An Introduction to

Bidirectional Forwarding Detection (BFD)

NANOG 39

Aamer Akhter / aa@cisco.com

ECMD, cisco Systems

Why BFD?

Methods needed to quickly determine forwarding failure

- Not everything is POS/SONET
  - Ethernet needs a solution for failure detection
- Layer 3 Data Forwarding plane needs a check
- Checking should not be bound to single hop
- Fast Hello needed for LDP, OSPF, ISIS, PIM, RSVP, BGP etc to catch same types of issues.

- BFD is a single Layer 3 protocol for detecting forwarding failures

- Other protocol timers can now be left at defaults

What is BFD?

Bi-directional Forwarding Detection:

- Extremely lightweight hello protocol
  - IPv4, IPv6, MPLS, P2MP
- 10s of milliseconds (technically, microsecond resolution) forwarding plane failure detection mechanism.
- Single mechanism, common and standardized
  - Multiple modes: Async (echo/non-echo), Demand
- Independent of Routing Protocols
- Levels of security, to match conditions and needs
- Facilitates close alignment with hardware

IETF Status

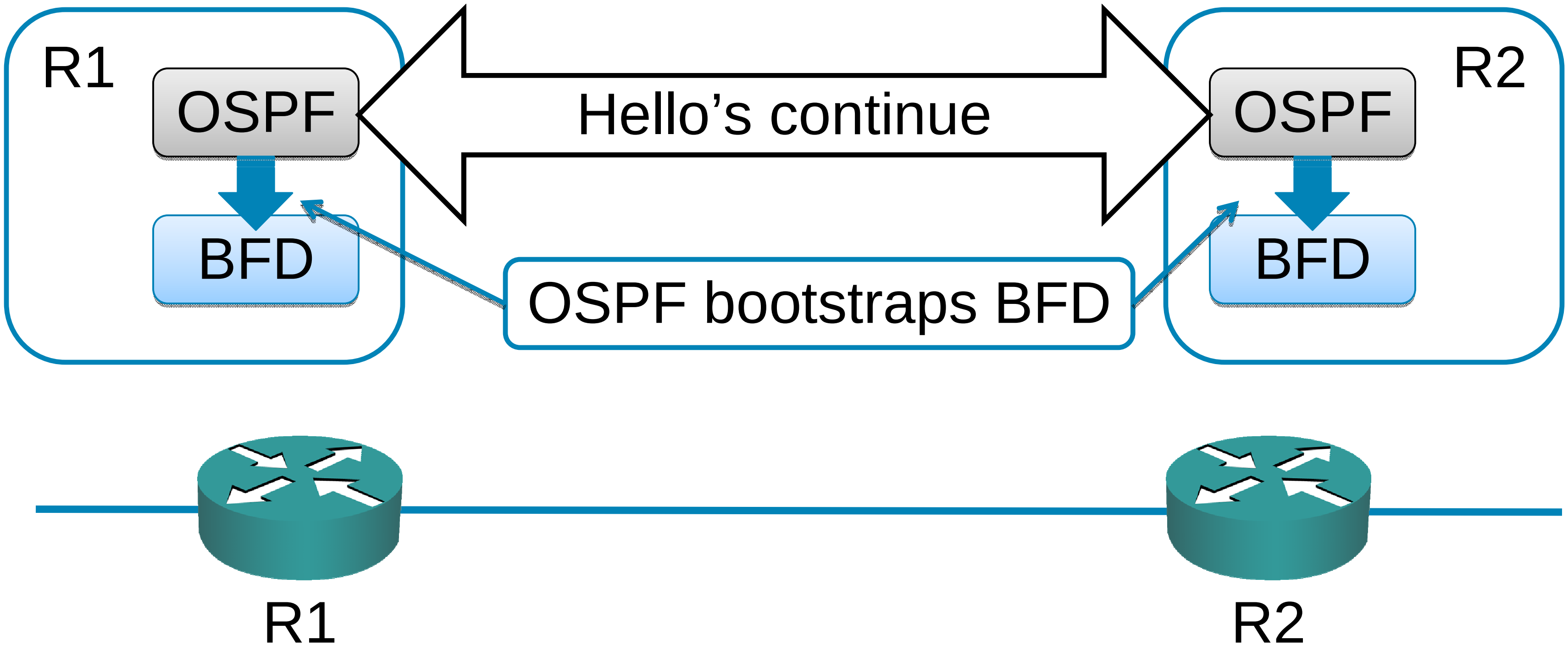
- BFD workgroup
- Base Draft (Ward and Katz) draft-ietf-bfd-base-xx
- Generic Application of BFD draft-iet-bfd-generic
- BFD for IPv4 and IPv6 (Single Hop) draft-ietf-bfd-v4v6
- BFD for Multihop Paths draft-ietf-bfd-multihop
- BFD For MPLS LSPs draft-ietf-bfd-mpls
- BFD MIB draft-ietf-bfd-mib

- Additonal BFD clients may not require standardization (eg statics client)

Basics of BFD Operation

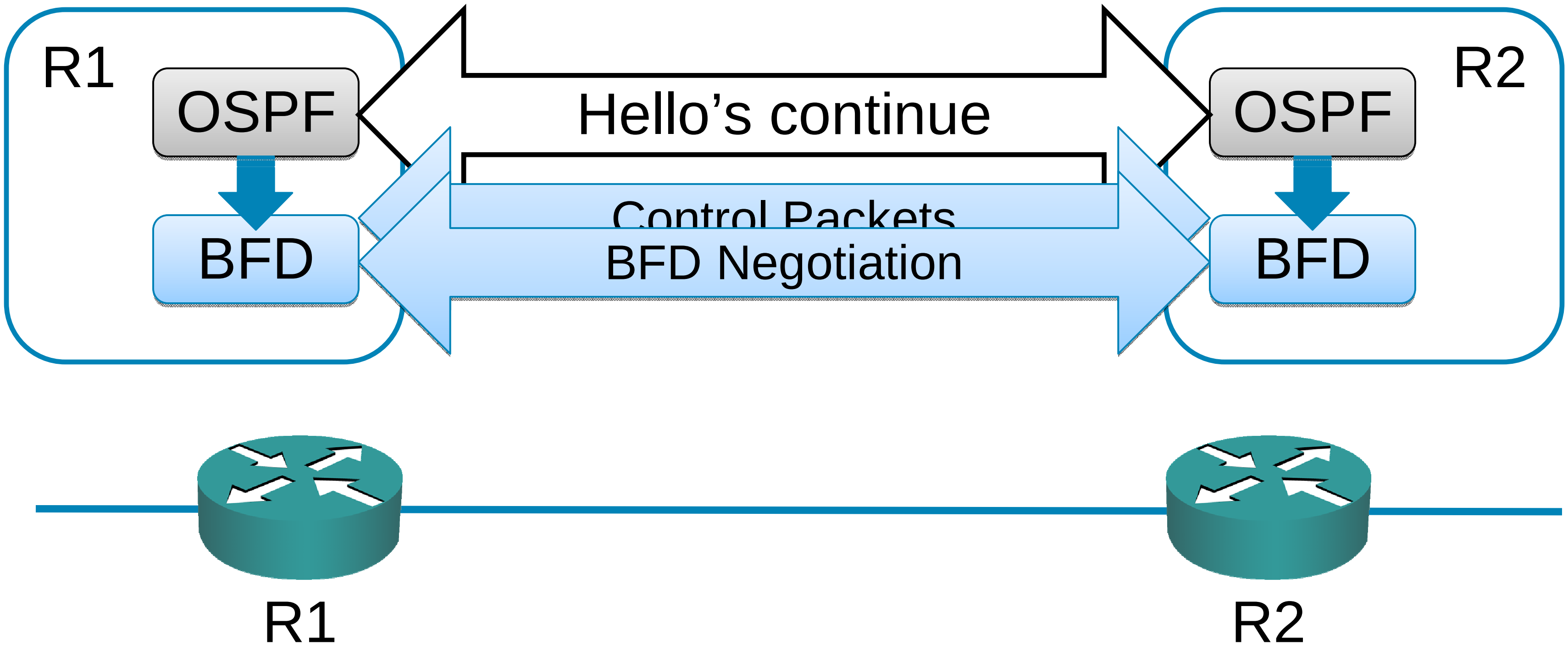
- Routing Protocol (BFD client) bootstraps BFD to create BFD session to a neighbor,
  - and to receive link status change notification.
- Receive and Transmit intervals are negotiated and configurable
- Two systems agree on method to detect failure
  - Via sending packets, watching counters etc
- In case of failure, BFD notifies BFD client
- BFD Client independently decides on action (if any)

BFD in Pictures (OSPF Example)



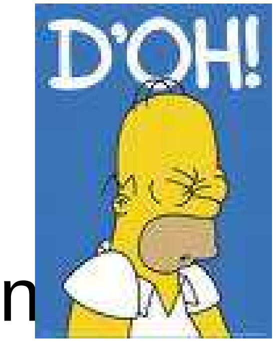
BFD in Pictures (async mode)

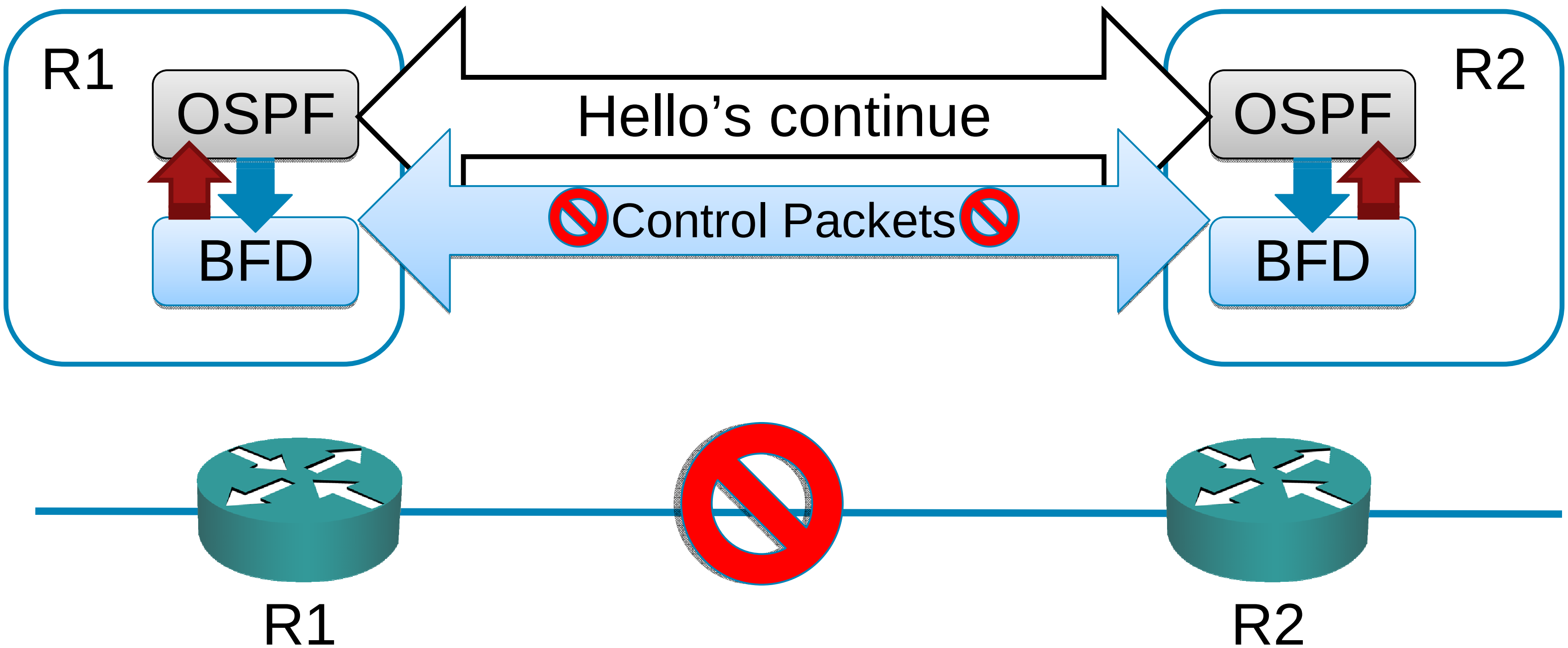
- OSPF Hello's at slow rate
- BFD Control packets maintain state and test forwarding plane



BFD in Pictures (async mode)

- BFD notifies OSPF of failure
- OSPF declares neighbor dead
  - Other protocols (ISIS, BGP) may take more granular action





BFD: More Details

Operating Modes
Control & Echo
Timers

Operating Modes

- Asynchronous mode
  - Echo Mode
  - Non-Echo Mode
- Demand Mode
  - Both systems set D bit in control packet
  - Implies that some other method is being used to check forwarding plane
    - Eg, looking at RX/TX counters on interface
  - BFD Polling mechanism will be used to verify liveliness as a secondary measure.

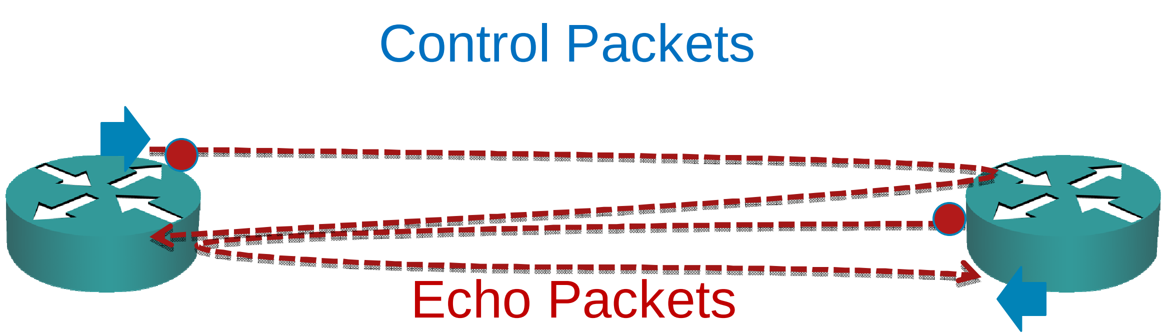
Control Packets and Echo Mode

- If echo function is not negotiated

  control packets sent at high rate to achieve Detection Time

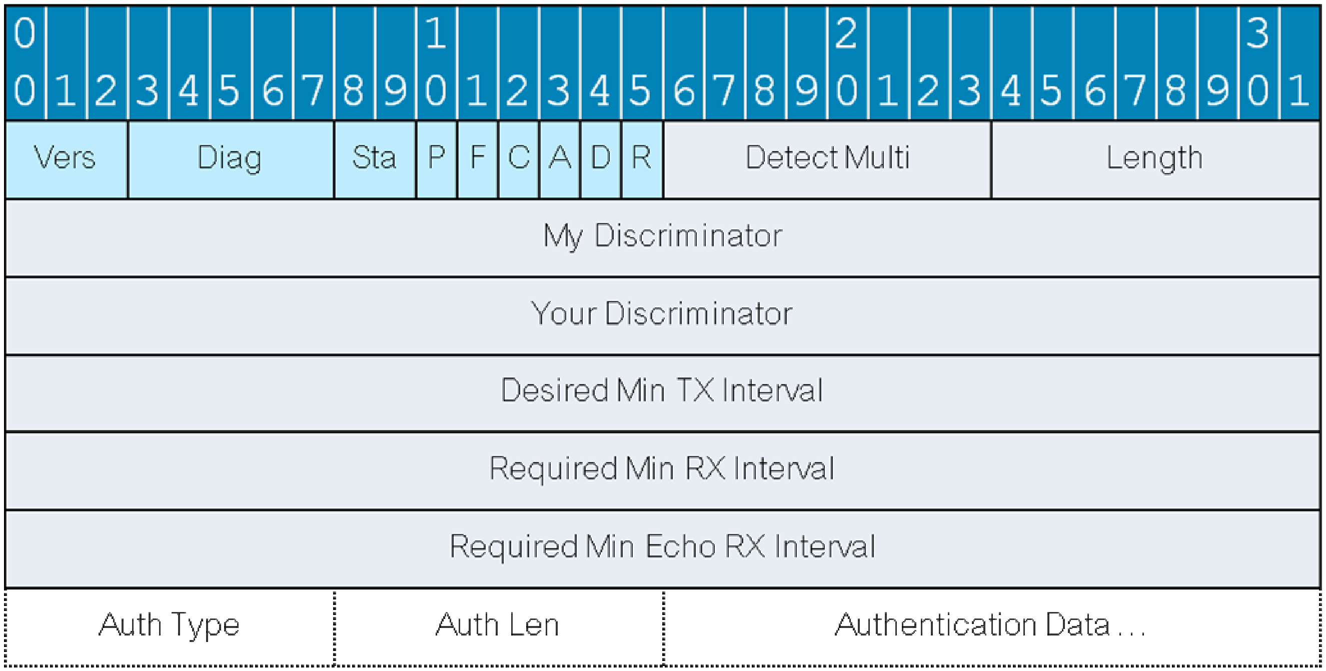
- If echo function is negotiated

  control packets sent at a slow rate (Negotiated Rate)
  self directed echo packets sent at high rate (Min Echo Rx Interval)



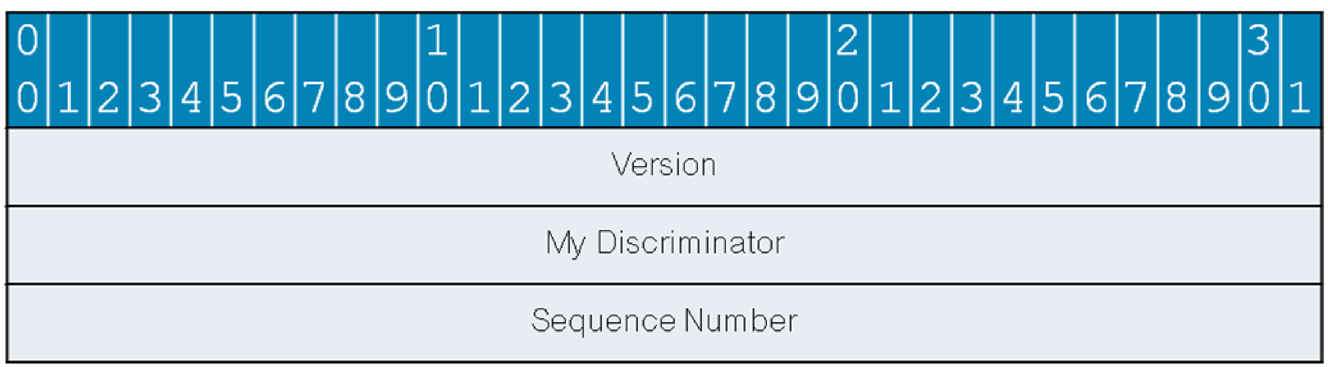
The BFD Control Packet

- Control Packets – control session state and parameters
  - Unicast Directed to BFD peer (IP address learnt via BFD client)
  - Are consumed by each BFD recipient
- Single hop: UDP sent to port 3784, source port (49152-65535)



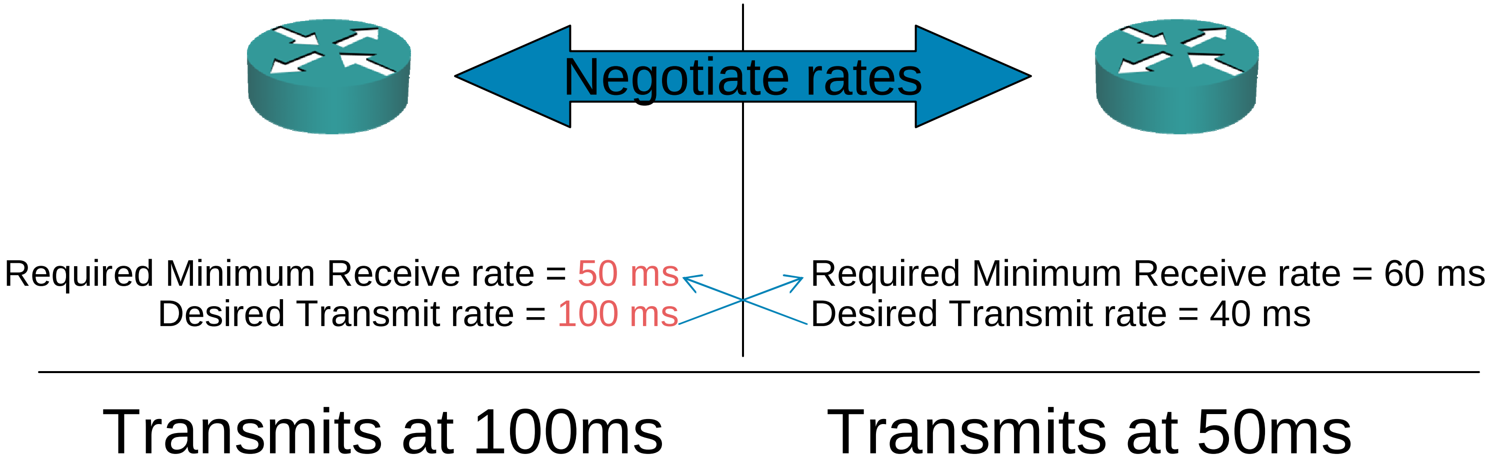
Echo Packets

- Echo Packets
  - Self directed
  - Low overhead check of fwding plane
  - Can be applied Asymmetrically
  - Format and content of packet determined by sender (implementation dependent)
    - Sent to UDP port 3785, for IPv4/v6 single hops



Timer negotiation

- Neighbors continuously negotiate their desired transmit and receive rates in terms of microseconds.
- The system reporting the slower rate determines the transmission rate.



Detection Time

- Time to detect failure ☺

  Not transmitted on the wire

- Adaptive Timers

  Less Restrictive/Tight Time Intervals

- Asynchronous mode (non-echo)

  Calculated by (remote Detect Mult) * (TX interval)

  Detect Mult is how many sequential packets can be missed before declaring down

- Asynchronous mode (echo)

  Detection Time Calculated by (local Detect Multiplier) * (local Echo RX interval)

  Loss of ‘local Detect Multiplier’ # of sequential packets causes failure

Detection Time

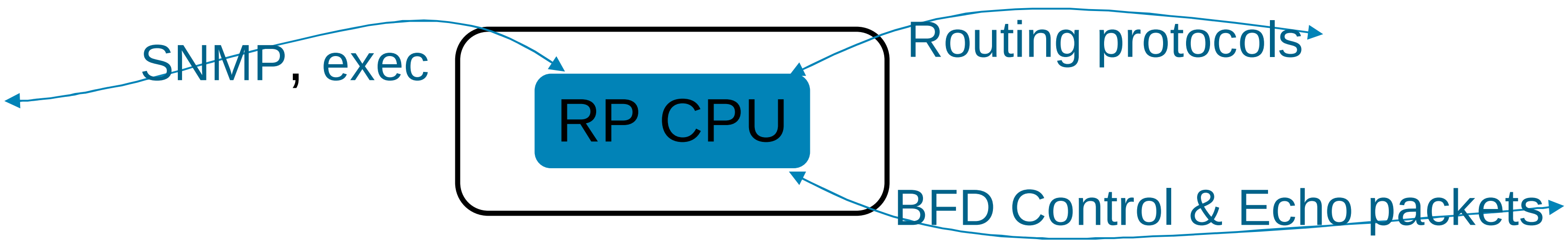
- Demand mode
  - Calculated only during Poll Sequence
  - Calculated by
    (Detect Multiplier) * (local TX interval)

BFD:
Implementation Models

Centralized
Distributed
Dedicated Hardware

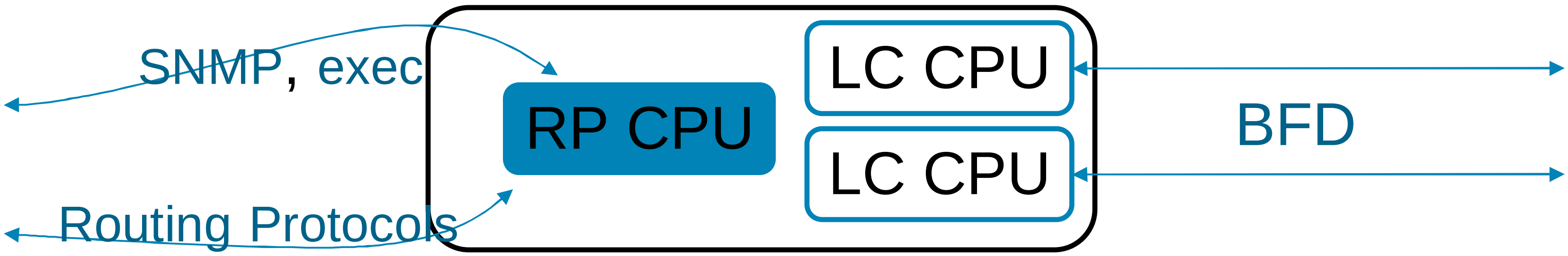
Centralized RP

- Shared CPU for all control plane (may also be shared for data plane)
  - BGP, OSPF, SNMP, exec
- Contention for CPU cycles ie:
  - BFD echo generation every 50ms
  - BGP UPDATE processing
  - IGP SPF runs
- Issue has always existed, but BFD may aggravate due to low timer values



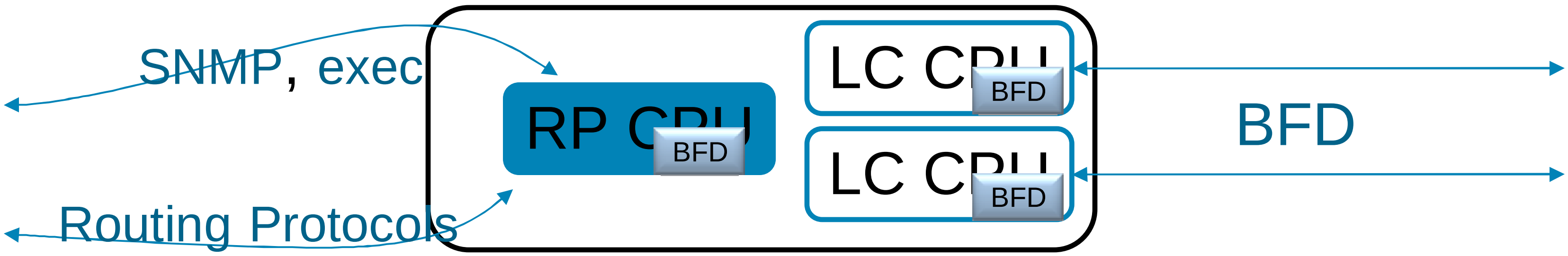
Distributed CPU

- BFD session maintenance implemented on distributed CPU (eg on line cards)
- LC CPUs generally lightly loaded
- RP can switchover (RP High Availability) w/o affecting BFD



(Semi) Dedicated Hardware

- BFD session maintenance implemented on dedicated or semi dedicated hardware (ie not GPU).
- May still be distributed for additional scalability
- Provides highest level of performance and timing precision
  - Allows more deterministic BFD performance



Control Plane Independent Bit
(Graceful Restart handling)

- BFD can inform peer via (c) bit of

  Control Plane Independence in BFD implementation

- Makes graceful restart system aware of BFD capability
- Eg: On distributed system, BFD will not be affected by rebooting control plane (ie grace-full restarts, etc)

  NSF can be checked by BFD!

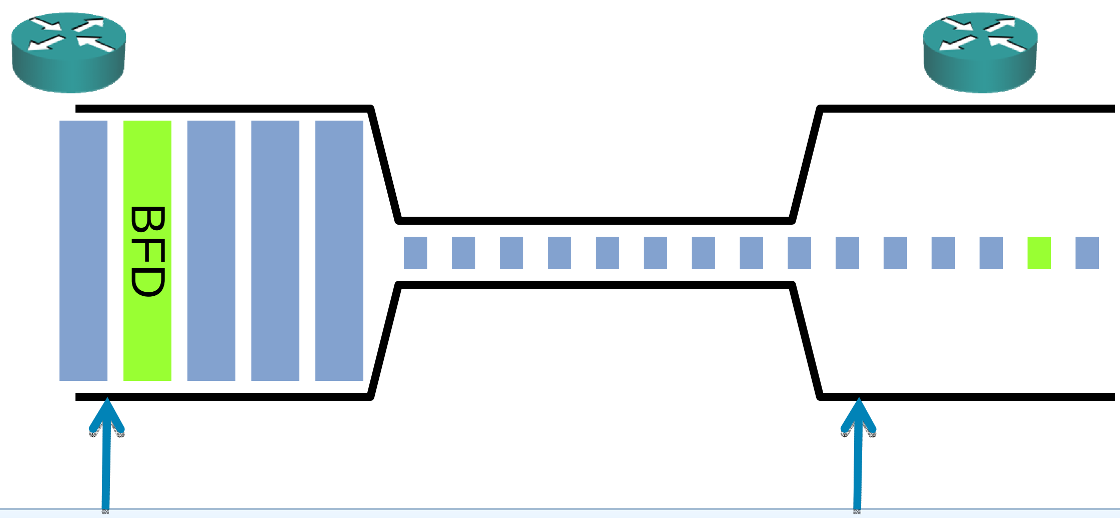
  - Eg: In case BFD is hosted by RP and will be affected by GR, (ie C bit clear):
  - BFD can temporarily ignore/disable/dilate timers during GR

Queuing (implementation)

- BFD packets can get stuck behind regular packets
- Host IP/UDP protocol stack can be bypassed.

  Requires BFD to have direct access to HW queues (no output features)

  Might be difficult on distributed architecture where BFD session is on RP & interface on LC ☺



Is internal to router and should give priority to BFD

BFD: Considerations

Queuing
Scalability vs. Performance
Security

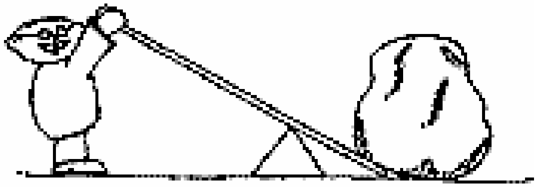
Queuing / Latency (not implementation architecture)

- Be realistic: Don’t set detection time to 50ms for transatlantic links!
- Multi-hop (and echo mode) BFD still requires queue management on transit routers
    - Marked with Precedence 6

      Prec 6 may need QoS policy configured on transit routers
    - Depending on link speed, BFD may need to be in LLQ
    - Verify that the correct data path is validated (ie DS-TE, DSCP PBR, MTR can put packets on different paths)

Scalability and Low Timer Values

- Possible to move resources around
  - BFD to LC,
    dedicated BFD CPU,
    relaxing timers etc



- Practical and finite limits on resources
- Lower BFD timer values and <u>strict detection timing</u> will decrease scalability (# of sessions) for that resource
  - Adaptive timers take care of situations w/o strict detection timing needs
  - Implementation may protect you by disallowing enabling of sessions that can't be maintained.

Scalability and Low Timer Values

- Testing under your 'real-world' conditions is essential
- Aggregate BFD pps per LC/CPU creates a composite between sessions and timer values
- Spreading BFD amongst multiple resources (line cards)
- BFD does not invalidate operator experience with low BGP KA or IGP hello times– just changes the game

False Positives / Oscillations

- Use adaptive timers / echo mode / demand mode

- Why they can happen:
- Generally, implementation issues
- Conditions change in network, but nothing really wrong

  When testing, account for stress conditions, not best conditions

  BGP updates
  IGP recalculations
  SNMP polls
  Traffic bursts

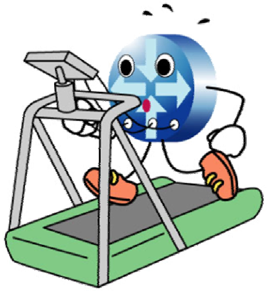
- Stress can be transient

  or related to new services that cross perf. threshold

Tuning Timers

- BFD allows timer renegotiation during session
- Adaptive Timers (all modes have this)
    - Less restrictive, can automatically adapt for slow local/remote system
    - Puts actual fault detection time into grey area
- Examples:
    - Control Mode: (slow remote) avg of last few rxvd pkts is used
    - Echo Mode: (slow self)
        - RX Count is done on packets actually transmitted, not what *should* have been transmitted
        - Detection Time = loosing ‘detect multi’ # of packets (regardless of when they were sent out)

Tuning Timers



- Need to monitor state of router
  - SNMP traps on consistently slow RX/TX, etc
- Are BFD packets being sent/rcvd at the configured value?
  - Increase or decrease interval's accordingly (if needed)

Security

- TTL checking

  single hop

    255 on sending, check for 255 on receipt

- Authentication (more work for router)

  Multi-hop applicability

  Key ID’s allow for rollover

  Simple Password

  MD5

  SHA1
  Meticulous Keyed

BFD Summary

- Solves a real issue with fast forwarding plane checks
- Extremely lightweight hello protocol
  - IPv4, IPv6, MPLS, P2MP
- 10s of milliseconds (technically, microsecond resolution) forwarding plane failure detection mechanism.
- Single mechanism, common and standardized
  - Multiple modes: Async (echo/non-echo), Demand
- Independent of Routing Protocols
- Facilitates close alignment with hardware

Q and A