Optical Networks

Marco Polverini

Network Infrastructures
"Sapienza" University

A.A. 2023/24

Outline

Introduction to Optical Networks

Control and Management

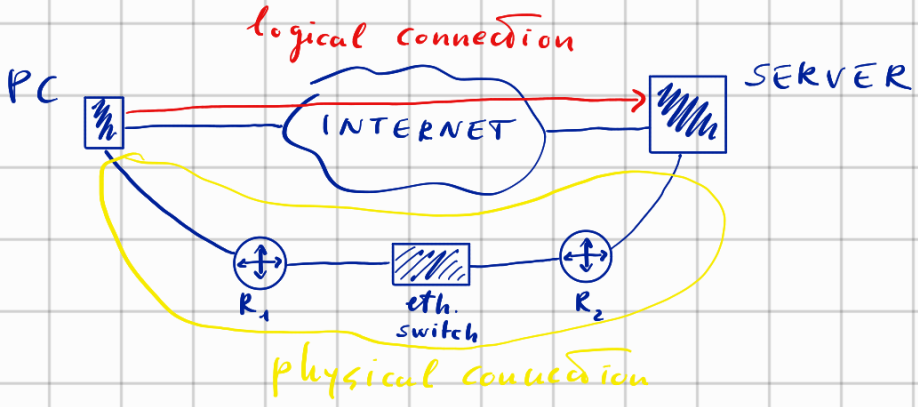
Introduction to Optical Networks

When we want to know a path that a packet does it's possible to use a specific tool called traceroute.

The connection between 2 distant hosts it's just a logical connection.

How the logical connection actually works it's a physical connection matter.

i.e.

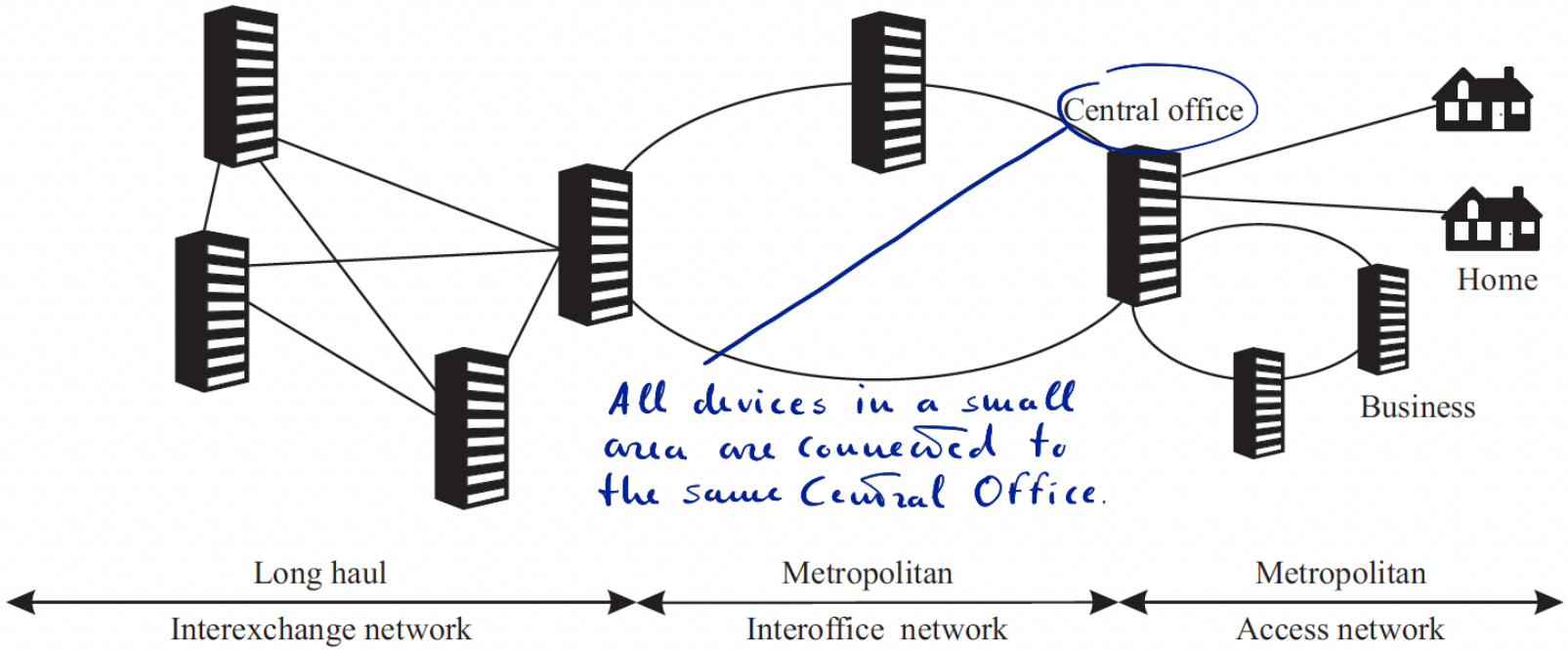


Telecommunications Network Architecture

- Transport Networks (TNs) are public infrastructures operated by service providers named *carriers*
- Carriers provide a variety of services:
  - telephone and leased line services
  - interconnect Internet Service Providers
  - provide bulk bandwidth to other carriers

The TN provide a physical interconnection between 2 hosts the required it.

(the scheme does not include mobile connectivity)



Different parts of a public network

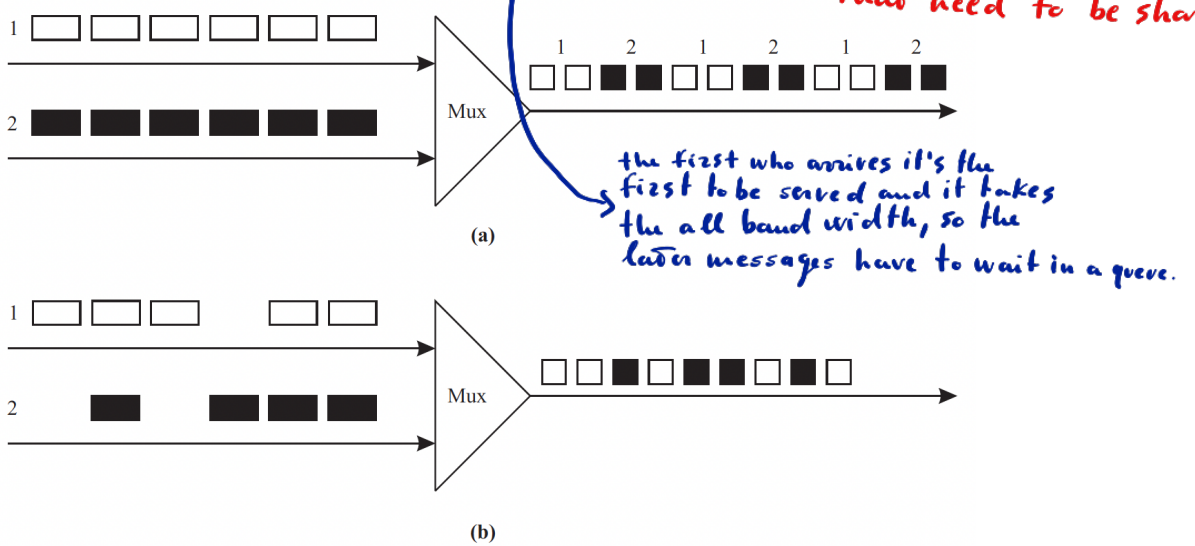
- The network can be broken up into:
  - metro network: is the part of the network that lies within a large city or a region
    - metro access network: extends from a central office out to individual businesses or homes
    - interoffice network: connects groups of central offices within a city or region
  - long-haul: network interconnects cities or different regions
- Different parts of the network may be owned and operated by different carriers
- The nodes in the network are central offices, sometimes also called points of presence (POPs)
- Links between the nodes consist of fiber pairs and, in many cases, multiple fiber pairs
  - links in the long-haul network tend to be very expensive to construct
- Two topologies are used: ring and mesh

Services, Circuit Switching, and Packet Switching

- Service can be:
  - Connection oriented: sender and receiver connect each other before communication happen
  - Connectionless: source sends messages to the receiver whenever it has something to send
- Core devices work according to one of the following switching paradigm:
  - circuit switching: static multiplexing (connection oriented service)
  - packet switching: statistical multiplexing

fixed assigned resources and deterministic performances.

multiplexing allows us to divide a channel that need to be shared.



the first who arrives it's the first to be served and it takes the all band width, so the latter messages have to wait in a queue.

Put in the correct order the different phases of the circuit setup procedure

✔ La combinazione corretta era:

1. the source node ask to the network the creation of a new circuit
2. the network compute a path from the source to the destination node
3. the network checks if, in each link of the path, there is an available sub-channel
4. new rules are inserted in the switching table of the network nodes
5. the clients are notified that the circuit is up
6. the communication happens
7. the network monitors the circuit
8. the network tears down the circuit

It means that it's mandatory to check the connection in order to always satisfy the user requests.

Optical Networks

- Optical Networks (ONs) can deliver bandwidth in a flexible manner where and when needed
- Optical fiber
  - offers much higher bandwidth than copper cables
  - is less susceptible to various kinds of electromagnetic interference and other undesirable effects
- Two generations of optical networks
  - First generation
    - optics essentially used for transmission and simply to provide capacity
    - switching and other intelligent network functions handled by electronics
  - Second generation
    - routing, switching, and intelligence in the optical layer

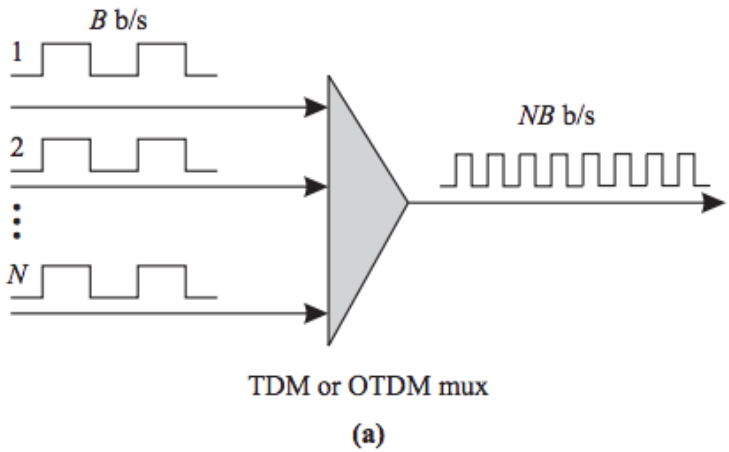
The transport network is usually diveded in 2 domain types:

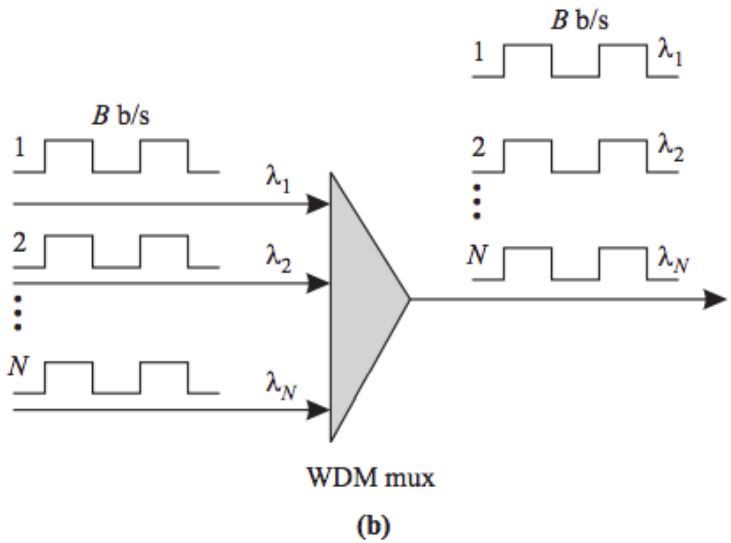
- electronic domain
- optical domain

Multiplexing Techniques

- Two ways of increasing the transmission capacity on a fiber:
  - Time Division Multiplexing (TDM) (1st generation of optical nets)
    - increase the bit rate (requires higher-speed electronics)
    - many lower-speed data streams are multiplexed into a higher-speed stream
  - Wavelength Division Multiplexing (WDM) (it's like a Frequency Division Multiplexing)
    - transmit data simultaneously at multiple carrier wavelengths over a fiber
    - virtual fibers

it requires to work in the electronic domain



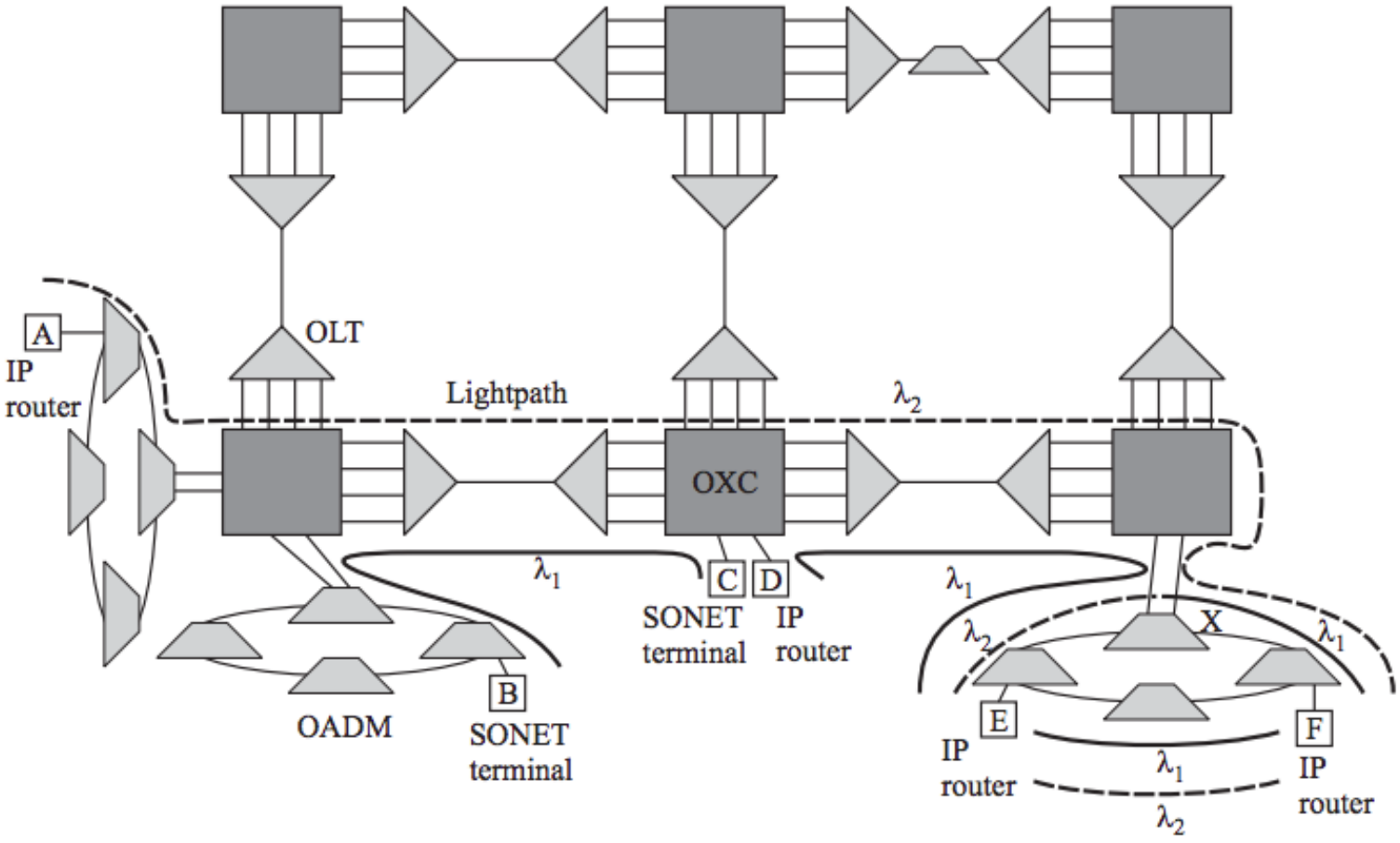


Second-Generation Optical Networks

- Also known as wavelength routed networks
- Main idea
  - incorporate some of the switching and routing functions into the optical part of the network
- The network provides lightpaths to its users
- Lightpaths are optical connections
  - carried end to end from a source node to a destination node
  - over a wavelength on each intermediate link
- At intermediate nodes the lightpaths are switched from one link to another link
- Lightpaths may be converted from one wavelength to another wavelength

Every user has its own color in order to be uniquely identified.

Second-Generation Optical Networks

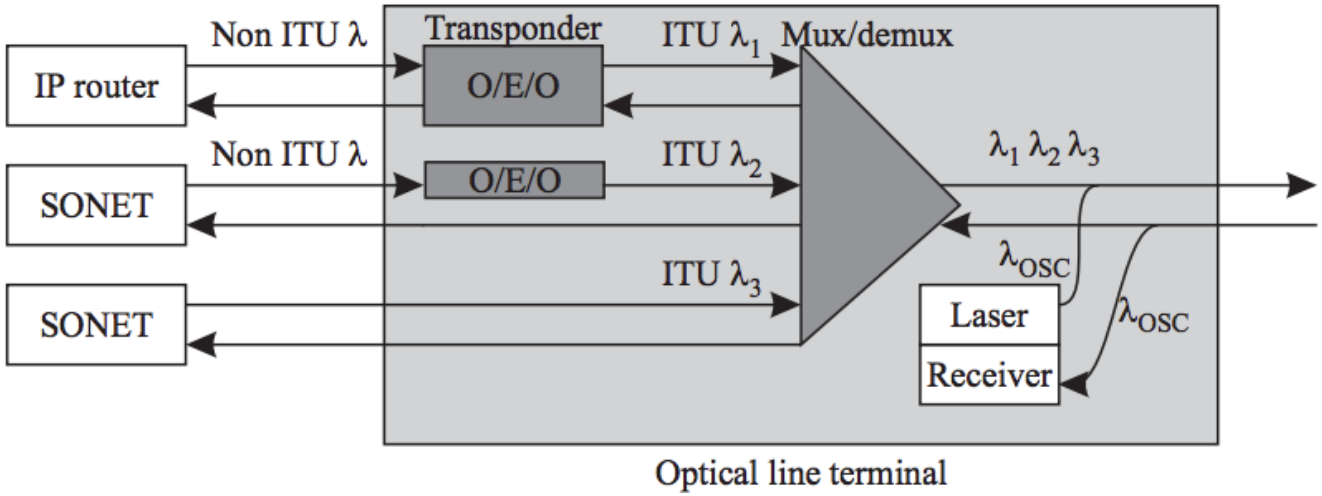


Optical Line Terminals

- OLTs are used at either end of a point-to-point link to multiplex and demultiplex wavelengths
- It is composed by three functional elements:
  - Transponders (the most expensive device into OLT)
  - Wavelength multiplexers
  - Optical amplifiers
- Transponder (a.k.a. optical-to-electrical-to-optical, O/E/O) adapts the signal coming in from a client of the optical network, and vice versa
  - converts the signal into a wavelength that is suited for use inside the optical network (from 1.3 μm to 1.55 μm)
  - adds OTN overhead (OPU, ODU, OTU, FEC, etc.)
  - monitors the bit error rate of the signal at the ingress and egress points in the network
- OLT also terminates an optical supervisory channel (OSC)

Optical Line Terminals

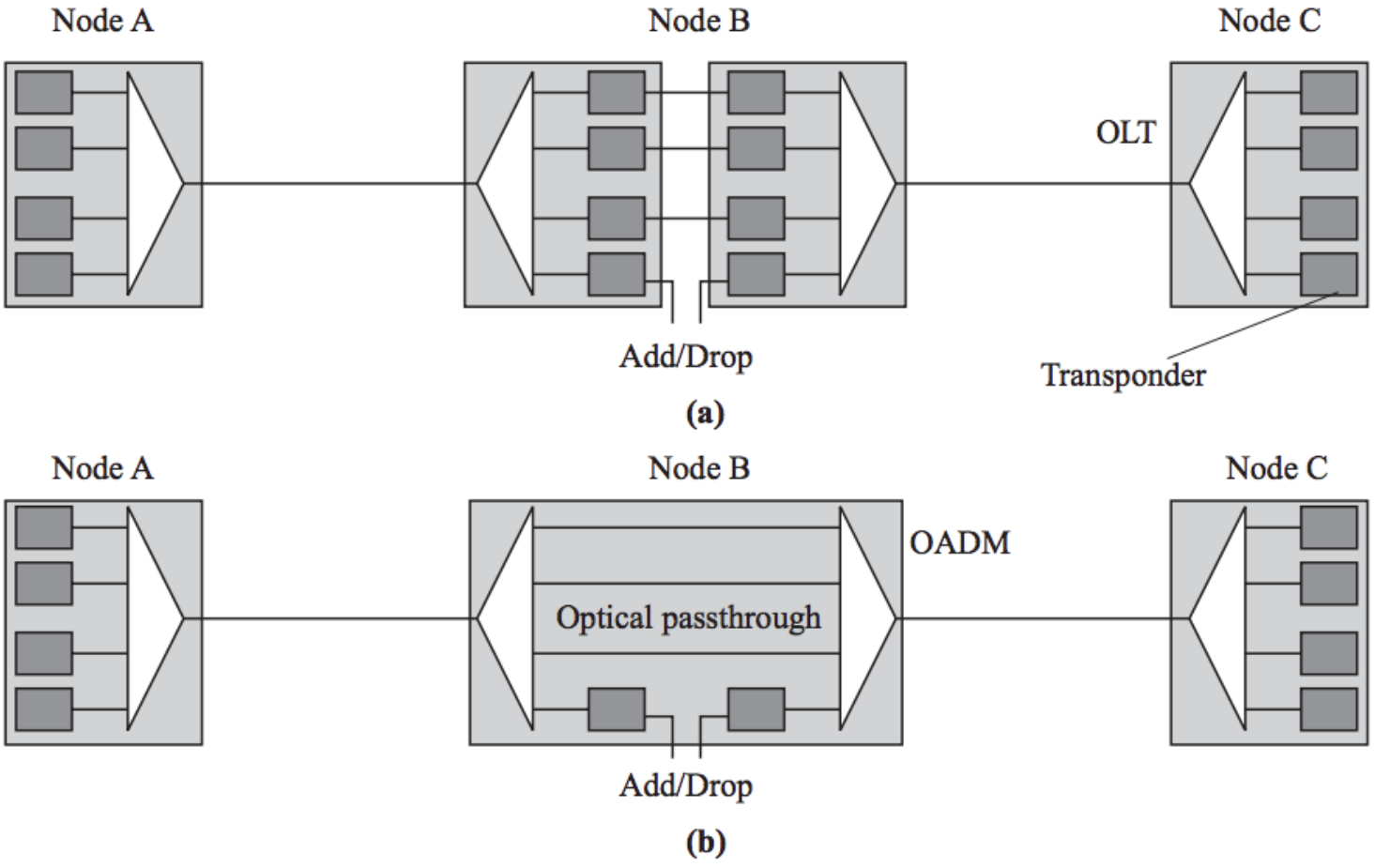
- Transponders typically constitute the bulk of the cost, footprint, and power consumption in an OLT
- Therefore, reducing the number of transponders helps minimize both the cost and the size of the equipment deployed



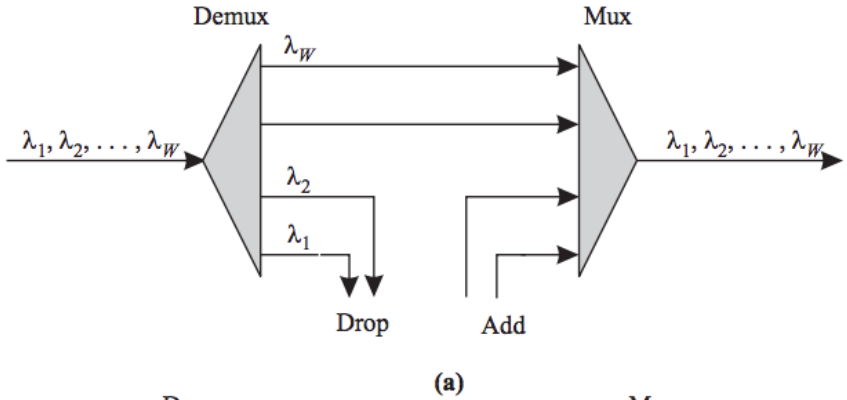
Usually there is a dedicated transponder to supervise the various colors.

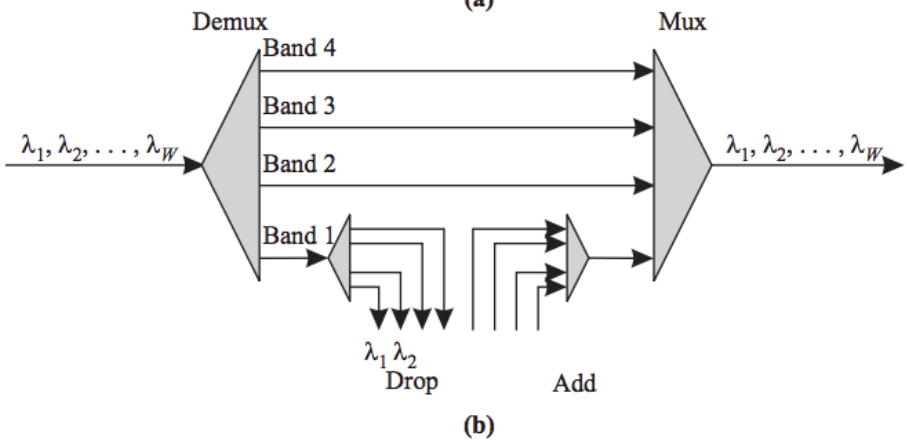
Optical Add/Drop Multiplexers

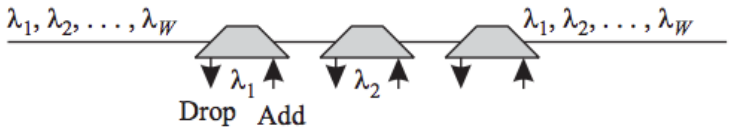
- Optical add/drop multiplexers (OADMs) provide a cost-effective means for handling passthrough traffic in both metro and long-haul networks

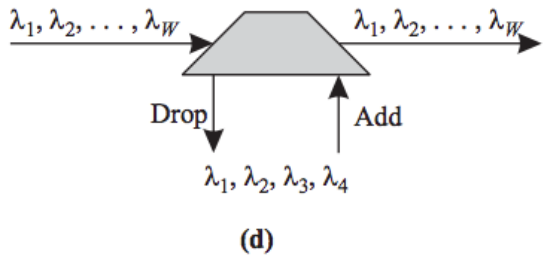


OADM Architectures



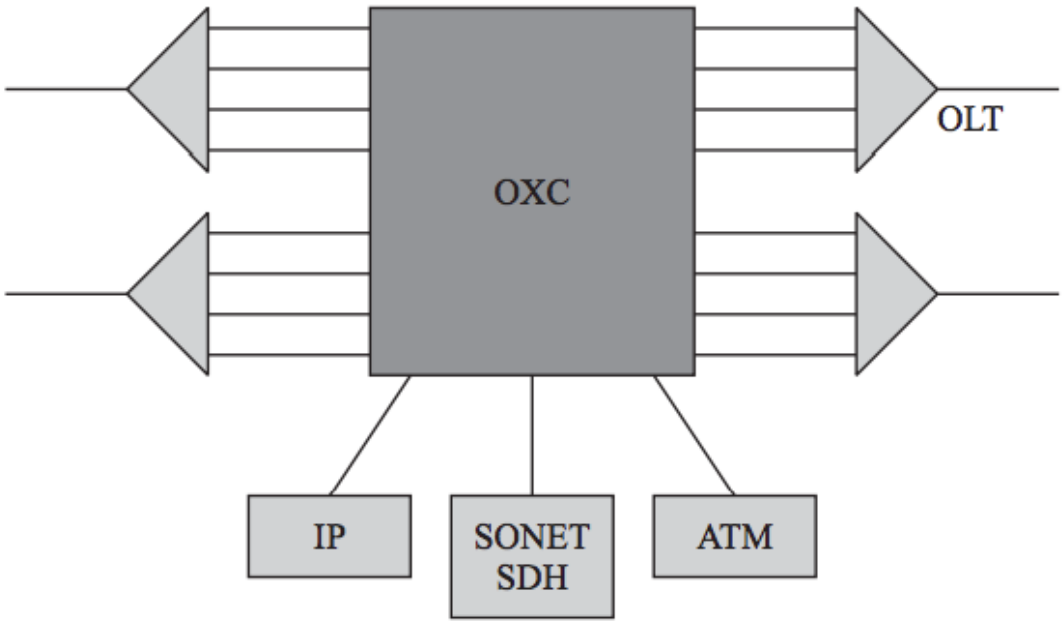






Optical Crossconnects

- OXC enables reconfigurable optical networks, where lightpaths can be set up and taken down as needed
- OXCs allow to handle complex topologies and large number of wavelengths

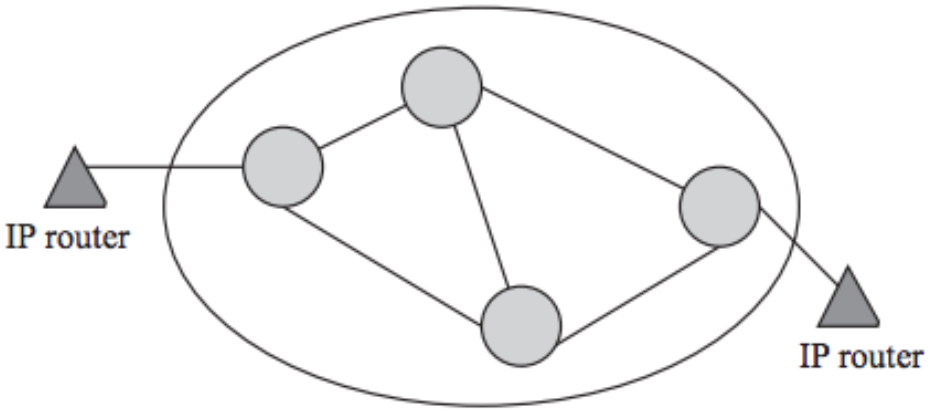


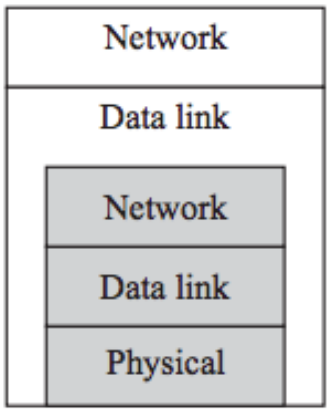
Optical Crossconnects

- An OXC provides several key functions in a large network:
  - service provisioning
    - provisioning of lightpaths in a large network in an automated manner (no manual patch panel connections)
    - reconfigure lightpaths to respond to traffic changes
  - protection
    - detect failures and rapidly reroute lightpaths around the failure
  - bit rate transparency
    - switch signals with arbitrary bit rates and frame formats
  - wavelength conversion
    - change the wavelength of an incoming signal before transmitting it
  - multiplexing and grooming
    - multiplexing and grooming capabilities to switch traffic internally at much finer granularities
    - this time division multiplexing has to be done in the electrical domain

The Optical Layer

- Network architectures can be organized by means of the ISO/OSI model
- A more realistic layered model for today's networks would employ multiple protocol stacks residing one on top of the other



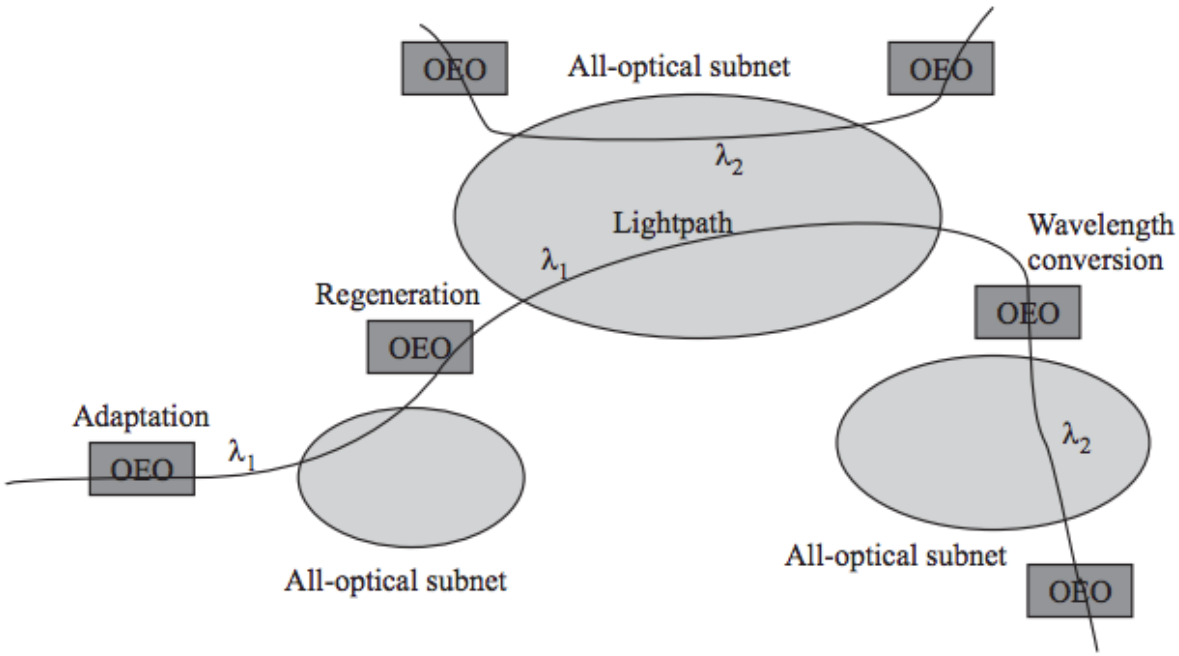


Transparency and All-Optical Networks

- Lightpaths are service transparent
  - once the lightpath is set up, it can accommodate different types of services
    - e.g., the telephone network had this property (a channel can be used to transfer voice, data, fax, etc.)
- Advantages:
  - data is carried from its source to its destination in optical form
  - no optical-to-electrical conversions along the way
- Hard to realize:
  - analog signals require higher SNR with respect to digital ones
- Optical networks almost always include a fair amount of electronics

Transparency and All-Optical Networks

- Electronics plays a crucial role in performing the intelligent control and management functions
- Electronic is required
  - at the edge of the network
    - to adapt the signals entering the optical domain
  - in the core of the network
    - for regeneration and wavelength conversion



Transparency and All-Optical Networks

- Electronic regenerators reduce the transparency of the network
- Three types of electronic regeneration techniques for digital data
  - 1R: regeneration (can be seen as an Optical Amplifier)
    - PRO: supports analog signals
    - CONS: poor performance
  - 2R - regeneration with reshaping
    - PRO: offers transparency to bit rates
    - CONS: limits the number of regeneration steps allowed due to the accumulated jitter
  - 3R - regeneration with retiming and reshaping
    - PRO: produces a "fresh" copy of the signal
    - CONS: it eliminates transparency to bit rates and the framing protocols

Network Evolution

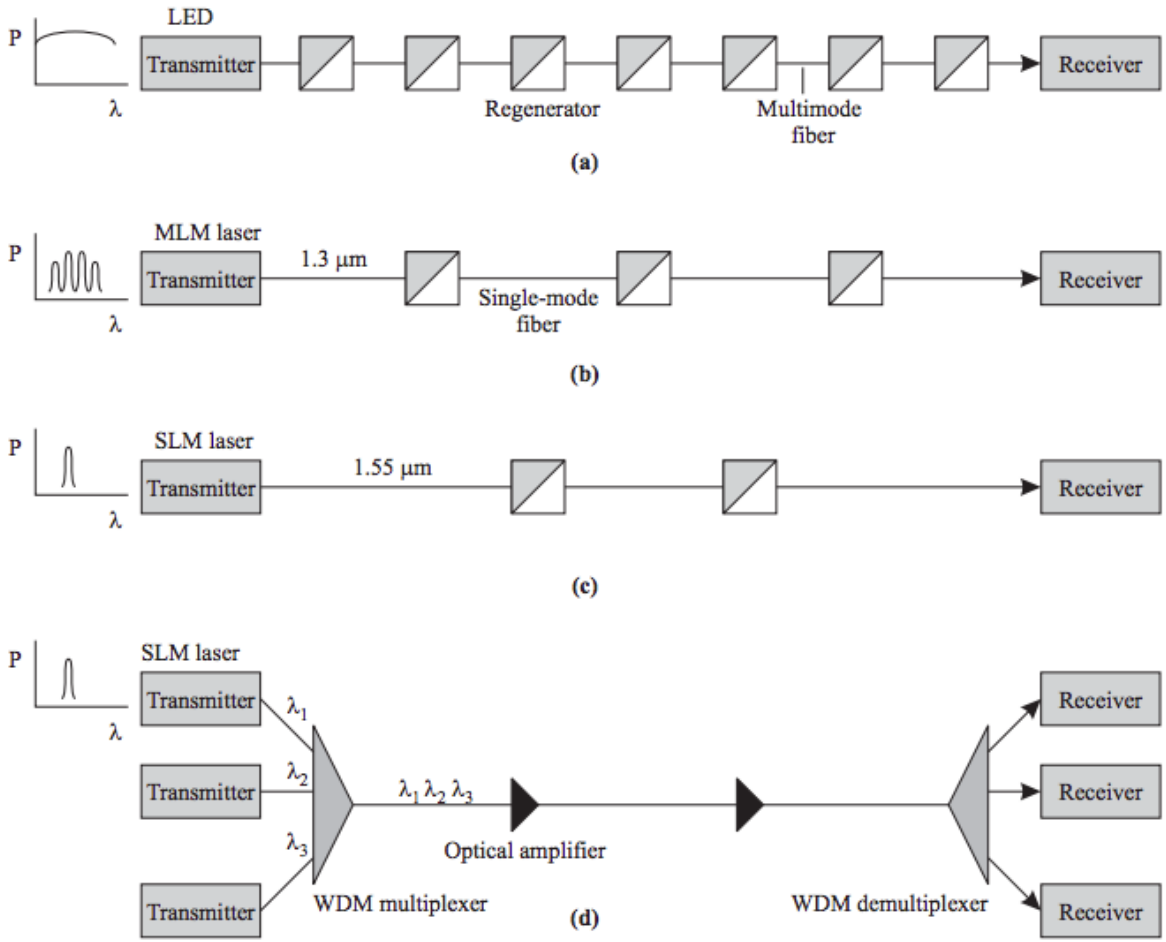
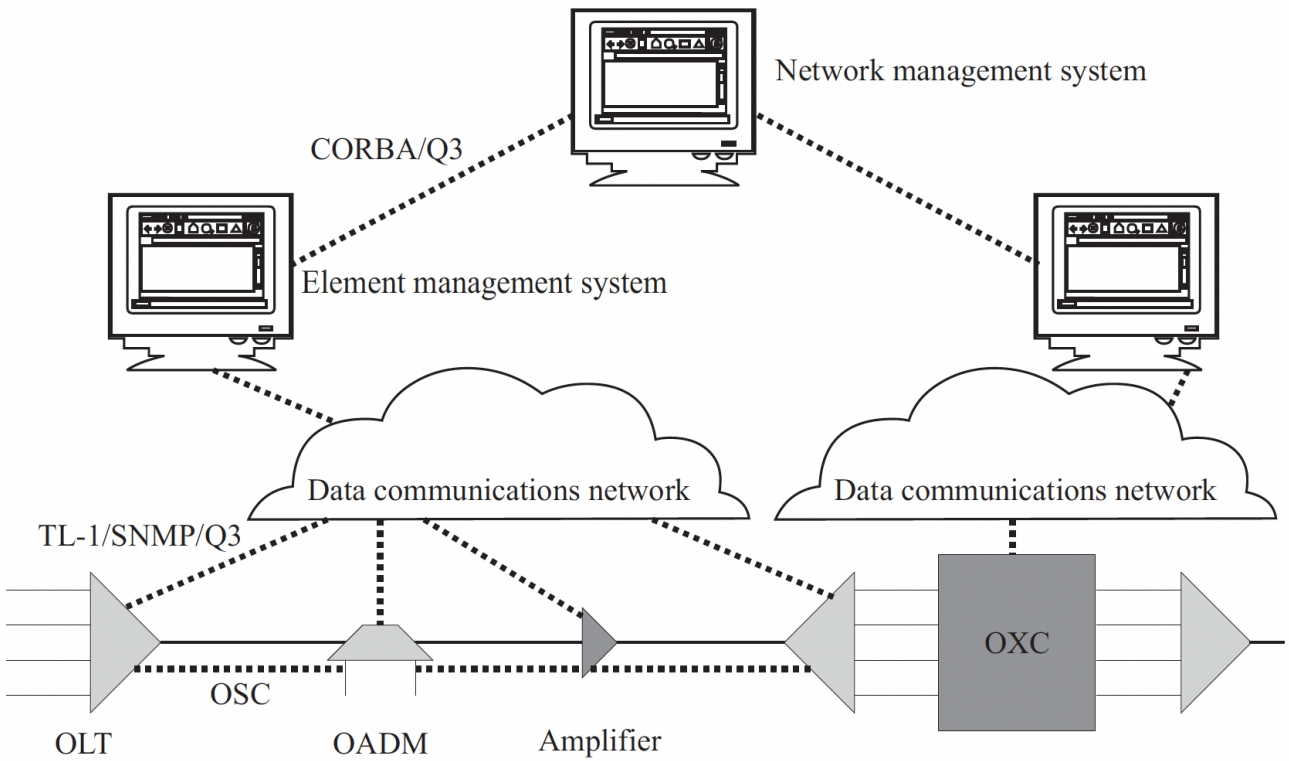


Figure 1.13 Evolution of optical fiber transmission systems. (a) An early system using LEDs over multimode fiber. (b) A system using MLM lasers over single-mode fiber in the 1.3 μm band to overcome intermodal dispersion in multimode fiber. (c) A later system using the 1.55 μm band for lower loss, and using SLM lasers to overcome chromatic dispersion limits. (d) A current-generation WDM system using multiple wavelengths at 1.55 μm and optical amplifiers instead of regenerators. The P-λ curves to the left of the transmitters indicate the power spectrum of the signal transmitted.

Control and Management

Management Framework

- Most functions of network management are implemented in a centralized manner by a hierarchy of management systems
- Due to latency, some management functions are performed in a decentralized manner (e.g., responding to failures and setting up and taking down connections)



Optical Layer Services and Interfacing

- Client layers can specify to the optical layer the following services during lightpath setup:
  - the endpoints to interconnect
  - the amount of bandwidth that is required
  - it can specify if an adaptation function is needed at the ingress or egress point
  - the targeted Bit Error Rate (BER)
  - the level of protection against failure events
  - requirements related to jitter and maximum end to end delay
- Enabling the delivery of these services requires a control and management interface between the optical layer and the client layer
  - The simple interface used today is through the management system
  - It works fine as long as lightpaths are set up fairly infrequently and remain nailed down for long periods of time

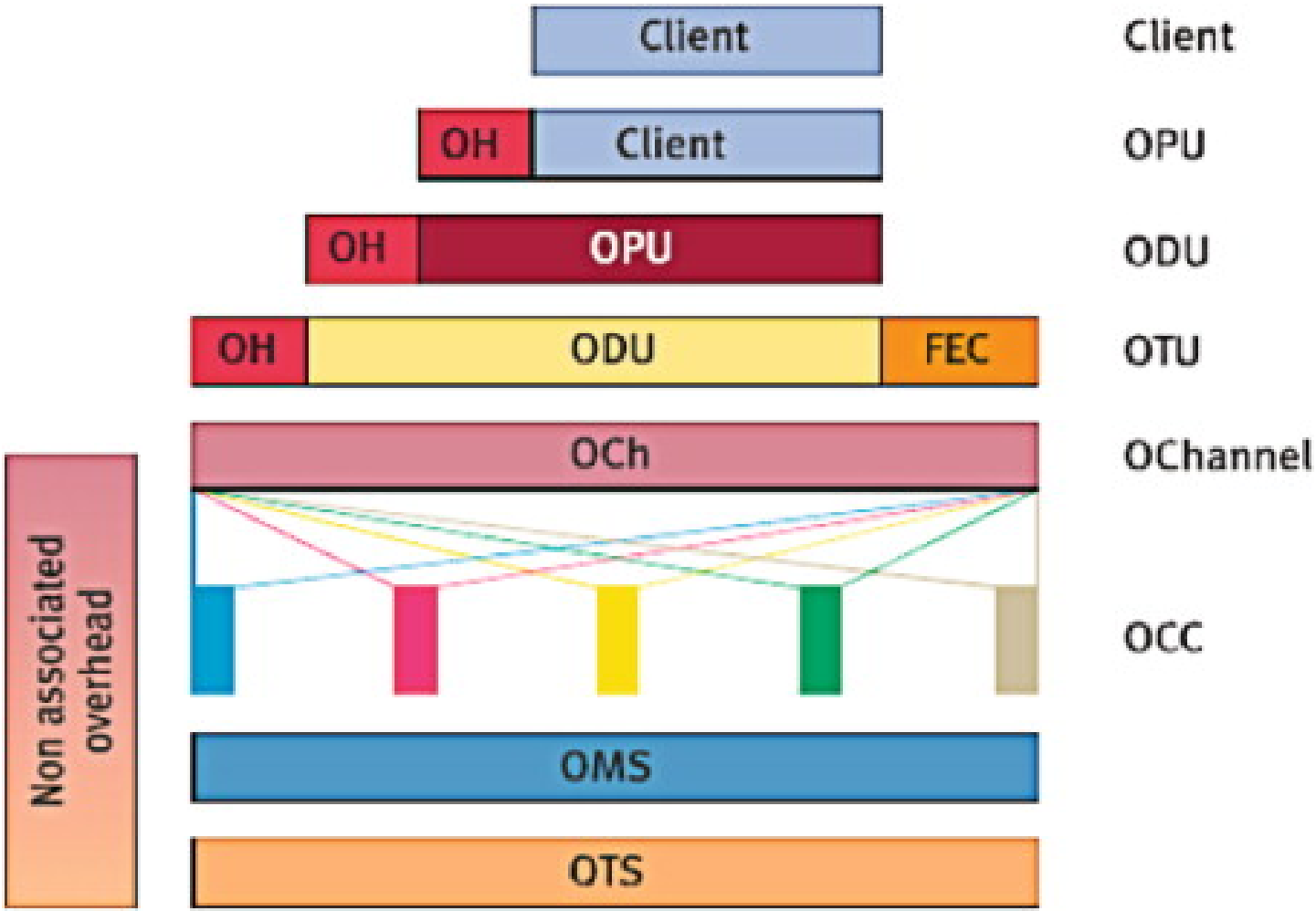
Layers within the Optical Layer

- The optical layer performs several functions (multiplexing, switching and routing, and performance monitoring)
- In order to help delineate management functions it is useful to further subdivide the optical layer into several sublayers:
  - *Optical Channel* layer (OCh):
    - takes care of end to end routing of the lightpaths
  - *Optical Multiplex Section* layer (OMS):
    - each link between OLTs or OADMs represents an optical multiplex section carrying multiple wavelengths
  - *Optical Transmission Section* layer (OTS):
    - link between two optical amplifier stages

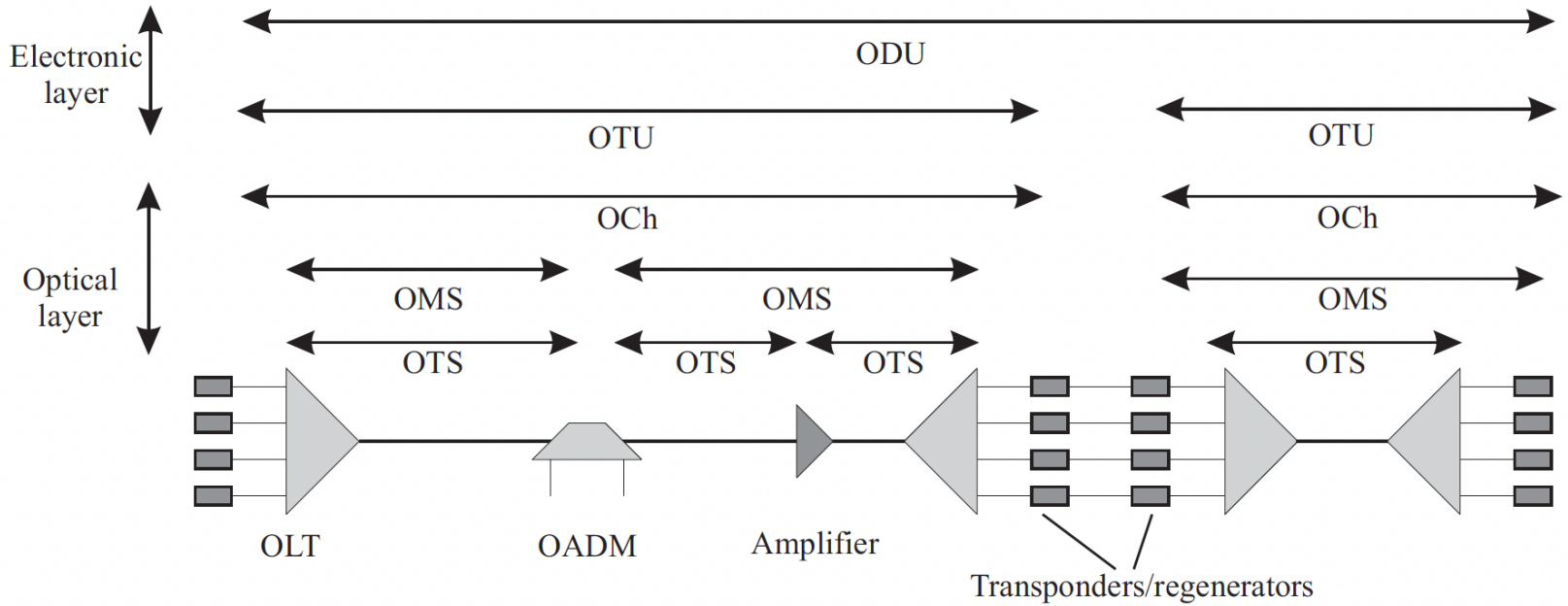
Layers within the Electronic Layer

- Optical channel transport unit (OTU)
  - delineate OTN frames
  - provide identification of the optical connection
  - monitor bit error rate (BER) performance
  - carry alarm indicators to signal failures
  - provide a communication channel between the end points of the optical connection
- Optical channel data unit (ODU)
  - as OTU, but at a higher layer
  - includes the Optical channel Payload Unit (OPU) sublayer that adapts client signals to the OTN frames

OTN Encapsulation



Layers within the Optical Layer



Performance and Fault Management

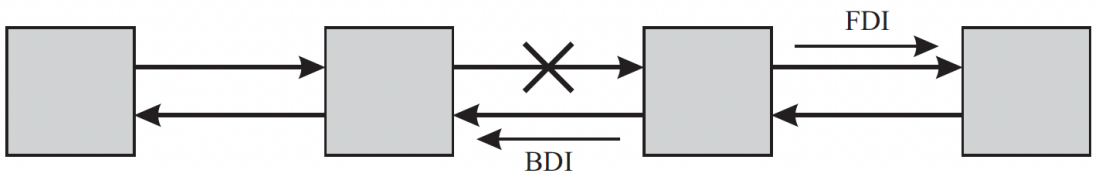
- The goal of *performance management* is to enable service providers to provide guaranteed quality of service to the users of their network
- This usually requires:
  - monitoring of the performance parameters for all the connections
  - taking any actions necessary to ensure that the desired performance goals are met
- *Fault management* involves:
  - detecting problems in the network
  - alerting the management systems appropriately through alarms
- Fault management also includes restoring service in the event of failures

BER and Optical Trace

- *BER*
  - The bit error rate (BER) is the key performance attribute associated with a lightpath
  - The BER can be detected only when the signal is available in the electrical domain, typically at regenerator or transponder locations
  - Overhead inserted in OTN frames, which consists of parity check bytes, allows for BER computation
- *Optical Trace*:
  - Lightpaths pass through multiple nodes and through multiple cards within the equipment deployed at each node.
  - It is desirable to have a unique identifier associated with each lightpath
  - This identifier is called an optical path trace
  - The trace enables the management system to identify, verify, and manage the connectivity of a lightpath

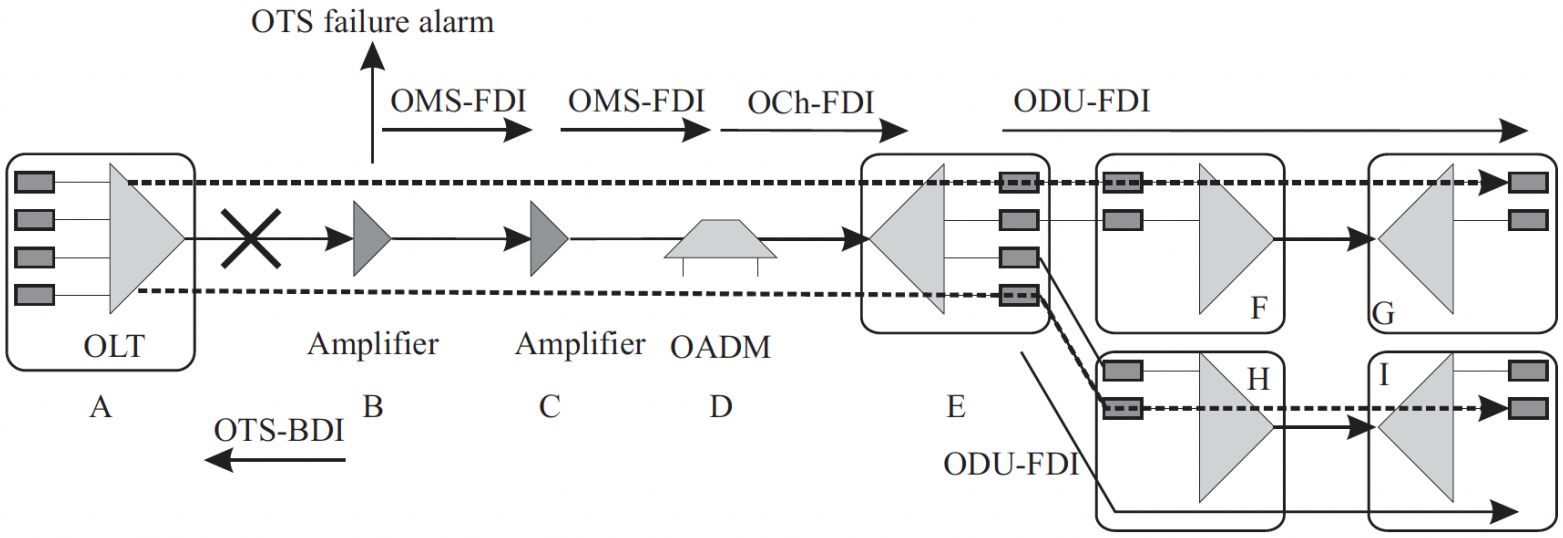
Alarm Management

- In a network, a single failure event may cause multiple alarms to be generated
  - in a network with 32 lightpaths on a given link, each traversing through two intermediate nodes, the failure of a single link could trigger a total of 129 alarms
- Alarm management it is required to identify the root-cause alarm of the failure and suppress the redundant alarms
  - Alarm suppression is accomplished by using a set of special signals, called the forward defect indicator (FDI) and the backward defect indicator (BDI)



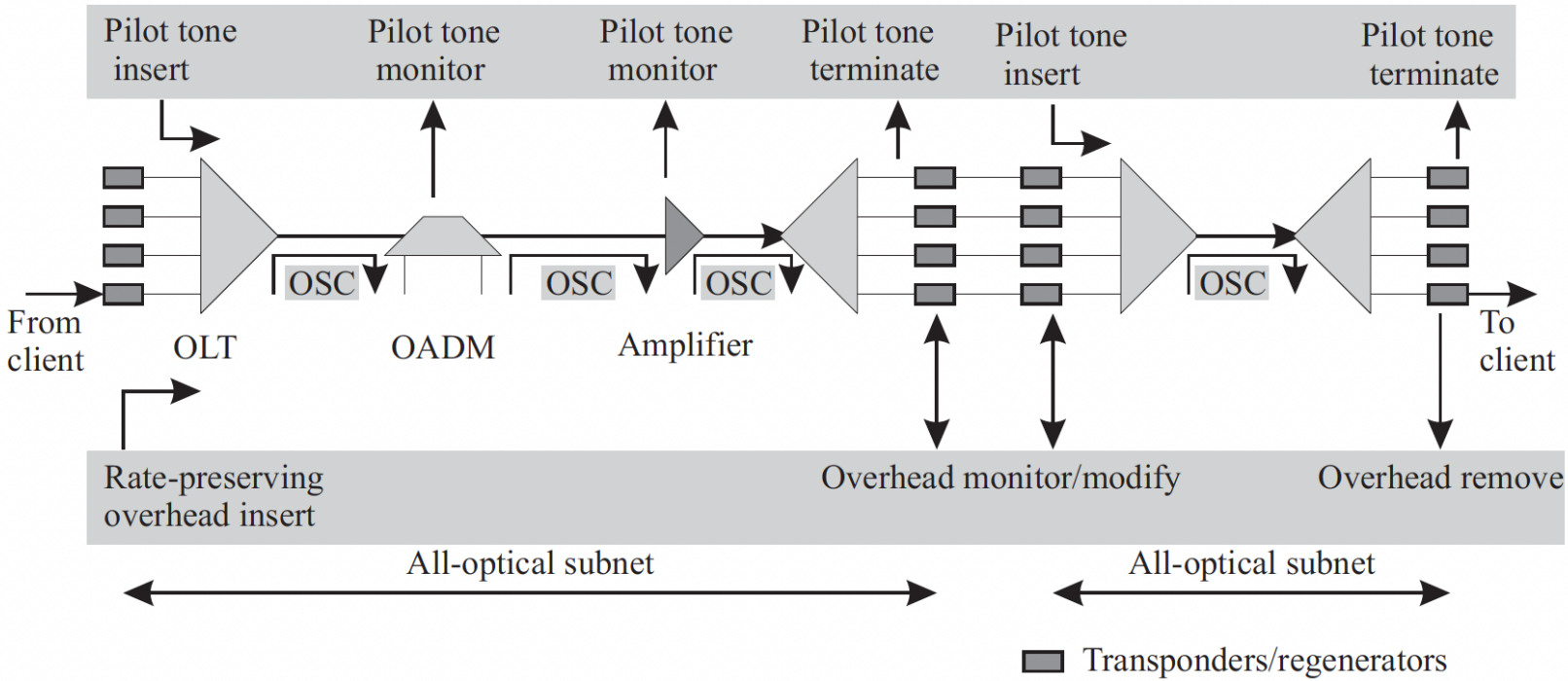
Alarm Management: Example

- When a link fails, the node downstream of the failed link inserts an FDI signal downstream to the next node
- The FDI signal propagates rapidly, and nodes further downstream receive the FDI and suppress their alarms
- The node also sends a BDI signal upstream to the previous node, to notify that node of the failure
- FDI and BDI are sent at different sublayers of the optical layer



Optical Layer Overhead

- Supporting the optical path trace, defect indicators, and BER measurement requires the use of some sort of overhead in the optical layer



Optical Layer Overhead

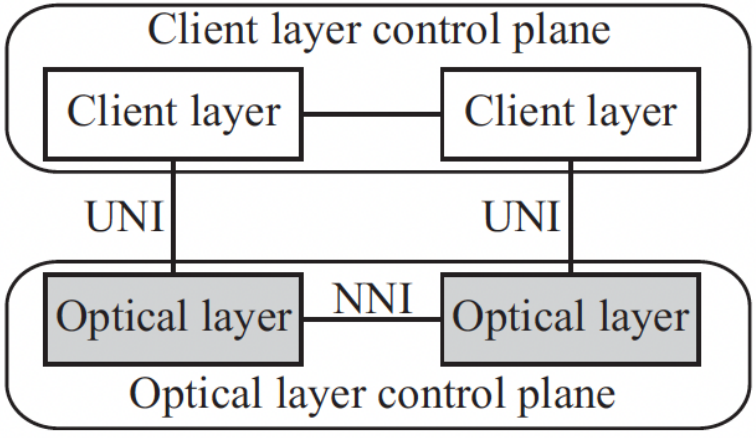
	All-Optical Subnet		End-to-End
Application	**OSC**	**Pilot Tone**	**Rate-Preserving**
Trace	OTS OCh		OTU ODU
DIs	OTS OMS OCh	None	OTU ODU
Performance monitoring	None	Optical power	BER
Client signal compatibility	Any	Any	Any

Connection Management

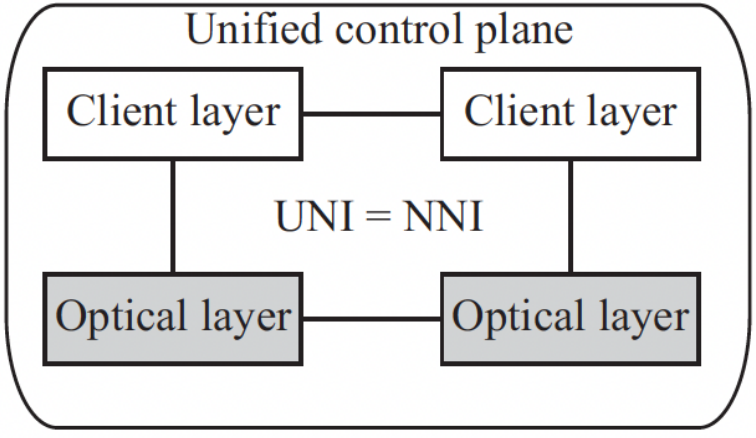
- Connection management deals with setting up connections, keeping track of them, and taking them down when they are not needed anymore
- Two different approaches
  - client-server model
    - the client layer (IP, SDH, ATM, etc.) asks to the server layer the establishment of a connection (lightpath) without having knowledge of the internal structure of the optical network
    - centralized control
    - suitable as long as lightpaths are set up fairly infrequently
  - peer model
    - tight coupling between the client and optical layers: the optical layer primarily serves a single client (IP)
    - distributed control
    - useful if there is a need to set up and take down connections rapidly

Interaction with Other Layers

- *Overlay Model*



- *Peer Model*



Distributed Connection Management

- Distributed connection control has several components
  - topology management: each node in the network maintains a database of the network topology and the current set of resources available
    - OSPF–Traffic Engineering (OSPF-TE) used to distribute link state info
  - link management: nodes monitor the status of the link by exchanging periodic "hello"
    - BER can also be considered to assess the status of a link
  - route computation: routing algorithm applied on the topology database
  - signaling protocol: set of messages used to set up a connection once the path has been calculated
    - RSVP Traffic Engineering (RSVP-TE) and Constraint-based Routing LDP (CR-LDP)

Network Infrastructures A.A. 23/24

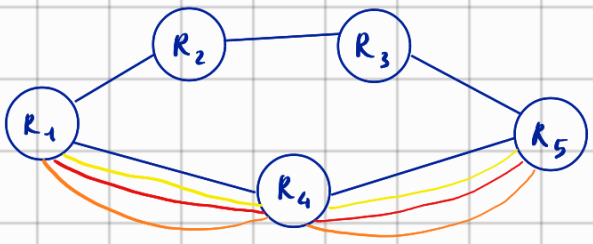
2 - Multi Protocol Label Switching

SAPIENZA
UNIVERSITÀ DI ROMA

Now we're in the electrical domain and we're going to to talk about IP networks.
As we know an IP network it's a connectionless environment.

One of the most important function, for an IP network, is routing.
In order to perform that function every router utilize a routing table that has to be populated.
To find packet paths over the network, various algorithms are used.
One of them is Dijkstra, it finds the shortest path from the sender to the destination.
Not always that's the best choice because of the high traffic concentration over a specific router.

i.e.

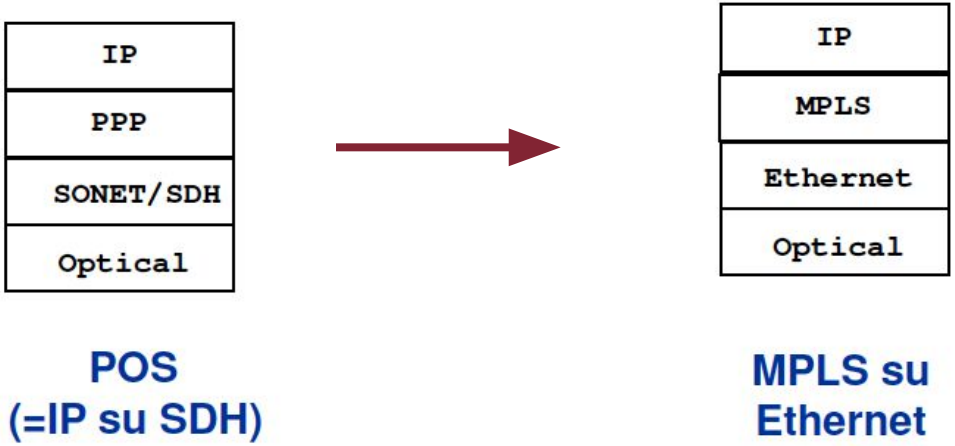


$R_1 - R_4 - R_5$ it's the shortest path, but as shown is also the only one to be used and that's not really good.

The needs of a TLC provider of data services

- Offer services to its customers
  - good “Quality of Service” – QoS
  - Service Level Agreements – SLAs
- Run its backbone in a cost-effective way
  - convergence of services (example voice and data services)
- For its Data backbone the provider needs good support for:
  - Virtual Private networks
  - Traffic Engineering → it's important in order to let our network to be flexible when have to find the path from A to B.
  - Protection/restoration mechanism → against failures
- Connection-oriented packet switching technologies represent a good answer to these needs

MPLS over Ethernet

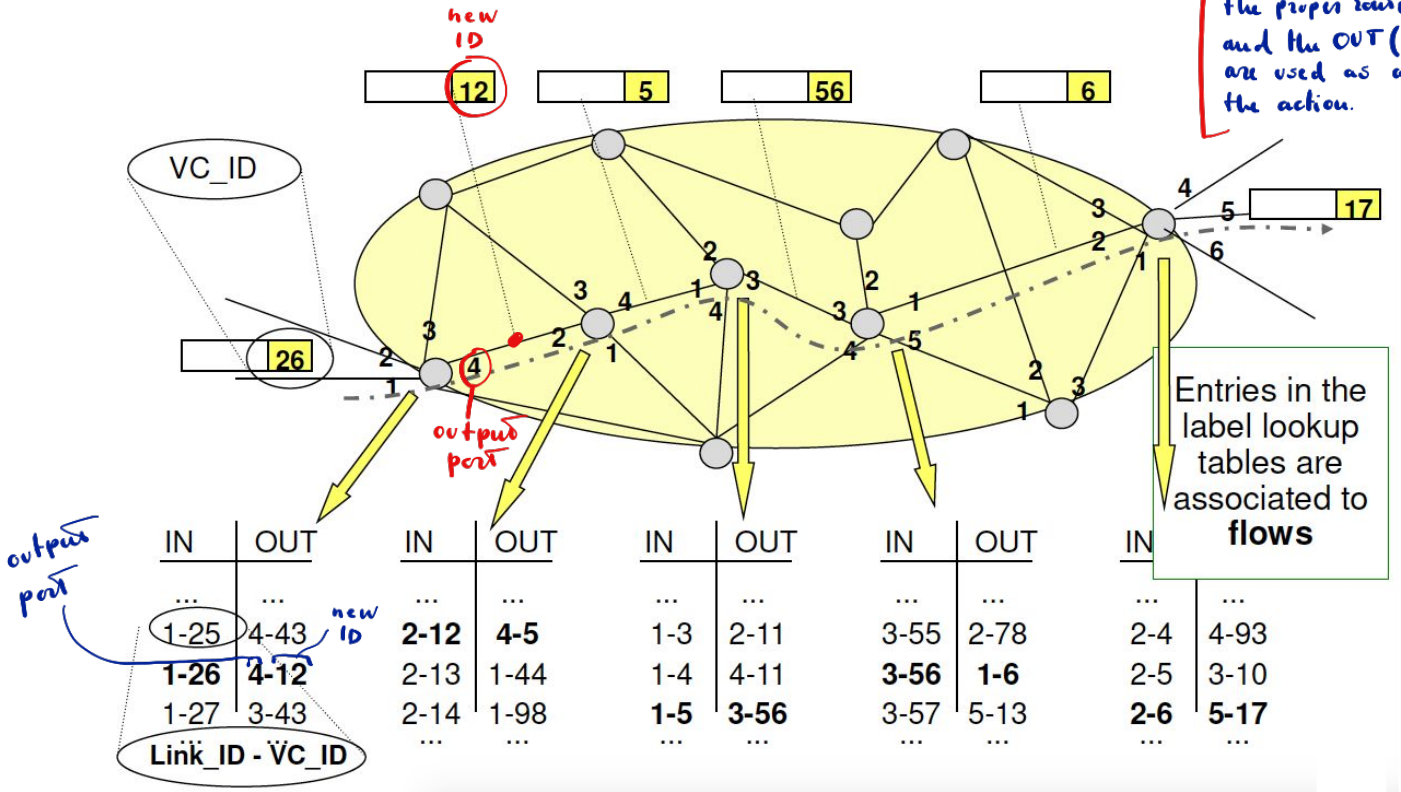


POS
(=IP su SDH)

MPLS su Ethernet

Connection-oriented packet network

The IN (input) information are used to perform the matching in order to find the proper route to save the packet, and the OUT (output) info are used as arguments for the action.



IN	OUT
...	...
1-25	4-43
1-26	**4-12**
1-27	3-43
...	...

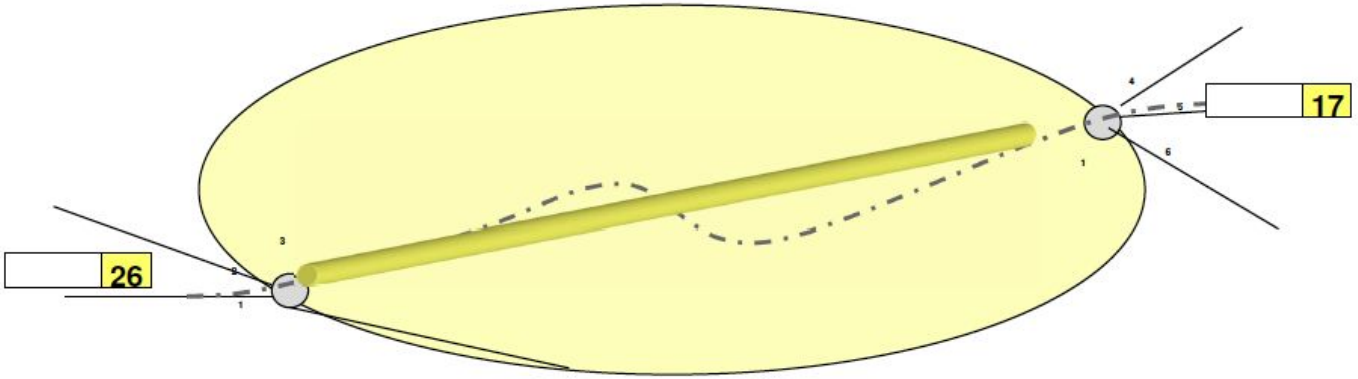
IN	OUT
...	...
2-12	**4-5**
2-13	1-44
2-14	1-98
...	...

IN	OUT
...	...
1-3	2-11
1-4	4-11
1-5	**3-56**
...	...

IN	OUT
...	...
3-55	2-78
3-56	**1-6**
3-57	5-13
...	...

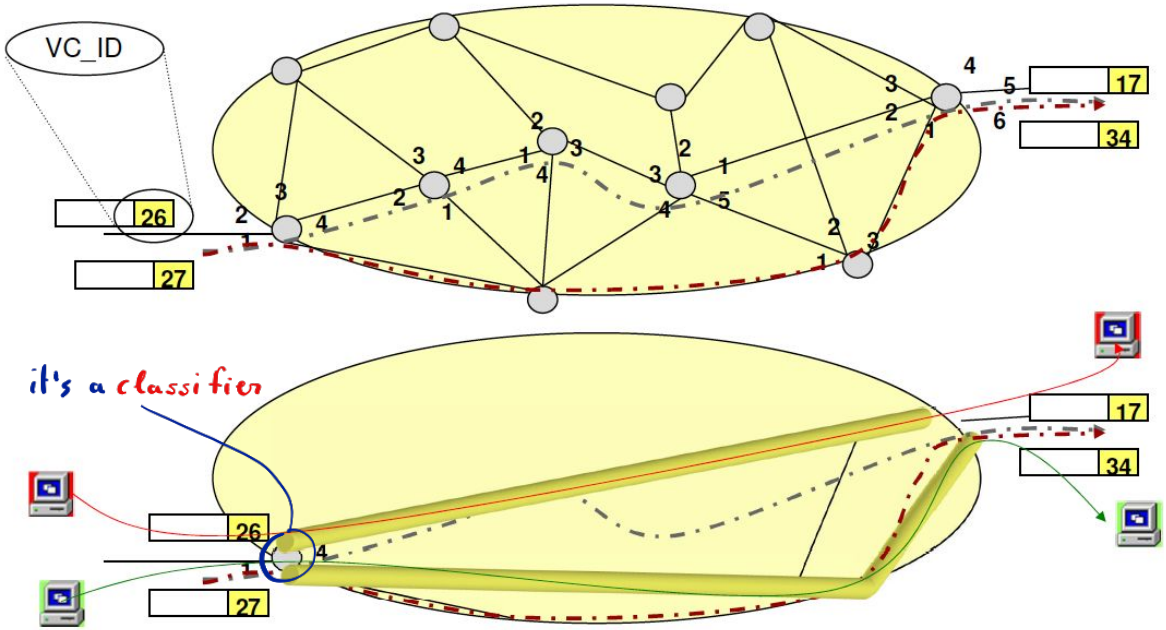
IN	OUT
...	...
2-4	4-93
2-5	3-10
2-6	**5-17**
...	...

Connection-oriented packet network



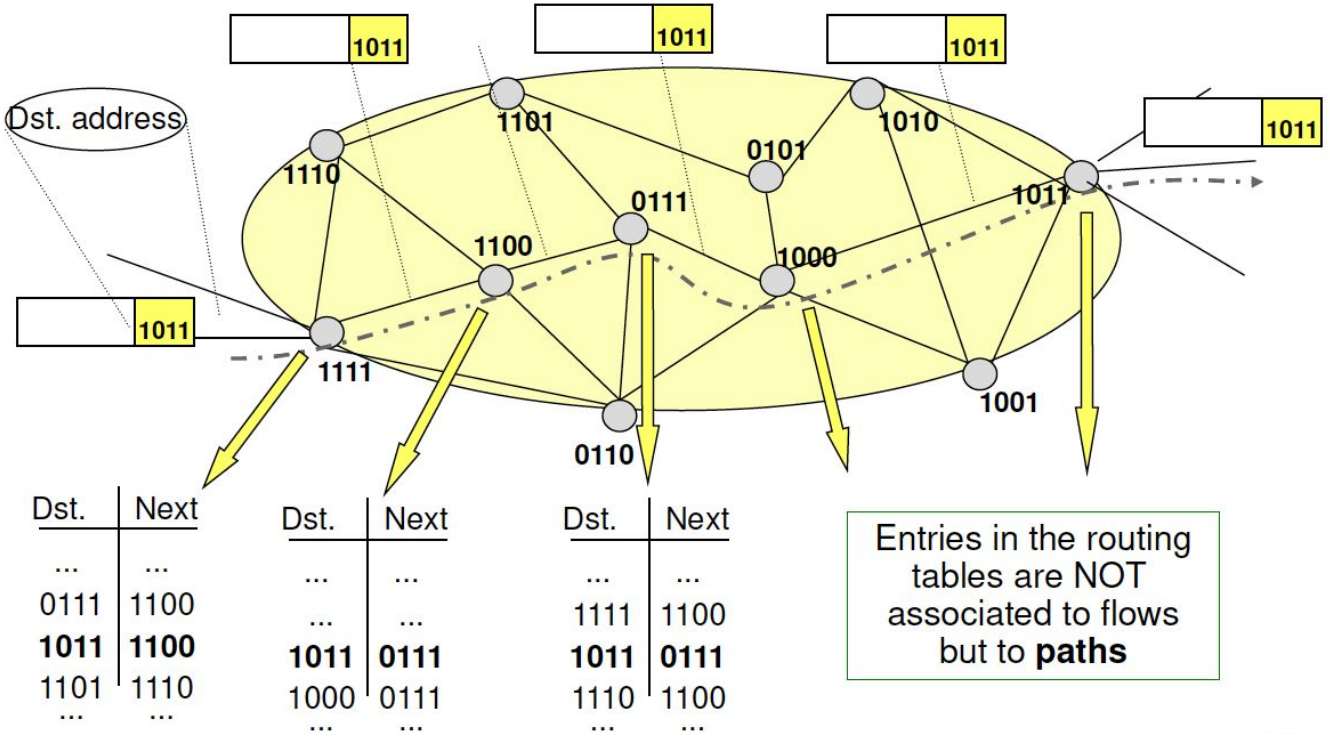
The connection can be seen as a **tunnel** from the ingress to the egress

Connection-oriented packet network

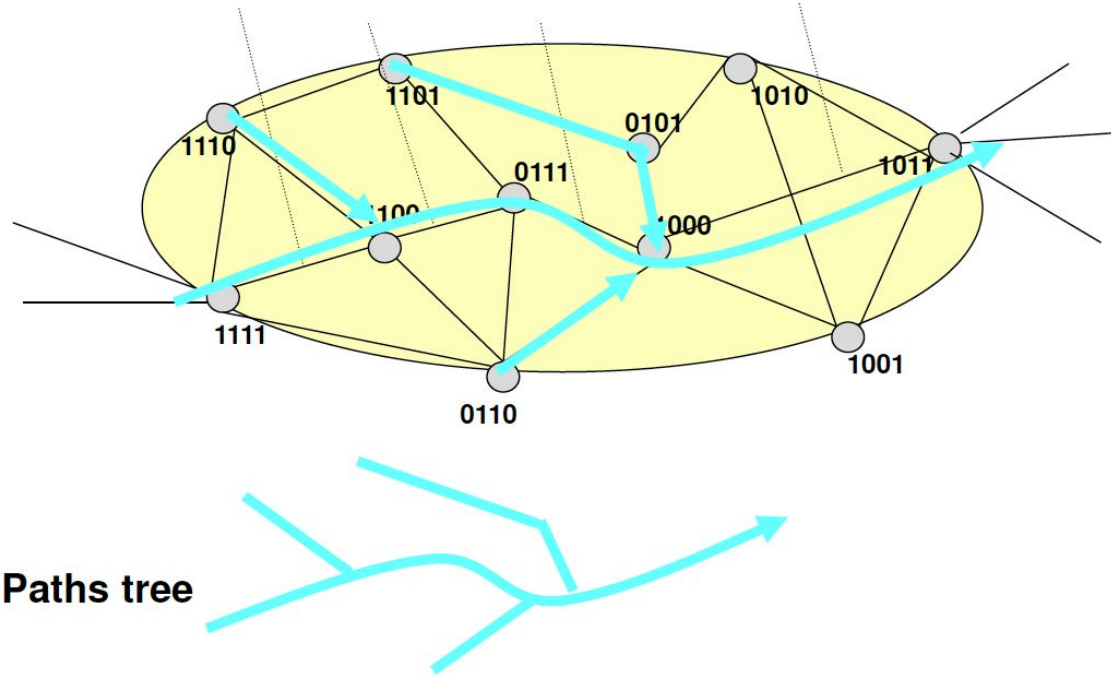


path differentiation in connection oriented packet networks

Connection-less packet network

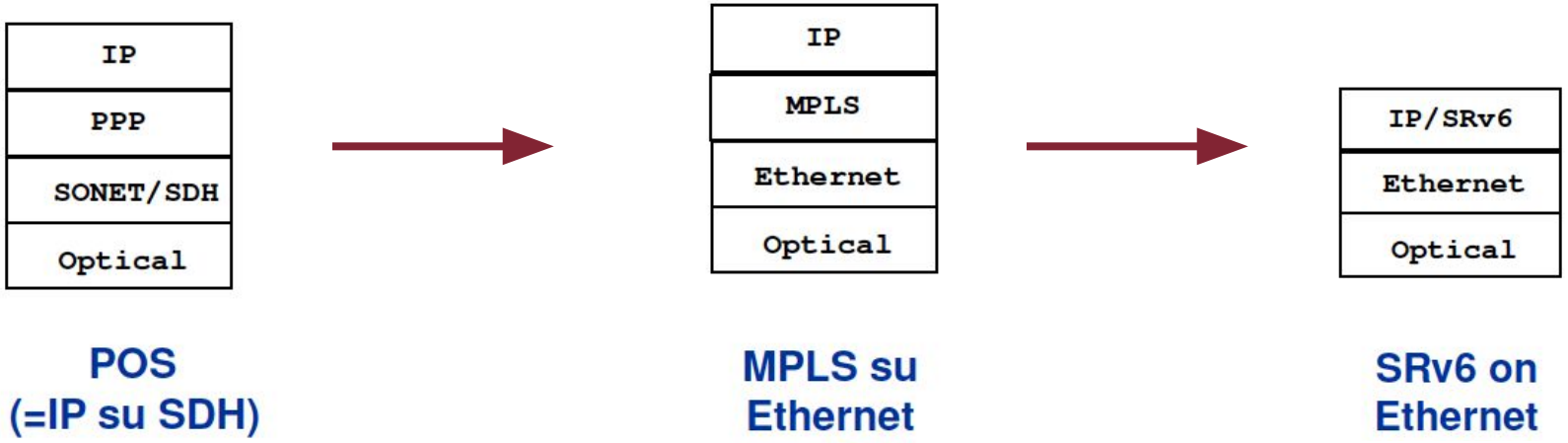


Connection-less packet network



path aggregation in connection-less packet networks

What next? removing MPLS…



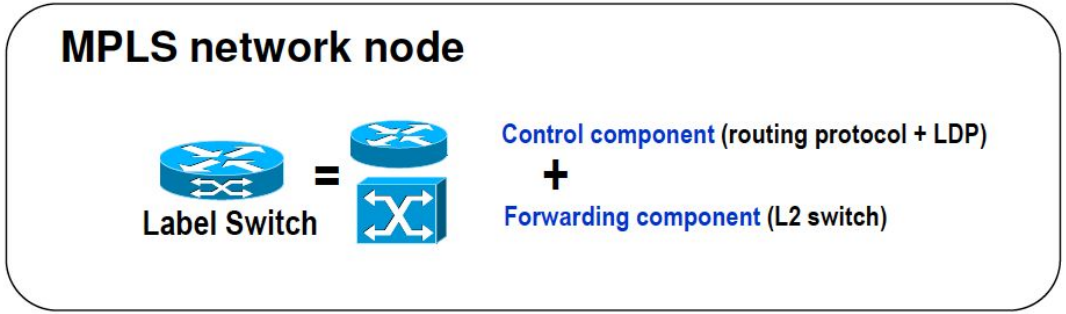
MPLS Architecture

Multi Protocol Label Switching

it means that's possible to encapsulate any type of packet into a MPLS packet.

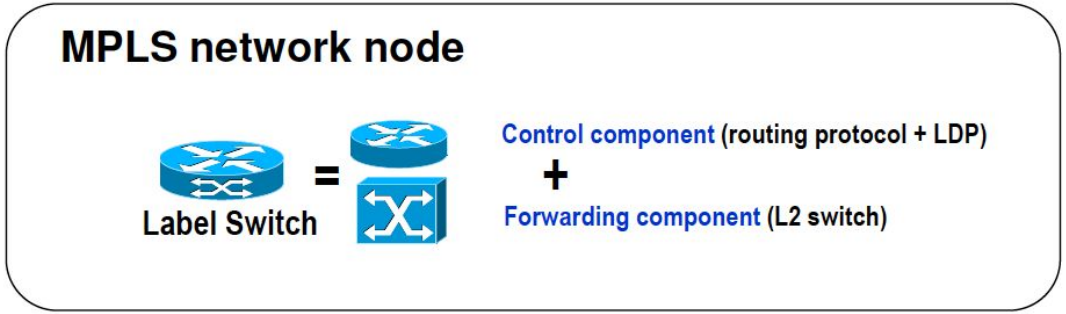
- The basic idea behind this architecture is to add a fixed length identifier, called **Label**, that can be used by internetworking device to forward packet based on Label switching
- MPLS is independent both from the underlying technology (Frame Relay, ATM, SDH, Ethernet etc.) and from the upper networking protocol (but only IP is used…)

MPLS network node



- **Control Component:**
  - Level 3 ‘intelligence’ (IP addressing, IP routing)
  - A set of modules for Label allocation and Label binding and for exchanging Label allocation information with adjacent nodes
- **Forwarding Component:**
  - Forwarding with Label switching paradigm

MPLS network node

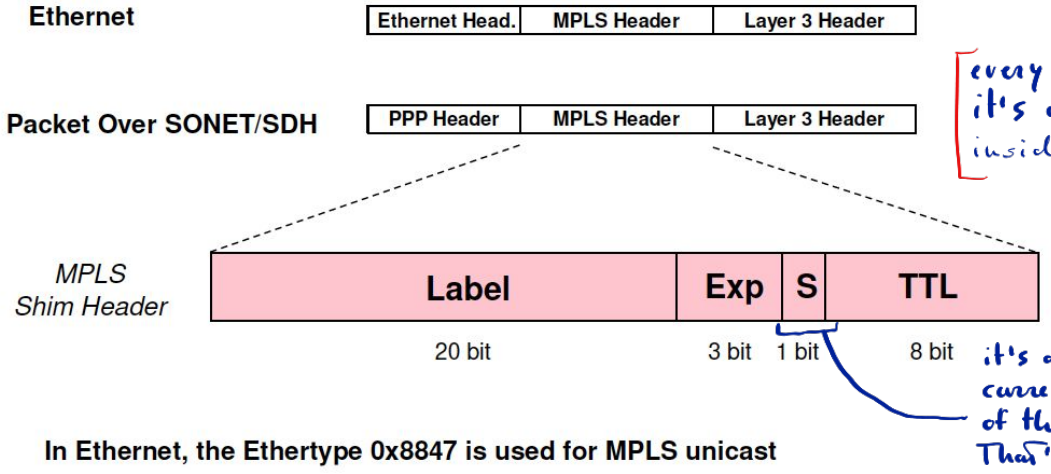


- The Control Component and the Forwarding component are **integrated within an IP Router** that becomes "IP/MPLS router"
- The MPLS Forwarding Component is an **extension of IP forwarding capability**

Label encoding

The Transport Network (TN) encapsulates every incoming IP packet in a MPLS packet. It means adding a MPLS header to the packet.

- The MPLS label is carried in a MPLS header (also known as “shim header”)
- It is inserted between Layer 2 and Layer 3 header



every header in the stack it's a specific tunnel nested inside the others.

it's a flag that shows if the current MPLS header it's the bottom of the headers stack. That's because MPLS supports multiple level encapsulation.

Terminology

- **Label Edge Router (LER):** border node for a MPLS «domain» → ingress/egress node of a tunnel.
  - it forwards packets to and from the MPLS domain, inserting and removing the Label to packets entering and going out from the domain
- **Label Switching Router (LSR):** a node that perform the Label switching within the MPLS domain → middle node.
- **Label Distribution Protocol (LDP):** together with traditional IP routing protocols, a LDP is used to distribute the Labels among MPLS devices → it creates the tunnel.
- **Forwarding Equivalence Class (FEC):** a set of IP packets that are forwarded in the same way and receive the same treatment → it represents the packet label.
- **Label Switched Path (LSP):** the path across one or more LSRs (followed by packet belonging to a FEC) → it's a MPLS tunnel.
  - it corresponds to the "Virtual Circuit" concept

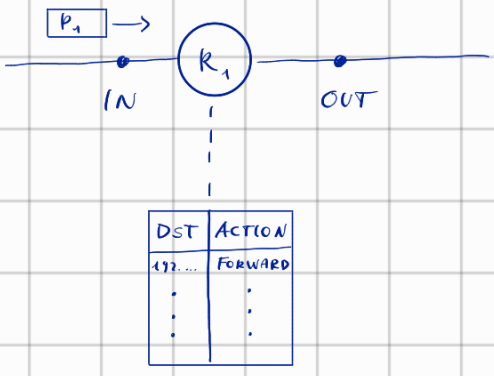
MPLS Dataplane procedures

- The **Ingress LER** of a MPLS domain analyzes the IP header of the packet, classifies the packet, adds the MPLS label and forwards to the next hop LSR
- In the MPLS domain the packet is forwarded along the LSP according to the Labels
- The **Egress LER** removes the Label and the packet is forwarded based on the IP destination address
- Three basic actions:
  - **PUSH** operation - add the MPLS label → usually performed by an edge node
  - **POP** operation - remove the MPLS label → performed by an edge node, the egress.
  - **SWAP** operation - change the label (in an intermediate node) → transit nodes.

Actions performed by a IP Router (IPR)

When a router receives a packet has to check its table in order to try to match the dst address of the packet with an action to perform.

i.e.



The paradigm is called Match - Action, in the case of an IP router, the match it's the dst address and the action is the action linked to the specific address.

There are 2 main actions: FORWARD / DROP

Actions performed by a Label Switching Router (LSR)

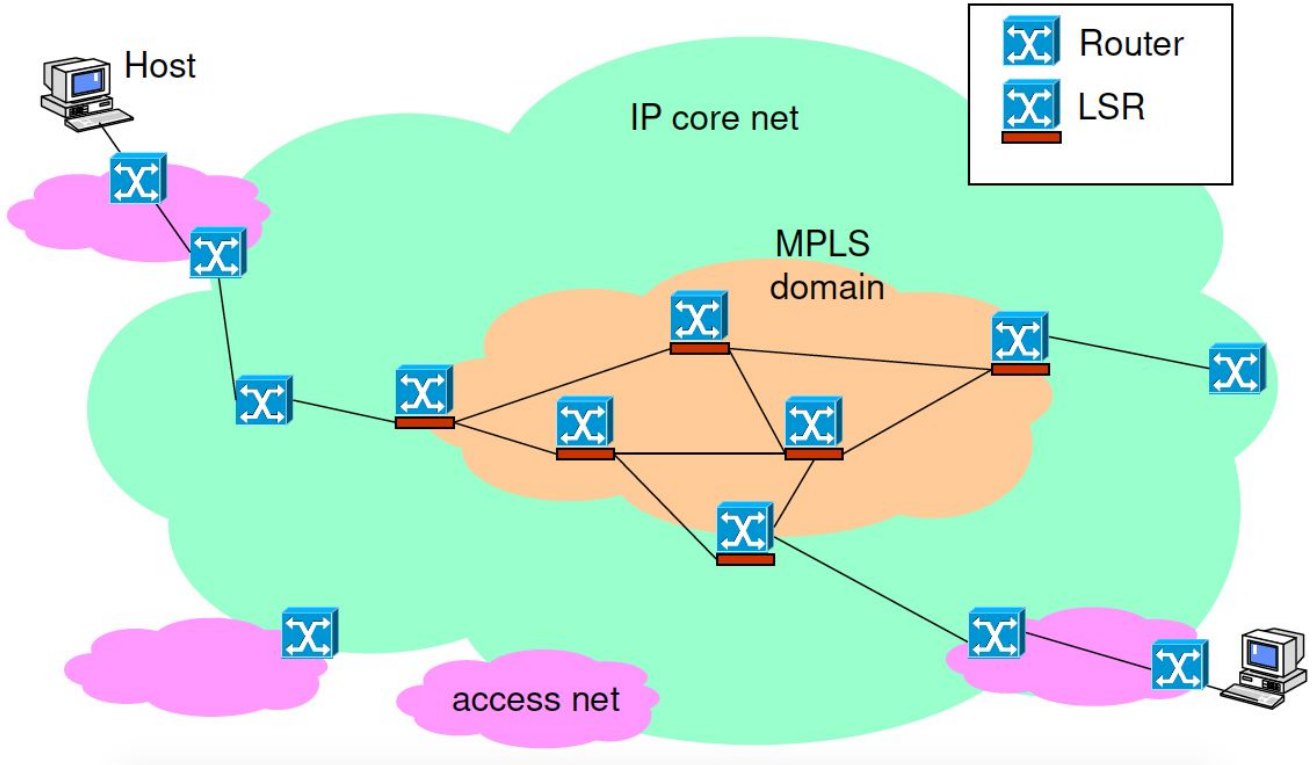
In this case the router has to match the incoming packet label in its table.

The fact that the router has to match a number (label) and not an address prefix makes the process much simpler and it's called Exact Matching.

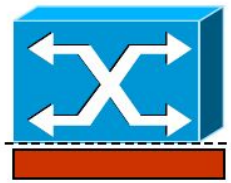
The actions in that case are:

1) Label Swap
2) Push Header (performed by the classifier)
3) Pop Header (performed by the classifier)

MPLS: example



Structure o of a Label Switched Router



IP Header processing + [LDP]

Destination	Next Hop
151.99.5.0	A
12.119.53.0	A
161.80.5.0	B
31.0.0.0	C

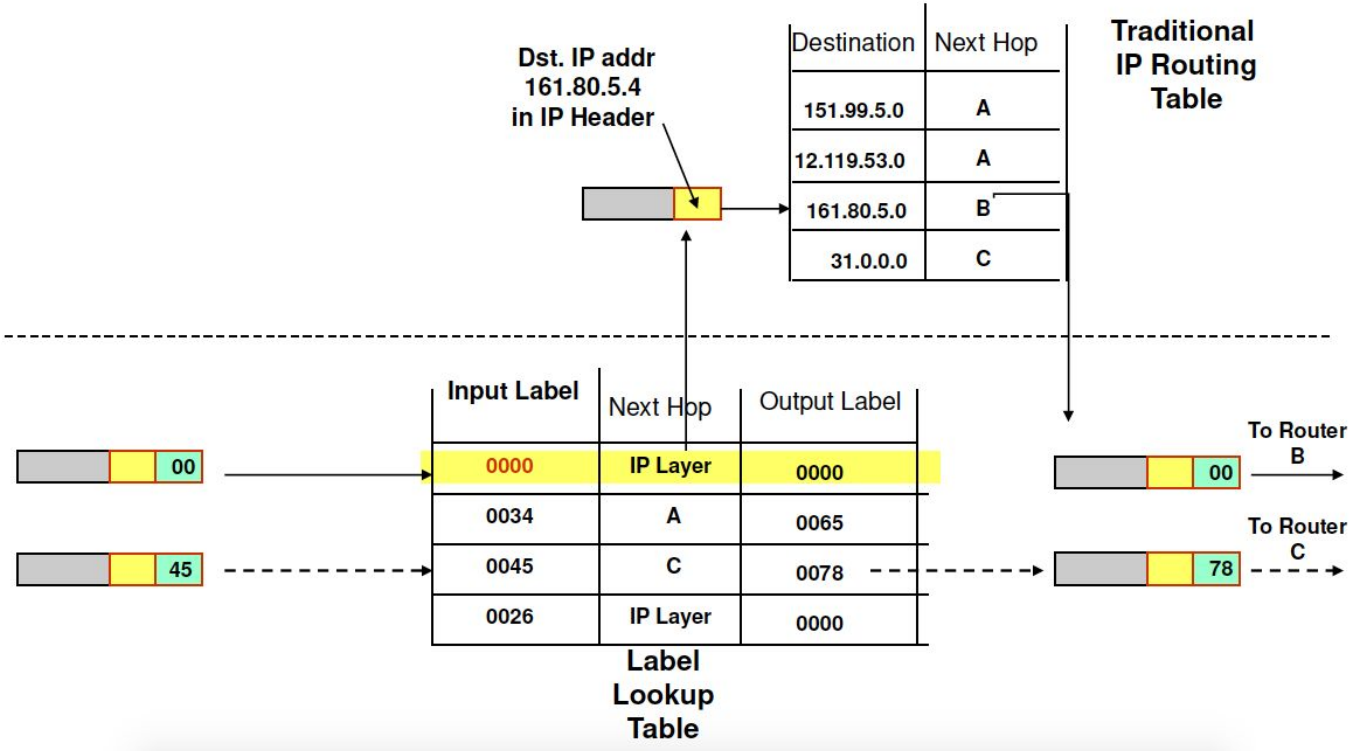
Traditional IP Routing Table

LABEL processing

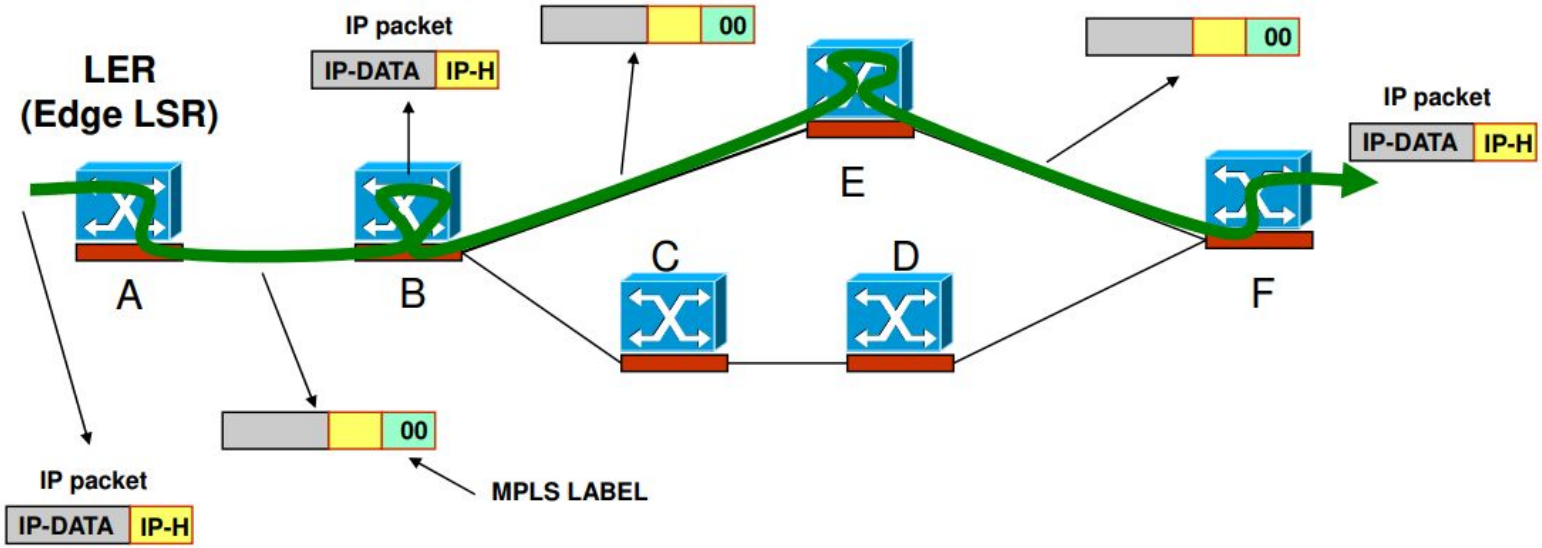
Input Label	Next Hop	Output Label
0000	IP Layer	0000
0034	A	0065
0045	C	0078
0026	C	0000

Label Lookup Table

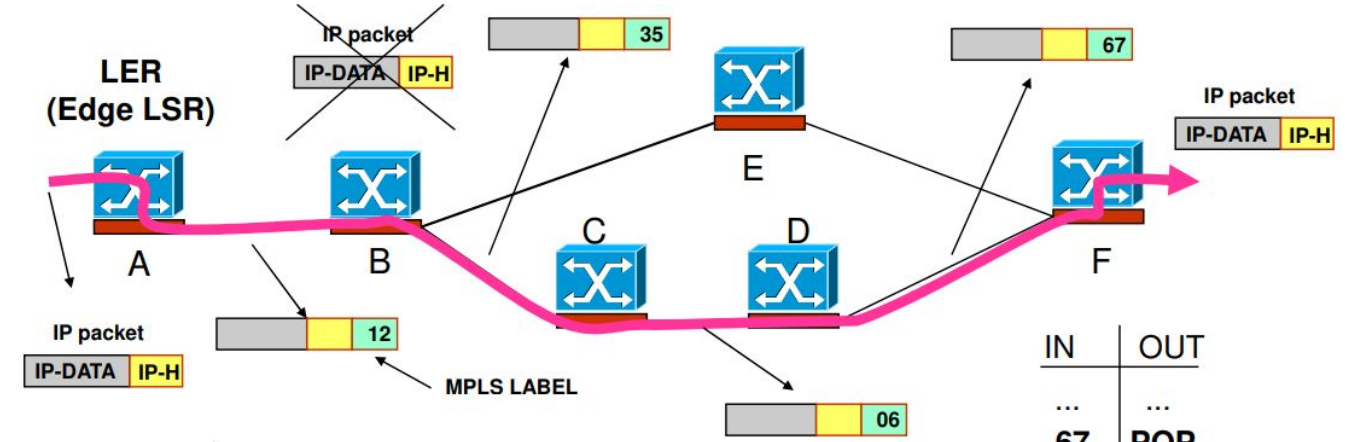
Label Switched Router



Connectionless forwarding along the IP path



Connection oriented forwarding along a LSP



FEC	OUT
...	...
160.80.1.0/24	B - PUSH 12
...	...

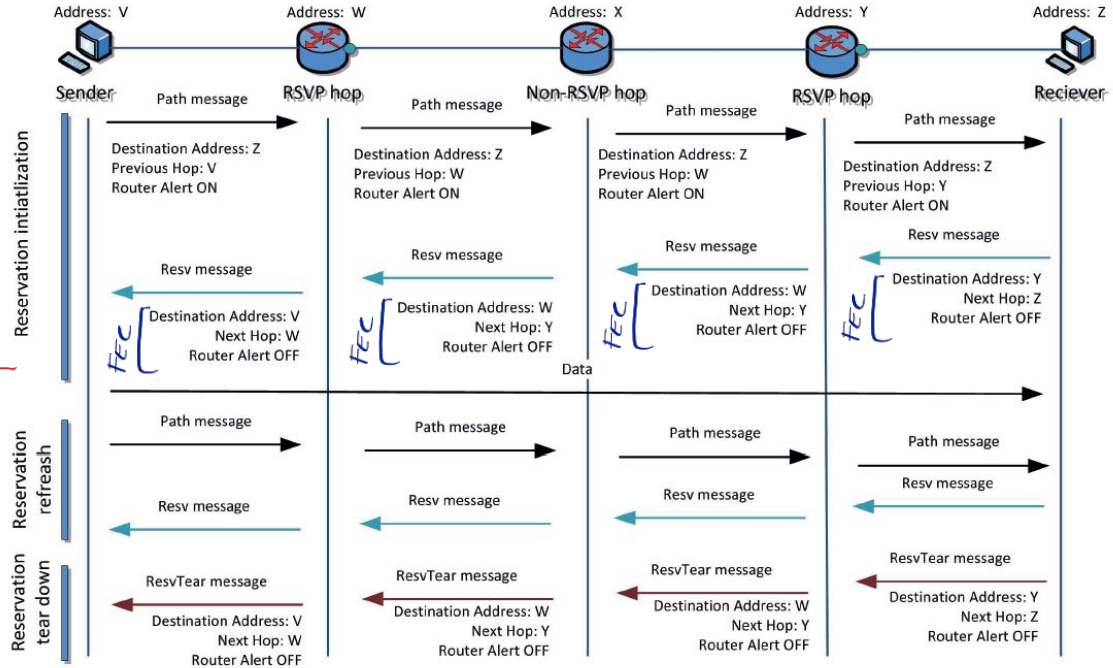
IN	OUT
...	...
67	POP
...	...

Classification in the Ingress LER based on the FEC

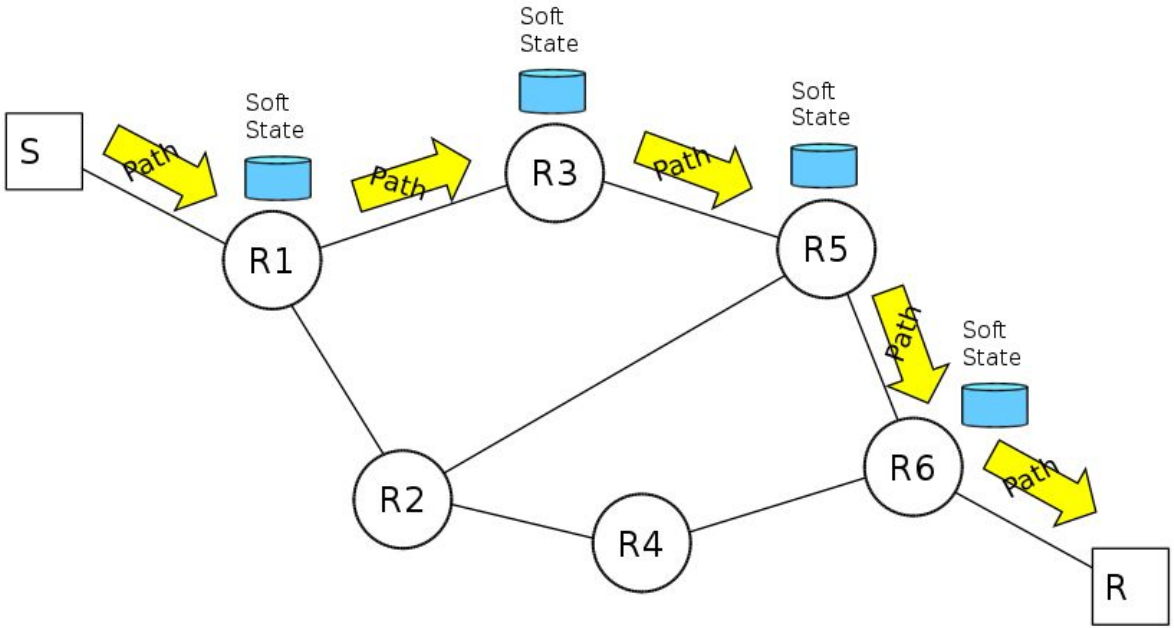
In the Egress LER the Label switching table instructs to POP the MPLS label and process the packet at the IP level

Control Plane: LSP setup

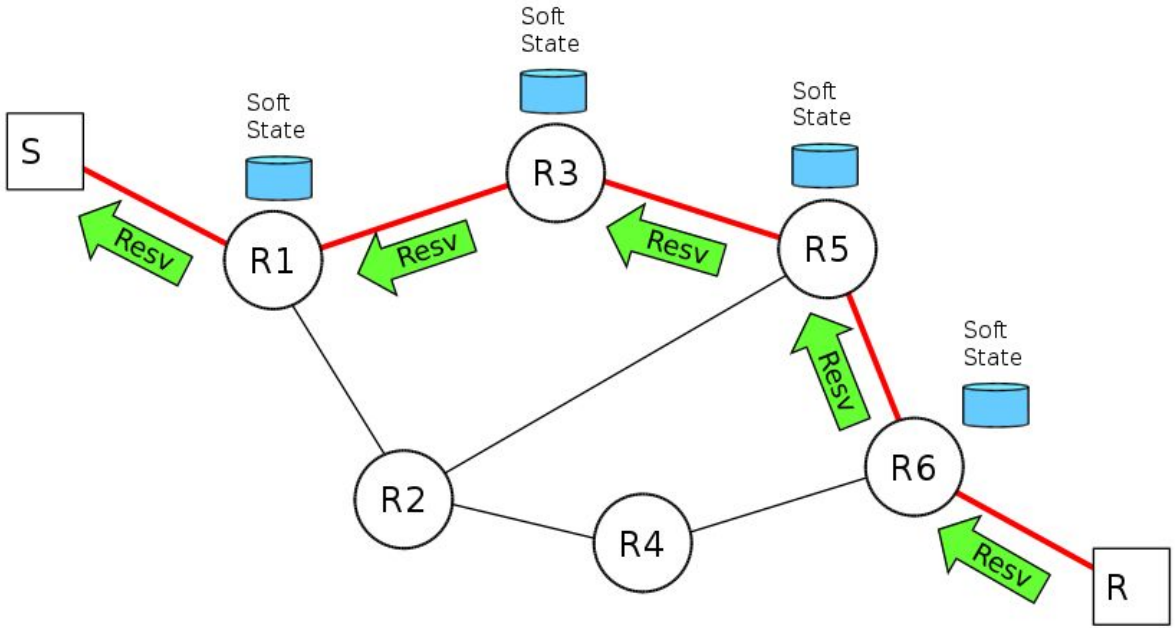
It's the process that constructs the application flow in the tunnel.



LSP Setup: example

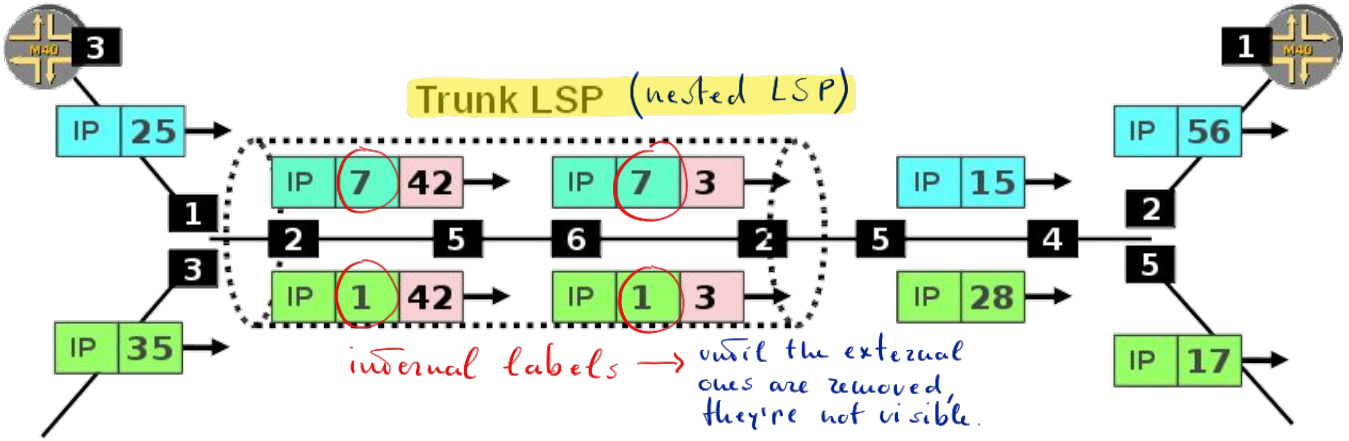


LSP Setup: example



Label stack

To know how many nested tunnel are there we can just count the MPLS labels on a packet.



MPLS Table

In	Out
(1, 25)	(2, 7, Push [42])
(3, 35)	(2, 1, Push [42])

MPLS Table

In	Out
(5, 42)	(6, 3)

MPLS Table

In	Out
(2, 3)	(5, Pop[3]) (7,15) (1,28)

MPLS Table

In	Out
(4, 15)	(2, 56)
(4, 28)	(5, 17)

Main reasons for MPLS adoption

- Why ISPs have adopted MPLS?
  - The main advantage of MPLS for an ISP is that it provides tools to better control the networking layer, useful to:
    - build new services
    - simplify some procedures
    - optimize network utilization
- In particular, MPLS has been used by ISPs in their backbone for:
  - Virtual Private Networks (VPN)
  - Fault protection
  - Traffic Engineering
  - Quality of Services

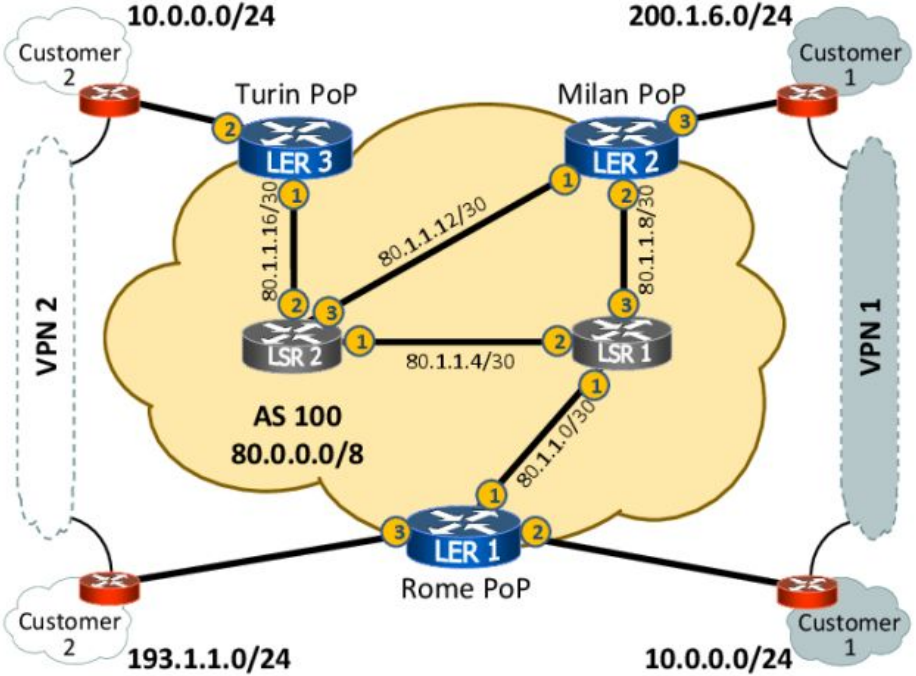
VPN, Traffic Engineering and Fast Re-Route

MPLS Virtual Private Networks

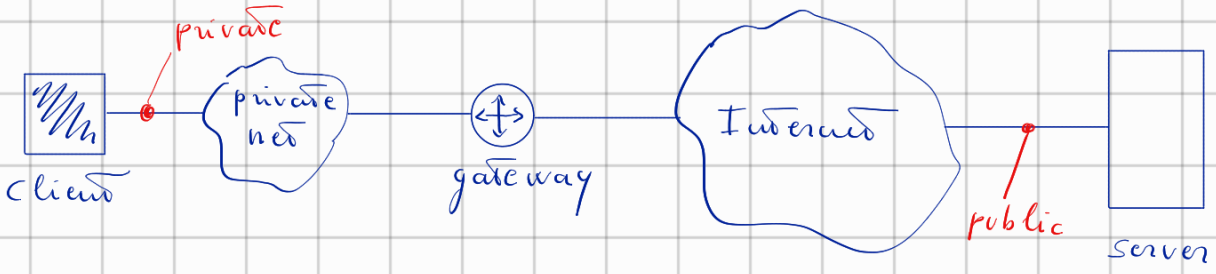
- Let us consider a company that needs to run an IP Intranet
  - A Virtual Private Network (VPN) offers a separated (private) network over a shared transport infrastructure
    - How can we use an IP based network as transport infrastructure?
- The two main features of a VPN services are:
  - security/privacy
  - support of private IP addressing
- MPLS is a simple and efficient solution for both features, as the forwarding is based on the Label and not on the IP destination address of the packets

This transport network is a MPLS one.

Problem Statement



From private to public network



In order to let the client communicate with the server (and viceversa) the gateway has to perform NAT (Network Address Translation).

Instead, when source and distination addresses are both private, NAT is not longer useful and the gateway has to use a VPN service.
This servia creates a tunnel over a public network (i.e. Internet).

Checkmate VPNs in Three Moves

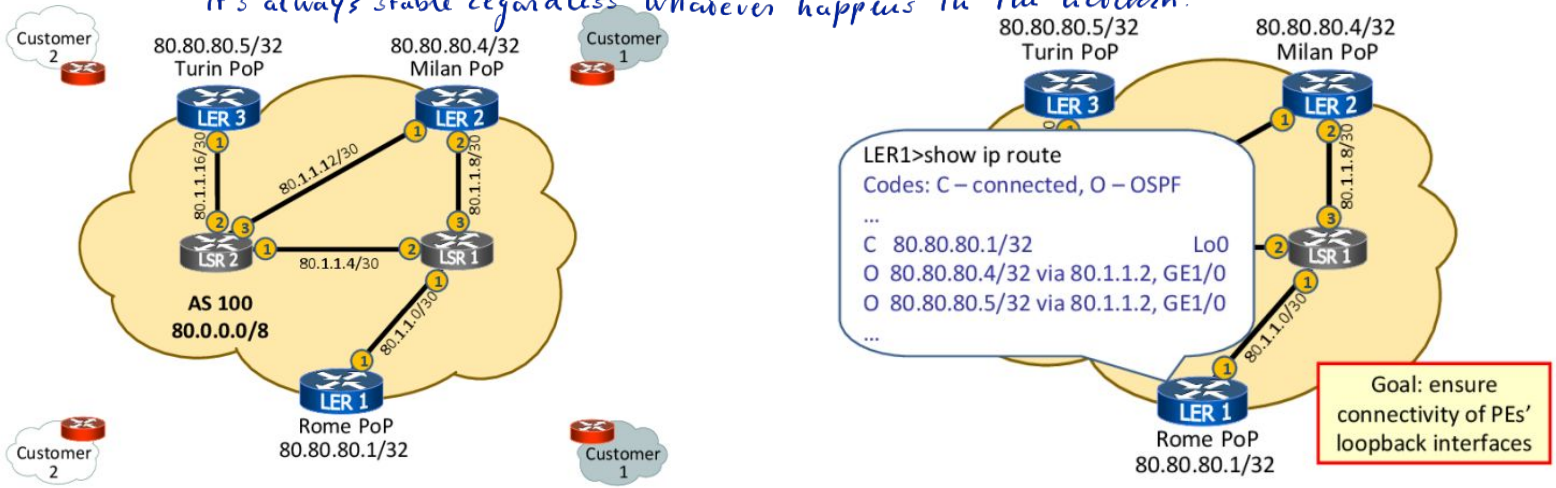
- Move 1: Achieve any-to-any IP connectivity among PEs,
- Move 2: Define a signalling mechanism to distribute customer prefixes among PEs, and
- Move 3: Define an encapsulation mechanism to transport packets from one PE to another across the network

Move 1 – Any-to-any IP connectivity among PEs

It means we want to be sure that every device in the network it's able to communicate with the others.

assign a loopback address to each PE router and use an IGP (e.g. OSPF or IS-IS) to announce these addresses as /32 prefixes in order to ensure any-to-any connectivity among them

it's a logical interface that, from its creation on it's always stable regardless whatever happens in the network.

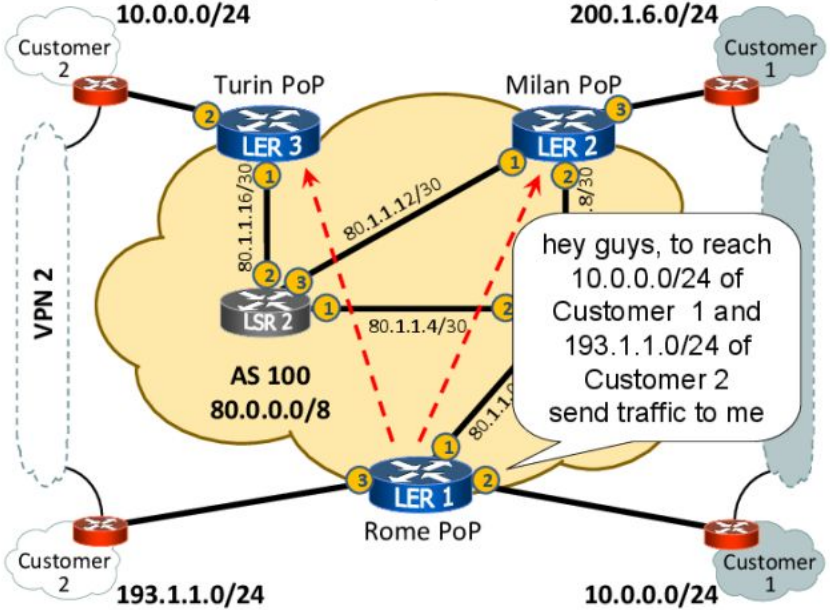


Move 2 – Use BGP to distribute customer prefixes

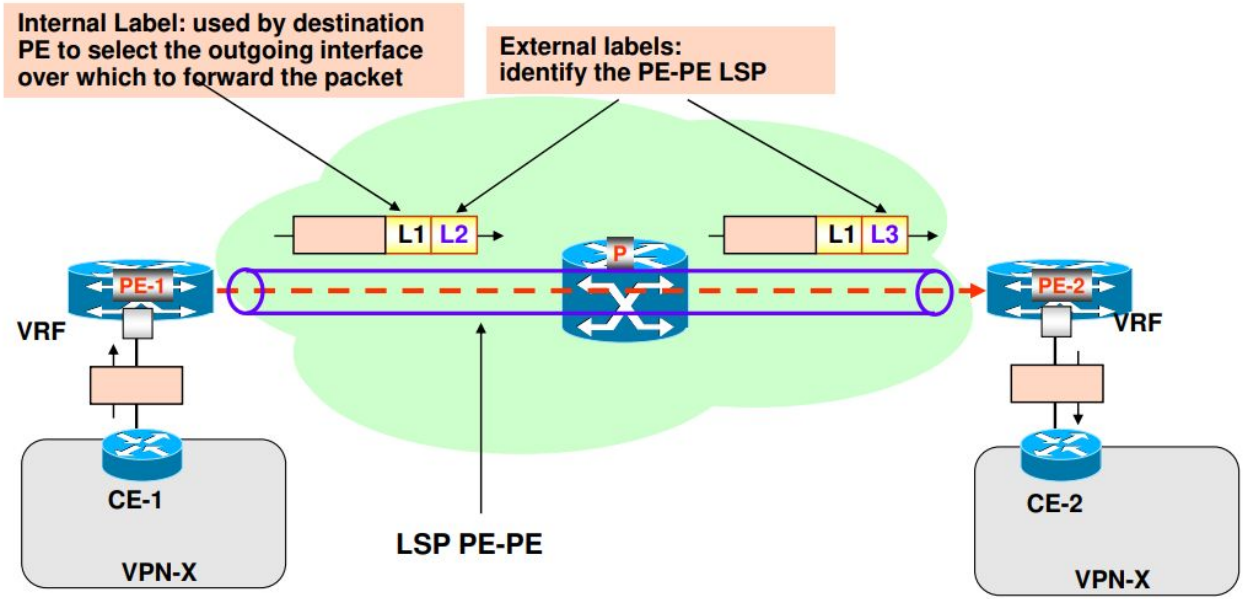
Border Gateway Protocol, it's a routing protocol that allows 2 nodes in different nets, to receive a BGP update msg that contains all the reachable prefixes passing through a specific router.

- Multi-Protocol BGP (MP-BGP) used as signaling protocol to distribute reachability information about customer prefixes
- MP-BGP treats VPNs as a separate address family
  - the concept of the "customer" (i.e., the "L3VPN identifier") is defined
- PE routers establish a full-mesh of iBGP peerings
  - a PE announces to all the other PEs the customer prefixes that it can reach via the CE router it is connected to

So the BGP update msg will contain also info about the owner of the specific address so it's possible to distinguish the addresses

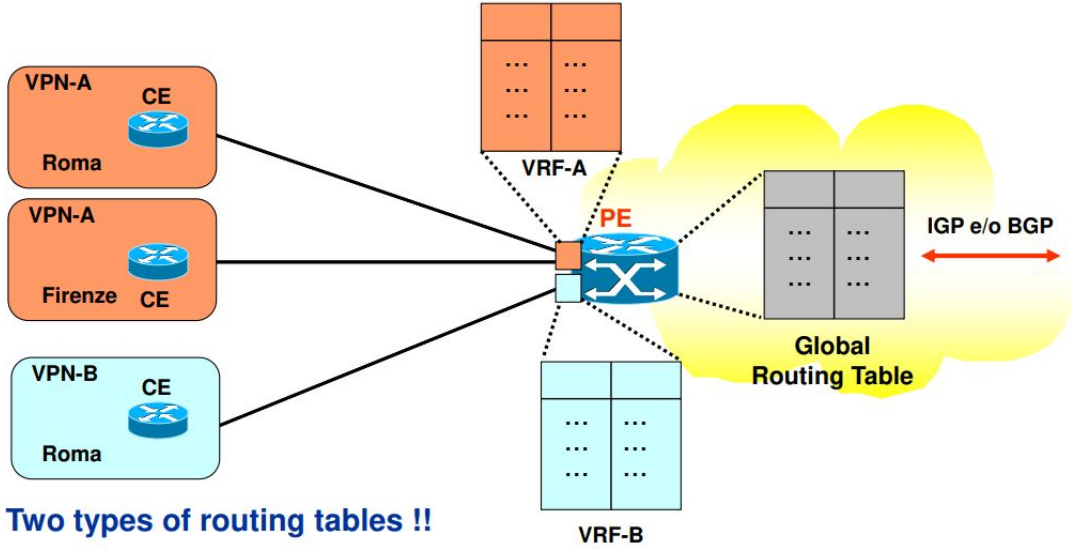


Move 3 – Use MPLS encapsulation among PEs



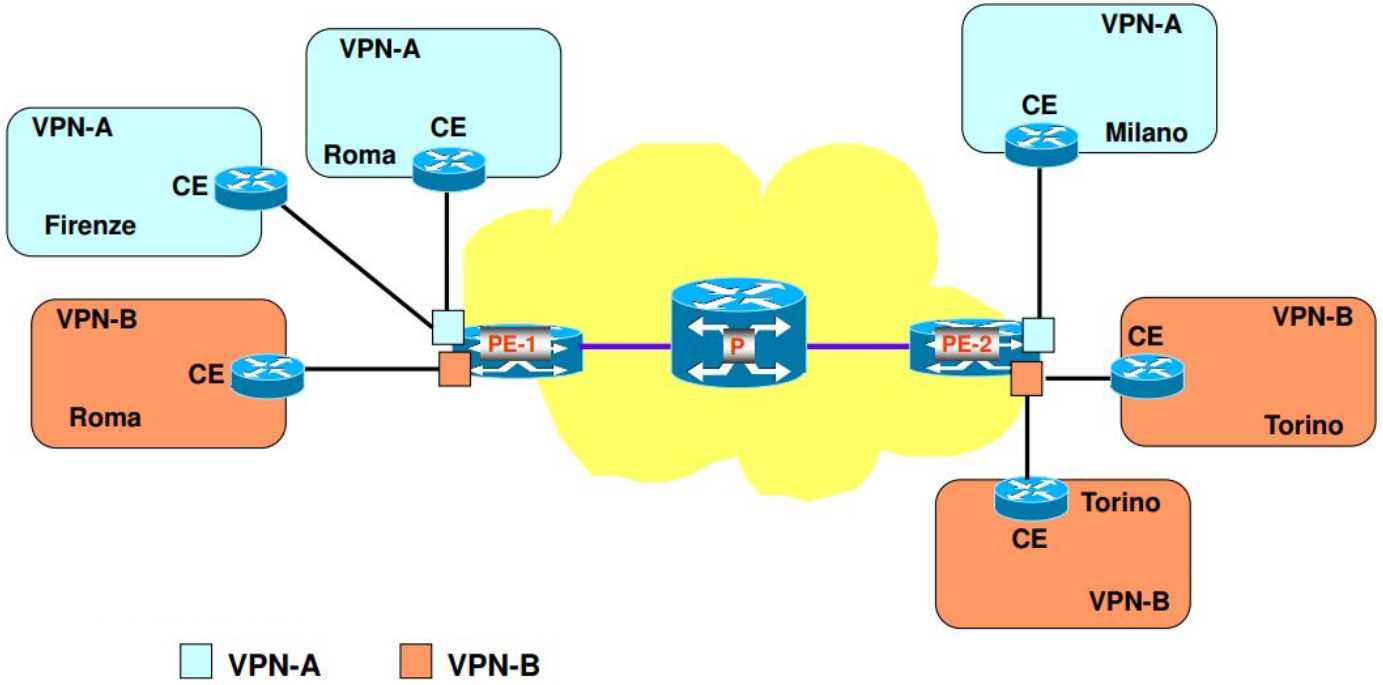
- The PE-PE LSPs can be “Hop-by-Hop” or “Explicitly Routed”

Routing Tables

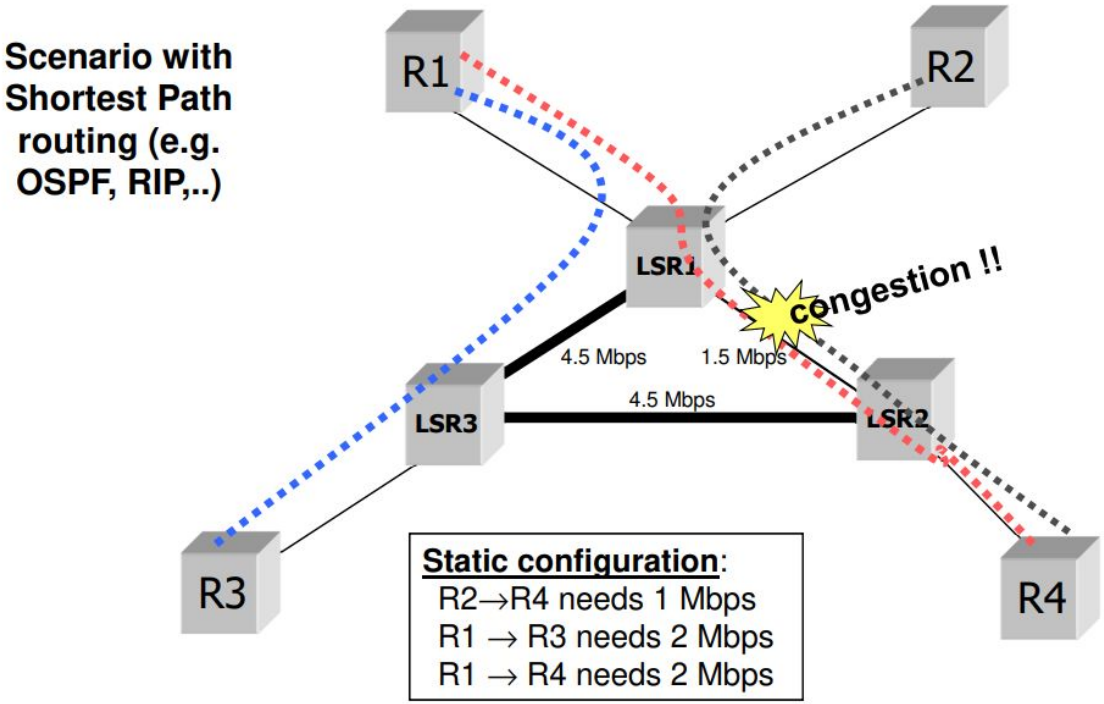


- Two types of routing tables:
  - Global routing table
  - VRF (VPN Routing & Forwarding)

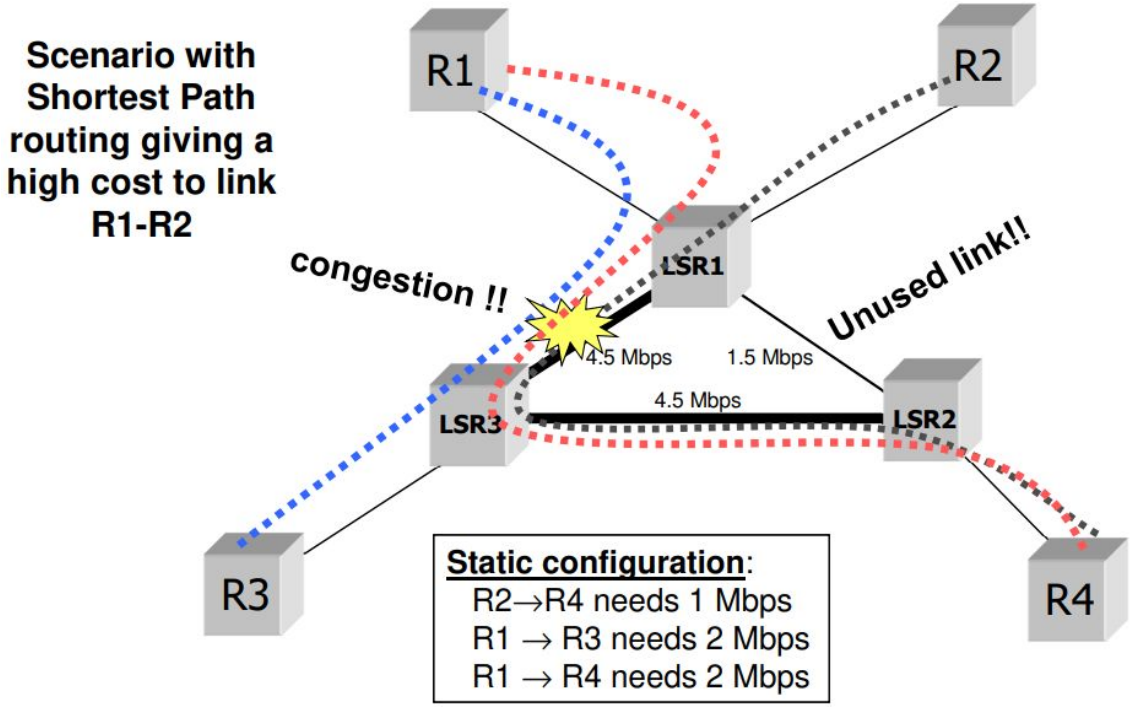
VPN Routing & Forwarding (VRF)



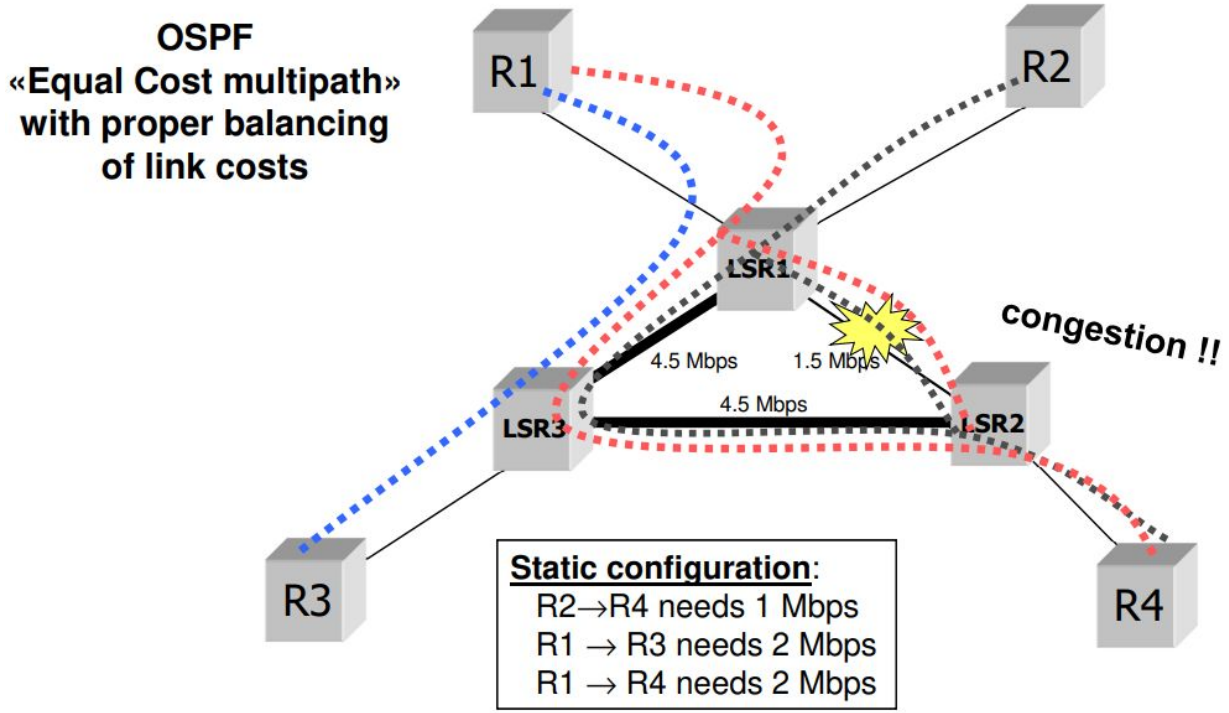
LSP for Traffic Engineering



LSP for Traffic Engineering



LSP for Traffic Engineering



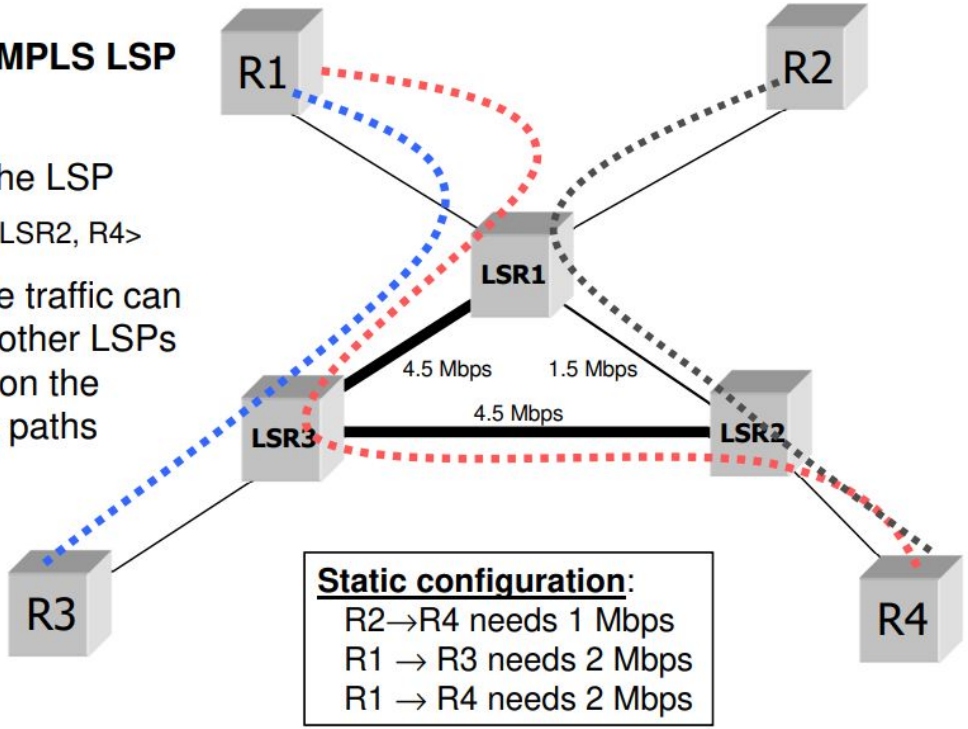
LSP for Traffic Engineering

Scenario with MPLS LSP

- R1->R4 traffic is tunneled in the LSP

<R1, LSR1, LSR3, LSR2, R4>

- the resto of the traffic can be assigned to other LSPs or it can be left on the default shortest paths



Static configuration:
R2→R4 needs 1 Mbps
R1 → R3 needs 2 Mbps
R1 → R4 needs 2 Mbps

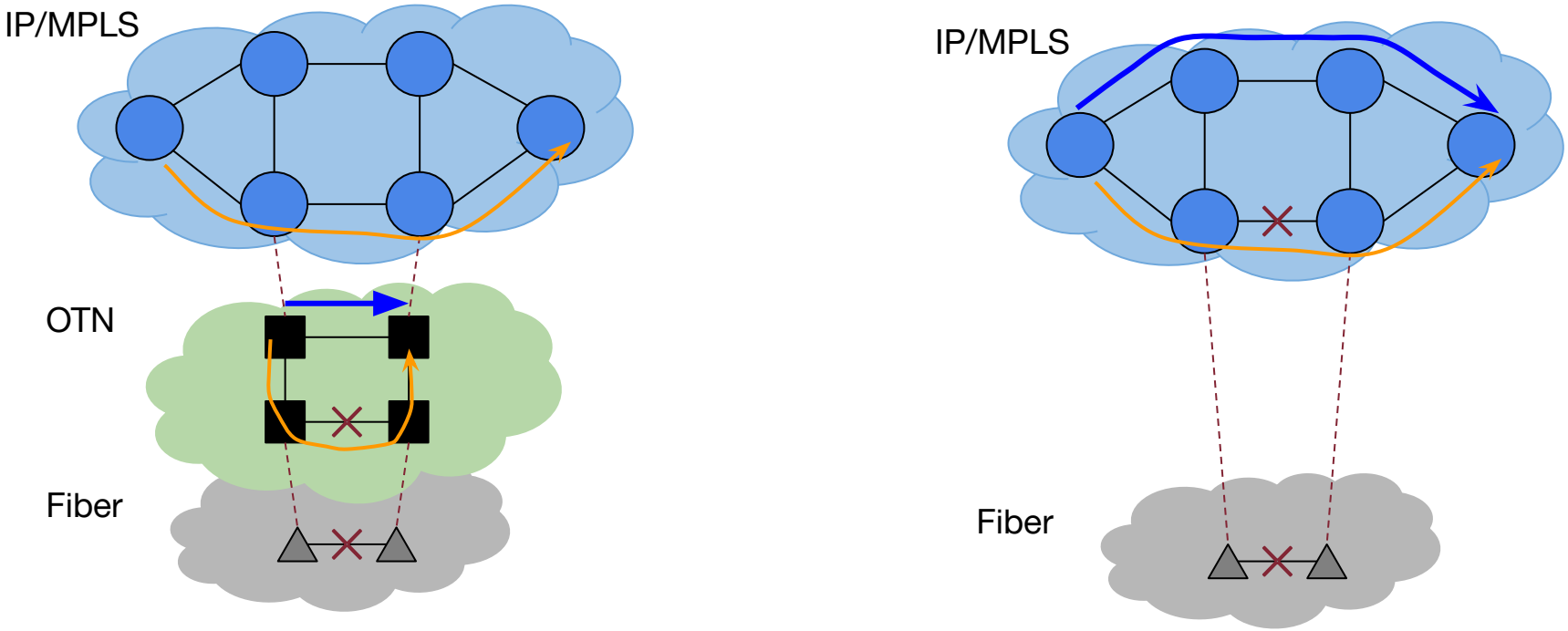
Traffic engineering with MPLS

- Who chooses the path?
  - An external entity chooses the path and then the routers are configured accordingly
  - The routers autonomously choose the path that follow some QoS constraint
- How to choose the path?
  - The path is chosen for each new LSP to be setup (“on-line”)
  - Global optimization (“off-line”) on all paths in the network

Protection mechanisms

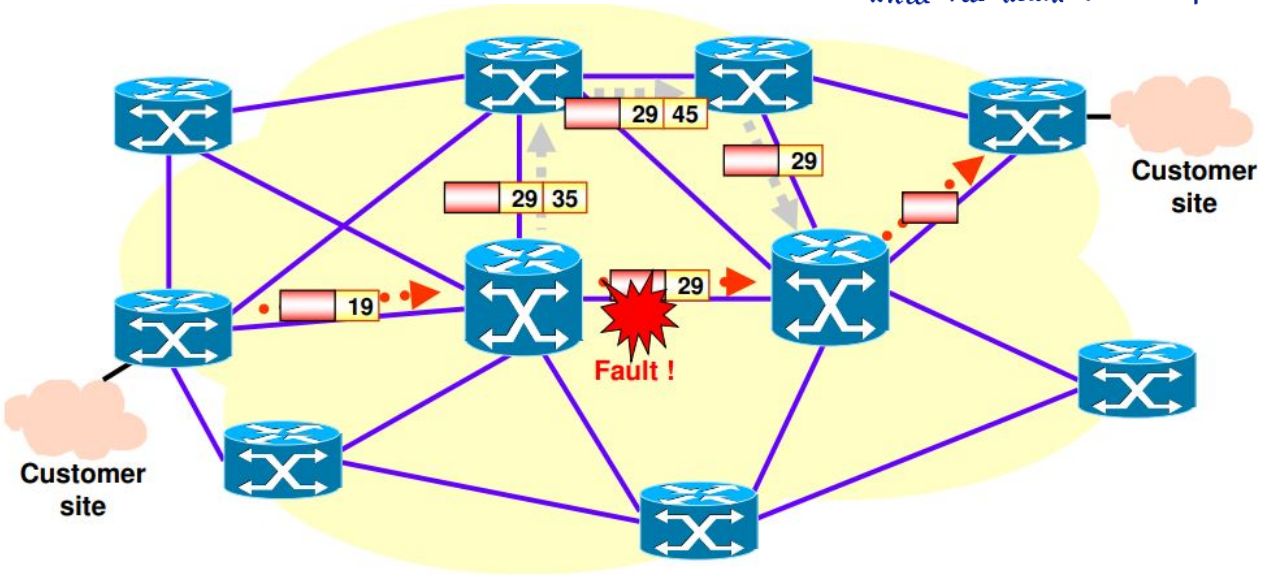
- A traditional IP network provides a failure recovery mechanism which is effective but slow (usually from seconds to tens of seconds)
  - This is based on dynamic routing protocols like OSPF and it is suitable for Best Effort traffic
- OTN network offer protection mechanism that operate with very short time scales (e.g. < 50 ms)
- MPLS can support protection mechanism
  - the goal is to provide performance (failure recovery time) similar to OTN ones

Automatic Protection Switching

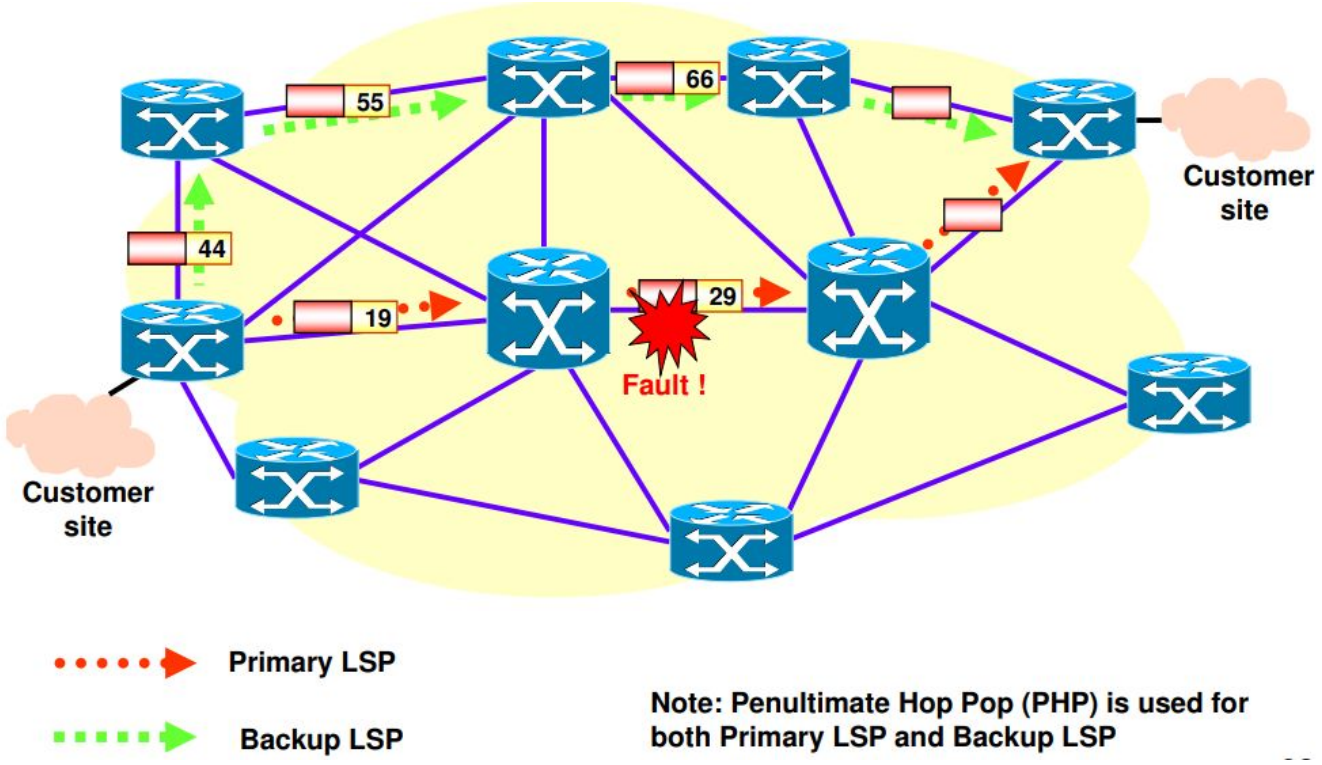


Fast rerouting – link protection

It's possible to perform it using the LSP trunk (MPLS multiple labels).
In a specific device it's added a PUSH action that pushes the label to follow the alternative path but just when the main one has problems.



backup LSP – path protection



Internet Protocol version 6 (IPv6)

Network Infrastructures A.Y. 2023/24

Outline

SAPIENZA
UNIVERSITÀ DI ROMA

- IPv6 basics
  - design principle
  - IPv6 header
  - Extension Header
  - Path MTU Discovery
- IPv6 Addressing
  - type of addresses
  - unicast
  - subnetting
- IPv6 Transition Mechanism
  - Dual Stack
  - Tunneling

Outline

SAPIENZA
UNIVERSITÀ DI ROMA

- IPv6 basics
  - design principle
  - IPv6 header
  - Extension Header
  - Path MTU Discovery
- IPv6 Addressing
  - type of addresses
  - unicast
  - subnetting
- IPv6 Transition Mechanism
  - Dual Stack
  - Tunneling

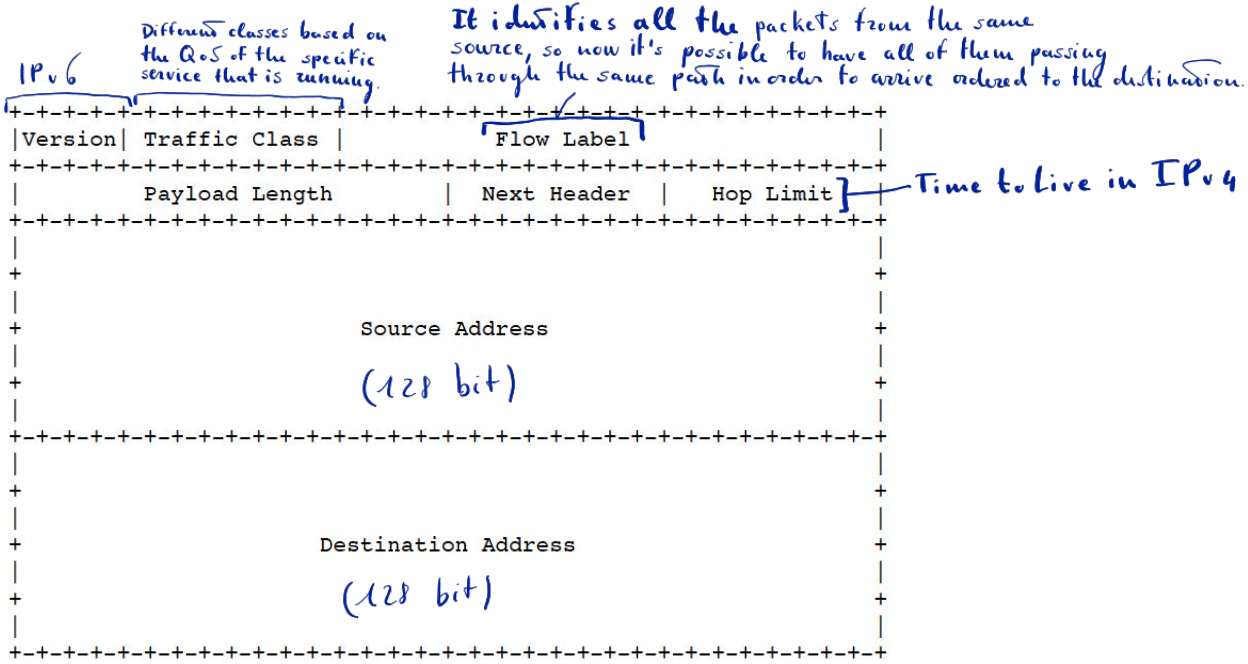
Design Principle

- IP version 6 (IPv6) is a new version of the Internet Protocol (IP), designed as the successor to IP version 4 (IPv4)
- The changes from IPv4 to IPv6 fall primarily into the following categories:
  - expanded addressing capabilities
  - header format simplification
  - improved support for extensions and options
  - flow labeling capability
  - authentication and privacy capabilities

IPv6 Header Format

SAPIENZA
UNIVERSITÀ DI ROMA

This is the basic header. It contains just the mandatory fields.



IPv6 Header Format

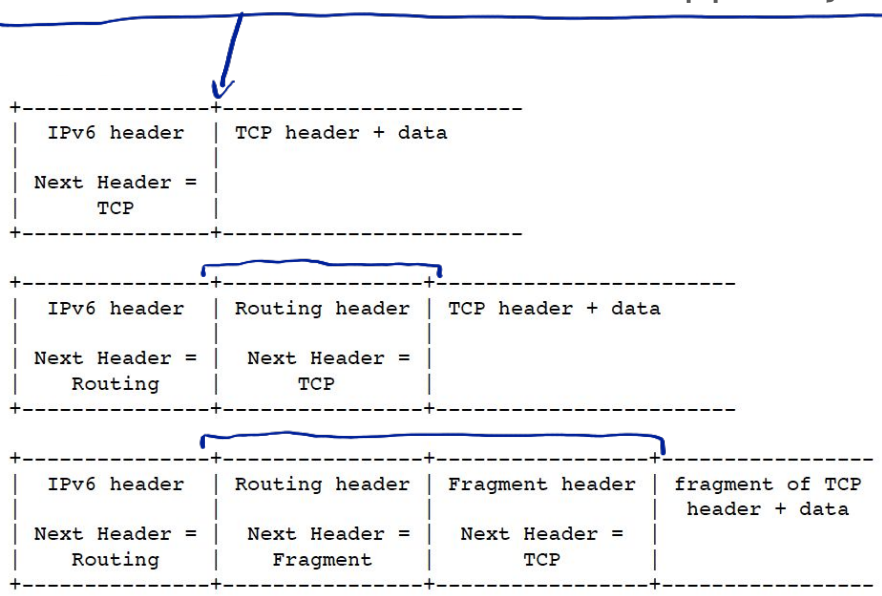
SAPIENZA
UNIVERSITÀ DI ROMA

- **Version**
  - (4-bit) Internet Protocol version number = 6
- **Traffic Class**
  - (8-bit) Traffic Class field
- **Flow Label**
  - (20-bit) flow label
- **Payload Length**
  - (16-bit) length of the IPv6 payload, i.e., the rest of the packet following this IPv6 header, in octets
- **Next Header**
  - (8-bit) identifies the type of header immediately following the IPv6 header
- **Hop Limit**
  - (8-bit) decremented by 1 by each node that forwards the packet. When forwarding, the packet is discarded if Hop Limit was zero when received or is decremented to zero
- **Source Address**
  - (128-bit) address of the originator of the packet
- **Destination Address**
  - (128-bit) address of the intended recipient of the packet (possibly not the ultimate recipient, if a Routing header is present)

IPv6 Extension Headers

SAPIENZA UNIVERSITÀ DI ROMA

- In IPv6, optional internet-layer information is encoded in separate headers that may be placed between the IPv6 header and the upper layer header in a packet



Processing an IPv6 packet

SAPIENZA
UNIVERSITÀ DI ROMA

- Extension headers are not processed, inserted, or deleted by any node along a packet's delivery path, until the packet reaches the node identified in the Destination Address field of the IPv6 header
- At the destination node, normal demultiplexing on the Next Header field of the IPv6 header invokes the module to process
  - the first extension header, or
  - the upper-layer header if no extension header is present
- The contents and semantics of each extension header determine whether or not to proceed to the next header
- Extension headers must be processed strictly in the order they appear in the packet

(some) Extension Headers

SAPIENZA
UNIVERSITÀ DI ROMA

- A full implementation of IPv6 includes implementation of the following extension headers:
  - Hop-by-Hop Options → it's an option that's processed in every node the packet passes through.
  - Fragment
  - Destination Options
  - Routing
  - Authentication → authentication of a packet.
  - Encapsulating Security Payload → encryption of a packet's payload.

Every application that wants to use IPv6 must implement the functionalities to elaborate these 2 services.

Extension Header Order

SAPIENZA UNIVERSITÀ DI ROMA

- When more than one extension header is used in the same packet, it is recommended that those headers appear in the following order:

1. IPv6 header
2. Hop-by-Hop Options header
3. Destination Options header
4. Routing header
5. Fragment header
6. Authentication header
7. Encapsulating Security Payload header
8. Destination Options header
9. Upper-Layer header

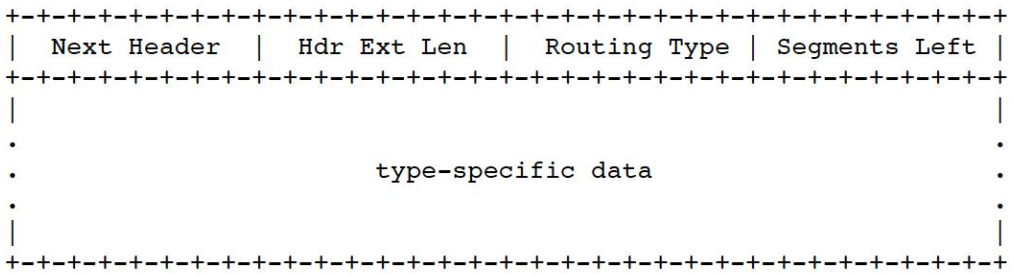
it's the only one that could appear twice.

The intermediate routers don't check the optional headers by default, because that's done just by the destination. If this header is specified then also every devices could check them.

Routing Header

SAPIENZA
UNIVERSITÀ DI ROMA

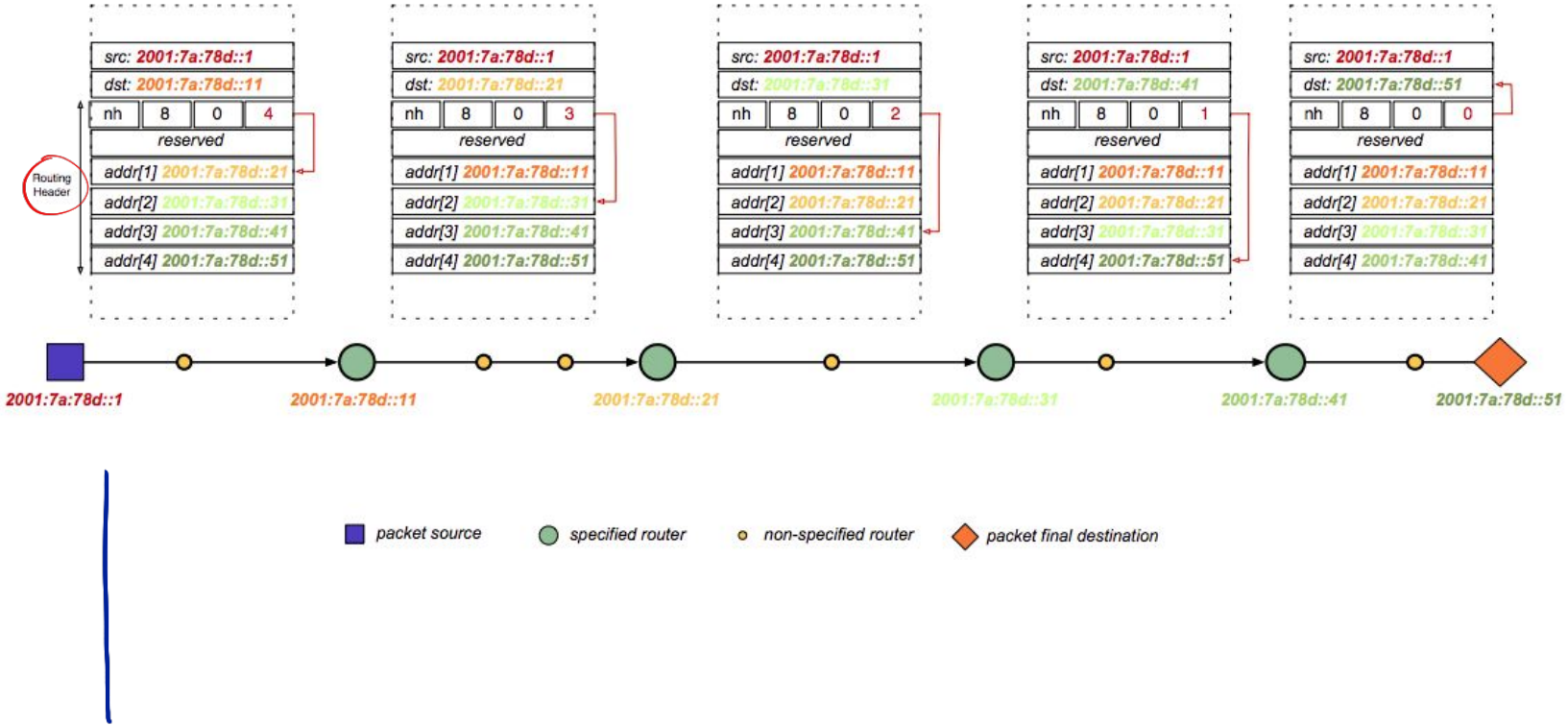
- The Routing header is used by an IPv6 source to list one or more intermediate nodes to be "visited" on the way to a packet's destination



- **Routing Type**
  - (8-bit) identifier of a particular Routing header variant
- **Segments Left**
  - (8-bit) number of explicitly listed intermediate nodes still to be visited before reaching the final destination

Type 0 Routing Header

SAPIENZA
UNIVERSITÀ DI ROMA



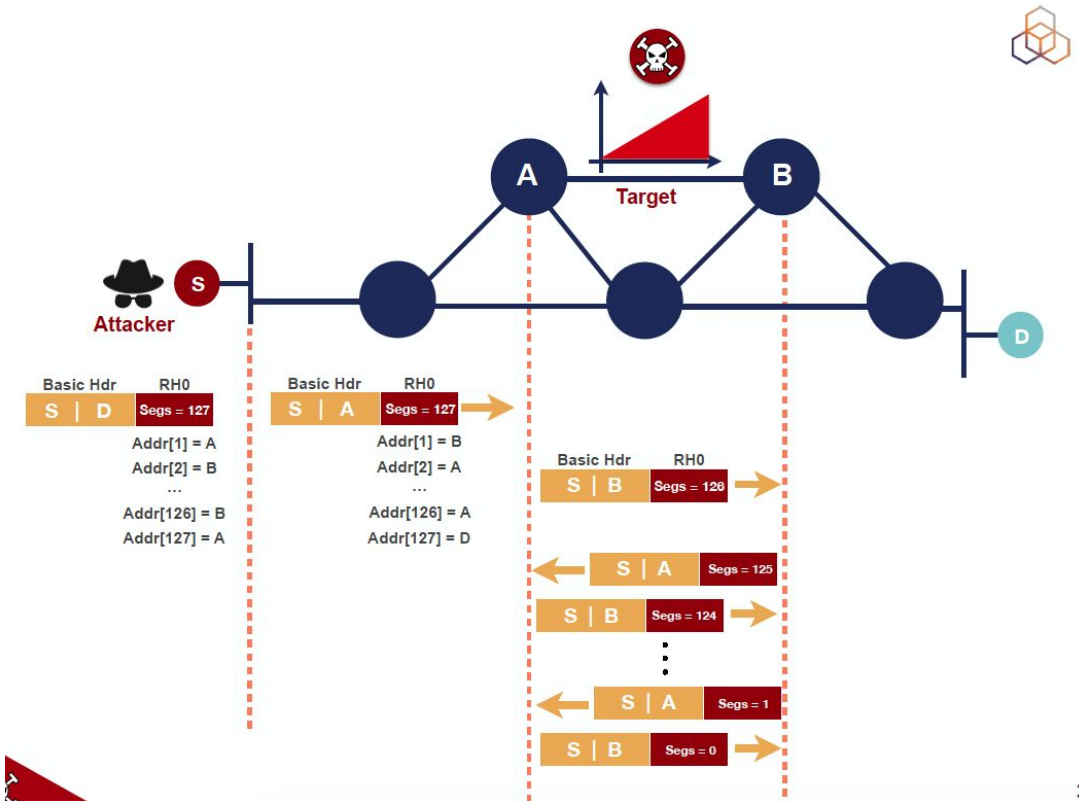
In order to obtain the explicit routing, it's possible to "cheat".
Instead of inserting, in the message's destination, the actual one, the field'll be filled with the mid-point router. Following this way we're sure about the path that the message is going to take.

Type 0 Routing Header

- A single RH0 may contain multiple intermediate node addresses, and the same address may be included more than once in the same RH0
- This allows a packet to be constructed such that it will oscillate between two RH0-processing hosts or routers many times
- This allows a stream of packets from an attacker to be amplified along the path between two remote routers, which could be used to cause congestion along arbitrary remote paths and hence act as a denial-of-service mechanism

Type 0 Routing Header

SAPIENZA
UNIVERSITÀ DI ROMA



Fragment Header

SAPIENZA
UNIVERSITÀ DI ROMA

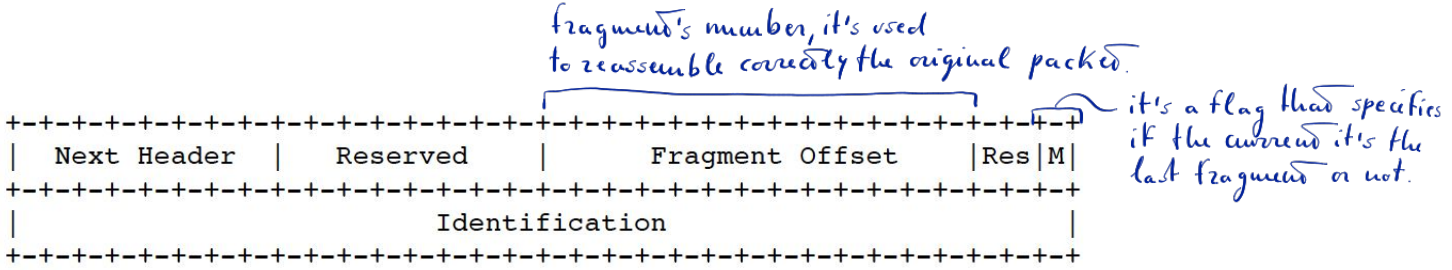
- The Fragment header is used by an IPv6 source to send a packet larger than would fit in the path MTU to its destination
- Unlike IPv4, fragmentation in IPv6 is performed only by source nodes, not by routers along a packet's delivery path
  - **pros:** less processing, less delay, overhead reduction
  - **cons:** security threats, need of path MTU discovery procedure
- IPv6 attempts to minimise the use of fragmentation
  - by minimising the supported MTU size
  - by allowing only the hosts to fragment datagrams
- IPv6 requires that every link in the Internet have an MTU of 1280 octets or greater

MTU

It's the maximum transmission unit.
Usually Internet's MTU is 1500 Byte.
If a message is bigger than network MTU, then fragmentation is performed (only by source nodes).
When all the fragments of a packet reach the destination, they're reassembled.

Fragment Header (similar to IPv4)

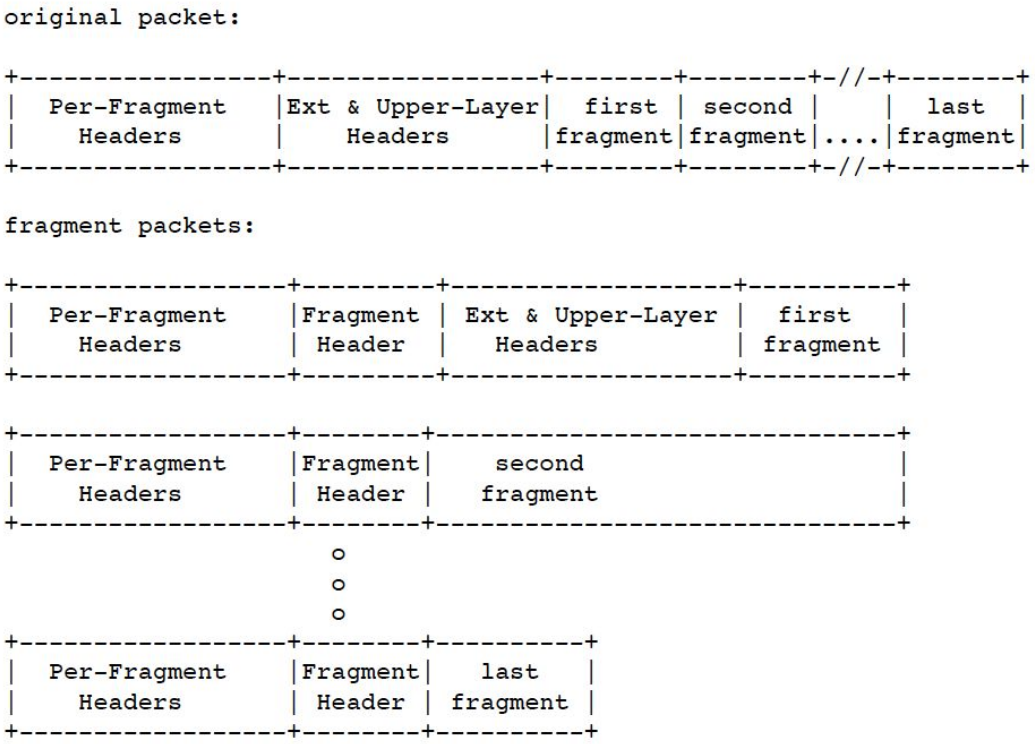
SAPIENZA
UNIVERSITÀ DI ROMA



- **Fragment Offset** defines the offset, in 8-octet units, of the data following this header relative to the start of the Fragmentable Part of the original packet
- **M flag** is a bit set to 1 when more fragments will follow or 0 if this is the last fragment
- **Identification** defines the fragments which belong to the same packet
    - this number must be different than that of any other fragmented packet sent recently with the same Source Address and Destination Address

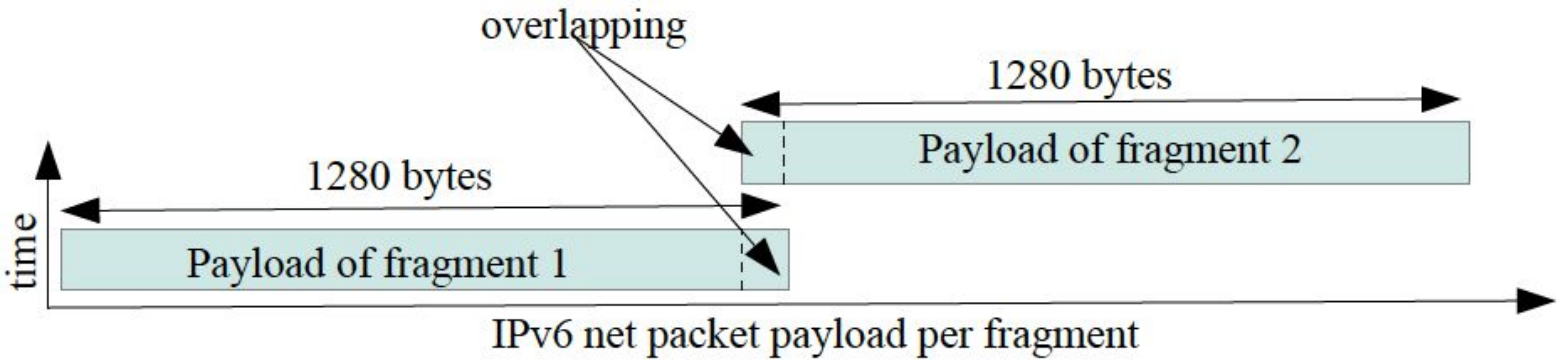
Fragmentation

SAPIENZA
UNIVERSITÀ DI ROMA



Security Threats related to Fragmentation

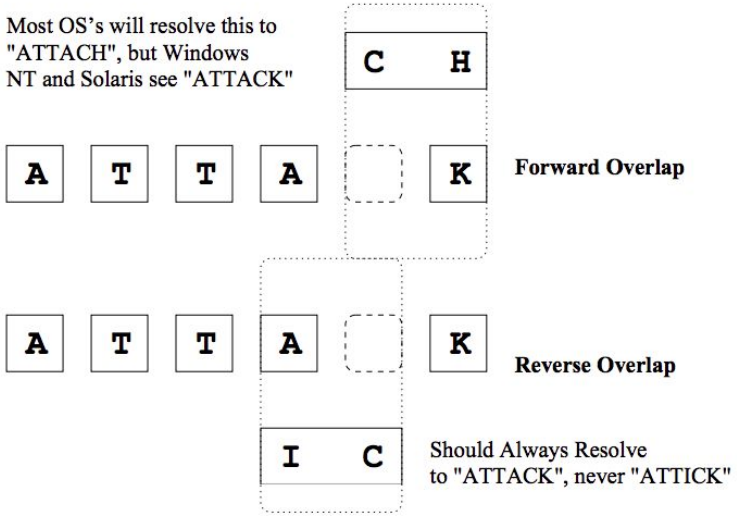
SAPIENZA
UNIVERSITÀ DI ROMA



Security Threats related to Fragmentation

SAPIENZA
UNIVERSITÀ DI ROMA

- Fragmentation overlap occurs when fragments of differing sizes arrive out of order and in overlapping positions
- Reconstruction process in different hosts might generate different packets
- This presents problems for an IDS
- An attacker that understands the specific inconsistency between an end system and an IDS can obscure her attack



That was a problem for the early IPv6 standardization because of the fact there wasn't a unique rule to follow, but everyone could solve it by its own.

There is a network function called IDS (Intrusion Detection System).
It has the job to check the incoming msgs with a DB of malicious msgs, if there is a match it raises an alert.

Packet Size Issues

- IPv6 requires that every link in the Internet have an MTU of 1280 octets or greater
- On any link that cannot convey a 1280-octet packet in one piece, link specific fragmentation and reassembly must be provided at a layer below IPv6
- It is strongly recommended that IPv6 nodes implement Path MTU Discovery, in order to discover and take advantage of path MTUs greater than 1280 octets
- A minimal IPv6 implementation may simply restrict itself to sending packets no larger than 1280 octets, and omit implementation of Path MTU Discovery

Path MTU Discovery

SAPIENZA
UNIVERSITÀ DI ROMA

- Path MTU (PMTU) is equal to the minimum link MTU of all the links in a path

- When one IPv6 node has a large amount of data to send to another node, the data is transmitted in a series of IPv6 packets
- These packets can have a size
  - less than or equal to the PMTU
  - larger than PMTU
    - fragmentation is required

Path MTU Discovery

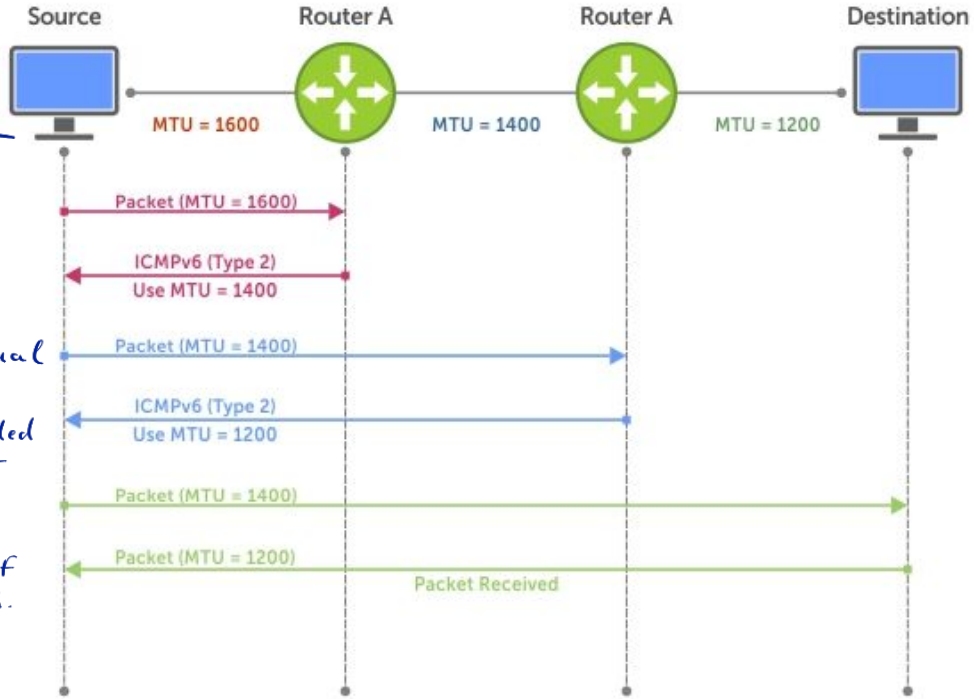
- It is preferable that a source node sends packets having the largest size that can successfully traverse the path to the destination without the need for IPv6 fragmentation
- **Problem**
  - Source node does not know the PMTU
- **Solution**
  - Path MTU Discovery
- PMTUD it is an iterative procedure based on ICMP Packet Too Big messages that allows the source to discover the PMTU

Path MTU Discovery

SAPIENZA
UNIVERSITÀ DI ROMA

In IPv6 only the source nodes are allowed to fragment messages.
This is the better solution because:

- it reduces the computational time in every router.
- if fragmentation is not needed the fragmentation bits are not carried.
- in that way, it's possible to create the smallest amount of packets reducing the overhead.



Outline

SAPIENZA
UNIVERSITÀ DI ROMA

- IPv6 basics
  - design principle
  - IPv6 header
  - Extension Header
  - Path MTU Discovery
- IPv6 Addressing
  - type of addresses
  - unicast
  - subnetting
- IPv6 Transition Mechanism
  - Dual Stack
  - Tunneling

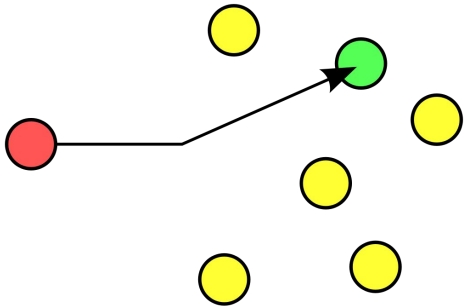
IPv6 Address

SAPIENZA
UNIVERSITÀ DI ROMA

- IPv6 addresses are 128-bit identifiers for interfaces and sets of interfaces
- There are three types of addresses:
  - **Unicast**
    - An identifier for a single interface
    - A packet sent to a unicast address is delivered to the interface identified by that address
  - **Anycast**
    - An identifier for a set of interfaces (typically belonging to different nodes)
    - A packet sent to an anycast address is delivered to one of the interfaces identified by that address (the "nearest" one)
  - **Multicast**
    - An identifier for a set of interfaces
    - A packet sent to a multicast address is delivered to all interfaces identified by that address
- There are no broadcast addresses in IPv6

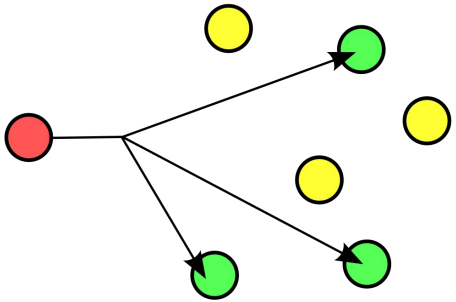
IPv6 Addresses

SAPIENZA
UNIVERSITÀ DI ROMA



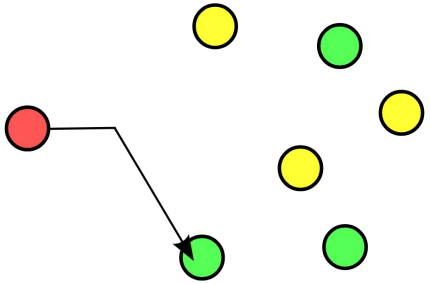
Unicast

Link Local
Global Unicast



Multicast

(special case: Broadcast)



Anycast

(i.e. DNS servers, you need the answer just from one of them, the first one that answer the request)

Text Representation of Addresses

SAPIENZA UNIVERSITÀ DI ROMA

- The general form for an IPv6 address is x:x:x:x:x:x:x:x
  - 'x's are one to four hexadecimal digits of the eight 16-bit pieces of the address
- **Examples**

  ABCD:EF01:2345:6789:ABCD:EF01:2345:6789

  2001:DB8:0:0:8:800:200C:417A

  The leading zeros in every piece are removed.

  - it is not necessary to write the leading zeros in an individual field
  - there must be at least one numeral in every field

Address Compression

SAPIENZA
UNIVERSITÀ DI ROMA

- It is common for addresses to contain long strings of zero bits
  - the use of "::" indicates one or more groups of 16 bits of zeros
  - the "::" can only appear once in an address → this is because otherwise it's impossible to know how many groups are missing from where.

2001:DB8:0:0:8:800:200C:417	2001:DB8::8:800:200C:417	*a unicast address*
FF01:0:0:0:0:0:0:101	FF01::101	*a multicast address*
0:0:0:0:0:0:0:1	::1	self reference to the internal stack [*the loopback address*
0:0:0:0:0:0:0:0	::	*the unspecified address*

IPv6 Address Prefix → 48 bits

SAPIENZA
UNIVERSITÀ DI ROMA

- An IPv6 address prefix is represented by the notation:

  ipv6-address/prefix-length

  - **ipv6-address** is an IPv6 address
  - **prefix-length** is a decimal value specifying how many of the leftmost contiguous bits of the address comprise the prefix
- **Examples**
  - 2001:0DB8:0000:CD30:0000:0000:0000:0000/60
  - 2001:0DB8::CD30:0:0:0:0/60
  - 2001:0DB8:0:CD30::/60

The smallest IPv6 network has a /64 subnet mask.

Prefix Representation

SAPIENZA UNIVERSITÀ DI ROMA

2001:0DB8:0000:CD30:0000:0000:0000:0000/60

Prefix

prefix length (in bits)

- the following are NOT legal representations of the above prefix

zero's missing

2001:0DB8:0:CD3/60

2001:0DB8::CD30/60 -> 2001:0DB8:0000:0000:0000:0000:0000:CD30

2001:0DB8::CD3/60 -> 2001:0DB8:0000:0000:0000:0000:0000:0CD3

Address Type Identification

SAPIENZA
UNIVERSITÀ DI ROMA

Address type	Binary prefix	IPv6 notation
Unspecified	00...0 (128 bits)	::/128
Loopback	00...1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-Local unicast	1111111010	FE80::/10
Global Unicast	(everything else)	

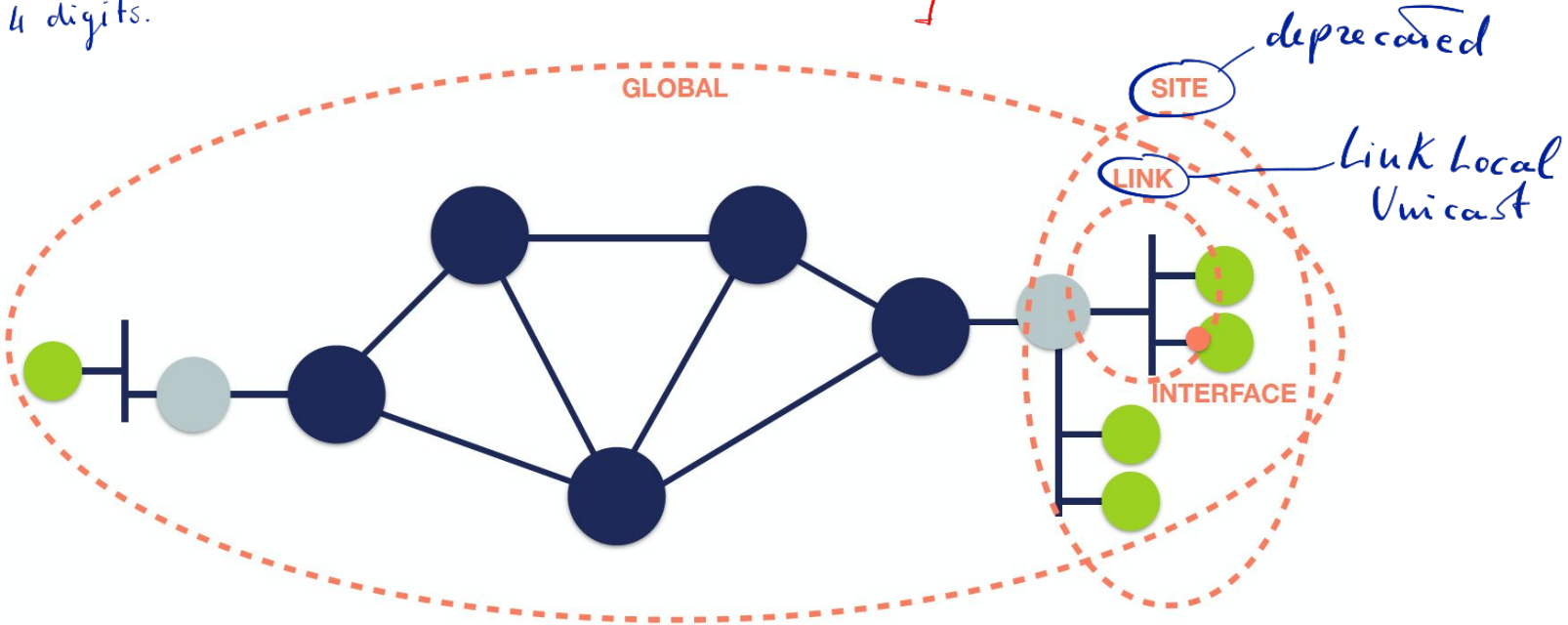
- Anycast addresses are taken from the unicast address spaces

It allows to communicate with other local (physically connected) addresses.

Address Scope

SAPIENZA
UNIVERSITÀ DI ROMA

It's possible to determine the address scope simply observing the first 4 digits.

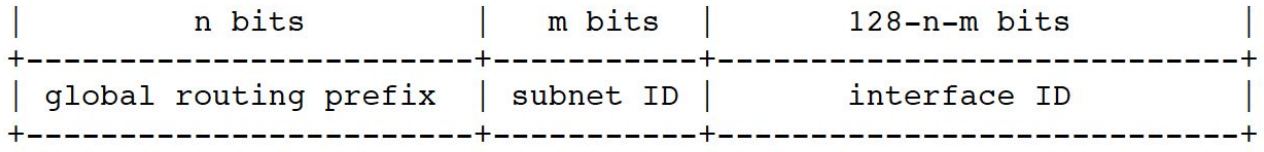


Global Unicast Addresses

to communicate with other devices outside from local network.

SAPIENZA UNIVERSITÀ DI ROMA

- The general format for **IPv6 Global Unicast** addresses is as follows:

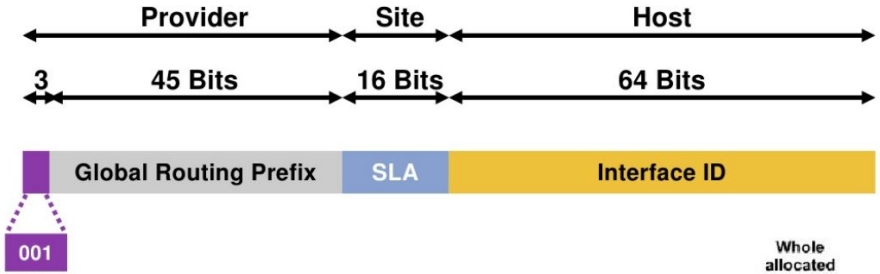


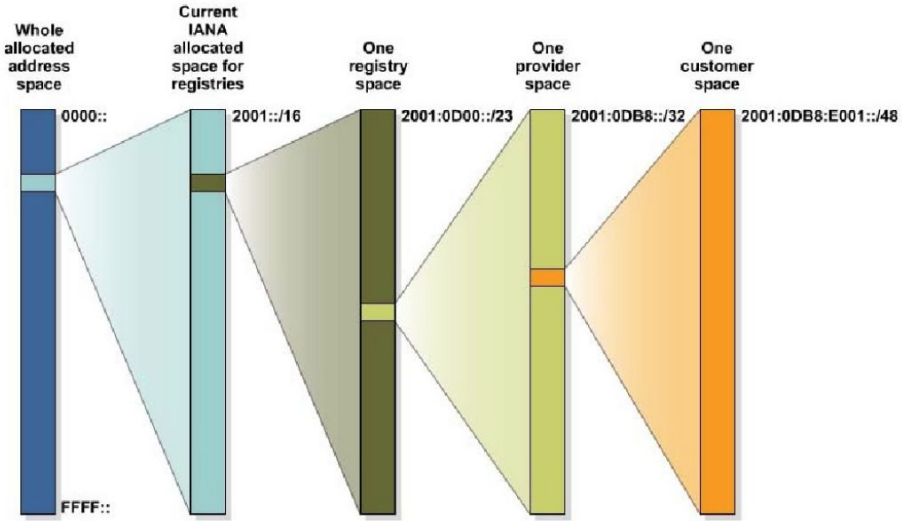
  where:

  - the global routing prefix is a value assigned to a site
  - the subnet ID is an identifier of a link within the site
  - interface ID is determined starting from the link layer address
- All Global Unicast addresses other than those that start with binary 000 have a 64-bit interface ID field (i.e., n + m = 64)

Global Unicast Addresses

SAPIENZA
UNIVERSITÀ DI ROMA



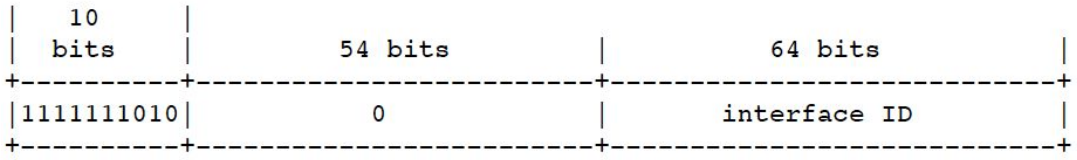


Link-Local IPv6 Unicast Addresses

SAPIENZA UNIVERSITÀ DI ROMA

- **Link-Local addresses** are designed to be used for addressing on a single link for purposes such as
  - automatic address configuration
  - neighbor discovery
  - when no routers are present

to communicate with devices in the same local network.



- Routers must not forward any packets with Link-Local source or destination addresses to other links

IPv6 Subnetting - Overview

SAPIENZA
UNIVERSITÀ DI ROMA

- Subnetting with IPv6 is not drastically different than subnetting with IPv4, we just need to keep a few things in mind:
  - Each character in an IPv6 address represents 4 bits
    - Since 0xF is 1111 in binary, it's easy to fall back into an IPv4 habit and forget that 0x11 is actually 0001 0001 in binary
  - Each IPv6 set represent 16 bits (4 characters at 4 bits each)
    - Keeping this in mind can make breaking up subnets a bit easier
  - Once it's in binary nothing changes!

https://community.cisco.com/t5/networking-documents/ipv6-subnetting-overview-and-case-study/ta-p/3125702

Setting the Ground Rules

SAPIENZA
UNIVERSITÀ DI ROMA

- The leading practice is to receive at least a /48 prefix from an ISP
  - This leaves you with 2^{80} bits to manipulate (128 bit address - 48 bits that can't be changed = 80 bits to use)
    - More bits than the entire IPv4 address space!
- According to RFC 4291 the current recommended smallest prefix is a /64
- With so many addresses in IPv6 there isn't the same need for address conservation as there is in IPv4
  - You can assign a /64 to a point-to-point link and not feel guilty
- This gives us one block of hex digits, or 16-bits, to use for subnetting
  - One block might not sound like much, but 16-bits is half of the entire IPv4 address space

Defining the Site ID

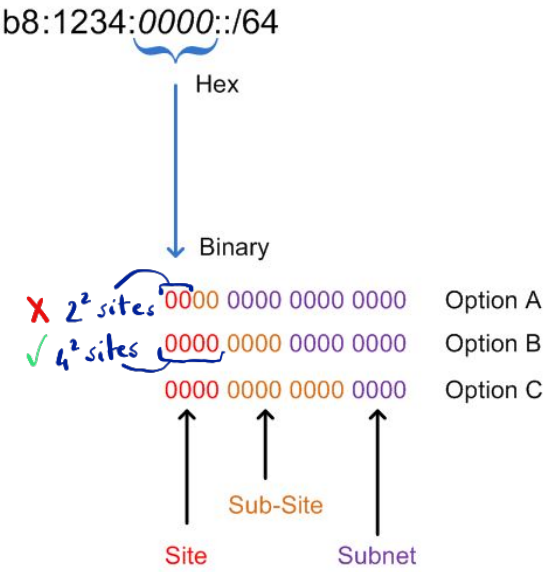
SAPIENZA UNIVERSITÀ DI ROMA

- In order to allow for proper route aggregation and summarization you should define Site IDs that you can use at each location (an office, data center or geographic region)
- This is where we need to define at least a Site ID, and possibly a sub-Site ID

big area

region within a site (check the map)

Example



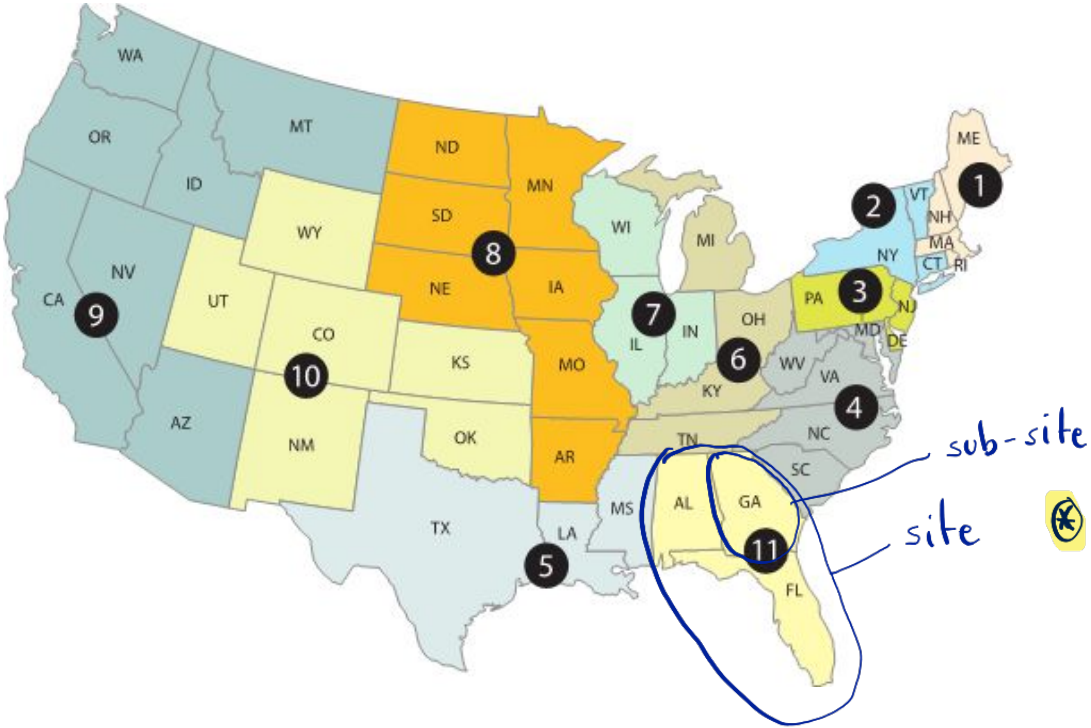
IPv6 Subnetting Case Study

SAPIENZA
UNIVERSITÀ DI ROMA

- Let’s consider a mid-sized company with offices and data centers across the United States
- **Assigned block:** 2001:db8:abcd::/48
  - We need to allocate this across our enterprise
- We have branches in most states
  - let’s use Option B, giving us 16 sites, 16 sub-sites and 256 subnets per site
    - "site" == geographic region of the country
    - “sub-site” == a city within the geographic region

IPv6 Subnetting Case Study

SAPIENZA
UNIVERSITÀ DI ROMA



The sites that we are rolling IPv6 to are in:

- San Francisco (Site 9)
- Seattle (Site 9)
- Omaha (Site 8)
- Newark (Site 3)
- New York City (Site 2)
- Boston (Site 1)

Site Addresses (in the example above)

SAPIENZA
UNIVERSITÀ DI ROMA

- Site 0 - 2001:db8:abcd:0000::/52 (for future use)
- Site 1 - 2001:db8:abcd:1000::/52
- Site 2 - 2001:db8:abcd:2000::/52
- Site 3 - 2001:db8:abcd:3000::/52
- ...
- Site 8 - 2001:db8:abcd:8000::/52
- Site 9 - 2001:db8:abcd:9000::/52
- Site 10 - 2001:db8:abcd:a000::/52 (for future use)
- Site 11 - 2001:db8:abcd:b000::/52 (for future use)
- Site 12 - 2001:db8:abcd:c000::/52 (for future use)

Sub-site Addresses

SAPIENZA
UNIVERSITÀ DI ROMA

- site 1
  - Boston - 2001:db8:abcd:1100::/56
  - others for future use
- site 2
  - New York City - 2001:db8:abcd:2000::/56
- site 3
  - Newark - 2001:db8:abcd:3f00::/56
- site 8
  - Omaha - 2001:db8:abcd:8000::/56
- site 9
  - San Francisco - 2001:db8:abcd:9100::/56
  - Seattle - 2001:db8:abcd:9200::/56

site number it's the highlighted one in the addresses.

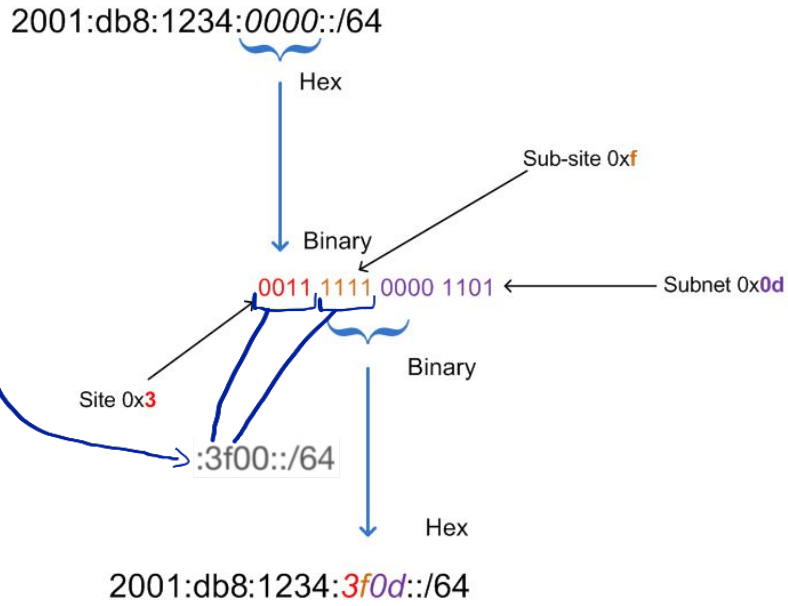
subsites for different cities

Subnet Addresses

SAPIENZA UNIVERSITÀ DI ROMA

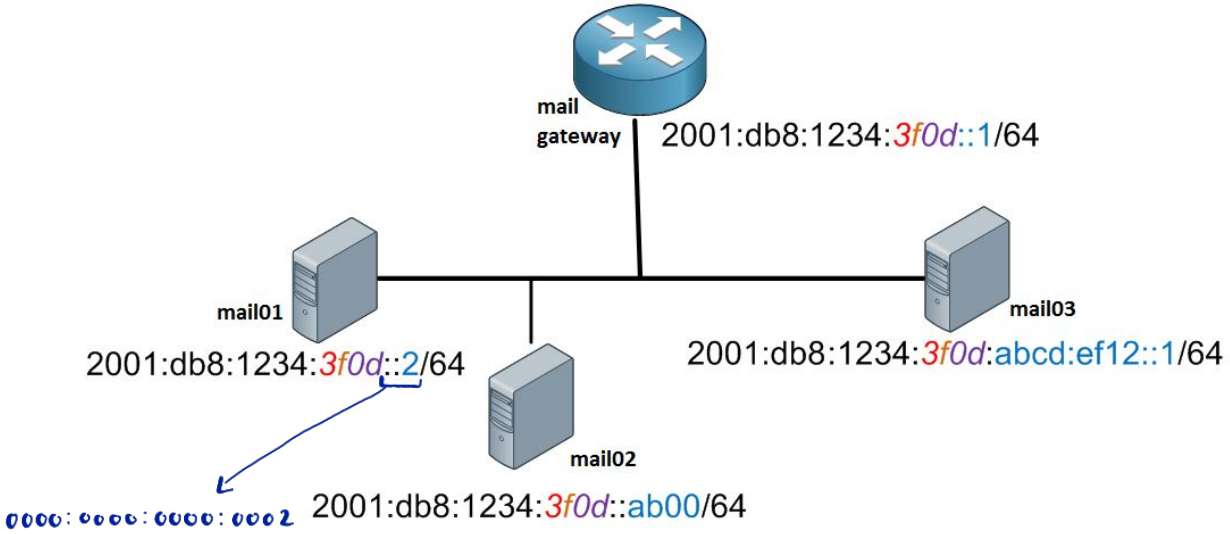
We will use our Newark site as an example

- Firewall Outside: 2001:db8:abcd:3f00::/64
- Webservers: 2001:db8:abcd:3f01::/64
- Database Servers: 2001:db8:abcd:3f02::/64
- ....
- Mail Servers: 2001:db8:abcd:3f0d::/64
- ....
- Management: 2001:db8:abcd:3fee::/64
- Loopbacks: 2001:db8:abcd:3fff::/64



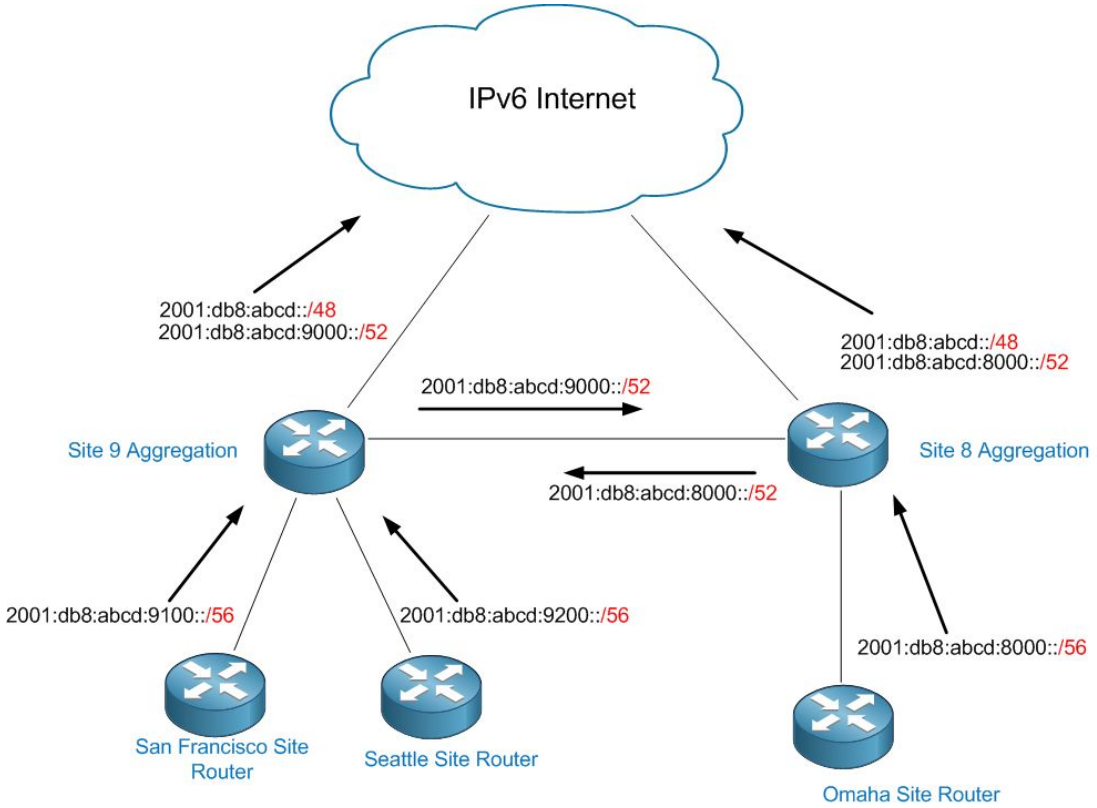
Assigning Addresses to hosts

SAPIENZA
UNIVERSITÀ DI ROMA



Mail Servers: 2001:db8:1234:3f0d::/64

Routing


IPv6 Internet
2001:db8:abcd::/48
2001:db8:abcd:9000::/52
2001:db8:abcd::/48
2001:db8:abcd:8000::/52
2001:db8:abcd:9000::/52
Site 9 Aggregation
Site 8 Aggregation
2001:db8:abcd:8000::/52
2001:db8:abcd:9100::/56
2001:db8:abcd:9200::/56
2001:db8:abcd:8000::/56
San Francisco Site Router
Seattle Site Router
Omaha Site Router

Outline

SAPIENZA
UNIVERSITÀ DI ROMA

- IPv6 basics
  - design principle
  - IPv6 header
  - Extension Header
  - Path MTU Discovery
- IPv6 Addressing
  - type of addresses
  - unicast
  - subnetting
- IPv6 Transition Mechanism
  - Dual Stack
  - Tunneling

Overview

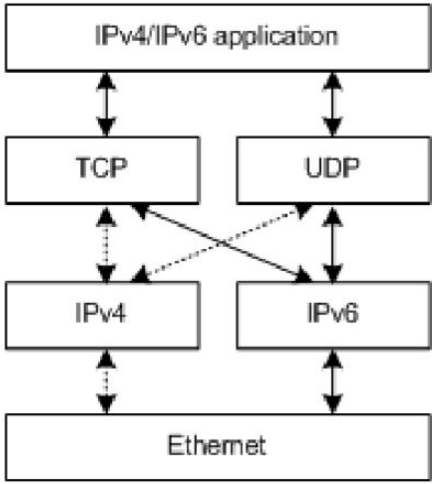
SAPIENZA
UNIVERSITÀ DI ROMA

- The key to a successful IPv6 transition is compatibility with the large installed base of IPv4 hosts and routers
- Two mechanism to manage transition
  - **dual IP layer** (also known as dual stack) → in order to adapt communication with other hosts.
    - complete support for both IPv4 and IPv6 in hosts and routers
  - **tunneling** of IPv6 over IPv4 → in order to communicate with other hosts through a different IP version.
    - establishing point-to-point tunnels by encapsulating IPv6 packets within IPv4

Dual IP Layer Operation

SAPIENZA UNIVERSITÀ DI ROMA

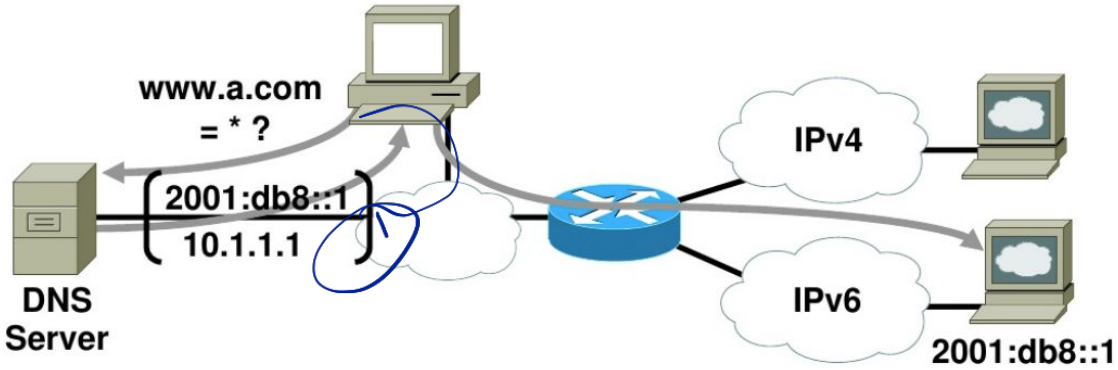
- The most straightforward way for IPv6 nodes to remain compatible with IPv4-only nodes is by providing a complete IPv4 implementation (IPv6/IPv4 nodes)
  - Address Configuration
    - nodes may be configured with both IPv4 and IPv6 addresses (e.g., DHCPv4 + SLAAC)
  - DNS
    - A new resource record type named "AAAA" has been defined for IPv6 addresses
    - IPv6/IPv4 nodes must provide resolver libraries capable of dealing with IPv4 "A" records as well as IPv6 "AAAA" records



By default the DNS doesn't know IPv6 addresses.

Dual IP Layer Operation

SAPIENZA
UNIVERSITÀ DI ROMA



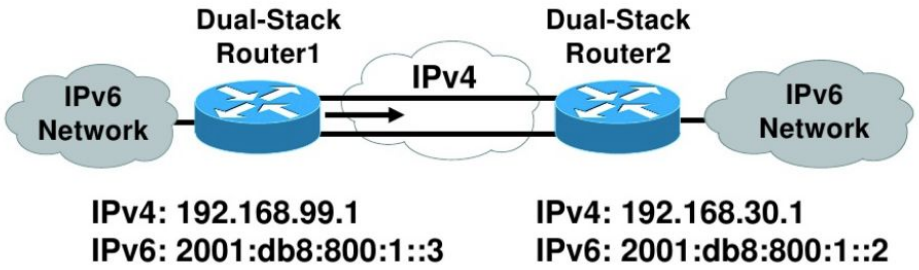
Configured Tunneling Mechanisms

SAPIENZA
UNIVERSITÀ DI ROMA

- Tunneling provides a way to utilize an existing IPv4 routing infrastructure to carry IPv6 traffic
  - while the IPv6 infrastructure is being deployed, the existing IPv4 routing infrastructure can be used to carry IPv6 traffic
- IPv6 datagrams forwarded over regions of IPv4 routing topology by encapsulating them within IPv4 packets
- Tunneling can be used in a variety of ways
  - Router-to-Router
  - Host-to-Router
  - Host-to-Host
  - Router-to-Host

Manually Configured IPv6 over IPv4 Tunnel

SAPIENZA UNIVERSITÀ DI ROMA




```
router1#

interface Tunnel0
 ipv6 enable
 ipv6 address 2001:db8:c18:1::3/127
 tunnel source 192.168.99.1
 tunnel destination 192.168.30.1
 tunnel mode ipv6ip
```

```
router2#

interface Tunnel0
 ipv6 enable
 ipv6 address 2001:db8:c18:1::2/127
 tunnel source 192.168.30.1
 tunnel destination 192.168.99.1
 tunnel mode ipv6ip
```

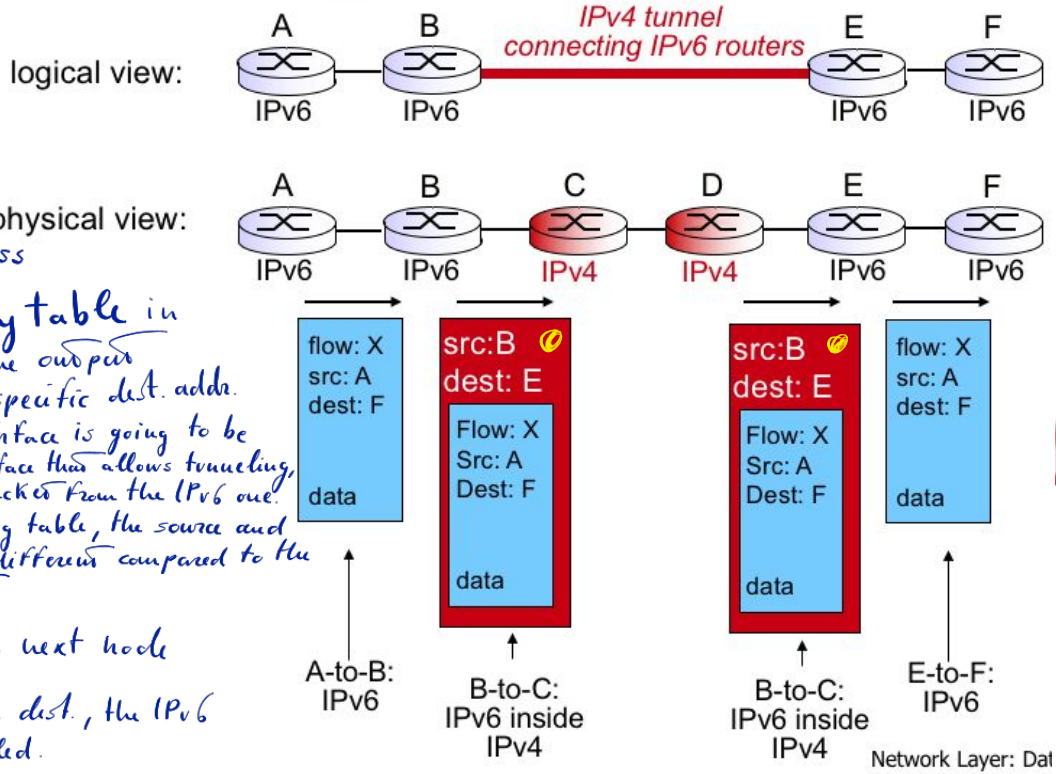
Configured Tunneling Mechanisms

SAPIENZA
UNIVERSITÀ DI ROMA

- The underlying mechanisms for tunneling are
  - The **entry node of the tunnel**
    - creates an encapsulating IPv4 header and transmits the encapsulated packet
  - The **exit node of the tunnel**
    - receives the encapsulated packet,
    - reassembles the packet if needed,
    - removes the IPv4 header,
    - processes the received IPv6 packet
- The determination of which packets to tunnel is usually made by routing information on the encapsulator (e.g., via a routing table)

Encapsulation

SAPIENZA
UNIVERSITÀ DI ROMA



Ordered Process

1) check dest. address
2) go into the routing table in order to check the output interface for the specific dest. addr. In this case the interface is going to be a TUN, a special interface that allows tunneling, so creates the IPv4 packet from the IPv6 one. Note that the routing table, the source and dest. addresses are different compared to the purious IPv6 packet.
3) forward it to the next node
4) when it reaches the dest., the IPv6 packet will be revealed.

The encapsulation process put an IPv4 header on top of the IPv6 packet.

In the new IPv4 header the source node is the one that performed the encapsulation.

Segment Routing

Network Infrastructures A.Y. 2023/24

Outline

- Introduction to Segment Routing
- Segment Routing Policy
- SRv6 and Segment Routing extension Header
- Network Programming

It's the substitute of MPLS.

↓

Multi Protocol
Label Switching

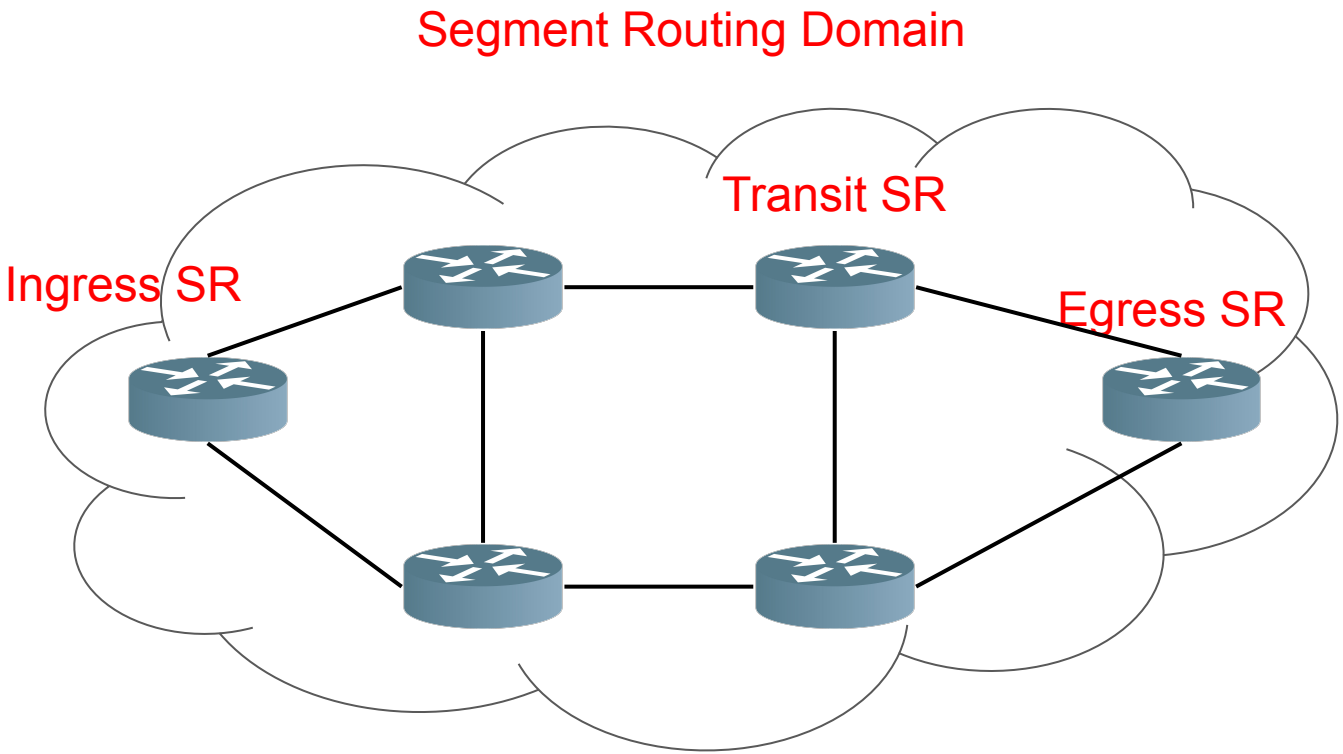
Outline

- Introduction to Segment Routing
- Segment Routing Policy
- SRv6 and Segment Routing extension Header
- Network Programming

SR overview

- Segment Routing (SR) leverages the source routing paradigm
- A node steers a packet through an ordered list of instructions, called segments
- A segment can represent any instruction, topological or service-based
- A segment can have a semantic:
  - local to an SR node
  - global within an SR domain
- SR allows to enforce a flow through any topological path while maintaining per-flow state only at the ingress nodes to the SR domain

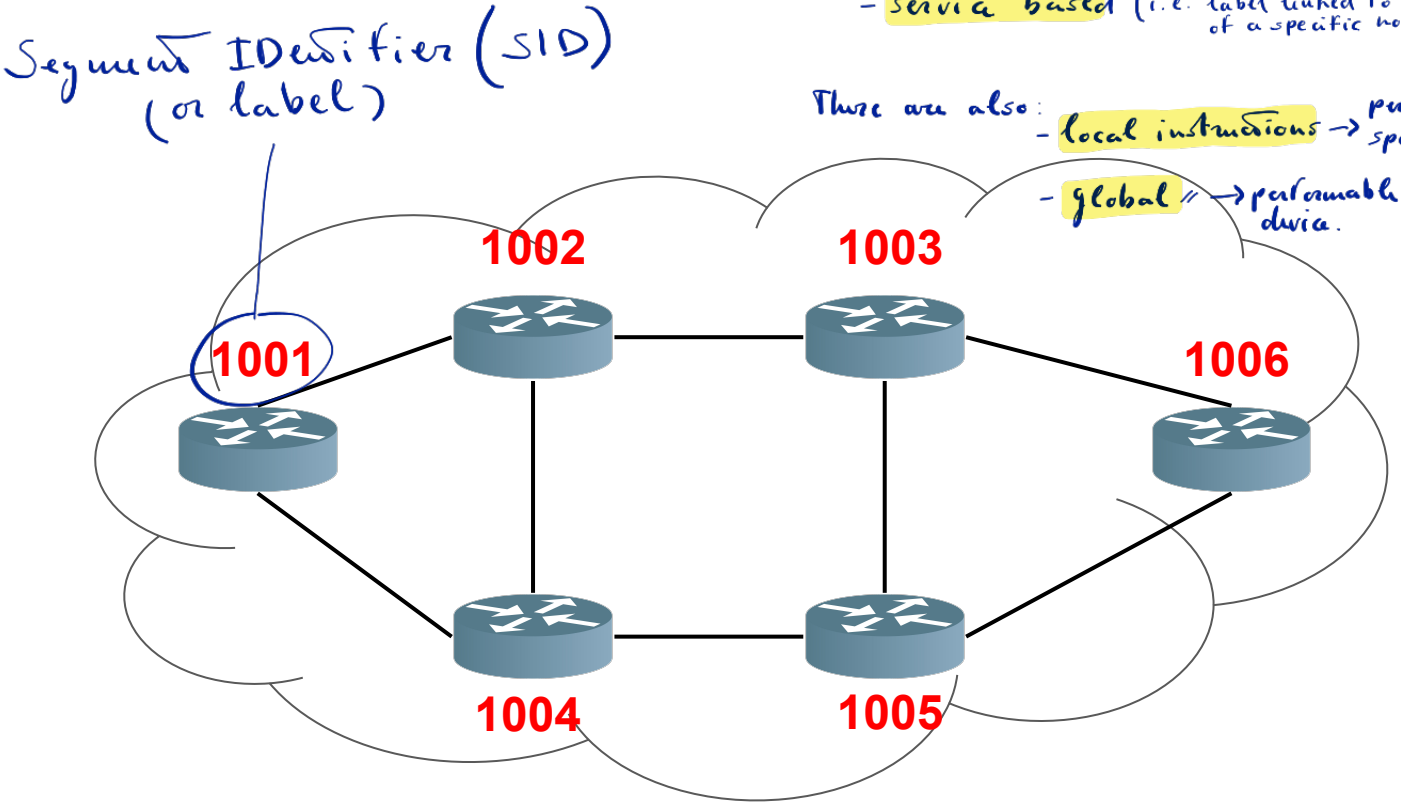
SR terminology



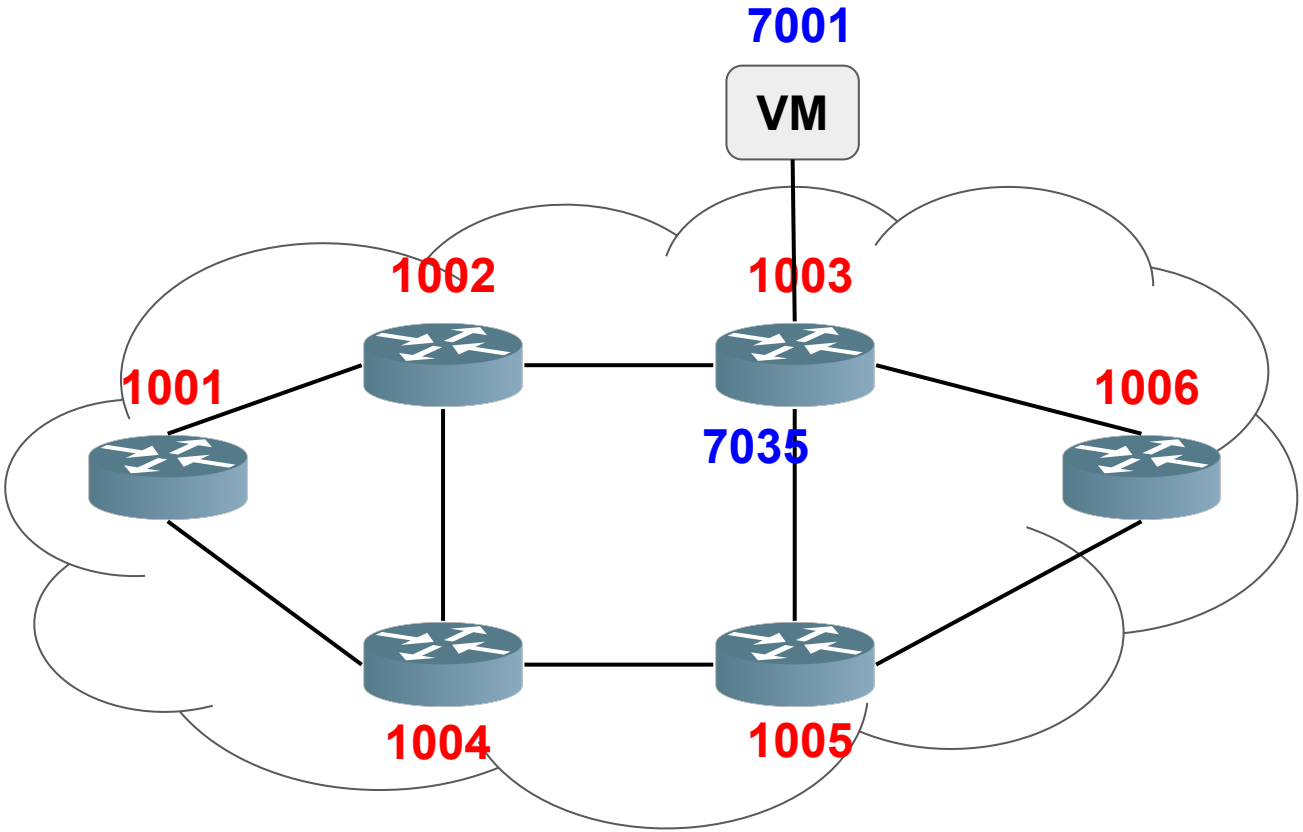
SR idea

Every SID corresponds to an instruction.
There are 2 types of instructions:
- topological based (i.e. node/interface identifiers)
- service based (i.e. label linked to a specific service of a specific node.)

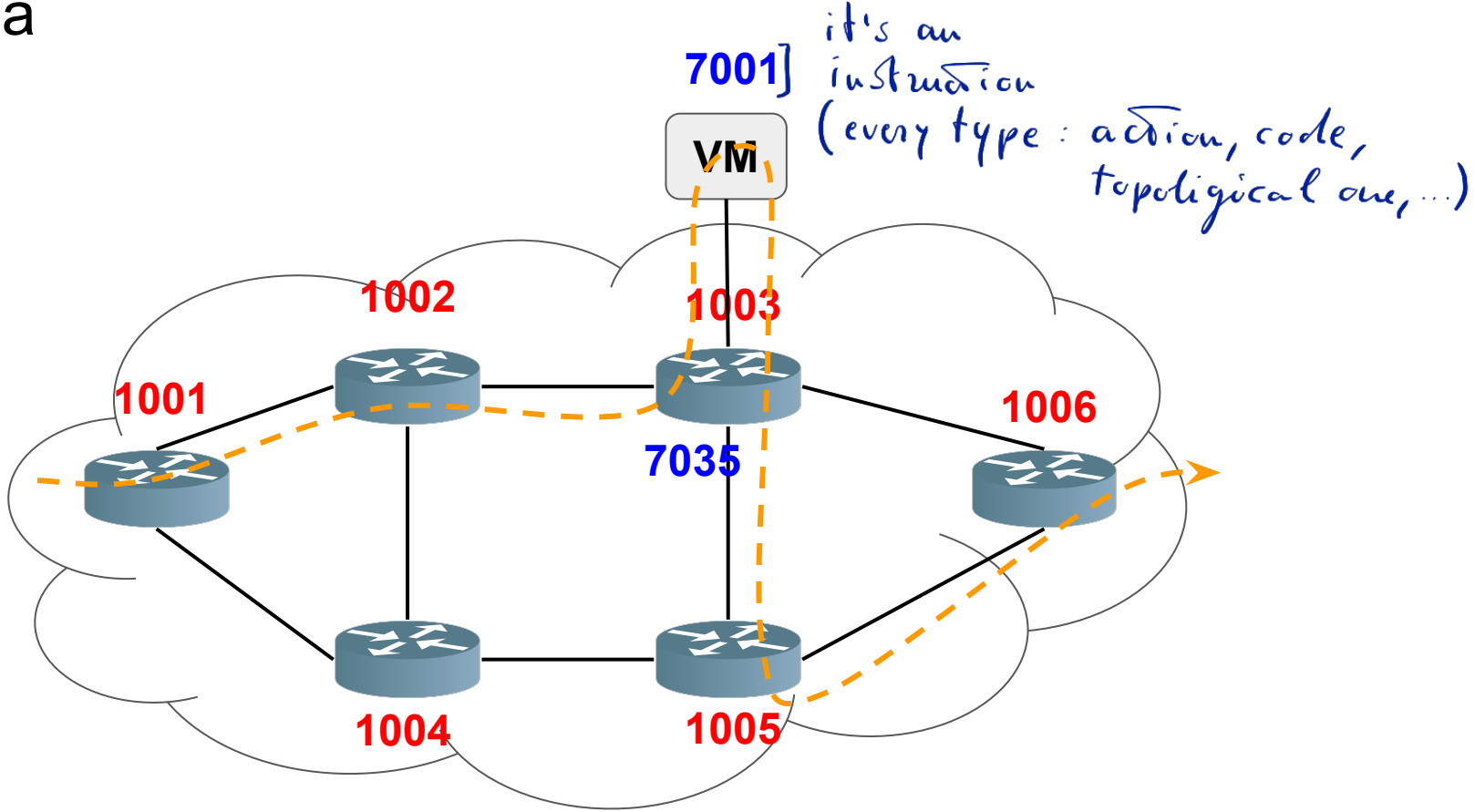
There are also:
- local instructions → performable just by a specific device.
- global → performable by every device.



SR idea



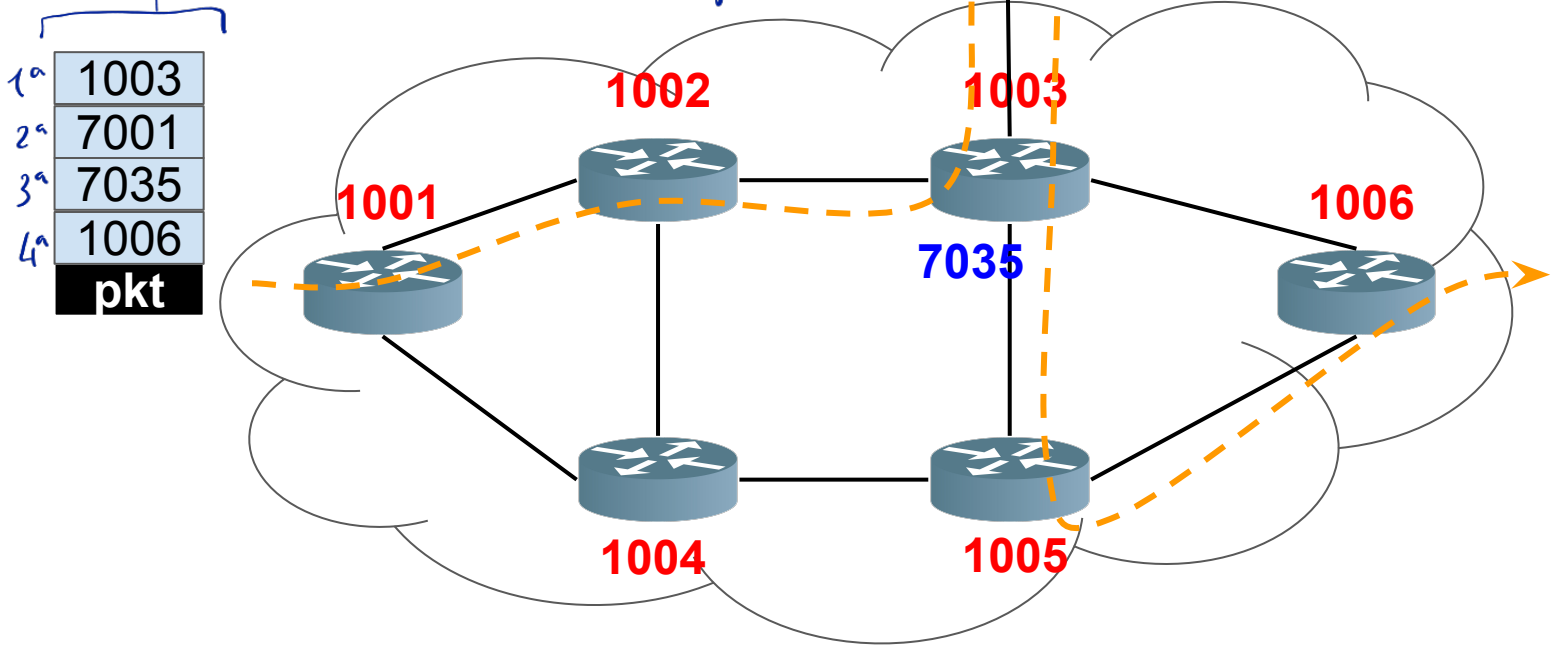
SR idea


7001]
it's an instruction
(every type: action, code, topological one, ...)
VM
1001
1002
1003
1004
1005
1006
7035

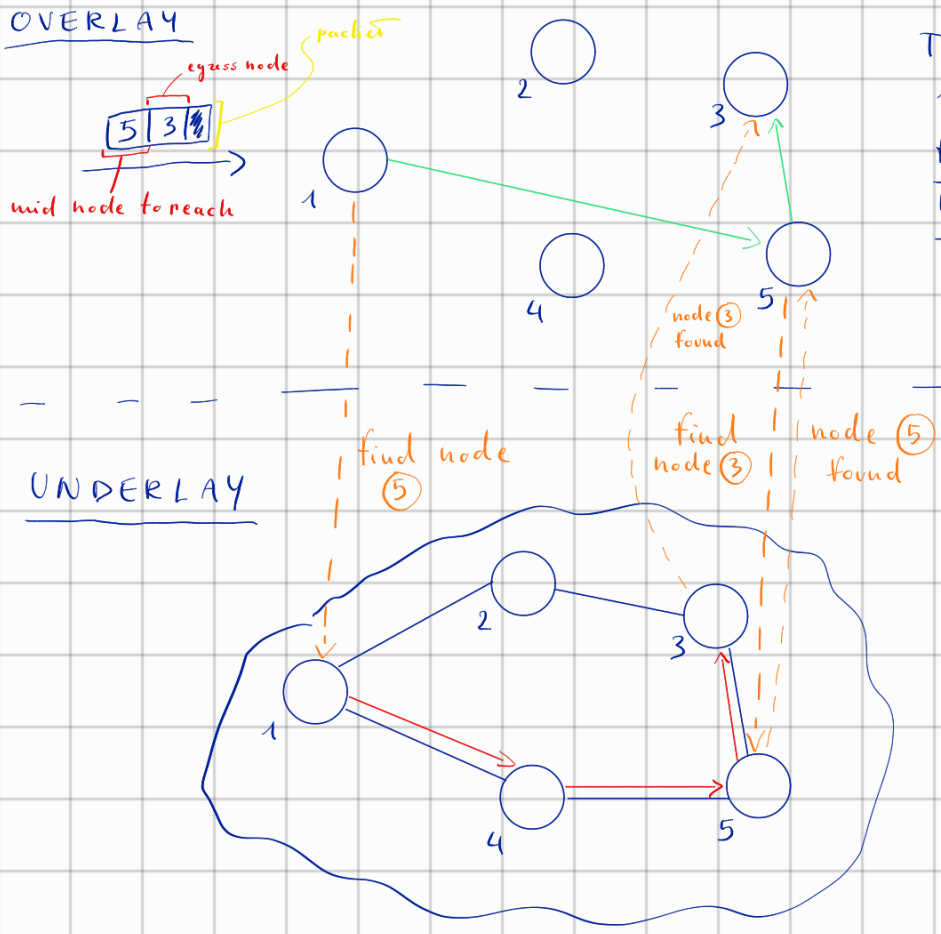
SR idea

segments path given to the source node.

Note: it just indicates the intermediate nodes and the final destination but not the "hop by hop" path. In order to send segments the shortest path algorithm is used.



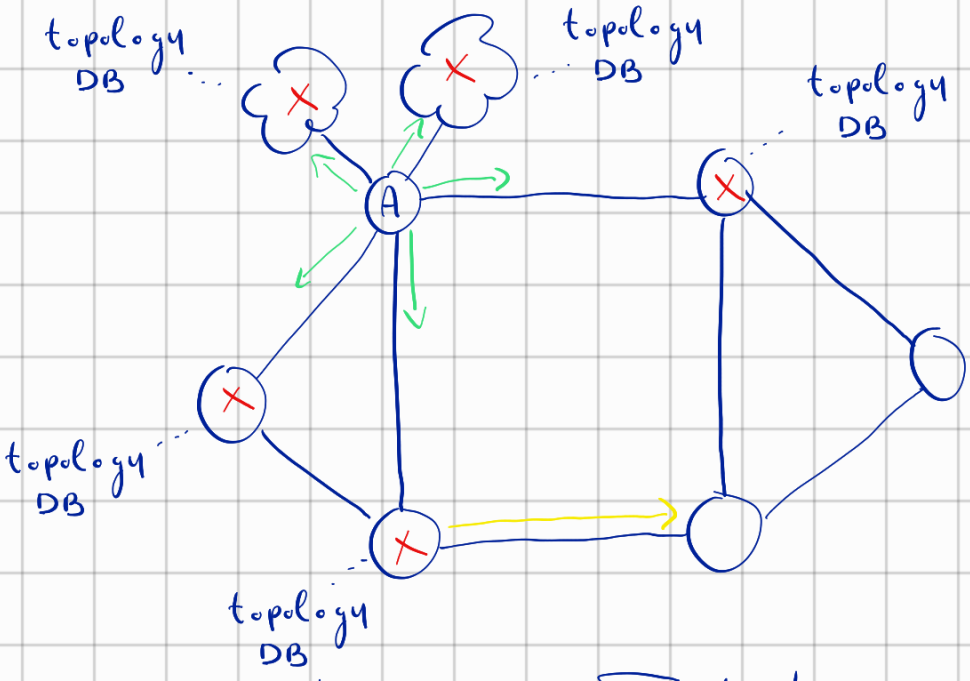
How Segment Routing



The green path is the logic path taken by the incoming.
The real path is transparent to the overlay and is performed in the underlay.

The red path is the shortest path actually performed.
Every time a planned node is reached is notified to the overlay (dashed lines).
The overlay's nodes also ask the underlay to find the path to specific nodes.

OSPF sids advertising



The node A is connected to 5 networks.
It send a Link State Advertisement (LSA) in order to let all the directly connected nodes to build their topology DB.
Once a node receive a LSA starts building is own topology DB.
After then a flooding system will spread the received LSA to other nodes not directly connected.

SR control plane

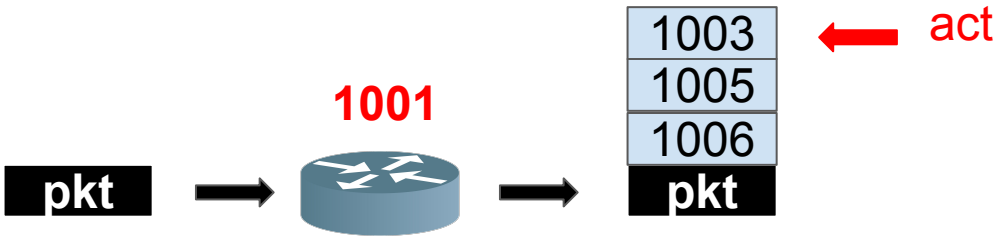
- The SR architecture supports any type of control-plane: distributed, centralized or hybrid.
- In a **distributed scenario**:
  - the segments are allocated and signaled by OSPF or BGP
  - a node individually decides to steer packets on a source-routed policy
  - a node individually computes the source-routed policy
- In a **centralized scenario**:
  - the segments are allocated and instantiated by an SR controller
  - the SR controller decides which nodes need to steer which packets on which source-routed policies
  - the SR controller computes the source-routed policies
- A **hybrid scenario** complements a base distributed control-plane with a centralized controller
  - for example, when the destination is outside the IGP domain, the SR controller may compute a source-routed policy on behalf of an IGP node

SR data plane

- The SR architecture can be instantiated on various dataplanes
  - SR over MPLS (SR-MPLS)
  - SR over IPv6 (SRv6)
- Segment Routing can be directly applied to the MPLS architecture with no change on the forwarding plane
- Segment Routing can be applied to the IPv6 architecture with a new type of routing header called the SR header (SRH)

SR forwarding

- SR nodes have a forwarding table that specifies the operation to perform on a received packet

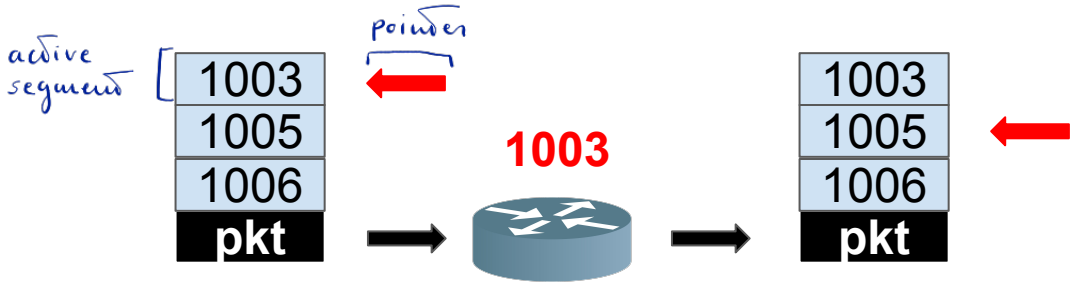


- Three different operations:
  - PUSH: the instruction consisting of the insertion of a segment at the top of the segment list

We can say that 1001 it's an ingress node because the incoming packet doesn't have a sids list.

SR forwarding

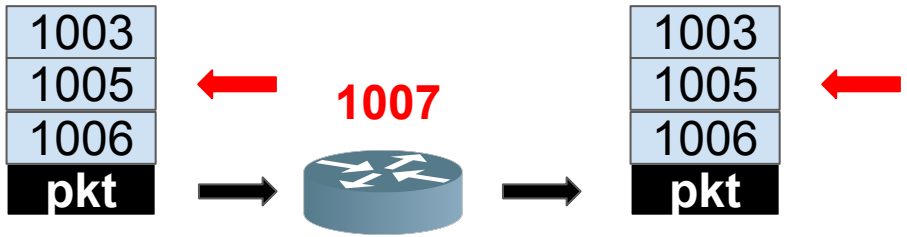
- SR nodes have a forwarding table that specifies the operation to perform on a received packet



- Three different operations:
  - PUSH: the instruction consisting of the insertion of a segment at the top of the segment list
  - NEXT: when the active segment is completed, NEXT is the instruction consisting of the inspection of the next segment (it performs the POP action of the active segment when it's done.)

SR forwarding

- SR nodes have a forwarding table that specifies the operation to perform on a received packet



- Three different operations:
  - (PUSH in MPLS) PUSH: the instruction consisting of the insertion of a segment at the top of the segment list
  - (POP in MPLS) NEXT: when the active segment is completed, NEXT is the instruction consisting of the inspection of the next segment
  - (SWAP in MPLS) CONTINUE: the active segment is not completed and hence remains active

Outline

- Introduction to Segment Routing
- Segment Routing Policy
- SRv6 and Segment Routing extension Header
- Network Programming

[Usually the policies are implemented in the edge nodes.]

Every node has its own SR policy DB where are stored all the possible paths.
When the ingress node receive a message to deliver it performs ranking of the candidate paths just then it forward the message through it (and all the messages with the same destination).

SR Policy

- An SR Policy is identified through the tuple

  <headend, color, endpoint>

  path
  ingress node
  destination

  These tuples are stored in a policies DB of a specific node.

- The headend is the node where the policy is instantiated/implemented
- The endpoint indicates the destination of the policy
  - headend and endpoint are specified as an IPv4 or IPv6 address
- The color is a 32-bit numerical value that associates the SR Policy with an intent (e.g., low-latency)

In order to know which is the best policy to apply it's needed to classify the incoming packets and choose the best one.
When the policy is chosen, it's needed to refer to another DB (of a specific color - policy) that has stored all the available SLs for the specified policy (the active one and the backups).
segment lists

Candidate Path and Segment List

- An SR Policy is associated with one (or more) candidate path
- A candidate path is itself associated with a Segment-List (SID-List)
  - a SID-List represents a specific source-routed way to send traffic from the head-end to the endpoint of the corresponding SR policy
- A candidate path is either dynamic or explicit
- A headend may be informed about a candidate path for an SR Policy by various means including:
  - local configuration
  - PCE

SR Policy: summary

SR policy POL1 <headend, color, endpoint>

Candidate-path CP1

Preference 200

Weight W1, SID-List1 <SID11...SID1i>

Weight W2, SID-List2 <SID21...SID2j>

Candidate-path CP2

Preference 100

Weight W3, SID-List3 <SID31...SID3i>

Weight W4, SID-List4 <SID41...SID4j>

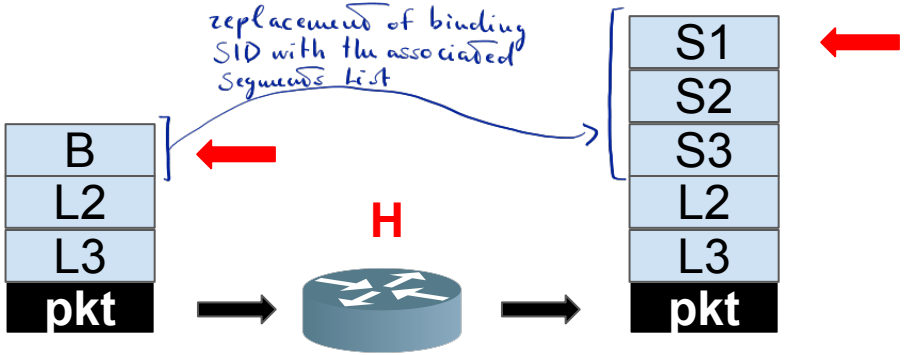
Just one of them is the working one, the other is the backup segment list.

Binding SID

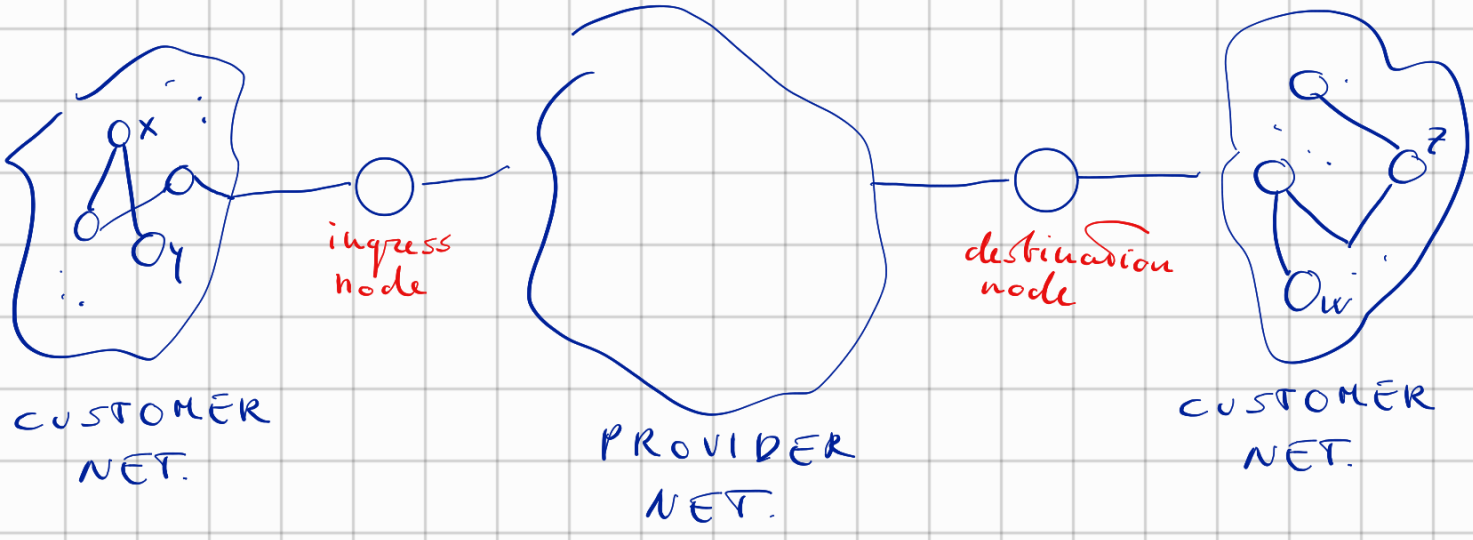
[It's the SID assigned to a policy]

[It's a local instruction]

- An SR Policy installs a **BSID** entry in the forwarding plane with the action of steering the packets matching this entry to the selected path
- Let us assume that headend **H** has a valid SR Policy **P** of SID-List **<S1, S2, S3>** and BSID **B**



- **H** has steered the packet in the SR policy **P**



It's possible to have the need to pass through an external network (provider) and thanks to BSID it's possible to specify a specific policy we want (low latency) without let the external network espose something.
So we can specify the path nodes for our net (customer) and then create a BSID for when we have to pass through provider net.

es.

[x, y, BSID, w, z]

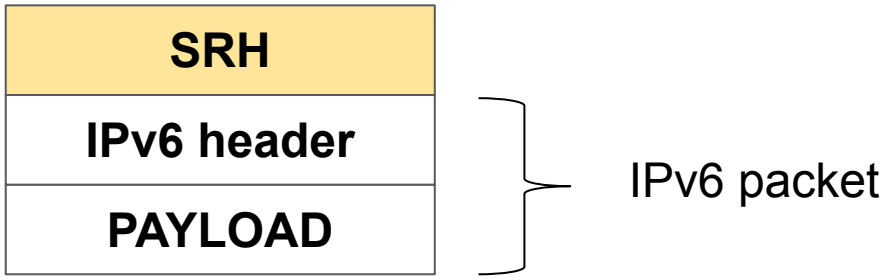
Outline

- Introduction to Segment Routing
- Segment Routing Policy
- SRv6 and Segment Routing extension Header
- Network Programming

SRH instantiation

- The source based routing model in case of SR over IPv6 dataplane (SRv6) is realized through the instantiation of the Segment Routing Header (SRH)
- The SRH is added to the packet by its source:
  - At the node originating the packet (host, server)
  - At the ingress node of an SR domain

routing type header, that didn't exist before.



Segment Routing Extension Header (SRH)

Most common

- A new type of the Routing Header is defined
  - the Segment Routing Header (SRH)

pointer to the correct instruction being executed

next header	hdr ext len	routing type	segments left
last entry	flags	tag	
segments list [0] (128 bit IPv6 address)			
.....			
segments list [n] (128 bit IPv6 address)			
optional type length value objects (variable)			

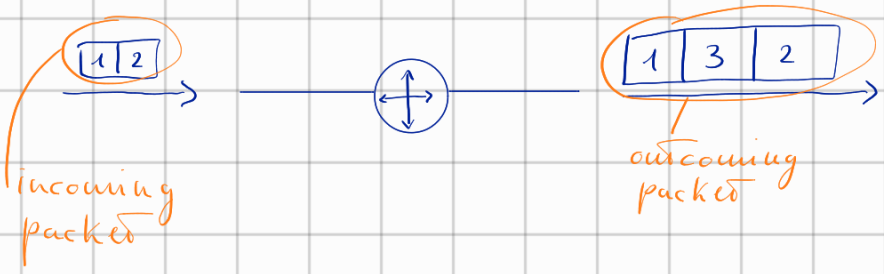
SIDs (Segments List)

Segment Routing Extension Header (SRH)

- **Next Header**: Identifies the type of header immediately following the SRH
- **Hdr Ext Len**: length of the SRH header in 8-octet units
- **Routing Type**: TBD, to be assigned by IANA (suggested value: 4)
- **Segments Left**: it contains the index, in the Segment List, of the next segment to inspect
  - Segments Left is decremented at each segment
- **Last Entry**: contains the index, in the Segment List, of the last element of the Segment List
- **Flags**: 8 bits of flags
- **Tag**: tag a packet as part of a class or group of packets (packets sharing the same set of properties)
- **Segment List[n]**: 128 bit IPv6 addresses representing the nth segment in the Segment List
- **Type Length Value** (TLV)

There are different ways to add the SRH to a IPv6 packet. These are the options:

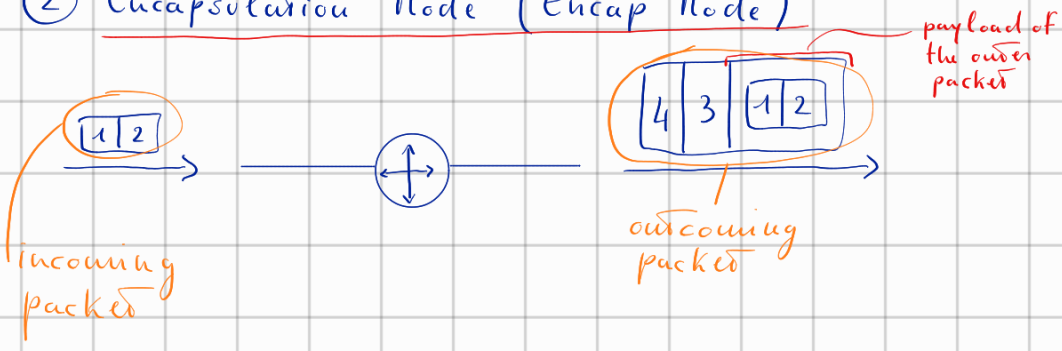
① Insert Mode



LEGENDA
① IPv6 Header
② Payload
③ SRH

This mode is usable just if the customer (that sends the packet) is using IPv6. That is because in order to add an additional header the packet must be an IPv6 one.

② Encapsulation Mode (Encap Mode)

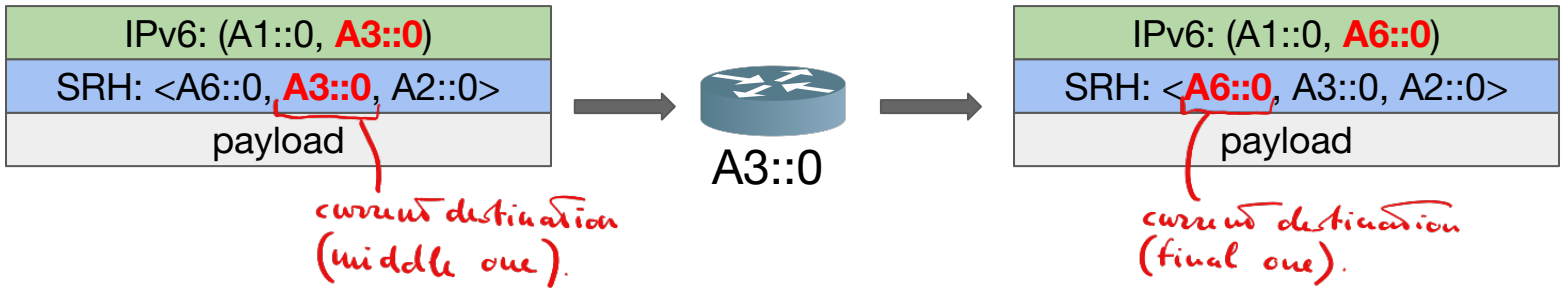


LEGENDA
① IPv6 Header
② Payload
③ SRH
④ Outer Header

This 2nd mode is the most used because it's more flexible. That is because the original packet, once is encapsulated, is not checked anymore (so it could be either IPv6 or IPv4, it doesn't matter).

SRH processing

- For the SRH holds the following property:
  - Only the router whose address is in the DA field of the packet header MUST inspect the SRH
- Segment Routing in IPv6 networks implies that the segment identifier is moved into the DA of the packet



- The DA of the packet changes at each segment termination/completion

My Local SID Table

IPv6 address

- An SRv6-capable node N maintains a MyLocalSID Table
- This table contains all the local SRv6 segments explicitly instantiated at node N
- N is the parent node for these SID's
- Every SRv6 local SID instantiated has a specific instruction bounded to it

It's common to create a loop net, connected to a node, in order to create SIDs for a specific node, we can create within the dummy network addresses range.
It allows to have the same prefix for all the behaviours in the previously created (dummy) net.
Now it's possible, thanks to the locator, to know where specific services are located in order to ask for them.

MyLocalSID Table

SID	instruction → behaviour / function
...	...

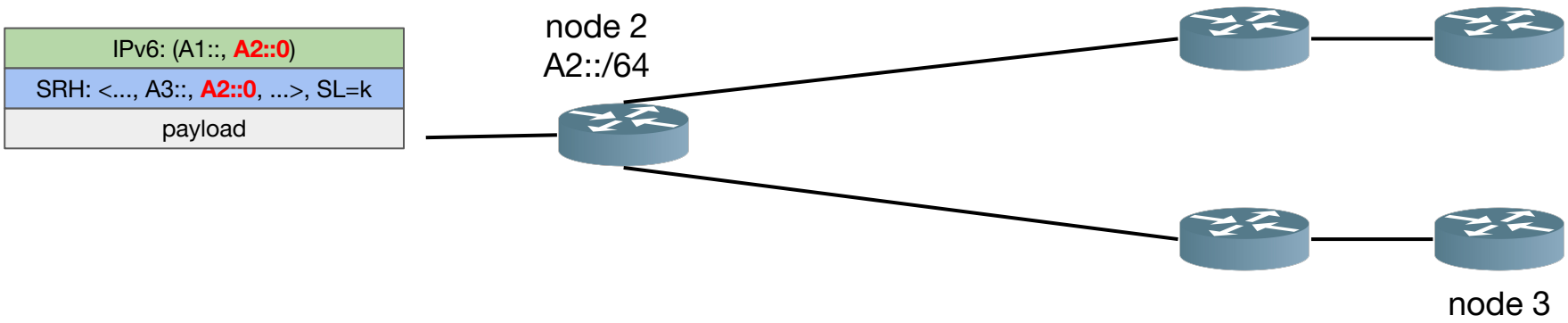
SRH Functions

- Segment Routing architecture defines a segment as an instruction or, more generally, a set of instructions (function)
- Two SRv6 basic functions are:
  - **End**
    - the endpoint (End) function is the base of the source routing paradigm
    - it consists of updating the DA with the next segment and forward the packet accordingly
  - **End.X** → send out the message through a specific node.
    - the endpoint layer-3 cross-connect function

Endpoint Function (End)

- When a node receives a packet destined to DA "S" and:
  - S is an entry in the MyLocalSID table
  - the function associated with S is "End"

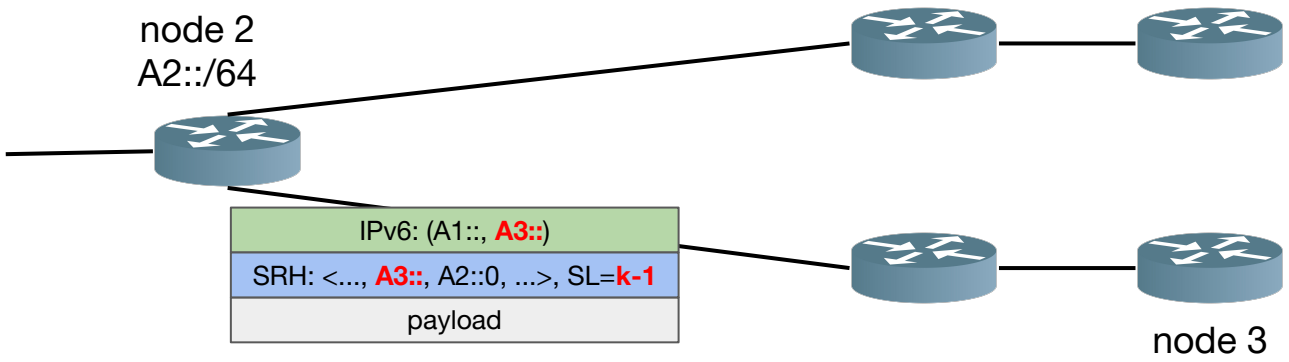
1. **IF** SegmentsLeft > 0 **THEN**
2. decrement SL
3. update the IPv6 DA with SRH[SL]
4. FIB lookup on updated DA
5. forward accordingly to the matched entry
6. **ELSE**
7. drop the SRH



Endpoint Function (End)

- When a node receives a packet destined to DA "S" and:
  - S is an entry in the MyLocalSID table
  - the function associated with S is "End"

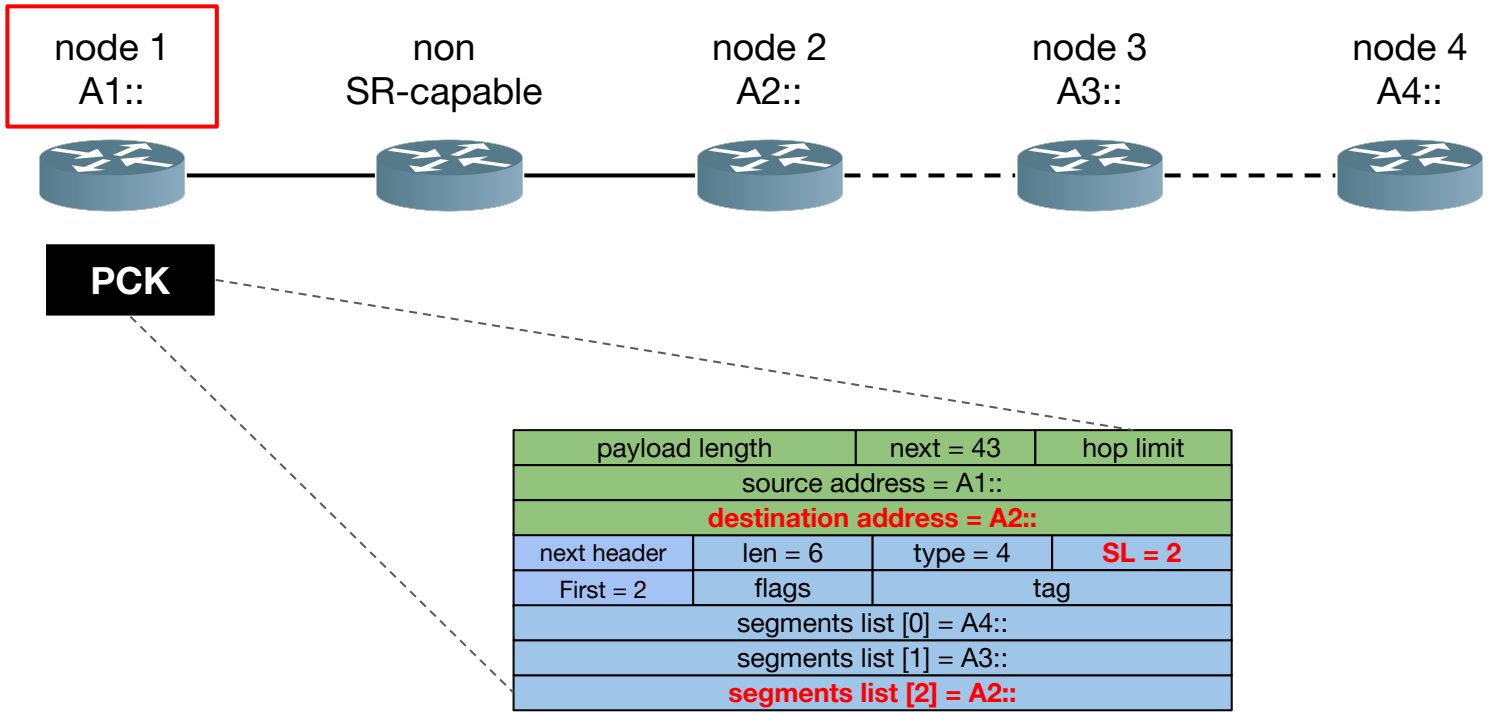
1. **IF** SegmentsLeft > 0 **THEN**
2. decrement SL
3. update the IPv6 DA with SRH[SL]
4. FIB lookup on updated DA
5. forward accordingly to the matched entry
6. **ELSE**
7. drop the SRH



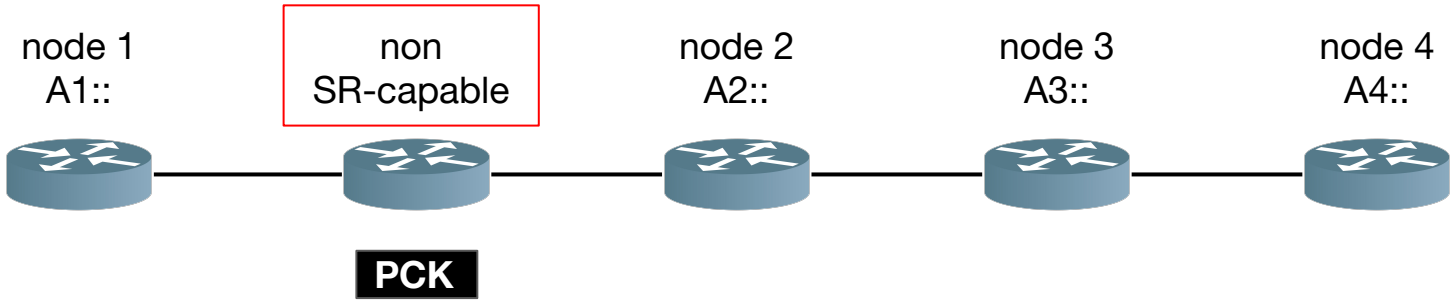
Source Node

- Source Node is SR-capable
- SRH is created with:
  - Segment List in reversed order of the path
    - Segment List [0] is the **LAST** segment
    - Segment List [n-1] is the **FIRST** segment
    - Segment left is set to **n-1**
    - First segment is set to **n-1**
- IP DA is set to the first segment
- Packet is sent according to the IP DA
  - Normal IPv6 forwarding

Source Node


node 1
A1::
non
SR-capable
node 2
A2::
node 3
A3::
node 4
A4::
PCK
payload length
next = 43
hop limit
source address = A1::
destination address = A2::
next header
len = 6
type = 4
SL = 2
First = 2
flags
tag
segments list [0] = A4::
segments list [1] = A3::
segments list [2] = A2::

Non-SR Transit Node



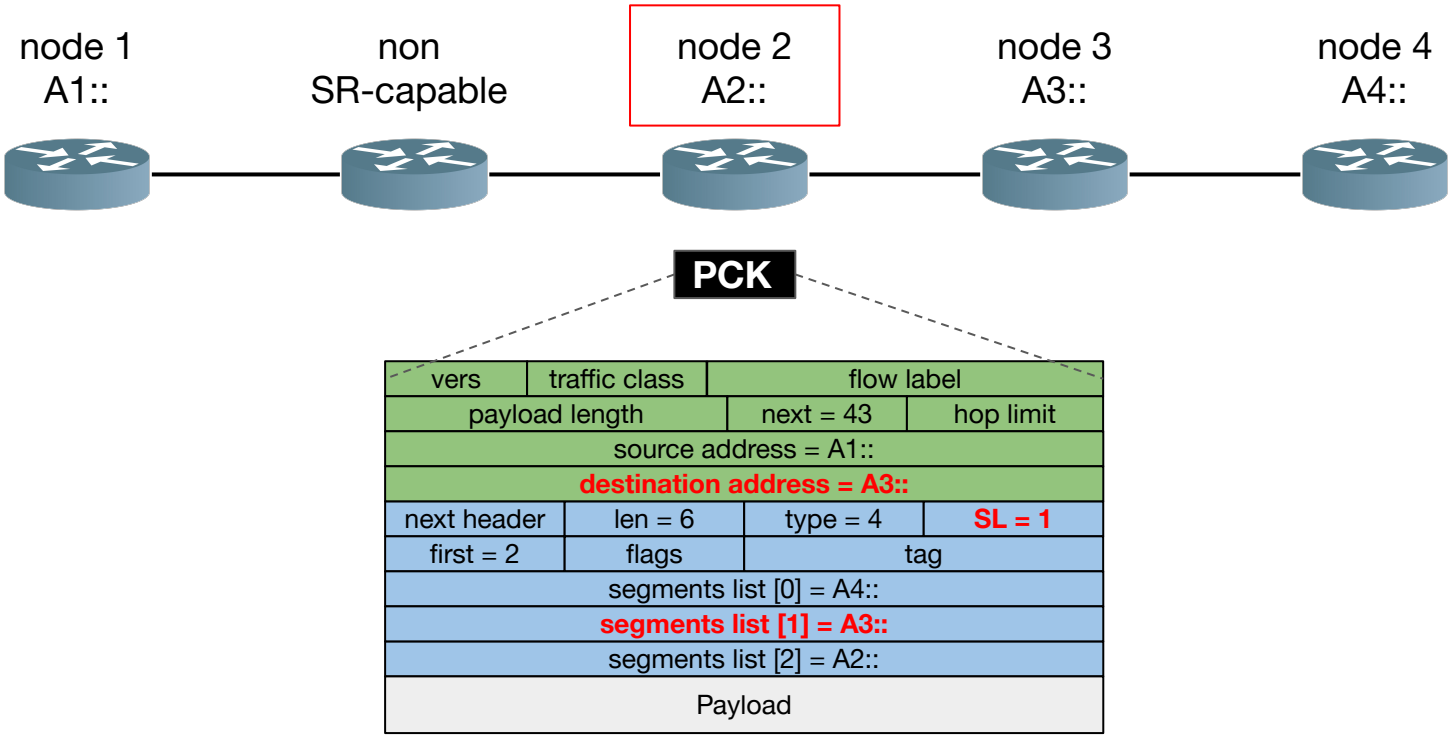
- Plain IPv6 forwarding
- Solely based on IPv6 destination address
- No SRH inspection or update

it's not capable to handle headers, so it's not possible to use it as a middle point. It's capable just to perform packets forwarding.

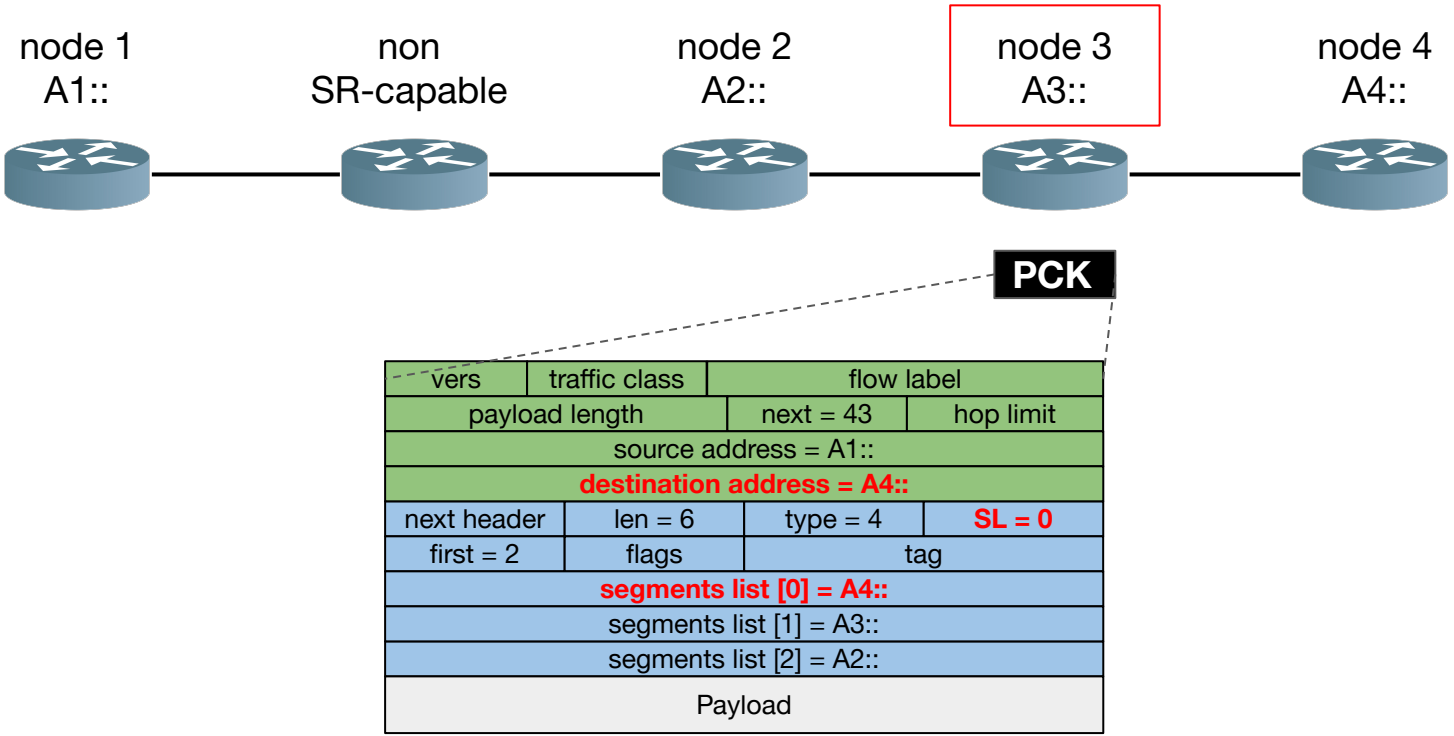
SR Segment Endpoints

- **SR Endpoints**: SR-capable nodes whose address is in the IP DA
- SR Endpoints inspect the SRH and do:
  - IF Segments Left > 0, THEN
    - Decrement Segments Left (-1)
    - Update DA with Segment List [Segments Left]
    - Forward according to the new IP DA

SR Segment Endpoints

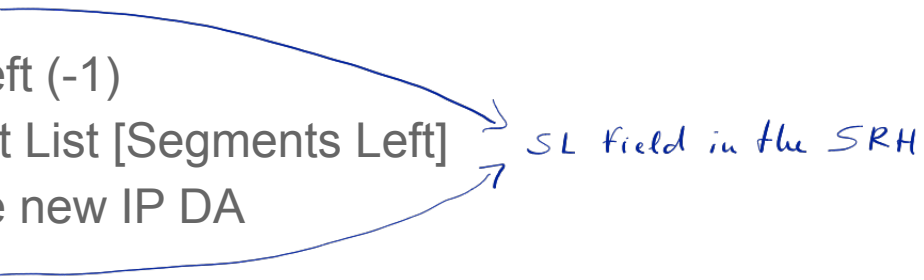


SR Segment Endpoints

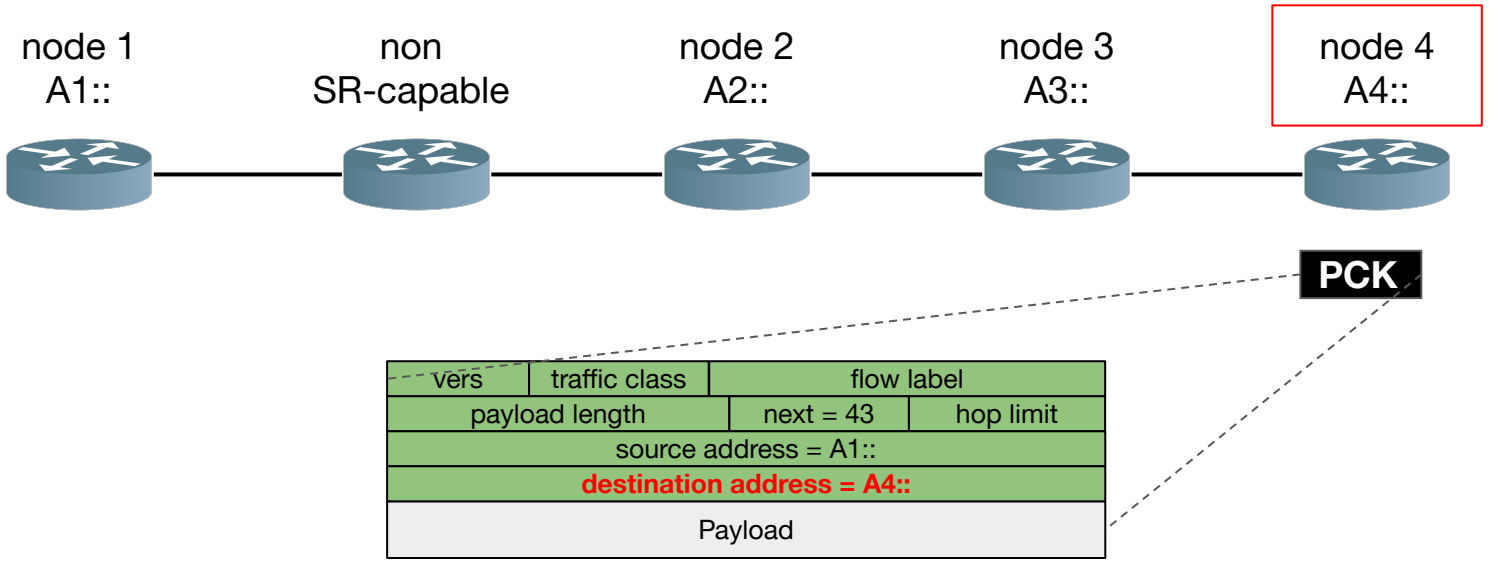


SR Segment Endpoints

- SR Endpoints: SR-capable nodes whose address is in the IP DA
- SR Endpoints inspect the SRH and do:
  - IF Segments Left > 0, THEN
    - Decrement Segments Left (-1)
    - Update DA with Segment List [Segments Left]
    - Forward according to the new IP DA
  - ELSE (Segments Left = 0)
    - Remove the IP and SR header
    - Process the payload



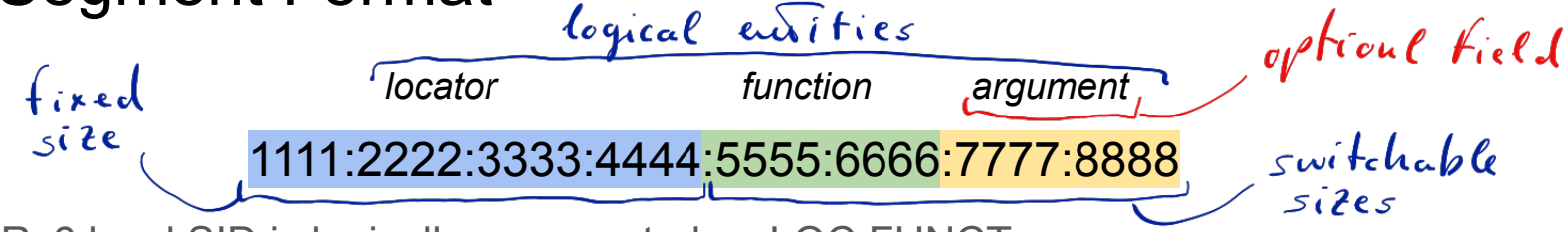
SR Segment Endpoints



Outline

- Introduction to Segment Routing
- Segment Routing Policy
- SRv6 and Segment Routing extension Header
- Network Programming

SRv6 Segment Format



- An SRv6 local SID is logically represented as LOC:FUNCT
  - LOC is the L most significant bits
  - FUNCT is the 128-L least significant bits
- L is called the locator length and is flexible
- Most often the LOC part of the SID is routable and leads to the node which owns that SID
- The FUNCT part of the SID is an opaque identification of a local function bound to the SID
- A function may require additional arguments
  - the SRv6 Local SID will have the form LOC:FUNCT:ARGS

This SID could be consider as a proper function, and since is not applied only on a single machine but on multiple machines, it's a network program.

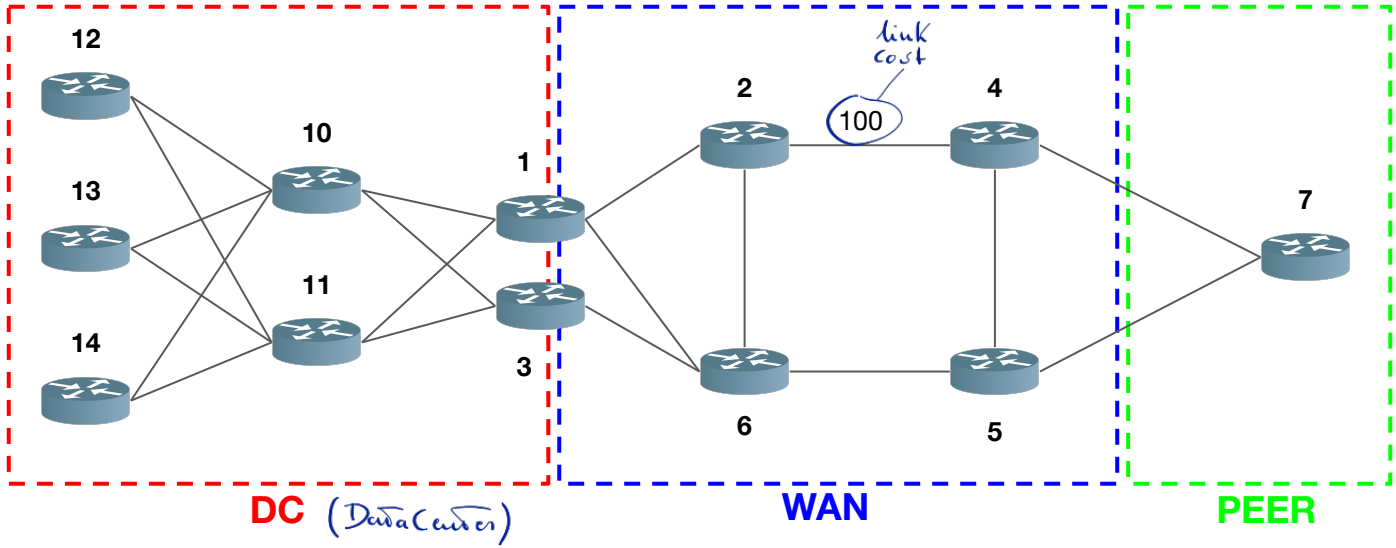
SRv6 Functions

- Each entry of the MyLocalSID Table indicates the function associated with the local SID
- In practice, any function can be attached to a local SID
  - a node N can bind a SID to a local VM which can apply any complex function on the packet
- Some examples:
  - End
  - End.X
  - End.B6

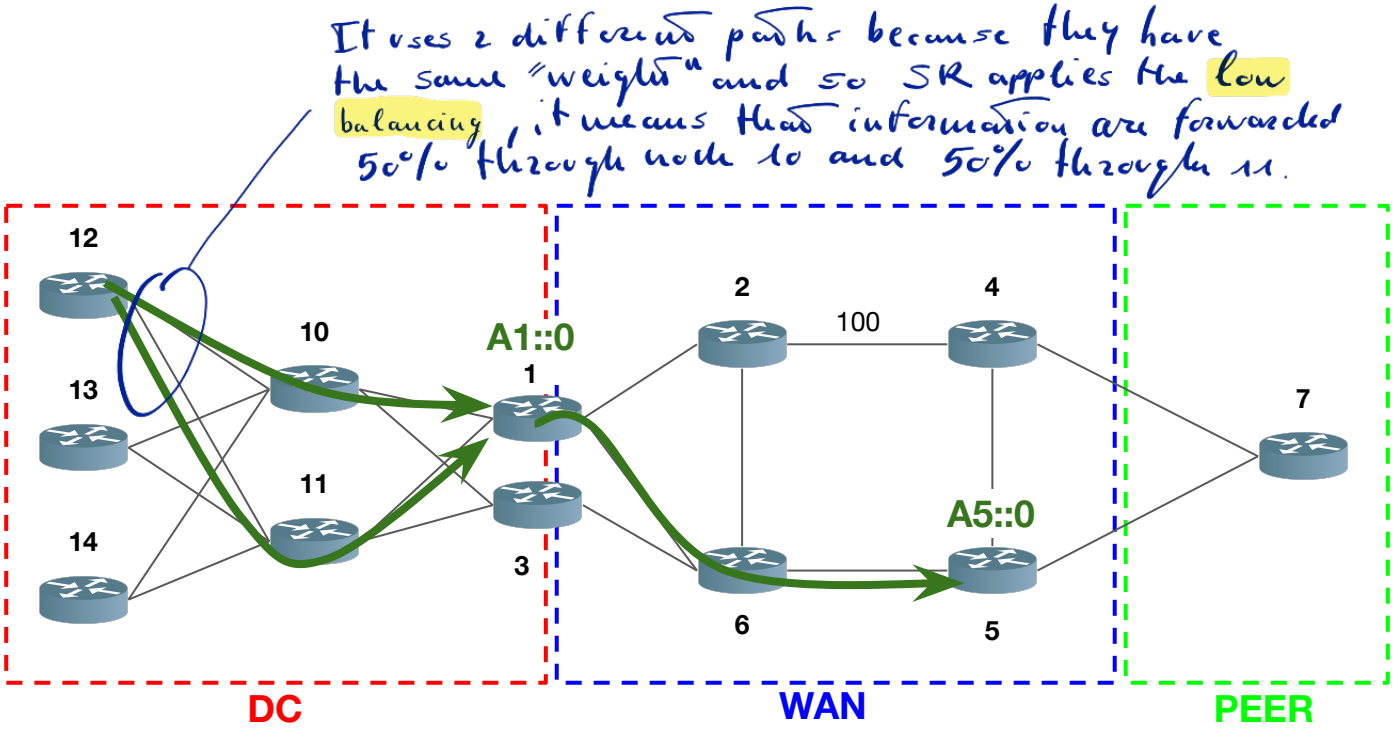
SID allocation for illustration purpose

- Node K advertises prefix AK::/64
- The function is encoded in the last 64 bits
  - **0** denotes the **End** function
  - **CJ** denotes the **End.X** function on **link CJ**

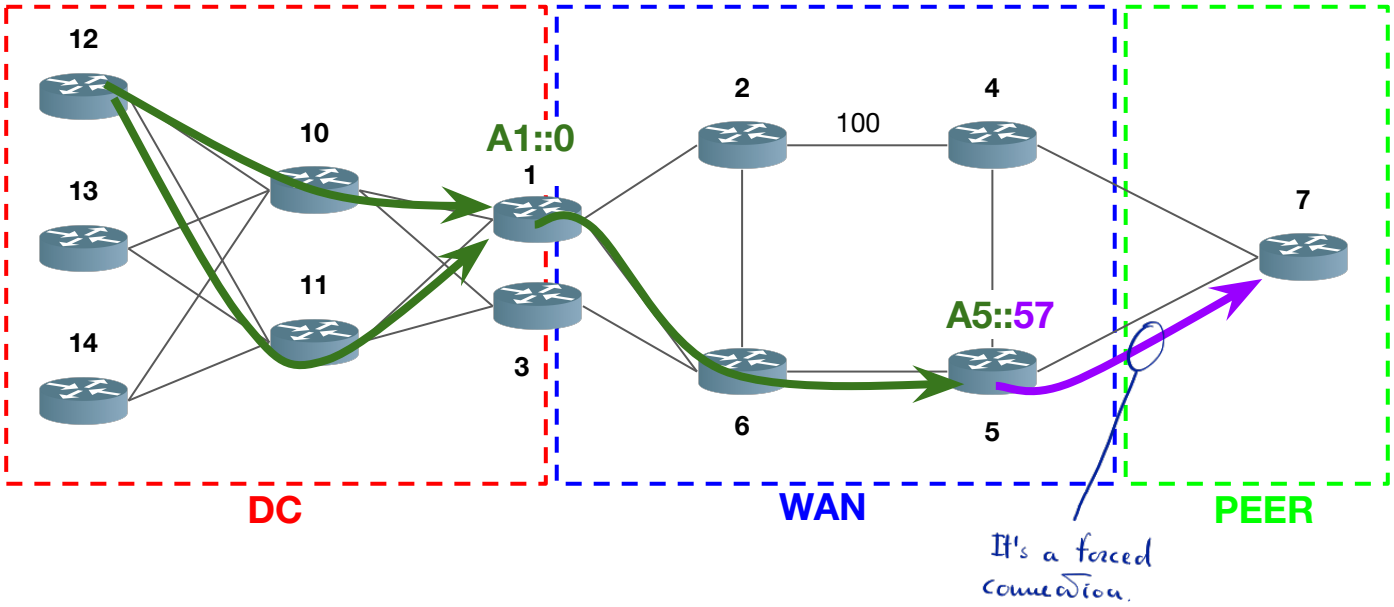
[Arguments are optional.]
[Locator and Function are mandatory.]



A1::0 and then A5::0



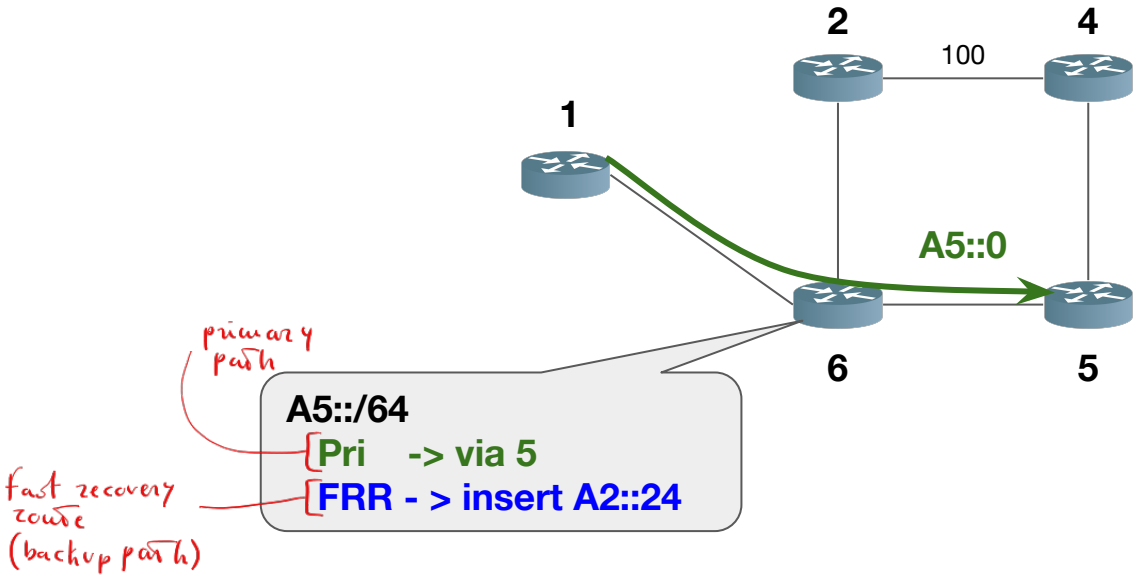
A1::0 and then A5::57



TILFA

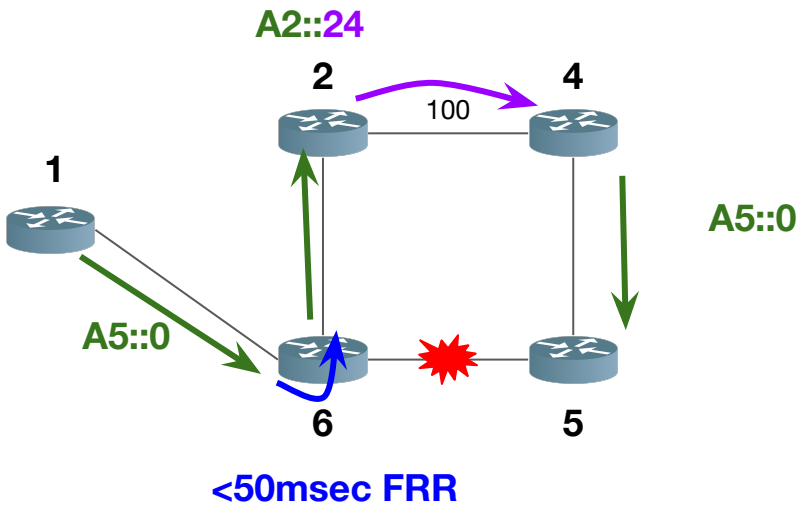
This is one of the cases where policies are applied on middle nodes.
[Usually they're applied on edge nodes.]

- 50 msec protection upon local link, node or SRLG failure

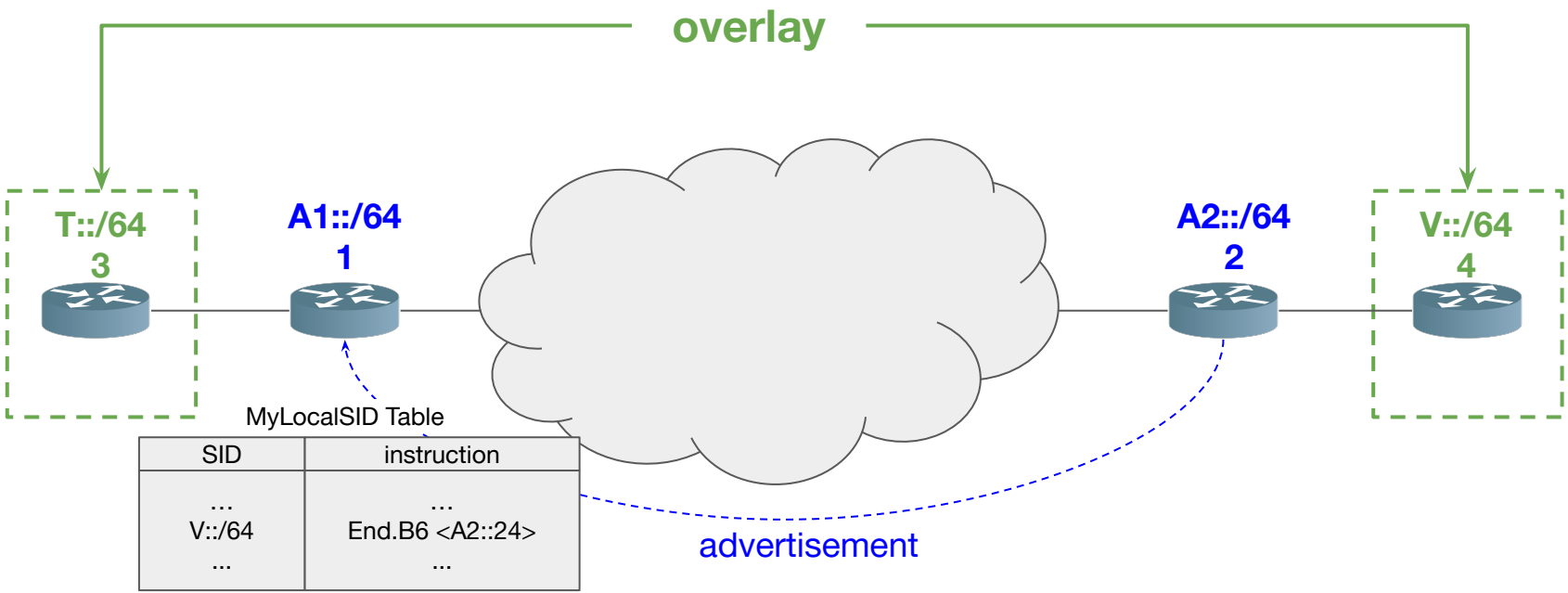


TILFA

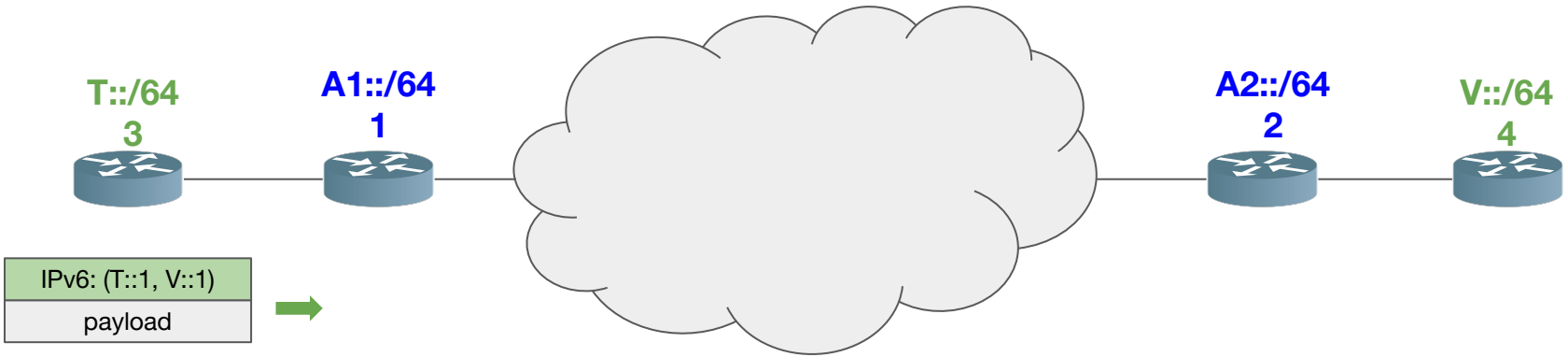
- 50 msec protection upon local link, node or SRLG failure



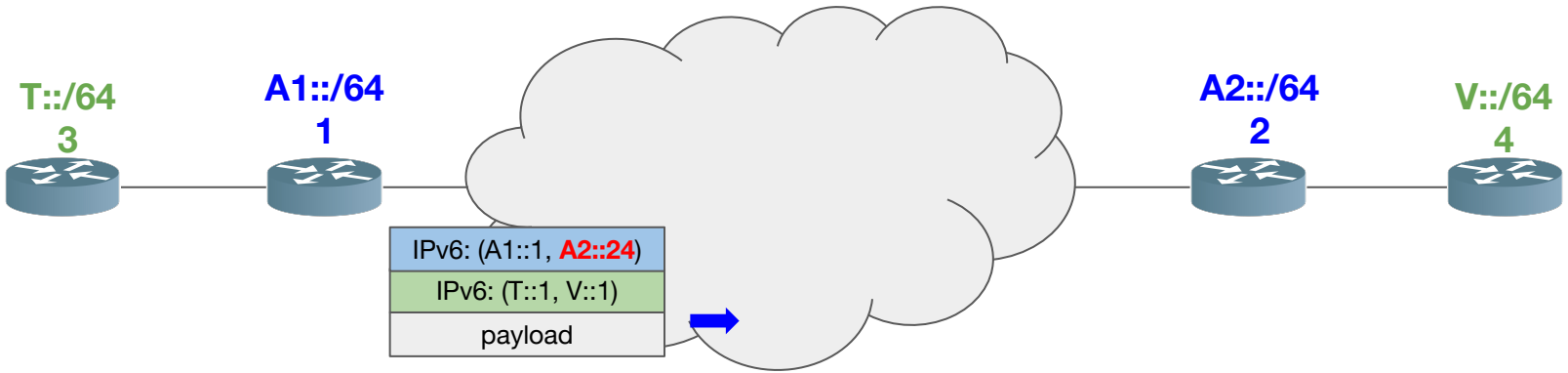
Overlay


overlay
T::/64
3
A1::/64
1
A2::/64
2
V::/64
4
MyLocalSID Table
SID
instruction
...
V::/64
...
...
End.B6 <A2::24>
...
advertisement

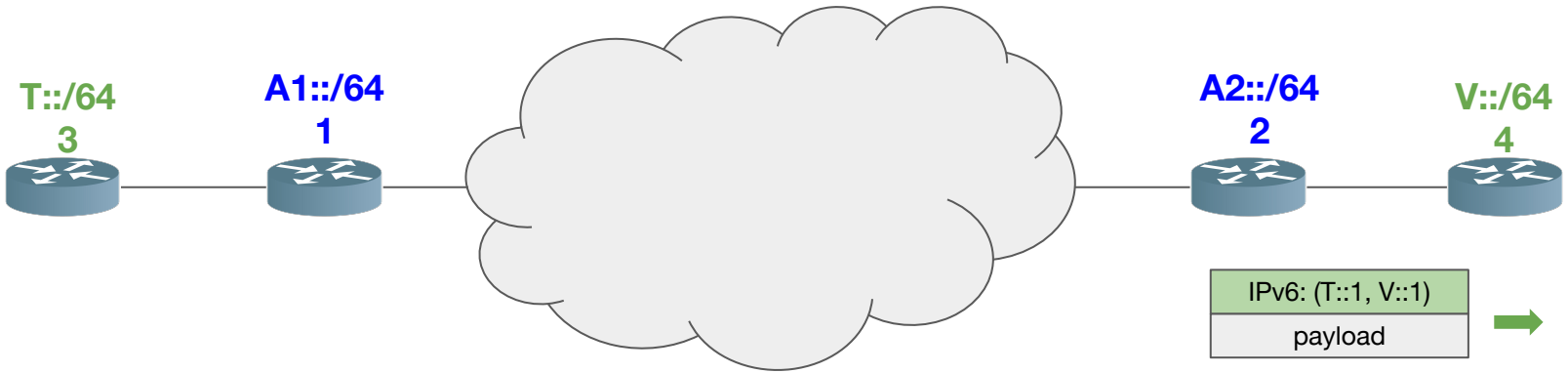
Overlay



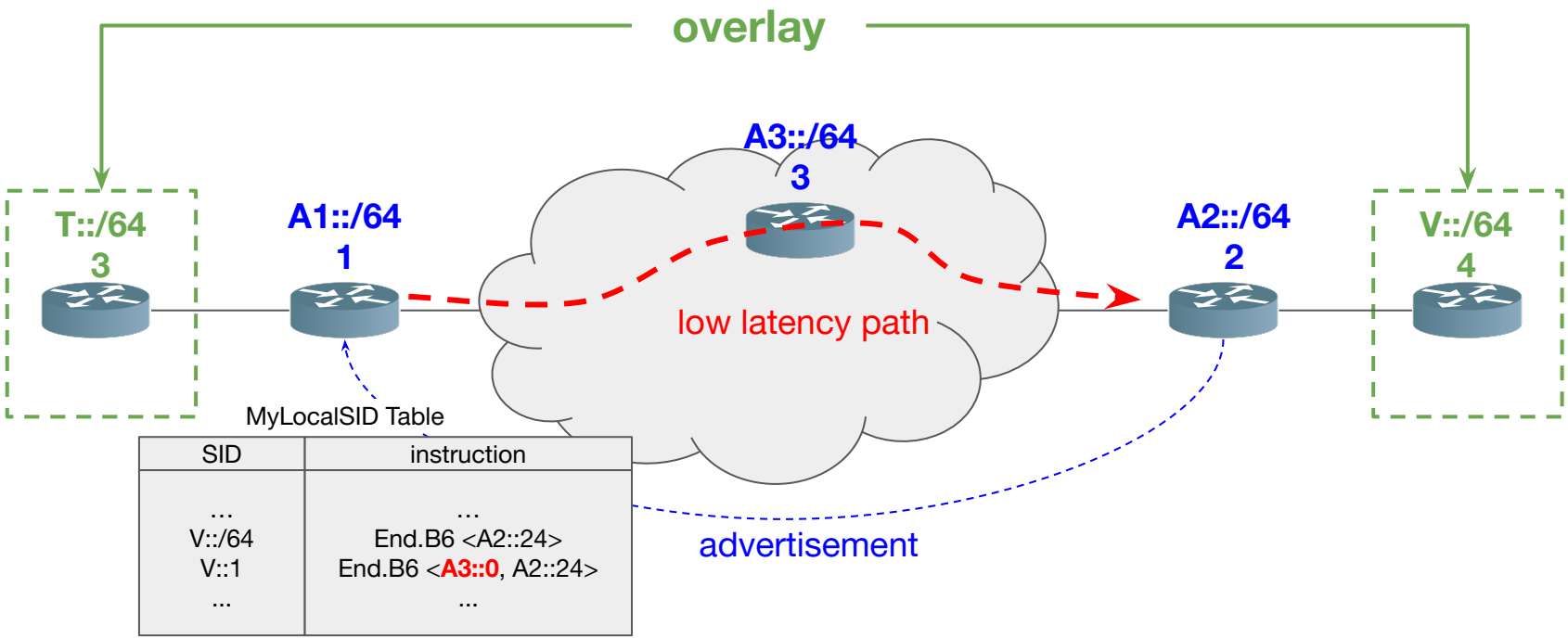
Overlay



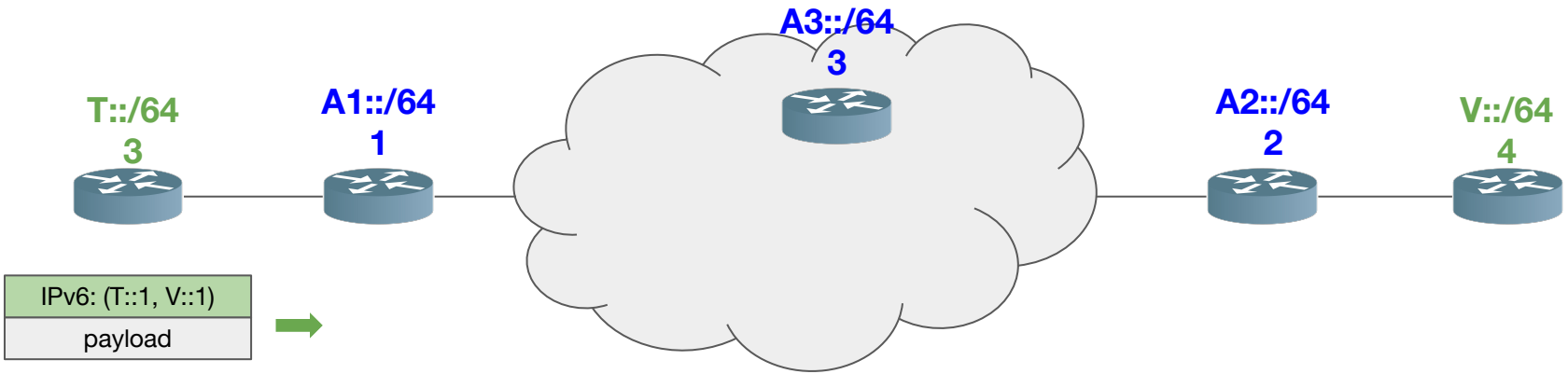
Overlay



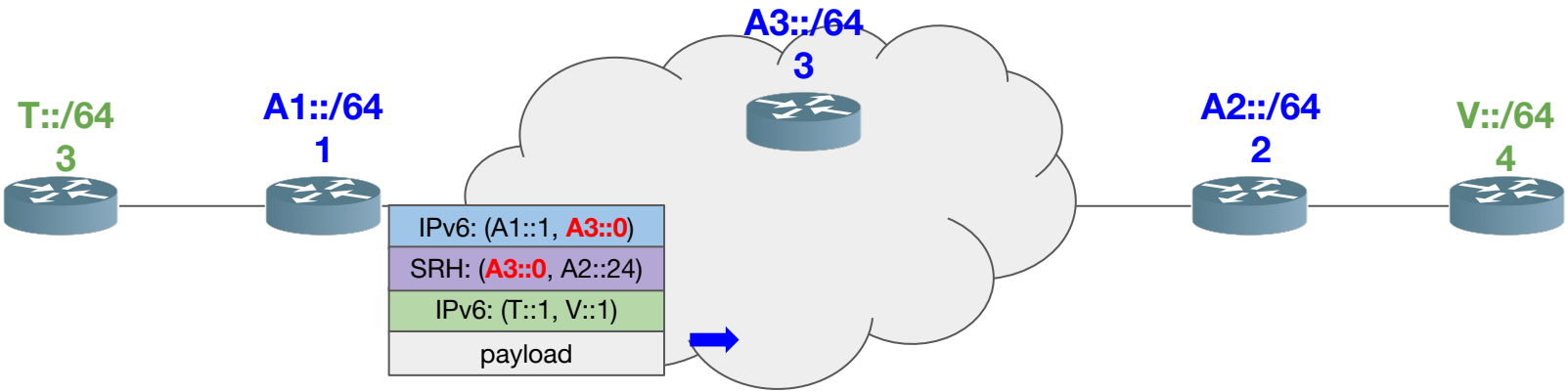
Overlay with underlay SLA



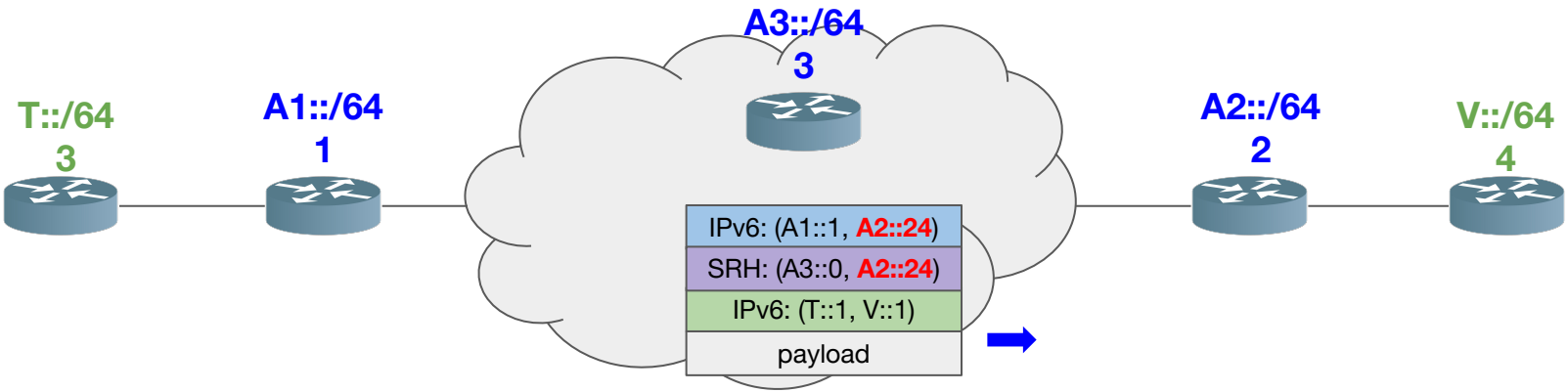
Overlay with underlay SLA



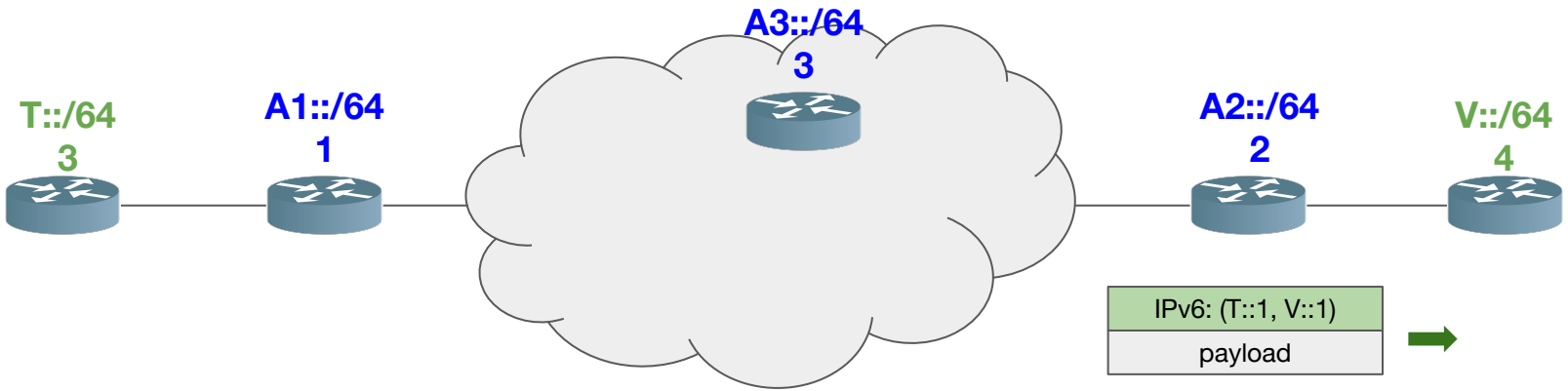
Overlay with underlay SLA



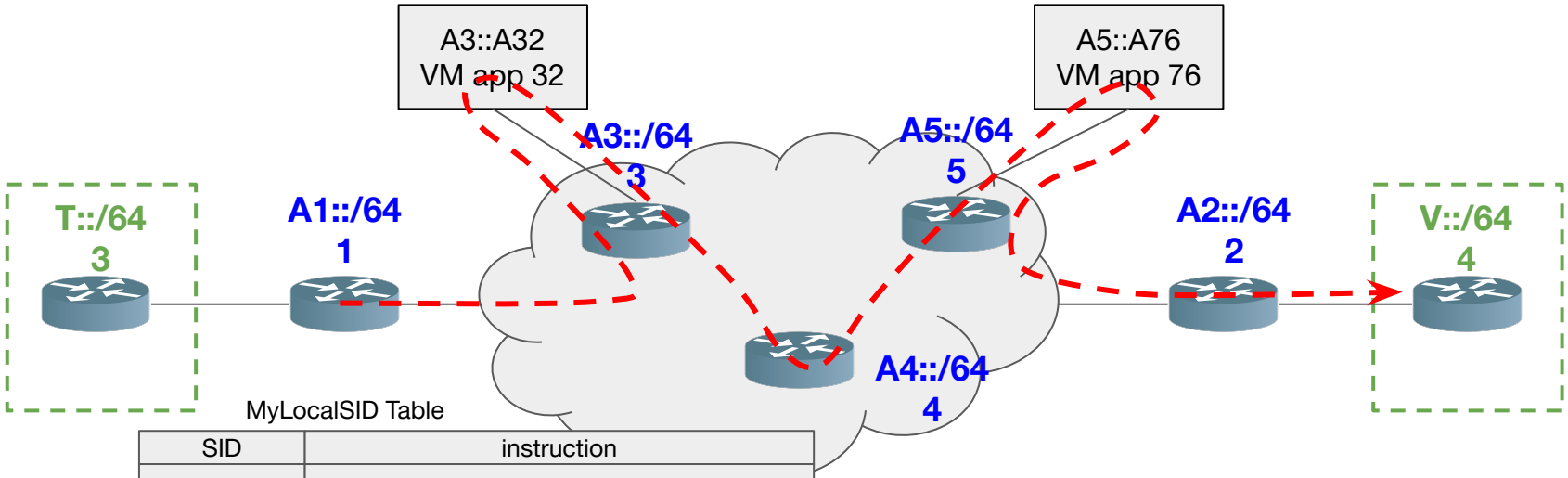
Overlay with underlay SLA



Overlay with underlay SLA



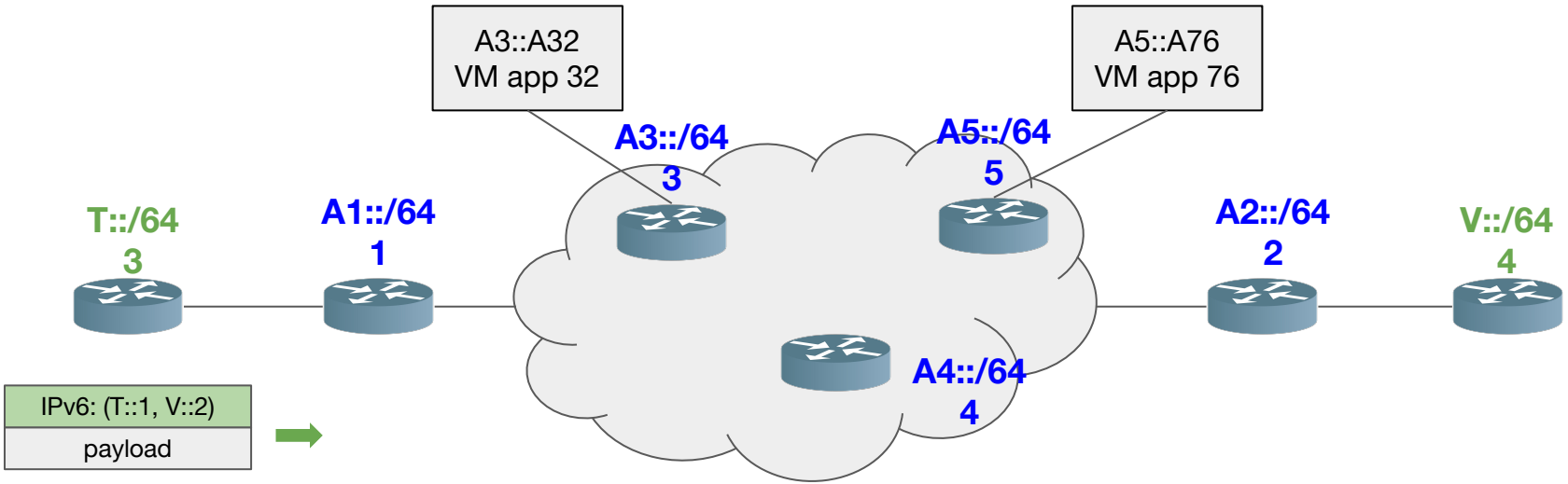
Integrated NFV



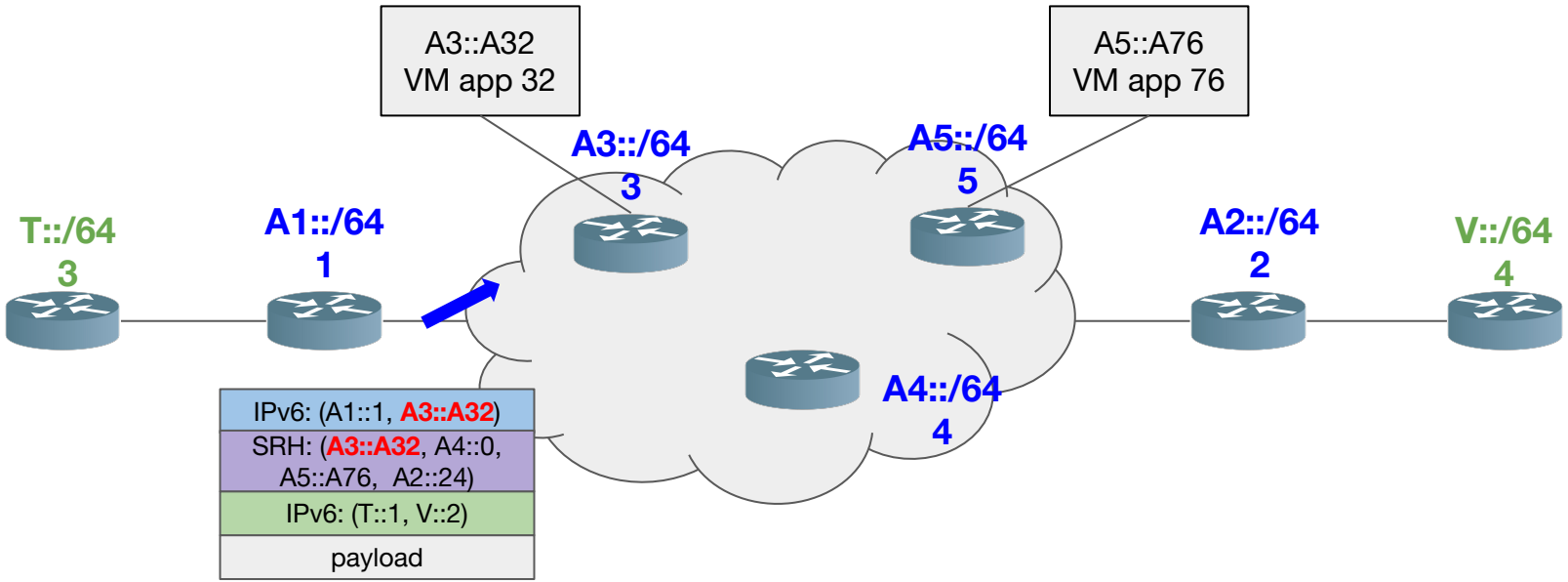
MyLocalSID Table

SID	instruction
… V::/64 V::1 V::2 …	… End.B6 <A2::24> End.B6 <A3::0, A2::24> End.B6 <A3::A32, A4::0, A5::A76, A2::24> …

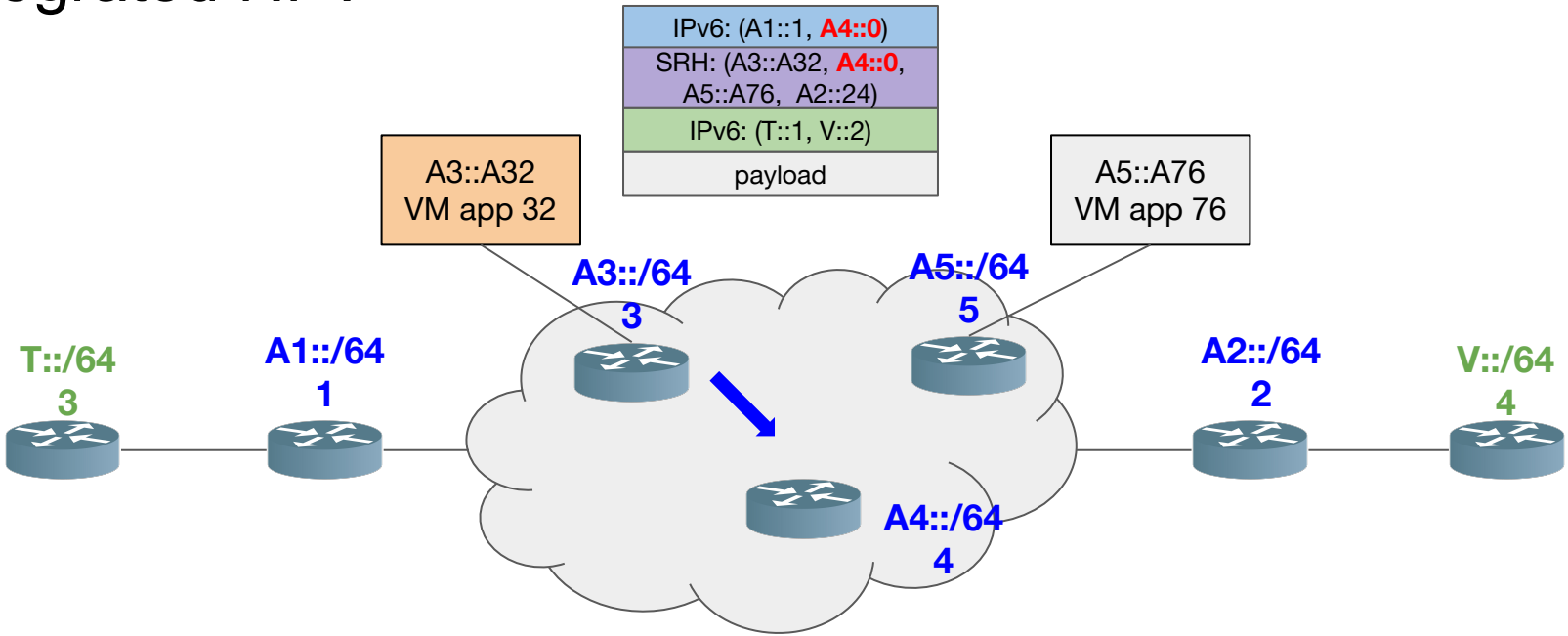
Integrated NFV



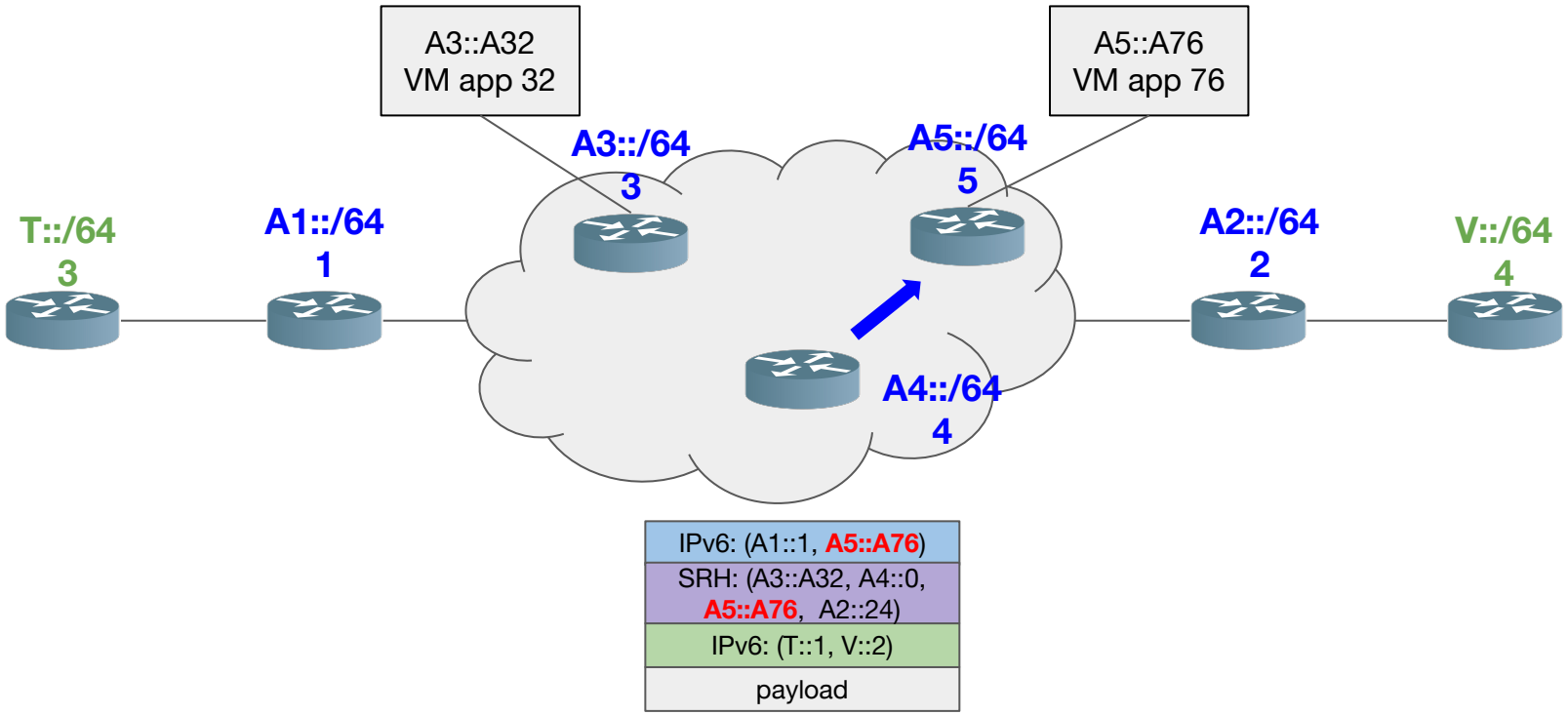
Integrated NFV



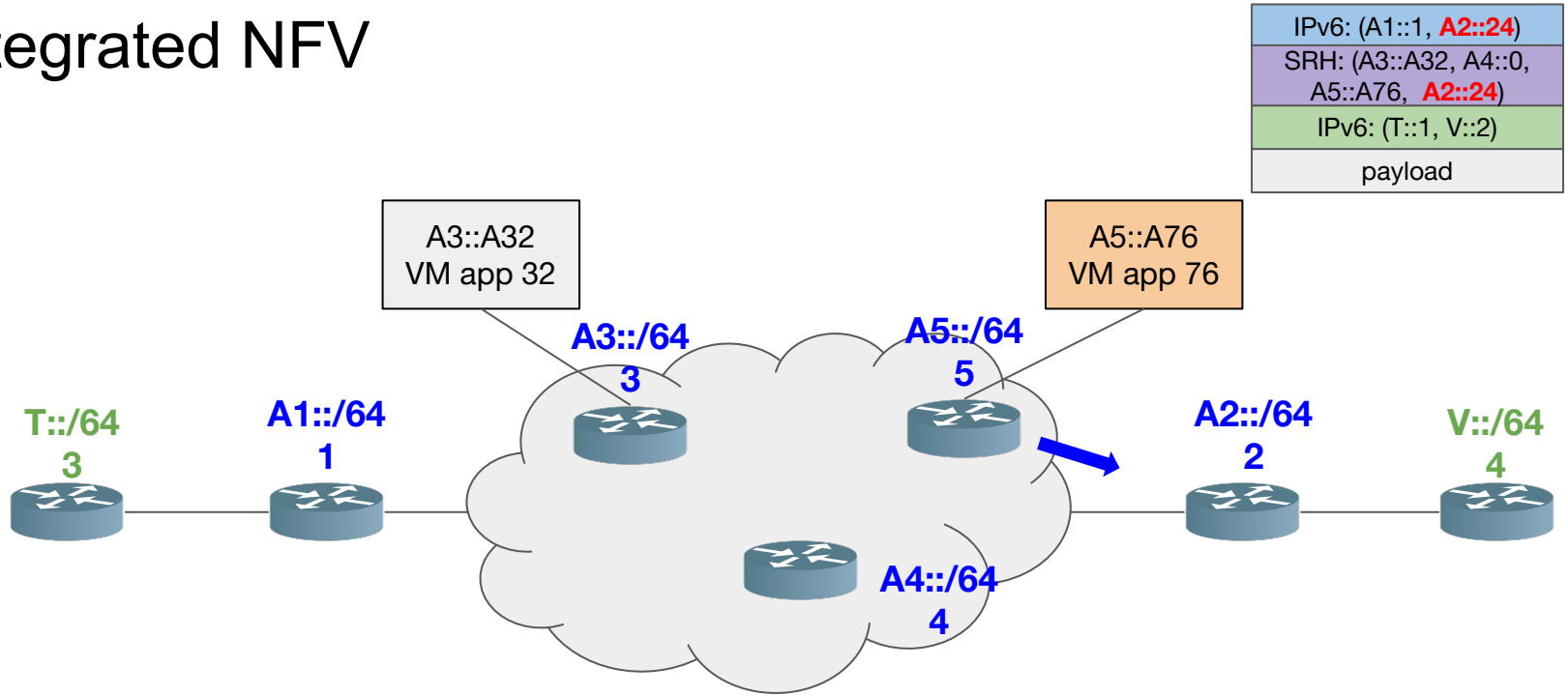
Integrated NFV



Integrated NFV



Integrated NFV



Integrated NFV