Introduction to Seamless BFD

Nagendra Kumar Nainar

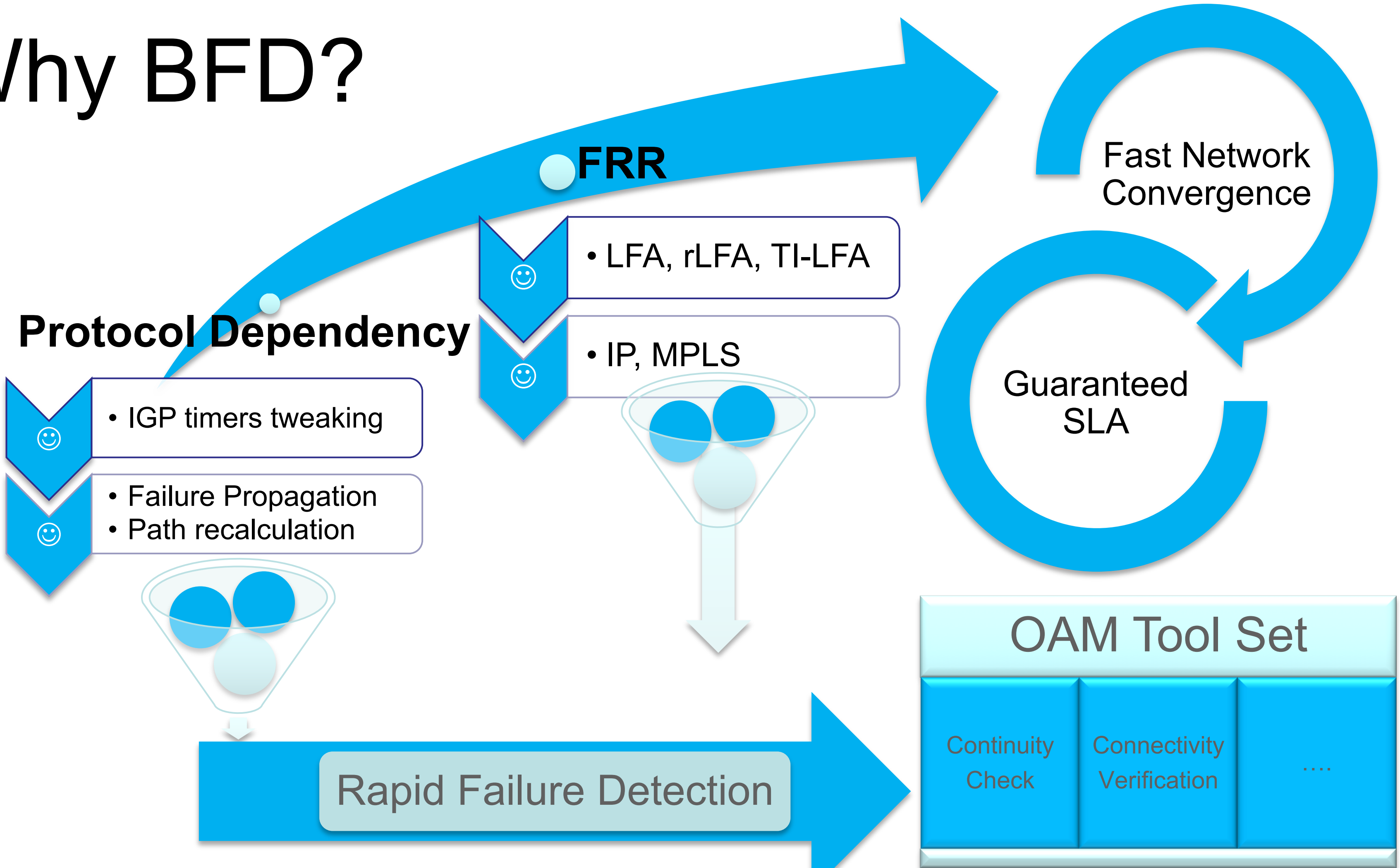
Cisco Systems

naikumar@cisco.com

Acknowledgement

- Carlos Pignataro
- Reshad Rahman

Why BFD?



Protocol Dependency

☺
- IGP timers tweaking

☺
- Failure Propagation
- Path recalculation

FRR

☺
- LFA, rLFA, TI-LFA

☺
- IP, MPLS

Fast Network Convergence

Guaranteed SLA

Rapid Failure Detection

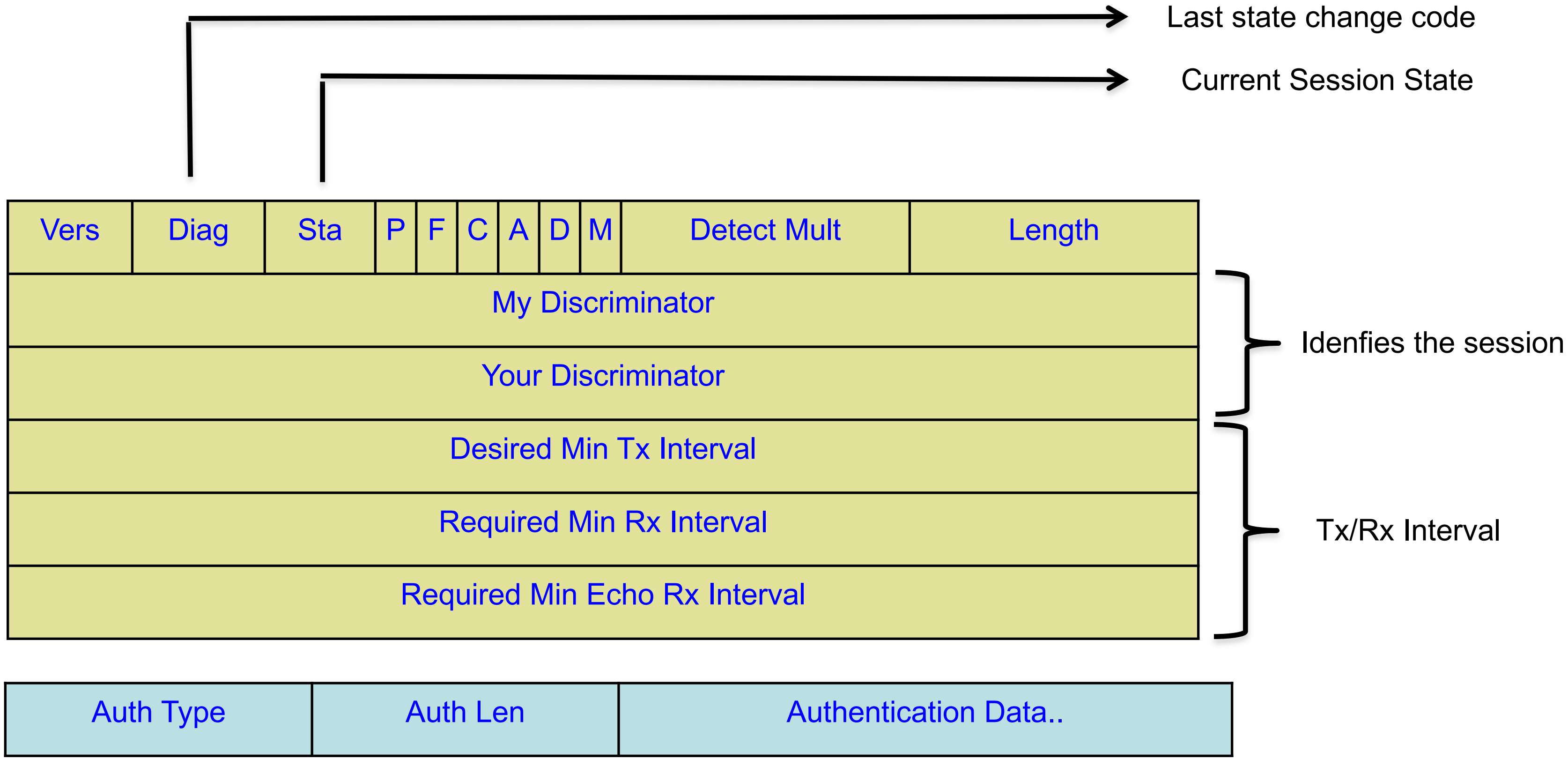
OAM Tool Set

Continuity Check	Connectivity Verification	….

What is BFD?

- BFD stands for **B**idirectional **F**orwarding **D**etection.
  - “BFD can provide failure detection on any kind of path between systems, including direct physical links, virtual circuits, tunnels, MPLS Label Switched Paths (LSPs), multihop routed paths, and unidirectional links (so long as there is some return path, of course).”
- Failure Detection protocol enabled within any IP Path and Clients (like routing protocol or interface instance) can subscribe to its update.
- Fast and lightweight generic failure detection mechanism.
  - Timer negotiations.
  - Sub-second failure detection.
  - Fixed sized control header.
  - Diagnostic relay.
  - Media independent (Ethernet, POS, Serial, etc).
  - Runs over UDP, data protocol independent (IPv4, IPv6, LSP).
  - Application independent (IGP/Tunnel liveliness, FRR trigger, etc).

BFD Overview – Header



Last state change code

Current Session State

Vers | Diag | Sta | P | F | C | A | D | M | Detect Mult | Length

My Discriminator

Your Discriminator

Desired Min Tx Interval

Required Min Rx Interval

Required Min Echo Rx Interval

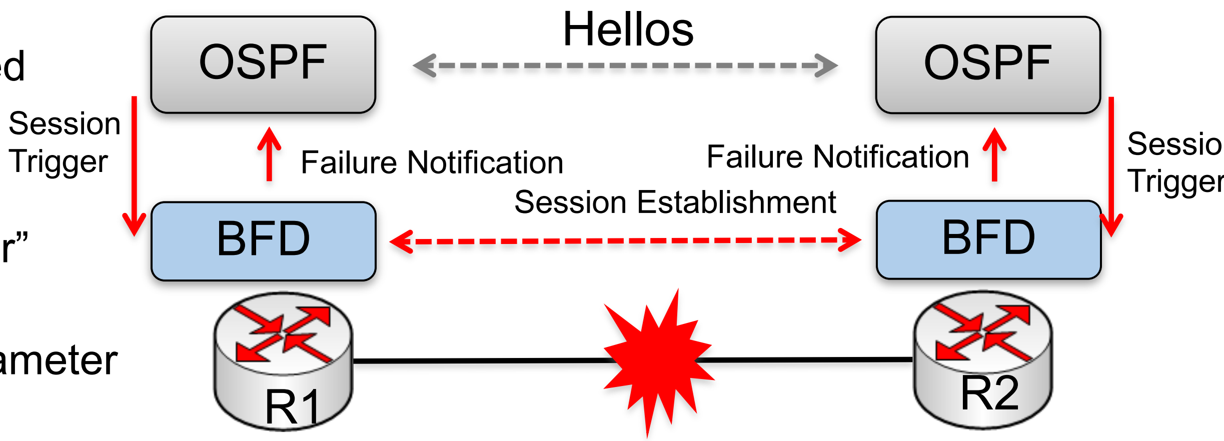
Idenfies the session

Tx/Rx Interval

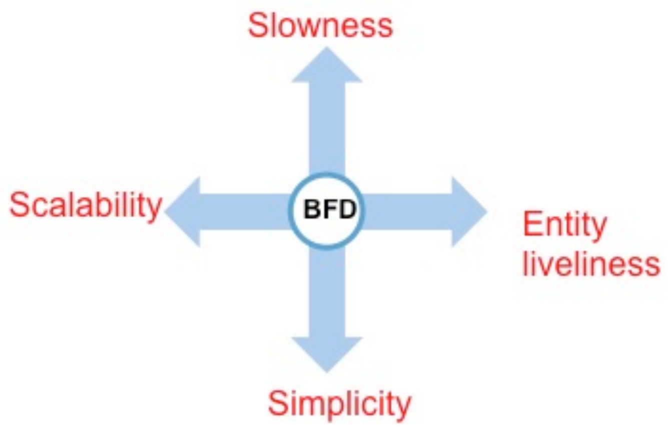
Auth Type | Auth Len | Authentication Data..

BFD Overview – Basic Operation

- Session Trigger
  - No Discovery mechanism
  - Client Application triggered (IGP/MPLS-TE)
- Session Establishment
  - Per session “Discriminator” assigned
  - 3 way handshake for parameter negotiation
  - Receive/Transmit Timer Negotiation
- Failure Notification
  - BFD notifies the client on failure



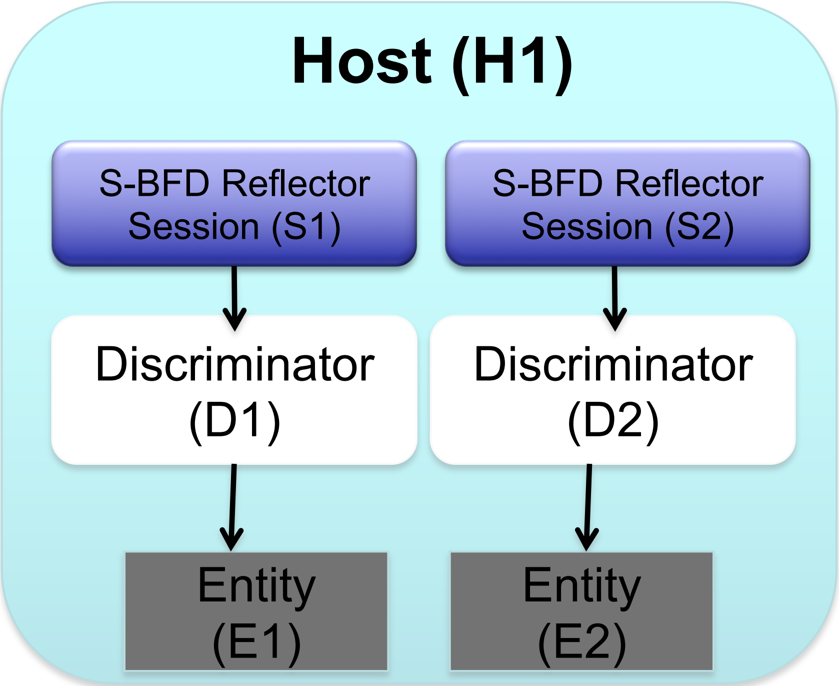
Challenges



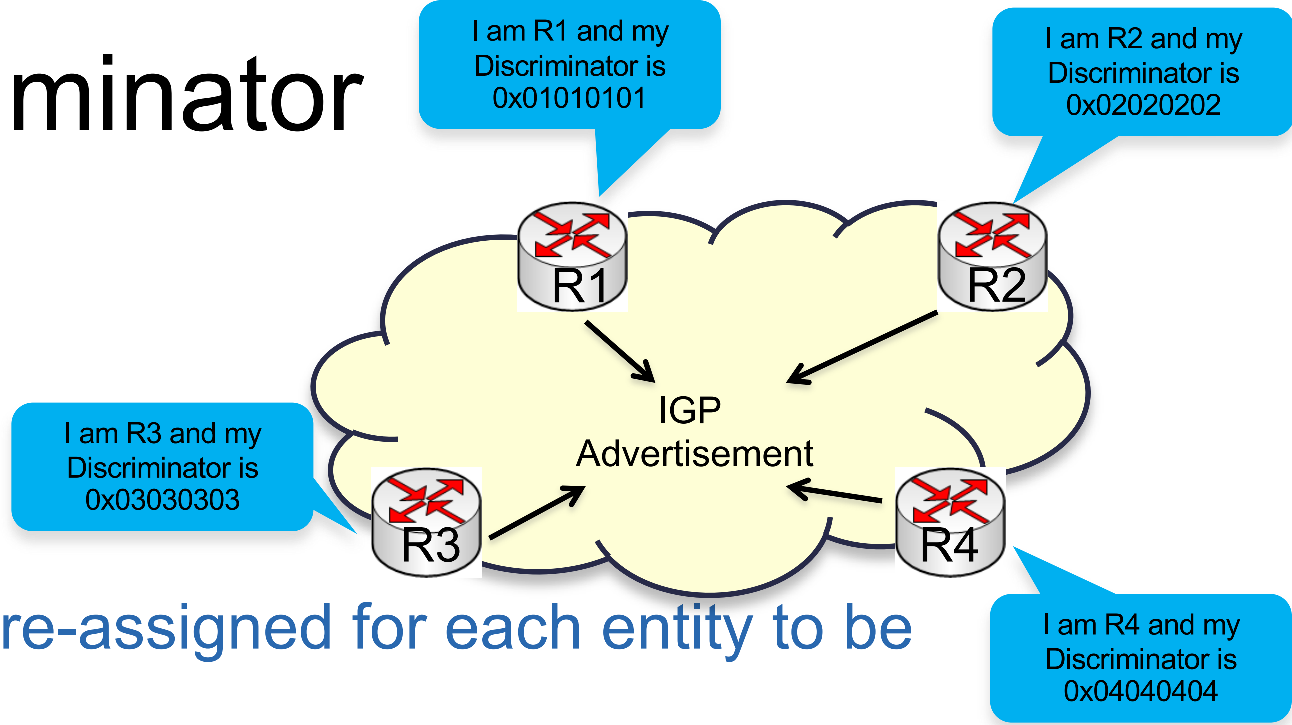
- Stateful
  - ❑ State entries on both ends
- Scalability
  - ❑ Inefficient Resource usage
- Slow Handshake
  - ❑ 3-way Discriminator handshake
  - ❑ Lacks rapidness in initial path verification
- Technology Adherence
  - ❑ SDN/nFV etc.
  - ❑ "Continuity Check" (Path) to "Liveliness check" (Entity)

S-BFD Overview

- RFC 7880
  - Protocol definition.
  - Defines 2 “main” components
- S-BFD Discriminator
  - Unique value pre-assigned to each monitoring entity
  - Propagated to all nodes
- S-BFD Reflector Session
  - Responds if Your Discriminator (YD) matches local S-BFD Discriminator

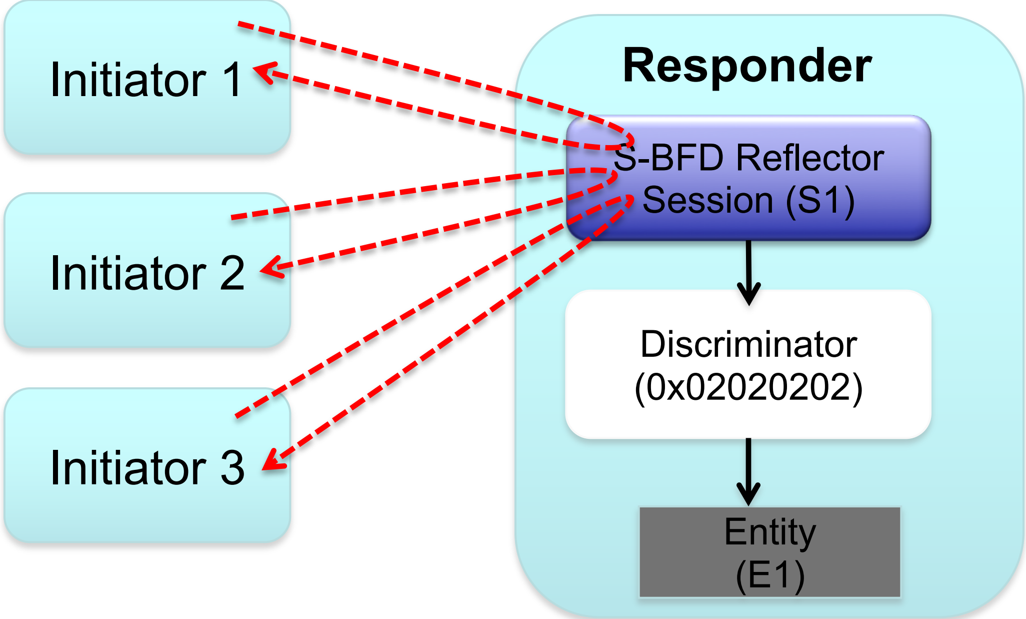


S-BFD Discriminator


I am R1 and my Discriminator is 0x01010101
I am R2 and my Discriminator is 0x02020202
R1
R2
IGP Advertisement
I am R3 and my Discriminator is 0x03030303
R3
R4
I am R4 and my Discriminator is 0x04040404

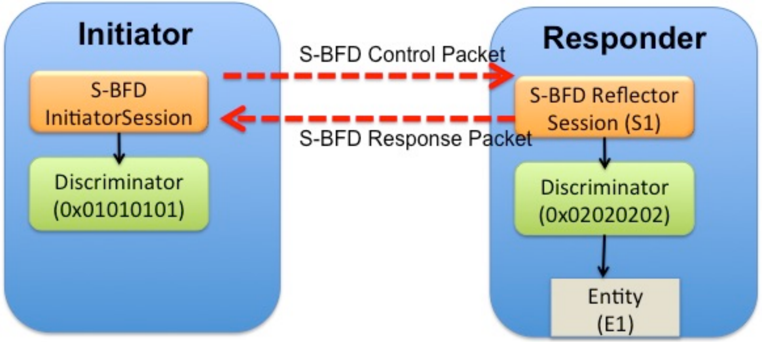
- A Unique value pre-assigned for each entity to be monitored.
  - Entity can be a node, resource, virtual host etc.
- Pre-assigned S-BFD Discriminator value is advertised to all nodes within the domain.
  - OSPF (RFC7884), ISIS (RFC7883) , L2TP (RFC7885) protocol extensions available.

S-BFD Reflector Session



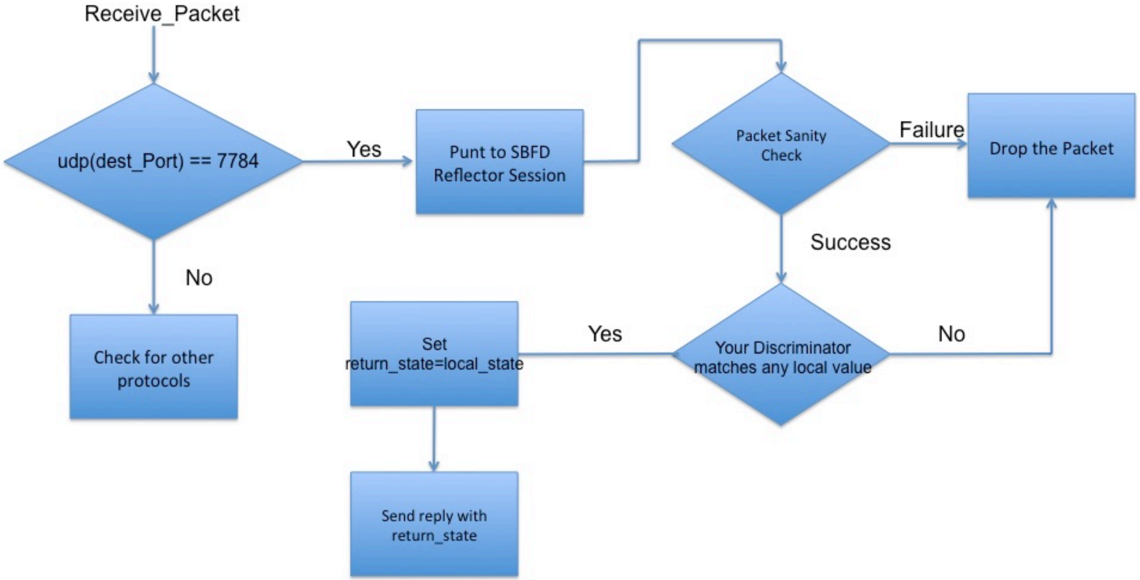
- Each node hosting the entity to be monitored creates a reflector session.
  - One session for each Discriminator.
- Responds if “Your Discriminator” matches local value.
  - No per session state entries.

S-BFD Initiator behavior



- Initiator instructed to monitor any resource.
  - Manual instruction, Protocol instruction etc
- Queries Discriminator value from local IGP database.
  - Eliminates the need for 3-way hand shake
- Generate S-BFD Control packet as below:
  - “My Discriminator” as any random value
  - “Your Discriminator” as “02020202” - the value assigned and advertised by Responder for Entity E1.
  - “State” to a value describing the local state.
  - Set “Demand” bit in control packet.

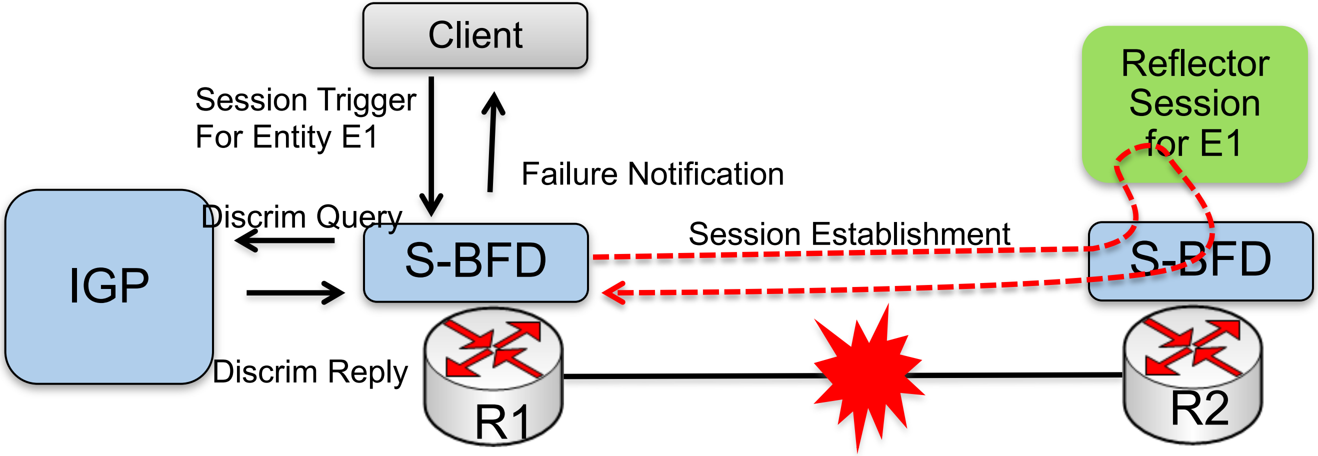
S-BFD Responder behavior



Receive_Packet
udp(dest_Port) == 7784
Yes
Punt to SBFD Reflector Session
Packet Sanity Check
Failure
Drop the Packet
No
Check for other protocols
Success
Set return_state=local_state
Yes
Your Discriminator matches any local value
No
Send reply with return_state

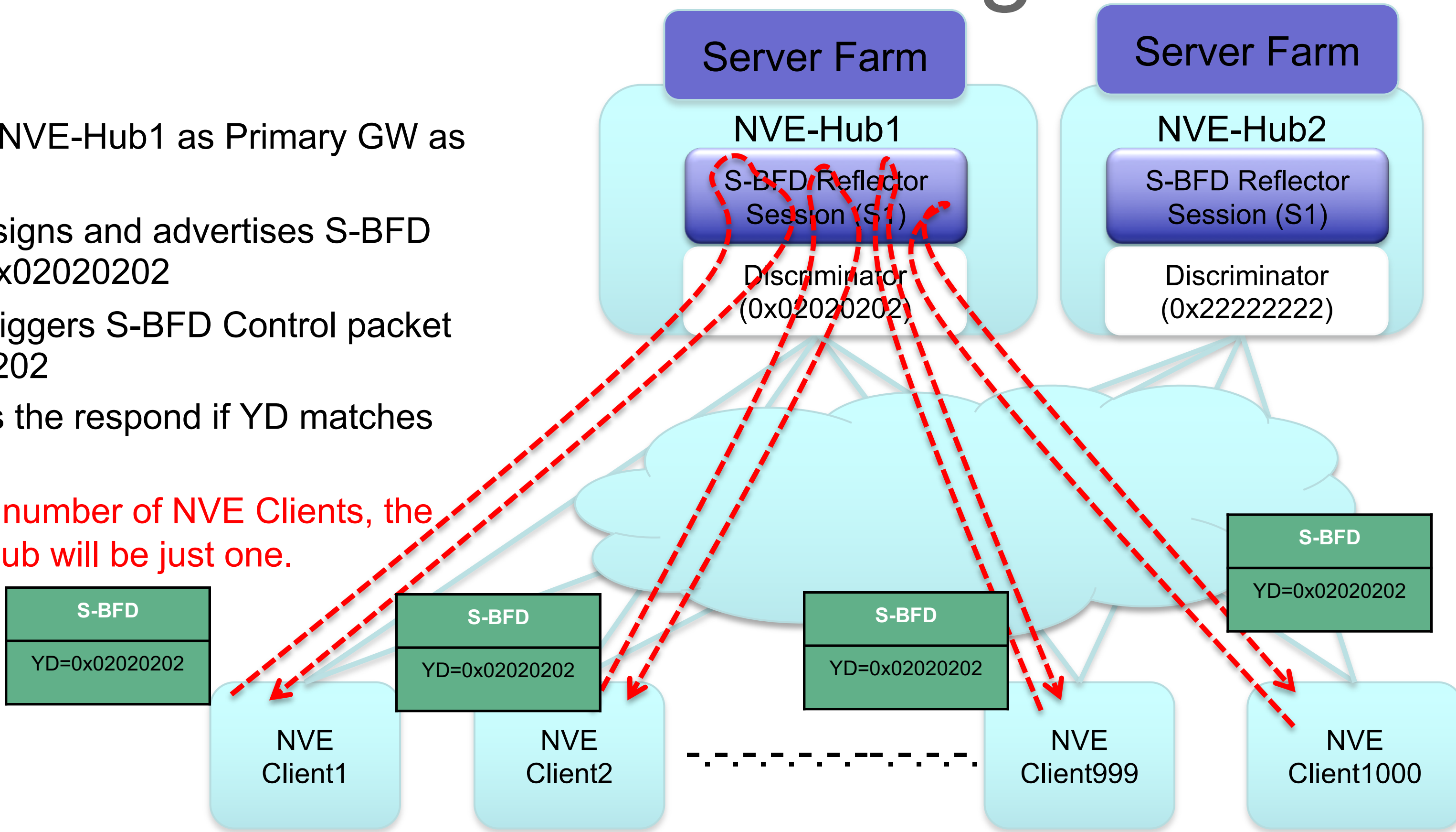
- Receiver replies if "Your Discriminator" matches local value.
- Replies with the status of the entity (UP/Down)
- No "per-session" state entries created.

S-BFD Overview – Basic Operation

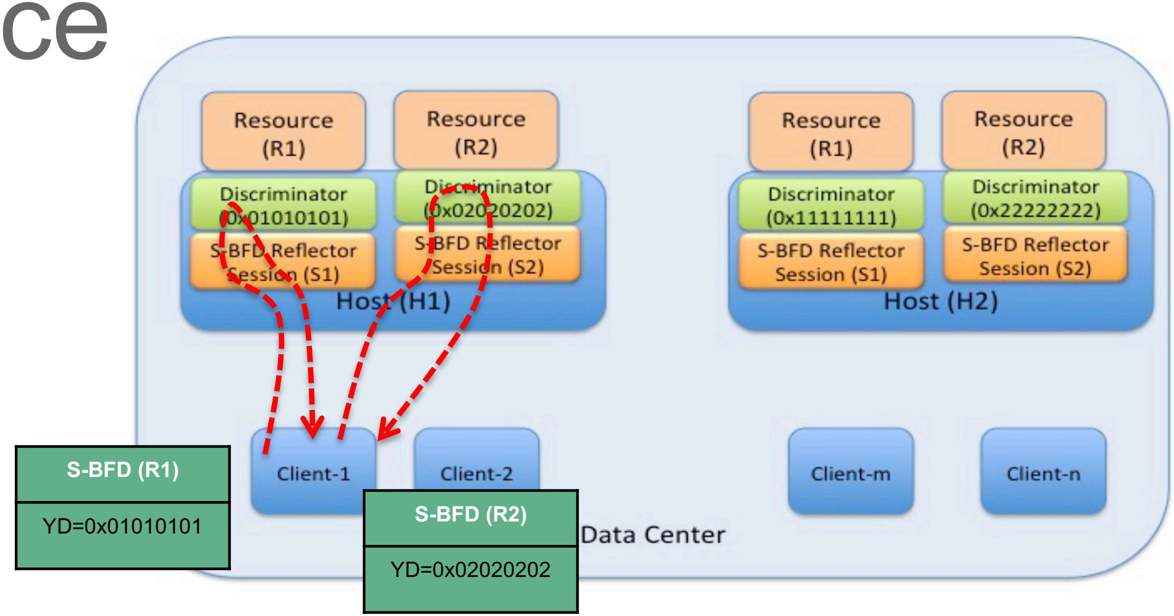


Uni-directional Path Monitoring

- NVE Clients uses NVE-Hub1 as Primary GW as long as it is up.
- NVE-Hub1 pre-assigns and advertises S-BFD Discriminator as 0x02020202
- Each NVE client triggers S-BFD Control packet with YD=0x02020202
- NVE-Hub1 reflects the respond if YD matches local value.
- Irrespective of the number of NVE Clients, the session on NVE-Hub will be just one.



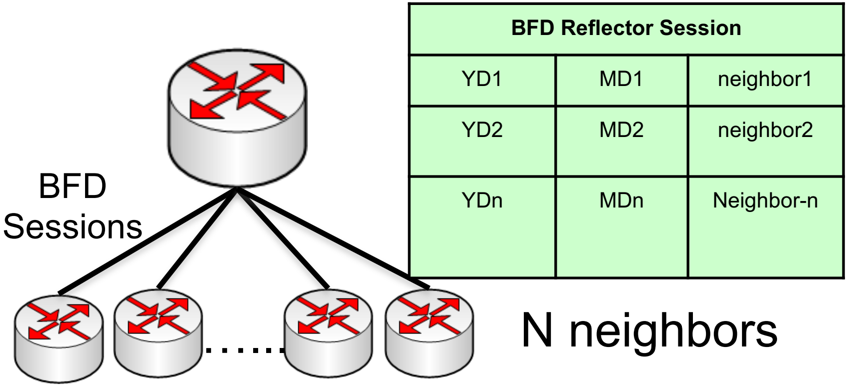
Virtual Resource Monitoring


Resource (R1)
Resource (R2)
Discriminator (0x01010101)
Discriminator (0x02020202)
S-BFD Reflector Session (S1)
S-BFD Reflector Session (S2)
Host (H1)
Resource (R1)
Resource (R2)
Discriminator (0x11111111)
Discriminator (0x22222222)
S-BFD Reflector Session (S1)
S-BFD Reflector Session (S2)
Host (H2)
S-BFD (R1)
YD=0x01010101
Client-1
Client-2
S-BFD (R2)
YD=0x02020202
Data Center
Client-m
Client-n

- In Virtual Datacenter environment, Clients monitor resource within physical host.
- A unique Discriminator assigned for each such resource.
- Clients uses the respective Discriminator and send the control packet to physical host.

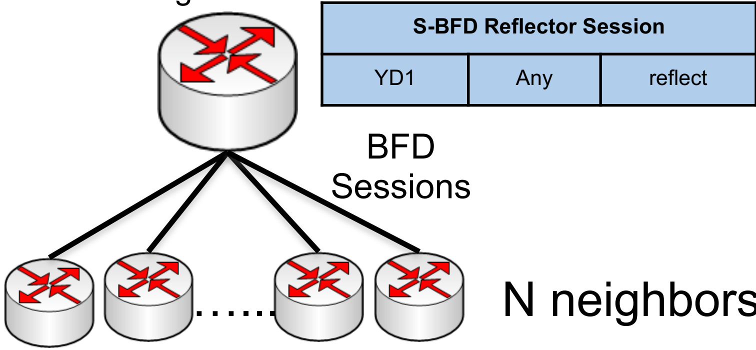
Benefits of S-BFD – State Efficiency & Scalability

Number of session state is directly proportional to neighbors



BFD Reflector Session		
YD1	MD1	neighbor1
YD2	MD2	neighbor2
YDn	MDn	Neighbor-n

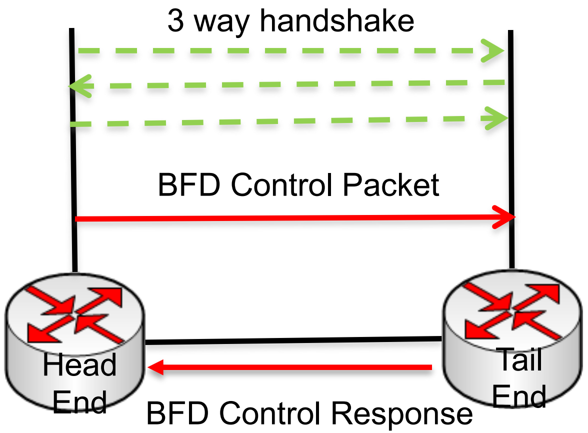
One reflector session maintained irrespective of the number of neighbors

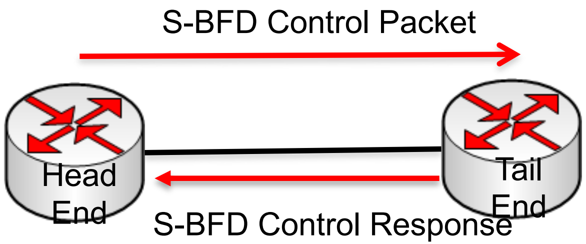


S-BFD Reflector Session		
YD1	Any	reflect

- Drastically reduces the number of state entries

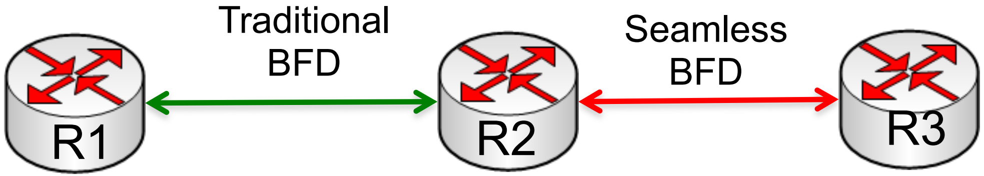
Benefits of S-BFD - Rapidness





- Eliminates 3 way handshake
- No Discriminator negotiation

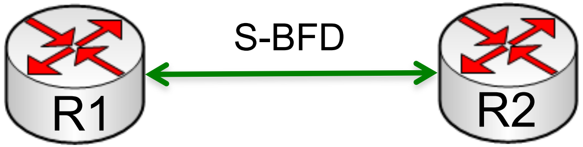
Benefits of S-BFD – Backward Compatible



- Backward Compatible with BFD.
- Re-uses same Diag Codes.
- Assigns Discriminator outside BFD range

Seamless BFD Evaluation

Evaluation – Test Environment



- Tested with variable number of sessions between test units.
- Testing done with session incremented every few hundreds.
- Metric measured includes Memory consumption, CPU utilization, time taken to establish the session.
- Above metrics are measured for each iteration with different number of sessions.

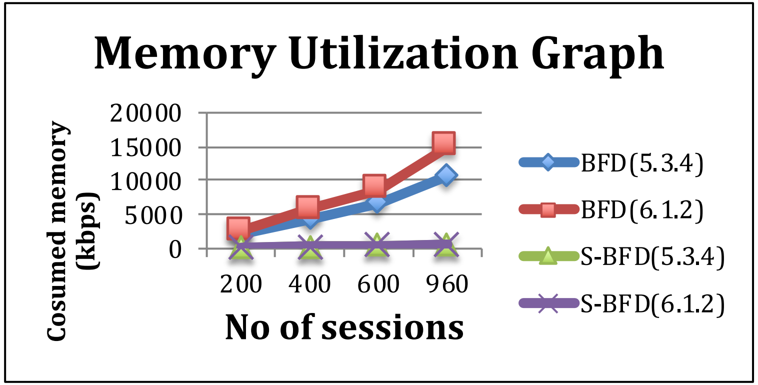
Test Environment:
Number of Sessions: ~1000
Metrics studied: Memory consumption, CPU Utilization for BFD process, Time taken for session establishment.

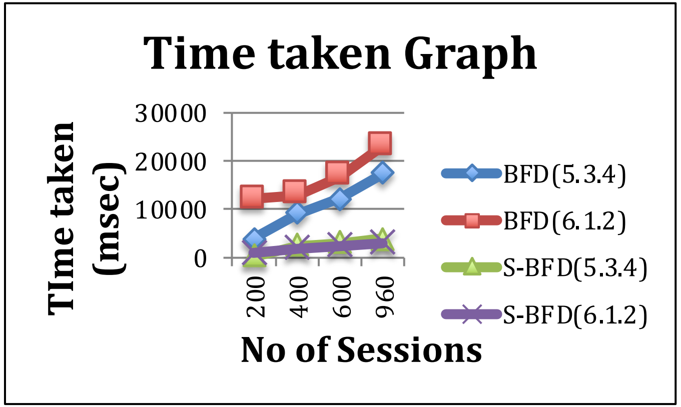
**The statistics are collected from testing environment and is not absolute value but relative values*

Evaluation Results

Traditional BFD				
No. of Sessions	**Memory Consumed (in KB)**	**Time taken for session establishment (in msec)**	**CPU Util**	**Rate of control packets**
200	2138	4012	0.18%	Variable*
400	4437	9120	0.85%	Variable*
600	6679	12238	1.0.2%	Variable*
960	10672	17331	2.03%	Variable*

Seamless BFD				
No. of Sessions	**Memory Consumed (in KB)**	**Time taken for session establishment (in msec)**	**CPU Util**	**Rate of control packets**
200	122	200	0.12%	Variable*
400	245	2510	0.68%	Variable*
600	369	3011	0.94%	Variable*
960	590	4000	1.92%	Variable*





*****_The statistics are collected from testing environment and is not absolute value but relative values_

Q&A