Segment Routing with MPLS Data Plane

SR-MPLS

Version 0.4

M Abdullah Al Naser
Datacom Professional
mnaser.net@gmail.com

The objective of this presentation is to discuss the need and benefits of ***Segment Routing with MPLS data plane (SR-MPLS)*** along with some shortcomings of ***LDP/RSVP-TE based MPLS*** in production network.

This presentation maybe useful for a network engineer running typical MPLS network for years to learn “Segment Routing” from very basic and also for a CxO to think about to move towards SDN era for the sake of service quality.

SR-MPLS 101

MPLS Intro

IP Based Forwarding

- A method to forward packets
  - Based on destination IP address
  - Same lookup process at each hop takes time
- Replaced Frame Relay, ATM in telecommunication network
  - IP came as a best effort forwarding capability
  - Variable length packet header
  - Connectionless; reliability ensured by transport layer
  - Flexible but has less QoS capability
- TE not possible

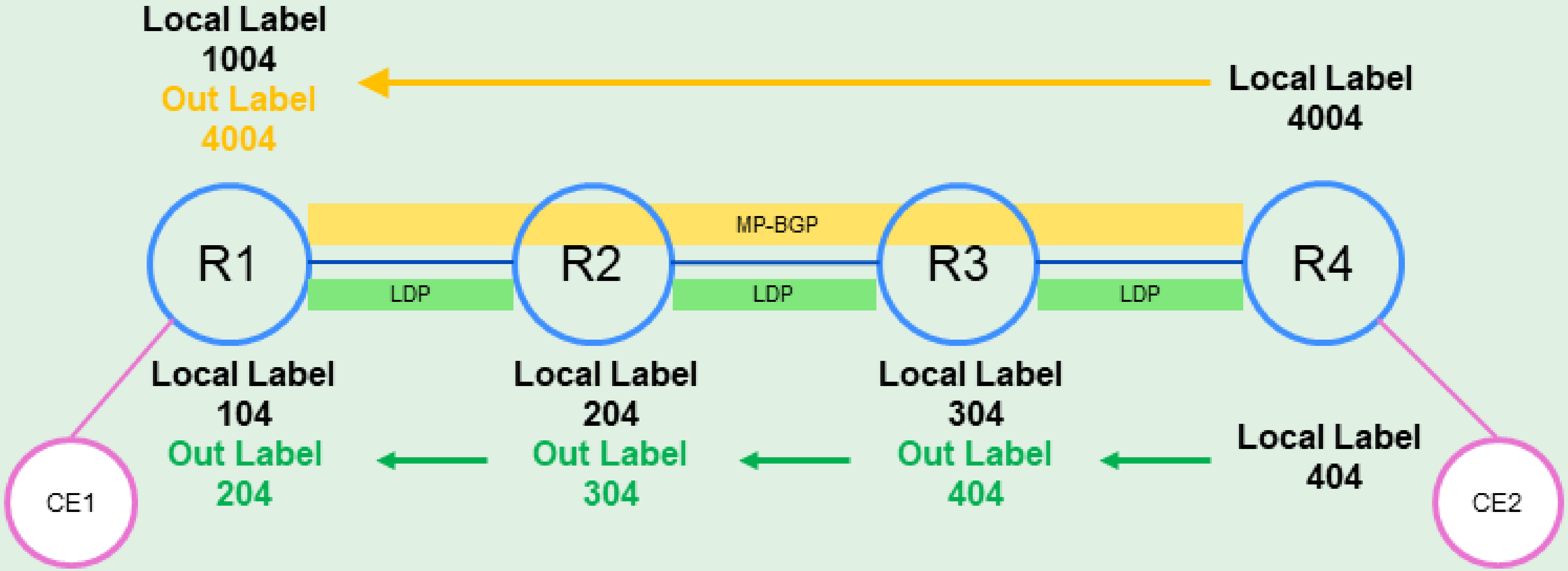
MPLS Intro

- A label-based approach to forward traffic
  - Labels are in a fixed 32-bit header
  - Label Distribution Protocol (LDP)
  - Label Operation: PUSH, SWAP, POP
- Overlay tunnelling for customer services
  - L3VPN, L2VPN
  - BGP-free core; RR requires for large network
- TE possible: RSVP-TE

MPLS LDP

Control Plane

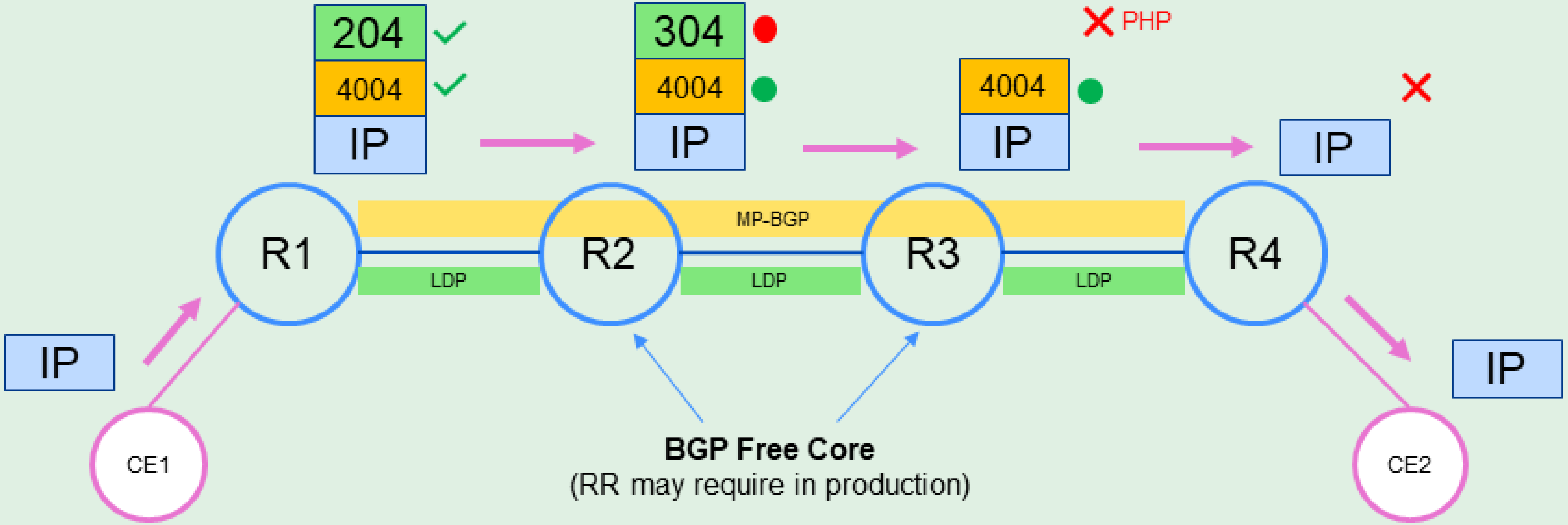
Label distribution from R4 to R1



MPLS LDP

Forwarding Plane
Traffic forwarding from R1 to R4

- ✓ Label pushed
- ● Label unchanged
- ● Label swapped
- ✕ Label popped



BGP Free Core
(RR may require in production)

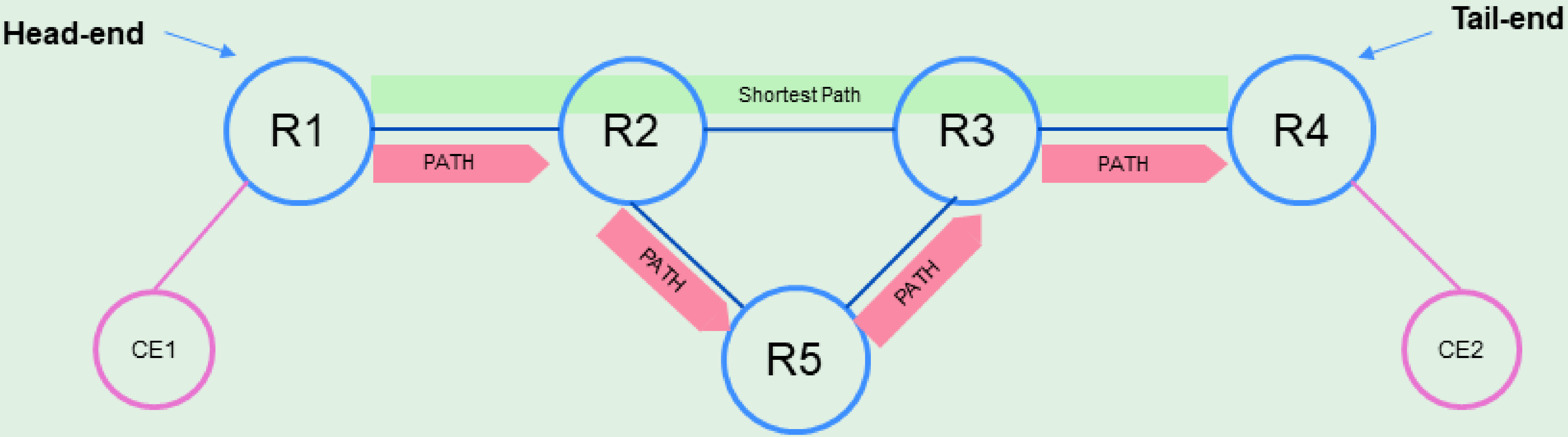
MPLS TE

- To steer traffic through longer path
  - To utilize unused links
  - To avoid congestion on busy links
- RSVP-TE
  - Creates circuits; state is signalled hop-by-hop
  - Need to configure link constraints (BW and other attributes)
  - Distribute TE info using IGP extension
  - PCALC (CSPF) to calculate best path
  - Carries TE labels to signal tunnel from head-end to tail-end PE
- Push traffic into the tunnel
  - Manual, Auto-route
  - Tunnel fallback provides less traffic disruption

MPLS TE

Control Plane

RSVP PATH messages reserve BW and request for MPLS label

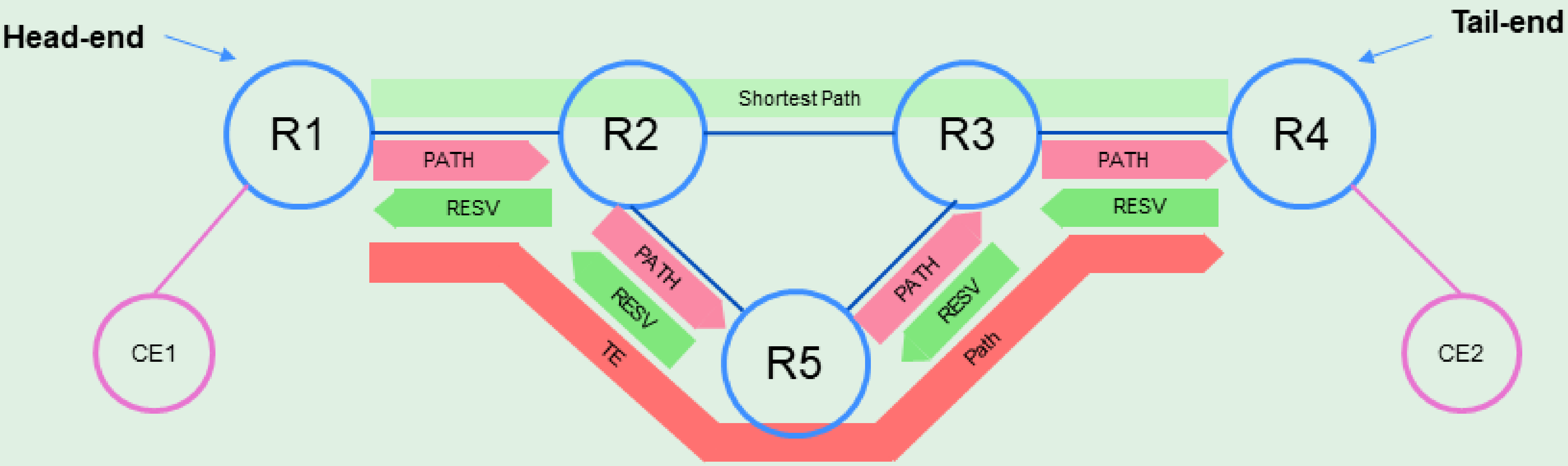


MPLS TE

Control Plane

RSVP RESV messages carry MPLS label

TE PATH is signaled "hop-by-hop" based on ERO !!!



MPLS TE

Shortcomings:

- Tunnel is not bidirectional
- IGP tracks link constraints and floods periodically
  - Maintains state
  - TE tunnel need to be re-optimized frequently
- RSVP-TE isn’t ECMP friendly (N^2*K tunnel required)
  - Doesn’t scale well with inter-domain (IGP) network

Additional overhead for both router and it’s administrator

Why is RSVP-TE still useful?

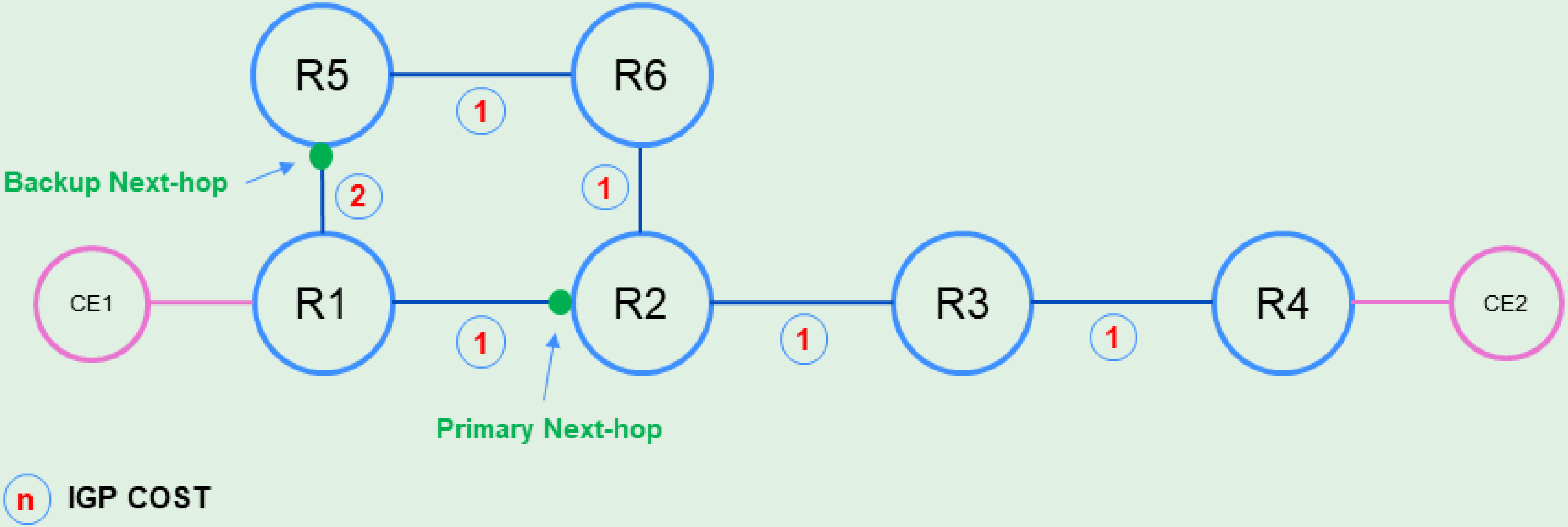
Fast Re-route (FRR)

- Pre-computed backup path
  - Backup next-hop installed in forwarding table
  - Minimize reconvergence time
  - Can be used to protect tunnels (links/nodes)
  - Widely used in carrier backbone
- Must have LFA – Loop Free Alternative
- Doesn’t provide optimum path all the time
- Very difficult to configure and manage

Fast Re-route (FRR)

FRR with Loop Free Alternative (LFA)

Case-1

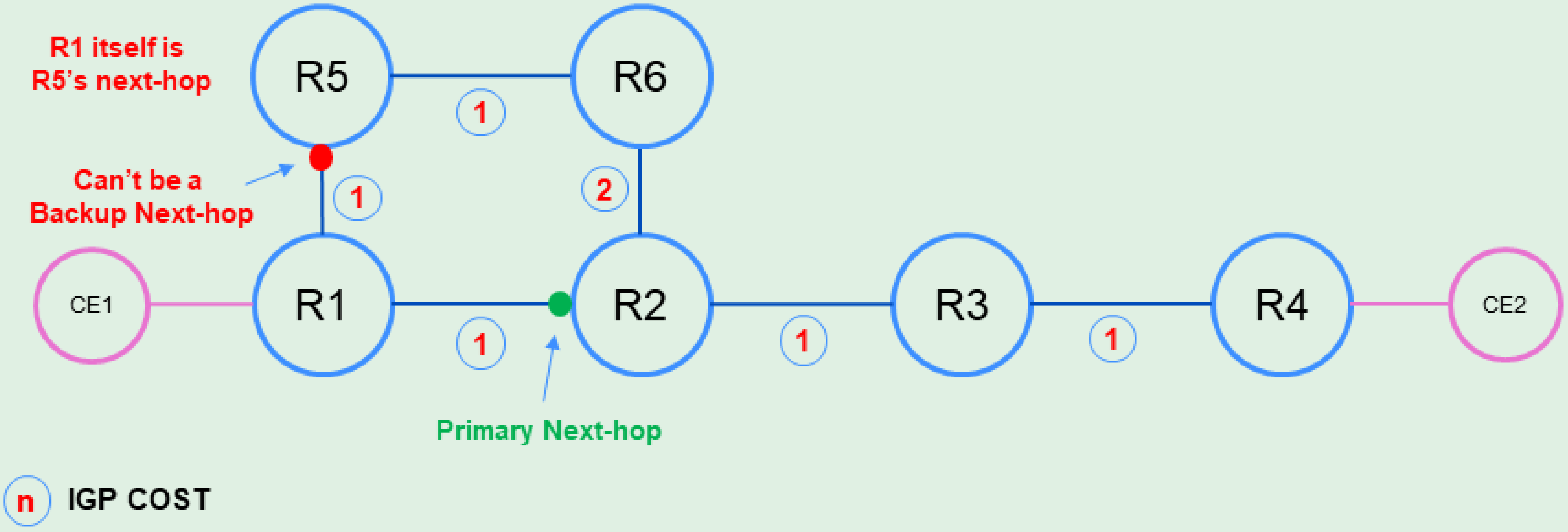


Fast Re-route (FRR)

FRR with Loop Free Alternative (LFA)

Case-2

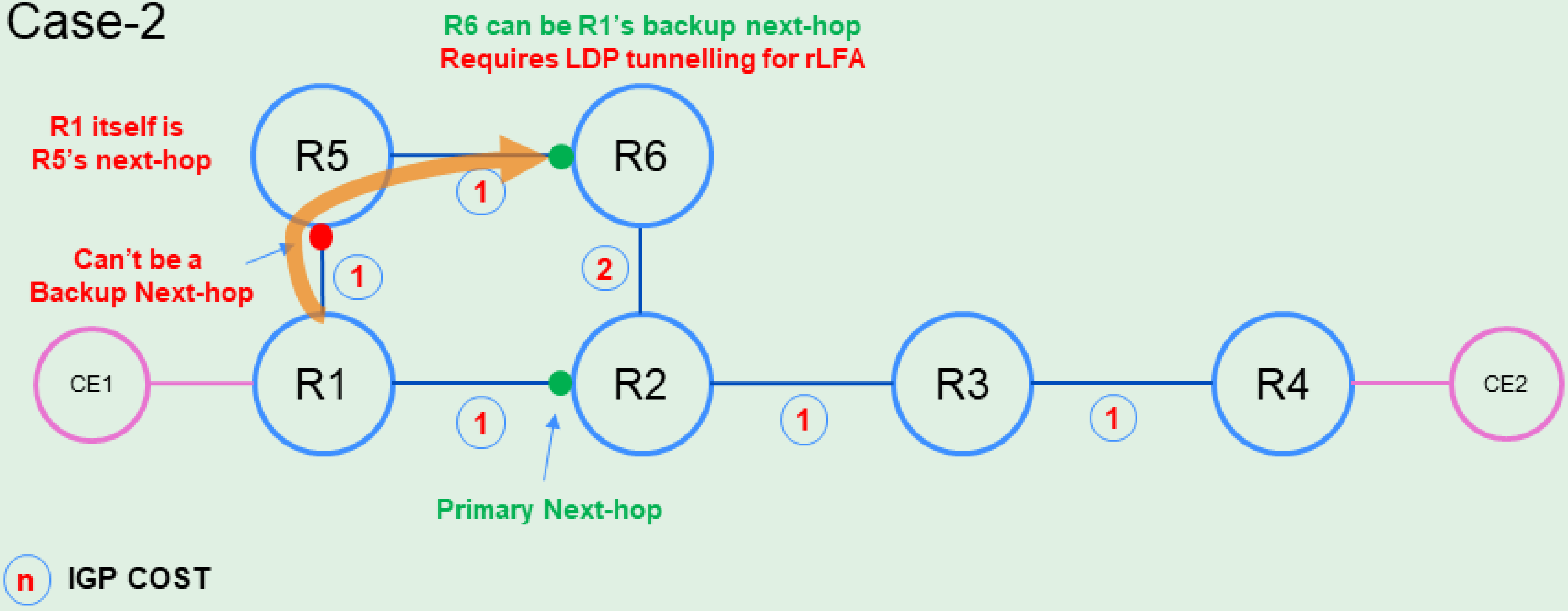
Micro-loop happens !!!



Fast Re-route (FRR)

FRR with Remote Loop Free Alternative (RLFA)

Case-2

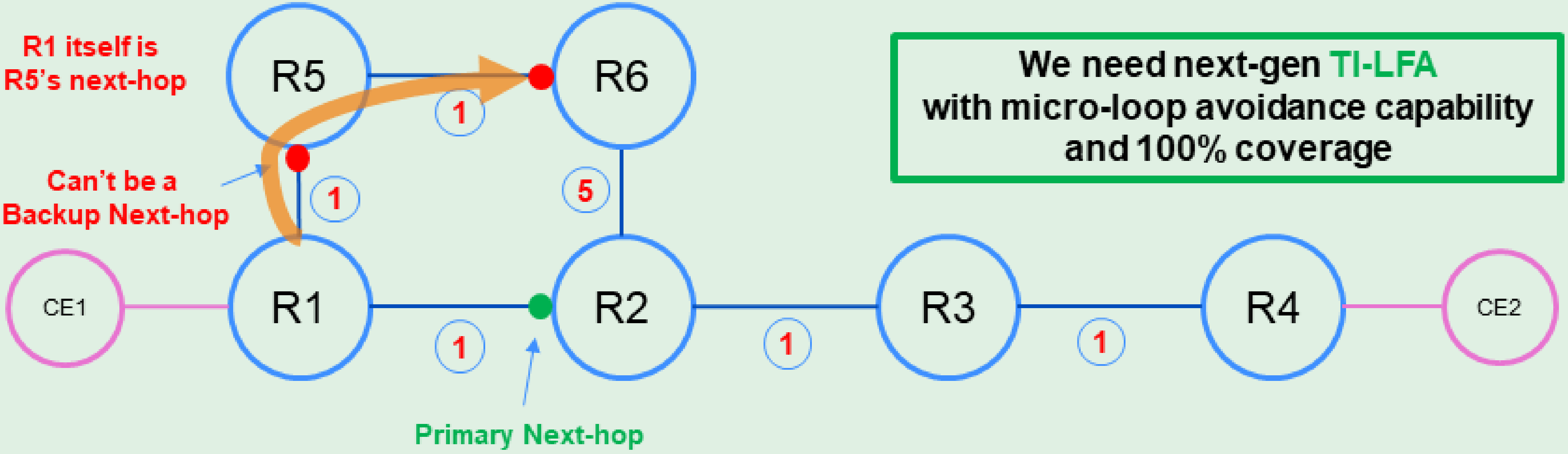


Fast Re-route (FRR)

FRR with RLFA is also not always Loop Free

Case-3

R6 can't be R1's backup next-hop
Because R1 itself is in R5's transit path



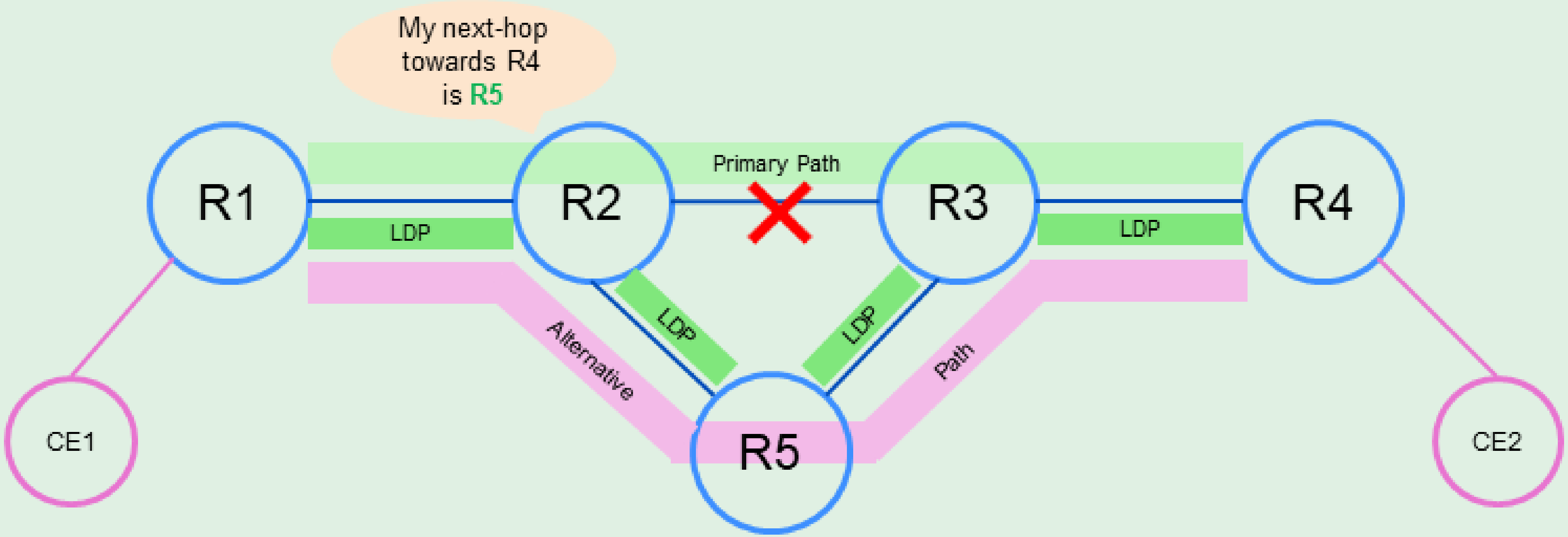
We need next-gen TI-LFA
with micro-loop avoidance capability
and 100% coverage

n IGP COST

IGP-LDP Sync

Primary Path Goes Down (!)

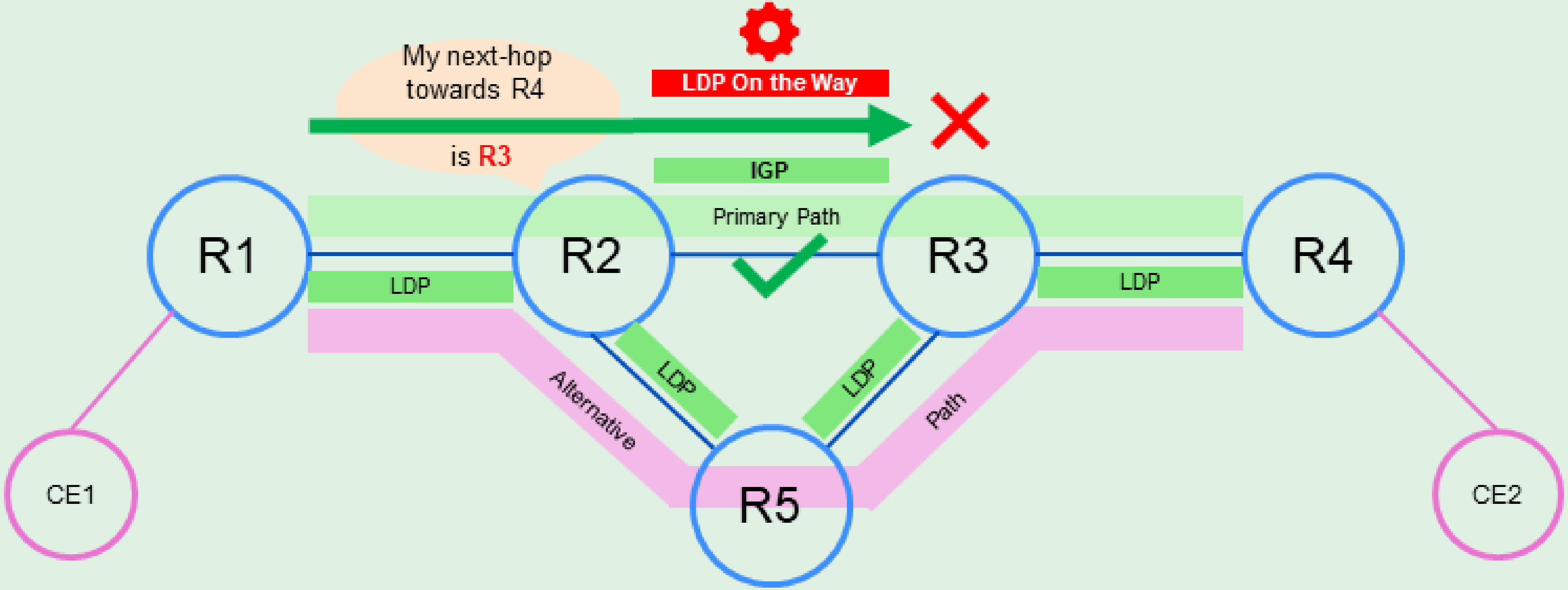
Customer traffic re-routed via alternative path



IGP-LDP Sync

Primary Path Goes Up (!!!)

Customer traffic dropped when re-routed via primary path – BROKEN LSP



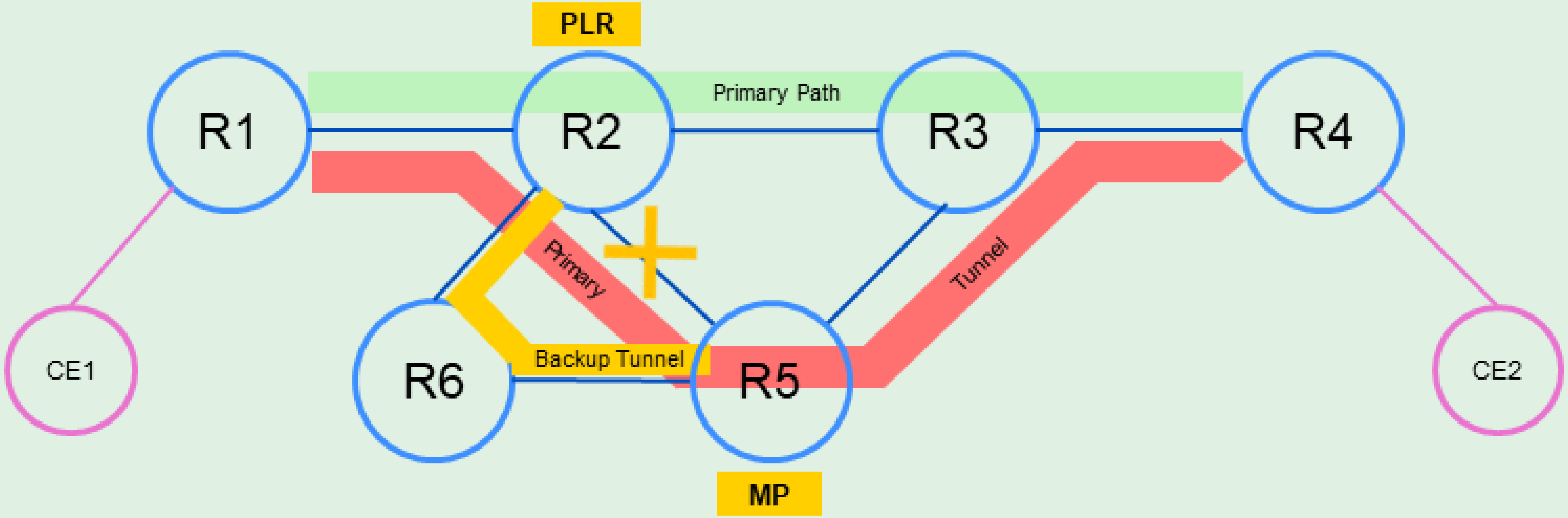
IGP-LDP Sync requires to be configured on each LDP enabled link

SR-MPLS 101

FRR with RSVP-TE

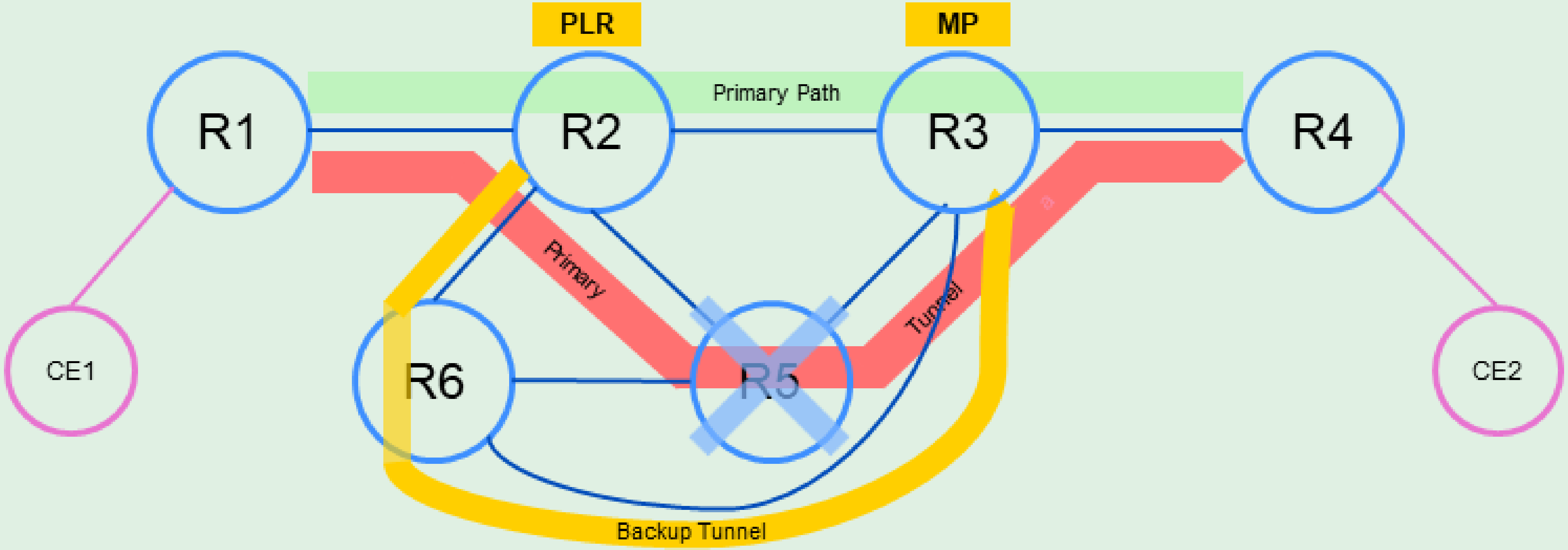
FRR with RSVP-TE

- **Link Protection**



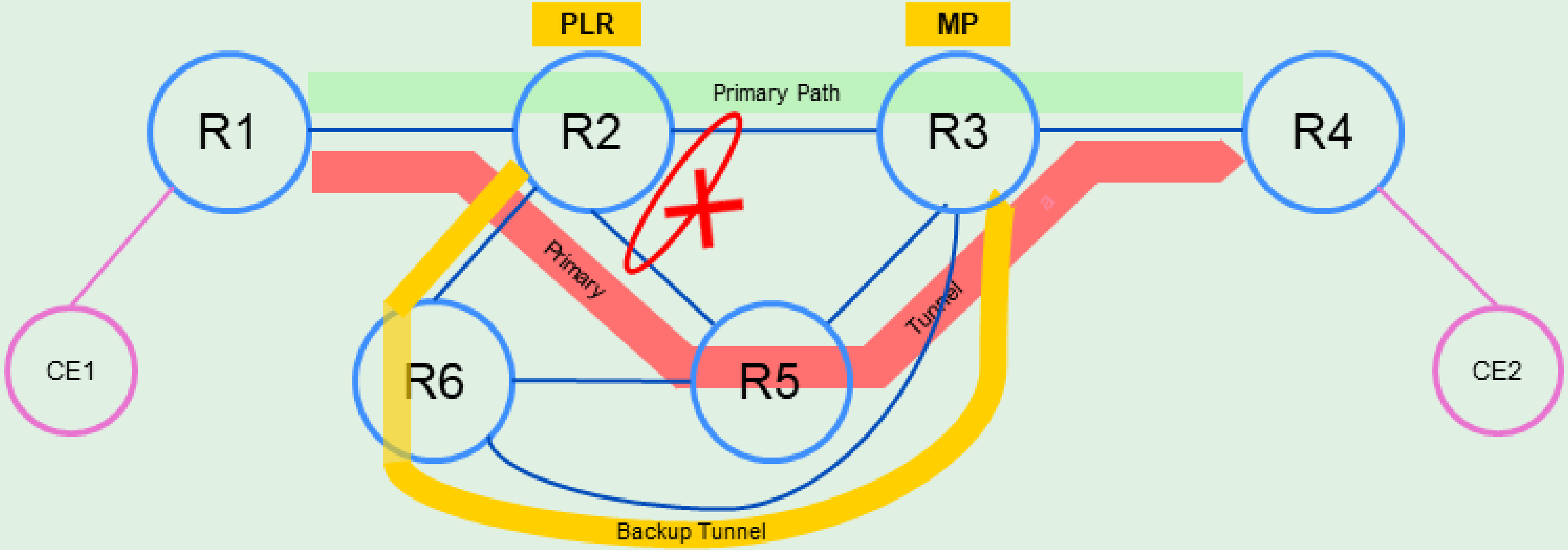
FRR with RSVP-TE

- **Node Protection**



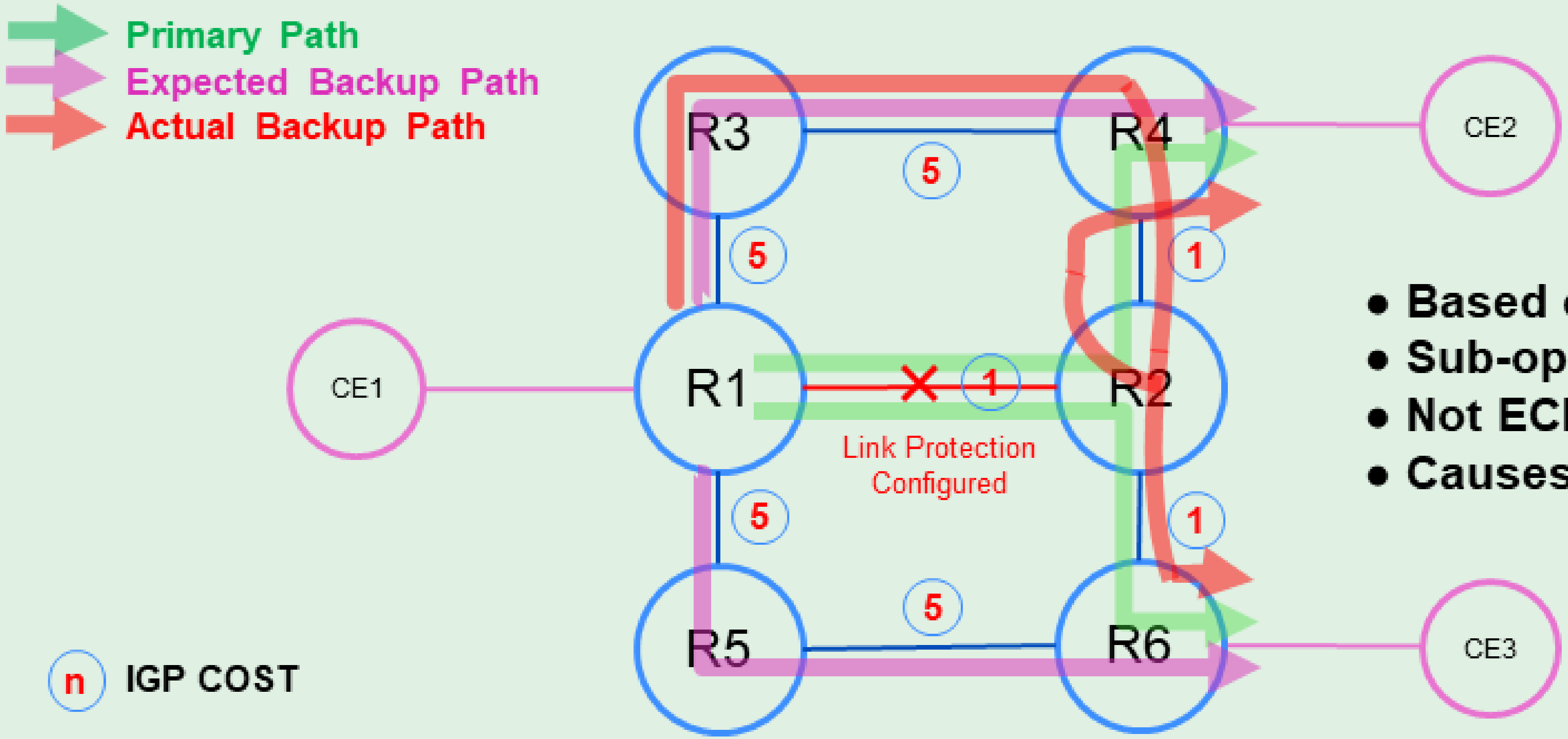
FRR with RSVP-TE

- **SRLG Protection**



FRR with RSVP-TE

Typical FRR may provide sub-optimal path



- Based on circuit concepts
- Sub-optimal Path
- Not ECMP friendly
- Causes link congestion

MPLS (Cont.)

Shortcomings:

- Doesn’t deal with performance anymore
  - CEF, ASIC etc
- Requires IGP-LDP synchronization
- TE is not scalable
- FRR is supported but very difficult to configure and manage
- Not suitable for very large carrier network

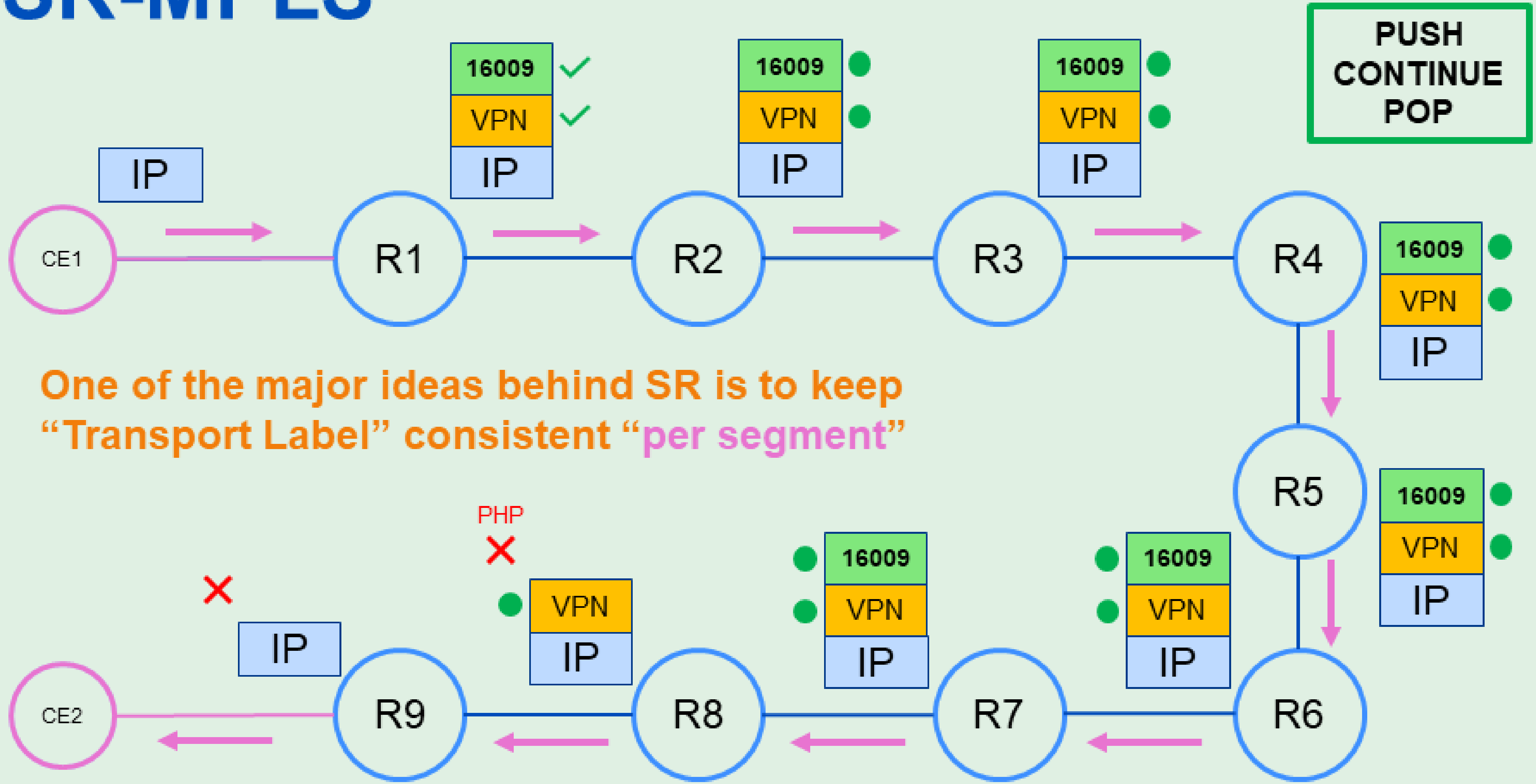
SR-MPLS 101

SR-MPLS Intro

SR-MPLS

- LDP+RSVP control plane is complex
- SR control plane with IGP
  - SR is an extension of IGP
  - MPLS is still as the forwarding plane
- No LDP, No RSVP-TE are required

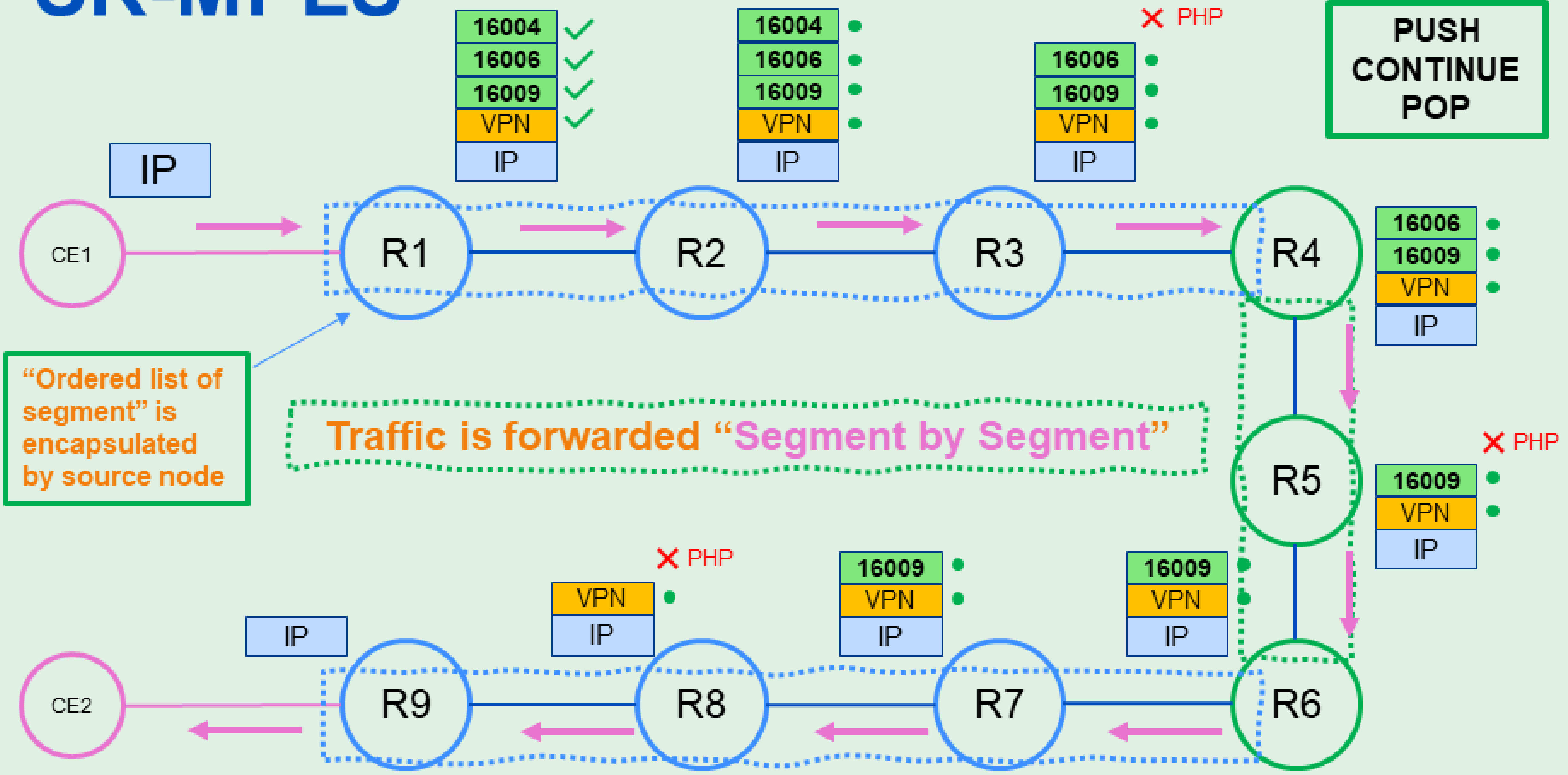
SR-MPLS



One of the major ideas behind SR is to keep "Transport Label" consistent "per segment"

Prepared by - M Abdullah Al Naser

SR-MPLS

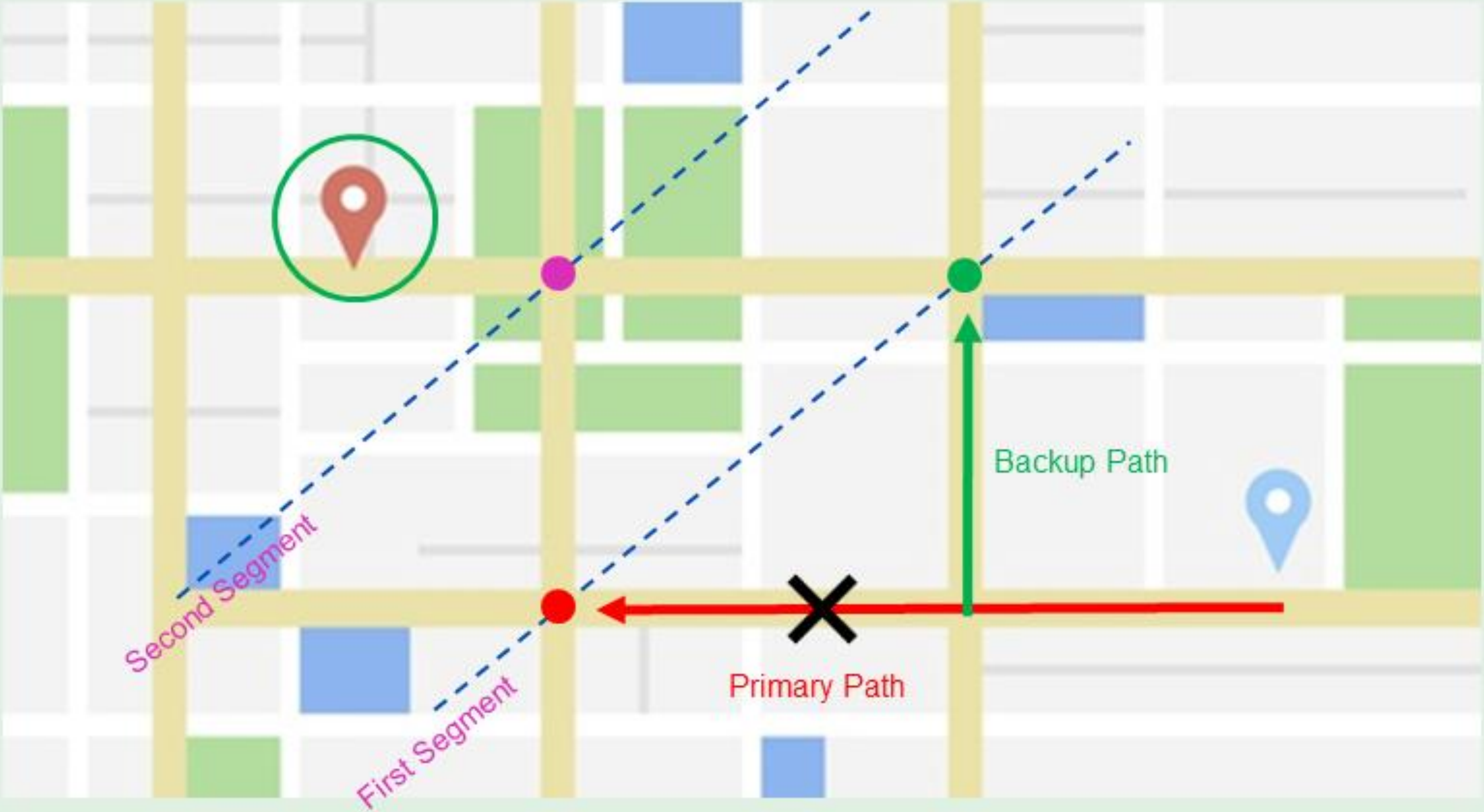


PUSH
CONTINUE
POP

"Ordered list of segment" is encapsulated by source node

Traffic is forwarded "Segment by Segment"

SR-MPLS



SR-MPLS

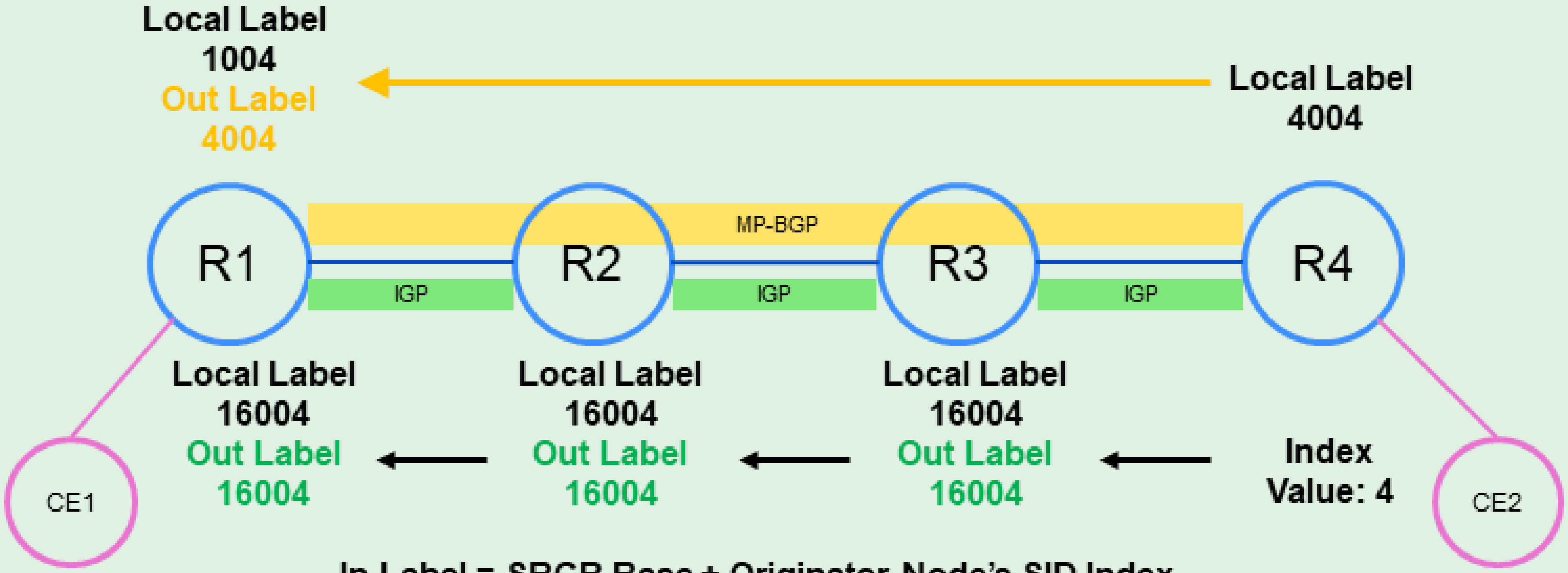
- Source routing paradigm – SPRING
  - SPRING – Source Packet Routing in Networking
  - Source node can encapsulate path information using label stack
  - Intermediate nodes don’t need to maintain any state
- Inserts an “ordered list of segment” with the packet
  - A segment is an instruction – topological, service chaining etc
  - Segment ID (SID) is equivalent to the LDP Label (in case of SR-MPLS)
  - Each SID indicates a segment; the top SID indicates active segment
- IGP carries SID information
  - No LDP, No RSVP

SR-MPLS

Control Plane

Label generation on each router to reach R4

SRGB (all router)
16000 - 23999



In Label = SRGB Base + Originator Node's SID Index

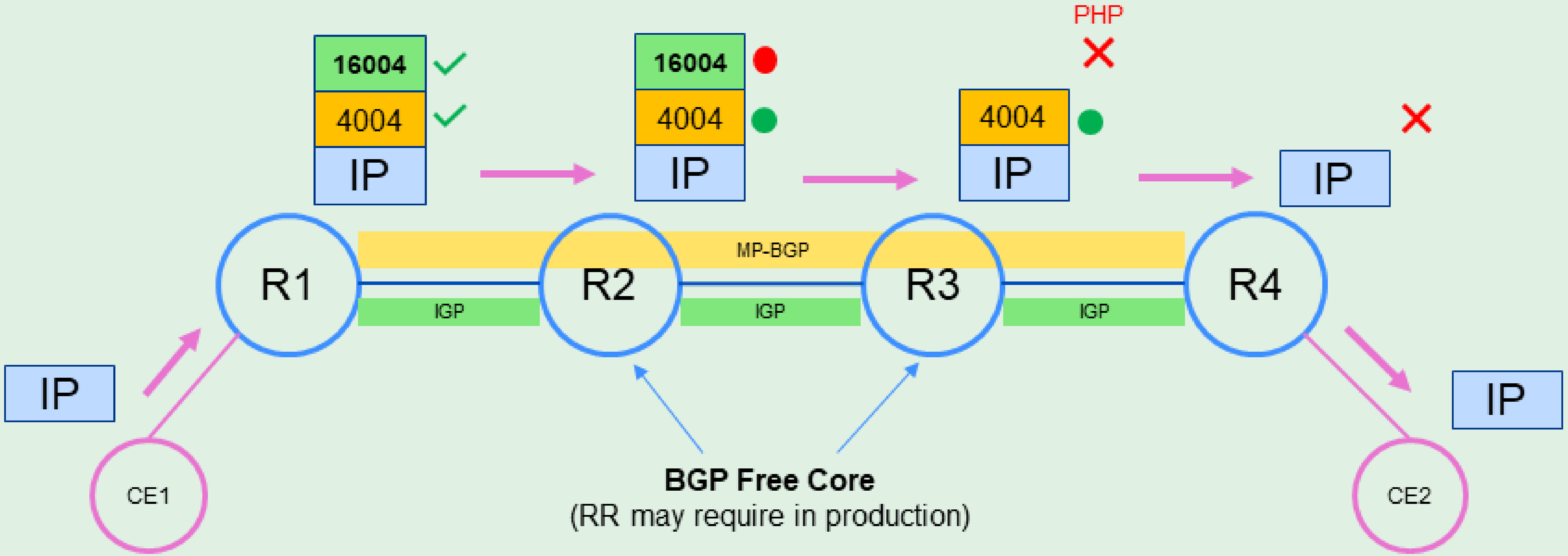
Out Label = Next-hop's SRGB Base + Originator Node's SID Index

Transport (Out) labels are consistent

SR-MPLS

Forwarding Plane

Traffic forwarding from R1 to R4



BGP Free Core

(RR may require in production)

Transport (Out) labels are consistent

TE and FRR would be easier

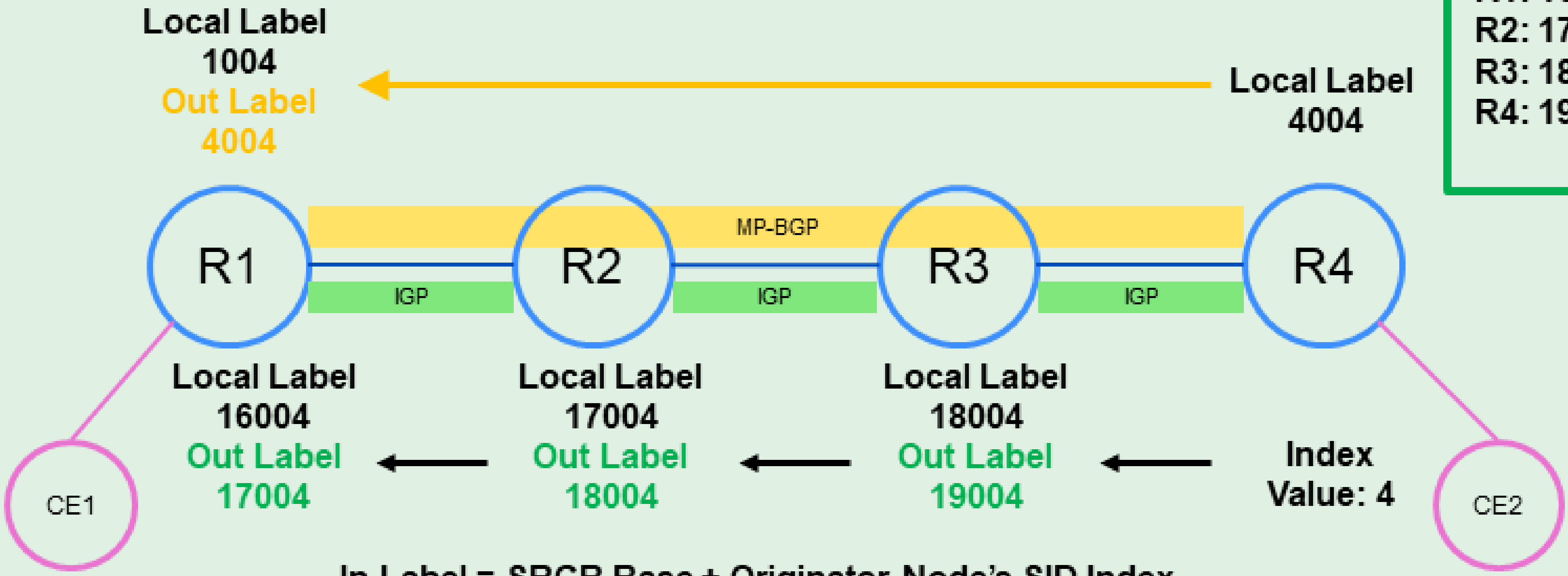
SR-MPLS

Control Plane

Label generation on each router to reach R4

What if the SRGB is not same on routers

R1: 16000 - 16999
R2: 17000 - 17999
R3: 18000 - 18999
R4: 19000 - 19999



In Label = SRGB Base + Originator Node's SID Index

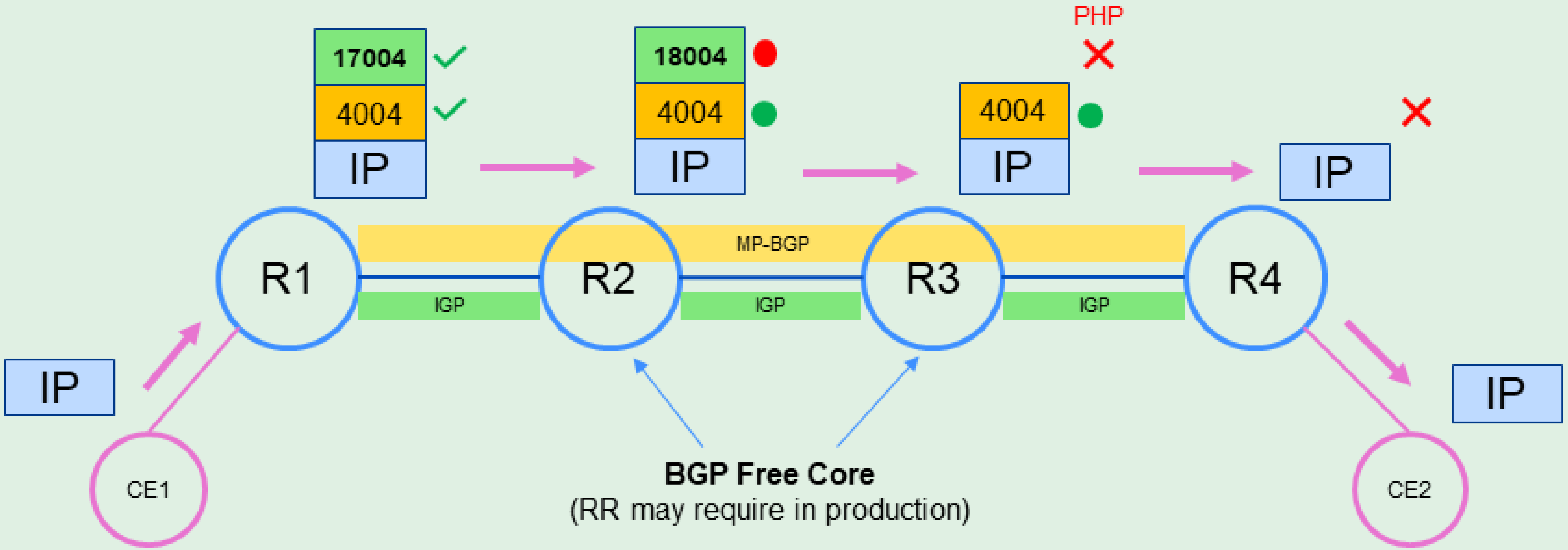
Out Label = Next-hop's SRGB Base + Originator Node's SID Index

Transport (Out) labels are inconsistent

SR-MPLS

Forwarding Plane

Traffic forwarding from R1 to R4



BGP Free Core
(RR may require in production)

Transport (Out) labels are inconsistent
TE and FRR would be difficult

SR-MPLS

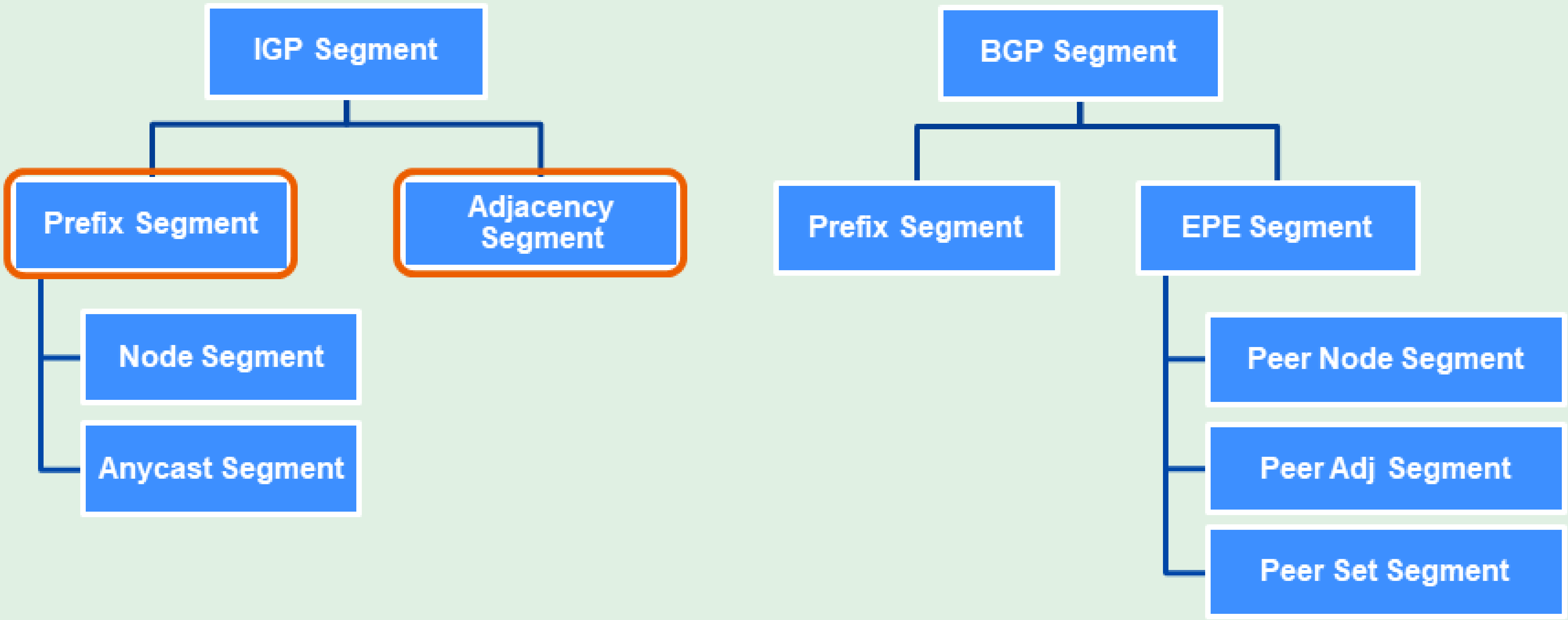
Global Segment

- Global segments are for routers' loopback
- Assigned from Segment Routing Global Block (SRGB)
  - Ranges from 16000 to 23999 for interoperability
  - Intended design; highly recommended to keep it same on all router
- Keeps transport label consistent per segment (i.e. within IGP domain)

Local Segment

- Local segments are for routers' P2P links
  - Assigned from dynamic label range
- Supports advanced traffic engineering and FRR
  - TE path doesn't need to be signalled hop-by-hop

Segment ID



IGP Control Plane

IGP Segment

Prefix Segment

Node Segment

Anycast Segment

Adjacency Segment

BGP Control Plane

BGP Segment

Prefix Segment

EPE Segment

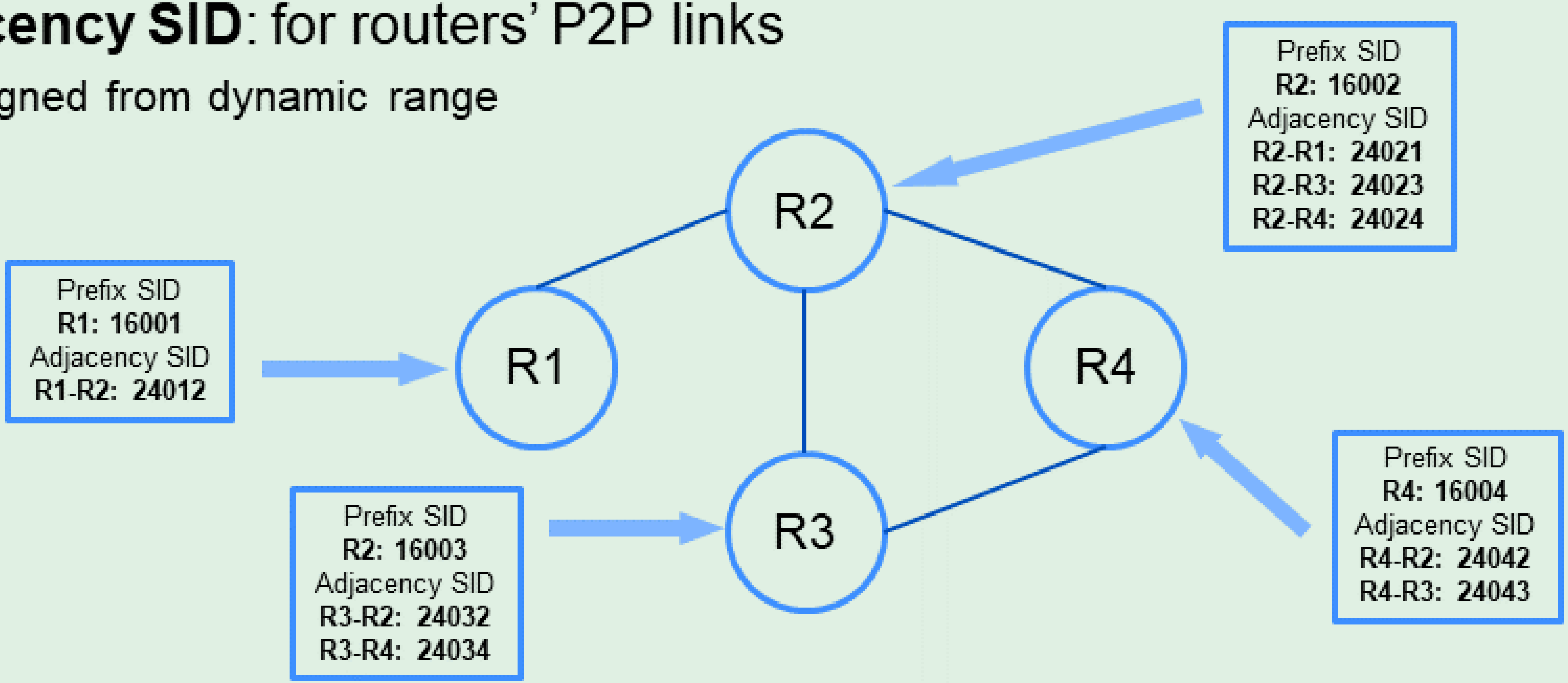
Peer Node Segment

Peer Adj Segment

Peer Set Segment

Segment ID

- **Prefix SID**: for routers' loopbacks
  - Assigned from SRGB
- **Adjacency SID**: for routers' P2P links
  - Assigned from dynamic range



Prefix SID
R1: 16001
Adjacency SID
R1-R2: 24012

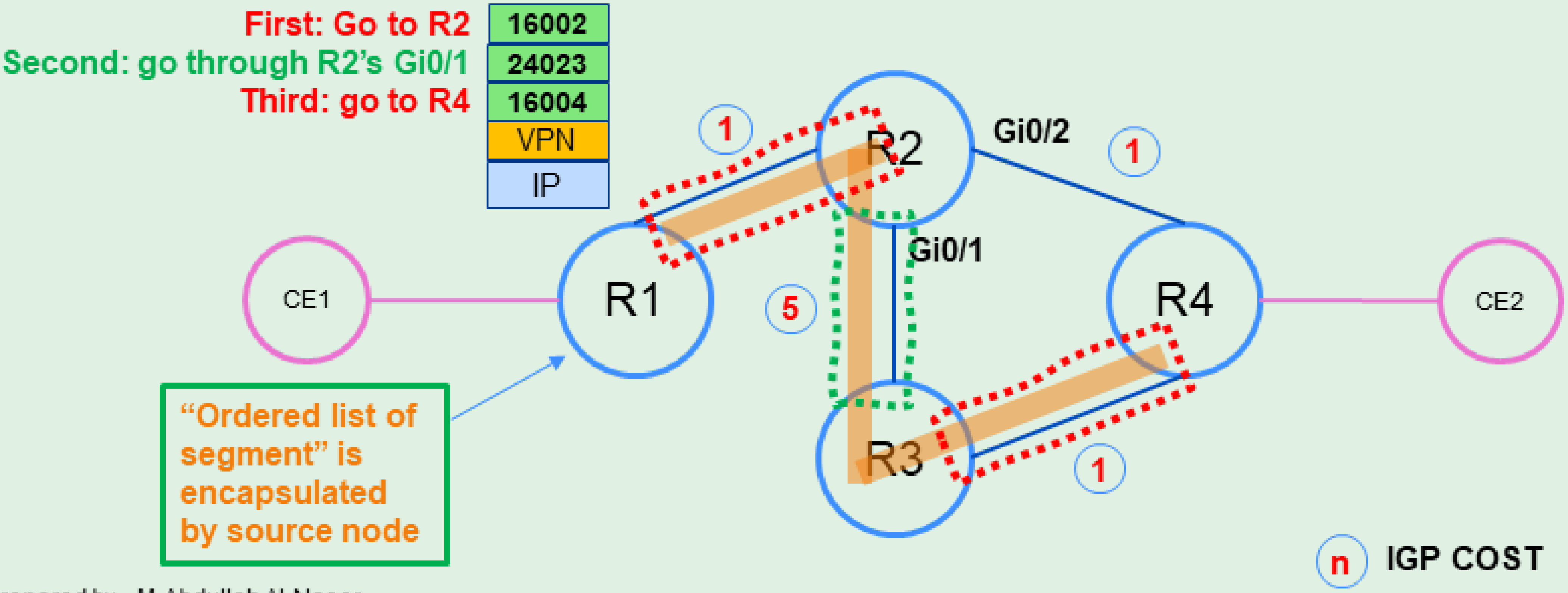
Prefix SID
R2: 16002
Adjacency SID
R2-R1: 24021
R2-R3: 24023
R2-R4: 24024

Prefix SID
R2: 16003
Adjacency SID
R3-R2: 24032
R3-R4: 24034

Prefix SID
R4: 16004
Adjacency SID
R4-R2: 24042
R4-R3: 24043

SID List

- Explicit path can be defined using local segments
- Useful for TE and FRR



SR-MPLS Features

- FRR with TI-LFA
  - Avoids micro-looping
- Scalable TE
  - SR Policy; and no more tunnel
- Can be a part of SDN (for inter-domain network)
  - PCEP and BGP-LS

SR-MPLS Fast Convergence

Segment Routing TI-LFA

Please Note:

- This slide deck is a continuation of previous one “SR-MPLS 101”
- Audience are expected to have prior basic knowledge on MPLS and Segment Routing

FRR with TI-LFA

Fast Reroute with TI-LFA

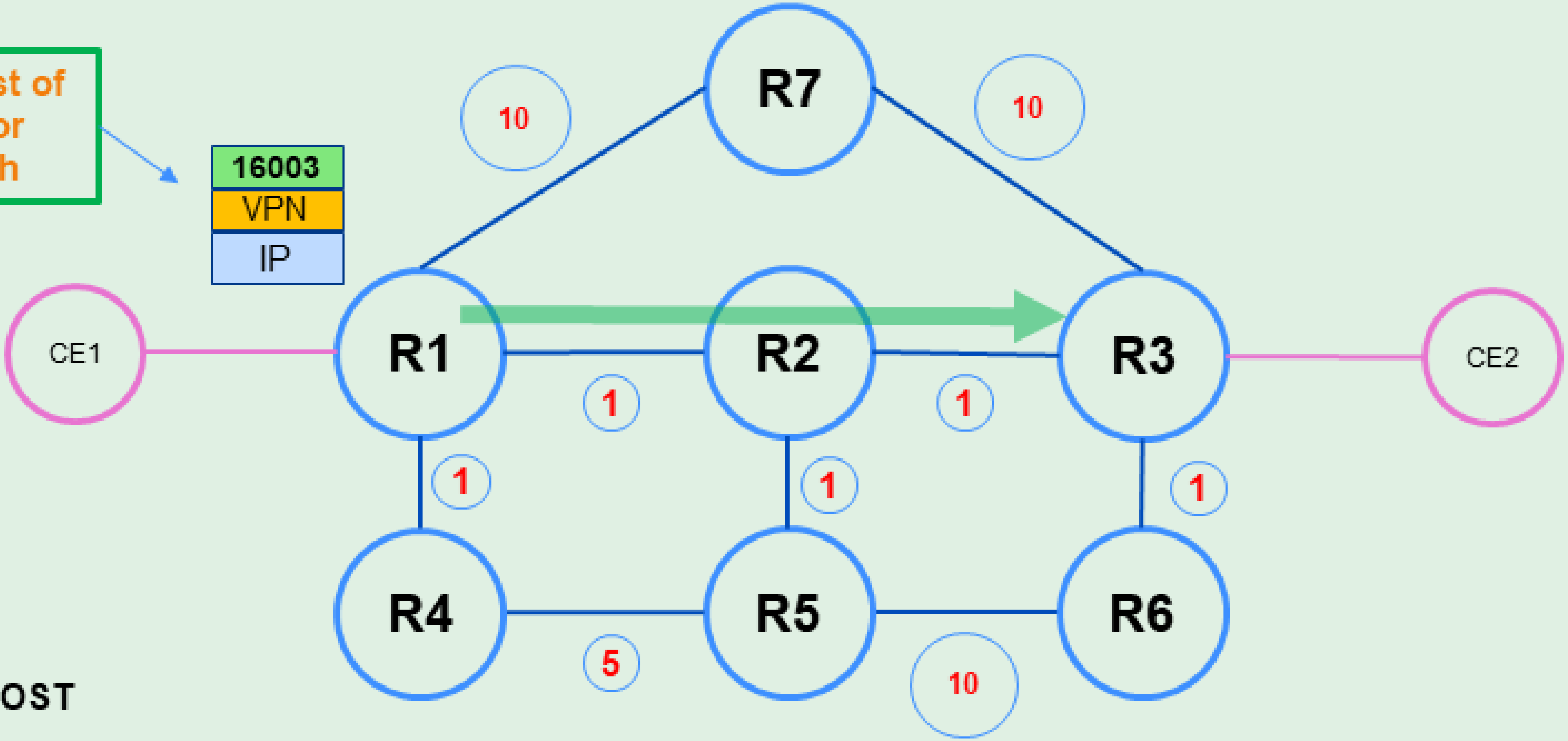
- Topology Independent LFA
  - Sub 50ms convergence time
  - Minimizes amount of dropped traffic during convergence
- Protects links, nodes and SRLGs
- Maintains better SLA

FRR with TI-LFA

TI-LFA Link Protection

Topology is fully functional
R1's Primary path is in forwarding table

"Ordered list of segment" for primary path

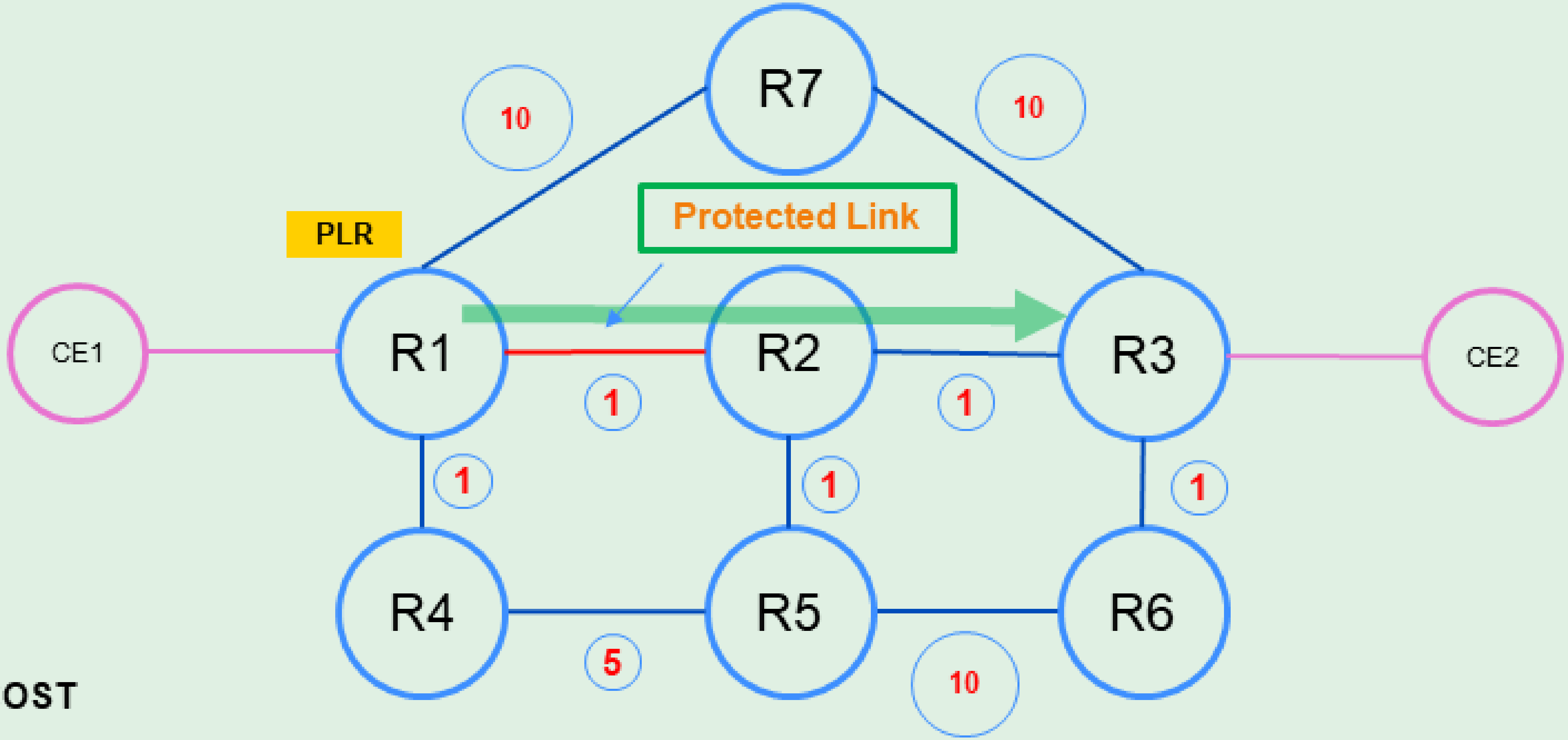


n IGP COST

FRR with TI-LFA

TI-LFA Link Protection

R1 is aware of the backup path in advance in case of link failure to R2



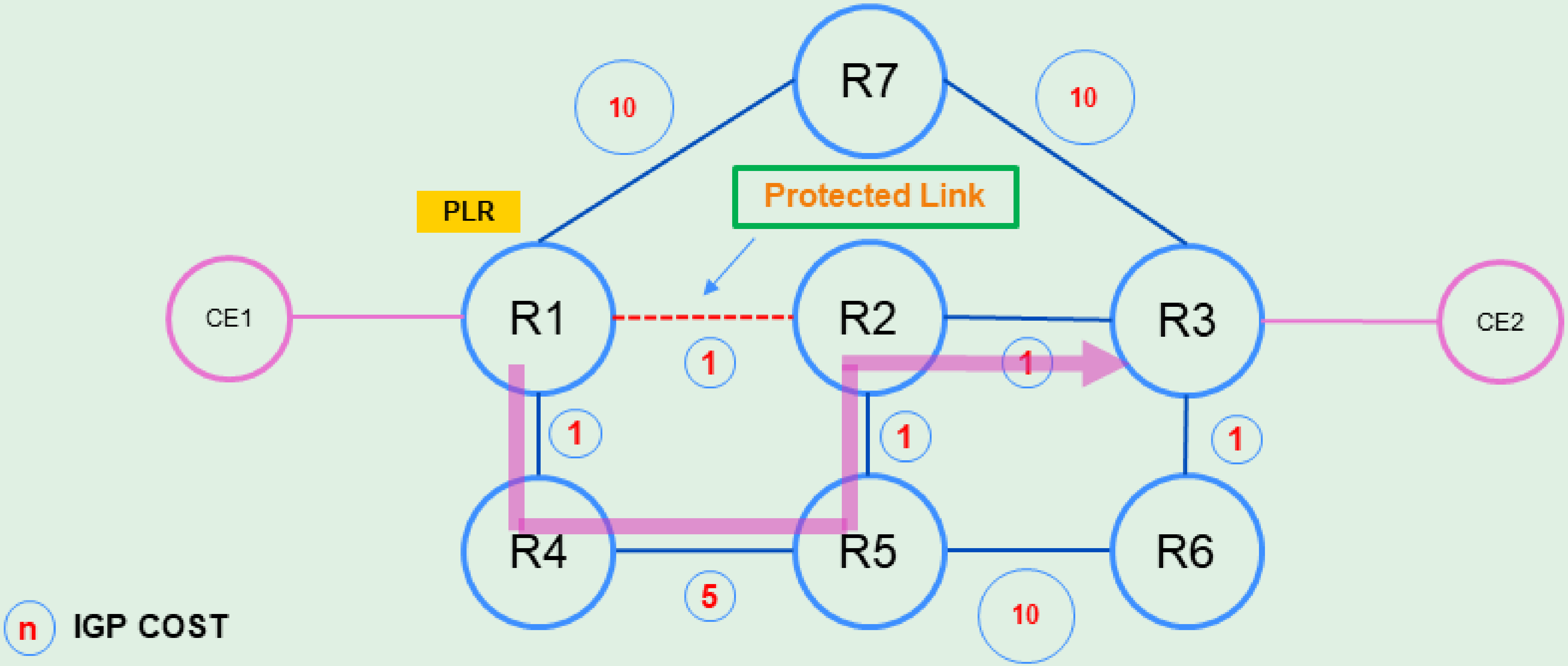
ⓝ IGP COST

Prepared by - M Abdullah Al Naser

FRR with TI-LFA

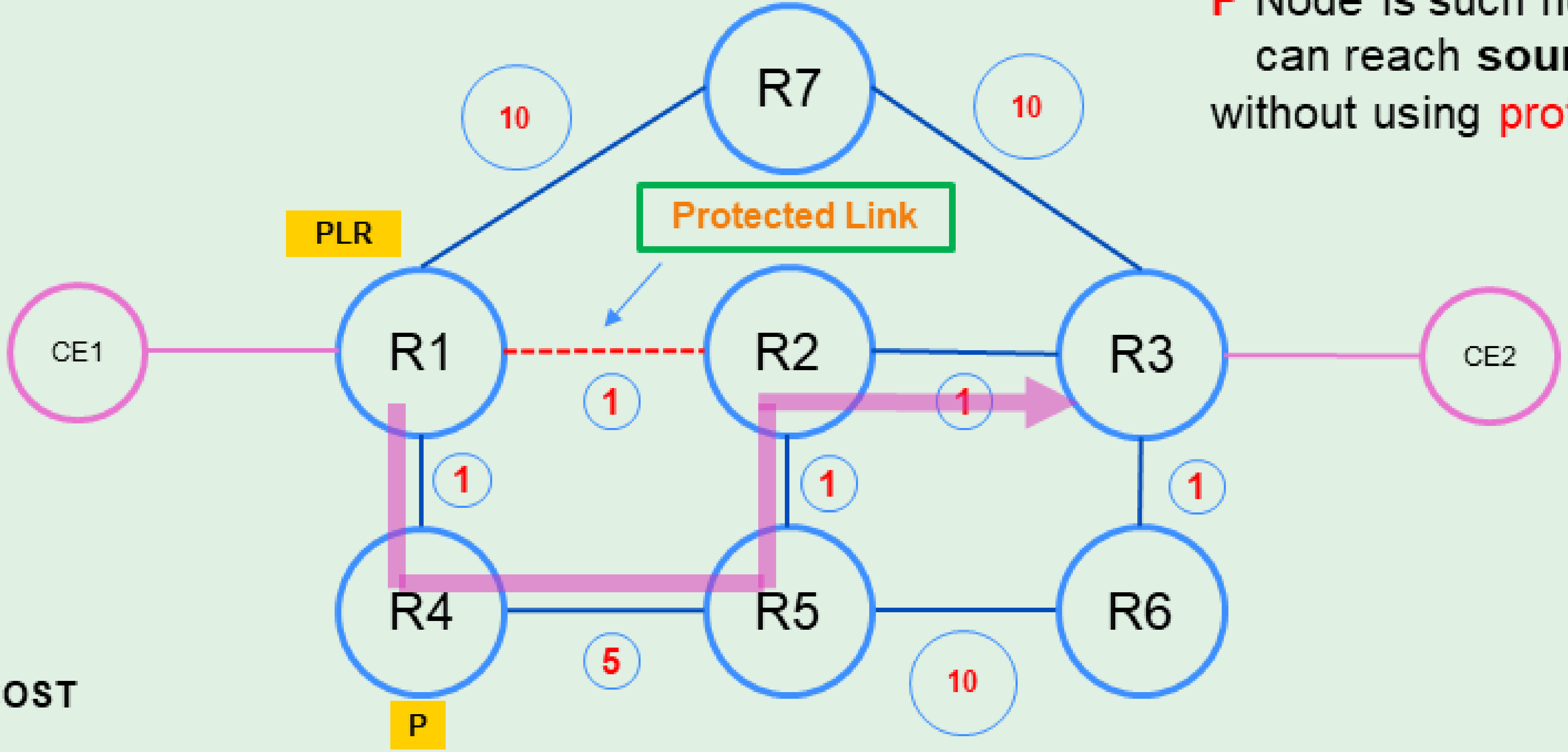
TI-LFA Link Protection

R1 prunes protected link before link failure occurs and find out post convergence path



FRR with TI-LFA

TI-LFA Link Protection



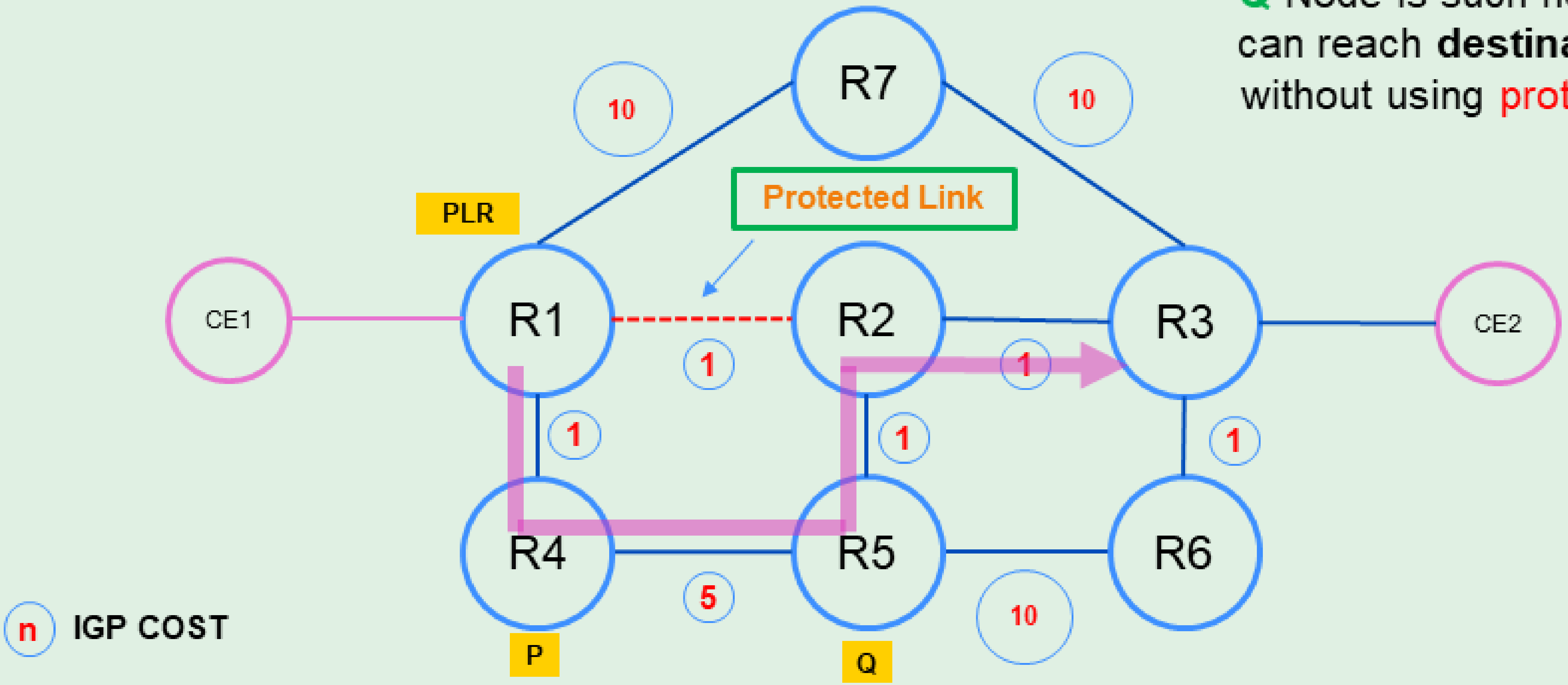
n IGP COST

R1 defines the P Node

P Node is such node which can reach **source (R1)** without using protected link

FRR with TI-LFA

TI-LFA Link Protection



R1 defines the Q Node

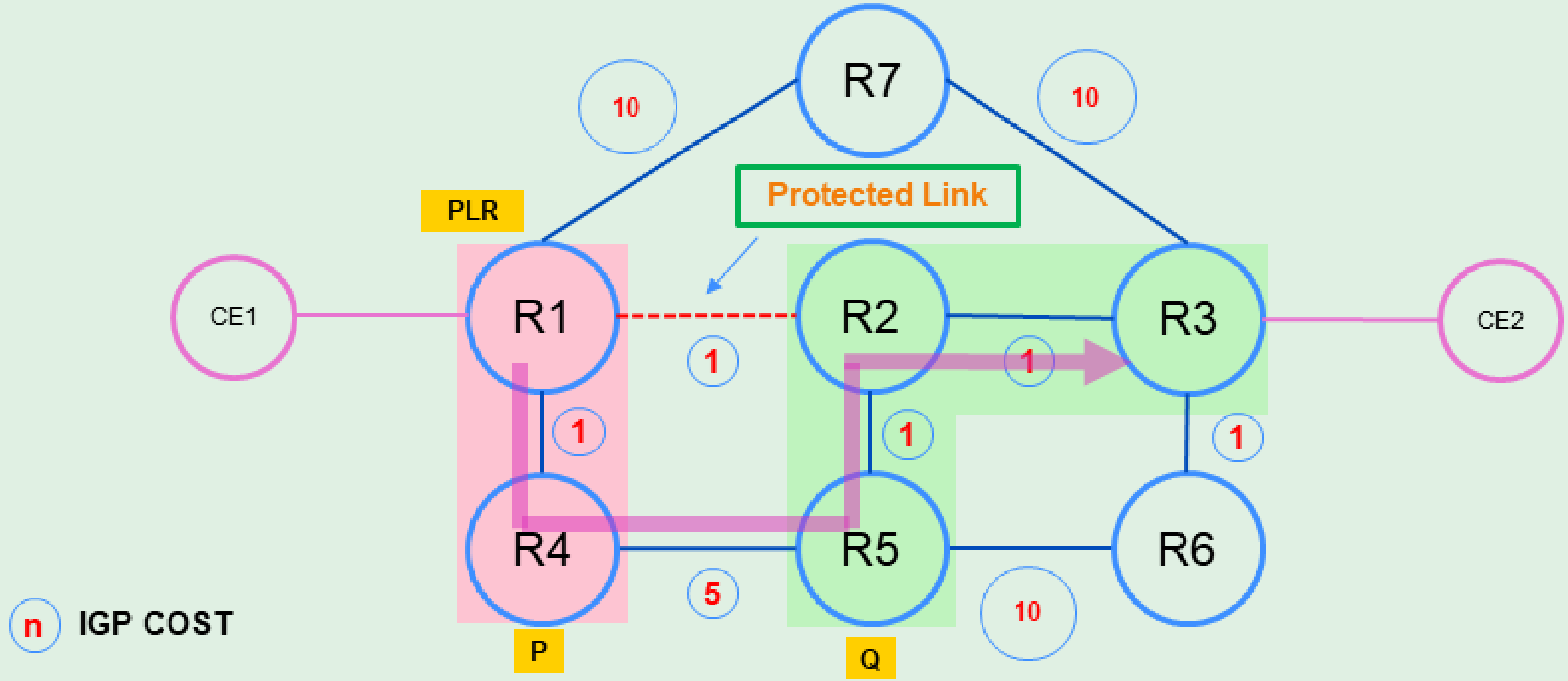
Q Node is such node which can reach **destination (R3)** without using protected link

Prepared by - M Abdullah Al Naser

FRR with TI-LFA

TI-LFA Link Protection

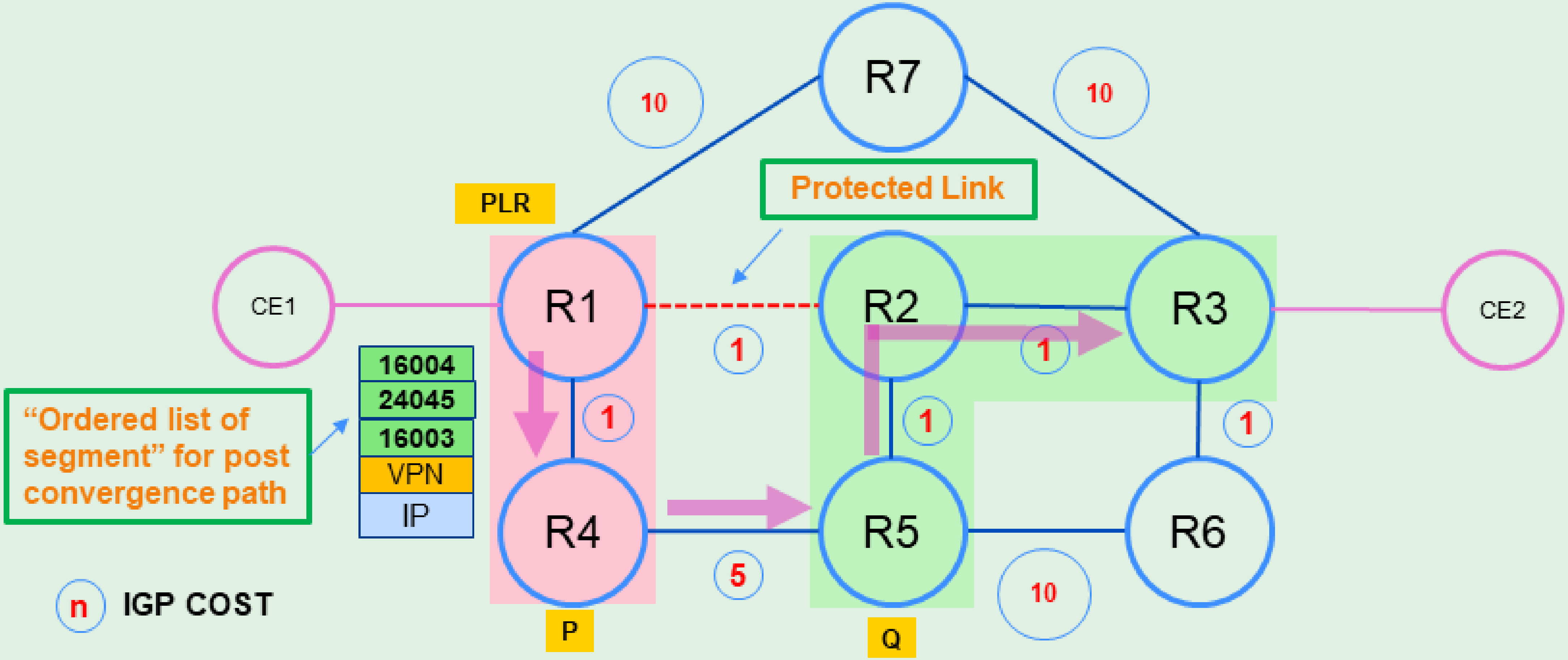
P Space and Q Space



FRR with TI-LFA

TI-LFA Link Protection

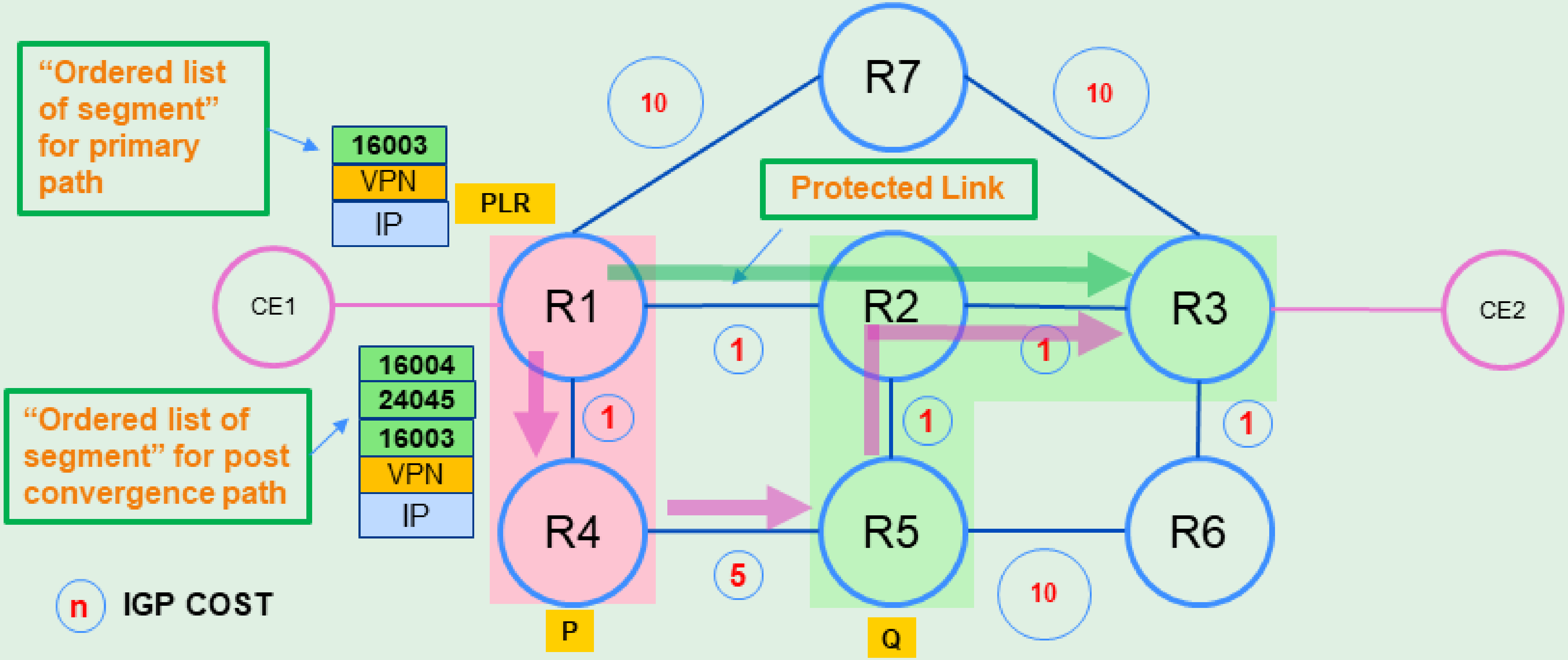
R1 builds SID-list based on each segment it found along the post convergence path



FRR with TI-LFA

TI-LFA Link Protection

R1 programs the precomputed SID-List it in forwarding table

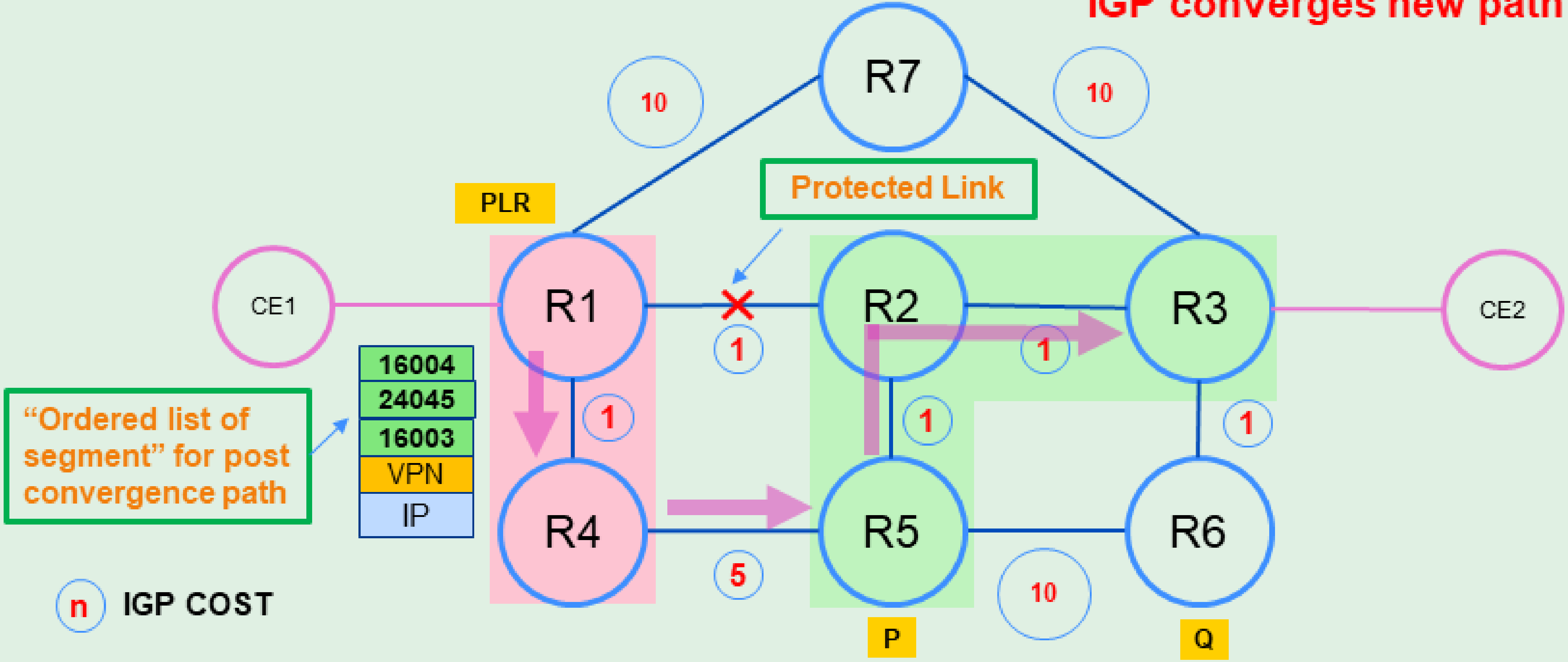


"Ordered list of segment" for primary path
16003
VPN
IP
PLR
"Ordered list of segment" for post convergence path
16004
24045
16003
VPN
IP
R7
10
10
Protected Link
CE1
R1
R2
R3
CE2
1
1
1
1
1
R4
R5
R6
5
10
P
Q
n
IGP COST

FRR with TI-LFA

TI-LFA Link Protection

When protected link goes down R1 instantly use post convergence SID-list even before IGP converges new path



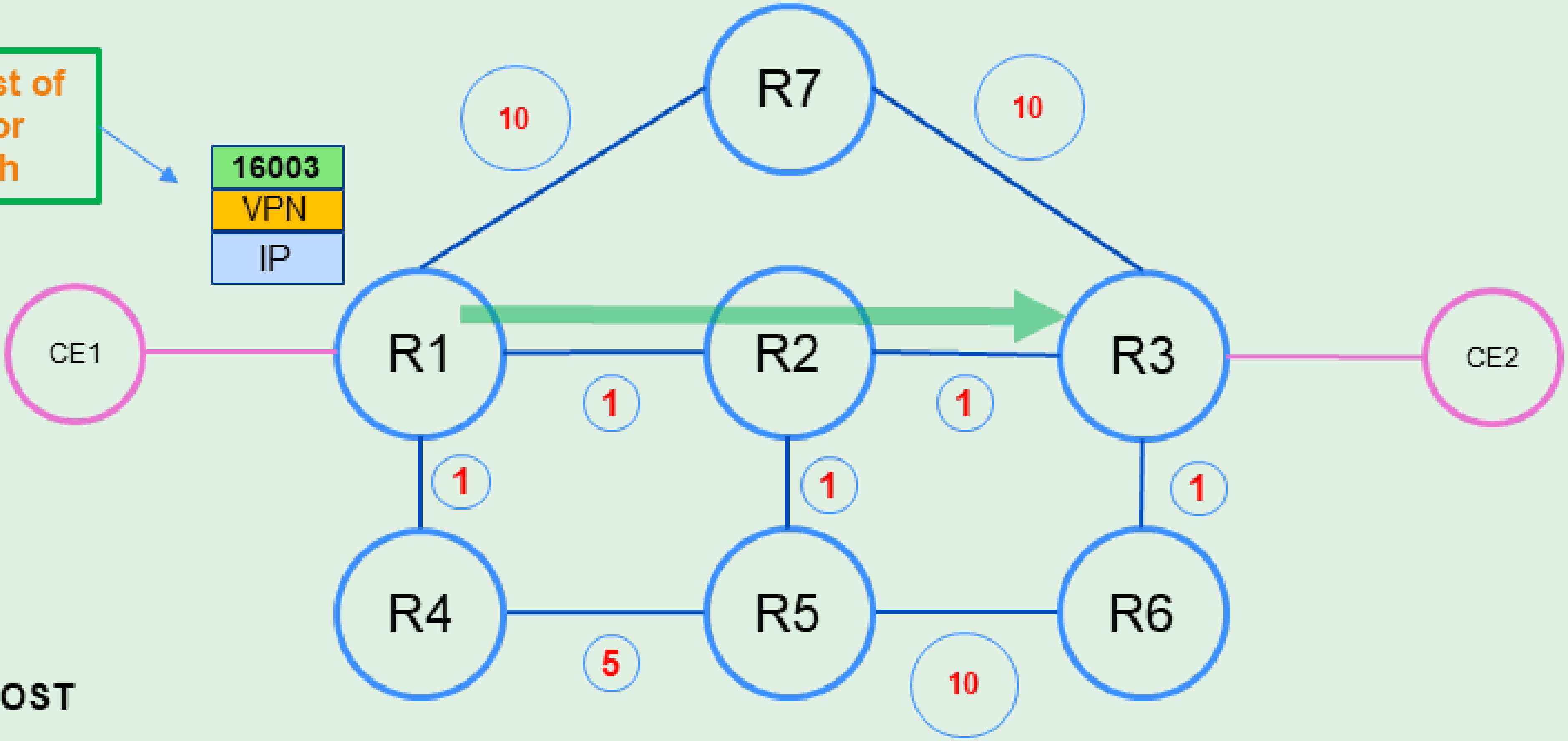
n IGP COST

FRR with TI-LFA

TI-LFA Node Protection

Topology is fully functional
R1's Primary path is in forwarding table

"Ordered list of segment" for primary path

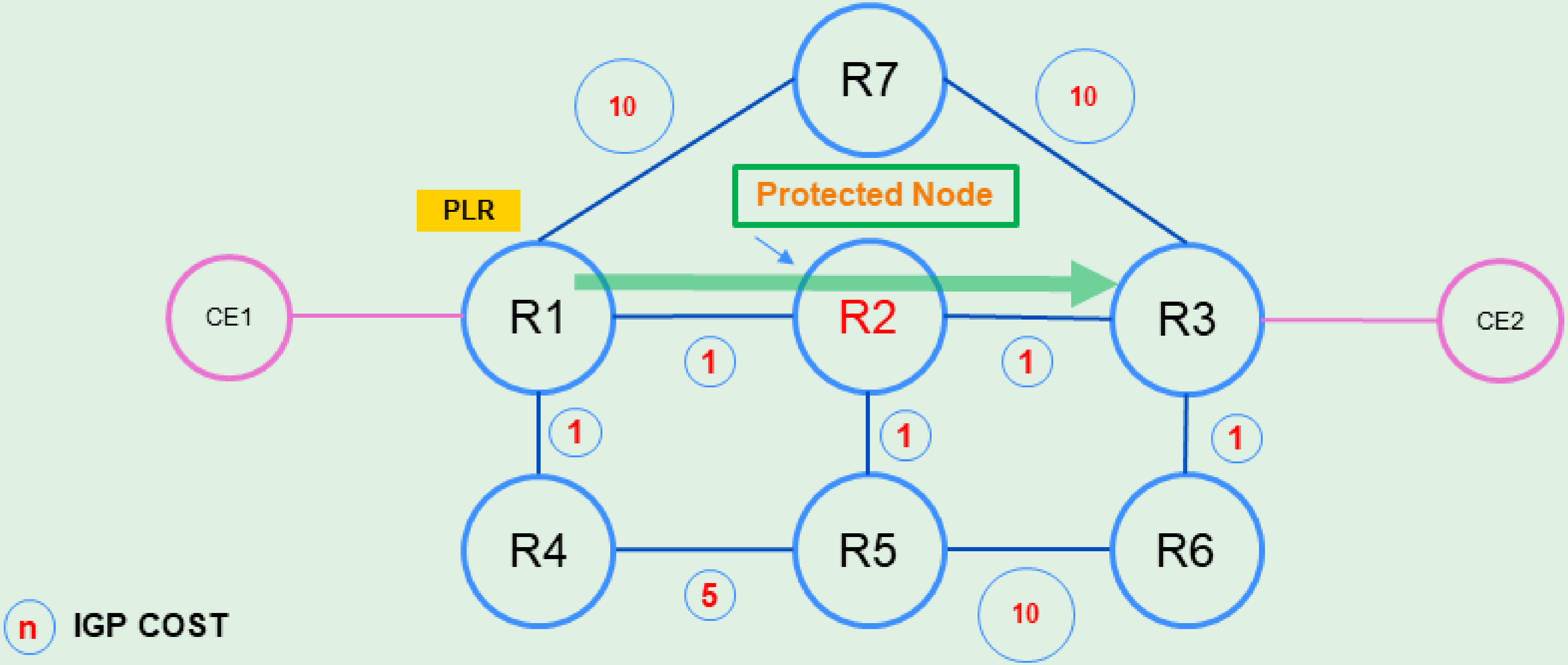


n IGP COST

FRR with TI-LFA

TI-LFA Node Protection

R1 is aware of the backup Node in advance in case of failure of R2

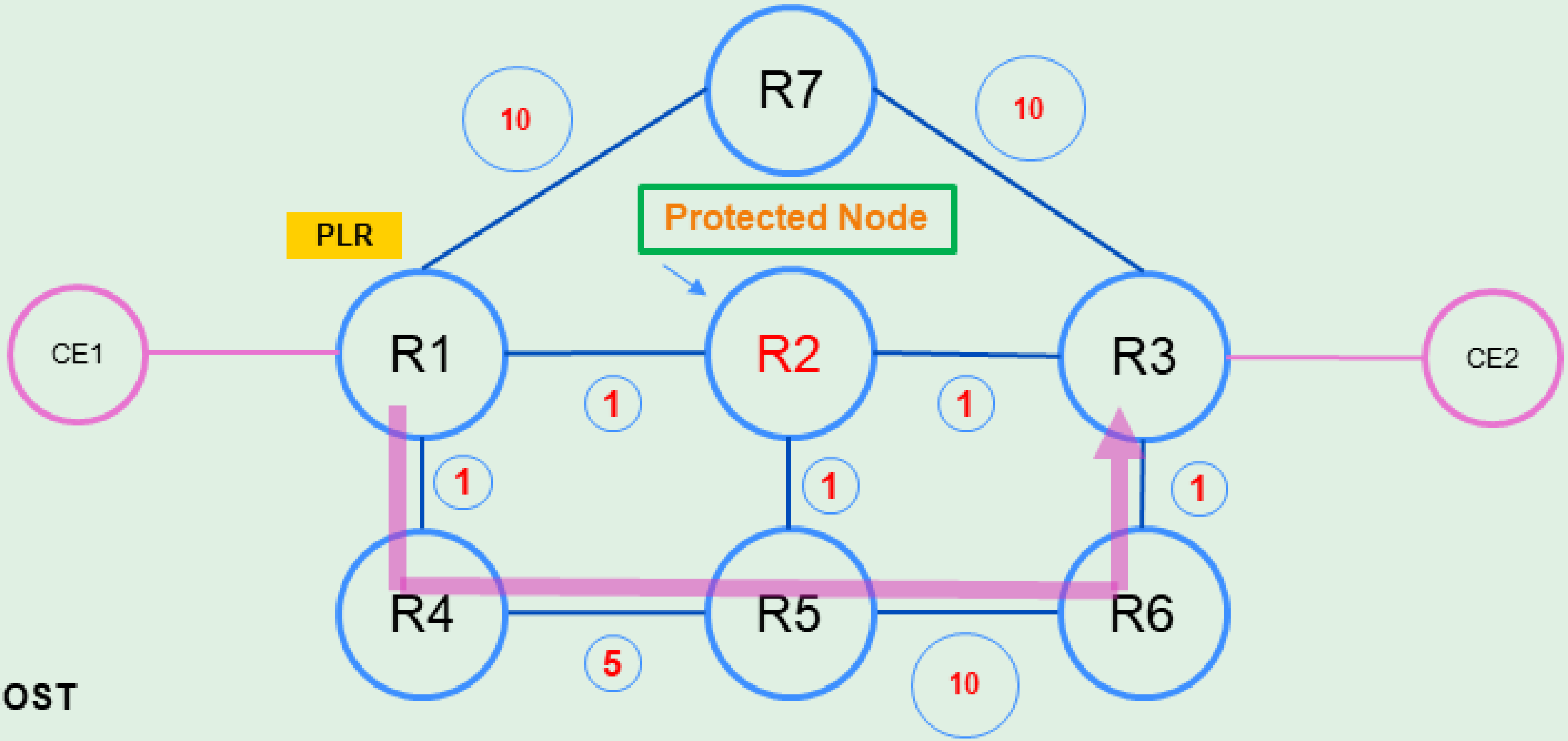


Prepared by - M Abdullah Al Naser

FRR with TI-LFA

TI-LFA Node Protection

R1 prunes protected node before node failure occurs and find out post convergence path



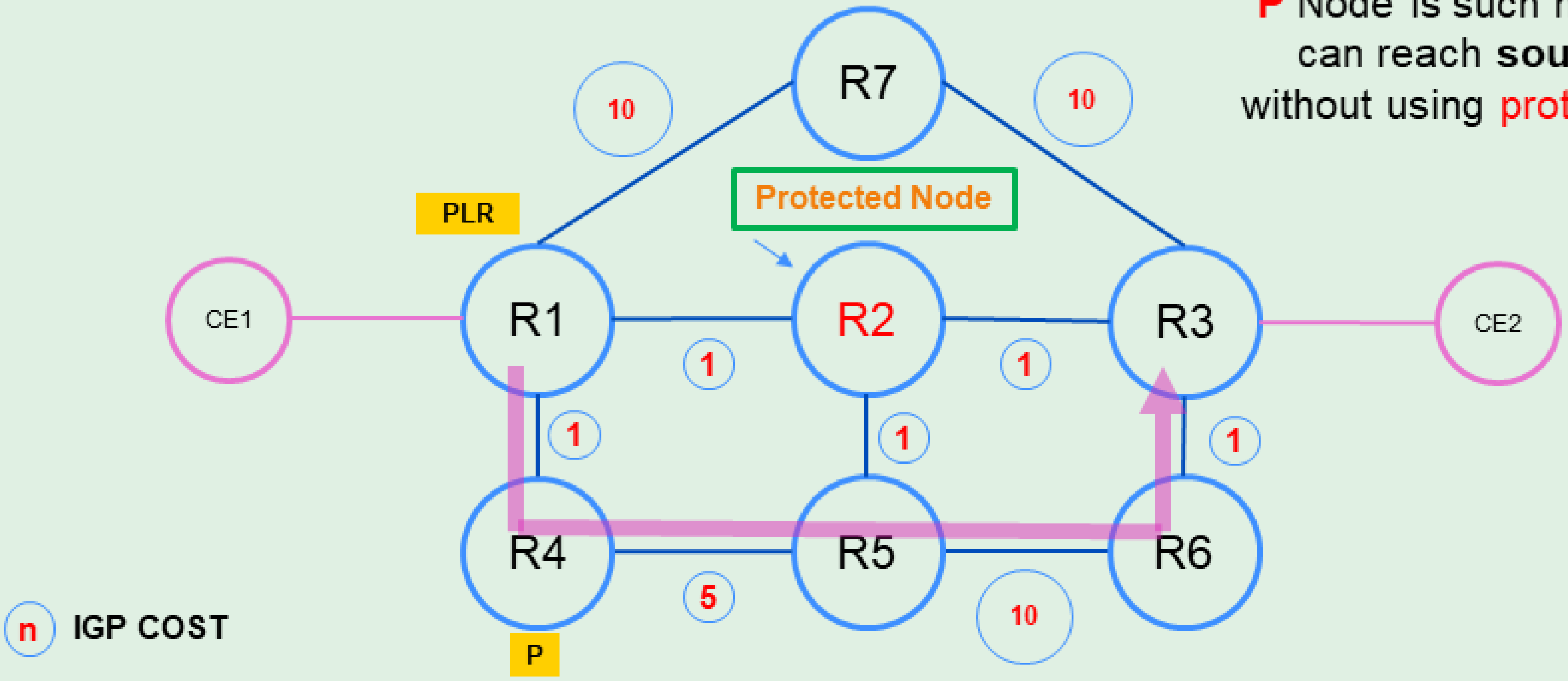
IGP COST

FRR with TI-LFA

TI-LFA Node Protection

R1 defines the P Node

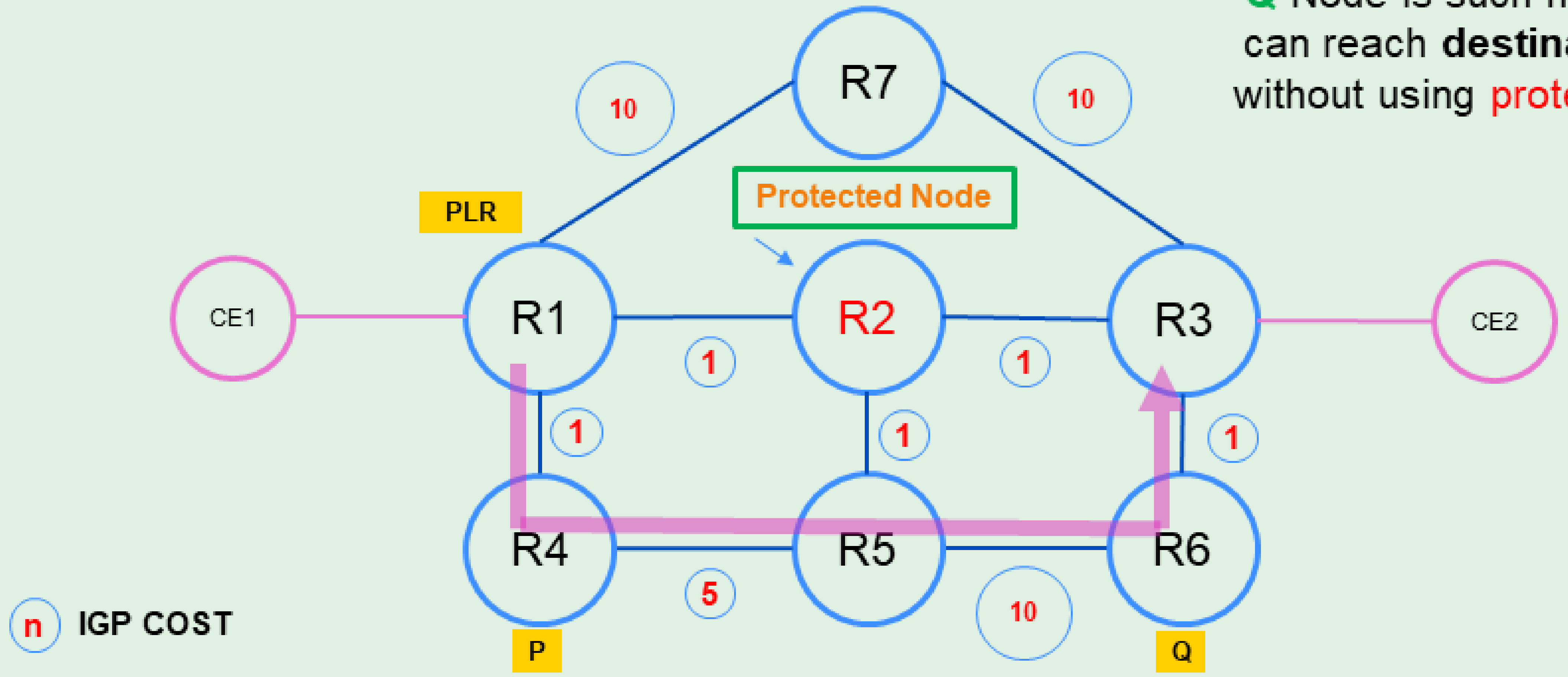
P Node is such node which can reach **source (R1)** without using protected node



Prepared by - M Abdullah Al Naser

FRR with TI-LFA

TI-LFA Node Protection



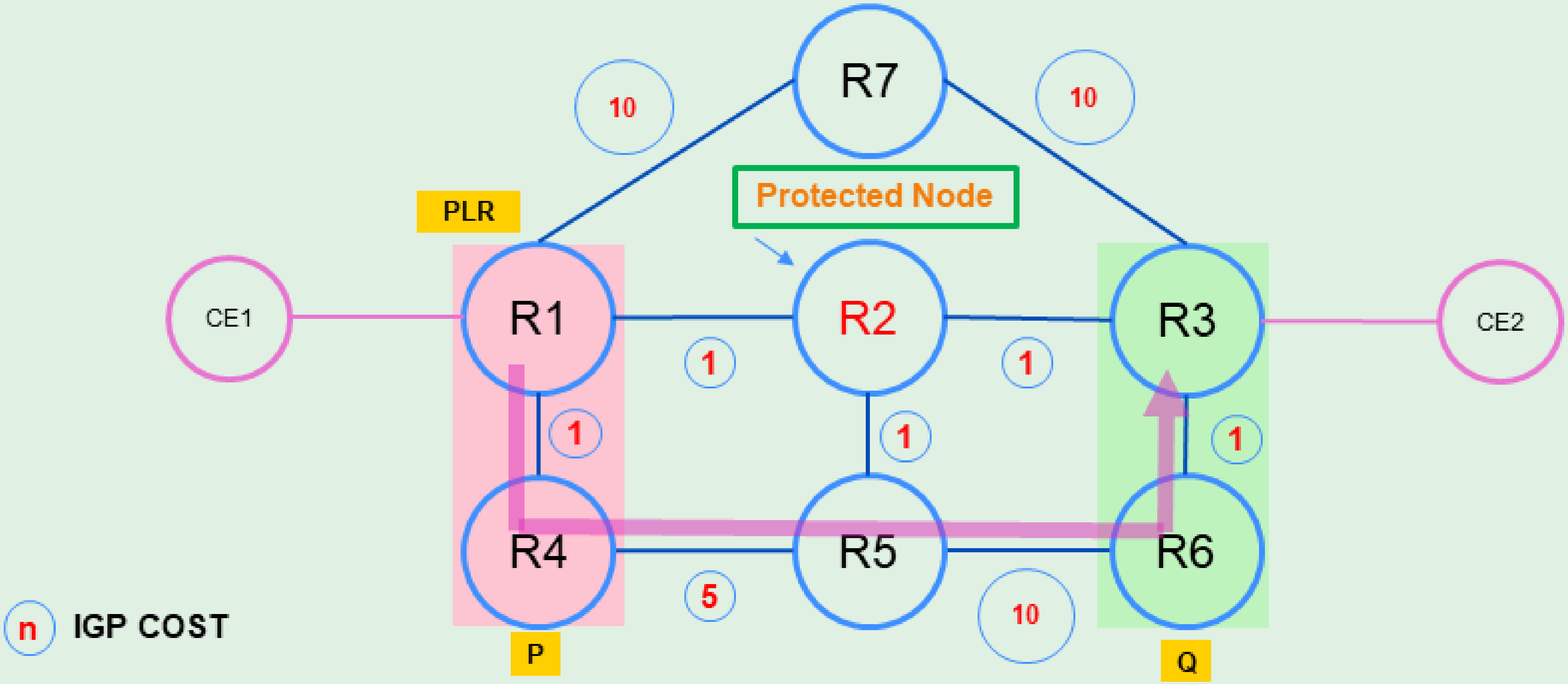
R1 defines the Q Node

Q Node is such node which can reach **destination (R3)** without using protected node

FRR with TI-LFA

TI-LFA Node Protection

P Space and Q Space

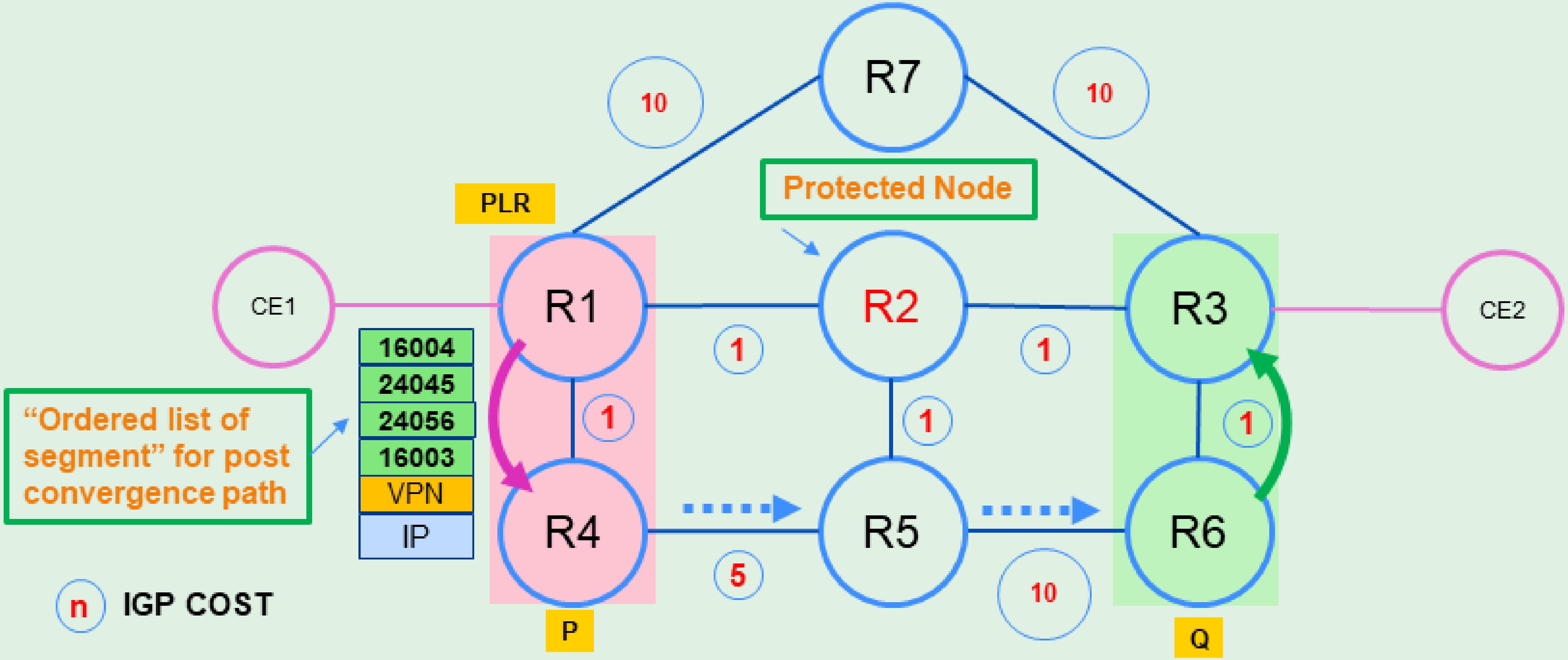


Prepared by - M Abdullah Al Naser

FRR with TI-LFA

TI-LFA Node Protection

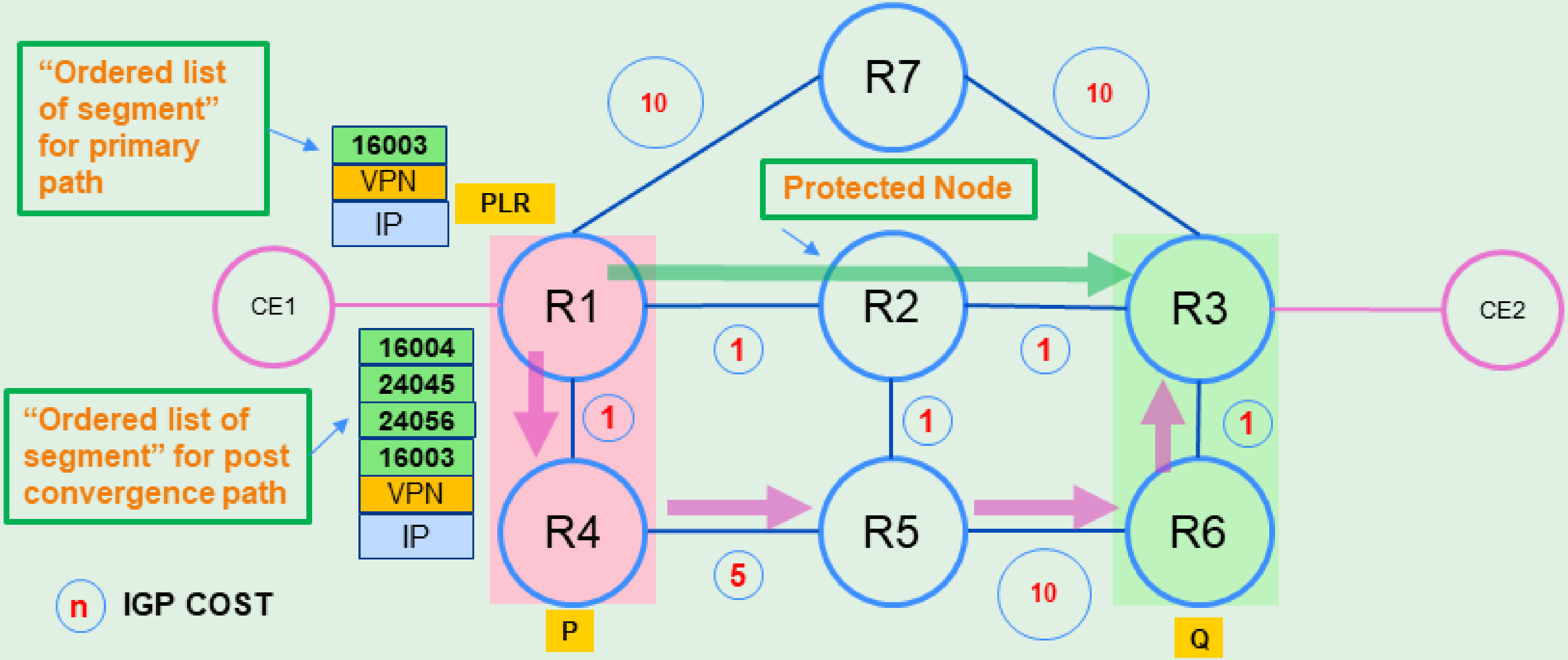
R1 builds SID-list based on each segment it found along the post convergence path



FRR with TI-LFA

TI-LFA Node Protection

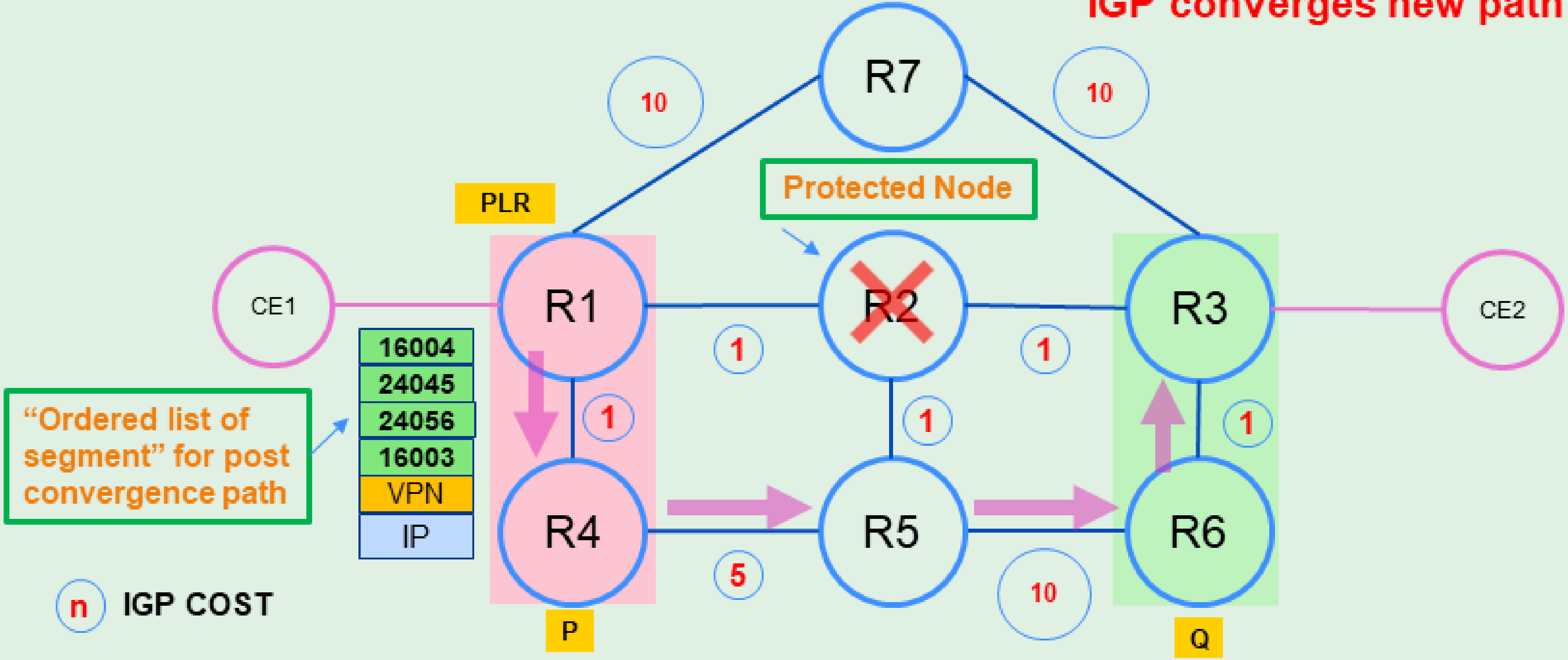
R1 programs the precomputed SID-List it in forwarding table



FRR with TI-LFA

TI-LFA Node Protection

When protected node goes down R1 instantly use post convergence SID-list even before IGP converges new path



16004
24045
24056
16003
VPN
IP

“Ordered list of segment” for post convergence path

n IGP COST

FRR with TI-LFA

Fast Reroute with TI-LFA

- SR must be implemented to have TI-LFA
- 100% topology coverage
- Avoids micro-loop
- ECMP friendly
- Protects IP and LDP traffic also

SR-MPLS Fast Convergence

TI-LFA: Few more…

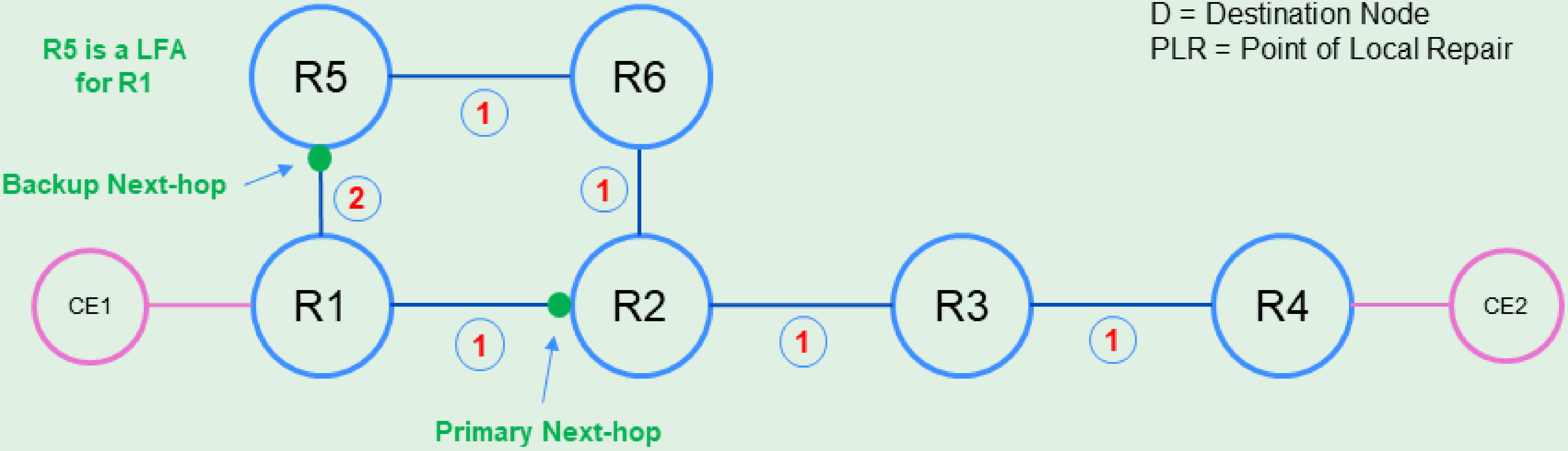
FRR Conditions

LFA Condition

- Distance (N, D) < Distance (N, PLR) + Distance (PLR, D)

Case 1:

N = LFA Candidate
D = Destination Node
PLR = Point of Local Repair



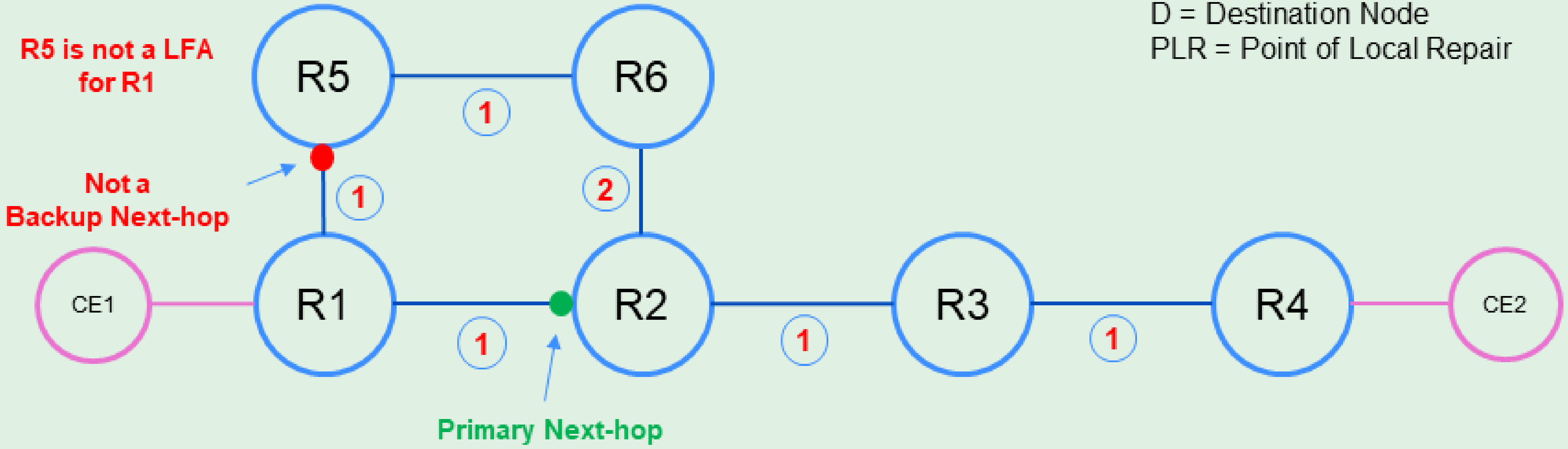
FRR Conditions

LFA Condition

- Distance (N, D) < Distance (N, PLR) + Distance (PLR, D)

Case 2:

N = LFA Candidate
D = Destination Node
PLR = Point of Local Repair



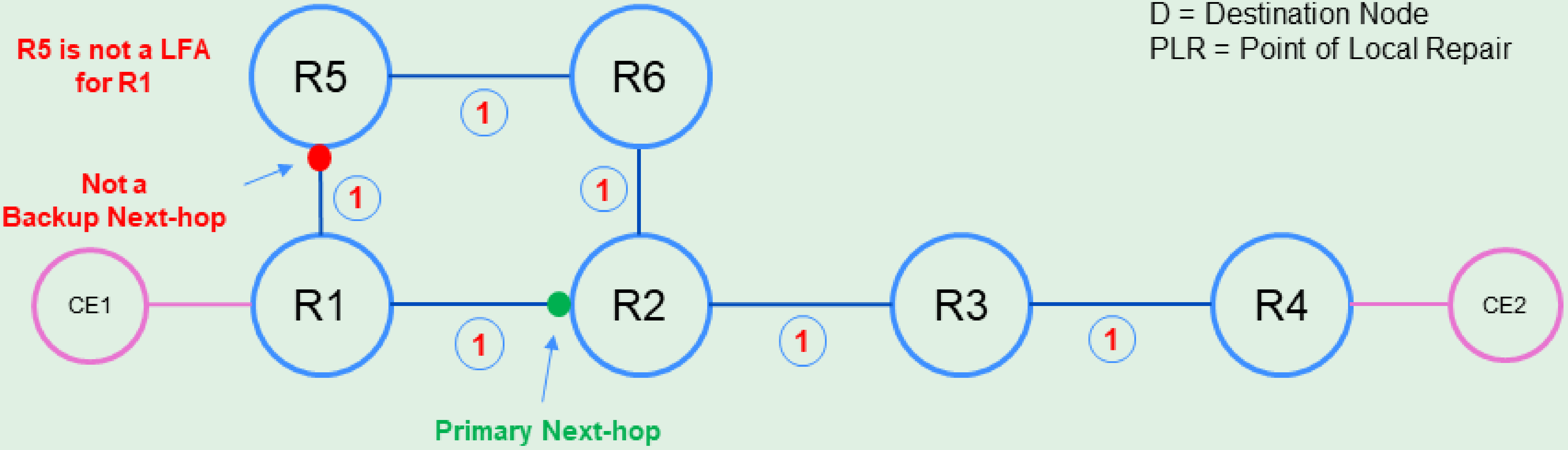
FRR Conditions

LFA Condition

- Distance (N, D) < Distance (N, PLR) + Distance (PLR, D)

Case 3:

N = LFA Candidate
D = Destination Node
PLR = Point of Local Repair

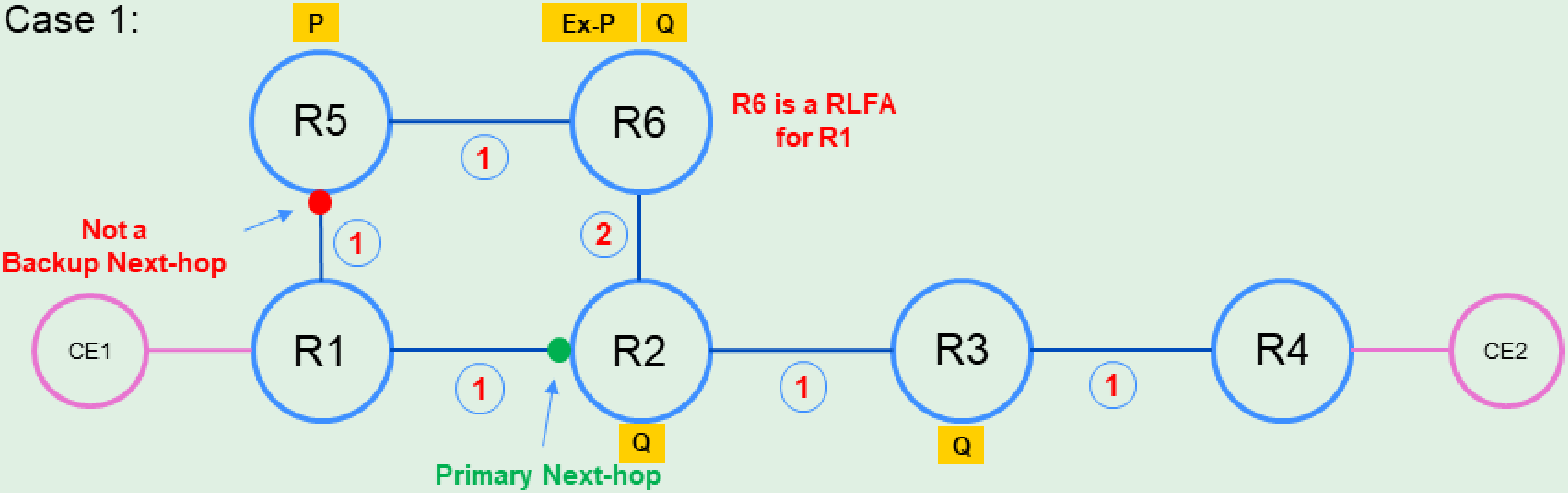


FRR Conditions

RLFA Condition

- **P** node: Closest Node to PLR whose backup path is not via protected link
- **Q** node: Closest Node to DESTINATION whose backup path is not via protected link
- A **PQ** node can be used as RLFA

Case 1:

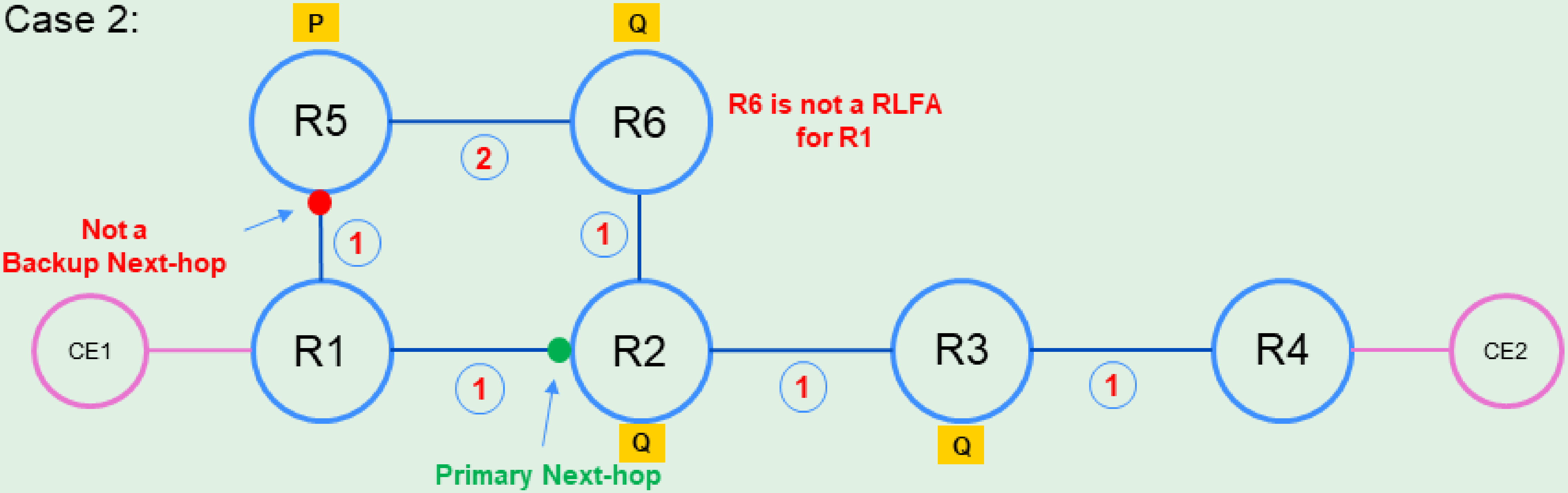


FRR Conditions

RLFA Condition

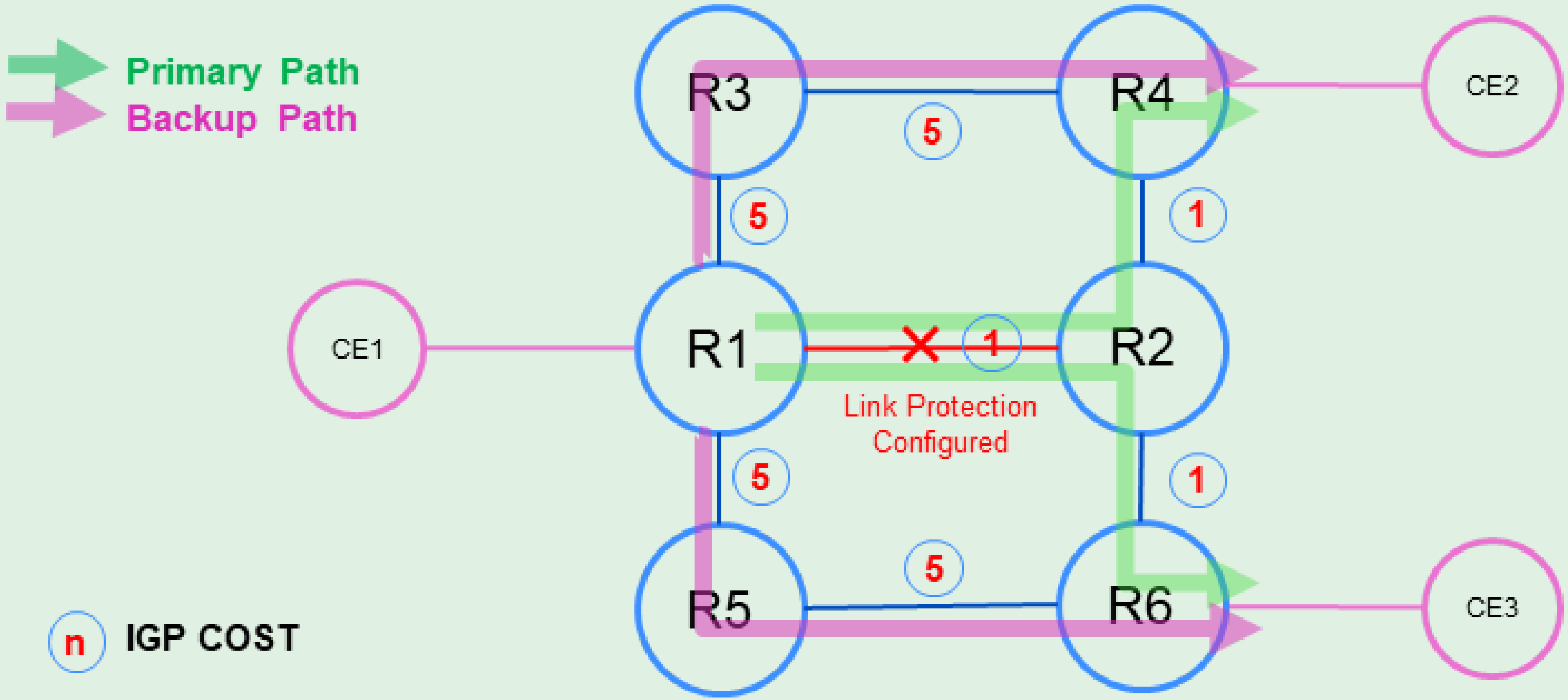
- **P** node: Closest Node to PLR whose backup path is not via protected link
- **Q** node: Closest Node to DESTINATION whose backup path is not via protected link
- A **PQ** node can be used as RLFA

Case 2:



TI-LFA Protection

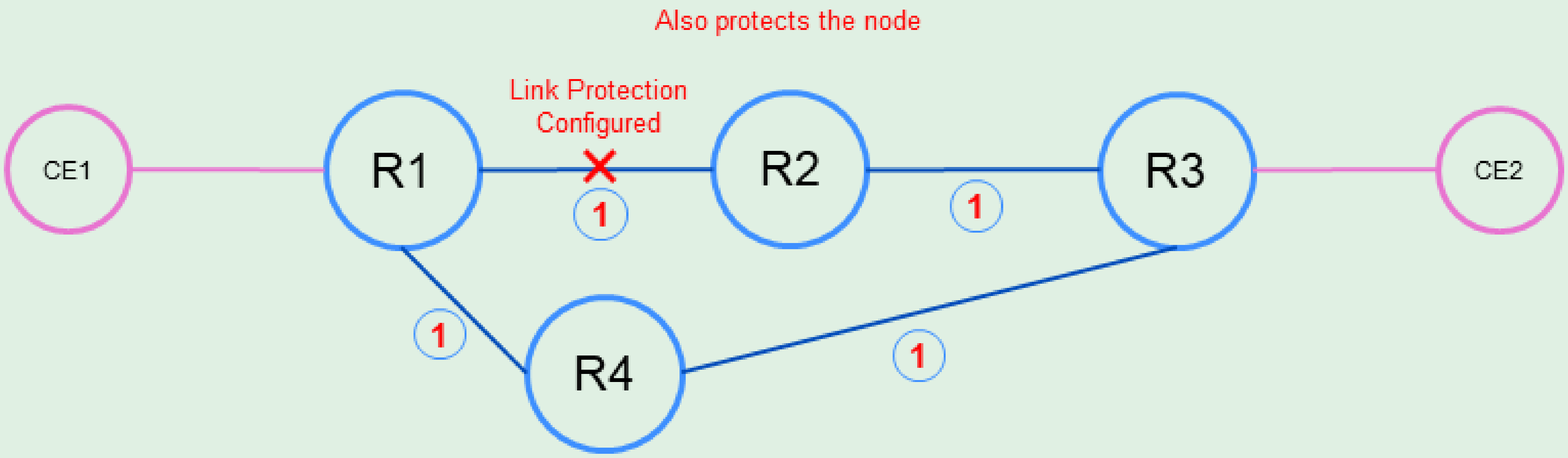
- Destination (Prefix) based FRR; not based on links
  - Results in more optimized path



TI-LFA Protection

- Link protection *sometimes* can protect the node also
- Node protection can *always* protect the link

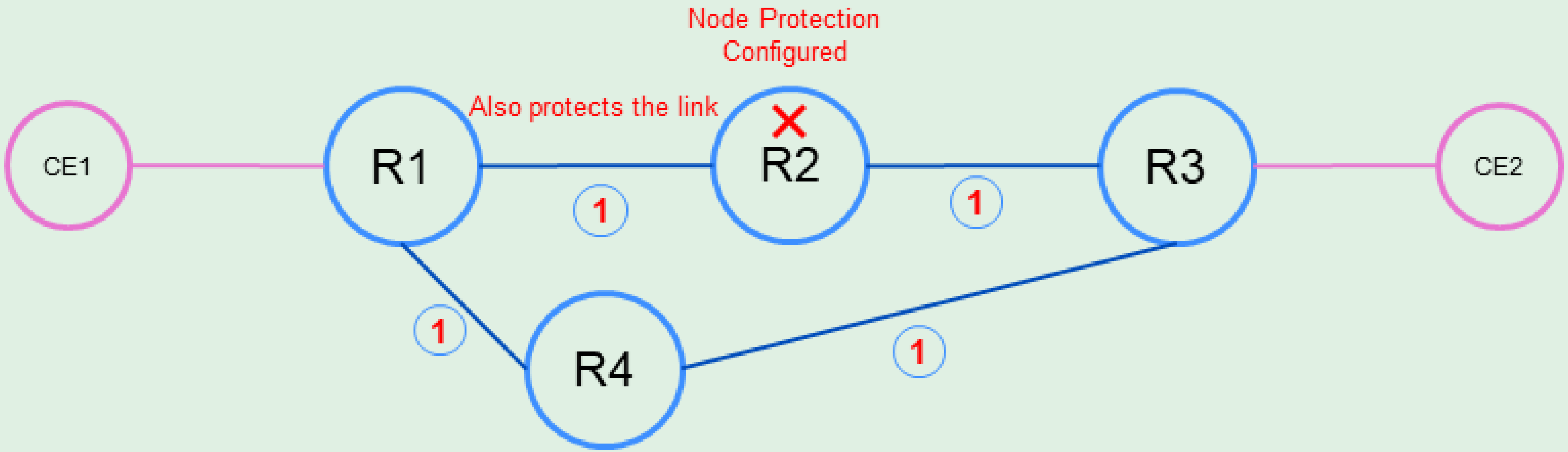
Case: 1



TI-LFA Protection

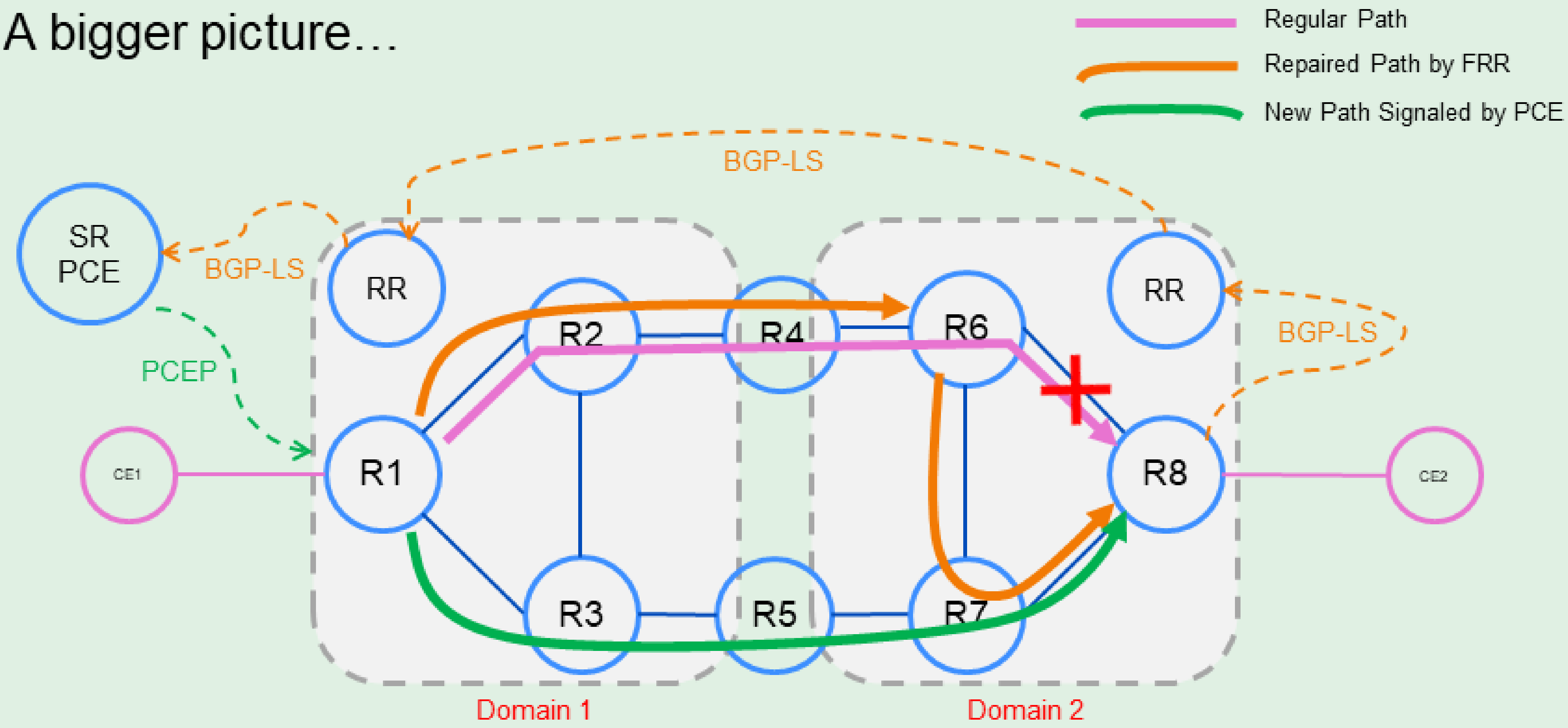
- Link protection *sometimes* can protect the node also
- Node protection can *always* protect the link

Case: 2



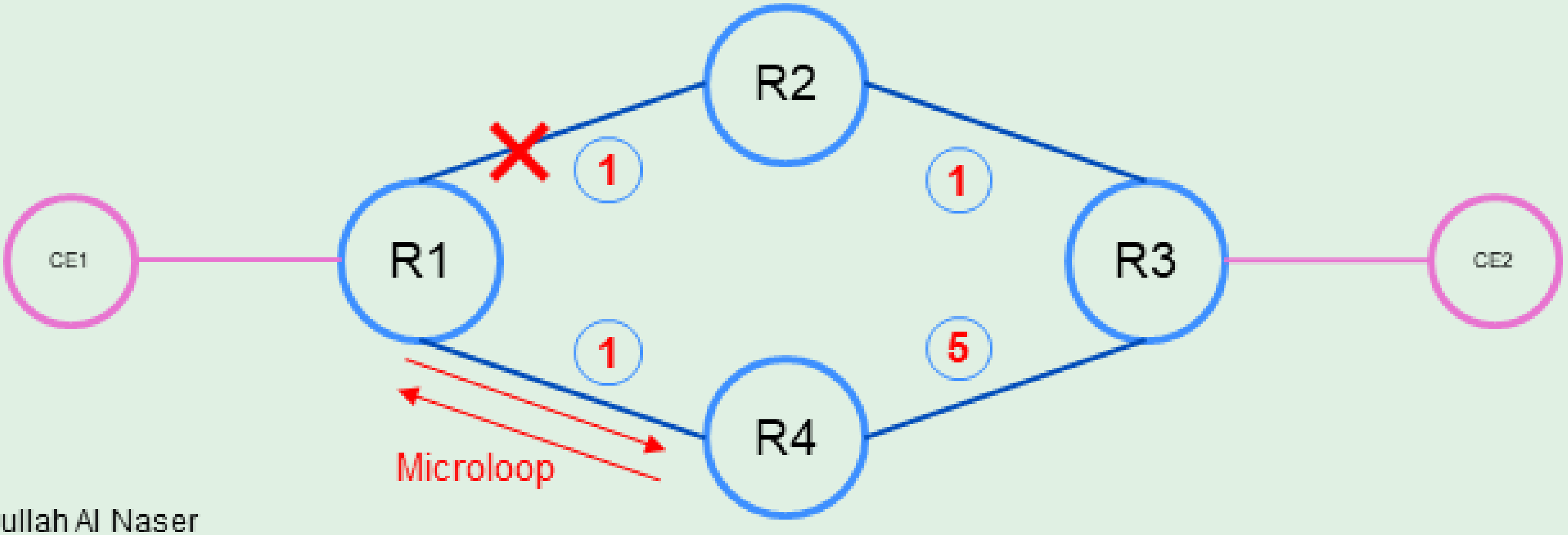
TI-LFA Protection

- A bigger picture…



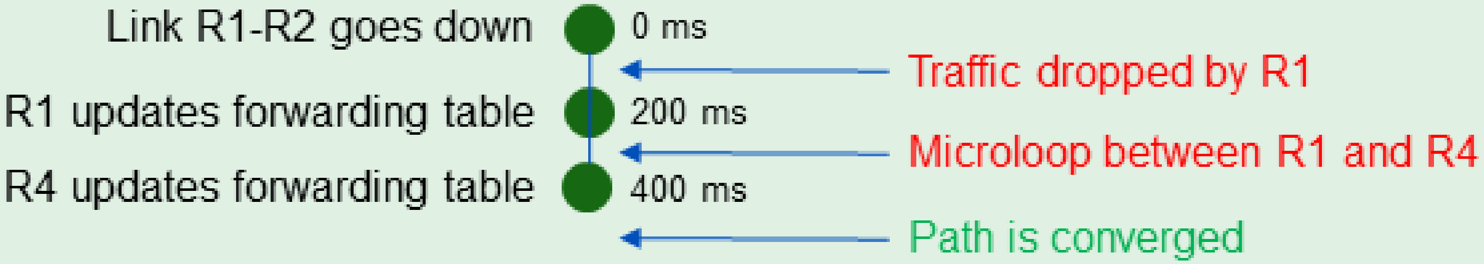
Microloop

- Microloop: Transient packet forwarding loop during path convergence
- Reasons of microloop:
  - Topological change propagation may need few times
  - Forwarding table may not be updated at the same time or in a proper order
  - Depends on CPU, Memory and link utilization also
  - Intermittent Link flapping



Microloop

- Without TI-LFA
  - Causes packet loss due to microloop



Link R1-R2 goes down — 0 ms

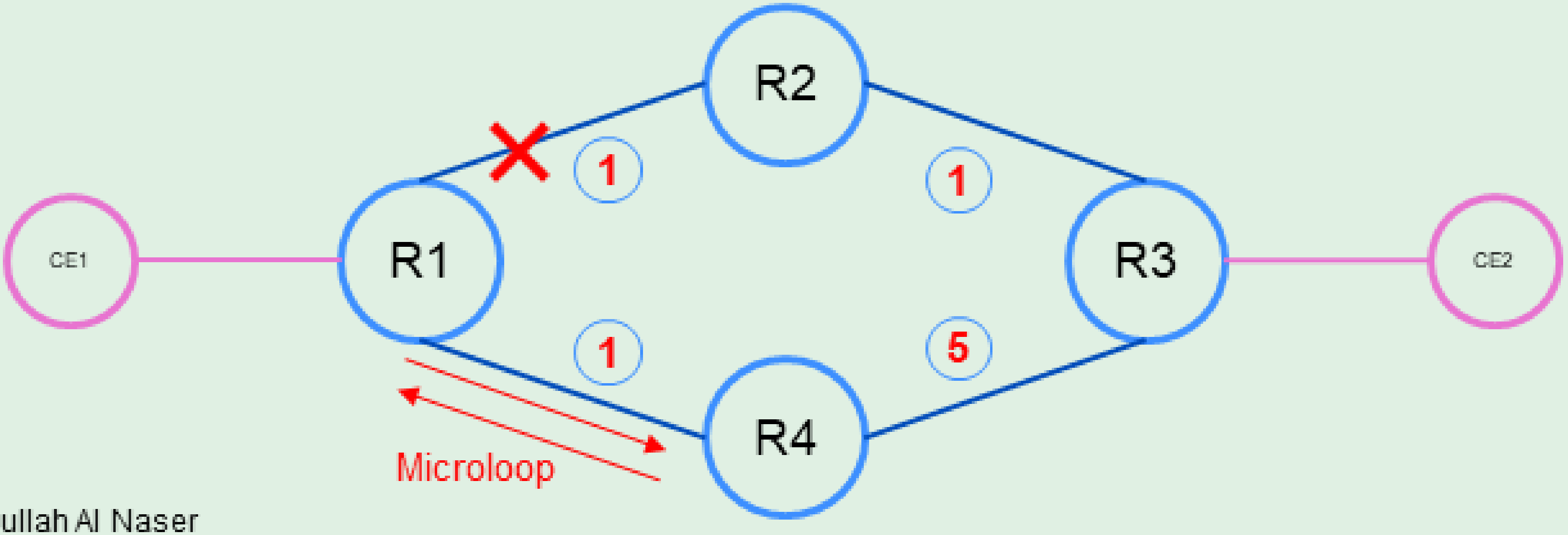
Traffic dropped by R1

R1 updates forwarding table — 200 ms

Microloop between R1 and R4

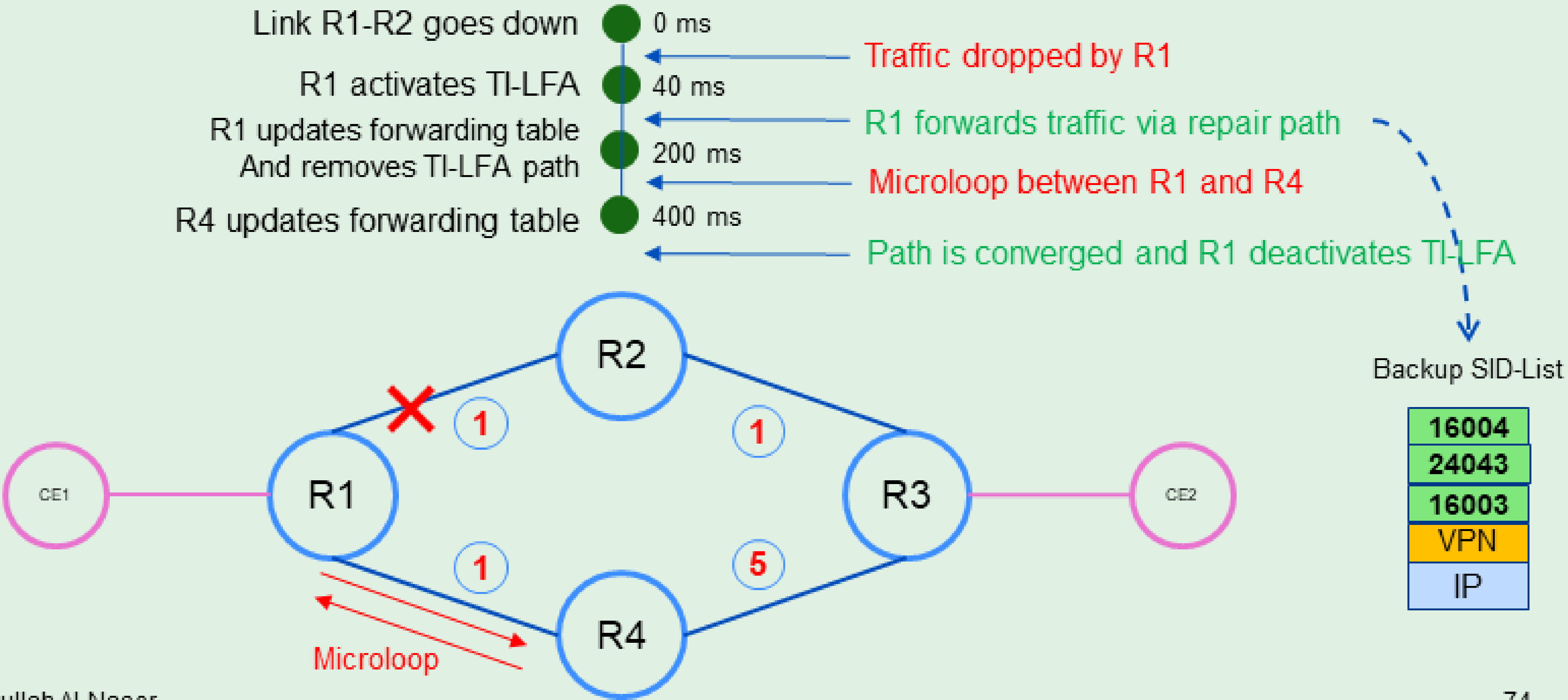
R4 updates forwarding table — 400 ms

Path is converged



Microloop

- With TI-LFA; but no microloop avoidance enabled
  - Much faster convergence but still microloop is there



Link R1-R2 goes down — 0 ms

Traffic dropped by R1

R1 activates TI-LFA — 40 ms

R1 forwards traffic via repair path

R1 updates forwarding table
And removes TI-LFA path — 200 ms

Microloop between R1 and R4

R4 updates forwarding table — 400 ms

Path is converged and R1 deactivates TI-LFA

Backup SID-List

16004
24043
16003
VPN
IP

Microloop Avoidance

- With TI-LFA; and microloop avoidance enabled
  - Much faster convergence and no microloop is there

Link R1-R2 goes down ● 0 ms

Traffic dropped by R1

R1 activates TI-LFA ● 40 ms

R1 forwards traffic via repair path

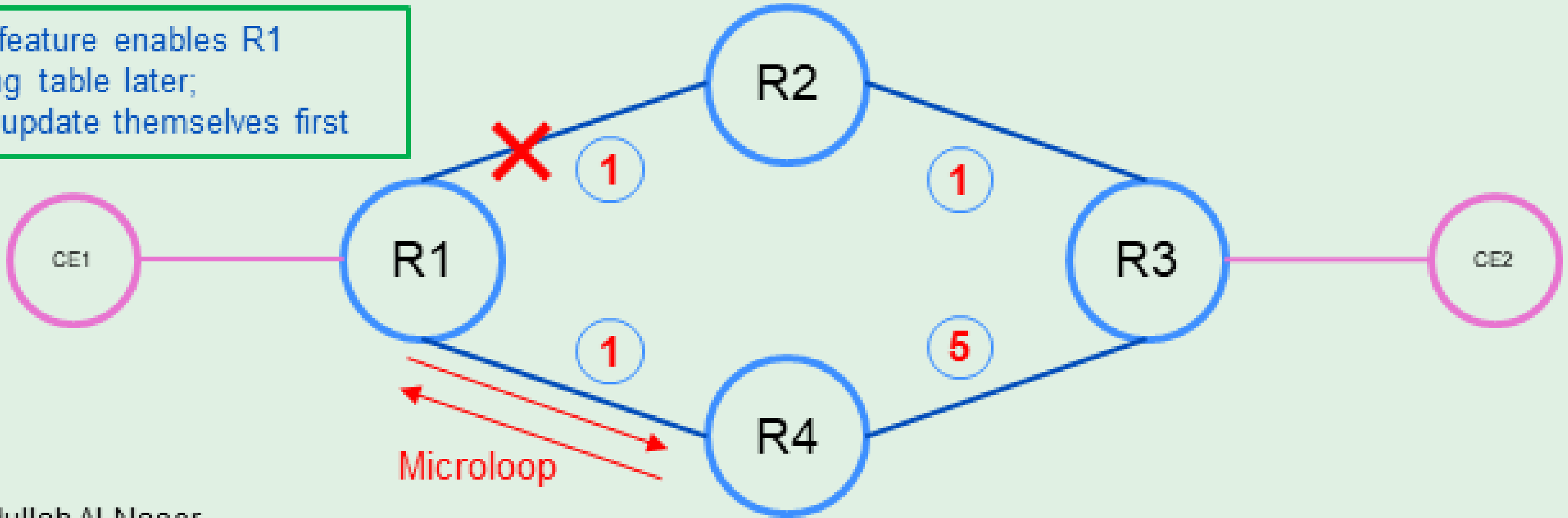
R4 updates forwarding table ● 400 ms

Traffic forwarded via repair path (Continued)

R1 updates forwarding table
And removes TI-LFA path ● 5000 ms

Path is converged and R1 deactivates TI-LFA

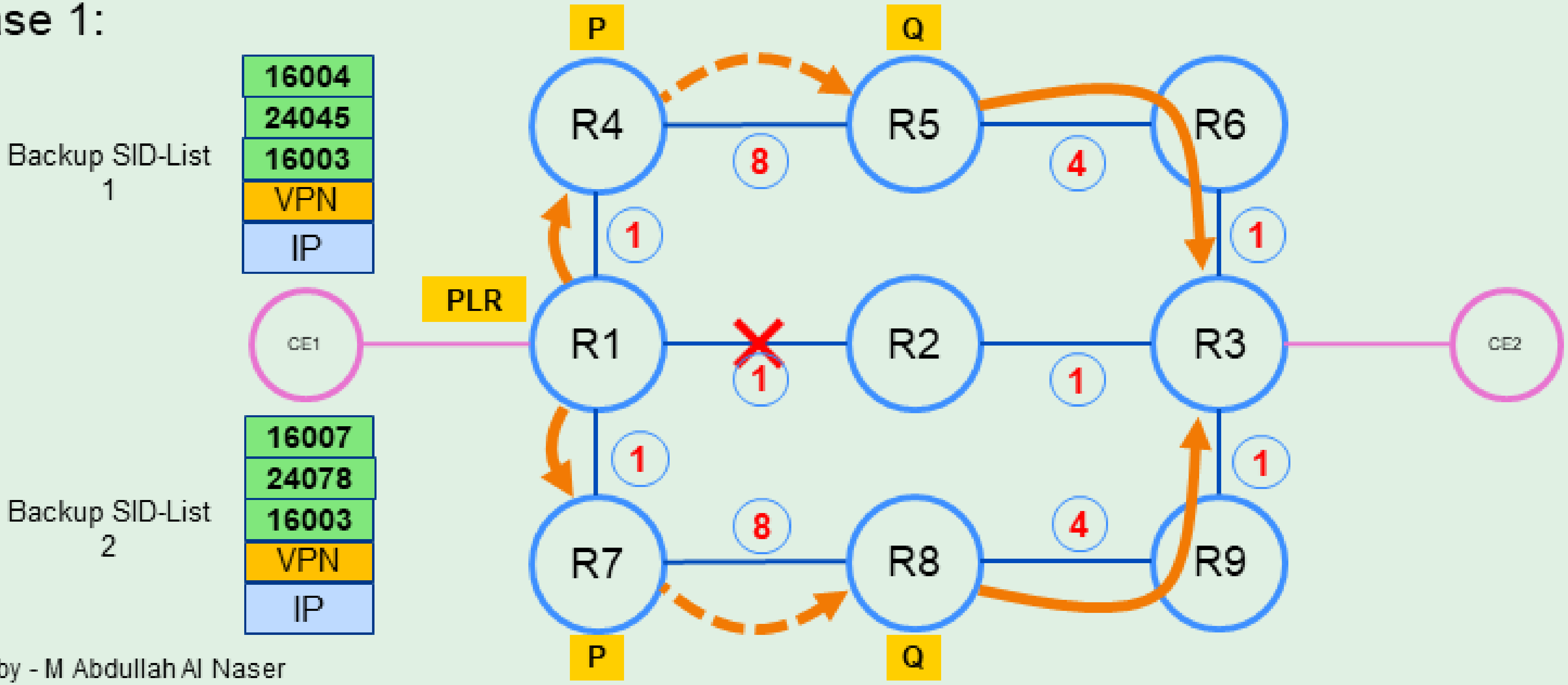
Microloop avoidance feature enables R1
to update its forwarding table later;
to let other routers to update themselves first



TI-LFA Load Balancing

- Prefix-SID is ECMP aware
- PLR can load balance over multiple PQ pairs
  - Selects **one PQ pair per destination** (based on hash function)

Case 1:



Backup SID-List 1

16004
24045
16003
VPN
IP

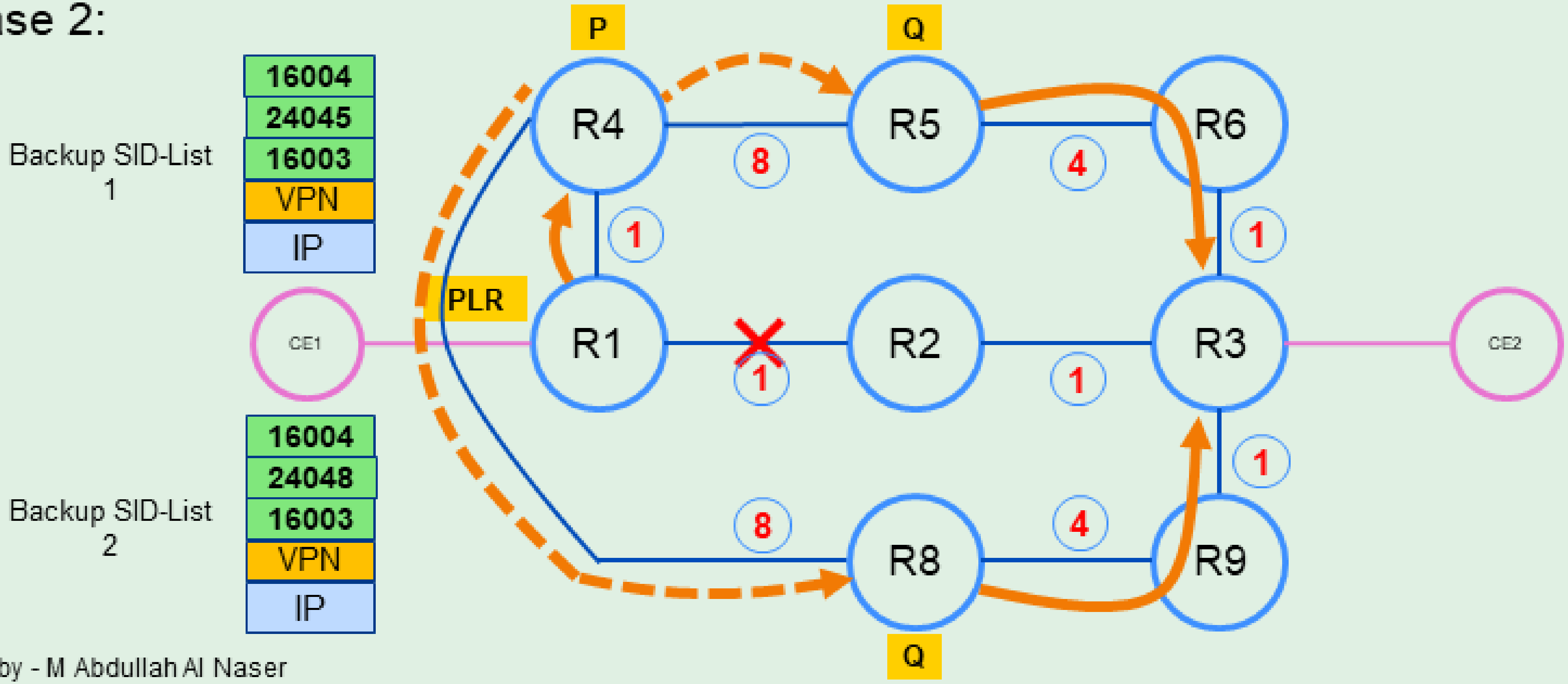
Backup SID-List 2

16007
24078
16003
VPN
IP

TI-LFA Load Balancing

- Prefix-SID is ECMP aware
- PLR can load balance over multiple PQ pairs
  - Selects **one PQ pair per destination** (based on hash function)

Case 2:



TI-LFA Load Balancing

- Prefix-SID is ECMP aware
- PLR can load balance over multiple PQ pairs
  - Selects **one PQ pair per destination** (based on hash function)

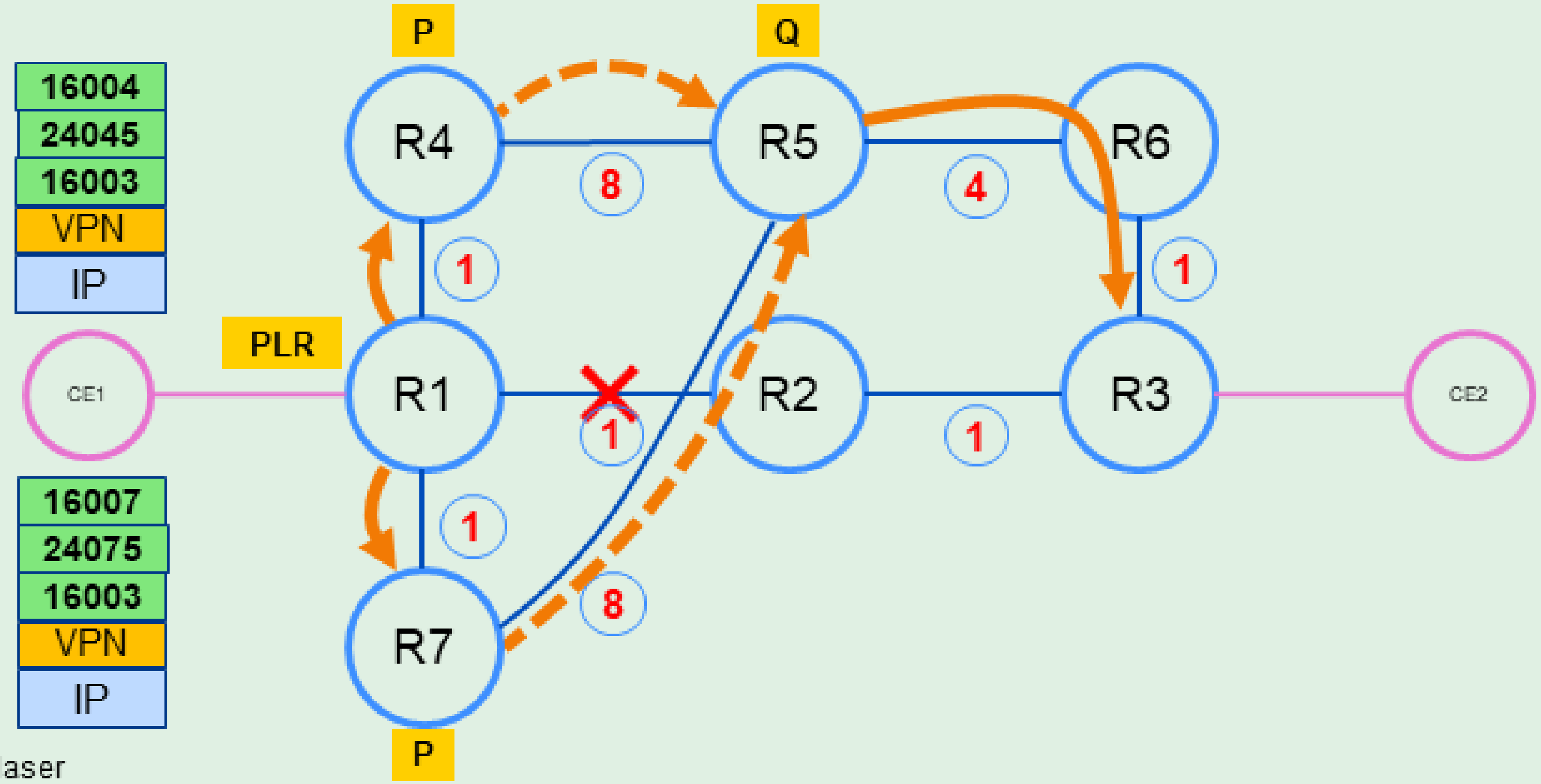
Case 3:

Backup SID-List 1

16004
24045
16003
VPN
IP

Backup SID-List 2

16007
24075
16003
VPN
IP



SR-MPLS Fast Convergence

Lab - Demo of *Microloop Prevention and Load Balancing*

SR-MPLS Fast Convergence

Lab - Demo of *TI-LFA FRR* vs RSVP FRR

SR-MPLS Traffic Engineering

Segment Routing TE (SR-TE)

Please Note:

- This slide deck is a continuation of previous one “SR-MPLS 101”
- Audience are expected to have prior basic knowledge on MPLS and Segment Routing

SR-TE

Segment Routing Traffic Engineering

- To steer traffic through longer path
- SR Policy to be implemented at ingress PE
  - No more tunnel from head-end to tail-end
- No state to be maintained at intermediate nodes
  - No RSVP-TE means no additional signalling
- Supports built-in FRR with TI-LFA
- ECMP friendly

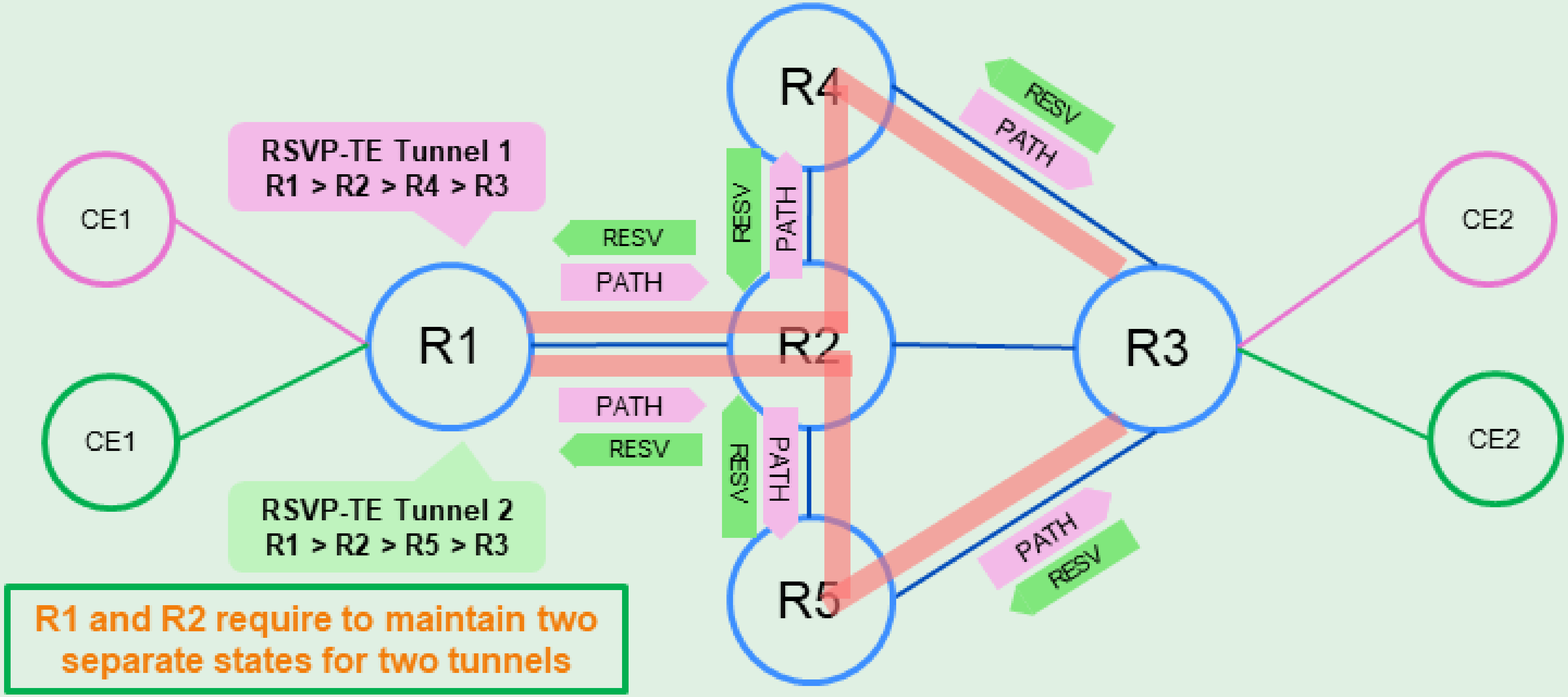
SR-TE

Segment Routing Traffic Engineering

- SR Policy
  - Intended design
  - Highly scalable
  - Better optimization
  - Automated traffic steering
- Can be applied in either centralized or distributed manner
  - Explicit Path: CLI or SDN Controller
  - Dynamic Path: Based on constrain (i.e link delay)

SR-TE

- **RSVP-TE**



RSVP-TE Tunnel 1
R1 > R2 > R4 > R3

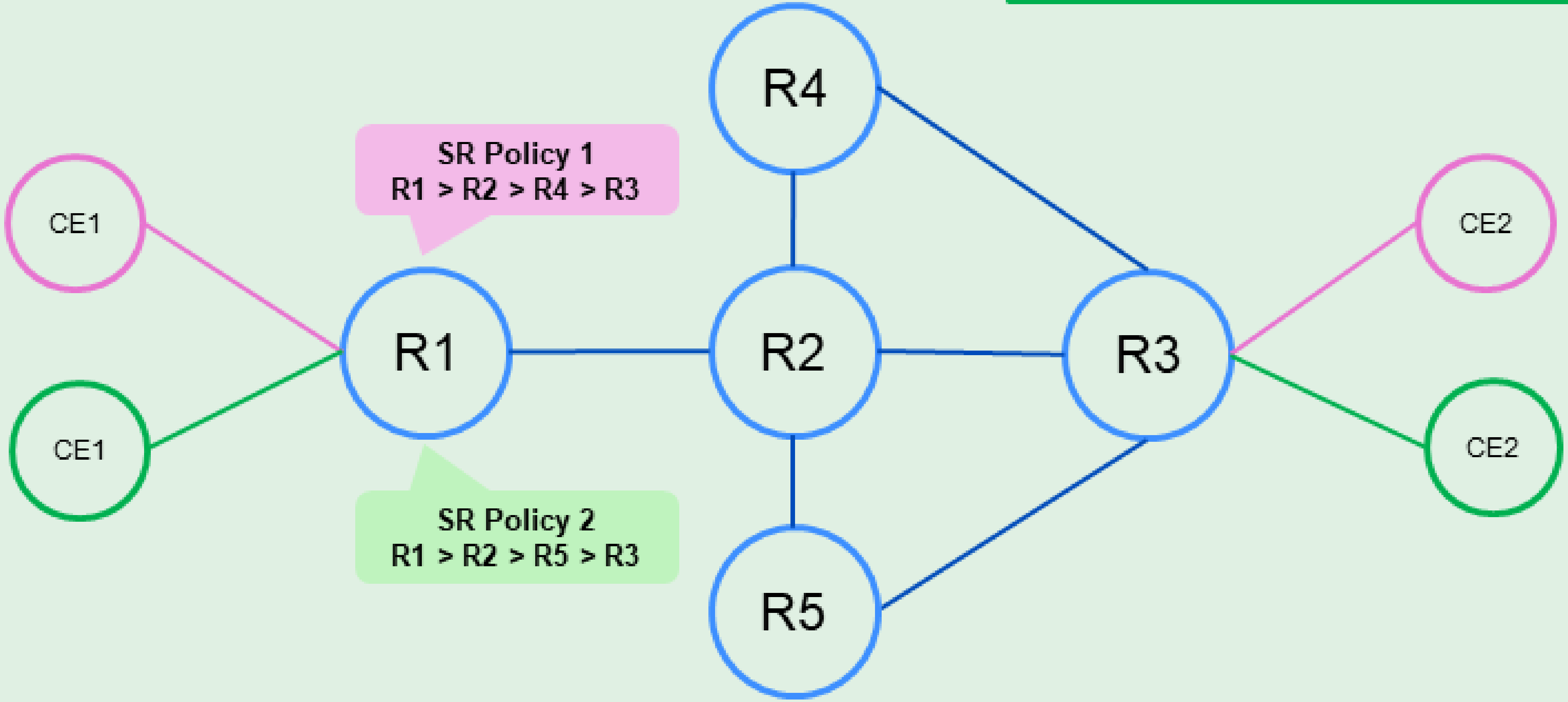
RSVP-TE Tunnel 2
R1 > R2 > R5 > R3

R1 and R2 require to maintain two separate states for two tunnels

SR-TE

- **SR-TE**

No tunnel required

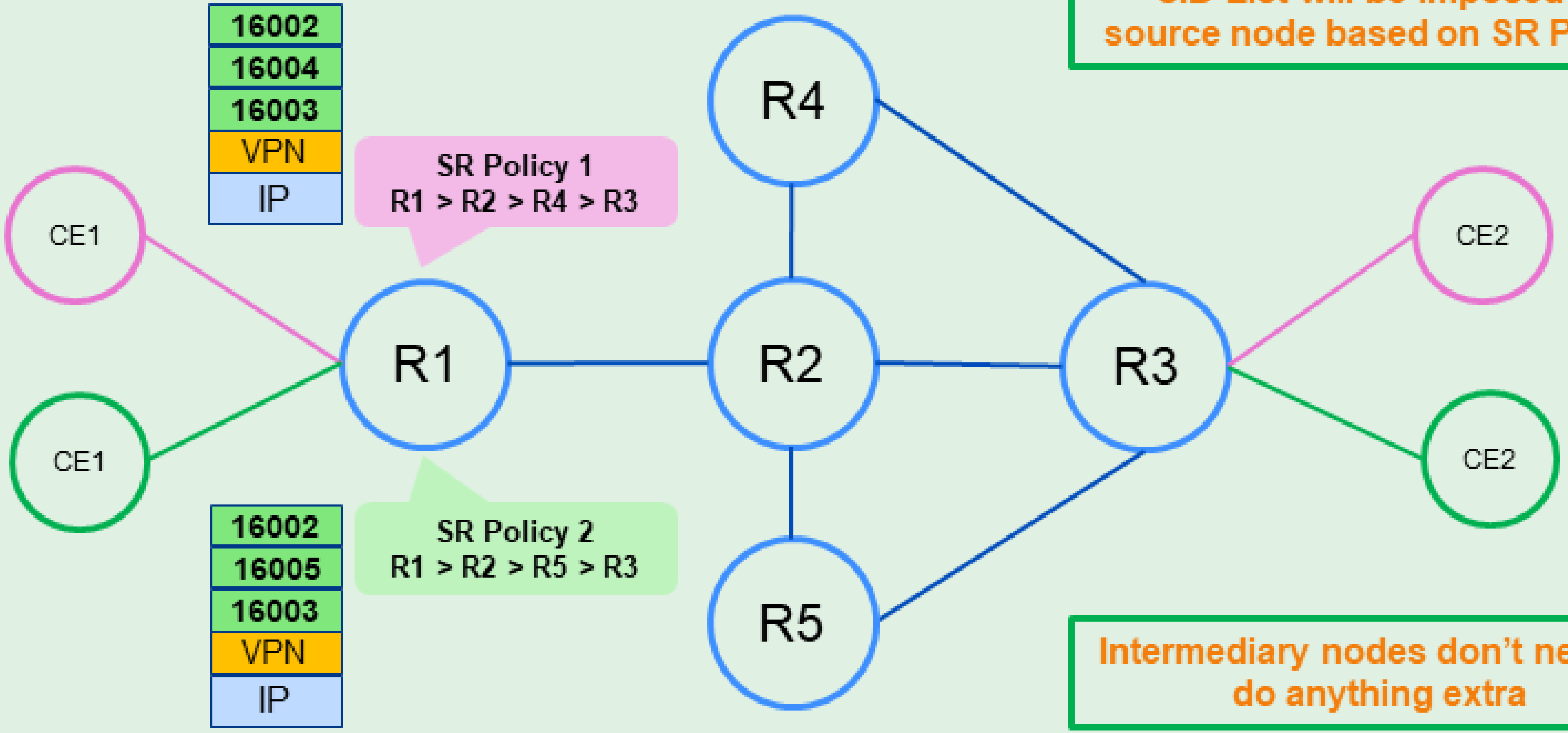


SR Policy 1
R1 > R2 > R4 > R3

SR Policy 2
R1 > R2 > R5 > R3

SR-TE

- **SR-TE**



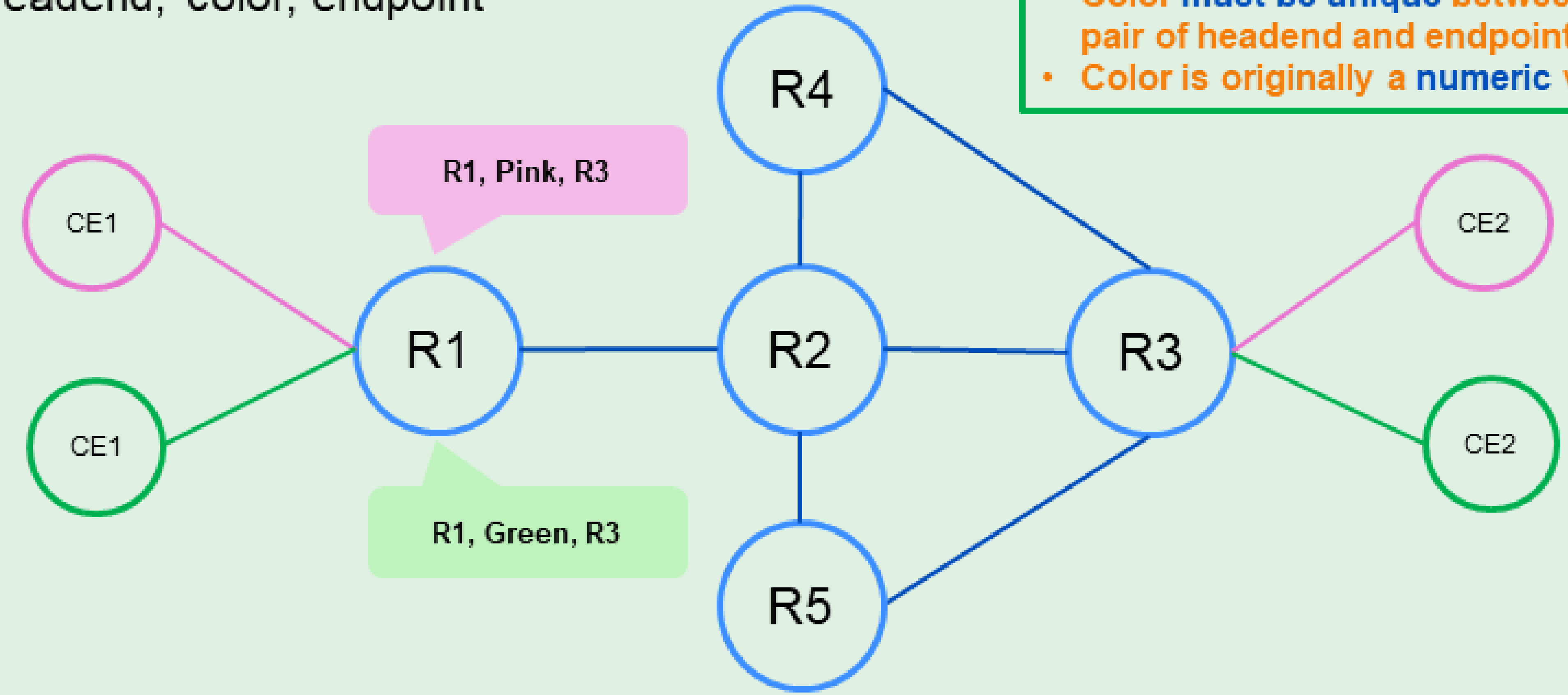
SID-List will be imposed at source node based on SR Policy

Intermediary nodes don't need to do anything extra

SR-TE

- **SR Policy**
  - headend, color, endpoint

- Color is used to express the intent
- Color must be unique between a pair of headend and endpoint
- Color is originally a numeric value

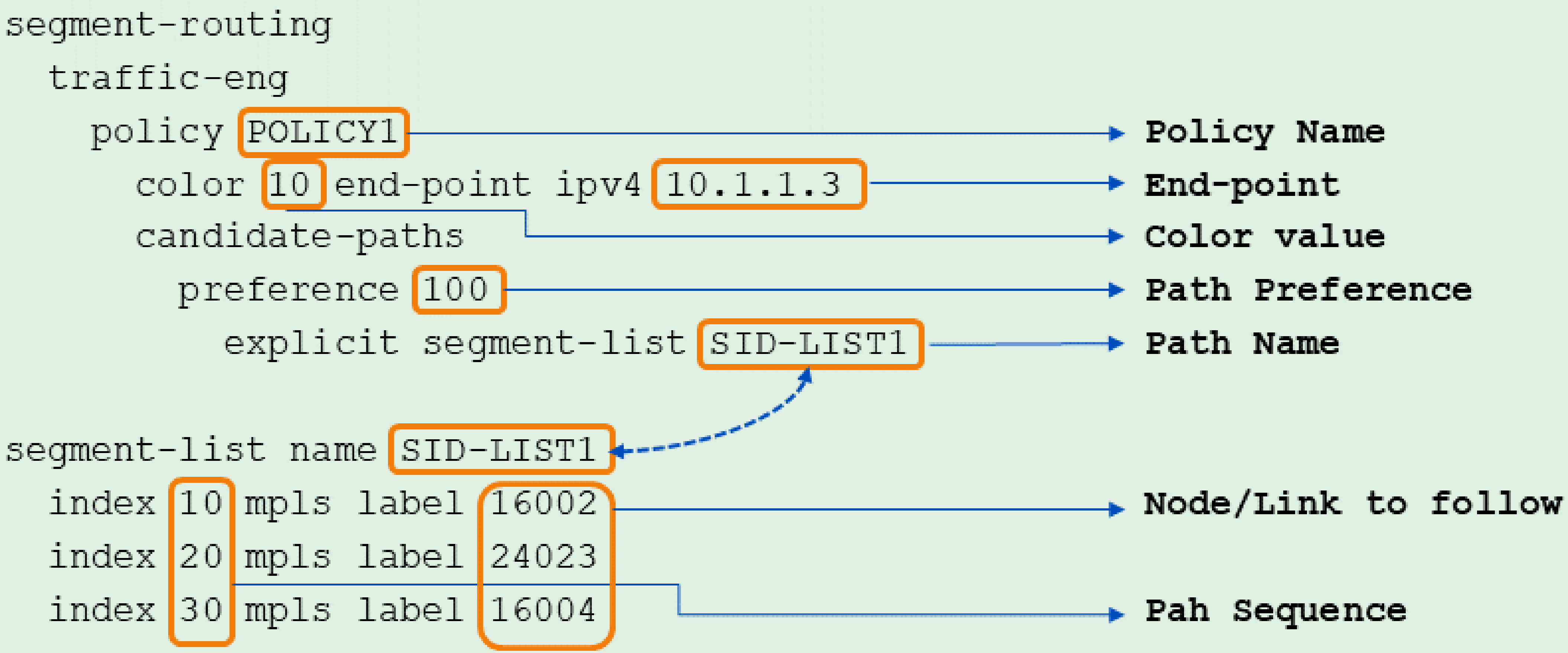


SR-MPLS Traffic Engineering

SR Policy

SR Policy

SR Policy: Configuration example



```
segment-routing
  traffic-eng
    policy POLICY1
      color 10 end-point ipv4 10.1.1.3
      candidate-paths
        preference 100
          explicit segment-list SID-LIST1

segment-list name SID-LIST1
  index 10 mpls label 16002
  index 20 mpls label 24023
  index 30 mpls label 16004
```

Policy Name

End-point

Color value

Path Preference

Path Name

Node/Link to follow

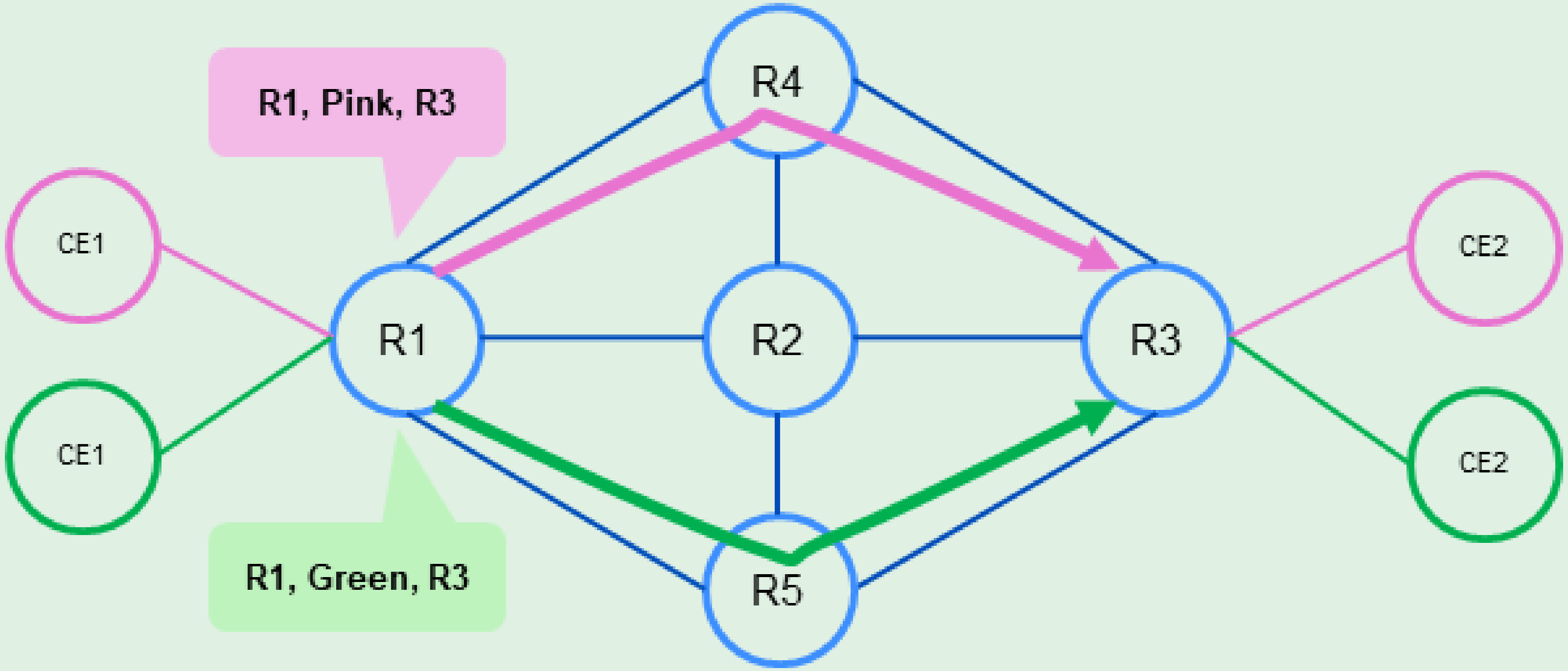
Pah Sequence

SR Policy

- A **SR Policy** is consisting of “headend, color, endpoint”

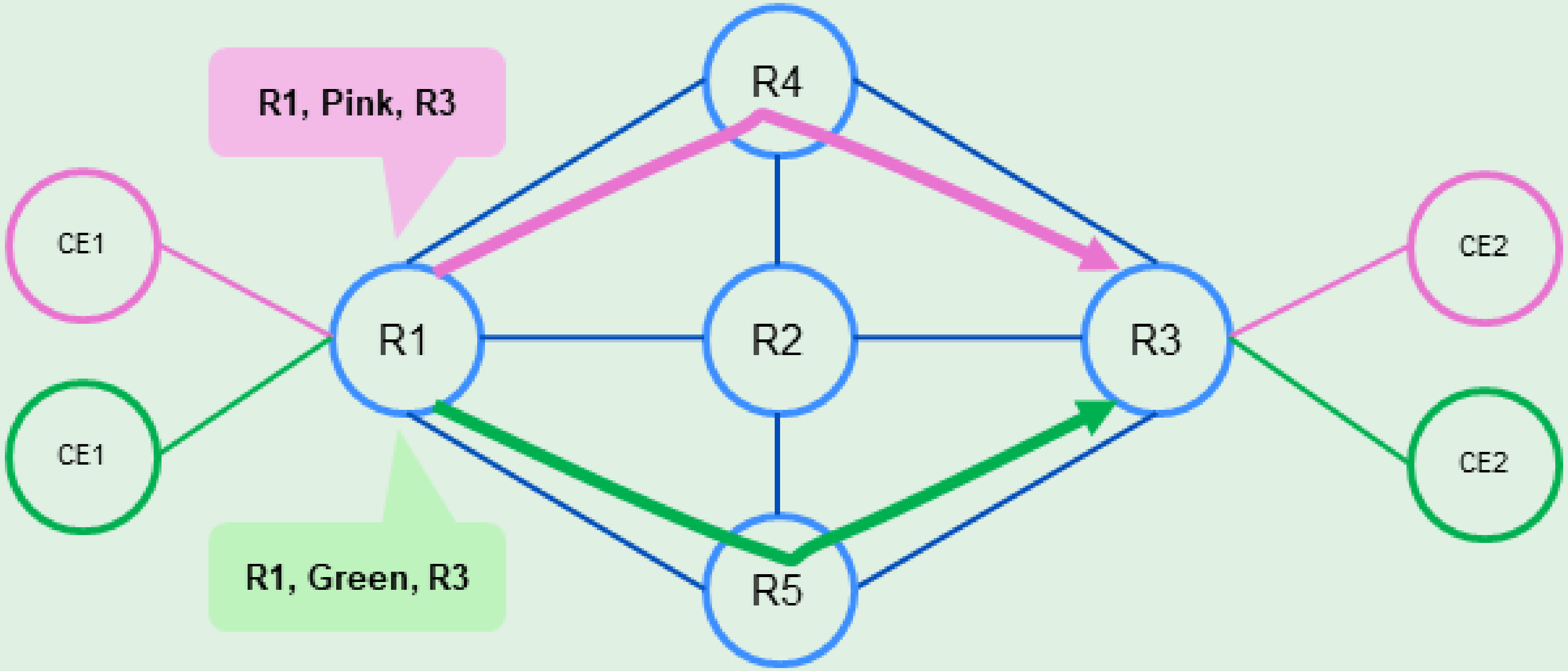
```
policy POLICY1
  color 10 end-point ipv4 10.1.1.3
```

```
policy POLICY2
  color 20 end-point ipv4 10.1.1.3
```



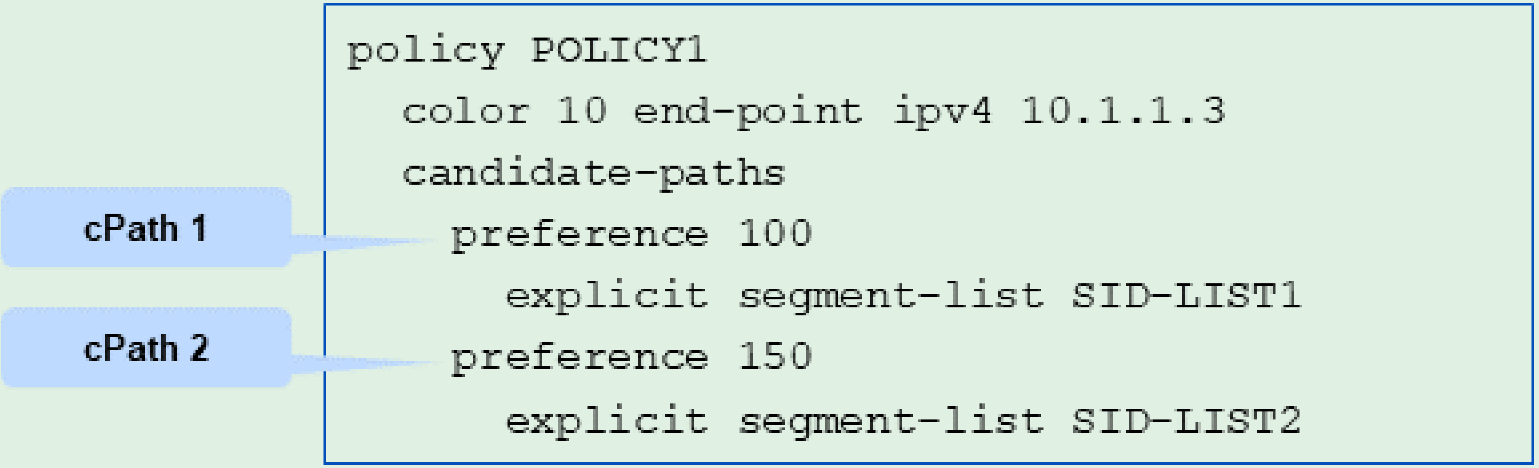
SR Policy

- **Color** is an intent or business requirements
  - Shortest Path, Minimum Delay, To avoid any link/node, Automated traffic steering etc
- **Color** is a numeric value and **must be unique** per H, E pair



SR Policy

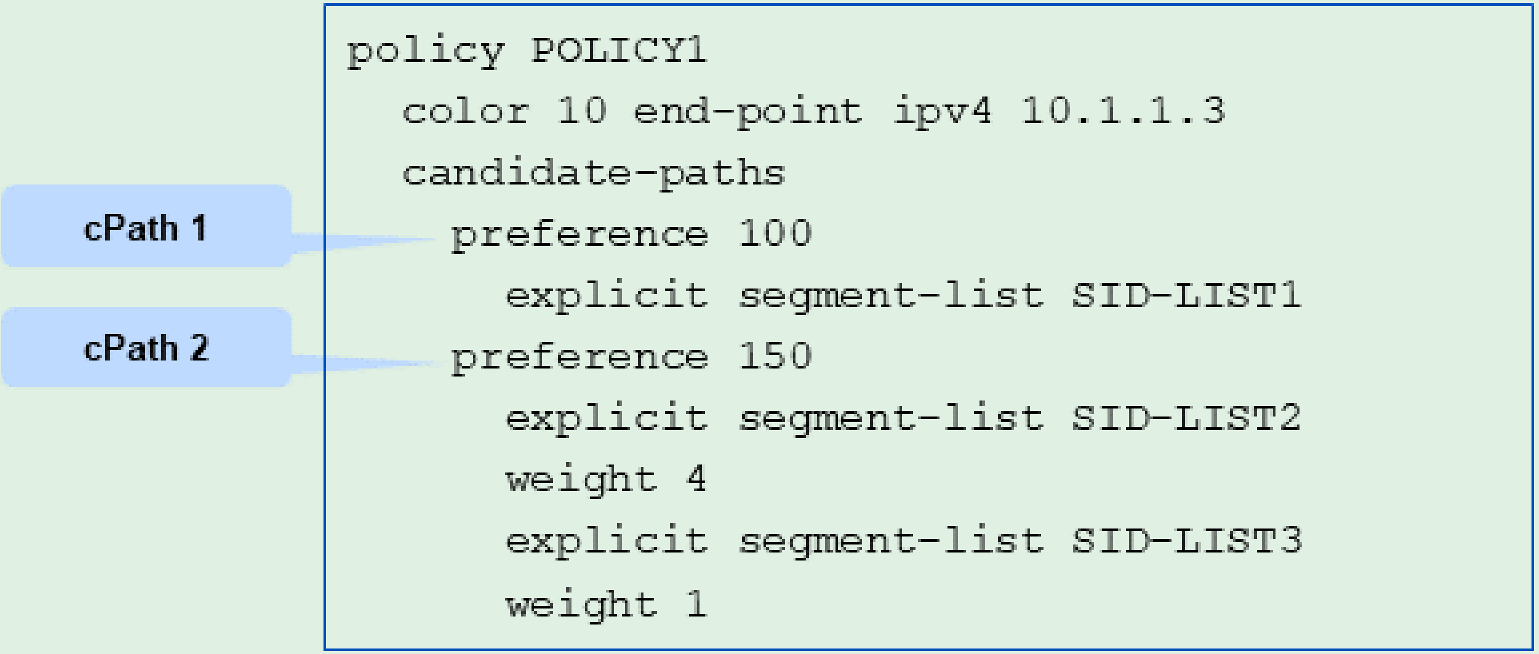
- Candidate Path
  - SR policy contains one or more **Candidate Path (cPath)**
  - cPath has a preference value
    - Higher value is preferred; Not based on any protocol priority
  - cPath can be configure by CLI or PCE or dynamically



```
policy POLICY1
  color 10 end-point ipv4 10.1.1.3
  candidate-paths
    preference 100
      explicit segment-list SID-LIST1
    preference 150
      explicit segment-list SID-LIST2
```

SR Policy

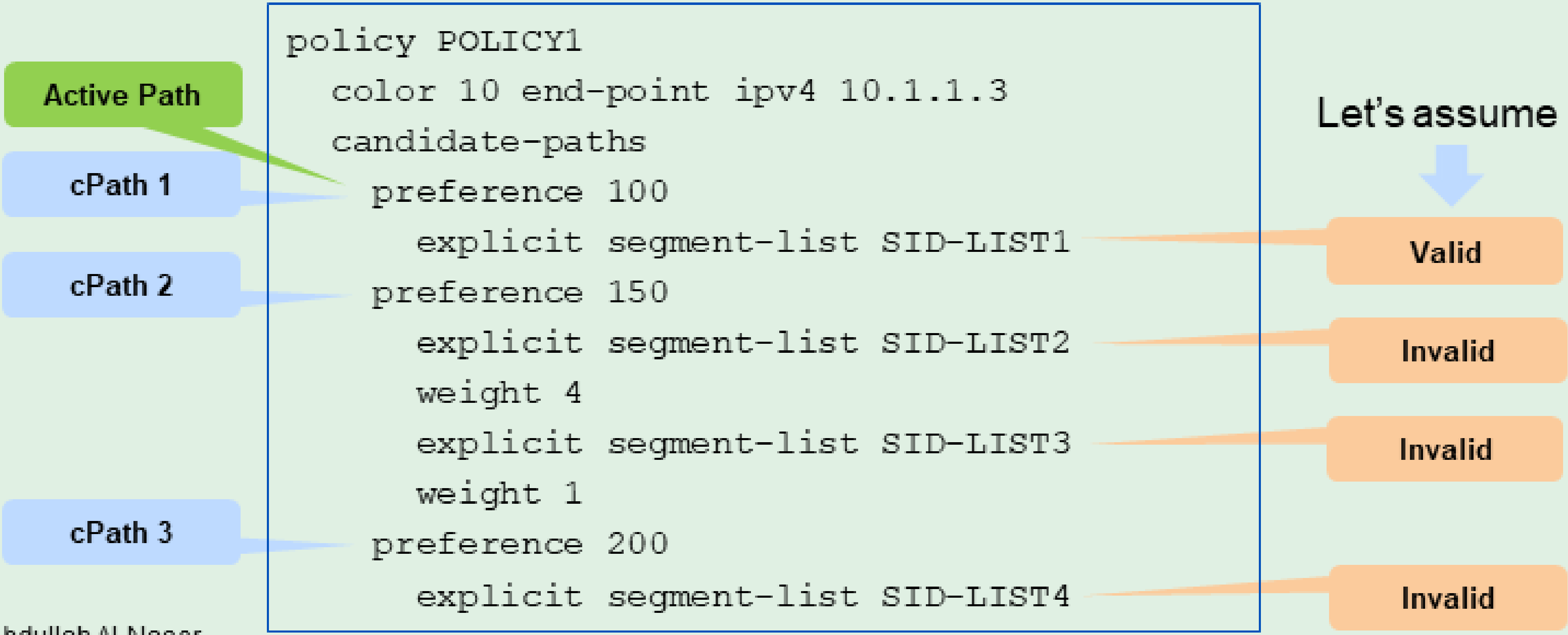
- Candidate Path
  - Multiple SID-LIST can be configured using **weight** value
  - Weight is used to load share the traffic



```
policy POLICY1
  color 10 end-point ipv4 10.1.1.3
  candidate-paths
    preference 100
      explicit segment-list SID-LIST1
    preference 150
      explicit segment-list SID-LIST2
      weight 4
      explicit segment-list SID-LIST3
      weight 1
```

SR Policy

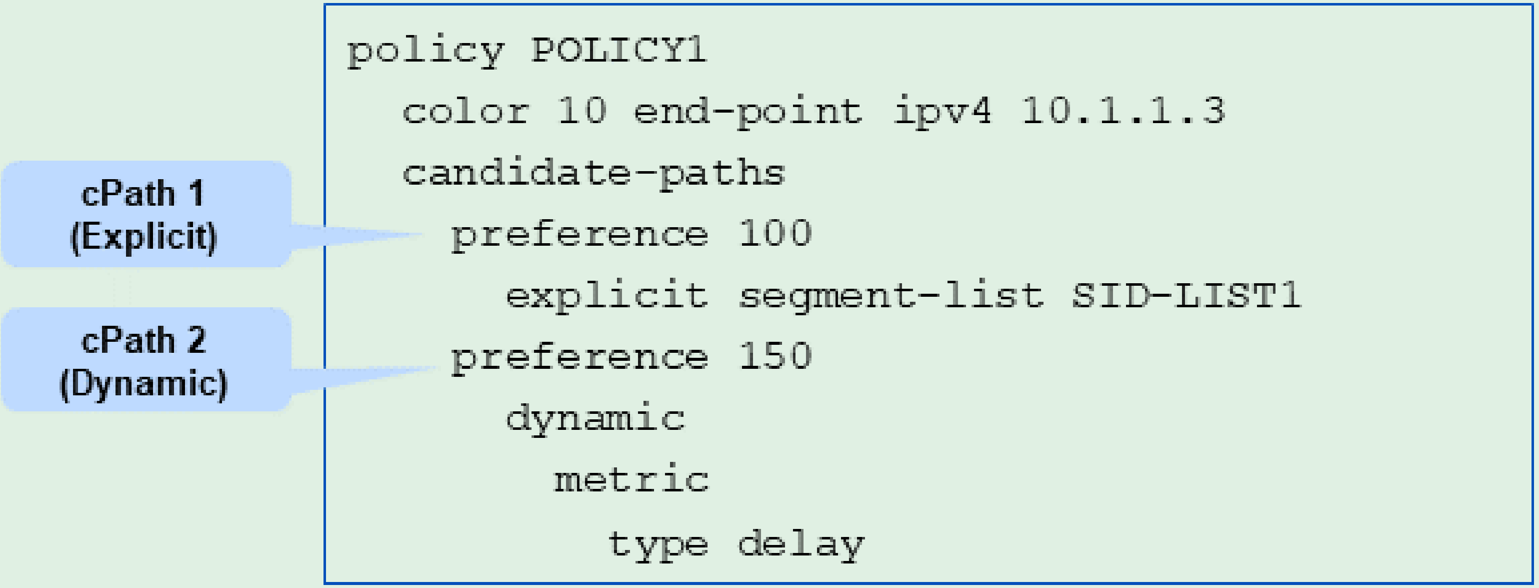
- Candidate Path
  - Multiple cPath can be source of an **active path**; and programmed in forwarding table
  - A path **must be valid** to be an active path
    - Tie-breaker also available in case of multiple active path



```
policy POLICY1
  color 10 end-point ipv4 10.1.1.3
  candidate-paths
    preference 100
      explicit segment-list SID-LIST1
    preference 150
      explicit segment-list SID-LIST2
      weight 4
      explicit segment-list SID-LIST3
      weight 1
    preference 200
      explicit segment-list SID-LIST4
```

SR Policy

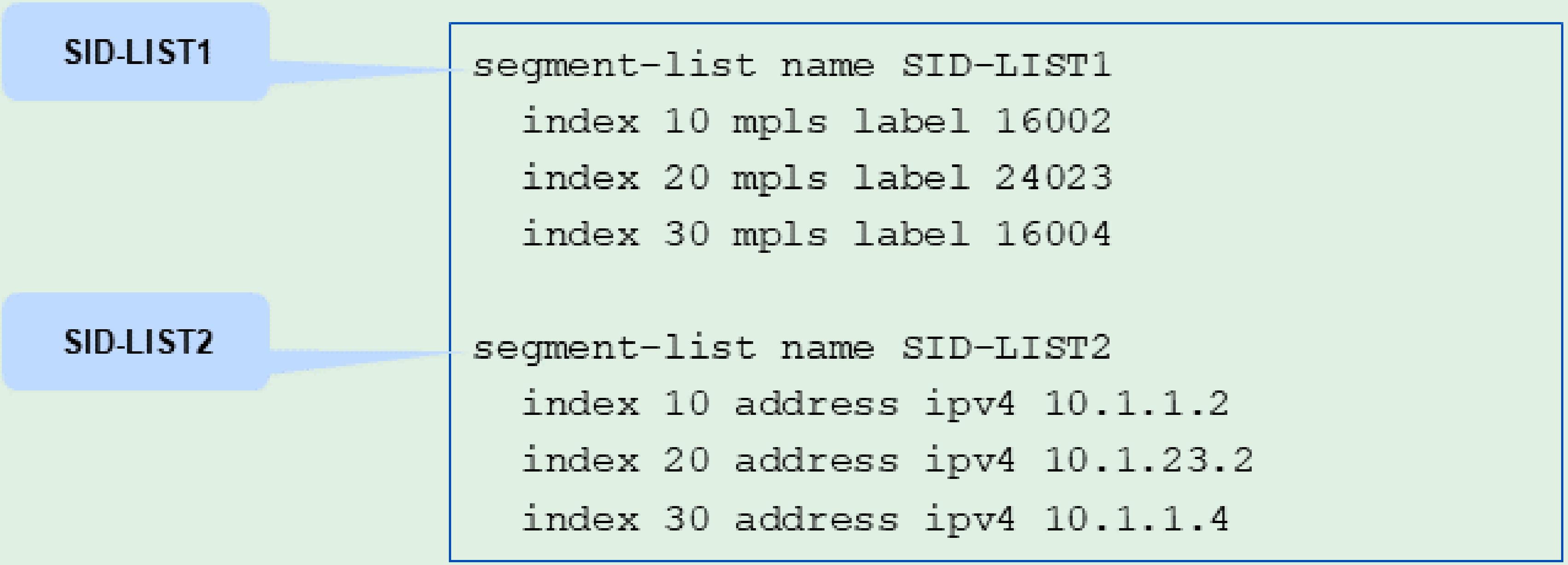
- Explicit cPath
  - Configured by CLI or PCE
- Dynamic cPath
  - Configured dynamically based on link attributes (like low delay)
  - SR Native Algorithm to translate intent to SID-LIST



```
policy POLICY1
  color 10 end-point ipv4 10.1.1.3
  candidate-paths
    preference 100
      explicit segment-list SID-LIST1
    preference 150
      dynamic
        metric
          type delay
```

SR Policy

- Explicit cPath can be configured:
  - By defining SID (MPLS Label)
  - By defining Segment Descriptor (Prefix or Adjacency)
- Headend needs to validate SID-LIST before using it



```
segment-list name SID-LIST1
  index 10 mpls label 16002
  index 20 mpls label 24023
  index 30 mpls label 16004

segment-list name SID-LIST2
  index 10 address ipv4 10.1.1.2
  index 20 address ipv4 10.1.23.2
  index 30 address ipv4 10.1.1.4
```

SR Policy

- Dynamic cPath can be configured:
  - By headend or by PCE
    - By defining SID (MPLS Label), or by defining Segment Descriptor (Prefix or Adjacency)
  - Path re-computed automatically
- May have optimization objectives
  - Minimum metric or delay
- May have set of constraints
  - Maximum cumulative metric, delay or other constraints

SR Policy

- Dynamic cPath
  - *Database* contains all information
    - Headend's SR-TE DB may contain info of the local domain only
    - PCE's SR-TE DB contains info of all domains
  - *Computation Engine* to compute optimized path
    - Translate intents into SID-List
    - Distributed computation
    - Centralized computation
  - Headend and PCE use same algorithm
    - SR Native Algorithm

SR Policy

- Dynamic cPath Distributed Computation
  - Done by headend router
  - Headend may contain low delay path
    - Flooded by IGP
    - Stored in SR-TE-DB
- Dynamic cPath Centralized Computation
  - Done by PCE
  - PCE collects info from PCC (headend) using BGP-LS
  - PCC requests for path info (based on optimization objectives and constraints)
    - Path Request, Reply and Report
  - PCE sends path to PCC in the form of a SID-List
  - PCC can delegate the path option to PCE

SR Policy

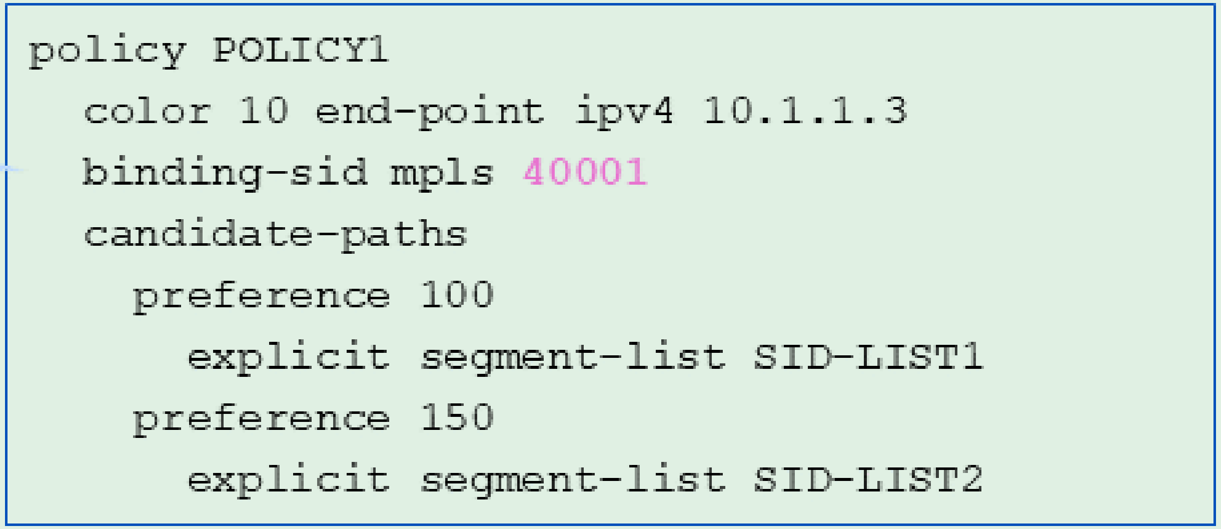
Some use cases:

- SID-List
  - Controller based multi domain network
    - Doesn’t have full visibility at headend
  - No need multiple resolve
- Segment Descriptor
  - In case of full visibility
  - Operator may need headend monitoring
  - Dynamic labels (Adj-SID) are difficult to guess
    - Exception: Adj-SID, Peering SID can also be configured manually

SR Policy

- Binding SID (BSID)
  - BSID is the **identifier** of a SR Policy
  - Configured automatically; or can be configured manually
  - Can be used as **In Label** in multi-domain network

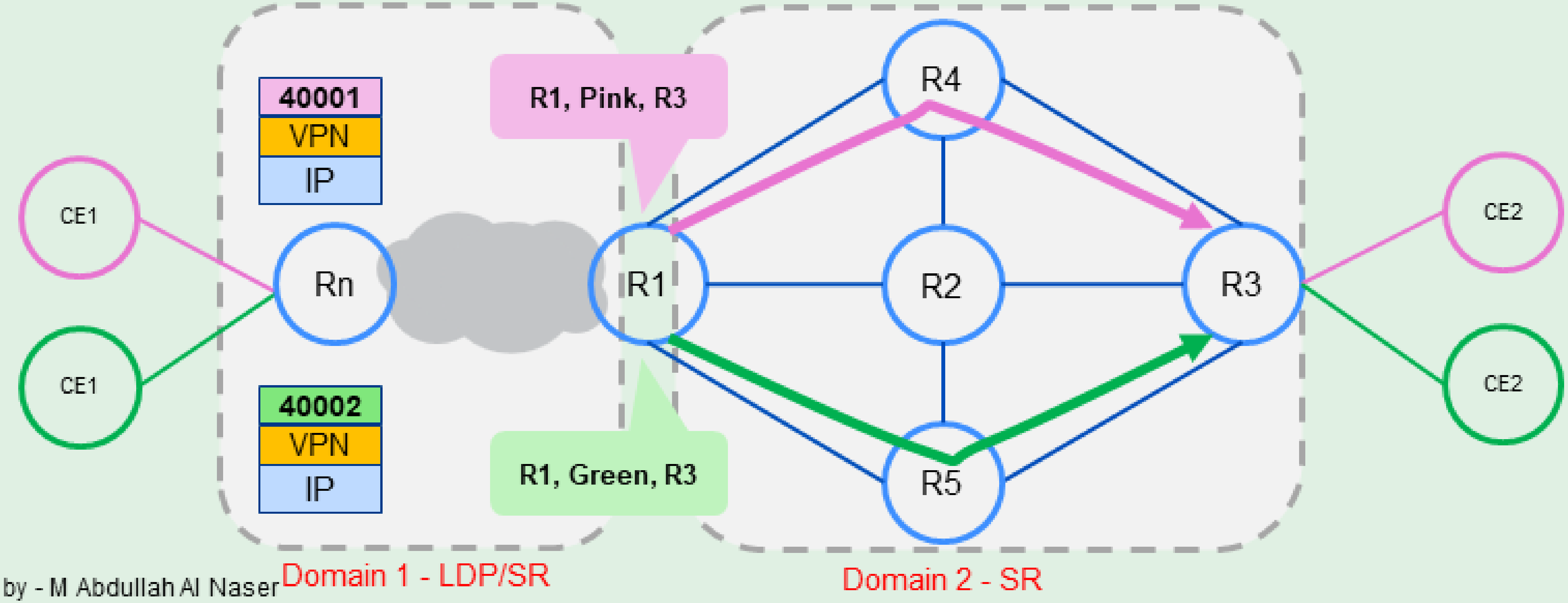
BSID



```
policy POLICY1
  color 10 end-point ipv4 10.1.1.3
  binding-sid mpls 40001
  candidate-paths
    preference 100
      explicit segment-list SID-LIST1
    preference 150
      explicit segment-list SID-LIST2
```

SR Policy

- Binding SID (BSID)
  - BSID is the **identifier** of a SR Policy
  - Configured automatically; or can be configured manually
  - Can be used as **In Label** in multi-domain network

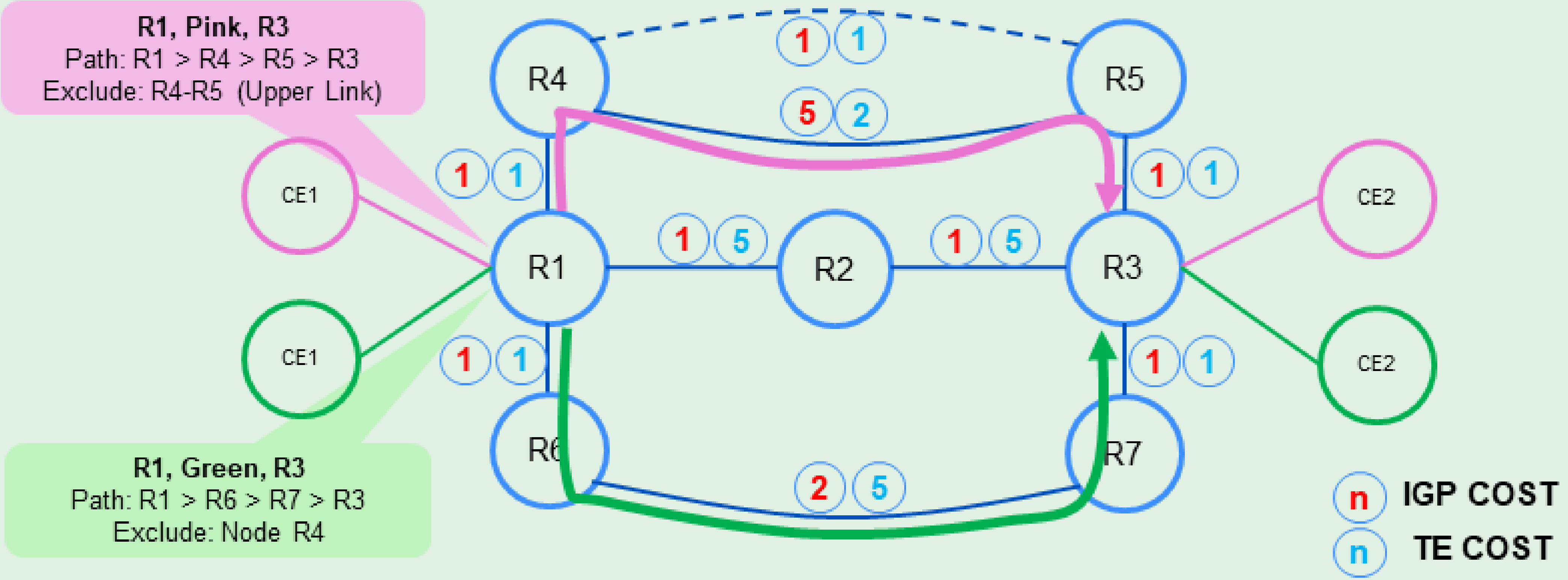


Constraints

- Constraints based TE
- Constraints can be:
  - Include and/or exclude TE Affinity
  - Include and/or exclude IP Address
  - Include and/or exclude SRLG
    - Shared Link Resource Group
  - Maximum accumulated metric
    - IGP, TE, Delay
  - Maximum number of SID in the SID-LIST
  - Disjoint path from another SR Policy

Constraints

- Link / Node Exclusion



R1, Pink, R3
Path: R1 > R4 > R5 > R3
Exclude: R4-R5 (Upper Link)

R1, Green, R3
Path: R1 > R6 > R7 > R3
Exclude: Node R4

Constraints

- Link Exclusion

```
#All Router#
router ospf 1
  distribute link-state

#R1, R4#
segment-routing
  traffic-eng
    affinity-map
      color PINK bit-position 0

#R4#
interface gi0/0/0/1
  affinity color PINK
```

```
#R1#
policy POLICY1
  color 10 end-point ipv4 10.1.1.3
  candidate-paths
    preference 100
      dynamic
        metric type te
      constraints
        affinity
          exclude-any color PINK
```

R1, Pink, R3
Path: R1 > R4 > R5 > R3
Exclude: R4-R5 (Upper Link)

Constraints

- Node Exclusion

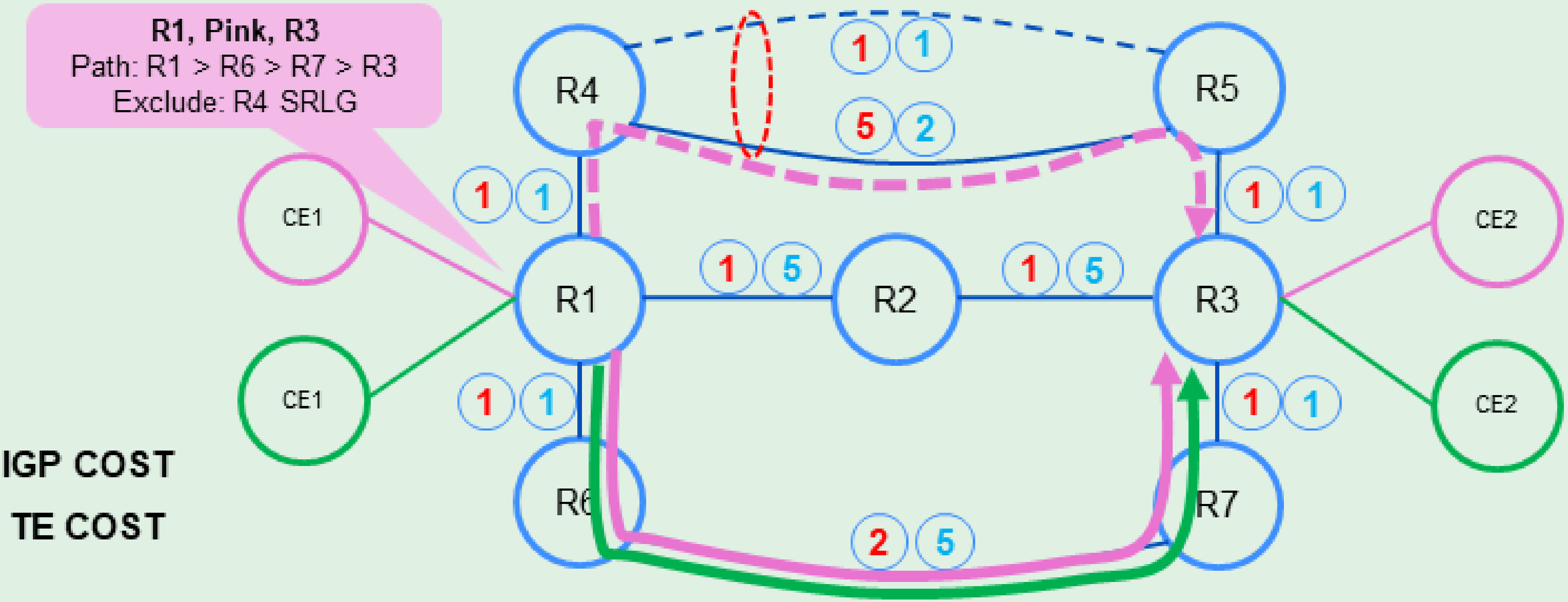
```
#R1#
prefix-set PSET1
  10.1.1.4
  end-set
```

```
#R1#
policy POLICY2
  color 20 end-point ipv4 10.1.1.3
  candidate-paths
    preference 100
      dynamic
        metric type te
      constraints
        address
          exclude PSET1
```

R1, Green, R3
Path: R1 > R6 > R7 > R3
Exclude: Node R4

Constraints

- SRLG Exclusion
  - SRLG means multiple links share same resource
  - i.e. Fiber path, conduit, patch panel etc



R1, Pink, R3
Path: R1 > R6 > R7 > R3
Exclude: R4 SRLG

n IGP COST

n TE COST

Constraints

- SRLG Exclusion
  - Shared Link Resource Group

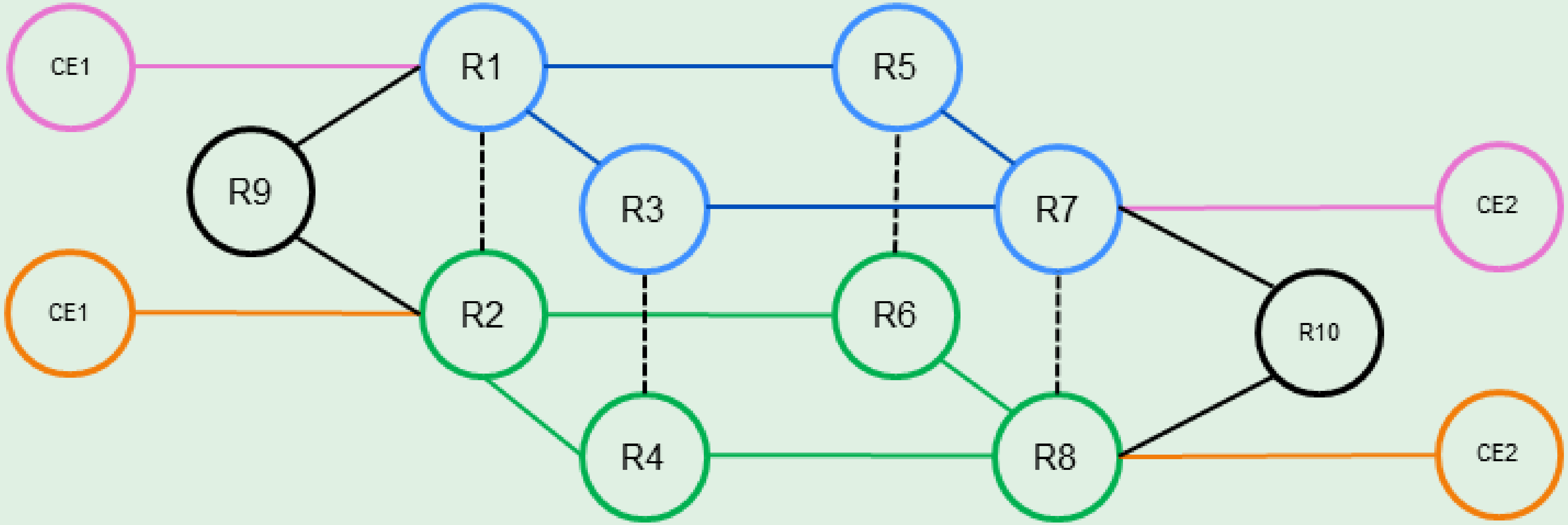
```
#R4#
srlg
  interface gi0/0/0/1
    10 value 1111
  interface gi0/0/0/1
    10 value 1111
```

```
#R1#
policy POLICY1
  color 10 end-point ipv4 10.1.1.3
  candidate-paths
    preference 100
      dynamic
        metric type te
      constraints
        srlg
          exclude 1111
```

R1, Pink, R3
Path: R1 > R6 > R7 > R3
Exclude: R4 SRLG

Constraints

- Disjoint Path
  - SR-TE can compute a path which is disjoint from another path in same disjoint-group
  - Link, Node, SRLG
  - Plain A, Plain B etc
    - Avoid congestion, maintaining SLA, Load balancing etc.



Path Validation

Explicit Path

- SID-List must be validated:
  - Should have at least one SID
  - Weight should be larger than 0
    - default is 1
  - Headend to resolve IPs in exp-path in to MPLS labels
  - Headend to resolve first SID to find out reachable next-hop
  - Any constraints can be considered (optional)
- If any intermediary link fails:
  - Adj-SID is withdrawn
  - Corresponding node will perform TI-LFA
    - For around 15 minutes at maximum

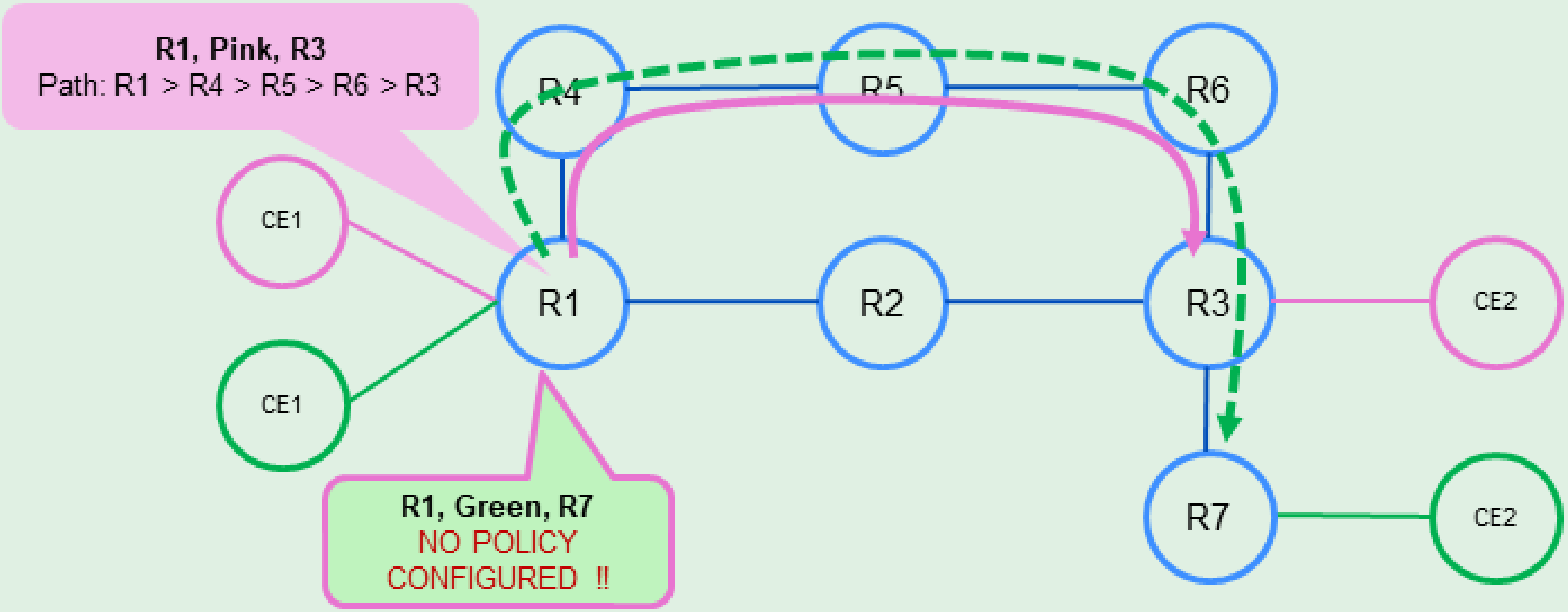
Path Validation

Dynamic Path

- SID-List must be validated:
  - Should have at least one SID
  - Weight should be larger than 0
    - default is 1
  - Headend to resolve IPs in exp-path in to MPLS labels
  - Headend to resolve first SID to find out reachable next-hop
  - Any constraints can be considered (optional)
- If any intermediary link fails:
  - Adj-SID is withdrawn
  - Corresponding node will perform TI-LFA
    - For around 15 minutes at maximum

Traffic Steering

- Auto Route
  - IGP steers all the traffic towards destination and other downstream nodes
  - Localized behaviour on headend



R1, Pink, R3
Path: R1 > R4 > R5 > R6 > R3

R1, Green, R7
NO POLICY
CONFIGURED !!

Traffic Steering

- Auto Route
  - Limited to local IGP area
  - Limited to per-BGP next-hop
  - Streets all service traffic destined for BGP next-hop
- Policy Based Routing
  - Can steer traffic for any specific VPN, or PW
- Static Route
  - Requires static route to specified next-hop for individual services
- Anything better than these?
  - Automated Steering

SR-MPLS Traffic Engineering

Lab - Demo of SR Policy

Thank you!