BGP Prefix Independent Convergence

draft-rtgwg-bgp-pic-00

Authors :

Ahmed Bashandy, Cisco Systems

Clarence Filsfils, Cisco Systems

Pradosh Mohapatra, Cisco Systems

Presenter :

Ahmed Bashandy

IETF85, Nov/2012

Atlanta, USA

Agenda

- Problem
- Solution
- Recovery from various failure scenarios

Agenda

- Problem
- Solution
- Recovery from various failure scenarios

What do we want to Achieve?

- *Serial* nature of reachability propagation

  ⇒BGP convergence is *inherently* slow

- Can we adjust the data structure in time complexity that does ***not*** depend on BGP prefixes?

  – If a path is lost or gained, modifications should be independent of the number of prefixes

  – If there is a recovery, recover as soon as possible independent of the number of prefixes

Objective

Make re-convergence after topology change <u>independent</u> of the number of BGP prefixes

Terminology

- Leaf: A prefix or local label container datastructure
- Path: a recursive or non-recursive path
  - May be primary or backup
- Pathlist: an array of paths
  - Each path carries its own pathindex
- OutLabel-Array: Array of outgoing labels and/or label actions*
  - A possibly different outlabel-array attached to each leaf
  - Each entry represents an outgoing label and/or label action for a path in the pathlist**
- Adjacency:
  - The L2 information to send a packet to a directly connected router

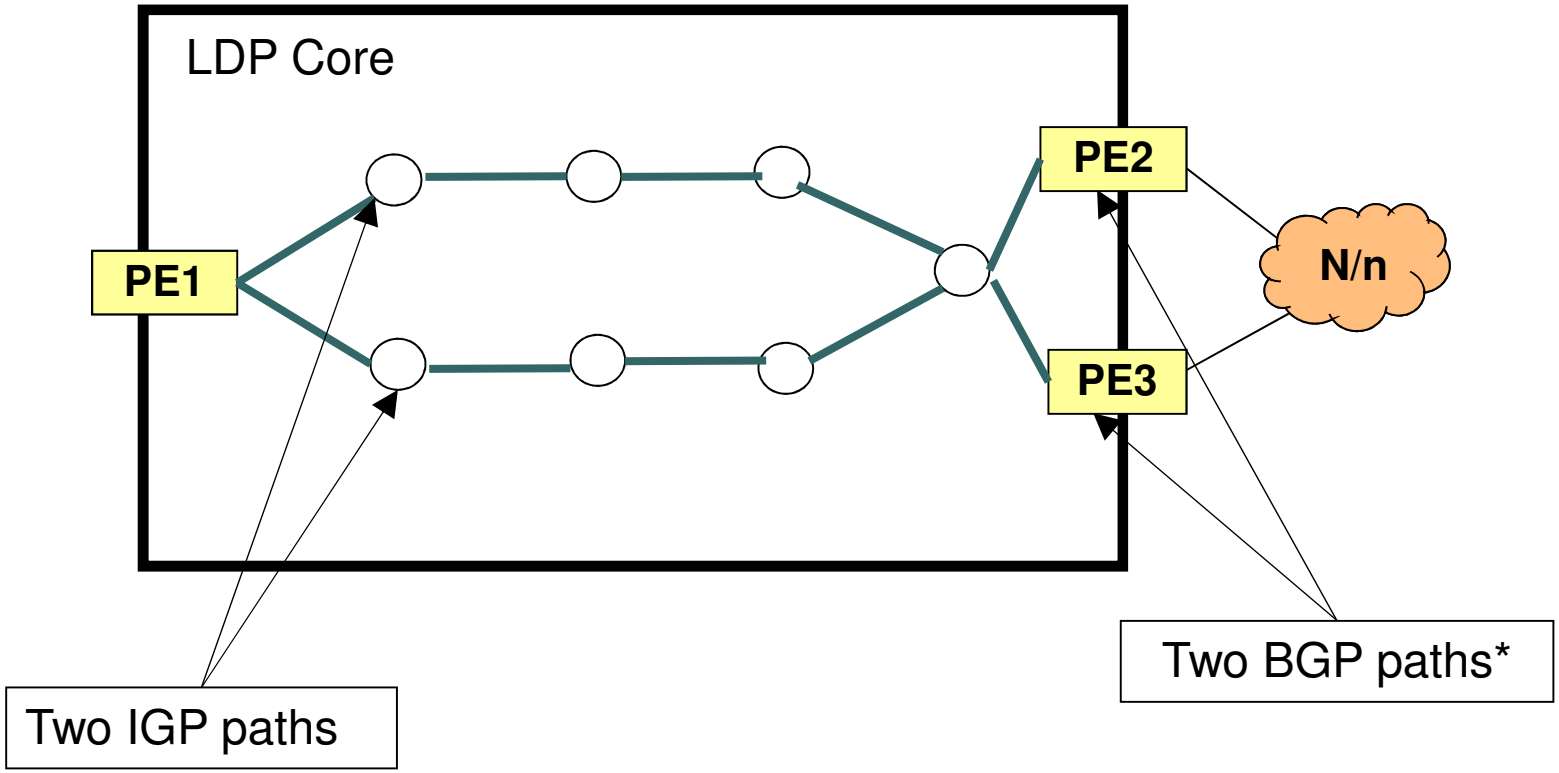
Agenda

- Problem
- Solution
- Recovery from various failure scenarios

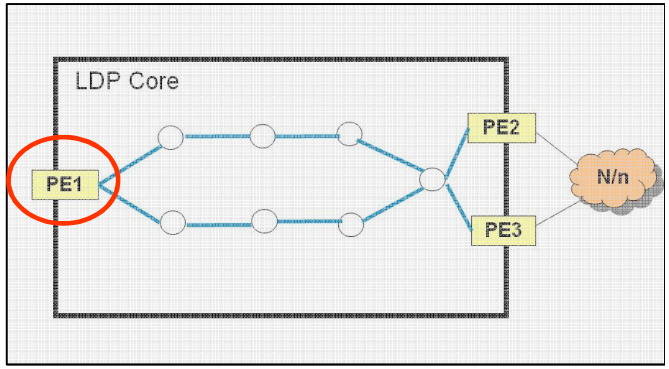
Basic Idea

- BGP-PIC is really a FIB feature
- *Hierarchical* and *shared* forwarding chains
- On topology changes
  - FIB modifies pathlists immediately without modifying leaves
  - Restore traffic very quickly
  - BGP modifies leaves later at a slower pace
- Behavior is internal to the router
  - Completely *transparent* to operator
  - *Incrementally* deployable

Sample Topology: VPN in LDP Core



Forwarding Chain on *Ingress* PE



VPN-L21
VPN-L31

VPN leaf 1

LDP-L11
LDP-L21

PE2
IGP leaf

PE2

PE2
PE3

Shared
BGP Pathlist
(even among VRFs)

NH1
NH2

Shared
IGP Pathlist

Adj1

Adj2

PE3

PE3
IGP leaf

VPN leaf 2

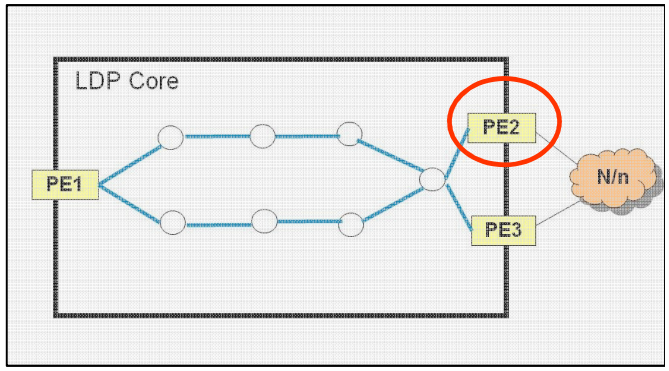
VPN-L22
VPN-L32

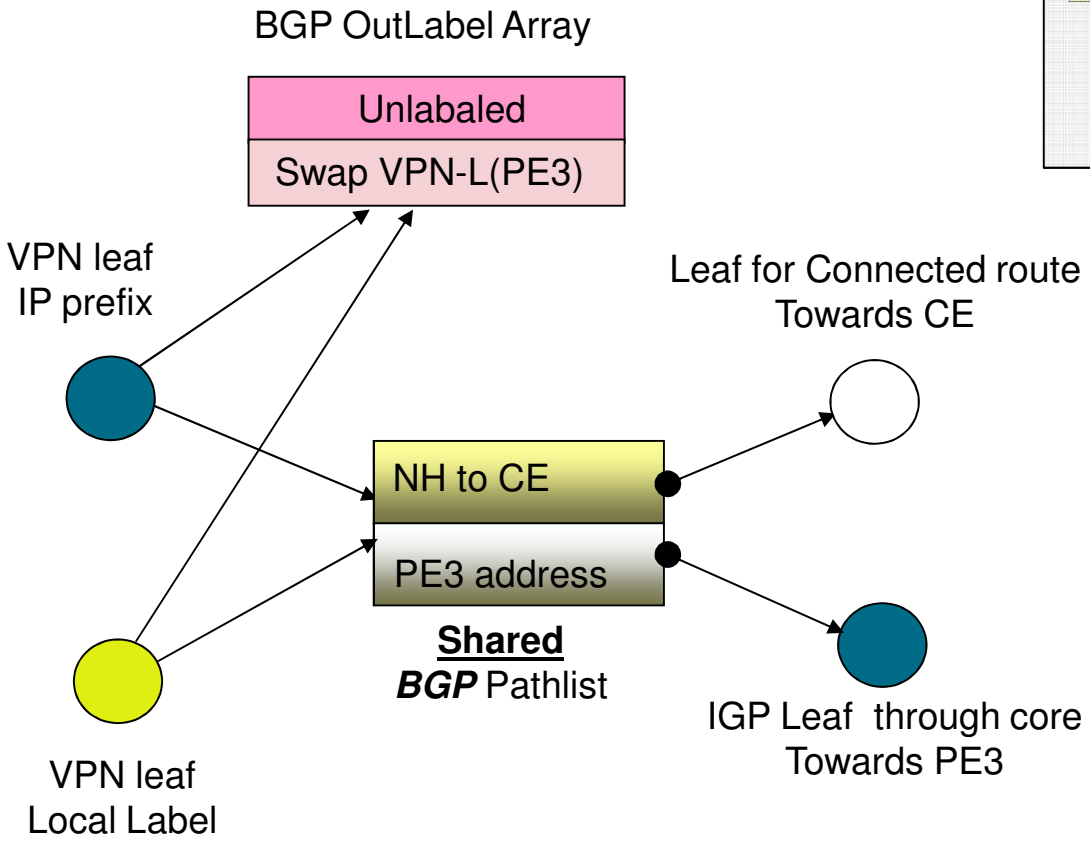
BGP OutLabel Array

LDP-L12
LDP-L22

IGP OutLabel Array

Forwarding Chain on *Egress* PE2

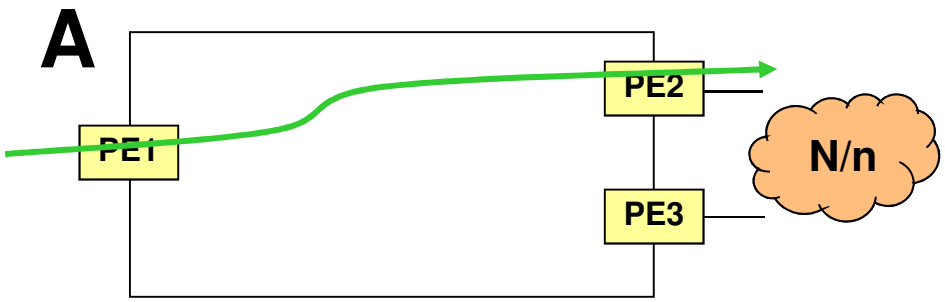




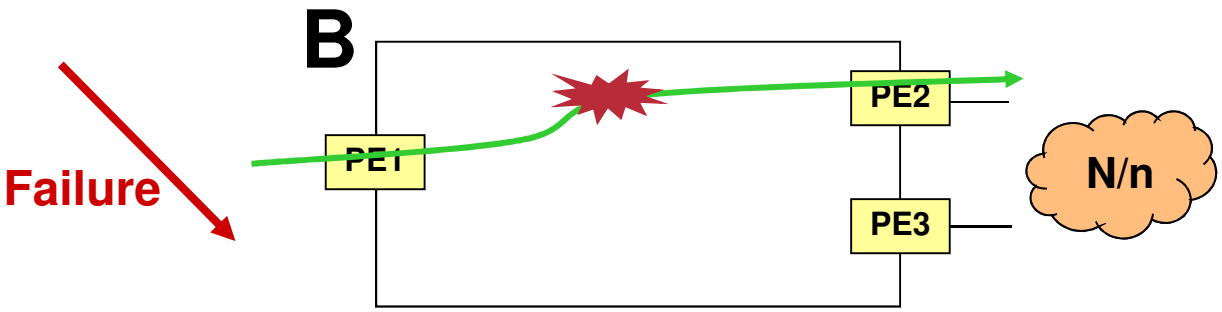
Agenda

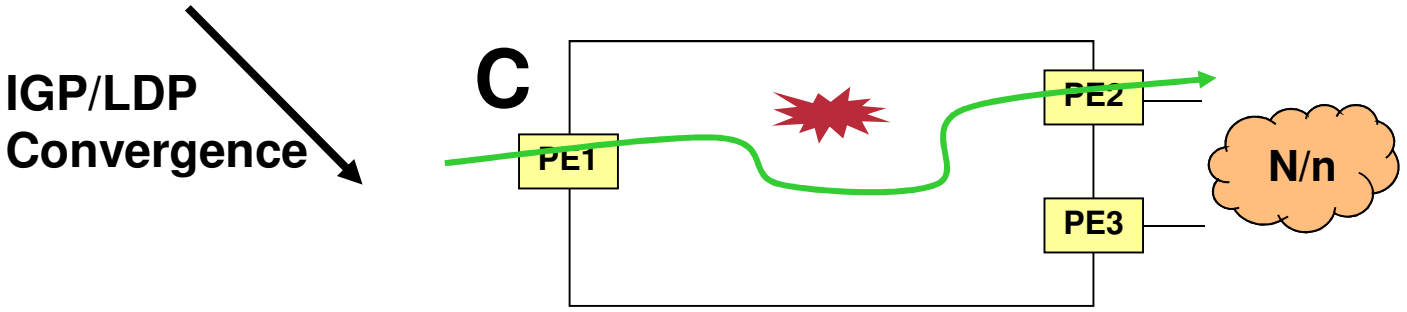
- Problem
- Solution
- Recovery from various failure scenarios

Core Link/Node Failure

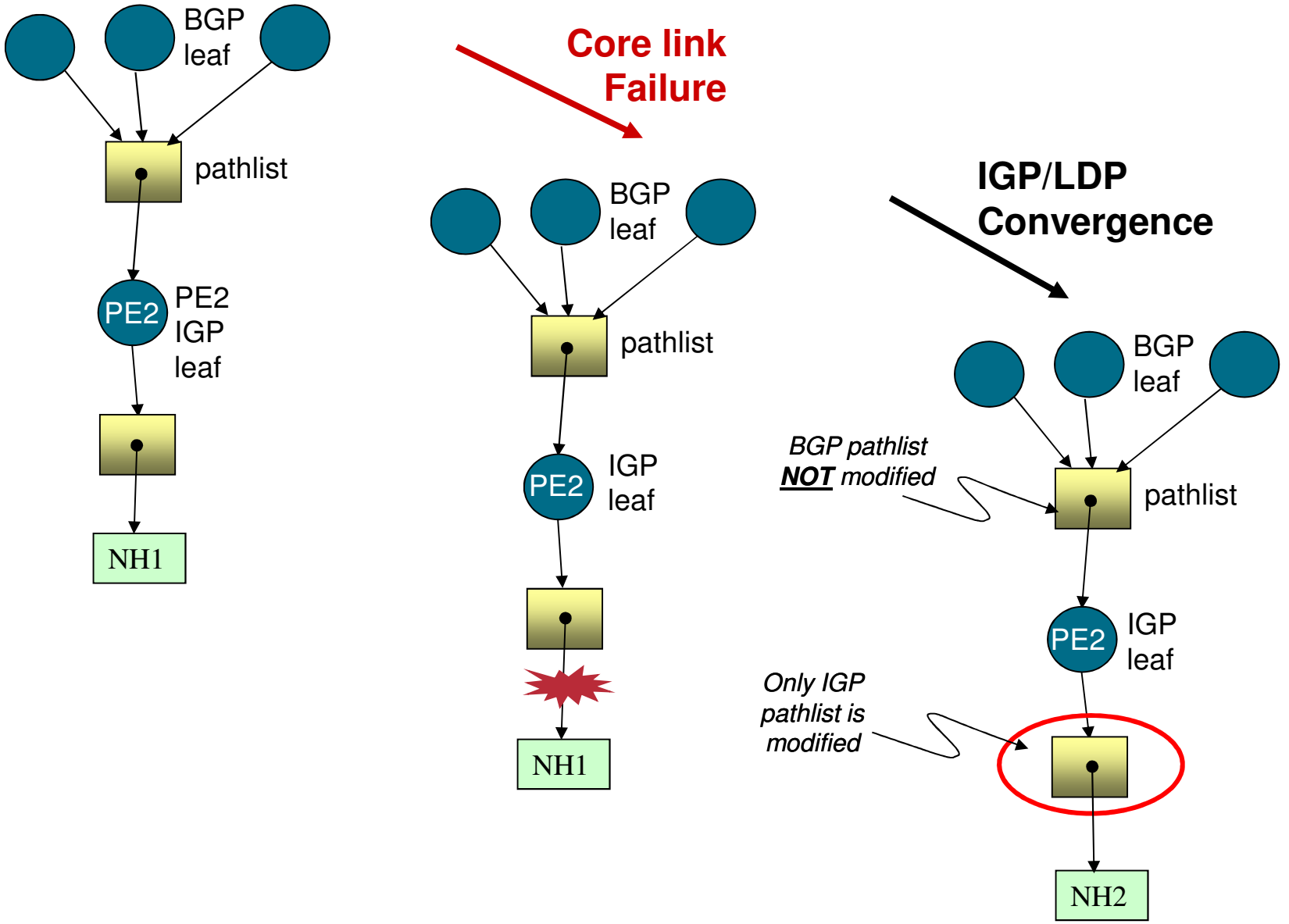


Works whether BGP is running In the core or not

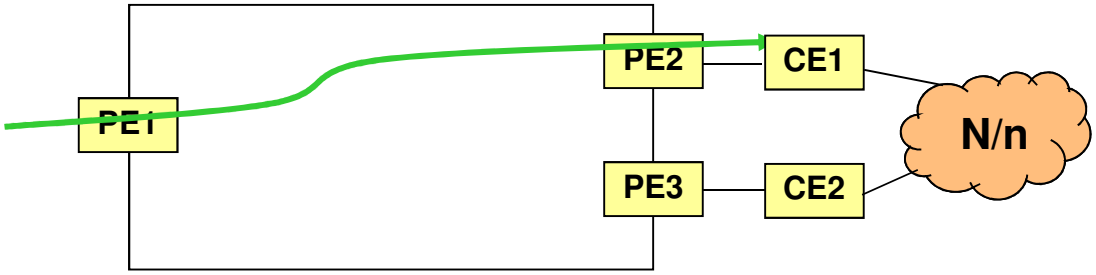




Core Link/Node Failure: FIB on PE1



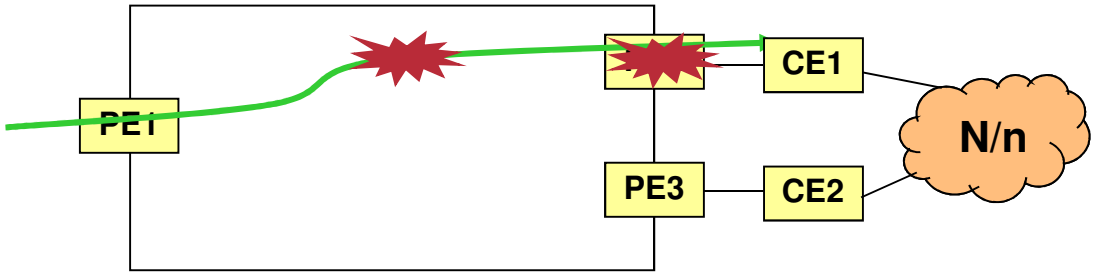
Unipath PE Node* Failure: Protection at *Ingress* PE



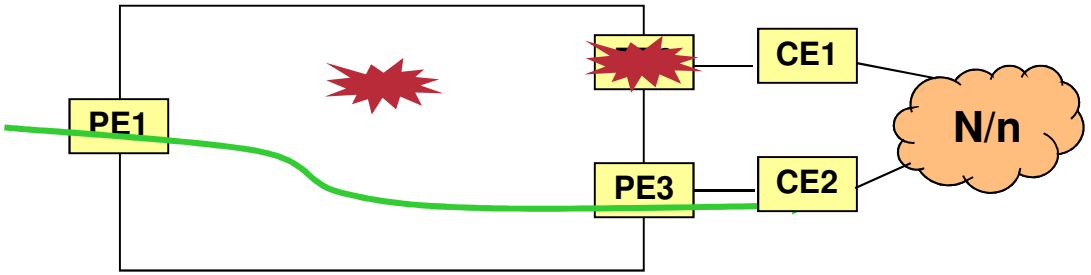
Works whether BGP is running in the core or not**

PE3 must be pre-programmed as a backup path

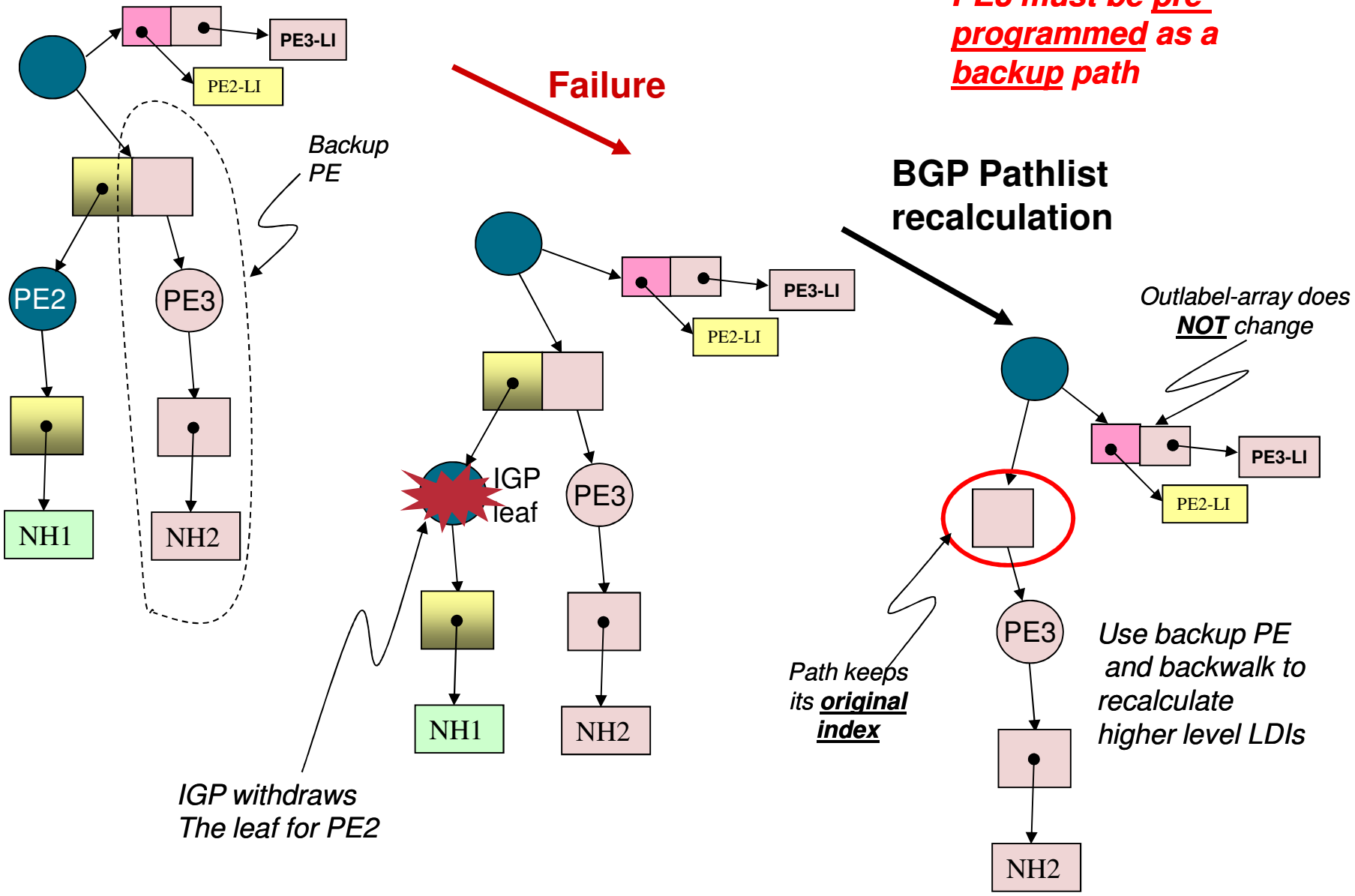
Core or Edge node Failure



FIB Correction

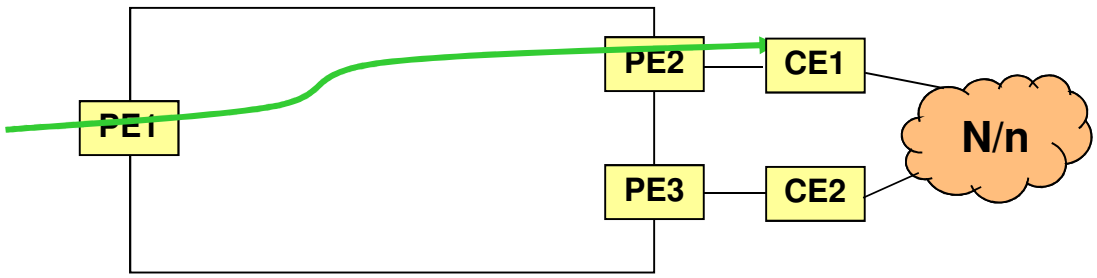


Unipath PE Node/PE-CE Link failure: FIB on PE1



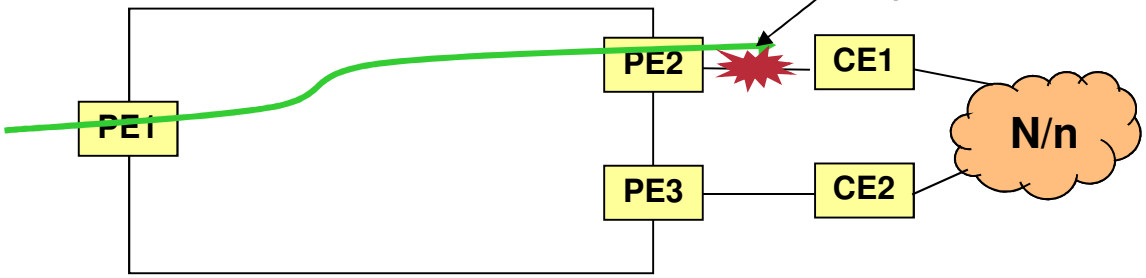
Unipath PE-CE Link Failure: Protection at ***Egress*** PE

More useful for BGP-free core

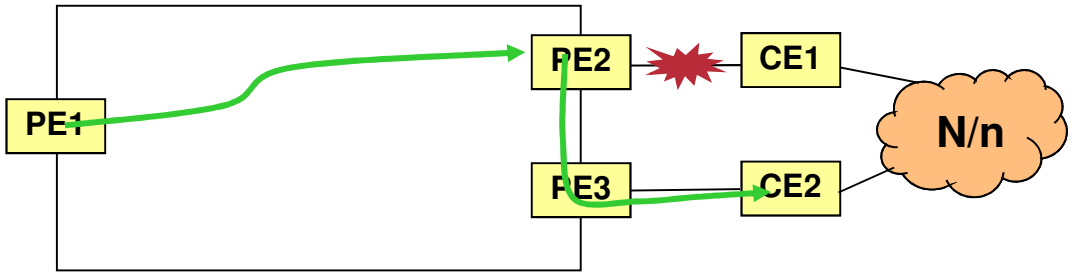


***Locally** detect failure*
50ms recovery !!

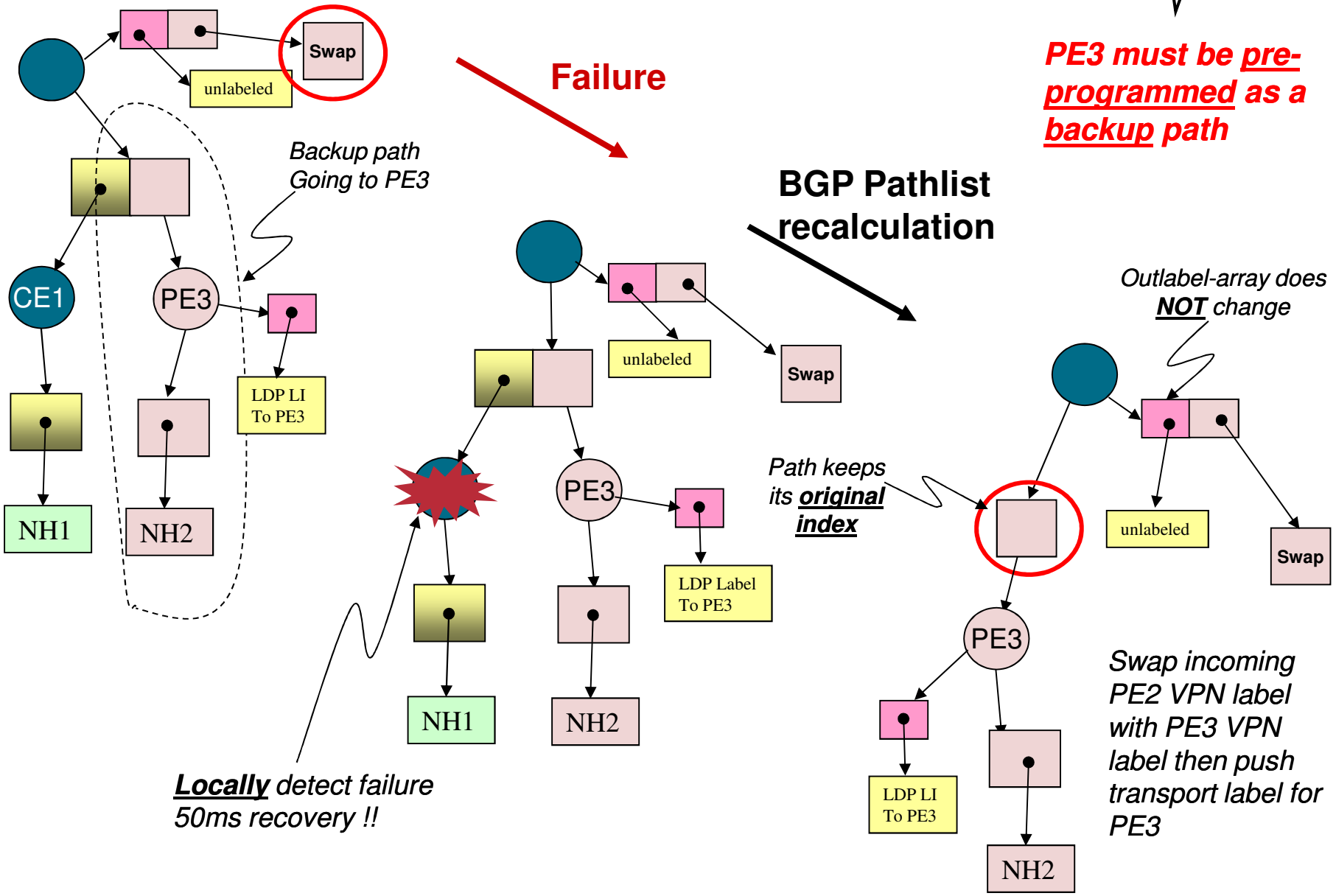
Failure



FIB Correction



Unipath PE-CE Link failure: FIB on PE2



Conclusion

- Single elegant design to handle many convergence/protection cases in a BGP prefix independent time