IP Routing Fast Convergence

Miloslav Kopka

Agenda

- Introduction
- FIB
- Routing Fast Convergence

  LFA

  BGP PIC

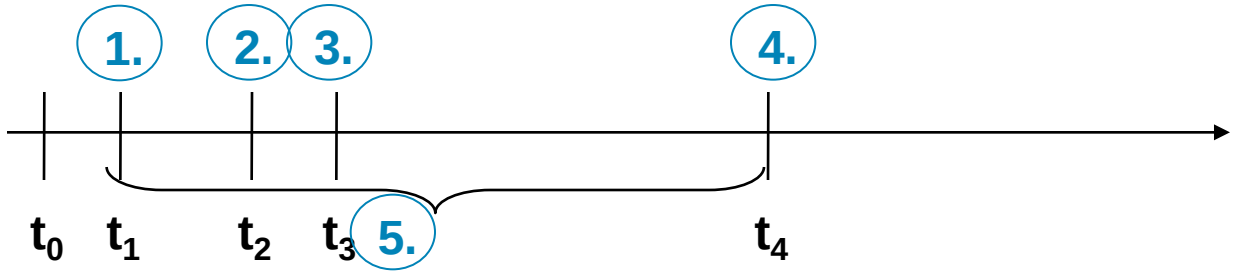
Introduction

Loss of Connectivity – LoC

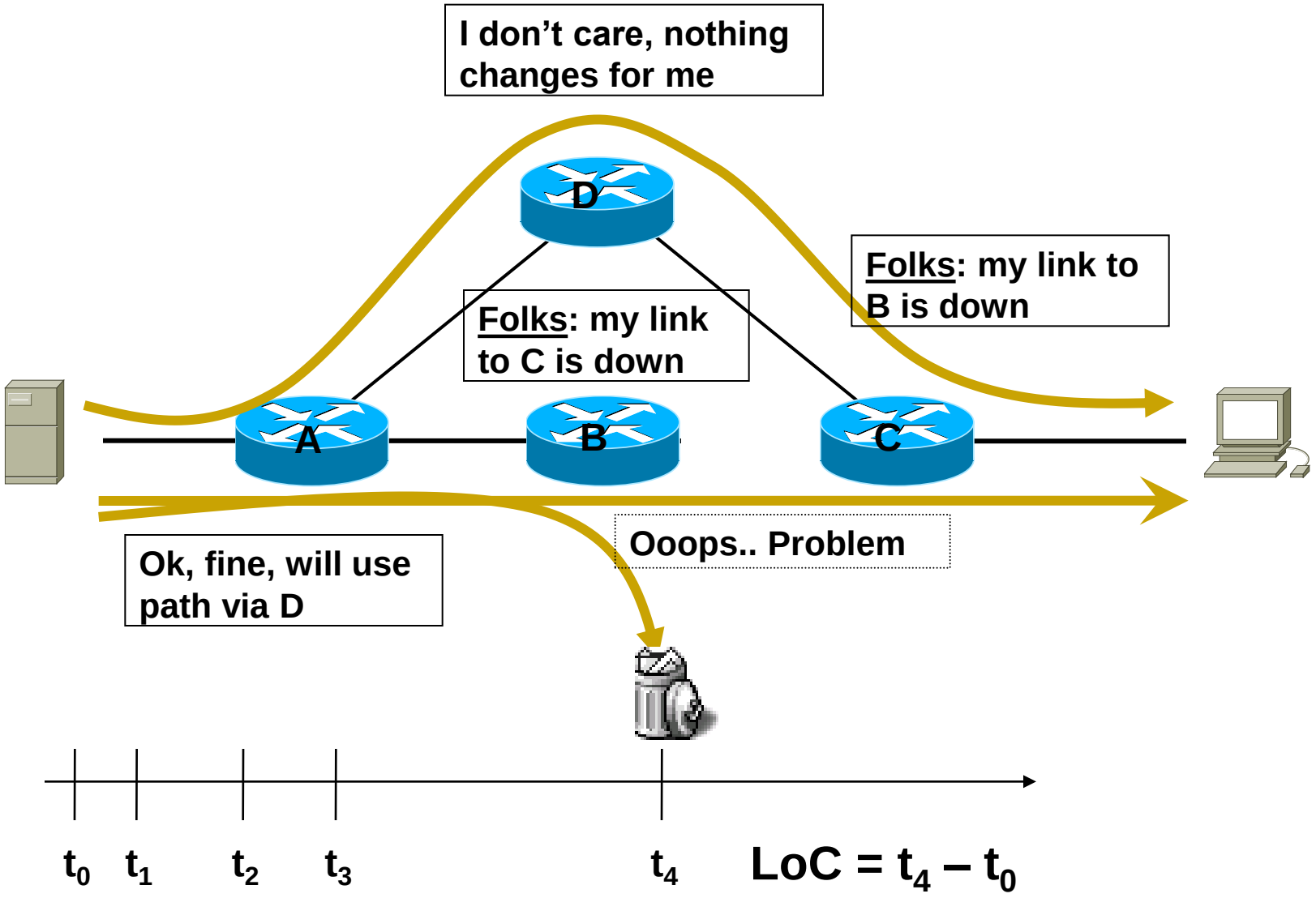
- Link and/or Node Failures cause packet loss until routing has converged
  - Loss can be caused by black-holes and/or micro-loops, until all relevant nodes in the path have converged
- Time between failure and restoration is called **LoC**
- How much LoC is acceptable?
  - Minutes?
  - Seconds?
  - Milliseconds?
- How **often** is LoC acceptable?

Routing Convergence Components

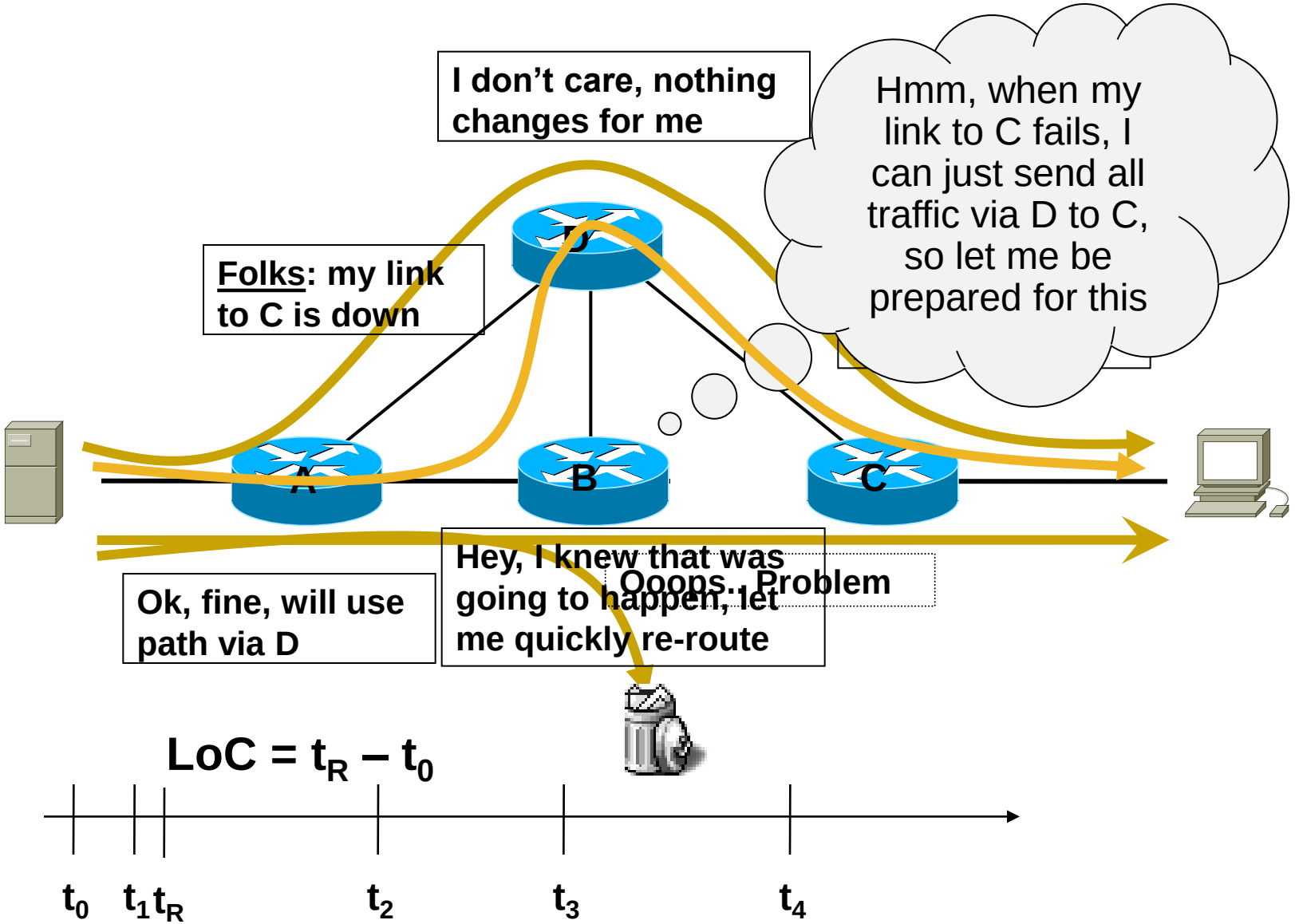
1. Failure Detection
2. Failure Propagation (flooding, etc.)
3. Topology/Routing Recalculation
4. Update of the routing and forwarding table (RIB & FIB)
5. Router Platform/Infrastructure – Fast CPU + Dedicated Control Plane



Routing Convergence Components



Fast Convergence vs. Fast ReRoute



Fast ReRoute Benefits

- Backup path is pre-computed
- Reaction to failure is local, immediate propagation and processing on other nodes is not required
- LoC as low as a few milliseconds can be achieved (provided the failure is detected quickly)

  Prefix independency is key
- Cool, so why is not everyone using it?

  It is either topology dependant or can be complex to implement (i.e. MPLS TE FRR), but we're working on this…

  It only addresses core/transit node/link failures (edge node failures still require routing convergence)

  It is not implemented on all platforms

Failure detection

- Detecting the failure is one of the most critical and often one of the most challenging part of network convergence
- Failure Detection can occur on different levels/layers

  Physical Layer (0 & 1) - G.709 (IPoDWDM), SONET/SDH (POS), Ethernet autonegotiation

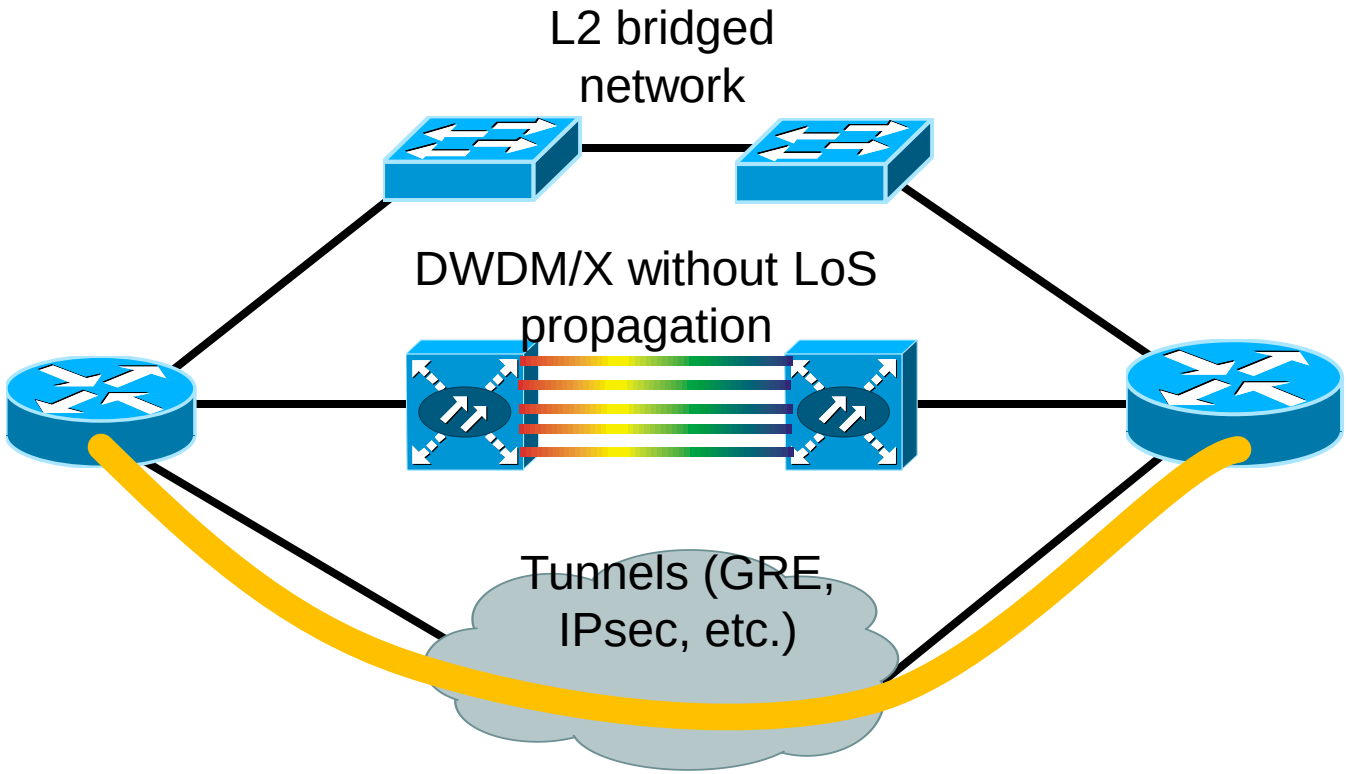
  Transport Layer (2) - PPP or HDLC keepalives, Frame-Relay LMI, ATM-OAM, Ethernet-OAM

  Network Layer (3) – IGP Hello, BFD

  Application (not covered here)

Failure Detection at Layer 3

- In some environments, some or all failures require checks at Layer 3 (i.e. IP)



Layer 3 – Network/Routing Layer

- All IGPs (EIGRP, OSPF and ISIS) use HELLOs to maintain adjacencies and to check neighbour reachability
- Hello/Hold timers can be tuned down (“Fast Hellos”), however it is not recommend doing so because
  - This does not scale well to larger number of neighbors
  - Not a robust solution, high CPU load can cause false-positives
  - Having said this: Works reasonably well in small & controlled environments, for example Campus networks

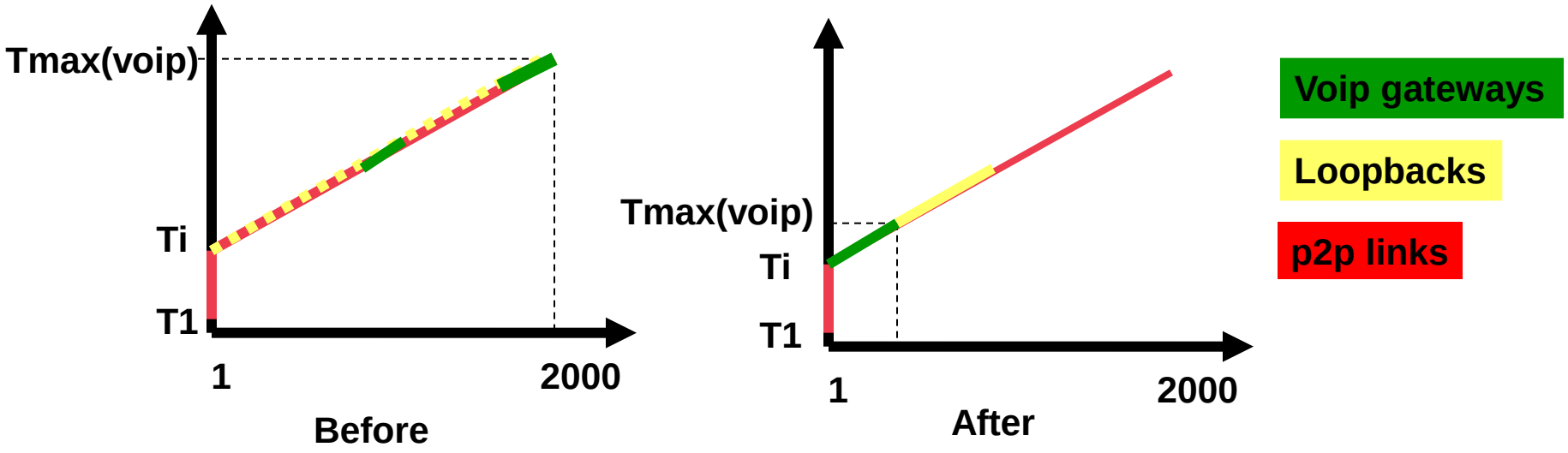
➔ We need another solution: Use BFD!

RIB & FIB Update

- Most “expensive” convergence component
- Linear dependency on the # of changed prefixes
- RIB/FIB update is measured on a per entry change and depends on
  - RP CPU speed
  - IP vs MPLS (MPLS requires more update time per entry)
- Distribution delay: RIB → FIB → LC-FIB → HW-FIB

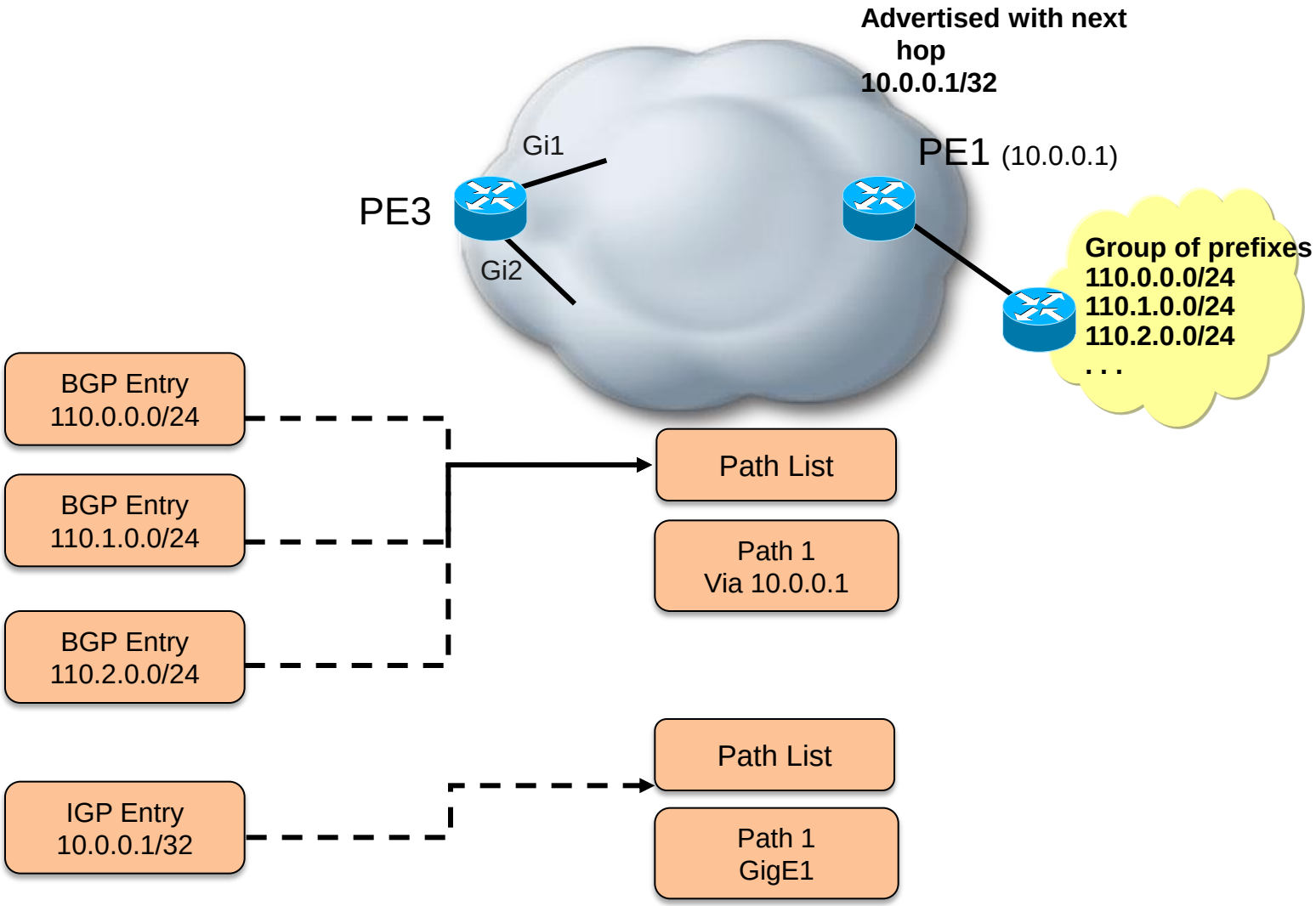
RIB/FIB – How much control?

- Prefix ordering is an important feature to reduce convergence time

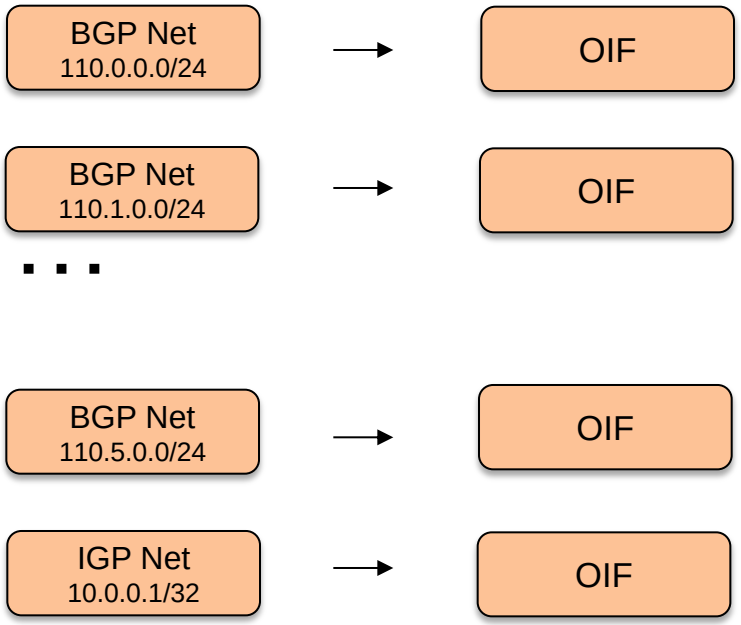


- Prefix Prioritization / Local RIB/ Prefix Suppression
- Design rule: Keep your IGP slim

Prefixes, path lists and paths

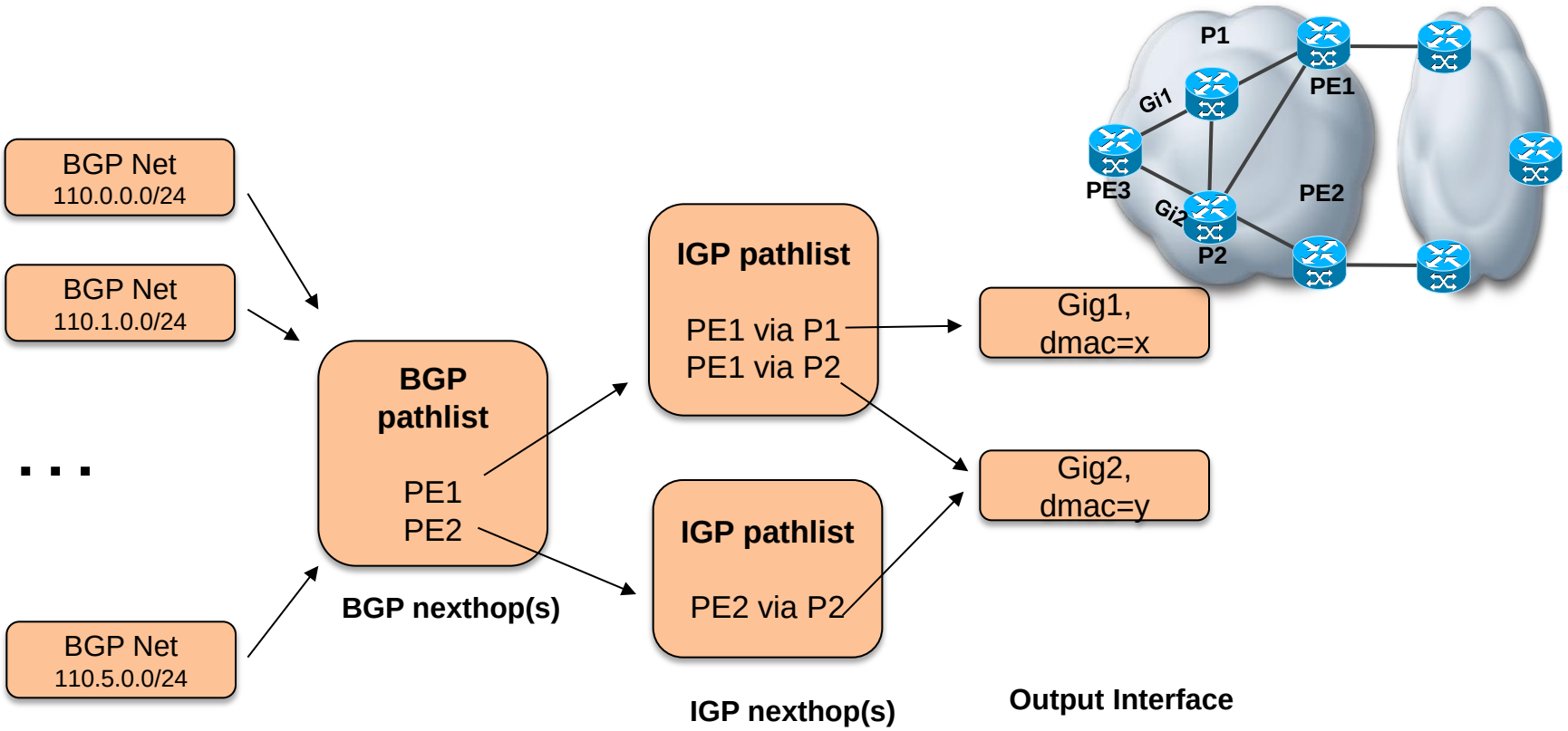


Non Optimal: Flat FIB



- Each BGP FIB entry has its own local Outgoing Interface (OIF) information (ex: GigE1)
- Forwarding Plane must directly recurse on local OIF Information
- Original Cisco implementation and still in use both by us and competition
- FIB changes can take long, dependent on number of prefixes affected

Right Architecture: Hierarchical FIB



- Pointer Indirection between BGP and IGP entries allow for immediate update of the multipath BGP pathlist at IGP convergence
- Only the parts of FIB actually affected by a change needs to be touched
- Used in newer IOS and IOS-XR (all platforms), enables Prefix Independent Convergence

IP Fast ReRoute – LFA-FRR

LFA Fast Reroute Overview

- **LFA – Loop Free Alternate**
- "The goal of IP Fast-Reroute is to reduce failure reaction time to 10s of milliseconds by using a **pre-computed alternate next-hop**, in the event that the currently selected primary next-hop fails, so that the alternate can be rapidly used when the failure is detected."

  -draft-ietf-rtgwg-ipfrr-spec-base-12

- IP Fast Reroute is an **IP based mechanism** that reduces traffic disruption to 10s of milliseconds in event of link or node failure
- A failure is **locally repaired by router next to failure** before routers in network re-converge around such failure
- **Differs from MPLS Traffic Engineering** in many aspects . . .

LFA Fast Reroute Buildings Blocks

- In order to achieve Fast Reroute, we need the forwarding engine (CEF/FIB) to hold a primary and a backup path for each prefix. The backup path is pre-computed using LFA mechanism so that we can very rapidly switch to it when a failure is detected without further computation.
- For MPLS we also need to hold in MPLS forwarding engine (MFI) primary path and backup path labels
- Backup paths are computed AFTER the primary path and so do not delay normal convergence
- In order to be prefix independent and to minimize backup switchover time, the forwarding engine (FIB) requires to be hierarchical.
- A fast detection mechanism is required to trigger the forwarding engine to switch from the primary path to the backup path. (BFD, IPoDWDM, LFS, proactive protection, …)

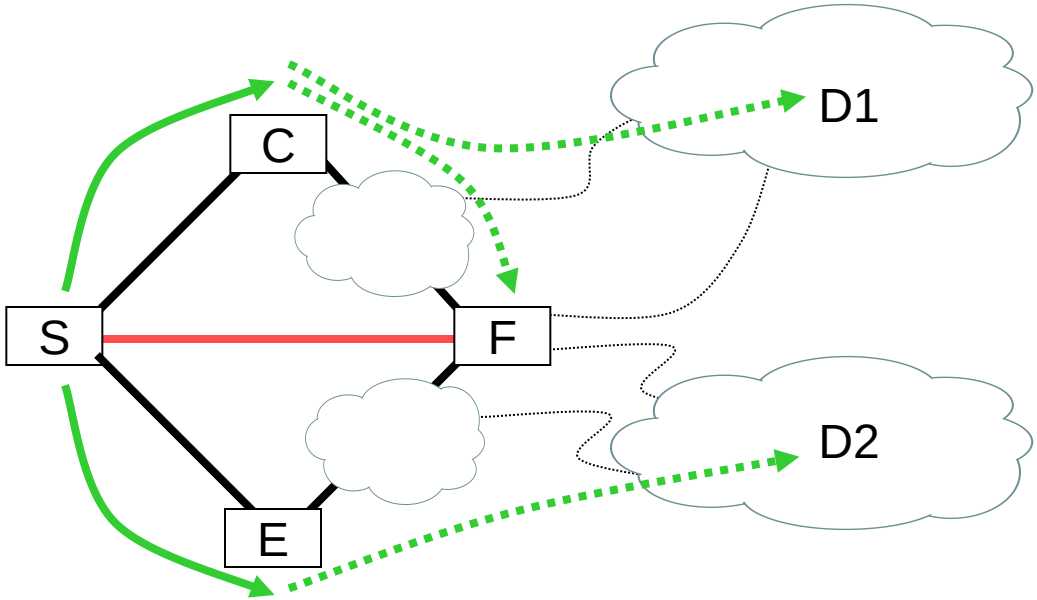
Terminology

- **D**: we will compute the LFA for this destination IGP prefix
- **S**: the router computing the LFA to D
- **F**: a neighbor of S, we typically look at the LFA's for the prefixes normally routed over link SF
- **Nx**: a neighbor of S
- Default link cost: 10
- Default link bandwidth: 10

Terminology

- Path: Outgoing interface and nhop
- Primary Path: One of (possibly many) least-cost paths
- Secondary: A costlier path than the best.
- Backup: an outgoing interface/nhop which is used to replace another one that went down. It can be:
  - another primary ECMP nhop
  - a secondary LFA routing path
  - a TE-tunnel could be a backup

Terminology



- LFA: Loop-Free Alternate

  N is an LFA for S's primary path to D via F if ND < NS + SD

  Node-protecting LFA if: ND < NF + FD

Per-Prefix LFA FRR

- IGP pre-computes a backup path per IGP prefix

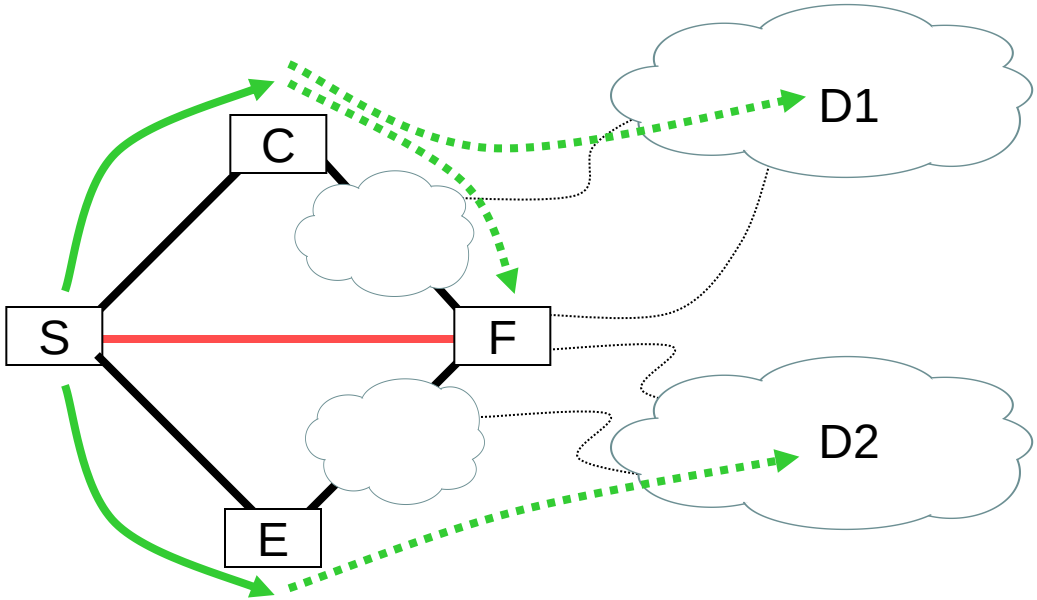
  IGP Route 1.1.1.1/32

  Primary Path: PoS1

  Backup Path: <LFA>

- FIB pre-installs the backup path in dataplane
- Upon local failure, all the backup paths of the impacted prefixes are enabled in a prefix-independent manner (<50msec LoC)

Per-Prefix LFA Algorithm



- For IGP route D1, S's primary path is link SF.
- S checks for each neighbor N (<>F) whether ND1 < NS + SD1 (Eq1)

  "does the path from the neighbor to D1 avoid me?"

  If so, it is a loop-free alternate (LFA) to my primary path to D1

Guaranteed-Node-Protecting

- Definition: ND < NF + FD (Eq2)

  "does the path from the neighbor to D avoid node F?"

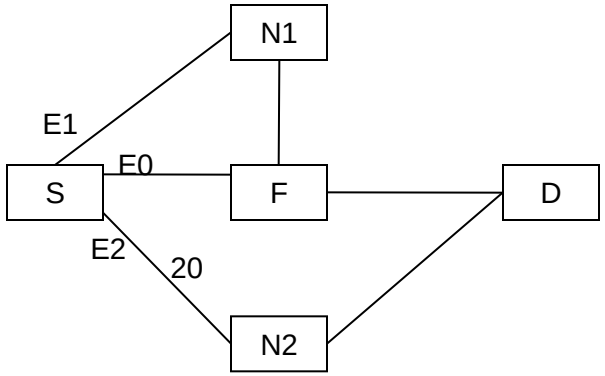
- S's primary path to D: E0
- S's LFA candidates: E1 and E2

  E1: not guaranteed node protecting

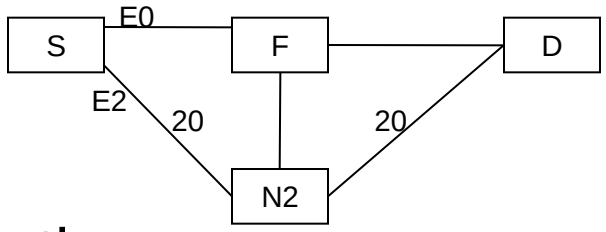
  N1 does not meet Eq2

  E2: guaranteed node protecting

  N2 meets Eq2



De Facto Node Protection

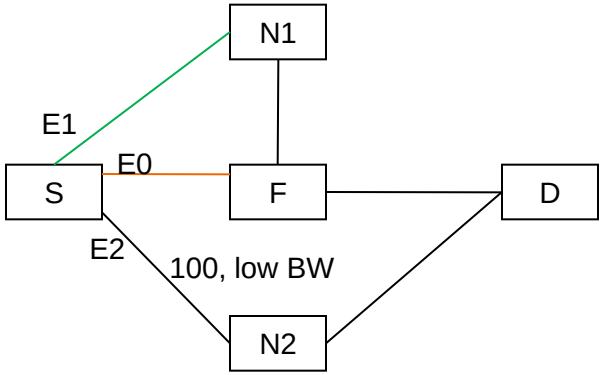


- Choosing a guaranteed node-protecting LFA is not the only way to benefit from LFA node protection
- A non-guaranteed node protecting LFA candidate might turn to be node protecting. We call this “De Facto Node Protection”

  N2 is not guaranteed node protecting for D: 20 !< 10+10

  But if F fails, N2 will trigger its own LFA for the path N2FD (via N2D) and hence the traffic SD avoids F!

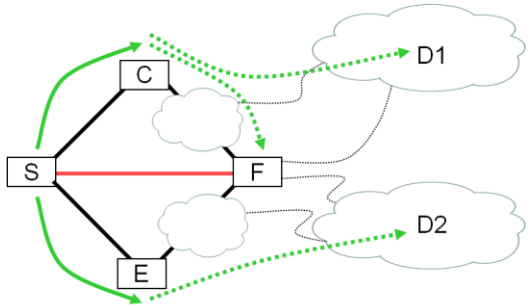
A guaranteed-Node-protecting is not always better



- Failure Statistics indicate that node failures are very rare
- Simulation indicates that non-guaranteed node-protecting LFA's actually provide De Facto Node Protection in many cases
- Priviledging a guaranteed node-protecting LFA might lead to using a non-optimal path

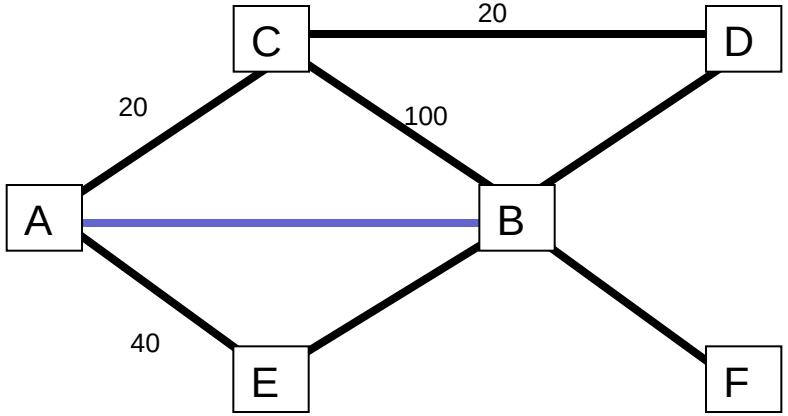
  SN2D involves a low-BW link (SN2). It would be used if F would really go down. When F is up and just the link SF goes down, it is much more efficient to use N1.

Per-Link LFA Algorithm



- A subcase of per-prefix LFA
- IGP computation changes (slightly)
  - one single per-prefix LFA is computed: for address “F”
  - “a valid backup path to F itself is valid for any destination reached via F”
- RIB/FIB representation does not change
  - one backup path per primary path
  - all the backup paths are the same
    - D1: primary path SF, backup path: SC
    - D2: primary path SF, backup path: SC

Example – Control Plane Output



IGP Per-Link LFA Result:

Link AB is protected via E

IGP Per-Prefix LFA Result:

D's LFA is via C
B's LFA is via E
E's LFA is via E
F's LFA is via E

Per-Prefix vs Per-Link

- Better Coverage

  Each prefix is analyzed independently. Most often it is the prefix at the other end which has no LFA…

- Better Capacity Planning

  Each flow is backed up on its own optimized shortest path to the destination

- Better Node Protection

  Can be verified algorithmically

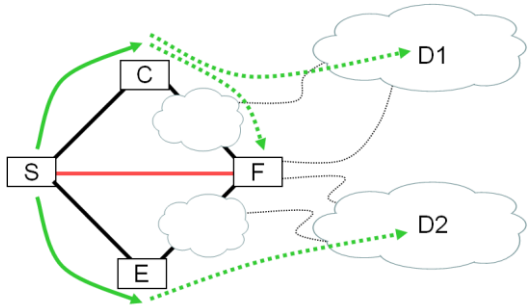
- Same Dataplane Scale

- Higher Control Plane CPU consumption

One backup path per primary path

- IGP will select one and only one backup path per primary path
- Need to select an LFA among multiple candidates (tie-break)
- Tie-break works as BGP Best-Path
  - a set of consecutive rules
  - each rule discards candidates
  - scheme stops when one single path remains
  - if a rule excludes all paths, then the rule is skipped
- Tie-breaking rules cannot eliminate all candidates
- The default Tie breaking order is configurable

LDP support



- The backup path for destination D must contain the label bound by the backup neighbor
  - backup label is dependent on (IGP destination, backup path)
- This is why, whether the IGP computes per-prefix or per-link, the RIB and FIB representation is always per-prefix
  - this allows to store the per-path dependent backup label

No degradation for IGP FC

- Per-Prefix LFA Computation is throttled by its own independent exp-backoff
- An LFA computation does not start until the primary computation is finished.
- An ongoing LFA computation is interrupted if a new primary computation is scheduled.

Restrictions

- A link is either configured for per-link LFA OR per-prefix LFA
- A given interface should be configured for LFA FRR or TE FRR
- Per-prefix LFA should be configured on only point-to-point interfaces.

  If Ethernet is used, it should be running in p2p mode. However LAN interfaces could be used as backup.

Operation

- Show command to provide the % of coverage
  - across all IGP routes
  - per IGP priority (critical, high, medium, low)
  - per primary interface
- Show command to display the selected backup path per primary IGP path

Benefits

- Simple
  - the router computes everything automatically
- <50msec
  - pre-computed, pre-installed, enabled on link down in a prefix independent manner
  - prefix-independent for any prefix type
  - Leverage Hierarchical dataplane FIB
- Deployment friendly
  - no IETF protocol change, no interop testing, incremental deployment

Benefits

- Good Scaling
- No degradation on IGP convergence for primary paths
- Node Protection or De Facto

  an LFA can be chosen on the basis of the guaranteed-node protection

  simulation indicate that most link-based LFA's anyway avoid the node (ie. De Facto Node Protection)

Constraints

- Topology dependent

  availability of a backup path depends on topology

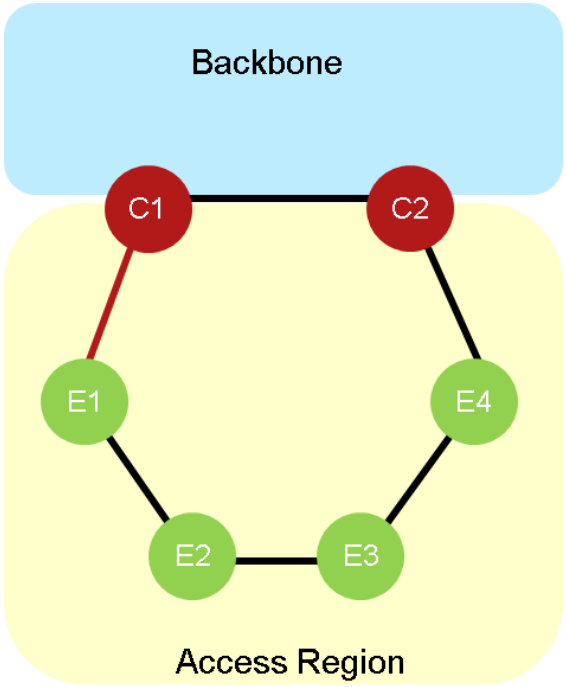
  Is there a neighbor which meets Eq1?

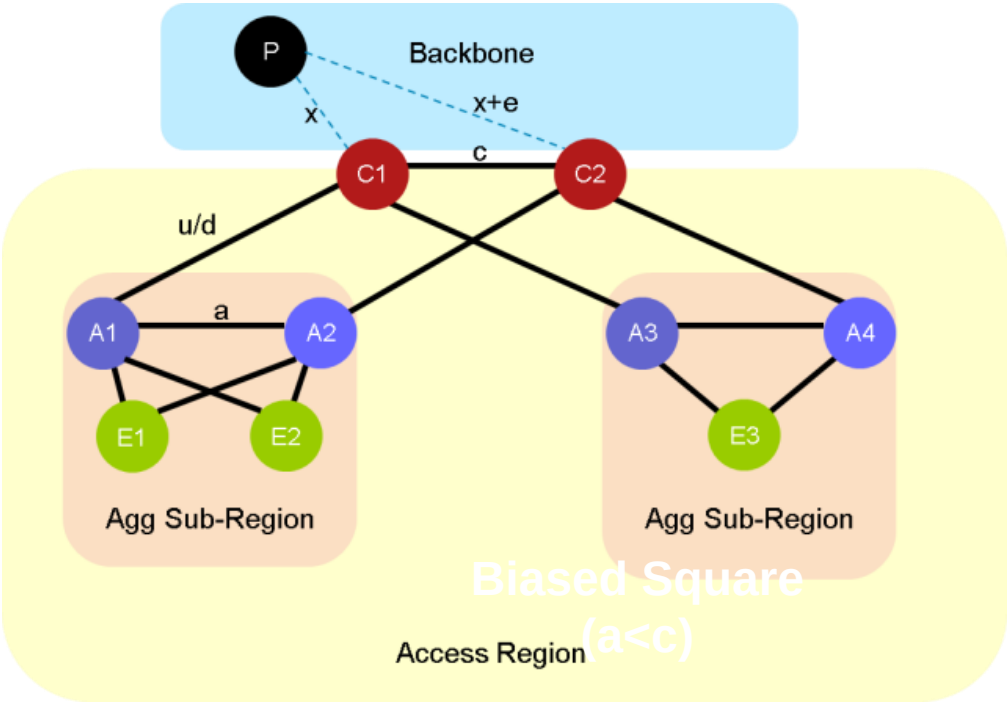
LFA Coverage is *really* excellent

- 11 *real* Core Topologies
  - average coverage: 94% of destinations
  - 5 topologies higher than 98% coverage
- *Real* Aggregation
  - simple design rules help ensure 100% link/node protection coverage for most frequent *real* aggregation topologies
  - RFC6571
  - Sweet Spot
    - A simple solution is essential for access/aggregation as it represents 90% of the network size hence complexity

High interest for access/aggregation

- Is there a way to also support the ring and “biased square”?

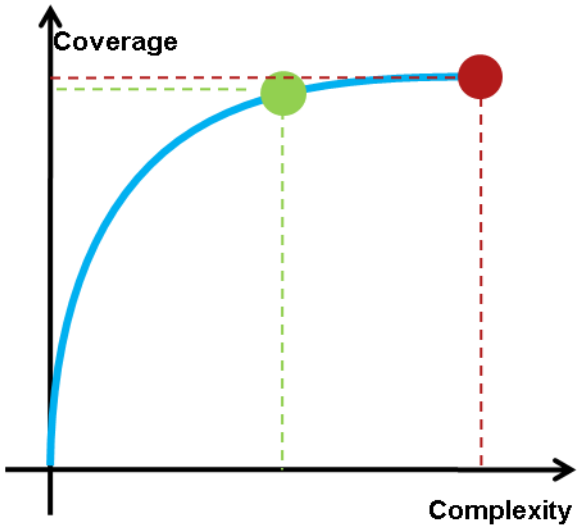




Fast ReRoute – Remote LFA

Remote-LFA Objective

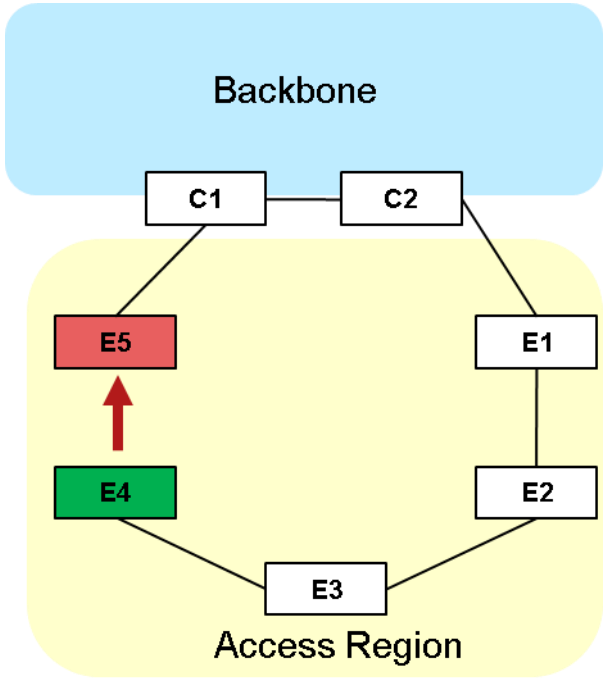
- Absolutely keep Per-prefix LFA benefits
  - simplicity
  - incremental deployment
- Increase coverage for *real* topologies
  - primarily for ring and biased-square access topologies
  - potentially for core topology
  - “98/99%” is seen as good-enough
    - 100% coverage is “icing on the cake”



The Ring

- No LFA protection in the ring

  if E4 sends a C1-destined packet to E3, E3 sends it back to E4



Remote LFA (aka PQ Algorithm)

- Any node which meets the P and Q properties

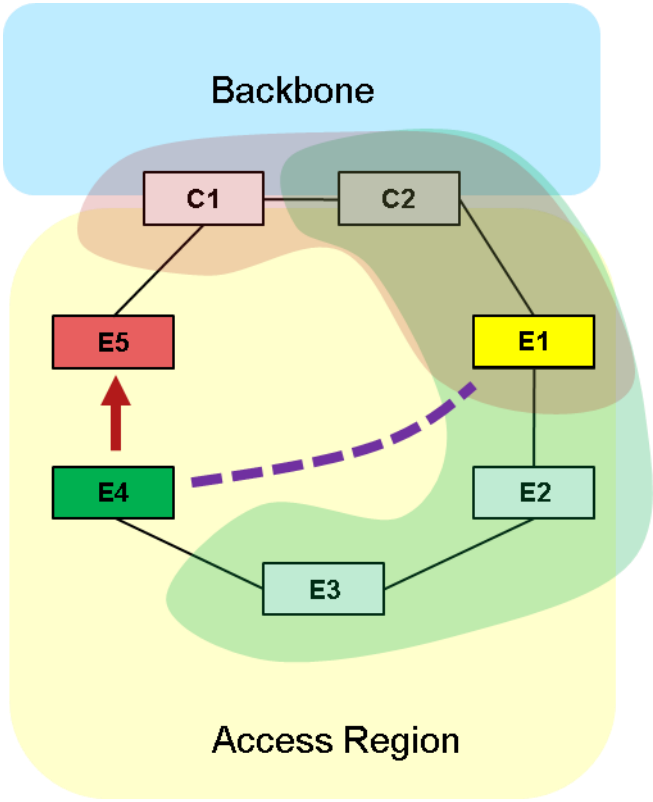
  P: the set of nodes reachable from E3 without traversing E4E5

  Q: the set of nodes which can reach E5 without traversing E4E5

- Best PQ node

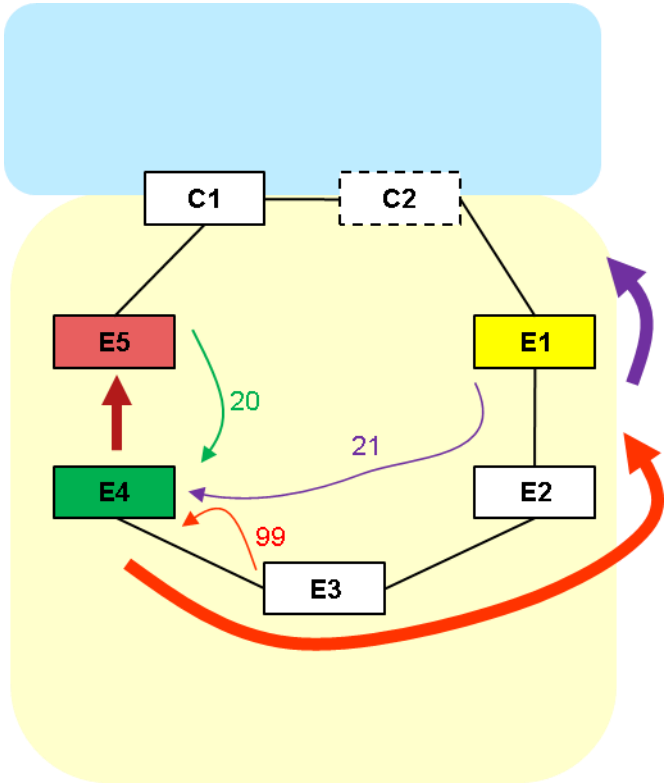
  the closest from E4: E1

- Establish a directed LDP session with the selected PQ node



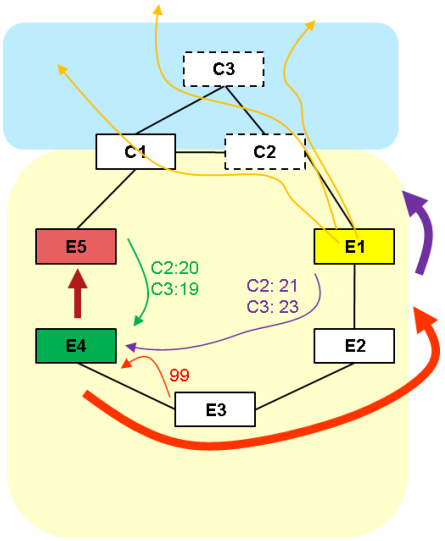
Remote LFA Protection

- E4's LIB
  - E5's label for FEC C2 = 20
  - E3's label for FEC E1 = 99
  - E1's label for FEC C2 = 21
- E4's FIB for destination C2
  - Primary: out-label = 20, oif = E5
  - Backup: out-label = 21
    - oif = [push 99, oif = E3]



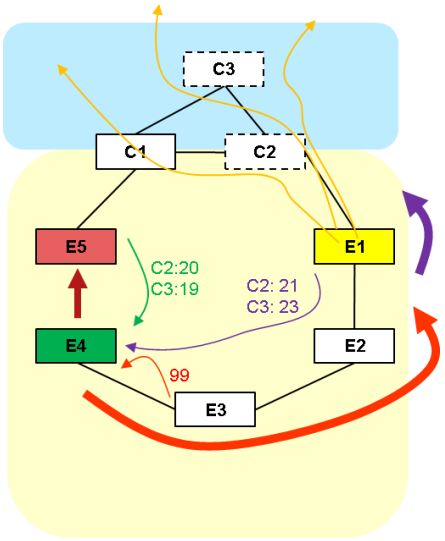
RLFA is LFA from a remote node (E1)

Per-Link Computation



- E4 computes a single PQ node for the link E4E5
  - P space: does not require any new computation
    - Nodes whose shortest-path avoid E4E5
    - Nodes whose shortest-path is E4E5 but have an LFA
  - Q space: one dijkstra per link (1-2ms Typical)
    - Dijkstra rooted at E5 with the reverse metrics and with E5E4 branch pruned
- Per-Prefix LFA as if E1 was directly connected neighbor
- Extremely low CPU requirement

Per-Prefix Backup



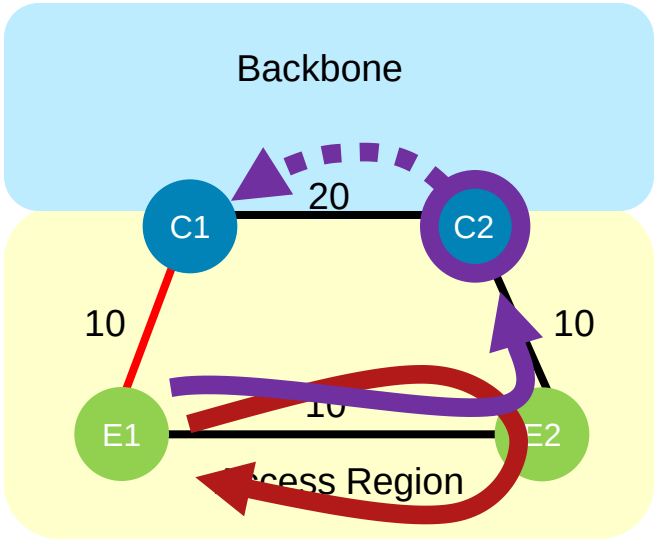
- Exactly like per-prefix LFA but as if the PQ node was directly connected
- Hence, all per-prefix LFA benefits are preserved
  - each packet uses its shortest-path from the PQ node
  - Excellent capacity planning
  - “De Facto” Node protection
    - (100% node protection for ring and biased square)

Remote LFA applied to Biased Square

- Upon E1C1 failure, E1 has no per-prefix LFA to C1

  E2 routes C1 via E1

- With Remote LFA, upon E1C1 failure, E1 forwards the packets destined to C1 towards the PQ node (C2) from where the packet reaches C1

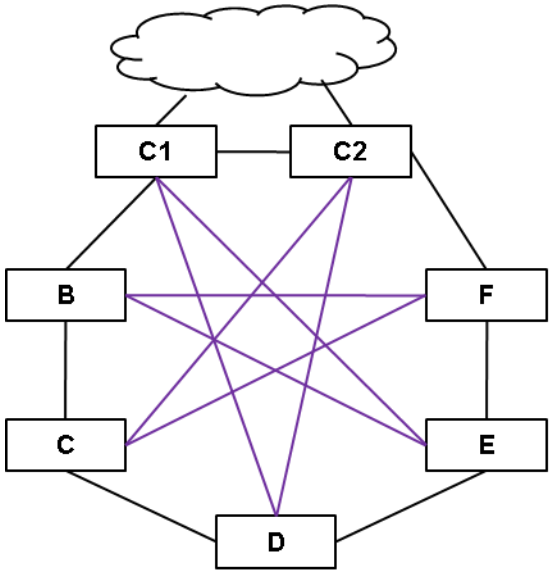


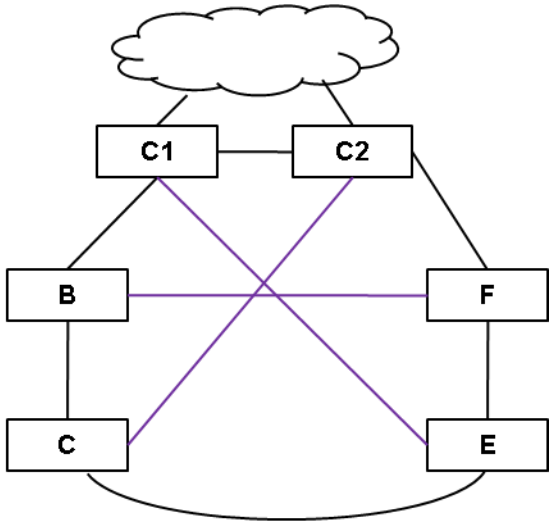
Targeted LDP session

- Upon computation of a new PQ node X, the local router R establishes a targeted LDP session with PQ node X
- X must be configured to accept Targeted LDP session

  mpls ldp discovery targeted-hello accept [from <peer-acl>]
- Same security model as PWE, VPLS and LDP over Full-Mesh TE deployments

Targeted LDP – Scalable

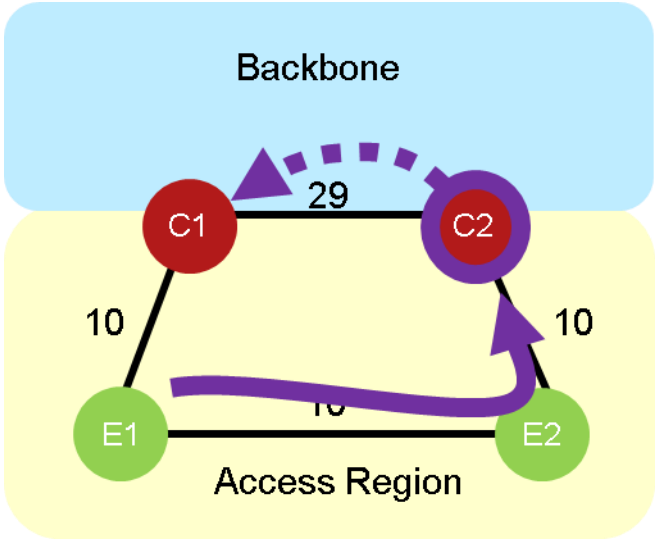




- Odd ring: 2 LDP additional sessions per node
- Even ring: 1 LDP additional session per node
- Biased Square: 1 LDP additional session per node

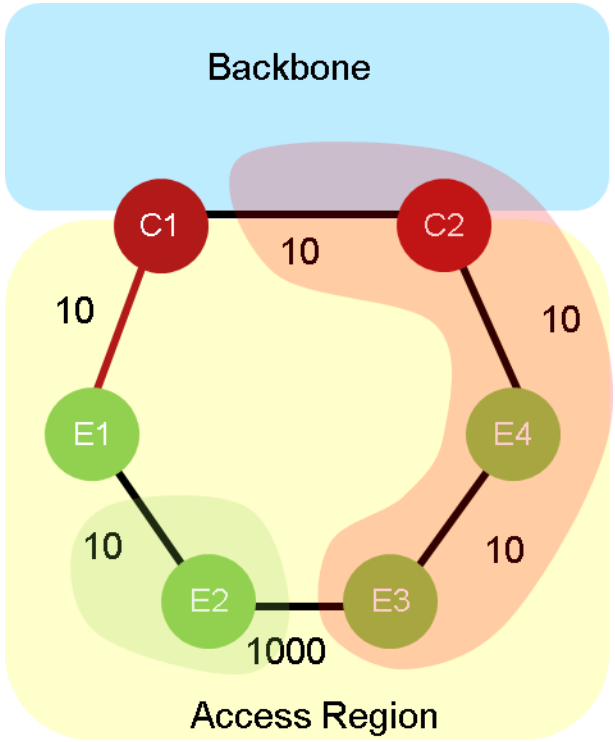
Very simple rules – RFC6571

- In a square, any metric should be less than the sum of the 3 other links
- If C1–C2 Metric is 30 Remote LFA does not work



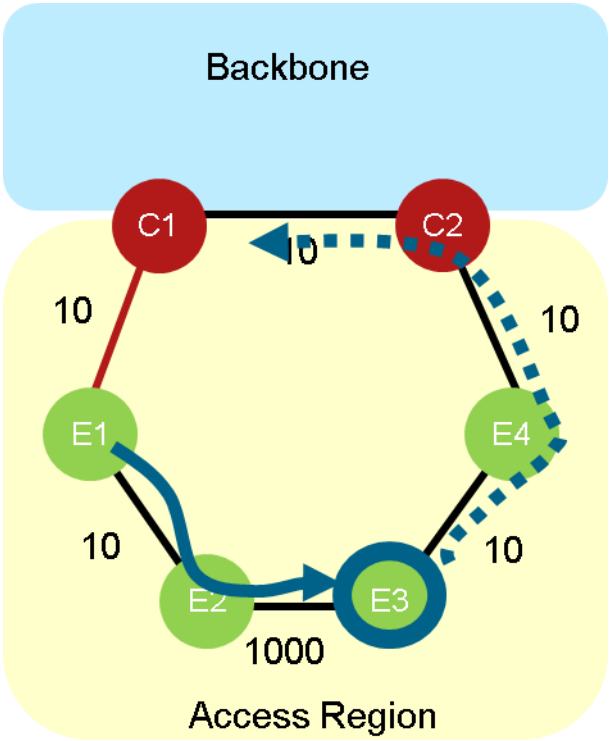
Not yet 100%-guaranteed…

- E1 has no LFA for C1
  - E2 routes back
- E1 has no Remote LFA for C1
  - P and Q intersection is null



100% - Icing on the cake

- When the P and Q space do not intersect, then setup an RSVP-LSP to the closest Q node
  - RSVP allows an explicit path and hence a path that avoids the primary link
- Very few RSVP-LSP's
- Automatically – Detect the Case
  - Build Explicit Path
  - Signal RSVP
- 100% guarantee
- Node protection



Remote LFA Benefits

- Seamless integration with Per-Prefix LFA
  - Packets take their shortest paths from the PQ node
  - Destinations use per-prefix LFA onto physical oif when available (i.e. per-prefix LFA), and per-prefix LFA onto LDP LSP (i.e. Remote LFA) otherwise
- Simple
  - Automated computation, negligible CPU, low T-LDP requirement
- Incremental Deployment
  - New code only at the protecting node
- Meet the real coverage requirements
  - backbone and access/aggregation

BGP Fast Convergence

BGP “Fast” Convergence

- Where/Why do we need BGP Fast Convergence?
  - Layer 3 VPN Convergence
  - Peering/Upstream router or link failure
  - Usually contained within a single AS
- IGPs have a few thousand prefixes max, with a couple of hundred important ones
  - Optimizing the implementation for FC is relatively easy
- BGP scales to > million routes
  - Tuning the protocol and implementation to provide sub-second convergence without sacrificing the scalability is impossible

BGP Control-Plane Convergence Components I: Core failure

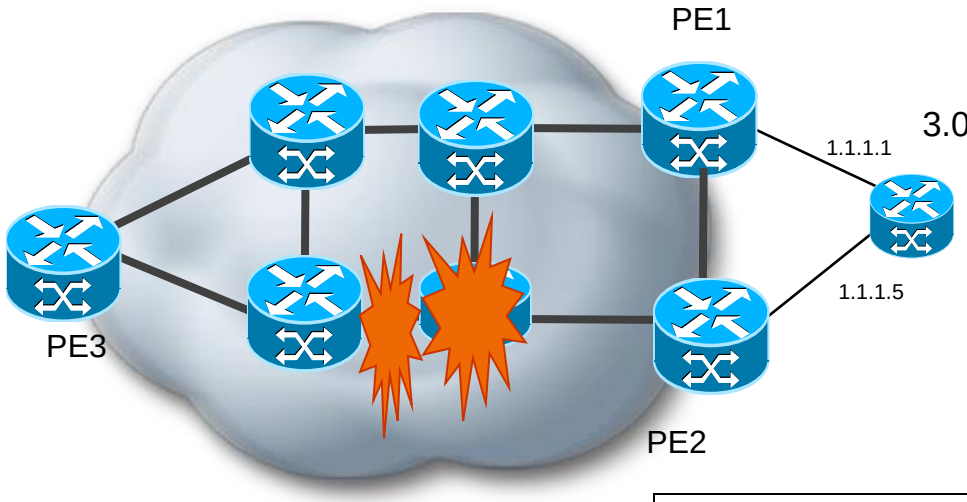
1. Core Link or node goes down

2. IGP notices failure, computes new paths to PE1/PE2

3. IGP notifies BGP that a path to a next-hop has changed

4. PE3 identifies affected paths, runs best path, path to PE2 no longer as good as the one to PE1

5. Updates RIB/FIB, traffic continues




```
3/8, NH 1.1.1.1
     NH <PE2>
…
```

```
3/8, NH <PE1>
     NH
…    <PE2>
```

```
3/8, NH 1.1.1.5
     NH <PE1>
…
```

BGP Control-Plane Convergence Components II: Edge Node Failure

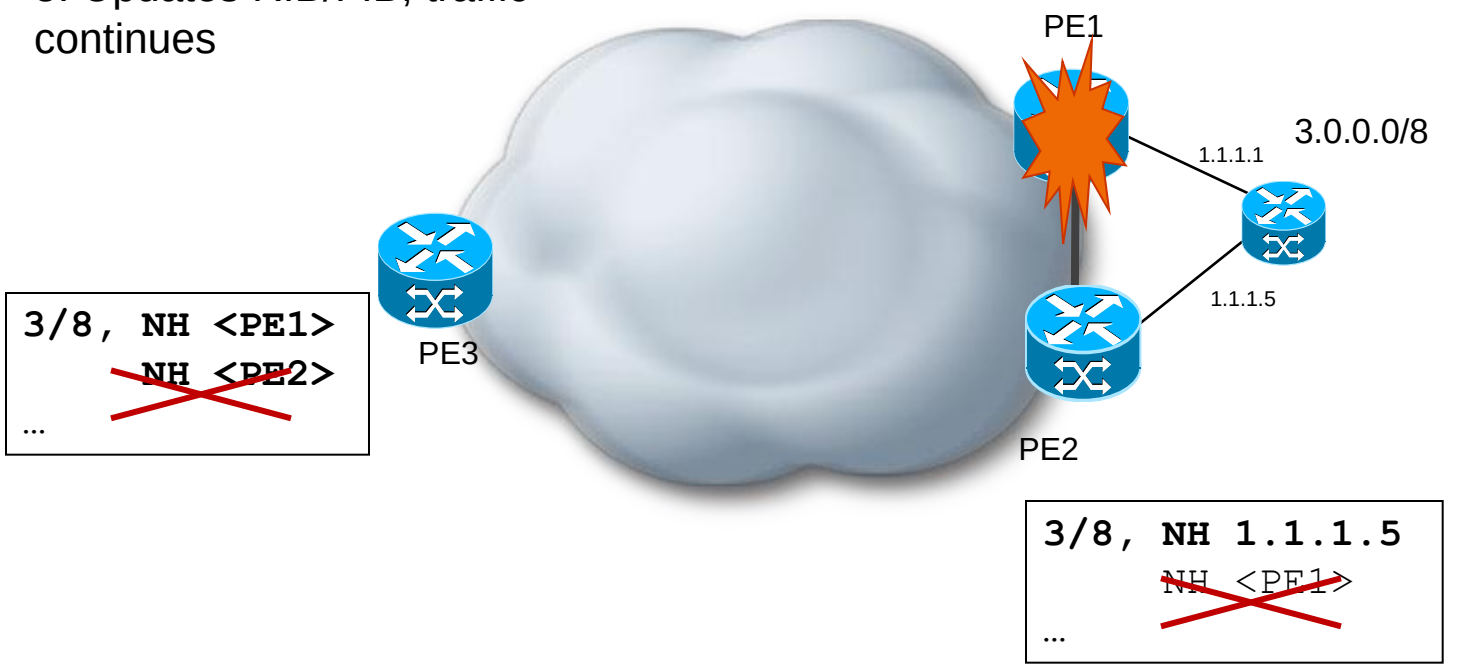
Edge Node (PE1) goes down

2. IGP notices failure, update RIB, remove path to PE1

3. IGP notifies BGP that path to PE1 is now longer valid

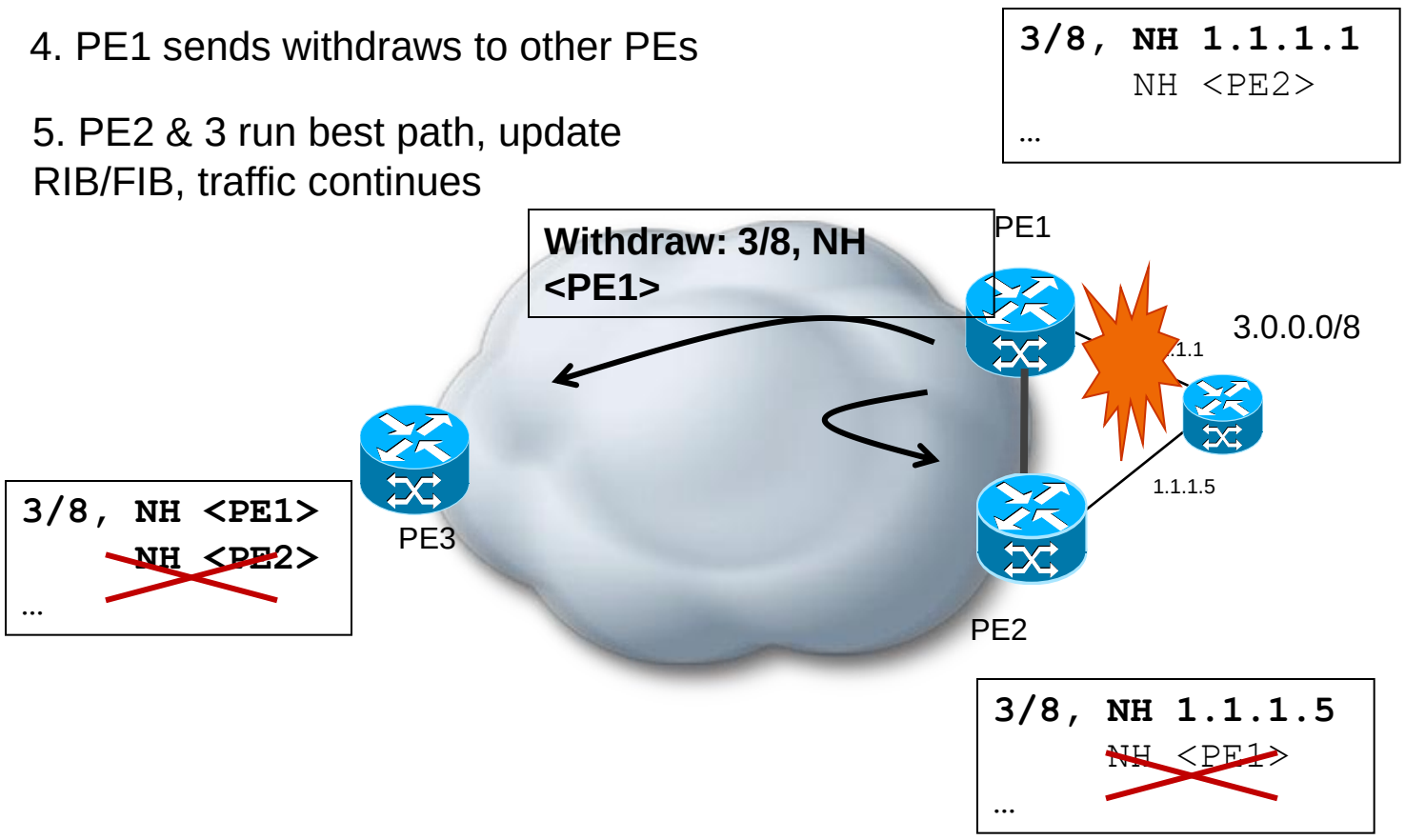
4. PE3 identifies affected paths, runs best path, removes paths via PE1

5. Updates RIB/FIB, traffic continues



BGP Control-Plane Convergence Components III: Edge Neighbour Failure (with next-hop-self)

1. Edge link on PE1 goes down
2. eBGP session goes down
3. PE1 identifies affected paths, runs best path
4. PE1 sends withdraws to other PEs
5. PE2 & 3 run best path, update RIB/FIB, traffic continues



BGP Control-Plane Convergence Components

- Failure Detection
- Reaction to Failure
- Failure Propagation
- RIB/FIB Update

Control vs. Data Plane Convergence

- Control Plane Convergence

  For the topology after the failure, the optimal path is known and installed in the dataplane

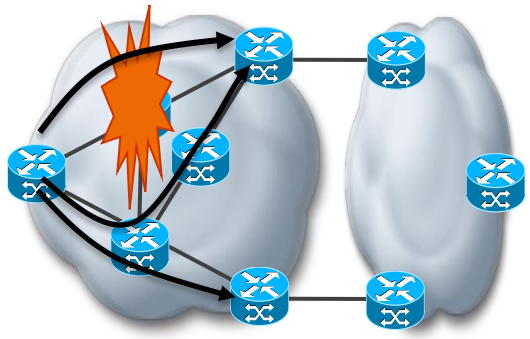
  May be extremely long (table size)

- Data Plane Convergence

  Once IGP convergence has detected the failure, the packets are rerouted onto a valid path to the BGP destination

  While valid, this path may not optimal one from a control plane convergence viewpoint

  We want this behaviour, in a prefix-independent way – that's what this session is all about

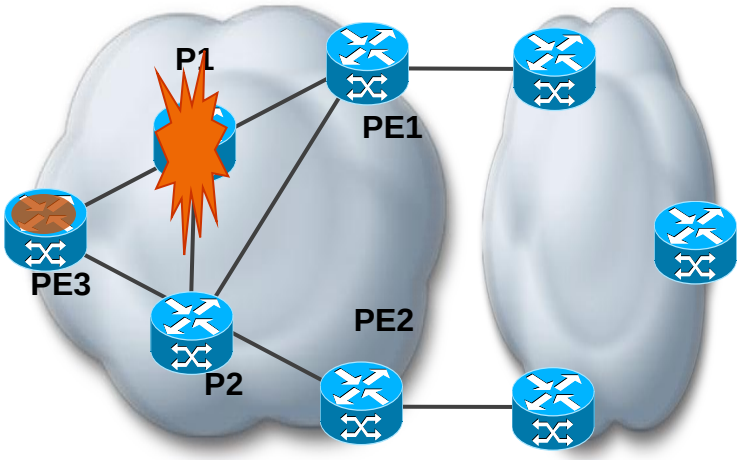


BGP Control-Plane Fast Convergence – Summary

- In reasonably small L3VPN environments (< 1000 pfx), sub-second can be achieved for most failures
  - be aware of active/standby policies, takes much longer
- However, with larger number of VRFs and/or routing tables, sub-5 seconds is a more realistic target
- Internet routing table convergence takes minutes

We want prefix-independent behaviour
➔ BGP Prefix Independent Convergence (BGP-PIC)

BGP PIC Core

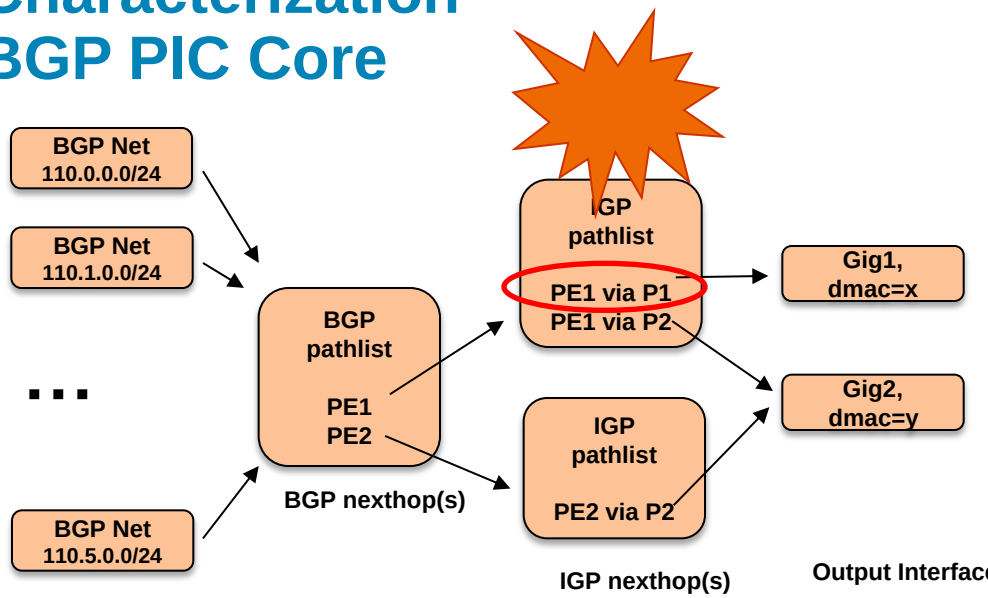


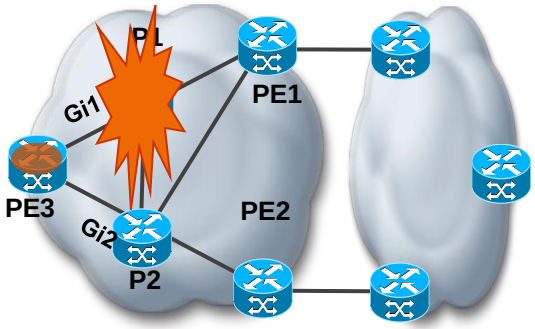
- **Core Failure**: a failure within the analyzed AS
- IGP convergence on PE3 leads to a **modification** of the RIB path to PE1

  BGP Dataplane Convergence is finished assuming the new path to the BGP Next Hop is leveraged immediately (**BGP PIC Core**)

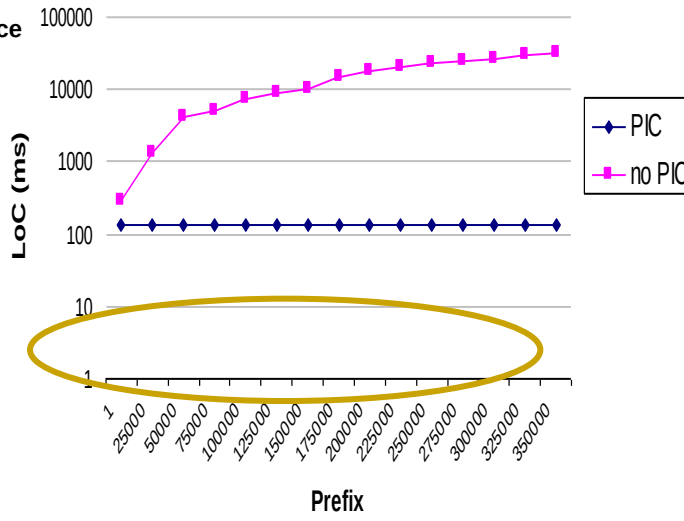
  BGP NHT sends a “modify” notification to BGP which may trigger BGP Control-Plane Convergence. This may be long without impacting Tight-SLA experience

Characterization
BGP PIC Core



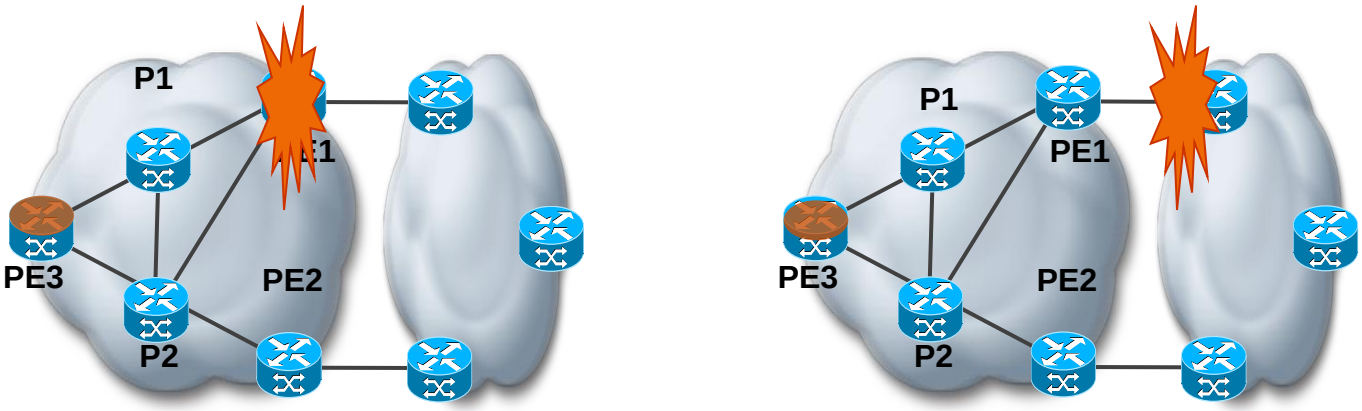


- As soon as IGP converges, the IGP pathlist memory is updated, and hence all children BGP PL's leverage the new path immediately
- Optimum convergence, Optimum Load-Balancing, Excellent Robustness



BGP PIC Edge

PE1 does not set next-hop-self

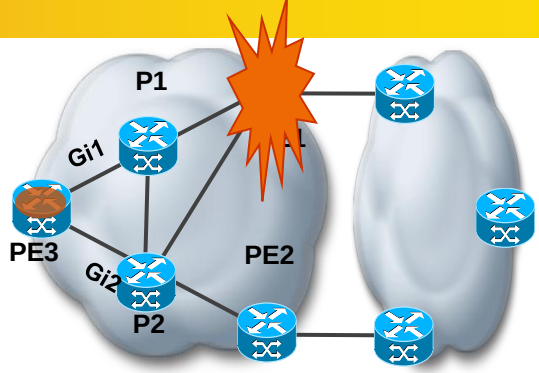


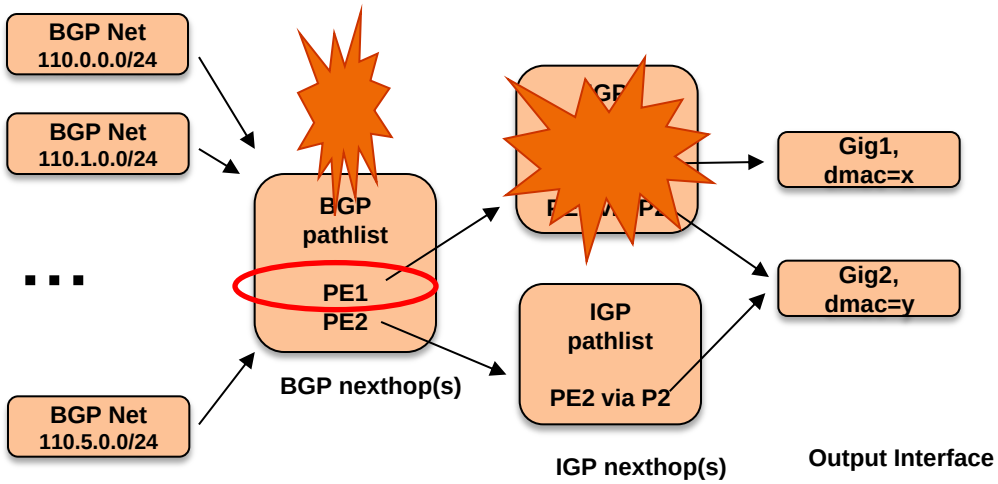
- **Edge Failure**: a failure at the edge of the analyzed AS
- IGP convergence on PE3 leads to a **deletion** of the RIB path to BGP Next-Hop PE1

  BGP Dataplane Convergence is kicked in on PE3 (**BGP PIC Edge**) and immediately redirects the packets via PE2

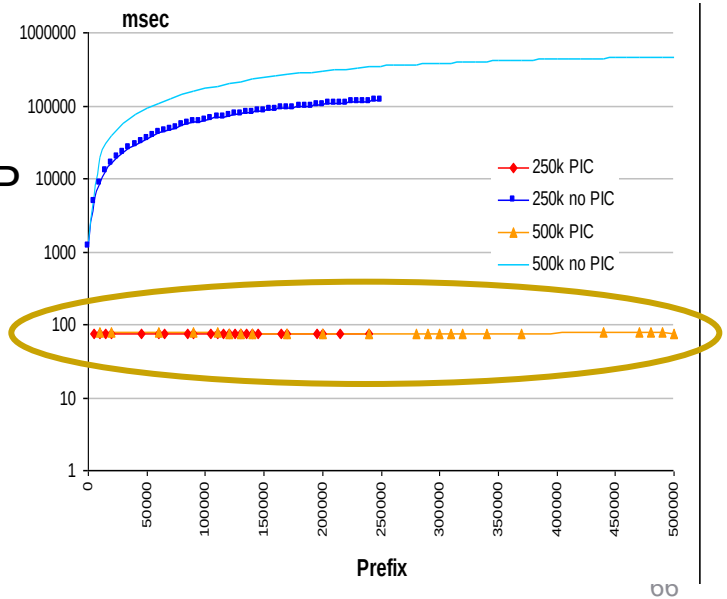
  BGP NHT sends a “delete” notification to BGP which triggers BGP Control-Plane Convergence. This may be long, but without impacting Tight-SLA experience

Characterization
BGP PIC Edge

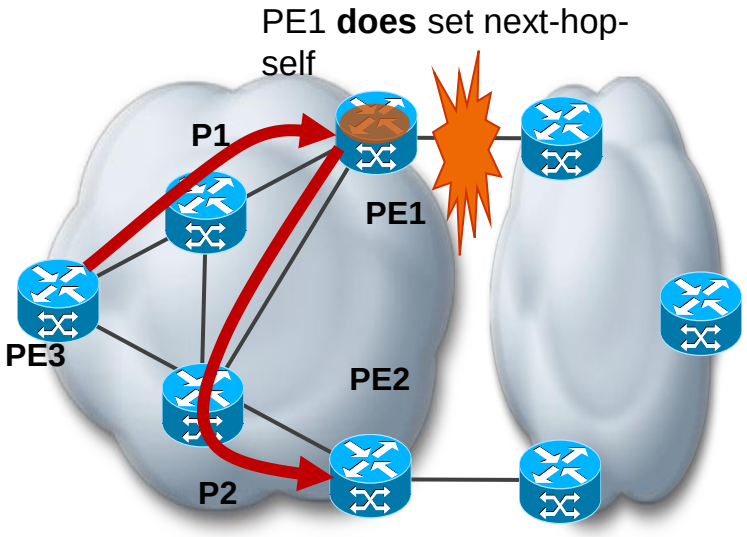




- At IGP Convergence time, the complete IGP pathlist to PE1 is deleted
- SW FIB walks the linked list of parent BGP path lists and in-place modify them to use alternate next hops (ECMP or backup). BGP Path lists are shared, so this is efficient



BGP PIC Edge and Next-Hop Self



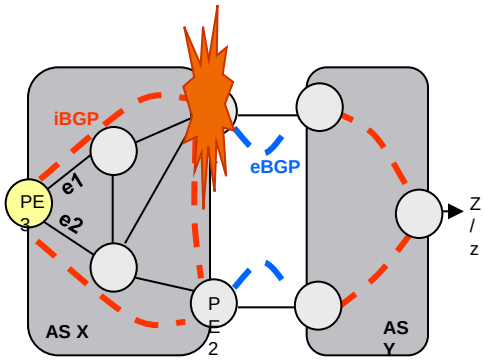
- With the edge device setting next-hop to its loopback (next-hop-self), edge link going down does not change next-hop as seen on other routers

  Failure goes unnoticed by others!

- However: Next-hop on edge device with failing link goes away, so this device can react in PIC-Edge fashion

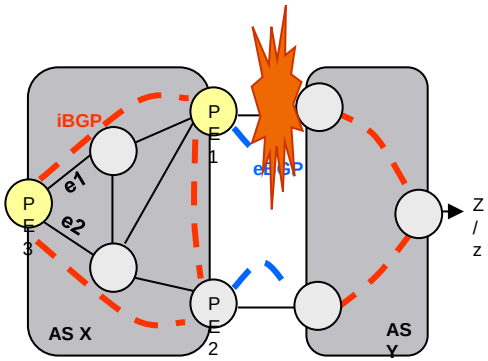
  Traffic re-routed via core to alternate edge

Summary: BGP PIC Edge application points



PE3 must be the reacting point as PE1 is down.

PE3's reaction is triggered by IGP convergence

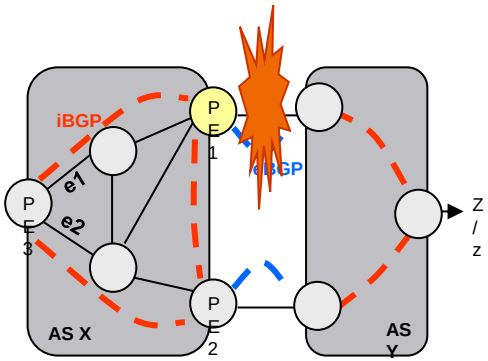


PE1 does not set next-hop-self

PE3 and PE1 may be the reacting point.

PE3 reaction is triggered by IGP convergence

PE1 reaction is triggered by local interface failure



PE1 sets next-hop-self

PE1 is the reacting point.

PE1 reaction is triggered by local interface failure

Note: PE3 is blind in this case as the next-hop is PE1

Key Take-Aways

- PIC Core and PIC-Edge leverage hierarchical forwarding structure
  - PIC Core: Path towards Next-Hop changes – IGP LoadInfo changed
  - PIC Edge: Next-Hop goes away – BGP Path list changed
- ➔ All BGP prefixes (no matter how many!!) converge as quickly as their next-hop
- Generally, IGP is responsible for next-hop convergence ➔ BGP convergence depends on IGP convergence

- So? What do I need to do to speed up my BGP convergence with BGP PIC???

Designing for BGP PIC

Designing (for) BGP PIC

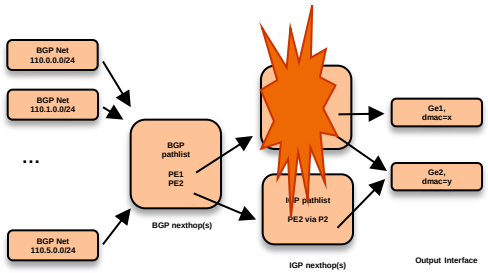
BGP PIC is a forwarding / data plane layer feature, so what's there to design???

Well, there is a bit:

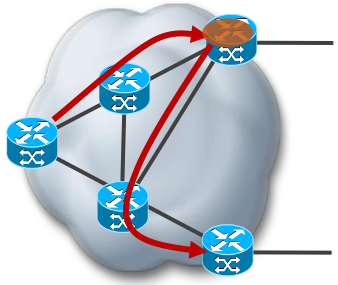
- BGP data plane convergence depends on how quickly the next-hop(s) converges (or is deleted), which boils down to

  Fast failure detection

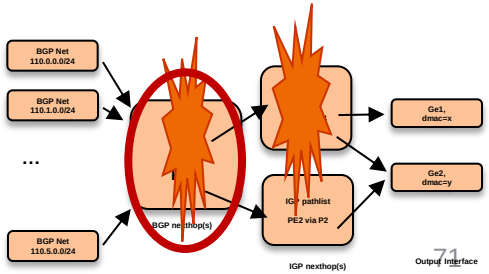
  Fast IGP convergence



- For PIC Edge, we need some form of tunnelling/encapsulation between edge devices



- For BGP PIC-Edge, we need to have an alternative/backup next-hop



BGP PIC Design I: Fast IGP Convergence

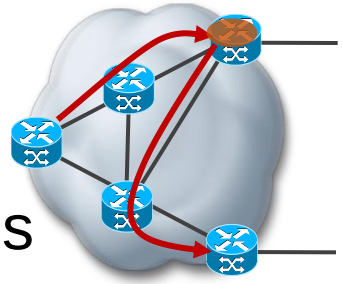
Details of IGP Convergence tuning is outside the scope of this session, but in principle

1. Provide for fast failure detection (BFD, carrier-delay)
   Avoid tuning down BGP hello/hold timers for failure detection
2. For IOS, tune OSPF/ISIS LSA- and SPF-throttle timers

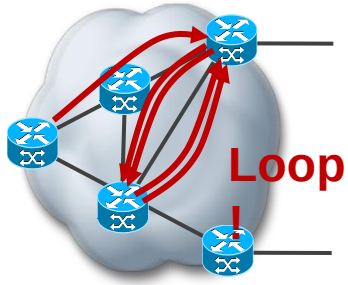
   Reduces convergence from ~5 down to <1 sec

   NX-OS and IOS-XR already tuned reasonably fast
3. Prioritize next-hop prefixes (i.e. PE loopback addresses)
   to make sure they converge before less important prefixes
   (i.e. link addresses or the like)
4. Keep the IGP slim and clean, use point-to-point adjacencies, carry customer routes in BGP
5. Evaluate Fast ReRoute techniques (like LFA-FRR) to further improve convergence

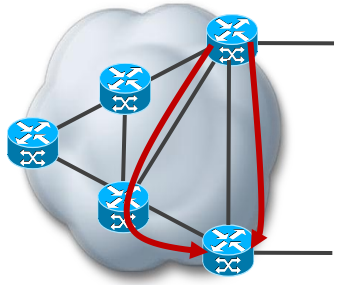
BGP PIC Design II: PE – PE Encapsulation



- Some BGP-PIC Edge convergence scenarios lead to edge device forwarding packets on to alternate edge, back via the core
- Core routers might be unaware of the failure (yet) and send packets back to the previous edge device, causing a loop

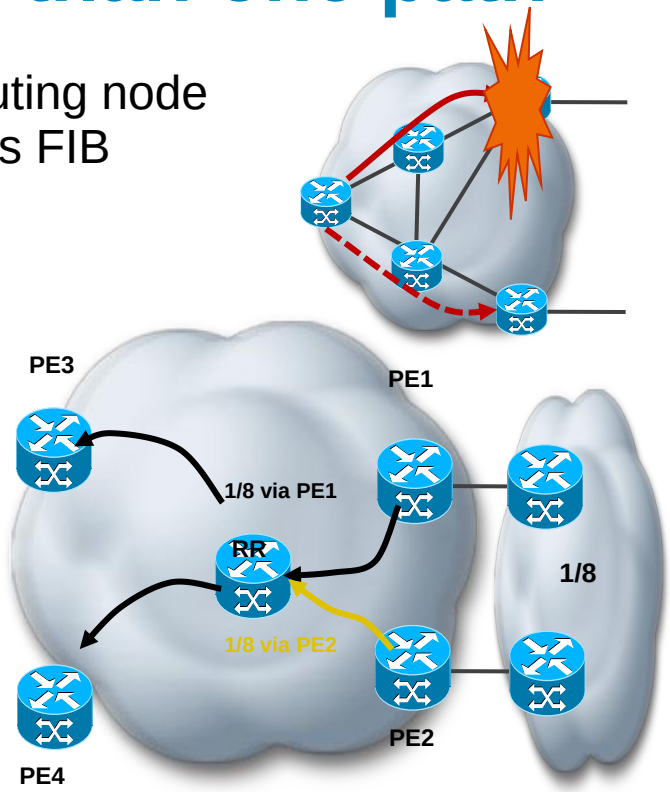


- Solution: Ensure there is no intermediate IP destination lookup, via means of encapsulation:
  - Direct adjacency between edges (physical link or GRE tunnel)
  - Using MPLS LSPs/Tunnels in the core



BGP PIC Design III: More than one path

- When a PE/next-hop goes away, the re-routing node needs already a backup/alternate path in its FIB
- This sounds rather obvious, but can be non-trivial in some scenarios

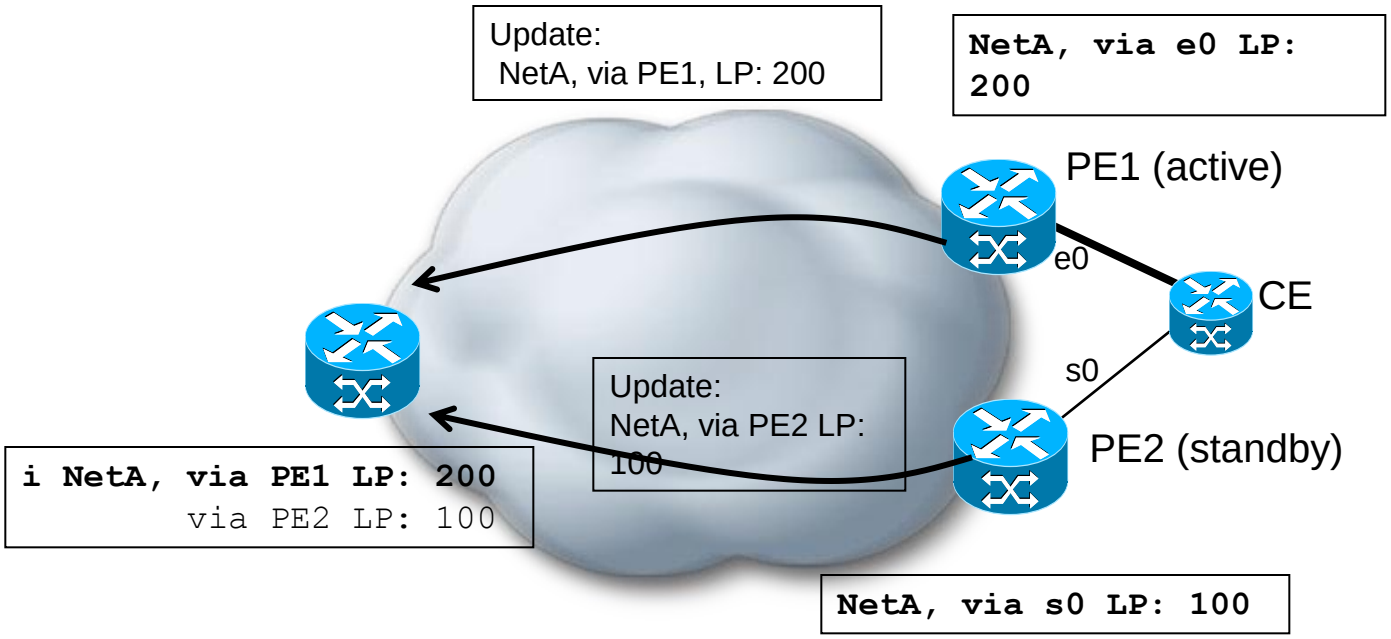


- Scenario 1: Route Reflectors

- Scenario 2: Active/Standby Routing Policies
  let's start with this one first...

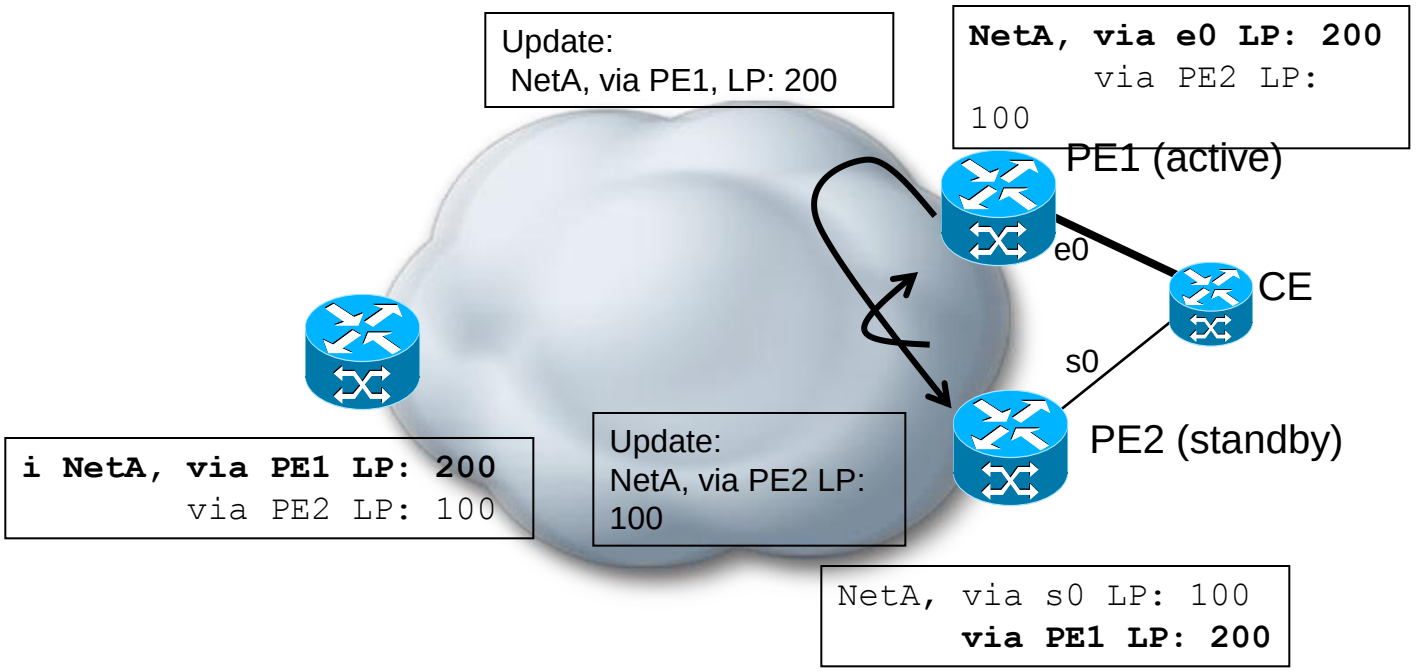
Scenario 2:
BGP Active/Standby – The Problem (1/3)

Initial State: CE just comes up



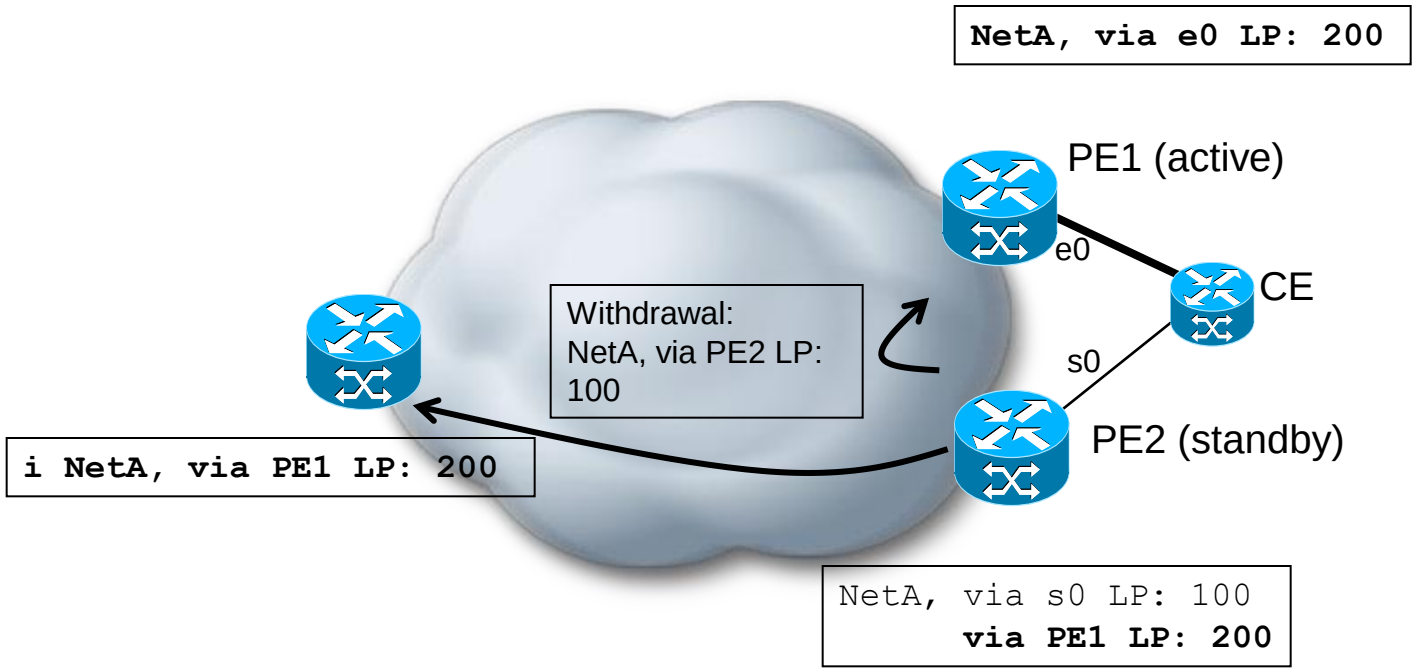
BGP Active/Standby – The Problem (2/3)

Initial State: CE just comes up



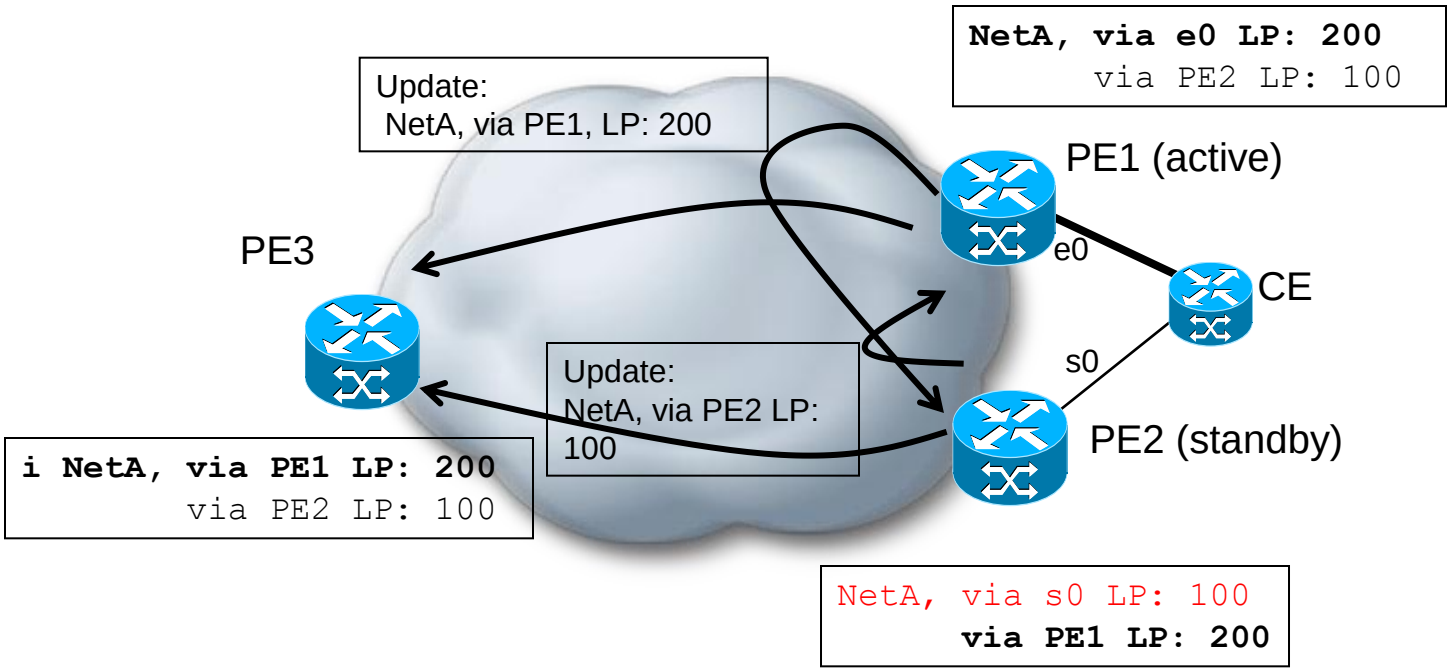
BGP Active/Standby – The Problem (3/3)

- PE2 withdraws its eBGP path as it is no longer best
- But now all other PEs are left with a single path – no alternate path for PIC-Edge to converge to!!!



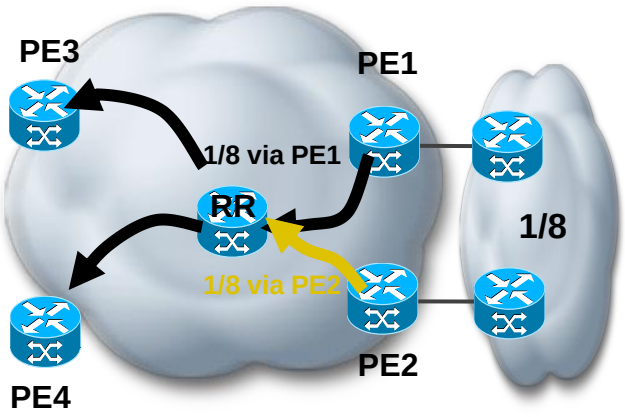
BGP Active/Standby – The Solution: BGP Best-External

- PE2 keeps advertising his best external path
- Requires no BGP protocol change (local change only)
- Availability: 12.2SRE/15.0S/MR, XE3.1 and XR3.9



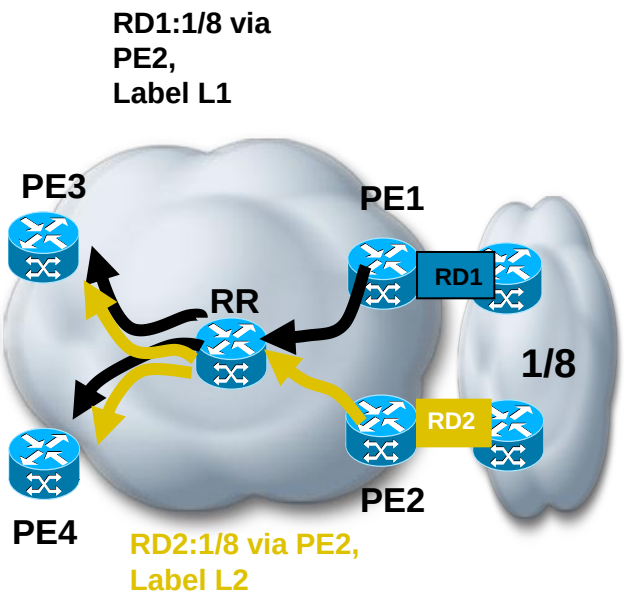
Scenario1:
iBGP Path Propagation and Reflection

- Regular BGP BestPath algorithm leads to an RR only reflecting one path, namely its best path
- Without explicit policy either hot-potato policy prevails (rule 8) or the lowest neighbor address (rule 13) is used as tiebreaker
- What can we do to propagate both paths to PE3/4?



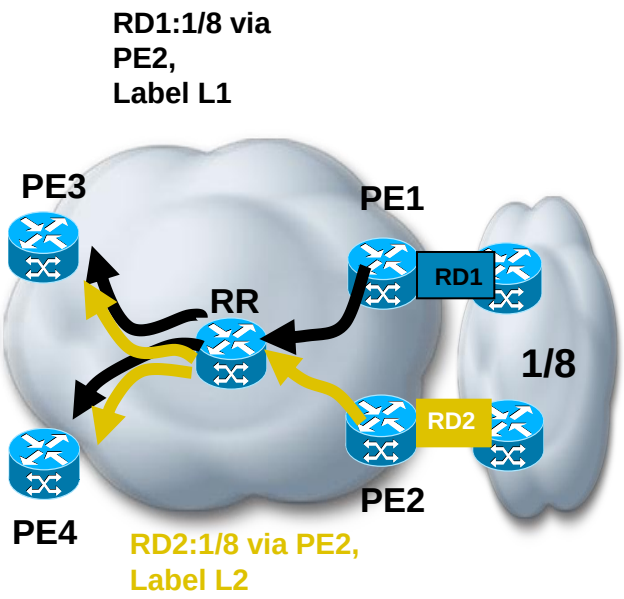
Path Diversity in L3VPN Environments

- Important sites are dual homed
- Unique RD allocation ensures both paths are learned, even through route reflectors
- For active/backup scenarios, “best-external” is required



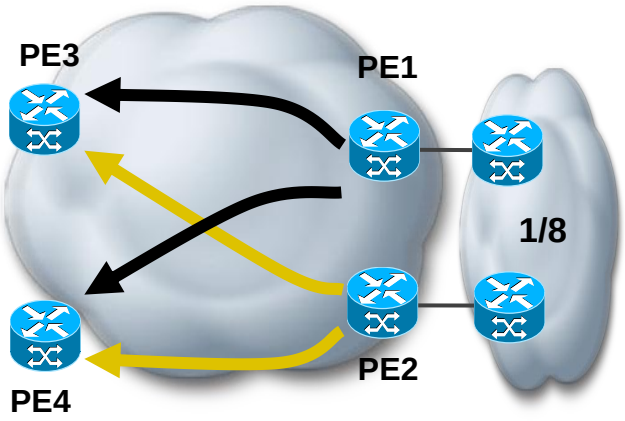
Solution 1: Internet in a VRF

- Unique RD allocation ensures both paths are learned, even through route reflectors
- Consider per-vrf or per-CE label allocation when advertising full Internet routing table within a VRF



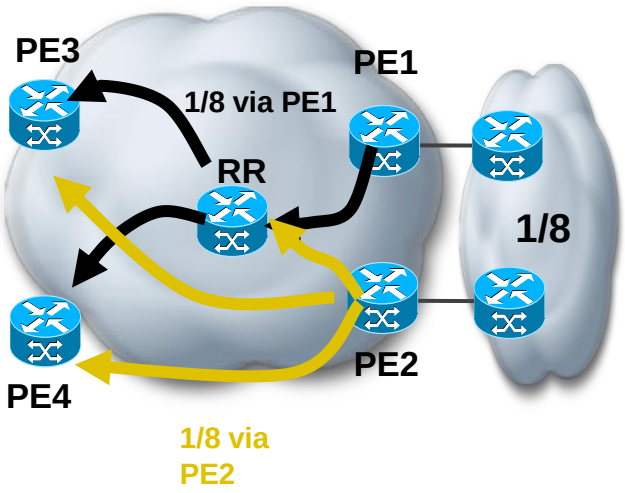
Solution 2: No RR

- Full iBGP mesh
- Yes, I am serious ☺, at least for reasonably sized and static environments



Solution 3: RR + partial iBGP mesh

- Done in practice by several operators
- Very specific and hence difficult to abstract any rule. Just know it exists and analyses the possibility.



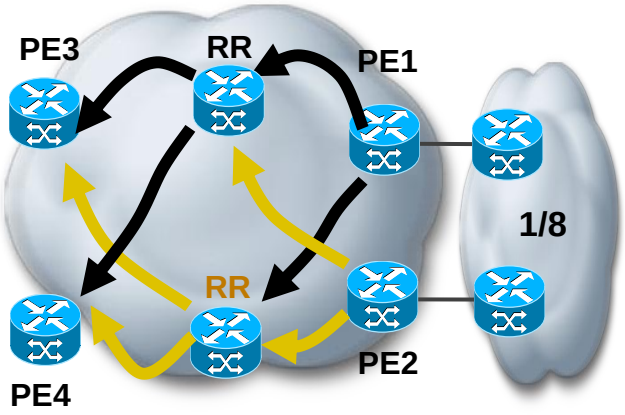
Solution 4: Engineered RR

- Some operators place their RR's in the topology to ensure they select different paths. Their PE's are clients to multiple RR's.

  the top RR is closer to PE1 and selects the black path

  the bottom RR is closer to PE2 and selects the blue path

  Above behaviour can also be achieved using specific BGP policies

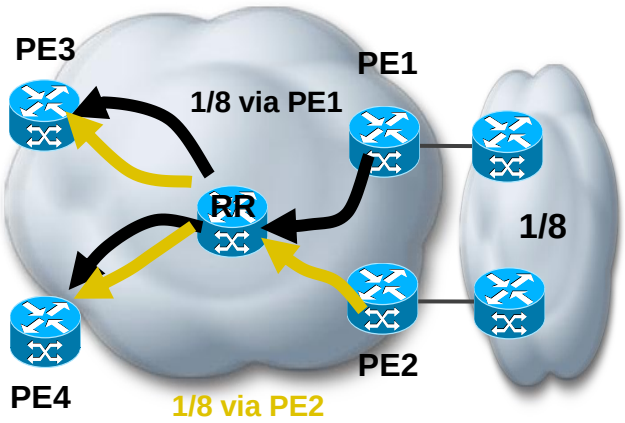


Solution 5: AddPath

- New Capability to allow a BGP speaker to advertise more than one path (“The holy grail”)

  Available in IOS-XR 4.0, IOS-XE 3.7, 15.2(4)S, 15.3T

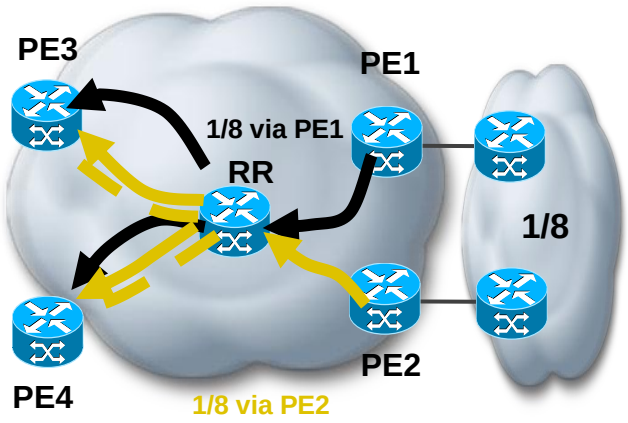
  Requires support for this functionality on RR and PEs



http://www.cisco.com/en/US/docs/ios-xml/ios/iproute_bgp/configuration/xe-3s/asr1000/irg-additional-paths.html

Solution 6: BGP Diverse Paths (aka "Shadow RR")

- New feature (IOS-XE 3.4S) allows a RR to advertise a 2nd best path
- Two deployment models:
  1. RR maintains two iBGP session to PEs (shown below)
     "Primary" connection advertises best path (PE1)
     "Secondary" connection (or secondary RR) advertises next-best-path (PE2)
  2. Dual RRs, first RR advertises best, the other RR the blue/2nd best path
- No update on PEs/RR-clients required

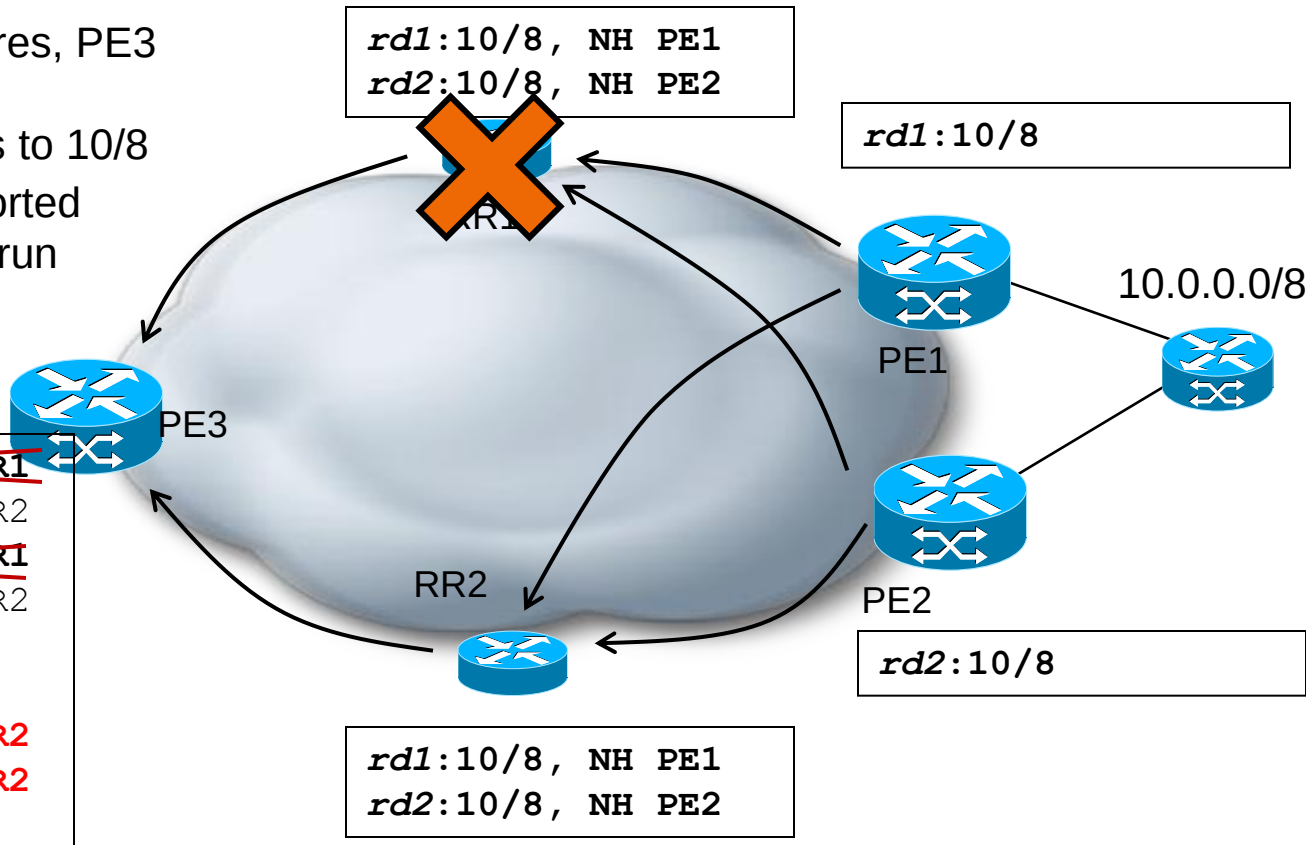


http://www.cisco.com/en/US/docs/ios/ios_xe/iproute_bgp/configuration/guide/irg_diverse_path_xe.html

L3VPN: RR Redundancy (IOS)
The Problem

Assumptions: Use of unique RDs per PE

1. RR1 goes down
   Traffic not affected as RR1 is not in forwarding path
2. After iBGP hold-time expires, PE3 purges routes from RR1
   ➔ VRF left without routes to 10/8
3. Alternative paths not imported until next import-scanner run (up to 15 seconds)
   not in 15M/S or SRE




```
> rd1:10/8, NH PE1, from RR1
  rd1:10/8, NH PE1, from RR2
> rd2:10/8, NH PE2, from RR1
  rd2:10/8, NH PE2, from RR2

vrf import
> rd3:10/8, NH PE1, from RR2
> rd3:10/8, NH PE2, from RR2
```

By default, BGP only imports a single best-path from each rd:prefix

L3VPN: RR Redundancy (IOS) The Solution

- Import more than the best path

```
router bgp ..
 address-family ipv4 vrf …
  maximum-paths [ibgp 2] import 4
```

or in 12.2SRE/15.0M

```
router bgp ..
 address-family ipv4 vrf …
  import path selection all
  import path limit 4
```

Be aware of addtl. memory consumption (~100 bytes/path)

```
> rd1:10/8, NH PE1, from RR1
  rd1:10/8, NH PE1, from RR2
> rd2:10/8, NH PE2, from RR1
  rd2:10/8, NH PE2, from RR2

vrf import
> rd3:10/8, NH PE1, from RR1
  rd3:10/8, NH PE1, from RR2
> rd3:10/8, NH PE2, from RR1
  rd3:10/8, NH PE2, from RR2
```

PE3

RR Failure

- RR Failure causes no immediate LoC as Forwarding Plane not affected
- BGP Hello's detect RR failure
- PE will switch to paths received from redundant RR
  - Make sure all paths are imported
- Goal: Minimize non-redundant state as much as possible

Designing (for) BGP PIC – Summary

- Work on the baseline – Improve your IGP convergence
- Use BFD for speedy eBGP failure detection
- Consider enabling MPLS to provide PE-PE tunnelling
- Ensure you have multiple paths – and keep this **always** in the back of your mind, whatever BGP designs/services you are coming up with

  This one is the hardest one to fix once solutions/etc. are deployed


CISCO