S-BFD

IETF 88, Vancouver, Canada

Nobo Akiya
David Ward
Carlos Pignataro
Nagendra Kumar
Manav Bhatia
Mallik Mudigonda
Santosh P K
Tarek Saad
Siva Sivabalan
Aswatnarayan Raghuram
Glenward Hayden

Why Another Flavor of BFD?

- Existing BFD is excellent for bi-directional reachability validation scenarios
- S-BFD provides improved control, flexibility and simplified operations to initiator for even wider range of scenarios and use-cases
- Why?
  - Faster reachability verifications
  - Reduction of false failures
  - Built-in fault isolation
  - Better fits those difficult with existing BFD: anycast, centralized controller initiation, etc

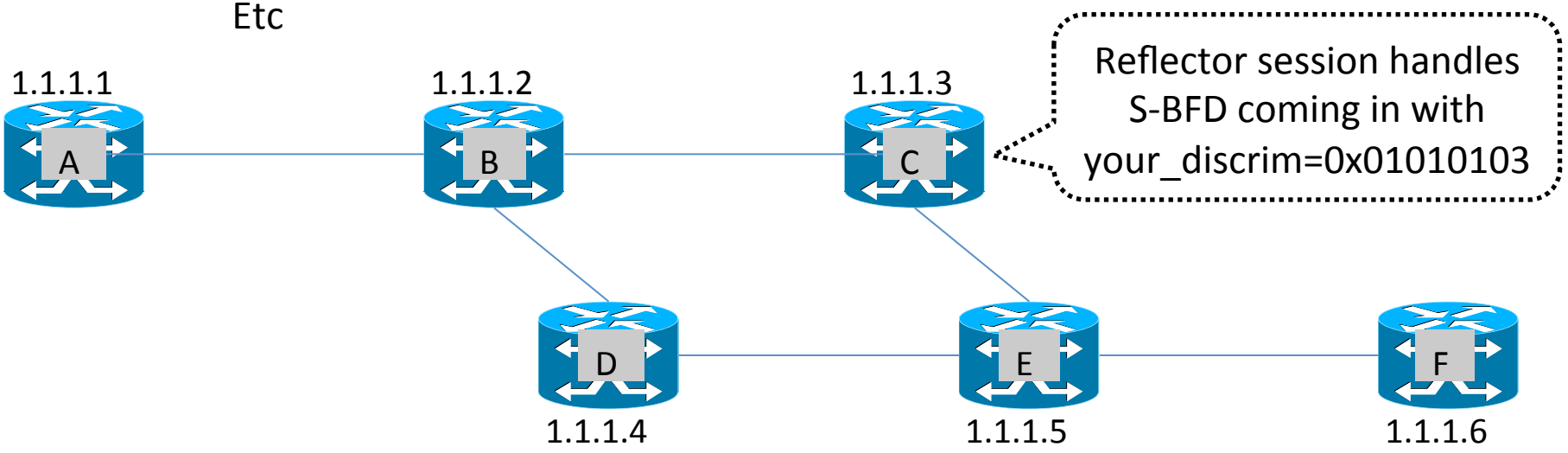
How S-BFD Works? [1]

- Pre-create reflector sessions in the network
  - Allocate discrim for local network identifier
  - Create reflector session to listen for S-BFD packets coming in with your_discrim = allocated discrim

    A allocates 0x01010101 discrim for 1.1.1.1 IPv4 address

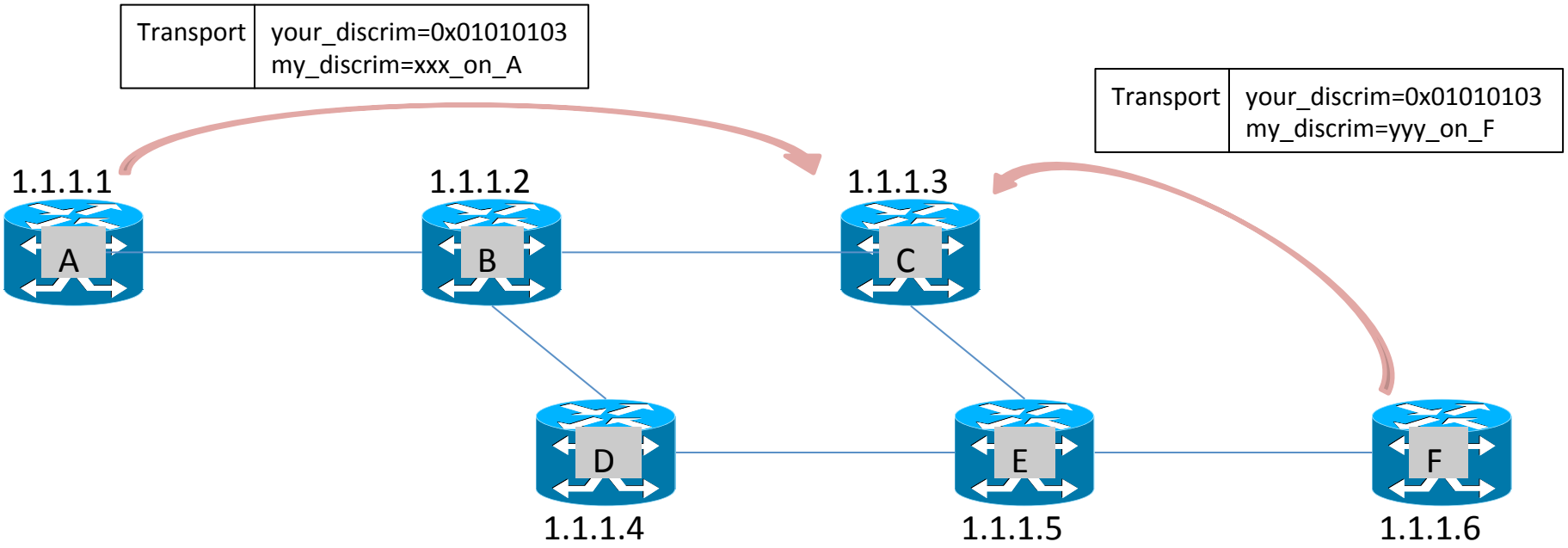
    B allocates 0x01010102 discrim for 1.1.1.2 IPv4 address

    Etc



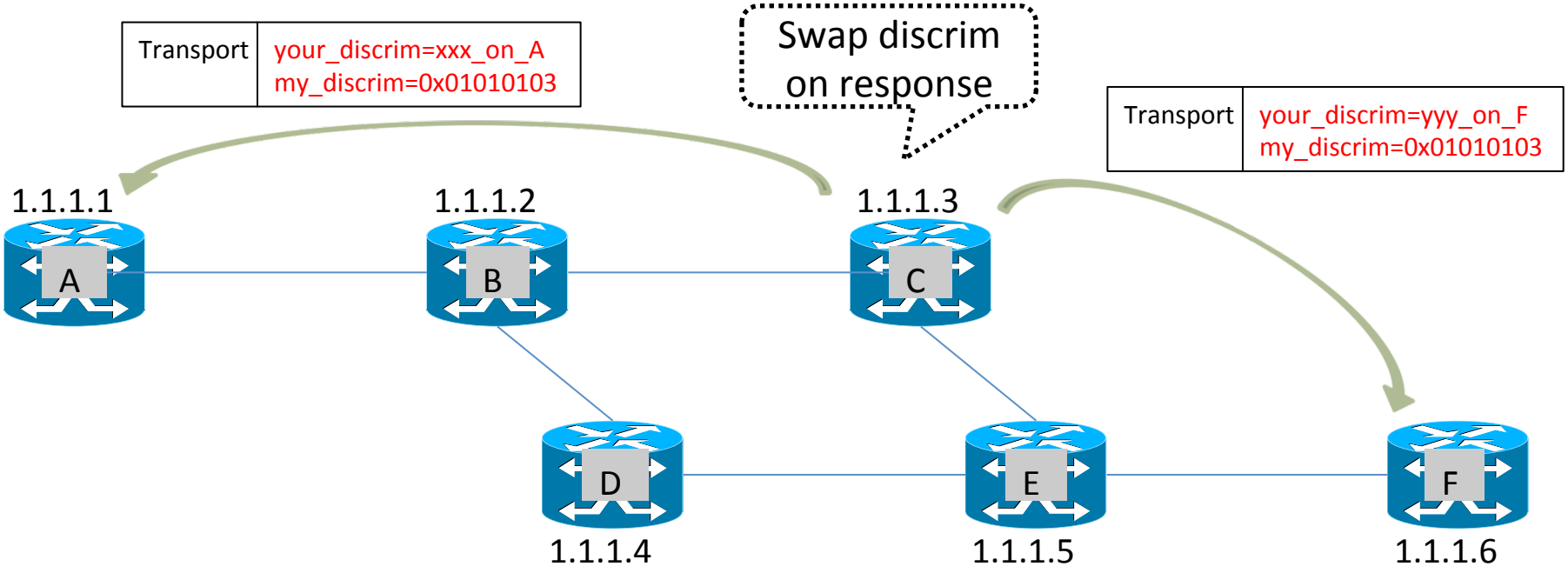
How S-BFD Works? [2]

- Initiator to send S-BFD packet to reflector session
  - Any transport
  - your_discrim=<discrim of intended target>
  - my_discrim=<locally allocated for this initiator instance>



How S-BFD Works? [3]

- Reflector session to send response S-BFD packets
  - Only handles your_discrim for me, otherwise drop
  - Send response S-BFD packets back to initiator
  - Single reflector session can handle multiple initiators



Yes …

- S-BFD is like one-sided echo
  - But … works for IP multihop, “loop” only on intended target and preserves minimal communication between end points
- S-BFD is like demand mode
  - But … no per session state on egress, no bootstrapping, and allows one-to-many (many initiators to one reflector session)
- S-BFD is like ping/traceroute
  - But … comes w/ great performance and scalability of BFD as result of still using fixed BFD header, and supports multiple transports

S-BFD Alert Discriminator

- Same discriminator allocated as reflection point in multiple network nodes, called Alert Discrim
- Your_discr=<alert discrim> can solicit response from those network nodes
- S-BFD Path Tracing Example
  - Multihop S-BFD detects failure …
  - Send S-BFD packets with your_discr=<alert discrim>, with incrementing TTL
  - Record source IP address from received responses.
  - Using anything else may traverse ECMP differently!
- (Alert Discrim + Diag) can indicate various hints

S-BFD Drafts

- draft-akiya-bfd-seamless-base-02
- draft-akiya-bfd-seamless-ip-00
- draft-akiya-bfd-seamless-sr-00
- draft-akiya-bfd-seamless-alert-discrim-00

Major Changes in Base Versions

- -00 -> -01

  `In addition, reflector BFD session SHOULD transmit response BFD control packet on the same interface on which it received the packet from initiator.`

- -01-> -02
  - New single UDP destination port for S-BFD
    - Separate discriminator pool MAY be implemented
  - Initiator state machine (Down/Up/AdminDown)

Next Steps

- Add more contents for security aspect in base
  - Spoofed packets can cause loops, D bit to fix?
  - Clarifications on authentications on S-BFD
- Polish up IP/SR/AlertDiscrim documents
- Would like WG review
- Request for WG adoption [near future]

Thank you!

Questions/Comments?

Backup Slide

(in case we need to discuss with topology)