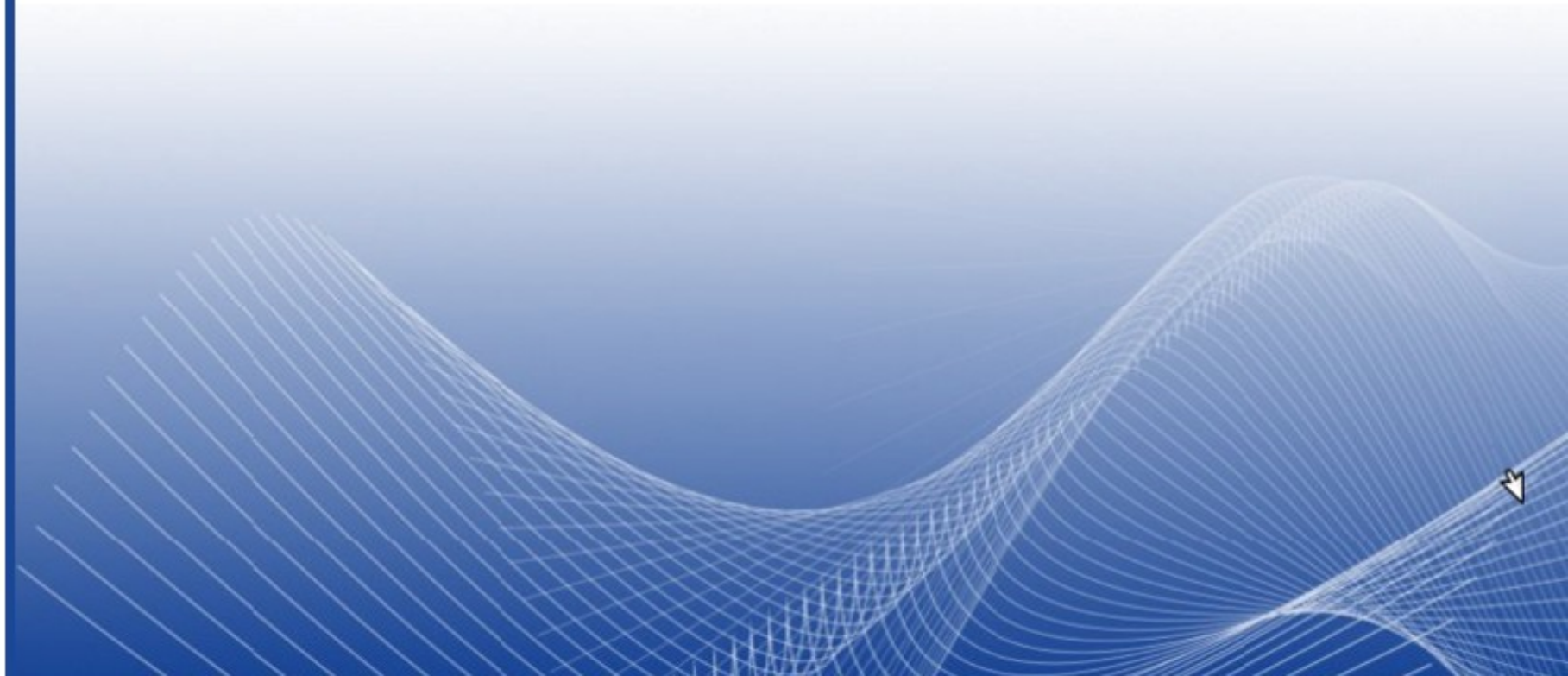
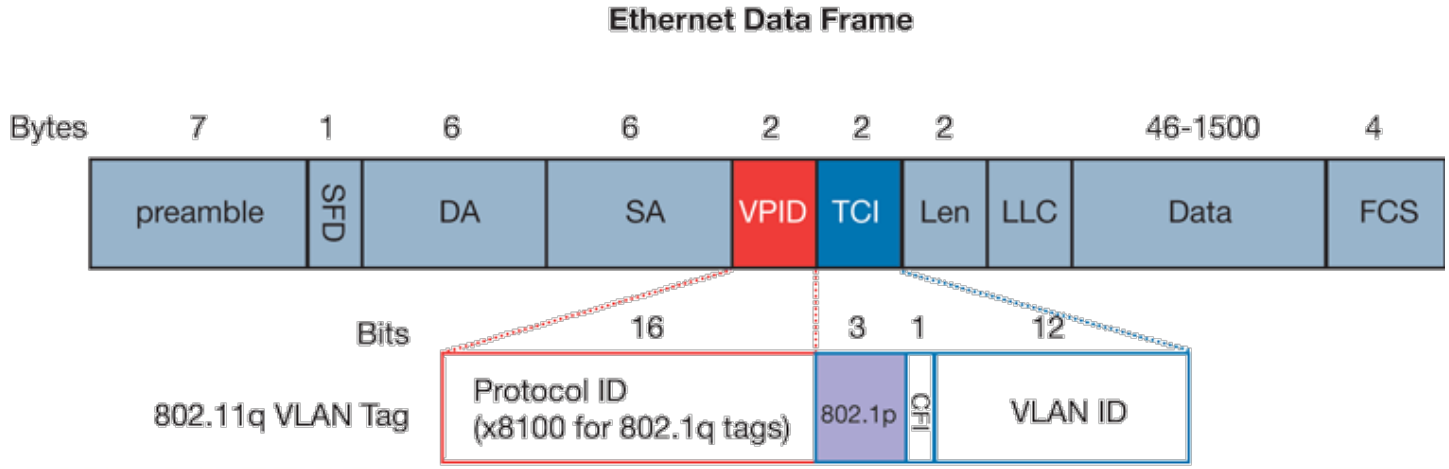
QoS
Layer 2
Layer 3 IntServ and DiffServ



universidade de aveiro
deti.ua.pt

Layer 2 QoS

- Layer 2 Ethernet switches rely on 802.1p standard to provide QoS.
  - 802.1p is part of the IEEE 802.1Q (VLAN tagging).
  - One of the tag fields, the Tag Control Information, is used by 802.1p in order to differentiate between the classes of service.
    - Allows different QoS classes on the same VLAN.
  - The three most significant bits of the Tag Control Information field known as Priority Code Point (PCP) are used to define frame priority.
    - PCP can be defined based on arrival port, terminal packet with 802.1p, or Layer 3 QoS (IP DSCP values).



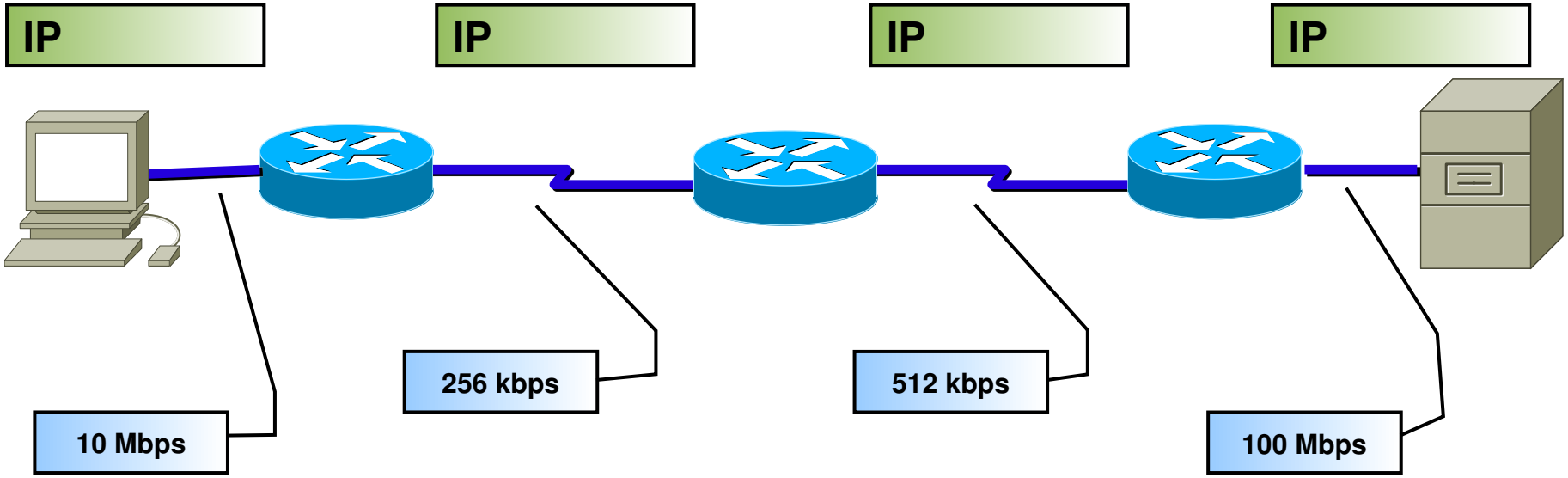
Layer 3/IP QoS? Because ...

- Application X is slow! (not enough BANDWIDTH)
- Video broadcast occasionally stalls! (DELAY temporarily increases – JITTER)
- Phone calls over IP are no better than over satellite! (too much DELAY)
- Phone calls have really bad voice quality! (too many phone calls – ADMISSION CONTROL)
- ATM (the money-dispensing-type) are non responsive! (too many DROPs)
- ...

What Causes ...

- Lack of bandwidth – multiple flows are contesting for a limited amount of bandwidth
- Too much delay – packets have to traverse many network devices and links that add up to the overall delay
- Variable delay – sometimes there is a lot of other traffic which results in more delay
- Drops – packets have to be dropped when a link is congested

Available Bandwidth

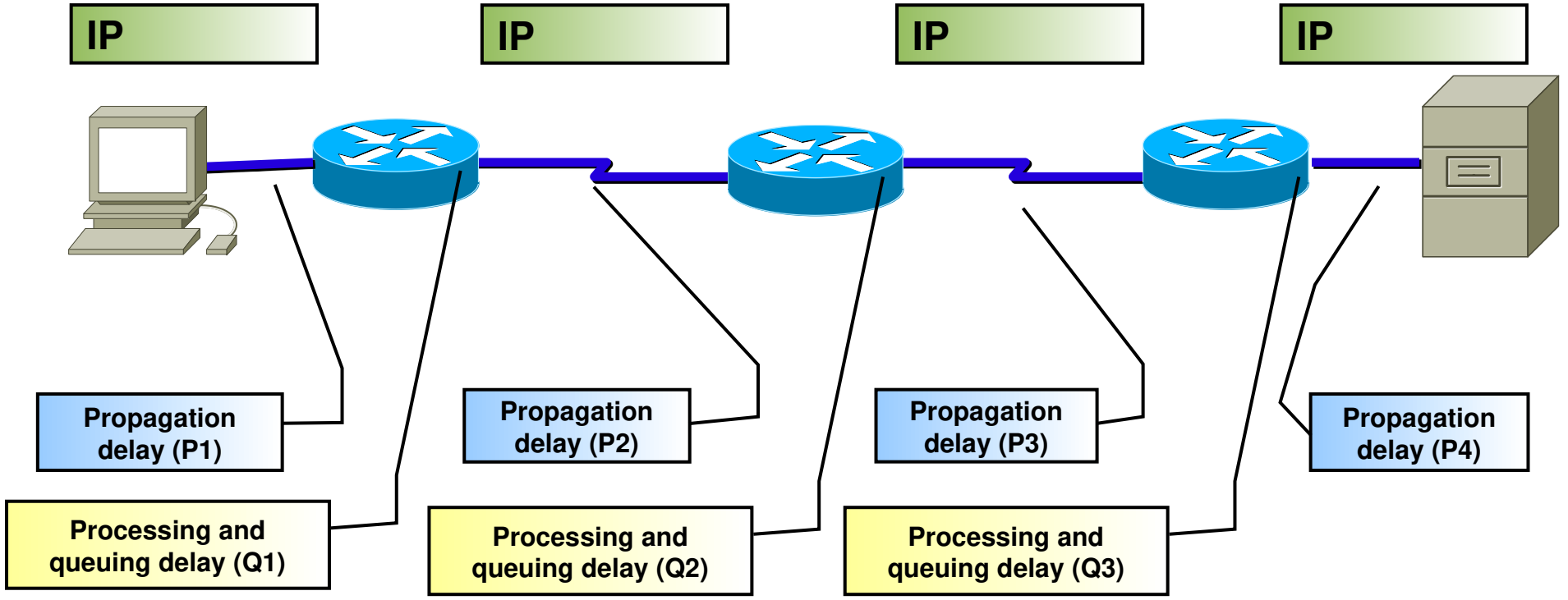


BW_{max} = min(10M, 256k, 512k, 100M)=256kbps

$BW_{avail} = BW_{max}/Flows$

- Maximum available bandwidth equals the bandwidth of the weakest link
- Multiple flows are contesting for the same bandwidth resulting in much less bandwidth being available to one single application.

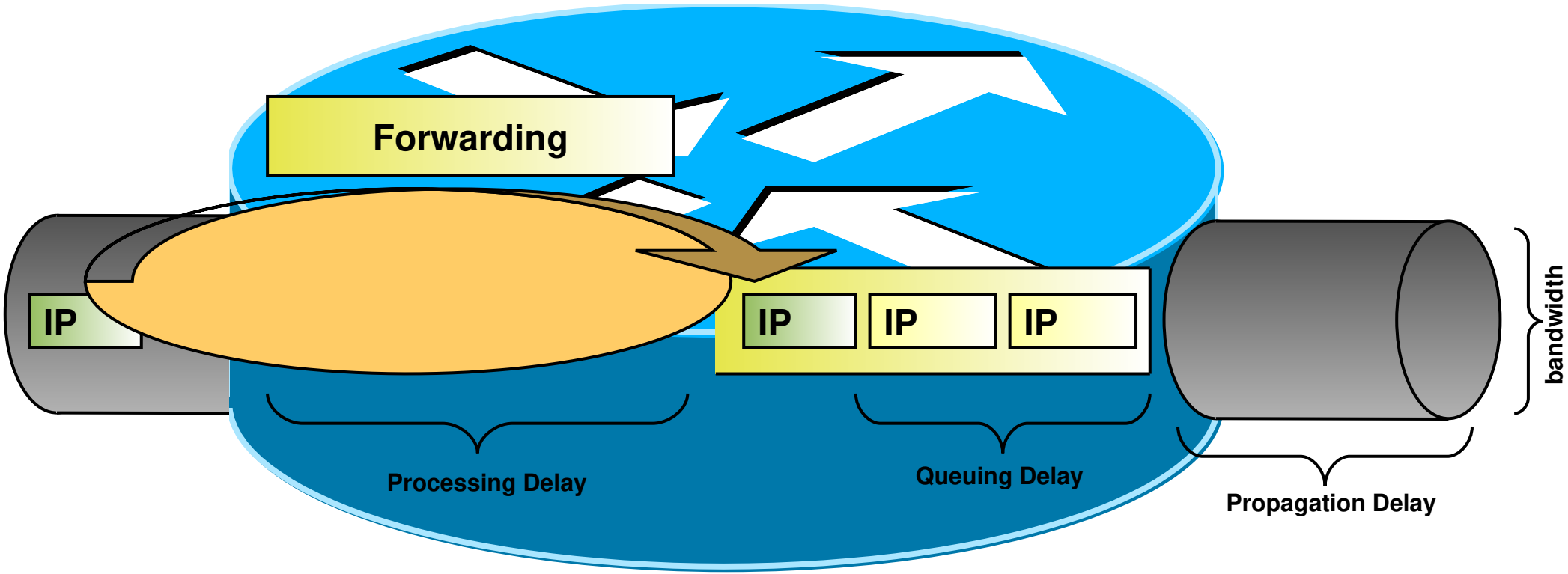
End-to-end Delay



Delay = P1 + Q1 + P2 + Q2 + P3 + Q3 + P4 = X ms

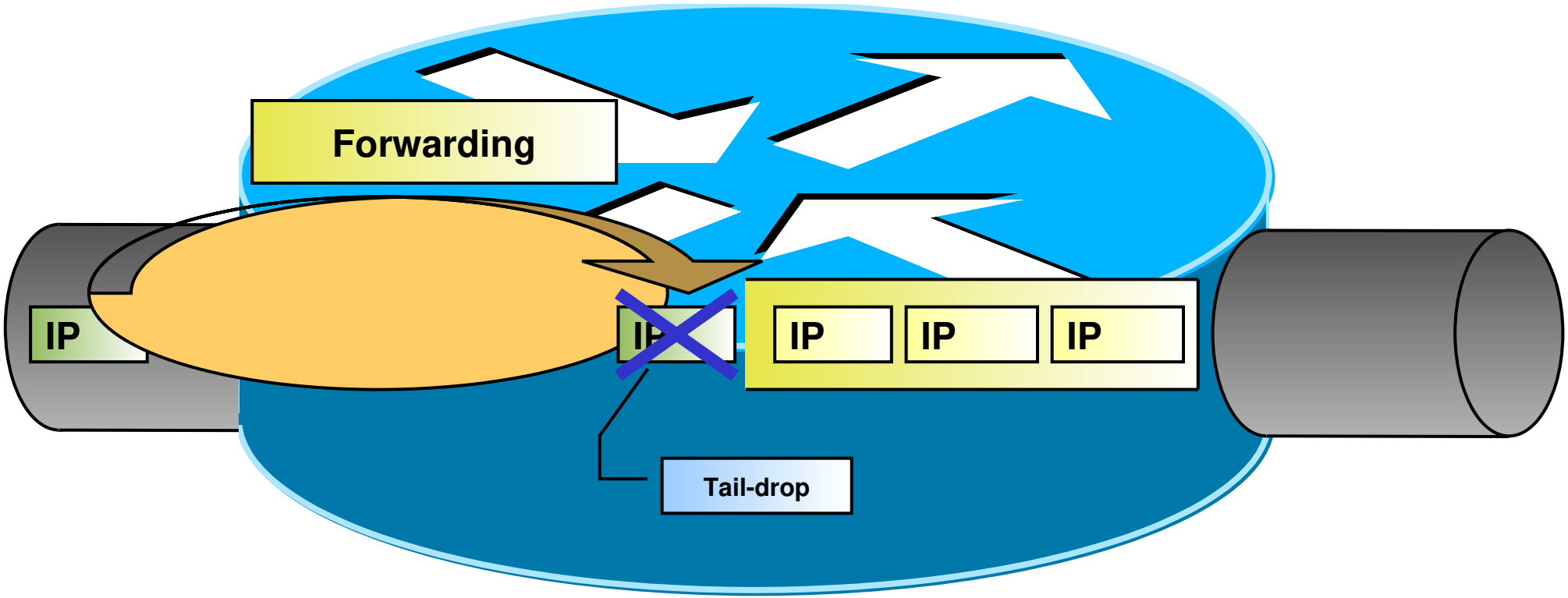
- End-to-end delay equals a sum of all propagation, processing and queuing delays in the path
- Propagation delay is fixed, processing and queuing delays are unpredictable in best-effort networks

Processing and Queuing Delay



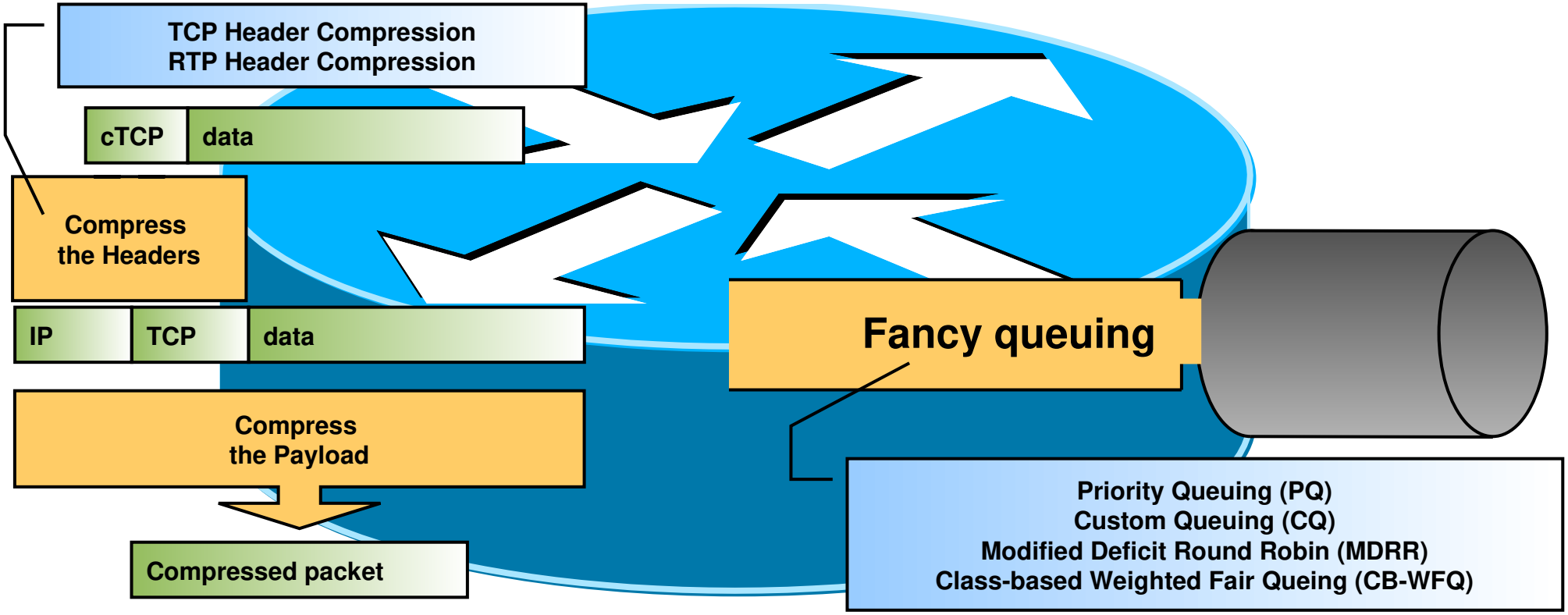
- Processing Delay is the time it takes for a router to take the packet from an input interface and put it into the output queue of the output interface.
- Queuing Delay is the time a packets resides in the output queue of a router.
- Propagation or Serialization Delay is the time it takes to transmit a packet.

Packet Loss



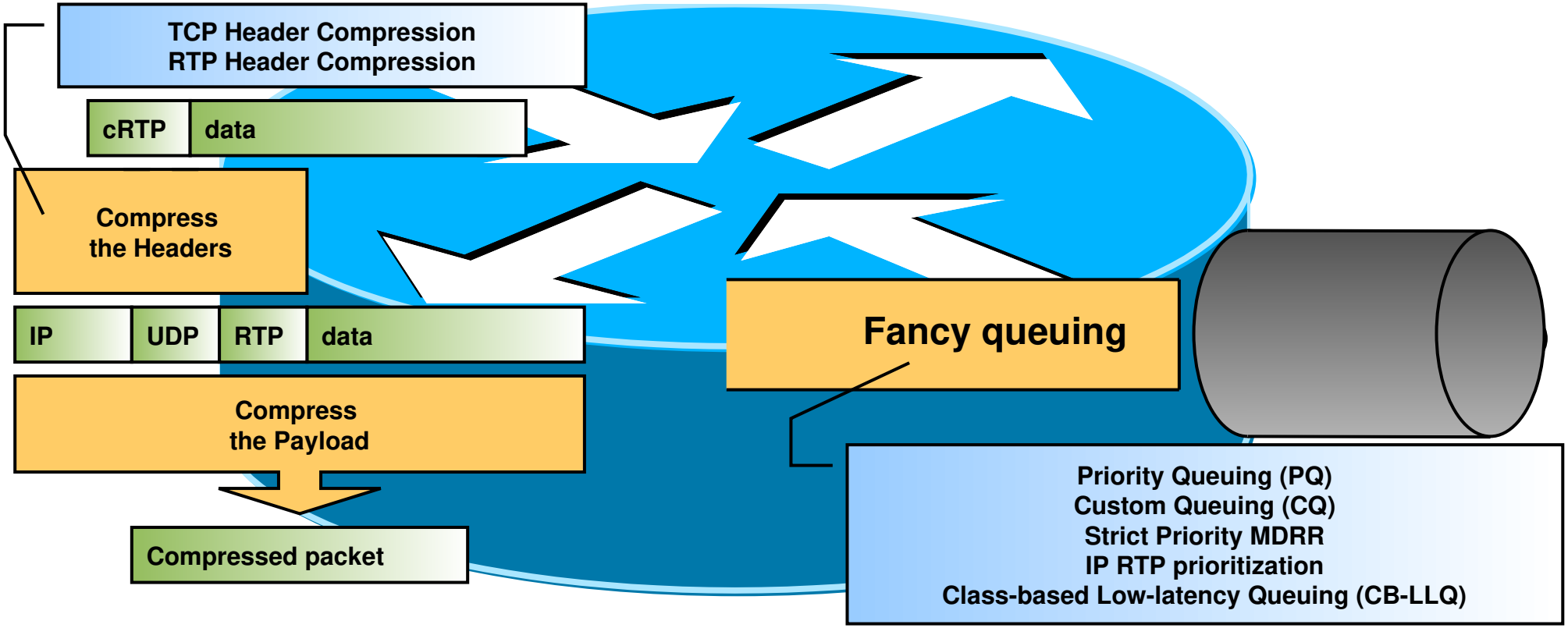
- Tail-drops occur when the output queue is full. These are the most common drops which happen when a link is congested.
- There are also many other types of drops that are not as common and may require a hardware upgrade (input drop, ignore, overrun, no buffer, ...). These drops are usually a result of router congestion.

How to Increase Available Bandwidth?



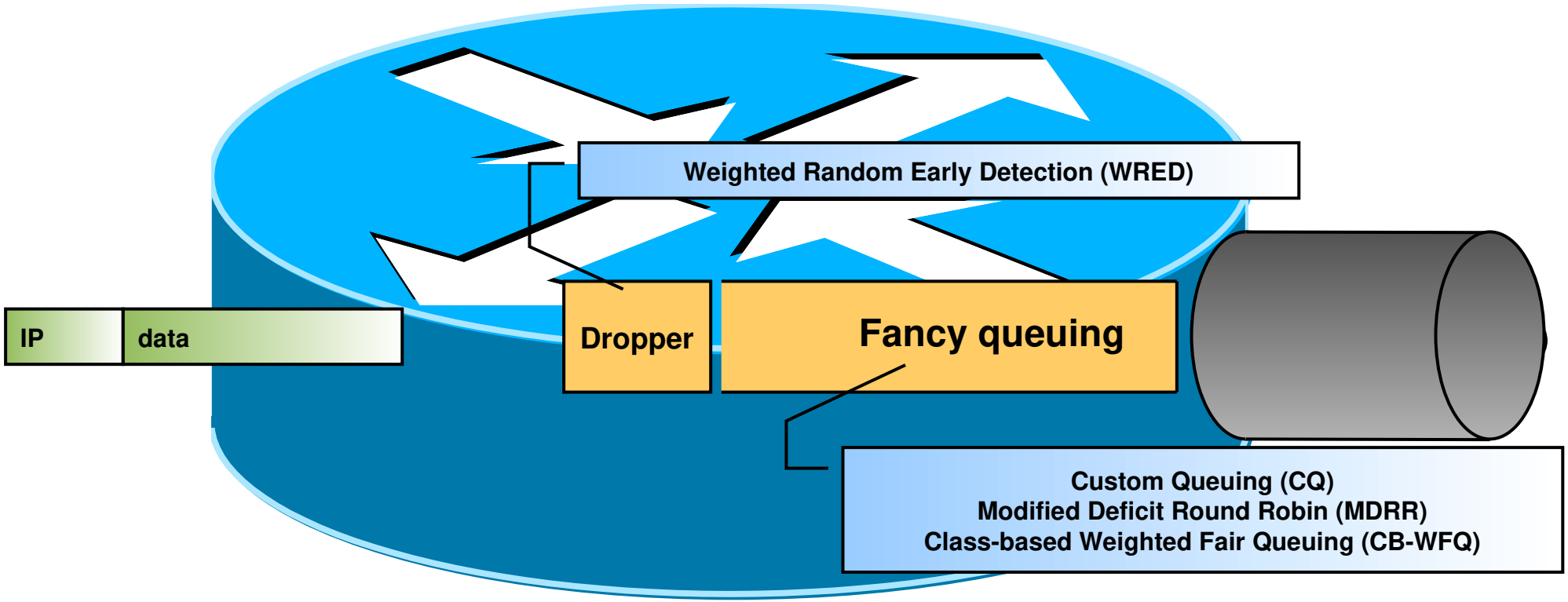
- Upgrade the link. The best solution but also the most expensive.
- Take some bandwidth from less important applications.
- Compress the payload of layer-2 frames.
- Compress the header of IP packets.

How to Reduce Delay?



- Upgrade the link. The best solution but also the most expensive.
- Forward the important packets first.
- Compress the payload of layer-2 frames (it takes time).
- Compress the header of IP packets.

How to Prevent Packet Loss?



- Upgrade the link. The best solution but also the most expensive.
- Guarantee enough bandwidth to sensitive packets.
- Prevent congestion by randomly dropping less important packets before congestion occurs

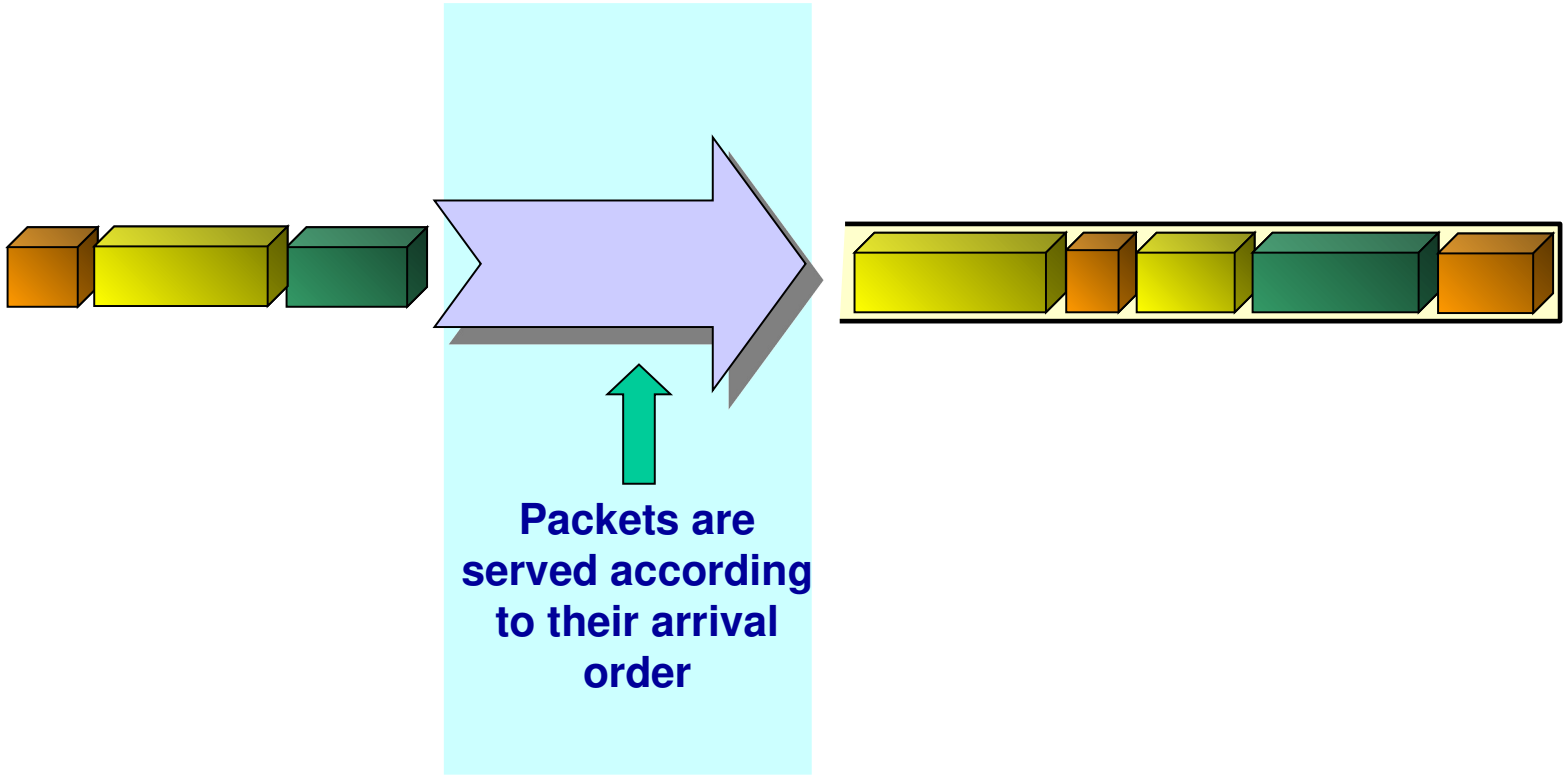
Traffic Terminology

- Flow: a single instance of an application-to-application flow of packets which is identified by source address, source port, destination address, destination port and protocol ID.
- Traffic stream: an administratively significant set of one or more flows which traverse a path segment. A traffic stream may consist of a set of active flows which are selected by a particular classifier.
- Traffic profile: a description of the temporal properties of a traffic stream such as average and peak rate and burst size.

Scheduling/Queuing algorithms

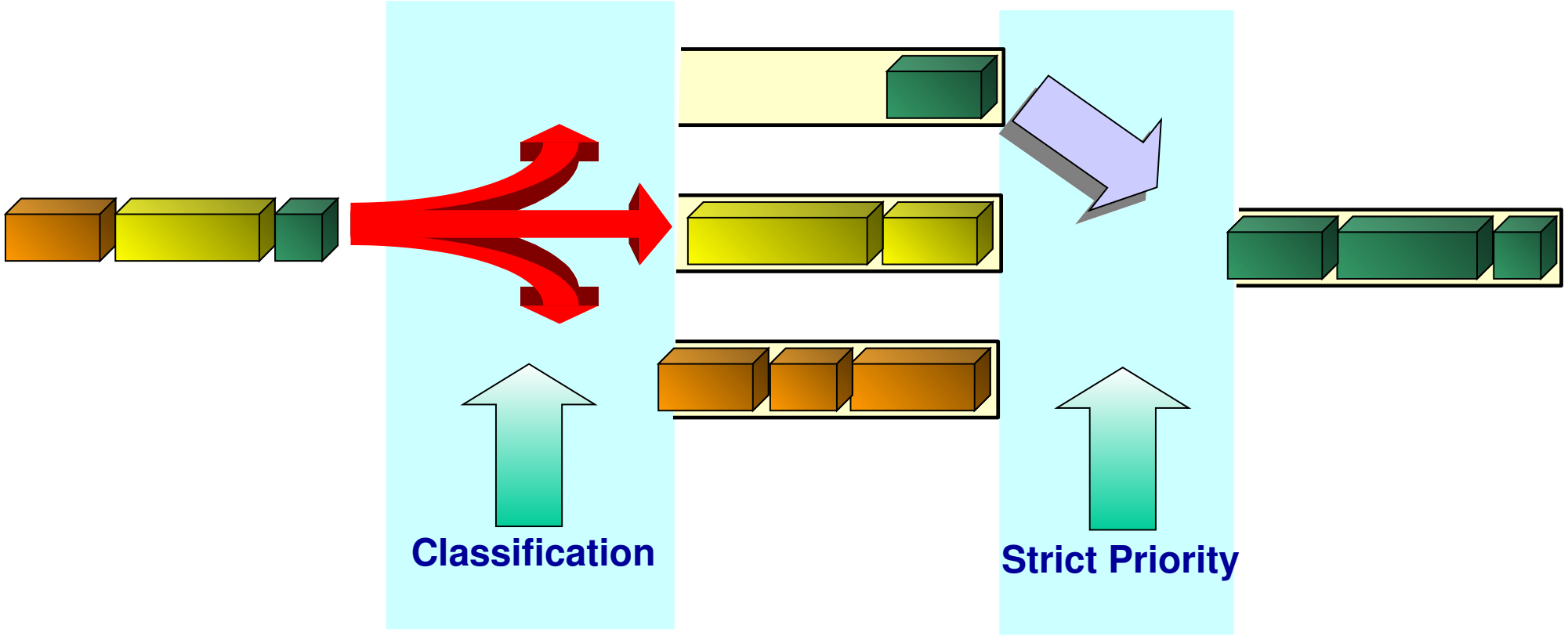
- Scheduling algorithms: decide the order packets from different flows are served in a queue
- Work conserving scheduling algorithms guarantee that the server is not occupied if and only if there is no packets waiting to be served
- Examples of work conserving scheduling algorithms:
  - FIFO
  - Strict priority (priority queuing)
  - Fair Queuing
  - Weighted Fair Queuing

First In First Out (FIFO)



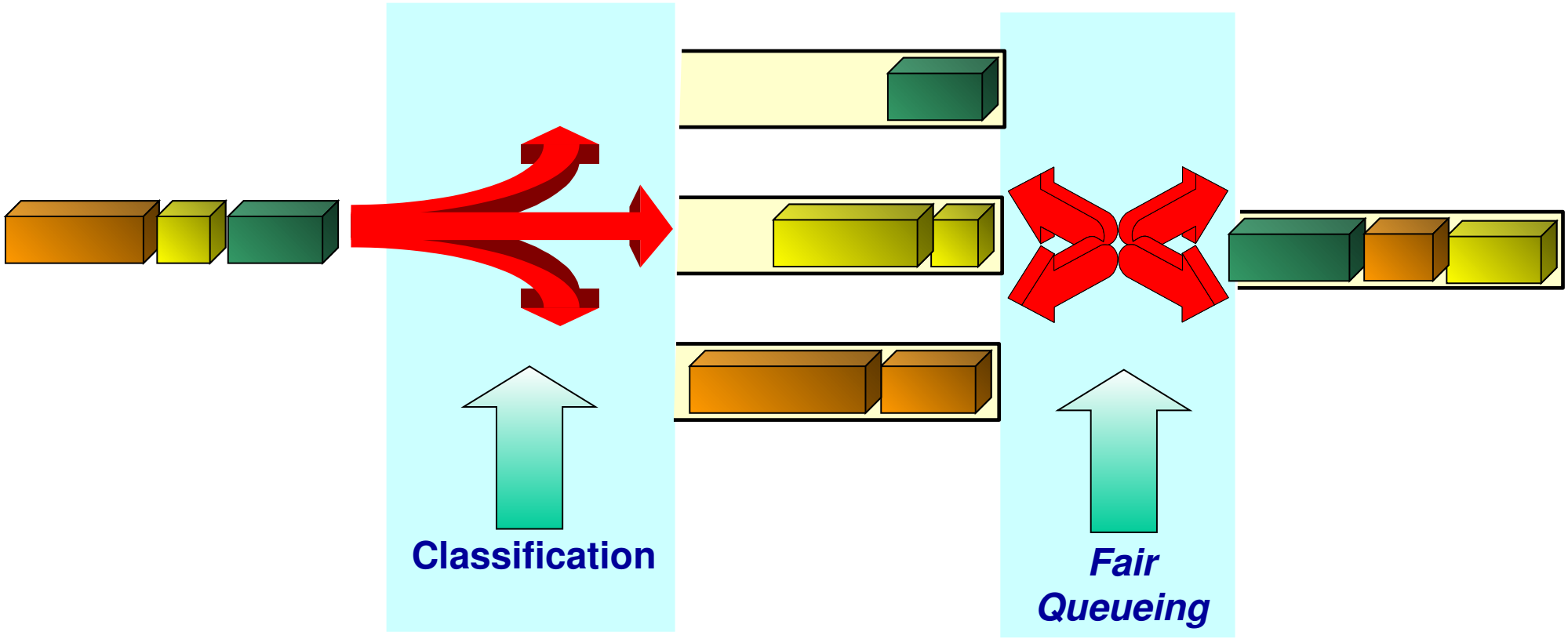
- Does not involve any ordering processing.
- Does not enable QoS differentiation.
- Flows having n times more traffic receive n times more service.
- On finite length queues, flows having smaller size packets receive more service.

Strict priority (Priority Queuing)



- Involves traffic classification according to priority.
- Higher priority traffic is always served before lower priority traffic.
- Enables QoS differentiation.
- Higher priority flows can prevent lower priority flows from receiving any service.

Fair Queueing (FQ)



- Involves traffic classification on different queues.
- Transmission bandwidth is equally distributed over non-empty queues.
- Enables QoS assignment.

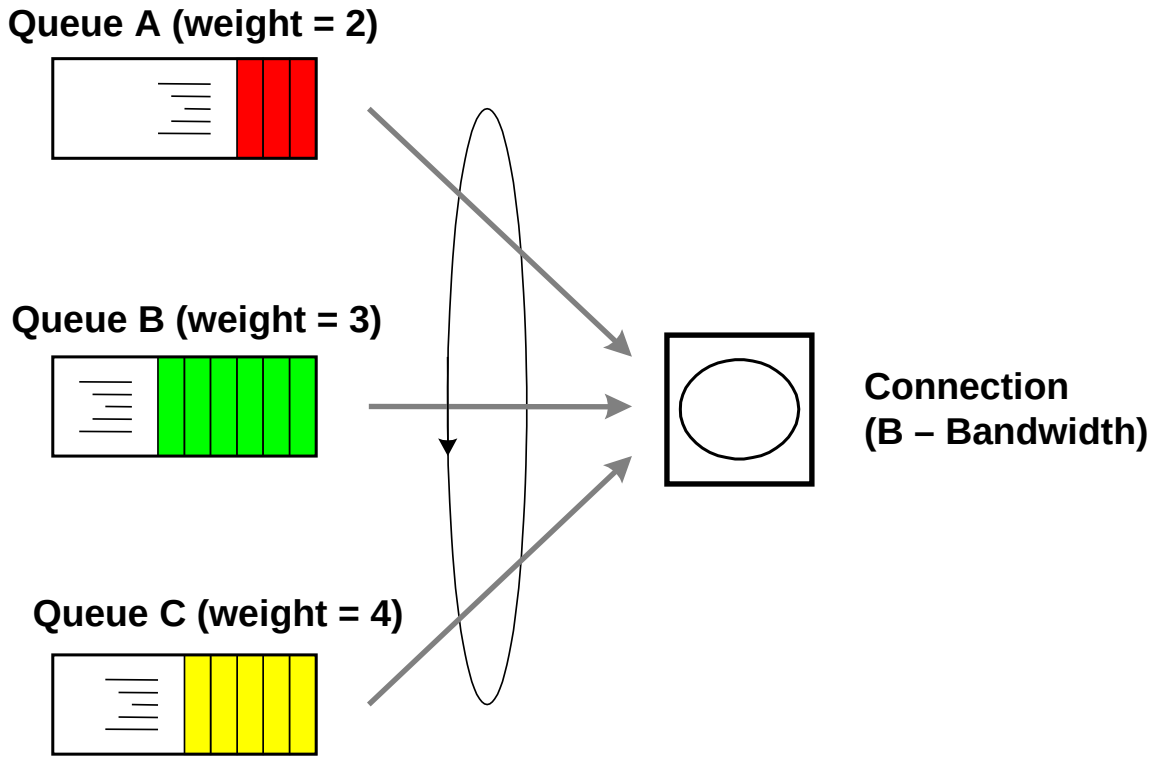
Weighted Fair Queuing (WFQ)

- This algorithm guarantees that each queue gets a percentage of the connection bandwith that is, al least, equal to its weight divided by the sum of all queues' weights

$$R_A = \frac{2}{2+3+4} B$$

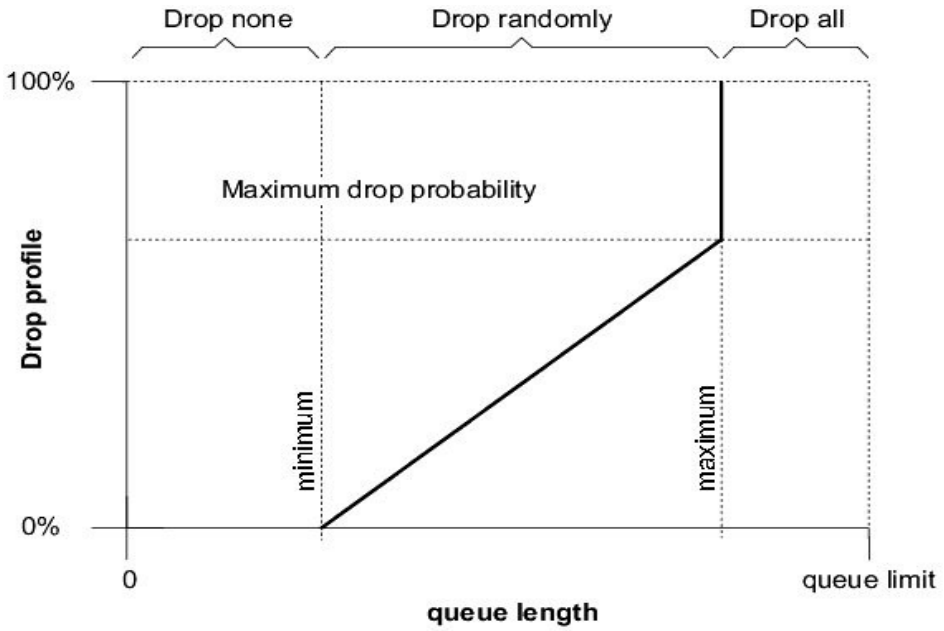
$$R_B = \frac{3}{2+3+4} B$$

$$R_C = \frac{4}{2+3+4} B$$



Dropper Mechanisms

- Random Early Detection (RED)
  - Congestion avoidance mechanism that takes advantage of TCP's congestion control
  - Randomly drops packets prior to periods of high congestion
  - Indirectly (by TCP) tells the packet source to decrease its transmission rate.
- Weighted RED (WRED)
  - Drops packets selectively based on priority classes
  - Higher priority traffic is delivered with a higher probability than lower priority traffic



How can QoS be Applied?

- Best effort – no QoS is applied to packets (default behavior)
- Integrated Services model – applications signal to the network that they require special QoS
- Differentiated Services model – the network recognizes classes that require special QoS

“Integrated *Services*” Architecture

Integrated Services (IntServ) Architecture

- The Integrated Services model (RFC1633) was introduced to guarantee a predictable behavior of the network for these applications
- For flows requiring Quality of Service, it is necessary to perform resource reservation on the flow paths between sources and destinations
  - Reservations are made flow-by-flow
- As opposed to the “best effort” service, the network implements a mechanism to control the admission of reservations (“call admission control”)
  - Flows that were not given any reservation are treated as “best effort” traffic

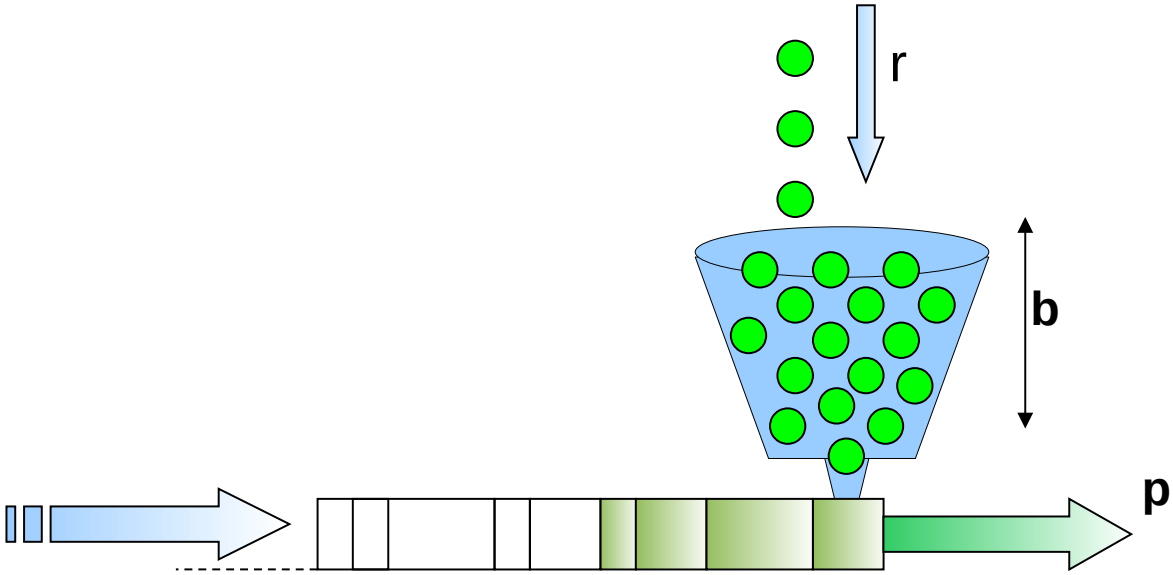
IntServ Classes of Service

- *Controlled Load* (RFC 2211)
  - Provides a service very similar to the "*best effort*" service in a non-congestioned network
  - Terminal stations should feel that a high percentage of their packets is delivered with routers' *queuing delays* very close to zero
- *Guaranteed Service* (RFC 2212)
  - Provides a maximum delay for all IP packets

- Both services demand that the sender condition the packet sending process according to a "*token bucket*" model

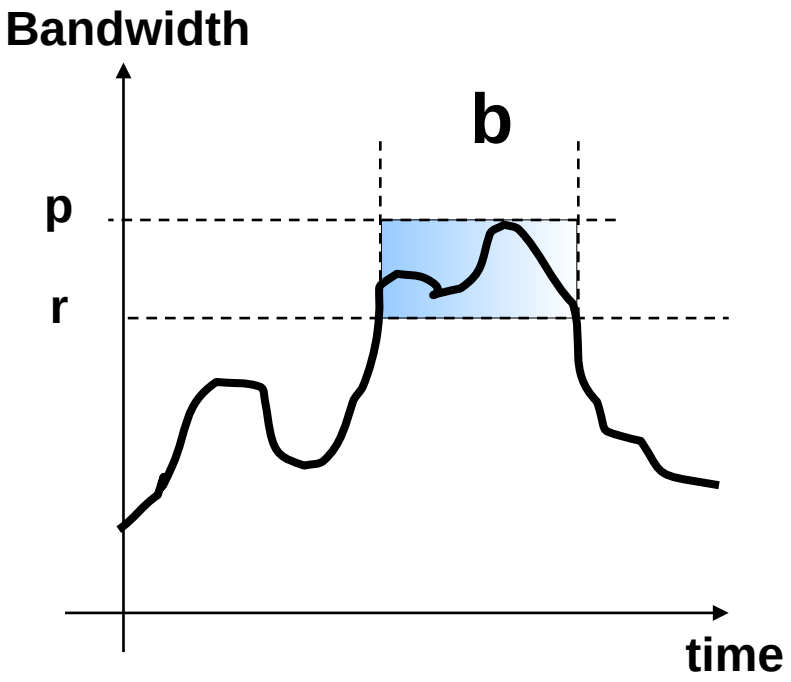
Traffic Characterization at the Sender

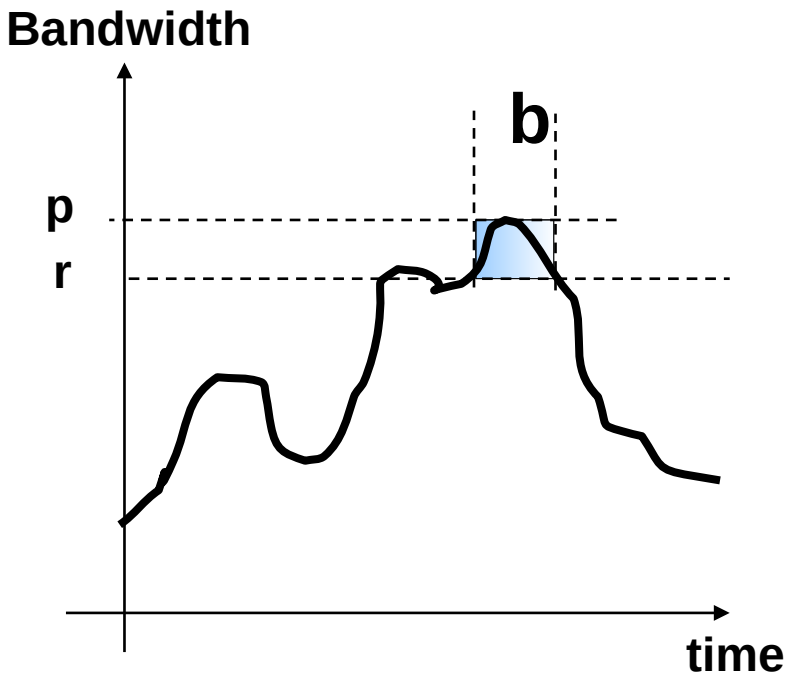
Token Bucket model



- **r = token filling rate (bytes/s)**
- **b = bucket size (bytes)**
- **p = maximum transmission rate (bytes/s)**
- **M = maximum packet size (bytes)**
- **m = minimum packet size (bytes) – each packet having a lower size will be considered as a size *m* packet**

Traffic Characterization at the Sender

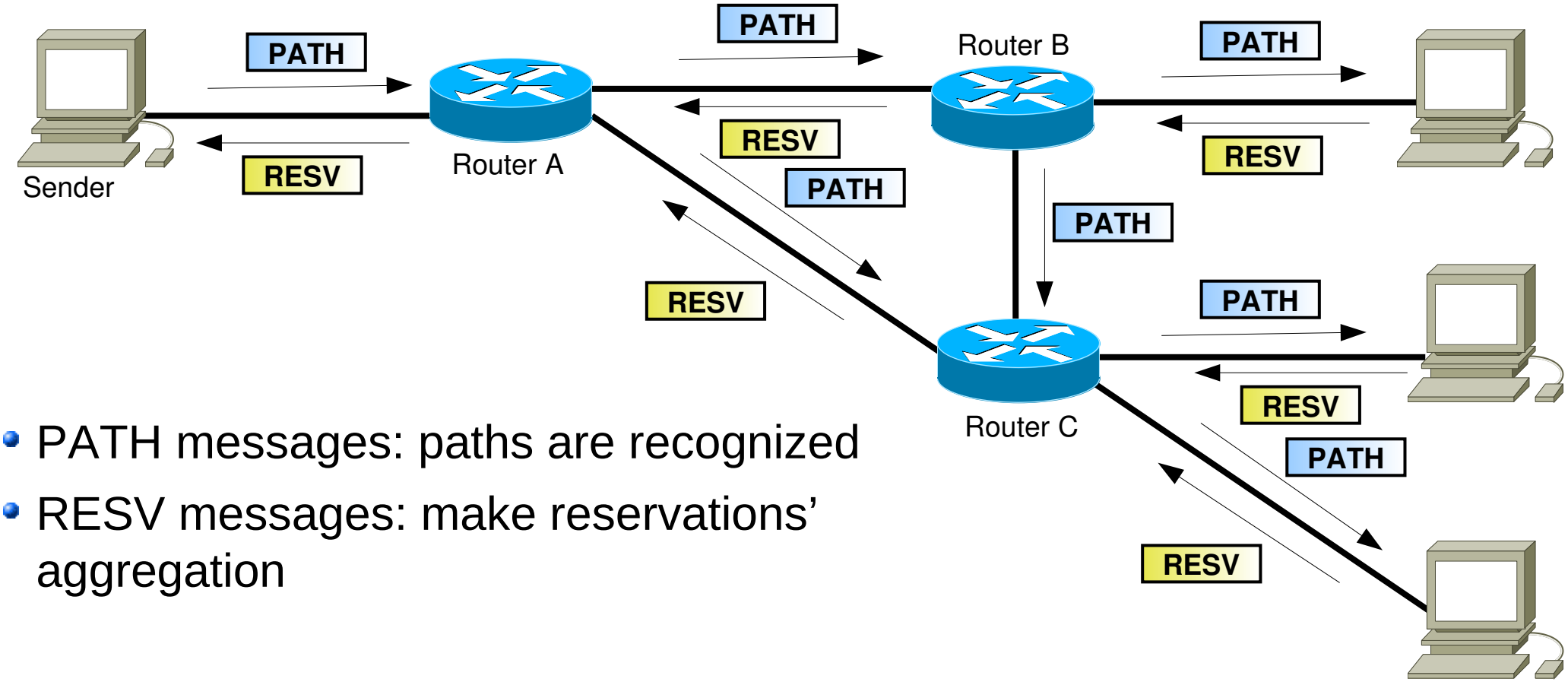




RSVP

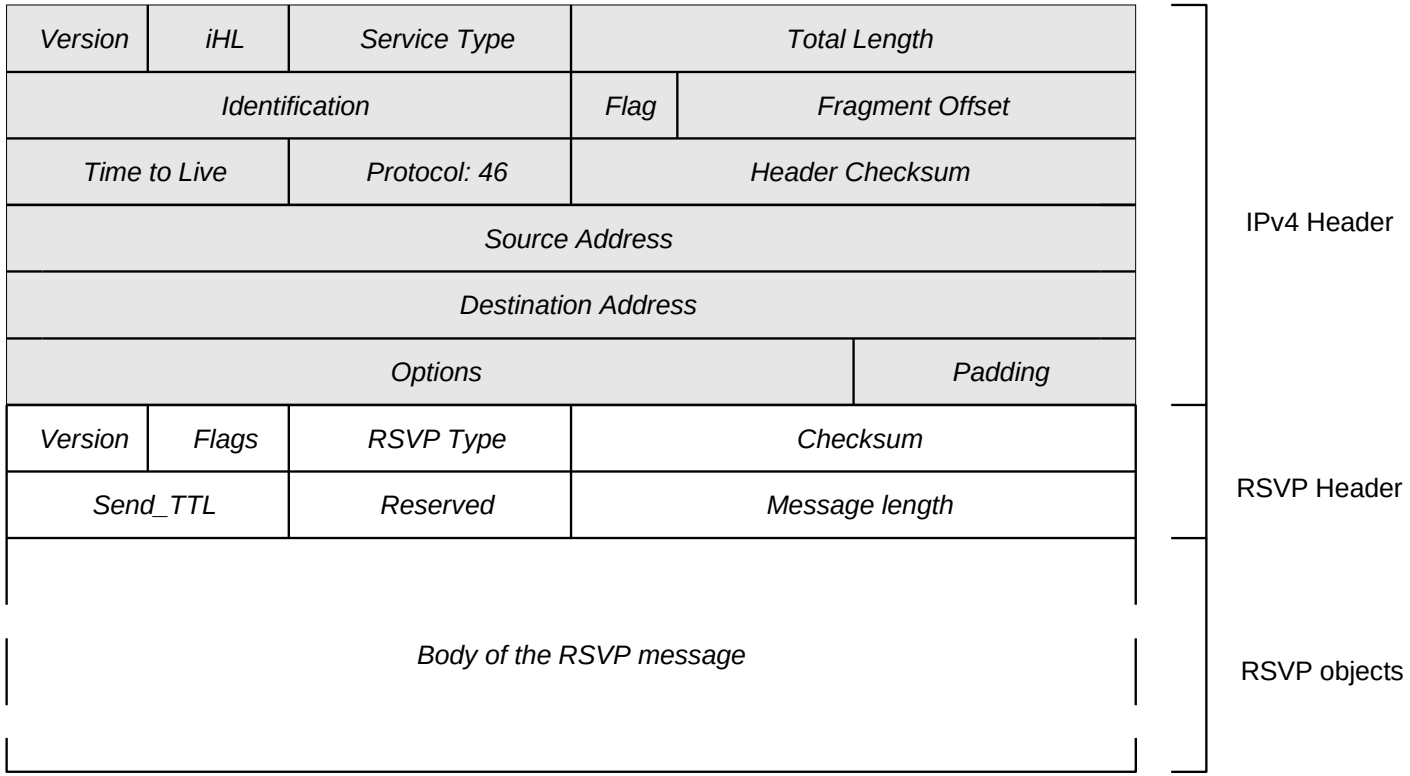
- The resource ReSerVation Protocol (RSVP) was developed to communicate resource needs between hosts and network devices (RFC 2205-2215)
- RSVP allows:
  - The source do describe the characteristics of the IP packets flow
  - Destinations to describe the reservation they want
  - Routers to know how to process the packets flow in order to fulfill the requested reservation
- Encapsulated on IP; protocol type = 46 (0x2E)
- Signaling is based on the exchange of PATH and RESV messages
  - PATH announces the traffic characteristics at the sender
  - RESV achieves reservations that were initiated by the receivers
  - If the reservation is not possible, a RESV ERR message is sent
- The routers reservation states have to be periodically refreshed (soft states)
- RSVP is typically used by applications carrying voice or video over IP networks (initiated by a host)
- RSVP with extensions is also used by MPLS Traffic Engineering to establish MPLS/TE tunnels (initiated by a router)

RSVP Signaling

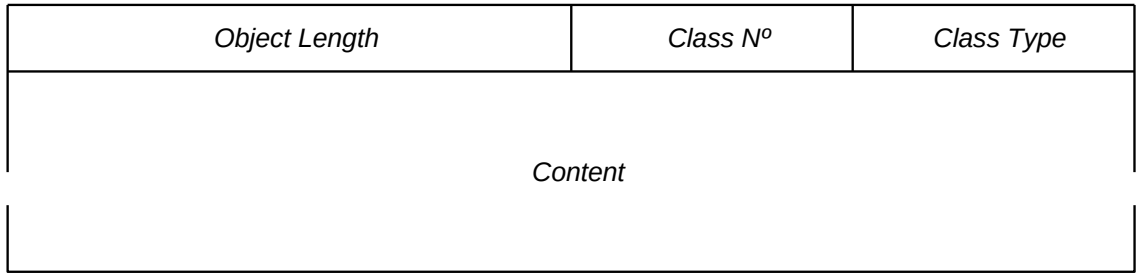


- PATH messages: paths are recognized
- RESV messages: make reservations' aggregation

Format of the RSVP messages



Format of each RSVP object:



RSVP messages

- PATH (*Type* = 0x01)
  - Tspec ("flow traffic specification"): contains the parameters that describe the traffic source based on the "Token Bucket" model
- RESV (*Type* = 0x02)
  - Tspec: the same that was received on the PATH message
  - FilterSpec ("*filter specification*"): contains the flow descriptor that enables routers to identify packets belonging to this reservation (source address, destination address, protocol type, source port number, destination port number, any combination of these parameters)
  - Rspec ("*flow reservation specification*"): contains the parameters describing the reservation that the receiver wants to become supported
    - Rspec is specified if the receiver wants a service of the "*guaranteed service*" type; when it is not specified, it means that the receiver wants a service of the "*controlled load*" type

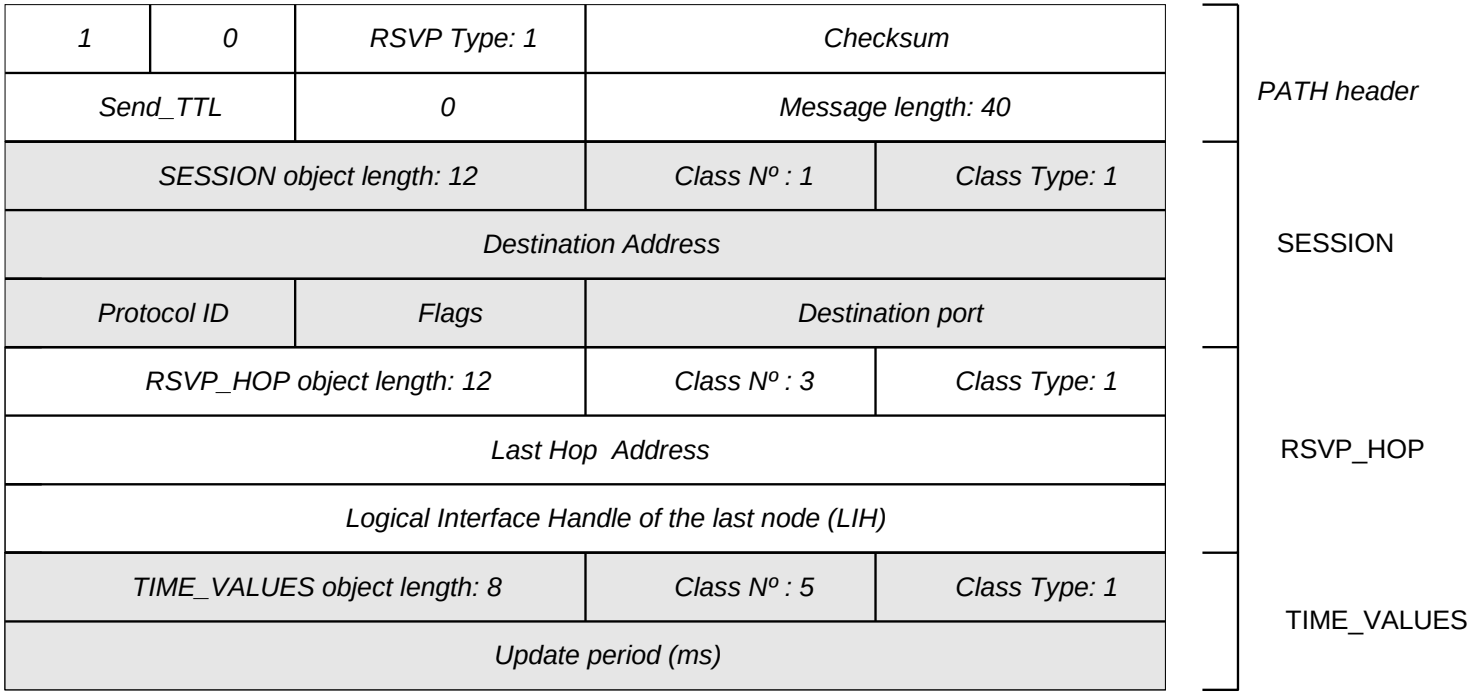
RSVP Parameters

RSVP Parameters	Description
TOKENBUCKETRATE (r)	TSpec: Rate of arriving tokens
TOKENBUCKETSIZE (b)	TSpec: Size of bucket
PEAKRATE (p)	TSpec: Maximum bit rate of the flow
MINIMUMPOLICEDUNIT (m)	TSpec: Minimum packet size considered
MAXIMUMPACKETSIZE (M)	TSpec: Maximum packet size
RATE (R)	RSpec*: Reservation rate
DELAYSLACKTERM	RSpec*: Tolerance of the requested delay

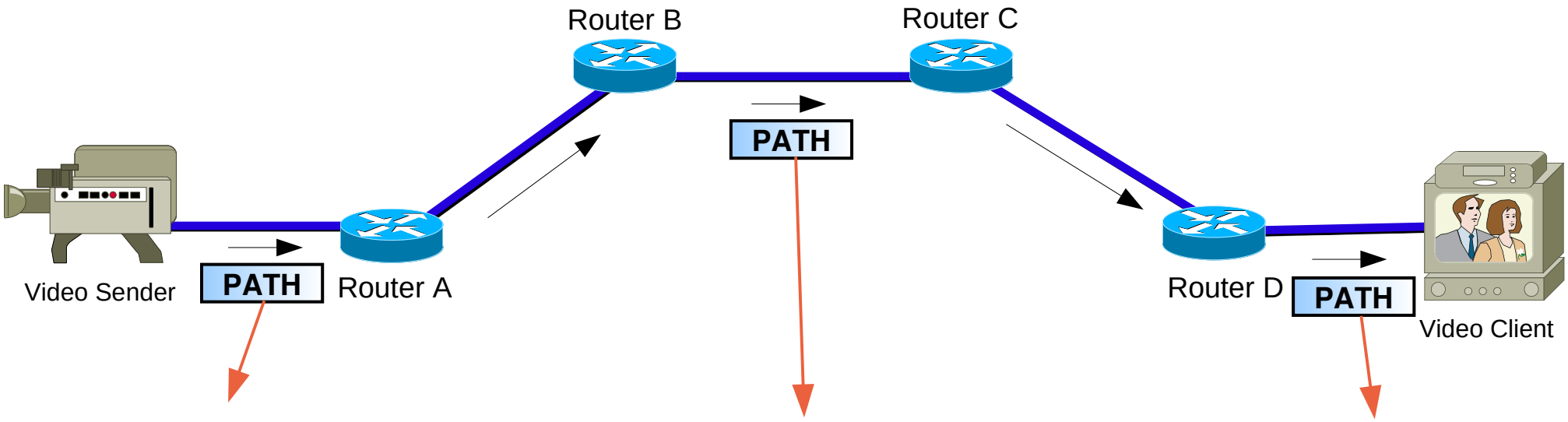
* RSpec is specified only for *Guaranteed Services*

RSVP PATH

- This message includes three mandatory RSVP objects (besides FLOWSPEC):
- SESSION – Identifies the session by the destination IP address, destination port and protocol ID
- RSVP_HOP – Indicates to the next router the sending IP address and port
- TIME_VALUES – Indicates the time period between successive sendings of PATH messages



RSVP PATH (Example)



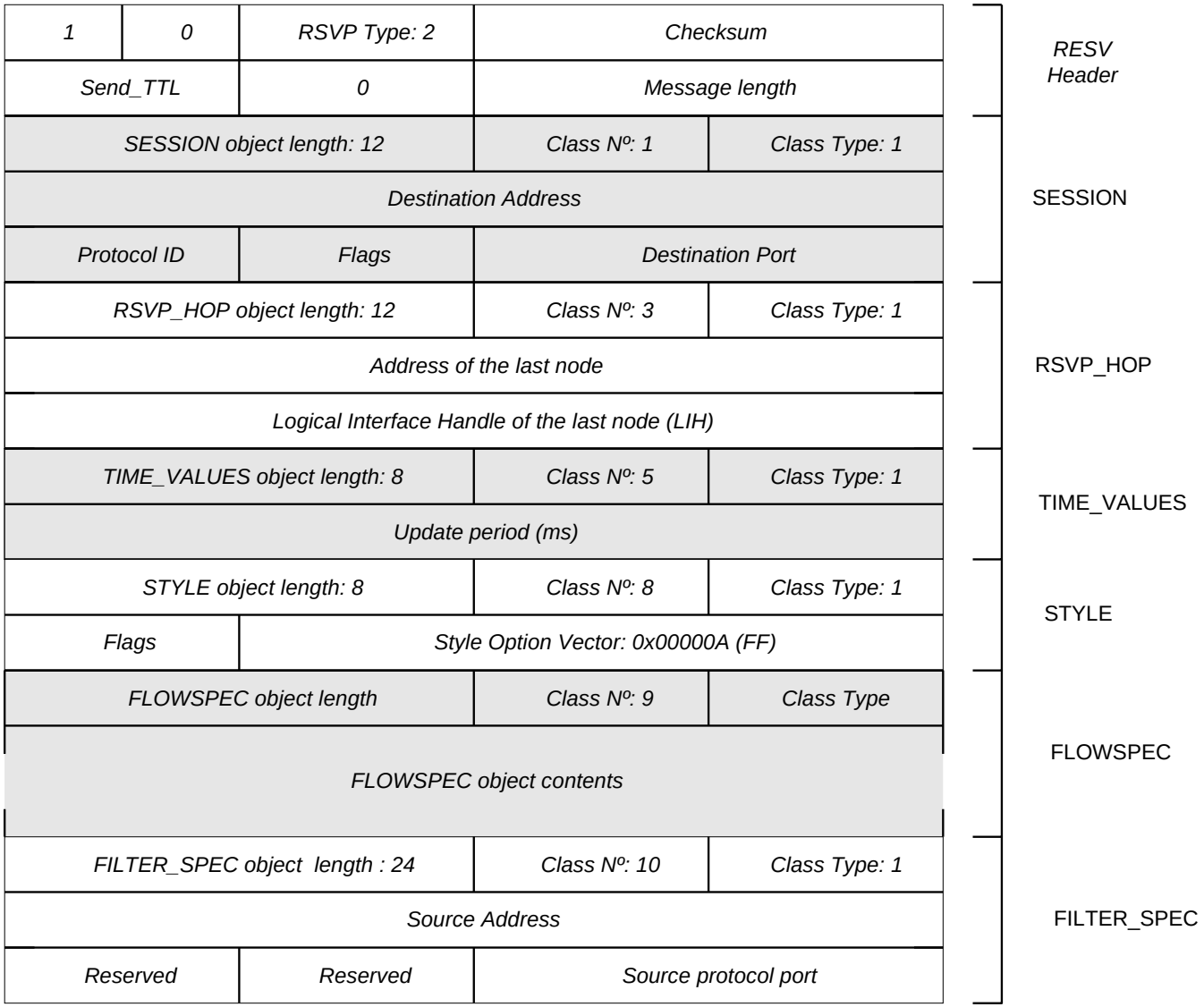
Vs.: 4	iHL: 5	Service	Total Length: 60
Identification		Flg	Fragment Offset
Time to Live	Protocol: 46		Header Checksum
Source Address: **Video Server**			
Destination Address: **Video Client**			
1	0	Type: 1	Checksum
Send_TTL	0		Message Length: 40
SESSION Length.: 12		Class Nº: 1	Class Type: 1
Destination Address: **Video Client**			
Protocol ID	Flags		Destination port
RSVP_HOP Length. : 12		Class Nº: 3	Class Type: 1
Last Hop Address: **Video Server**			
Logical Interface Handle of the last node (LIH)			
TIME_VALUES Length: 8		Class Nº: 5	Class Type: 1
Update Period (ms)			

Vs.: 4	iHL: 5	Service	Total Length: 60
Identification		Flg	Fragment Offset
Time to Live	Protocol: 46		Header Checksum
Source Address: **Video Server**			
Destination Address: **Video Client**			
1	0	Type: 1	Checksum
Send_TTL	0		Message Length: 40
SESSION Length: 12		Class Nº: 1	Class Type: 1
Destination Address: **Video Client**			
Protocol ID	Flags		Destination Port
RSVP_HOP Length: 12		Class Nº: 3	Class Type: 1
Last Hop Address: **Router B**			
Logical Interface Handle of the last node (LIH)			
TIME_VALUES Length: 8		Class Nº: 5	Class Type: 1
Update Period (ms)			

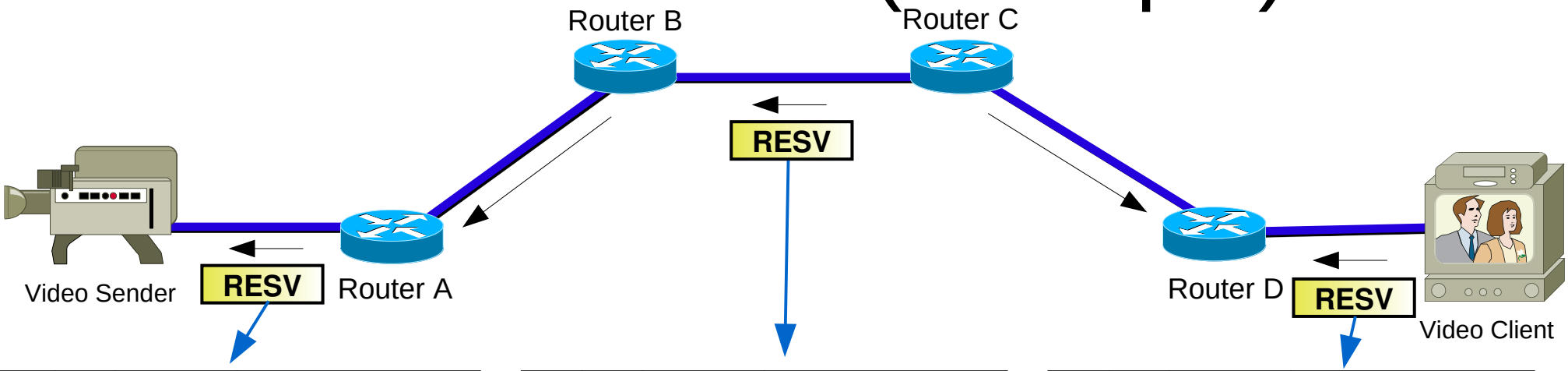
Vs.: 4	iHL: 5	Service	Total Length: 60
Identification		Flg	Fragment Offset
Time to Live	Protocol: 46		Header Checksum
Source Address: **Video Server**			
Destination Address: **Video Client**			
1	0	Type: 1	Checksum
Send_TTL	0		Message Length: 40
SESSION Length: 12		Class Nº: 1	Class Type: 1
Destination Address: **Video Client**			
Protocol ID	Flags		Destination Port
RSVP_HOP Length: 12		Class Nº: 3	Class Type: 1
Last Hop Address: **Router D**			
Logical Interface Handle of the last node (LIH)			
TIME_VALUES Length: 8		Class Nº: 5	Class Type: 1
Update Period (ms)			

RSVP RESV

- STYLE – Identifies the style of the reservation
- FLOWSPEC – Includes TSpec and RSpec
- FILTER_SPEC – Indicates the necessary information for the packet classifier



RSVP RESV (Example)



Vs.: 4	iHL: 5	Service	Total Length
Identification		Flg	Fragment Offset
Time to Live	Protocol: 46		Header Checksum
Source Address: **Router A**			
Destination Address: **Video Server**			
1	0	Type: 2	Checksum
Send_TTL	0		Message Length
SESSION Length: 12		Class Nº: 1	Class Type: 1
Destination Address: **Video Client**			
Protocol Id	Flags		Destination protocol port
RSVP_HOP Length: 12		Class Nº: 3	Class Type: 1
Address of the last node: **Router A**			
Logical Interface Handle of the last node (LIH)			
TIME_VALUES Length: 8		Class Nº: 5	Class Type: 1
Update period (ms)			
STYLE Object Length : 8		Class Nº: 8	Class Type: 1
Flags	Style Option Vector: 0x00000A (FF)		
FLOWSPEC Length		Class Nº: 9	Class Type
FLOWSPEC object contents			
FILTER_SPEC Length: 12		Class Nº: 10	Class Type: 1
Source Address: Video Server			
Reserved	Reserved		Source protocol port

Vs.: 4	iHL: 5	Service	Total Length
Identification		Flg	Fragment Offset
Time to Live	Protocol: 46		Header Checksum
Source Address: **Router C**			
Destination Address: **Router B**			
1	0	Type: 2	Checksum
Send_TTL	0		Message Length
SESSION Length: 12		Class Nº: 1	Class Type: 1
Destination Address: **Video Client**			
Protocol Id	Flags		Destination protocol port
RSVP_HOP Length: 12		Class Nº: 3	Class Type: 1
Address of the last node: **Router C**			
Logical Interface Handle of the last node (LIH)			
TIME_VALUES Length: 8		Class Nº: 5	Class Type: 1
Update period (ms)			
STYLE Object Length : 8		Class Nº: 8	Class Type: 1
Flags	Style Option Vector: 0x00000A (FF)		
FLOWSPEC Length		Class Nº: 9	Class Type
FLOWSPEC object contents			
FILTER_SPEC Length: 12		Class Nº: 10	Class Type: 1
Source Address: Video Server			
Reserved	Reserved		Source protocol port

Vs.: 4	iHL: 5	Service	Total Length
Identification		Flg	Fragment Offset
Time to Live	Protocol: 46		Header Checksum
Source Address: **Video Client**			
Destination Address: **R outer D**			
1	0	Type: 2	Checksum
Send_TTL	0		Message Length
SESSION Length: 12		Class Nº: 1	Class Type: 1
Destination Address: **Video Client**			
Protocol Id	Flags		Destination protocol port
RSVP_HOP Length: 12		Class Nº: 3	Class Type: 1
Address of the last node: **Video Client**			
Logical Interface Handle of the last node (LIH)			
TIME_VALUES Length: 8		Class Nº: 5	Class Type: 1
Update period (ms)			
STYLE Object Length : 8		Class Nº: 8	Class Type: 1
Flags	Style Option Vector: 0x00000A (FF)		
FLOWSPEC Length		Class Nº: 9	Class Type
FLOWSPEC object contents			
FILTER_SPEC Length: 12		Class Nº: 10	Class Type: 1
Source Address: Video Server			
Reserved	Reserved		Source protocol port

RSVP Reservation Styles

- “Fixed Filter” (Style Option Vector = 0x00000A)
  - The receiver specifies a reservation value for each sender
- “Wildcard Filter” (Style Option Vector = 0x000011)
  - The receiver specifies a unique reservation value to receive traffic from any sender
- “Explicit Filter” (Style Option Vector = 0x000012)
  - The receiver specifies a list of senders from which it wants to receive information and a unique reservation value to receive traffic from the specified senders

- On RSVP RESV messages:
  - The reservation style is declared by the STYLE object
  - Senders are declared on the FILTER_SPEC object

Other RSVP messages

- PATH ERR (*Type* = 0x03):
  - Sent by routers in error situations
- RESV ERR (*Type* = 0x04):
  - Sent by routers when a reservation cannot be supported
- PATH TEAR (*Type* = 0x05):
  - Sent by senders when information they finish transmitting information
- RESV TEAR (*Type* = 0x06):
  - Sent by receivers when they do not want a reservation anymore
- RESV CONFIRMATION (*Type* = 0x07):
  - Sent by routers to confirm the establishment of a reservation

RSVP characteristics

- Multipoint-multipoint model (simplex)
- Reservations initiated by the receivers
- Temporized reservations (soft state)
- Separation between reservation and routing
- Separation between reservation and packet filtering
- Different reservation styles
- Aggregation of reservations

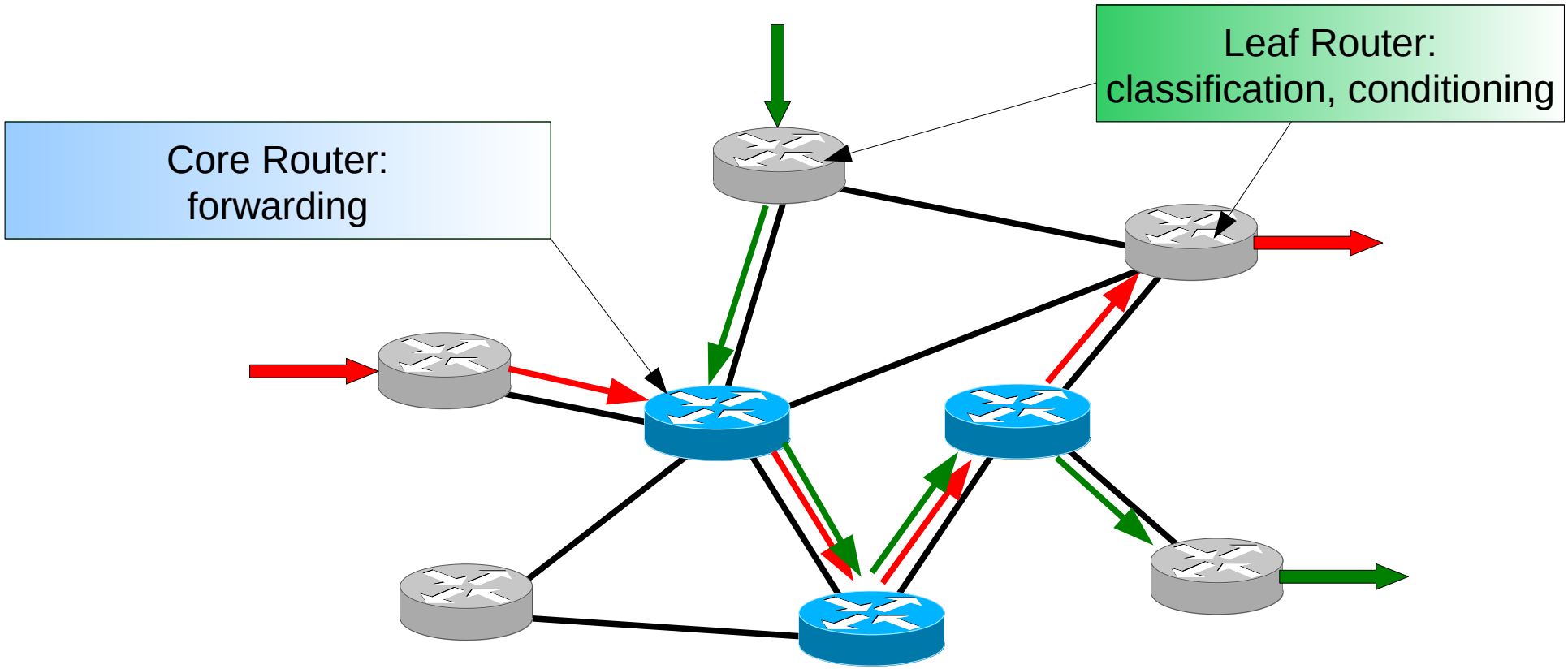
“*Differentiated Services*” Architecture

Differentiated Services (*DiffServ*) Architecture

- Problems of the *Integrated Services* architecture:
  - Routers maintain information about the state of the end-to-end reservations
    - Poor scalability
  - Routers determine attendance order based on multiple fields (source and destination addresses, protocol, source and destination port)
    - Penalizes performance
  - Supports only two service classes: "*controlled load*" and "*guaranteed service*"
    - Low flexibility
  - Demands end-to-end RSVP signaling
    - High reservation establishment times
- Thus, DiffServ architecture:
  - By contract, the traffic flow from each client is classified as belonging to a particular class
  - Treats flow classes that demand the same Quality of Service
  - At the network entrance, packets are marked as belonging to the contracted class and packet scheduling is based on the packet mark

Basic ideas

- Implement simple routing operations on the network *core routers* and leave complex operations to the network *edge routers*.
- It only defines functional elements that enable the support of any service class.



Differentiated Services Model

- Differentiated Services model describes services associated with traffic classes
- Complex traffic classification and conditioning is performed at network edge resulting in a per-packet Differentiated Services Code Point (DSCP).
- No per-flow/per-application state in the core
- Core only performs simple ‘per-hop behavior's’ on traffic aggregates
- Goal is scalability

DiffServ Elements

- The service defines QoS requirements and guarantees provided to a traffic aggregate;
- The conditioning functions and per-hop behaviors are used to realize services;
- The DS field value (DS Code Point) is used to mark packets to select a per-hop behavior
- Per-hop Behavior (PHB) is realized using a particular QoS mechanism
- Provisioning is used to allocate resources to traffic classes

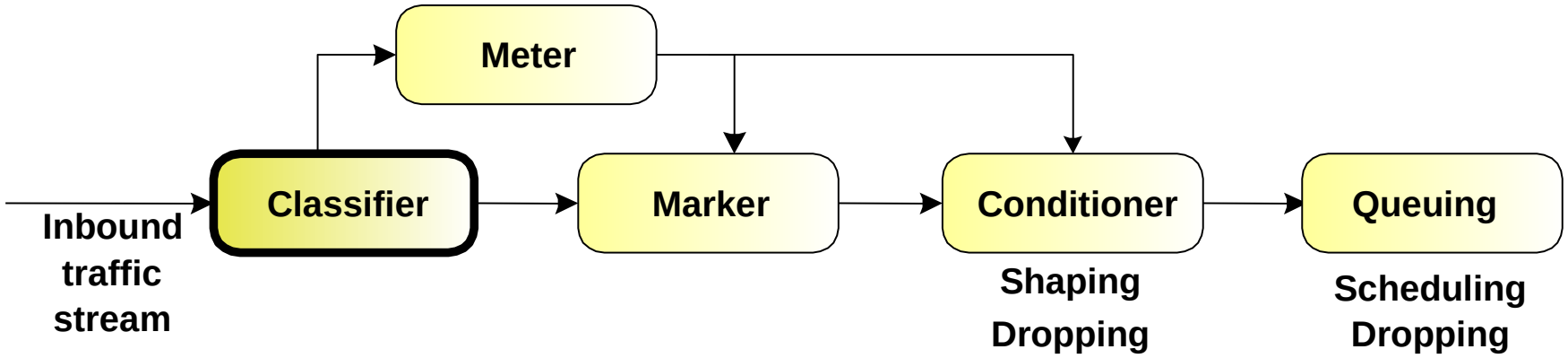
Traffic Terminology

- Behavior Aggregate (BA) is a collection of packets with the same DS code point crossing a link in a particular direction.
- Per-Hop Behavior (queuing in a node) externally observable forwarding behavior applied at a DS-compliant node to a DS behavior aggregate.
- PHB Mechanism: a specific algorithm or operation (e.g., queuing discipline) that is implemented in a node to realize a set of one or more per-hop behaviors.

DiffServ QoS Actions

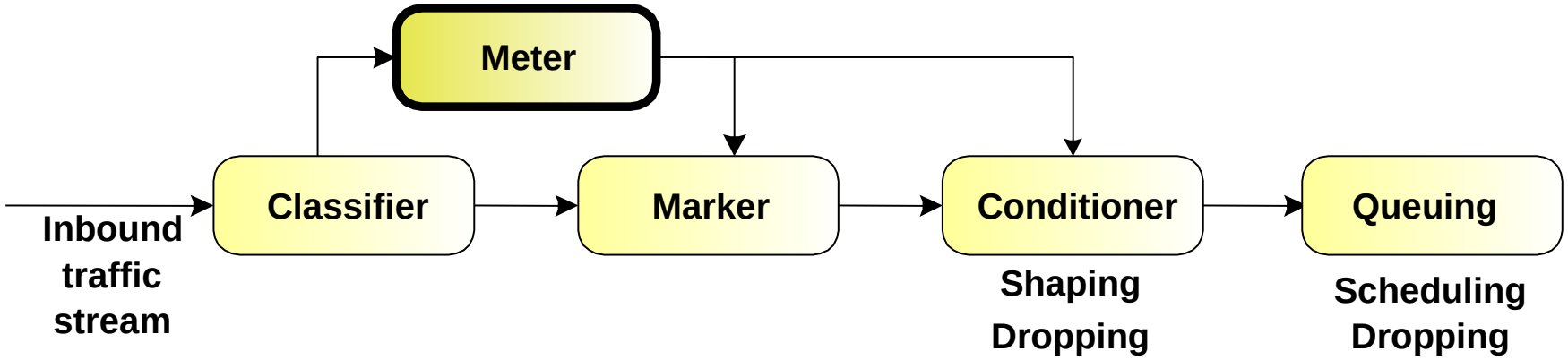
- Classification – Each class-oriented QoS mechanism has to support some type of classification (access lists, route maps, class maps, etc.)
- Metering – Some mechanisms measure the rate of traffic to enforce a certain policy (e.g. rate limiting, shaping, scheduling, etc.)
- Dropping – Some mechanisms are used to drop packets (e.g. random early detection)
- Policing – Some mechanisms are used to enforce a rate limit based on the metering (excess traffic is dropped)
- Shaping – Some mechanisms are used to enforce a rate limit based on the metering (excess traffic is delayed)
- Marking – Some mechanisms have the capability to mark packets based on classification and/or metering (e.g. CAR, class-based marking, etc.)
- Queuing – Each interface has to have a queuing mechanism
- Forwarding – There are several supported forwarding mechanisms (process switching, fast switching, CEF switching, etc.)

DiffServ Mechanisms



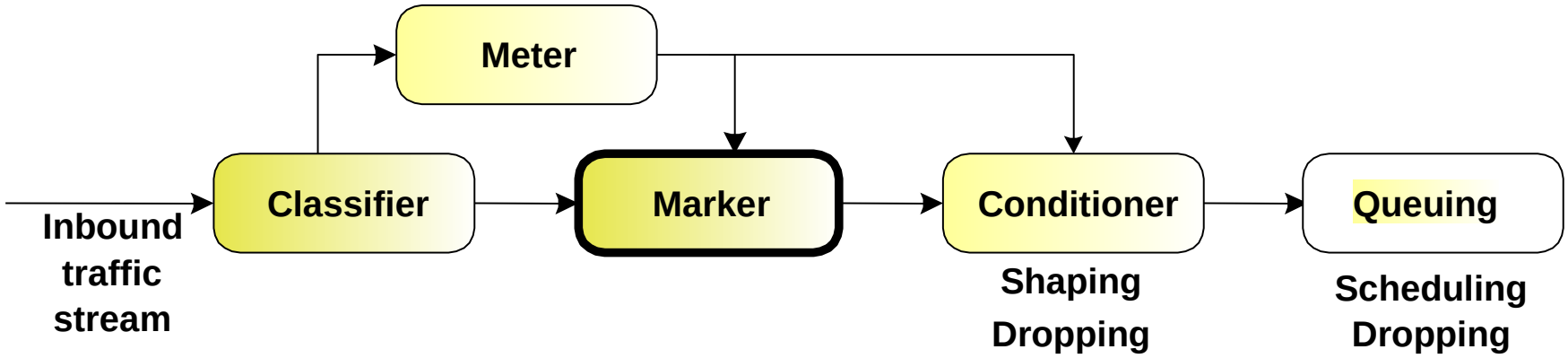
- Most traditional QoS mechanisms include extensive built-in classifiers
  - Committed Access Rate (CAR)
  - QoS Policy Propagation via BGP (QPPB)
  - Route-maps
  - Queuing mechanisms
  - ...

DiffServ Mechanisms



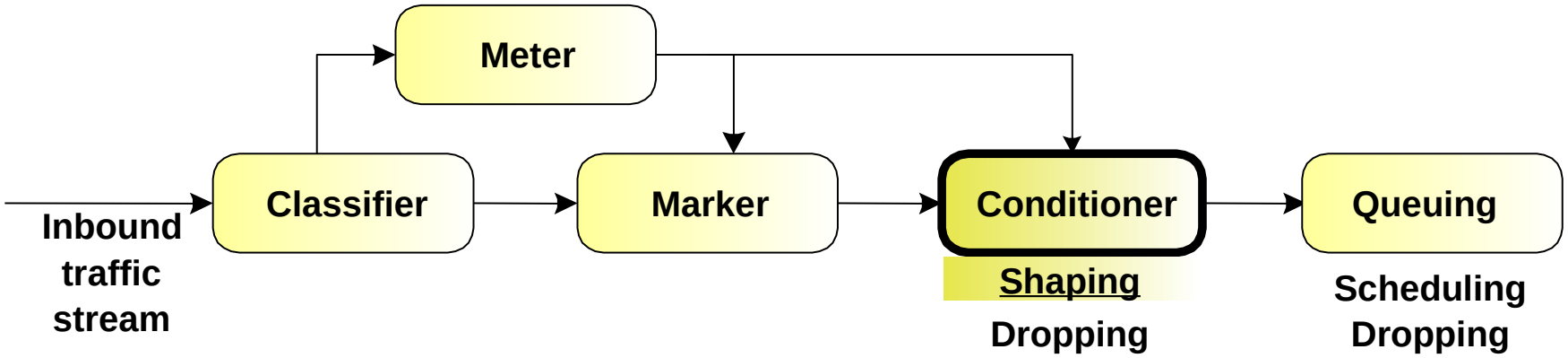
- Token Bucket model is used for metering
  - Committed Access Rate (CAR)
  - Generic Traffic Shaping (GTS)
  - Frame Relay Traffic Shaping (FRTS)
  - Class-based Weighted Fair Queuing (CB-WFQ)
  - Class-based Low Latency Queuing (CB-LLQ)
  - Class-based Policing
  - Class-based Shaping
  - IP RTP Prioritization

DiffServ Mechanisms



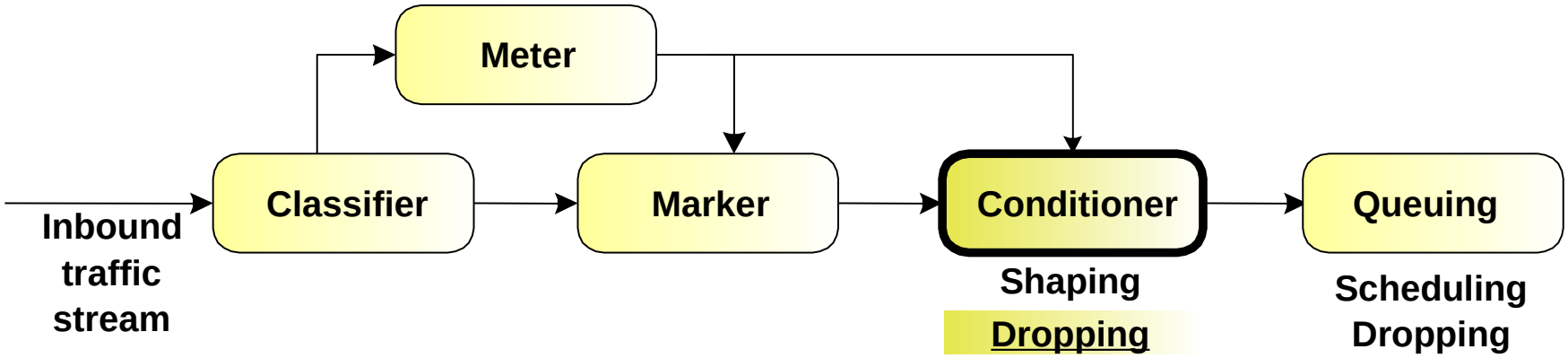
- Marker is used to set:
  - IP precedence
  - DSCP
  - QoS group
  - MPLS experimental bits
  - Frame Relay DE bit
  - ATM CLP bit
  - IEEE 802.1Q or ISL CoS
- Marking mechanisms:
  - Comitted Access Rate (CAR)
  - QoS Policy Propagation through BGP (QPPB)
  - Policy-based Routing (PBR)
  - Class-based Marking

DiffServ Mechanisms



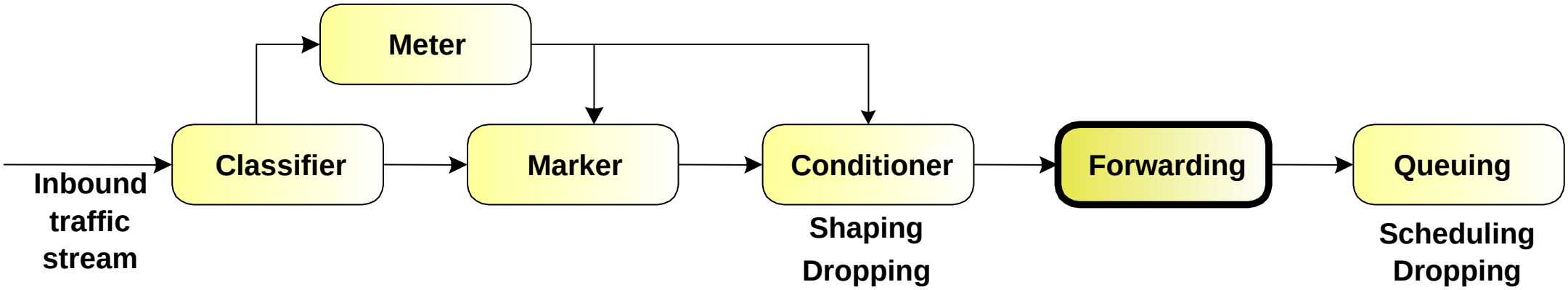
- Shaping mechanisms:
  - Generic Traffic Shaping (GTS)
  - Frame Relay Traffic Shaping (FRTS)
  - Class-based Shaping
  - Hardware shaping on ATM VC

DiffServ Mechanisms



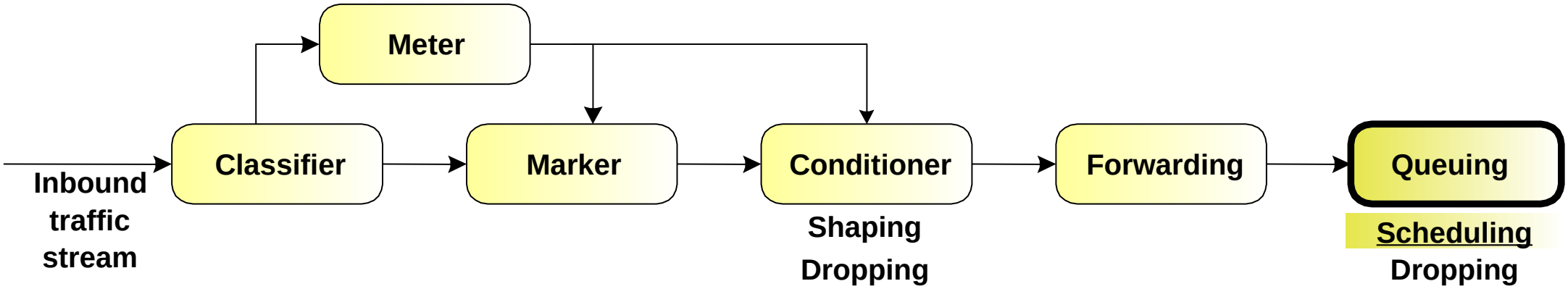
- Dropping mechanisms
  - Committed Access Rate (CAR) and Class-based Policing can drop packets that exceed the contractual rate
  - Weighted Random Early Detection (WRED) can randomly drop packets when an interface is nearing congestion

DiffServ Mechanisms



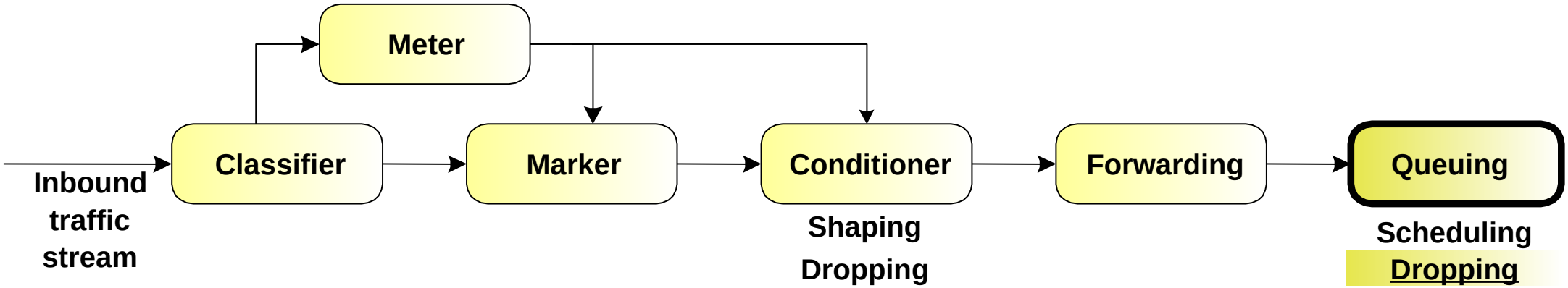
- Forwarding mechanisms
  - Routing
  - e.g. Cisco Express Forwarding (CEF)

DiffServ Mechanisms



- Traditional queuing mechanisms
  - FIFO, Priority Queuing (PQ), Custom Queuing (CQ)
- Weighted Fair Queuing (WFQ) family
  - WFQ, dWFQ, CoS-based dWFQ, QoS-group dWFQ
- Advanced queuing mechanisms
  - Class-based WFQ, Class-based LLQ

DiffServ Mechanisms

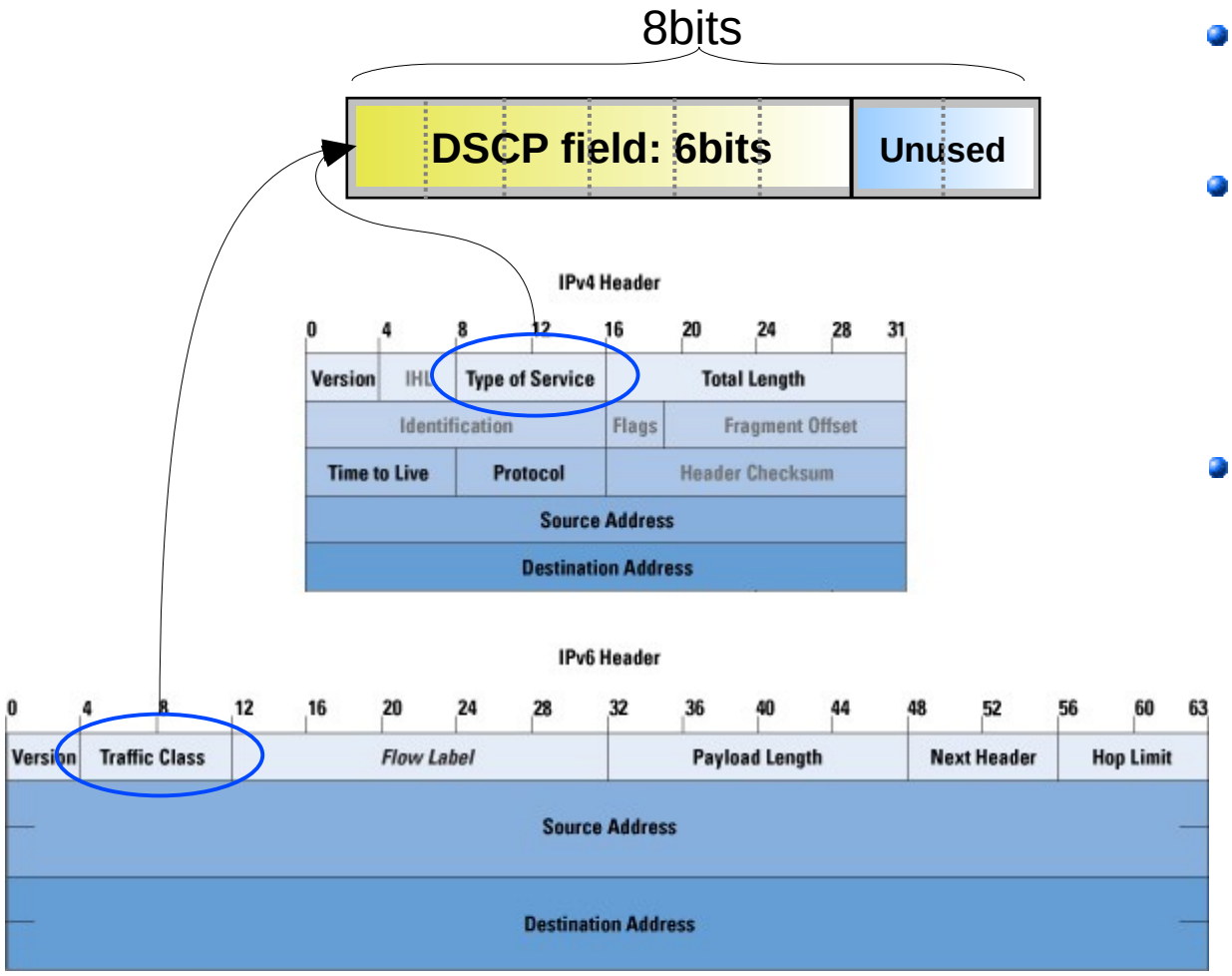


- Dropping mechanisms
  - Tail drop on queue congestion
  - WFQ has an improved tail-drop scheme
  - WRED randomly drops packets when nearing congestion

Functional elements

- The functional elements of the DiffServ architecture are:
- Edge Routers:
  - Classify packets: they mark each packet on the Type of Service field of the IPv4 header or Traffic Class field of the IPv6 header
  - Condition traffic: for example, they use a “Token Bucket” to verify if the incoming traffic is conforming to the contracted traffic and, if not,
    - delay excess traffic or
    - drop excess traffic
- Core Routers:
  - Identify treatment that should be given to packets based on their mark and according to a Per-Hop-Behavior (PHB)

Edge Routers: Traffic classification



- All classification and QoS revolves around the DSCP field
- Format
  - DSCP – Differentiated Service Code Point (6bits)
  - CU – Currently Unused (2bits)
- Packets are marked on the
  - Type of Service (TOS) field of the IPv4 header
  - Traffic Class field of the IPv6 header

Routing on *Core Routers*

- Different PHBs (*Per-Hop-Behaviors*) result on different network performances
- PHBs do not specify which queuing mechanisms should be used
- PHBs examples:
  - x% of the physical bandwidth is attributed to packets from Class A during any time interval of a given specified length
  - Packets from Class A are always served before Class B packets
  - Packets from Class A are served with twice the service bandwidth that is attributed to Class B packets

Per Hop Behaviors

- The Default PHB
  - Traditional best effort service
    - DSCP value (recommended) of “000000”
- Class-Selector PHBs
  - To preserve backward compatibility with the IP-precedence scheme
    - DSCP values of the form “xxx000”
  - These PHBs ensure that DS-compliant nodes can co-exist with IP-precedence aware nodes
- Expedited Forwarding (EF) PHB
  - Provides a low-loss, low-latency, low-jitter, and assured bandwidth service
  - Recommended DSCP value for EF is “101110”
- Assured Forwarding PHB
  - Can be provide different forwarding assurances.
    - For example, traffic can be divided into gold, silver, and bronze classes, with gold being allocated 50 percent of the available link bandwidth, silver 30 percent, and bronze 20 percent
  - The AFxy PHB defines four AFx classes: AF1, AF2, AF3, and AF4

Expedited Forwarding

- Expedited Forwarding (EF) PHB:
  - Ensures a minimum departure rate
  - Guarantees bandwidth – the class is guaranteed an amount of bandwidth with prioritized forwarding
  - Polices bandwidth – the class is not allowed to exceed the guaranteed amount (excess traffic is dropped)
- DSCP value: “101110”; looks like IP precedence 5 to non-DS compliant devices

EF PHB Implementations

- Priority Queuing
- IP RTP Prioritization
- Class-based Low-latency Queuing (CB-LLQ)
- Strict Priority queuing within Modified Deficit Round Robin (MDRR)

Assured Forwarding

- Assured Forwarding (AF) PHB:
  - Guarantees bandwidth
  - Allows access to extra bandwidth if available
- Four standard classes (AF1, AF2, AF3 and AF4)
- DSCP value range: “aaadd0” where “aaa” is a binary value of the class and “dd” is drop probability

DiffServ Service Classes

- *Default* (DE) → DSCP = 000000 = 0
  - *best-effort* service with a single FIFO-type queue
- *Expedited Forwarding* (EF) → DSCP = 101110 = 46
  - Service of the “virtual leased line” type
  - Provides control for losses, delay and delay variance inside a specified maximum bandwidth
- *Assured Forwarding* (AF)
  - Provides a relative Quality of Service (AF*i* is served with more bandwidth than AF*j*, for *i*<*j*)
  - On each class, there are 3 precedence levels for dropping packets in case of congestion

Drop	Class 1	Class 2	Class 3	Class 4
Low	001010 AF11 DSCP 10	010010 AF21 DSCP 18	011010 AF31 DSCP 26	100010 AF41 DSCP 34
Medium	001100 AF12 DSCP 12	010100 AF 22 DSCP 20	011100 AF32 DSCP 28	100100 AF42 DSCP 36
High	001110 AF13 DSCP 14	010110 AF23 DSCP 22	011110 AF33 DSCP 30	100110 AF43 DSCP 38

AF PHB Definition

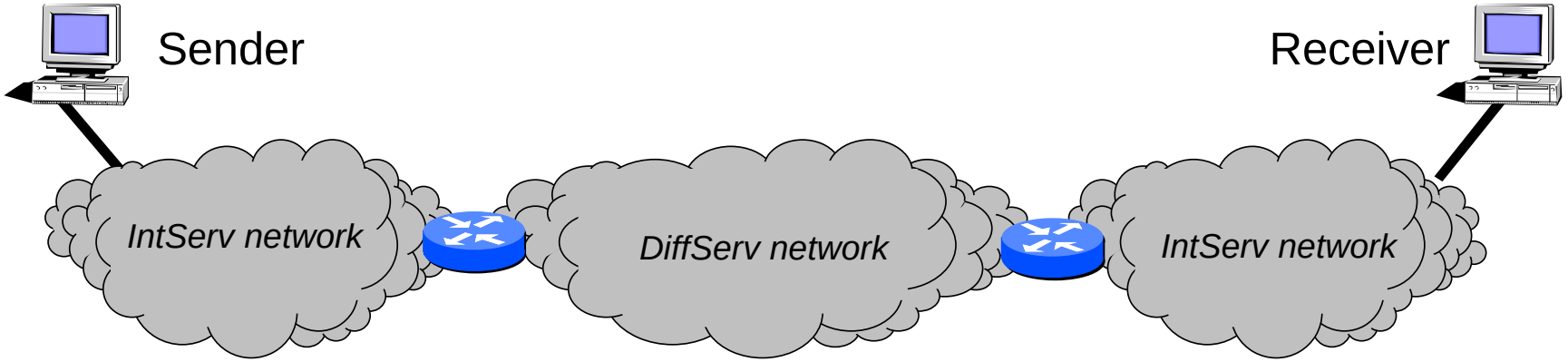
- A DS node MUST allocate a configurable, minimum amount of forwarding resources (buffer space and bandwidth) per AF class
- Excess resources may be allocated between non-idle classes. The manner must be specified.
- Reordering of IP packets of the same flow is not allowed if they belong to the same AF class

AF PHB Implementation

- CBWFQ (4 classes) with WRED within each class
- (M)DRR with WRED within each class
- Optionally Custom Queuing (does not support differentiated dropping)

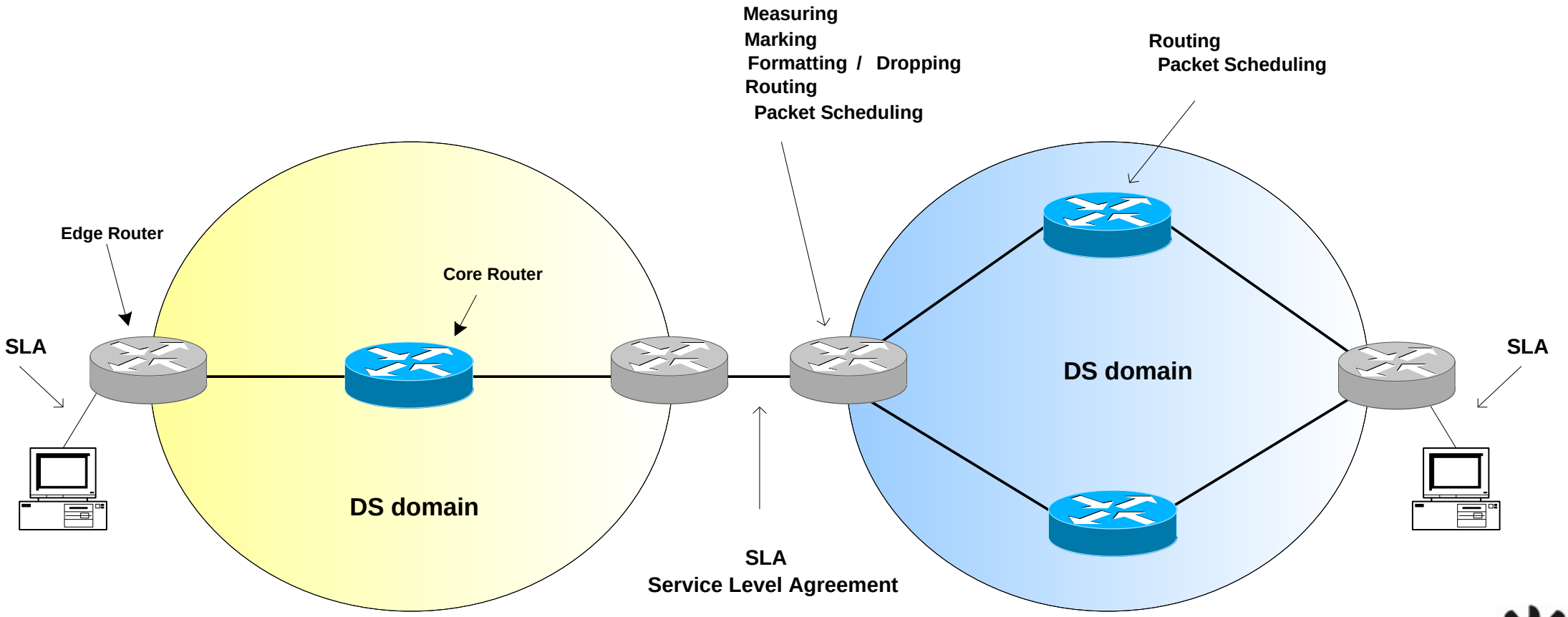
IntServ and *DiffServ* integration

- Use:
  - *IntServ* architecture (appropriate for small networks) on access networks
  - *DiffServ* architecture (appropriate for large networks) on transit networks
- Border routers of both network types:
  - Classify RSVP requests on the appropriate *DiffServ* service classes
  - If the are no sufficient resources, they refuse the RSVP reservation requests
- Advantages:
  - Provide *IntServ* services on large networks
  - Provide explicit admission control instead of SLAs on *DiffServ networks*



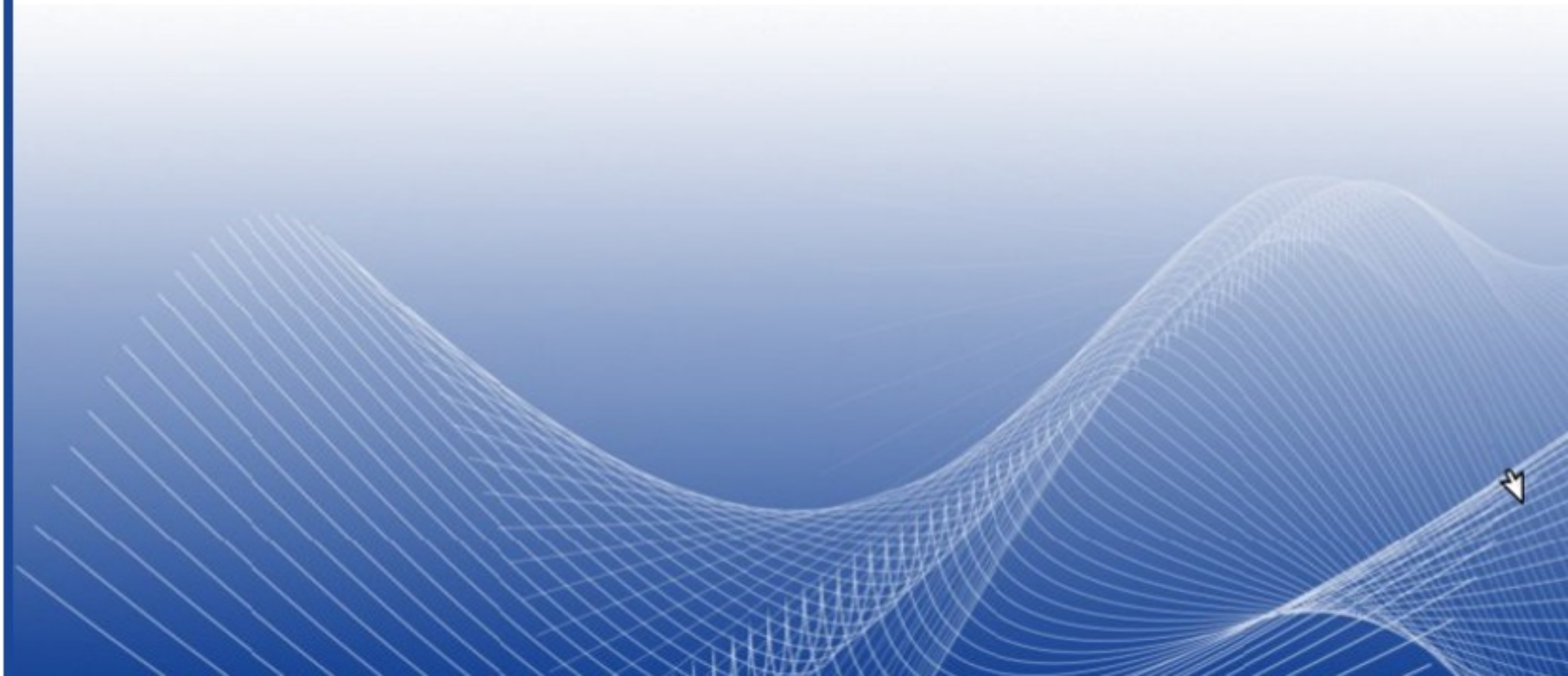
DiffServ Domains and SLAs

- Quality of Service that is provided to a client is configured:
  - By management (traffic conditioning configuration is made on the respective Edge Router)
  - According to the Service Level Agreement (SLA)



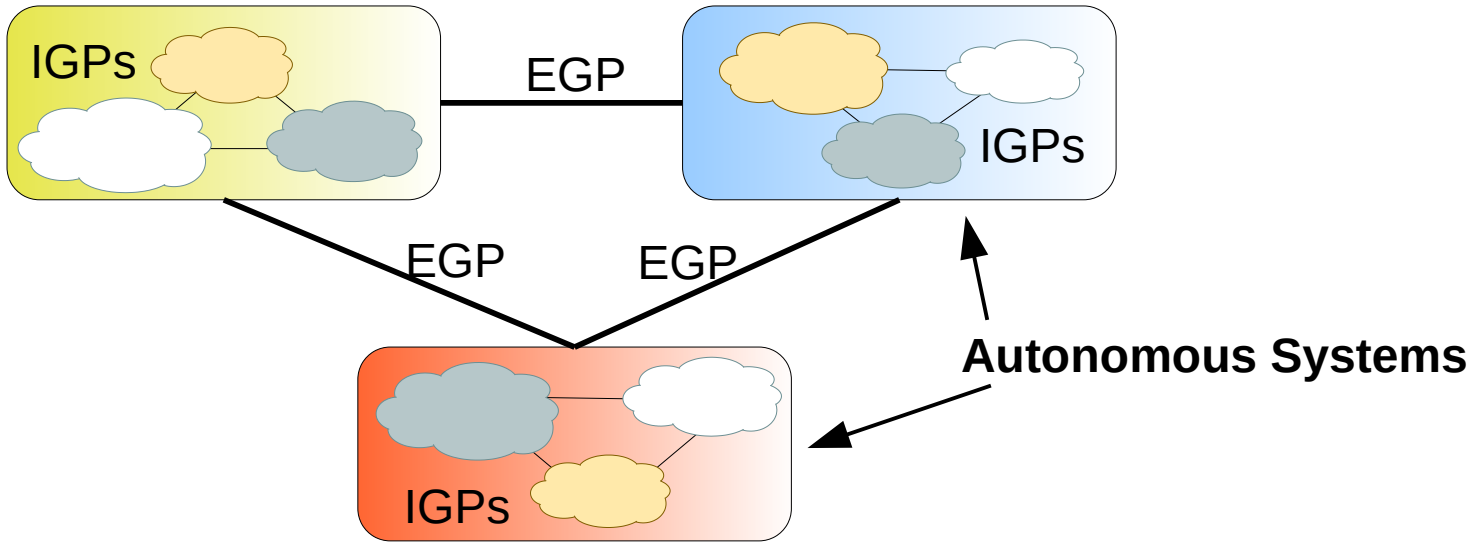
External Routing (BGP and MP-BGP)

Arquitectura de Comunicações



universidade de aveiro
deti.ua.pt

Border Gateway Protocol (BGP)



- Border Gateway Protocol Version 4 of the protocol (BGP4) was deployed in 1993 and currently is the protocol that assures Internet connectivity
- BGP is mainly used for routing between Autonomous Systems
- Autonomous System (AS) is a network under a single administration
  - One or more network operators with a common well defined global routing policy

AS Numbers

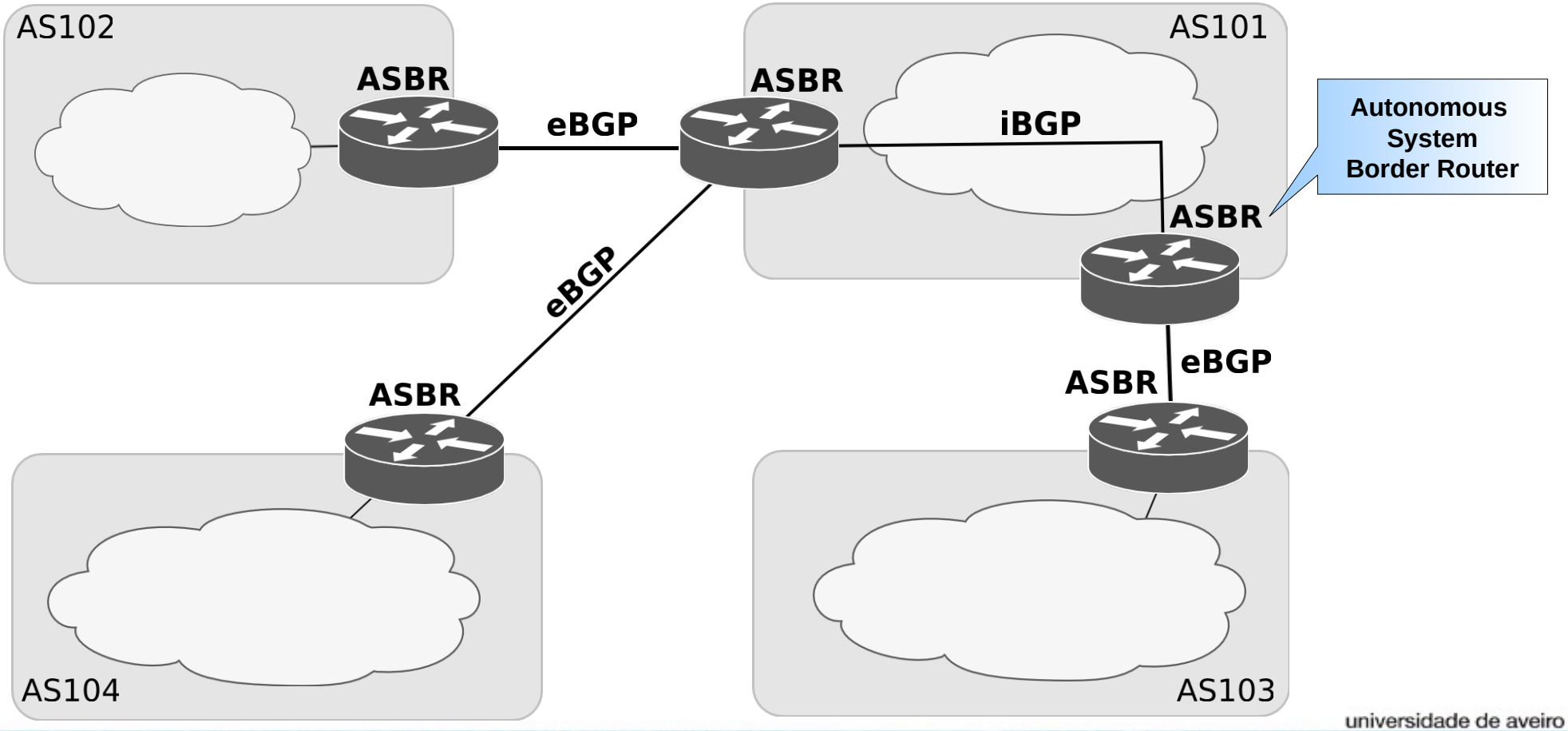
- Allocated ID by InterNIC and is globally unique
- RFC 4271 defines an AS number as 2-bytes
  - Private AS Numbers = 64512 through 65535
  - Public AS Numbers = 1 through 64511
    - 39000+ have already been allocated
    - We will eventually run out of AS numbers
- Need to expand AS size from 2-bytes to 4-bytes
- RFC4893 defines BGP support for 4-bytes AS numbers
  - 4,294,967,295 AS numbers
  - As of January 1, 2009, all new Autonomous System numbers issued will be 4-byte by default, unless otherwise requested.
  - The full binary 4-byte AS number is split two words of 16 bits each
    - Notation:
    - <higher2bytes in decimal>.<lower2bytes in decimal>
    - Example1: AS 65546 is represented as “1.10”
    - Example2: AS 50000 is represented as “0.50000”
  - Cannot have a “flag day” solution

BGP Neighbor Relationships

- Often called peering
  - Usually manually configured into routers by the administrator
- Each neighbor session runs over TCP (port 179)
  - Ensures reliable data delivery
- Peers exchange all their routes when the session is first established
- Updates are also sent when there is a topology change in the network or a change in routing policy
- BGP peers exchange session KEEPALIVE messages
  - To avoid extended periods of inactivity.
  - Low keepalive intervals can be set if a fast fail-over is required

Internal BGP (iBGP) & External BGP (eBGP)

- Neighbor relations can be established between
  - Same AS routers (Internal BGP – iBGP).
  - Different AS routers (External BGP – eBGP).
- Routers that implement neighbor relations are called an Autonomous System Border Router (ASBR).

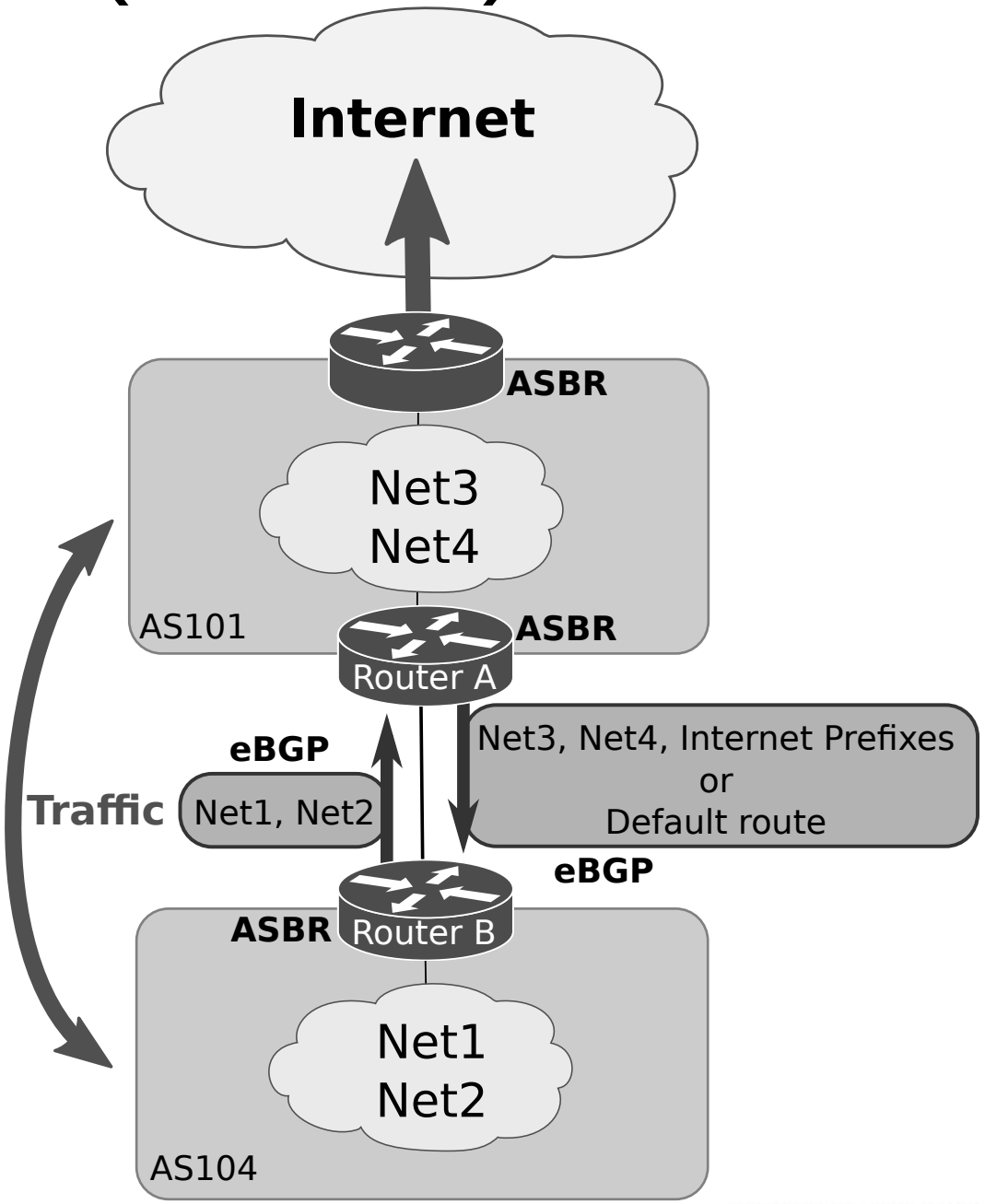


External and Internal BGP

- External BGP (eBGP) is used between AS.
- Internal BGP (iBGP) is used within AS.
- A BGP router never forwards a path learned from one iBGP peer to another iBGP peer even if that path is the best path.
  - An exception is when a router is configured as route-reflector.
- A BGP forward the routes learned from one eBGP peer to both eBGP and iBGP peers.
  - Filters can be used to modify this behavior.
- iBGP routers in an AS **must maintain an iBGP session with all other iBGP routers** in the AS (iBGP Mesh).
  - To obtain complete routing information about external networks.
  - Most networks also use an IGP, such as OSPF.
  - Additional methods can be used to reduce iBGP Mesh complexity.
    - Route reflectors, private AS, ...

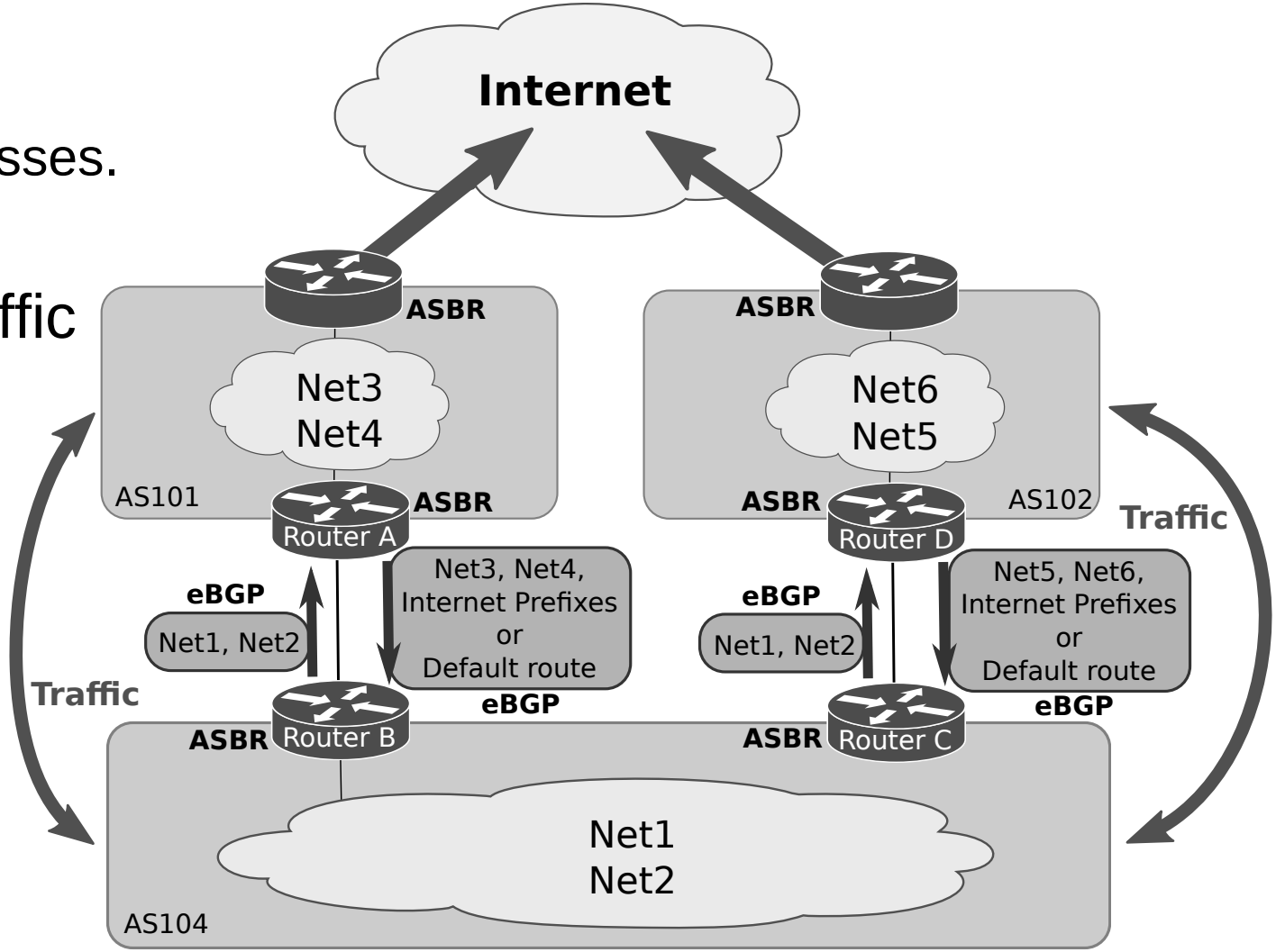
Single-homed (or Stub) AS

- AS has only one border router (ASBR)
  - Single Internet access.
  - Single ISP.



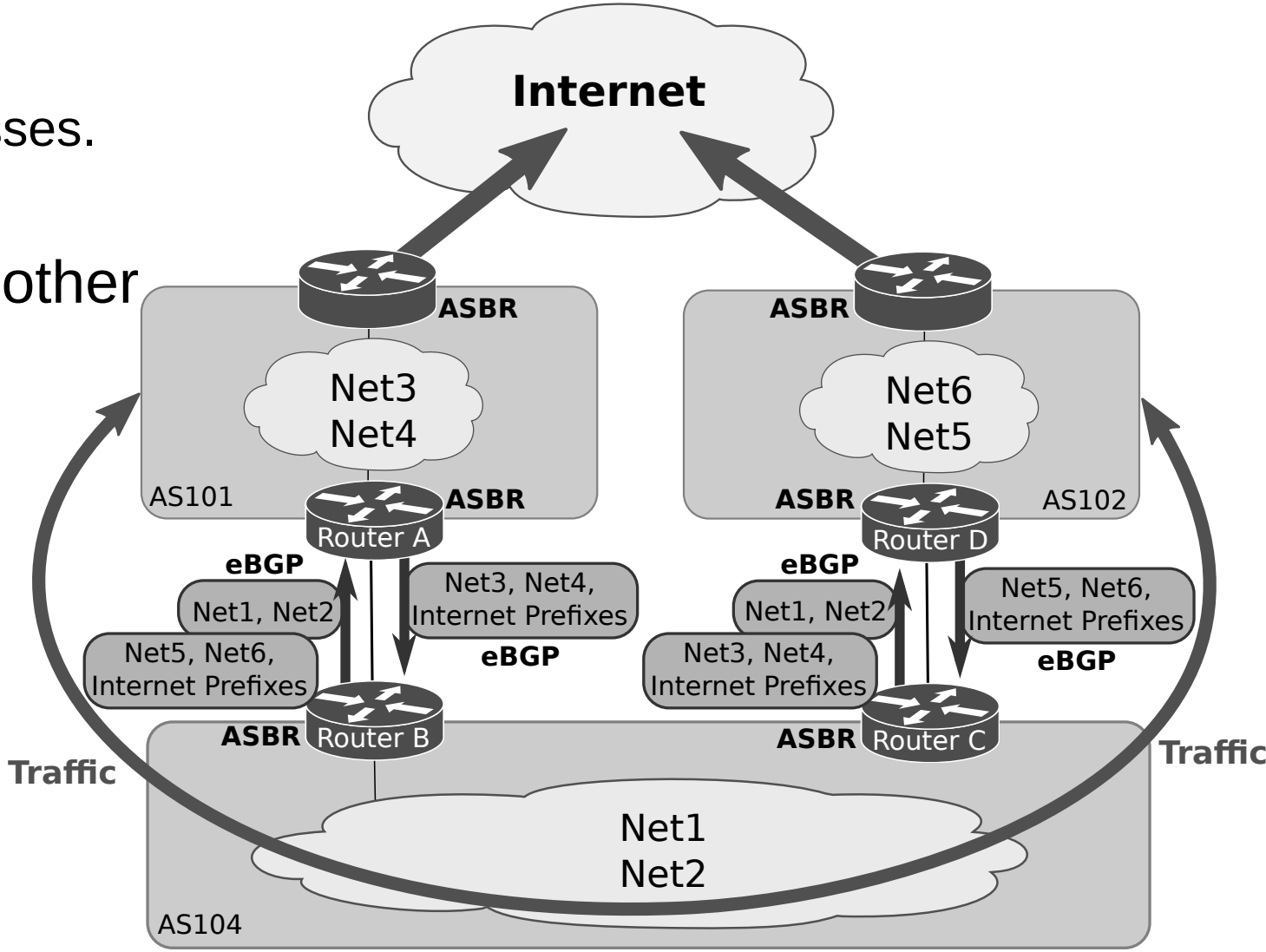
Multi-homed Non-transit AS

- AS has more than one border router (ASBR)
  - Multiple Internet accesses.
  - Multiple ISP.
- Does not transport traffic from other AS.



Multi-homed Transit AS

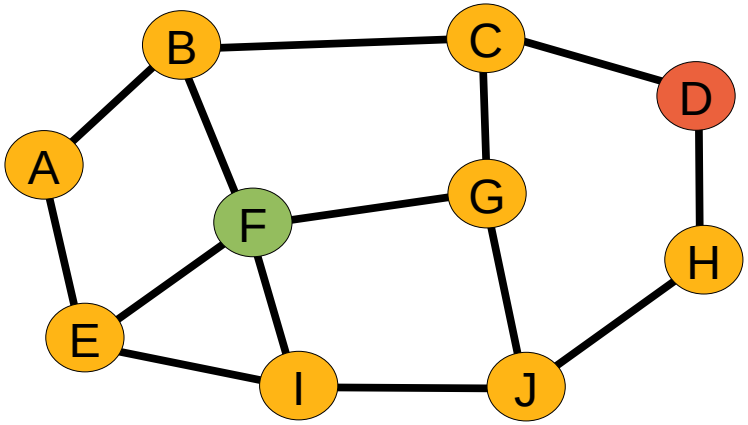
- AS has more than one border router (ASBR).
  - Multiple Internet accesses.
  - Multiple ISP.
- Transports traffic from other AS.



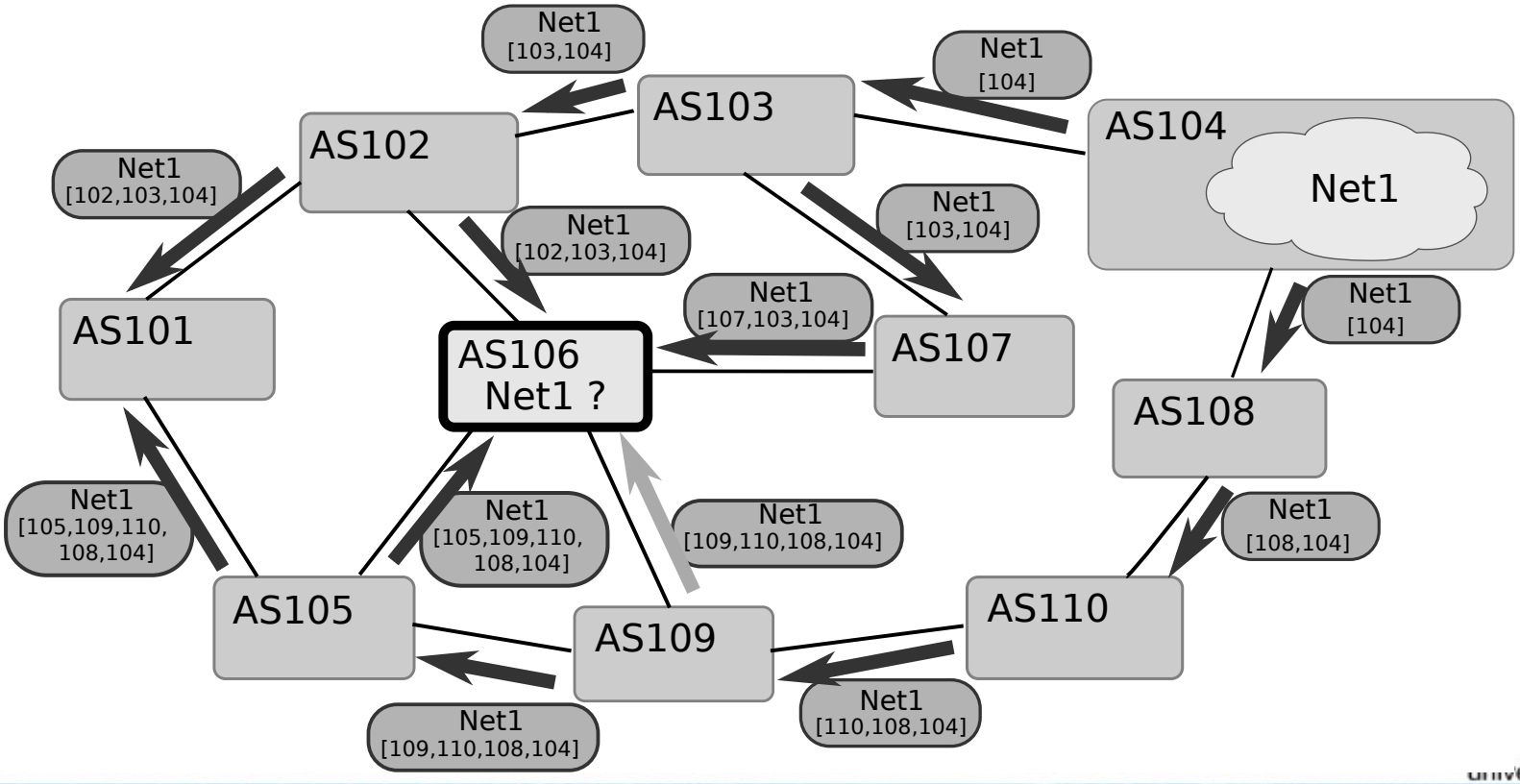
Path-vector

- BGP is a path-vector protocol
- Although it is essentially a distance-vector protocol that carries a list of the AS traversed by the route
  - Provides loop detection
- An EBGP speaker adds its own AS to this list before forwarding a route to another EBGP peer
- An IBGP speaker does not modify the list because it is sending the route to a peer within the same AS
  - AS list cannot be used to detect the IBGP routing loops

Path vector

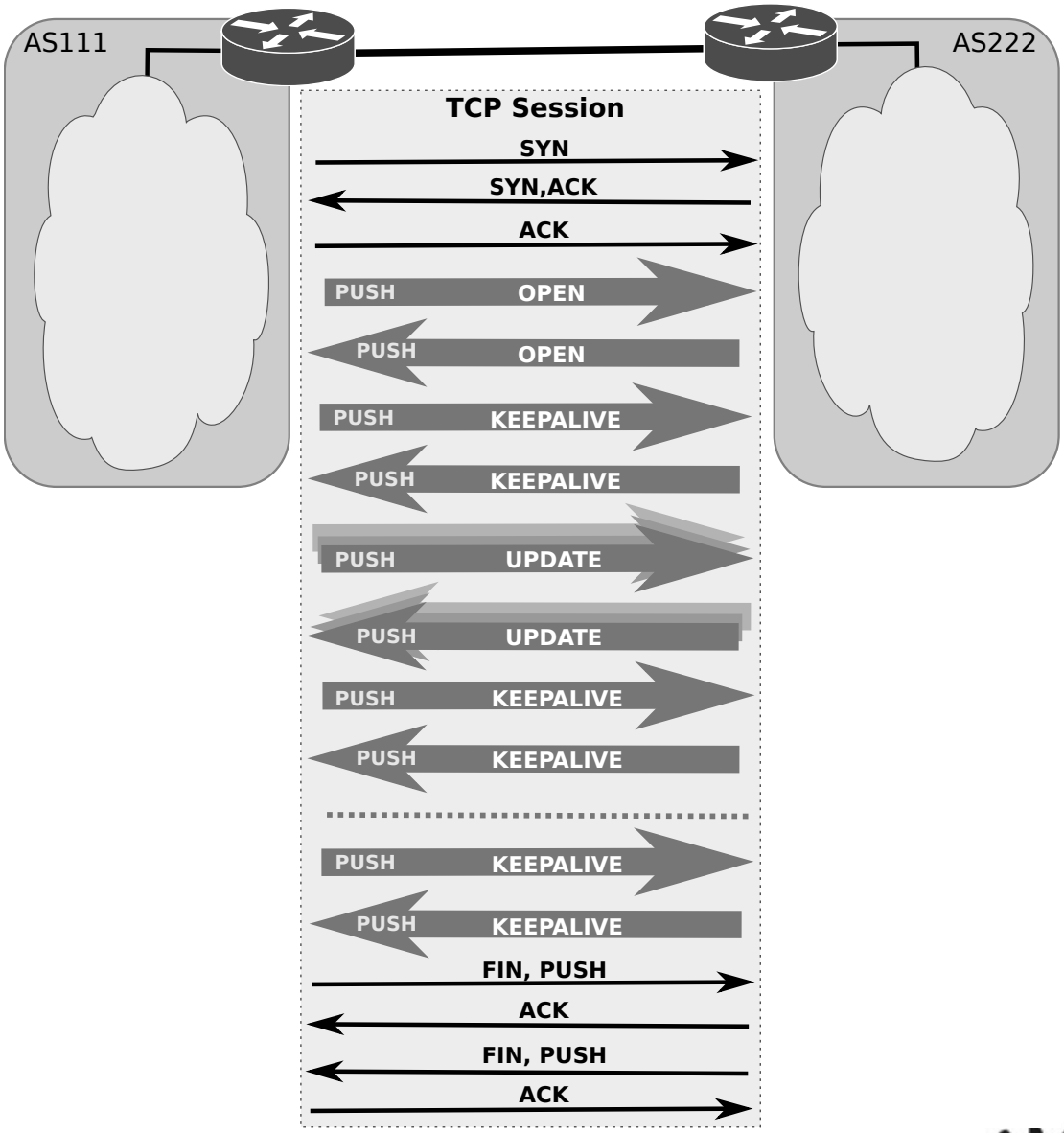


- F receives from its neighbors different paths to D:
  - De B: “I use BCD”
  - De G: “I use GCD”
  - De I: “I use IFGCD”
  - De E: “I use EFGCD”



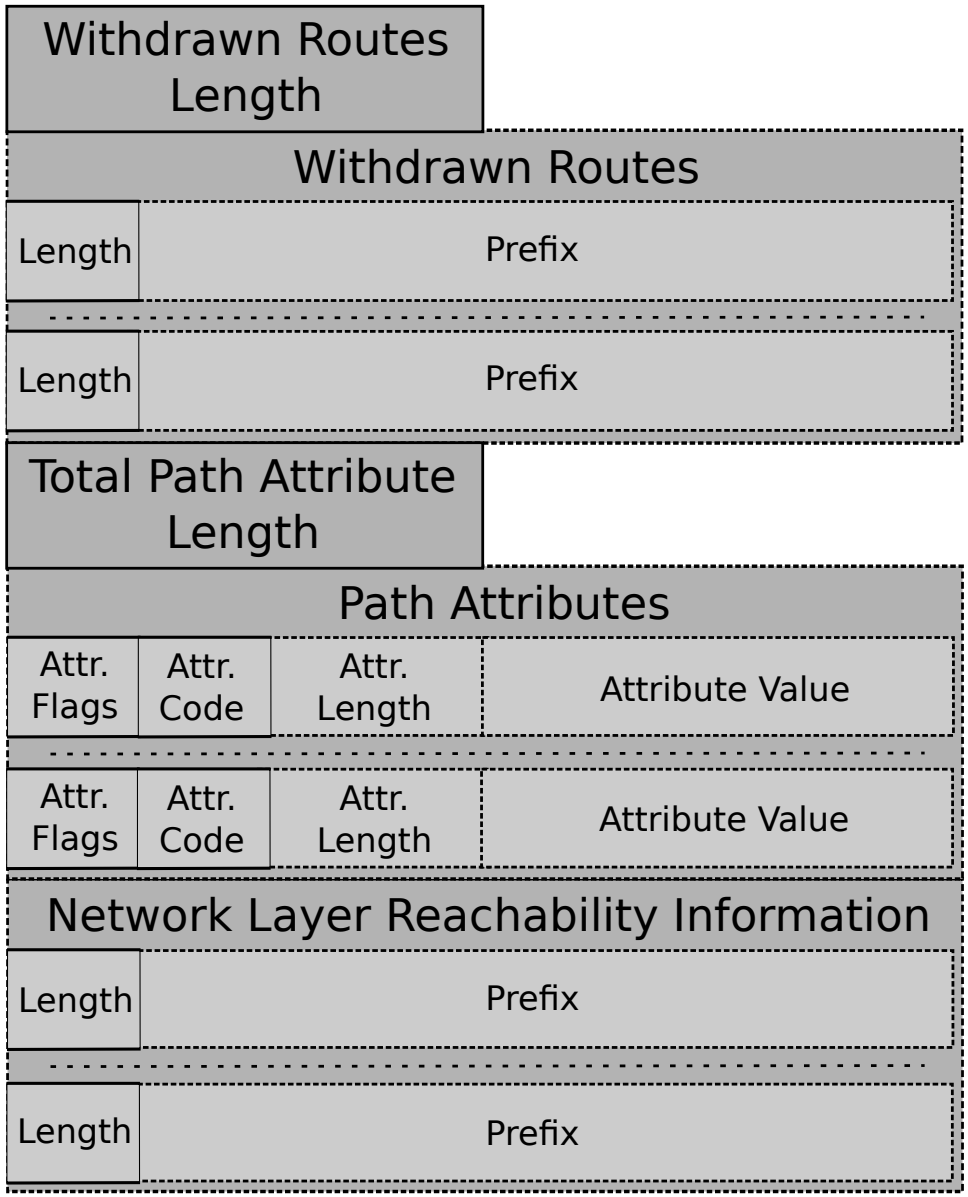
BGP Messages

- OPEN messages are used to establish the BGP session.
- UPDATE messages are used to send routing prefixes, along with their associated BGP attributes (such as the AS-PATH).
- KEEPALIVE messages are exchanged whenever the keepalive period is exceeded, without an update being exchanged.
- NOTIFICATION messages are sent whenever a protocol error is detected, after which the BGP session is closed.

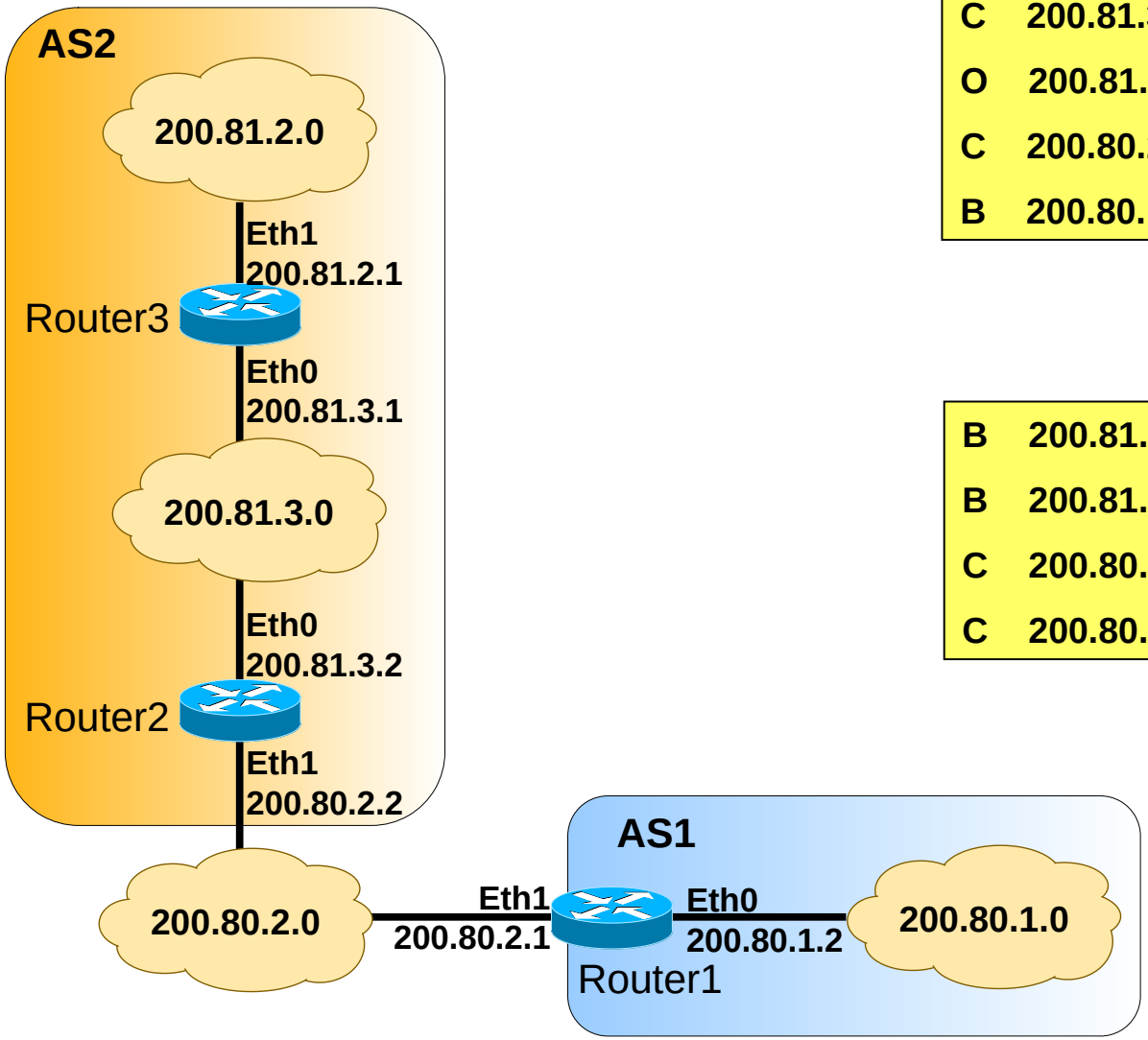


Update Message

- Withdrawn routes – List of IP networks no longer accessible.
- Path attributes – parameters used to define routing and routing policies.
- Network layer reachability information – List of IP networks with connectivity.



Example



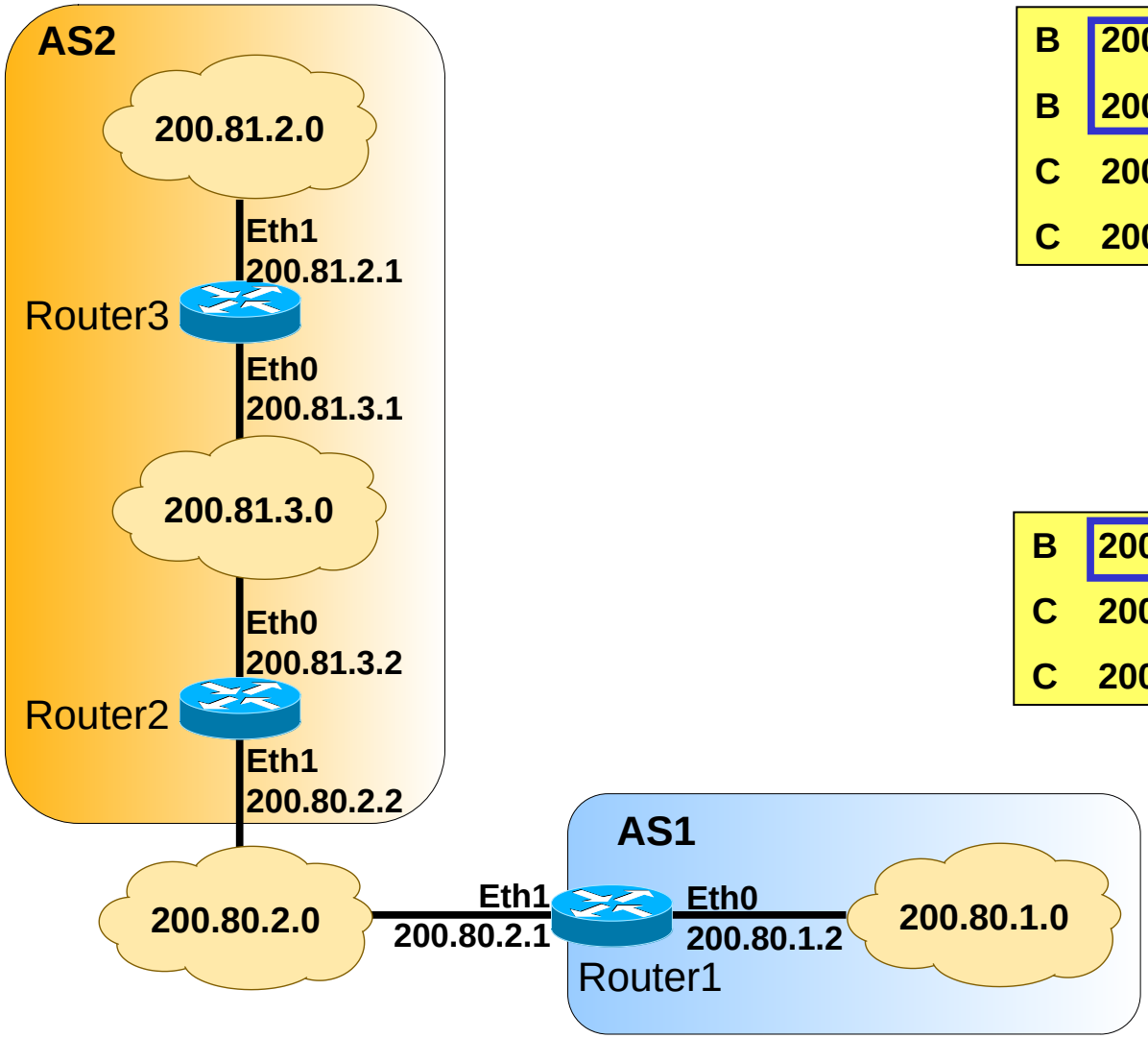
C	200.81.3.0/24 is directly connected, Ethernet0
O	200.81.2.0/24 [110/20] via 200.81.3.1, 00:01:12
C	200.80.2.0/24 is directly connected, Ethernet1
B	200.80.1.0/24 [20/0] via 200.80.2.1, 00:00:29

Router 2's routing table

B	200.81.3.0/24 [20/0] via 200.80.2.2, 00:01:58
B	200.81.2.0/24 [20/0] via 200.80.2.2, 00:01:57
C	200.80.2.0/24 is directly connected, Ethernet1
C	200.80.1.0/24 is directly connected, Ethernet0

Router 1's routing table

Example – BGP networks aggregation



Before aggregation

```
B   200.81.3.0/24 [20/0] via 200.80.2.2, 00:01:58
B   200.81.2.0/24 [20/0] via 200.80.2.2, 00:01:57
C   200.80.2.0/24 is directly connected, Ethernet1
C   200.80.1.0/24 is directly connected, Ethernet0
```

Router 1

After aggregation

```
B   200.81.2.0/23 [20/0] via 200.80.2.2, 00:01:06
C   200.80.2.0/24 is directly connected, Ethernet1
C   200.80.1.0/24 is directly connected, Ethernet0
```

Router 1

BGP Attributes

- A BGP attribute, or path attribute, is a metric used to describe the characteristics of a BGP path.
- Attributes are contained in update messages passed between BGP peers to advertise routes. There are 4+1 categories of BGP attributes.
  - Well-known Mandatory (included in BGP updates)
    - AS-path, Next-hop, Origin.
  - Well-known Discretionary (may or may not be included in BGP updates)
    - Local Preference, Atomic Aggregate.
  - Optional Transitive (may not be supported by all BGP implementations)
    - Aggregator, Community, AS4_Aggregator, AS4_path.
  - Optional Non-transitive (may not be supported by all BGP implementations)
    - If the neighbor doesn't support that attribute it is deleted
    - Multi-exit-discriminator (MED).
  - Cisco-defined (local to router, not advertised)
    - Weight

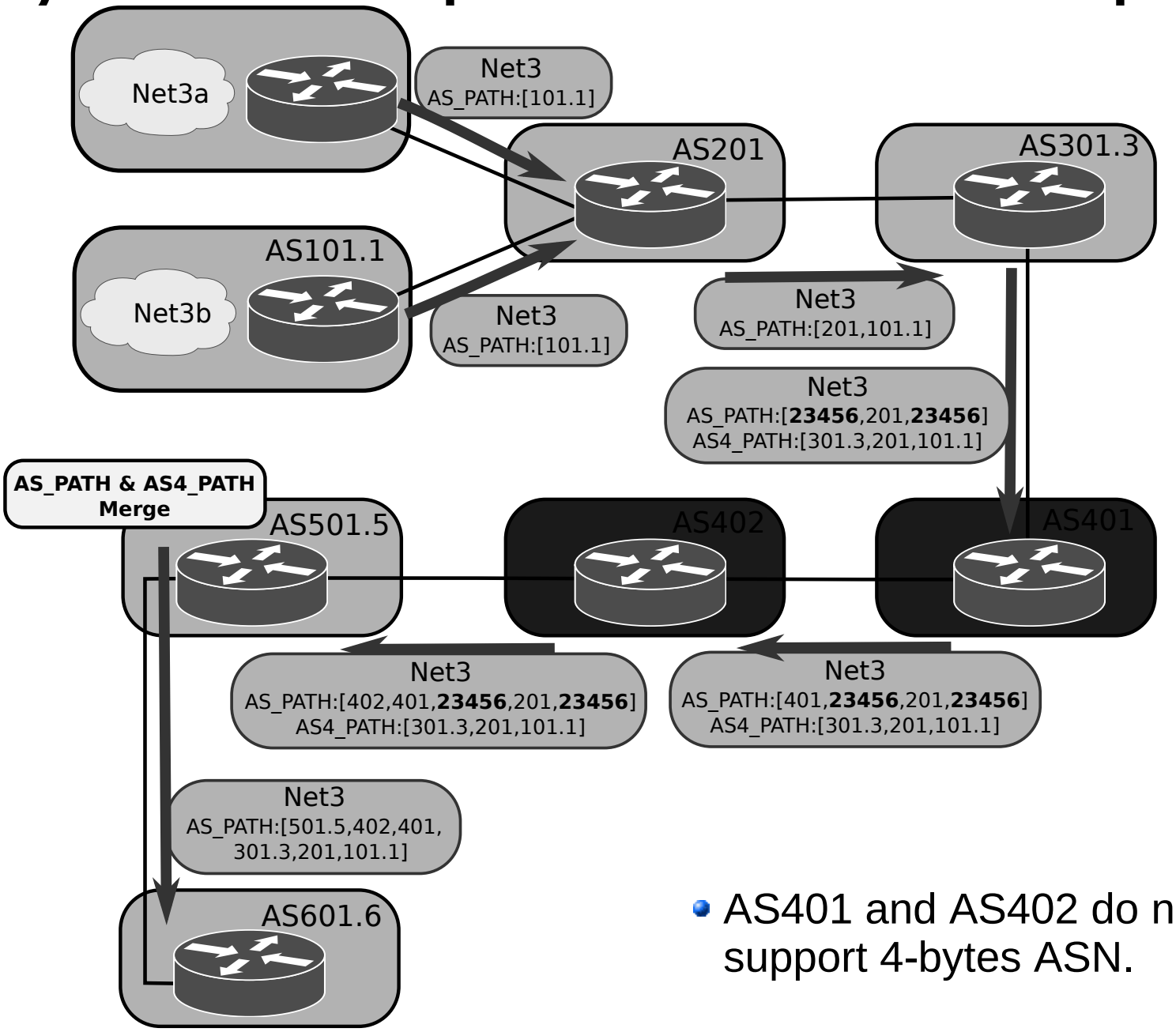
AS-PATH and ORIGIN Attributes

- AS-PATH
  - When a route advertisement passes through an autonomous system, the AS number is added to an ordered list of AS numbers that the route advertisement has traversed.
- ORIGIN
  - Indicates how BGP learned about a particular route. Can take three possible values:
    - IGP (0) value is set if the route is interior to the originating AS, resulting from an explicit inclusion of a network within the BGP routing process by means of manual configuration.
    - INCOMPLETE (2) value is set if the route is learned by other means, namely, route redistribution from other routing processes into the BGP routing process.
    - EGP (1) is no longer used in modern networks.

AS4_PATH & AS4_AGGREGATOR

- AS4_PATH attribute has the same semantics as the AS_PATH attribute, except that it is optional transitive, and it carries 4-bytes AS numbers.
- AS4_AGGREGATOR attribute has the same semantics as the AGGREGATOR attribute, except that it carries a 4-bytes AS number.
- 4-byte AS support is advertised via BGP capability negotiation
  - Speakers who support 4-byte AS are known as NEW BGP speakers
  - Those who do not are known as OLD BGP speakers
- New Reserved AS number
  - AS_TRANS = AS 23456
    - 2-byte placeholder for a 4-byte AS number
    - Used for backward compatibility between OLD and NEW BGP speakers
- Receiving UPDATEs from a NEW speaker
  - Decode each AS number as 4-bytes
  - AS_PATH and AGGREGATOR are effected
- Receiving UPDATEs from an OLD speaker
  - AS4_AGGREGATOR will override AGGREGATOR
  - AS4_PATH and AS_PATH must be merged to form the correct as-path
- Merging AS4_PATH and AS_PATH
  - AS_PATH → [275 250 225 23456 23456 200 23456 175]
  - AS4_PATH → [100.1 100.2 200 100.3 175]
  - Merged AS-PATH → [275 250 225 100.1 100.2 200 100.3 175]

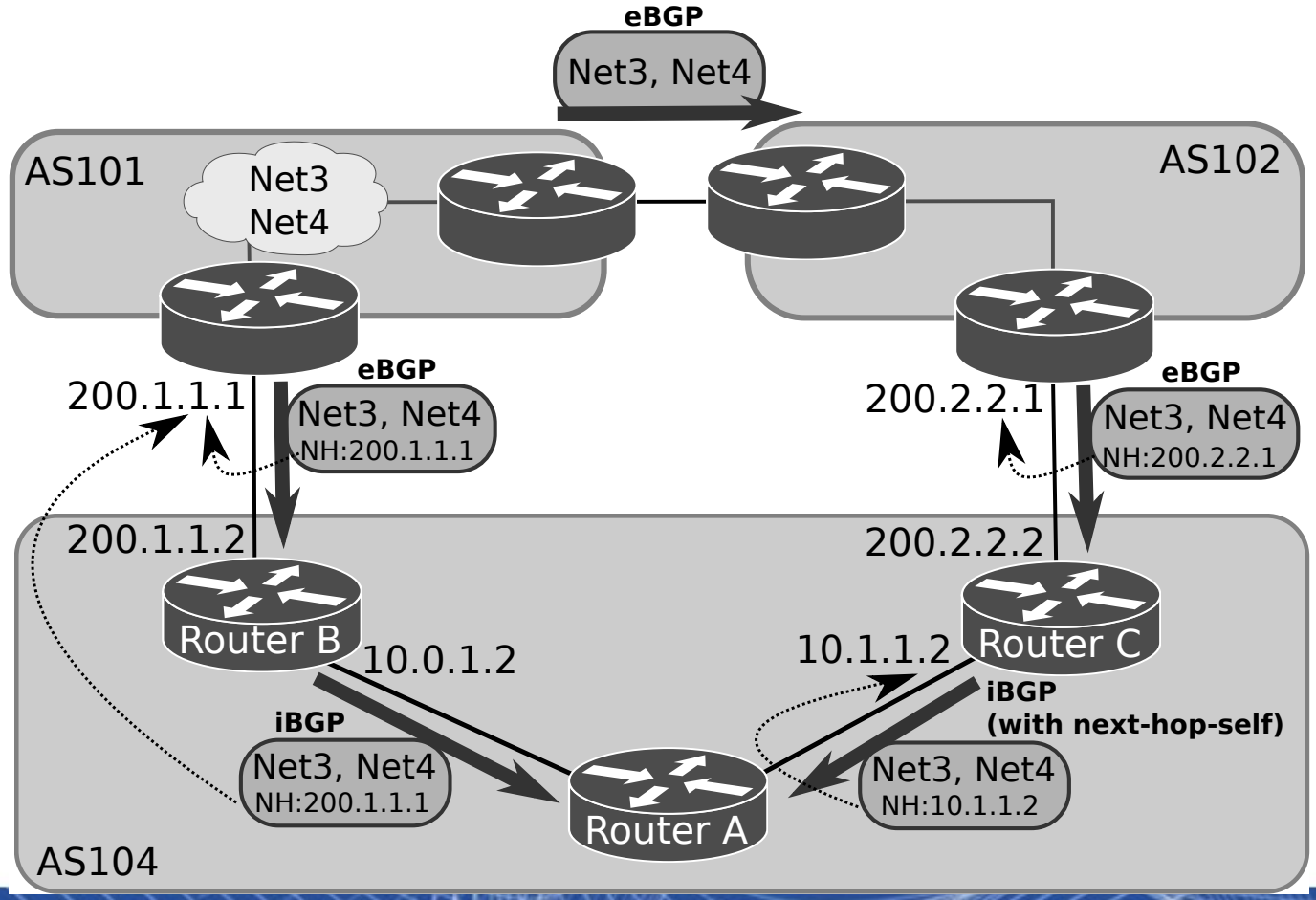
4-bytes AS Operational Example



- AS401 and AS402 do not support 4-bytes ASN.

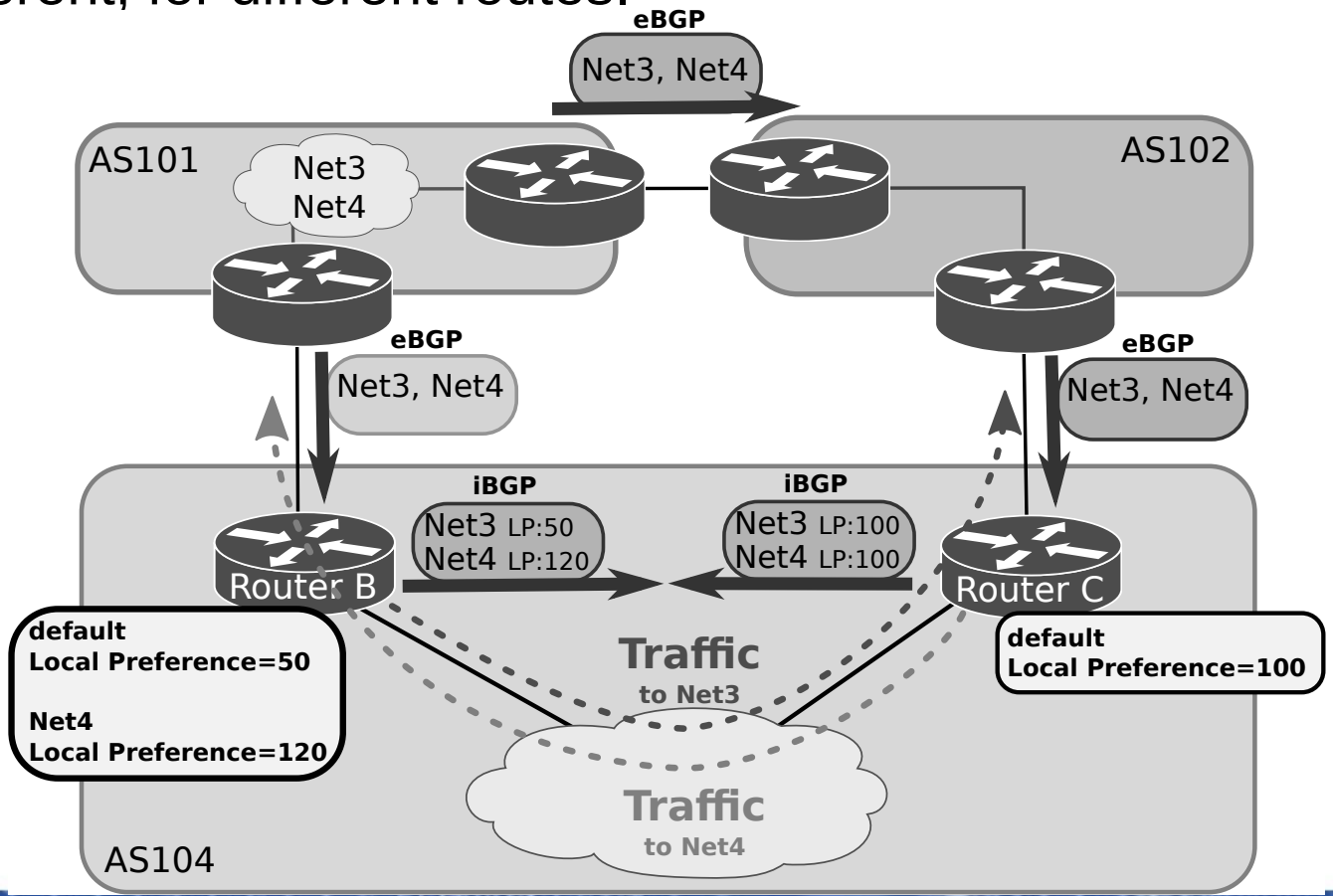
Next-Hop Attribute

- The eBGP next-hop attribute is the IP address that is used to reach the advertising router
- For eBGP, the next-hop address is the IP address of the connection between the peers
- For iBGP, the eBGP next-hop address is carried into the local AS
  - By configuration the AS border router can be the next-hop to iBGP neighbors



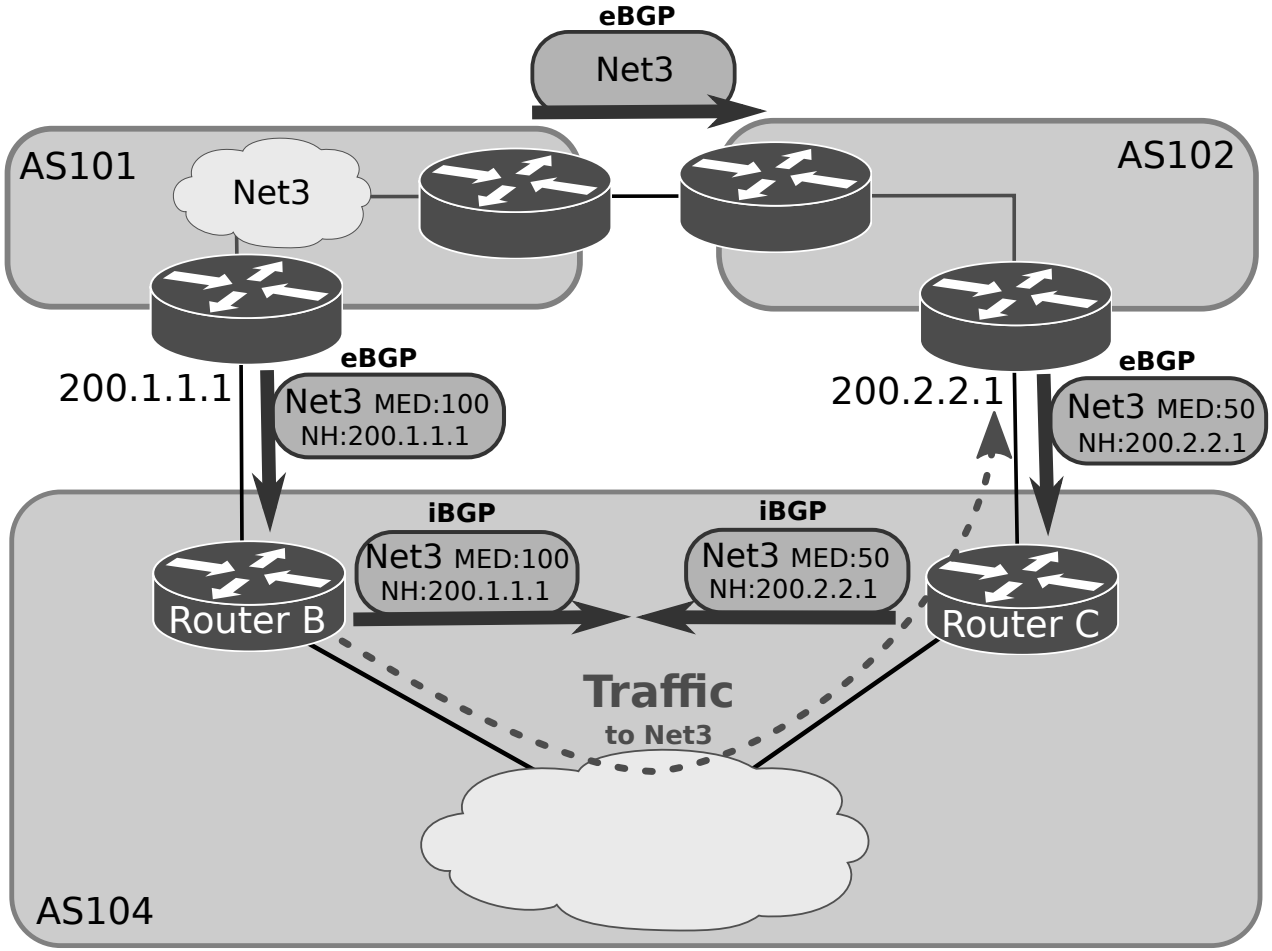
Local Preference Attribute

- The local preference attribute is used to choose an exit point from the local autonomous system (AS).
  - **Higher value** is preferred.
- The local preference attribute is propagated throughout the local AS.
- Can be different, for different routes.



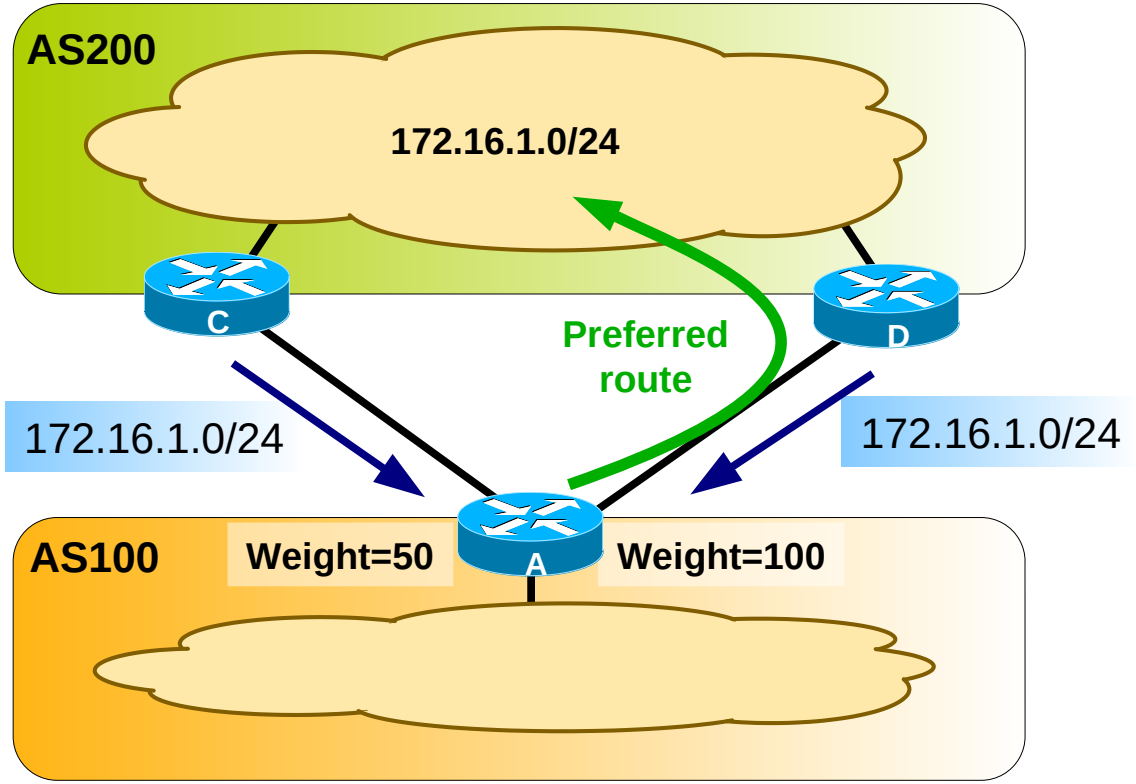
Multi-Exit Discriminator Attribute (MED)

- The multi-exit discriminator (MED) or metric attribute is used as a suggestion to an external AS.
- The external AS that is receiving the MEDs may be using other BGP attributes for route selection.
- The **lower value** of the metric is preferred.
- MED is designed to influence incoming traffic.

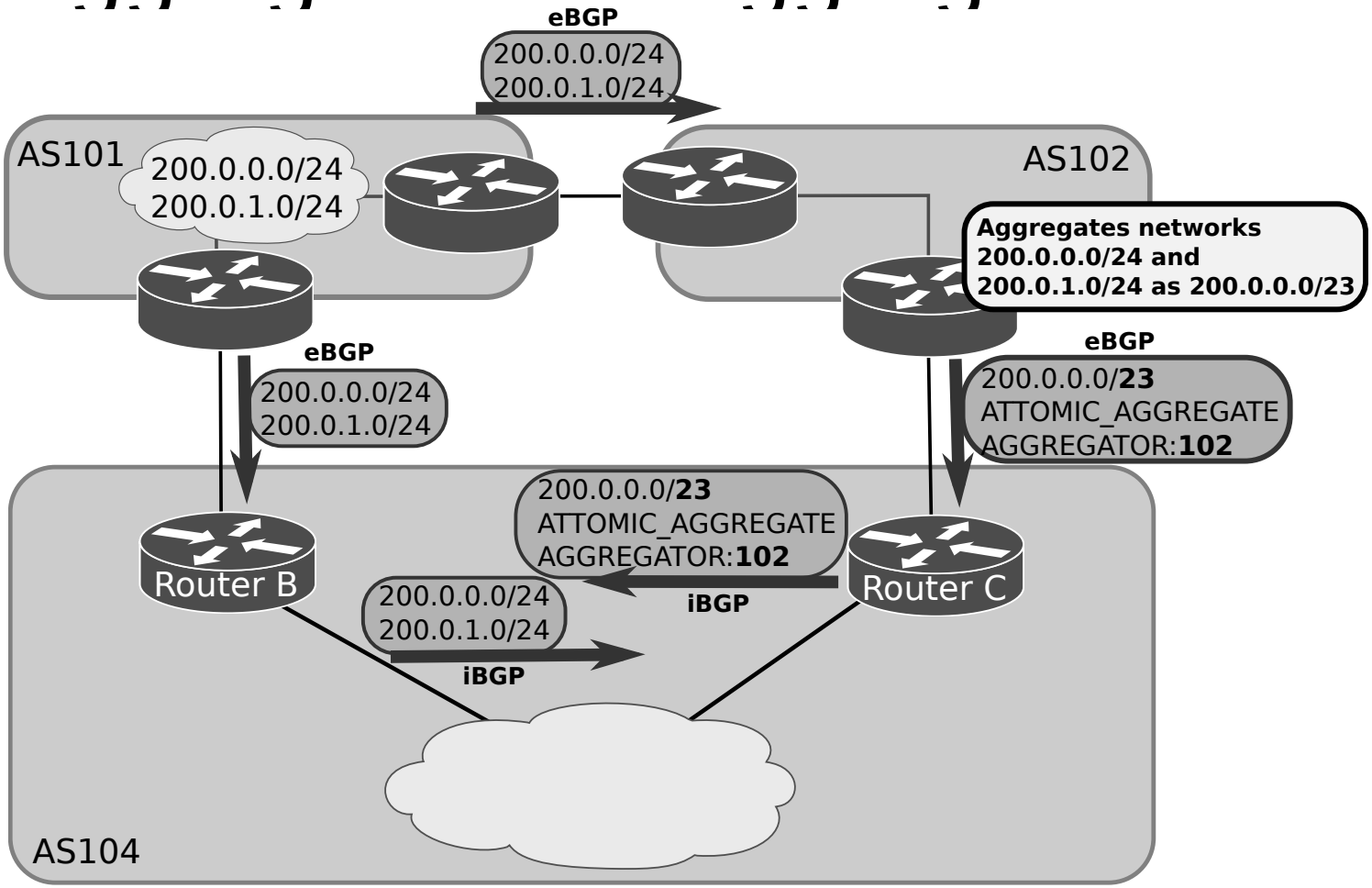


Weight Attribute

- Weight is a Cisco-defined attribute that is local to a router.
- The weight attribute is not advertised to neighboring routers.
- If the router learns about more than one route to the same destination, the route with the **highest weight** will be preferred.

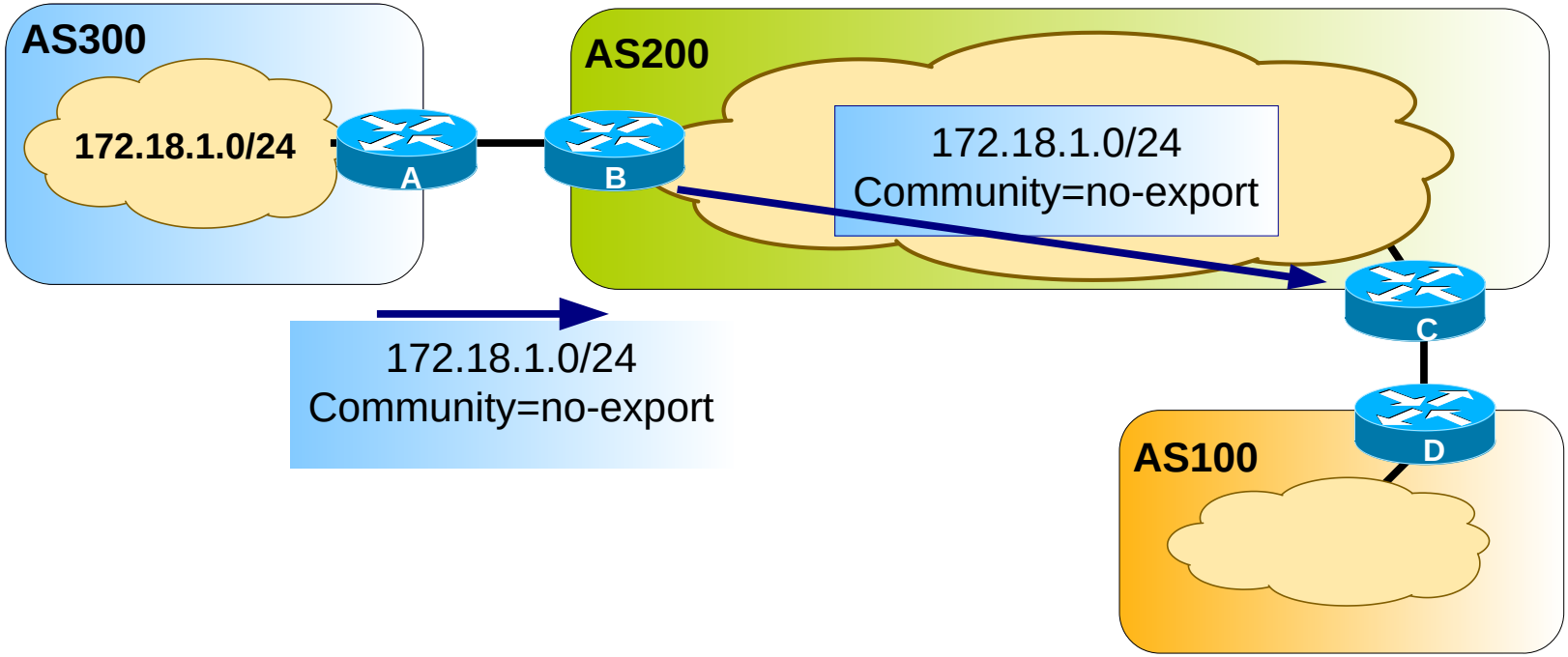


Atomic Aggregate and Aggregator Attributes



- Atomic Aggregate
  - Is used to alert routers that specific routes have been aggregated into a less specific route.
  - When aggregation like this occurs, more specific routes are lost.
- Aggregator
  - Provides information about which AS performed the aggregation.
  - And the IP address of the router that originated the aggregate.

Community Attribute



- Used to group routes that share common properties so that policies can be applied at the group level
- Predefined community attributes are:
  - no-export - Do not advertise this route to EBGP peers
  - no-advertise - Do not advertise this route to any peer
  - internet - Advertise this route to the Internet community; all routers in the network belong to it
- General communities format is ASnumber:Cnumber
  - e.g. 300:1, 200:38, etc...

BGP Path Selection

- BGP may receive multiple advertisements for the same route from multiple sources.
- BGP selects only one path as the best path.
- BGP puts the selected path in the IP routing table and propagates the path to its neighbors. BGP uses the following criteria, in the order:
  - Largest weight (Cisco only)
  - Largest local preference
  - Path that was originated locally
  - Shortest path
  - Lowest origin type (IGP lower than EGP, EGP lower than incomplete)
  - Lowest MED attribute
  - Prefer the external path over the internal path
  - Closest IGP neighbor

Multi-Protocol Border Gateway Protocol (MP-BGP)

MP-BGP Description

- Extension to the BGP protocol
- Carries routing information about other protocols/families:
  - IPv6 Unicast
  - Multicast (IPv4 and IPv6)
  - 6PE - IPv6 over IPv4 MPLS backbone
  - Multi-Protocol Label Switching (MPLS) VPN (IPv4 and IPv6)
- Exchange of Multi-Protocol Reachability Information (NLRI)

MP-BGP Attributes

- New non-transitive and optional attributes
  - MP_REACH_NLRI
    - Carry the set of reachable destinations together with the next- hop information to be used for forwarding to these destinations
  - MP_UNREACH_NLRI
    - Carry the set of unreachable destinations
- Attribute contains one or more triples
  - Address Family Information (AFI) with Sub-AFI
    - Identifies protocol information carried in the Network Layer Reachability Information
  - Next-hop information
    - Next-hop address must be of the same family
- Reachability information

MP-BGP Negotiation Capabilities

- MP-BGP routers establish BGP sessions through the OPEN message
  - OPEN message contains optional parameters
  - If OPEN parameters are not recognized, BGP session is terminated
  - A new optional parameter: CAPABILITIES
- OPEN message with CAPABILITIES containing:
  - Multi-Protocol extensions (AFI/SAFI)
  - Route Refresh
  - Outbound Route Filtering

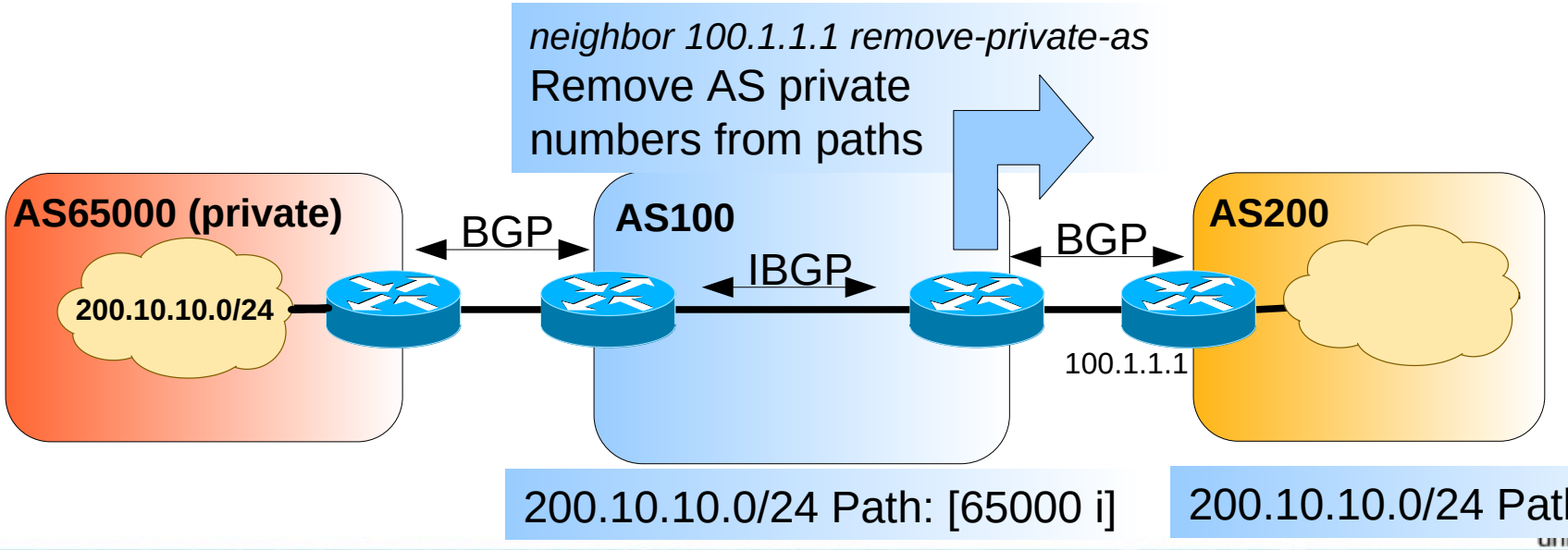
MP-BGP New Features for IPv6

- IPv6 Unicast
  - MP-BGP enables the creation of IPv6 Inter-AS relations
- IPv6 Multicast
  - Unicast prefixes for Reverse Path Forwarding (RPF) checking
  - RPF information is disseminated between autonomous systems
  - Compatible with single domain Rendezvous Points or Protocol Independent Multicast-Source Specific Multicast (PIM-SSM)
  - Topology can be congruent or non-congruent with the unicast one
- IPv6 and label (6PE)
  - IPv6 packet is transported over an IPv4 MPLS backbone
- IPv6 VPN (6VPE)
  - Multiple IPv6 VPNs are created over an IPv4 MPLS backbone

Advanced BGP

Private BGP AS

- Private autonomous system (AS) numbers range from 64512 to 65535
- When a customer network is large, the ISP may assign an AS number:
  - Permanently assigning a **Public** AS number in the range of 1 to 64511
    - Should have a unique AS number to propagate its BGP routes to Internet
    - Done when a customer network connects to two different ISPs, such as multihoming
  - Assigning a **Private** AS number in the range of 64512 to 65535.
    - It is not recommended that you use a private AS number when planning to connect to multiple ISPs in the future



BGP AS Routing Policies

```
aut-num:      AS15525
as-name:      PTPRIMENET
descr:        PT Prime Autonomous System
descr:        Corporate Data Communications Services
descr:        Portugal
import:       from AS1930 action pref=100;
              accept AS-RCCN # RCCN
import:       from AS3243 action pref=200;
              accept AS-TELEPAC # Telepac
import:       from AS5516 action pref=100;
              accept AS5516 # INESC
import:       from AS5533 action pref=100;
              accept AS-VIAPT # Via NetWorks Portugal
import:       from AS8657 action pref=300;
              accept ANY # CPRM
import:       from AS12305 action pref=100;
              accept AS12305 # Nortenet
import:       from AS1897 action pref=100;
              accept AS1897 AS9190 AS13134 AS15931 # KPN Qwest
import:       from AS13156 action pref=100;
              accept AS13156 # Cabovisao
import:       from AS8824 action pref=100;
              accept AS8824 AS15919 # Eastecnica
.................
```

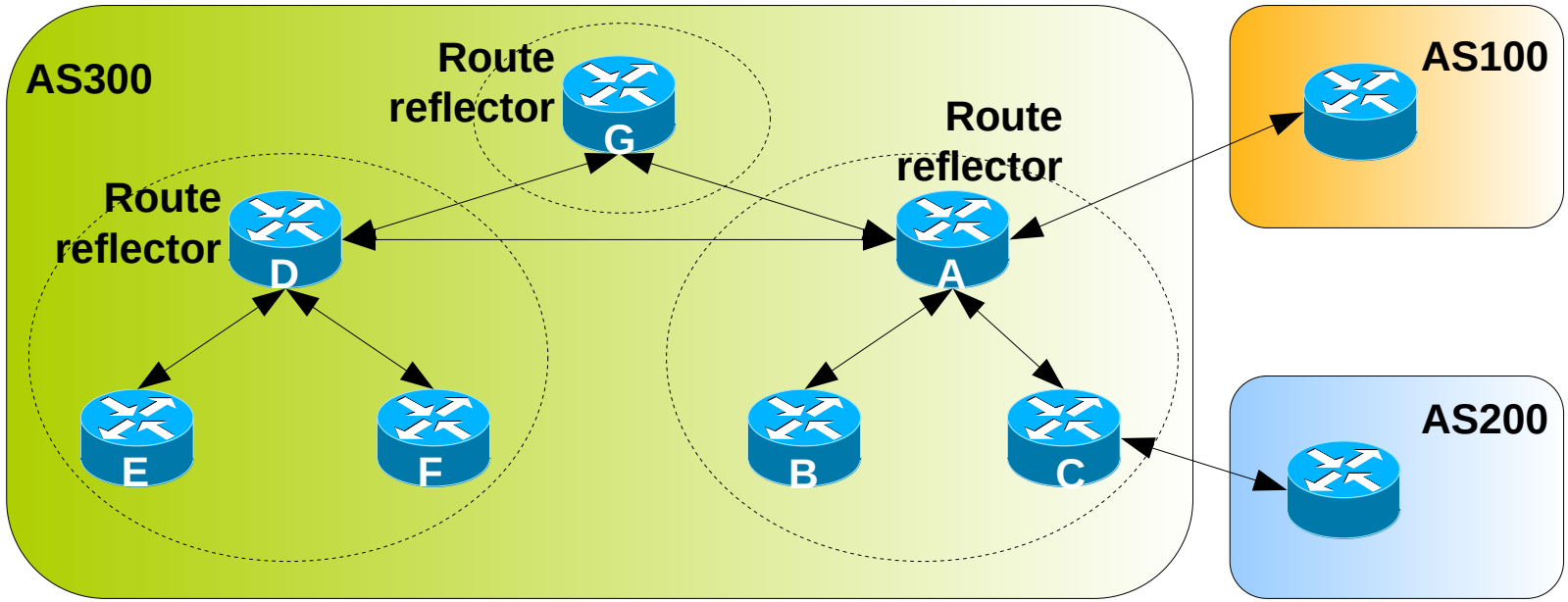
```
export:       to AS1897 announce RS-PTPRIME # KPNQwest
export:       to AS1930 announce RS-PTPRIME # RCCN
export:       to AS3243 announce RS-PTPRIME # Telepac
export:       to AS5516 announce {0.0.0.0/0} # INESC
export:       to AS5533 announce RS-PTPRIME # Via NetWorks Portugal
export:       to AS8657 announce RS-PTPRIME # CPRM
export:       to AS8824 announce RS-PTPRIME  # Eastecnica
export:       to AS8826 announce {0.0.0.0/0} # Siemens
export:       to AS9186 announce RS-PTPRIME # ONI
export:       to AS12305 announce RS-PTPRIME # Nortenet
export:       to AS12353 announce RS-PTPRIME # Vodafone Portugal
export:       to AS13156 announce RS-PTPRIME # Cabovisao
export:       to AS13910 announce ANY # register.com
export:       to AS15931 announce ANY # YASP Hiperbit
export:       to AS24698 announce RS-PTPRIME # Optimus
export:       to AS25005 announce ANY # Finibanco
export:       to AS25253 announce {0.0.0.0/0} # CGDNet
export:       to AS28672 announce ANY # BPN
export:       to AS31401 announce {0.0.0.0/0} # SICAMSERV
export:       to AS39088 announce {0.0.0.0/0} # Santander-Totta
export:       to AS41345 announce RS-PTPRIME  # Visabeira
export:       to AS43064 announce  RS-PTPRIME # Teixeira Duarte
export:       to AS43643 announce  ANY # TAP
.........................
```

From RIPE database
http://www.db.ripe.net

BGP Synchronization

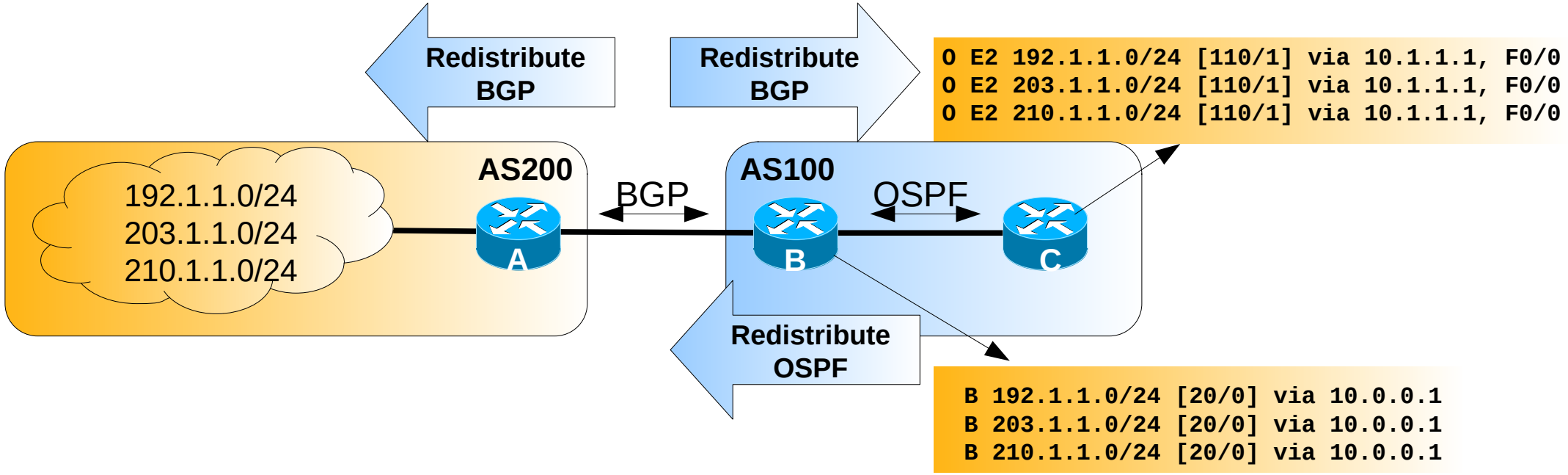
- Synchronization states that, if your AS passes traffic from another AS to a third AS, BGP should not advertise a route before all the routers in your AS have learned about the route via IGP.
- BGP waits until IGP has propagated the route within the AS. Then, BGP advertises the route to external peers.

BGP Route Reflectors



- Without a route reflector, the network requires a full iBGP mesh within AS300.
- The route reflector and its clients are called a cluster.
  - Router A is configured as a route reflector, iBGP peering between Routers B and C (and others) is not required.
  - Router D is configured as a route reflector, iBGP peering between Routers E and F (and others) is not required.
- Full IBGP mesh between route reflector Routers.

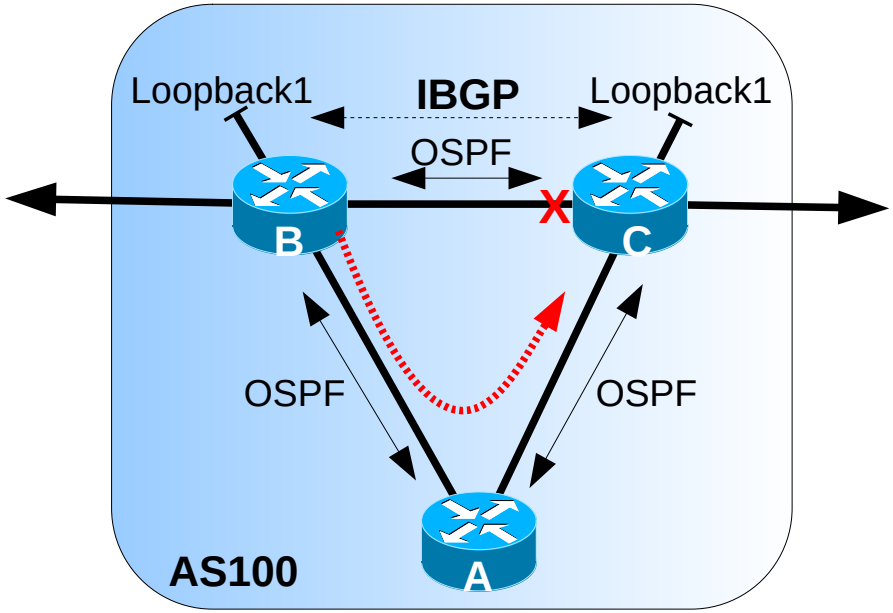
Routes Redistribution



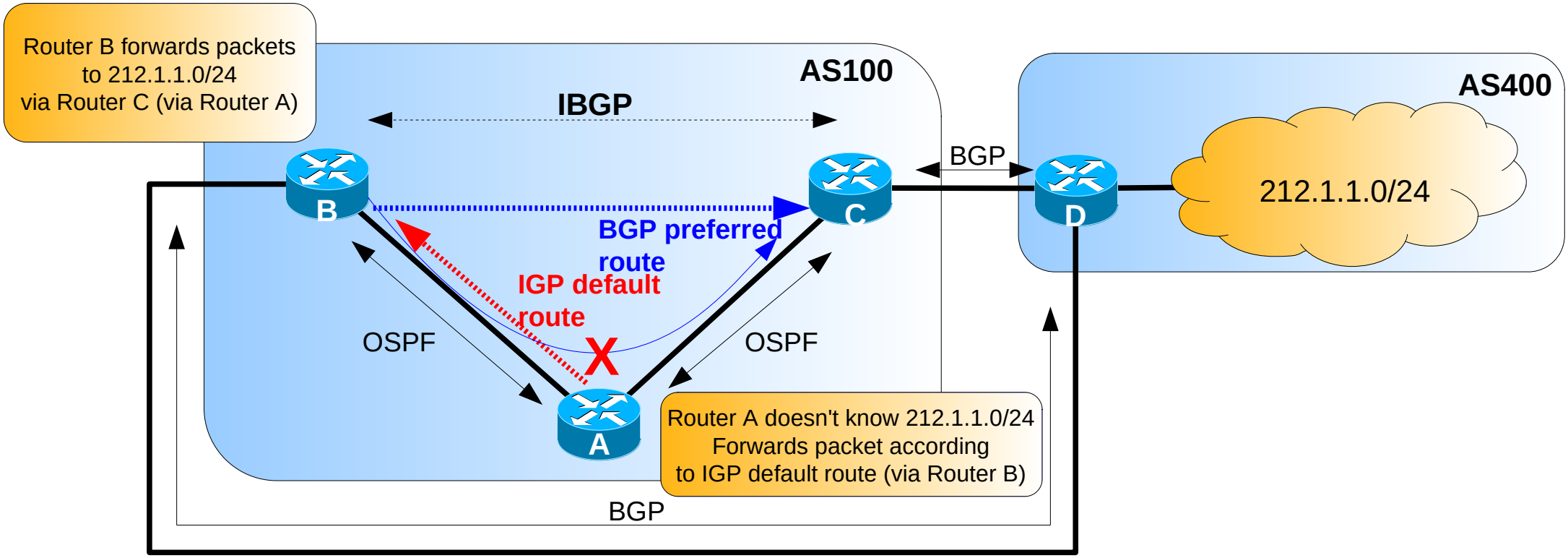
- Redistributing IGP routes by BGP will:
  - Simplify BGP configuration (advantage)
  - And BGP will announce only internal networks with connectivity (advantage)
- Redistributing BGP routes by IGP protocols will:
  - Make internal routes know all external routes (disadvantage/advantage?)
  - Increase routing tables size in internal routers (disadvantage)
    - Decrease routing time, imposes memory requirements, ...
  - Avoid the usage of internal default routes (disadvantage/advantage?)

BGP Neighborhood Resilience

- BGP neighbor relations between physical interfaces are dependent on interface stability/status
- (Virtual) neighbor relations using Loopback interfaces/addresses
  - Loopback interfaces are virtual and software based
    - If the router is active Loopback interfaces are always active
  - Neighbor relation is active while a path exists between the virtual networks
    - (Alternative) Routing provided by IGPs

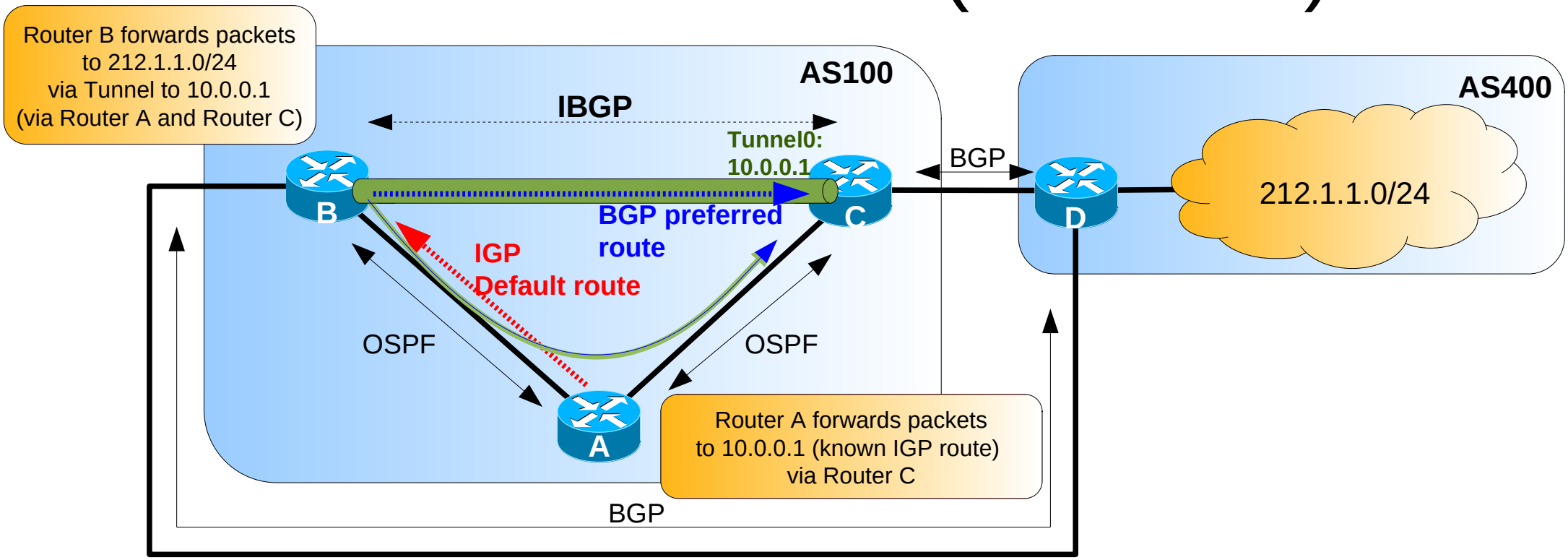


BGP and IGP conflicts



- Routing conflicts may arise with
  - Internal routers without BGP
  - No redistribution of BGP routes by IGP
  - IGP default routes
  - BGP preferred routes (with no agreement with IGP default routes)
- Solutions
  - Adjust IGP default routes
  - Adjust BGP preferred routes (e.g. with local preference)
  - BGP neighborhood and Internal routing via IP-IP tunnels

BGP over Tunnels (over IGP)

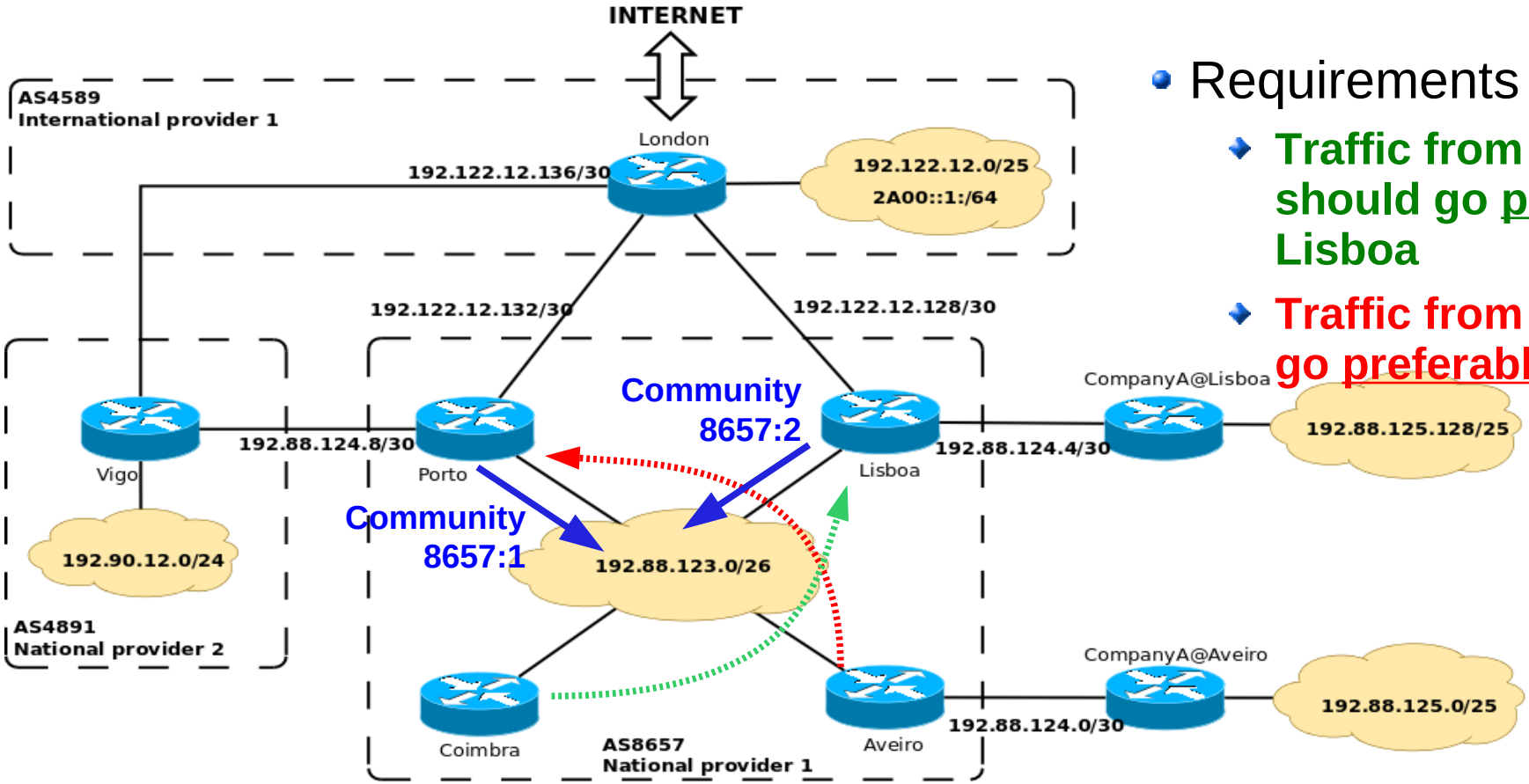


- IP-IP tunnels to solve BGP/IGP routing conflicts
  - Tunnels manually configured
    - Between physical or Loopback interfaces
  - BGP neighborhood via Tunnel
  - BGP routes learned via Tunnel (next hop is remote Tunnel end-point)
  - Tunnel “network” distributed internally via IGP
- In Router A, to any packet destined to an outside network it's forwarded via Tunnel
  - A new IP header is added, new IP destination address is the remote Tunnel end-point
  - Internally, packet is routed according to the new IP header (Tunnel end-points IP addresses)

BGP Filtering and Route Maps

- Sending and receiving BGP updates can be controlled by using a number of different filtering methods.
- BGP updates can be filtered based on:
  - Route information,
  - Path information,
  - Communities.
- Route maps are used with BGP to
  - Control and modify routing information.
  - Define the conditions by which routes are redistributed between routing domains.

BGP Case Studies



- Requirements
  - **Traffic from Coimbra should go preferably by Lisboa**
  - **Traffic from Aveiro should go preferably by Porto**

- @Porto
  - Route-map applied to all BGP announced external routes/nets
  - Adds BGP attribute: **Community 8657:1**
- @Lisboa
  - Route-map applied to all BGP announced external routes/nets
  - Adds BGP attribute: **Community 8657:2**

- @Aveiro
  - Route-map applied to all BGP received routes/nets
  - If **Community 8657:1** → **Local-preference 200**
  - If **Community 8657:2** → **Local-preference 100**
- @Coimbra
  - Route-map applied to all BGP received routes/nets
  - If **Community 8657:1** → **Local-preference 100**
  - If **Community 8657:2** → **Local-preference 200**

BGP Community Attribute (real data)

- TeliaNet Global Network

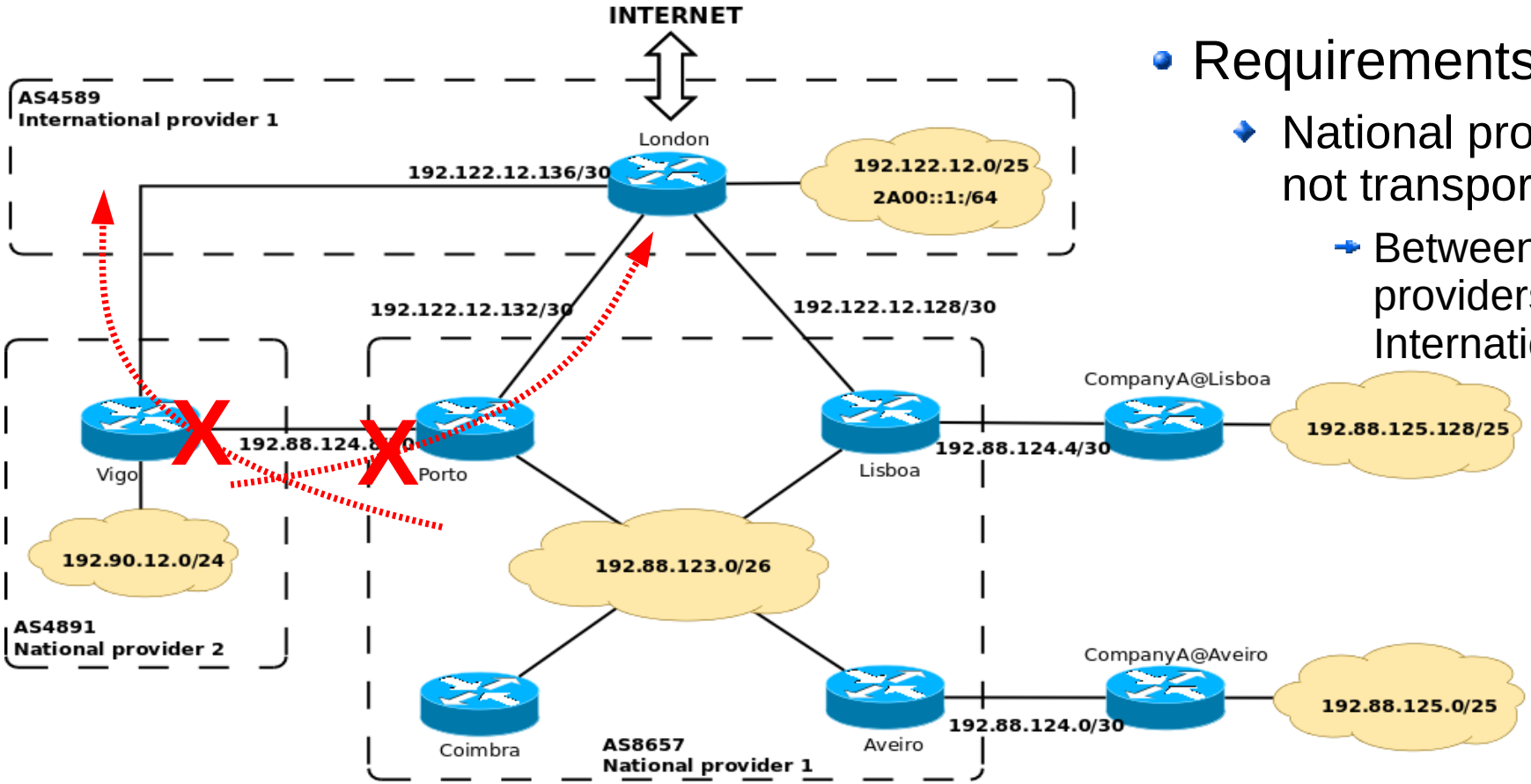
```
remarks:        BGP COMMUNITY SUPPORT FOR AS1299 TRANSIT CUSTOMERS:
remarks:
remarks:        Community Action
remarks:        --------- ------
remarks:        1299:50   Set local pref 50 within AS1299 (lowest possible)
remarks:        1299:150  Set local pref 150 within AS1299 (equal to peer, backup)
remarks:
remarks:        European peers/ix-points          US peers/ix-points              Asia peers/ix-points
remarks:        Community  Action                Community  Action               Community  Action
remarks:        ---------  ----------------------  ---------  ----------------------  ---------  ----------------------
remarks:        1299:200x  All peers Europe incl:    1299:500x  All peers US incl:       1299:700x  All peers Asia incl:
...                -
remarks:        1299:250x   Sprint/1239           1299:550x   Sprint/1239           -
remarks:        1299:251x   Savvis/3561           1299:551x   Savvis/3561           -
remarks:        1299:252x   Verio/2914            1299:552x   Verio/2914            -
remarks:        1299:253x   Abovenet/6461         1299:553x   Abovenet/6461         -
remarks:        1299:254x   FT/5511               1299:554x   FT/5511               1299:754x   FT/5511
remarks:        1299:255x   GBLX/3549             1299:555x   GBLX/3549             1299:755x   GBLX/3549
remarks:        1299:256x   Level3/3356           1299:556x   Level3/3356           -
remarks:        1299:257x   UUnet/702             1299:557x   UUnet/701             -
remarks:        1299:558x   AT&T/7018             1299:758x   AT&T/2687
remarks:        1299:259x   Telefonica/12956       1299:559x   Telefonica/12956       -
remarks:        1299:260x   BT/Concert/5400        -                                  -
remarks:        1299:261x   Qwest/209             1299:561x   Qwest/209             -
remarks:        1299:263x   Teleglobe/6453         1299:563x   Teleglobe/6453         -
remarks:        1299:264x   DTAG/3320             1299:564x   DTAG/3320             -
remarks:        1299:268x   AOL/1668              1299:568x   AOL/1668              -
remarks:        1299:269x   Tiscali/3257          1299:569x   Tiscali/3257          1299:769x   Tiscali/3257
remarks:        1299:270x   UPC/6830              -                                  -
remarks:        1299:273x   Cogent/174            1299:573x   Cogent/174            -
remarks:        1299:274x   Telecom Italia/6762     1299:574x   Telecom Italia/6762     1299:774x   Telecom Italia/6762
remarks:        1299:275x   Tele2/1257            -                                  -
...
remarks:        1299:284x   Cable & Wireless DE/1273 1299:584x   Cable & Wireless DE/1273 -
remarks:        1299:286x   KPN/286               -                                  -
remarks:        1299:287x   China Netcom/4837       1299:587x   China N
remarks:        1299:288x   China Telecom/4134      1299:588x   China T
```

From RIPE database

https://apps.db.ripe.net/

e.g., https://apps.db.ripe.net/db-web-ui/#/query?bflag=false&dflag=false&rflag=true&searchtext=as1299

BGP Case Studies

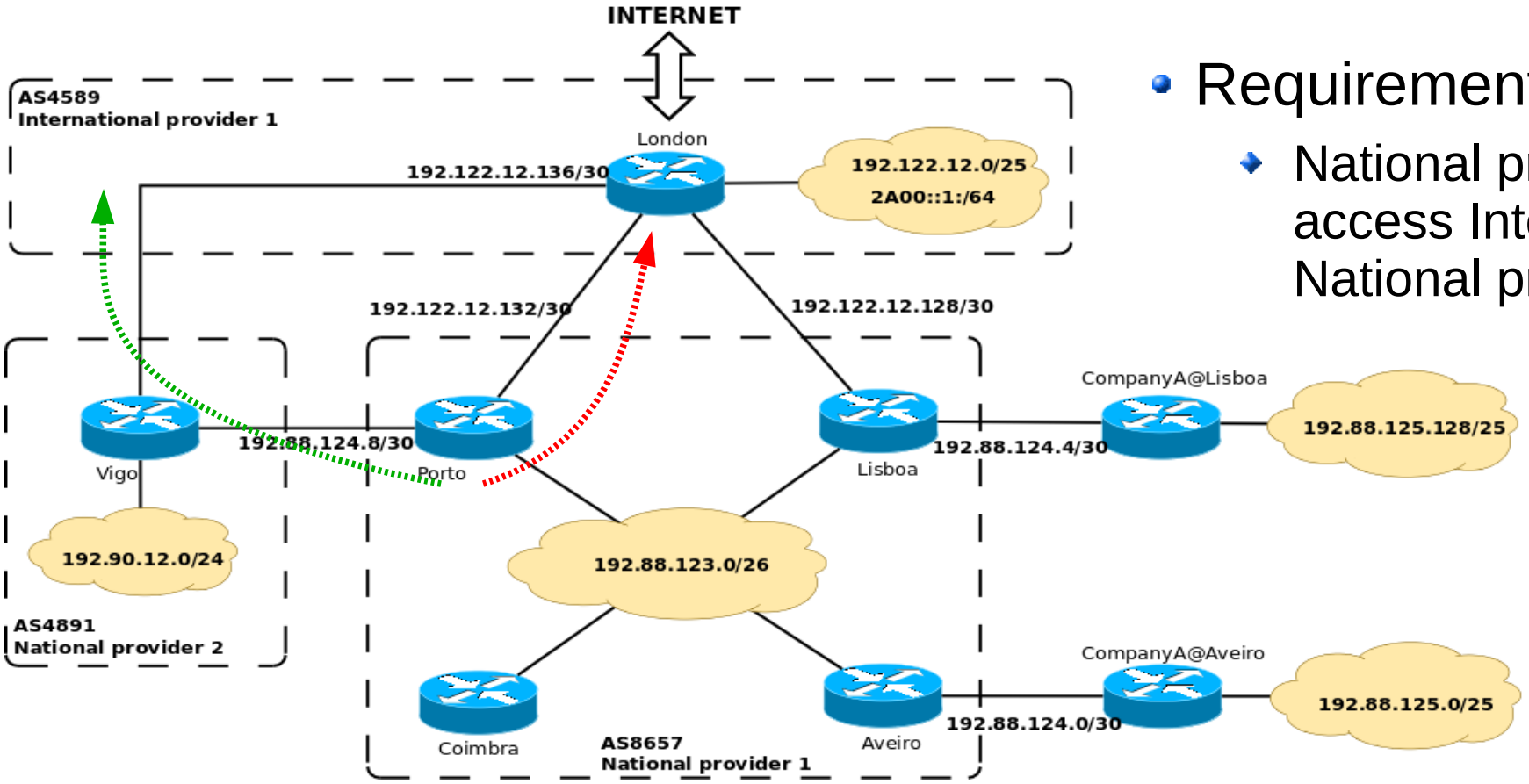


- Requirements
  - National providers should not transport traffic
    - Between other national providers and the International provider

- @Porto, @Lisboa
  - Route-map applied to all external BGP announcements
  - Announce only internal routes/nets
    - Empty path “^$”

- @Vigo
  - Route-map applied to all external BGP announcements
  - Announce only internal routes/nets
    - Empty path “^$”

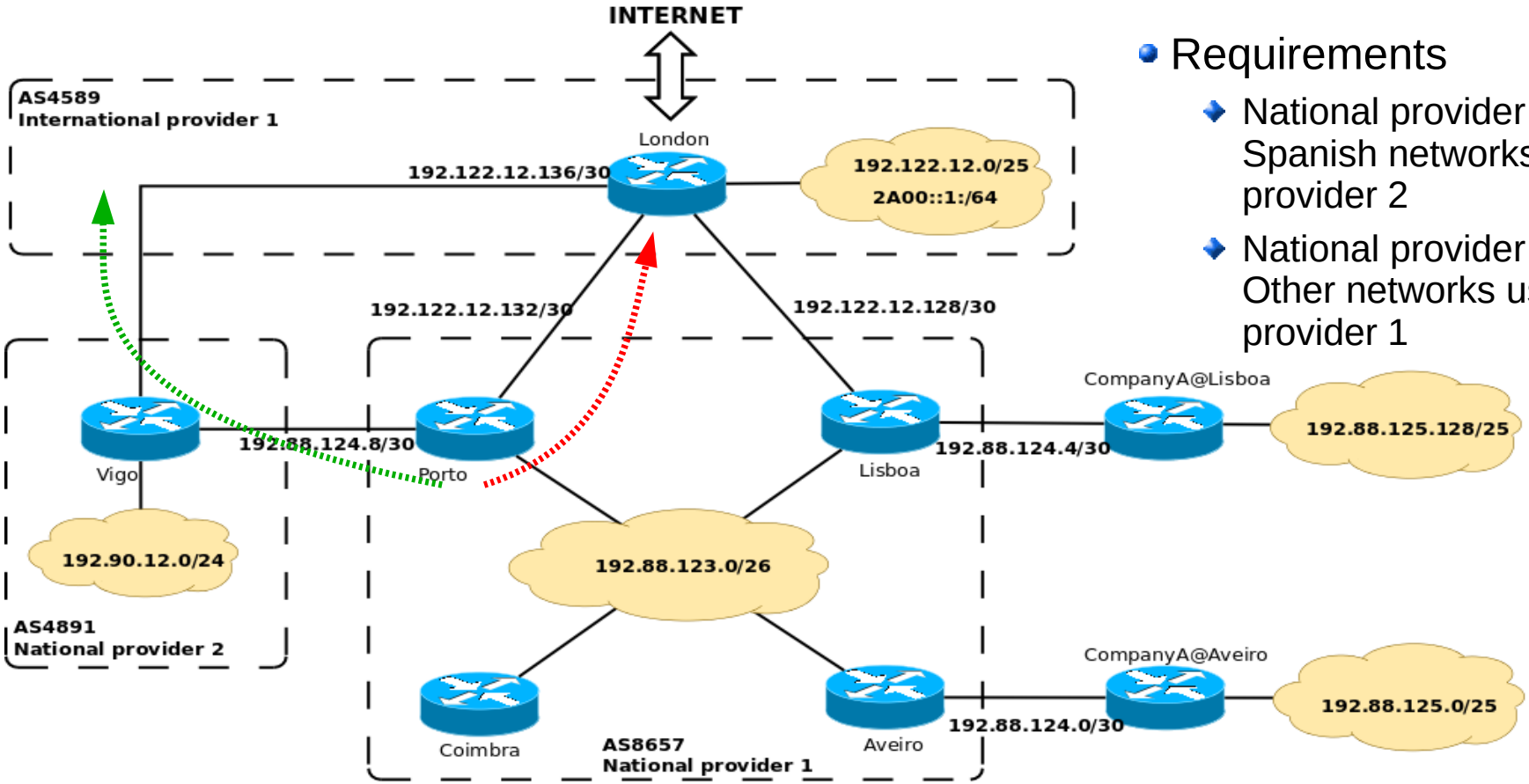
BGP Case Studies



- Requirements
  - National provider 1 should access Internet using National provider 2

- @Porto, @Lisboa
  - Route-map applied to all BGP announcements received
  - If Path contains “4891” → **Local-preference 200**
  - If Path does not contain “4891” → **Local-preference 100**

BGP Case Studies

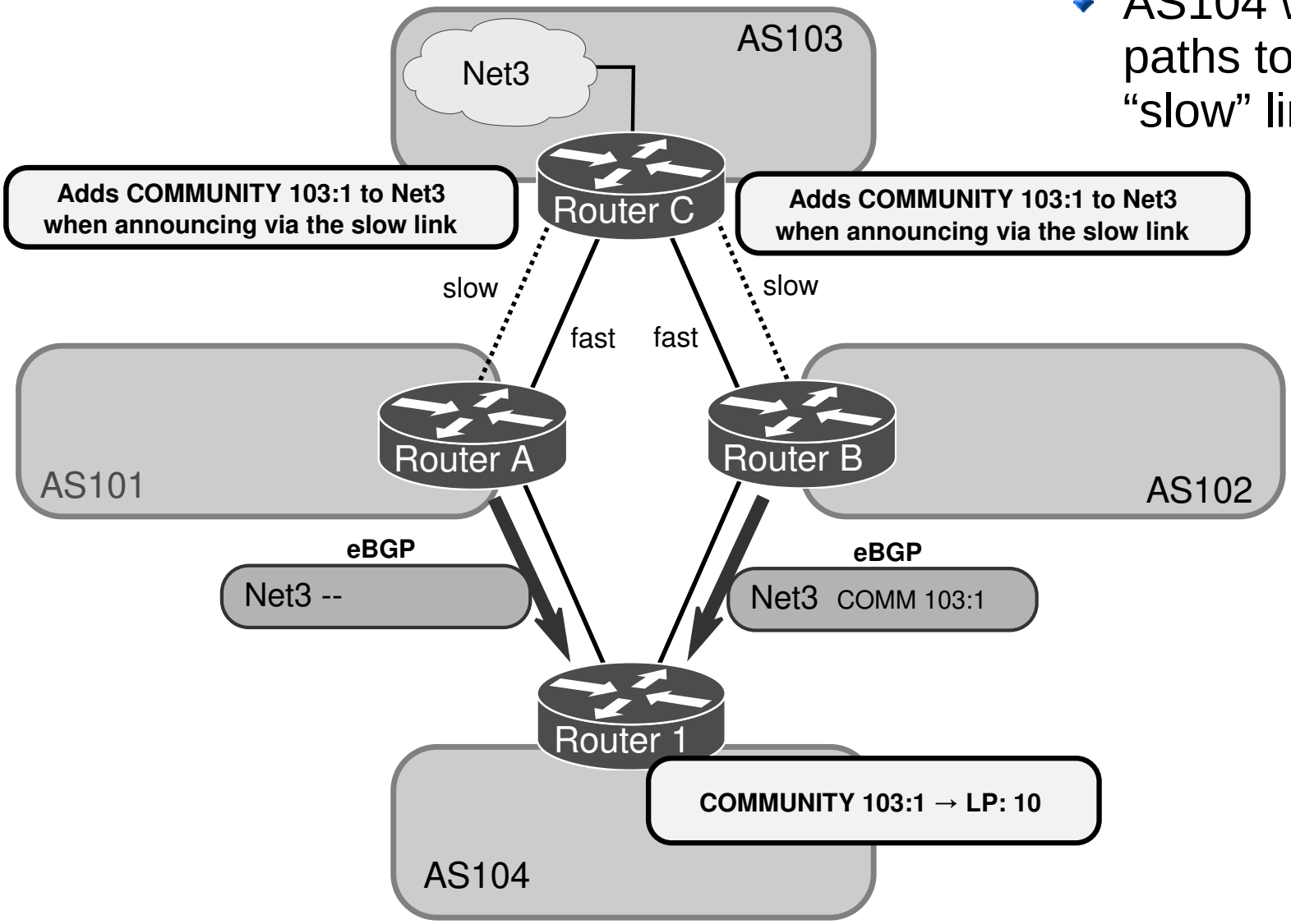


- Requirements
  - National provider 1 should access Spanish networks using National provider 2
  - National provider 1 should access Other networks using International provider 1

- @Porto, @Lisboa
  - Route-map applied to all BGP announcements received
    - E.g. known Spanish operators AS: 4891, 7654, 9876 and 3352
  - If Path starts (from right to left) with “4891$ or 7654$ or 9876$ or 3352$” and ends in “^4891” → **Local-preference 200**
  - If Path does not start with “4891$ or 7654$ or 9876$ or 3352$” and ends in “^4891” → **Local-preference 50**
  - Assuming default Local-preference 100.

BGP Case Studies

- Requirements
  - AS104 wants to avoid paths to Net3 that use “slow” links.



Traffic Tunneling
&
Overlay Networks

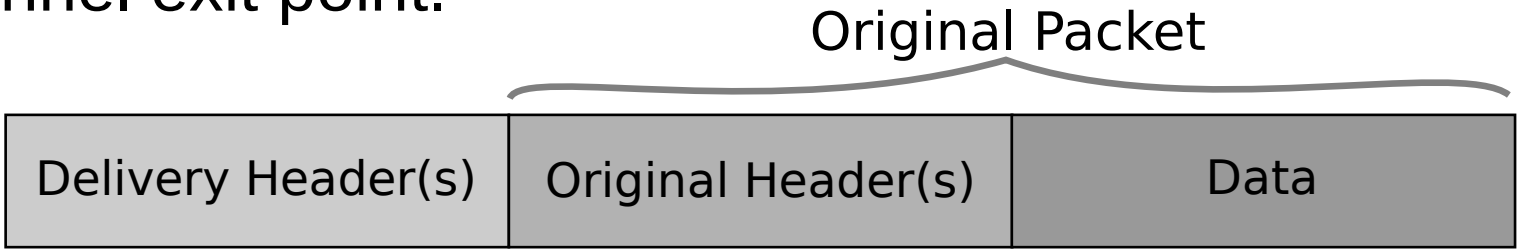
Redes de Comunicações II

Licenciatura em
Engenharia de Computadores e Informática
DETI-UA

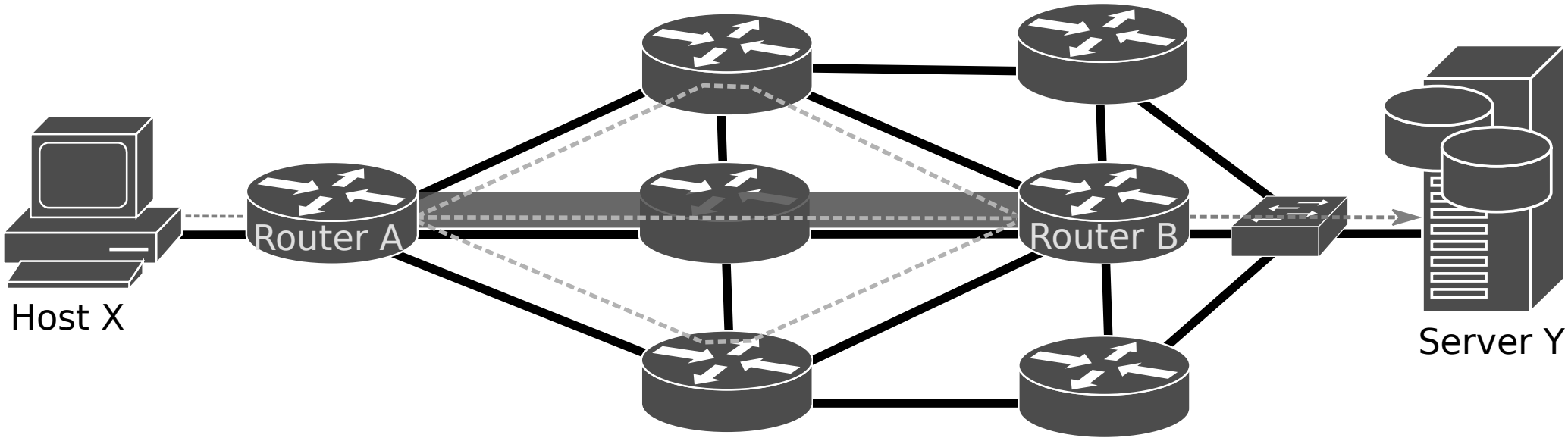
universidade de aveiro
deti.ua.pt

Traffic Tunnel Concept

- Main purposes
  - Guarantee that a packet that reaches a network node will reach a specific secondary network node independently of the intermediary nodes routing processes,
  - Guarantee the delivery of a packet to a remote node when the intermediary nodes do not support the original packet network protocol, and,
  - Define a virtual channel that adds additional data transport features in order to provide differentiated QoS, security requirements and/or optimized routing.
- Achieved by adding, at the tunnel entry point, one or more protocol headers to the original packets to handle their delivery to the tunnel exit point.



Tunnel End-Points



Delivery protocol(s)	Original protocol(s)	Data
Source: A address Destination: B address	Source: X address Destination: Y address	

Virtual Tunnel Interface (VTI)

- Logical construction that creates a virtual network interface that can be handled as any other network interface within a network equipment.
- A tunnel does not require to have any network addresses other the ones already bound to the end-point router.
- However, most implementations impose that a network address must be bound to a tunnel interface in order to enable IP processing on the interface.
  - The tunnel interface may have a explicitly bound network address or reuse an address of another interface already configured on the router.

```
#interface Tunnel 1
 #ip address 10.1.1.1 255.255.255.252
 #ipv6 address 2001:A:A::1/64
 #ip unnumbered FastEthernet0/0
 #ipv6 unnumbered FastEthernet0/0
 #ip ospf cost 10
 #ipv6 ospf 1 area 0
 #tunnel mode ipip
 #tunnel source FastEthernet0/0
 #tunnel destination 200.2.2.2
```

VTI Requirements

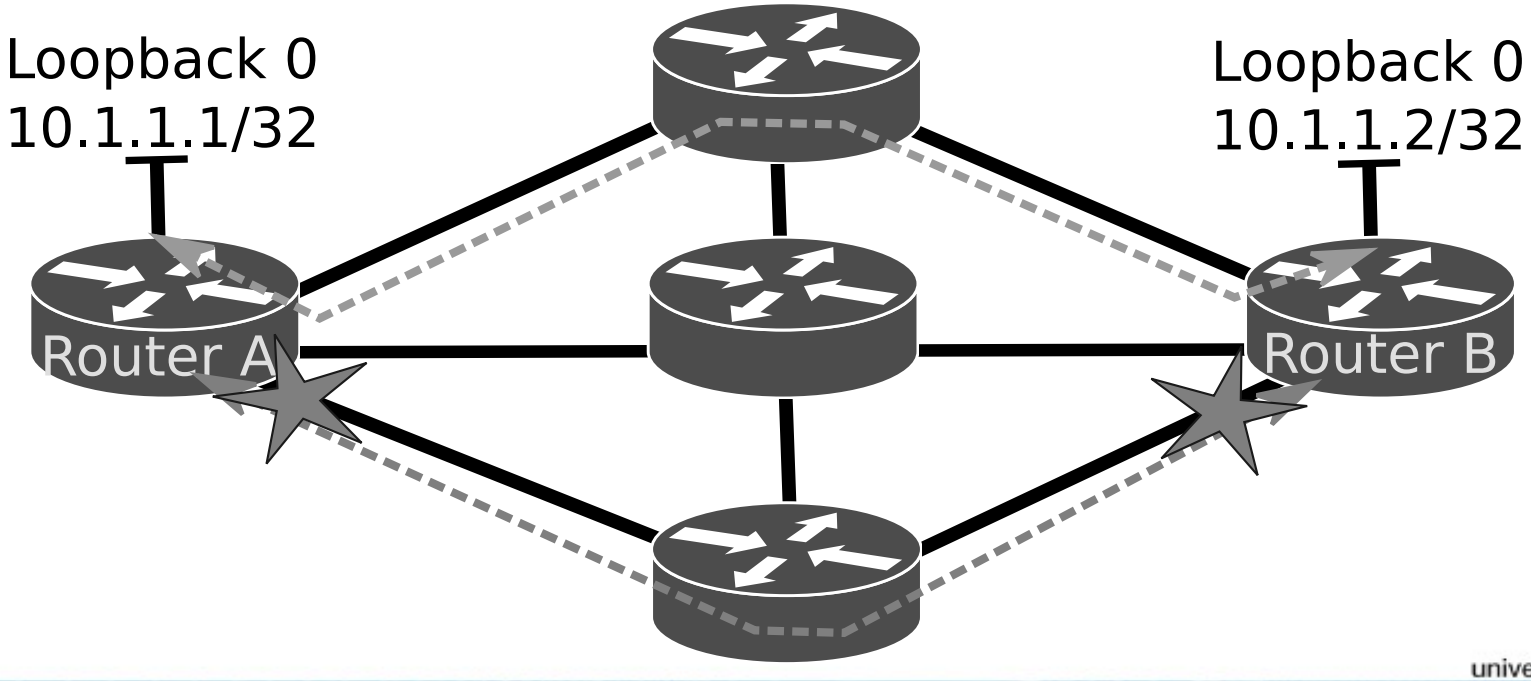
- A numeric identifier,
- A bounded IP address, this will enable IP processing,
  - Add the tunnel interface to the routing table and allow routing via the interface,

```
1  #interface Tunnel 1
2   #ip address 10.1.1.1 255.255.255.252
3   #ipv6 address 2001:A:A::1/64
4   #ip unnumbered FastEthernet0/0
5   #ipv6 unnumbered FastEthernet0/0
6   #ip ospf cost 10
7   #ipv6 ospf 1 area 0
8   #tunnel mode ipip
9   #tunnel source FastEthernet0/0
10  #tunnel destination 200.2.2.2
```

- A defined mode or type of tunnel,
  - Availability of tunnel models depends on the Router model, operating software and licenses.
- Tunnel source,
  - Defined as the name of the local interface or IPv4/IPv6 address depending on the type of the tunnel.
- Tunnel destination,
  - Defined as a domain name or IPv4/IPv6 address depending on the type of the tunnel.
  - This definition is not mandatory for all types of tunnels because in some cases the tunnel end-point is determined dynamically.
- May optionally have additional configurations for routing, security and QoS purposes.

Loopback Interfaces as End-Points

- Loopback interface is another logical construction that creates a virtual network interface completely independent from the remaining physical and logical router network interfaces.
- The main propose of a loopback interface is to provide a network address to serve as router identifier in remote network configurations and distribute algorithms.
- The main advantage of using loopback interfaces as tunnel end-points, is the creation of a tunnel not bounded to any individual network card/link that may fail.

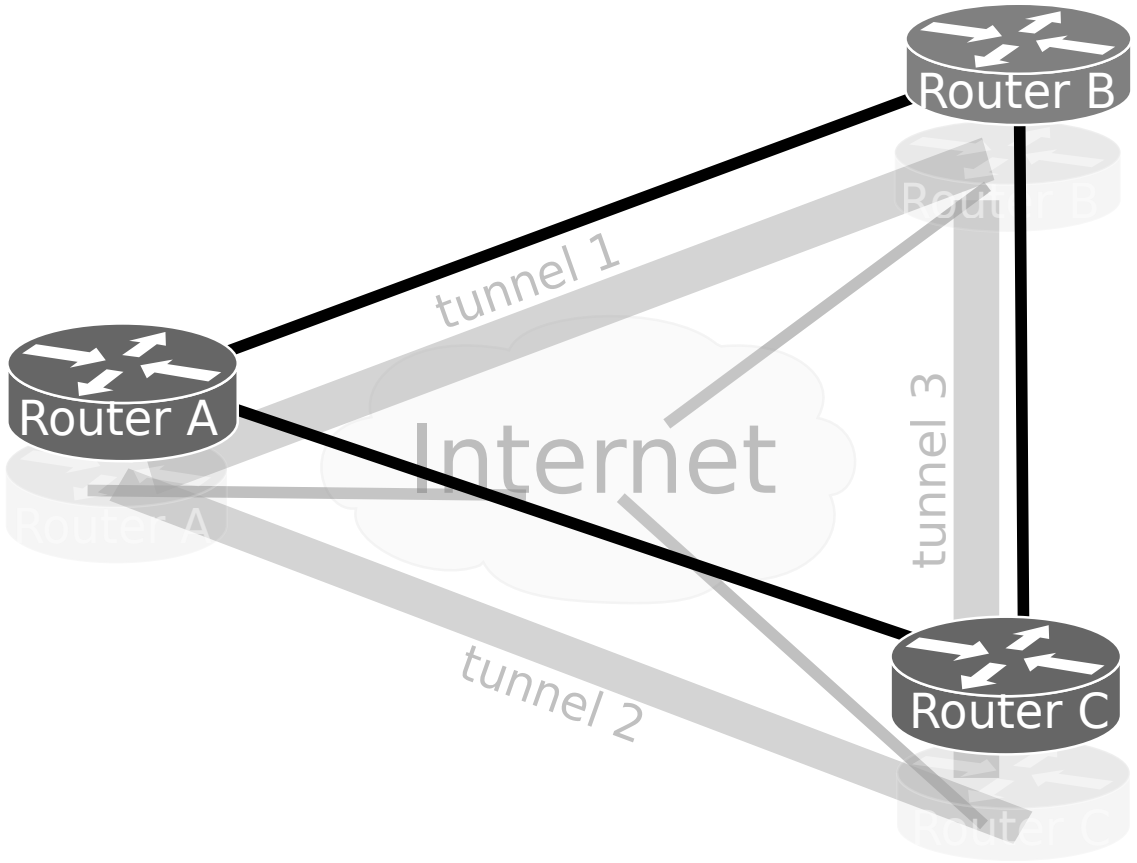


IP Tunnel Types

- IPv4-IPv4
  - Original IPv4 packets are delivered using IPV4 as network protocol.
- GRE IPv4
  - Original packets protocol (any network protocol) is defined by GRE header and delivered using IPv4 as network protocol.
- IPv6-IPv6
  - Original IPv6 packets are delivered using IPv6 as network protocol.
- GRE IPv6
  - Original packets protocol (any network protocol) is defined by a GRE header and delivered using IPv6 as network protocol.
- IPv6-IPv4
  - Original IPv6 packets are delivered using IPv4 as network protocol.
- IPv4-IPv6
  - Original IPv4 packets are delivered using IPv6 as network protocol.

Overlay Network

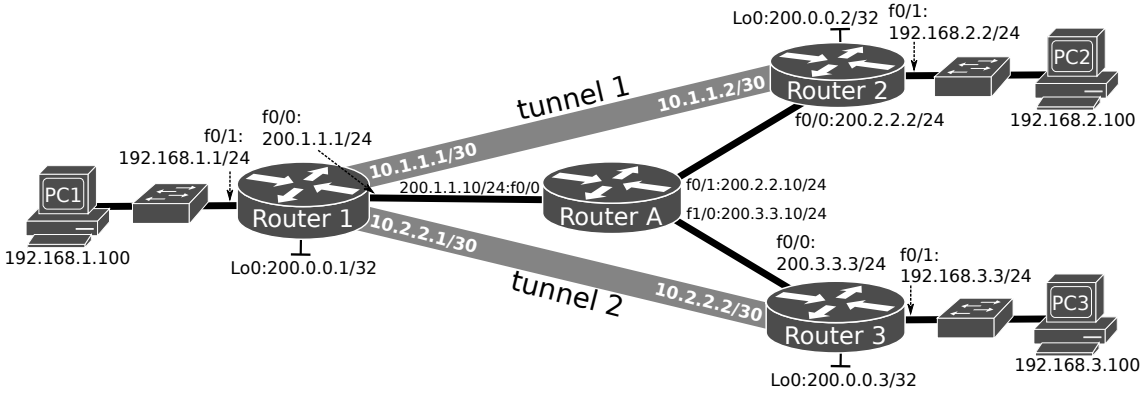
- An overlay network can be defined as a virtual network defined over another network.
  - For a specific purpose like private transport/routing policies, QoS, security.
- The underlying network can be physical or also virtual.
  - May result in multiple layers of overlay networks.
- When any level of privacy protocol is present on an overlay network is designated by Virtual Private Network (VPN).



Routing Through/Between Tunnels

- Static Routes

```
#ip route 192.168.2.0 255.255.255.0 Tunnel1
#ip route 192.168.2.0 255.255.255.0 10.1.1.2
#ipv6 route 2001:A:1::/64 Tunnel1
#ipv6 route 2001:A:1::/64 2001:0:0::2
#ip route 192.168.2.100 255.255.255.255 10.1.1.2
#ipv6 route 2001:A:1::100/128 2001:0:0::2
```



- Route-maps

```
#access-list 100 permit ip host 192.168.1.100 192.168.2.0 255.255.255.0
#route-map routeT1
 #match ip address 100
 #set ip next-hop 10.1.1.2
#interface FastEthernet0/1
 #ip policy route-map routeT1
```

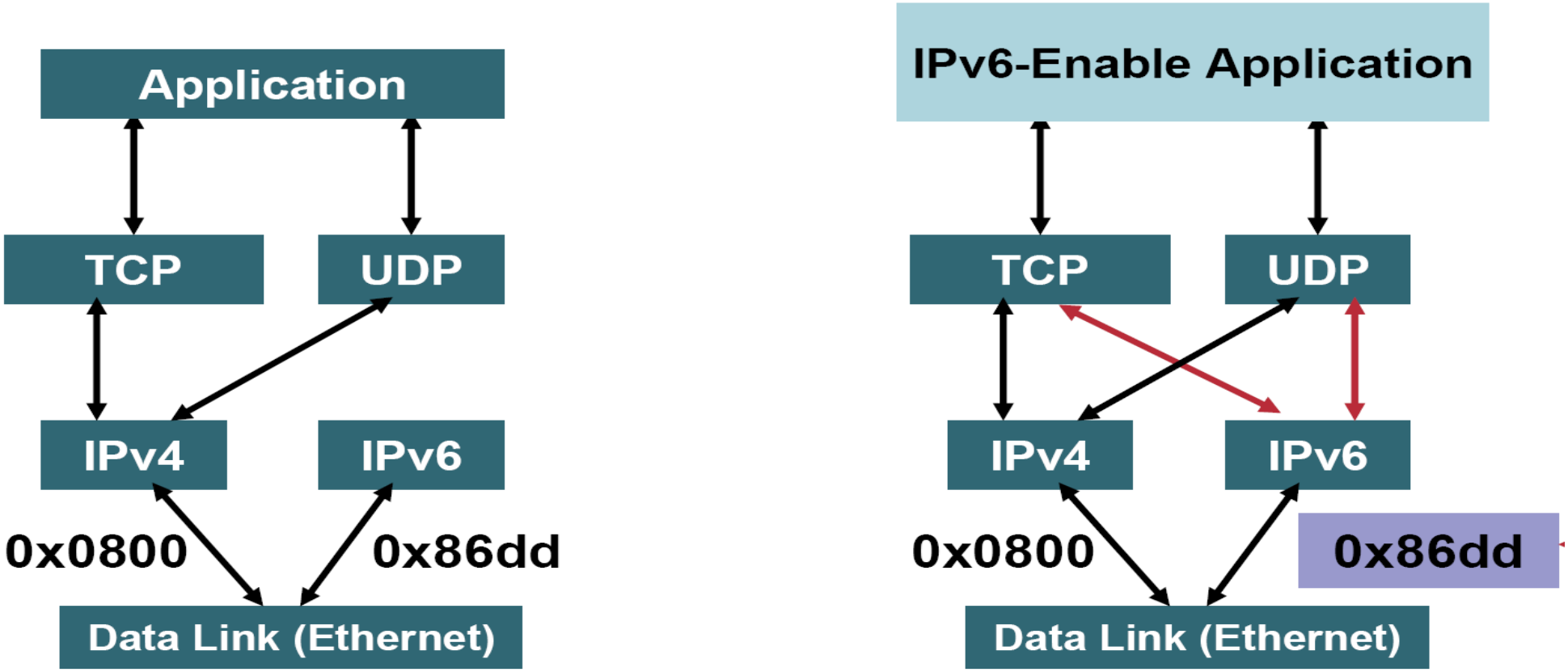
- Dynamic Routing
  - Multiple (distinct) routing processes.
    - One per overlay network, and
    - One for the underlying network.

```
#router ospf 1
 #network 200.1.1.0 0.0.0.255 area 0
 #network 200.0.0.1 0.0.0.0 area 0
!
#router ospf 2
 #network 10.0.0.0 0.255.255.255 area 0
 #network 192.168.0.0 0.0.255.255 area 1
```

IPv6 Deployment Techniques

- Deploying IPv6 using dual-stack backbones
  - IPv4 and IPv6 applications coexist in a dual IP layer routing backbone
  - All routers in the network need to be upgraded to be dual-stack
- IPv6 over IPv4 tunnels
  - Manually configured
    - With and without Generic Routing Encapsulation (GRE)
  - Semiautomatic tunnel mechanisms
  - Fully automatic tunnel mechanisms (IPv4-compatible and 6to4)

Dual Stack



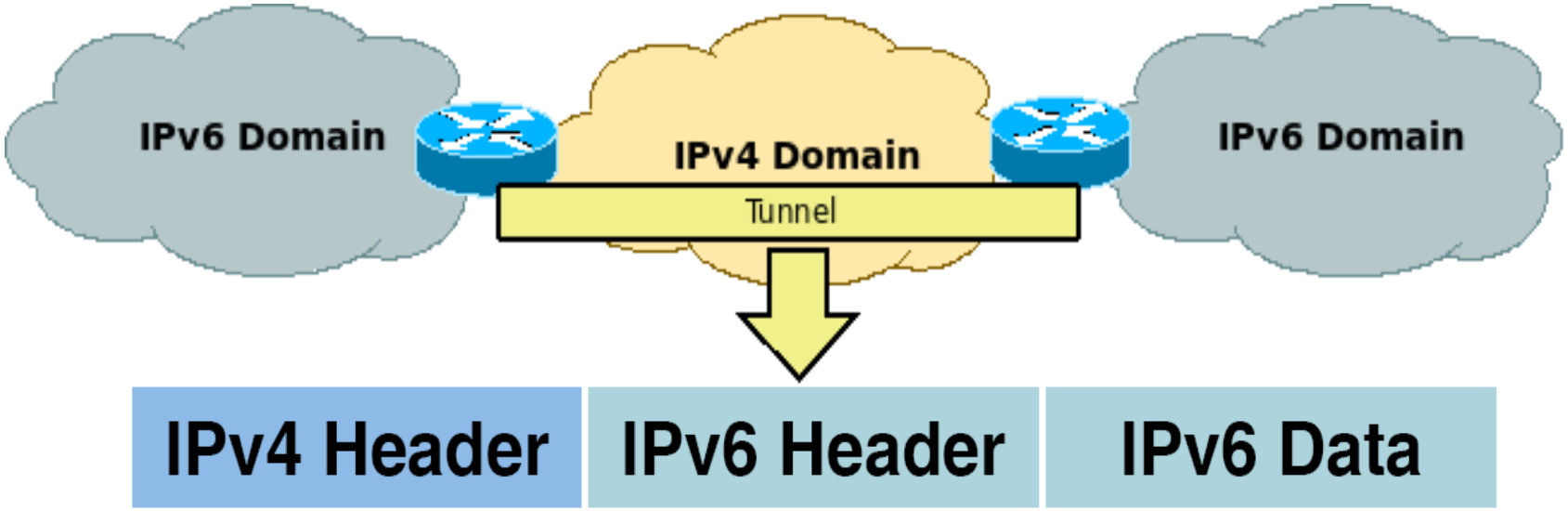
- Applications may talk to both
- Choice of the IP version is based on DNS responses and application preferences

IPv6 Overlay Tunneling

- Manual
  - IPv6 Manually Configured IPv6 over IPv4
  - IPv6 over IPv4 GRE Tunnel
- Semi-automatic mechanisms
  - Tunnel Broker
  - Teredo
  - Dual Stack Transition Mechanism (DSTM)
- Automatic mechanisms
  - Automatic IPv4 Compatible Tunnel (deprecated)
  - 6to4 Tunnel
  - ISATAP Tunnels

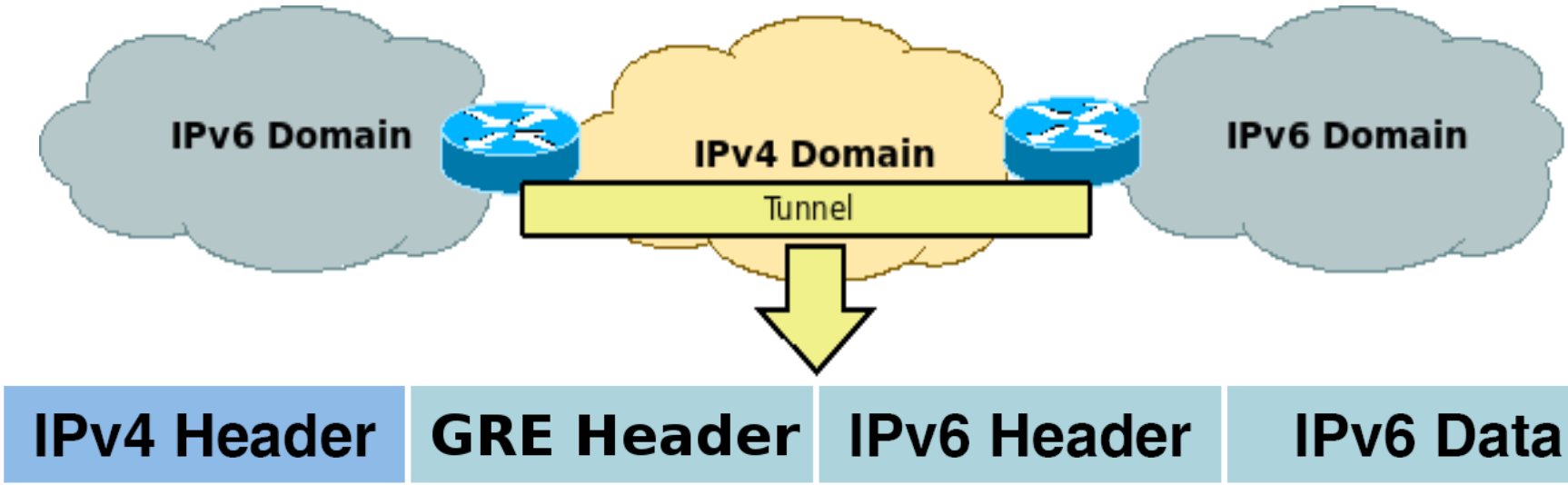
IPv6 Manually Configured

- Permanent link between two IPv6 domains over an IPv4 backbone
- Primary use is for stable connections that require regular secure communication between
  - Two edge routers, end system and an edge router, or for connection to remote IPv6 networks
- Tunnel between two points
- Complex management



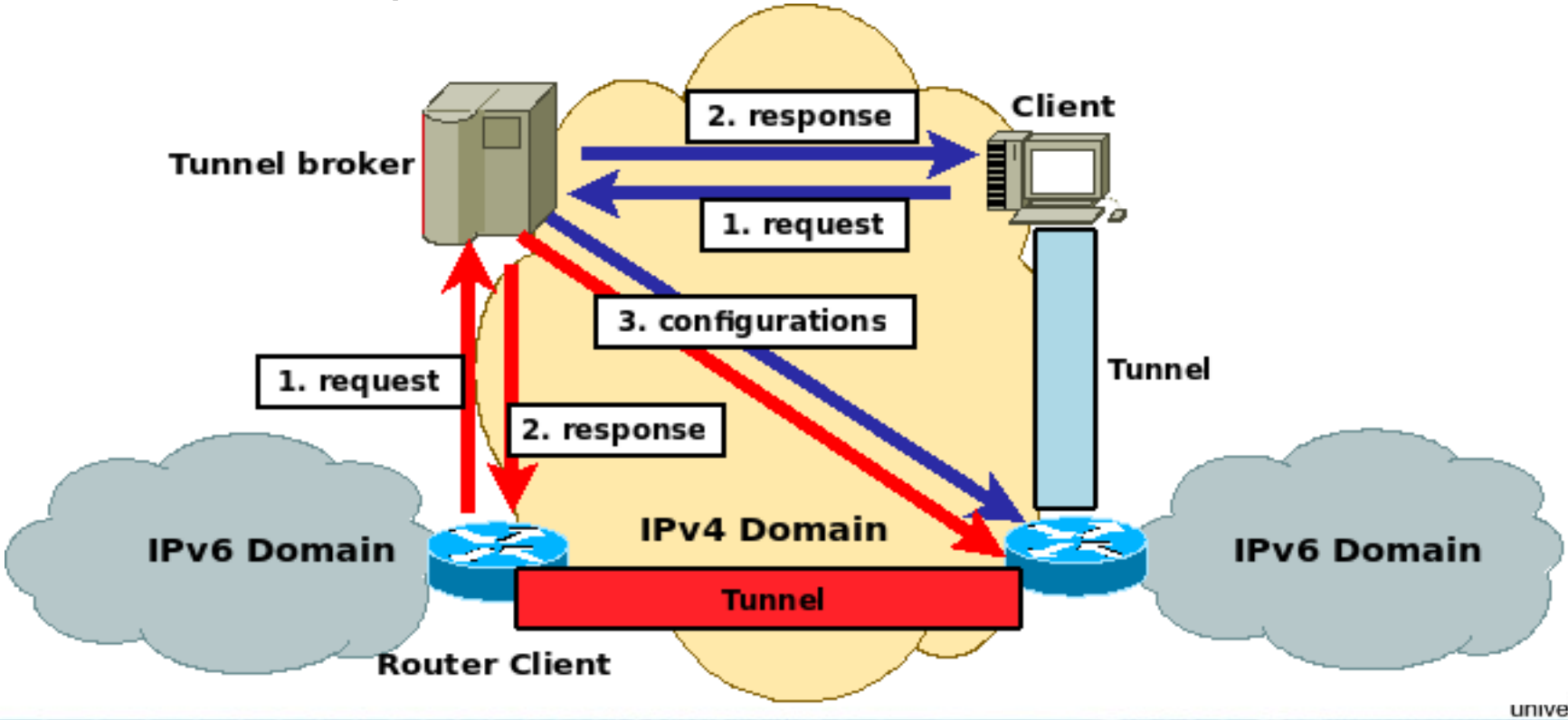
IPv6 over IPv4 GRE Tunnel

- Uses the standard GRE tunneling technique
  - GRE – Generic Route Encapsulation
- Also must be manually configured
- Primary use is for stable connections that require regular stable communications
- IPv4 over IPv6 also possible



Tunnel Broker

- A tunnel broker service allows IPv6 applications on dual-stack systems access to an IPv6 backbone
- Automatically manages tunnel requests and configuration
- Potential security implications
  - Broker is a single point of failure
- Most common implementation: Teredo.

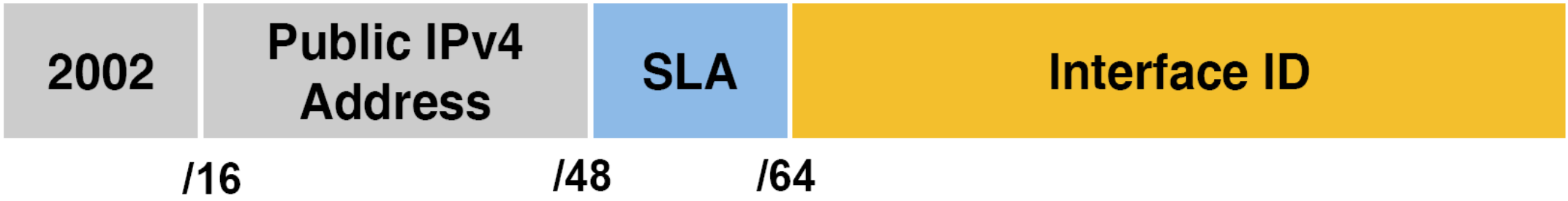


Automatic IPv4 Compatible Tunnel

- IPv4 tunnel end-point address is embedded within the destination IPv6 address
- An automatic IPv4-compatible tunnel can be configured between edge routers or between an edge router and an end system.
- Systems must be dual-stack
- Communication only with other IPv4-compatible sites
- This tunneling technique is currently deprecated

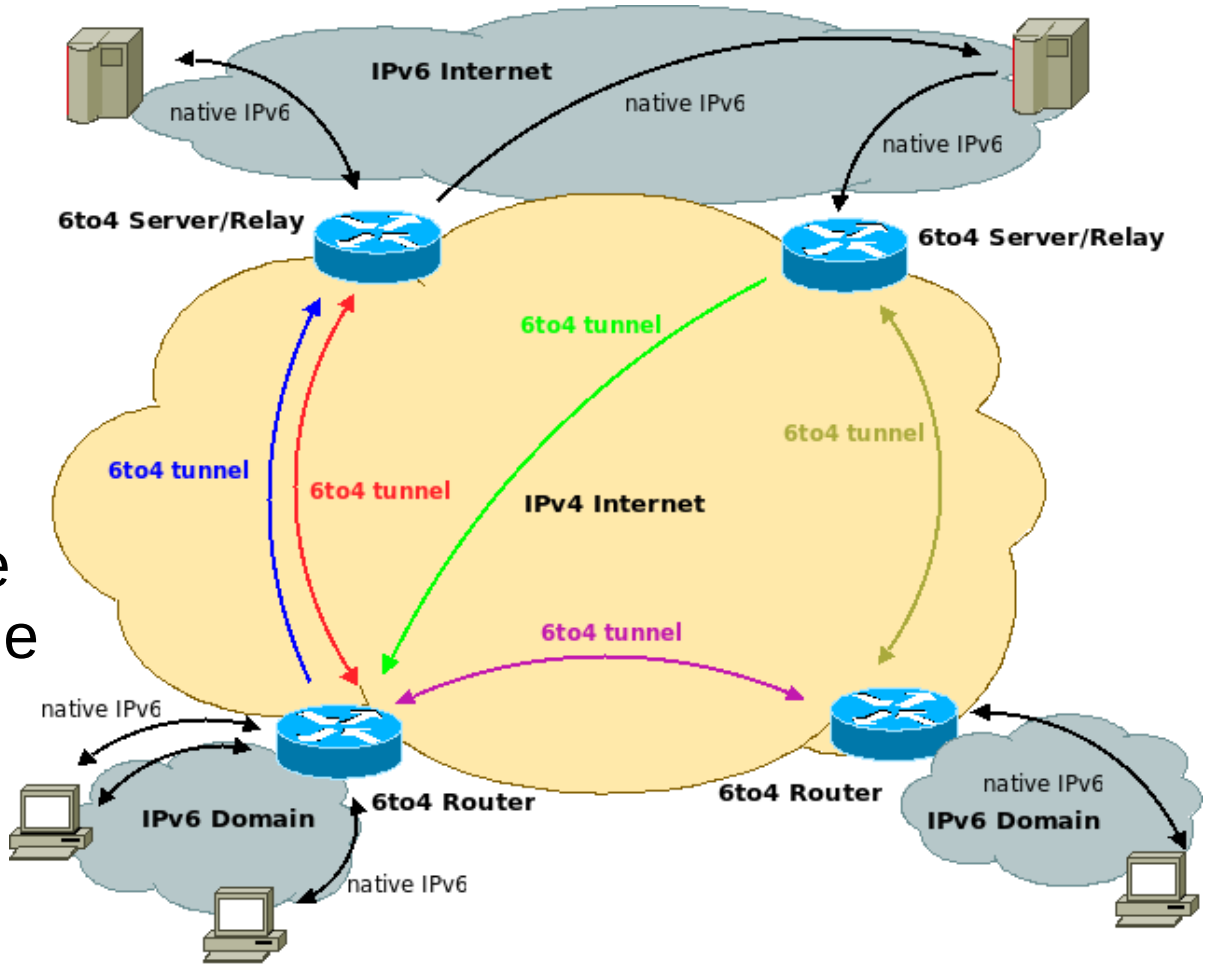
Automatic 6to4 Tunnels

- IPv4 tunnel end-point address is embedded within the destination IPv6 address
  - Automatic 6to4 tunnel allows isolated IPv6 domains to connect over an IPv4 network
  - Unlike the manually configured tunnels are not point-to-point, they are multipoint tunnels
  - 6to4 host/router needs to have a globally addressable IPv4 address
  - Cannot be located behind a NAT box
  - Unless the NAT box supports protocol 41 packets forwarding
  - Address format is:



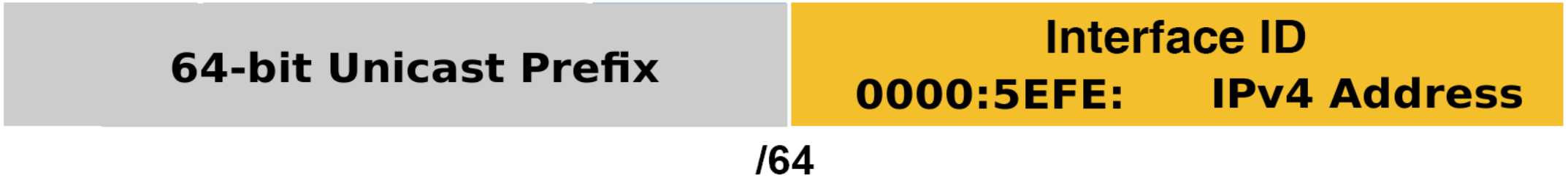
6to4 Relay Routers

- 6to4 router
  - Connects 6to4 hosts from a IPv6 domain and
    - Other 6to4 routers
    - The IPv6 Internet through a 6to4 relay router
- 6to4 relay router
  - Connects 6to4 routers on the IPv4 Internet and hosts on the IPv6 Internet.



ISATAP Tunnels

- Intra-site Automatic Tunnel Address Protocol
- Point-to-multipoint tunnels that can be used to connect systems within a site
- Used to tunnel IPv4 within as administrative domain to create a virtual IPv6 network over a IPv4 network
- Scalable approach for incremental deployment
- Encode IPv4 Address in IPv6 Address within the interface ID



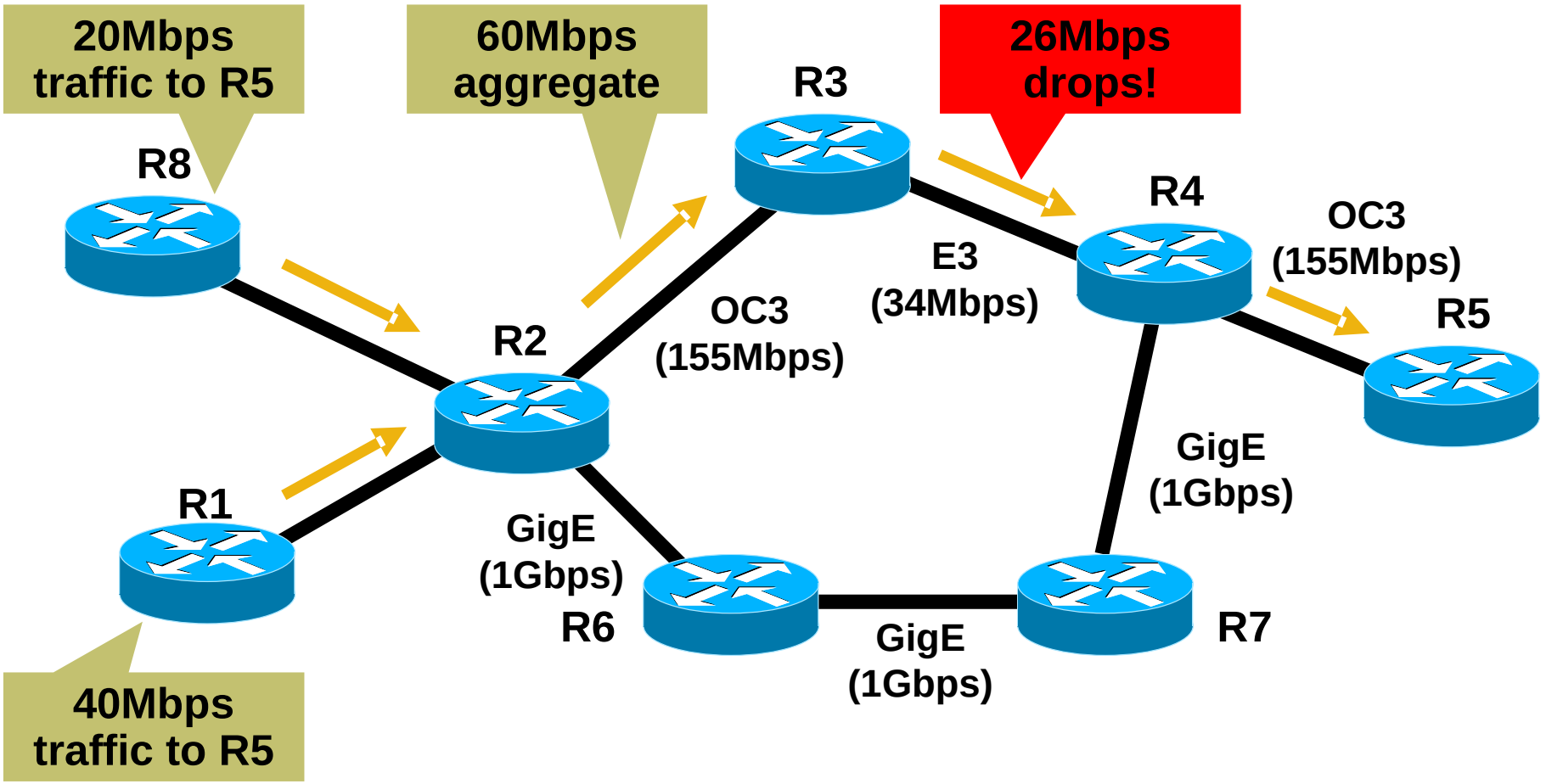
Traffic Engineering (TE)
&
Multiprotocol Label Switching (MPLS)

universidade de aveiro
deti.ua.pt

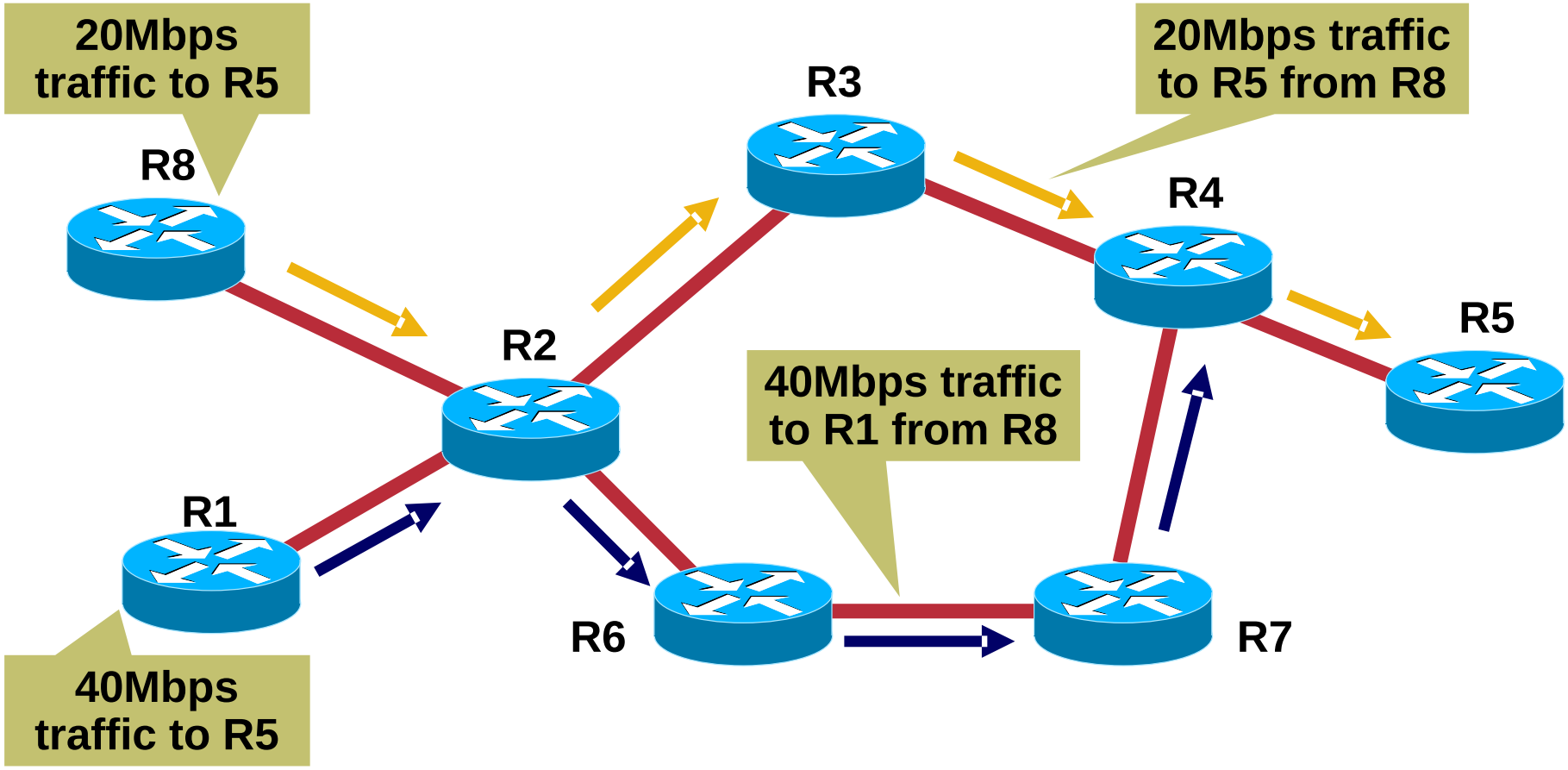
Traffic Engineering (TE)

- Network Engineering
  - Build your network to carry your predicted traffic!
  - Traffic patterns are impossible to predict!
  - Routing is based on the destination and does not allow to take the maximum possible advantage of the network resources.
  - IP source routing (using options field of IP header) is not usable in practice due to security reasons.
- Traffic Engineering
  - Manipulate your traffic path to fit your network!
    - Can be done with routing protocol costs (difficult deployment), or MPLS.
    - With RIP or OSPF or ANY OTHER IGP it is not possible to condition multiple traffic flows.
  - Increase efficiency of bandwidth resources.
    - Prevent over-utilized (congested) links whilst other links are under-utilized.
  - Ensure the most desirable/appropriate path for some/all traffic.
    - Override the shortest path selected by the routing protocols.

Shortest Path and Congestion



A TE Solution



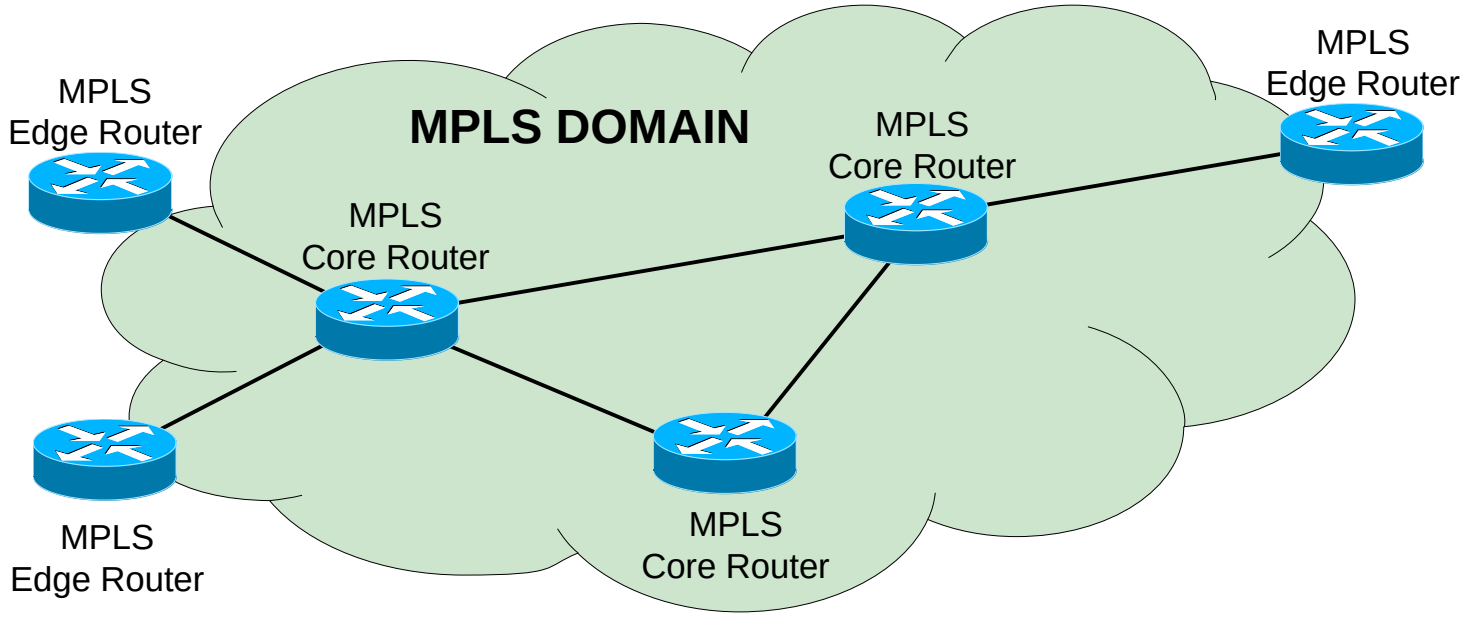
Tunnels are UNI-DIRECTIONAL

Normal path: R8 > R2 > R3 > R4 > R5

Tunnel path: R1 > R2 > R6 > R7 > R4

Multiprotocol Label Switching (MPLS)

- Packets are labeled at the source with the label of the first hop.
- As a packet travels from one router to the next, each router makes an independent forwarding decision for that packet based on a label.
- Advantages
  - Simplification of the packet routing process on routers.
  - Traffic engineering capability.
  - Simplification of the network management (a single protocol layer).

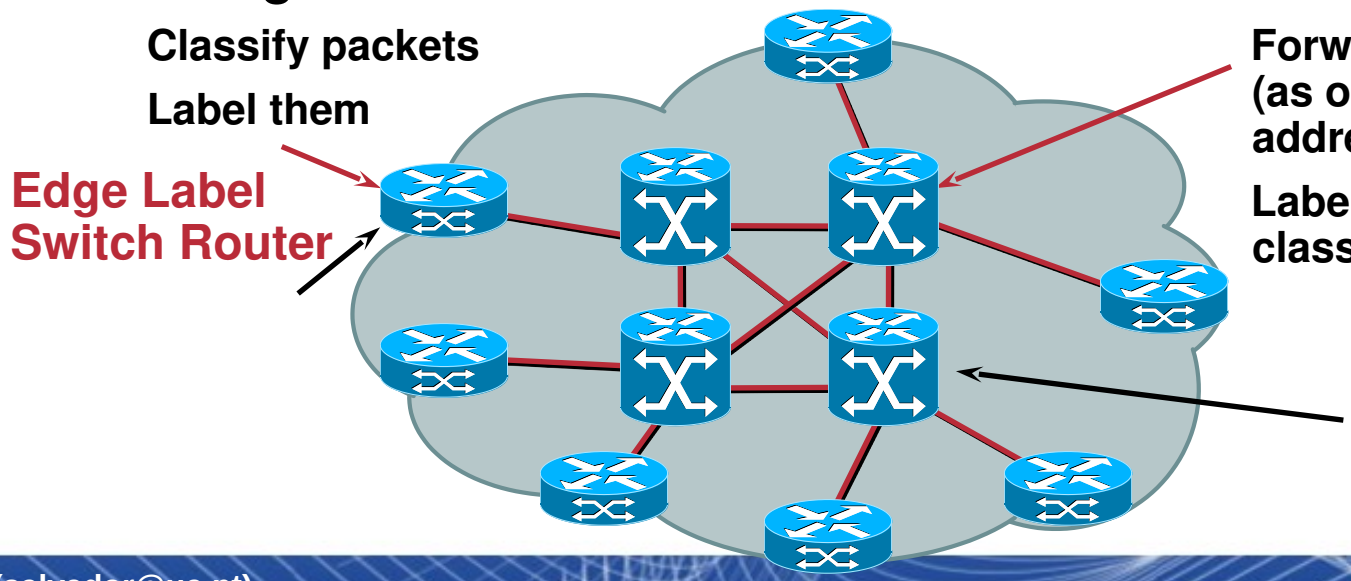


MPLS Fundamentals

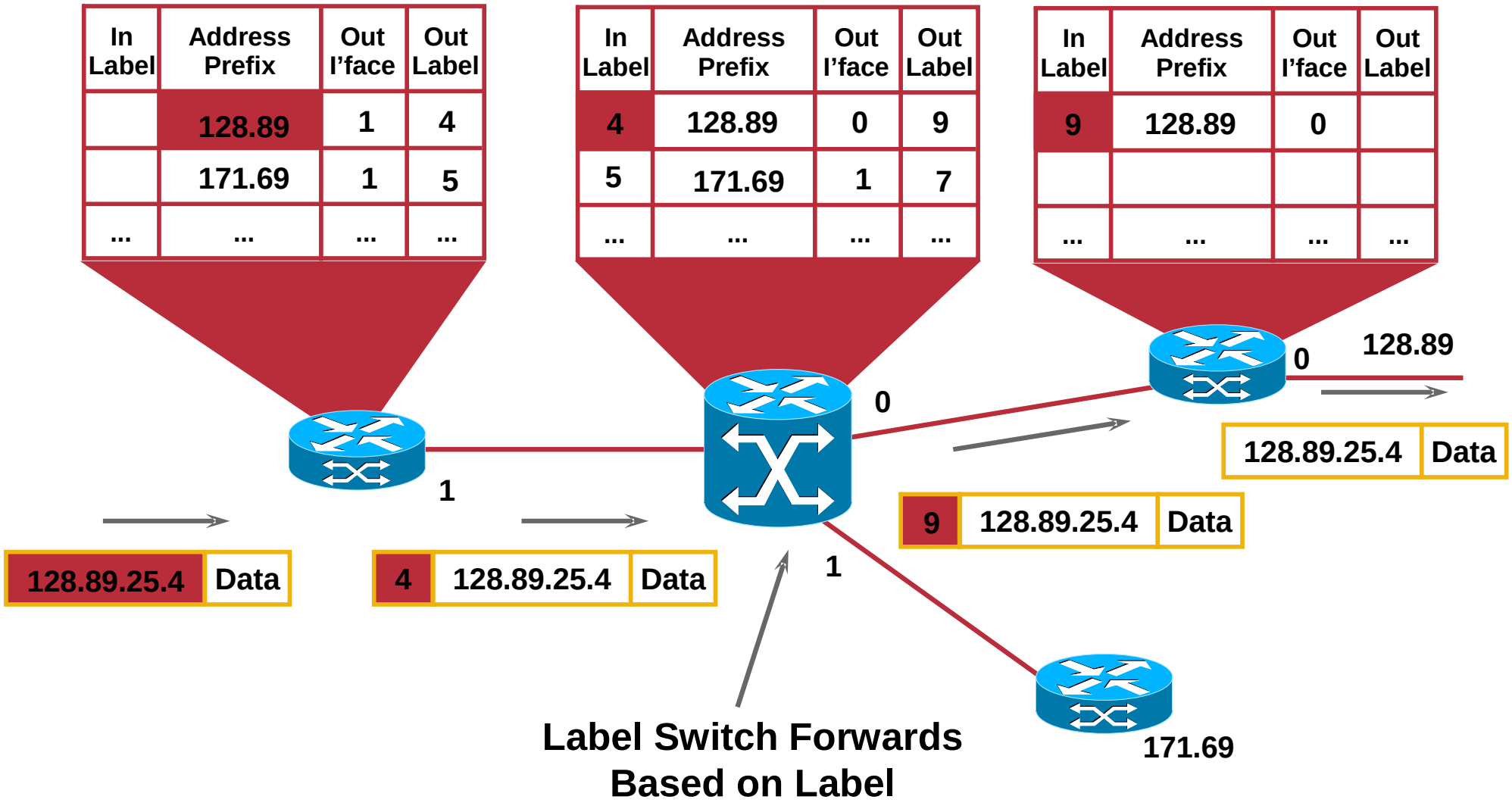
- Based on the label-swapping and forwarding paradigm.
- As a packet enters an MPLS network, it is assigned a label based on its **Forwarding Equivalence Class (FEC)** as determined at the edge of the MPLS network.
- FECs are groups of packets forwarded over the same **Label Switched Path (LSP)** by **Label Switching Routers (LSR).**
- Need a mechanism that will create and distribute labels to establish LSP paths.
- Separated into two planes:
  - Control Plane - Responsible for maintaining correct label tables among Label Switching Routers.
  - Forwarding Plane - Uses label carried by packet and label table maintained by LSR to forward the packet.

- **At Edge:**
  - **Classify packets**
  - **Label them**

- **In Core:**
  - **Forward using labels (as opposed to IP address)**
  - **Label indicates service class and destination**

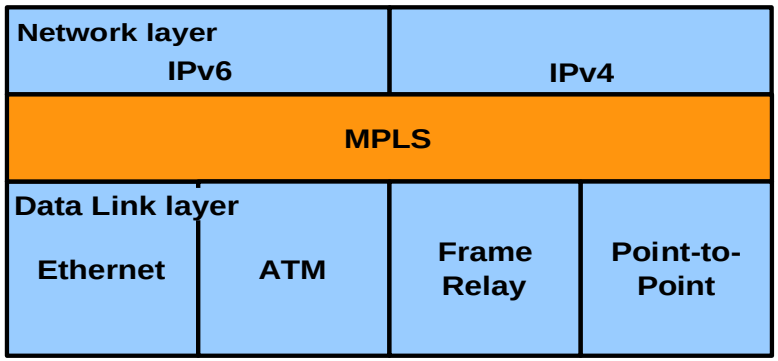


MPLS Switching

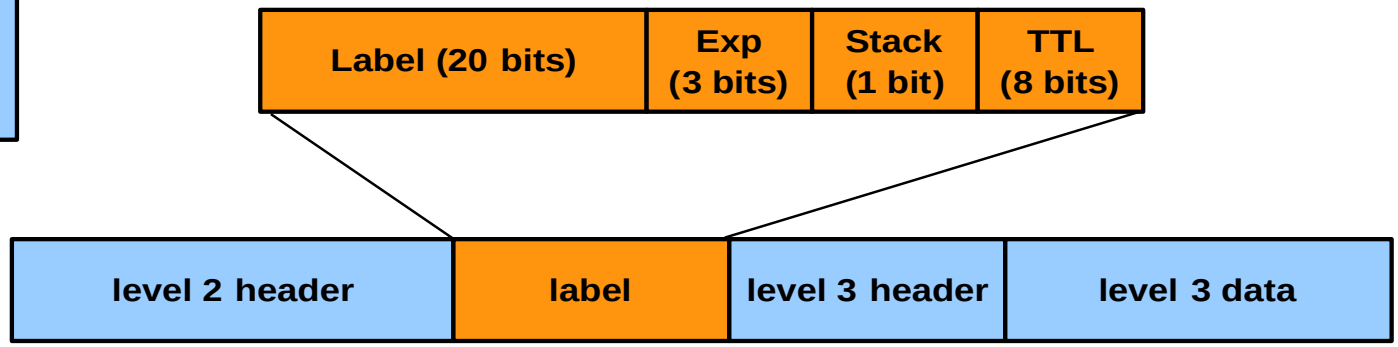


MPLS Labels

- On some Data Link (level 2) technologies, label is given by the appropriate fields of their header.
  - ATM technology : VPI (Virtual Path ID) and VCI (Virtual Channel ID) fields.
  - Frame Relay technology: DLCI (Data Link Connection Identifier) field.
- On other Data Link technologies (Point-to-Point, Ethernet), the label is inserted between layer 2 and layer 3 headers.
- Label is a 20-bit field that carries the actual value of the Label.
- TTL field is IP independent – Similar purpose.

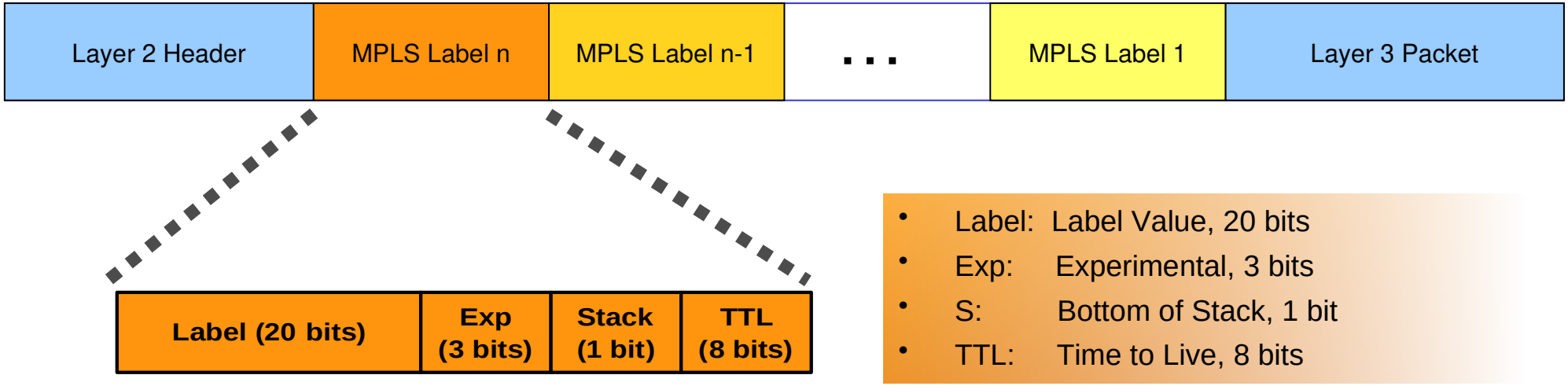


- Label: Label Value, 20 bits
- Exp: Experimental, 3 bits
- S: Bottom of Stack, 1 bit
- TTL: Time to Live, 8 bits



MPLS Label Stacking

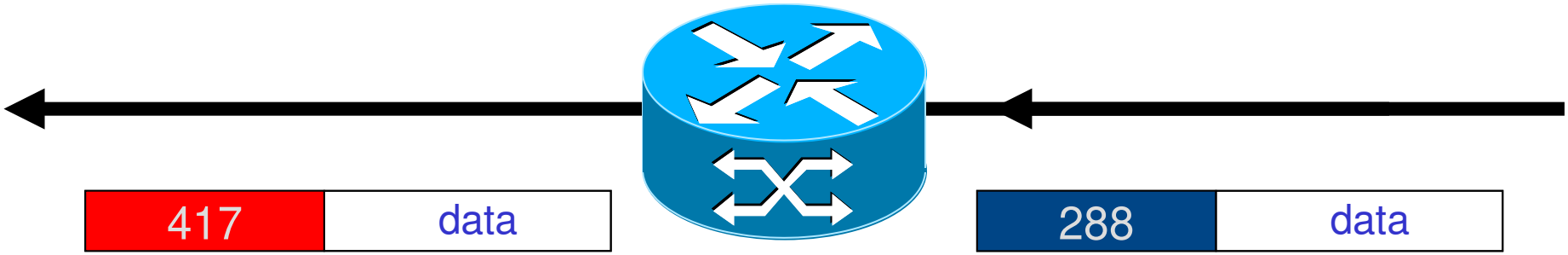
RFC 3032: MPLS Label Stack Encoding



- Label: Label Value, 20 bits
- Exp: Experimental, 3 bits
- S: Bottom of Stack, 1 bit
- TTL: Time to Live, 8 bits

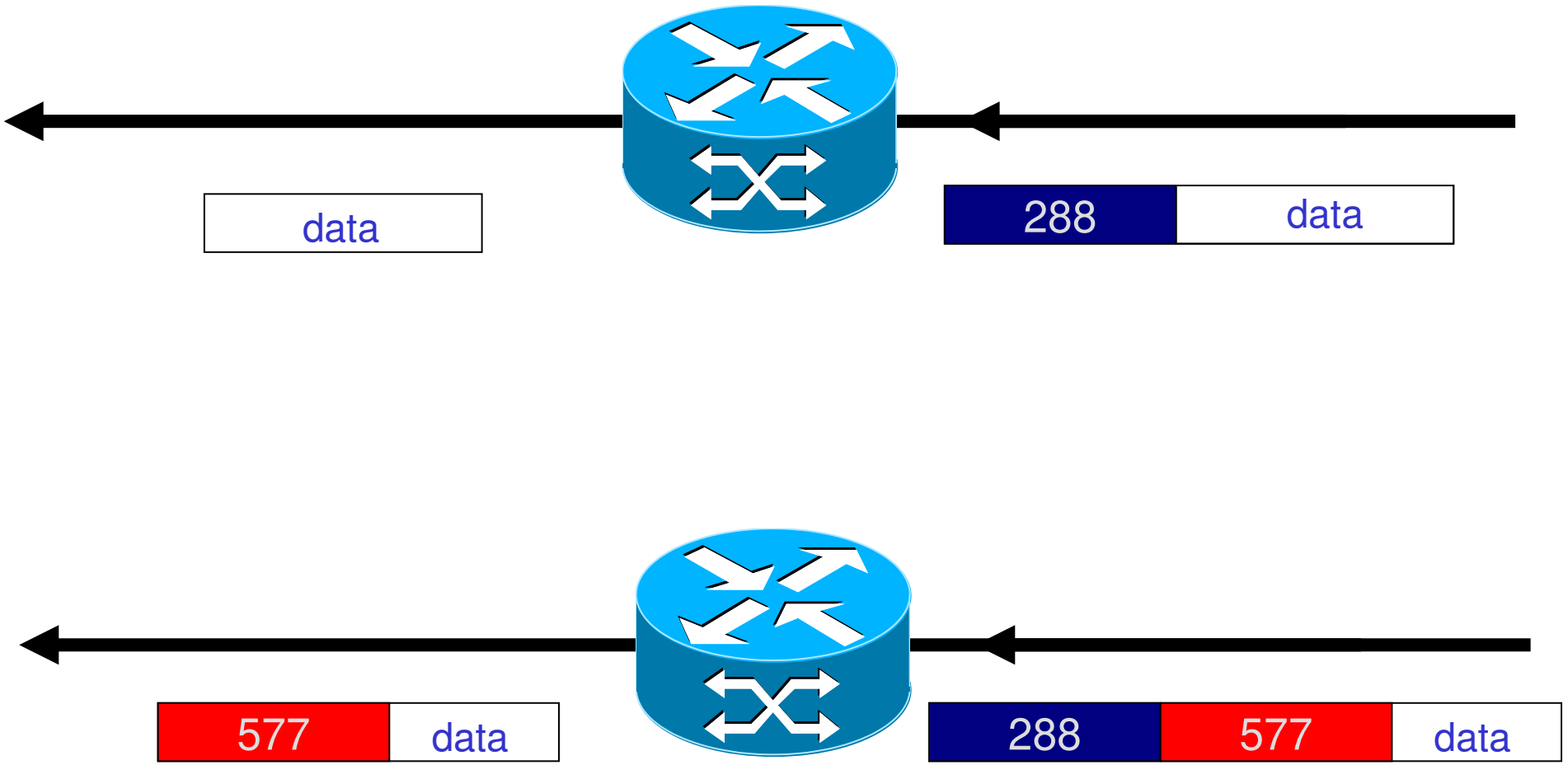
- Labels are arranged in a stack to support multiple services:
  - Inner labels are used to designate services, FECs, etc.
  - Outer label is used to switch the packets in MPLS core.
- Bottom of Stack (S) bit is set to one for the last entry in the label stack (i.e., for the bottom of the stack), and zero for all other labels.

Forwarding via Label Swapping

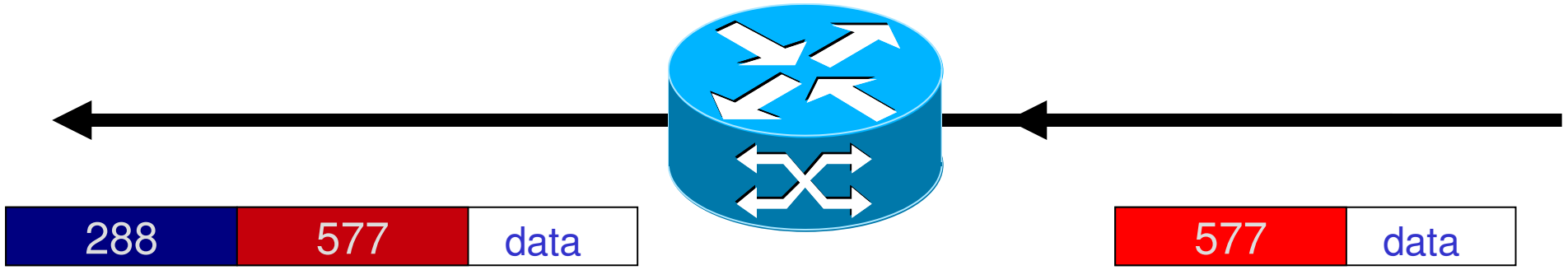


Labels are short, fixed-length values.

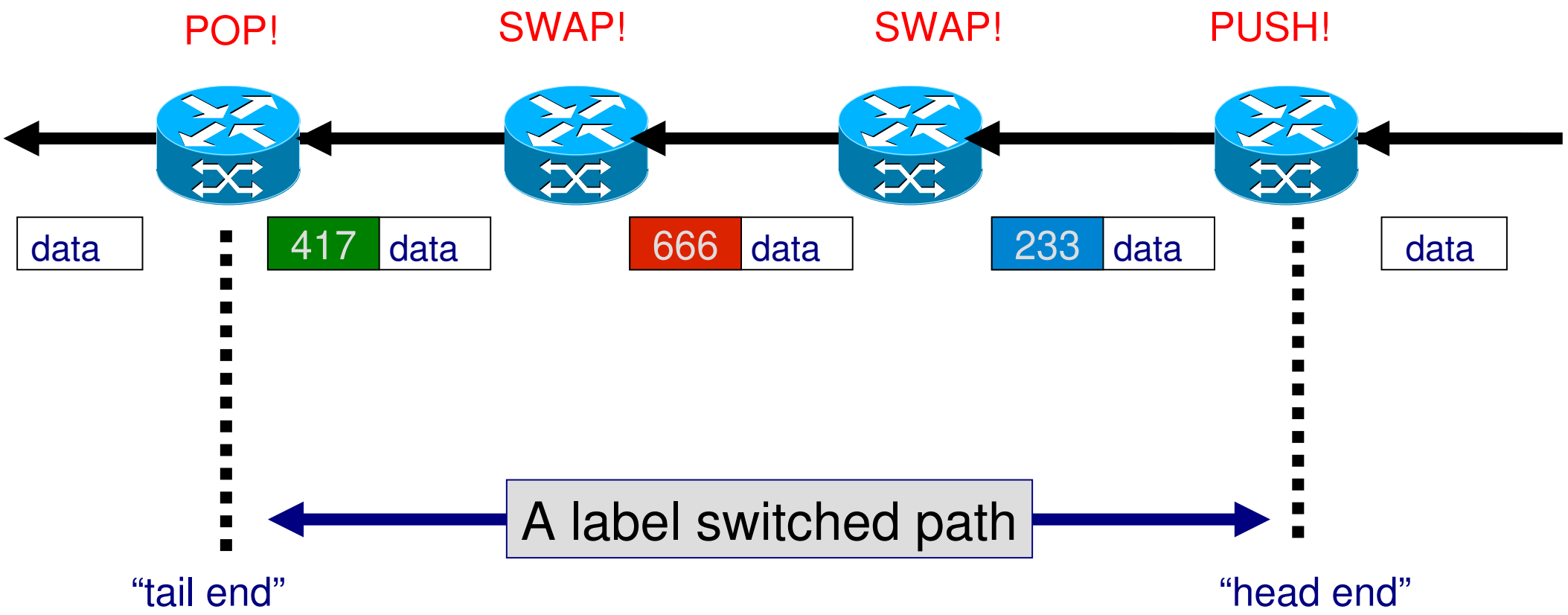
Popping Labels



Pushing Labels

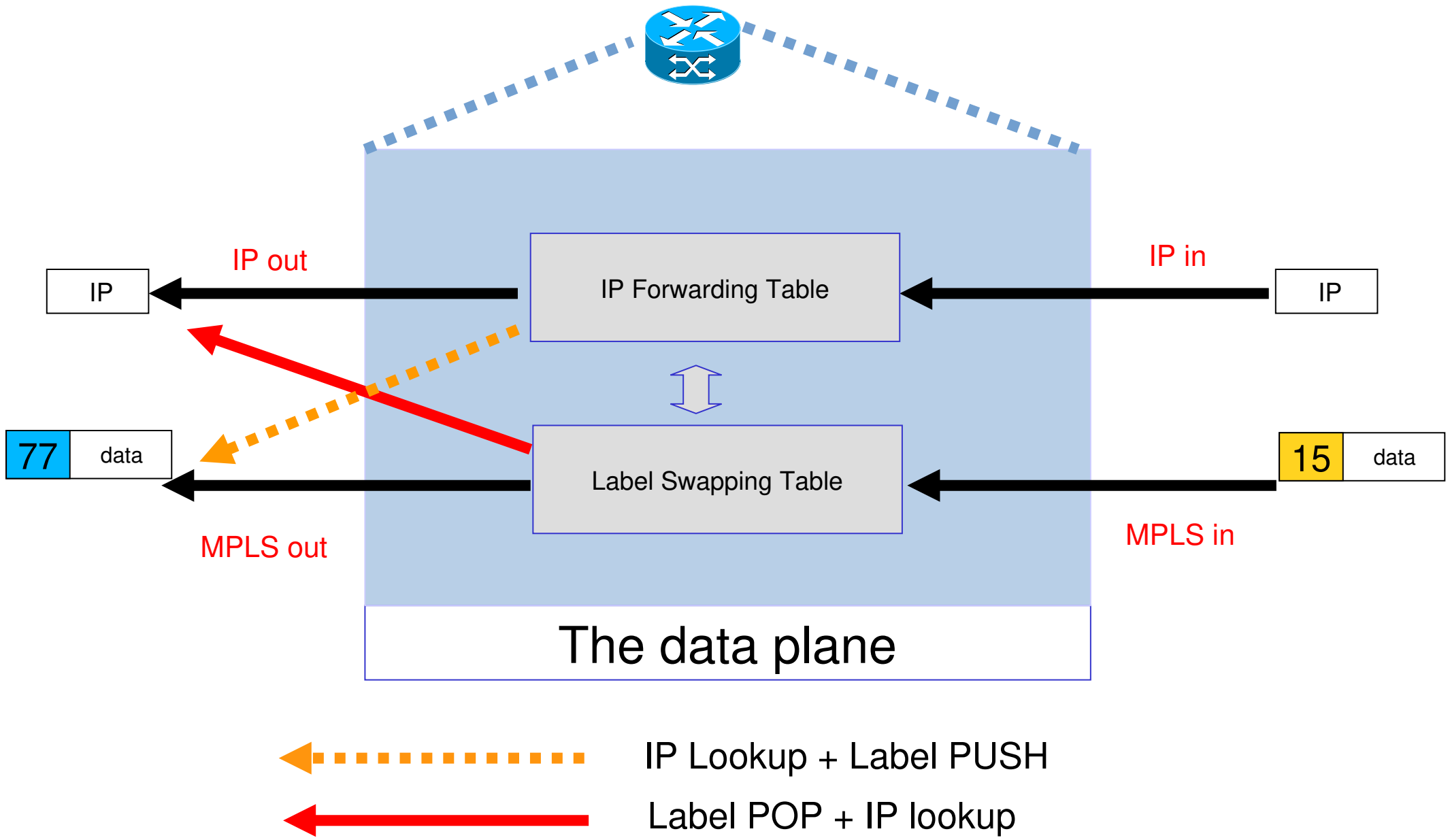


A Label Switched Path (LSP)

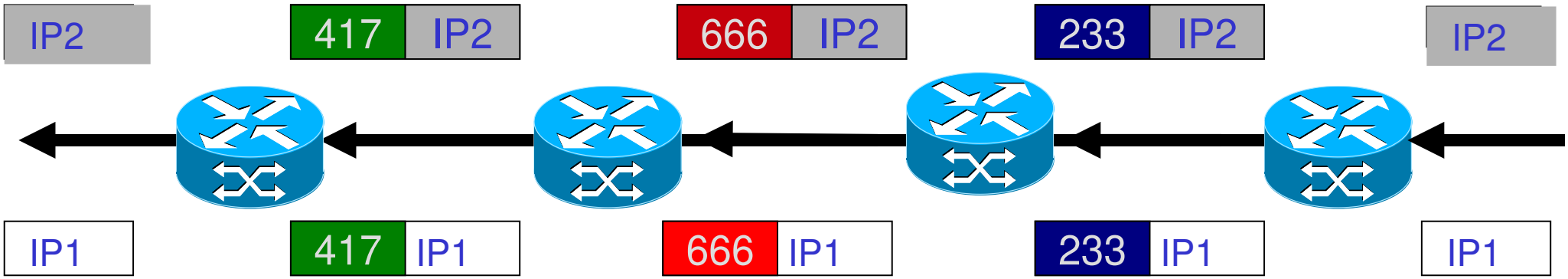


Often called an MPLS tunnel: payload headers are not Inspected inside of an LSP. Payload could be MPLS …

Label Switched Router



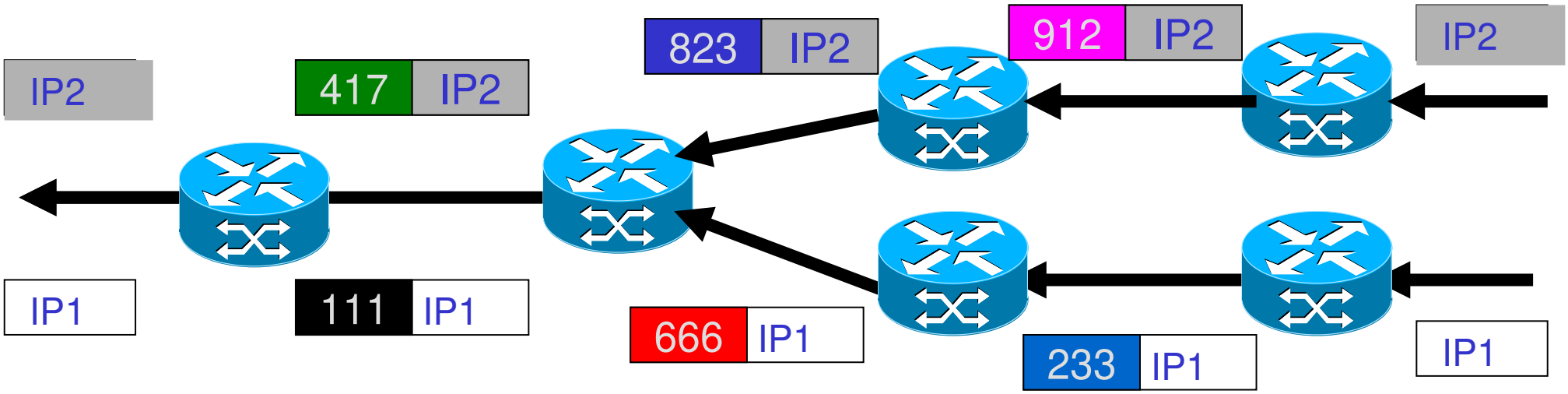
Forwarding Equivalence Class (FEC)

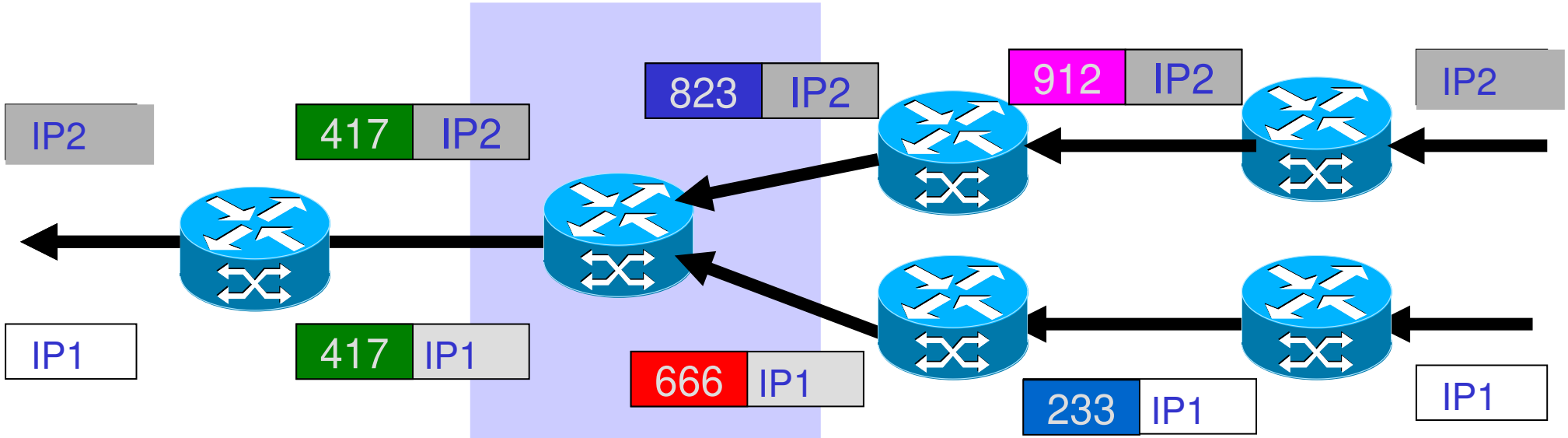


Packets IP1 and IP2 are forwarded in the same way --- they are in the same FEC.

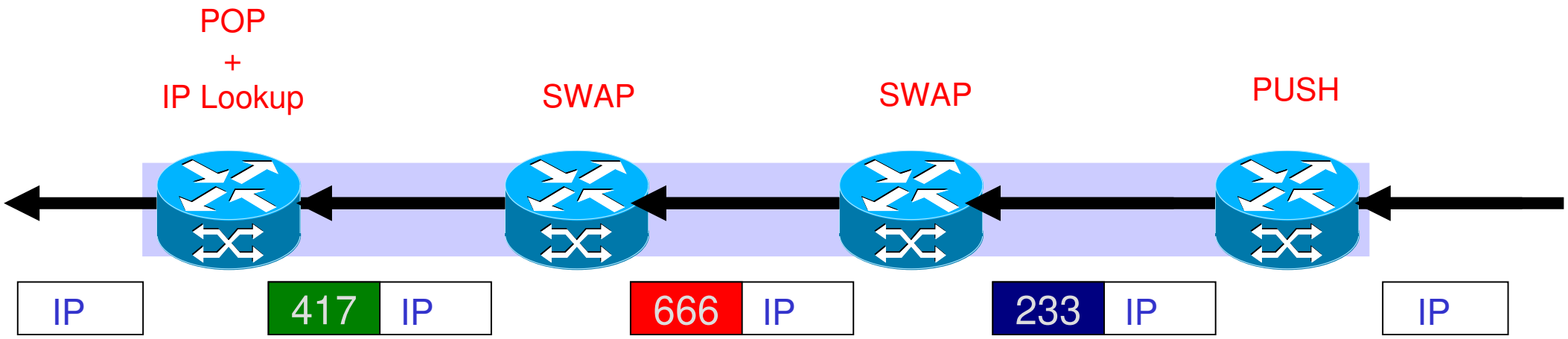
Network layer headers are not inspected inside an MPLS LSP. This means that inside of the tunnel the LSRs do not need full IP forwarding table.

LSP Merge

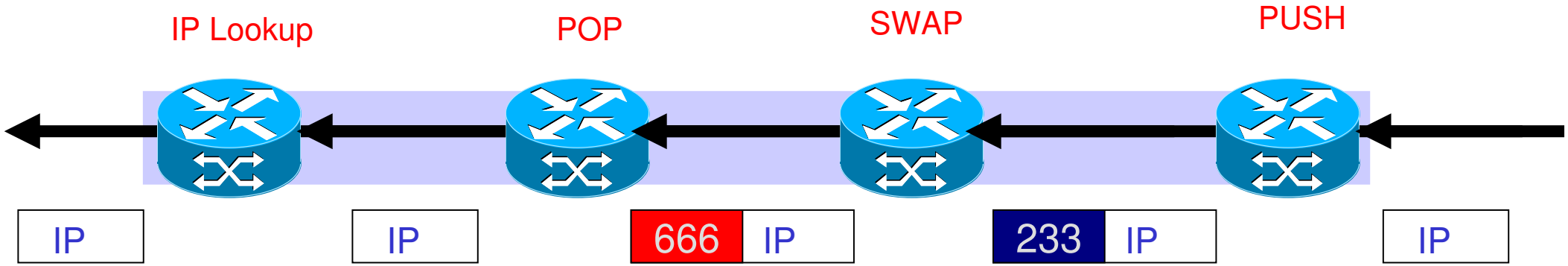

IP2
417 IP2
823 IP2
912 IP2
IP2
IP1
111 IP1
666 IP1
233 IP1
IP1


IP2
417 IP2
823 IP2
912 IP2
IP2
IP1
417 IP1
666 IP1
233 IP1
IP1
LSP merge

Penultimate Hop Popping (PHP)

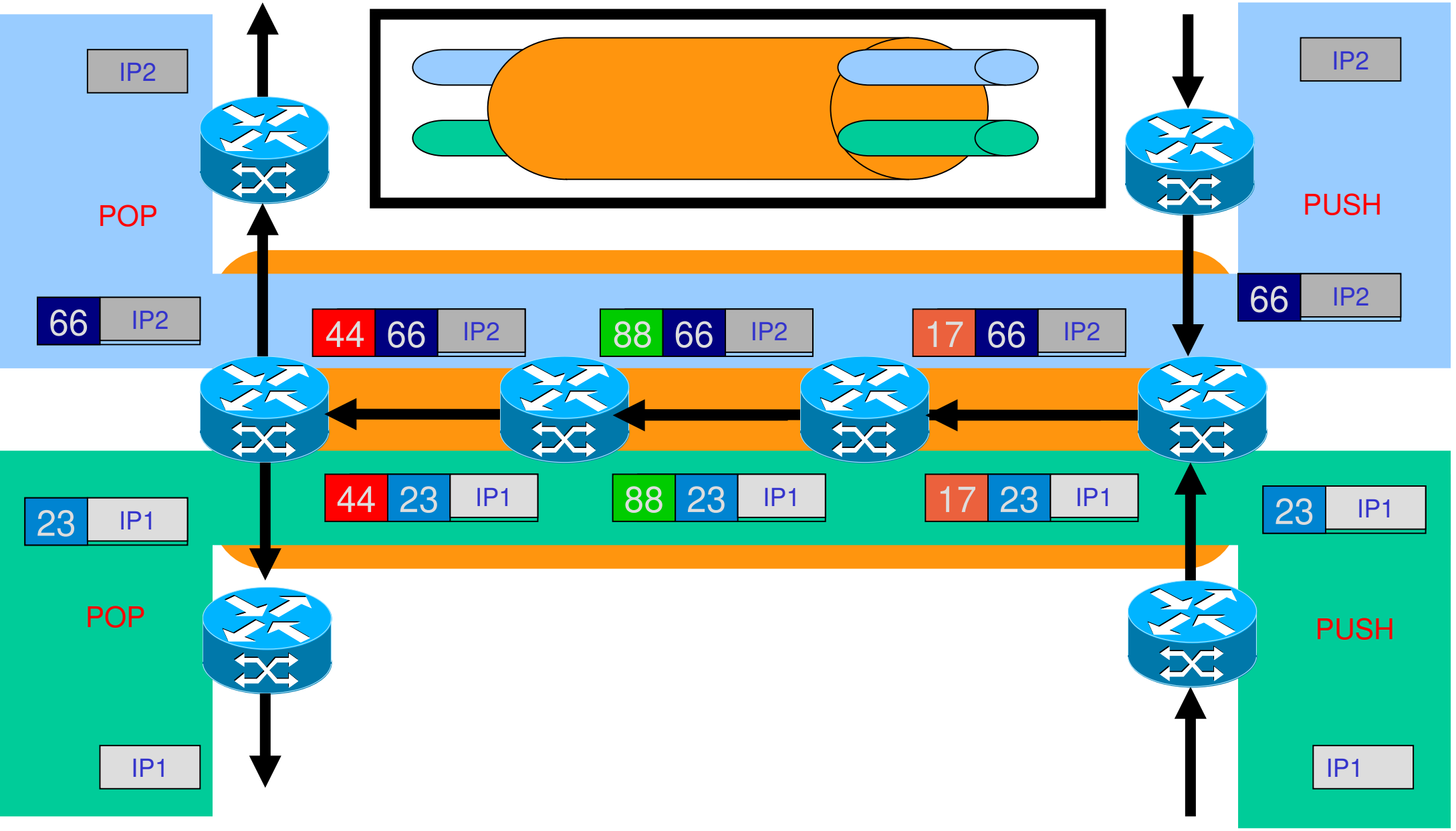


Without PHP



With PHP - Reduces Label Edge Router load

LSP Hierarchy via Label Stacking



Label Distribution Protocols

- Unconstrained routing
  - Label Distribution Protocol (LDP).
  - Path is chosen based on IGP shortest path.
- Constrained routing
  - Constrained by explicit path definition and/or performance requirements (e.g., available bandwidth).
  - Resource Reservation Protocol with Traffic Engineering (RSVP-TE).
    - Evolution of RSVP to support traffic engineering and label distribution.
  - Constrained based Routing LDP (CR-LDP).
    - Evolution of LDP to support constrained routing.
    - Deprecated!
- MPLS VPN scope
  - MP-BGP using address family VPN IPv4 and family specific MP_REACH_NLRI attribute.

Label Distribution Protocol (LDP)

RFC 5036: LDP Specification. (10/2007)

- Dynamic distribution of label binding information.
- LSR discovery.
- Reliable transport with TCP.
- Incremental maintenance of label swapping tables (only deltas are exchanged).
- Designed to be extensible with Type-Length-Value (TLV) coding of messages.
- Modes of behavior that are negotiated during session initialization
  - Label distribution control (ordered or independent).
  - Label retention (liberal or conservative).
  - Label advertisement (unsolicited or on-demand).

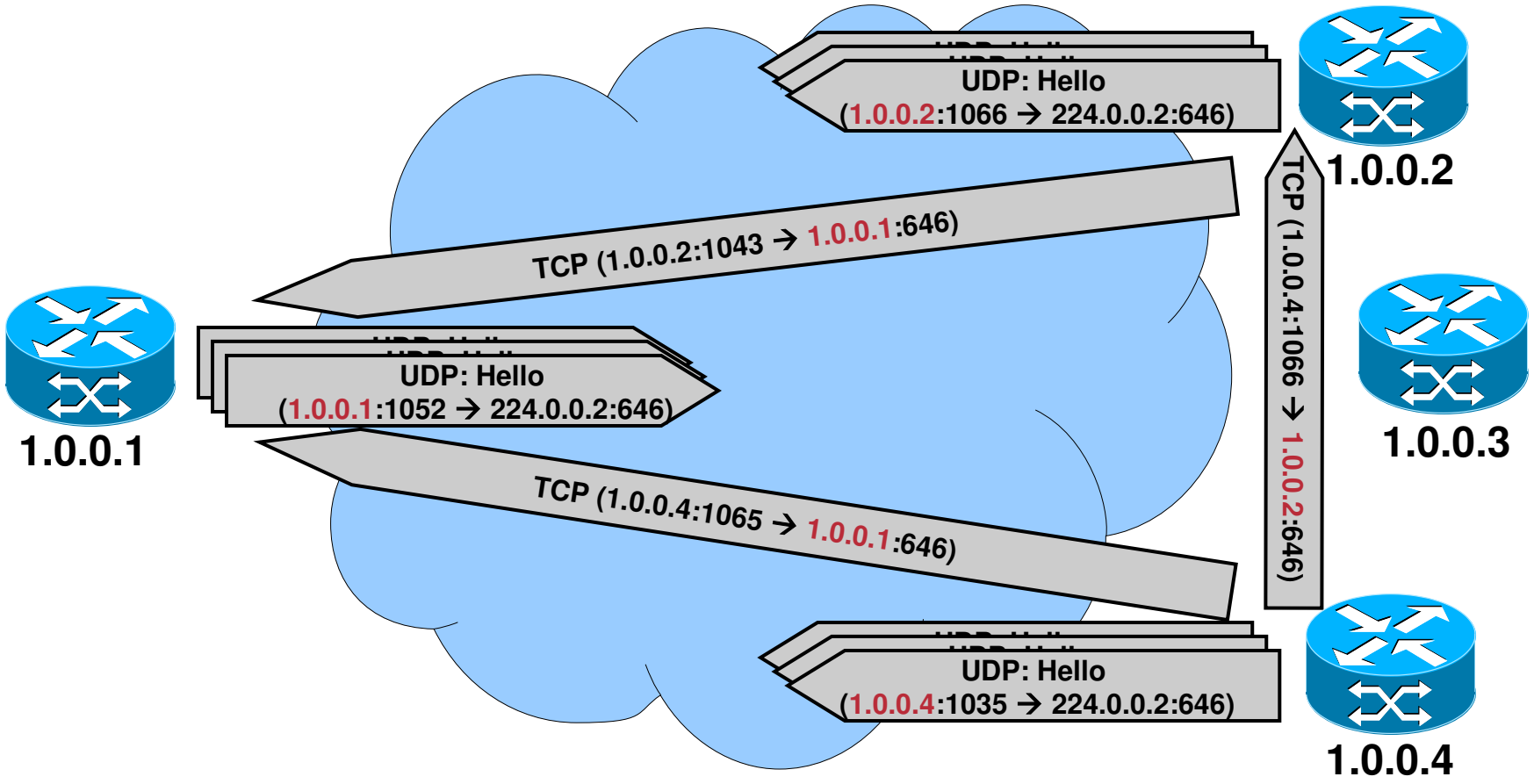
LDP Messages

- Discovery messages
  - Announce and maintain the presence of an LSR in a network.
  - **Hello Messages** (UDP) sent to “all-routers” multicast address.
  - Once neighbor is discovered, a LDP session is established over TCP.
- Session messages
  - Establish (**Initialization Message**) and maintain (**KeepAlive Message**) sessions between LDP peers.
- Advertisement messages
  - When a new LDP session is initialized and before sending label information an LSR advertises its interface addresses with one or more **Address Messages**.
  - An LSR withdraw previously advertised interface addresses with **Address Withdraw Messages**.
  - Create, change, and delete label mappings for FECs.
    - **Label Mapping, Label Request, Label Abort Request, Label Withdraw,** and **Label Release Messages.**
- Notification messages
  - Provide advisory information and to signal error information.

LDP Session Establishment

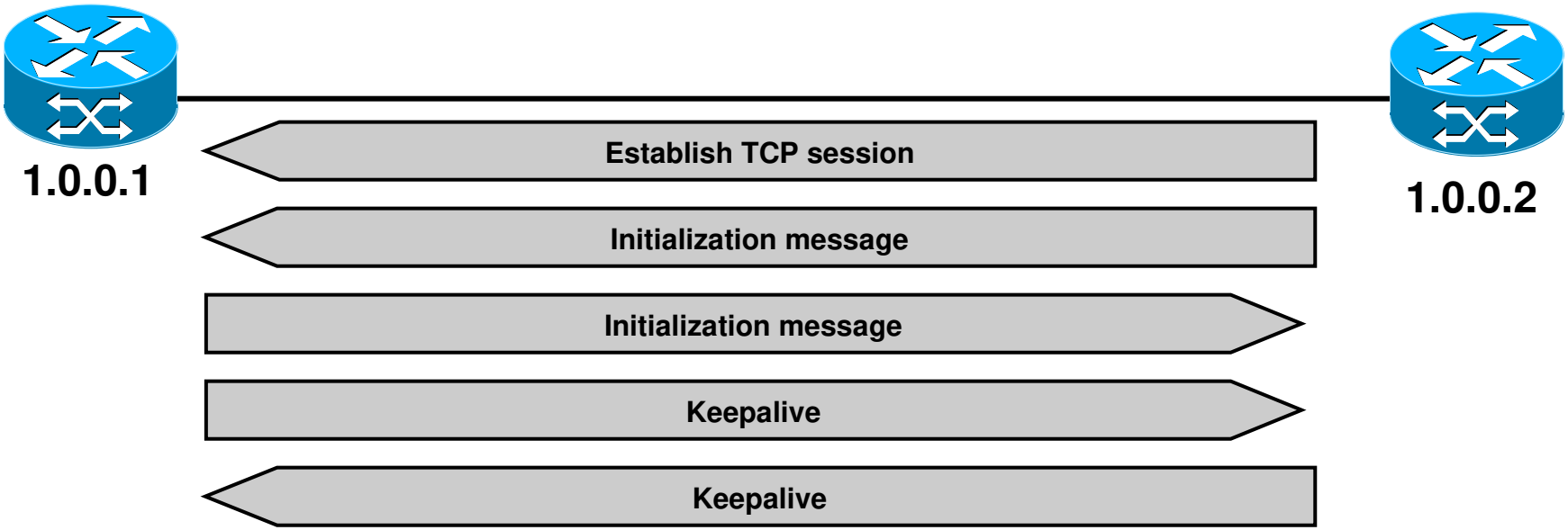
- Hello messages (UDP) are periodically sent on all interfaces enabled for MPLS to a “all-routers” multicast address (224.0.0.2).
- If there is another router on that interface it will respond by trying to establish a LDP/TCP session with the source of the hello messages.
- Both TCP and UDP messages use well-known LDP port number 646.

LDP Neighbor Discovery



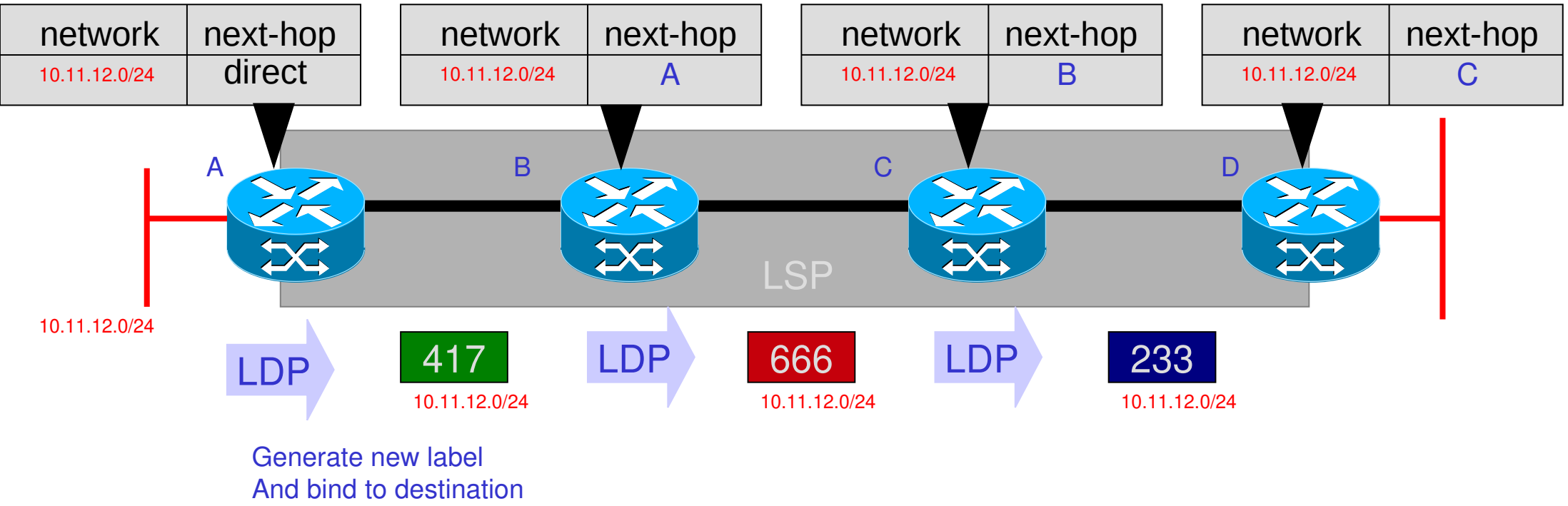
- LDP Session is started by the router with higher IP address.

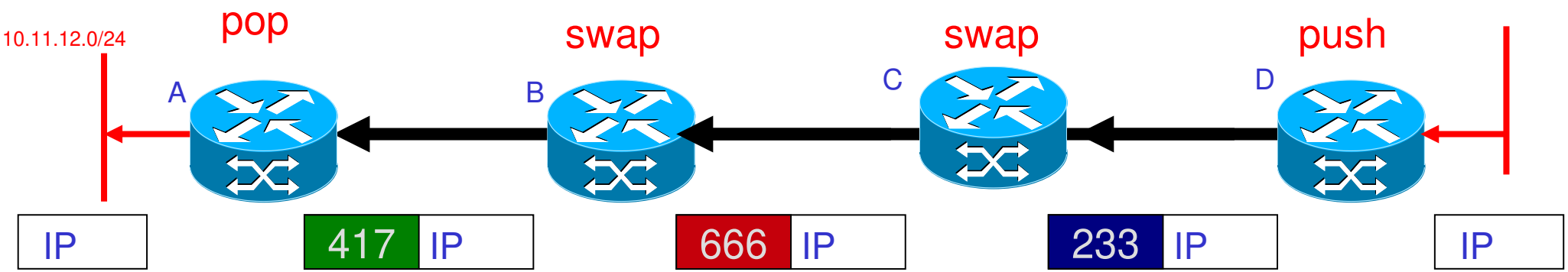
LDP Session Negotiation



- Peers first exchange initialization messages.
- The session is ready to exchange label mappings after receiving the first keepalive.
  - Keepalives are resent periodically to maintain the LDP/TCP session active.

LDP and Hop-by-Hop routing





Constraint Based Routing

Basic components

1. Specify path constraints
2. Extend topology database to include resource and constraint information
3. Find paths that do not violate constraints and optimize some metric
4. Signal to reserve resources along path
5. Set up LSP along path (with explicit route)
6. Map ingress traffic to the appropriate LSPs

Problem here: OSPF areas hide information for scalability. So these extensions work best only within an area…

Extend Link State Protocols (IS-IS, OSPF)

Extend RSVP or LDP or both!

Problem here: what is the "correct" resource model for IP services?

Note: (3) could be offline,
or online (perhaps an extension to OSPF)

Resource Reservation + Label Distribution

Two competing approaches:

Add label distribution and explicit routes to a resource reservation protocol

RSVP-TE

\+

RSVP

RSVP-TE:
RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels

CR-LDP

\+

LDP

Add explicit routes and resource reservation to a label distribution protocol

CR-LDP
RFC 3212: Constraint-Based LSP Setup using LDP

As of February 2003, the IETF MPLS working group deprecated CR-LDP and decided to focus purely on RSVP-TE.

RFC 3468: The Multiprotocol Label Switching (MPLS) Working Group decision on MPLS signaling protocols

Resource Reservation Protocol with Traffic Engineering (RSVP-TE)

RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels. (12/2001)
RFC 5151: Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions. (2/2008)

- Evolution of RSVP.
- To map traffic flows onto the physical network topology through label switched paths, requires resource and constraint network information.
  - Provided by Extend Link State Protocols (IS-IS or OSPF with TE extensions).
    - RFC 3630: Traffic Engineering (TE) Extensions to OSPF Version 2. (9/2003)
    - RFC 5305: IS-IS Extensions for Traffic Engineering. (10/2008)

Extensions to RSVP for LSP Tunnels

- The SENDER_TEMPLATE (or FILTER_SPEC) object together with the SESSION object uniquely identifies an LSP tunnel (flow).
- LSP Tunnel related new objects
  - Explicit Route
    - Carried in PATH and contains a series of variable-length data items called sub-objects.
    - Possible sub-objects: IPv4 prefix, IPv6 prefix, and autonomous system number.
  - Label Request
    - Carried in PATH requesting a label for a specific tunnel/flow.
    - Request cab be without label range, with an ATM label range, or with an Frame Relay label range.
  - Label
    - Carried in RESV messages and contain a single label for a specific tunnel/flow.
  - Record Route
    - Carried in PATH and RESV, used to collect detailed path information and useful for loop detection and diagnostics.
  - Session Attribute
    - Carried in PATH, used to define the type and name of the session/tunnel/flow, also used to define priority values.
- LSP Tunnel related new object types
  - Session object new types
    - LSP_TUNNEL_IPv4 and LSP_TUNNEL_IPv6
  - Sender Template object new types
    - LSP_TUNNEL_IPv4 and LSP_TUNNEL_IPv6
  - Filter Specification object new types
    - LSP_TUNNEL_IPv4 and LSP_TUNNEL_IPv6

RSVP-TE PATH and RESV (example)

```
Resource ReserVation Protocol (RSVP): PATH Message. SESSION: IPv4-LSP
▷ RSVP Header. PATH Message.
▷ SESSION: IPv4-LSP, Destination 192.2.0.11, Tunnel ID 2, Ext ID c002000a.
▷ HOP: IPv4, 200.10.2.10
▶ TIME VALUES: 30000 ms
▷ EXPLICIT ROUTE: IPv4 200.10.2.2, IPv4 200.2.11.2, IPv4 200.2.11.11,
▷ LABEL REQUEST: Basic: L3PID: IP (0x0800)
▷ SESSION ATTRIBUTE: SetupPrio 7, HoldPrio 7, SE Style,  [RA_t2]
▷ SENDER TEMPLATE: IPv4-LSP, Tunnel Source: 192.2.0.10, LSP ID: 8.
▷ SENDER TSPEC: IntServ, Token Bucket, 18750 bytes/sec.
▷ ADSPEC
```

```
▽ Resource ReserVation Protocol (RSVP): RESV Message. SESSION: IPv4-LSP
  ▷ RSVP Header. RESV Message.
  ▷ SESSION: IPv4-LSP, Destination 192.2.0.11, Tunnel ID 2, Ext ID c002000a.
  ▷ HOP: IPv4, 200.10.2.2
  ▷ TIME VALUES: 30000 ms
  ▷ STYLE: Shared-Explicit (18)
  ▷ FLOWSPEC: Controlled Load: Token Bucket, 18750 bytes/sec.
  ▷ FILTERSPEC: IPv4-LSP, Tunnel Source: 192.2.0.10, LSP ID: 8.
  ▷ LABEL: 19
```

Traffic Engineering Extensions to OSPF

- RFC 3630: Traffic Engineering (TE) Extensions to OSPF Version 2. (9/2003)
- OSPF Traffic Engineering (TE) extensions are used to advertise TE Link State Advertisements (TE-LSAs) containing information about TE-enabled links.
  - Traffic Engineering LSA is a type 10 Opaque LSAs, which have an area flooding scope.
- TE-LSA contains one of two possible top-level Type Length Values (TLVs)
    - **Router Address**: specifies a stable IP address of the advertising router that is always reachable if there is any connectivity to it; this is typically implemented as a "loopback address";
    - **Link**: describes a single link with a a set of sub-TLVs (Link type, Link ID, Local interface IP address, Remote interface IP address, Traffic engineering metric, Maximum bandwidth, Maximum reservable bandwidth, Unreserved bandwidth, and Administrative group.
- The information made available by these extensions can be used to build an extended link state database
  - Can be used to:
    - Monitoring the extended link attributes;
    - Local constraint-based source routing;
    - Global traffic engineering.

OSPF-TE Opaque Area Database

- Router Address TLV

```
LS age: 250
  Options: (No TOS-capability, DC)
  LS Type: Opaque Area Link
  Link State ID: 1.0.0.0
  Opaque Type: 1
  Opaque ID: 0
  Advertising Router: 192.2.0.2
  LS Seq Number: 80000001
  Checksum: 0xDACD
  Length: 28
  Fragment number : 0

    MPLS TE router ID : 192.2.0.2
    Number of Links : 0
```

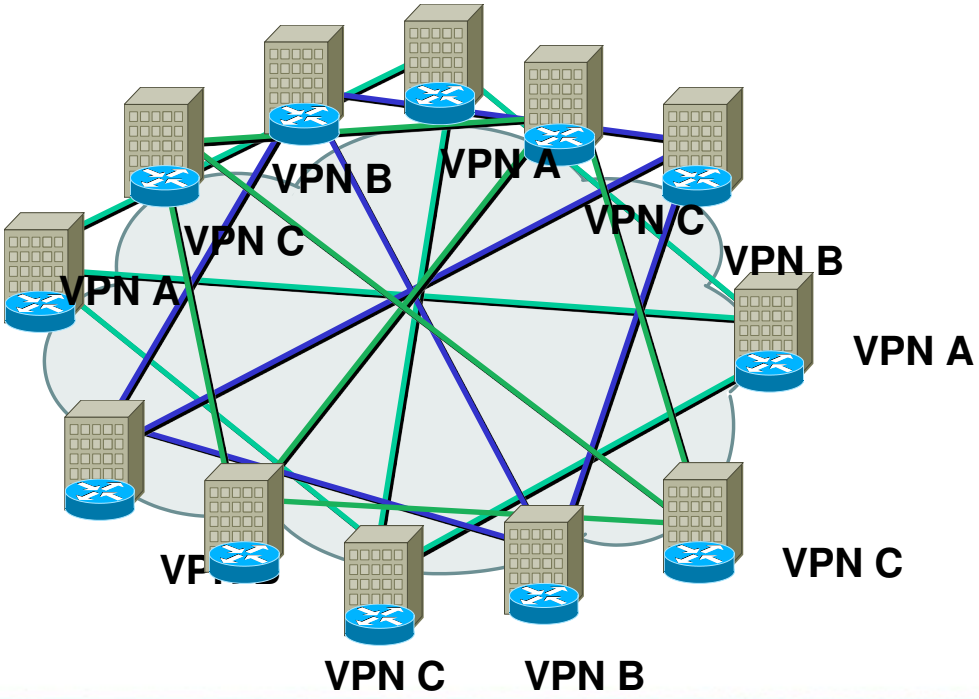
- Link TLV

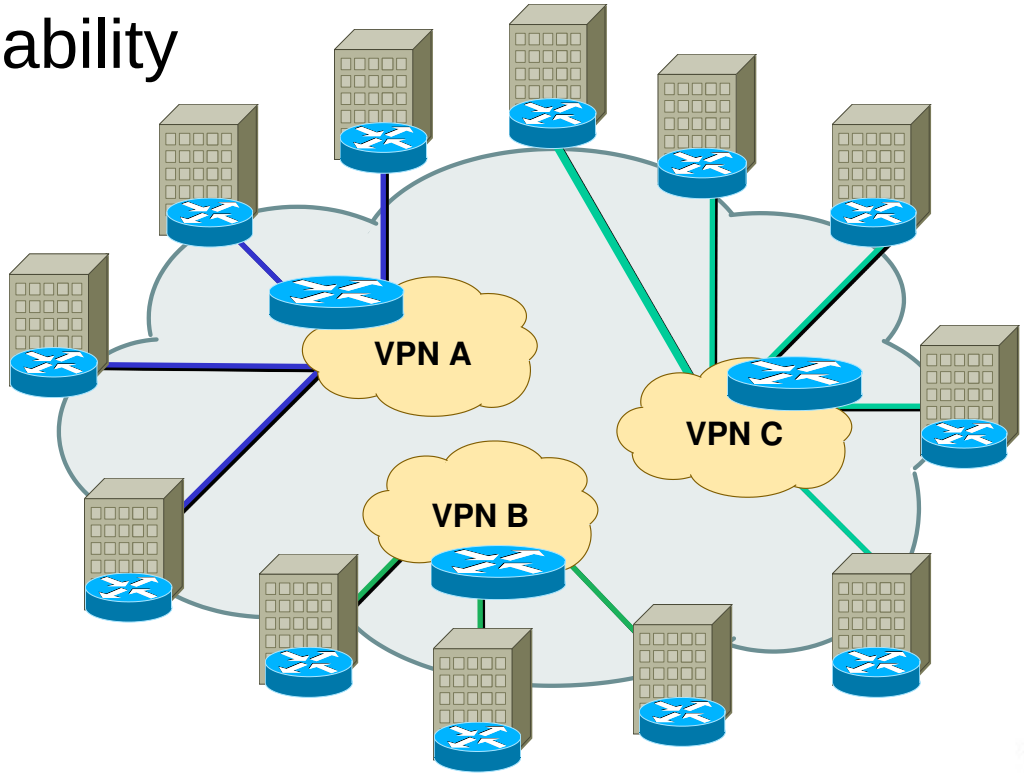
```
LS age: 246
 Options: (No TOS-capability, DC)
 LS Type: Opaque Area Link
 Link State ID: 1.0.0.2
 Opaque Type: 1
 Opaque ID: 2
 Advertising Router: 192.2.0.2
 LS Seq Number: 80000001
 Checksum: 0x2FBB
 Length: 124
 Fragment number : 2

   Link connected to Broadcast network
     Link ID : 200.1.2.2
     Interface Address : 200.1.2.2
     Admin Metric : 1
     Maximum bandwidth : 12500000
     Maximum reservable bandwidth : 64000
     Number of Priority : 8
     Priority 0 : 64000       Priority 1 : 64000
     Priority 2 : 64000       Priority 3 : 64000
     Priority 4 : 64000       Priority 5 : 64000
     Priority 6 : 64000       Priority 7 : 64000
     Affinity Bit : 0x0
     IGP Metric : 1
   Number of Links : 1
```

MPLS Layer 3 VPNs

MPLS L3 VPNs using BGP (RFC2547)

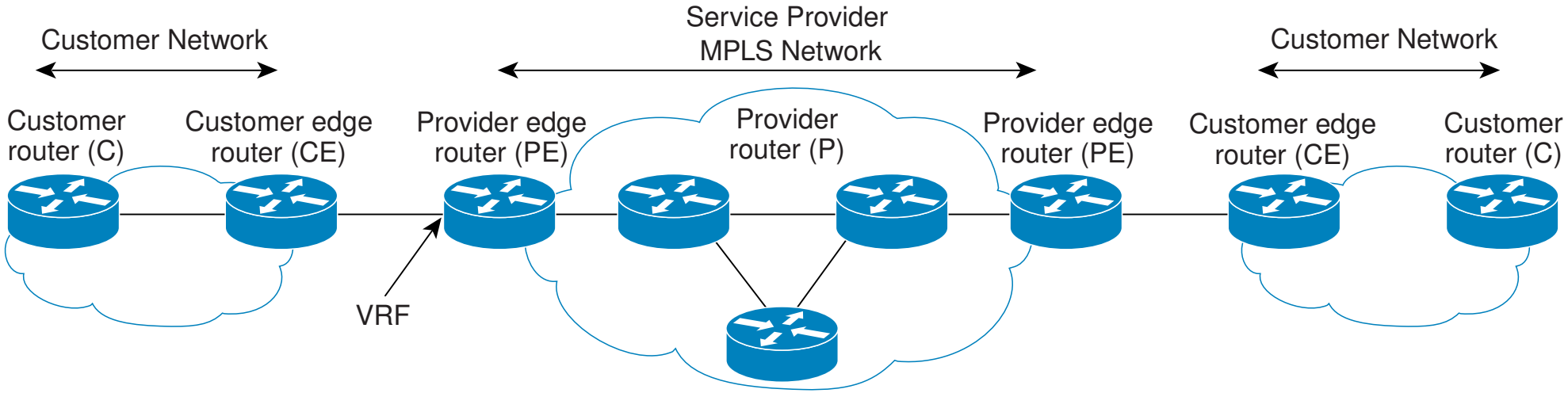
- End user perspective
  - Virtual Private IP service.
  - Simple routing – just point default to provider.
  - Full site-site connectivity without the usual drawbacks (routing complexity, scaling, configuration, cost).
- Major benefit for provider – scalability





universidade de aveiro

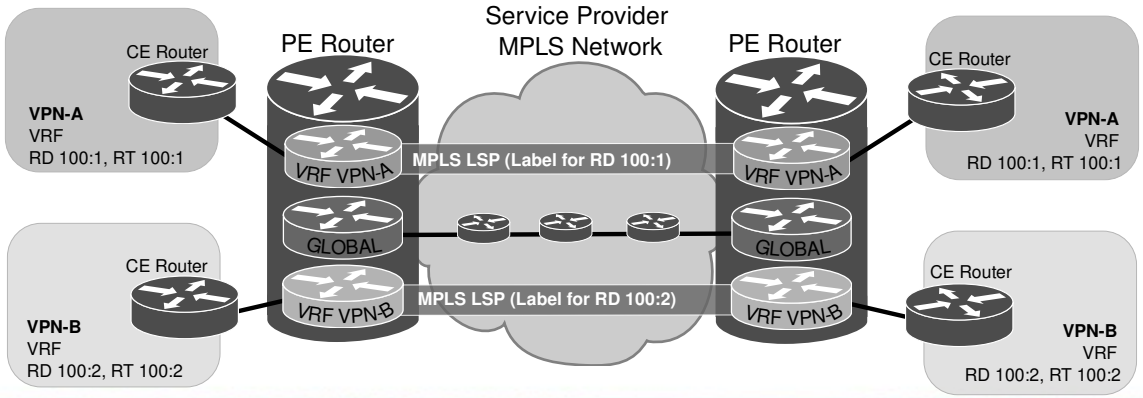
MPLS VPN Terminology



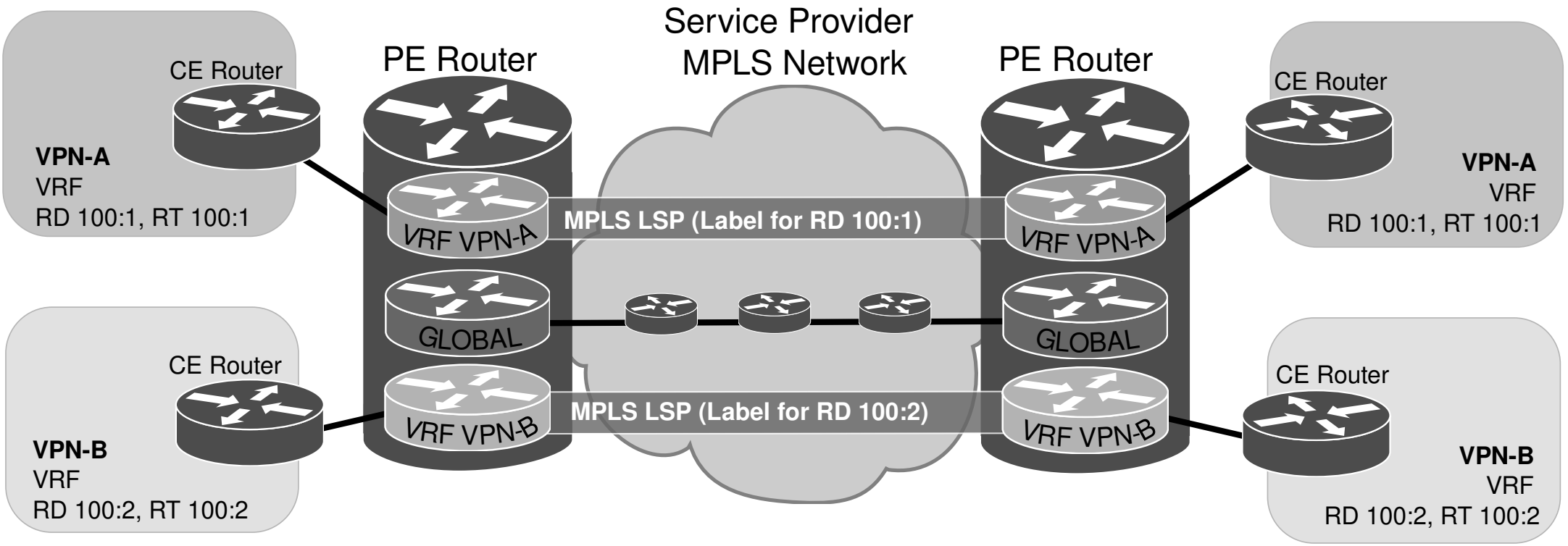
- Customer router (C) is connected only to other customer devices.
- Customer Edge (CE) router peers at Layer 3 to the Provider Edge (PE).
  - The PE-CE Interface runs either a dynamic routing protocol (eBGP, RIPv2, EIGRP, or OSPF) or has static routing (Static, Connected).
- Provider (P) router, resides in the core of the provider network.
  - Participates in the control plane for customer prefixes. The P router is also referred to as a Label Switch Router (LSR), in reference to its primary role in the core of the network, performing label switching/swapping of MPLS traffic.
- Provider Edge (PE) router, sits at the edge of the MPLS SP network.
  - In an MPLS VPN context, separate VRF routing tables are allocated for each user group.
  - Contains a global routing table for routes in the core SP infrastructure.
  - The PE is sometimes referred to as a Label Edge Router (LER) or Edge Label Switch Router (ELSR) in reference to its role at the edge of the MPLS cloud, performing label imposition and disposition.

Virtual Routing and Forwarding (VRF)

- Virtual Routing and Forwarding (VRF) instance, is separate from the global routing table that exists on PE routers.
- PE routers maintain separate routing tables:
  - Global routing table
    - Contains all PE and P routes (perhaps BGP).
    - Populated by the VPN backbone IGP .
  - VRF table
    - Routing and forwarding table associated with one or more directly connected sites (CE routers).
    - VRF is associated with any type of interface, whether logical or physical (e.g. sub/virtual/tunnel) .
    - Interfaces may share the same VRF if the connected sites share the same routing information.
    - Routes are injected into the VRF from the CE-PE routing protocols for that VRF and any MP-BGP announcements that match the defined VRF.



MPLS-VPN & VRF


Service Provider
MPLS Network
PE Router
PE Router
CE Router
CE Router
CE Router
CE Router
VPN-A
VRF
RD 100:1, RT 100:1
VPN-A
VRF
RD 100:1, RT 100:1
VPN-B
VRF
RD 100:2, RT 100:2
VPN-B
VRF
RD 100:2, RT 100:2
VRF VPN-A
VRF VPN-A
GLOBAL
GLOBAL
VRF VPN-B
VRF VPN-B
MPLS LSP (Label for RD 100:1)
MPLS LSP (Label for RD 100:2)

Route Distinguisher

- To differentiate 10.0.0.0/8 in VPN-A from 10.0.0.0/8 in VPN-B.
  - 64-bit quantity.
- Configured as ASN:YY or IPADDR:YY.
  - Almost everybody uses ASN.
- Purely to make a route unique.
  - Unique route is now RD:Ipaddr (96 bits) plus a mask on the IPAddr portion.
  - So customers don’t see each others routes.

```
!
ip vrf VPN-A
rd 100:1
route-target export 100:1
route-target import 100:1
```

Route Target

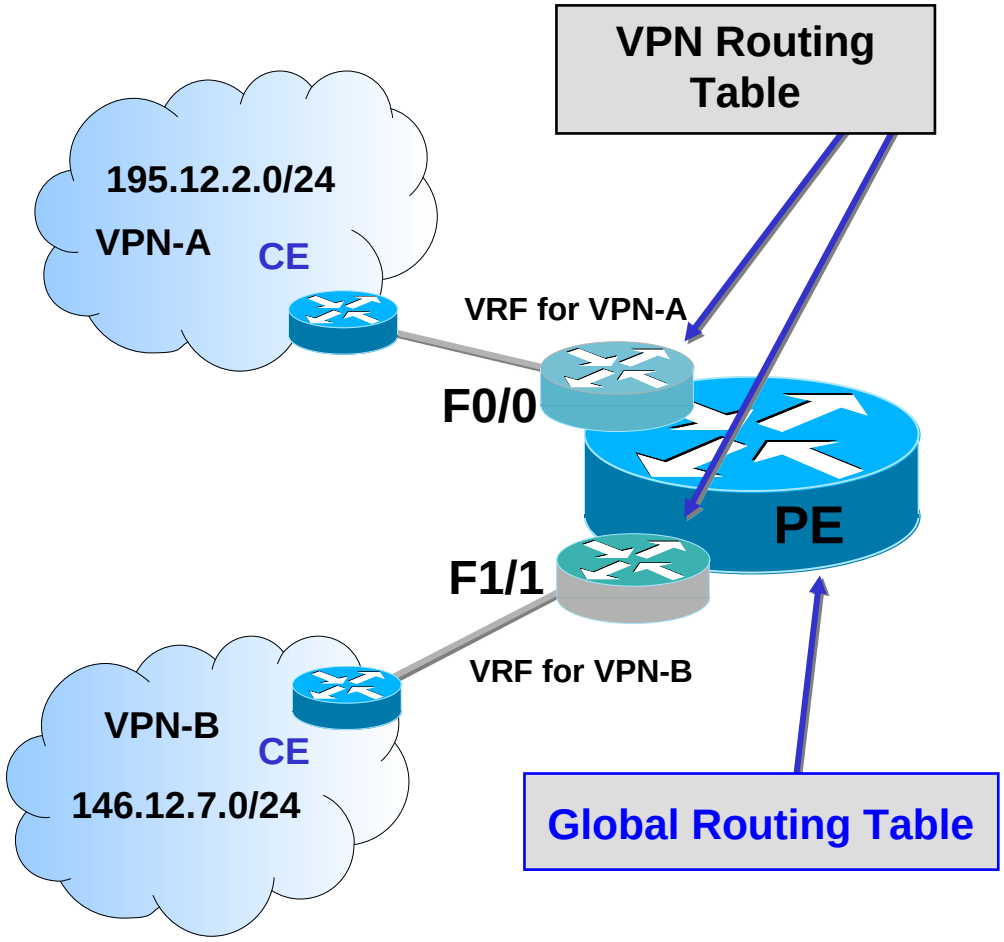
- Creates or adds to a list of VPN extended communities used to determine which routes are imported by a VRF.
- To control policy about who sees what routes.
- 64-bit quantity (2 bytes type, 6 bytes value).
- Carried as an extended community.
  - Typically written as ASN:YY.

```
!
ip vrf VPN-A
rd 100:1
route-target export 100:1
route-target import 100:1
```

- Each VRF ‘imports’ and ‘exports’ one or more RTs.
  - Exported RTs are carried in VPNv4 BGP.
  - Imported RTs are local to the box.
- A VRF PE that imports an RT installs that route in that VRF routing table.
- Allows the interconnection of different VLAN by importing/exporting other VPN routes (other RTs).
  - (Private) Routes should not conflict!

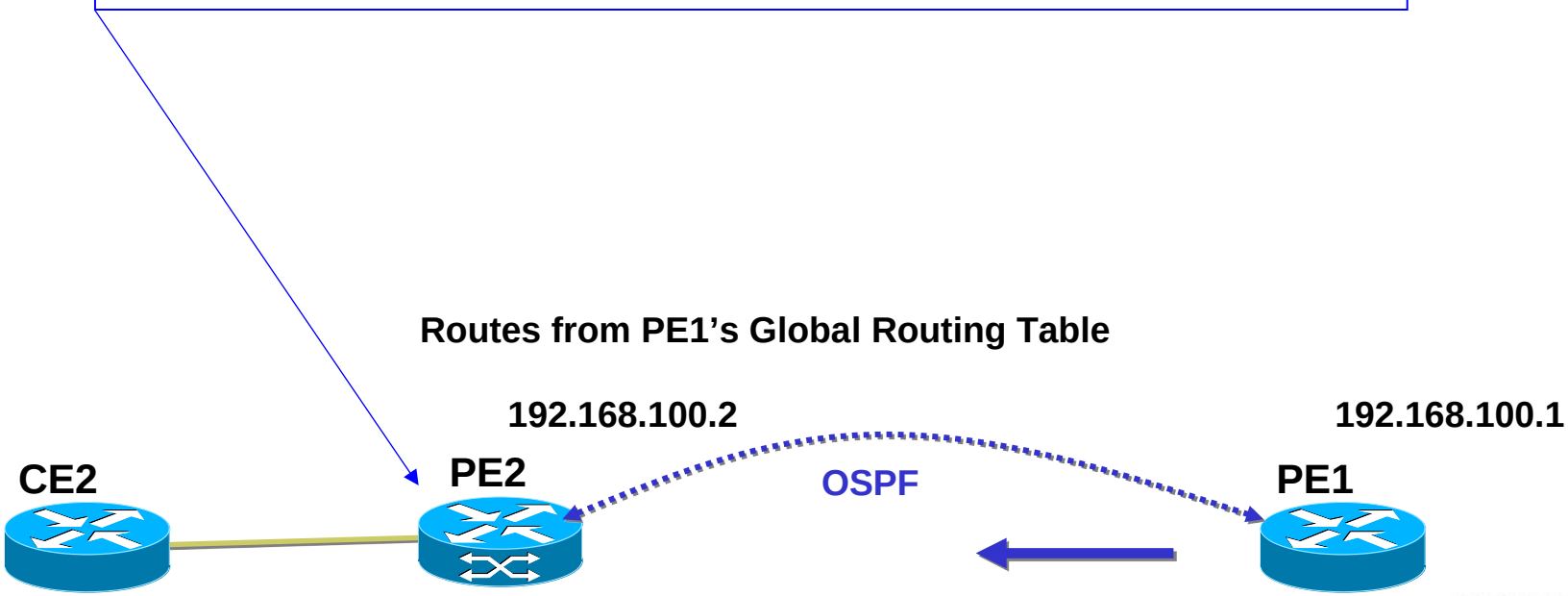
VRF Interface Definition

- Define a unique VRF for interface F0/0.
- Define a unique VRF for interface F1/1
  - Packets will never go between interfaces F0/0 and F1/1.
  - Unless Each other RT are imported.
- Uses VPNv4 to exchange VRF routing information between PE's.



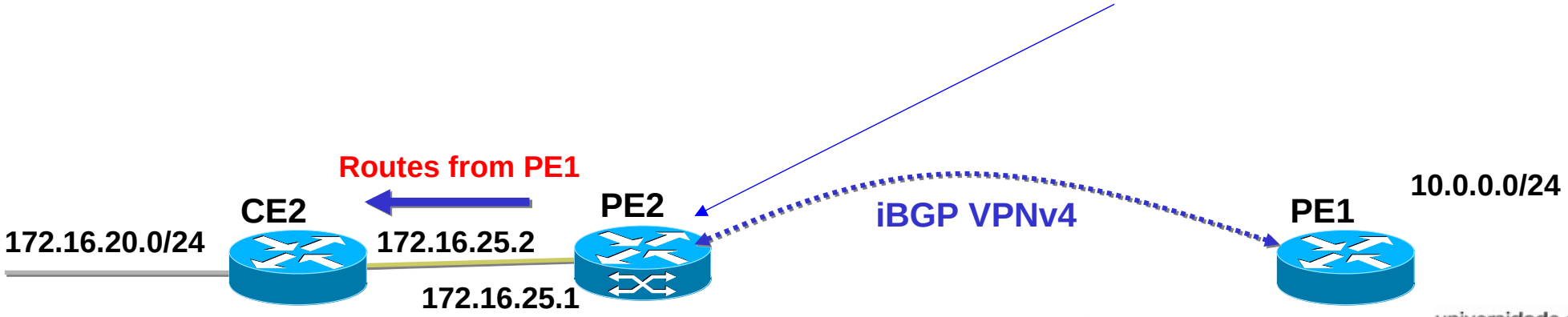
PE Router – Global Routing Table Output

```
PE2#sh ip route

Gateway of last resort is not set

C    192.168.1.0/24 is directly connected, Ethernet0/0
     192.168.100.0/32 is subnetted, 3 subnets
O       192.168.100.1 [110/11] via 192.168.1.1, 00:04:27, Ethernet0/0
C       192.168.100.2 is directly connected, Loopback0
O       192.168.100.3 [110/11] via 192.168.1.3, 00:04:27, Ethernet0/0
```

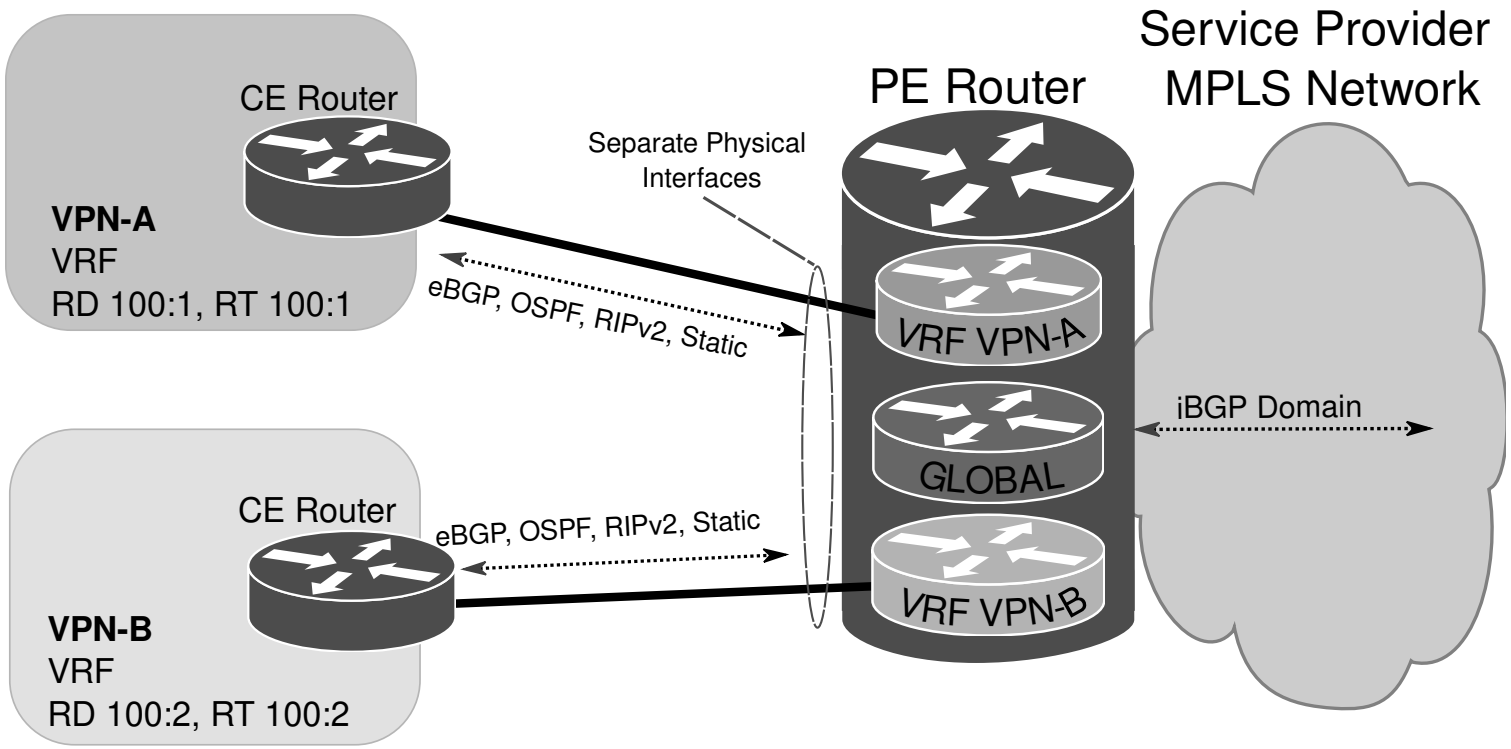


PE Router – VRF Routing Table Output

```
PE2#sh ip route vrf RED
Routing Table: RED

Gateway of last resort is 192.168.100.1 to network 0.0.0.0

     172.16.0.0/16 is variably subnetted, 8 subnets, 3 masks
C       172.16.25.0/30 is directly connected, Serial4/0
C       172.16.25.2/32 is directly connected, Serial4/0
B       172.16.20.0/24 [20/0] via 172.16.25.2, 00:07:04
     10.0.0.0/24 is subnetted, 1 subnets
B       10.0.0.0 [200/307200] via 192.168.100.1, 00:06:28
B*   0.0.0.0/0 [200/0] via 192.168.100.1, 00:07:03
```



VRF Route Population



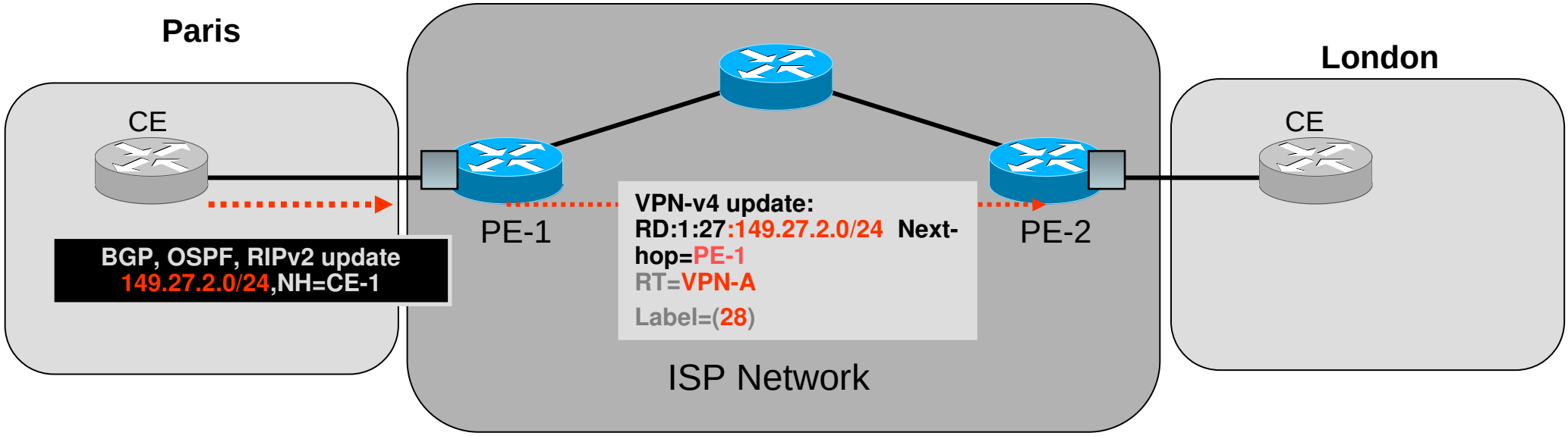
- VRF is populated locally through PE and CE routing protocol exchange.
  - EBGP, OSPF, RIPv2, and Static routing.
  - “Connected” is also supported.
- Separate routing context for each VRF.
  - Routing protocol context (e.g., MP-BGP).
  - Separate process (e.g., OSPF).

Carrying VPN Routes in BGP

- Need some way to get the VRF routing information off the PE and to other Pes.
- This is done with MP-BGP.
- Additions to MP-BGP to carry MPLS-VPN info:
  - Route Target (RT) sent in EXTENED_CIMMUNITY attribute.
  - MP_REACH_NLRI attribute for Labeled VPN IPv4 (VPNv4) address family,
    - VPN IPv4 network.
    - Route Distinguisher (RD).
    - MPLS Label.

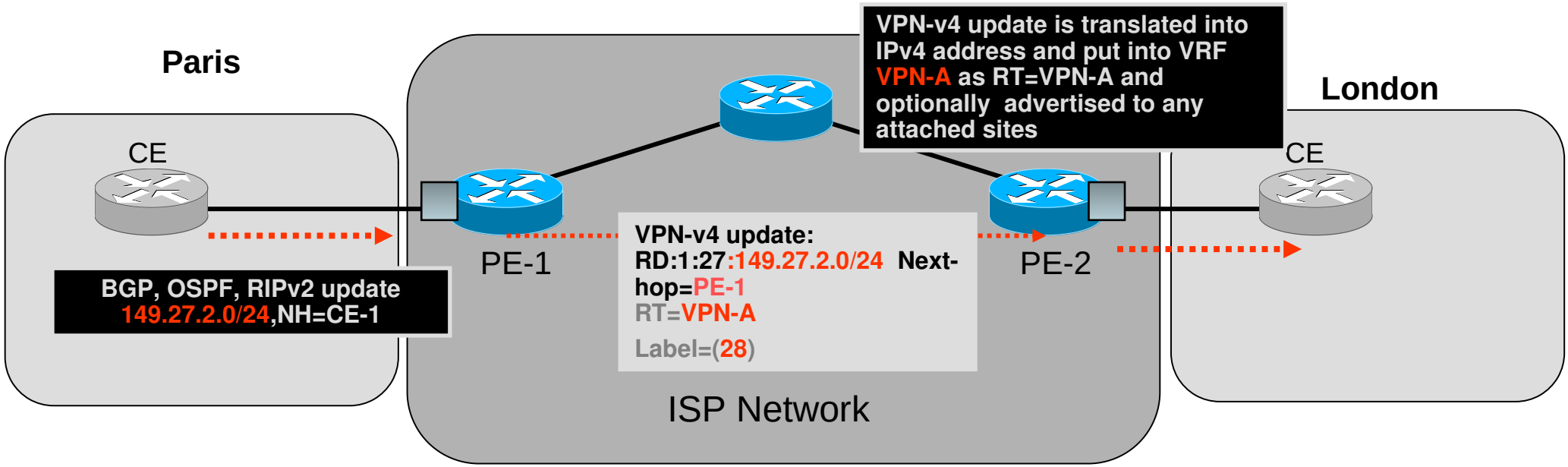
```
Border Gateway Protocol - UPDATE Message
  Marker: ffffffffffffffffffffffffffffffff
  Length: 91
  Type: UPDATE Message (2)
  Withdrawn Routes Length: 0
  Total Path Attribute Length: 68
▽ Path attributes
  ▷ Path Attribut - ORIGIN: INCOMPLETE
  ▷ Path Attribut - AS_PATH: empty
  ▷ Path Attribut - MULTI_EXIT_DISC: 0
  ▷ Path Attribut - LOCAL_PREF: 100
  ▽ Path Attribut - EXTENDED_COMMUNITIES
    ▷ Flags: 0xc0: Optional, Transitive, Complete
    Type Code: EXTENDED_COMMUNITIES (16)
    Length: 8
    ▽ Carried extended communities: (1 community)
      ▷ Community Transitive Two-Octet AS Route Target: 200:1
  ▽ Path Attribut - MP_REACH_NLRI
    ▷ Flags: 0x80: Optional, Non-transitive, Complete
    Type Code: MP_REACH_NLRI (14)
    Length: 33
    Address family: IPv4 (1)
    Subsequent address family identifier: Labeled VPN Unicast (128)
    ▷ Next hop network address (12 bytes)
    Subnetwork points of attachment: 0
    ▽ Network layer reachability information (16 bytes)
      ▷ Label Stack=24 (bottom) RD=200:1, IPv4=192.1.1.0/25
```

VRF Population of MP-BGP



- PE routers translate into VPN-V4 route
- Assigns an RD and RT based on configuration
- Re-writes Next-Hop attribute (to PE loopback)
- Assigns a label based on VRF and/or interface
- Sends MP-BGP update to all PE neighbors

VRF Population of MP-BGP

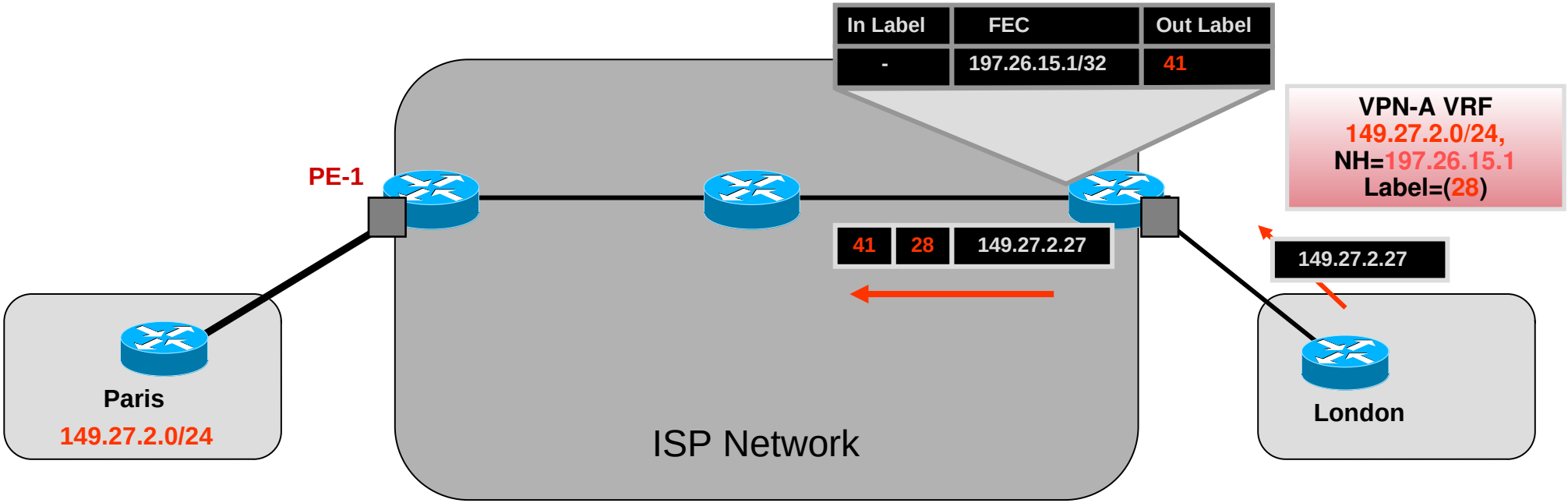


- Receiving PE routers translate to IPv4
  - Insert the route into the VRF identified by the RT attribute (based on PE configuration)
- The label associated to the VPN-V4 address will be set on packets forwarded towards the destination

MPLS/VPN Packet Forwarding

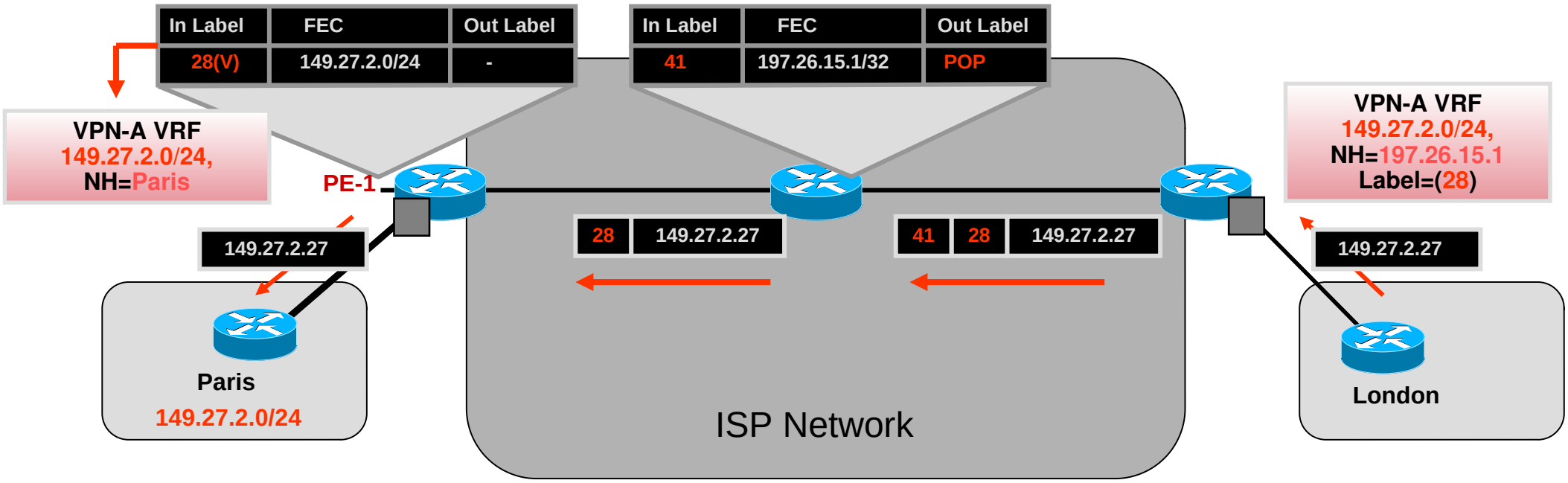
- **Between PE and CE, regular IP packets (currently)**
- **Within the provider network—label stack**
  - **Outer label: “get this packet to the egress PE”**
  - **Inner label: “get this packet to the egress CE”**
- **MPLS nodes forward packets based on TOP label!!!**
  - any subsequent labels are ignored
- **Penultimate Hop Popping procedures used one hop prior to egress PE router (shown in example)**

MPLS/VPN Packet Forwarding



- Ingress PE receives normal IP packets
- PE router performs IP Longest Match from VPN FIB (Forwarding Table), finds iBGP next-hop and imposes a stack of labels <IGP, VPN>

MPLS/VPN Packet Forwarding



- Penultimate PE router removes the IGP label
  - Penultimate Hop Popping procedures (implicit-null label)
- Egress PE router uses the VPN label to select which VPN/CE to forward the packet to
- VPN label is removed and the packet is routed toward the VPN site

Things to Note

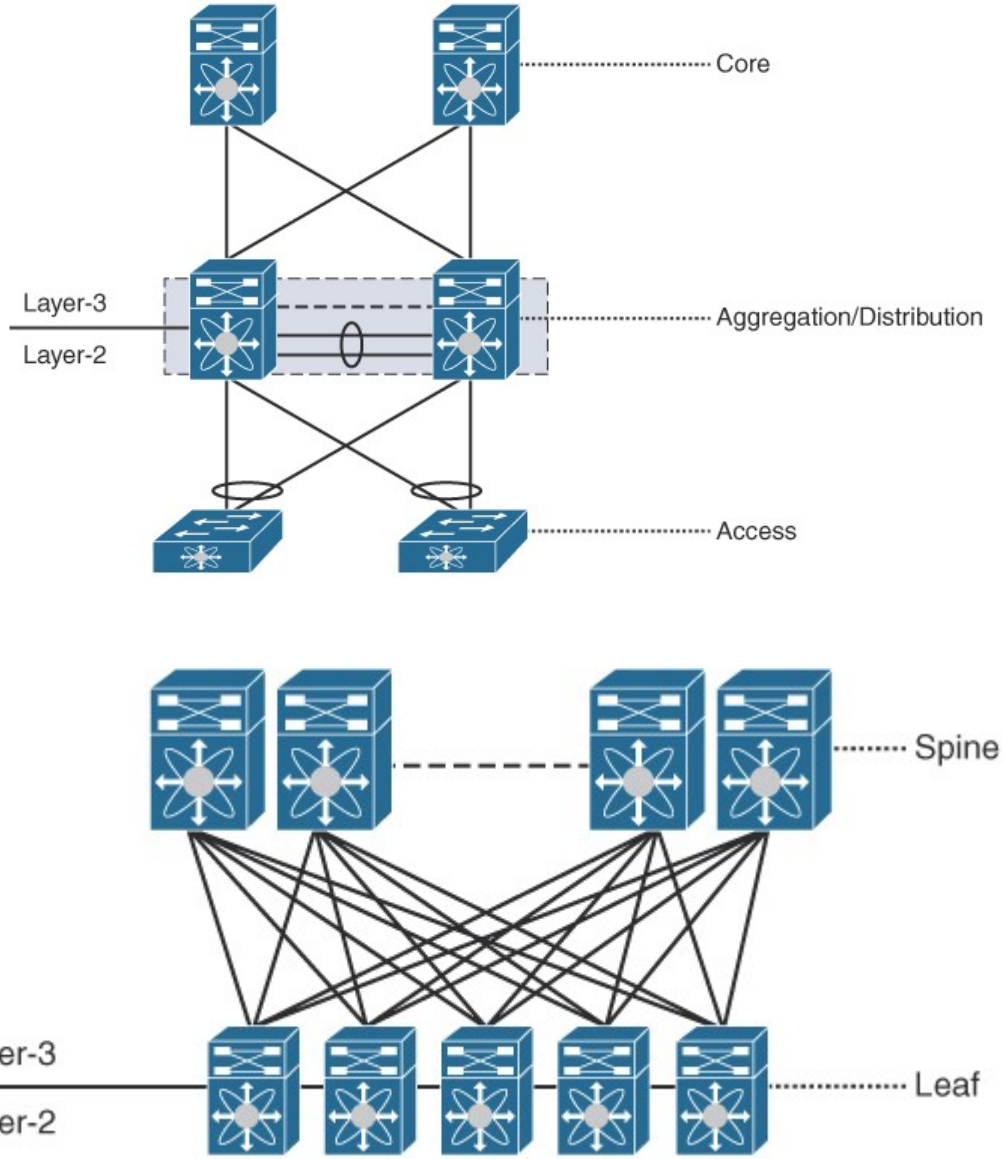
- Core does not run VPNv4 BGP!
  - Same principle can be used to run a BGP-free core for an IP network,
- CE does not know it's in an MPLS-VPN!
- Outer label is from LDP/RSVP (Core LSP).
  - Getting packet to egress PE is mutually independent to MPLS-VPN.
- Inner label is from MP-BGP (VPN LSP).
  - Inner label is there so the egress PE can have the same network in multiple VRFs.

Layer 2 VPN
VXLAN and BGP EVPN

universidade de aveiro
deti.ua.pt

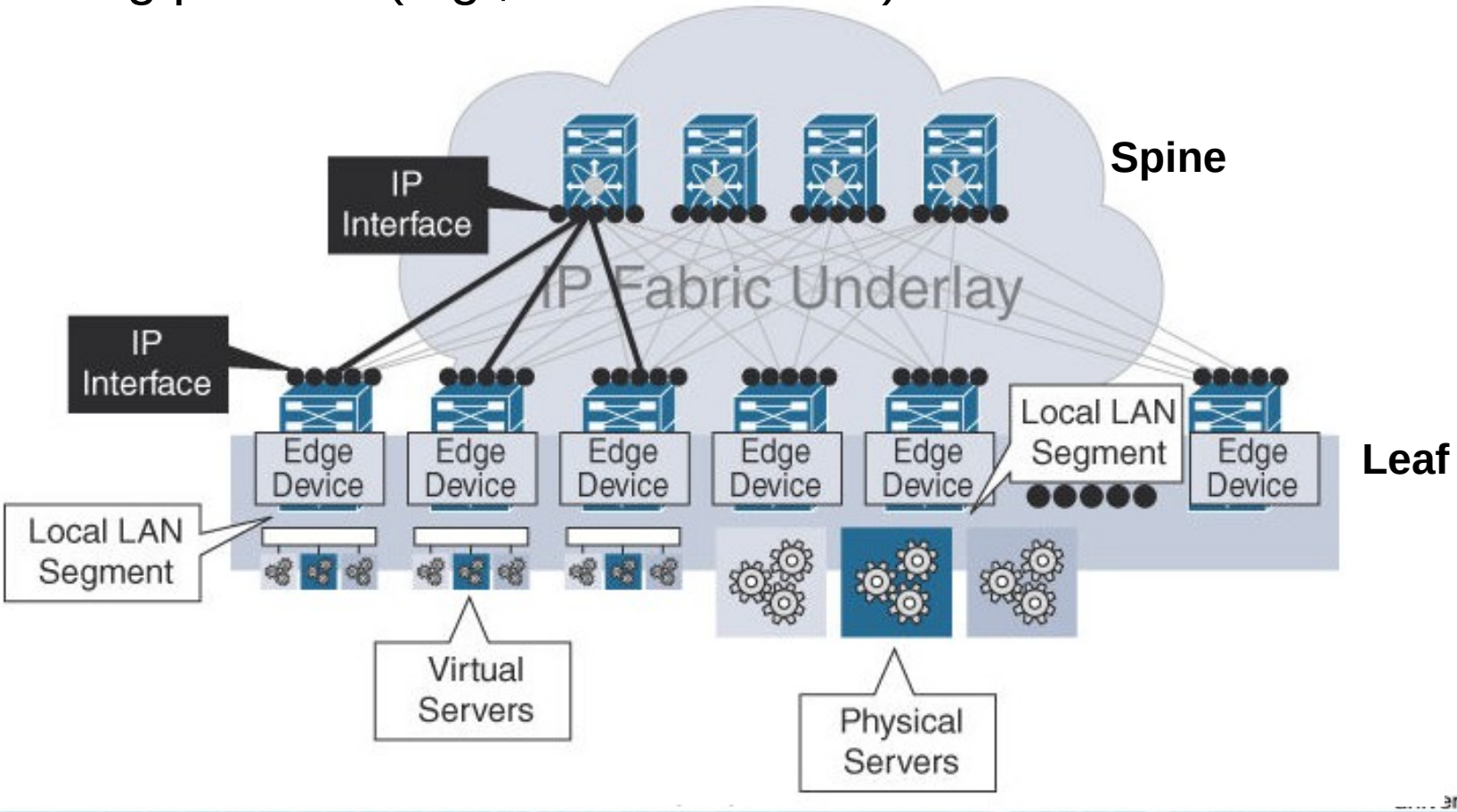
Datacenter CLOS Topology

- With large-scale data center deployments, three-tier topologies have become scale bottlenecks.
- The classic three-tier topology evolved to a CLOS topology.
  - Original designed by Charles Clos in 1950 to find a more efficient way to handle telephonic call transfers.
- Eliminating the need for STP the network evolved to greater stability and scalability.
- Layer 3 moves to the Access Layer.
- Usually called Spine-and-Leaf Architecture.



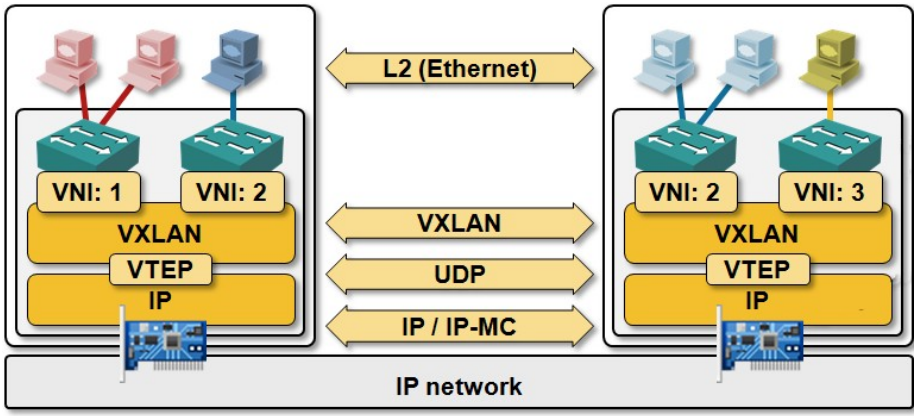
Spine-and-Leaf Architecture

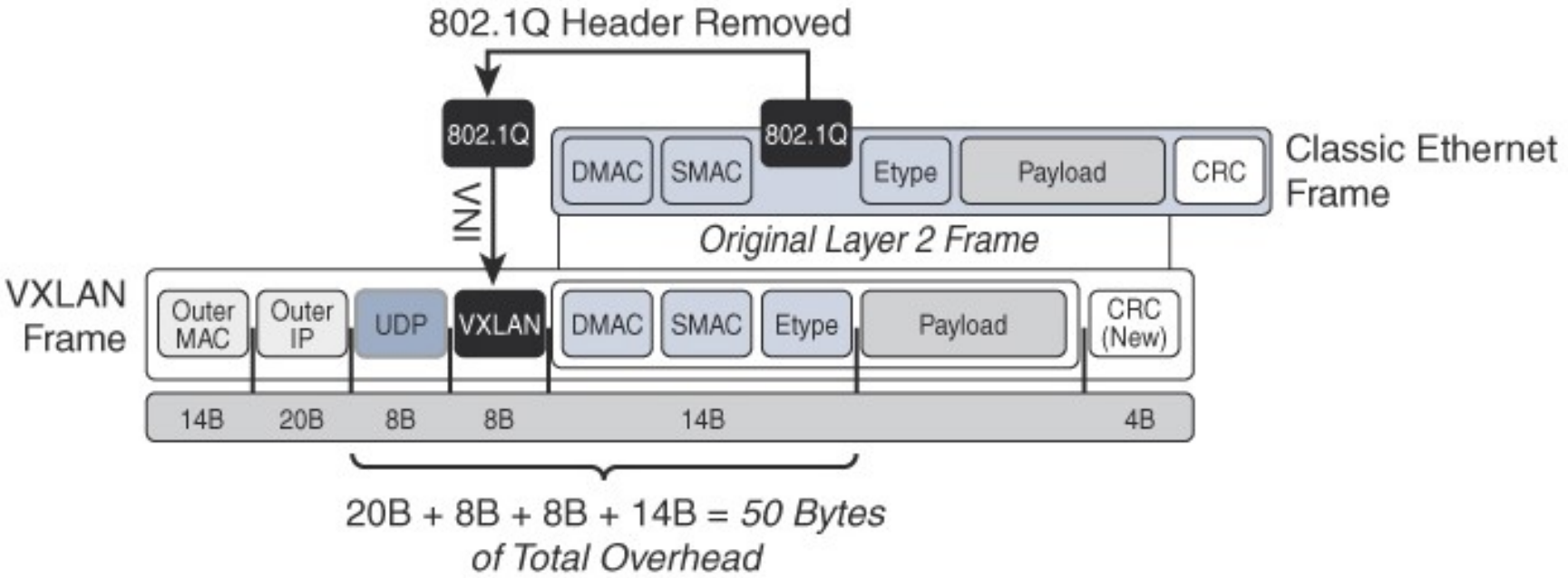
- The access layer with Layer 3 support is typically called the Leaf layer.
- The aggregation layer that provides the interconnection between the various leafs is called the Spine layer.
- The IP underlay transport between Spines and Leaves requires an IGP routing protocol (e.g., OSPF or IS-IS).



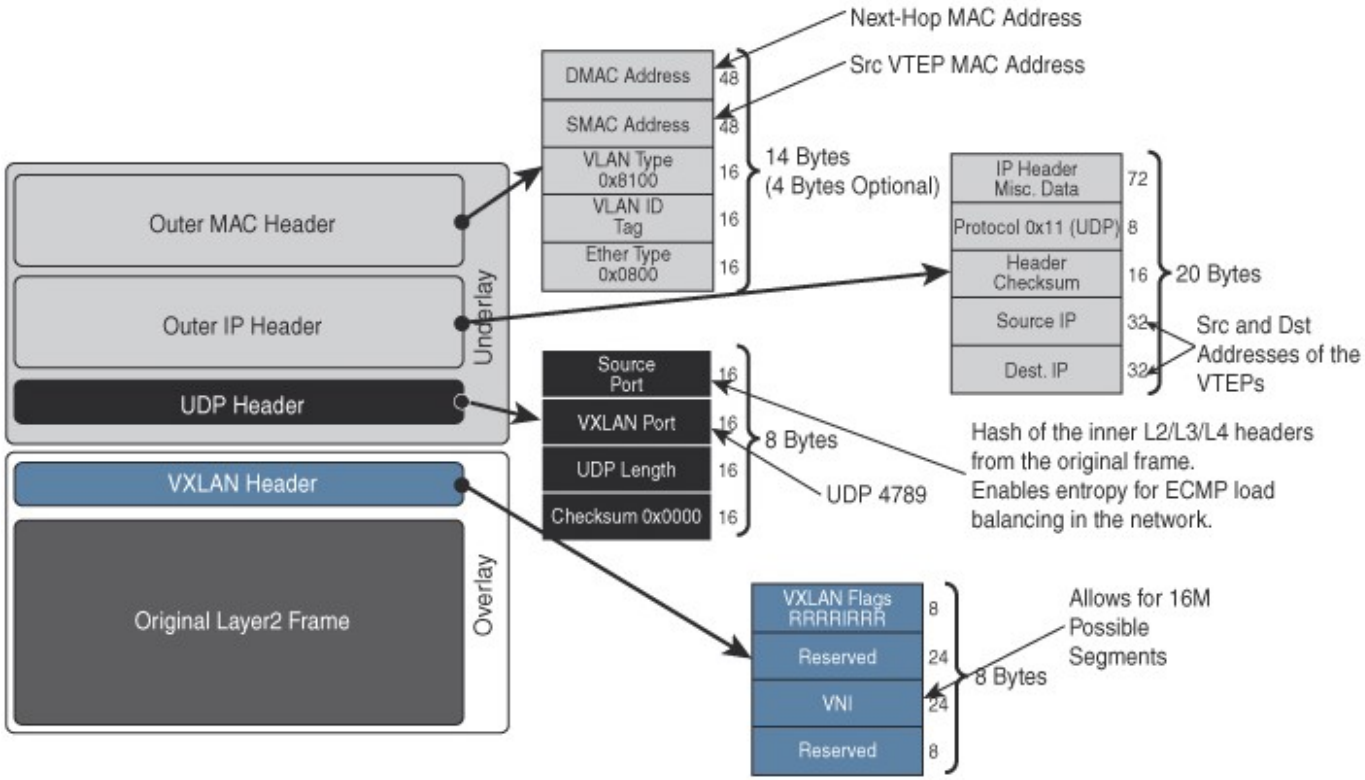
Virtual Extensible LAN (VXLAN)

- Encapsulates OSI Layer 2 Ethernet frames within Layer 4 UDP/IP datagrams .
  - Default port 4789.
- VLAN may be additionally identified by a **VNI field** with 24 bits.
  - 802.1Q tag only has 12 bits.
- The original inner 802.1Q header of the Layer 2 Ethernet frame is removed and mapped to a VNI to complete the VXLAN header.





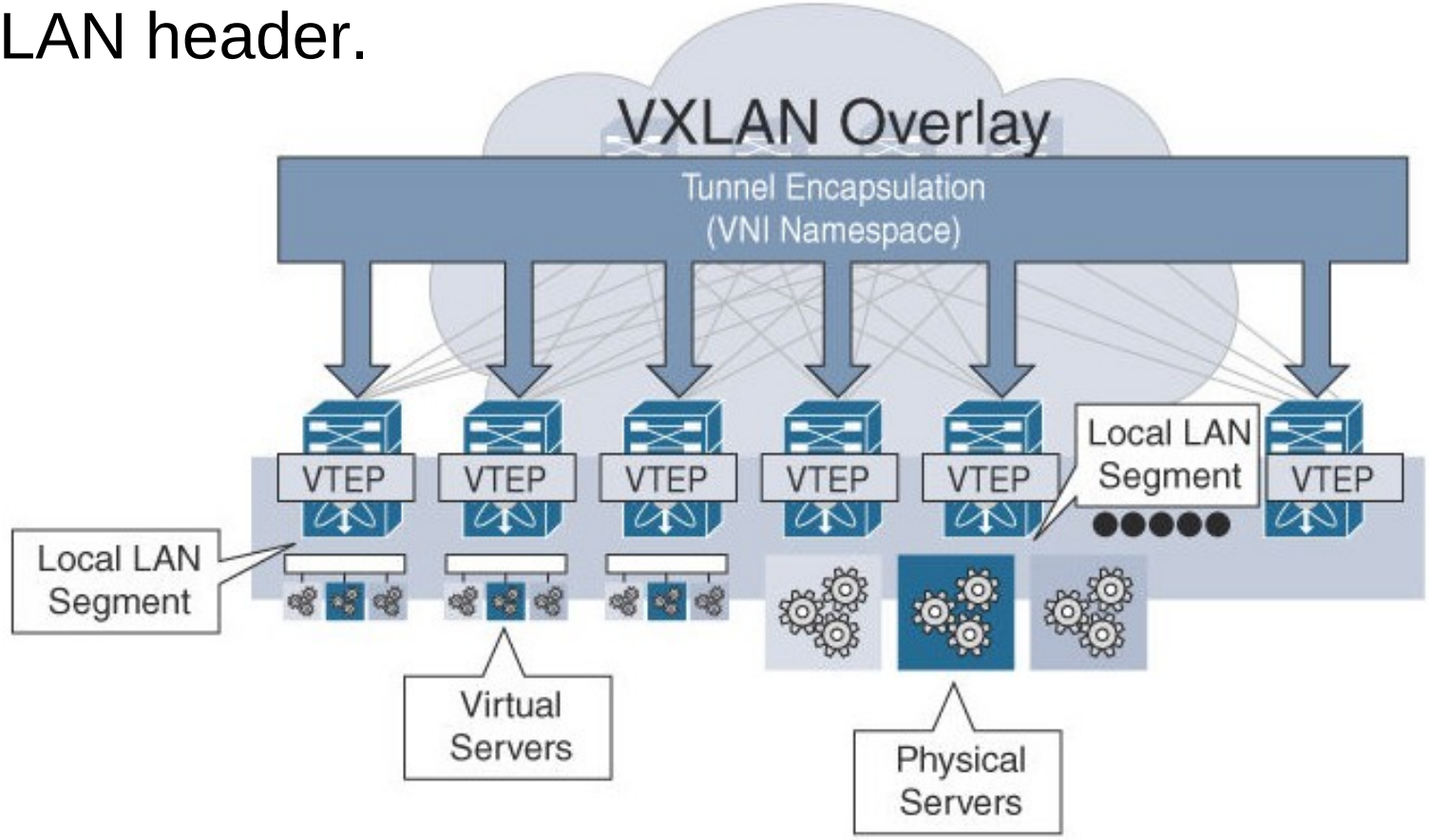
VXLAN Header/Packet




```
▸ Ethernet II, Src: ca:01:25:92:00:08 (ca:01:25:92:00:08), Dst: 0c:32:45:6d:00:00 (0c:32:45:6d:00:00)
▸ Internet Protocol Version 4, Src: 192.0.0.3, Dst: 192.0.0.1
▸ User Datagram Protocol, Src Port: 56255, Dst Port: 8472
▾ Virtual eXtensible Local Area Network
  ▸ Flags: 0x0800, VXLAN Network ID (VNI)
    Group Policy ID: 0
    VXLAN Network Identifier (VNI): 101
    Reserved: 0
▸ Ethernet II, Src: 0c:88:63:63:00:01 (0c:88:63:63:00:01), Dst: Private_66:68:00 (00:50:79:66:68:00)
▸ Internet Protocol Version 4, Src: 10.1.3.100, Dst: 10.1.1.100
▸ Internet Control Message Protocol
```

VTEP (VXLAN Tunnel Endpoint)

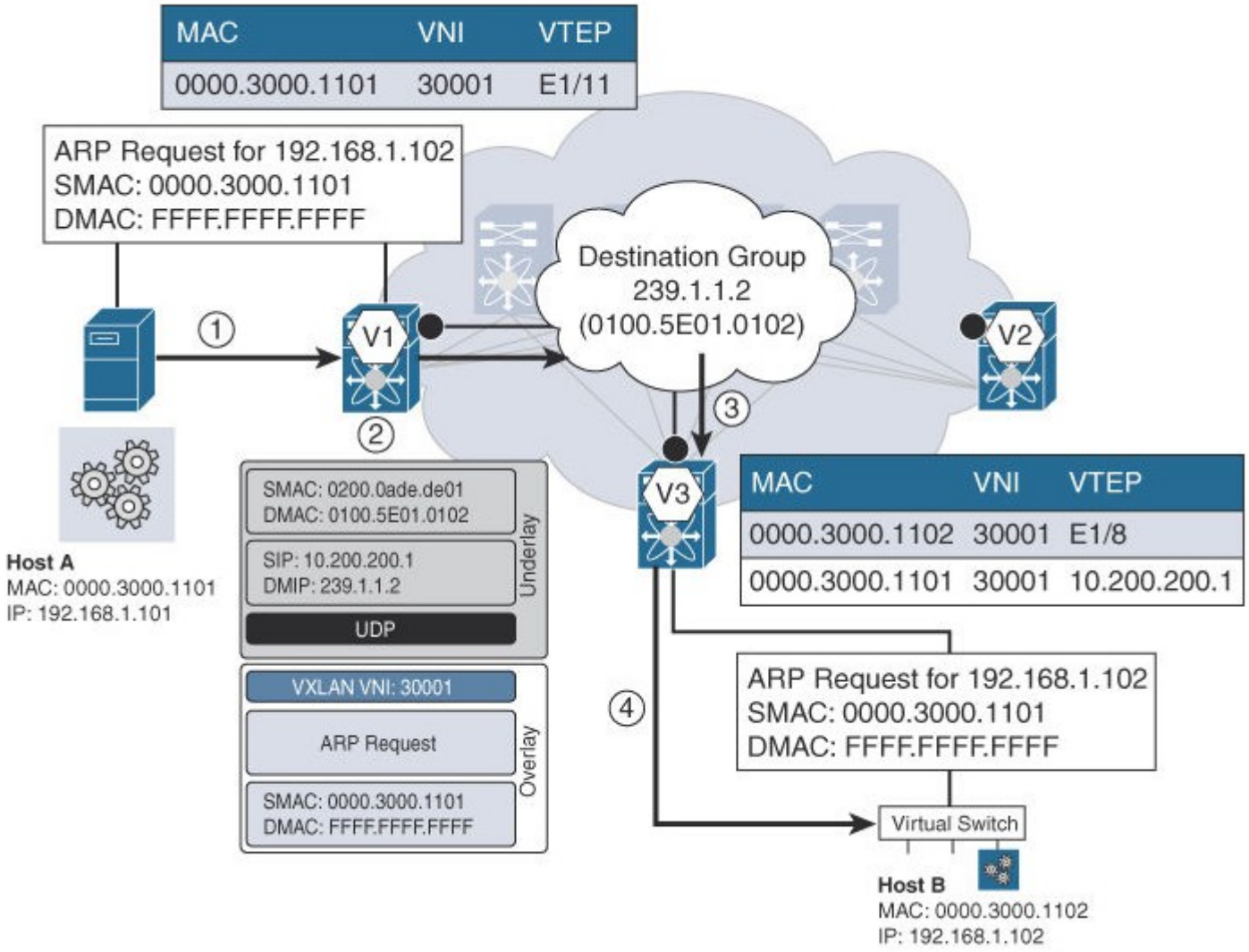
- The edge devices in a VXLAN network have the VXLAN Tunnel Endpoints (VTEP).
- Are responsible for encapsulation and decapsulation of the VXLAN header.



VTEP: VXLAN Tunnel Endpoint
VNI/VNID: VXLAN Network Identifier

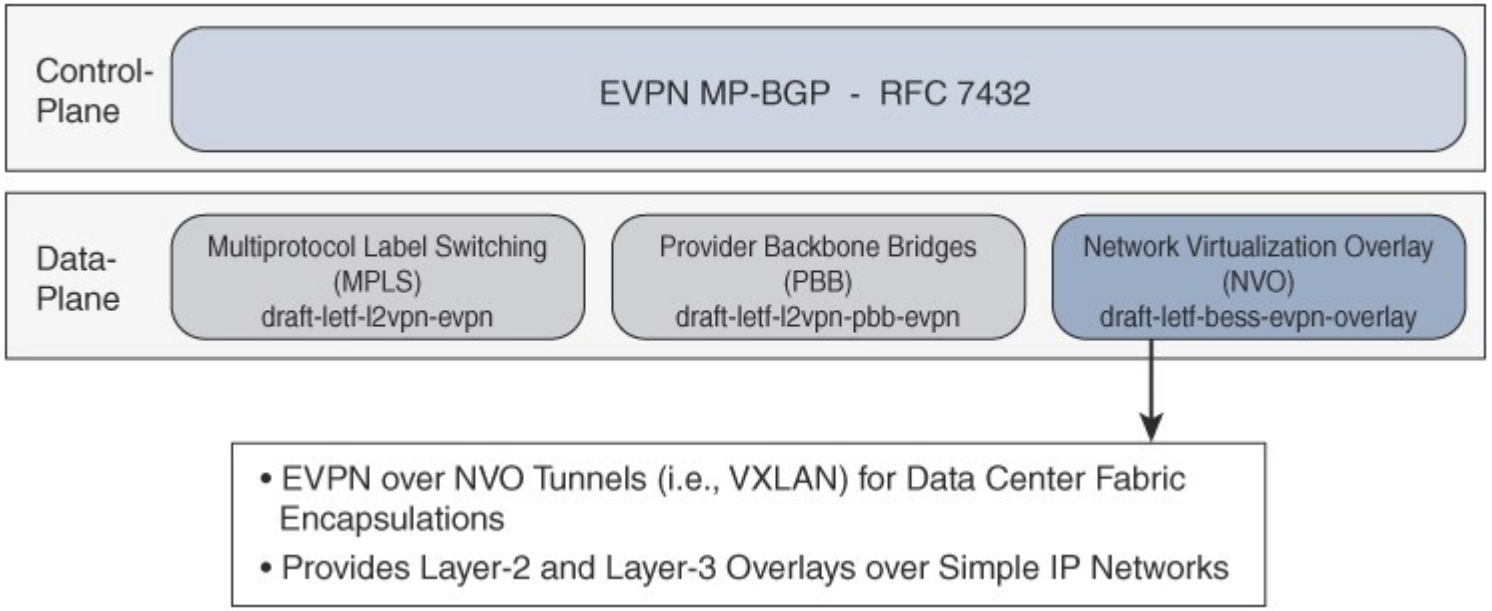
VXLAN Flood and Learn

- The multidestination traffic is flooded over the VXLAN between VTEPs.
  - To learn about the host MACs located behind the VTEPs so that subsequent traffic can be unicast.
  - This is referred to as an F&L mechanism.
- A native F&L based approach is far from optimal since the broadcast domain for a VXLAN now spans Layer 3 boundaries.



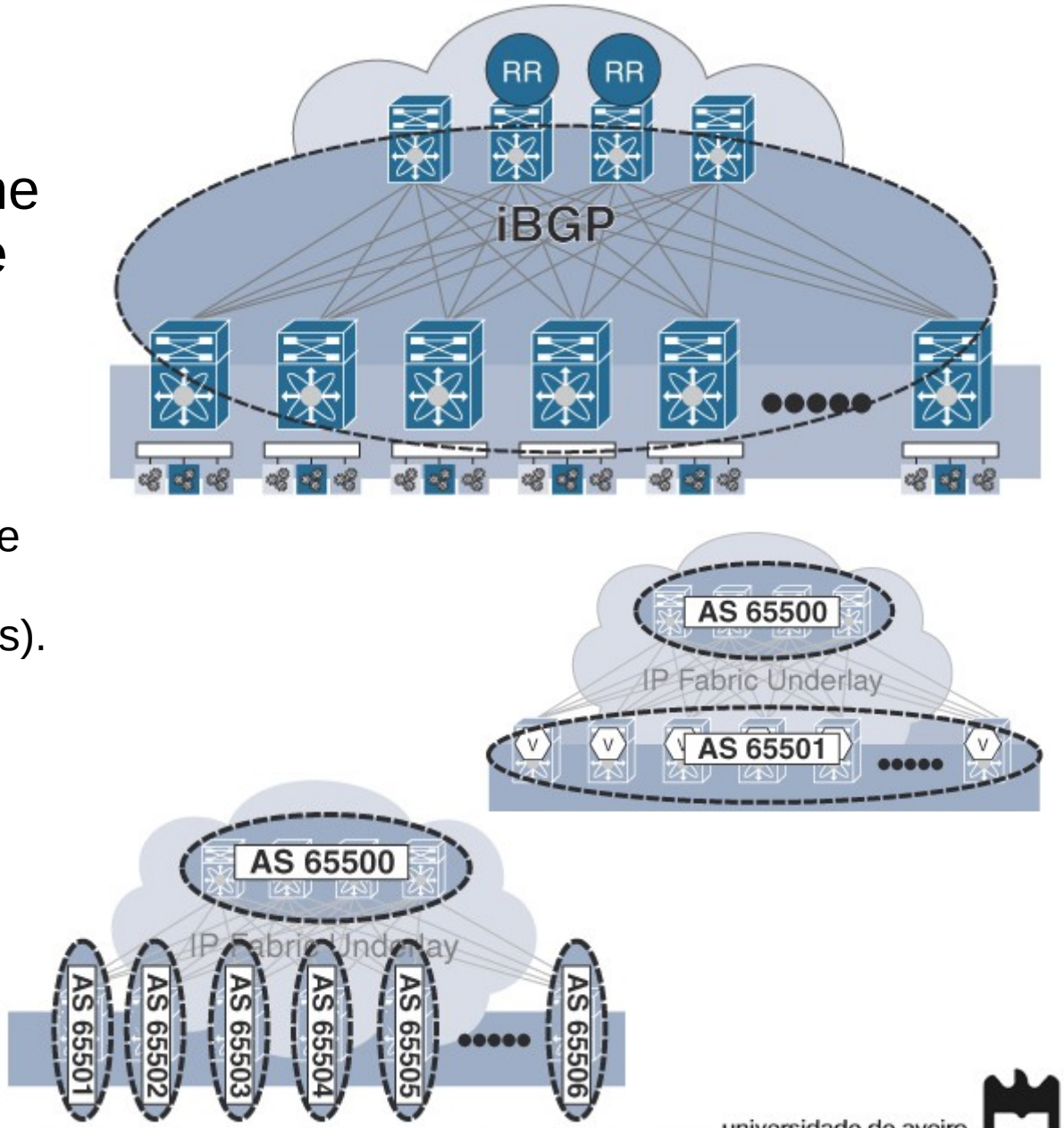
EVPN MP-BGP

- To mitigate the VXLAN Flood and Learn problem, it was introduced the concept of Ethernet VPN with MP-BGP, provided by the Address family L2VPN EVPN.
- The Address family L2VPN EVPN provides a method to transport VPN-aware Layer 2 (MAC) and Layer 3 (IP) information across a single MP-BGP peering session.
- The EVPN MP-BGP RFC allow for multiple data plane transport: MPLS, PBB and NVO.
  - A possible (and more common) EVPN over NVO solution for datacenters is VXLAN.



BGP EVPN wih VXLAN

- BGP is used to announce and learn remote VTEP addresses.
- VXLAN is used to transport to the specific remote VTEP where the destination device is.
- BGP relations can be:
  - Only internal BGP.
    - To avoid a full BGP mesh, Route Reflectors should be used (usually all or some of the spines).
  - External BGP relations between private AS.
    - Leaf in a single private AS.
    - Each Leaf is a private AS.



EVPN Route types

- Route Type-2
  - Defines the MAC/IP advertisement route.
  - Responsible for the distribution of MAC and IP address reachability information.
- Route Type-3
  - Called “inclusive multicast Ethernet tag route”.
  - Used to create the a distribution list for unknown unicast, multicast and broadcast packets (ingress replication).
  - Provides a way to replicate multidestination traffic in a unicast.

RD (8 Octets)
ESI (10 Octets)
Ethernet Tag ID (4 Octets)
MAC Address Length (1 Octet)
MAC Address (6 Octets)
IP Address Length (1 Octet)
IP Address (0, 4, or 16 Octets)
MPLS Label1 (3 Octets)
MPLS Label2 (0 or 3 Octets)

RD (8 Octets)
ESI (10 Octets)
Ethernet Tag ID (4 Octets)
IP Address Length (1 Octet)
Originating Router's IP Address (4 or 16 Octets)

EVPN Route Type-2

```
Border Gateway Protocol - UPDATE Message
  Marker: ffffffffffffffffffffffffffffffff
  Length: 144
  Type: UPDATE Message (2)
  Withdrawn Routes Length: 0
  Total Path Attribute Length: 121
 Path attributes
  Path Attribute - MP_REACH_NLRI
   Flags: 0x90, Optional, Extended-Length, Non-transitive, Complete
    Type Code: MP_REACH_NLRI (14)
    Length: 83
    Address family identifier (AFI): Layer-2 VPN (25)
    Subsequent address family identifier (SAFI): EVPN (70)
   Next hop: 192.0.0.3
    Number of Subnetwork points of attachment (SNPA): 0
   Network Layer Reachability Information (NLRI)
    EVPN NLRI: MAC Advertisement Route
      Route Type: MAC Advertisement Route (2)
      Length: 33
      Route Distinguisher: 0001c00000030003 (192.0.0.3:3)
     ESI: 00:00:00:00:00:00:00:00:00:00
      Ethernet Tag ID: 0
      MAC Address Length: 48
      MAC Address: Private_66:68:02 (00:50:79:66:68:02)
      IP Address Length: 0
     IP Address: NOT INCLUDED
      VNI: 101
    EVPN NLRI: MAC Advertisement Route
      Route Type: MAC Advertisement Route (2)
      Length: 37
      Route Distinguisher: 0001c00000030003 (192.0.0.3:3)
     ESI: 00:00:00:00:00:00:00:00:00:00
      Ethernet Tag ID: 0
      MAC Address Length: 48
      MAC Address: Private_66:68:02 (00:50:79:66:68:02)
      IP Address Length: 32
      IPv4 address: 10.1.3.100
      VNI: 101
  Path Attribute - ORIGIN: IGP
  Path Attribute - AS_PATH: empty
  Path Attribute - LOCAL_PREF: 100
  Path Attribute - EXTENDED_COMMUNITIES
   Flags: 0xc0, Optional, Transitive, Complete
    Type Code: EXTENDED_COMMUNITIES (16)
    Length: 16
   Carried extended communities: (2 communities)
    Encapsulation: VXLAN Encapsulation [Transitive Opaque]
    Route Target: 100:101 [Transitive 2-Octet AS-Specific]
```

- Announces a MAC address and respective IP address of a remote device.
  - And respective next-hop.
- EXTENDED_COMMUNITY attribute is uses to announce the type of encapsulation and the route target.
- Sent when Leaf device learns a new MAC address.

EVPN Route Type-3

```
▾ Border Gateway Protocol - UPDATE Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 122
    Type: UPDATE Message (2)
    Withdrawn Routes Length: 0
    Total Path Attribute Length: 99
  ▾ Path attributes
    ▾ Path Attribute - MP_REACH_NLRI
      ▸ Flags: 0x90, Optional, Extended-Length, Non-transitive, Complete
        Type Code: MP_REACH_NLRI (14)
        Length: 28
        Address family identifier (AFI): Layer-2 VPN (25)
        Subsequent address family identifier (SAFI): EVPN (70)
      ▸ Next hop: 192.0.0.2
        Number of Subnetwork points of attachment (SNPA): 0
      ▾ Network Layer Reachability Information (NLRI)
        ▾ EVPN NLRI: Inclusive Multicast Route
            Route Type: Inclusive Multicast Route (3)
            Length: 17
            Route Distinguisher: 0001c00000020002 (192.0.0.2:2)
            Ethernet Tag ID: 0
            IP Address Length: 32
            IPv4 address: 192.0.0.2
    ▸ Path Attribute - ORIGIN: IGP
    ▸ Path Attribute - AS_PATH: empty
    ▸ Path Attribute - MULTI_EXIT_DISC: 0
    ▸ Path Attribute - LOCAL_PREF: 100
    ▸ Path Attribute - ORIGINATOR_ID: 192.0.0.2
    ▸ Path Attribute - CLUSTER_LIST: 192.0.0.1
    ▸ Path Attribute - EXTENDED_COMMUNITIES
    ▾ Path Attribute - PMSI_TUNNEL_ATTRIBUTE
      ▸ Flags: 0xc0, Optional, Transitive, Complete
        Type Code: PMSI_TUNNEL_ATTRIBUTE (22)
        Length: 9
        Flags: 0
        Tunnel Type: Ingress Replication (6)
        VNI: 102
      ▸ Tunnel ID: tunnel end point -> 192.0.0.2
```

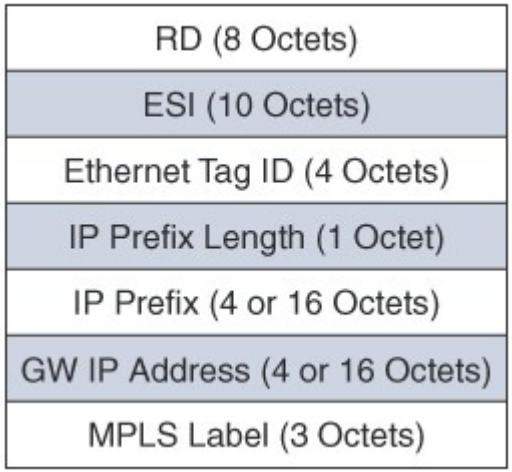
- Defines the next hop for unknown unicast, multicast and broasvast.
- Must also carry a Provider Multicast Service Interface (PMSI) Tunnel attribute.
  - Defines tunnel type.
  - For EVPN with VXLAN the tunnel type is “Ingress Replication”.
- Sent when a new Leaf (BGP peer) is added.

BGP Route Table – L2VPN EVPN

```
# show bgp l2vpn evpn
BGP table version is 1, local router ID is 192.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN type-1 prefix: [1]:[EthTag]:[ESI]:[IPlen]:[VTEP-IP]:[Frag-id]
EVPN type-2 prefix: [2]:[EthTag]:[MAClen]:[MAC]:[IPlen]:[IP]
EVPN type-3 prefix: [3]:[EthTag]:[IPlen]:[OrigIP]
EVPN type-4 prefix: [4]:[ESI]:[IPlen]:[OrigIP]
EVPN type-5 prefix: [5]:[EthTag]:[IPlen]:[IP]

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 192.0.0.1:2
*> [3]:[0]:[32]:[192.0.0.1]
                    192.0.0.1                              32768 i
                    ET:8 RT:100:102
...
*>i[2]:[0]:[48]:[00:50:79:66:68:01]
                    192.0.0.2                100      0 i
                    RT:100:101 ET:8
...
*>i[2]:[0]:[48]:[00:50:79:66:68:02]:[32]:[10.1.3.100]
                    192.0.0.3                100      0 i
                    RT:100:101 ET:8
...
*>i[3]:[0]:[32]:[192.0.0.3]
                    192.0.0.3                100      0 i
                    RT:100:101 ET:8
```

Layer 3 VPN over EVPN with VXLAN

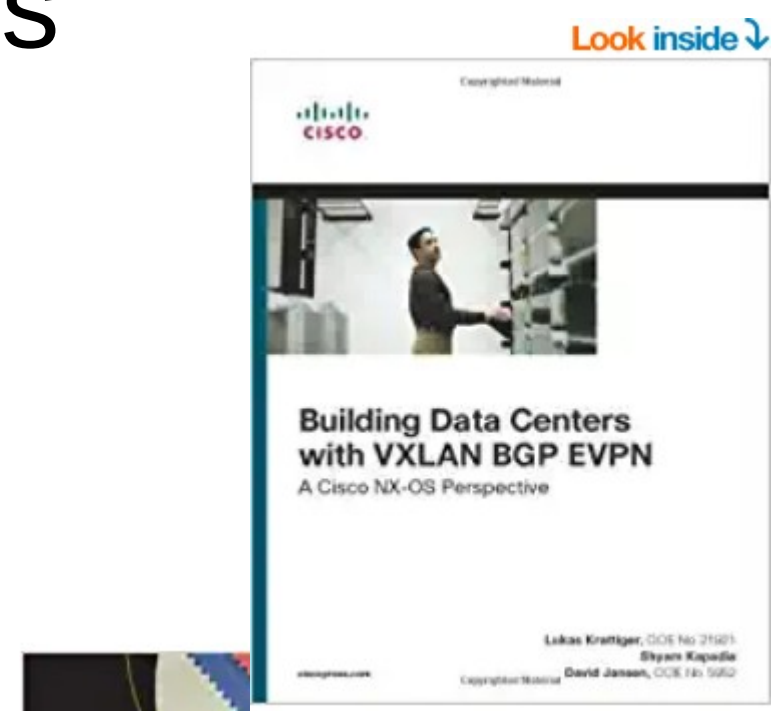
- As an alternative Layer3 VPN to MPLS VPN, it is possible to create a Layer3 VPN over an EVPn with VXLAN.
  - Using announcements of Route Type-5.
- Route Type-5
  - Announces IP network prefixes.




```
▾ Border Gateway Protocol - UPDATE Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 121
    Type: UPDATE Message (2)
    Withdrawn Routes Length: 0
    Total Path Attribute Length: 98
  ▾ Path attributes
    ▾ Path Attribute - MP_REACH_NLRI
      ▸ Flags: 0x90, Optional, Extended-Length, Non-transitive, Complete
        Type Code: MP_REACH_NLRI (14)
        Length: 45
        Address family identifier (AFI): Layer-2 VPN (25)
        Subsequent address family identifier (SAFI): EVPN (70)
      ▸ Next hop: 192.0.0.1
        Number of Subnetwork points of attachment (SNPA): 0
      ▾ Network Layer Reachability Information (NLRI)
        ▾ EVPN NLRI: IP Prefix route
            Route Type: IP Prefix route (5)
            Length: 34
            Route Distinguisher: 00010a0101010004 (10.1.1.1:4)
          ▸ ESI: 00:00:00:00:00:00:00:00:00:00
            Ethernet Tag ID: 0
            IP prefix length: 16
            IPv4 address: 10.1.0.0
            IPv4 Gateway address: 0.0.0.0
            VNI: 101
    ▸ Path Attribute - ORIGIN: INCOMPLETE
    ▸ Path Attribute - AS_PATH: empty
    ▸ Path Attribute - MULTI_EXIT_DISC: 0
    ▸ Path Attribute - LOCAL_PREF: 100
    ▾ Path Attribute - EXTENDED_COMMUNITIES
      ▸ Flags: 0xc0, Optional, Transitive, Complete
        Type Code: EXTENDED_COMMUNITIES (16)
        Length: 24
      ▾ Carried extended communities: (3 communities)
        ▸ Encapsulation: VXLAN Encapsulation [Transitive Opaque]
        ▸ Route Target: 100:101 [Transitive 2-Octet AS-Specific]
        ▸ EVPN Router's MAC: Router's MAC: 0c:32:45:6d:00:01 [Transitive EVPN]
```

References

- Building Data Centers with VXLAN BGP EVPN: A Cisco NX-OS Perspective (Networking Technology), 1st Edition, David Jansen, Lukas Krattiger, Shyam Kapadia, Cisco Press (March 31, 2017), ISBN-13: 978-1587144677.
- The Fast-Track Guide to VXLAN BGP EVPN Fabrics: Implement Today's Multi-Tenant Software-Defined Networks, 1st Edition, Rene Cardona, Apress (May 19, 2021), ISBN-13:978-1484269299.



SNMP

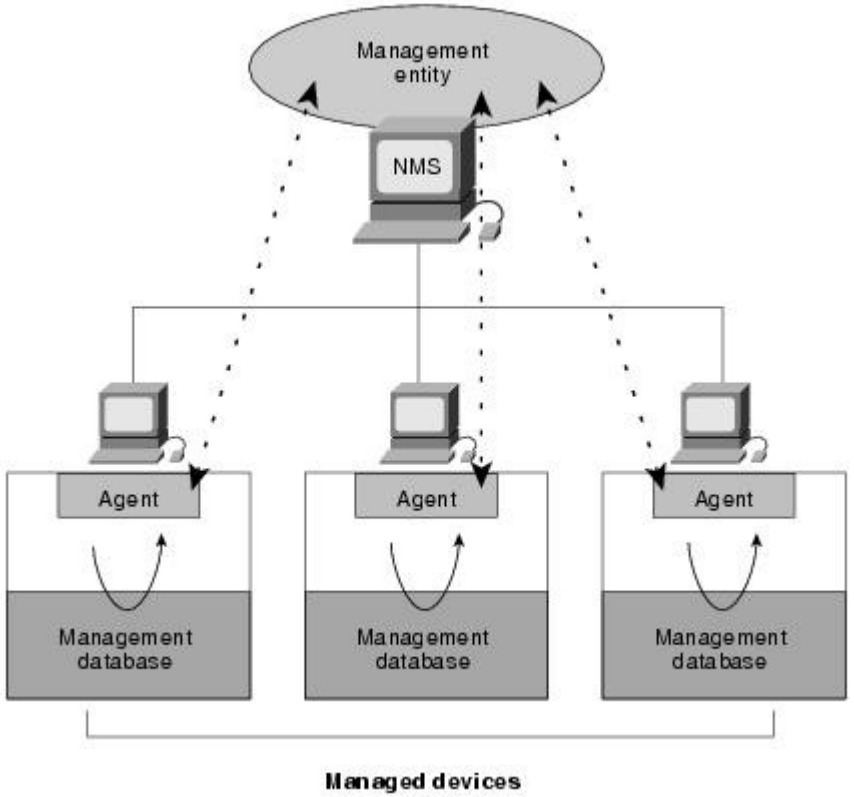
Simple Network Management Protocol

universidade de aveiro

deti.ua.pt

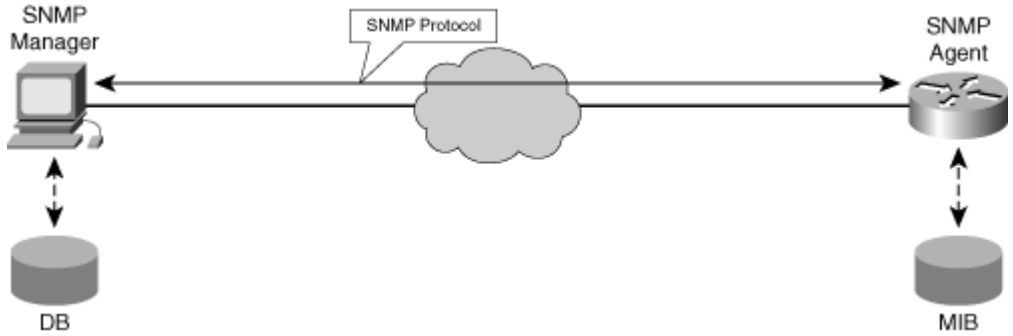
SNMP Basic Components

- An SNMP-managed network consists of three key components:
- Managed devices
  - Network node that contains an SNMP agent.
  - Collect and store management information and make this information available using SNMP.
  - Can be routers and access servers, switches and bridges, hubs, computer hosts, or printers.
- Agents
  - Network-management software module that resides in a managed device.
- Network-management systems (NMSs)
  - Executes applications that monitor and control managed devices.
  - Provide the bulk of the processing and memory resources required for network management.
  - One or more NMSs must exist on any managed network.



Data Collection Protocols: SNMP, SMI, and MIB

- SNMP is an Internet protocol developed by the IETF
- It is designed to facilitate the exchange of management information between network elements



- SNMP agent
  - A software module that resides in network elements; it collects and stores management information specified in the supported MIB modules. The SNMP agent responds to SNMP requests from an NMS station for information and actions. The SNMP agent can send fault notifications pro-actively to the SNMP manager.
- Managed object
  - A representation of something that can be managed.
  - Managed objects differ from variables, which are particular object instances.
- Management Information Base (MIB)
  - A collection of managed objects residing in a virtual information store.
  - A collection of related managed objects is defined in a specific MIB module.
  - A MIB can be considered a local data store at the network element.
- Syntax notation
  - A language used to describe managed objects in a machine-independent format
  - SNMP-based management systems use a subset of the International Organization for Standardization's (ISO) Open System Interconnection (OSI) Abstract Syntax Notation 1 (ASN.1, International Telecommunication Union Recommendation X.208) to define both the packets exchanged by the management protocol and the objects that are to be managed.
- Structure of Management Information (SMI)
  - Defines the rules for describing management information (the MIB). The SMI is defined using ASN.1.

SNMP Basic Commands

- Managed devices are monitored and controlled using four basic SNMP commands: read, write, trap, and traversal operations.
  - The **read** command is used by an NMS to monitor managed devices. The NMS examines different variables that are maintained by managed devices.
  - The **write** command is used by an NMS to control managed devices. The NMS changes the values of variables stored within managed devices.
  - The **trap** command is used by managed devices to asynchronously report events to the NMS. When certain types of events occur, a managed device sends a trap to the NMS.
  - Traversal operations are used by the NMS to determine which variables a managed device supports and to sequentially gather information in variable tables, such as a routing table.

SNMP: Polling

- Manager periodically asks the agent for new information

☺ Advantage: Manager completely controls the equipment, and knows all network details

☹ Disadvantage: Delay between event and its entry in the system, and unnecessary communication overhead:

  - Slow polling, slow answer to the events
  - Quick polling, quick reaction, but large bandwidth wastage

SNMP: Traps

- There is an event → trap is sent
- Trap contains appropriate information

  equipment name, time instant of event, type of event

☺ Advantage: information only generated when required

☹ Disadvantage:

- ☹ More resources required in the managed equipment
- ☹ Traps can be useless
  - If many events occur, bandwitdh can be wasted with all traps (thresholds can solve)
  - Since the agent has only a limited scope of the network, NMS may already know about the events.

- Traps&Polling
  - Event occurs → trap is sent
  - Manager performs polling to obtain the rest of information
  - Manager also performs periodic polling, as backup

SNMP Versions

Model	Level	Authentication	Encryption	What Happens
v1	noAuthNoPriv	Community String	No	Uses a community string match for authentication.
v2c	noAuthNoPriv	Community String	No	Uses a community string match for authentication.
v3	noAuthNoPriv	Username	No	Uses a username match for authentication.
v3	authNoPriv	MD5 or SHA	No	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithm.
v3	authPriv	MD5 or SHA	DES or AES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides DES 56-bit or CFB128-AES-128 encryption in addition to authentication based on the CBC-DES (DES-56) standard.

SNMPv1: security and authentication

- In its initial version, authorization and authentication were based on the notion of “SNMP community string”
- The “words of community” identify the permissions of the machine accessing the agent: read-only ou read-write
- By default, all systems are configured with the community strings:
  - public (read-only)
  - private (read-write)
- The words are case sensitive.

SNMPv2c and SNMPv3 versions

- SNMPv2 extensions
  - Structure of management information (SMI)
  - Manager-Manager capacity
  - New protocol operations
- SNMPv3 extensions
  - New message format
  - Message security
  - Access control

SNMPv3: Security

- Notion of “access control dependent on the user”
  - The agent mantains access rights information (policies) to different users in a data base
- Authentication: shared secret key
  - MD5 or SHA authentication passphrase hashes
- Privacy
  - Packet data may now be DES encrypted (future use allows additional encryption)
  - Passphrase defaults to authentication passphrase
  - Allows for unique Privacy passphrase
- Protection against replays: resort to nounces

SNMPv1 Message

- Version: SNMP version.
- Community: Community name, used for the authentication between an agent and the NMS.
  - In Get or GetNext operations, read community name is used for authentication;
  - In Set operation, write community name is used for authentication.
- Request ID: It is used to match a response to a request.
  - SNMP assigns a unique ID to each request.
- Error status: It is used in a response to indicate the errors when the agent processes the request
  - noError, tooBig, noSuchName, badValue, readOnly, and genErr.
- Error index: Provides the information of the variables that caused the error when an error occurs.
- Variable bindings: It is composed of a variable name and value.
- Enterprise: Type of the device that generates traps.
- Agent addr: Address of the device that generates traps.
- Generic trap: It includes coldStart, warmStart, linkDown, linkup, authenticationFailure, egpNeighborLoss and enterpriseSpecific.
- Specific trap: Specific trap information of a vendor.
- Time stamp: The amount of time between the time when the SNMP entity sending this message reinitialized and the time when traps were generated, that is, the value of sysUpTime.

SNMP message

Version	Community	SNMP PDU

Get/GetNext/Set PDU

PDU type	Request ID	0	0	Variable bindings

Response PDU

PDU type	Request ID	Error status	Error index	Variable bindings

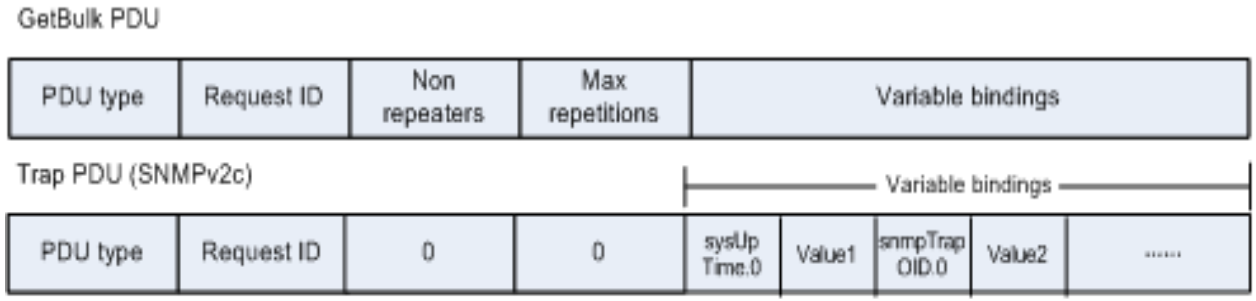
Trap PDU

PDU type	enterprise	Agent addr	Generic trap	Specific trap	Time stamp	Variable bindings

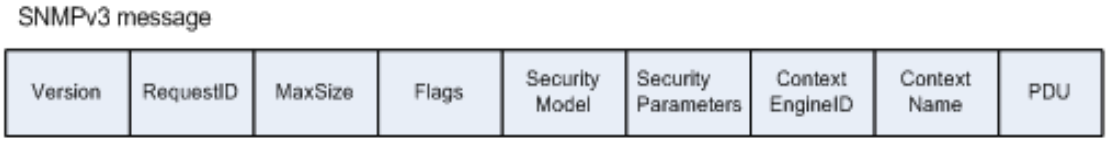
universidade de aveiro

SNMPv2c Message

- Compared with SNMPv1, GetBulk packets are added in SNMPv2c.
  - GetBulk operation corresponds to GetNext operation.
  - In a GetBulk operation, the setting of Non repeaters and Max repetitions parameters enables NMS to obtain data of many managed objects from an agent.
- In SNMPv2c, trap message format is different from that in SNMPv1.
  - SNMPv2c trap PDU adopts the format of SNMPv1 Get/GetNext/Set PDU, and sysUpTime and snmpTrapOID are used as variables in variable bindings to create a packet.



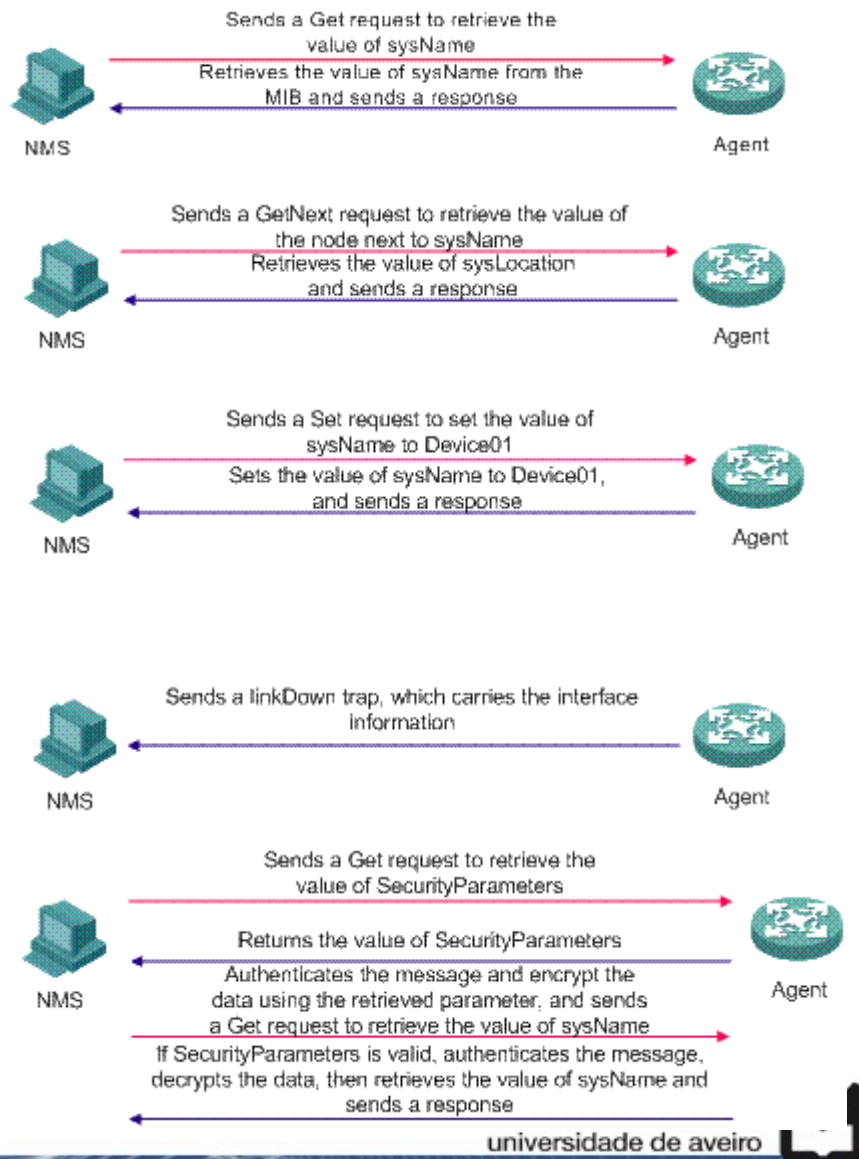
SNMPv3 Message



- SNMPv3 message format is modified, but the PDU format is the same as that in SNMPv2c.
- The entire SNMPv3 message can be authenticated, and EngineID, ContextName, and PDU are encrypted.
- RequestID, MaxSize, Flags, SecurityModel and SecurityParameters form the SNMPv3 message header.
- Fields:
  - RequestID
  - MaxSize: The maximum size of the message that the sender of the message can receive.
  - Flags: Message flag which occupies one byte. Only the lowest three bytes are valid. 0x0 indicates no authentication no privacy, 0x1 indicates authentication without privacy, 0x3 indicates authentication with privacy, and 0x4 indicates to send a report PDU.
  - SecurityModel: Message security model, in the range 0 to 3. 0 indicates any model, 1 indicates SNMPv1 security model, 2 indicates SNMPv2c security model, and 3 indicates SNMPv3 security model.
  - SecurityParameters includes the following fields:
    - AuthoritativeEngineID: Specifies the snmpEngineID of the authoritative SNMP engine involved in the exchange of the message, used for identification, authentication and encryption for an SNMP entity. This field refers to the source for a trap, response, or report, and to the destination for a Get, GetNext, GetBulk, or Set operation.
    - AuthoritativeEngineBoots: Specifies the snmpEngineBoots value at the authoritative SNMP engine involved in the exchange of the message. It indicates the number of times that this SNMP engine has initialized or reinitialized itself since its initial configuration.
    - AuthoritativeEngineTime: Specifies the snmpEngineTime value at the authoritative SNMP engine involved in the exchange of the message. It is used for time window check.
    - UserName: Specifies the user (principal) on whose behalf the message is being exchanged. Usernames configured on NMS and Agent must be the same.
    - AuthenticationParameters: A key used in authentication calculation. If no authentication is performed, this field is null.
    - PrivacyParameters: A parameter used in privacy calculation.
  - ContextEngineID: Uniquely identifies an SNMP entity. For a message received, this field decides how this message will be processed; for a message sent, this field is provided by the sender.
  - ContextName: Identifies a context. Must be unique within an SNMP entity.

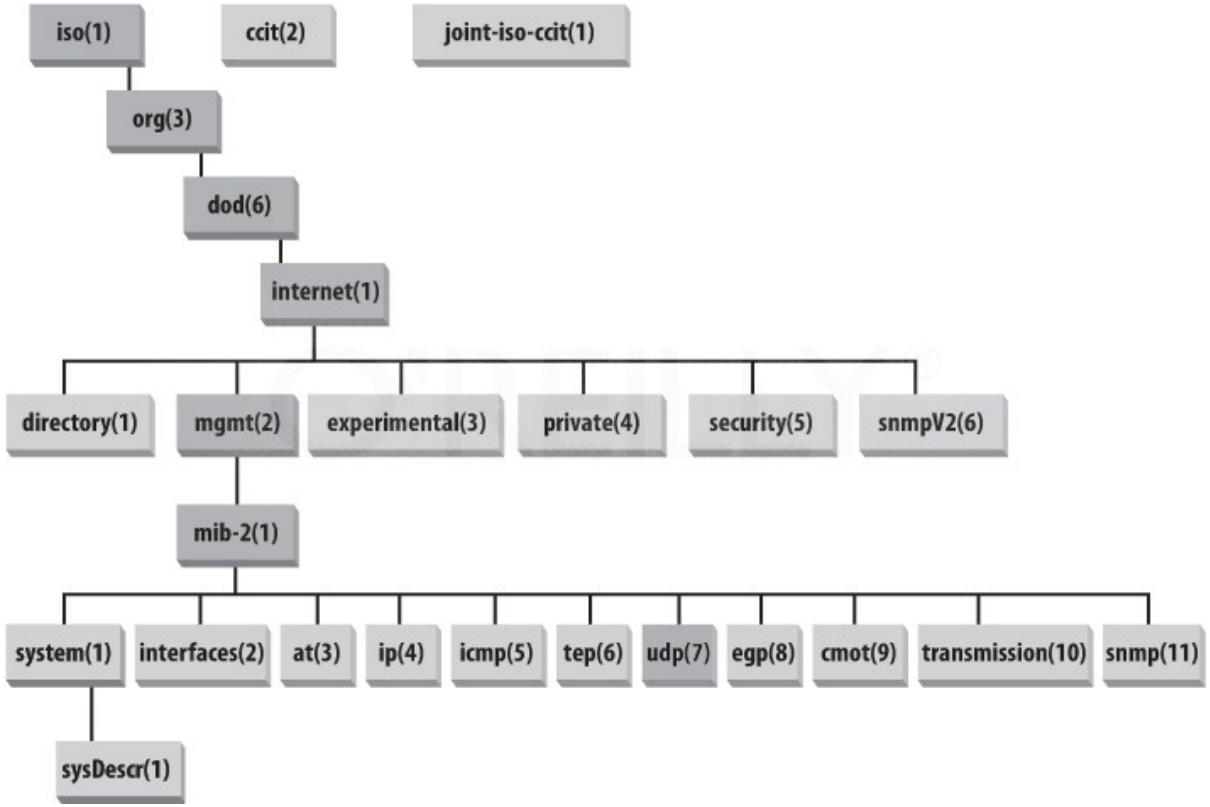
SNMP Operations

- SNMP provides the following five basic operations:
  - Get operation
    - Request sent by the NMS to the agent to retrieve one or more values from the agent.
  - GetNext operation
    - Request sent by the NMS to retrieve the value of the next OID in the tree.
  - Set operation
    - Request sent by the NMS to the agent to set one or more values of the agent.
  - Response operation
    - Response sent by the agent to the NMS.
  - Trap operation
    - Unsolicited response sent by the agent to notify the NMS of the events occurred.
- In SNMPv3 get operations are performed using authentication and encryption.



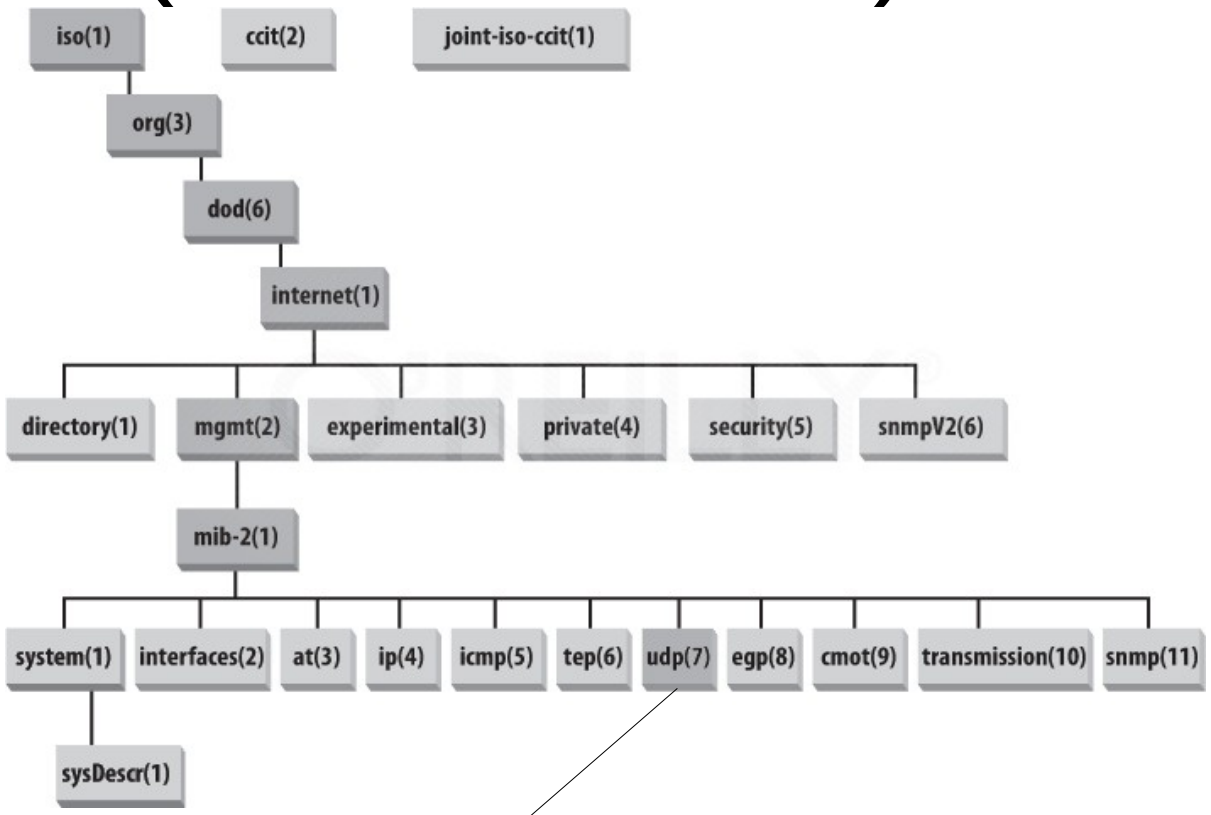
MIB Modules and Object Identifiers

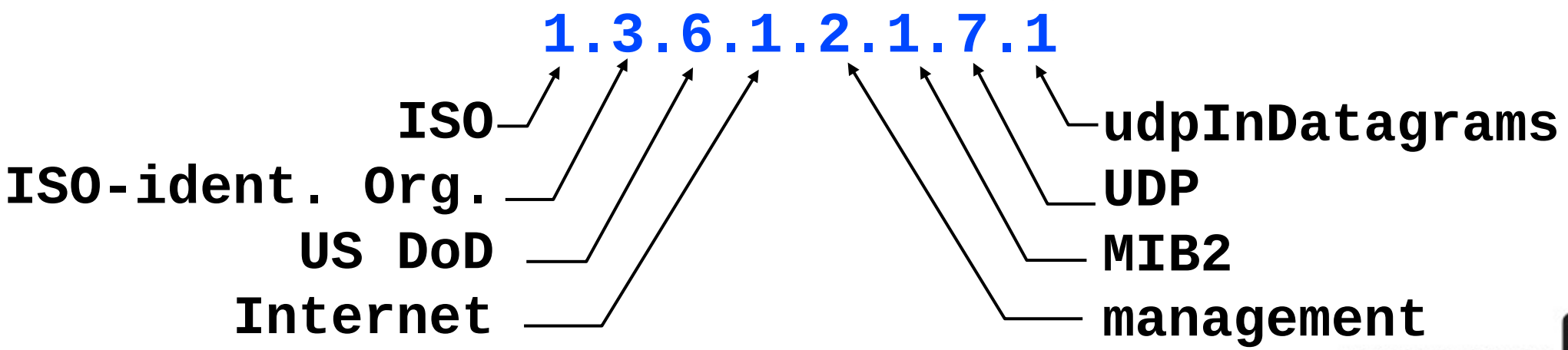
- An SNMP MIB module is a specification of management information on a device
- The SMI represents the MIB database structure in a tree form with conceptual tables, where each managed resource is represented by an object
- Object Identifiers (OIDs) uniquely identify or name MIB variables in the tree
  - Ordered sequence of nonnegative integers written left to right, containing at least two elements
  - For easier human interaction, string-valued names also identify the OIDs
    - MIB-II (object ID 1.3.6.1.2.1)
    - Cisco private MIB (object ID 1.3.6.1.4.1.9)
- The MIB tree is extensible with new standard MIB modules or by experimental and private branches
  - Vendors can define their own private branches to include instances of their own products



SNMP Names (numbers/OID)

- To nominate all possible objects (protocols, data, etc.) it is used an ISO Object Identifier (OID) tree:
  - Hierarchic nomenclature of objects
  - Each leaf of the tree has a name and number





universidade de aveiro

SNMP MIBs

- Management Information Base (MIB): set of managed objects, used to define information from equipments, and created by the manufacturer
- Example: UDP module

Object ID	Name	Type	Comments
1.3.6.1.2.1.7.1	UDPInDatagrams	Counter32	Number of UDP datagrams delivered to users.
1.3.6.1.2.1.7.2	UDPNoPorts	Counter32	Number of received UDP datagrams for which there was no application at the destination port.
1.3.6.1.2.1.7.3	UDPInErrors	Counter32	The number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.
1.3.6.1.2.1.7.4	UDPOutDatagrams	Counter32	The total number of UDP datagrams sent from this entity.

SMI: Data language definition

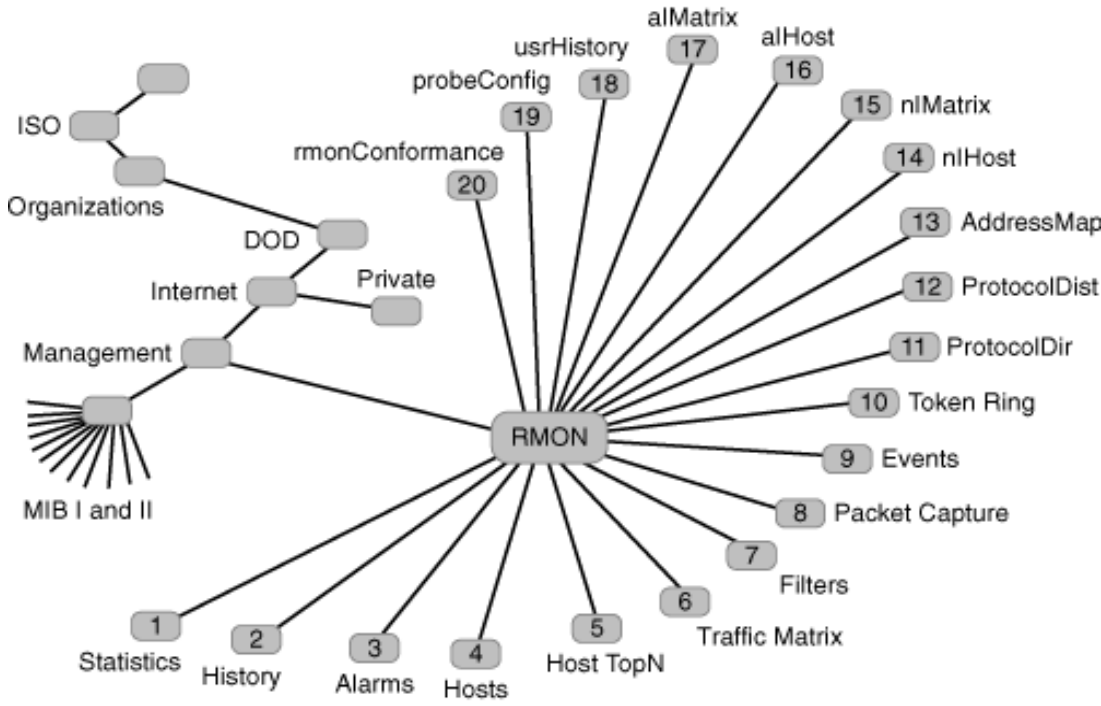
- Well-defined sintax and semantics of management information
  - Type of basic data
    - INTEGER, Integer32, Unsigned32, OCTET, STRING, OBJECT IDENTIFIED, IPaddress, Counter32, Counter64, Guage32, Tie Ticks,Opaque...
  - Type of object
    - Type of data, status, semantic of the managed object
  - Module identification
    - Collection of objects inter-related in the MIB

SMI: Data Types for Scalars

	SMIv1	SMIv2
SIMPLE TYPES:	INTEGER	INTEGER
	OCTET STRING	OCTET STRING
	OBJECT IDENTIFIER	OBJECT IDENTIFIER
	-	Integer32
APPLICATION-WIDE TYPES:	-	Unsigned32
	Gauge	Gauge32
	Counter	Counter32
	-	Counter64
	TimeTicks	TimeTicks
	IpAddress	IpAddress
	Opaque	Opaque
	NetworkAddress	-
PSEUDO TYPES:	-	BITS

RMON

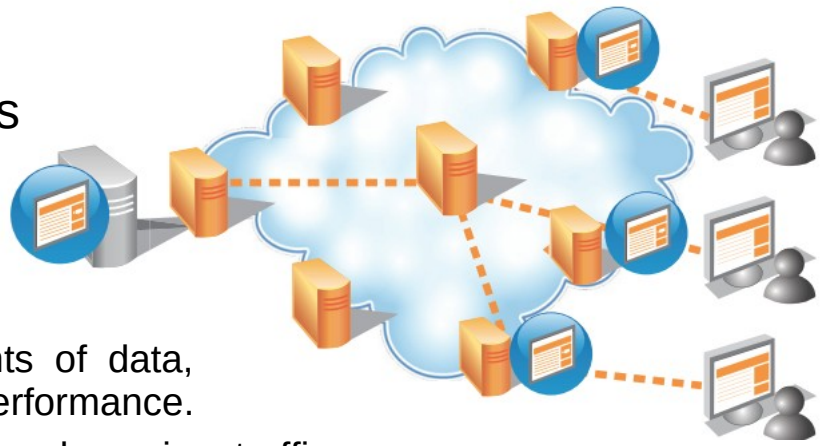
- RMON is a set of standardized MIB variables that monitor networks
  - All previously defined MIBs monitored only nodes
- RMON has 9 groups
  - Statistics, History, Alarm, Host, HostTopN, Matrix, Filter, Packet Capture, and Event
- The term RMON now is often used to refer to the concept of remote monitoring and to the entire series of RMON MIB extensions
- The main RMON MIB extensions are:
  - RMON 1 and RMON 2 MIBs - Remote Monitoring MIB versions 1 and 2
  - DSMON MIB - Remote Monitoring MIB Extensions for Differentiated Services
  - SMON MIB - Remote Network Monitoring MIB Extensions for Switched Networks
  - APM MIB - Application Performance Measurement MIB



Content Delivery/Distribution Networks (CDN)

universidade de aveiro
deti.ua.pt

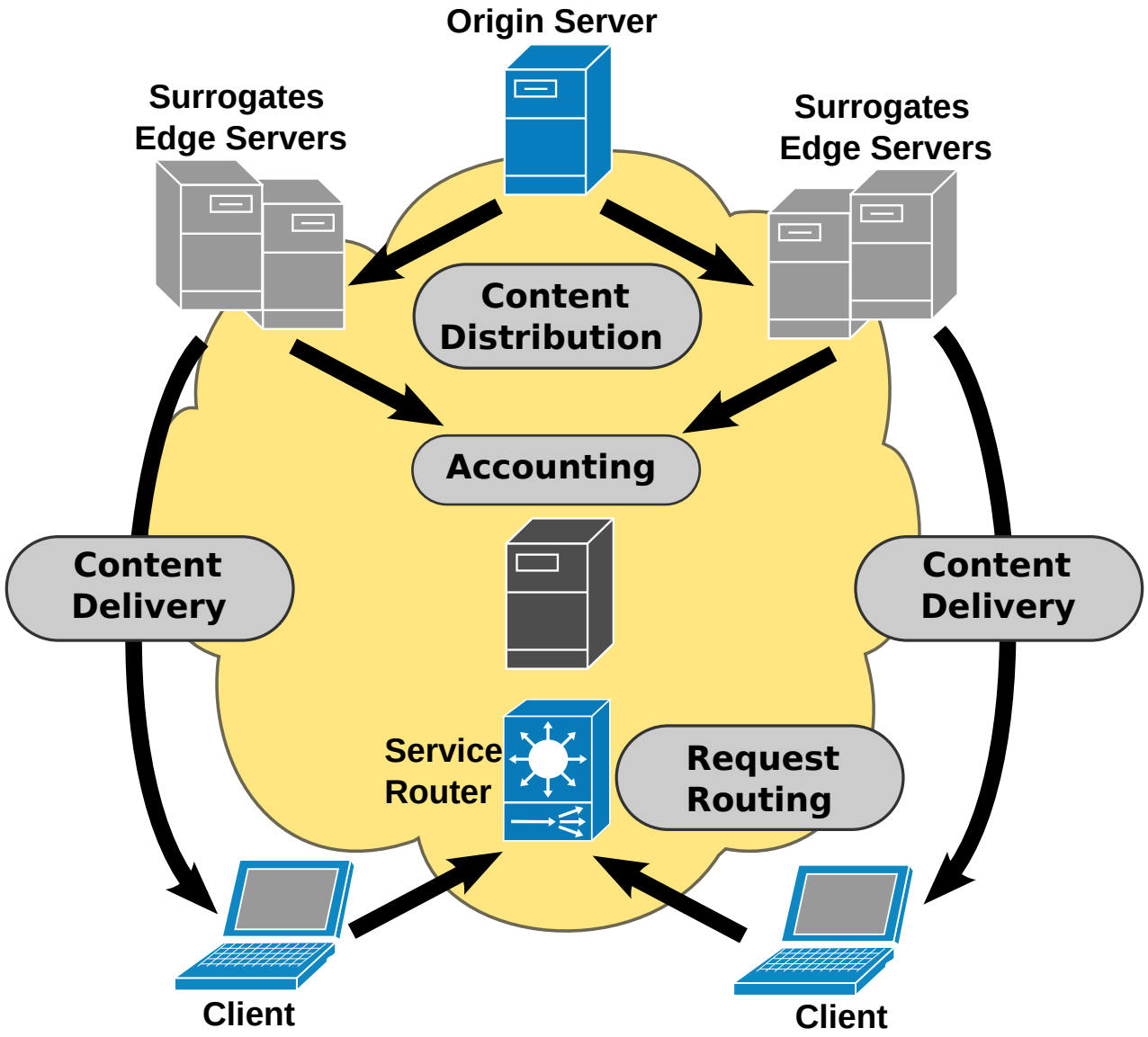
Content Delivery/Distribution Network (CDN)



- Consists of geographically distributed network of servers around the globe.
- Improvement goals:
  - Scalability
    - Ability to expand in order to handle new and large amounts of data, users and transactions without any significant decline in performance.
    - Dynamically allocation of resources to address flash crowds and varying traffic.
    - Acts as a shock absorber for traffic by automatically providing capacity-on-demand to meet the requirements
    - Avoids costly over-provisioning of resources and provides high performance to every user.
  - Security
    - Provides protection of content against unauthorized access and modification, distributed denial-of-service (DdoS) attacks, viruses, and other unwanted intrusions.
    - Eliminates the need for costly hardware and dedicated component to protect content and transactions.
  - Reliability, Responsiveness and Performance
    - Improves client access to content through delivering it from multiple locations.
    - The reliability and performance is affected by the distributed content location and routing mechanism, as well by data replication and caching strategies.
- Evolution
  - First Generation: Focused on Static or Dynamic Web Document.
  - Second Generation: Focused on Video-on-Demand (VoD), audio and video streaming.

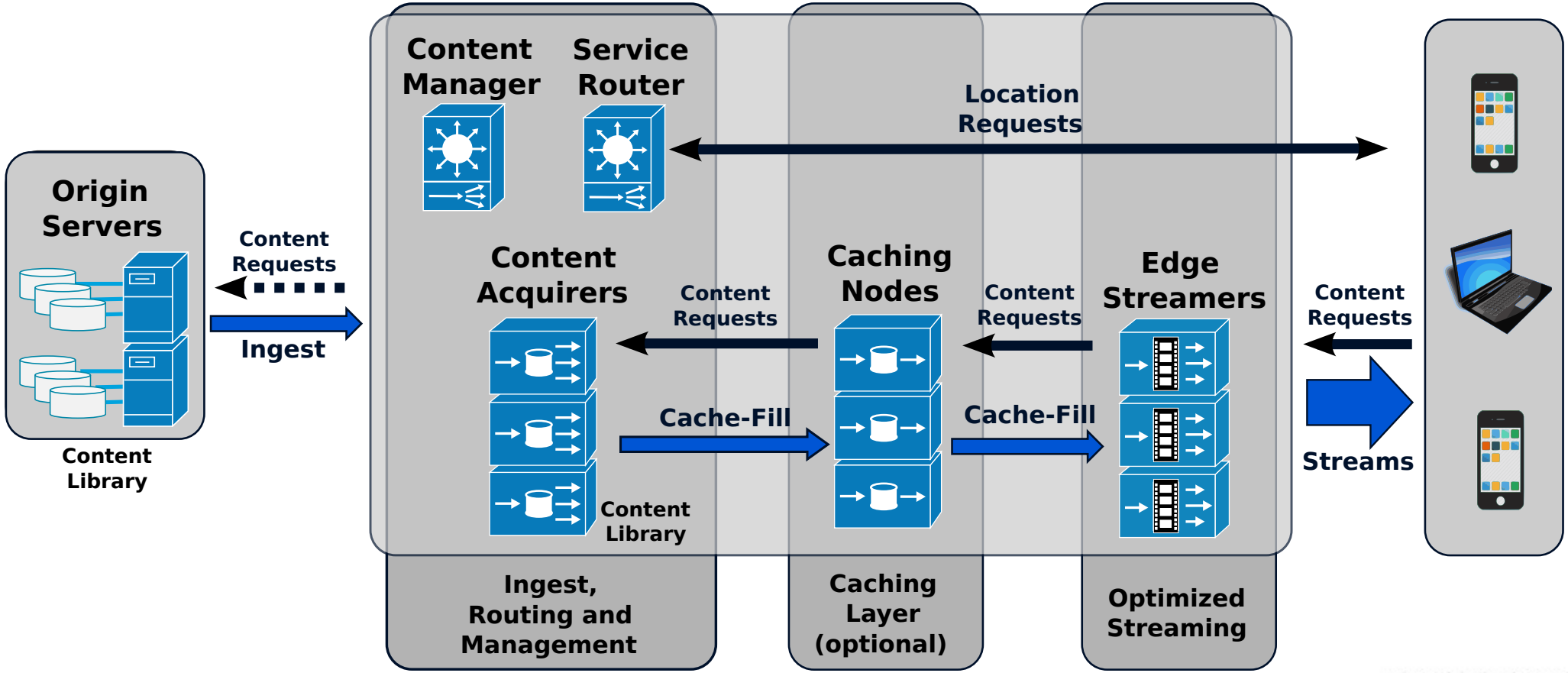
CDN Components

- **Content Delivery** Infrastructure
  - Delivering content to clients from Surrogates (Edge Servers).
- **Request Routing** Infrastructure
  - Steering or directing content request from a client to a suitable Surrogate.
- **Content Distribution** Infrastructure
  - Moving or replicating content from content source (origin server, content provider) to surrogates.
- **Accounting** Infrastructure
  - Logging and reporting of distribution and delivery activities.



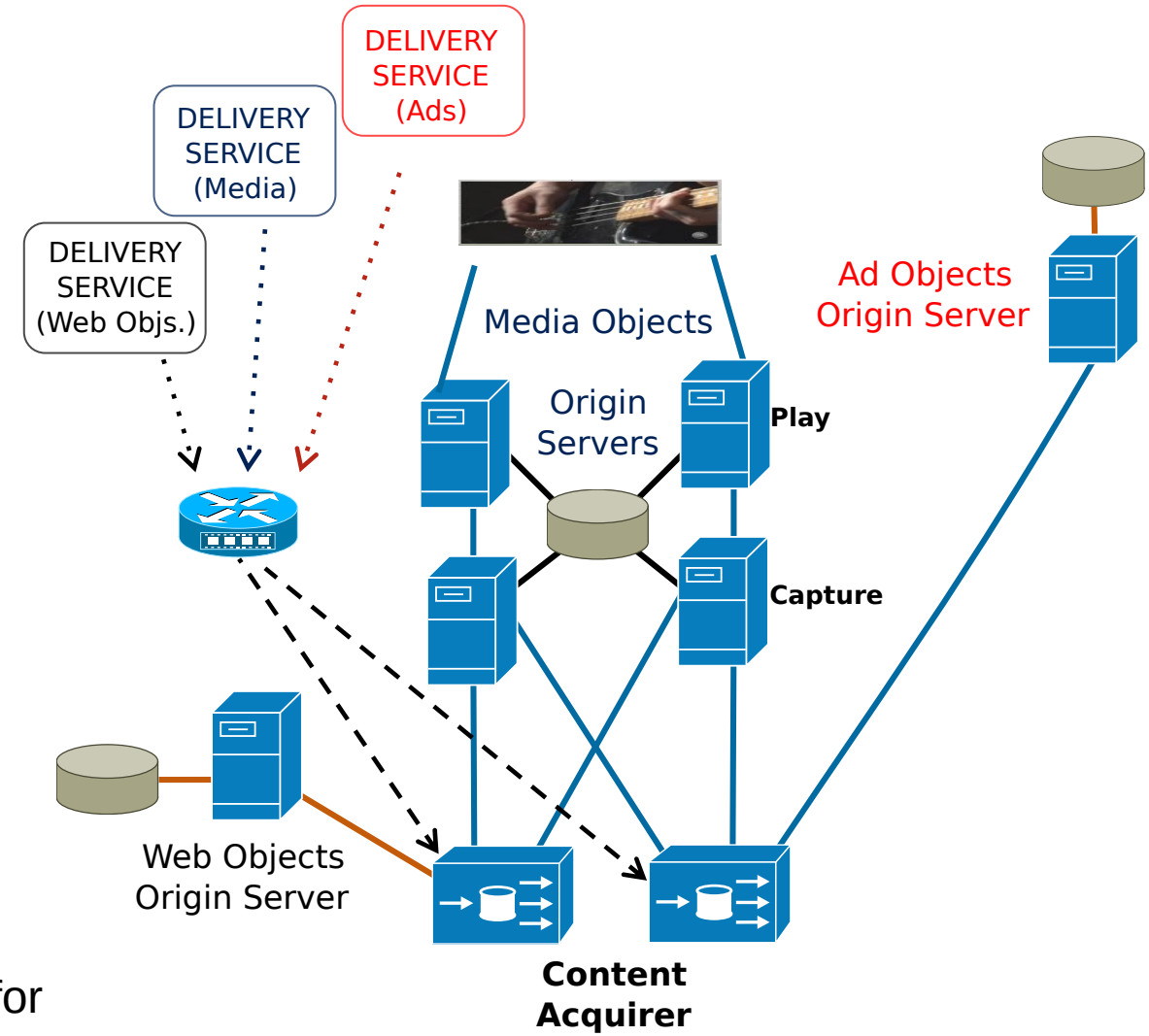
CDN for VoD and Streaming

- VoD and Streaming have QoS strict requirements.
- Surrogates become:
  - Content Acquirers
  - Cache Nodes
  - Edge Streamers



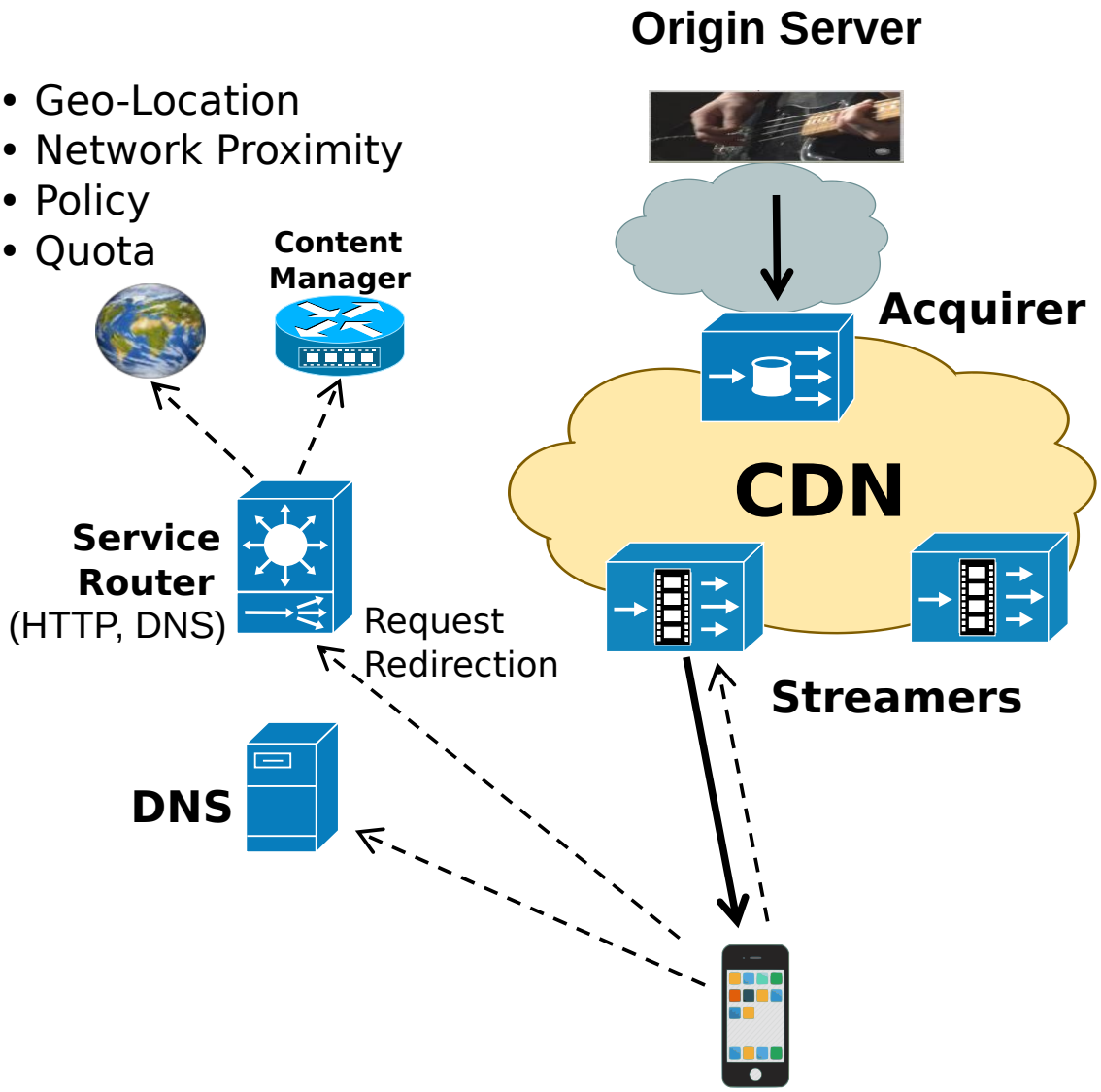
Origin Servers

- Origin Servers (OS)
  - Organized Media on Storage.
  - Authorize Acquirers.
  - Package Content.
- Ingest must be flexible, resilient and secure.
- CDN can ingest from multiple Origin Servers.
  - Local or Remote locations.
- Origins can be replicated.
  - Locally (load balancing).
  - Remotely (disaster recovery).
- Origins can have structure.
  - Security.
  - Capture/Recording/Playout separation for better scalability.

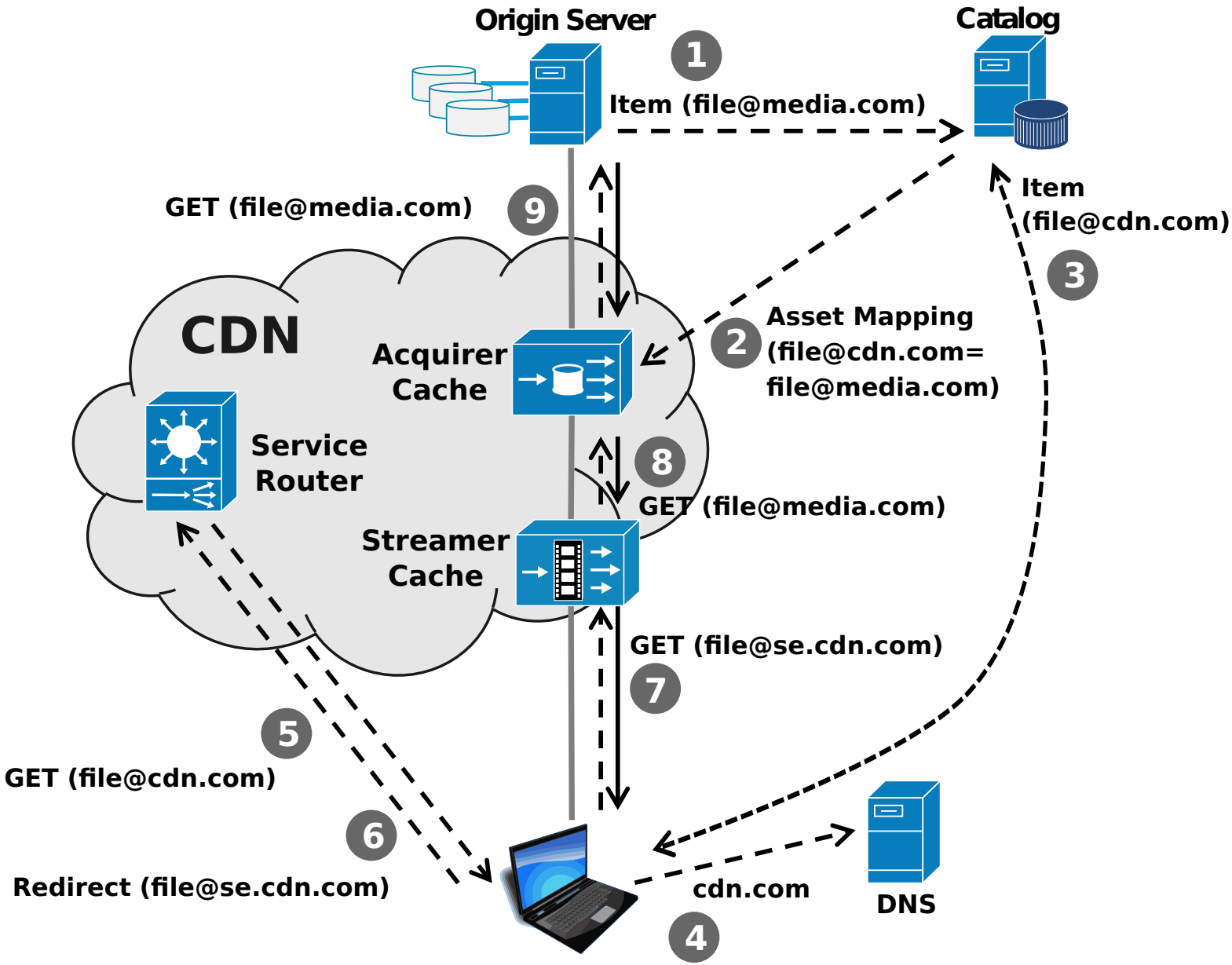


Routing Service

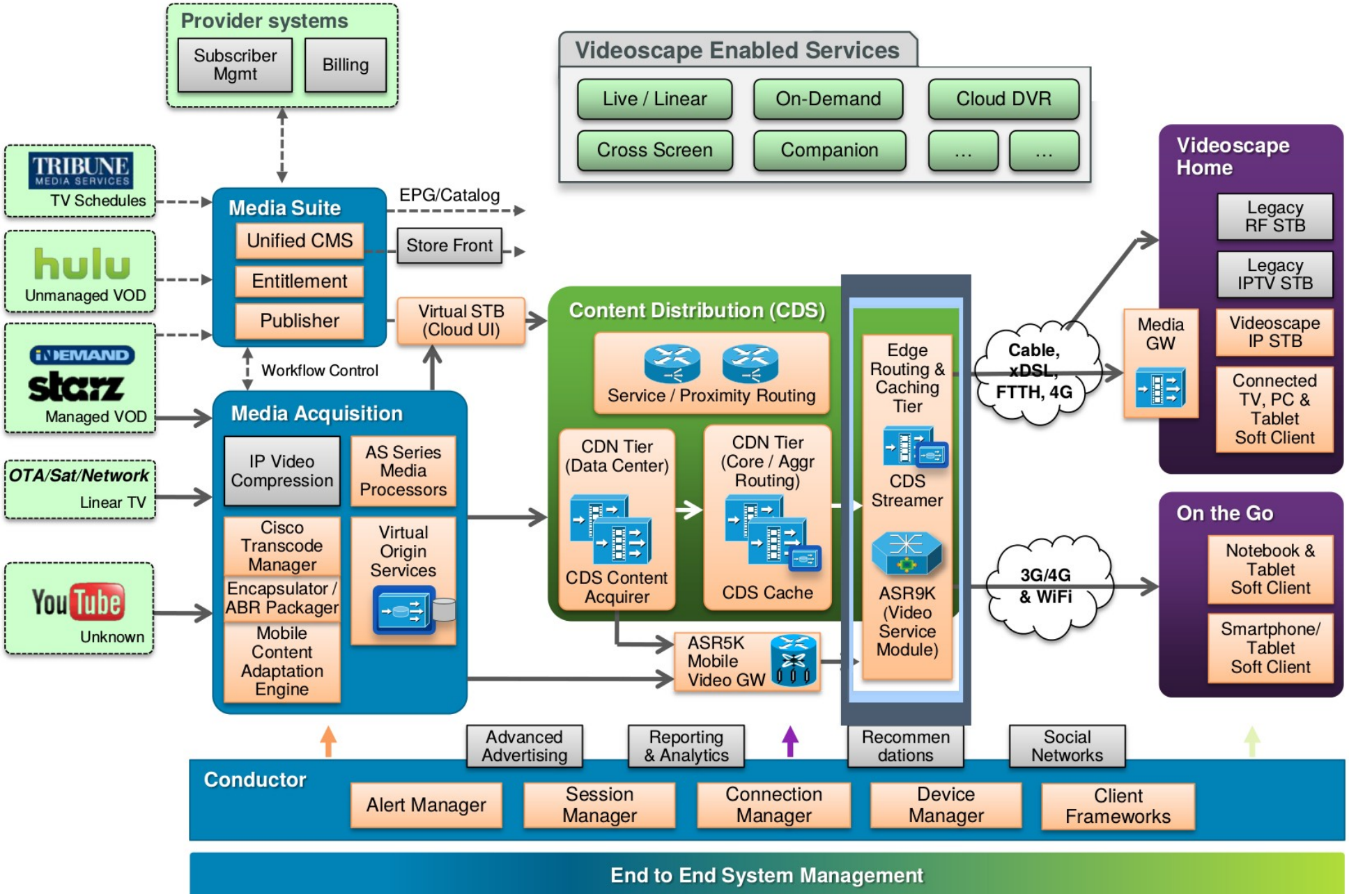
- Request Redirection model.
  - Service Router is the Authoritative DNS for “Delivery Service”.
- HTTP-based 30x redirection.
  - Service Router resolves domain name to its own IP address.
  - Service Router then uses HTTP 302/307 redirection to a Streamer.
- DNS-based redirection.
  - Service Router resolves domain to IP address of Streamer.
- Service Router Criteria.
  - Based on Client IP Address.
    - Determines Geo-Location, Network Proximity, Policy, and/or Quota.



CDN Caching


Origin Server
Catalog
1
Item (file@media.com)
GET (file@media.com)
9
Item
(file@cdn.com)
3
CDN
Acquirer
Cache
2
Asset Mapping
(file@cdn.com=
file@media.com)
Service
Router
8
GET (file@media.com)
Streamer
Cache
GET (file@se.cdn.com)
7
5
GET (file@cdn.com)
6
Redirect (file@se.cdn.com)
cdn.com
4
DNS

A CDN Architecture



Source: Cisco,Guillaume Gottardi, Next Generation Service Edge Architectures, CiscoLive London 2012

Alternative CDN Content Distribution

- Content Classification
  - Content manager assesses content popularity.
  - Content manager drives content distribution.
    - Popular Content is pre-positioned on the edge.
    - Less Popular content is dynamically cached from central site.
    - Unpopular content is off-loaded directly to Origin.
  - Content popularity may change!
- Peer to peer
  - Distributed Hash Table model.
  - Content can be cached anywhere.
  - Appropriate in fully meshed topologies.