MPLS

MULTI PROTOCOL LABEL SWITCHING

OVERVIEW OF MPLS, A TECHNOLOGY THAT COMBINES LAYER 3 ROUTING WITH LAYER 2 SWITCHING FOR OPTIMIZED NETWORK USAGE

Peter R. Egli
peteregli.net

Contents

1. Why MPLS (1/2)?

Why not ATM?

ATM (Asynchronous Transfer Mode) as a backbone transmission technology has a scalability problen.

➔ *ATM is connection-oriented:*

Between any pair of switches / routers, a separate PVC (Permanent Virtual Connection) must be set up. If optimal routing is required a full mesh of PVCs needs to be established (O(n^2) complexity). In large networks this becomes unfeasible.

➔ *ATM does not have powerful routing protocols:*

ATM does not provide such powerful routing protocols as IP (OSPF, BGP).

➔ *ATM is dying:*

ATM is slowly but steadily being replaced by other, simpler technologies.

Why not simple IP routing?

➔ *QoS is difficult to meet with traditional routers:*

Traditional routers have become the bottleneck in the backbone.

➔ *Routing is costly:*

Routing is costly ($) since it needs a lot of performance and memory.
(@ 2014 ca. 300‘000 BGP routes in Internet backbone; requires 60-120Mb memory to hold these routes)

1. Why MPLS (2/2)?

MPLS (RFC3031) combines the advantages of layer 3 routing with layer 2 packet forwarding (in hardware).

- **Routing in MPLS is done with IP routing protocols. No change to the existing Internet backbone routing infrastructure is required (AS/BGP4 for Internet backbone, OSPF for ‚private' or enterprise backbone routing).**
- **QoS (Quality of Service) is achieved through layer 2 switching.**
- **The routing tables size can be reduced through layer 2 switching.**
- **MPLS allows to add additional services like VPN.**

2. MPLS elements and terms (1/3):

LSR:
Label Switching Router. Core MPLS router/switch that switches packets based on label.

LER:
Label Edge Router; same as edge-LSR; a LER sits at the edge of a network and performs label push/pop = label imposition/disposition.

LSP:
Label Switched Path (is unidirectional). Path that an IP packet takes.

Ingress LSR:
The ingress LSR is an LER an as such the first MPLS hop in an LSP.
The ingress LSR performs the transition from IP routing and packet forwarding to MPLS switching (IP to MPLS).

Egress LSR:
The egress LSR is an LER as well and does the opposite operation of an ingress LSR.
The egress LSR terminates the MPLS LSP and performs the transition from MPLS switching to IP routing and packet forwarding (MPLS to IP).

Label swapping:
Ingress to egress label exchange (like ATM VPI/VCI).

2. MPLS elements and terms (2/3):

FEC:
Forwarding Equivalent Class. Set of packets that belong to the same forwarding treatment.

PHP:
Penultimate Hop Popping. The penultimate LSR removes (‚pops') the MPLS label and forwards the IP packet to reduce MPLS processing overhead.

LIB:
Label Information Base (Label to IP prefix binding table).
The LIB contains the label bindings (mappings) for every route prefix received via BGP.

FIB:
Forwarding Information Base (routing table), used in the ingress-LSR. The FIB is actually the same as the IP routing table but augmented with MPLS information to allow a decision about the forwarding method (forward incoming IP packet as a pure IP packet or label it and send it into an MPLS LSP).

LFIB:
Label Forwarding Information Base. The LFIB contains the mappings that are actually used by the label switching engine. It contains a subset of the label bindings of the LIB.

2. MPLS elements and terms (3/3):

Label binding:

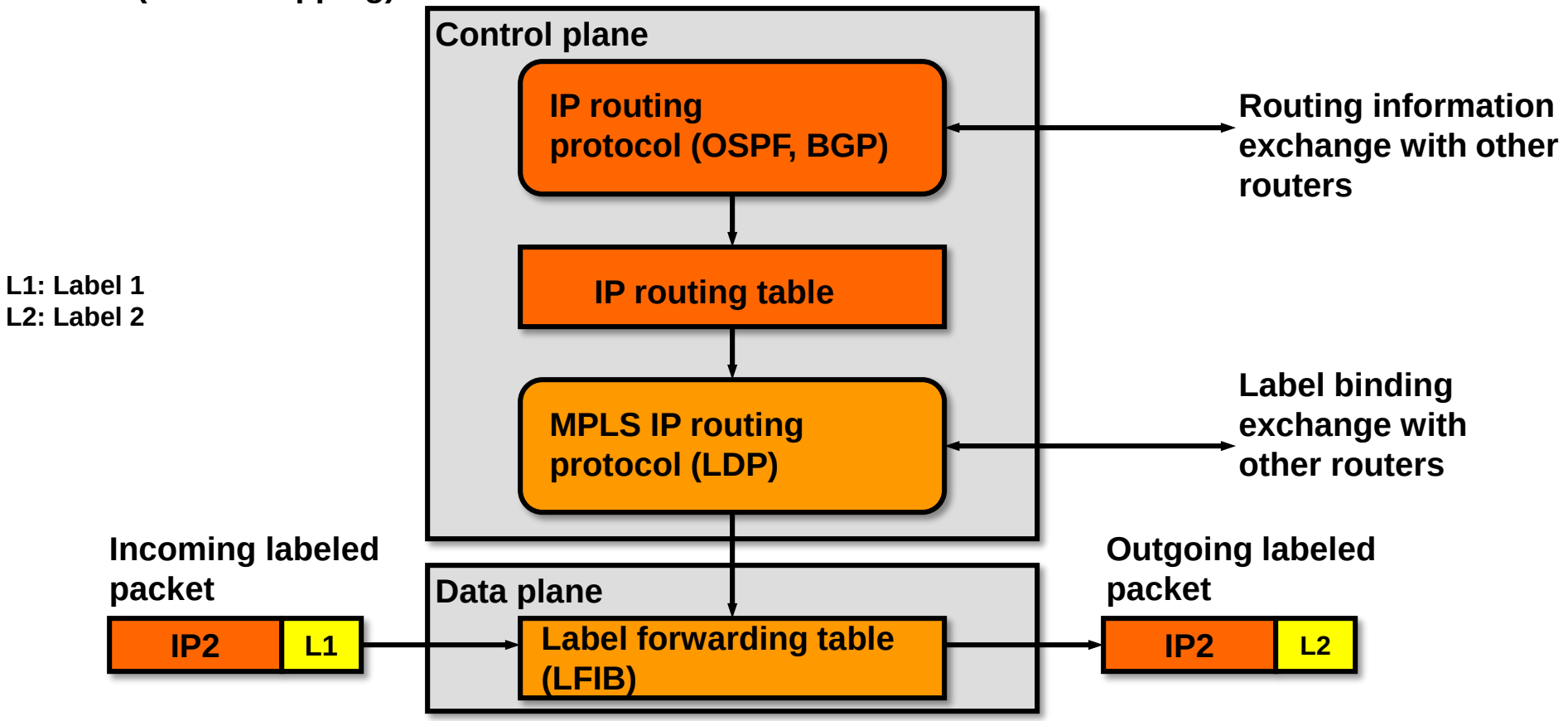
The label binding is the mapping of an IP prefix (FEC) to a label.

Label imposition:

Label imposition is the same as label pushing.

3. Basic architecture of an LSR MPLS node:

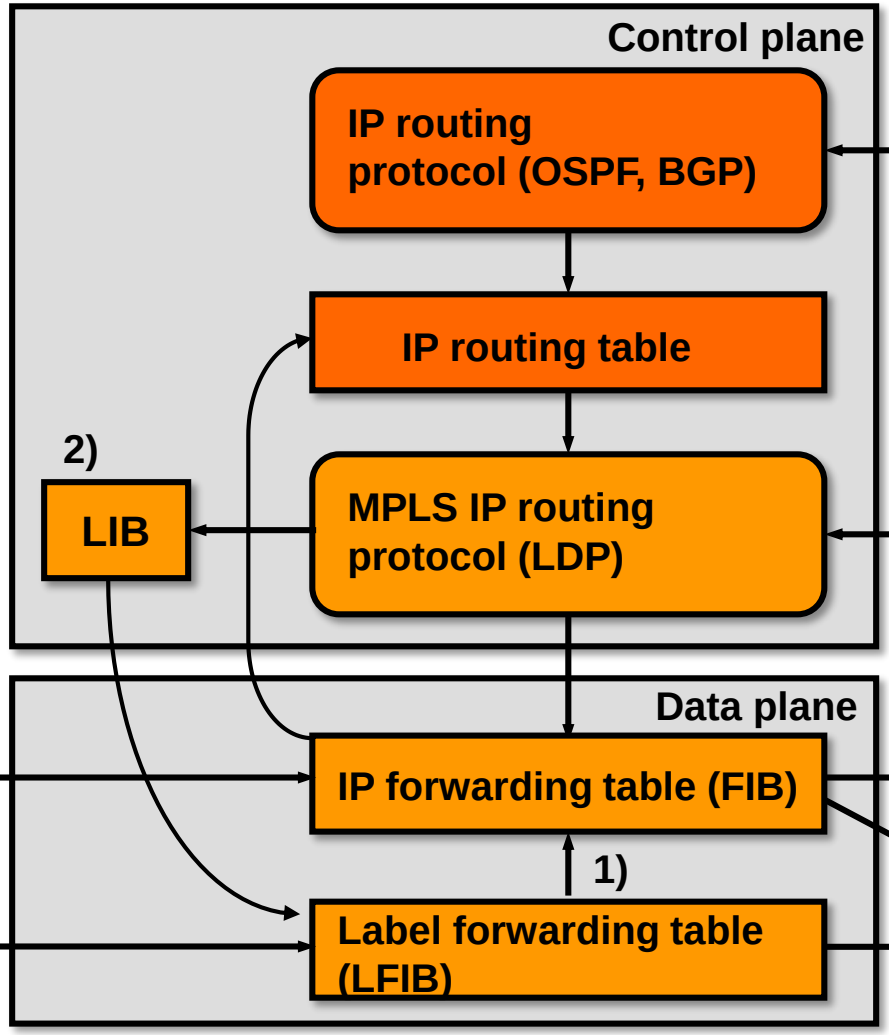
➔ MPLS consists of a Forwarding (data plane) and a Control (control plane) part.
➔ The control plane runs an ordinary IP routing stack with routing protocols.
➔ Additionally the control plane runs a label binding (label to IP prefix) exchange protocol with other routers (LDP and others).
➔ The data plane receives labeled packets, looks up the label to ascertain the outgoing interface and label (label swapping).



4. Basic architecture of an LER MPLS (MPLS edge node):

➔ An edge-LSR (LER) augments the general MPLS node architecture with an MPLS-enabled IP forwarding table (FIB) that allows desicion routing/switching on incoming IP packets.

1) Label removal and subsequent L3 route lookup.
2) The LIB contains all prefix to label mappings received by this LSR.



Routing information exchange with other routers (BGP4)

Label binding exchange with other routers (LDP)

Incoming IP packets

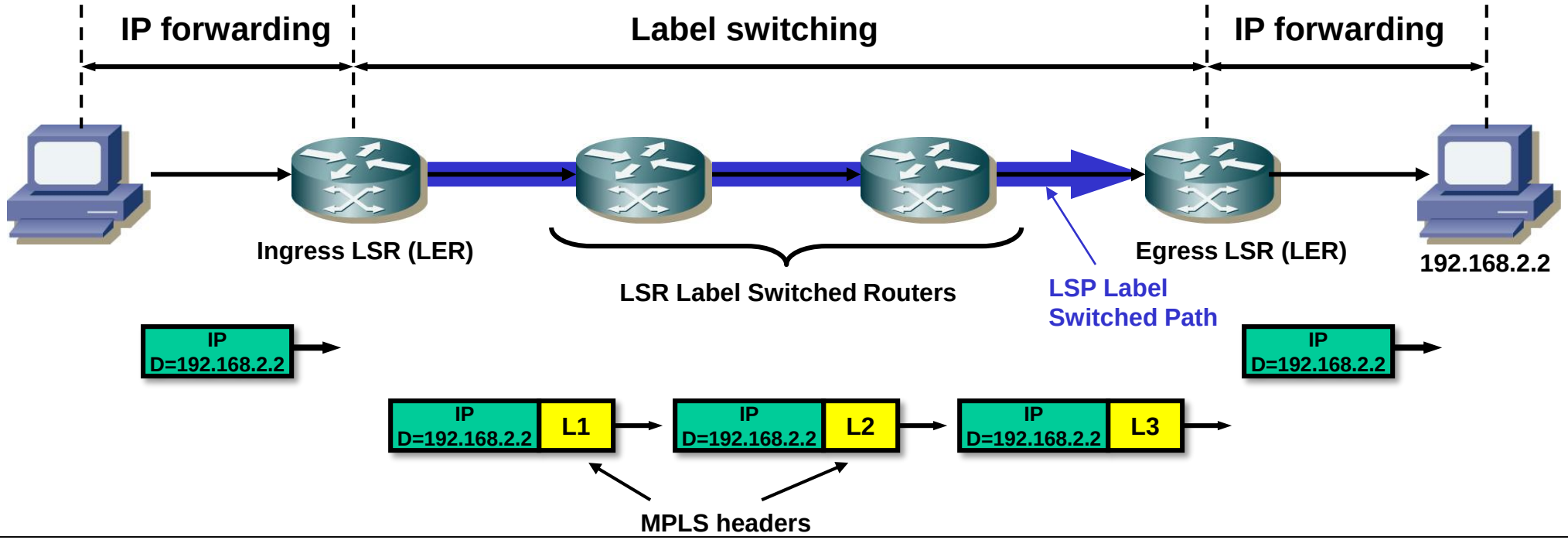
Outgoing IP packets

Incoming labeled packet

Outgoing labeled packets

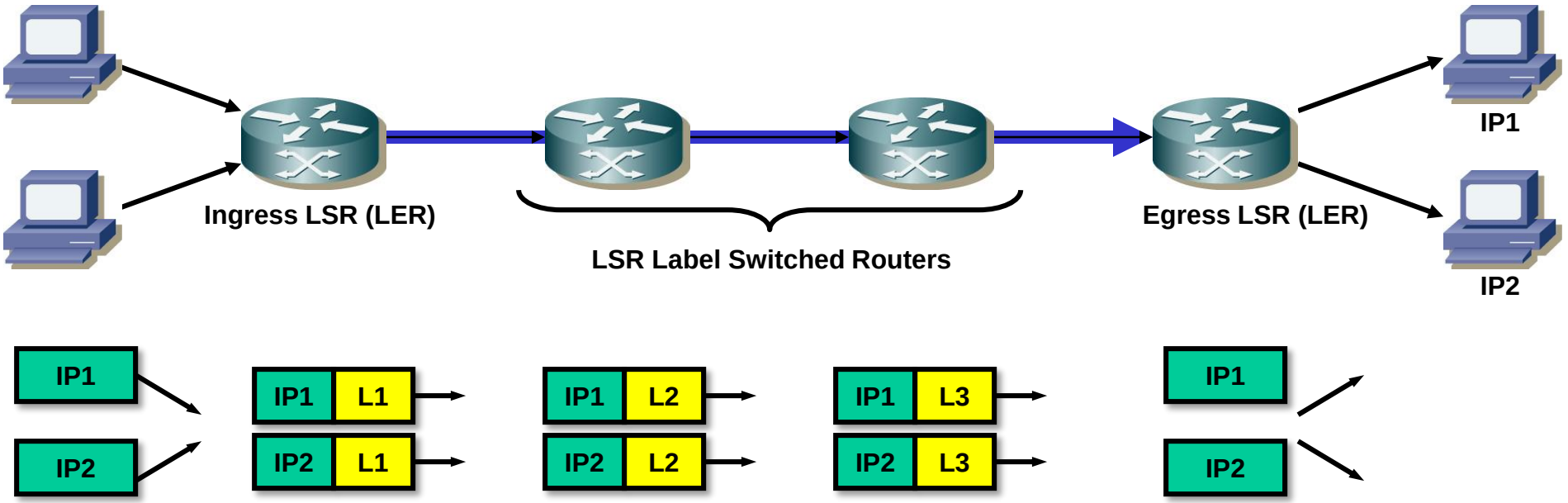
5. MPLS operation – label switching (1/2):

1. The ingress LSR receives the IP packet with Dest.=192.168.2.2, classifies it into an FEC and labels it with a label according to the FEC (L1). The FEC corresponds to a traditional destination subnet.
2. The core LSRs receive the packet, use the ingress label as an index into a lookup table and thus ascertains the egress label and interface on which to forward the packet (label swapping and switching).
3. The egress LSR receives the packet, removes the label and performs a traditional layer 3 routing lookup to forward the packet to the final destination.



5. MPLS operation – label switching (2/2):

➔Multiple streams of IP traffic (IP prefixes) can be mapped onto the same path (LSP).
E.g. both IP1 and IP2 in the example below are mapped into the same FEC (Forwarding Equivalent Class).
➔ The assignment of label (path) can be done based on VPN identifiers or QoS.



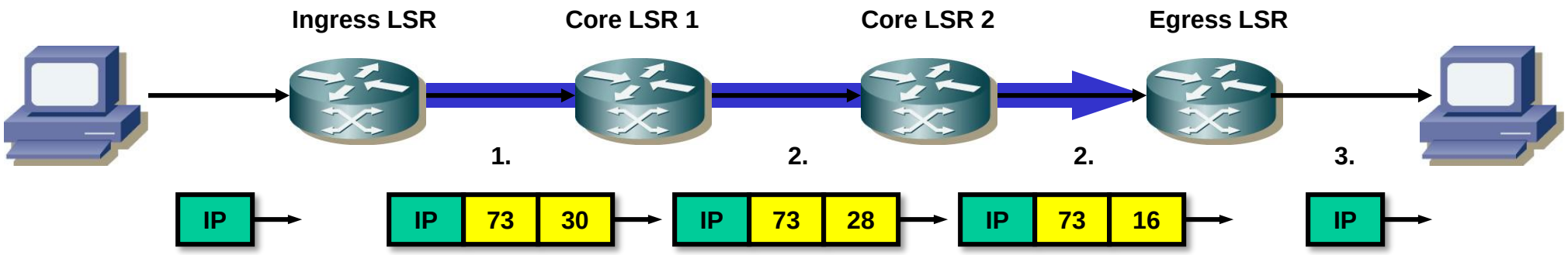
6. MPLS label stacking:

➔ Labels can be stacked for specific purposes, e.g. VPNs. The switching in the MPLS core uses only the outermost label. In case of VPNs the inner label is used for destination lookup in the egress LSR.

➔ Usually only 2 labels are stacked (inner label for VPN, outer label for path).

➔ Label stacking is used by RFC2547bis (MPLS based VPNs with BGP4).

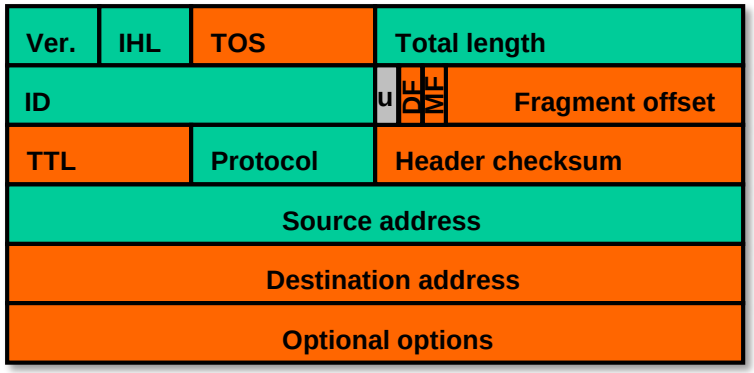
1. Ingress LSR performs route lookup to ascertain FEC. The IP packet belongs to a VPN (VPN membership of route prefixes along with label are exchanged with BGP in core).
2. Core LSR switch the packet downstream and exchange the outer label.
3. The egress LSR pops the outer label, performs a label lookup which tells it to pop the outer label. Then the egress LSR looks up the inner label that tells the LSR to which VPN the packet belongs. The egress LSR pops the inner label and performs a route lookup in the VPNs routing table. Eventually the egress LSR forwards the packet towards the destination.



7. IP routing versus MPLS switching (1/4):

➔ **Comparison of headers:**

IP header



MPLS Header

Label: Address
EXP: EXPerimental or EXPedite bits
S: If set to 1 indicates bottom of label stack in case of label stacking.
TTL: Time To Live (max. number of MPLS hops, like IP TTL) for loop detection.

Routing (packet forwarding) means:

1. Decrement TTL by 1; drop packet if TTL=0
2. Route lookup (packet leaves router through which interface?)
3. (Optional) Fragment packet if too big for outbound interface
4. Recalculate header checksum
5. Apply QoS (change TOS/DSCP value, put packet into priority queue)

Switching a packet with MPLS means:

1. Lookup of outbound interface with MPLS label; outbound LSP's label is looked up in the same step. Copy outbound label into label field in header.
2. Decrement TTL by 1
3. Apply QoS (queueing of MPLS packet).

➔ **The MPLS address is only 20 bits which represents an address space of 1 million entries. This reduces memory demands and allows using the label as a direct index into a lookup table.**
➔ **The MPLS address does not have prefixes („masks"). This allows using the label as a direct index into a lookup table.**
➔ **MPLS switching does not involve costly processing such as header checksum calculation.**

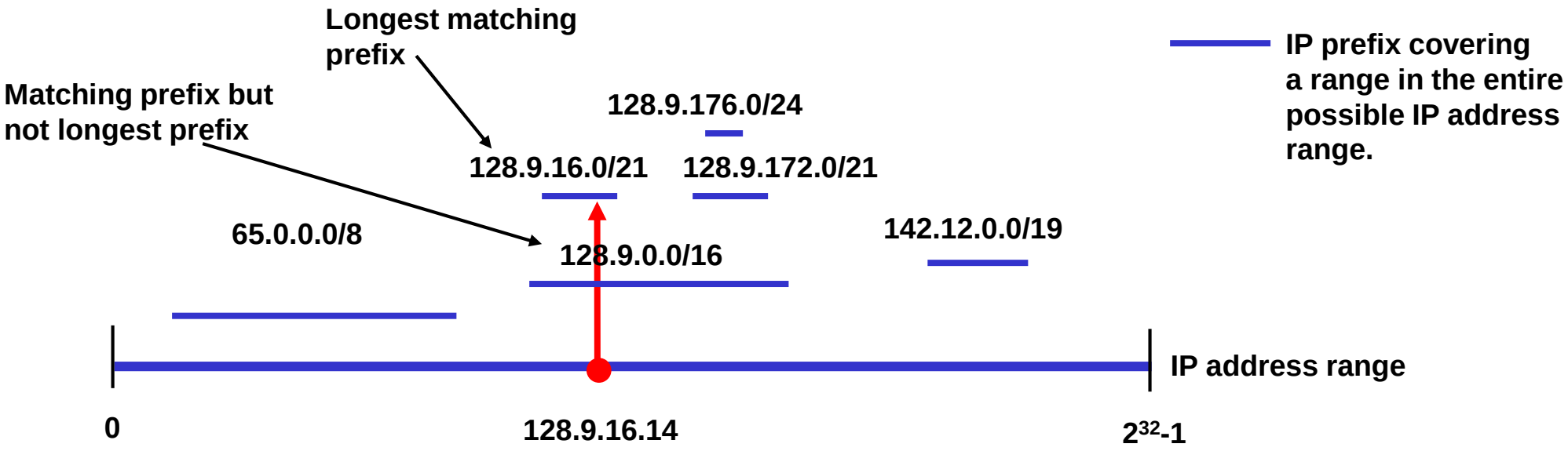
7. IP routing versus MPLS switching (2/4):

➔ **Why IP route lookups are costly:**

Route lookups = Find the longest matching prefix among all prefixes that match the destination address. Both 128.9.16.0/21 and 128.9.0.0/16 in the example below match the destination IP address 128.9.16.14 but the former is more specific (longer prefix) and thus is the correct routing entry.

➔ **The job of the router is to find this longest prefix matching route entry among possibly many other matching (but less specific) route entries.**

➔ **IPv6 still has prefixes (to allow route aggregation) but unlike IPv4 the address space is gigantic and can be partitioned more freely and thus allows easier aggregation.**



7. IP routing versus MPLS switching (3/4):

➔ **Patricia trie (tree with internal and external (=leaf) nodes) (1):**

The PATRICIA trie (Practical Algorithm to Retrieve Information Coded in Alphanumeric) is one of many route lookup algorithms (e.g. used in BSD kernel).

A. Build up of trie:

1. Generally the PATRICIA trie contains the prefixes as nodes. 0 bit in address prefix results in a node to the left of the current node, 1 bits result in a node to the right of the current node.

2. If there is only 1 child node it is removed (path compression). The remaining parent node stores the number of bits that were compressed and thus can be skipped. This reduces the number of necessary comparisons.

B. Search algorithm:

The router proceeds bit by bit of the destination IP address through the tree (decision left or right child node).

The router skips the specified number of bits as indicated in the nodes.

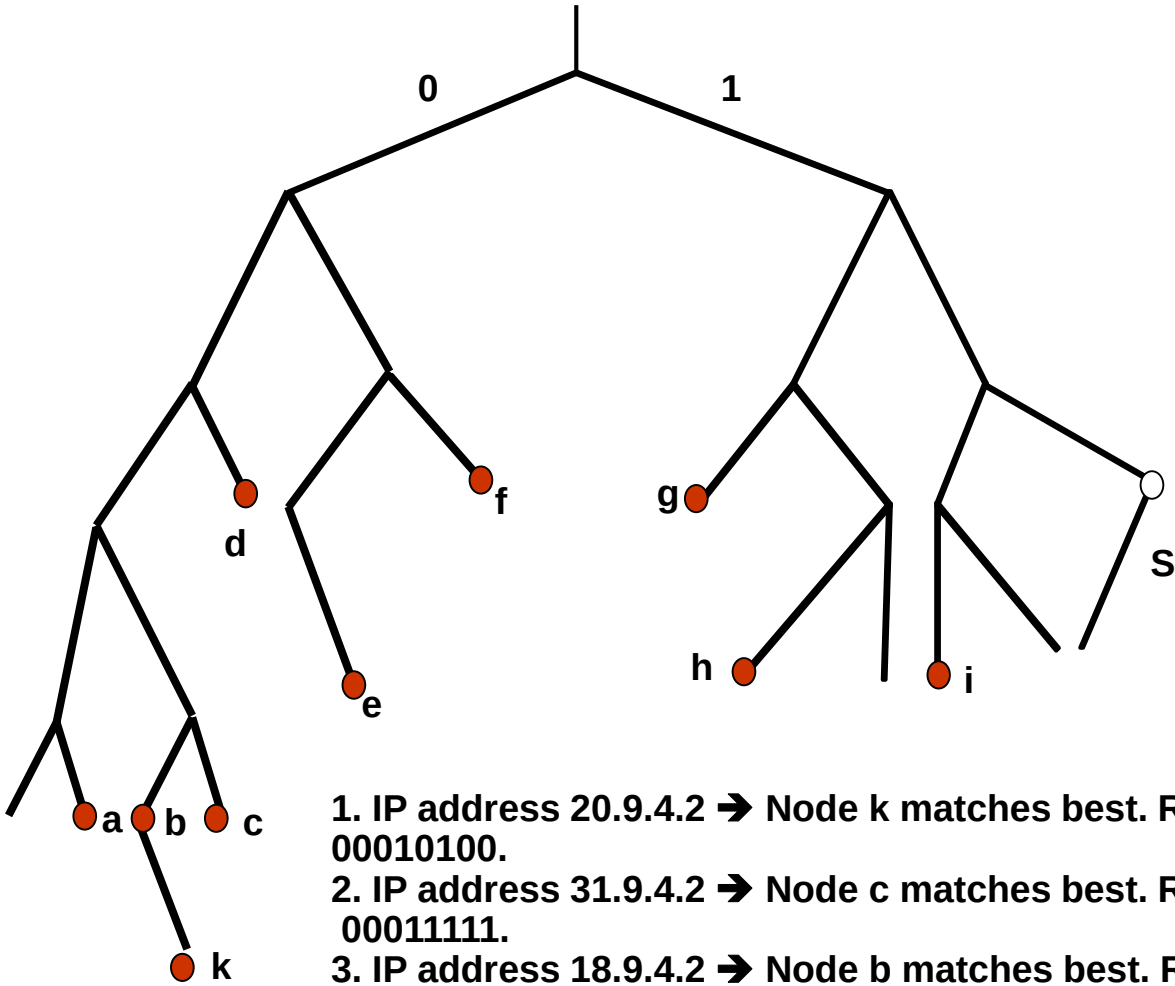
The PATRICIA trie does not allow to find an exact match but only a possible match. Therefore the router must perform a full match in the leaf node (after masking the input IP address with the number of bits that lead to this leaf node). If the destination IP matches the full prefix that the leaf node contains the route lookup is successful and the router finds the outgoing interface and next-hop-gw in the leaf node.

If not the router must go up the trie (towards the root) and do a mask/full match operation in each node until there is a match. If none of the nodes match the root node will (if configured so) contain the default route. If no default route is configured the route lookup has failed.

➔ **It is evident that the maintenance of the PATRICIA trie (add routes / nodes, remove routes / nodes based on routing protocol like OSPF, BGP) and the lookup are costly in terms of processing power (N.B.: The lookup is done anew per IP packet).**

7. IP routing versus MPLS switching (4/4):

➔ Patricia trie (tree with internal and external (=leaf) nodes) (2):



Example Prefixes:
a) 00001
b) 00010
c) 00011
d) 001
e) 0101
f) 011
g) 100
h) 1010
I) 1100
j) 11110000
k) 000101

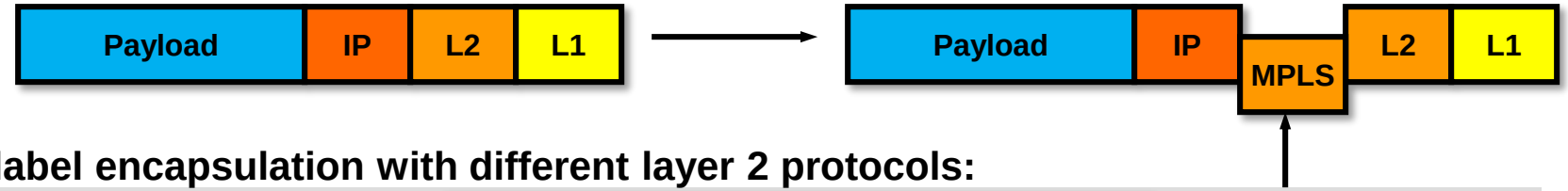
Prefix length

1. IP address 20.9.4.2 ➔ Node k matches best. Route in node k is taken. 00010100.
2. IP address 31.9.4.2 ➔ Node c matches best. Route in node c is taken. 00011111.
3. IP address 18.9.4.2 ➔ Node b matches best. Route in node b is taken. 00010010.

8. MPLS header position:

➔ Where is the MPLS header (label)?

MPLS, unlike other protocols, is inserted between existing layer 2 and layer 3 header („shim“ header):



➔ MPLS label encapsulation with different layer 2 protocols:

PoS (Packet over SONET):

Ethernet:

Frame Relay:

Label over ATM PVCs:

(subsequent cell)

ATM label switching:

(subsequent cell)

9. MPLS label distribution (1/2):

➔ **LDP Label Distribution Protocol:**

Label distribution propagates upstream from destination to source. In this way a LSP is already established when the last label binding „hits“ the edge-LSR (source of IP packet).

1. LDP hello packets are used to discover neighbors on links. LDP hello packets use IP broadcast or IP multicast and UDP.

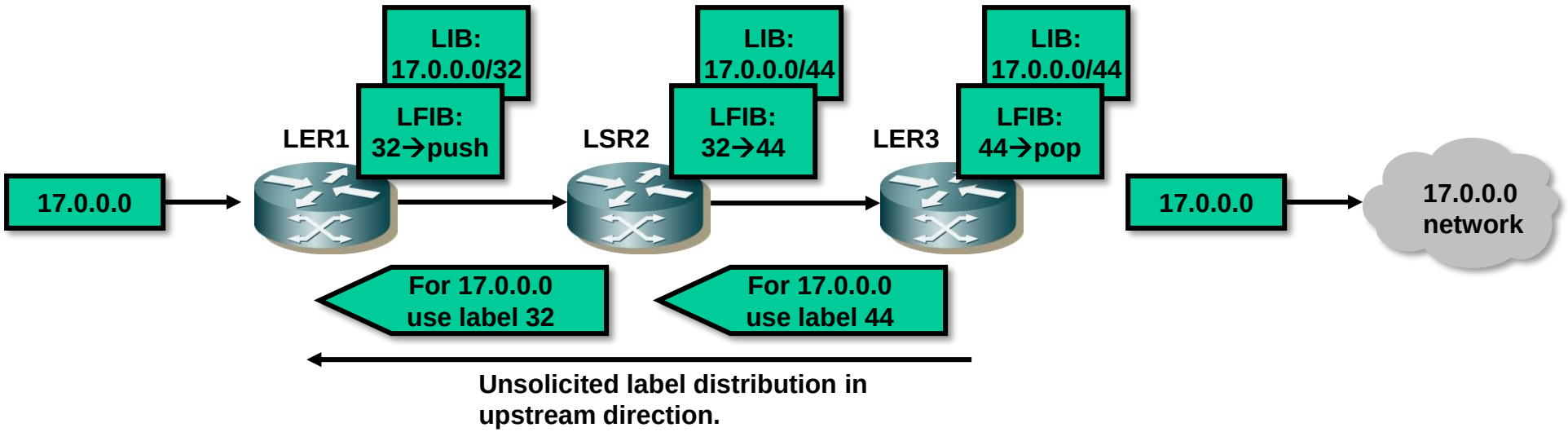
2. After the discovery the LDP neighbors establish an LDP session through a TCP connection (like BGP).

3. Every Edge-LSR creates IP prefix → label bindings for the network it is attached to and distributes these bindings via LDP to its upstream neighbors (into the MPLS cloud).

4. The neighbors fill their LIB with the LDP bindings received from both upstream and downstream LDP neighbors (LIB contains mapping IP/FEC→label).

5. The LSRs distribute all their label bindings to their adjacent LSRs (both upstream and downstream).

6. The LSRs fill their LFIB only with label bindings that were received from a downstream neighbor.



9. MPLS label distribution (2/2):

➔ LDP parameters:

1. Unsolicited vs. on-demand distribution:

Unsolicited: Label is distributed even if upstream LSR does not need prefix/label mapping.

2. Independent vs. ordered control:

Independent control allocation: Assignment of label to a new IP prefix (received via BGP) regardless of whether the router has already received a label mapping for the same route prefix from the downstream LSR.

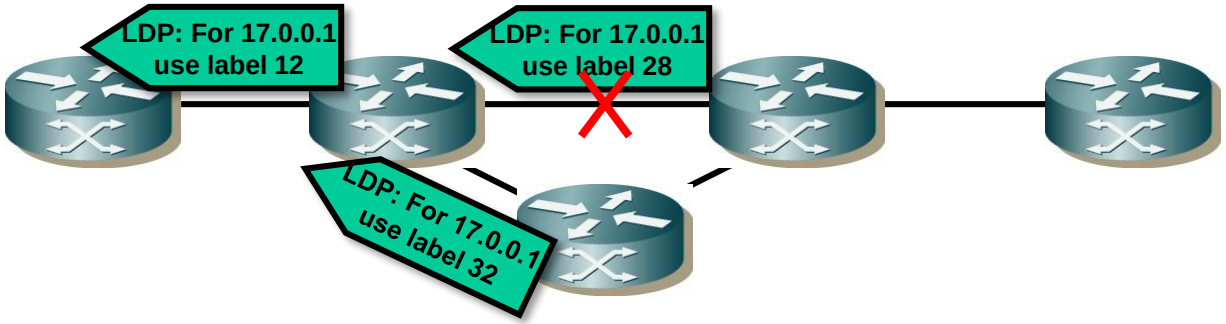
Ordered control allocation: Assignment of label only for prefixes where a downstream label already exists in LIB.

3. Liberal retention vs. conservative retention:

Conservative retention: An LSR only inserts label bindings into its LIB that are received from its current IP next hop gateway for this prefix.

Liberal retention: An LSR retains also label bindings that were not received from the current next hop gateway for that specific prefix.

➔ The combinations unsolicited distribution/independent control/liberal retention are used to provide faster convergence in case of a link failure:



Before link failure:

LIB:
17.0.0.0/28
17.0.0.0/32

LFIB:
12→28

After link failure:

LIB:
17.0.0.0/28
17.0.0.0/32

LFIB:
12→32

10. Penultimate hop popping PHP:

Without PHP the egress LSR performs the following steps on a received packet:

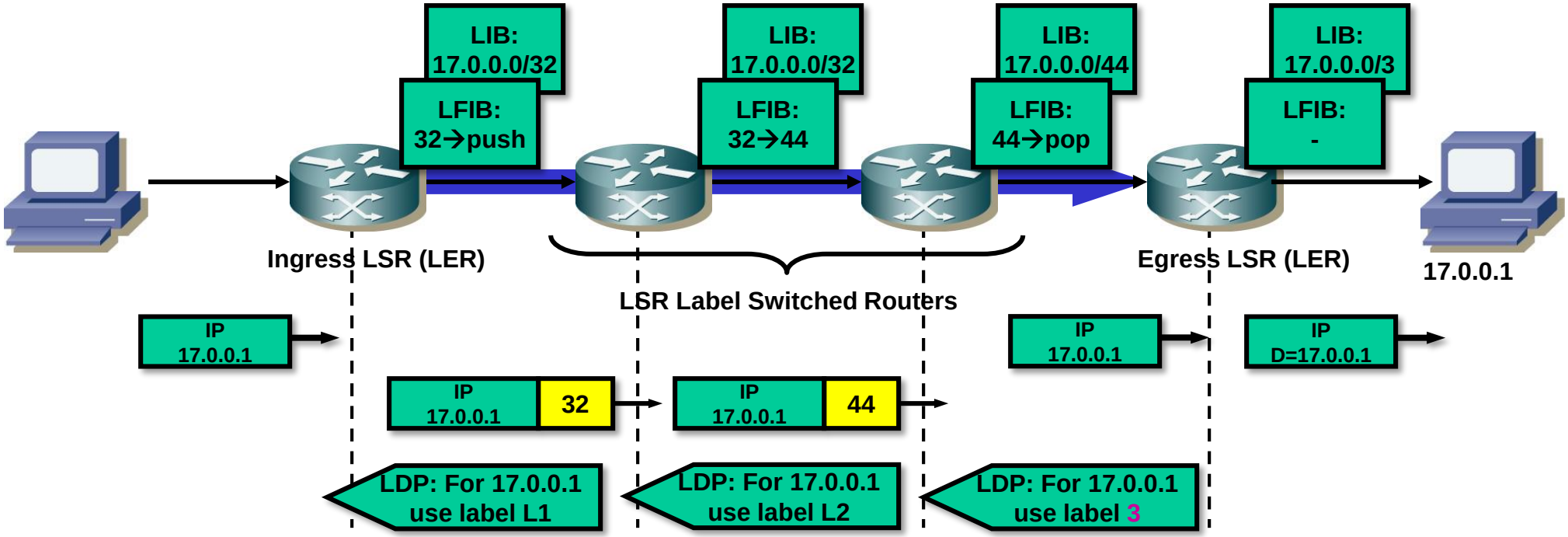
1. Inspect MPLS header and look up label. The lookup tells the LSR to pop the label (1st lookup).

2. Inspect IP packet and perform layer 3 (routing) lookup (2nd lookup).

3. Forward IP packet to destination.

The 2 lookups (MPLS and IP) present additional load on the egress edge-LSR. The MPLS label lookup can be removed without changing the MPLS logic.

➔ Thus the edge-LSR can request the upstream MPLS neighbor by sending a special LDP label (called *implicit null-label*, value 3). The penultimate LSR then pops the label and sends a pure IP packet to the egress-LSR.



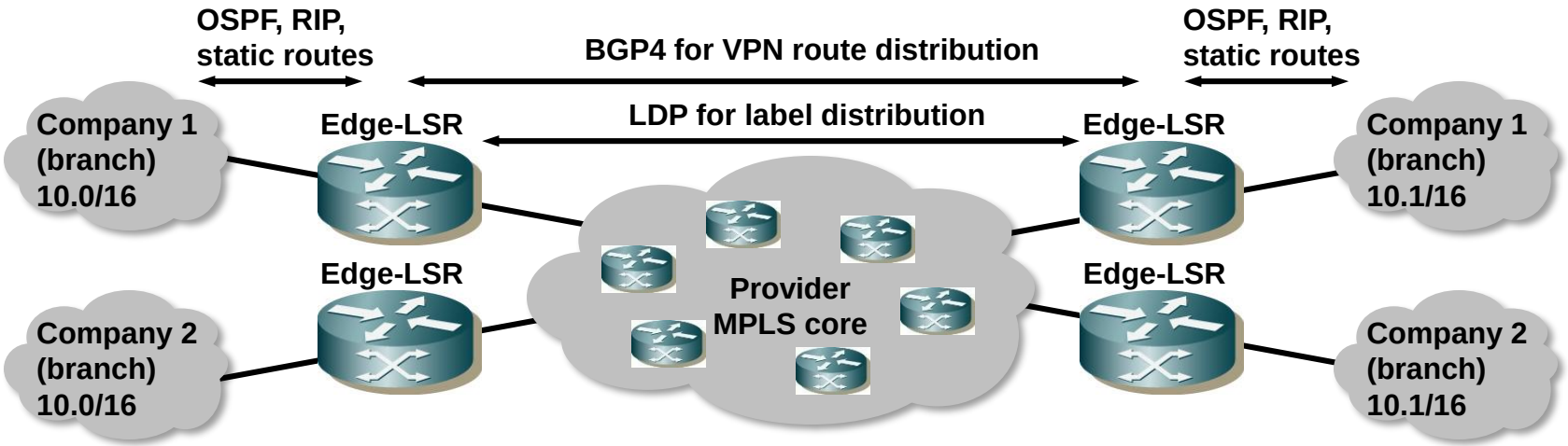
11. MPLS applications:

1. Backbone routing (switching, service provider network):

➔ Reduction of routes on backbone routers (lower memory demands).

➔ Faster routing (switching instead of routing).

2. Layer 3 VPNs (e.g. RFC2547bis):



3. QoS for IP-based networks:

➔ Assign IP packets to different traffic classes (FEC) based on different criteria (destination address, payload type (voice), TOS/DSCP bits etc.) and then prioritize packets.

➔ N.B.: MPLS per se does not provide QoS. MPLS is only a means to assign packets to classes and switch these classes differently.