

# Core network protocols and architectures

---

Multi-Protocol Label Switching

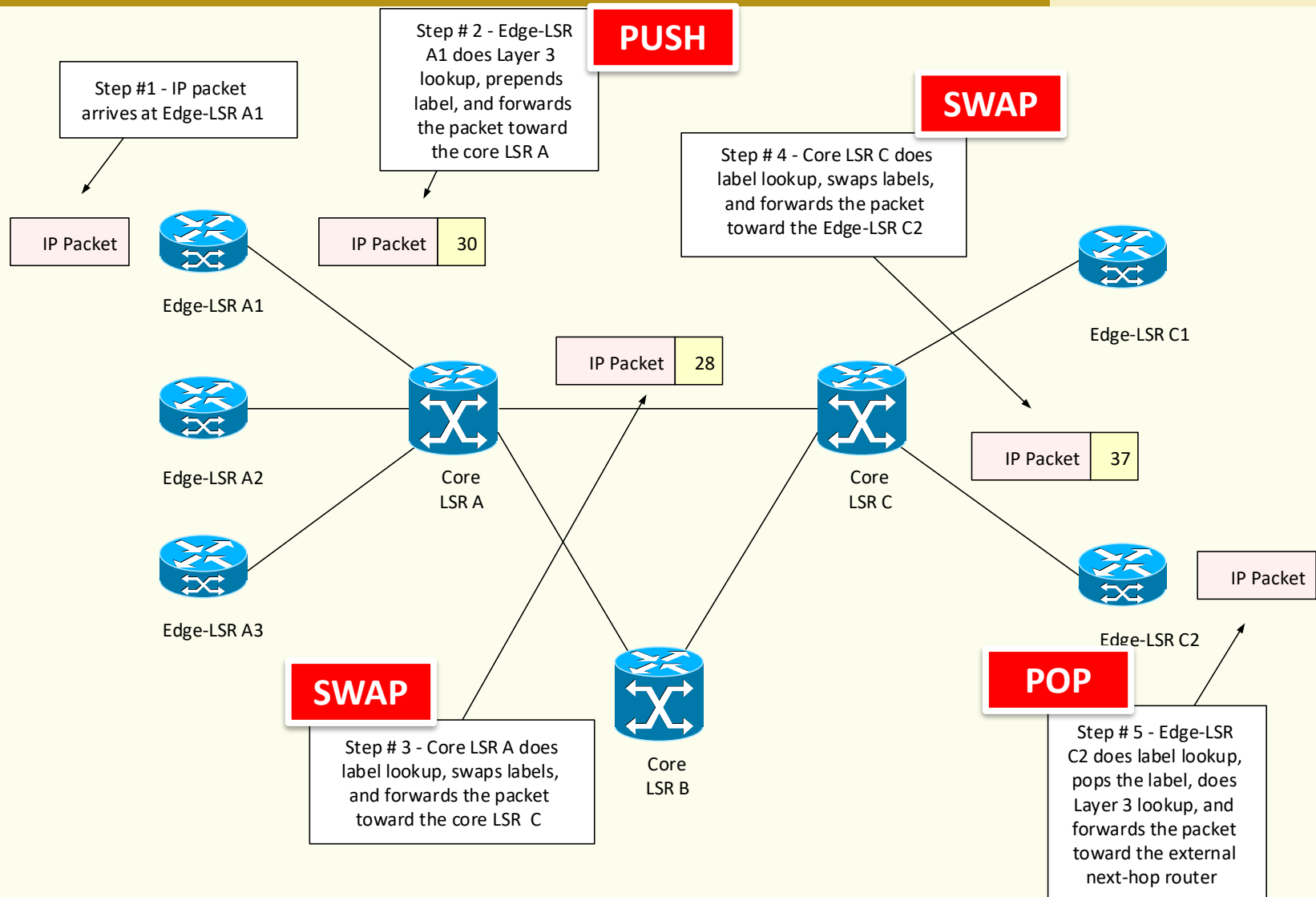
Data plane

Enzo Mingozzi

Professor @ University of Pisa

[enzo.mingozzi@unipi.it](mailto:enzo.mingozzi@unipi.it)

# MPLS data plane



- ```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+ Label
|                                     | TC   | S |                                   TTL       | Stack
+---+ Entry

Label:    Label Value, 20 bits
TC:       Traffic Class field, 3 bits
S:        Bottom of Stack, 1 bit
TTL:      Time to Live, 8 bits

```

# MPLS label encoding

- Ethernet

|            |          |           |
|------------|----------|-----------|
| Eth header | Label SE | L3 header |
|------------|----------|-----------|

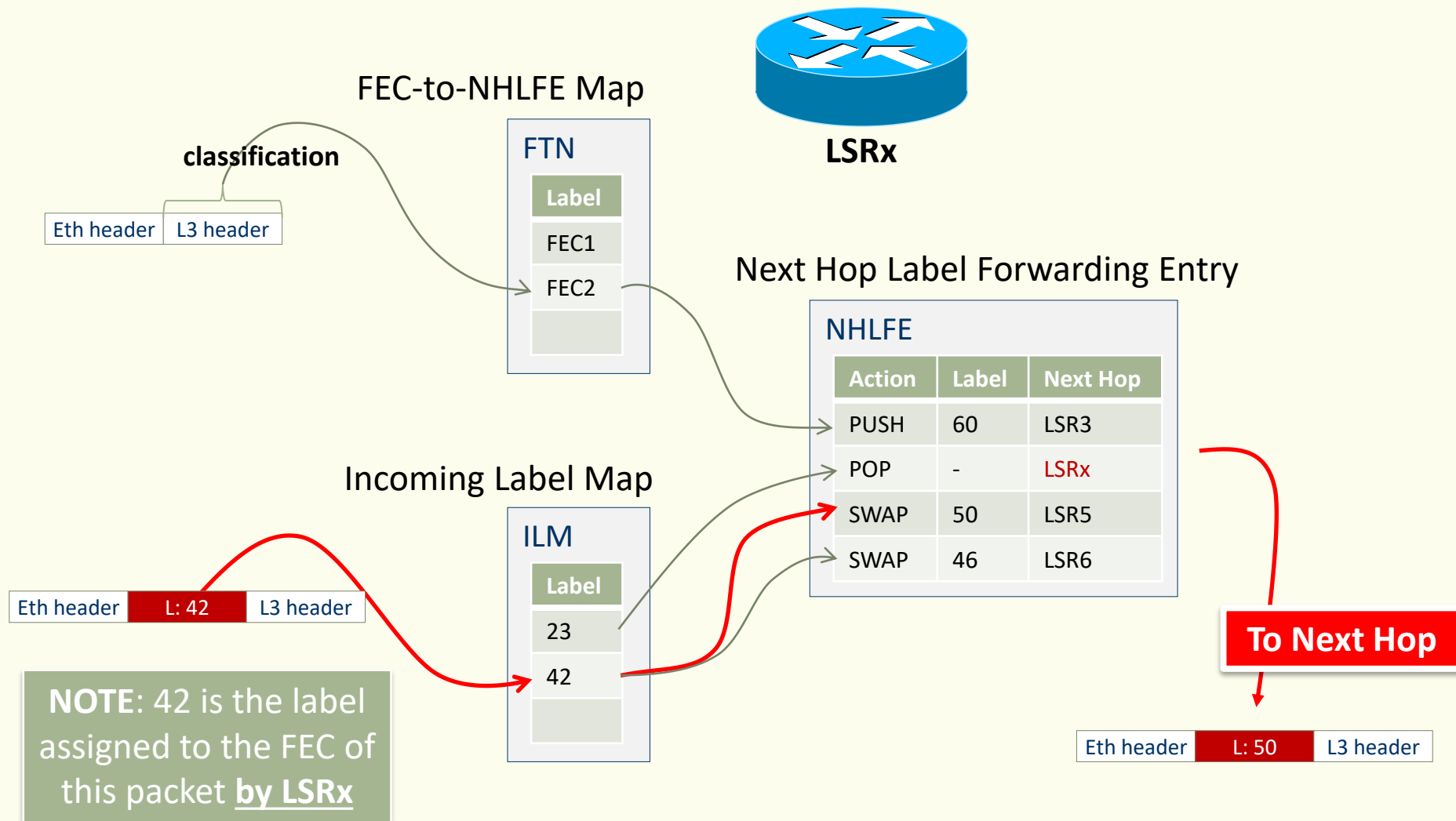
- Ethertype value 0x8847: MPLS unicast packet
- Ethertype value 0x8848: MPLS multicast packet

- PPP

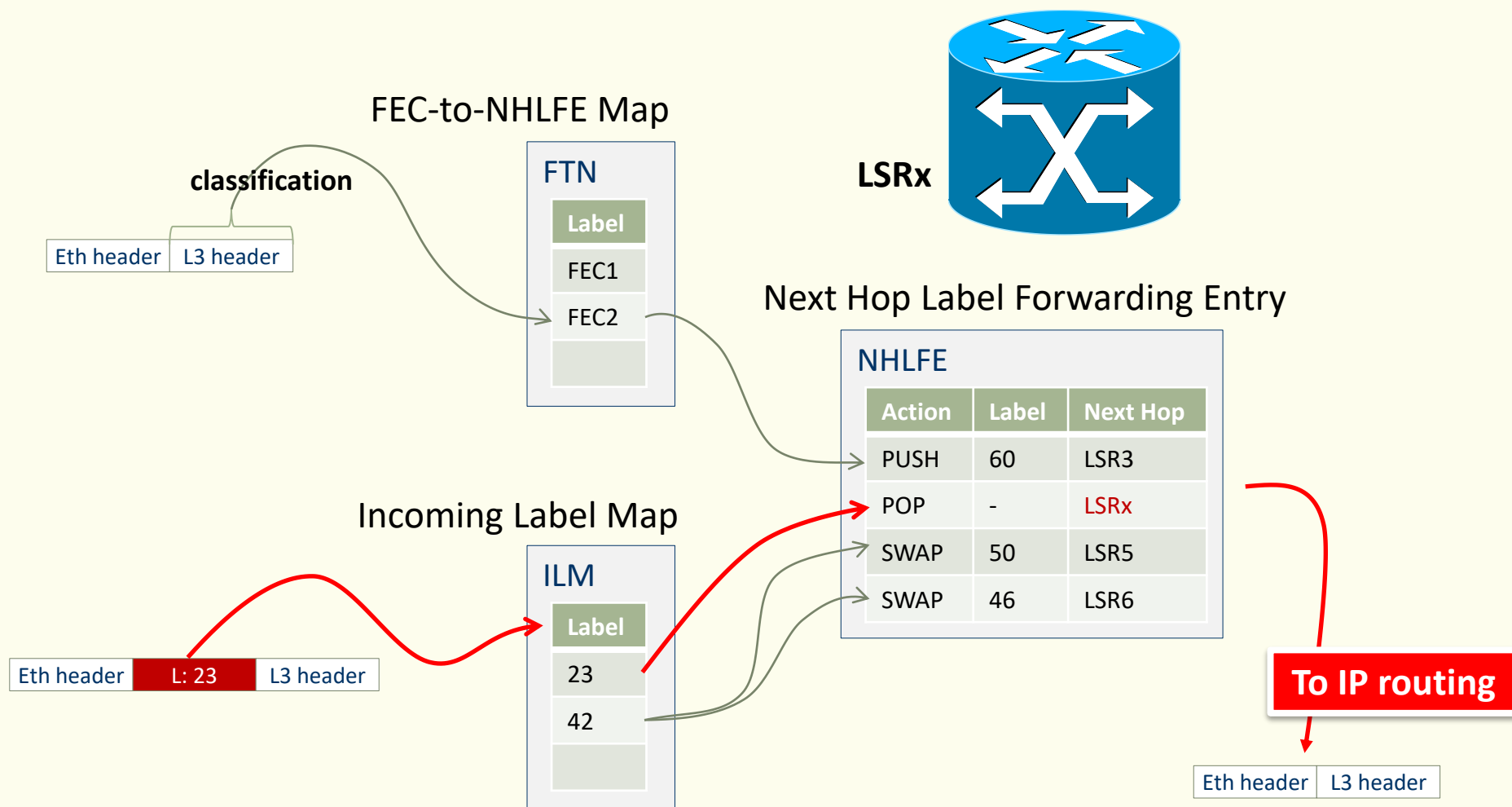
|            |          |           |
|------------|----------|-----------|
| PPP header | Label SE | L3 header |
|------------|----------|-----------|

- MPLS Control Protocol (MPLSCP) defined for initial negotiation
  - PPP protocol field: 0x8281
- L3 protocol identity is implicit (must be inferable from the label value)

# MPLS forwarding operation

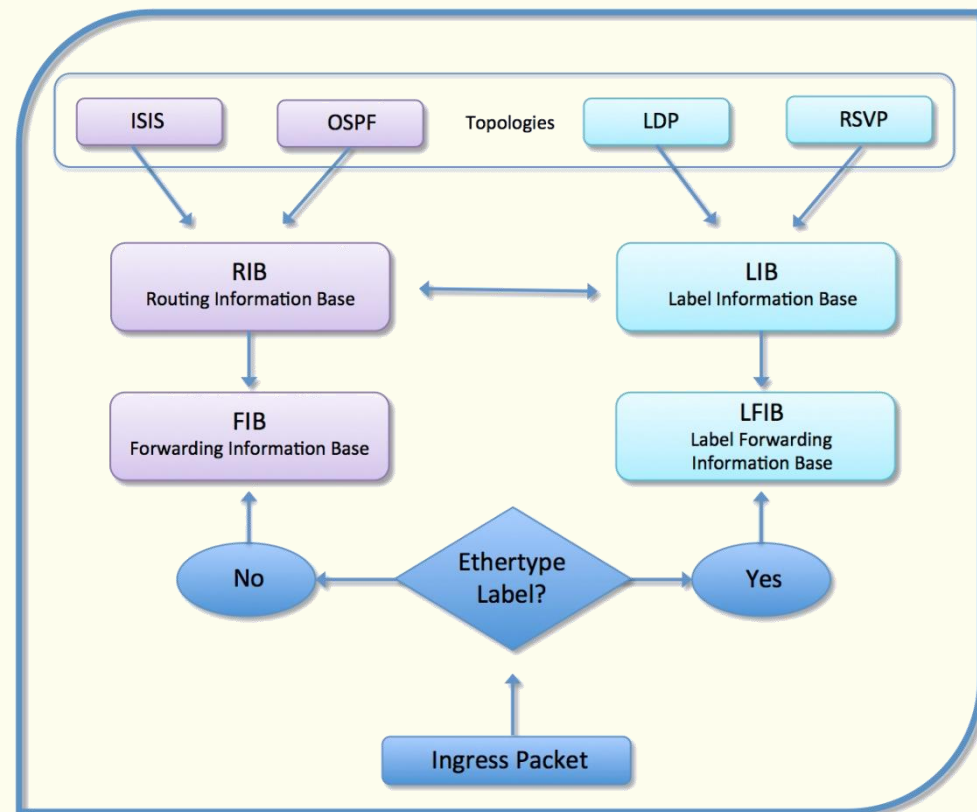


# MPLS forwarding operation

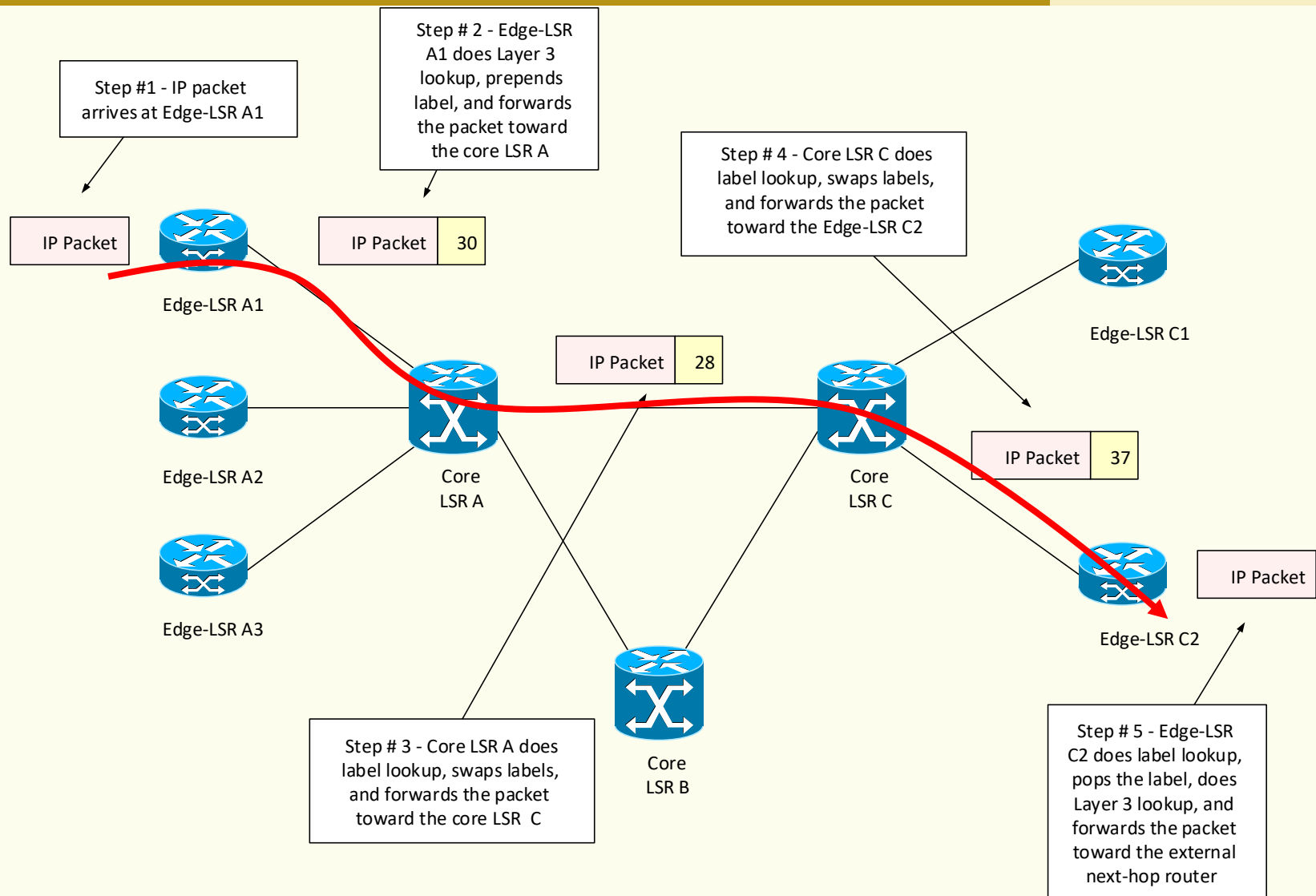


# MPLS forwarding

- In concrete implementations, the data plane structures may take different names and/or have different organization
- **Label Forwarding Information Base**



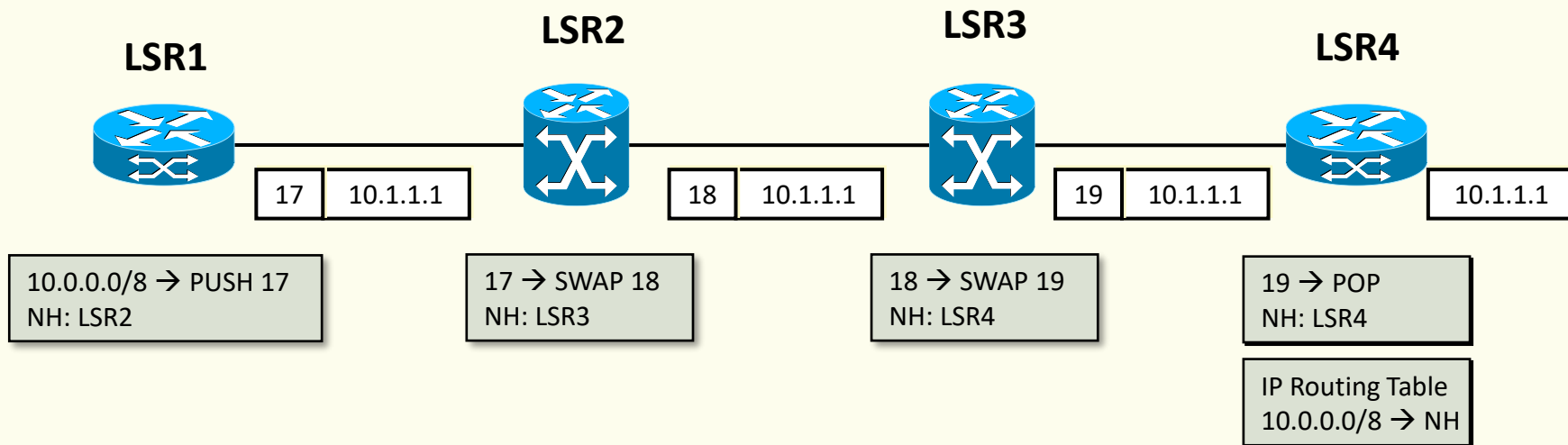
# Label Switched Path (LSP)





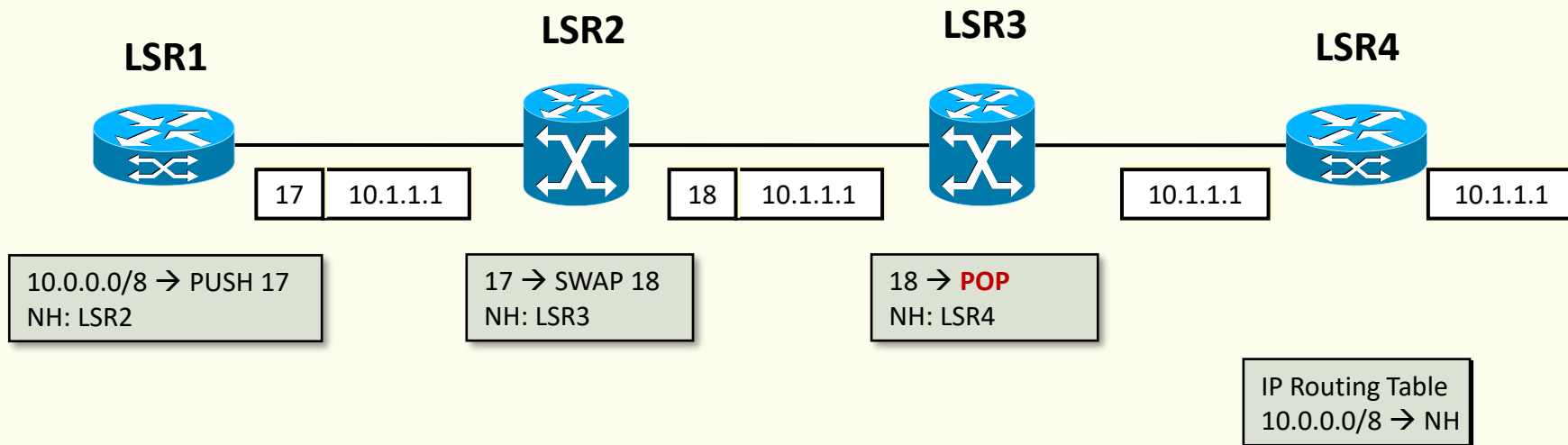
# Penultimate Hop Popping

- Double lookup at LSP egress
  - LFIB: remove the label
  - FIB: forward the IP packet based on IP next-hop address



# Penultimate Hop Popping

- Single lookup at LSP egress
  - ~~LFB: remove the label~~
  - FIB: forward the IP packet based on IP next-hop address



# Label stacking

- The MPLS data plane is based on a more general **stack model**
  - A labeled packet carries a number of labels  $m \geq 0$
  - Labels are organized as a last-in, first out stack
    - Label 1, bottom of stack
    - Label  $m$ , top of stack
  - Actions in NHLFEs always apply to the top label in the stack



# Label stacking and actions



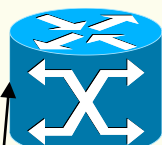
- **SWAP**  $L_x$ 
  - replace the label at the top of the label stack with a specified label  $L_x$
- **POP**
  - pop the label stack
- **PUSH**  $L_x$ 
  - push one new label  $L_x$  onto the label stack
  - Multiple pushes are allowed

# Label stacking – example

FEC: 10.1.0.0/16  
PUSH 30  
NH: LSR2



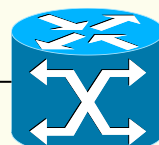
10.1.0.0/16



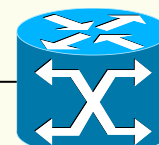
LSR1



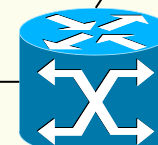
LSR2



LSR3



LSR4

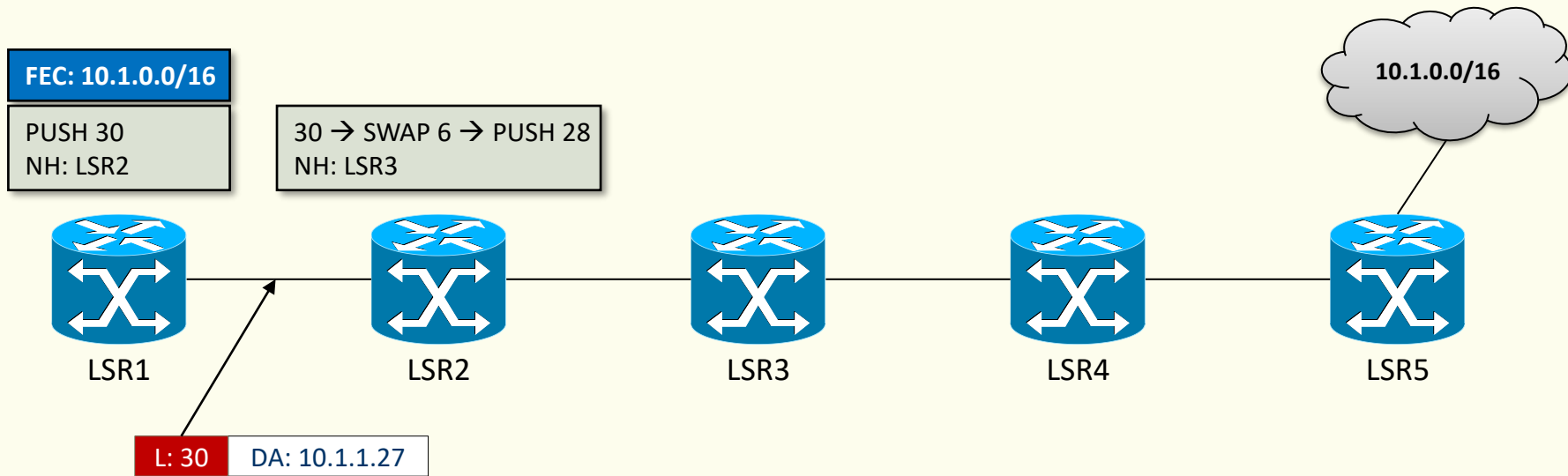


LSR5

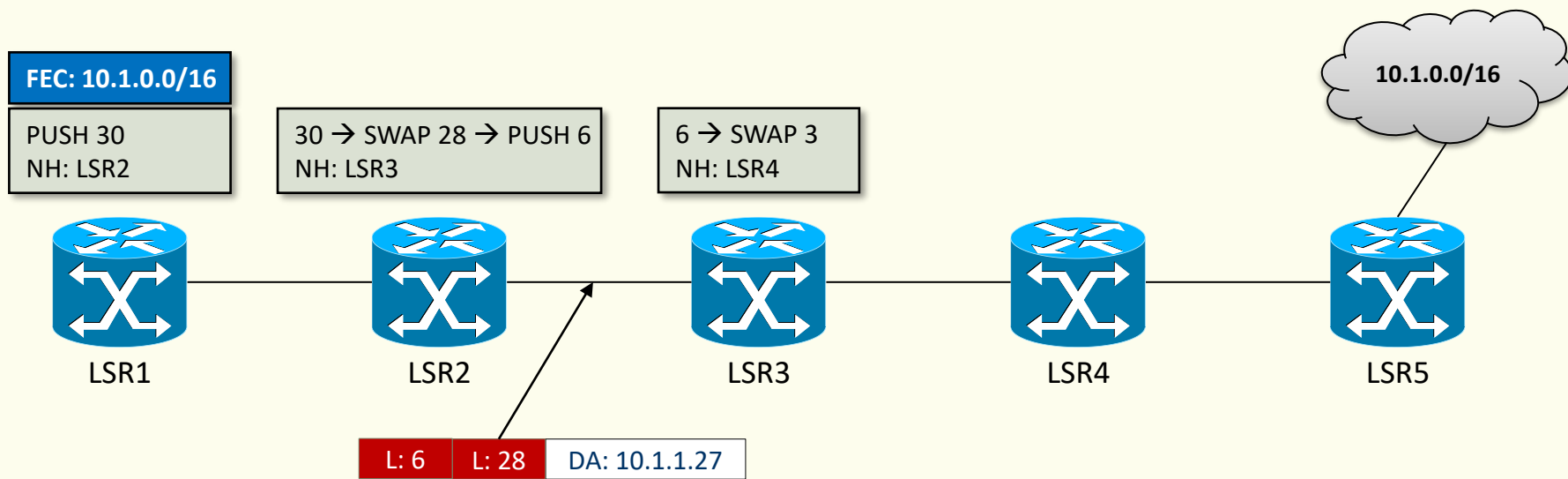
DA: 10.1.1.27

Incoming IP packet

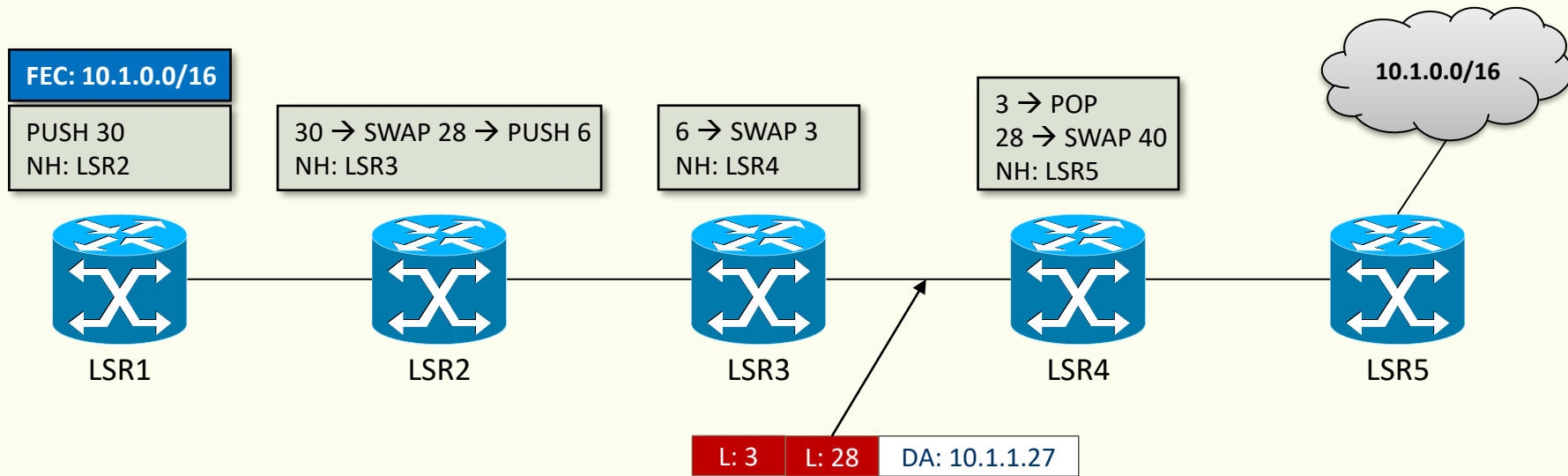
# Label stacking – example



# Label stacking – example

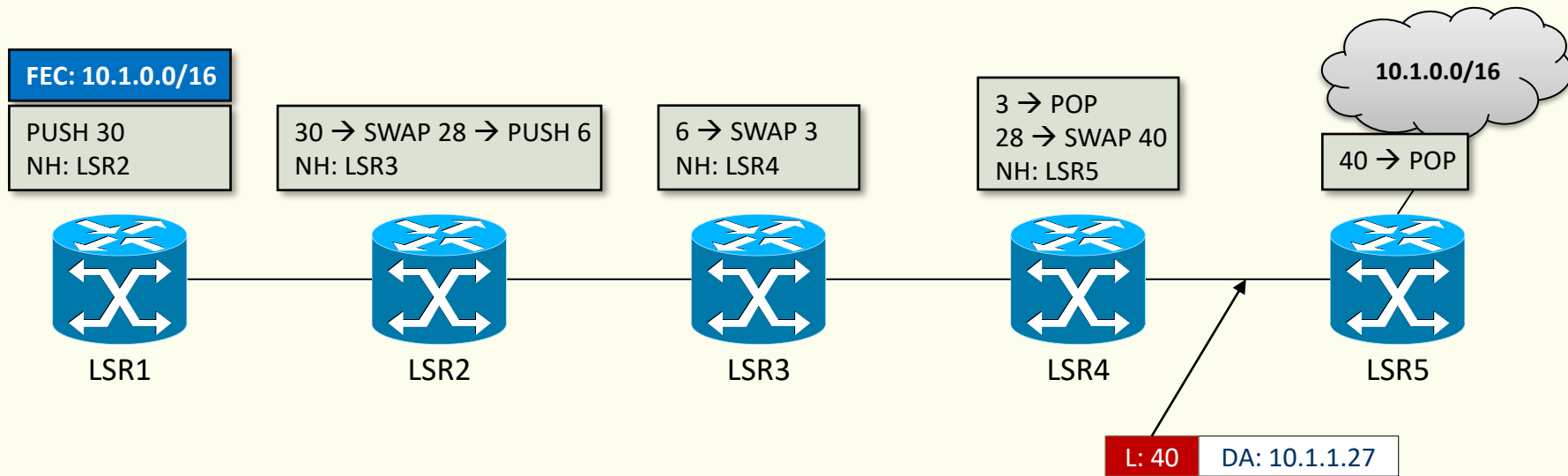


# Label stacking – example

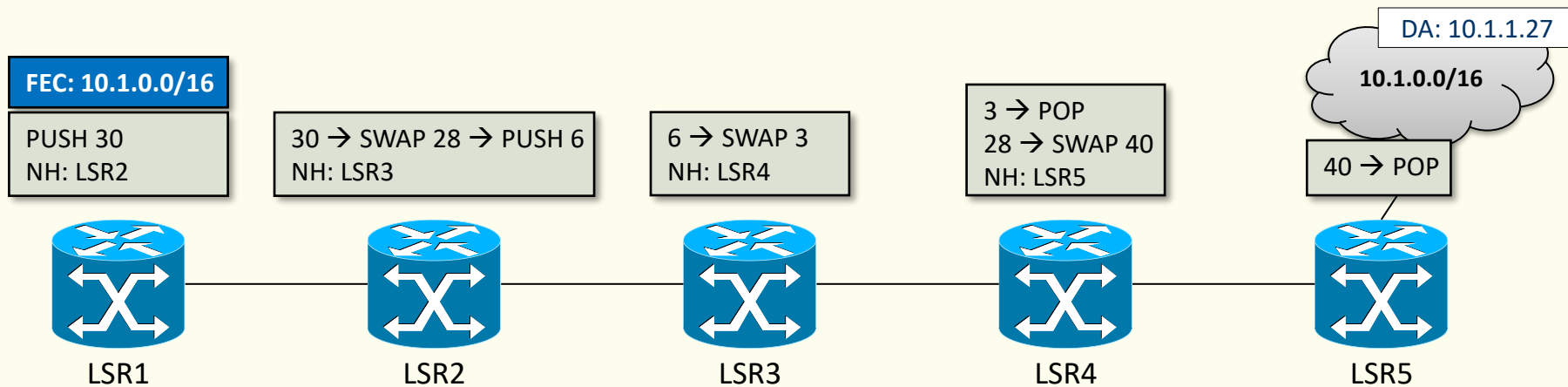




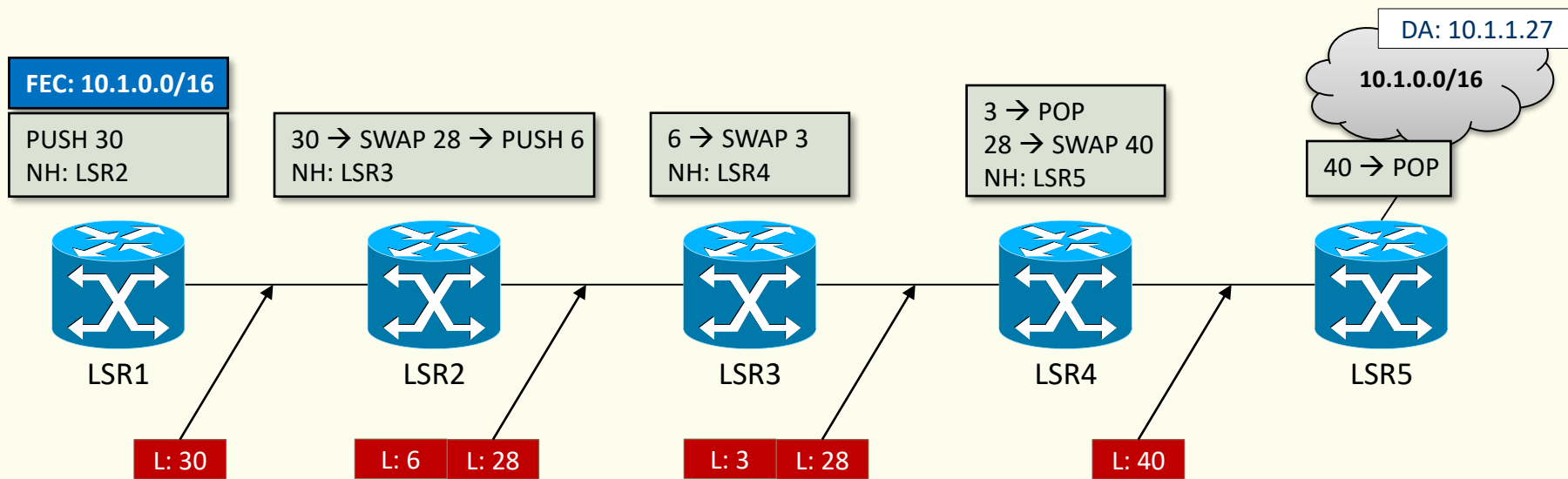
# Label stacking – example



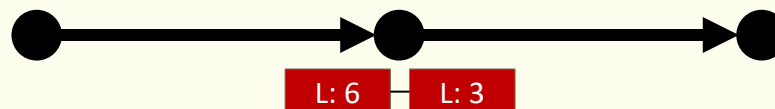
# Label stacking – example



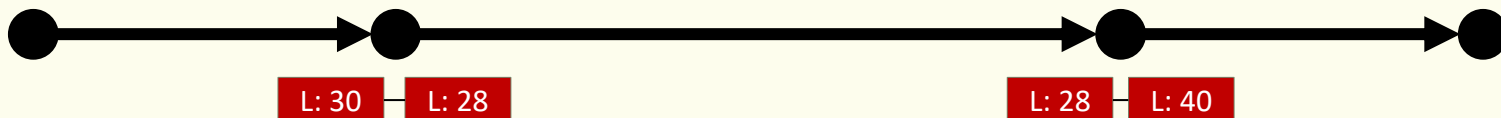
# Label stacking – example



LSP level 2

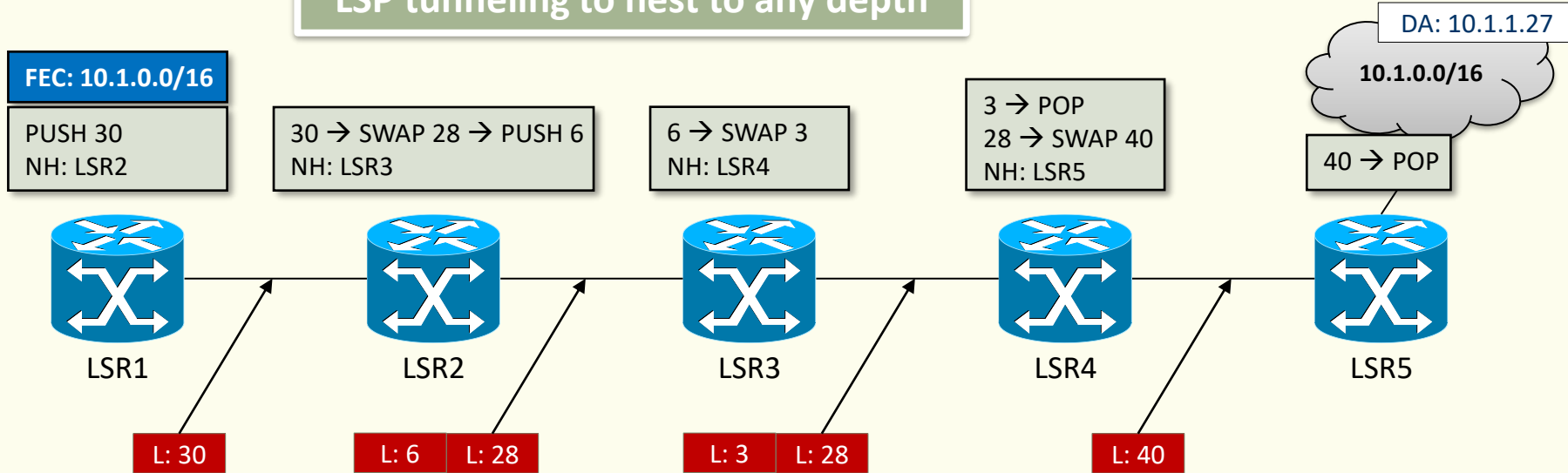


LSP level 1

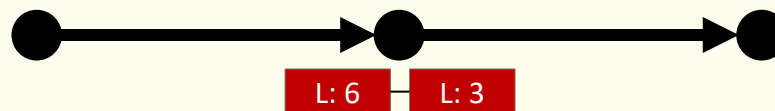


# Label stacking – example

The label stack mechanism allows LSP tunneling to nest to any depth



LSP level 2



LSP level 1

