

## How to use Bidirectional Forwarding Detection(BFD) in NetScaler?

Bidirectional Forwarding Detection commonly referred as BFD is a simple hello protocol that provides fast failure detection mechanism between two routers/nodes. It detects failures on any bi-directional forwarding paths, such as direct physical links, tunnels, virtual links, multi-hop paths across network devices. BFD is a mechanism that is independent of media, routing protocol, and data protocol used and validates the operation of the forwarding plane. Since it is not tied to any routing specific protocols it can be used as a generic failure detection method between network devices.

### Why BFD?

In Routing protocols, fault resolution happens in order of tens of seconds. Reducing the OSPF and BGP timers can bring it down to four to five seconds to detect failure, whereas BFD can do the trick in less than one second. This helps in faster network convergence, short application interruptions, and enhanced network reliability. Another advantage is that BFD is lightweight and can run completely on the data plane offloading the control plane CPU.

### How does BFD work?

BFD session establishment happens in a series of steps. Let us assume the routing protocol used here is OSPF(BGP follows similar steps for other routing protocols).

- 1) OSPF discovers neighbours by sending Hello packets and establishes adjacencies with the neighbouring nodes.
- 2) OSPF notifies BFD of the neighbour to be monitored by including source and destination address.
- 3) BFD uses the information to establish session and sends control packets at regular intervals to the peer nodes.

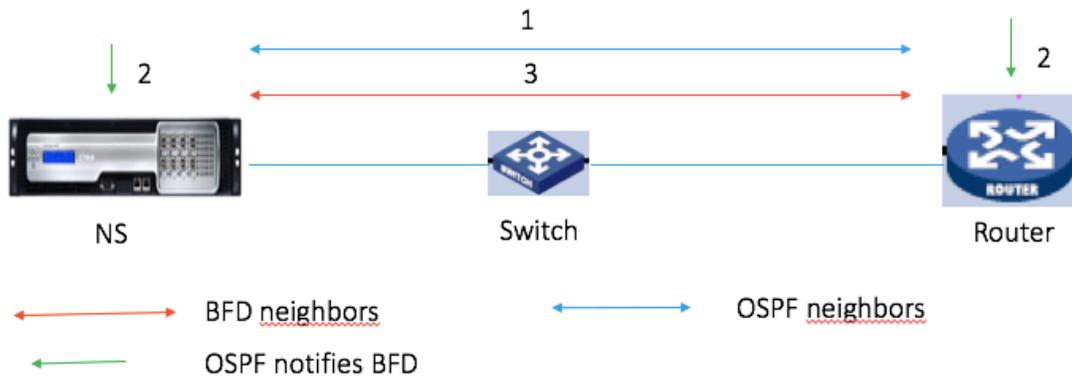


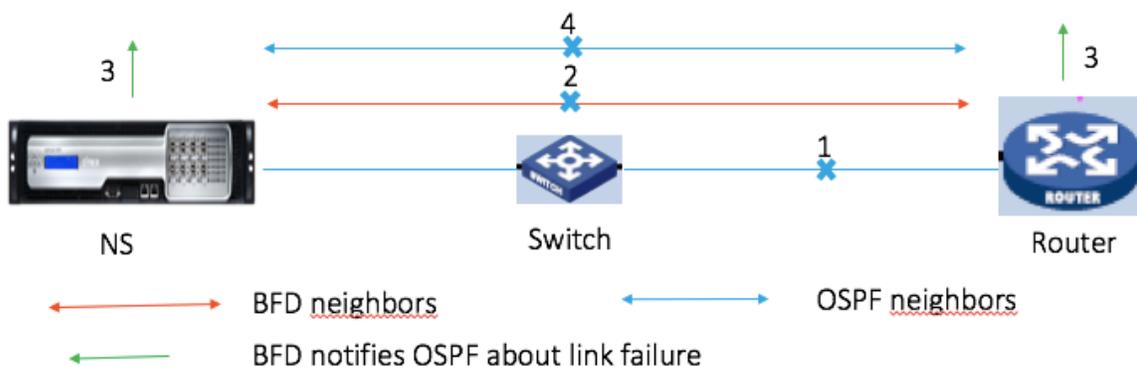
Fig: BFD session establishment

There are 3 modes of BFD operation:

1. Asynchronous - Control packets flow both directions periodically. (Note: This is supported in NetScaler)
2. Demand - Periodic control packets are not sent. After BFD session establishment, a node can ask the other system to stop sending BFD Control packets, except when the node feels needs to verify connectivity explicitly. (Note: This is not supported in NetScaler)
3. Echo mode- a stream of Echo packets is transmitted in such a way that it loops back through its forwarding path to the same node to verify connectivity. (Note: This is not supported in NetScaler)

Failure detection happens in a series of steps as mentioned below,

- 1) A link goes down between NetScaler and adjacent node
- 2) BFD sessions goes down immediately based on timer expiry
- 3) BFD notifies neighbour unreachability to OSPF
- 4) OSPF terminates the neighbour adjacency



*Fig: BFD failure detection*

How to configure BFD in NetScaler?

### **Enabling BFD for OSPF**

The below mentioned configuration has to be done at VTYSH to enable BFD for OSPF.

*Enable or Disable BFD for all interfaces at ospf router level*

```
ZebOS(config-router)#bfd all-interfaces
```

```
ZebOS(config-router)#no bfd all-interfaces
```

*Enable or disable BFD at interface level for OSPF*

```
ZebOS(config-if)#ip ospf bfd
```

```
ZebOS(config-if)#ip ospf bfd disable
```

### **Enabling BFD for OSPFv3**

*Enable or Disable BFD for all interfaces at ospfv3 router level*

```
ZebOS(config-router)#bfd all-interfaces
```

```
ZebOS(config-router)#no bfd all-interfaces
```

*Enable or disable BFD at interface level for OSPFv3*

```
ZebOS(config-if)#ipv6 ospf bfd
```

```
ZebOS(config-if)#ipv6 ospf bfd disable
```

### **Enabling BFD for BGP**

#### **BGP IPv4 Singlehop-peer**

```
ns(config-router)#neighbor <ipv4addr> fall-over bfd
```

```
ns(config-router)#no neighbor <ipv4addr> fall-over bfd
```

#### **BGP IPv4 Multihop-peer**

```
ns(config-router)#neighbor <mh-neighbor-ipv4addr> fall-over bfd multihop
```

```
ns(config-router)#no neighbor <mh-neighbor-ipv4addr> fall-over bfd multihop
```

### **BGP IPv6 Singlehop-peer**

```
ns(config-router)#neighbor <ipv6addr> fall-over bfd
```

```
ns(config-router)#no neighbor <ipv6addr> fall-over bfd
```

### **BGP IPv6 Multihop-peer**

```
ns(config-router)#neighbor <mh-neighbor-ipv6addr> fall-over bfd multihop
```

```
ns(config-router)#no neighbor <mh-neighbor-ipv6addr> fall-over bfd multihop
```

## **Configuring BFD Timers**

### **BFD Single-hop Session Timer**

*Configure BFD single-hop sessions timer and reception interval in millisecond, and the Hello multiplier in interface mode*

```
bfd singlehop-peer interval <100-30000> minrx <100-30000> multiplier <1-20>
```

*Multiplier indicates the number of packets to be dropped before declaring the interface as down*

*Minrx is the value that is advertised to other nodes as the time interval for reception by NetScaler*

*Interval is the time interval with which NetScaler sends hello packets*

*To unset the timers:*

```
no bfd singlehop-peer
```

*Ex: ZebOS(config)#interface vlan10*

```
ZebOS(config-if)#bfd singlehop-peer interval 100 minrx 100 multiplier 4
```

### **BFD Multihop Peer Timer**

*Configure BFD multihop-peer timer and reception intervals in milliseconds, and the Hello multiplier in config mode*

```
bfd multihop-peer A.B.C.D interval <100-30000> minrx <100-30000> multiplier <1-20>
```

*Unset the timers using no command:*

```
no bfd multihop-peer A.B.C.D
```

*Ex: ZebOS(config)#bfd multihop-peer 20.1.1.3 interval 100 minrx 100 multiplier 3*

### **BFD IPV6 Multihop Peer Timer**

```
bfd multihop-peer ipv6 X:X::X:X interval <100-30000> minrx <100-30000> multiplier <1-20>
```

*Unset the timers using no command:*

```
no bfd multihop-peer ipv6 X:X::X:X
```

*Ex: ZebOS(config)#bfd multihop-peer ipv6 3001:20::4 interval 100 minrx 100 multiplier 3*

### **BFD Passive mode**

*NetScaler will not initiate BFD control packets when passive mode is enabled. This is used where the device need not initiate control packets first. Once it receives a control packet, it triggers state change.*

*BFD Passive mode can be configured as follows in interface mode:*

```
ZebOS(config)#interface <vlanid>
```

```
ZebOS(config-if)#bfd passive
```

*To unset the passive setting:*

```
ZebOS(config-if)#no bfd passive
```

BFD in NetScaler is supported for both IPv4 and IPv6 . It is supported for routing protocols in the context of admin-partitions, on spotted SNIP in the context of cluster and on SNIP-IP in HA\_INC mode.