# BFD

Piotr Wojciechowski (CCIE #25543)

# About me

- Senior Network Engineer MSO at VeriFone Inc.
- Previously Network Solutions Architect at one of top polish IT integrators
- CCIE #25543 (Routing & Switching)
- Blogger – http://ccieplayground.wordpress.com
- Administrator of CCIE.PL board
  - The biggest Cisco community in Europe
  - Over 6100 users
  - 3 admin, 7 moderators
  - 48 polish CCIEs as members, 20 of them actively posting
  - About 150 new topics per month
  - About 1000 posts per month
  - English section available!

# Agenda

- High Availability Networks
- Convergence Time
- BFD Protocol
- BFD Caveats

# High Availability Networks

# High Availability Networks

| Availability | DPM | Downtime per Year (24x365) | | |
|---|---|---|---|---|
| 99.000% | 10000 | 3 Days | 15 Hours | 36 Minutes |
| 99.500% | 5000 | 1 Day | 19 Hours | 48 Minutes |
| 99.900% | 1000 | | 8 Hours | 46 Minutes |
| 99.950% | 500 | | 4 Hours | 23 Minutes |
| 99.990% | 100 | |  | 53 Minutes |
| 99.999% | 10 | | | 5 Minutes |
| 99.9999% | 1 | | | 30 Seconds |

- Reactive
- Proactive
- Predictive
- "High Availability"

# High Availability Networks

- System Level HA Features:
  - NSF, NSR
  - SSO
  - Stateful NAT
  - Stateful IPSec
  - Stateful Firewall
  - MPLS HA for L3VPN, L2VPN, TE, CSC etc.
  - Dual OS (ie. IOS-XE redundancy)

# High Availability Networks

- Network Level HA Features:
  - Resilient Network Design
  - Fast Convergence (routing protocols optimization)
  - Graceful restart
  - RSTP, RPVSTP, MST
  - HSRP/VRRP/GLBP
  - BFD
  - MPLS FRR
  - IP FRR
  - Pseudowire  Redundancy

# Convergence Time

# Routing Convergence Components

1. Failure detection
2. Failure information propagation
3. Topology and routing recalculation
4. Update of RIB and FIB tables

# Routing Convergence Components

Detection of Link layer failure

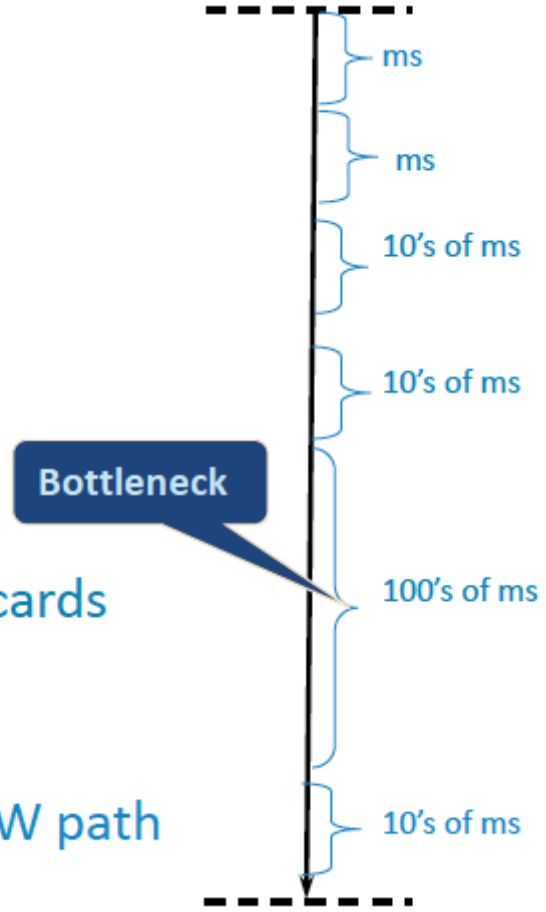
Report failure to Route Controller

Generate and flood an LSP

Trigger and Compute an SPF

Communicate new FIB entries to linecards

Install new FIB entries into linecard HW path



# Failure Detection

- Failure detection can occur on different layers:
  - Physical Layer
  - Data Link Layer
  - Network Layer
  - Application Layer

# Failure Detection

| | |
|---|---|
| Application | • BFD for VCCV, GRE, Fabric Path, TRILL OAM, Y1731 PM, 802.1ag CFM |
| Layer 3 | • BFD for BGP, OSPF, IS-IS, EIGRP, FHRP and static |
| MPLS (Layer 2.5) | • BFD for MPLS LSP, TE-FRR |
| Layer 2 | • UDLD, LACP, 802.3ah (Link OAM), 802.1ag CFM, Y.1731 FM, othre |
| Layer 1 | • Fiber cut, loss of physical link, auto-negotiation, Remote Fault Detection, other |

# BFD (Bidirectional Forwarding Protocol)

# Layer 3 Failure Detection

- Why do we need to detect failure on Layer 3?
  - Required if failure detection cannot be performed on Layer 1 or Layer 2 or takes to much time (for WAN links it can be even over 10 seconds)
  - Concerns over protocols software failure
  - Concerns over unidirectional failures
  - Control-plane failure detection – it takes usually 15 to 50 seconds in default configuration

# Layer 3 Failure Detection

- Layer 3 protocols use HELLO timers to:
  - Maintain adjacencies
  - Check neighbor reachability and detect failure

- Can we tune HELLO and HOLDTIME timers?
  - Yes, but safe in small and well controlled environments
  - Each interface may run multiple protocols
  - Increased CPU utilization
  - Challenges with ISSU and SSO
  - Configuration and design complexity
  - Challenges to achieve expected convergence times

# BFD

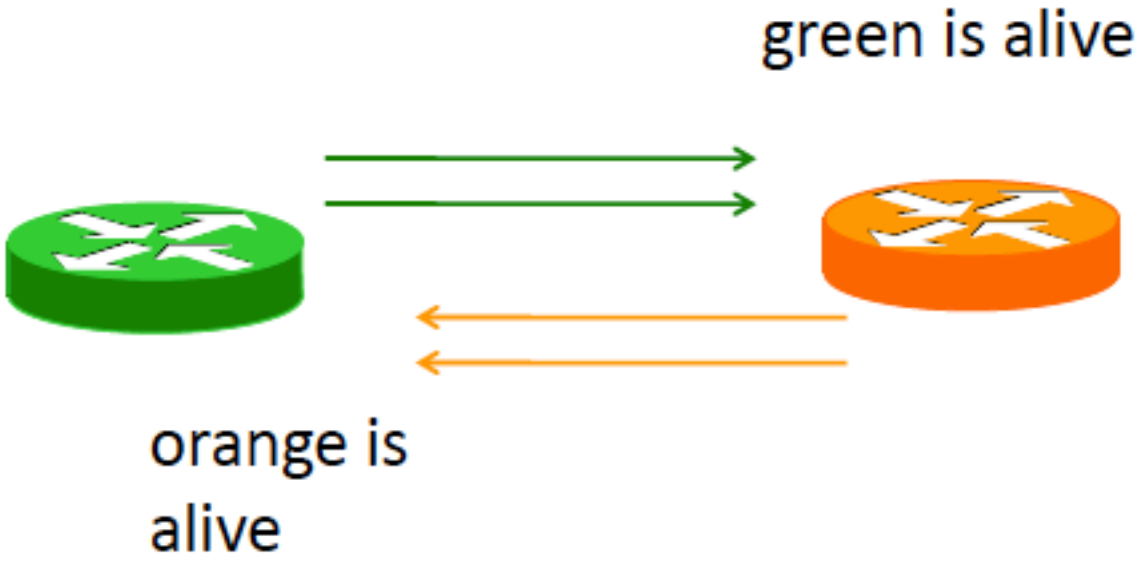
- Hello-type protocol designed to run over multiple transport protocols (IPv4, IPv6, MPLS, TRILL)
- Designed for sub-second Layer 3 failure detection
- Routing protocols are clients of the BFD and receives information as soon as BFD detects a neighbor loss
- Runs on physical, virtual and bundle interfaces (be aware of limitations!)
- Uses UDP ports 3784 and 3785

# BFD

- Advantages (some but not all):
  - Sub-second failure detection
  - Reduced control-plane load
  - Reduced link bandwidth usage
  - Timer negotiation
- Platform specific advantages:
  - Stateful restart
  - SSO and ISSU aware
  - Distribute architecture – I/O modules responsible for transmitting and receiving messages
  - Per-link implementations

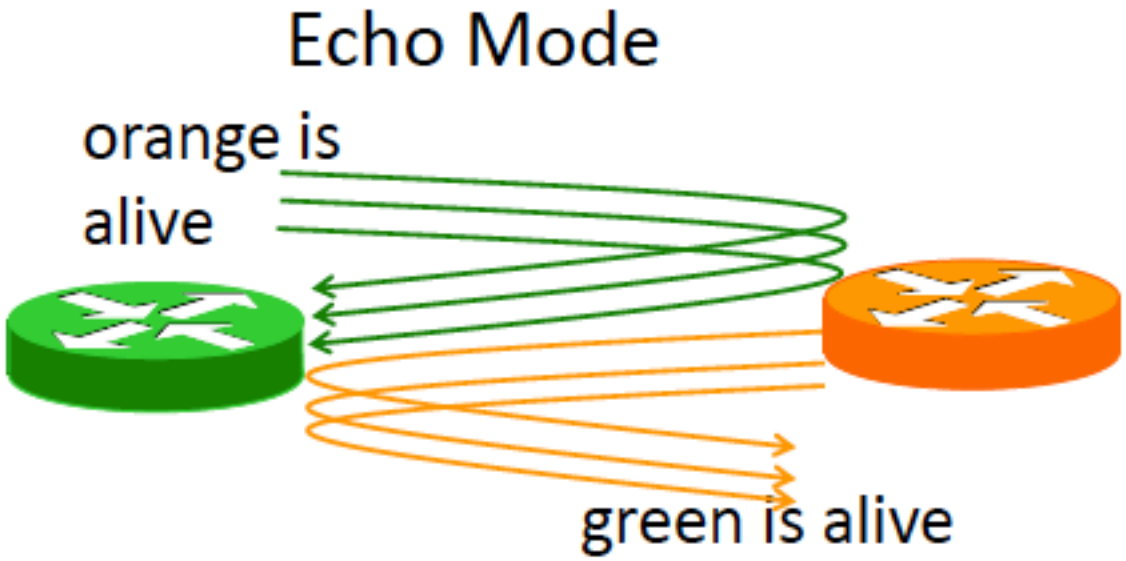
# BFD Operation Modes

- Asynchronous
  - Independent sessions
  - Hello packets sent at negotiated rate
  - Neighbor marked down if no packets received in time period defined as hello_interval*multiplier



# BFD Operation Modes

- Asynchronous + echo
  - Control packets sent at slower rate
  - Self-directed echo packets sent at fast negotiated rate are used for failure detection

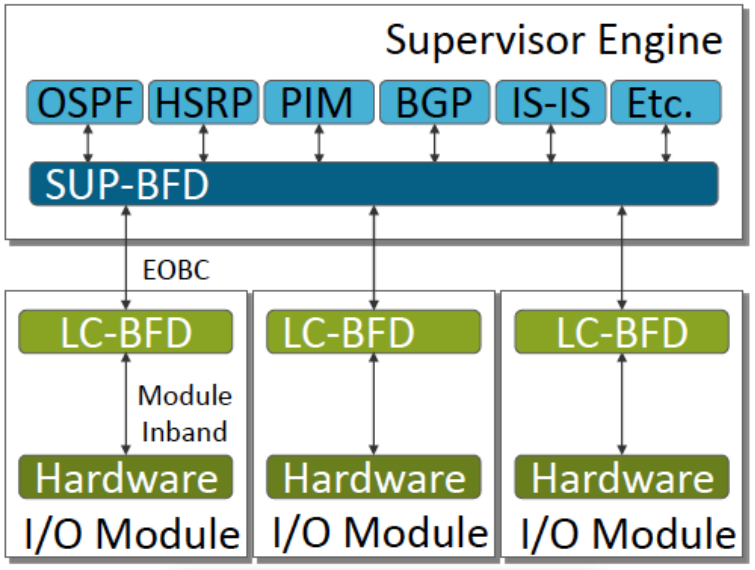


# BFD Operation Modes

- Demand mode
  - No *Hello* packets are exchanged after session is established
  - It is assumed that the endpoints have another way to verify connectivity to each other
  - Either host may still send *Hello* packets if needed
  - Demand mode have to be enabled in both direction separately

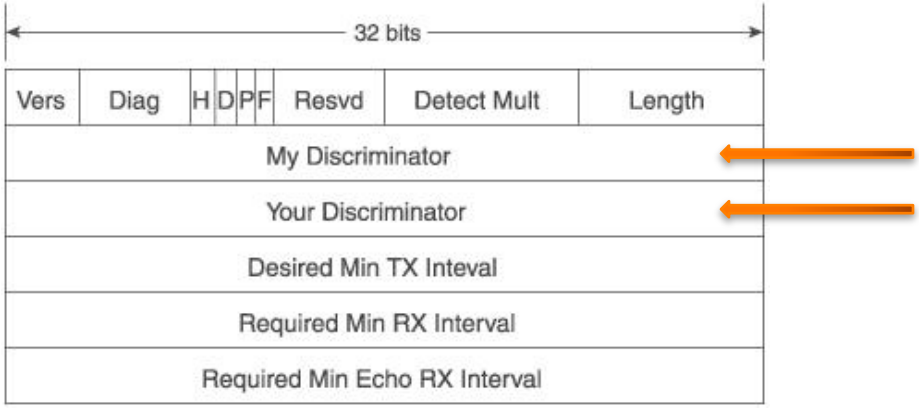
# BFD ON DISTRIBUTED ARCHITECTURE

- Nexus 7000
  - SUP-BFD process running on supervisor engine
    - Interface for BFD clients
  - LC-BFD process running on CPU of each line card
    - Generates BFD Hellos
    - Receives BFD Hellos



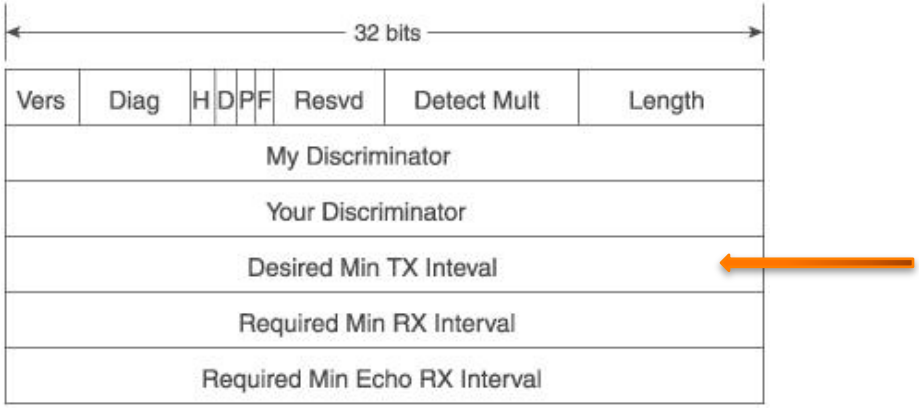
# BFD Packet

- Because multiple sessions can be active on single link Discriminators are used to identify and demultiplex packets for each session
- Your Discriminator is 0 on first packet to request neighbor ID.



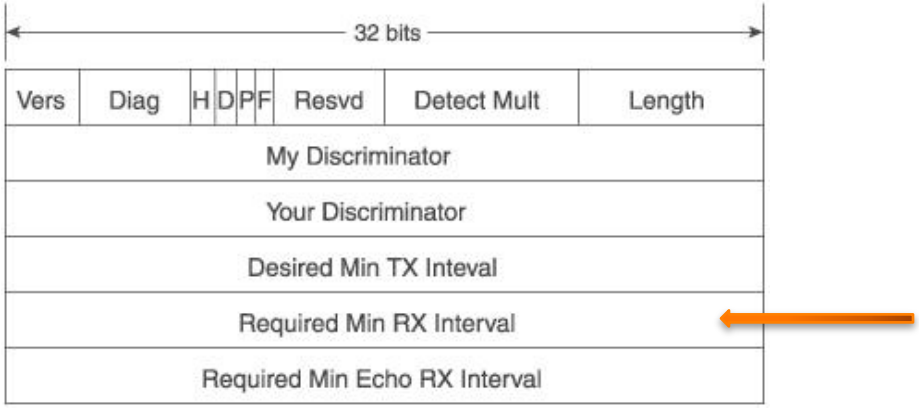
# BFD Packet

- **Desired Minimum Transmit Interval** is the minimum interval, in microseconds, that the originating system would like to use between transmitted control packets.



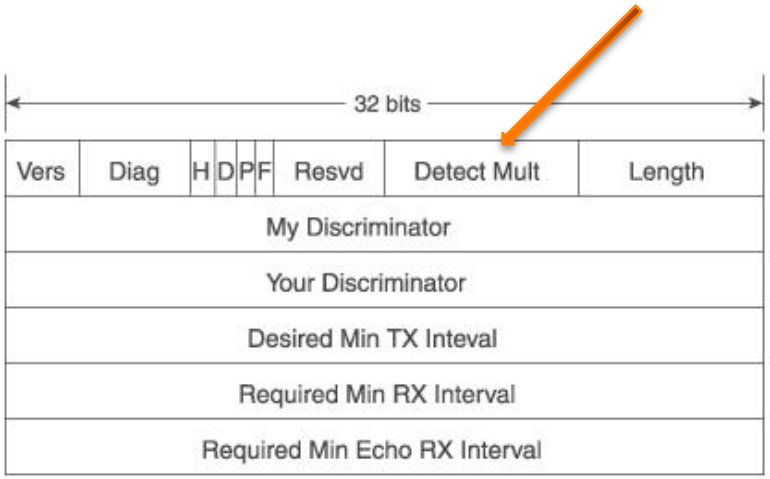
# BFD Packet

- **Required Minimum Receive Interval** is the minimum interval, in microseconds, that the originating system can receive control packets.



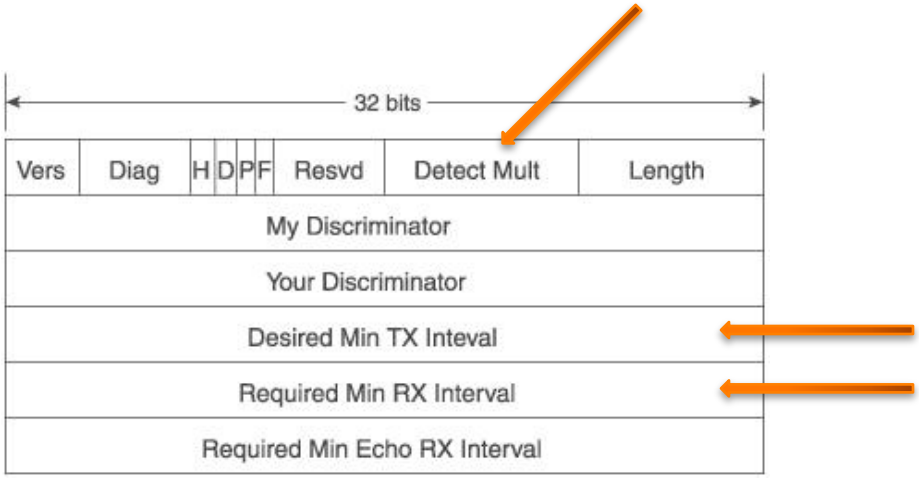
# BFD Packet

- **Detect Multiplier** is a nonzero integer multiplier applied to the negotiated transmission interval that specifies the period (the detection time) a system will wait to hear a control packet before declaring the BFD session down.



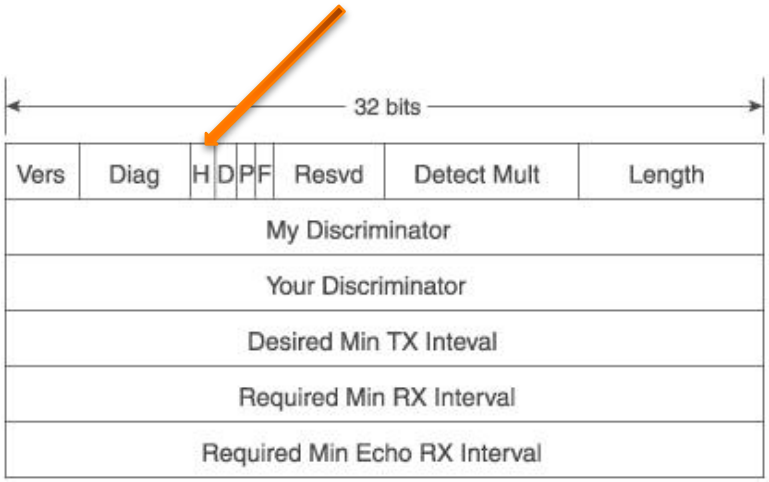
# BFD Packet

- The two timers and the detection multiplier are continuously negotiated, are independent in each direction, and can be changed at any time.
- Each system transmits the period it would like to transmit control packets, and the minimum period it is willing to receive control packets.



# BFD Packet

- **I Hear You (H)** is cleared (0) if the transmitting system either is not receiving BFD packets from the remote system or is in the process of tearing down the BFD session.

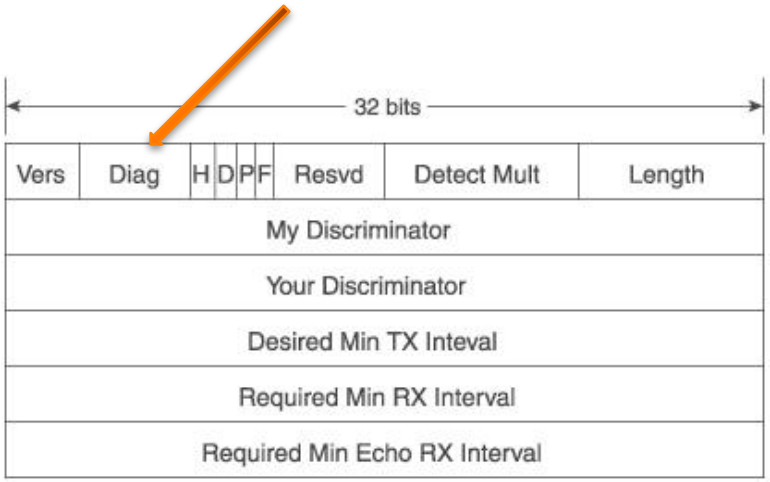


# BFD Packet

A diagnostic code specifying the local system's reason for the last transition of the session from Up to some other state.

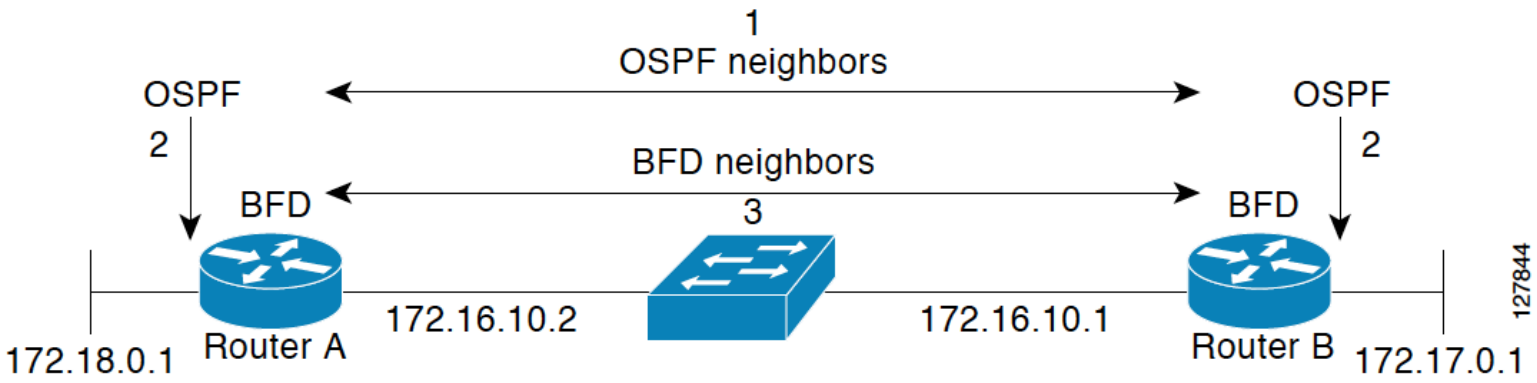
Possible values are:

- 0-No Diagnostic
- 1-Control Detection Time Expired
- 2-Echo Function Failed
- 3-Neighbor Signaled Session Down
- 4-Forwarding Plane Reset
- 5-Path Down
- 6-Concatenated Path Down
- 7-Administratively Down



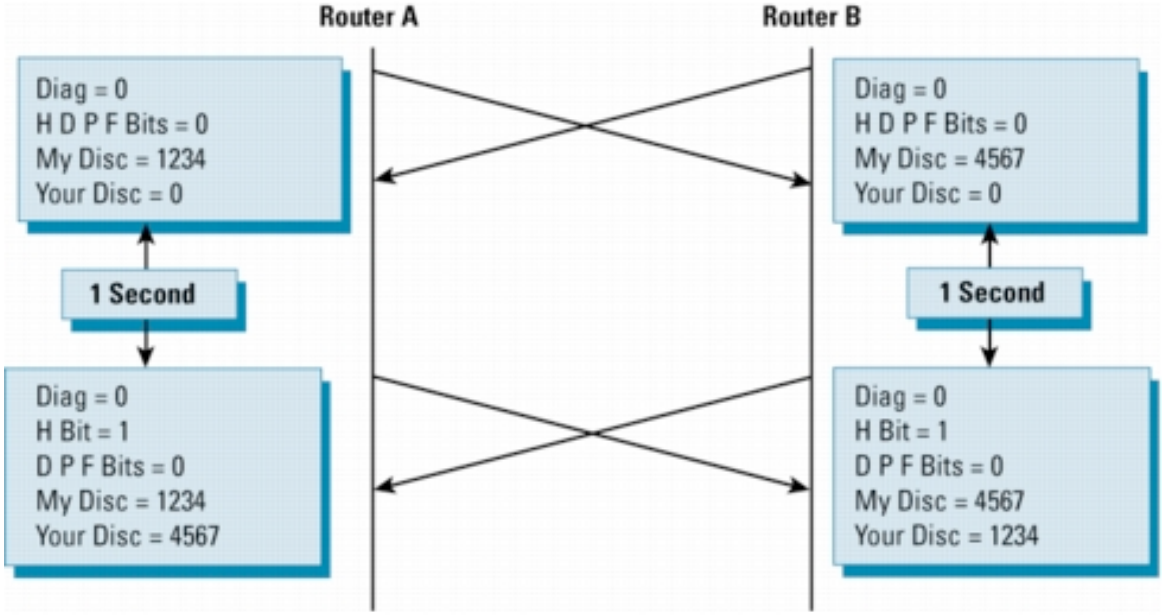
# BFD Neighbor relationship

- Establishing BFD Neighborship
  1. OSPF (or any other protocol) discovers neighbor
  2. OSPF sends request to BFD process to initiate BFD session with OSPF neighbor
  3. BFD session is established



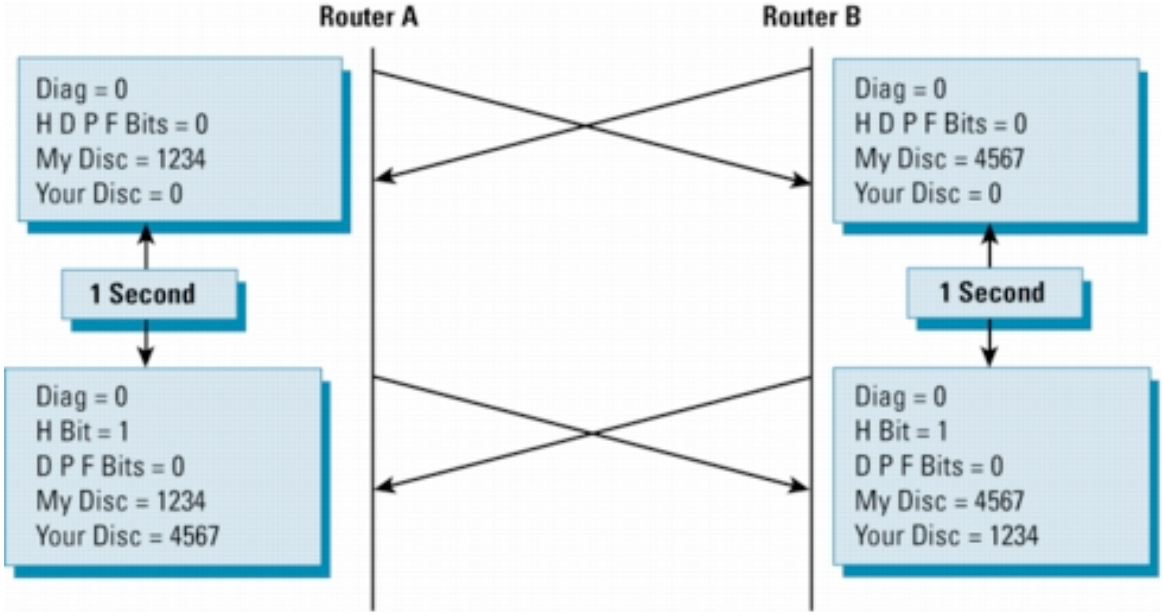
# BFD Initial Session Setup

- Initial packets are sent every second until session is established
- Flags initially set to 0
- My Discriminator will be set to a value which is unique on the transmitting router
- Your Discriminator is set to zero



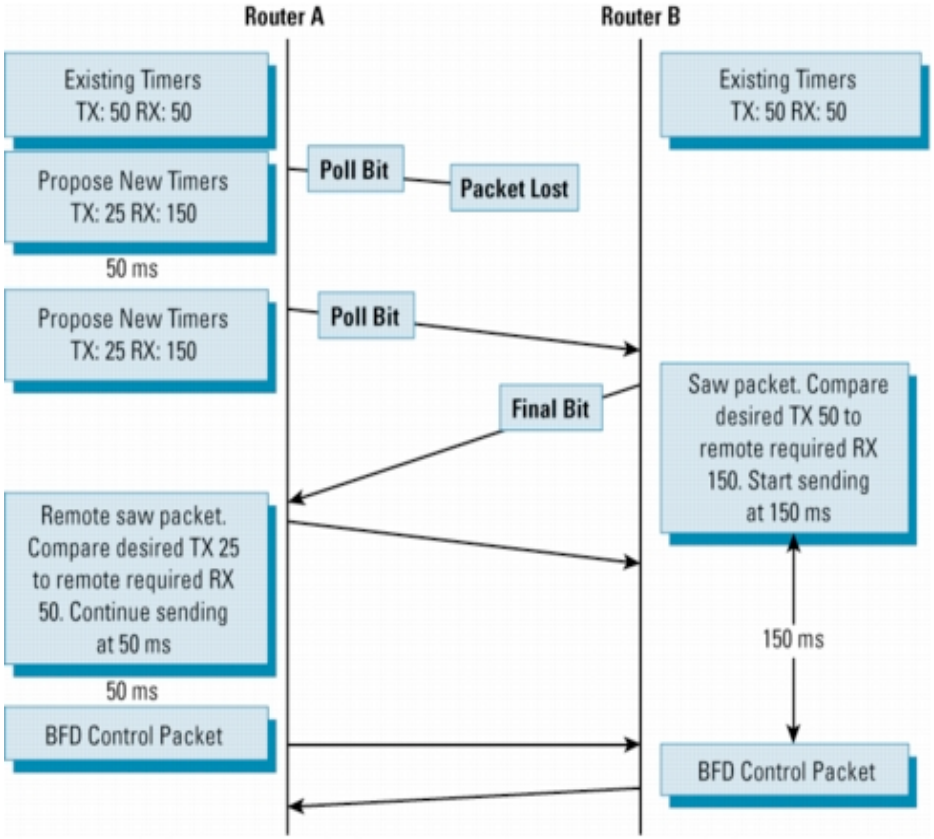
# BFD Initial Session Setup

- Remote router receives a BFD control packet
- It copy the value of the "My Discriminator" field into its own "Your Discriminator" field and set the H ("I Hear You") bit for any subsequent BFD control packets it transmits
- Session is now Established



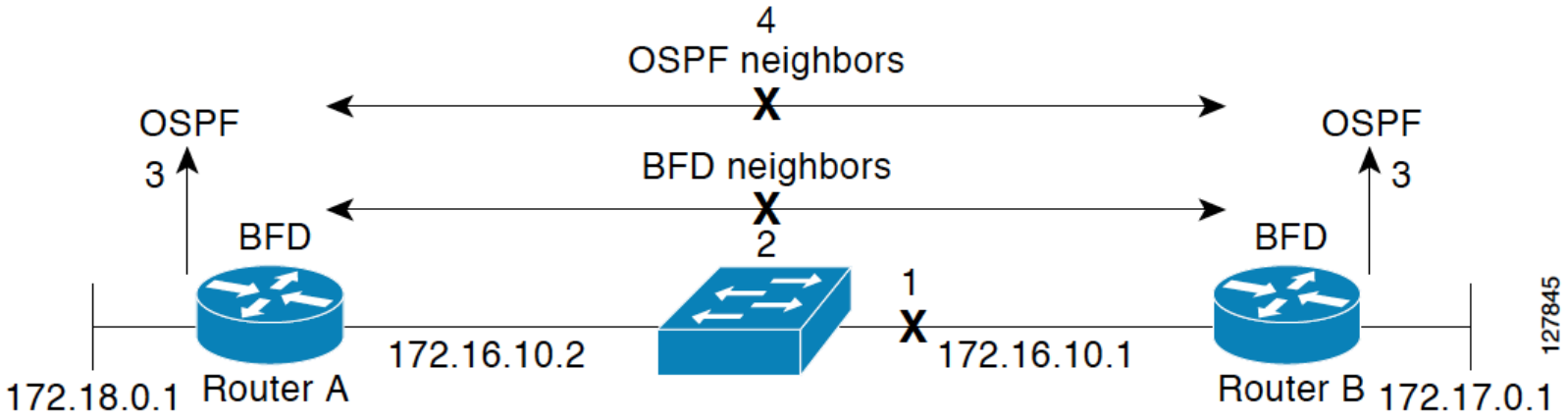
# BFD Timer Negotiation

- The device that changed its timers will set the P bit on all subsequent BFD control packets, until it receives a BFD control packet with the F bit set from the remote system
- Each system, upon receiving a BFD control packet will take the "Required Min RX Interval" and compare it to its own "Desired Min TX Interval" and take the greater (slower) of the two values and use it as the transmission rate for its BFD packets
- The slower of the two systems determines the transmission rate.



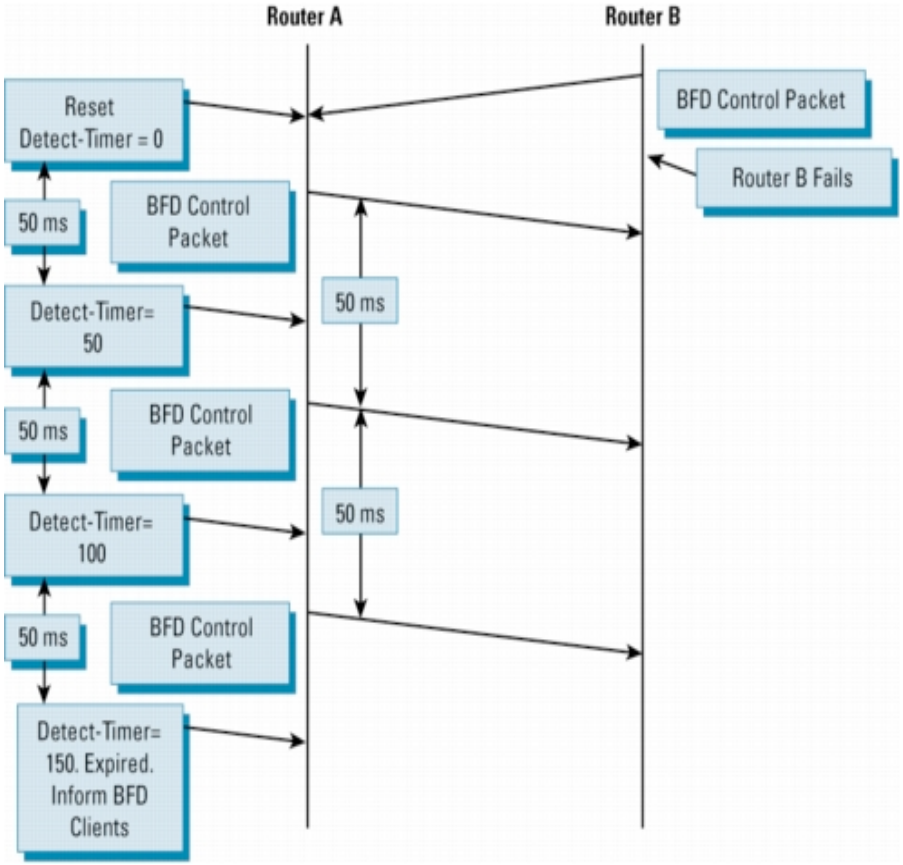
# BFD Failure Detection

- Detecting failure
  1. Network failure occurs
  2. BFD session tore down because of network failure
  3. BFD notifies OSPF process that neighbor is not reachable anymore
  4. OSPF process tears down neighbor relationship



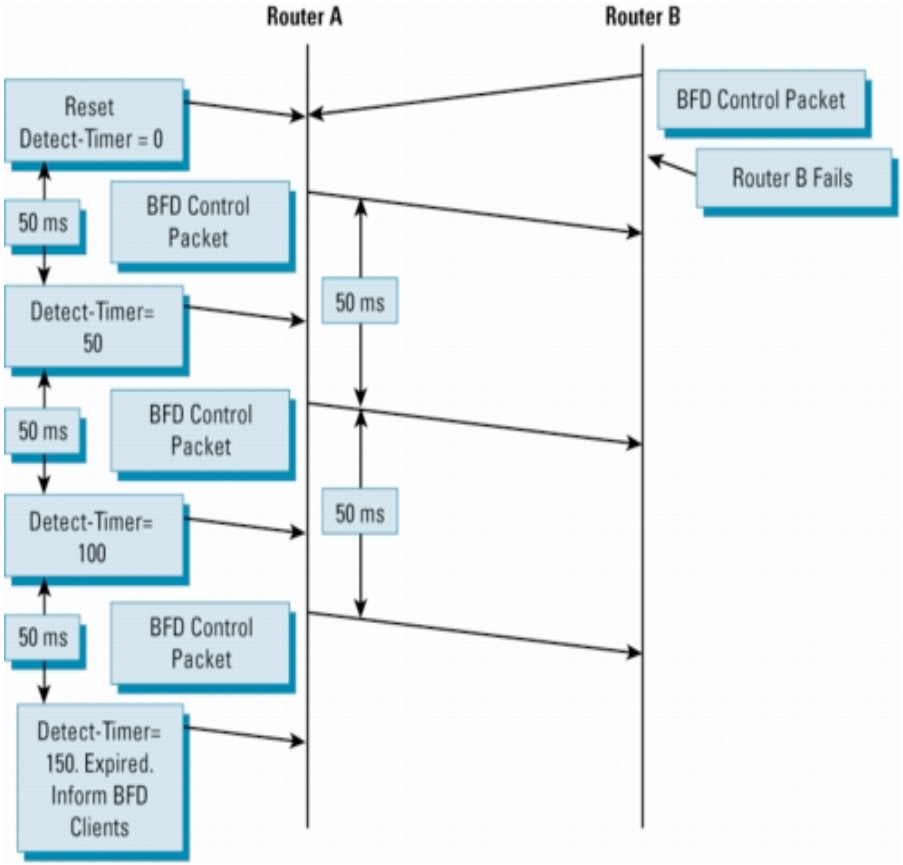
# BFD Failure Detection

- As long as each BFD peer receives a BFD control packet within the detect-timer period, the BFD session remains up and any routing protocol associated with BFD maintains its adjacencies.
- If a BFD peer does not receive a control packet within the detect interval, it informs any clients of that BFD session about the failure. It is up to the routing protocol to determine the appropriate response to that information.



# BFD Failure Detection

- In its next BFD control packet, Router A will set the diagnostic field to a value which indicates why the session was taken down. In this case, the diagnostic will be 1: Control Detection Time Expired.
- Diagnostics are useful to differentiate between real failures, versus administrative actions. For example, if the network administrator disabled BFD for this session, the diagnostic would be 7: Administratively Down.



# BFD Configuration – Nexus 7000

```
switch(config)# feature bfd
switch(config)# bfd interval <50-999>
min_rx <50-999> multiplier <1-50>
switch(config)# router ospf 1
switch(config-router)# bfd
```

# BFD Configuration – BFD for Static Route on ASR 1000

```
ip route static bfd Serial 2/0/0 10.201.201.2
ip route 10.0.0.0 255.0.0.0 Serial 2/0/0 10.201.201.2
!
interface Serial 2/0/0
 ip address 10.201.201.1 255.255.255.0
 bfd interval 500 min_rx 500 multiplier 5
```

# BFD Configuration – ASR 9000

```
RP/0/RSP0/CPU0:router(config)# router bgp 65000
RP/0/RSP0/CPU0:router(config-bgp)# bfd multiplier 2
RP/0/RSP0/CPU0:router(config-bgp)# bfd minimum-interval 20
RP/0/RSP0/CPU0:router(config-bgp)# neighbor 192.168.70.24
RP/0/RSP0/CPU0:router(config-bgp-nbr)# remote-as 2
RP/0/RSP0/CPU0:router(config-bgp-nbr)# bfd fast-detect
```

# BFD Configuration – Juniper SRX

```
[edit protocols ospf area 0.0.0.0]
user@switch# set interface ge-0/0/20.0 bfd-liveness-
detection minimum-interval 500
user@switch# set interface ge-0/0/2.0 bfd-liveness-
detection multiplier 3
user@switch# set interface ge-0/0/20.0 bfd-liveness-
detection full-neighbors-only
```

# BFD Configuration – F5 BIG-IP LTM

```
bigip (config-if)# bfd interval 100 minrx
200 multiplier
bigip (config)# bfd slow-timer 2000
```

*BGP:*

```
bigip (config-if)# neighbor 1.1.1.1
fallover bfd multihop
```

OSPF:

```
bigip (config)# bfd all-interfaces
bigip (config)# area 1 virtual-link
3.3.3.3 fallover bfd
```

# BFD Caveats – when maybe not use BFD

# General BFD caveats

- BFD can potentially generate false alarms-signaling a link failure when one does not exist. Because the timers used for BFD are so tight, a brief interval of data corruption or queue congestion could potentially cause BFD to miss enough control packets to allow the detect-timer to expire. While the transmission of BFD control packets is managed by giving them the highest possible queue priority, little can be done about prioritizing incoming BFD control packets.

# General BFD caveats

- BFD will consume some CPU resources
  - On non-distributed platforms, in-house testing has shown a minor 2% CPU increase (above baseline) when supporting one hundred concurrent BFD sessions*.
  - On distributed platforms, there is no impact on the main Route Processor CPU, except during BFD session setup and teardown. It is important to note that, because of this accelerated handling of BFD control packets, all output features are bypassed. Users cannot, for example, filter or apply Quality of Service (QoS) to transmitted BFD packets.

* based on Cisco documentation

# BFD – Different Platforms – Different features

| Dynamic State Preservation | ASR 1000 | ASR 9000 |
|---|---|---|
| Connectivity Protocols | FR, PPP, MLPPP, HDLC, 802.1Q, BFD (BGP, IS-IS, OSPF) | BFD (OSPF, BGP, IS-IS, Static) |
| Routing & IP Services | RP, HSRP, IPv6 NDP, uRPF, SNMP, GLBP, VRRP, NSR (MP-iBGP, eBGP), ISSU, GRE, | NSF (ISIS, OSPF, BGP), NSR (ISIS, OSPFv2, OSPFv3, BGP) |
| Multicast | IPv4 Multicast (IGMP), IPv6 Multicast (MLD, PIM-SSM, MLD Access group), MoFRR | NSF Multicast, BFD for PIM, MoFRR |
| MPLS Protocols | MPLS L3VPN, MPLS LDP , VRF-aware BFD,<br>Roadmap: NSR LDP, T-LDP | NSF (LDP, T-LDP, RSVP-TE)<br>NSR (LDP), BFD for MPLS FRR, VRF-aware BFD |
| Broadband | PPPoE, L2TP (LAC, LNS), DHCPv4/v6, AAA, session state (virtual templates), ISG, ANCP, LI | PPPoE (including nV) |
| Security | SSO, Stateful Inter-chassis redundancy for FW / NAT | Roadmap |
| SBC | SSO | Roadmap |

# BFD – Software Problems

- SXF is said to be terrible regarding BFD
- Some IOS versions might let you configure BFD on port-channel interfaces but it would not work (not supported feature)

# BFD – Is fast convergence required in network segment?

- What traffic we are sending? Is subsecond convergence required?
- If in case of short disruption BFD initiate failover can it cause more problems?

# Bundle links

- Current standard do not describe how to handle bundled links
- No standard --> different ideas how to do it
- Different ways of implementation results in incompatibility
- Two ways of implementation
  - Single BFD session
  - Per-link BFD session

# Bundle Links – Single BFD Session

- Single BFD session per destination
- Internal algorithm decides which line card would handle BFD
- Minimum interval and BFD mode depends on line card
- Different implementations even by same vendor:
  - Nexus 7000 – BFD Logical Mode (asynchronous with echo, one bundle link per BFD session used or round robin, depends on software)
  - ASR9000 – BFD Logical Bundles (asynchronous only, one bundle link per BFD session used, since 4.0.0 mode can be chosed)

# Bundle Links – per-link BFD Session

- BFD session per bundle member
- Master session on RP/SUP consolidates members states and communicates with clients
- Asynchronous on NX-OS, Asynchronous with Echo on IOS XR platform
- LACP required on Nexus

# BFD for Static Routes

- Let us detect if next-hop is alive
- Removes static route if next-hop if neighbor is marked down
- Must be configured on both ends

# Questions?

# Thank You