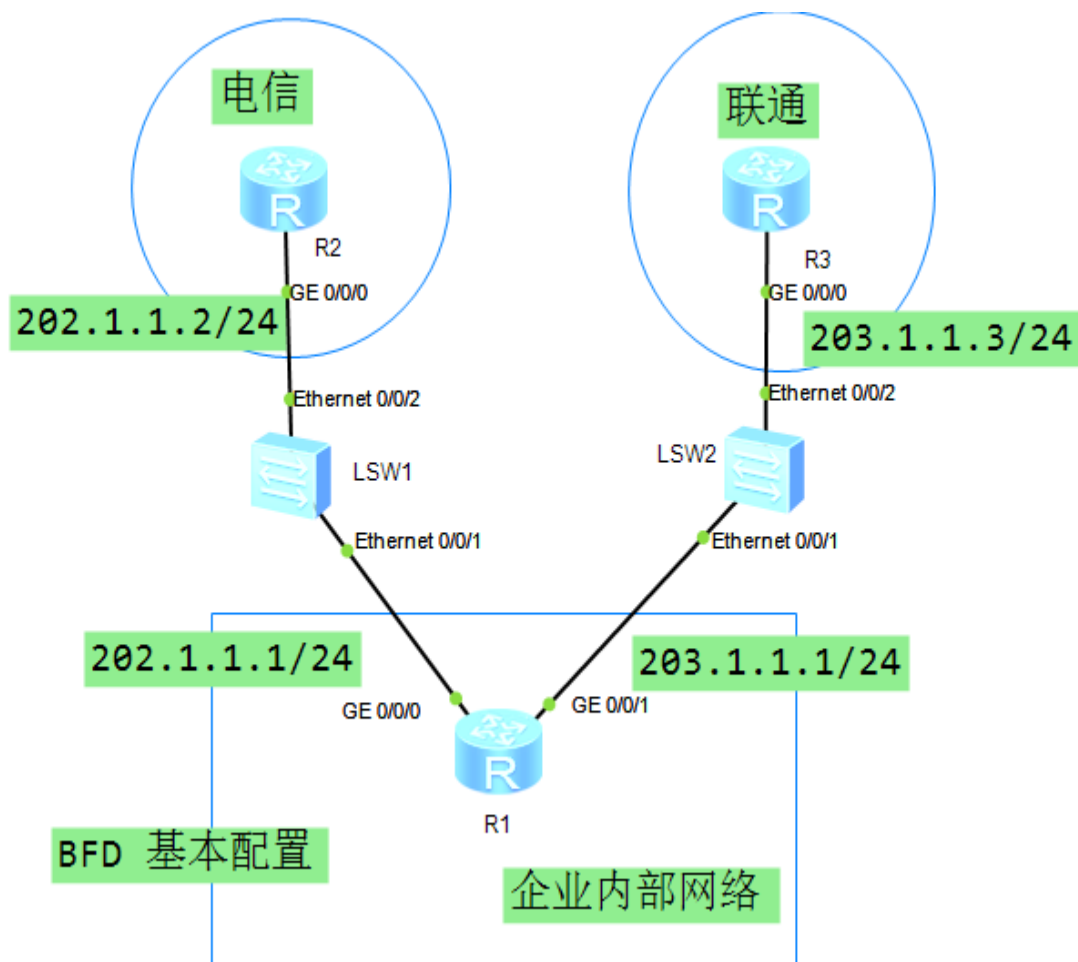
目前，主要的故障检测机制包括两大类：硬件检测机制和慢 Hello 检测机制。例如，SDH (Synchronous Digital Hierarchy) 链路告警检测机制就是一种硬件检测机制，其优点是发现故障的速度很快，缺点是有些传输介质在有的条件下是无法支持这样的硬件检测机制的。慢 Hello 检测机制通常是指路由协议的 Hello 机制，这种检测机制存在的主要问题是发现故障的速度较慢，一般需要秒级的时间，这对于高速链路来说一般是无法接受的，因为秒级的时间将造成大量的数据丢失。除了这

两类故障检测机制外，有的设备厂商提供了一些专用的故障检测机制。然而，在进行不同厂商的设备互联时，这样的专用检测机制通常又难以部署和实施。

BFD (Bidirectional Forwarding Detection) 双向转发侦测，技术就是为了解决现有的故障检测机制的不足而产生的。BFD 可以在相邻设备的转发引擎之间的通信通道上提供轻荷快速的故障检测能力，并在发现故障时即时通知上层应用。BFD 可以发现的故障包括接口故障、链路故障，甚至可以是转发引擎本身的故障等。

双向转发检测 BFD (Bidirectional Forwarding Detection) 是一套全网统一的检测机制，用于快速检测、监控网络中链路或者 IP 路由的转发连通状况。为改善网络性能，相邻系统之间应能快速检测到通信故障，更快地建立起备用通道恢复通信。

BFD 的检测机制可以概括为：首先在两个设备之间建立起 BFD 会话，然后相互周期性地发送 BFD 控制报文，如果一方在预定的时间范围内没有收到另一方发送的 BFD 控制报文，则认为传输路径上发生了故障。为满足快速检测故障的需求，BFD 规定发送和接收控制报文的时间间隔大致为微秒级别。但是，限于目前的设备处理能力，大部分厂商的设备在实际运用 BFD 时都只能达到毫秒级别。



企业网出口路由器 R1 存在两条访问互联网的线路，一个是连接到电信，另一个是连接到联通。正常情况，企业网访问互联网的主链路是经过电信网络，当电信网络故障就走联通备份链路

配置 IP 地址

R1:

```
undo ter mo
sys
sysname R1
user-interface console 0
idle-timeout 0 0
int g0/0/0
ip add 202.1.1.1 24
```

```
int g0/0/1
ip add 203.1.1.1 24
q
```

```
R2:
undo ter mo
sys
sysname R2
user-interface console 0
idle-timeout 0 0
int g0/0/0
ip add 202.1.1.2 24
q
```

```
R3:
undo ter mo
sys
sysname R3
user-interface console 0
idle-timeout 0 0
int g0/0/0
ip add 203.1.1.3 24
q
```

=====

配置默认路由及模拟故障

在企业网出口路由器 R1 配置备份缺省路由，R1 做为主链路

```
R1:
ip route-static 0.0.0.0 0 202.1.1.2
ip route-static 0.0.0.0 0 203.1.1.3 preference 65
```

```
dis ip routing-table protocol static
```

查询路由表，说明：优先级为 60 的是激活路由，是主链路；
优先级为 65 的是不激活路由，是备链路

Static routing table status : <Active>
Destinations : 1 Routes : 1

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
0.0.0.0/0	Static	60	0	RD	202.1.1.2

Static routing table status : <Inactive>
Destinations : 1 Routes : 1

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
0.0.0.0/0	Static	65	0	R	203.1.1.3

当 R2 与交换机 SW1 之间链路故障了，在 R1 的路由表中还是 active

Static routing table status : <Active>
Destinations : 1 Routes : 1

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
0.0.0.0/0	Static	60	0	RD	202.1.1.2

问题说明：

主链路中某段物理链路故障了，企业网出口路由器 R1 仍走原

来的主链路，很显然，此时将不能正常访问互联网了。

原因：

企业网出口路由器 R1 不能检测到与对端路由器之间的链路故障。

解决方法：

通过在路由器之间部署 BFD（双向转发检测），并将 BFD 与缺省路由进行关联，实现快速检测链路故障的效果。

=====

配置 BFD

全局启用 BFD 功能

配置 BFD 延迟 up 功能，默认为 0

bfd

delay-up 5

创建 BFD 会话

配置本地标识符

配置远端标识符

修改 BFD hello 报文的时间间隔 100ms

修改 BFD hello 报文的等待时间 100ms

修改最多等待接收 hello 报文的次数为 3 次，也就是说如果等待达到了 3 次仍没收到对方的 BFD hello 报文，则认为链路发生了故障。

提交配置

R1：

bfd

delay-up 5

quit

bfd R1toR2 bind peer-ip 202.1.1.2 interface g0/0/0
discriminator local 1

```
discriminator remote 2
min-tx-interval 100
min-rx-interval 100
detect-multiplier 3
commit
quit
```

```
R2 :
bfd
delay-up 5
quit
```

```
bfd R2toR1 bind peer-ip 202.1.1.1 interface g0/0/0
discriminator local 2
discriminator remote 1
min-tx-interval 100
min-rx-interval 100
detect-multiplier 3
commit
quit
```

查询 BFD 的会话信息 display bfd session all
[R1]display bfd session all

Local	Remote	PeerIpAddr	State	Type	InterfaceName

1	2	202.1.1.2	Up	S_IP_IF	GigabitEthernet0/0/0

将静态路由与 BFD 会话进行关联

```
R1:
ip route-static 0.0.0.0 0.0.0.0 202.1.1.2 track bfd-
```

session R1toR2

当 R2 与交换机 SW1 之间链路故障了，在 R1 的路由表会发生变化。

现象说明：当出口路由器与电信设备之间的物理链路故障时，BFD 可以迅速检测到，导致 BFD 会话 down，此时与 BFD 会话关联的静态路由也将失效，最终原来的备份路由（优先级为 65）升级为活跃路由，成为主路由；避免因物理链路故障导致的断网现象。

```
[R1]display bfd session all
```

Local	Remote	PeerIpAddr	State	Type	InterfaceName

1	2	202.1.1.2	Down	S_IP_IF	GigabitEthernet0/0/0

```
Static routing table status : <Active>
Destinations : 1          Routes : 1
```

Destination/Mask	Proto	Pre	Cost	Flags	NextHop
0.0.0.0/0	Static	65	0	RD	203.1.1.3

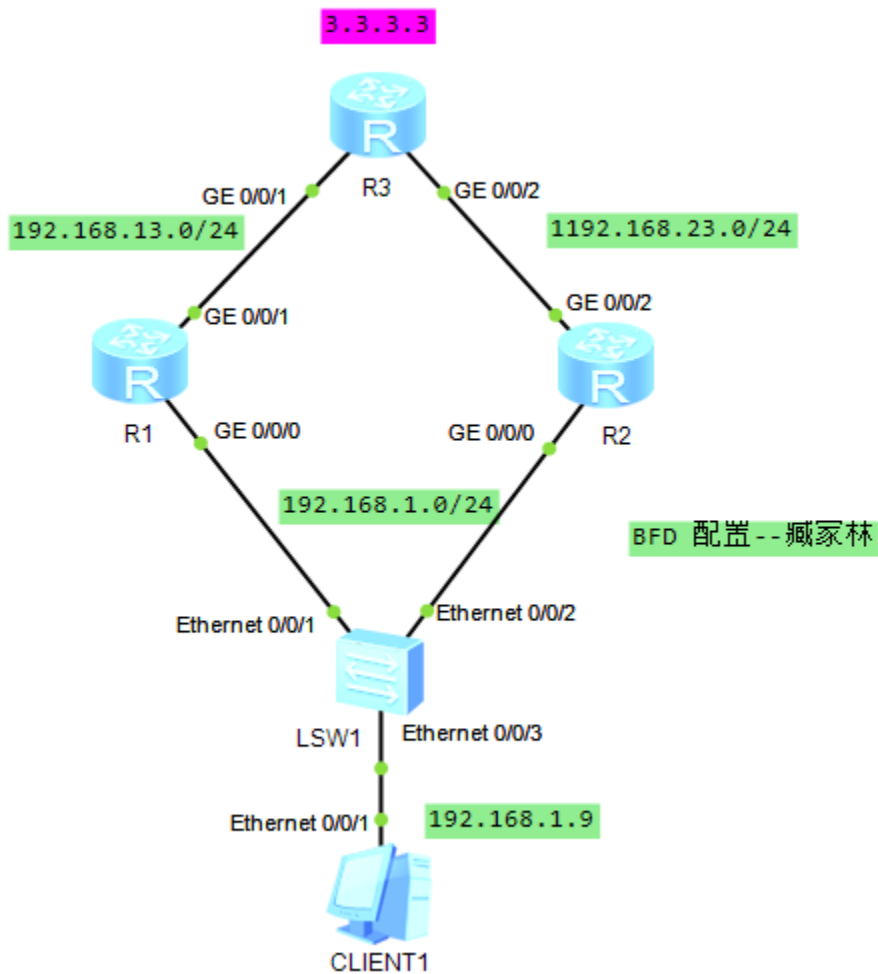
总结

1.BFD 的作用：用于快速检测、监控网络中链路或者 IP 路由的转发连通状况

2.BFD 在现网的部署很常见，可以与多种路由协议和 VRRP 等联合使用

=====

BFD 实验 2：BFD 与 VRRP 配置



R1 和 R2 之间配置 VRRP 备份组，VLAN10 的用户通过缺省网关访问 ISP 的 3.3.3.3 网络。其中，R1 和 R2 的 Dot1q 终结子接口支持 VRRP 功能。用户希望能够快速检测到链路故障，实现 VRRP 快速切换。

配置思路

- 1.配置路由协议，以实现 R1、R2 和 R3 设备之间互通。
- 2.配置 Switch 二层转发功能。
- 3.配置 R1 和 R2 的 Dot1q 终结子接口。

4.在 R1 和 R2 上创建 BFD 会话，以实现对 R1 和 R2 之间链路的检测。

5.分别在 R1 和 R2 的 GE0/0/0.1 接口上创建 VRRP 备份组 1，以实现链路备份。

配置路由协议，以实现 R1、R2 和 R3 设备之间互通。

R1:

```
undo ter mo
sys
sysname R1
user-interface console 0
idle-timeout 0 0
int g0/0/1
ip add 192.168.13.1 24
int g0/0/0.1
ip add 192.168.1.1 24
q
```

R2:

```
undo ter mo
sys
sysname R2
user-interface console 0
idle-timeout 0 0
int g0/0/2
ip add 192.168.23.2 24
int g0/0/0.1
ip add 192.168.1.2 24
q
```

R3:

```
undo ter mo
sys
sysname R3
```

```
user-interface console 0
idle-timeout 0 0
int loop 0
ip add 3.3.3.3 24
int g0/0/1
ip add 192.168.13.3 24
int g0/0/2
ip add 192.168.23.3 24
q
```

配置完成后在 R3 ping 直连测试连通性

```
R1:
ospf router-id 1.1.1.1
area 0
net 1.1.1.1 0.0.0.0
net 192.168.13.1 0.0.0.0
net 192.168.1.1 0.0.0.0
q
```

```
R2:
ospf router-id 2.2.2.2
area 0
net 2.2.2.2 0.0.0.0
net 192.168.23.2 0.0.0.0
net 192.168.1.2 0.0.0.0
q
```

```
R3:
ospf router-id 3.3.3.3
area 0
net 3.3.3.3 0.0.0.0
net 192.168.13.3 0.0.0.0
net 192.168.23.3 0.0.0.0
```

q

在 R3 上查看 OSPF 邻居的建立

<R3>display ospf peer brief

[R3]dis ospf peer bri

OSPF Process 1 with Router ID 3.3.3.3
Peer Statistic Information

Area Id	Interface	Neighbor id	State
0.0.0.0	GigabitEthernet0/0/1	1.1.1.1	Full
0.0.0.0	GigabitEthernet0/0/2	2.2.2.2	Full

=====

配置 Switch 二层转发功能

在 SW1 上配置 e0/0/1 和 e0/0/2 端口为 trunk，允许 VLAN 10 的数据通过，端口 e0/0/3 为 access，加入到 VLAN 10

SW1:

undo ter mo

sys

sysname SW1

user-interface console 0

idle-timeout 0 0

vlan 10

q

```
int e0/0/1
port link-type trunk
port trunk allow-pass vlan 10
int e0/0/2
port link-type trunk
port trunk allow-pass vlan 10
int e0/0/3
port link-type access
port default vlan 10
q
```

=====

配置 R1 和 R2 的 Dot1q 终结子接口

在 R1 和 R2 上配置 dot1q 报文，并开启 ARP 广播功能。

R1:

```
int g0/0/0.1
dot1q termination vid 10
arp broadcast enable
```

R2:

```
int g0/0/0.1
dot1q termination vid 10
arp broadcast enable
```

=====

在 R1 和 R2 上创建 BFD 会话，以实现 R1 和 R2 之间链路的检测

配置 BFD

以 R1 为例，全局启用 BFD 功能。

选做：

配置 BFD 延迟 up 功能，默认为 0

bfd

delay-up 50

修改 BFD hello 报文的时间间隔 100ms

修改 BFD hello 报文的等待时间 100ms

修改最多等待接收 hello 报文的次数为 3 次，也就是说如果等待达到了 3 次仍没收到对方的 BFD hello 报文，则认为链路发生了故障。

min-tx-interval 100

min-rx-interval 100

detect-multiplier 3

R1：

建立 BFD 会话，配置本地及远端标识符，
最后是提交 BFD 配置

bfd

q

bfd 1-2 bind peer-ip 192.168.1.2 interface g0/0/0.1

discriminator local 1

discriminator remote 2

commit

R2:

bfd

q

bfd 2-1 bind peer-ip 192.168.1.1 interface g0/0/0.1

discriminator local 2

discriminator remote 1

```
commit
q
```

查看 BFD 会话状态

```
<R1>display bfd session all
      dis bfd session all verbose
```

```
[R1]dis bfd session all
```

```
-----
Local Remote      PeerIpAddr      State      Type      InterfaceName
-----
1      2          192.168.1.2      Up          S_IP_IF      GigabitEthernet0/0/0.1
-----
Total UP/DOWN Session Number : 1/0
```

```
[R2]dis bfd session all
```

```
-----
Local Remote      PeerIpAddr      State      Type      InterfaceName
-----
2      1          192.168.1.1      Up          S_IP_IF      GigabitEthernet0/0/0.1
-----
Total UP/DOWN Session Number : 1/0
```

可以看到，BFD 会话已经是 UP 状态

=====

分别在 R1 和 R2 的 GE0/0/0.1 接口上创建 VRRP 备份组 1，
以实现链路备份

配置 VRRP 协议

R1 和 R2 使用子接口配置 VRRP，R1 为主网关 master, R2 为备份网关，虚拟 IP 地址为 192.168.1.254。

在 R1 的 g0/0/0.1 子接口下创建备份组 1，并配置 R1 在备份组中的优先级为 120。为了突出与配置了 BFD 之后的切换速度差异，现将 VRRP 的 Hello 报文间隔修改为 10s，默认为 1s, R2 也进行类似的操作。

R1:

```
int g0/0/0.1
vrrp vrid 1 virtual-ip 192.168.1.254
vrrp vrid 1 priority 120
vrrp vrid 1 timer advertise 10
dot1q vrrp vid 10
```

R2:

```
int g0/0/0.1
vrrp vrid 1 virtual-ip 192.168.1.254
vrrp vrid 1 timer advertise 10
dot1q vrrp vid 10
```

查看效果

在 PC 上长 ping 3.3.3.3

PC>ping 3.3.3.3 -t

如果 R1 与交换机之间的链路发生故障，VRRP 会把工作网关的角色从 R1 切换到 R2. 但是，由于现在还没有配置 BFD 功能，所以此切换过程比较缓慢，切换过程中丢包较多。

把 R1 的 g0/0/0.1 手工关闭，查看 PC 的丢包情况


```
From 3.3.3.3: bytes=32 seq=7 ttl=254 time=62 ms
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
Request timeout!
From 3.3.3.3: bytes=32 seq=27 ttl=254 time=62 ms
From 3.3.3.3: bytes=32 seq=28 ttl=254 time=46 ms
From 3.3.3.3: bytes=32 seq=29 ttl=254 time=47 ms
```

=====

配置 VRRP 监视 BFD

R1:

```
int g0/0/0.1
```

```
vrrp vrid 1 track bfd-session 1 reduced 40
```

R2:

```
int g0/0/0.1
```

```
vrrp vrid 1 track bfd-session 2
```

把 R1 的 g0/0/0.1 手工关闭，查看 PC 的丢包情况

配置完成后，再用 PC 来测试一下，发现丢的包就很少，很快就切换到另一个路由器上

```
PC>ping 3.3.3.3 -t
```

```
Ping 3.3.3.3: 32 data bytes, Press Ctrl_C to break
From 3.3.3.3: bytes=32 seq=1 ttl=254 time=156 ms
From 3.3.3.3: bytes=32 seq=2 ttl=254 time=94 ms
From 3.3.3.3: bytes=32 seq=3 ttl=254 time=78 ms
From 3.3.3.3: bytes=32 seq=4 ttl=254 time=78 ms
From 3.3.3.3: bytes=32 seq=5 ttl=254 time=94 ms
From 3.3.3.3: bytes=32 seq=6 ttl=254 time=94 ms
Request timeout!
From 3.3.3.3: bytes=32 seq=8 ttl=254 time=109 ms
From 3.3.3.3: bytes=32 seq=9 ttl=254 time=94 ms
From 3.3.3.3: bytes=32 seq=10 ttl=254 time=110 ms
From 3.3.3.3: bytes=32 seq=11 ttl=254 time=125 ms
```