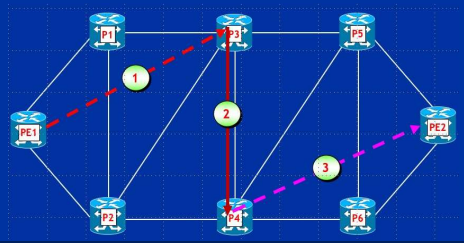
School of Advanced Networking by NAMEX

Reiss Romoli

Segment Routing: from theory to practice



Tiziano Tofoni

Copyright Tiziano Tofoni

Tiziano Tofoni
22/04/2026 16:35:26 UTC+0200

Copyright notes

- This set of slides, including the footnotes, is protected by copyright laws and international treaties. Title and copyright to the slides (including, but not limited to, all images, photographs, animations, videos, audio, music, and text), in accordance with Articles 12 et seq. of Italian Law 633/1941, are owned by the author, Tiziano Tofoni (hereinafter "the author").
- The slides must be used exclusively for study purposes in the courses taught by the author.
- Any other use or reproduction (including, but not limited to, reproductions on optical/magnetic media, computer networks, or printed media) in whole or in part is prohibited unless explicitly authorized in writing, in advance, by the author.
- The information contained in these slides is believed to be accurate as of the date of publication. It is provided for educational purposes only and not for use in designing systems, products, networks, etc. In any case, it is subject to change without notice. The author assumes no responsibility for the content of these slides (including, but not limited to, the accuracy, completeness, applicability, or timeliness of the information).
- In any case, this copyright notice must never be removed and must be retained even in partial uses.

Disclaimer

REISS ROMOLI

- **Vendor Examples**: References of vendors are used for illustration only. Many vendors offer similar solutions in the networking industry
- **No Affiliation**: I do not represent any vendor. All opinions are my own.
- **Information validity**: Content may become outdated due to rapid technology changes. Do not use it as your sole source of information.
- **Use at Your Own Risk**: Consult official documentation and professionals before implementing any concepts discussed. No liability is assumed for consequences of use.

AGENDA

Module 1: Introduction

Module 2: *Segment Routing* over MPLS data plane (SR-MPLS)

Module 3: *Segment Routing Traffic Engineering* (SR-TE)

Module 4: *Segment Routing* over IPv6 data plane (SRv6)

Copyright Tiziano Tofoni


Module 1: Introduction

#1 Why Segment Routing?

#2 Fundamental concepts

Copyright Tiziano Tofoni

Why deploying *Segment Routing*? (1/2)

REISS ROMOLI

- Control plane simplification
  - Elimination of additional protocols
  - Elimination of state in intermediate routers
- Scalability and flexibility
  - By eliminating state in intermediate routers, Segment Routing improves network scalability allowing for a greater number of paths and services
  - Simplified and more granular traffic engineering
- Multiple data planes
  - MPLS (SR-MPLS)
  - IPv6 (SRv6)

Why deploying *Segment Routing*? (2/2)

REISS ROMOLI

- Integration with SDN controllers
  - Automation and centralized management
  - Rapid response to changes
- Traffic protection
  - Path protection (Fast Reroute - FRR)
  - Support for node and/or link out-of-service protection mechanisms (e.g., TI-LFA, Topology Independent-Loop Free Alternate)
  - Definition of specific paths to balance load and prevent congestion on specific links
- Reduction of CAPEX and OPEX
  - Resource optimization: The ability to manage traffic more efficiently can lead to better utilization of existing network resources, delaying the need for hardware upgrades
  - Operational simplification: Fewer protocols and fewer states to manage mean reduced operational complexity

Copyright Tiziano Tofoni


Module 1: Introduction

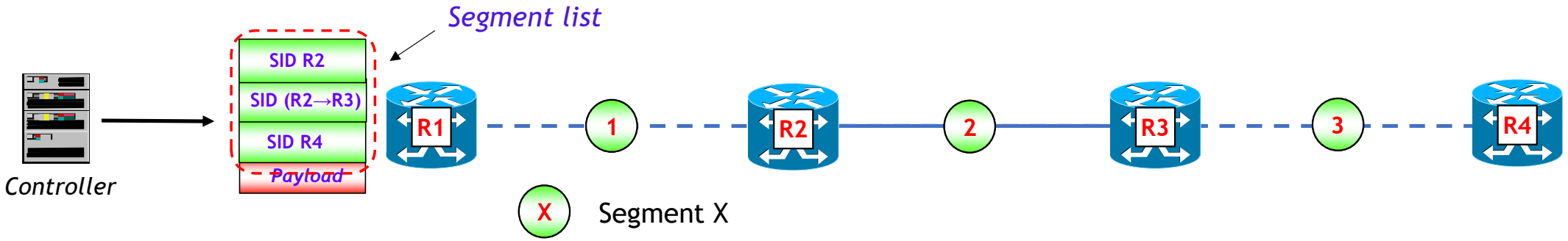
#1 Why Segment Routing?

#2 Fundamental concepts

Copyright Tiziano Tofoni

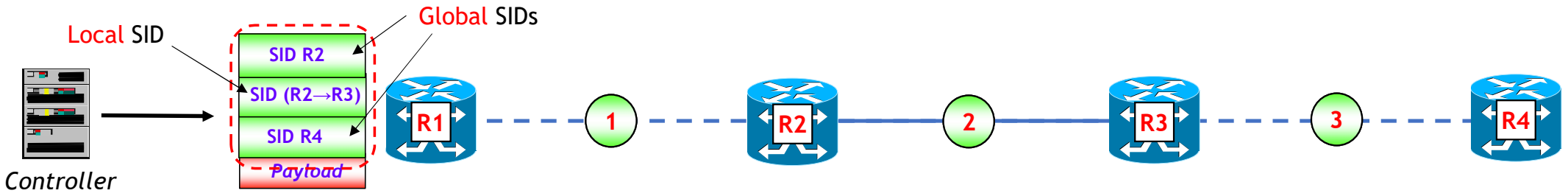
What is *Segment Routing*?

- Segment Routing is a modern variant of (old) source routing where an end-to-end path is divided into many segments
- Each segment is identified by a ***Segment ID*** (**SID**)
  - A SID can be an MPLS label (SR-MPLS) or an IPv6 address (SRv6)
  - A SID constitutes a routing "instruction" for the nodes traversed by the packet.
- Basic idea: Add a stack of SIDs (Segment Lists) to packets entering a routing domain, allowing them to be routed along a desired path.



Segment IDs (SIDs)

- Each SID identifies a network segment
  - A destination prefix
  - A simple network node
  - A link (or rather, IGP adjacency, which must be IS-IS or OSPF)
  - etc.
- Two types of SIDs
  - Local: The router that generates and advertises a local SID is the only one capable of using it
  - Global: Each router in the routing domain assigns (locally) a (global) SID, and all other routers can use it

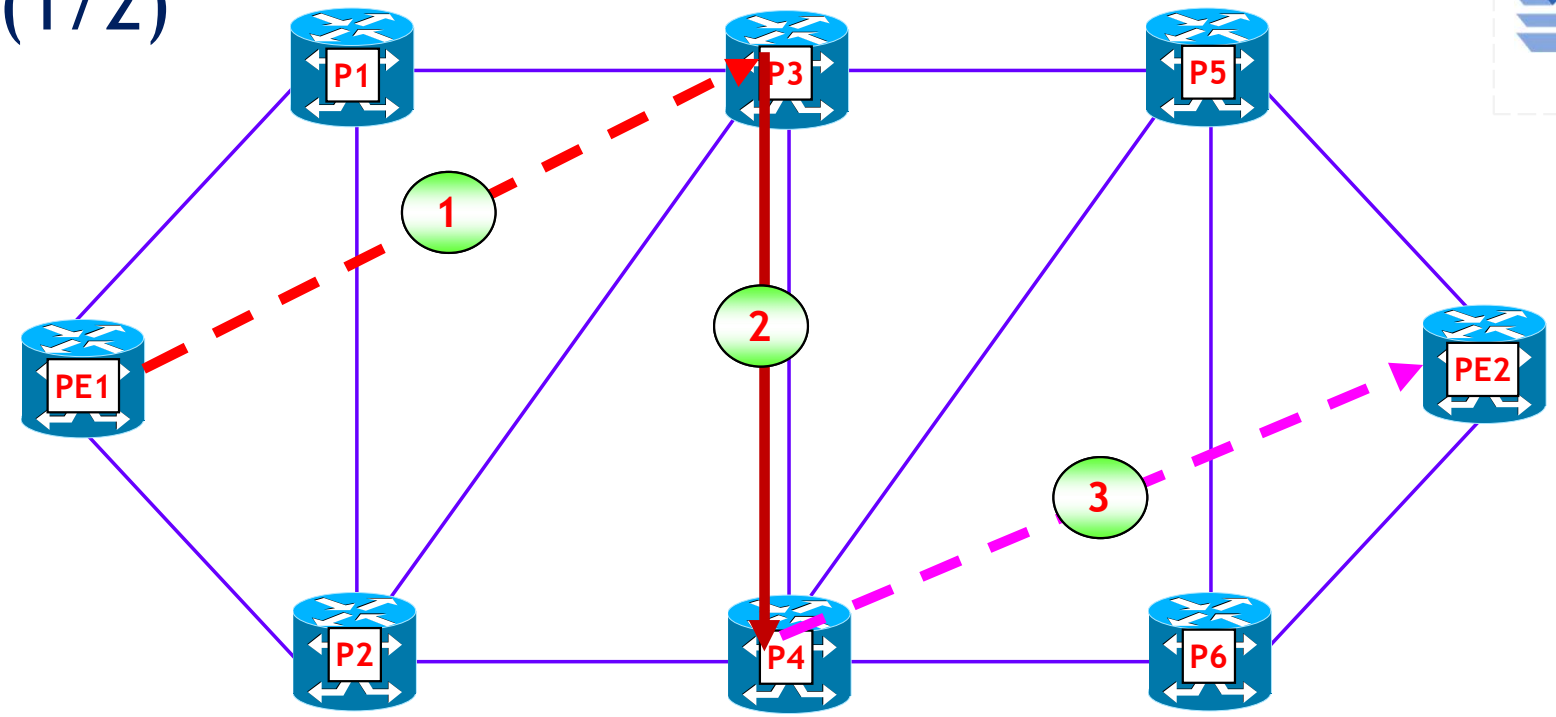


Forwarding plane

REISS ROMOLI

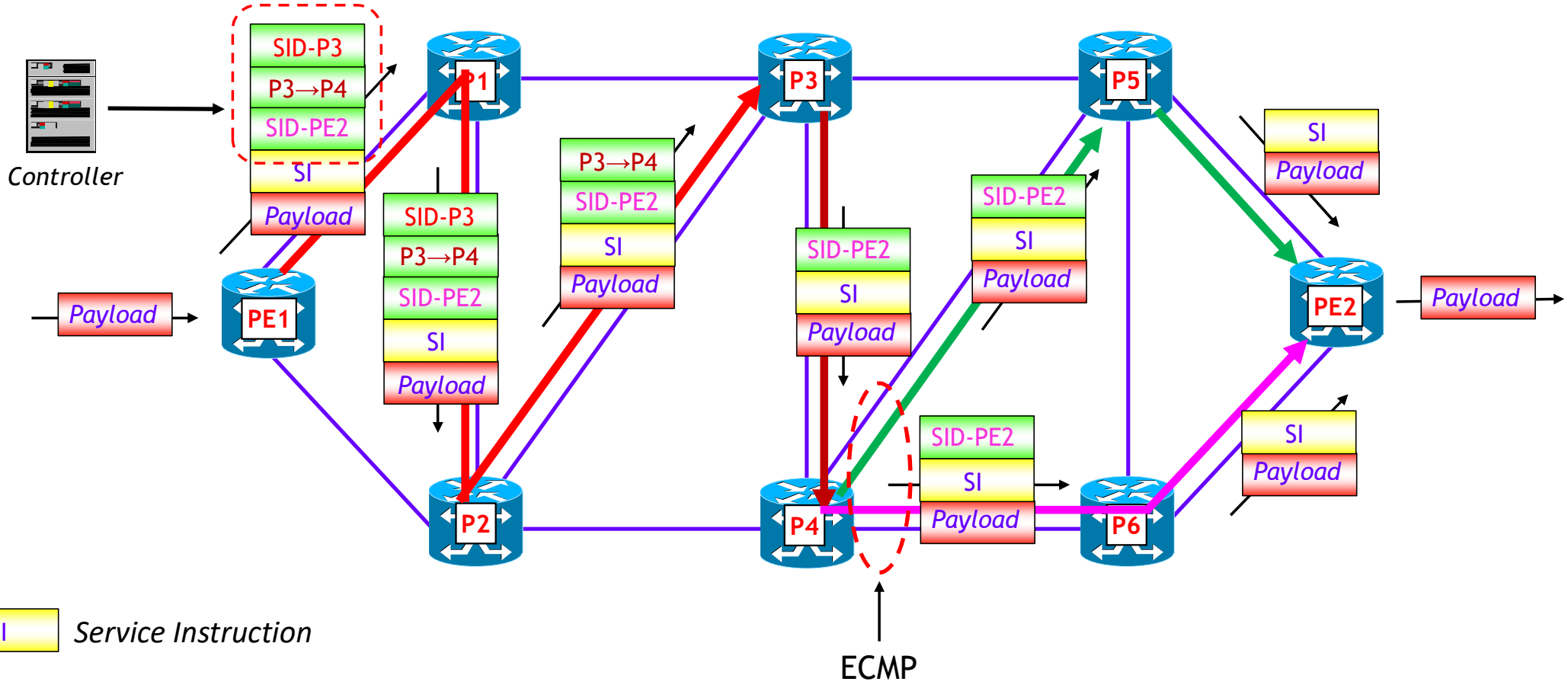
- **SR-MPLS**: SID=MPLS label
  - SR-MPLS reuses the MPLS data plane without any changes
  - An end-to-end path is represented by a stack of MPLS labels
- **SRv6**: SID=IPv6 address
  - IPv6 SRv6 uses an IPv6 forwarding plan
  - Need for a network with an IPv6 numbering plan
  - An end-to-end path can be represented by:
    - A set of IPv6 addresses collected in a Routing Extension Header (SRH type 4)
    - Or, a single IPv6 address with the SIDs encoded within the single address
- NOTE: Both forwarding planes are applicable to the transport of IPv4 and IPv6 traffic

Example (1/2)

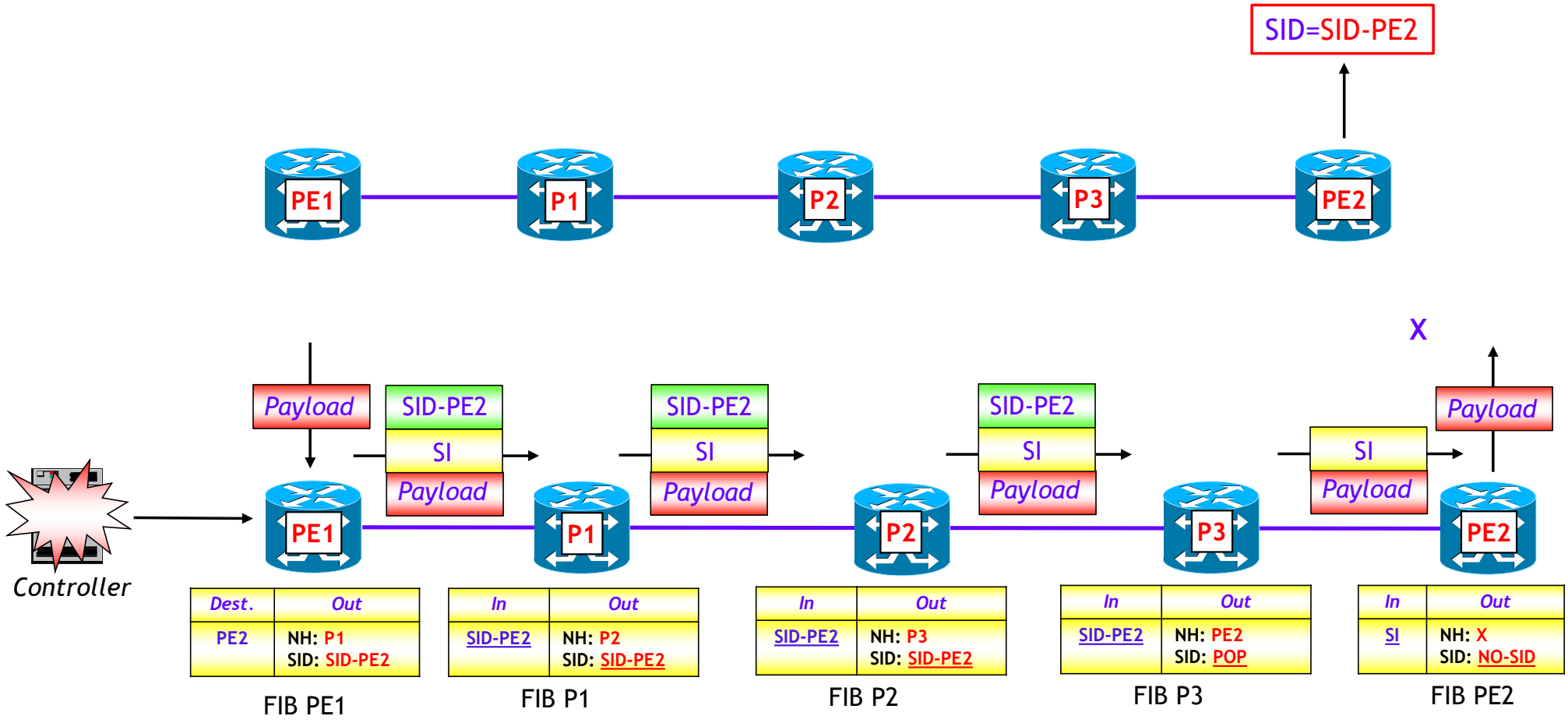


- PE1→PE2 path is made up of three segments
  - PE1→P3: global segment (SID=SID-P3)
  - P3→P4: local segment (SID=P3→P4)
  - P4→PE2: global segment (SID=SID-PE2)

Example (2/2)


Controller
SID-P3
P3→P4
SID-PE2
SI
Payload
P1
P3
P5
PE1
PE2
P2
P4
P6
Payload
SI
Service Instruction
ECMP

Example: path made up of only one segment



SIDs distribution

- SIDs distribution occurs through extensions to the IS-IS and OSPF protocols.
  - IS-IS extensions: use new TLV modules
  - OSPF extensions: use new opaque LSAs
- SR-MPLS
  - RFC 8667 - *IS-IS Extensions for Segment Routing*, dicembre 2019
  - RFC 8665 - *OSPF Extensions for Segment Routing*, dicembre 2019
- SRv6
  - RFC 9352 - *IS-IS Extensions to Support Segment Routing over the IPv6 Data Plane*, febbraio 2023
  - RFC 9513 - *OSPFv3 Extensions for Segment Routing over IPv6* (SRv6), dicembre 2023

Prefix-SID and Node-SID

REISS ROMOLI

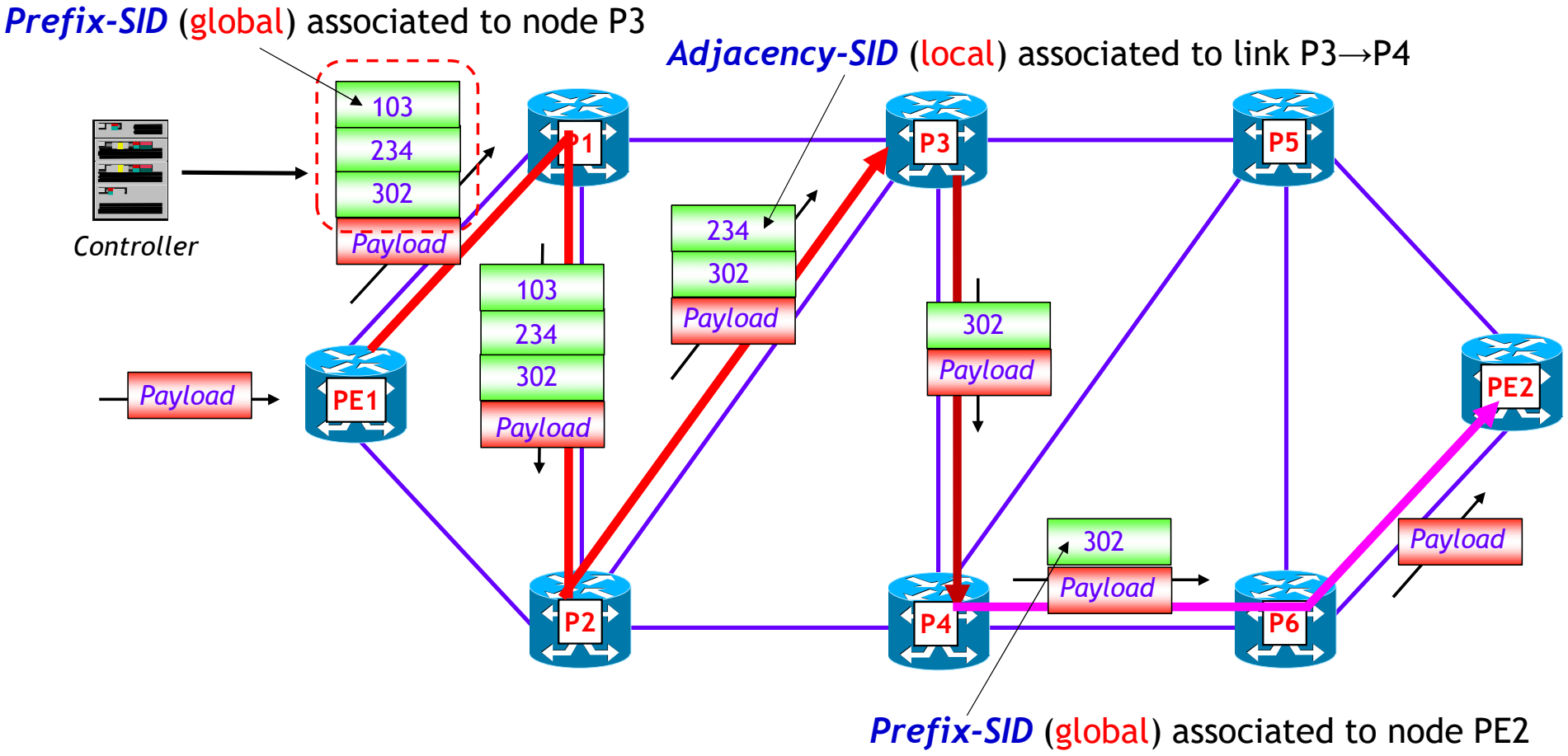
- A Prefix-SID identifies an IPv4/v6 prefix (for example, a subnet or a loopback interface address)
  - It is a global SID, meaning its value is unique and recognized by all nodes within the SR domain
  - It is used to reach a specific destination prefix using the optimal path determined by the algorithm used by the IGP protocol (e.g., Dijkstra's algorithm) or through a specific algorithm (typically subject to constraints)
- A Node-SID is a special type of Prefix-SID that identifies a specific node
  - It is typically associated with the IP address of a loopback interface of that node
  - Like the Prefix-SID, the Node-SID has global significance and is unique within the SR domain
  - Node-SIDs are tipically used to construct more complex traffic engineering paths

Copyright Tiziano Tofoni


Adjacency-SID

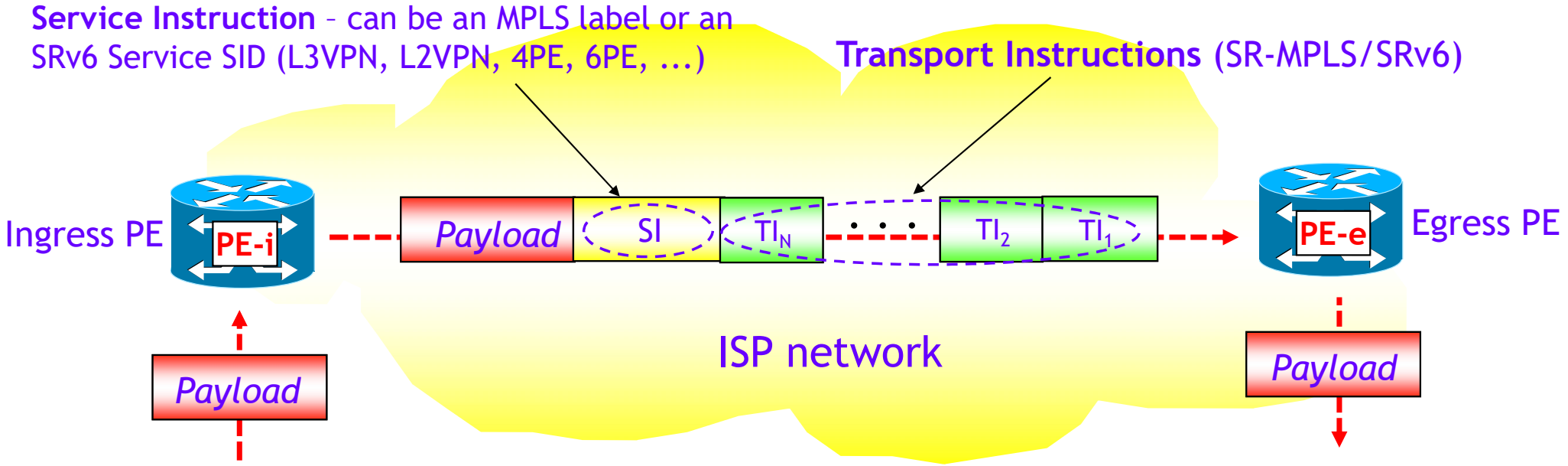
- It’s a specific type of segment that represents a unidirectional adjacency (a link) between two directly connected nodes
- They have local significance, meaning their value is significant only to the node that originated them
- They enforce a strict forwarding, meaning that traffic must traverse that specific link, regardless of the link cost or other potentially shorter paths calculated by the IGP Routing Protocol

To recap...


Prefix-SID (global) associated to node P3
Adjacency-SID (local) associated to link P3→P4
Controller
103
234
302
Payload
P1
P2
P3
P4
P5
P6
PE1
PE2
Payload
Prefix-SID (global) associated to node PE2

The role of Segment Routing in L2/L3VPN services

- **Segment Routing has no impact on L2/L3VPN services**
  - It only impacts how PE-to-PE traffic from various services is transported within the ISP's network



Module 2: *Segment Routing* over MPLS (SR-MPLS)

#1	Main aspectes
#2	SRGB and SRLB
#3	Interworking LDP/Segment Routing
#4	From LDP to Segment Routing: migration plan
#5	Topology Independent LFA (TI-LFA)

Copyright Tiziano Tofoni

SR-MPLS: pros and cons

REISS ROMOLI

- SR-MPLS is a simplification of the MPLS control plane
  - It does not use ad hoc protocols such as LDP or RSVP-TE for MPLS label distribution
  - MPLS labels (i.e., SIDs) are distributed through appropriate extensions to the OSPF and IS-IS protocols
  - IMPORTANT NOTE: SR-MPLS has no impact on L2/L3 MPLS VPN services!
- Pros
  - Reuse of the existing IP/MPLS network and the mature, consolidated MPLS data plane
    - No need to change the current IPv4 numbering plan, no need for IPv6...
  - No need for additional MPLS-specific protocols such as LDP and/or RSVP-TE
    - MPLS labels are announced through IGP protocol extensions (OSPF or IS-IS)
  - No need for additional features such as LDP-IGP synchronization
  - Enables better traffic protection via TI-LFA
- Cons
  - SID planning, which must be defined and configured manually (not necessary with LDP)

Copyright Tiziano Tofoni


Basic operation (1/5)

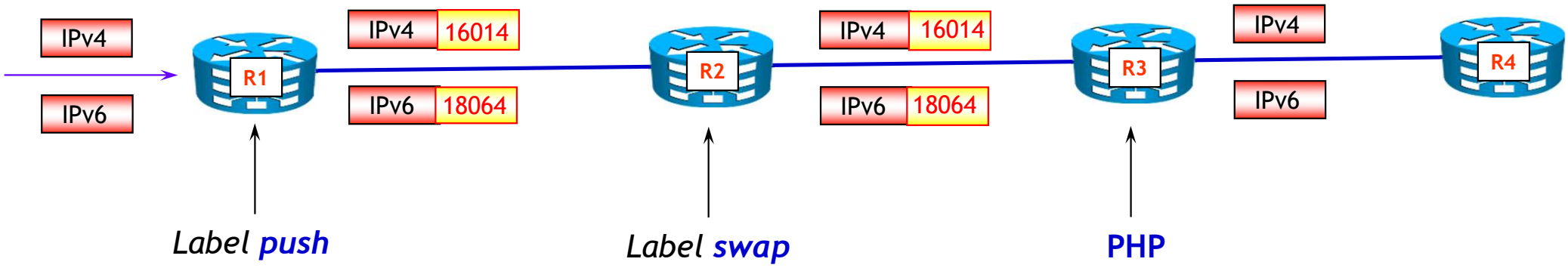
REISS ROMOLI

- SR-MPLS reuses the existing MPLS data plane
  - SID → MPLS label
  - *Segment List* → Label stack
- SR-MPLS reuses the *Penultimate Hop Popping* (PHP) mechanism or even *Ultimate Hop Popping* (UHP)
  - PHP is enabled by default
  - UHP, where required, must be enabled via configuration and uses the reserved Explicit Null label (=0 for IPv4, 2 for IPv6)
- A node imposes a Prefix-SID (SID associated with an IP prefix) if:
  - The destination itself or the next hop is a FEC to which a Prefix-SID has been associated
  - The downstream neighbor (i.e., the neighbor to the destination) is SR-enabled
  - The node is configured to prefer SR labels over labels advertised via LDP or RSVP-TE, or these ad hoc protocols are absent

Copyright Tiziano Tofoni


Basic operation (2/5)

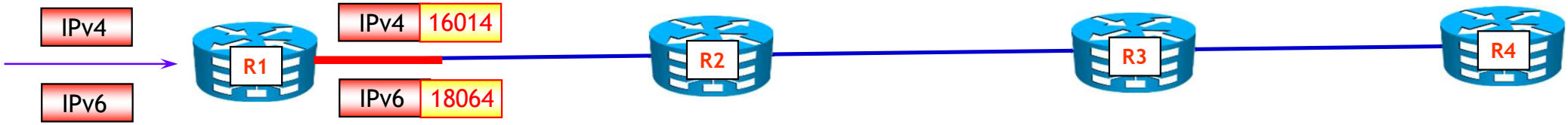
- Rx Loopback0→IPv4: 192.168.1.x/32 ; IPv6 2001:db8:f:1::x/128
- Global label block: 16.000-23.999
- Rx *Prefix-SID* Loopback0: IPv4 16.01x ; IPv6: 18.06x
  - Example: R4 advertises the *host routes* 192.168.1.4/32 and 2001:db8:f:1::4/128 with associated *Prefix-SID* 16.014 and 18.064 respectively



Basic operation (3/5)

```
RP/0/0/CPU0:R1# show cef 192.168.1.4/32
. . .
192.168.1.4/32, version 20, labeled SR, internal 0x1000001 0x81. . .
 local adjacency 172.30.12.2
 Prefix Len 32, traffic index 0, precedence n/a, priority 1
   via 172.30.12.2/32, GigabitEthernet0/0/0/0, 7 dependencies, weight 0,
   class 0                  [flags 0x0]
    path-idx 0 NHID 0x0 [0xa189a110 0x0]
    next hop 172.30.12.2/32
    local adjacency
     local label 16014      labels imposed {16014}
```

Label push




```
RP/0/0/CPU0:R1# show cef ipv6 2001:db8:f:1::4/128
. . .
2001:db8:f:1::4/128, version 17, labeled SR, internal 0x1000001 0x80 . .
. . .
 Prefix Len 128, traffic index 0, precedence n/a, priority 1
   via fe80::5200:ff:fe02:1/128, GigabitEthernet0/0/0/0, 6 dependencies,
   weight 0, class 0 [flags 0x0]
    path-idx 0 NHID 0x0 [0xa19370c8 0x0]
    next hop fe80::5200:ff:fe02:1/128
    local adjacency
     local label 18064      labels imposed {18064}
```

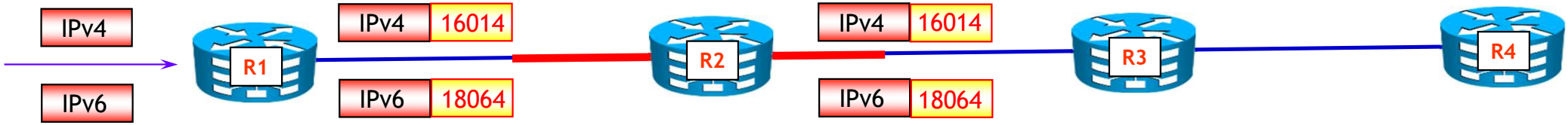
Label push

Basic operation (4/5)

REISS ROMOLI

Label swap

```
RP/0/0/CPU0:R2# show mpls forwarding labels 16014
. . .
Local  Outgoing    Prefix              Outgoing     Next Hop        Bytes
Label  Label       or ID               Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
16014  16014       SR Pfx (idx 14)     Gi0/0/0/1    172.30.23.3     5918
```



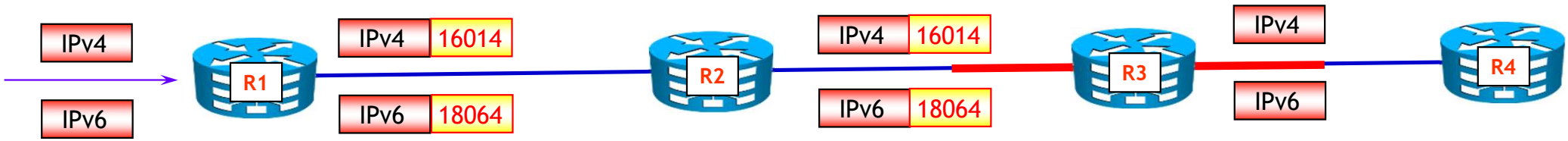
Label swap

```
RP/0/0/CPU0:R2# show mpls forwarding labels 18064
. . .
Local  Outgoing    Prefix              Outgoing     Next Hop        Bytes
Label  Label       or ID               Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
18064  18064       SR Pfx (idx 2064)   Gi0/0/0/1    fe80::5200:ff:fe03:2   \
```

Basic operation (5/5)

PHP

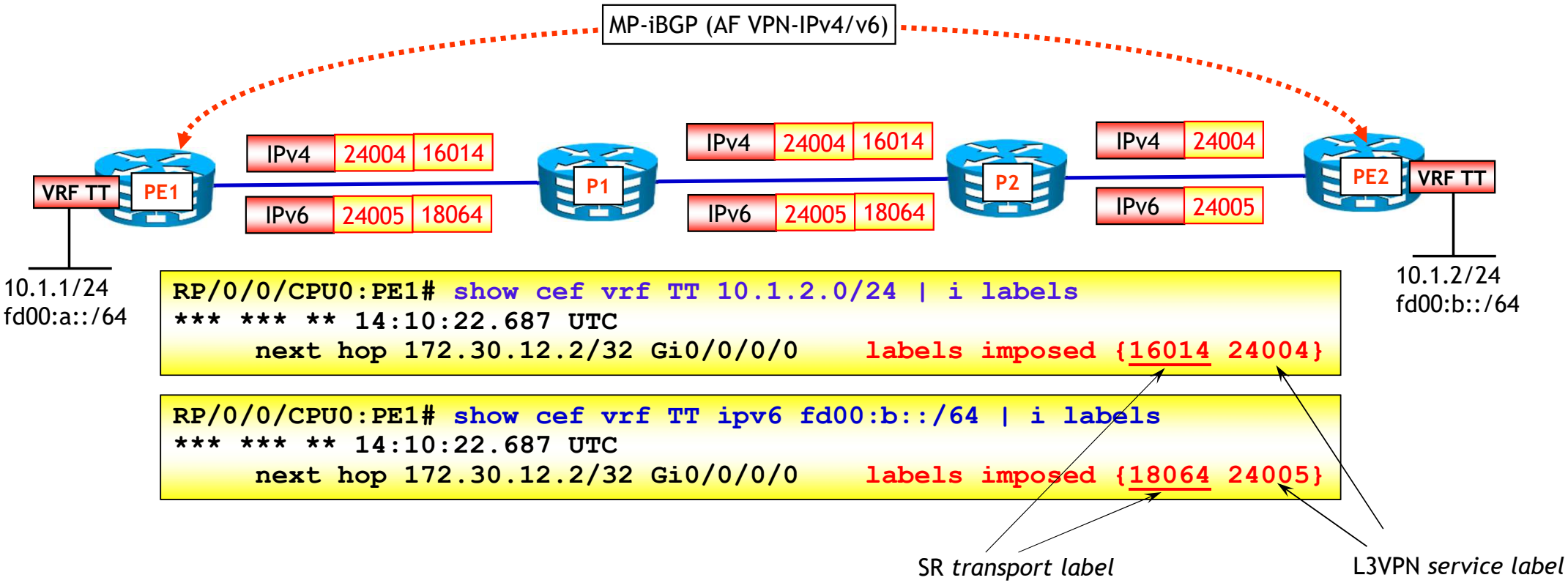
```
RP/0/0/CPU0:R3# show mpls forwarding labels 16014
. . .
Local  Outgoing    Prefix              Outgoing     Next Hop        Bytes
Label  Label       or ID               Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
16014  Pop         SR Pfx (idx 14)     Gi0/0/0/0    172.30.34.4     7248
```



PHP

```
RP/0/0/CPU0:R3# show mpls forwarding labels 18064
. . .
Local  Outgoing    Prefix              Outgoing     Next Hop        Bytes
Label  Label       or ID               Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
18064  Pop         SR Pfx (idx 2064)   Gi0/0/0/0    fe80::5200:ff:fe04:1   \
```

SR-MPLS and L3VPN



LDP vs RSVP-TE vs SR

REISS ROMOLI

	LDP	RSVP-TE	*Segment Routing*
Traffic Engineering support	No	Yes	Yex, with distributed and centralized control
Need for support from the IGP protocol	No	No	Yes
LSP P2MP support	Yes	Yes	Yes, via an external controller (PCE)
LSP MP2MP support	Yes	No	Not yet
Configuration simplicity	Yes	Yes, with *auto-tunnel*	Need to assign unique *Prefix-SID*
Control plane load	Low	Per-LSP state, *Soft-State*	Low
Deterministic *labels*	No	No	Only for global segments
Integration with SDN controllers	N/A	Yes (via PCEP)	Yes (via PCEP)
Fast ReRouting	Yes (*)	Yes, with backup tunnels	Yes, with TI-LFA (100% coverage, always)

(*) with coverage dependent on network topology

Copyright Tiziano Tofoni


Module 2: *Segment Routing* over MPLS (SR-MPLS)

#1	Main aspectes
#2	SRGB and SRLB
#3	Interworking LDP/Segment Routing
#4	From LDP to Segment Routing: migration plan
#5	Topology Independent LFA (TI-LFA)

Copyright Tiziano Tofoni


Segment Routing Global Block

REISS ROMOLI

- A Segment Routing Global Block (SRGB) is a block of labels reserved for global SIDs (Prefix-SIDs and Node-SIDs)
  - Each node locally allocates a SRGB
  - The allocated SRGB is advertised to other nodes via the IGP protocol
- An SRGB is identified by two parameters: base and amplitude
  - The base represents the first label, the amplitude is the number of allocable labels.
  - Example: if base=16,000 and amplitude=8,000, the globally allocable labels are all those belonging to the range [16,000-23,999]
- To ease the uniqueness of the MPLS label, an index may be defined via configuration for each node, and therefore the associated label is automatically assumed to be "base+index."
  - Some vendors, however, allow the configuration of the absolute value of the MPLS label t
  - IMPORTANT NOTE: The assigned indices must be different from node to node

Copyright Tiziano Tofoni


Recommended allocation

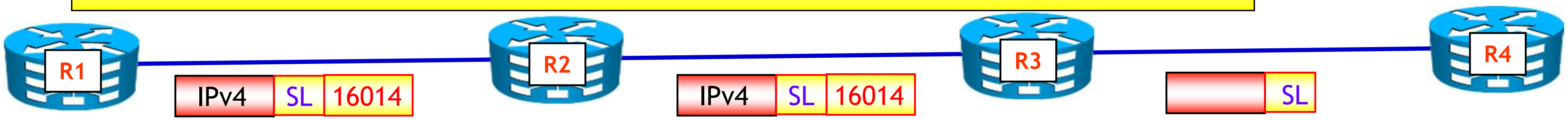
REISS ROMOLI

- *Best practice*: use the same SRGB on each node
  - "Recognizable" labels
  - Simplifies troubleshooting
  - Simplifies controller operations

- In case of routers from different vendors, this best practice must be managed via configuration
  - Example: Cisco IOS XE/XR (default: base=16.000 – amplitude=8.000)

Recommended allocation: example

```
RP/0/0/CPU0:R1#traceroute mpls ipv4 192.168.1.4/32
. . . < output omesso > . . .
  0 172.30.12.1 MRU 1500 [Labels: 16014 Exp: 0]
L 1 172.30.12.2 MRU 1500 [Labels: 16014 Exp: 0] 2 ms
L 2 172.30.23.3 MRU 1500 [Labels: implicit-null Exp: 0] 3 ms
! 3 172.30.34.4 5 ms
```



SRGB (R1)

16.000 (indice=0)
.
16.014 (indice=14)
.
23.999 (indice=7.999)

24.000
.
.
.
.
.
1.048.575

SRGB (R2)

16.000 (indice=0)
.
16.014 (indice=14)
.
23.999 (indice=7.999)

24.000
.
.
.
.
.
1.048.575

SRGB (R3)

16.000 (indice=0)
.
16.014 (indice=14)
.
23.999 (indice=7.999)

24.000
.
.
.
.
.
1.048.575

SRGB (R4)

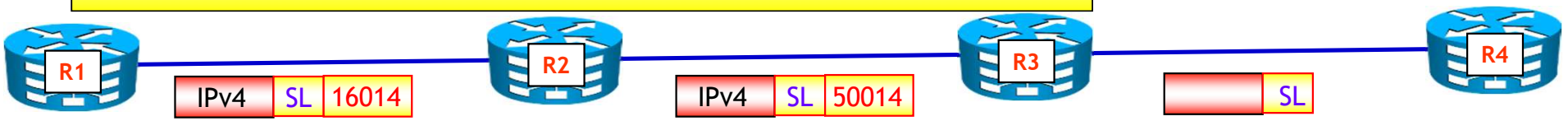
16.000 (indice=0)
.
16.014 (indice=14)
.
23.999 (indice=7.999)

24.000
.
.
.
.
.
1.048.575

Copyright Tiziano Tofoni

Allocation allowed but not recommended: example

```
RP/0/0/CPU0:R1#traceroute mpls ipv4 192.168.1.4/32
. . . < output omesso > . . .
  0 172.30.12.1 MRU 1500 [Labels: 16014 Exp: 0]
L 1 172.30.12.2 MRU 1500 [Labels: 50014 Exp: 0] 3 ms
L 2 172.30.23.3 MRU 1500 [Labels: implicit-null Exp: 0] 5 ms
! 3 172.30.34.4 4 ms
```



SRGB

16.000 (indice=0)
.
16.014 (indice=14)
.
23.999 (indice=7.999)
24.000
.
.
.
.
.
1.048.575

SRGB

16.000 (indice=0)
.
16.014 (indice=14)
.
23.999 (indice=7.999)
24.000
.
.
.
.
.
1.048.575

SRGB

16.000
.
49.999
50.000 (indice=0)
.
50.014 (indice=14)
.
59.999 (indice=9.999)
60.000
.
.
1.048.575

SRGB

16.000 (indice=0)
.
16.014 (indice=14)
.
23.999 (indice=7.999)
24.000
.
.
.
.
.
1.048.575

Segment Routing Local Block

REISS ROMOLI

- A Segment Routing Local Block (SRLB) is a block of labels reserved for the manual (static) definition of local SIDs (Adjacency-SIDs)
  - Each node allocates a block of labels for local use
  - Example: default SRLB in Cisco routers: 15,000 – 15,999

- Similar to an SRGB, an SRLB is identified by two parameters: base and amplitude

.

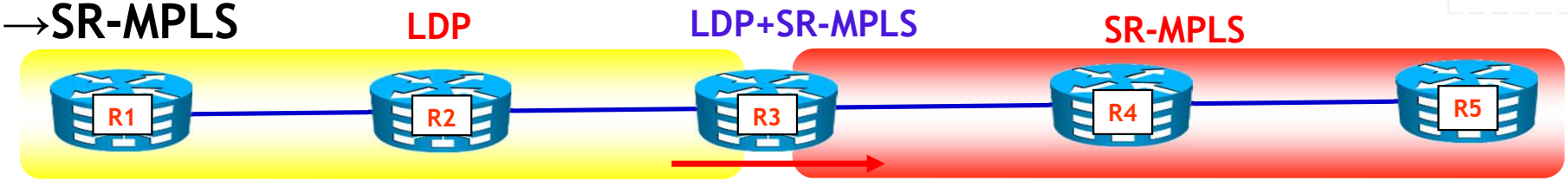
Module 2: *Segment Routing* over MPLS (SR-MPLS)

#1	Main aspectes
#2	SRGB and SRLB
#3	Interworking LDP/Segment Routing
#4	From LDP to Segment Routing: migration plan
#5	Topology Independent LFA (TI-LFA)

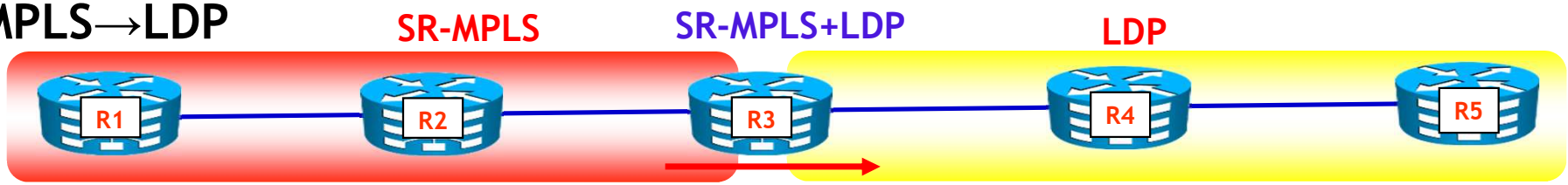
Copyright Tiziano Tofoni


Interworking models

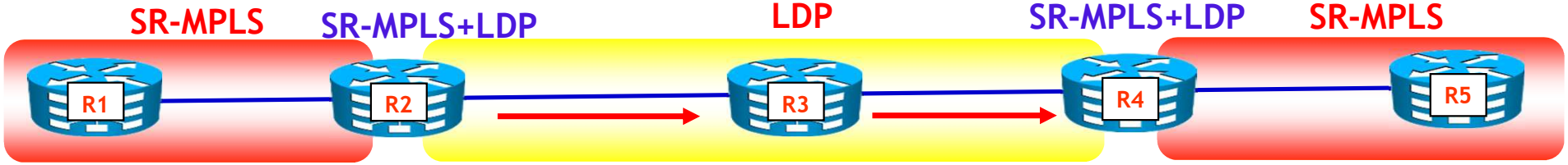
- LDP→SR-MPLS



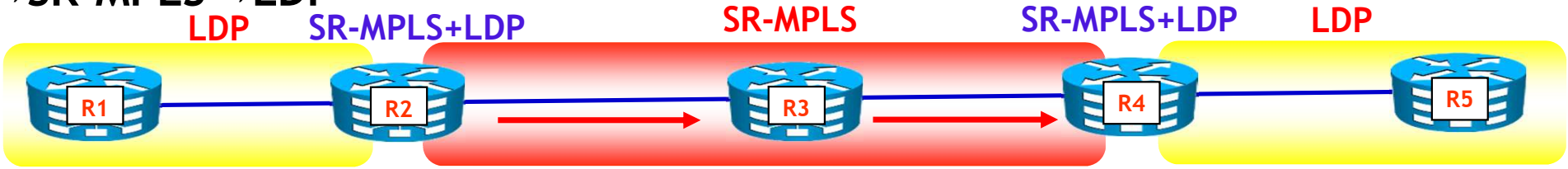
- SR-MPLS→LDP



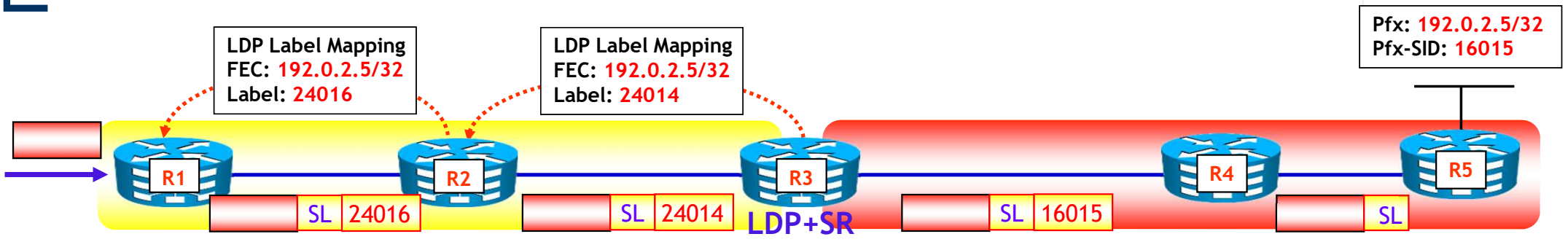
- SR-MPLS→LDP→SR-MPLS



- LDP→SR-MPLS→LDP



Interworking LDP→SR: example (1/2)



In label	Out label
24000	
24007	24016
.	
.	
.	
1048575	

In label	Out label
24000	
24016	24014
.	
.	
.	
1048575	

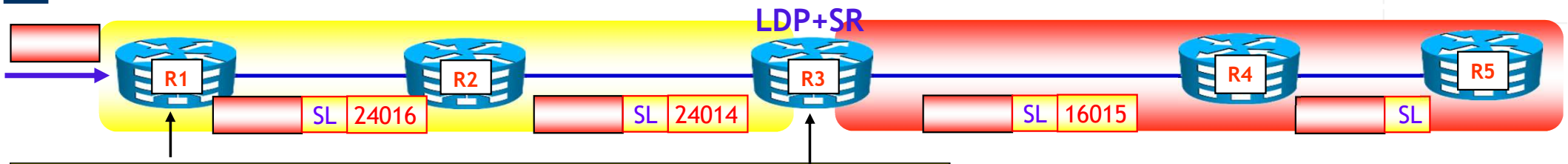
In label	Out label
16015	16015
24000	
24014	16015
.	
.	
.	
1048575	

SRGB

In label	Out label
16015	POP
24000	
.	
.	
.	
.	
1048575	

SRGB

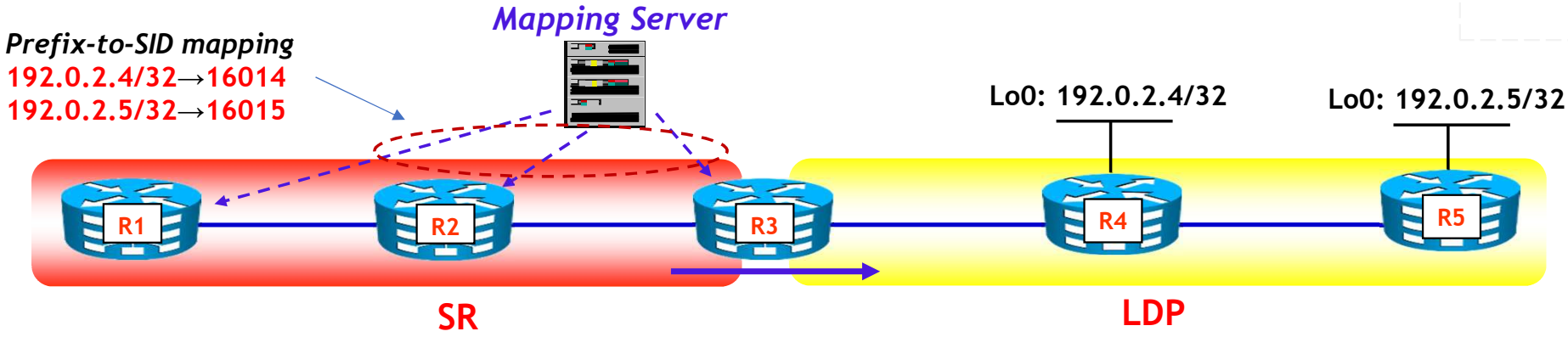
Interworking LDP→SR: example(2/2)




```
RP/0/0/CPU0:R1# traceroute mpls ipv4 192.0.2.5/32
*** *** 14:13:41.310 UTC
Tracing MPLS Label Switched Path to 192.0.2.5/32, timeout is 2 seconds
. . . < output omesso > . .
  0 172.30.12.1 MRU 1500 [Labels: 24016 Exp: 0]
L 1 172.30.12.2 MRU 1500 [Labels: 24014 Exp: 0] 0 ms
L 2 172.30.23.3 MRU 1500 [Labels: 16015 Exp: 0] 10 ms
L 3 172.30.34.4 MRU 1500 [Labels: implicit-null Exp: 0] 0 ms
! 4 172.30.45.5 10 ms
```

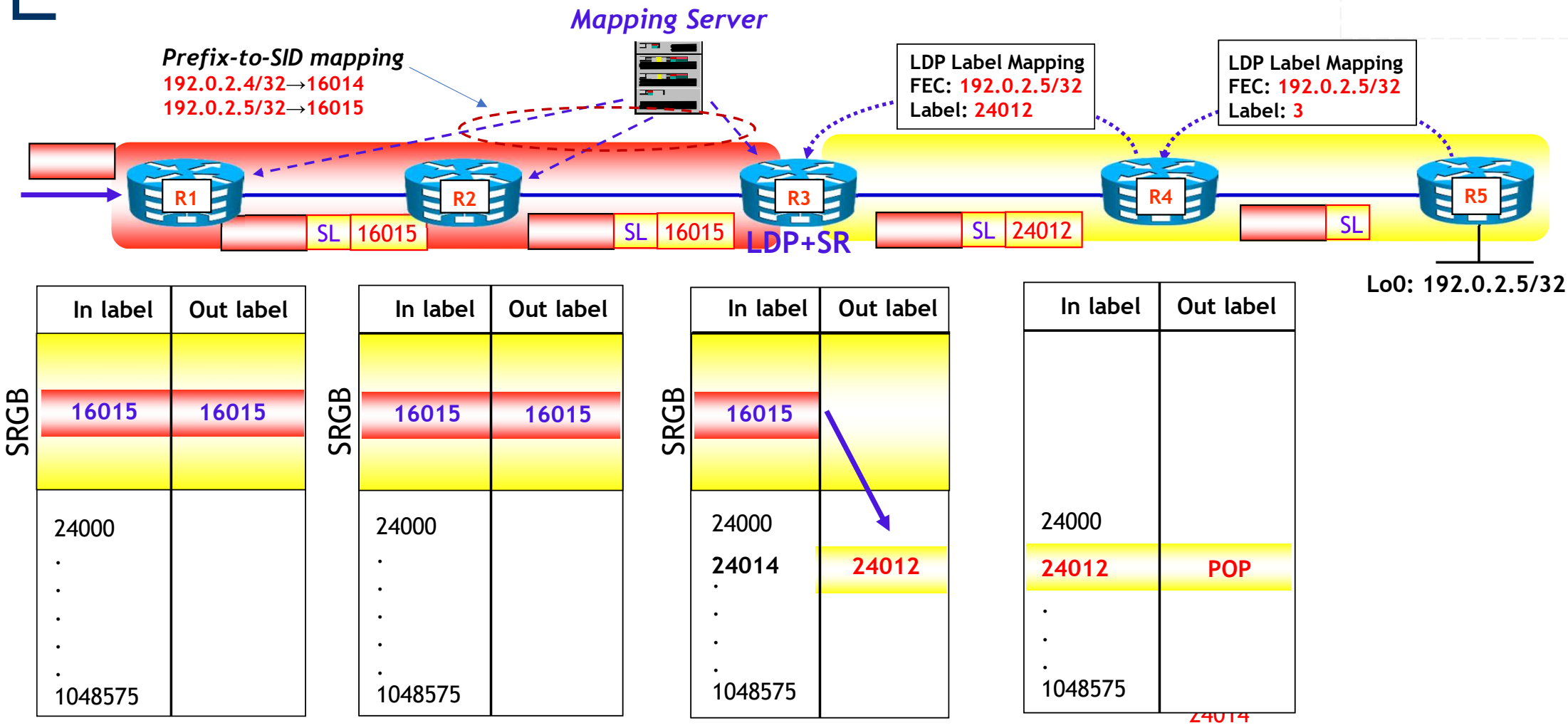
```
RP/0/0/CPU0:R3#show mpls ldp binding 192.0.2.5/32
*** *** 14:09:02.619 UTC
192.0.2.5/32, rev 40
        Local binding: label: 24014
        Remote bindings: (1 peers)
            Peer                 Label
            -----------------    ---------
            192.0.2.2:0          24016

RP/0/0/CPU0:R3#show mpls forwarding labels 24014
*** *** 14:09:24.567 UTC
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ----------
24014  16015       192.0.2.5/32      Gi0/0/0/0    172.30.34.4     248
```

Interworking SR→LDP: operation

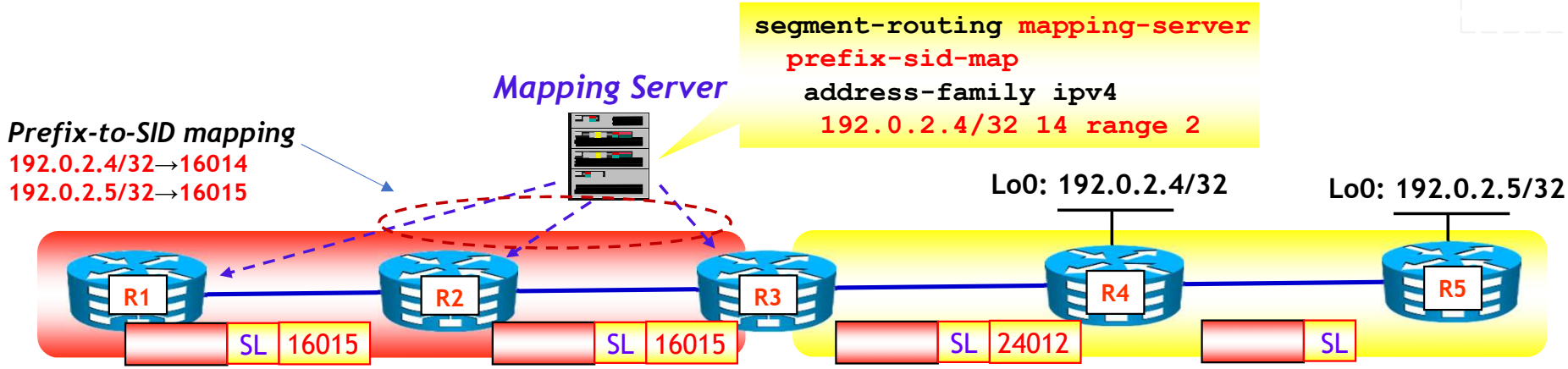


- A Mapping Server announces the Prefix-SIDs 16014 and 16015 associated with the Loopback0 interfaces of R4 and R5 (Prefix-to-SID mapping) on behalf of nodes R4 and R5
  - All SR-MPLS-enabled nodes receive the prefix-to-SID mapping
  - SR nodes use prefix-to-SID mapping and install in their LFIB the Prefix-SIDs associated by the Mapping Server with nodes R4 and R5
  - If no "native" Prefix-SID is available, the SR nodes use prefix-to-SID mapping
- Final result: nodes R1 and R2 have SR connectivity to nodes R4 and R5

Interworking SR→LDP: example (1/2)



Interworking SR→LDP: example (2/2)




```
RP/0/0/CPU0:R1# traceroute mpls ipv4 192.0.2.5/32
*** *** 16:27:10.493 UTC
Tracing MPLS Label Switched Path to 192.0.2.5/32, timeout is 2 seconds
. . . < output omesso > . .
  0 172.30.12.1 MRU 1500 [Labels: 16015 Exp: 0]
L 1 172.30.12.2 MRU 1500 [Labels: 16015 Exp: 0] 0 ms
L 2 172.30.23.3 MRU 1500 [Labels: 24012 Exp: 0] 8 ms
L 3 172.30.34.4 MRU 1500 [Labels: implicit-null Exp: 0] 9 ms
! 4 172.30.45.5 10 ms
```

```
RP/0/0/CPU0:R3#show mpls forwarding labels 16015
*** *** 16:46:21.174 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ---------
16015  24012       SR Pfx (idx 15)    Gi0/0/0/0    172.30.34.4     1224
```

Module 2: *Segment Routing* over MPLS (SR-MPLS)

#1	Main aspectes
#2	SRGB and SRLB
#3	Interworking LDP/Segment Routing
#4	From LDP to Segment Routing: migration plan
#5	Topology Independent LFA (TI-LFA)

Copyright Tiziano Tofoni


The migration plan

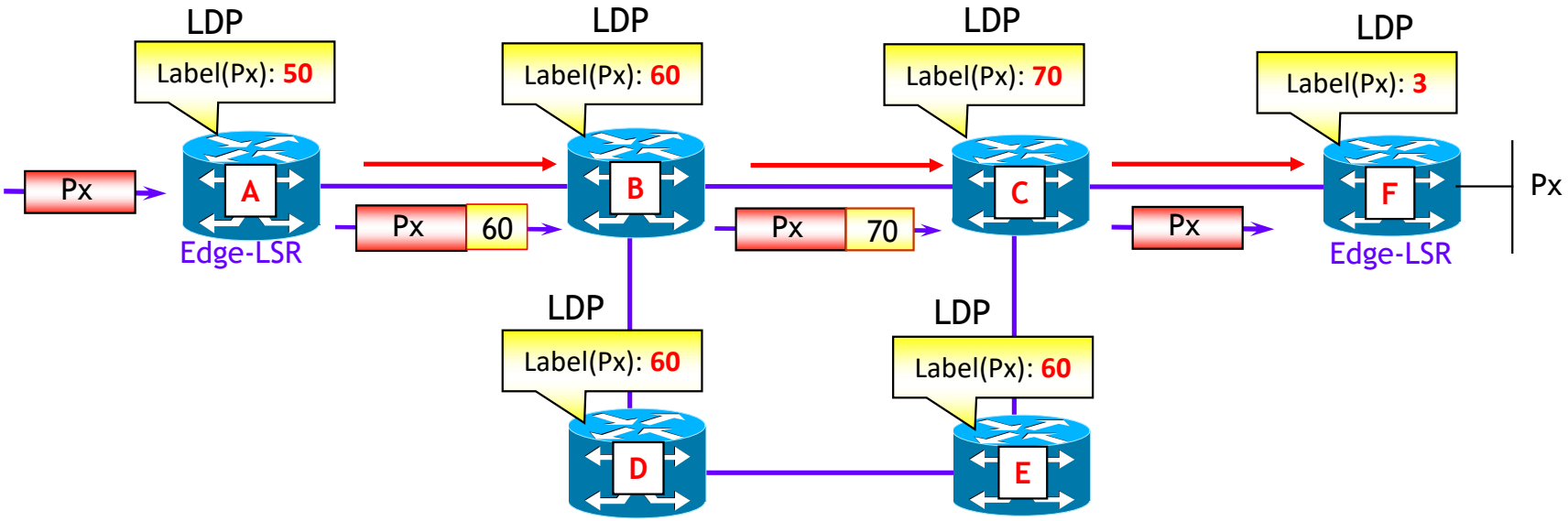
REISS ROMOLI

- Step 0: *Assess the network for Segment Routing*
  - Which devices support SR?
  - Which only support LDP?
  - Which require hardware and/or software updates?
- Step 1: Introduction of SR (in parallel to LDP)
  - Define the SRGB and eliminate any overlaps in the use of MPLS labels
  - Prefix-SIDs planning (global SIDs)
  - Verify the need for any Mapping Servers and their possible placement
  - Verify the SR control plane and data plane
  - Verify any SR↔LDP interworking (if necessary)
- Step 2: Reverse the LDP/SR preference order and verify operation
  - Possibly start with a subset of nodes
- Step 3: Disable LDP
- Step 4 (optional): Enable TI-LFA

Copyright Tiziano Tofoni

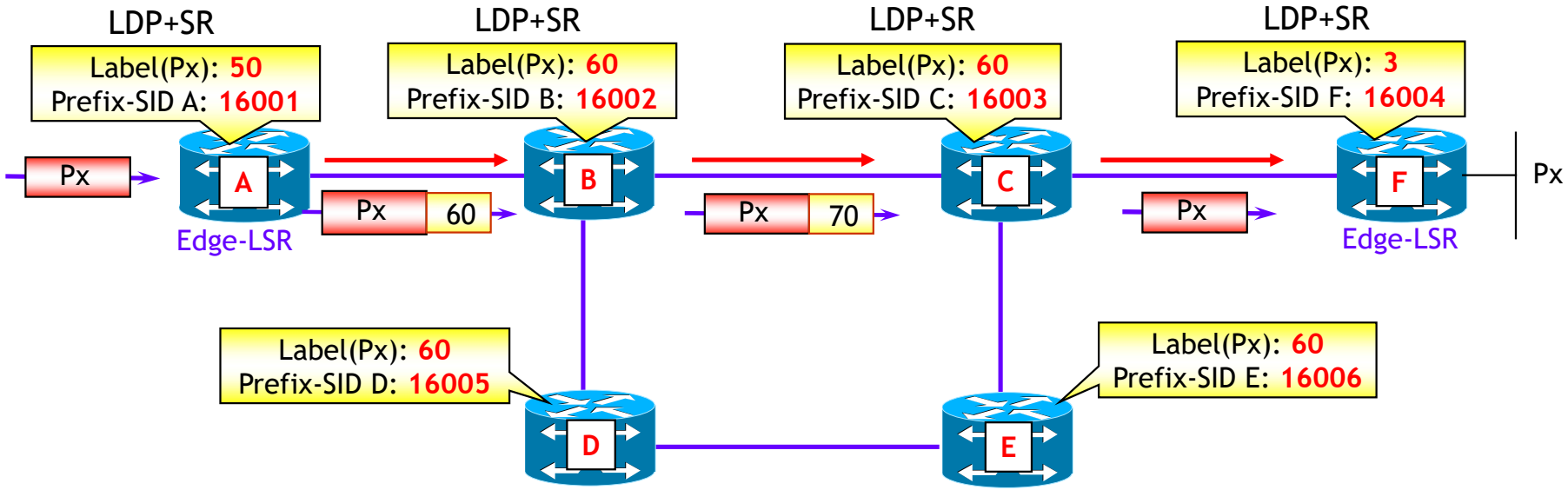

Initial state

- Assumptions:
  - The network uses only LDP to distribute <FEC, label> associations
  - All routers support SR
  - No need for a Mapping Server



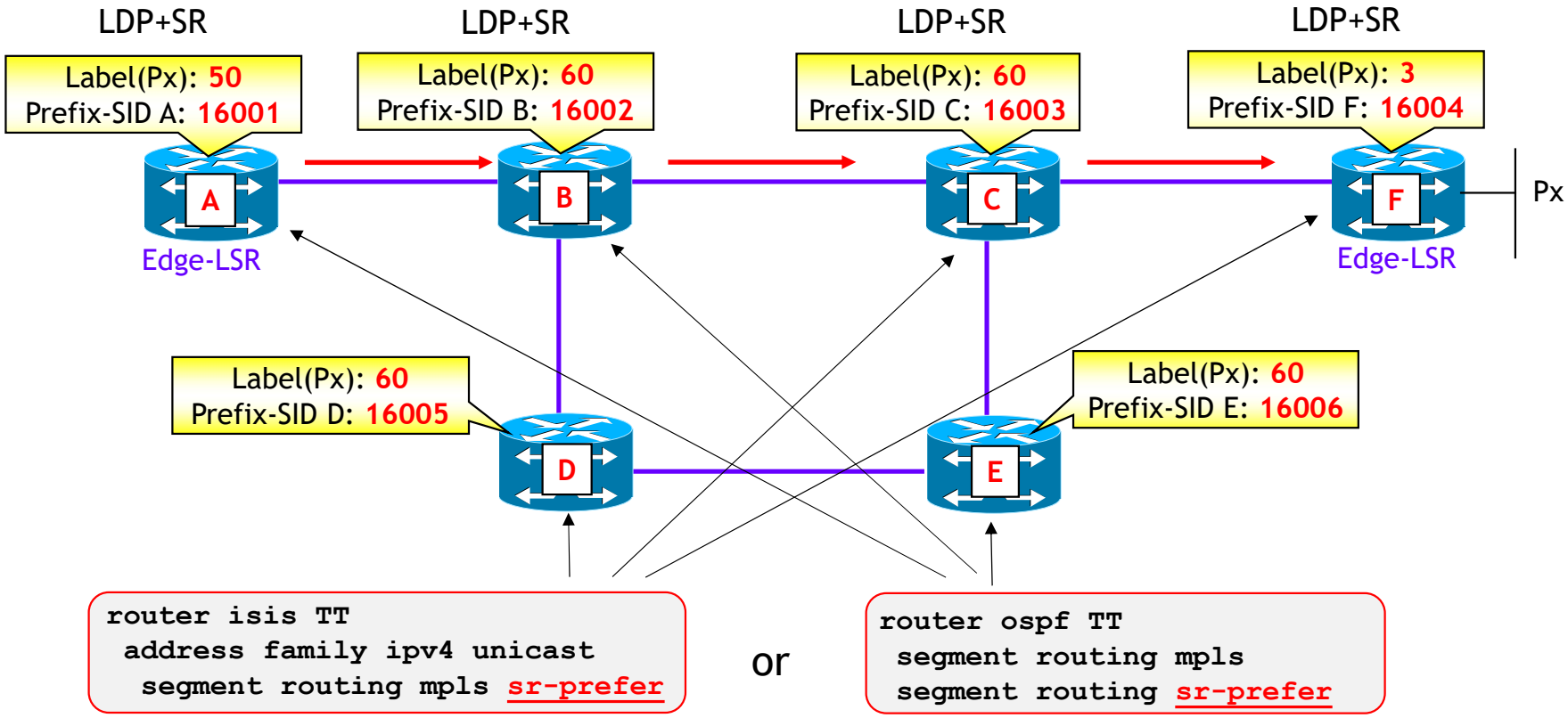
Enable Segment Routing

- **First step:** enable *Segment Routing* on each router
  - Define SRGB (best practice: same SRGB on all routers)
  - Check for any LDP/SR label overlaps
- NOTE: The order in which Segment Routing is activated on the routers is irrelevant



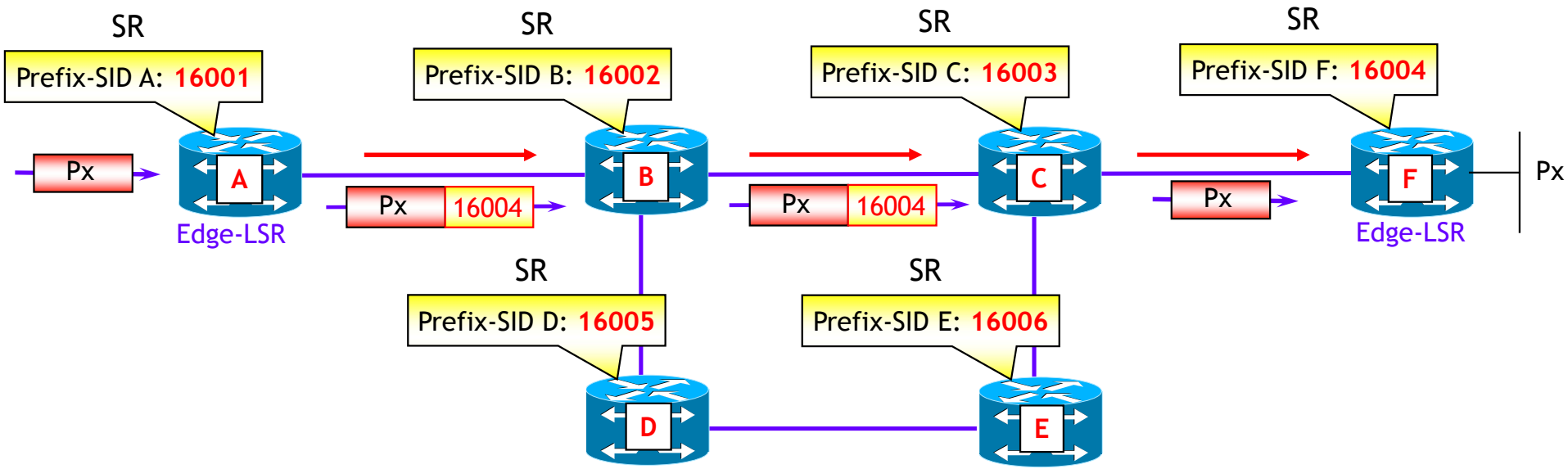
Change the preference order

- **Second step**: on all routers change the preference order (from LDP to SR)
  - The sequence in which it is done is irrelevant



Disable LDP

- **Third step**: on all routers disable LDP
  - The sequence in which it is done is irrelevant



Module 2: *Segment Routing* over MPLS (SR-MPLS)

#1	Main aspectes
#2	SRGB and SRLB
#3	Interworking LDP/Segment Routing
#4	From LDP to Segment Routing: migration plan
#5	Topology Independent LFA (TI-LFA)

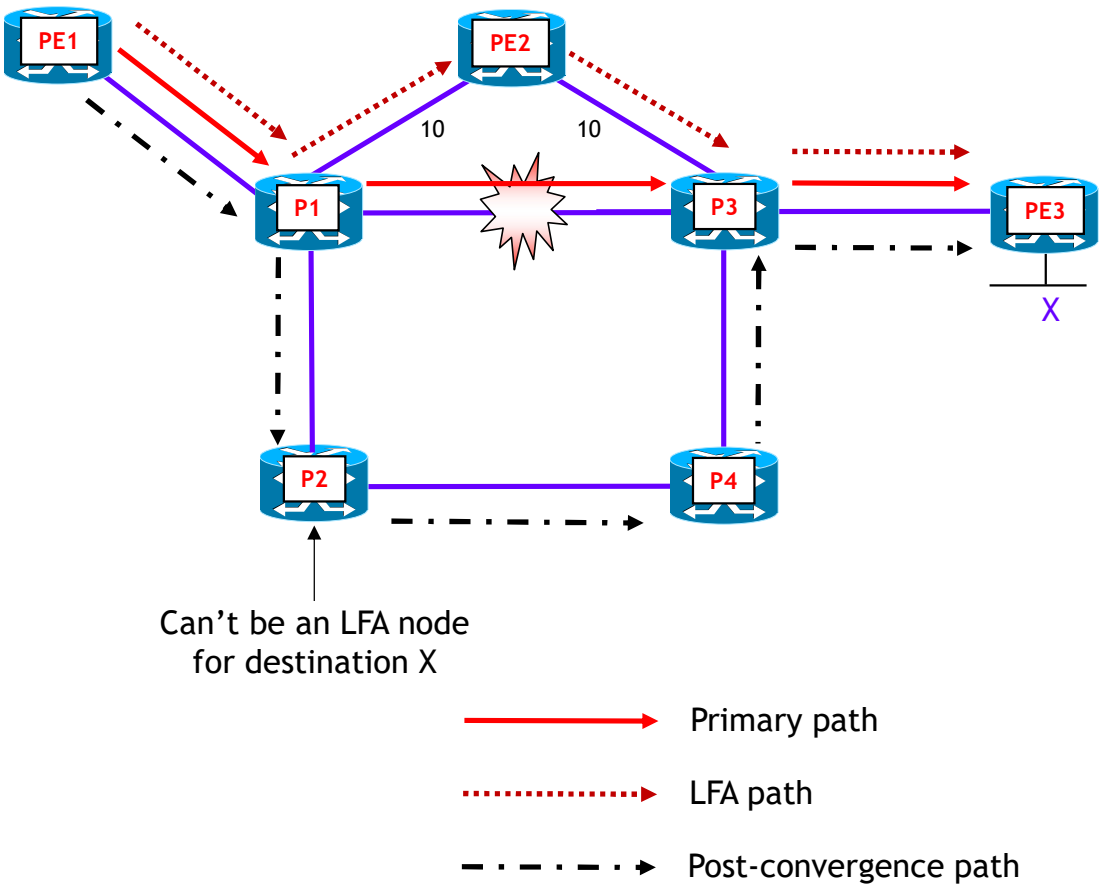
Copyright Tiziano Tofoni


Prologue: *Loop Free Alternate* (LFA)

- *Per-prefix* LFA: simple, automatic, local Fast Rerouting technique, sub-50 msec convergence

- Operation
  - The IGP protocol predetermines a primary and a backup (loop-free) next hop, if available
  - Both primary and backup next hops are programmed into the FIB

- In the event of an outage, all backup paths to the affected destinations are enabled (<50 msec LoC (Loss of Connectivity))



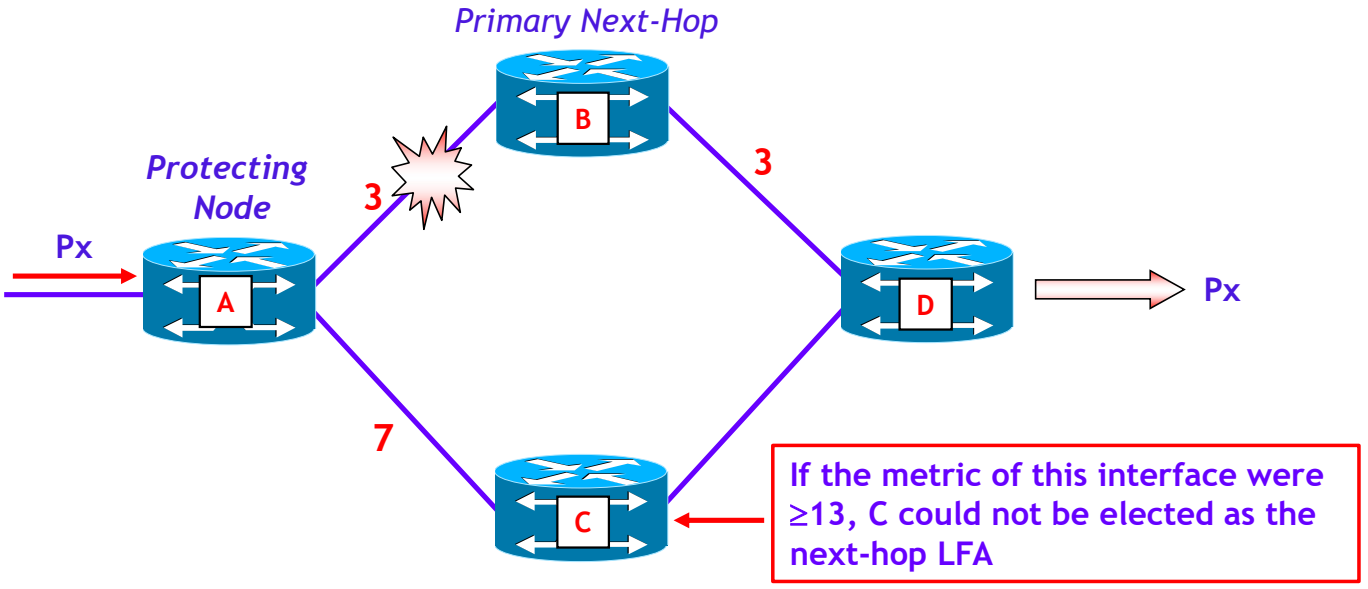
Classic LFA : pros

REISS ROMOLI

- Simple
  - The router determines the alternative Next-Hop automatically
  - No protocol changes, no interoperability issues, incremental deployment
- Low convergence time
  - Essentially equal to the time to detect an outage (typically <50 ms)
  - The alternate next-hop is pre-calculated and pre-installed in the FIB, then used when an outage is detected
  - Independent of the number of prefixes in the FIB
- Allows to protect traffic from links and/or nodes outages

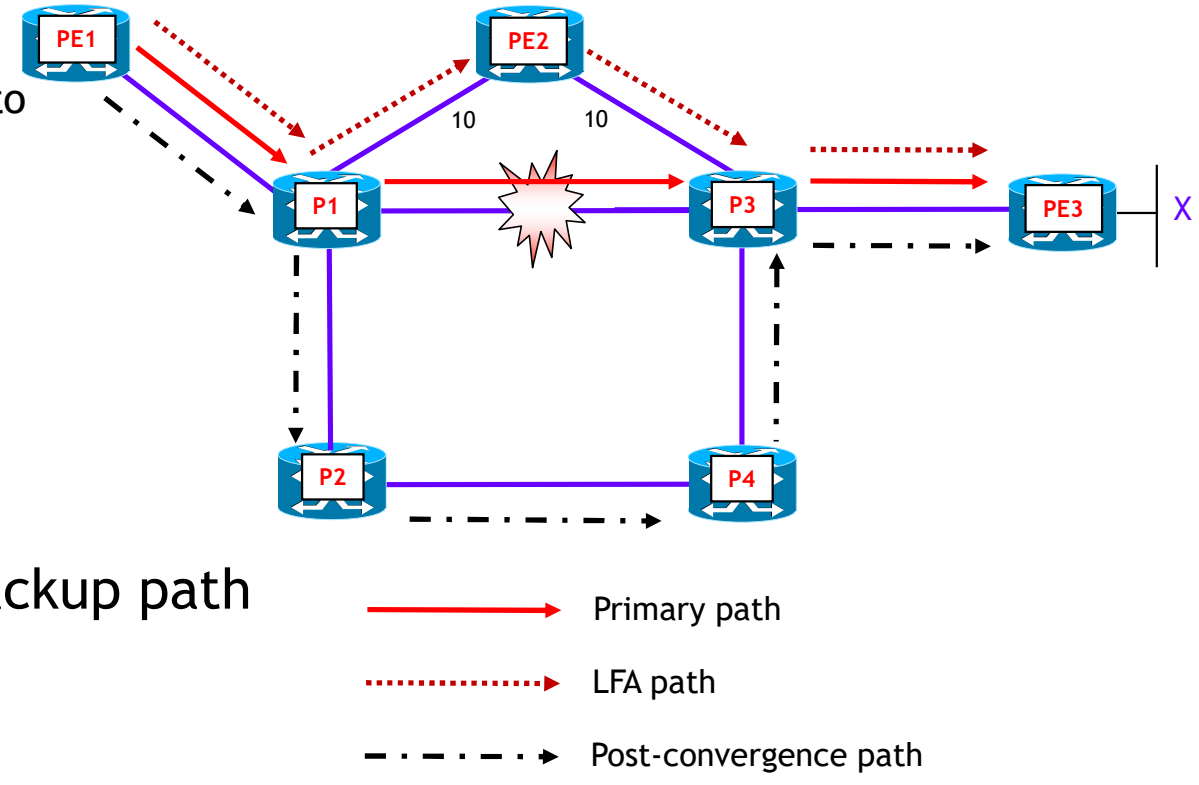
Classic LFA: cons (1/2)

- 100% coverage not always achievable: it depends on the logical network topology and IGP metrics
  - The coverage degree is the % of destination prefixes for which a LFA Next-Hop (i.e., a loop-free backup Next-Hop) can be determined
- Example



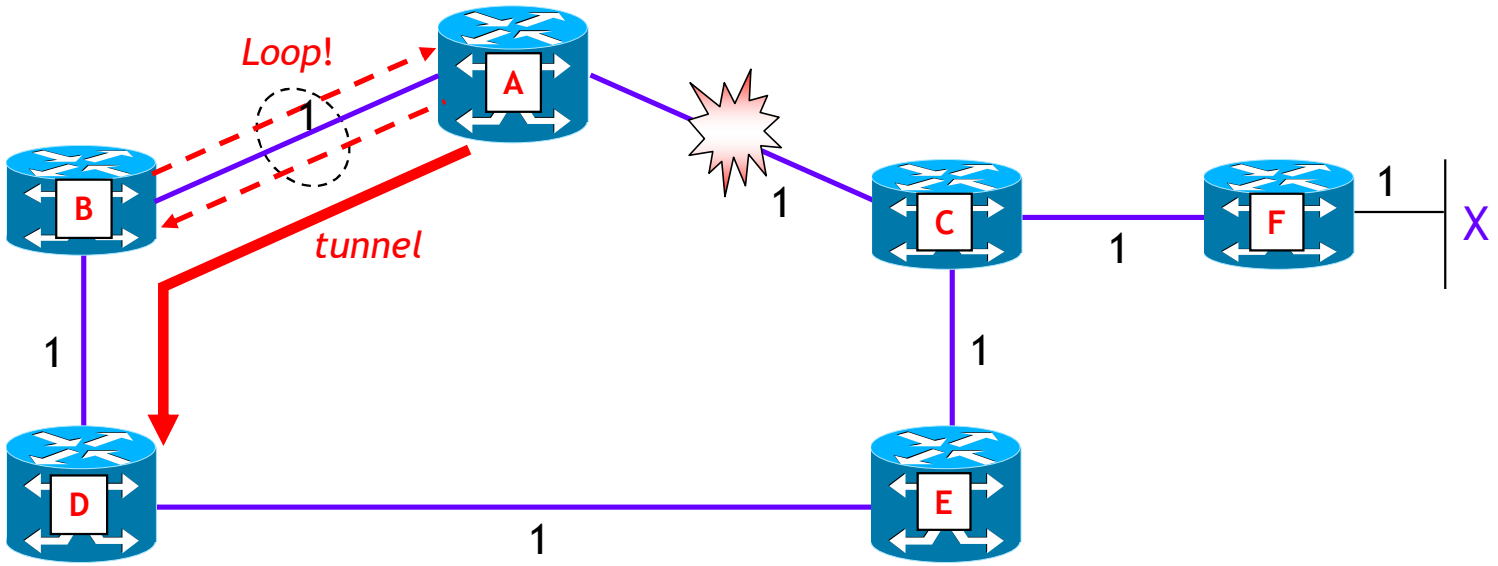
Classic LFA: cons (2/2)

- Backup path is sub-optimal
  - The path may not have enough bandwidth to support transit traffic
  - Node P1 uses a PE router (PE2) as a backup Next-Hop to protect a link of the core network
    - A basic design rule is to avoid using links to PEs as transit



- The ideal (and natural) choice of backup path is to use the post-convergence path

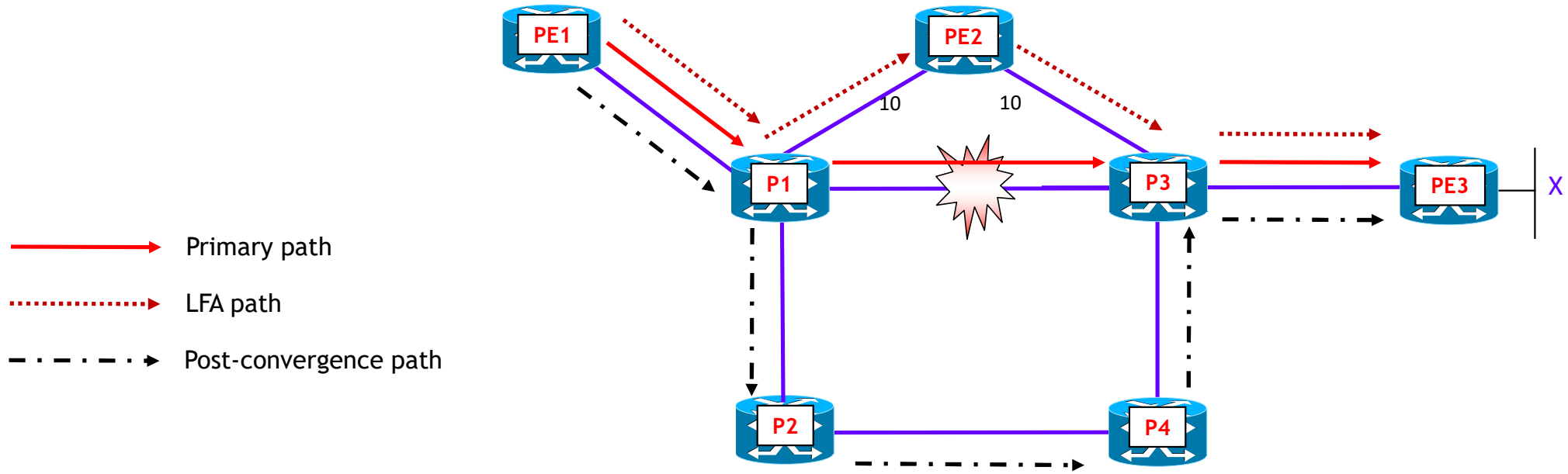
How to improve LFA coverage: Remote LFA



- Node B cannot be elected as the next-hop LFA for prefix X
- Node D can be elected as the next-hop LFA for prefix X, but is not directly connected
- Solution (Remote LFA): deploy a tunnel between A and D
  - The tunnel can be of any type : GRE, MPLS, etc.

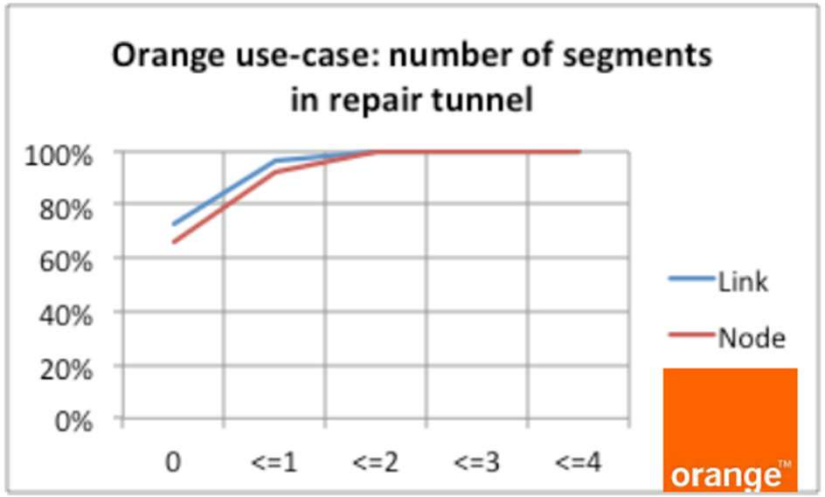
TI-LFA (Topology Independent-LFA) protection

- Guarantees 100% coverage for any network topology
- Use of post-convergence path as a backup path
- Use of Segment Routing to route traffic on post-convergence path



Post-convergence path: length of the *Segment List*

- Link protection with symmetric metrics
  - ≤ 2 segments (mathematically provable)
- Asymmetric metrics : SRLG groups, link, node protection
  - No theoretical limits
  - But in reality, things are simpler than they might seem ...
- *Case study* on a real network (Orange)
  - Link protection: 100%
    - ≤ 2 segments in 100% of cases
    - Node protection: ≤ 4 segments in 100% of cases
    - 99,72% ≤ 2 segments
    - 0,24% uses 3 segments
    - 0,04% uses 4 segments



Ref. Orange @ MPLS/SDN WC 2014 – "Fast Reroute Approach Using Segment Routing"

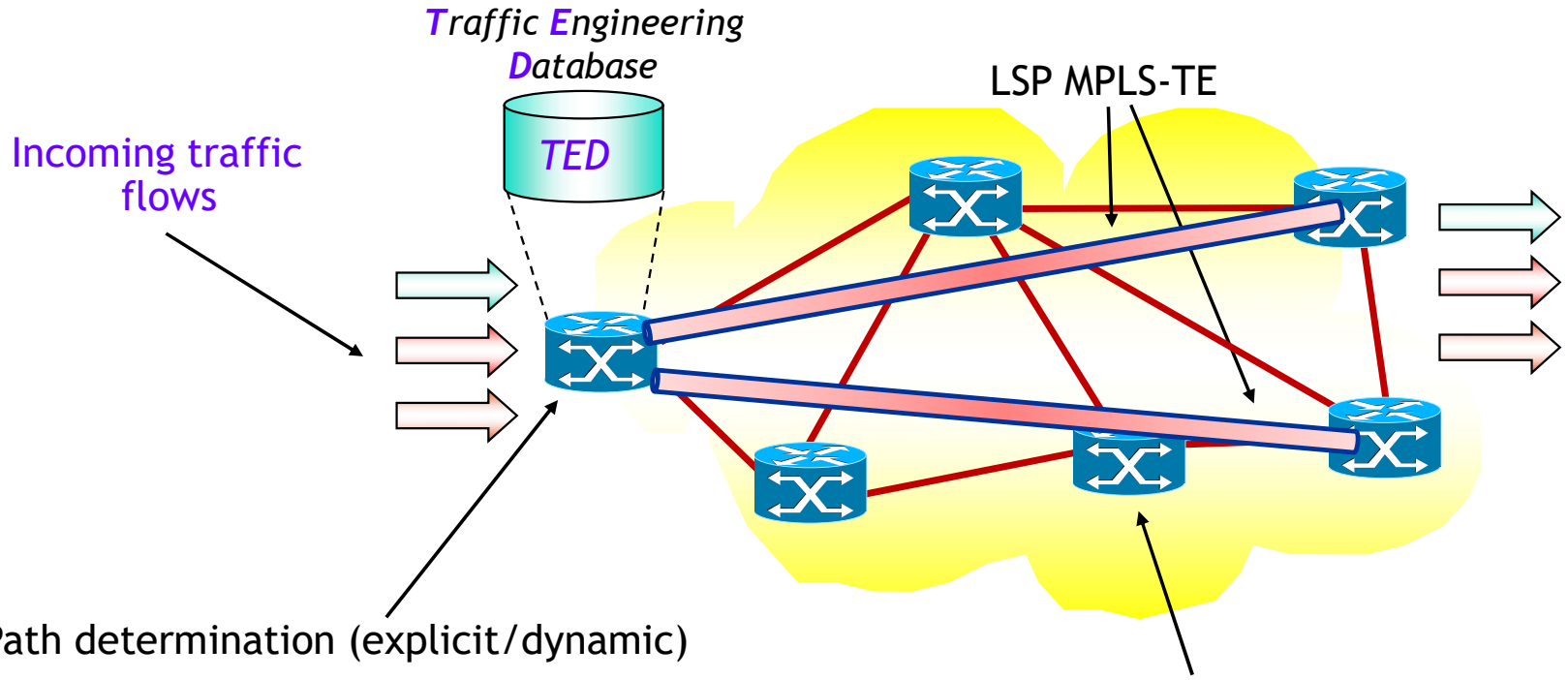
Module 3: *Segment Routing Traffic Engineering* (SR-TE)

#1 Basic aspects of *Traffic Engineering*

#2 SR *policy*

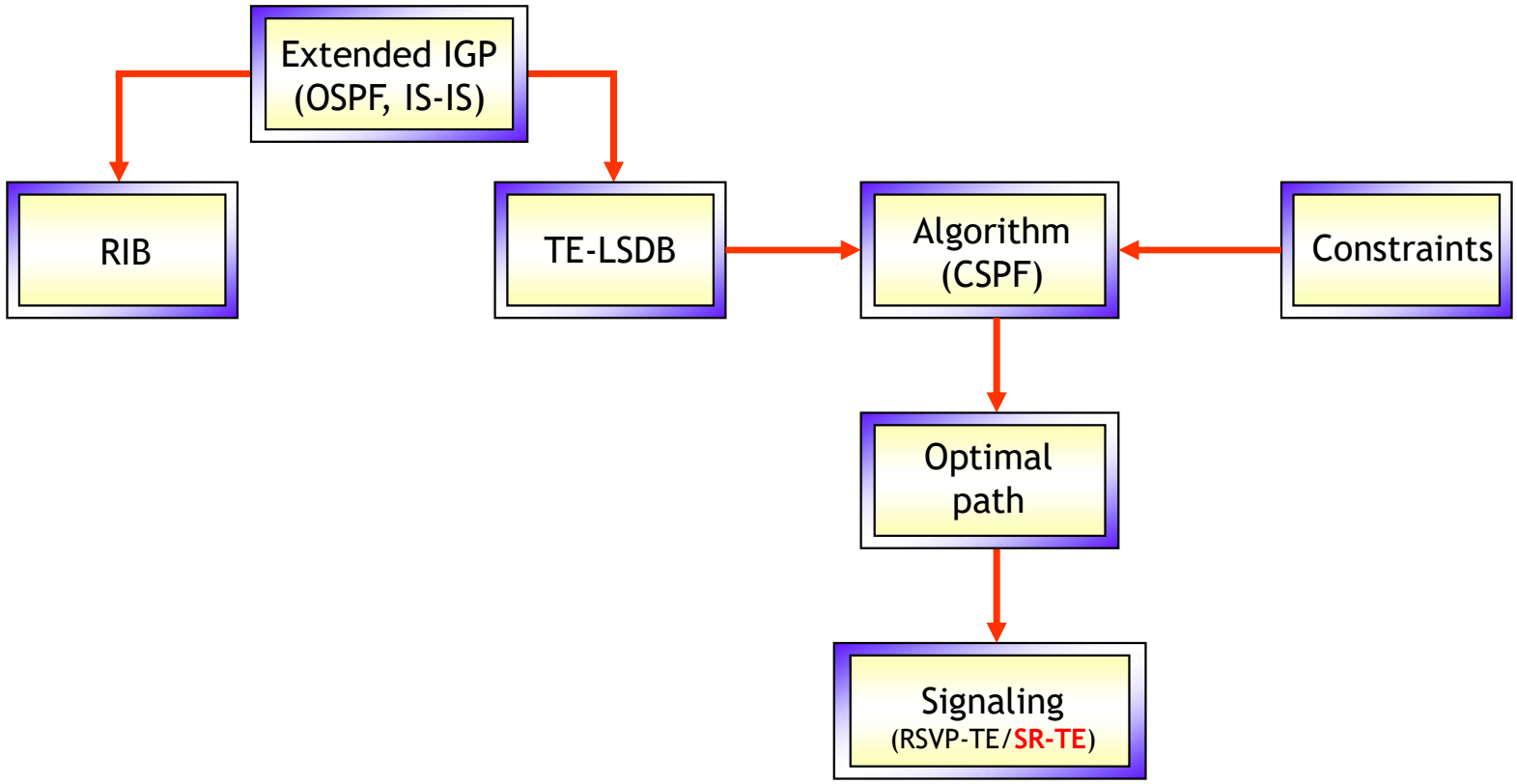
Copyright Tiziano Tofoni

Traffic Engineering over MPLS



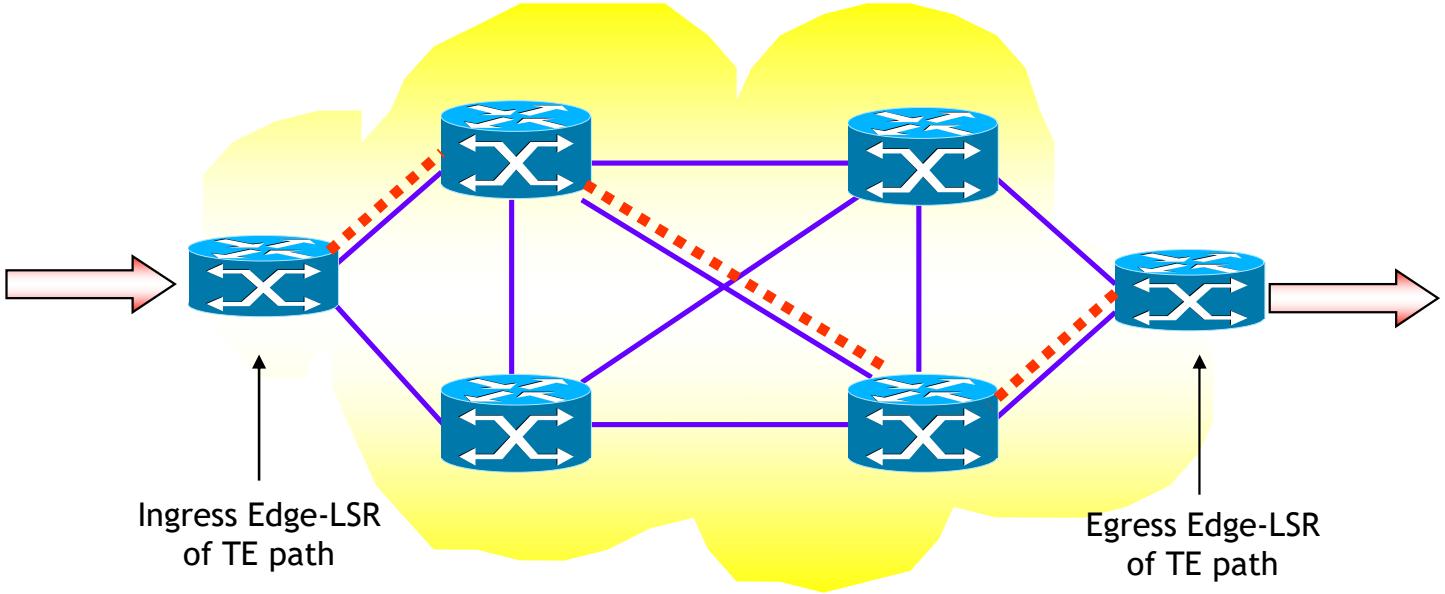
- Path determination (explicit/dynamic)
- Signaling (RSVP-TE, **SR-TE**)

The logical flow of operations



Path determination

- Two possible approaches
  - *Off-line*: through specific tools developed in-house or supplied by vendors
  - *On-line*: *Constraint-Based Routing (CBR)*



Module 3: *Segment Routing Traffic Engineering* (SR-TE)

#1 Basic aspects of *Traffic Engineering*

#2 SR *policy*

Copyright Tiziano Tofoni

WARNING: In this tutorial, SR-TE will be discussed exclusively on the MPLS data plane. The concepts of SR-TE on the IPv6 data plane are identical

Definition (1/2)

- An SR policy defines one or more paths between two Edge-LSRs, which are determined explicitly or dynamically
  - The SR policy also defines how these paths should be determined and the order of preference in which they are used
- An SR policy has three components: <*Head-end*, Color, *End-point*>
  - *Head-end*: This is the ingress Edge-LSR where the SR policy is instantiated
  - Color: A numeric value used to differentiate SR policies between the same Head-end and End-point
  - *End-point*: It is the egress Edge-LSR of the *SR policy*

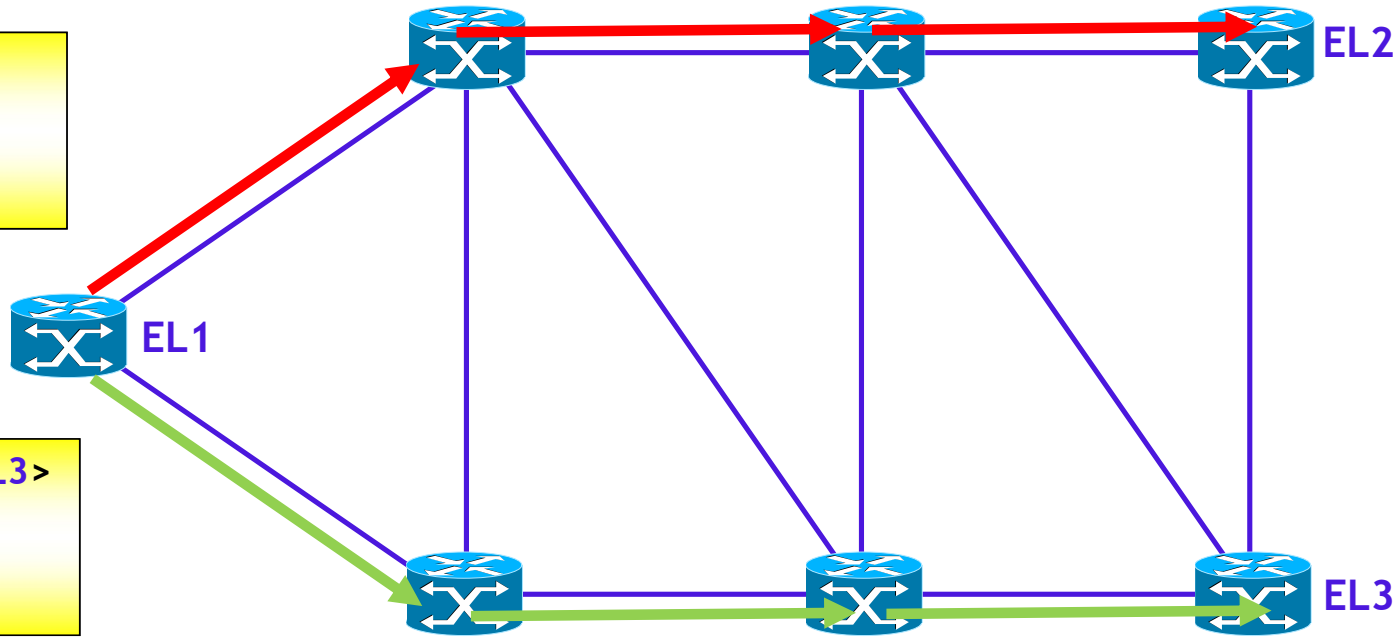
Copyright Tiziano Tofoni


Definition (2/2)

- Example of *SR policies*

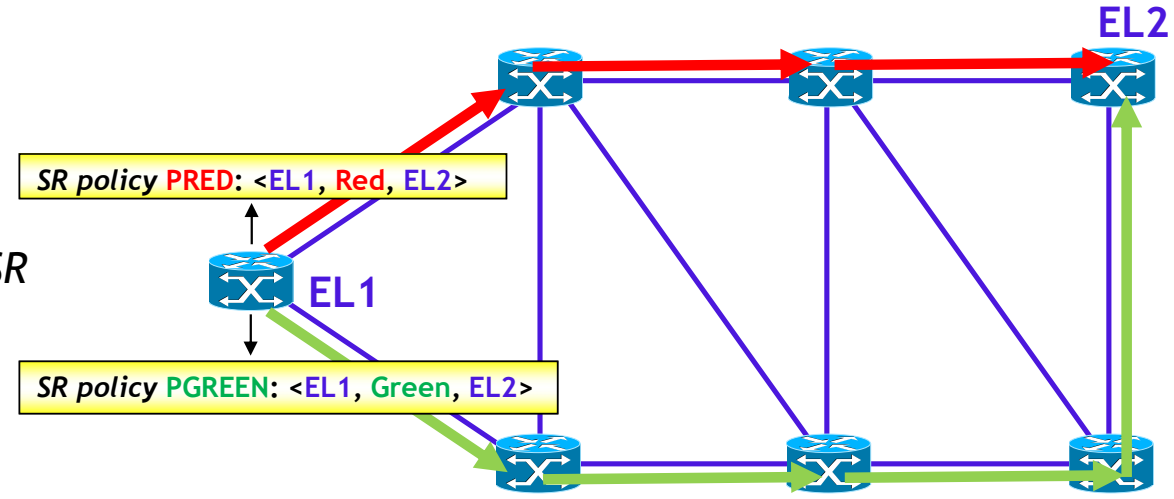
SR policy PRED: <EL1, red, EL2>
- *Head-end*: EL1
- Color: red
- *End-point*: EL2

SR policy PGREEN: <EL1, green, EL3>
- *Head-end*: EL1
- Color: green
- *End-point*: EL3



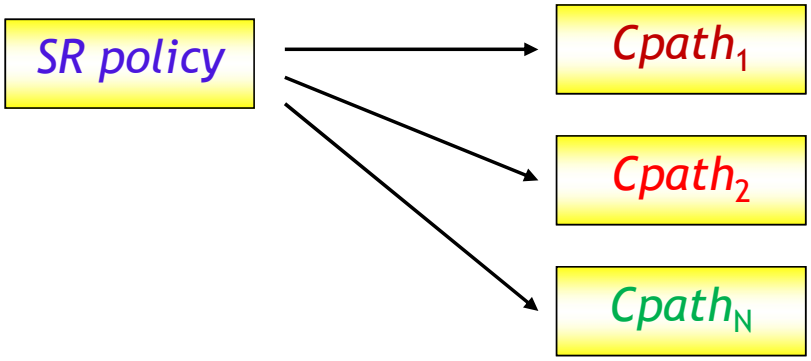
What is color for?

- Each SR policy is associated with a color
  - The color can be used to indicate a specific treatment (SLA policy)
- Between a pair of Head-end (H) and End-point (E) nodes, only one SR policy with a specific color (C) can exist
  - In other words: each triple <H, C, E> is unique
- Example
  - *Low-cost*=Red, *Low-delay*= Green
  - Forward Low-cost traffic to EL2 using the *SR policy* PRED <EL1, Red, EL2>
  - Forward Low-delay traffic to EL2 using the *SR policy* PGREEN <EL1, Green, EL2>



Candidate paths (1/2)

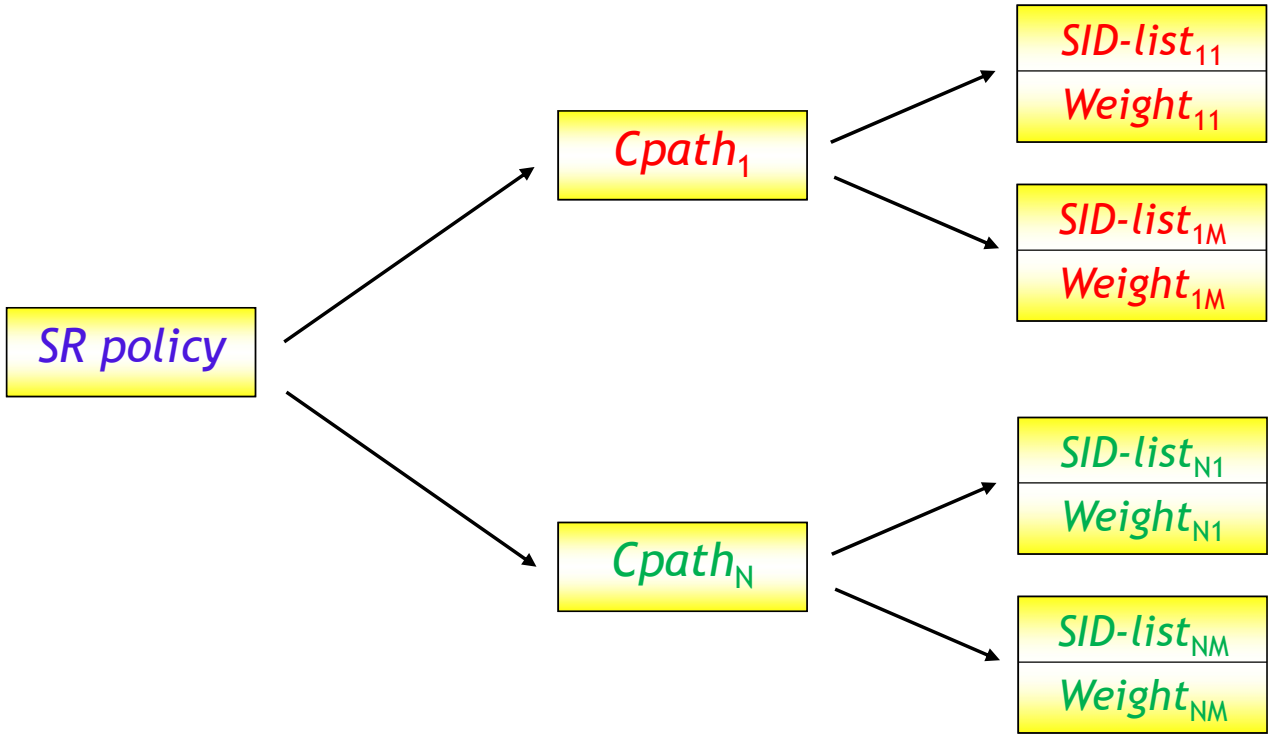
- An SR policy consists of one or more candidate paths (*Cpath*)



- Each Cpath can be determined explicitly or dynamically
- Each SR policy instantiates only one path in the RIB/FIB
  - The preferred valid Cpath is instantiated in the FIB

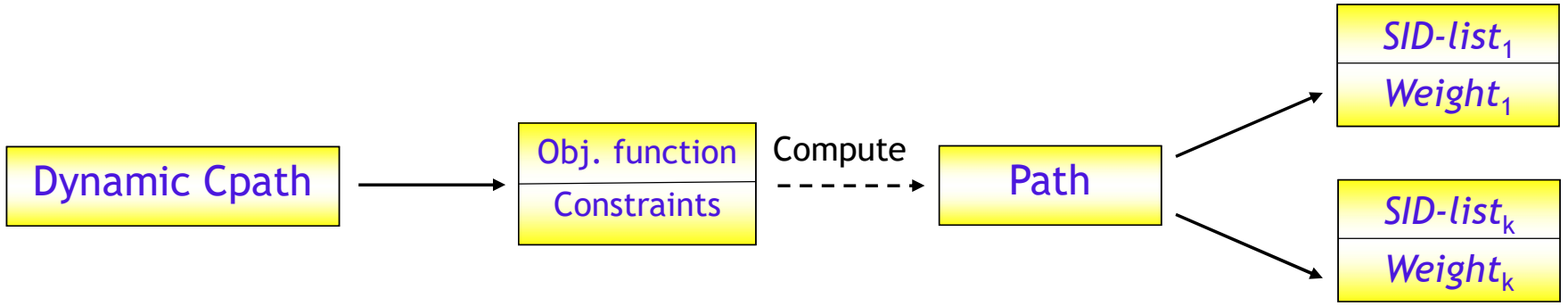
Candidate paths (2/2)

- Each Cpath consists of a Segment List (or SID-list) or a weighted set of Segment Lists (Weighted SID-lists)
- Traffic using a given SR policy is distributed among all the SID-lists on the path



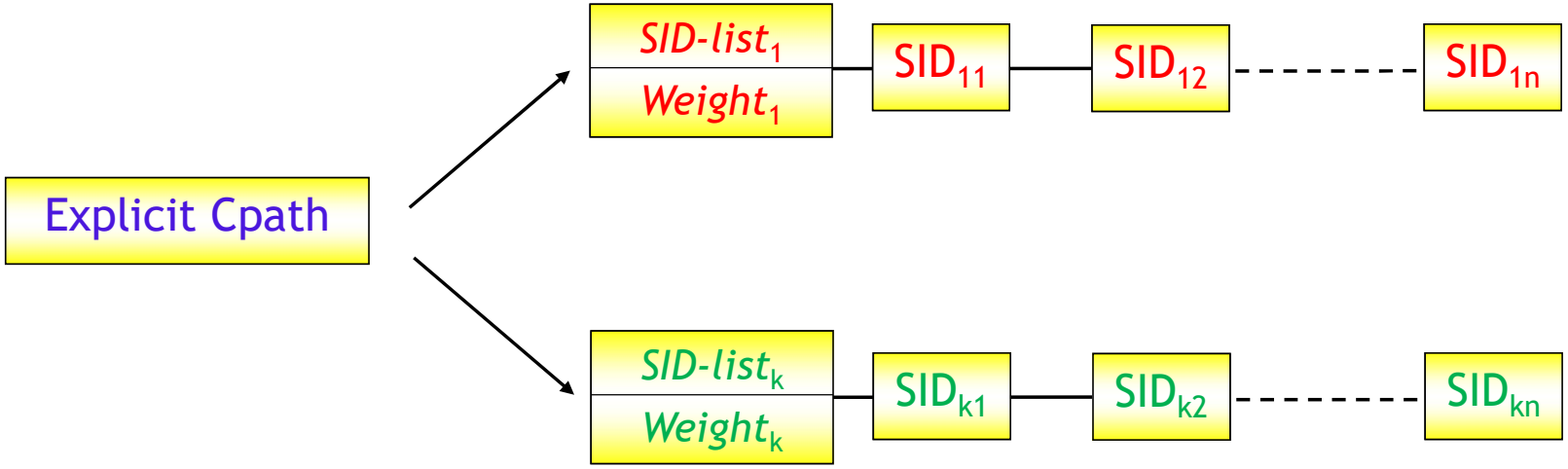
Dynamic Cpaths

- A dynamic Cpath is determined by specifying an objective function (e.g., minimum latency) and a set of constraints (e.g., exclude satellite links from the path)
- The Head-end determines the optimal path and encodes it in a SID list or set of SID lists
- When the head-end does not have sufficient topological information to determine the optimal path, it can delegate the computation to an external controller (**PCE**, **P**ath **C**omputation **E**lement)
  - Whenever there is a topological change in the network, the path is recalculated



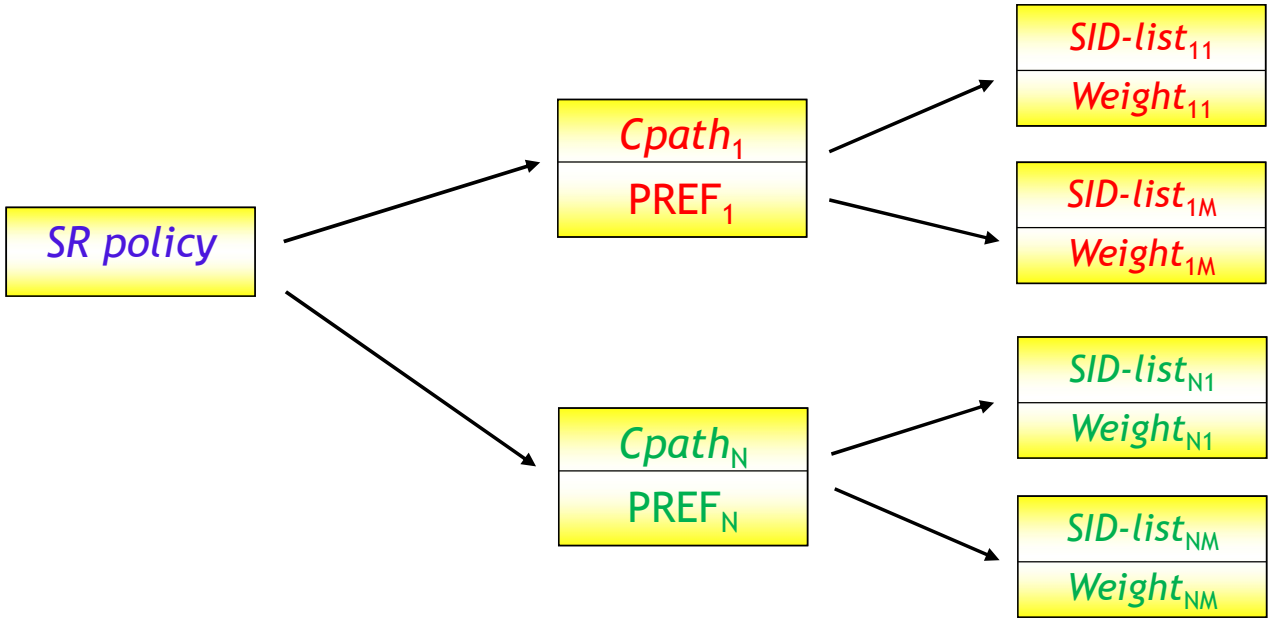
Explicit Cpaths

- An explicit Cpath consists of one or more manually determined SID lists



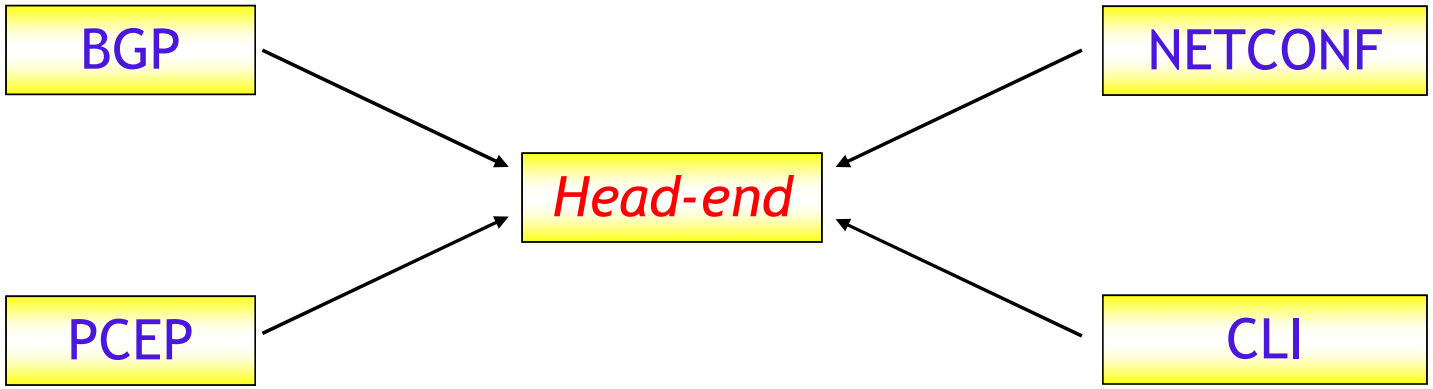
Cpath preference

- Each Cpath has a degree of Preference (PREF)
- The node chooses the Cpath with the highest PREF to insert into the RIB/FIB



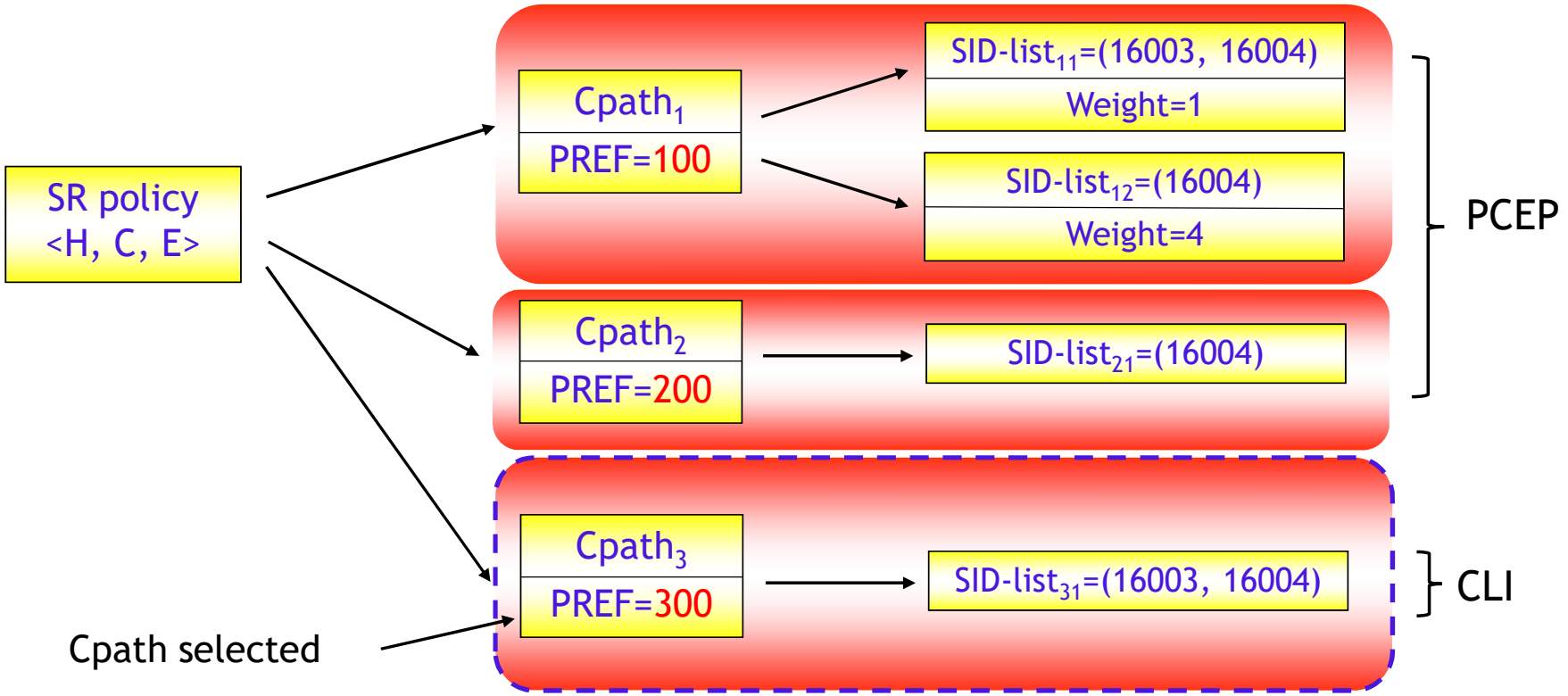
Head-end and *Cpath*

- A Head-end can be informed about the Cpaths of an SR policy (Head-end, Color, End-point) in various ways
  - BGP
  - PCEP
  - NETCONF
  - CLI



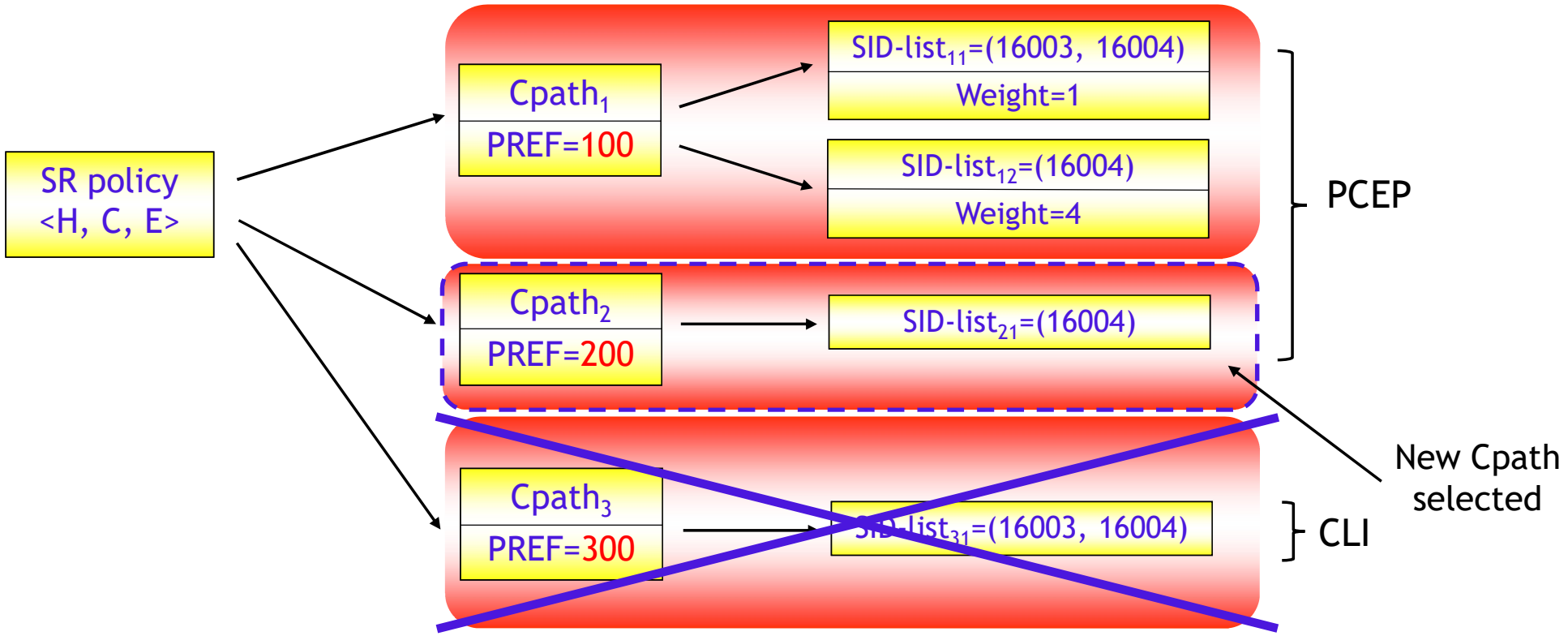
Cpath selection

- The choice is based, among all the valid Cpaths, only on the PREF value
  - The source protocol of a Cpath has no influence on that choice



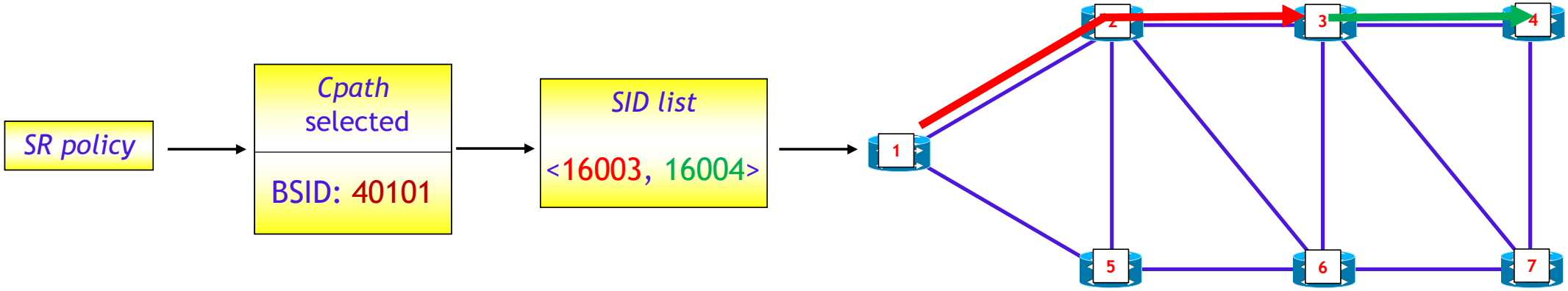
Selecting a new Cpath

- Whenever a Head-end learns a new Cpath or the current one selected is no longer valid, the selection process must be repeated



Active SR policy

- An SR Policy <H, C, E> of a Head-end H is active if it has at least one valid Cpath
- An active SR policy is installed in the FIB indexed by a Binding SID (BSID)

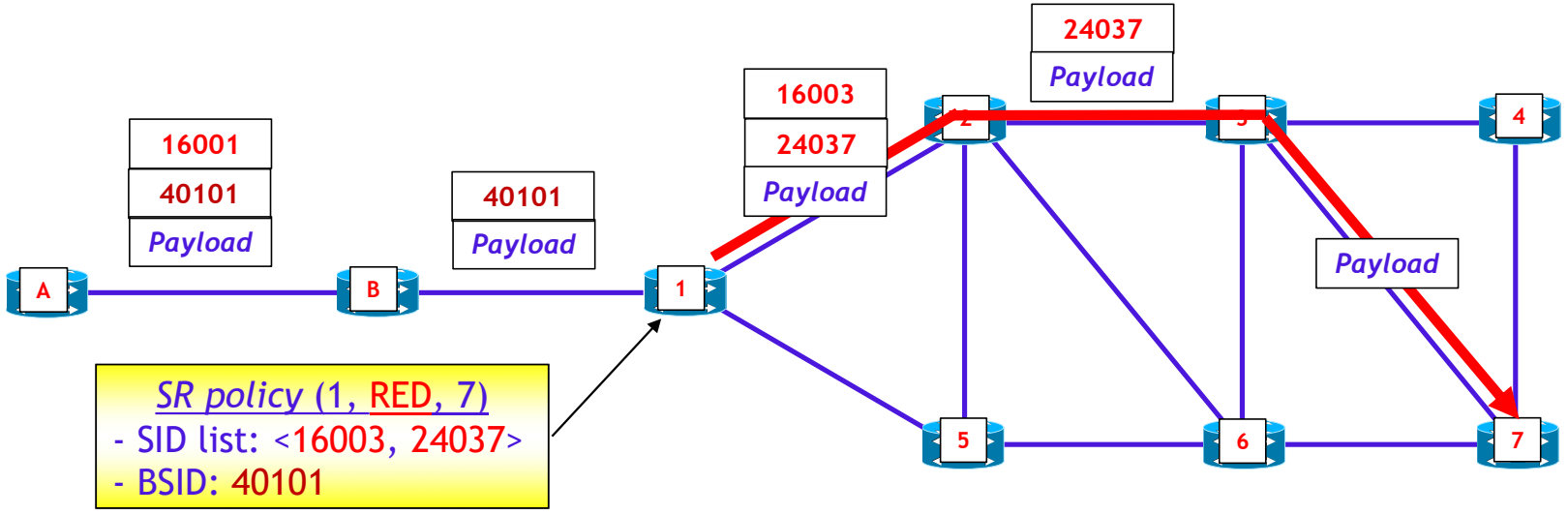


FIB node 1

In	Out	Output Intf	Fraction
40101	<16003, 16004>	→ Node 2	100%

What is a BSID used for?

- A BSID is a specific SID associated with an SR policy
- The function of a BSID in a Head-end is to route traffic using the associated SR policy



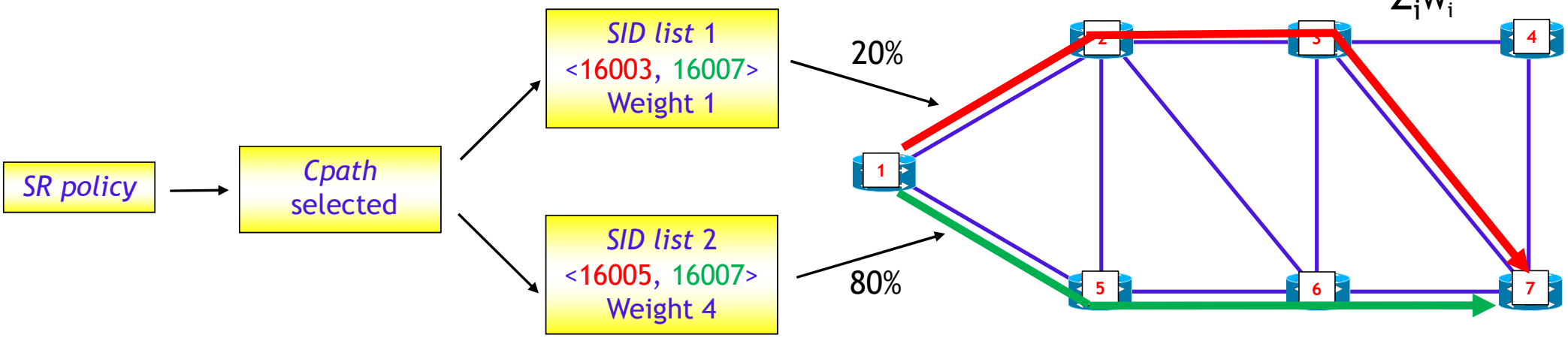
FIB node 1

In	Out	Output Intf	Fraction
40101	<16003, 24037>	→ Node 2	100%

Weighted SR policy

- If an active SR policy uses a Cpath with a weighted set of SID lists, traffic is distributed per flow on the paths defined by the SID lists according to their weight

$$Fraction_k = \frac{W_k}{\sum_i W_i}$$



FIB node 1

In	Out	Output Intf	Fraction
40101	<16003, 16007>	→ Node 2	20%
40101	<16005, 16007>	→ Node 5	80%

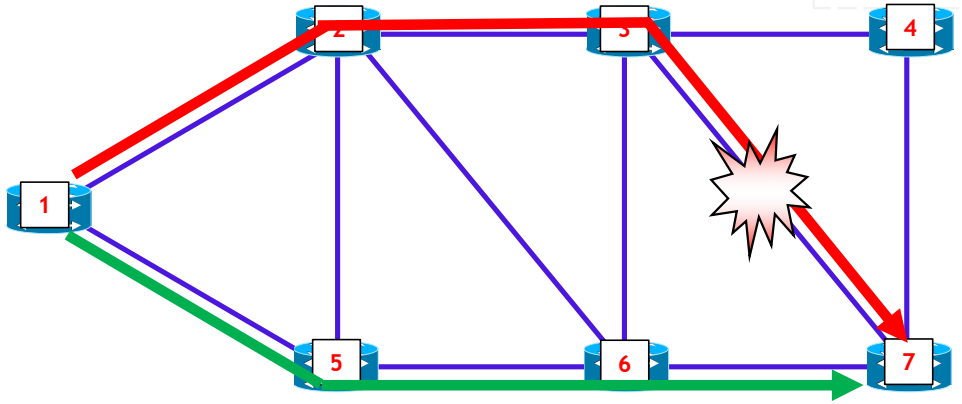
In case of a link or node outage...

- After a link or node goes out of service:
  1. Initially, traffic is protected by TI-LFA (if enabled)
  2. The Head-end learns about the outage via IGP flooding and invalidates all Cpaths containing the out of service link or node
  3. The head-end runs the selection process again and chooses the one with the highest PREF among all remaining Cpaths as the new Cpath to route
  4. The Head-end installs the new Cpath in the FIB and then begins forwarding traffic using the SR policy over the new path

- The process is reversible: when the link or node returns to service, the Head-end runs the selection process again and chooses the Cpath with the highest PREF, i.e., the one it was using before the outage

An example

SR policy (1, RED, 7)
Cpaths:
- PREF 200: <16002, 24023, 16007>
- PREF 100: <16005, 16007>



- FIB of node 1 before link 5→7 failure

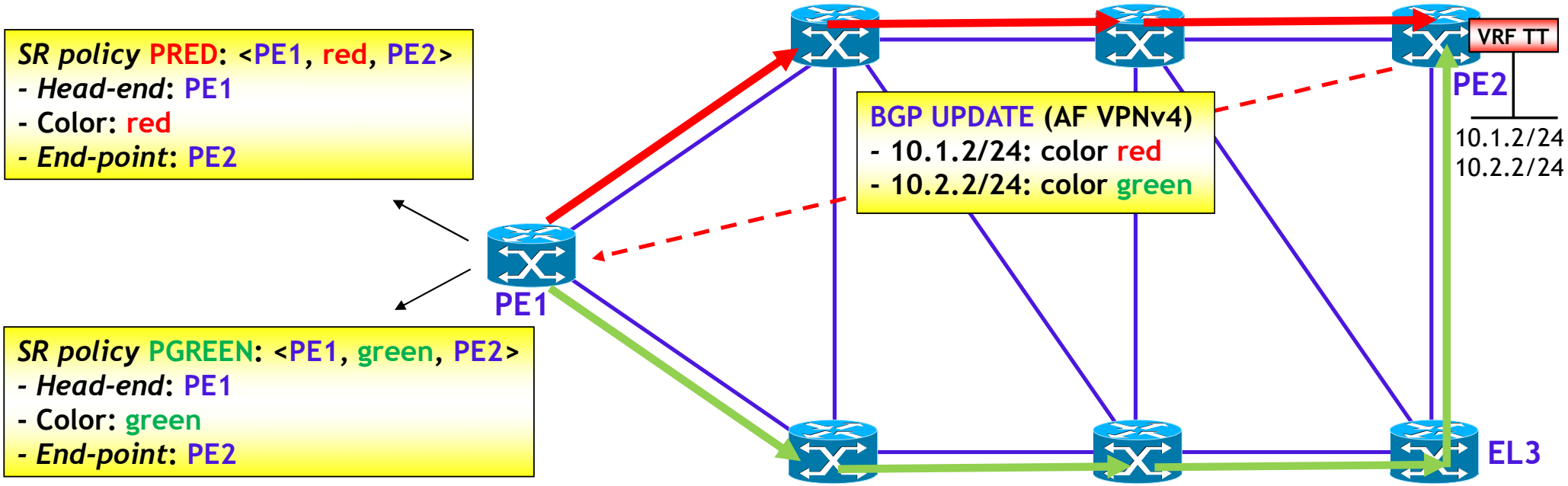
In	Out	Output Intf	Fraction
40101	<16002, 24023, 16007>	→ Node 2	100%

- FIB of node 1 after link 5→7 failure

In	Out	Output Intf	Fraction
40101	<16005, 16007>	→ Node 5	100%

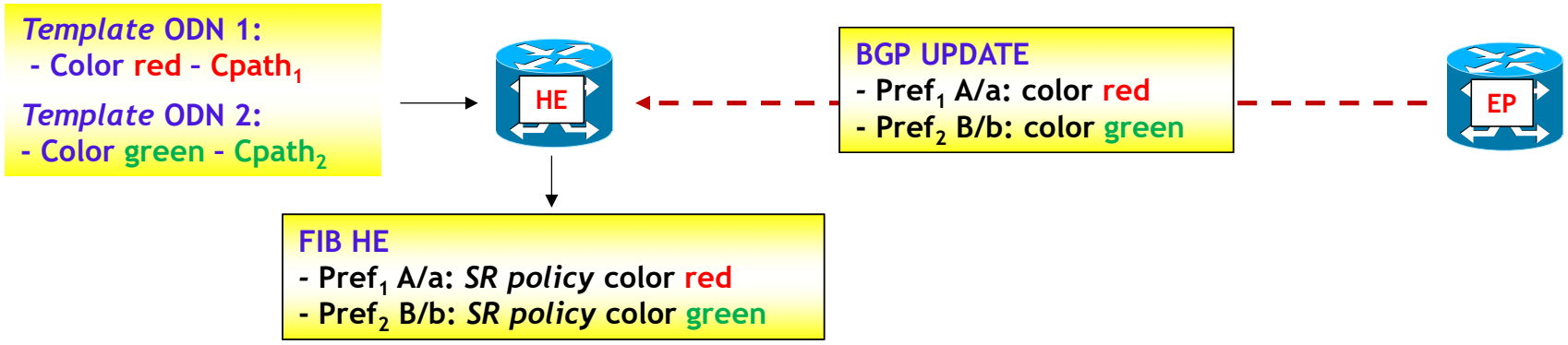
Automated steering

- **Problem**: How do you instruct a Head-end to route traffic using a specific SR policy?
- **Solution**: Use the color associated with the SR policy
  - The color is transported through a BGP Color Extended Community



On Demand Next-hop (ODN)

- The ODN functionality automates the creation of a dynamic SR policy on a Head-end, upon request from an endpoint
- How it works
  - Templates are defined on the Head-end that define specific paths (explicit or dynamic)
  - Each template is identified by a color that specifies how the dynamic path should be determined
  - Upon request from an endpoint via a BGP advertisement containing a specific BGP Color Extended Community value, if a template identified by the color value exists on the Head-end, an SR policy is created and installed in the FIB



Module 4: *Segment Routing* over IPv6 (SRv6)

#1	Main aspects
#2	SID coding: SRH and micro-SID (uSID)
#3	SID Processing at End Nodes
#4	SR-MPLS⟶SRv6: migration plan

Copyright Tiziano Tofoni

Basic operation

- SRv6 is an implementation of the Segment Routing paradigm that uses IPv6 addresses or parts thereof (micro-SIDs) as SIDs
  - SID → IPv6 address or part thereof (micro-SID)
  - *Segment List* → Set of IPv6 addresses or a single IPv6 address with the (micro-)SIDs encoded within it
  - As with SR-MPLS, SIDs are distributed through appropriate extensions to the OSPF(v3) and IS-IS protocols
  - IMPORTANT NOTE: SRv6 (like SR-MPLS) has no impact on L2/L3 MPLS VPN services!

- SRv6 packets, like MPLS packets, can carry any type of payload
  - Layer 3: IPv4, IPv6
  - Layer 2: Ethernet *tagged/untagged* (other types of Layer 2 are possible but not of practical interest)

SRv6 vs SR-MPLS (1/2)

- Header overhead
  - SRv6 uses IPv6 addresses as SIDs, so an SRv6 SID is four times longer than an MPLS label (32 bits)
  - SR-MPLS uses MPLS labels (20+12 bits), resulting in less packet header overhead
- Hardware requirements
  - SRv6, being a newer technology based on IPv6, requires specific hardware support to efficiently handle the Segment Routing (SRH) header, or, if using micro-SIDs, specific hardware for the shift-and-forward mechanism
  - SR-MPLS uses the existing MPLS data plane, meaning it can be deployed on existing MPLS hardware with simple software upgrades
- Transition Complexity
  - SRv6: Full adoption depends on widespread IPv6 deployment in the network infrastructure
  - SR-MPLS: Enables gradual migration into existing MPLS networks without requiring changes to the underlying IPv4+MPLS infrastructure

SRv6 vs SR-MPLS (2/2)

- Security and Fragmentation/DoS Attacks
  - The introduction of the Segment Routing Header (SRH) in SRv6 adds a new Extension Header to the IPv6 packet, which may have security implications not present in SR-MPLS
  - SR-MPLS: does not introduce any additional IPv4 or IPv6 headers, but only MPLS labels, reducing the attack surface
- NOC and Network Engineering Skills
  - SRv6: requires a good knowledge of IPv6 and IPv6 routing protocols such as IS-IS and/or OSPFv3
  - SR-MPLS: since this reuses an existing and well-known data plane, training for both the NOC and Network Engineering staff may be minimal or nonexistent
- In summary
  - SRv6 offers greater flexibility and better native integration with IPv6 for new deployments (greenfield)
  - SR-MPLS is often more practical and cost-effective for existing networks (brownfield) seeking to benefit from SR with less impact on hardware and a simpler transition

Module 4: *Segment Routing* over IPv6 (SRv6)

#1 Main aspects

#2 SID coding: SRH and micro-SID (uSID)

#3 SID Processing at End Nodes

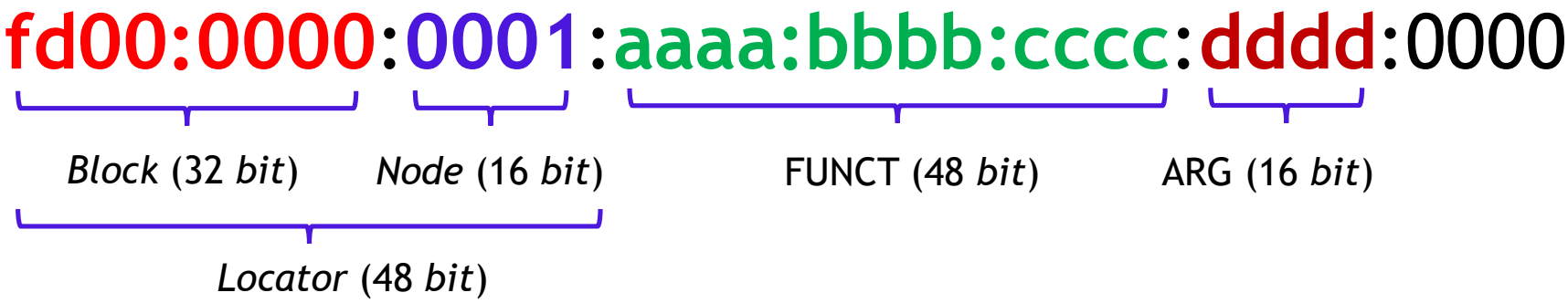
#4 SR-MPLS⟶SRv6: migration plan

Copyright Tiziano Tofoni

Generic format of an SRv6 SID

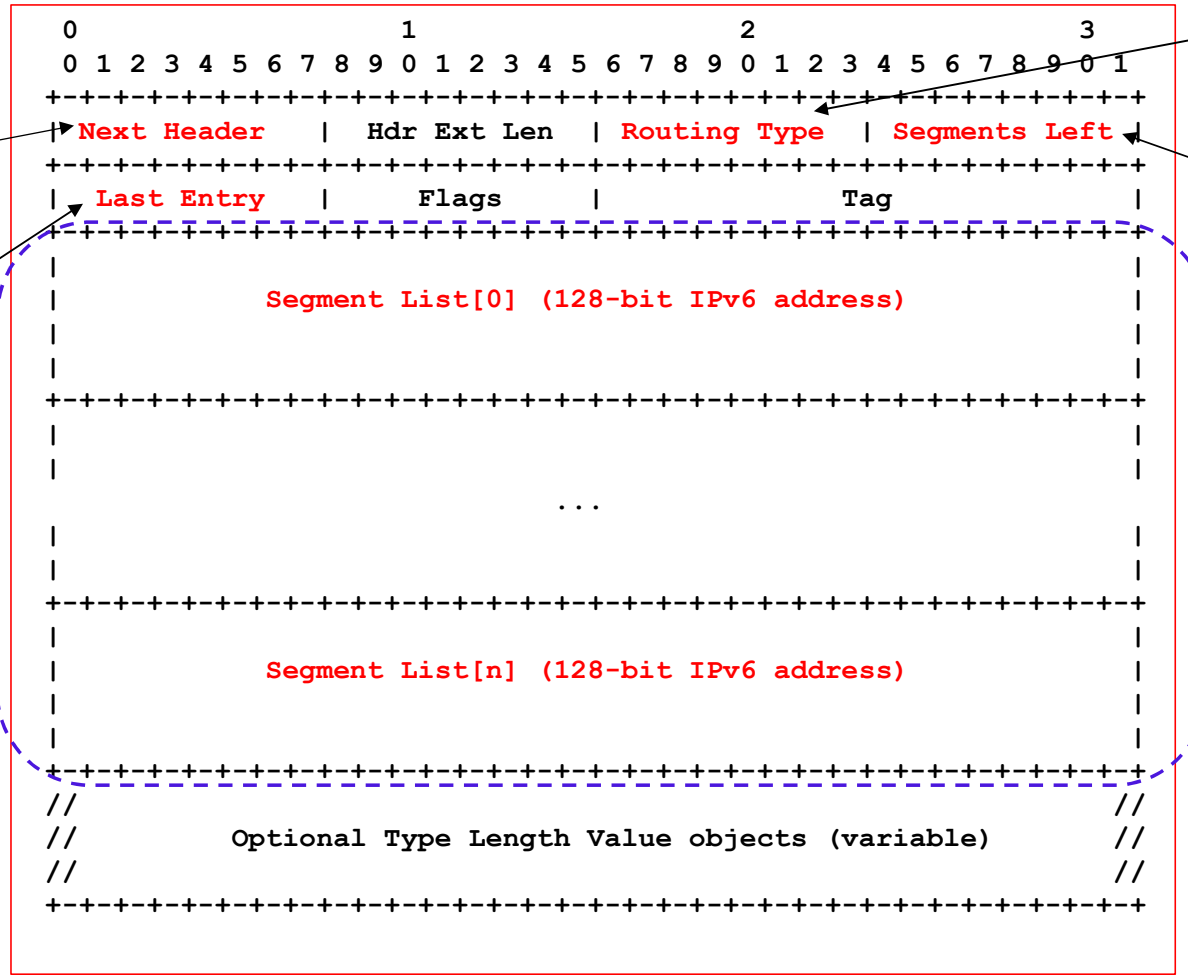
- An SRv6 SID is an IPv6 address that has the generic structure: **LOC**:**FUNCT**:**ARG** (RFC 8986)
  - **LOC** (*Locator*): These are the **L** most significant bits of the SID
  - **FUNCT** (Function): These are the **F** bits following the LOC – representing the local behavior associated with the SID
  - **ARG**: These are the **A** bits following the FUNCT – used in those special cases where further processing of the local behavior associated with the SID is required
  - NOTE: **L**+**F**+**A**≤128, if **L**+**F**+**A**<128 the **ARG** portion is padded with sufficient zeros to reach **L**+**F**+**A**=128
- The *Locator* can be represented as **B**:**N**
  - **B** (SRv6 SID *Block*): It is an IPv6 prefix allocated by the ISP for all the SIDs of the various nodes
  - **N** (*Node*): It is an identifier of the node to which the SID is associated
- Best practice: use IPv6 Unique Local addresses (fc00::/7) or the prefix 5f00::/16 reserved by IANA for SRv6 to define the SIDs

Generic format of an SRv6 SID: example



- L=48 *bit*
- F=48 *bit*
- A=16 *bit*
- L+F+A=112 *bit* (→ last 16 *bit* null)

Segment Routing Header (SRH) (RFC 8754)



Routing Type=4→SRH

4: IPv4
6: TCP
17:UDP
41: IPv6
. . .

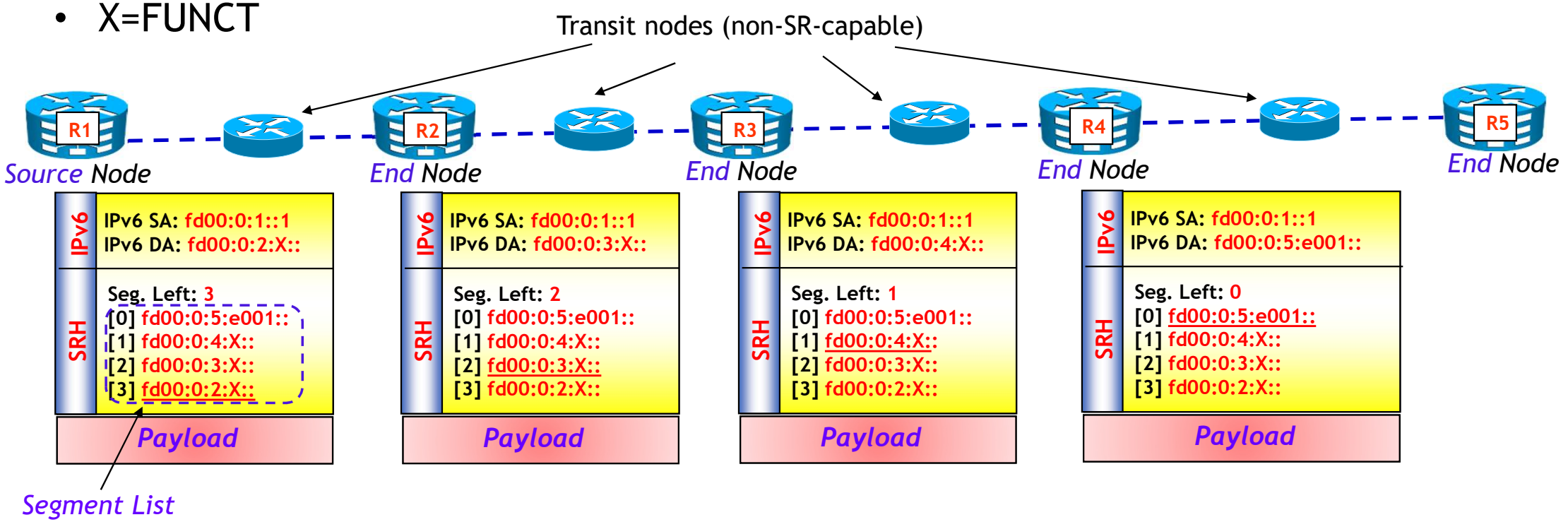
Number of SIDs left in the *Segment List*

Index (*zero-based*) of the last element of the list (coincides with the first SID)

Segment List (in reverse order)

Example of using SRH

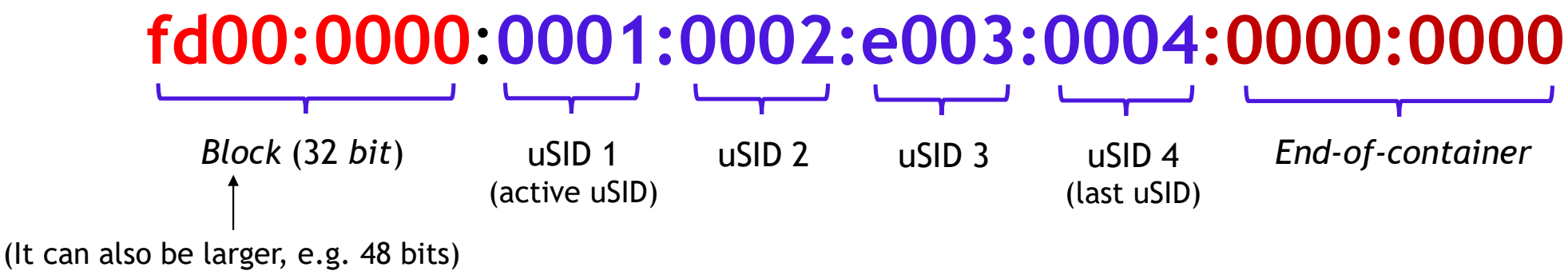
- *Block*=fd00::/32
- *Locator* node Rn=fd00:0:n::/48 (n=1, ..., 5)
- VPN SID (node R5)=fd00:0:5:e001:: (similar to an MPLS service label)
- X=FUNCT



Compressed *Segment List*: micro-SID

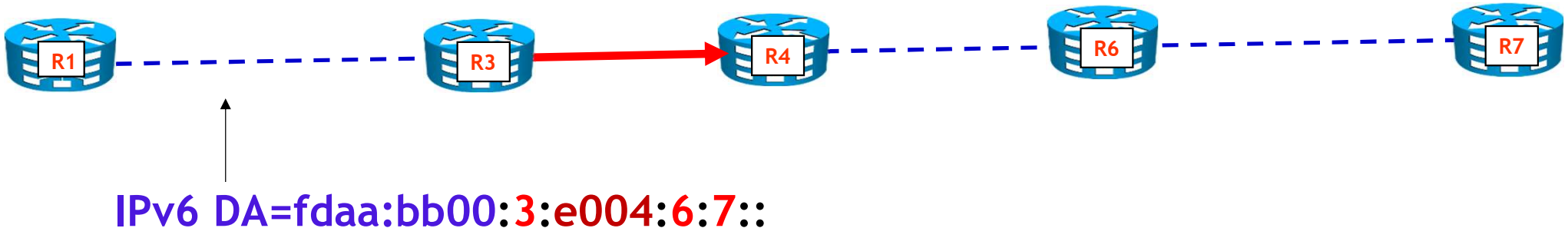
- In SRv6, each instruction (carried in the destination IPv6 address or in a SID within an SRH) can contain up to 6 micro-instructions (micro-SIDs, abbreviated as uSID)
  - Each uSID is 16 bits long
  - A 128-bit SRv6 instruction (SID) containing multiple micro-SIDs is called a uSID container
  - Each uSID container can contain up to six uSIDs
    - If it contains fewer than six uSIDs, each unused micro-instruction is set to 0x0 (End-of-container).

- Example (F3216 format)



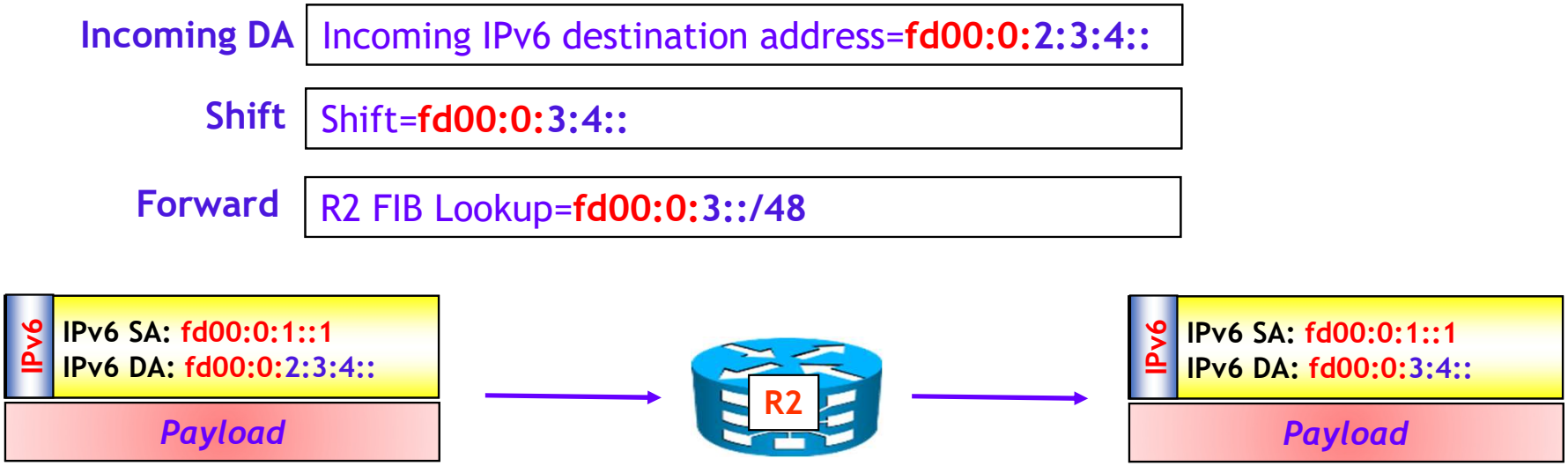
Global and local uSID

- A uSID can indicate a node or an IGP adjacency
  - A node is identified by a global uSID
  - An IGP adjacency is identified by a local uSID
- Example: definition of a path with source node R1, destination node R7, which passes through the segments R1-R3, R3→R4, R6-R7
  - Assumptions: SID block=fdaa:bb00::/32, uSID node Rn=0xn, uSID adjacency R3→R4=0xe004



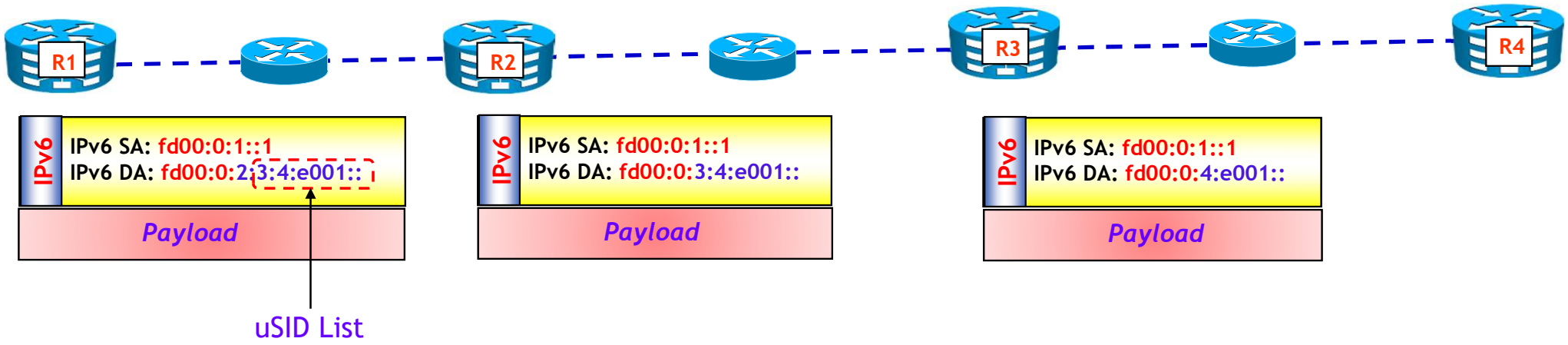
uSID processing

- The key packet processing operation for uSID at each SR Endpoint is called shift-and-forward
  - Designed for hardware efficiency



Example of using uSIDs

- *Block*=fd00::/32
- uSID node Rn=0xn (n=1, ..., 4)
- VPN uSID (node R4)=e001 (similar to an MPLS *service label*)



Module 4: *Segment Routing* over IPv6 (SRv6)

#1 Main aspects

#2 SID coding: SRH and micro-SID (uSID)

#3 SID Processing at End Nodes

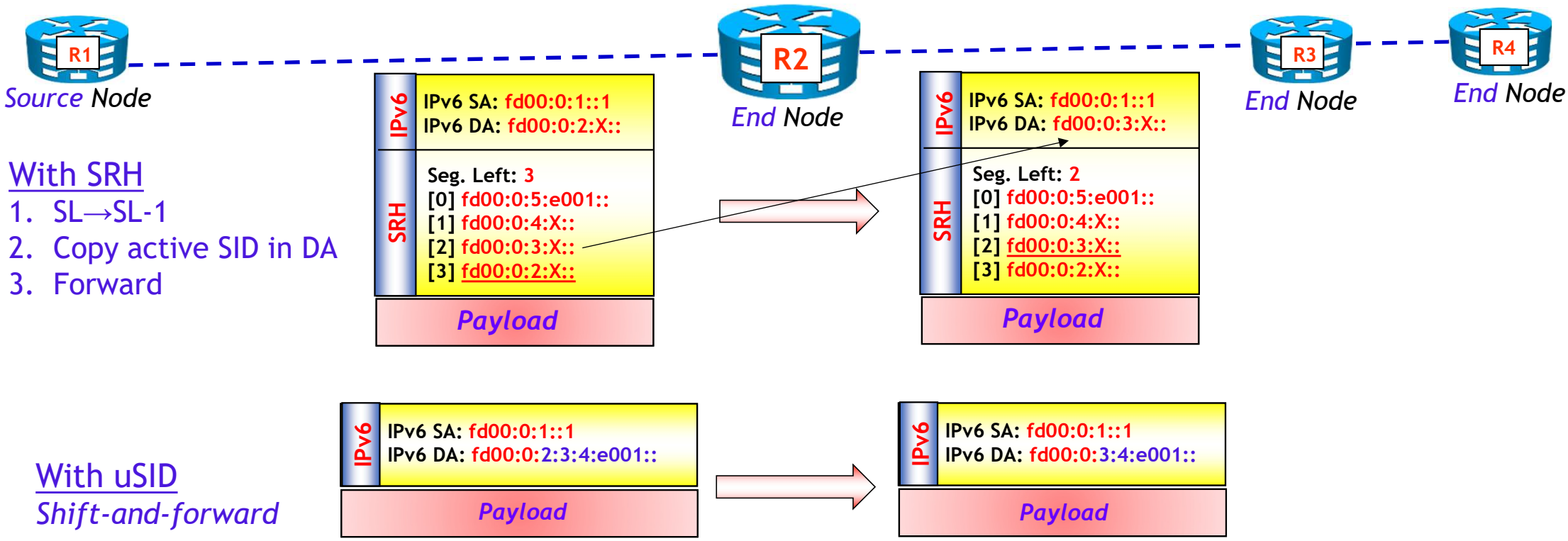
#4 SR-MPLS⟶SRv6: migration plan

Copyright Tiziano Tofoni


Processing SRv6 packets at End Nodes

- SRv6 provides many processing possibilities in End Nodes
  - End (uN)
    - with PSP
    - with USD
    - with PSP & USD
    - . . .
  - End.X (uA)
    - with PSP
    - with USD
    - with PSP & USD
    - . . .
  - End.DX4/End.DX6/End.DX2 (uDX4/uDX6/uDX2)
    - Endpoint with Decapsulation and and Xconnect
  - End.DT4/End.DT6/End.DT46 (uDT4/uDT6/uDT46)
    - Endpoint with Decapsulation and Table lookup
  - And many others... (see RFC 8986)

Default processing (End/uN)



With SRH

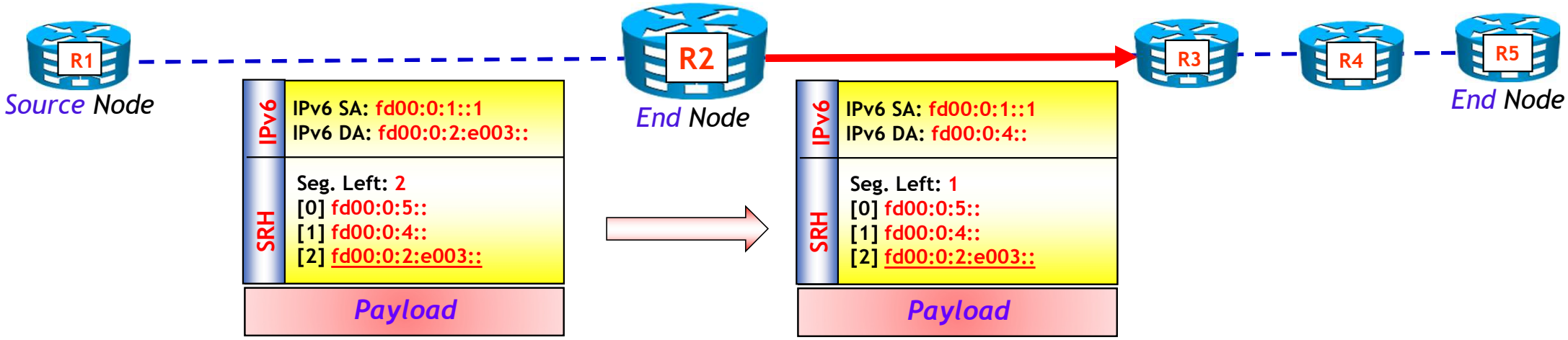
1. SL→SL-1
2. Copy active SID in DA
3. Forward

With uSID

Shift-and-forward

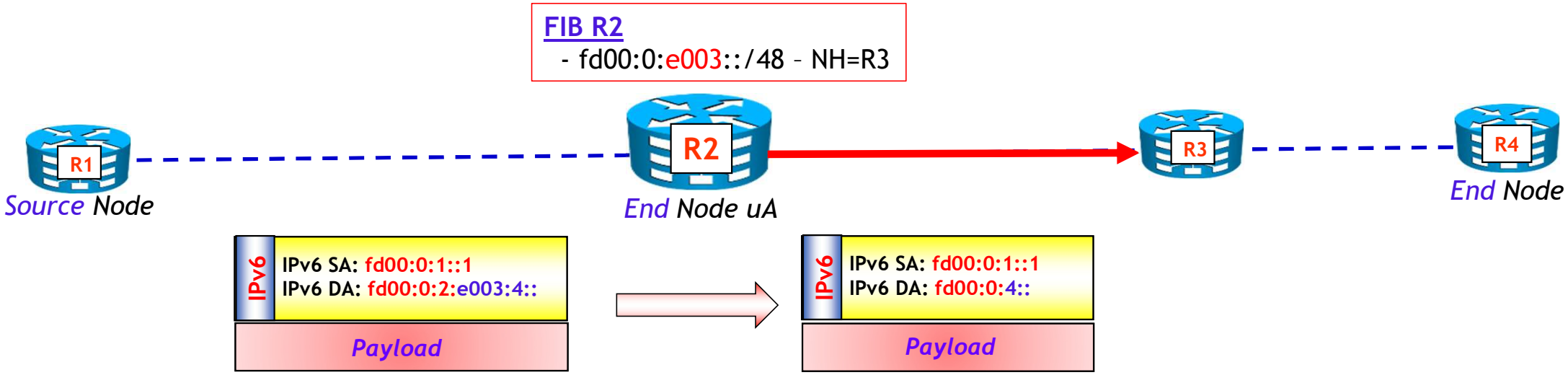
End.X

- Allows you to "force" traffic to a specific IGP adjacency (IGP-adjacency segment)



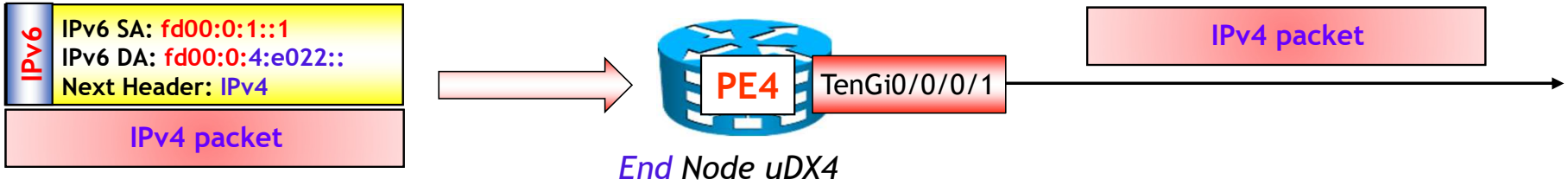
End.X with uSID (uA)

- Same as End.X (but with uSID)



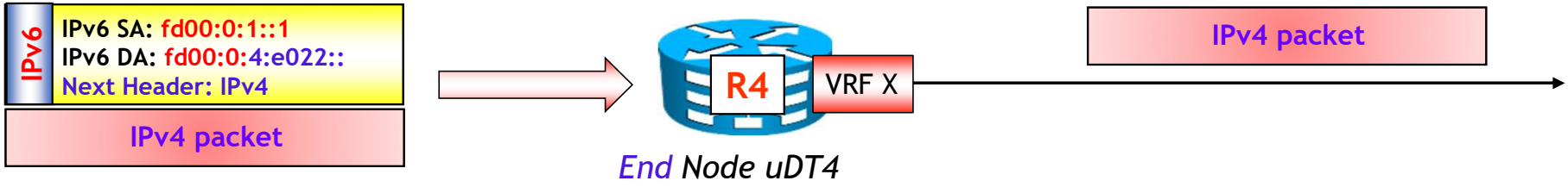
End.DX4/End.DX6/End.DX2 (uDX4/uDX6/uDX2)

- Allows you to decapsulate an SRv6 packet, obtaining the transported payload (IPv4/IPv6 packet or an L2 frame) and send it to a given interface outside the SRv6 domain
  - Same operation as per-CE label allocation in L3VPN MPLS services
  - **DX**=**D**ecapsulation and **X**connect
- NOTE: must necessarily be the last function of the SID/uSID list



End.DT4/End.DT6 (uDT4/uDT6)

- Allows you to decapsulate an IPv4 or IPv6 packet and perform a lookup on an IPv4 or IPv6 FIB
  - Same operation as per-VRF label allocation in L3VPN MPLS services
  - **DT**=**D***ecapsulation and* ***T****able lookup*
- NOTE: must necessarily be the last function of the SID/uSID list



Module 4: *Segment Routing* over IPv6 (SRv6)

#1 Main aspects

#2 SID coding: SRH and micro-SID (uSID)

#3 SID Processing at End Nodes

#4 SR-MPLS⟶SRv6: migration plan

Copyright Tiziano Tofoni

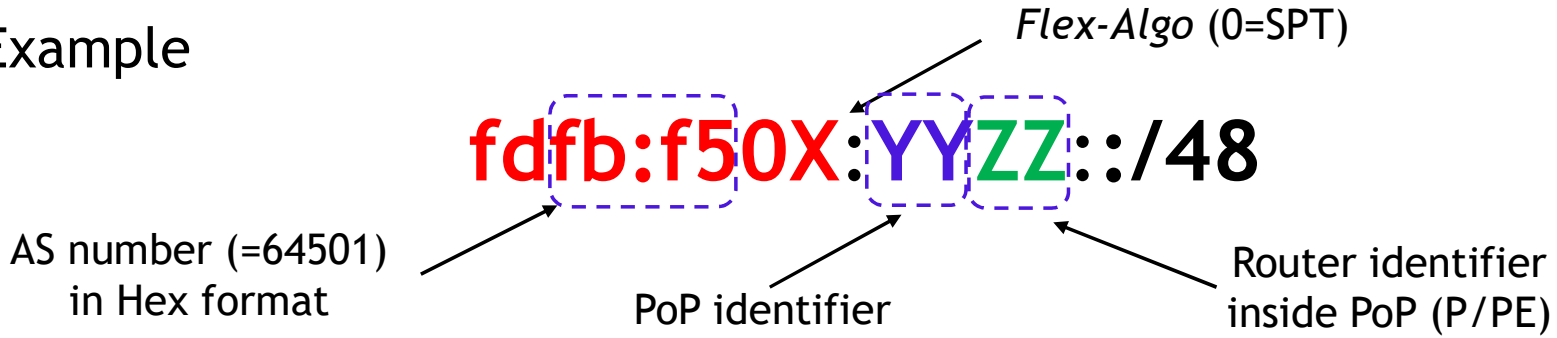
Migration plan

- **Step 0**: Network *assessment* with respect to SRv6
- **Step 1**: Definition of an IPv6 numbering plan and deployment of IPv6 IGP routing
  - IGP Protocol: Choose the one that is most supported by your vendor and most familiar to you
- **Step 2**: Definition of the Locators numbering plan
- **Step 3**: Service migration (L3VPN e/o L2VPN)
  - Possibly starting from a subset of sites
- **Step 4**: Disable LDP e/o SR-MPLS
- **Step 5**: (optional): Enable TI-LFA
- **IMPORTANT NOTE**: **The procedure can be performed on a production network without any traffic loss**

Definition of an IPv6 and SRv6 numbering plan

REISS ROMOLI

- To simplify the IPv6 numbering plan use:
  - For physical interfaces IPv6 Link-Local addresses (auto-generated by the router)
  - For Loopback interfaces IPv6 Unique-Local addresses (**fc00::/7**)
    - Alternative: use the prefix **5f00::/16** reserved by IANA for SRv6
- Locators numbering plan
  - Select the SRv6 SID Block and Format (es. F3216)
  - Choose the length of the Locators (*best practice* /48, to leave room for 6 uSID)
  - Define the structure (i.e., the meaning of the nibbles)
- Example



Choice of the IGP protocol

REISS ROMOLI

- IS-IS or OSPFv3 – no other choices available
- IS-IS vs OSPFv3

Feature	IS-IS	OSPFv3
Scope of use	Mainly ISP networks	ISP and enterprise networks
Messages transport	Over Layer 2	Over IPv6 (*Next Header*=89)
Hierarchy	Flexible (Level 1 and Level 2)	Rigid: area 0 mandatory
Scalability	Excellent, greater flexibility	Excellent, but more rigid
Implementation	Less widespread (more niche)	Widespread
Extensibility	High, thanks to the widespread use of TLV modules	Less simple (often requires new RFCs)

- Selection criteria: choose the one most supported by the vendor you use and, if support is equal, choose the one you are familiar with
- NOTE: Regarding SRv6 support, the most widely support by various vendors is for the IS-IS protocol

L3VPN services migration

REISS ROMOLI

- For L3VPN services, whether based on an MPLS data plane (with LDP and/or SR) or SRv6, the control plane is the same (apart some minor differences on service SID/label encoding)
  - Based on the BGP VPNv4-unicast address family (AFI/SAFI=1/128) for IPv4 L3VPN or BGP VPNv6-unicast address family (AFISAFI=2/128) for IPv6 L3VPN
- Initial assumptions
  - All PE routers are dual-connected, meaning they support the implementation of L3VPN services with both MPLS and SRv6 data planes
  - Active L3VPN services have been implemented on an MPLS data plane that uses only SR-MPLS
  - Separate Route Reflectors for VPNv4-unicast advertisements with BGP Next-Hop IPv4 and for VPNv4-unicast announcements with BGP Next-Hop IPv6
- *Best practice*: Start the migration with two CEs of a single L3VPN, verify that migration works and then extend gradually migration to all other CEs of all other active L3VPNs

Point-to-Point L2VPN services migration

- **Problem:** An Attachment Circuit (AC) of a CE on a PE cannot be assigned to multiple pseudowires
- **Solution:** Allow an AC to participate in two different pseudowires
  - In the case of EVPN-VPWS, Cisco IOS XR provides the **vpws-seamless-integration** command within the pseudowire configuration
- Migration plan
  1. Initial assumption: There is a pseudowire created via EVPN-VPWS on the SR-MPLS data plane
  2. Activate a second pseudowire on the SRv6 data plane using the **vpws-seamless-integration** command
  3. Verify that all pseudowires are in the up state
  4. Delete the pseudowire with the SR-MPLS data plane

Copyright Tiziano Tofoni

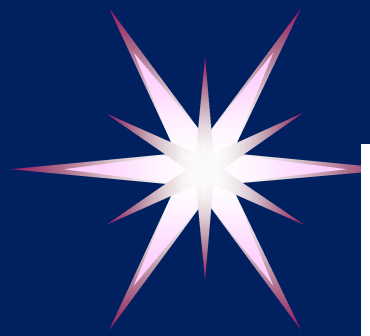

The end …

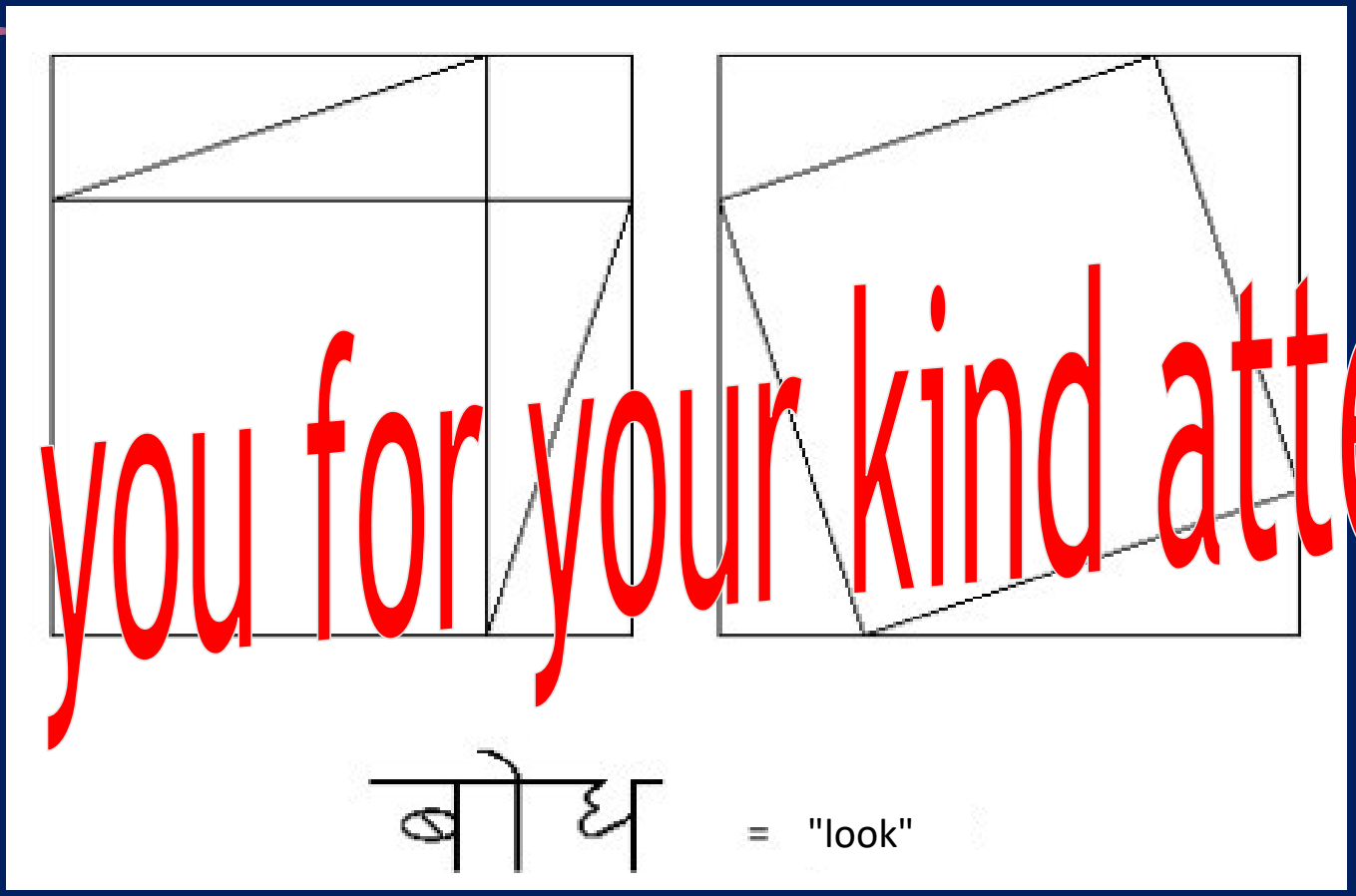
Whenever you're evaluating new technologies or architectures, try to figure out what business (not technology) problem you're really trying to solve, and whether the new shiny thing solves it or introduces another distracting layer of abstraction.

Source: blog.ipspace.net

Copyright Tiziano Tofoni


Last page (finally...)





Thank you for your kind attention

बोध = "look"

Special thanks to Nicola Modena and Ivan Pepelnjak for their valuable suggestions.

Copyright Tiziano Tofoni