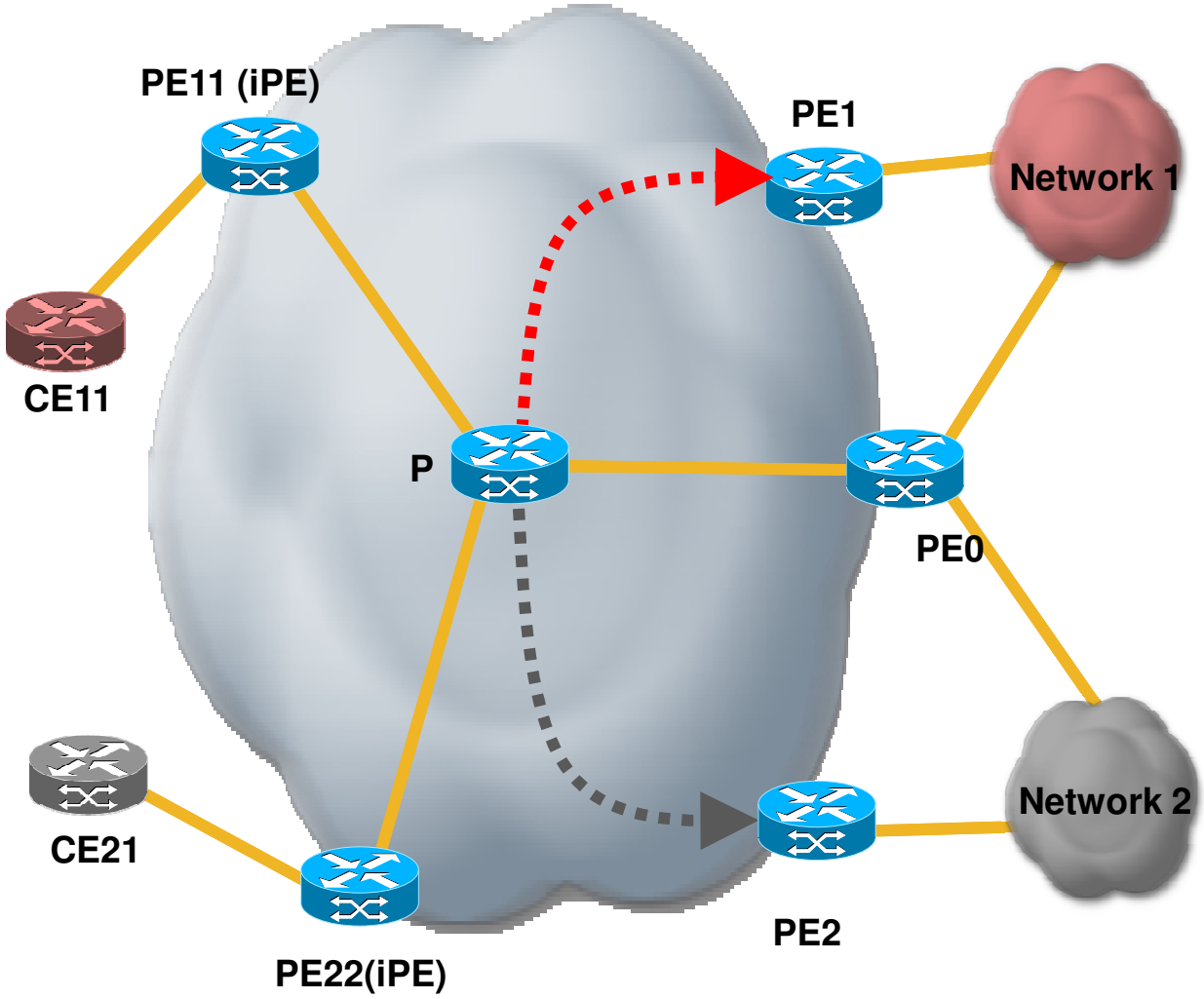
BGP Edge Node FRR

Ahmed Bashandy

Agenda

- Problem and Requirements
- 3 BGP-FRR Solutions
  - Solution 1: draft-bashandy-bgp-edge-node-frr-03
  - Solution 2: draft-bashandy-bgp-frr-vector-label-00
  - Solution 3: draft-bashandy-bgp-frr-mirror-table-00
- Comparison (if we have time)
  - Qualitative Comparison
  - Quantitative Comparison
  - Two main advantages and disadvantages of each solution

Problem

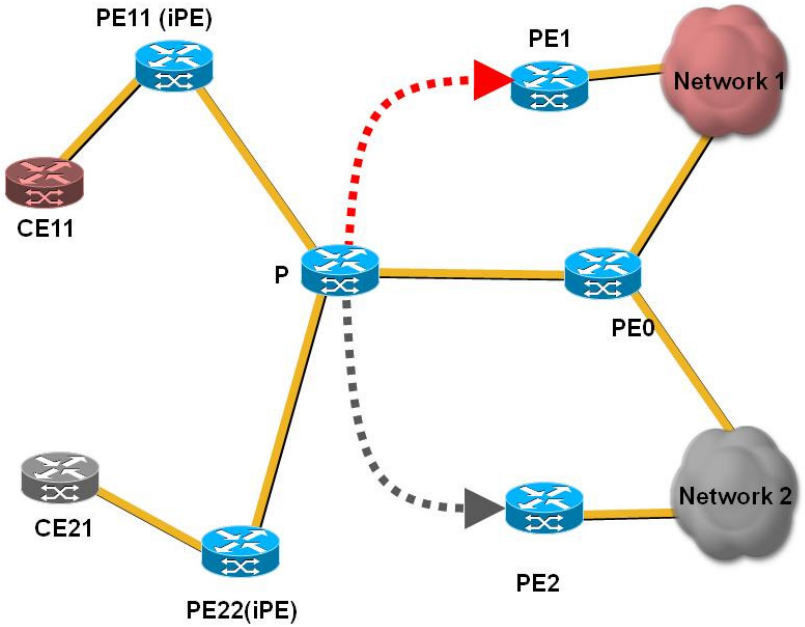


- PE0 is primary for both **Red** and **Grey**.
- P router redirects traffic to the ***correct*** repair PE
  - PE1 for **Red**
  - PE2 for **Gray**
- Correct ***BGP label*** must exist for correct forwarding on repair PE

Main Requirements

- Must have
  - Core remains BGP free
  - Minimize provisioning
  - Correct BGP label must exist when repairing
  - No multi-label lookup at steady state
- Should Have
  - IGP FRR not required
  - Works with MPLS and IP core
  - Minimal churn in the network
  - Minimal additional state in the network
  - Resistance to misconfig
- Good to have
  - No multilabel lookup even during repair
  - No churn in the network, only routers willing to participate gets some churn

Terminology



- Protected PE (**pPE**): A PE protected by BGP FRR (E.g. router PE0)
- Protected next-hop: (**pNH**): It is an IPv4 or IPv6 host address belonging to the protected egress PE. Traffic tunneled to this IP address will be protected by BGP FRR
- Repair PE (**rPE**): It is an egress PE other than the primary egress PE that can reach the protected prefix P/m through an external neighbor (E.g. routers PE1 and PE2)
- Repair next-hop (**rNH**): It is an IPv4 or IPv6 host address belonging by to the repair PE
- Repairing P router (**rP**): A core router that attempts to restore traffic when it detects, through local means, that the primary egress PE is no longer reachable without waiting for IGP or BGP to re-converge (E.g. router P)
- Ingress PE (**iPE**): A PE router that receives external traffic and forwards it outside the AS through a pPE (e.g. routers PE11 and PE22)

Overview of the Solutions

Scalable BGP FRR Protection against Edge Node Failure

draft-bashandy-bgp-edge-node-frr-03

Authors :

Ahmed Bashandy, Cisco Systems
Burjiz Pithawala, Cisco Systems
Keyur Patel, Cisco Systems

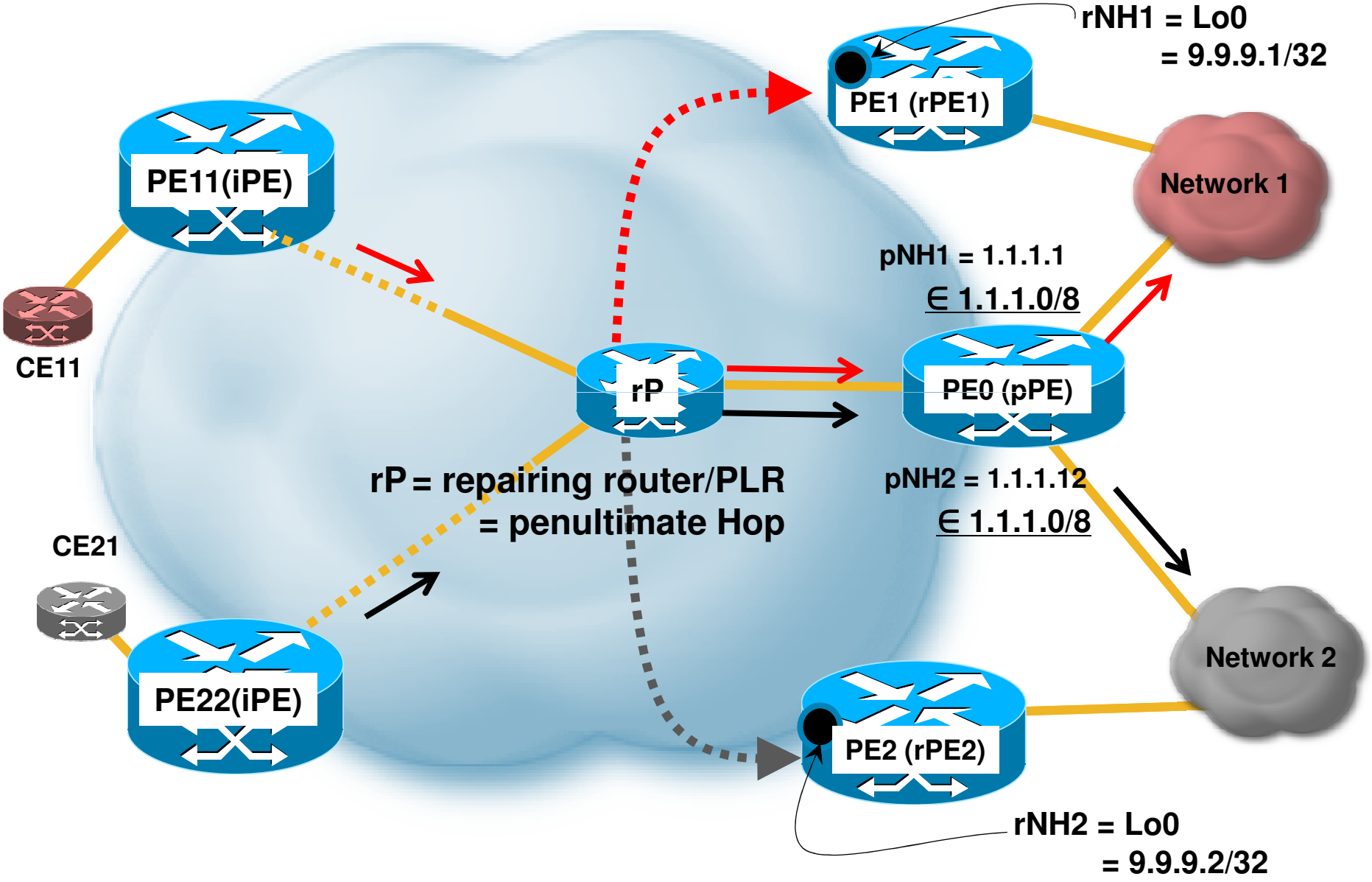
Presenter :

Ahmed Bashandy

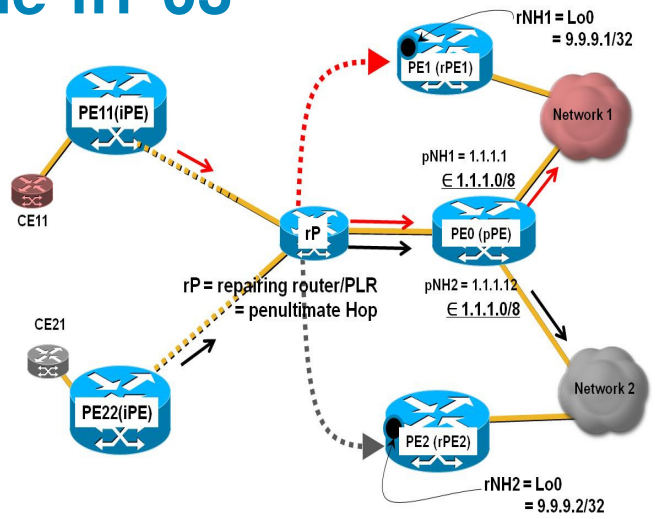
IETF85, Nov/2012

Atlanta, USA

Solution 1: One pNH per Protected-Repair PE Pair

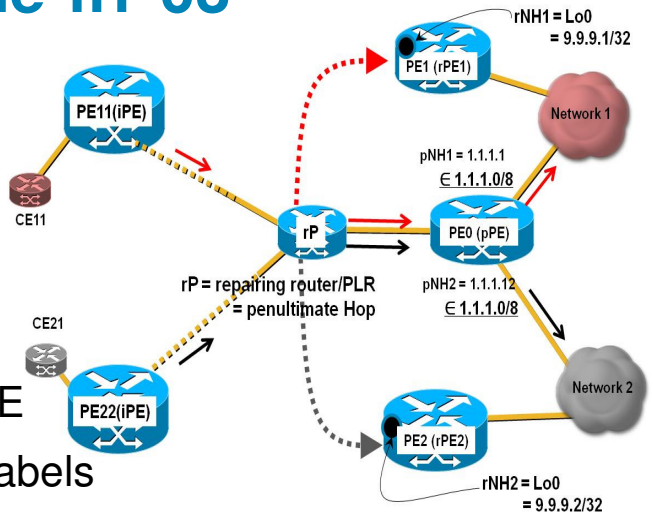


Solution 1: draft-bashandy-bgp-edge-node-frr-03
Control Plane



- **rPE:**
  - Allocates **a repair label rL** per CE
  - Advertises **rL** with protected prefixes
- **pPE:**
  - Allocates **a distinct pNH** for all prefixes protected by the **same rPE**
  - Advertise/Re-advertises prefixes with (**pNH**, **rL**) to **iPEs**
  - Advertises (**pNH**, **rNH**) to **rP**
- **rP:**
  - Advertises **pNH** with **the maximum metric**
  - Programs alternate path – label swap pNHL → rNHL

Solution 1: draft-bashandy-bgp-edge-node-frr-03
Data Plane



- **iPE**:
  - Pushes three labels: **vpnL**, **rL***, **pNHL**
- **Normal working conditions – steady state**
  - **rP**: pops two labels (**rL**, **pNHL**) and delivers packet to pPE
  - **pPE** will pop two labels if **rP** is not capable of popping 2 labels
  - Other nodes: standard behavior
- **pPE failure event – transient state**
  - **rP:**
    - swaps **pNHL** with **rNHL**
    - Re-routes traffic to **rNH**
  - **rPE:**
    - Receives traffic with **rL** as top label**
    - pops two labels (**rL, vpnL**),
    - looks up **rL** and forwards to the correct CE
  - Other nodes: standard behavior

BGP FRR Protection against Edge Node Failure Using Vector Labels

draft-bashandy-bgp-frr-vector-label-00

Authors :

Ahmed Bashandy, Cisco Systems
Maciek Konstantynowicz, Cisco Systems
Nagendra Kumar, Cisco Systems

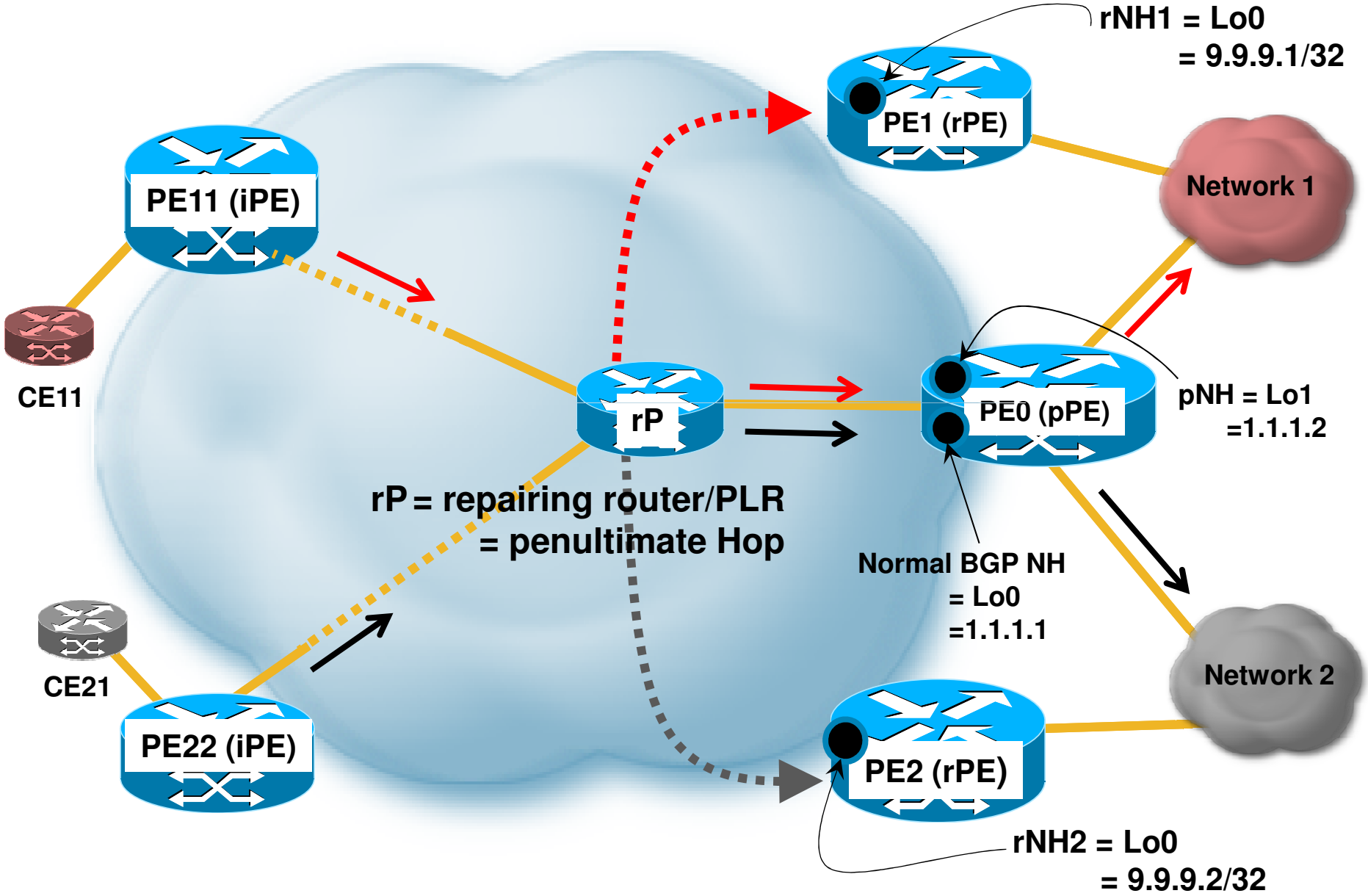
Presenter :

Ahmed Bashandy

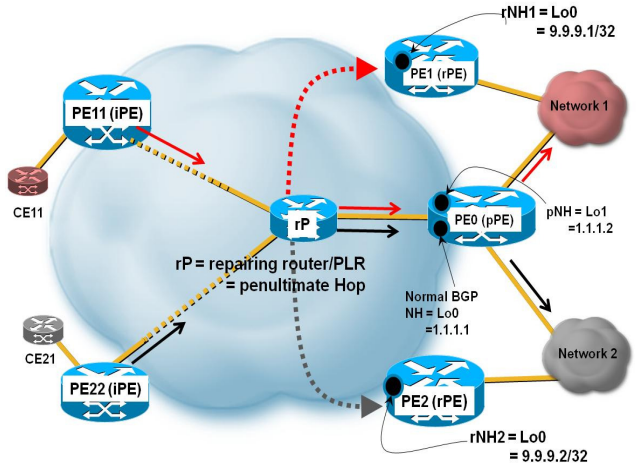
IETF85, Nov/2012

Atlanta, USA

Solution 2: draft-bashandy-bgp-frr-vector-label-00



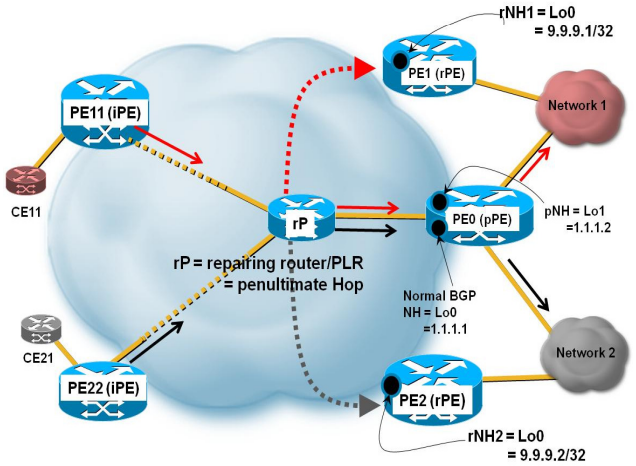
Solution 2: draft-bashandy-bgp-frr-vector-label-00
Control Plane



- **rPE:**
  - Allocates **a repair label rL** per CE
  - Advertises **rL** with protected prefixes
- **pPE:**
  - Allocates a **a vector label vL** for **every rPE**
  - Configure/auto-assign* a single protected next-hop **pNH** for the entire router
  - Advertises (***rNH, vL)*** binding to **iPE**s
    - e.g. can be done similar to biscuit tunnel “rfc5512”
  - Advertises (***pNH, rNH, vL)*** binding to **rP**
- **rP**:
  - Advertises **pNH** with **the maximum metric**
  - A separate **label context per pNH (i.e. per pPE)**
  - In the context of **pNH**
    - Programs repair path: *swap* **vL → rNHL**

Solution 2: draft-bashandy-bgp-frr-vector-label-00
Forwarding Plane

- **iPE**:
  - For each protected prefix
    - Chooses the **rPE*** and the corresponding **vL** and **rL**
  - Pushes four labels: **vpnL**, **rL**, **vL**, **pNHL****
- **Normal working conditions – steady state**
  - **rP**:
    - pops three labels (**rL**, **vL**, **pNHL**)
    - Delivers the packet to **pPE**
  - Other nodes: standard behavior
- **pPE failure event – transient state**
  - **rP**: Looks up **vL** in the label context of **pNH**
    - Re-routes traffic via repair path to **rNH**
      - swaps **vL** with **rNHL**
      - Send packet to **rNH**
  - **rPE:**
    - Receives traffic with **rL** as top label
    - pops two labels (**rL**,**vpnL**),
    - looks up **rL**
    - and forwards to the correct CE
  - Other nodes: standard behavior



BGP FRR Protection against Edge Node Failure Using Table Mirroring with Context Labels

draft-bashandy-bgp-frr-mirror-table-00

Authors :

Ahmed Bashandy, Cisco Systems
Maciek Konstantynowicz, Cisco Systems
Nagendra Kumar, Cisco Systems

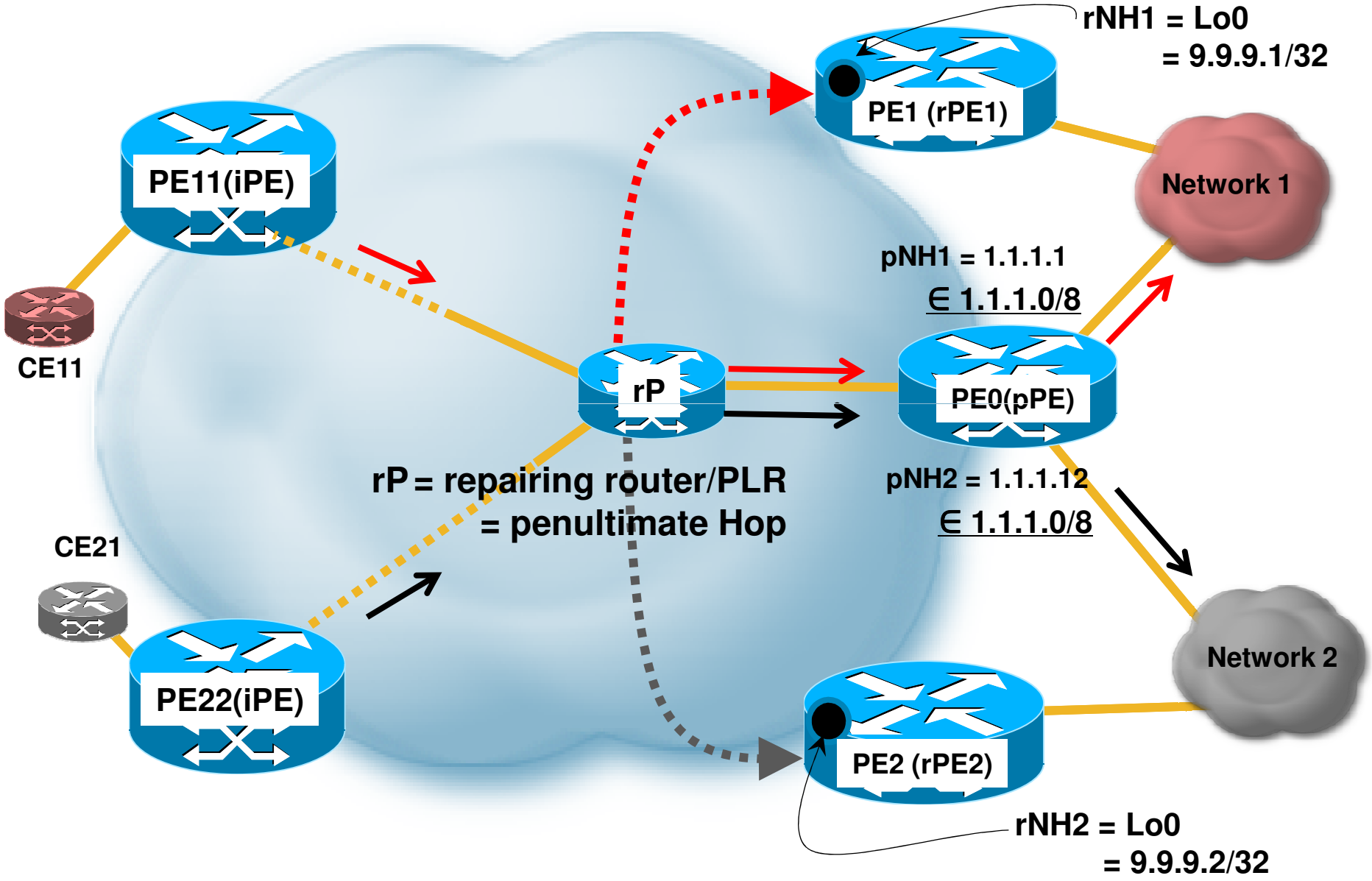
Presenter :

Ahmed Bashandy

IETF85, Nov/2012

Atlanta, USA

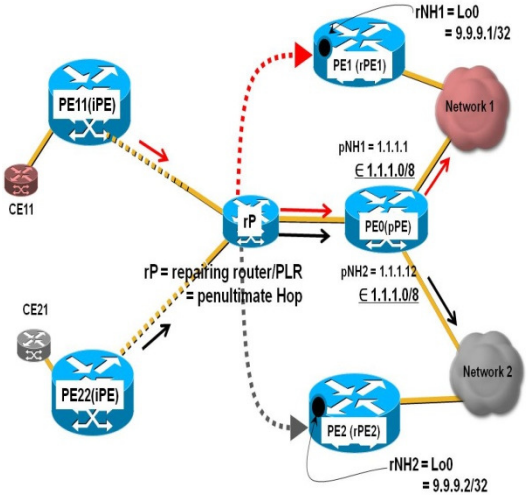
Solution 5: draft-bashandy-bgp-frr-mirror-table-00



Solution 5: draft-bashandy-bgp-frr-mirror-table-00 Control Plane

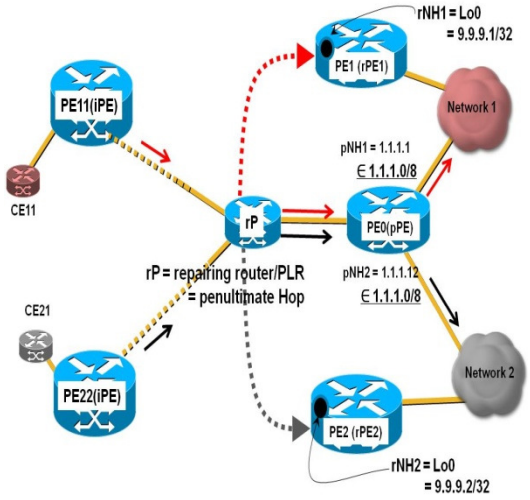
NOTE: Most of pPE and rPE configuration can be automated.

- **pPE:**
  - configure **a distinct pNH** for **every distinct rPE**
  - Advertise **pNH** as the BGP next-hop for all prefixes protected by **rPE**
- **rPE:**
  - configure **repair function** for prefixes with **pNH as their BGP next-hop**
  - Allocates **a distinct context label cL** for every **distinct pNH**
  - Mirrors prefixes with **pNH** as a BGP next-hop in the label context identified by **cL**
  - Advertises the **pNH** with high metric (e.g. max-metric – 1)
  - Advertises (**pNH**, **rNH, cL**) to **rP***
  - Advertises the label "**cL**" for **pNH** instead of the usual implicit NULL**



Solution 5: draft-bashandy-bgp-frr-mirror-table-00 Forwarding Plane

- **Normal working conditions – steady state**
  - All nodes: standard behavior
- **pPE failure event – transient state**
  - **rP:**
    - Pops the label for **pNH**
    - Pushes **cL** and re-routes traffic to **rPE**
  - **rPE:** Uses **cL** to identify the mirrored label
    - Looks up **vpnL** in the context of pPE,
    - finds the match with its local VPN table,
    - forwards the packet
  - **All other nodes:** standard behavior



Q & A

The Comparison

General Notes

- Solution 1, 2, 3
  - *Can co-exist in the same network and even on the same **pPE**, **rPE**, and **rP**
    - This can be done by having different attributes for advertising (**pNH, rNH, vL**), (**pNH**, **rNH**, **rL**)
  - No need for LFA support in the core
  - No need for explicit routing: Work in both MPLS and IP core

Label Swap/Pop/push Comparison at Steady State

Node	draft-bashandy-bgp-edge-node-frr-03	draft-bashandy-bgp-frr-vector-label-00	draft-bashandy-bgp-frr-mirror-table-00
pPE	No Change: Pop (1)	No Change: Pop (1)	No Change: Pop (1)
PHP	Pop 2	Pop 3	No change: Pop (1)
iPE	Push 3	Push 4	No Change: Push (2)

Label Swap/Pop/push Comparison at Failure

Node	draft-bashandy-bgp-edge-node-frr-03	draft-bashandy-bgp-frr-vector-label-00	draft-bashandy-bgp-frr-mirror-table-00
rPE	Pop (2)	Pop (2)	Pop (2)
rP	No Change: Swap (1)	Swap (2)	Swap (2)

Qualitative Comparison

Factor	draft-bashandy-bgp-edge-node-frr-03	draft-bashandy-bgp-frr-vector-label-00	draft-bashandy-bgp-frr-mirror-table-00
Loop Free re-routing on failure	Yes	Yes	Yes
Core remains BGP-free	Yes	Yes	Yes
Simple config	Medium because of the need to configure non-overlapping IP range	Yes	NO: because of the need to configure non-overlapping IP range or distinct pNHs on rPE and pPE
Correct VPN label when repairing	Yes	Yes	Yes
Immunity to misconfig	Yes	Yes	Yes/No*
Per-prefix label allocation	Yes	Yes	No because of mirroring

Qualitative Comparison

Factor	draft-bashandy-bgp-edge-node-frr-03	draft-bashandy-bgp-frr-vector-label-00	draft-bashandy-bgp-frr-mirror-table-00
Works with any tunneling protocol	Yes	Yes	Yes*
Single label lookup during **steady state**	Yes	Yes	Yes
Single label lookup **during repair****	Yes	No	No
Minimal Churn in the network	Medium	Yes	If the pNH are configured, then no churn
Extra state in the core	Medium***	Small	Medium***
Works on networks **without** IP FRR or TE FRR	Yes	Yes	Yes

Qualitative Comparison

Factor	draft-bashandy-bgp-edge-node-frr-03	draft-bashandy-bgp-frr-vector-label-00	draft-bashandy-bgp-frr-mirror-table-00
No Churn except on nodes participating in the solution	No*	Yes	If the pNH are configured, then no churn
No New Code on Penultimate Hop	No	No	No**
No New Code on Ingress PE	No	No	Yes
No New Label Pop Semantics	No	No	Yes
Repair is not another path to the same FEC***	Yes	Yes	Yes

Qualitative Comparison

Factor	draft-bashandy-bgp-edge-node-frr-03	draft-bashandy-bgp-frr-vector-label-00	draft-bashandy-bgp-frr-mirror-table-00
Ability to support per-CE label binding on the primary path	Yes	Yes	Yes
Forwarding Plane Complexity	Simple: The only additional complexity is popping 2 labels instead of 1	Medium **during repair only** because of vector label lookup	Medium **during repair only** because of context label lookup
Summary (Green = 1 Yellow = 2, Red = 3)	**30**	**28**	**30**

Quantitative Comparison: Factors

- **Additional Entries in the IP table**
  - The additional IP prefixes inserted in the FIB or RIB because of employing the BGP FRR scheme
- **Additional Entries in the label table**:
  - The extra labels inserted in the FIB
- **Additional BGP Mapping entries**:
  - Certain mappings are needed for BGP on a PE to maintain and advertise certain attributes to other PEs.
  - For example,
    - when rPE allocates a repair label "**rL**" on per-CE basis, then it needs to maintain one mapping entry
    - All prefixes reachable via this CE point to this mapping entry
    - This way BGP can advertise "**rL**" as an optional attribute to other PEs

Quantitative Comparison: Parameters

- **N(pPE)**: Total Number of protected PEs in the network
- **N(rPE/pPE)**: Average number of repair PEs protecting the routes on a given protected PE.
- **N(pPE/rPE)**: Average the number of protected PEs that a given repair PE protects
- **N(VPN/pPE)**: Average number of VPNs connected to a protected PE
- **N(rPE/VPN)**: Average number of repair PEs needed to protect all the routes belonging to a single VPN on a **pPE**.
  - For example, Suppose a VPN has 1000 prefixes and connected to the protected router PE0.
  - Suppose that 500 of the prefixes are reachable also via PE1 and the other 500 prefixes are also reachable via PE2
  - In that case, on the protected PE, N(rPE/VPN) = 2
- **N(VPN/rPE)**: Average number of VPNs connected to a repair PE

Quantitative Comparison: Formulas for pPE

Solution	Extra IP FIB	Exra LFIB	Extra BGP mappings
Solution 1: draft-bashandy-bgp-edge-node-frr-03	N(pPE) × N(rPE/pPE)	N(pPE) × N(rPE/pPE)	N(rPE/pPE)
Solution 2: draft-bashandy-bgp-frr-vector-label-00	1 × N(pPE)	1 × N(pPE)	N(rPE/pPE)
Solution 3: draft-bashandy-bgp-frr-mirror-table-00	N(pPE) × N(rPE/pPE)	N(pPE) × N(rPE/pPE)	N(rPE/pPE)

Quantitative Comparison: Formula for rP/PH

Solution	Extra IP FIB	Exra LFIB	Extra BGP mappings
Solution 1: draft-bashandy-bgp-edge-node-frr-03	N(pPE) × N(rPE/pPE)	N(pPE) × N(rPE/pPE)	zero
Solution 2: draft-bashandy-bgp-frr-vector-label-00	N(pPE)	N(pPE) × N(rPE/pPE) **Or** N(rPE)	Zero
Solution 3: draft-bashandy-bgp-frr-mirror-table-00	N(pPE) × N(rPE/pPE)	N(pPE) × N(rPE/pPE)	Zero

Quantitative Comparison: Formula for rPE/Protector

Solution	Extra IP FIB	Exra LFIB	Extra BGP mappings
Solution 1: draft-bashandy-bgp-edge-node-frr-03	N(pPE) × N(rPE/pPE)	N(pPE) × N(rPE/pPE) + N(VPN/rPE)	N(VPN/rPE)
Solution 2: draft-bashandy-bgp-frr-vector-label-00	N(pPE)	N(pPE) + N(VPN/rPE)	N(VPN/rPE)
Solution 3: draft-bashandy-bgp-frr-mirror-table-00	N(pPE) × N(rPE/pPE)	N(pPE/rPE) × N(VPNs/pPE) + N(pPE) × N(rPE/pPE)*	Zero

Quantitative Comparison: Formula for Solution Agnostic Nodes

Solution	Extra IP FIB	Exra LFIB	Extra BGP mappings
Solution 1: draft-bashandy-bgp-edge-node-frr-03	N(pPE) × N(rPE/pPE)	N(pPE) × N(rPE/pPE)	Zero
Solution 2: draft-bashandy-bgp-frr-vector-label-00	N(pPE)	N(pPE)	Zero
Solution 3: draft-bashandy-bgp-frr-mirror-table-00	N(pPE) × N(rPE/pPE)	N(pPE) × N(rPE/pPE)	Zero

Main Two Advantages of each Solution

- Solution 1: draft-bashandy-bgp-edge-node-frr-03
  - Simple forwarding plane
  - No switching performance drop even during failure
- Solution 2: draft-bashandy-bgp-frr-vector-label-00
  - Simplest Provisioning
  - Maximum scalability:
    - No single node need to maintain all or most of state
    - Minimum churn in the network*

Main Two Advantages of each Solution

- Solution 3: `draft-bashandy-bgp-frr-mirror-table-00`
  - No need to upgrade Ingress PE
  - Easy to use a centralized router*

Main Two Disadvantages of each Solution

- Solution 1 : `draft-bashandy-bgp-edge-node-frr-03`
  - The additional state injected in the network
  - Some Configuration complexity: Need to configure non-overlapping address ranges on different pPEs
- Solution 2: `draft-bashandy-bgp-frr-vector-label-00`
  - Non-trivial forwarding plane: Need to pop 3 labels at steady state
  - Need to upgrade iPE, pPE, and PHP

Main Two Disdvantages of each Solution

- Solution 3: `draft-bashandy-bgp-frr-mirror-table-00`
  - The additional state injected in the network
  - Configuration complexity: Need to configure correct pNH on rPE and pPE
    - Or else need to re-advertise some or all of the protected prefixes