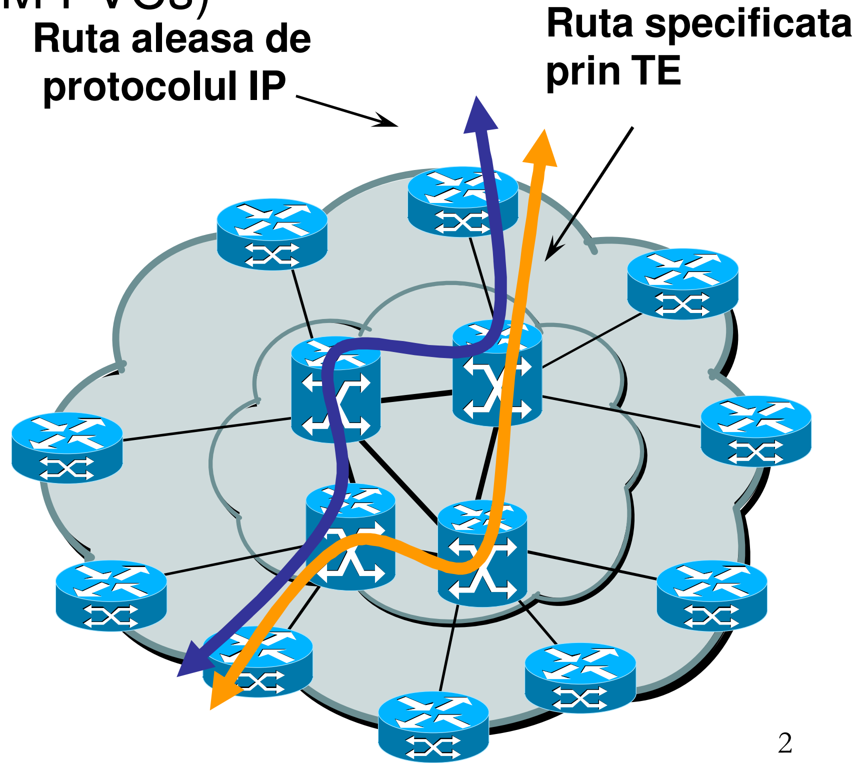
MultiProtocol Label Switching Traffic Engineering MPLS-TE

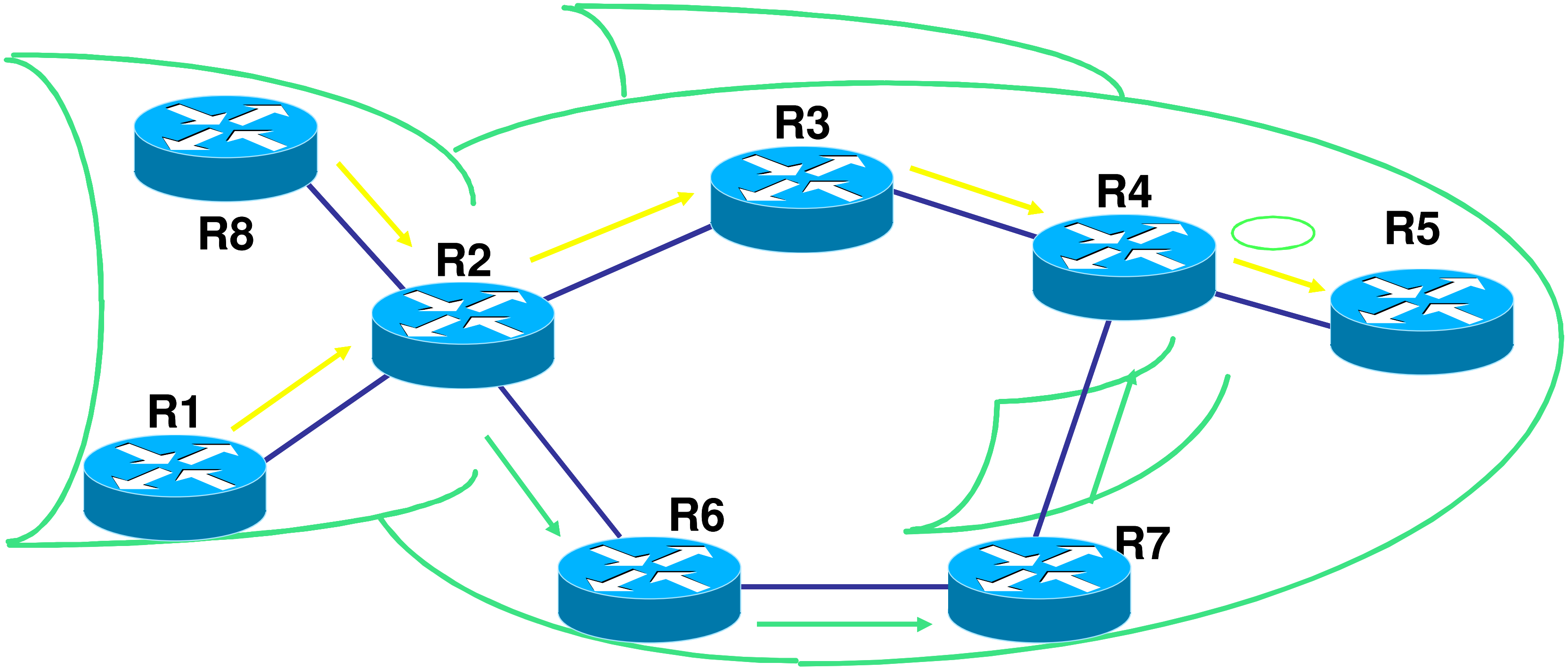
Traffic Engineering

- *TE: "…that aspect of Internet network engineering dealing with the issue of performance evaluation and **performance optimization of operational IP networks** …"*
- Two abstract sub-problems:
  - 1. Define a *traffic aggregate*

    (OC or T-carrier hierarchies, or ATM PVCs)
  - 2. *Map* the traffic aggregate *to an explicit setup path*
- Cannot do this in OSPF or BGP-4
  - OSPF and BGP-4 offer only a SINGLE path!



‘Fish problem’ in TE



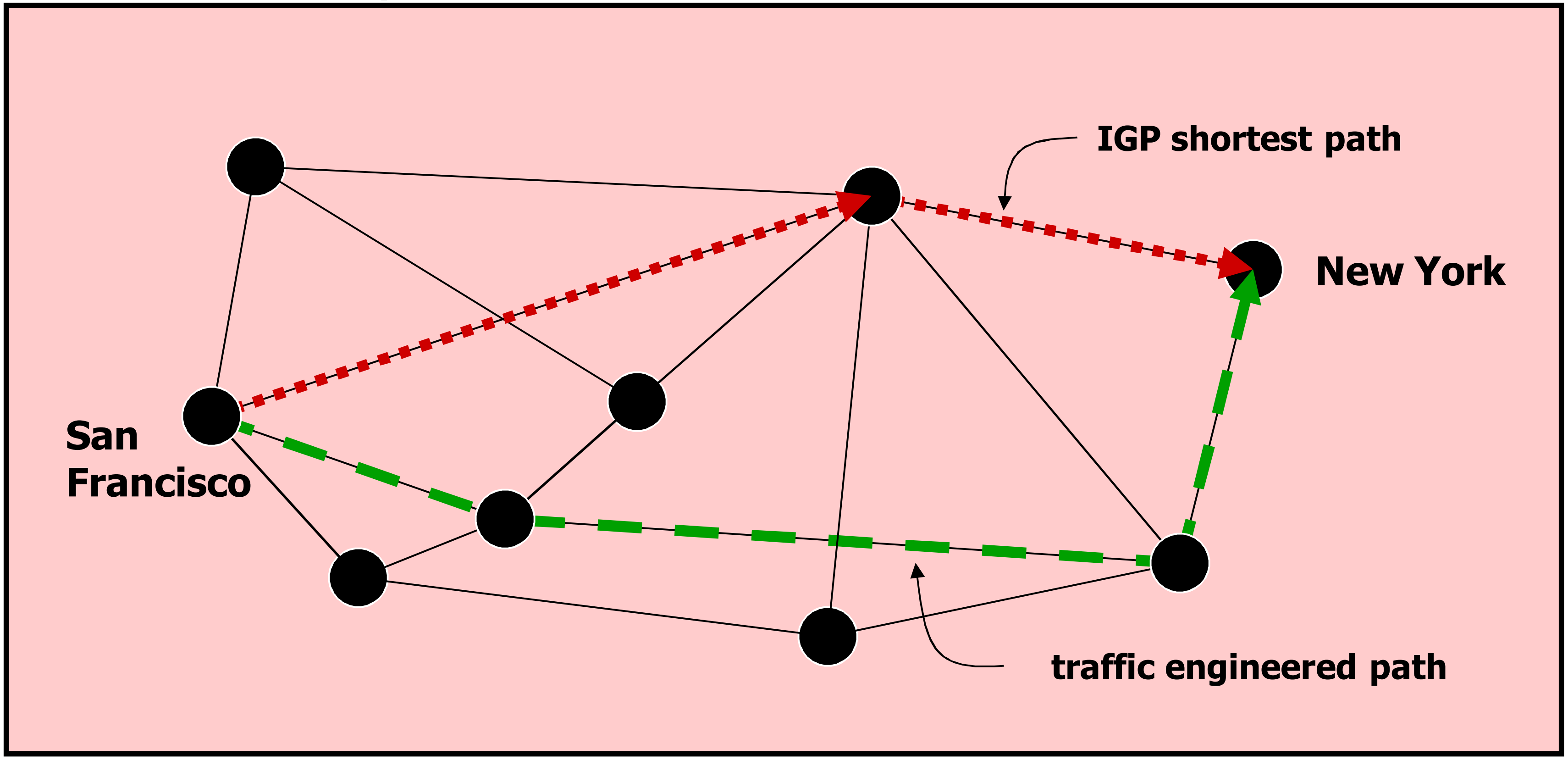
- •IP utilizeaza rutarea pe calea cea mai scurta
- •calea cea mai scurta nu e singura cale
- •celelalte cai pot fi subutilizate in timp ce caile cele mai scurte pot fi suprautilizate

Why not TE with OSPF/BGP?

- Internet connectionless routing protocols designed to find only one route (path)
  - The “connectionless” approach to TE is to change **link weights** in IGP (OSPF, IS-IS) or EGP (BGP-4) protocols
  - Assumptions: Quasi-static traffic, knowledge of demand matrix
- Limitations:
  - Performance is fundamentally limited by the **single** shortest/policy path nature:
    - All flows to a destination prefix mapped **to the same path**
  - Desire to map traffic to **different routes** (eg: for load-balancing reasons) => the single default route MUST be changed
  - Changing parameters (eg: OSPF link weights) changes routes AND changes the traffic mapped to the routes
  - Leads to *extra control traffic* (eg: OSPF floods or BGP-4 update message), *convergence problems* and *routing instability*
- Summary: Traffic mapping *coupled* with route availability in OSPF/BGP
  - MPLS *de-couples* traffic trunking (traffic aggregation) from path setup

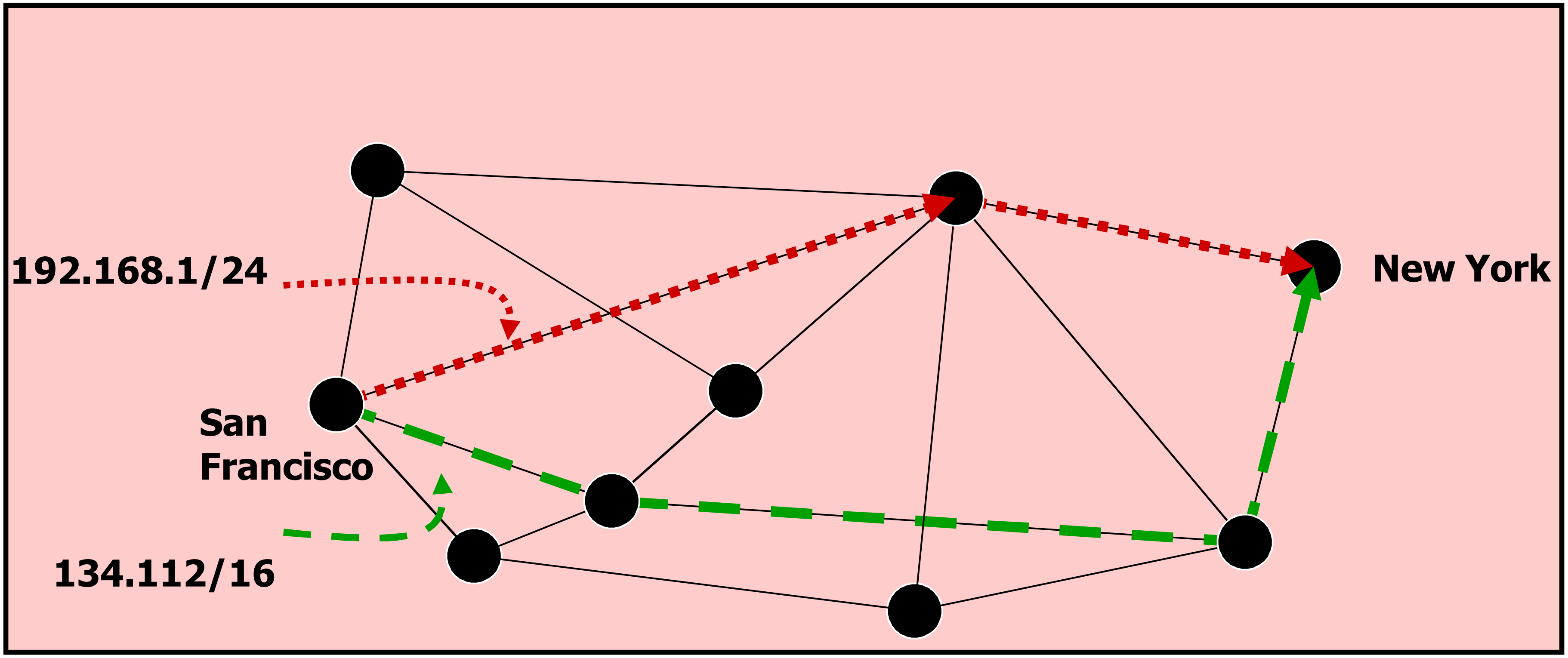
Traffic Engineering with MPLS

- Engineer unidirectional paths through your network without using (compulsory) the IGP's shortest path calculation



Traffic Engineering with MPLS

- IP prefixes (or traffic aggregates) can now be bound to MPLS Label Switched Paths (LSPs)

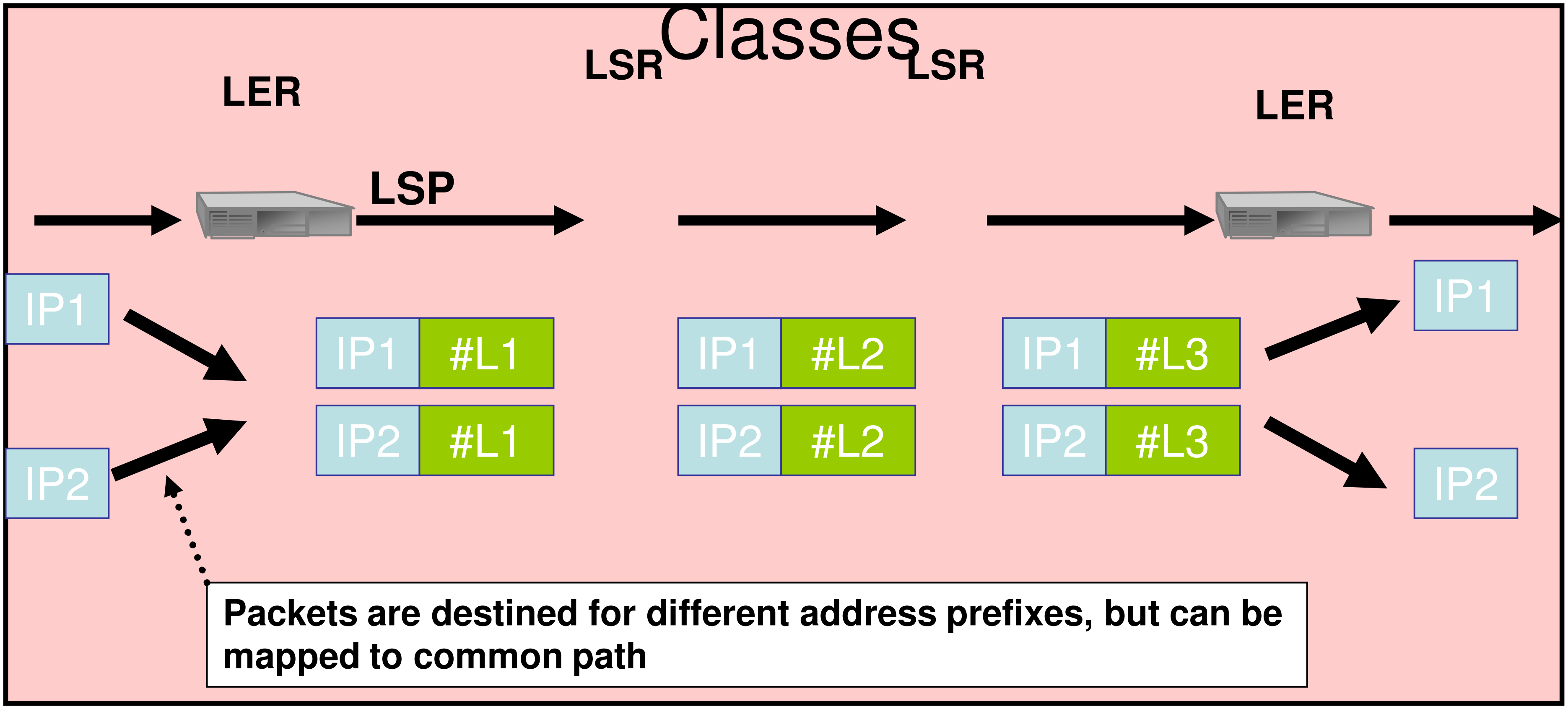


Reminding

In the traditional layer 3 forwarding paradigm, as a packet travels from one router to the next, an independent forwarding decision is made at each hop. The IP network-layer header is analyzed and the next hop is chosen based on this analysis and on the information in the routing table.

In the traffic engineering environment, the analysis of the packet header is performed just once—right before the packet enters the engineered path. The packet is assigned a label, which is a short, fixed-length value placed at the front of the packet. Routers in the traffic engineering path use labels as lookup indicies into the label forwarding table. For each label, this table stores forwarding information such as the router interface for which a labeled packet should be forwarded

Traffic Aggregates: Forwarding Equivalence Classes



Packets are destined for different address prefixes, but can be mapped to common path

- FEC = "A subset of packets that are all treated the same way by a router"
- The concept of FECs provides for a great deal of flexibility and scalability
- In conventional routing, a packet is assigned to a FEC at each hop (i.e. L3 look-up), in MPLS it is only done once, at the network ingress

Remember

The “Forwarding Equivalence Class” is an important concept in MPLS. An FEC is any subset of packets that are treated the same way by a router. By “treated” this can mean: forwarded out the same interface with the same next hop and label. It can also mean: given the same class of service, output on same queue, given same drop preference, and any other option available to the network operator.

When a packet enters the MPLS network at the ingress node, the packet is mapped into an FEC. The mapping can also be done on a wide variety of parameters: address prefix (or host), source/destination address pair, or ingress interface. This greater flexibility adds functionality to MPLS that is not available in traditional IP routing.

FECs also allow for greater scalability in MPLS.

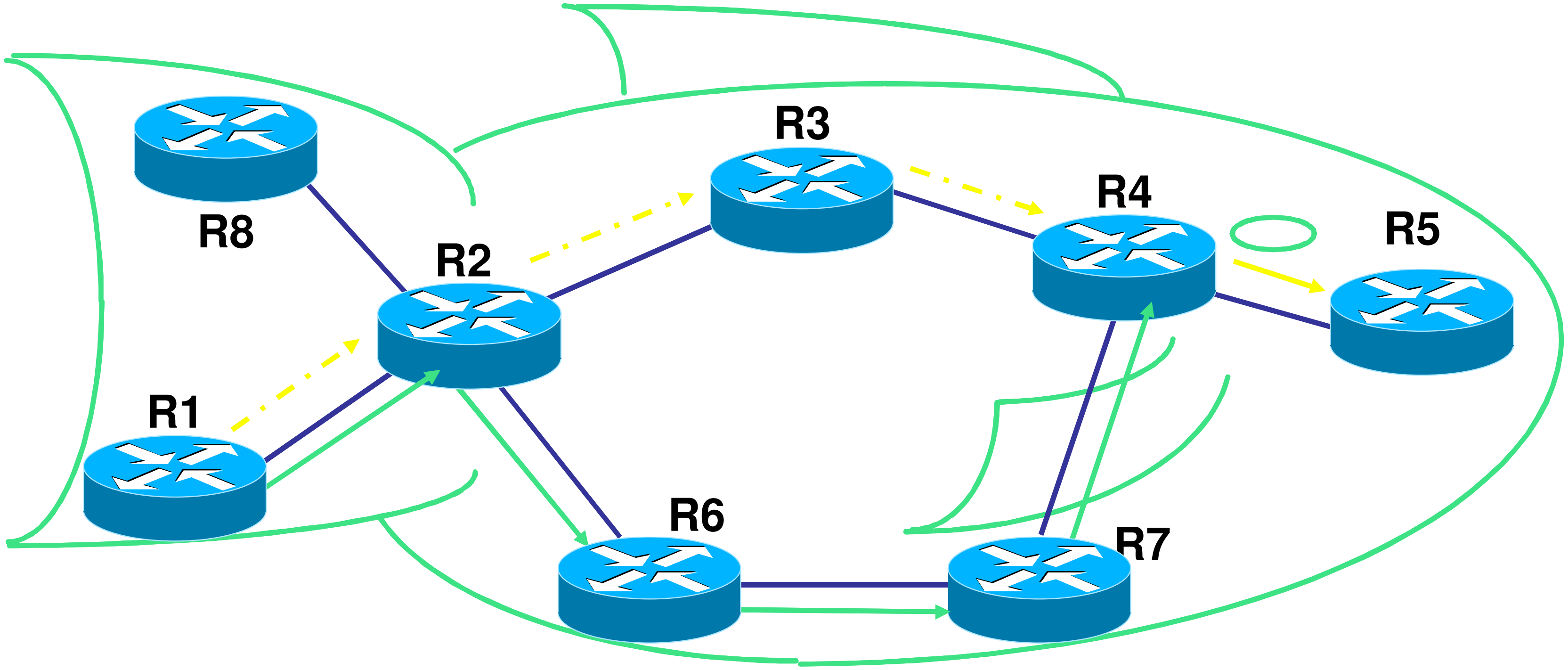
With MPLS, the aggregation of flows into FECs of variable granularity provides scalability that meets the demands of the public Internet as well as enterprise applications.

In the current Label Distribution Protocol specification, only three types of FECs are specified:

- IP Address Prefix
- Router ID
- Flow (port, dest-addr, src-addr etc.)

The specification states that new elements can be added as required.

LSP as a tunnel



Etichetele pot fi utilizate pentru stabilirea tunelelor

→ **ruta normala R1->R2->R3->R4->R5**

→ **tunel: R1->R2->R6->R7->R4**

MPLS as a Signaled TE Approach

- Features:
  - In MPLS, the choice of a route (and its setup) is orthogonal to the problem of traffic mapping onto a route
  - Signaling maps global IDs (addresses, path-specification) to local IDs (labels)
  - FEC mechanism for defining traffic aggregates, label stacking for multi-level opaque tunneling
- Issues:
  - Requires extensive upgrades in the network
  - Hard to inter-network beyond area boundaries
  - Very hard to go beyond AS boundaries (even in same organization)
  - Impossible for inter-domain routing across multiple organizations => inter-domain TE has to be connectionless

Hop-by-Hop vs. Explicit Routing

Hop-by-Hop Routing

- Distributes routing of control traffic
- Builds a set of trees either fragment by fragment like a random fill, or backwards, or forwards in organized manner.
- Reroute on failure impacted by convergence time of routing protocol
- Existing routing protocols are destination prefix based
- Difficult to perform traffic engineering, QoS-based routing

Explicit Routing

- Source routing of control traffic
- Builds a path from source to dest
- Requires manual provisioning, or automated creation mechanisms.
- LSPs can be ranked so some reroute very quickly and/or backup paths may be pre-provisioned for rapid restoration
- Operator has routing flexibility (policy-based, QoS-based,
- Adapts well to traffic engineering

Explicit routing shows great promise for traffic engineering

RSVP: “Resource reSerVation Protocol”

- A generic QoS signaling protocol
- An Internet control protocol
  - Uses IP as its network layer
- Originally designed for host-to-host
- Uses the IGP to determine paths
- RSVP is <u>not</u>
  - A data transport protocol
  - A routing protocol
- RFC 2205

Remember

The Resource Reservation Protocol (RSVP) is a generic signaling protocol that was originally designed to be used by applications to request and reserve specific Quality of Service (QoS) requirements across an internetwork. Resources are reserved hop-by-hop across the internetwork; each router receives the resource reservation request, establishes and maintains the necessary state for the data flow (if the requested resources are available), and forwards the resource reservation request to the next router along the path.

As this behavior implies, RSVP is an internetwork control protocol, similar to ICMP, IGMP, and routing protocols. It does not transport application data, nor is it a routing protocol. RSVP utilizes unicast and multicast routing protocols to discover paths through the internetwork by consulting existing routing tables.

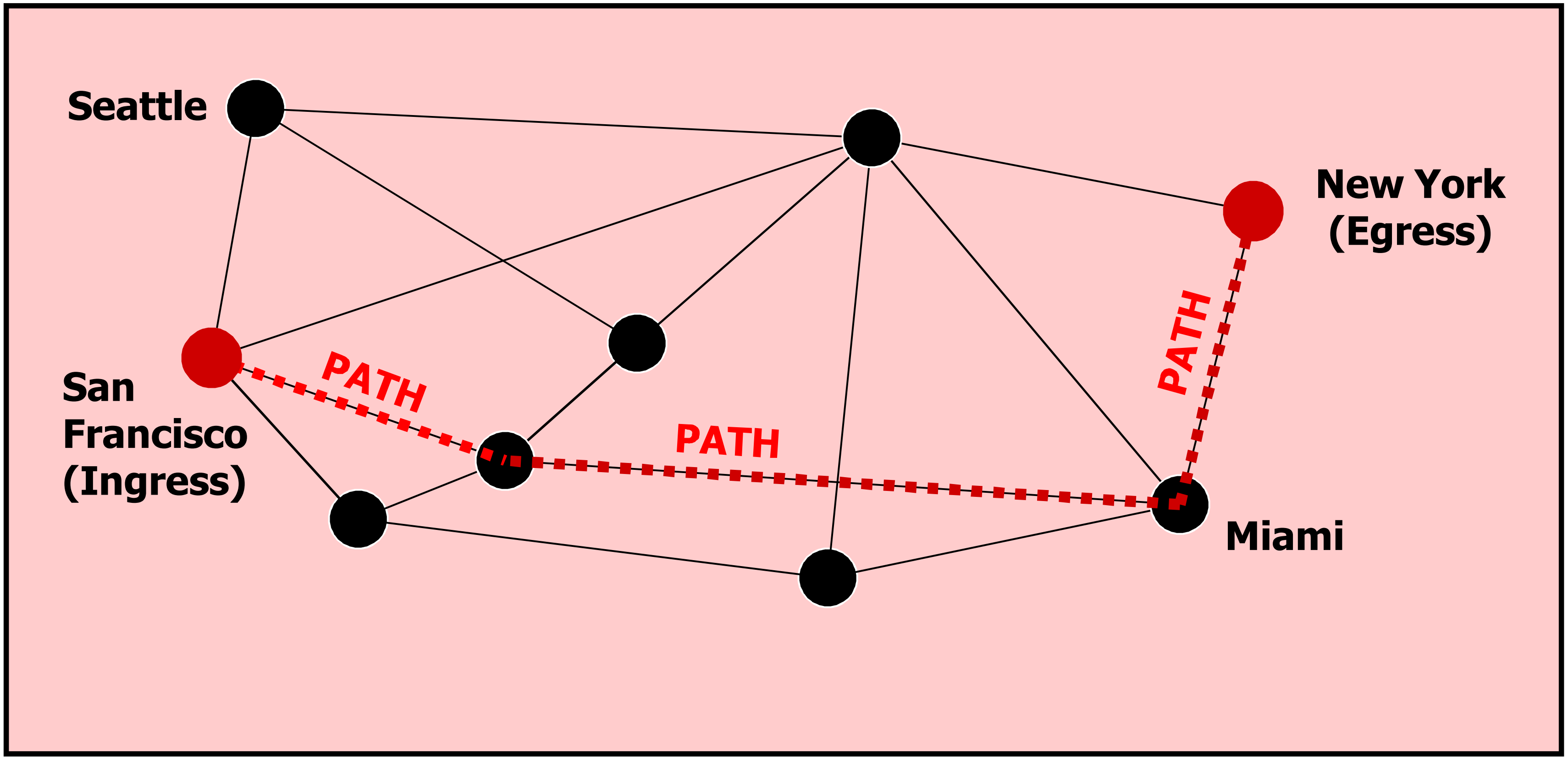
The present document describing RSVP is RFC 2205, *Resource Reservation Protocol (RSVP)-- Version 1 Functional Specification*

RSVP: Internet Signaling

- Creates and maintains distributed reservation state
- De-coupled from routing & also able to support IP multicast model:
  - Multicast trees setup by routing protocols, not RSVP (unlike ATM or telephony signaling)
- Key features of RSVP:
  - Receiver-initiated: scales for multicast
  - Soft-state: reservation times out unless refreshed
- Latest paths discovered through “PATH” messages (forward direction) and used by RESV mesgs (reverse direction).
  - Again dictated by needs of de-coupling from IP routing and to support IP multicast model

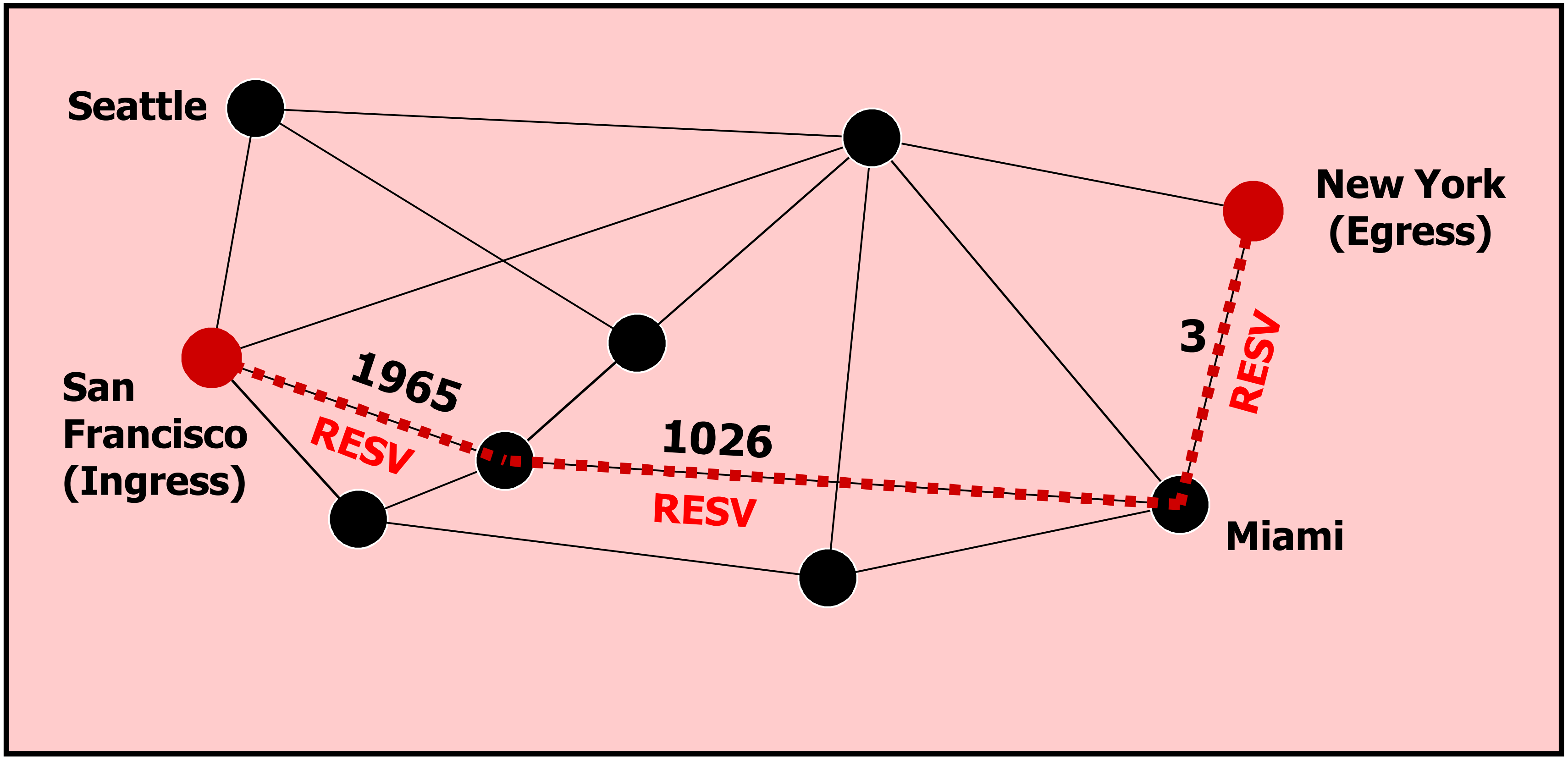
RSVP Path Signaling Example

- Signaling protocol sets up path from San Francisco to New York, reserving bandwidth along the way



RSVP Path Signaling Example

- Once path is established, signaling protocol assigns label numbers in reverse order from New York to San Francisco



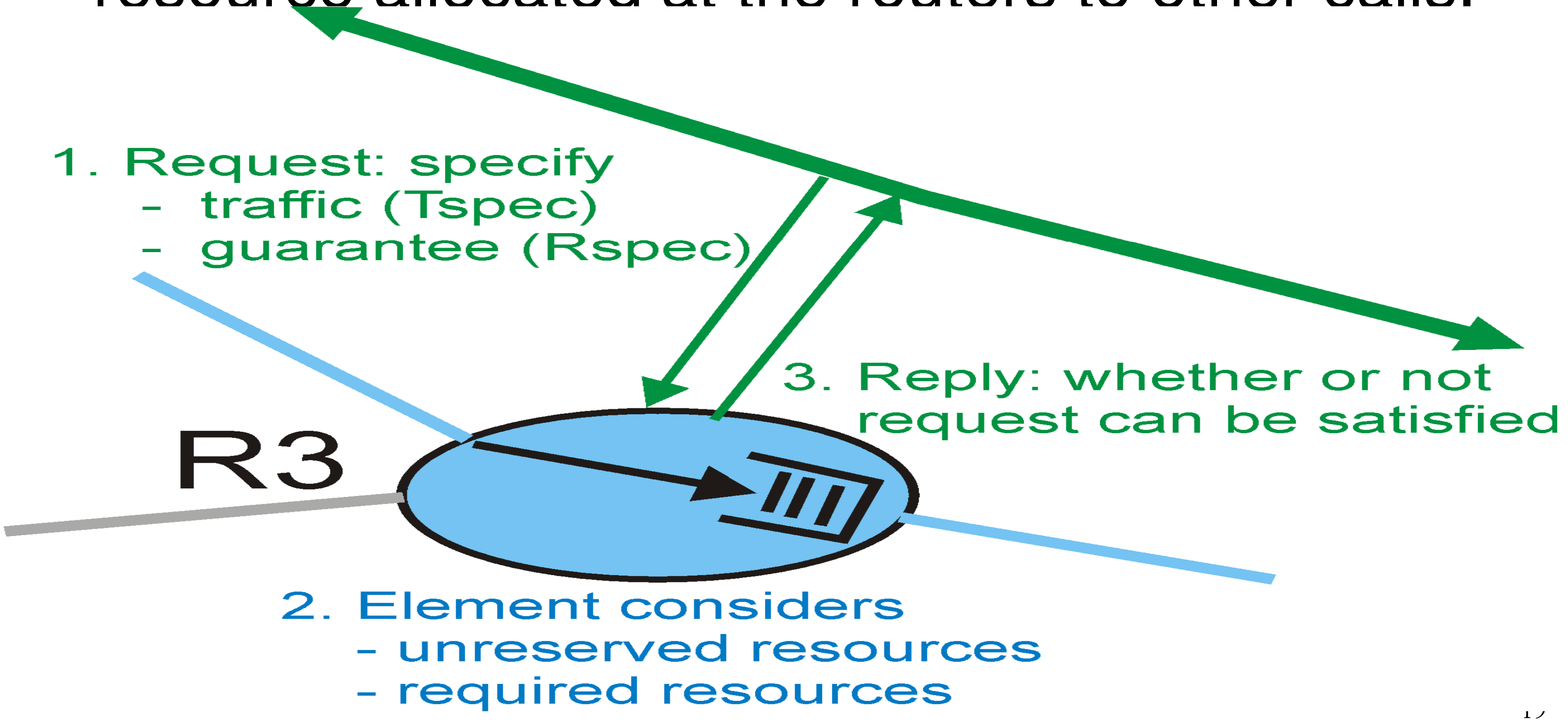
Call Admission

- Session must first declare its QoS requirements and characterize the traffic it will send through the network
- **R-spec**: defines the QoS being requested
- **T-spec**: defines the traffic characteristics
- A signaling protocol is needed to carry the **R-spec** and **T-spec** to the routers where reservation is required;

RSVP is a leading candidate for such signaling protocol

Call Admission

- Call Admission: routers will admit calls based on their R-spec and T-spec and base on the current resource allocated at the routers to other calls.



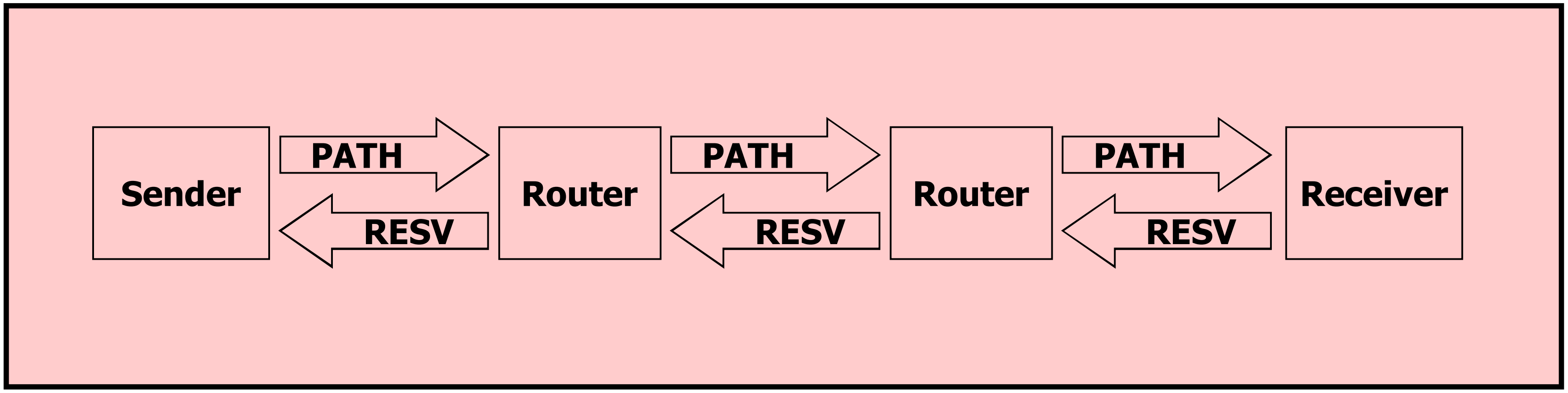
1. Request: specify
   - traffic (Tspec)
   - guarantee (Rspec)

2. Element considers
   - unreserved resources
   - required resources

3. Reply: whether or not request can be satisfied

Basic RSVP Path Signaling

- Reservation for simplex (unidirectional) flows
- Ingress router initiates connection
- “Soft” state
  - Path and resources are maintained dynamically
  - Can change during the life of the RSVP session
- Path message sent downstream
- Resv message sent upstream



Remember

RSVP requests resources for simplex data flows. Each reservation is made for a data flow from a specific sender to a specific receiver. While RSVP Path messages are exchanged between the sender and receiver, the resulting path itself is unidirectional.

Although the application data flow is from the sender to the receiver, the reservation itself is receiver-initiated. The sender notifies the receiver of a pending flow and characterizes the flow, and the receiver is responsible for requesting the resources. This design choice was made to accommodate heterogeneous receiver requirements, and for multicast flows in which multiple receivers will be joining and leaving a multicast group.

RSVP requests made to routers along the transit path cause each router to either reject the request for lack of resources, or establish a *soft state*. This is in contrast to a *hard state*, which is associated with virtual connections that remain established for the duration of the data transfer. Soft state means that the logical path set up by RSVP is not necessarily associated with a physical path through the internetwork. The logical path may change during its lifetime as the result of the sender changing the characterization of the traffic, causing the receiver to modify its reservation request, or the failure of a transit router.

The soft state is maintained by refreshing the soft state periodically. In standard RSVP implementations, this is done by sending PATH and RESV messages across the path.

MPLS Extensions to RSVP (RSVP-TE)

- **Path and Resv message objects**
  - Explicit Route Object (ERO)
  - Label Request Object
  - Label Object
  - Record Route Object
  - Session Attribute Object
  - Tspec Object (traffic specs)
- For more detail on contents of objects:

  daft-ietf-mpls-rsvp-lsp-tunnel-04.txt

  Extensions to RSVP for LSP Tunnels

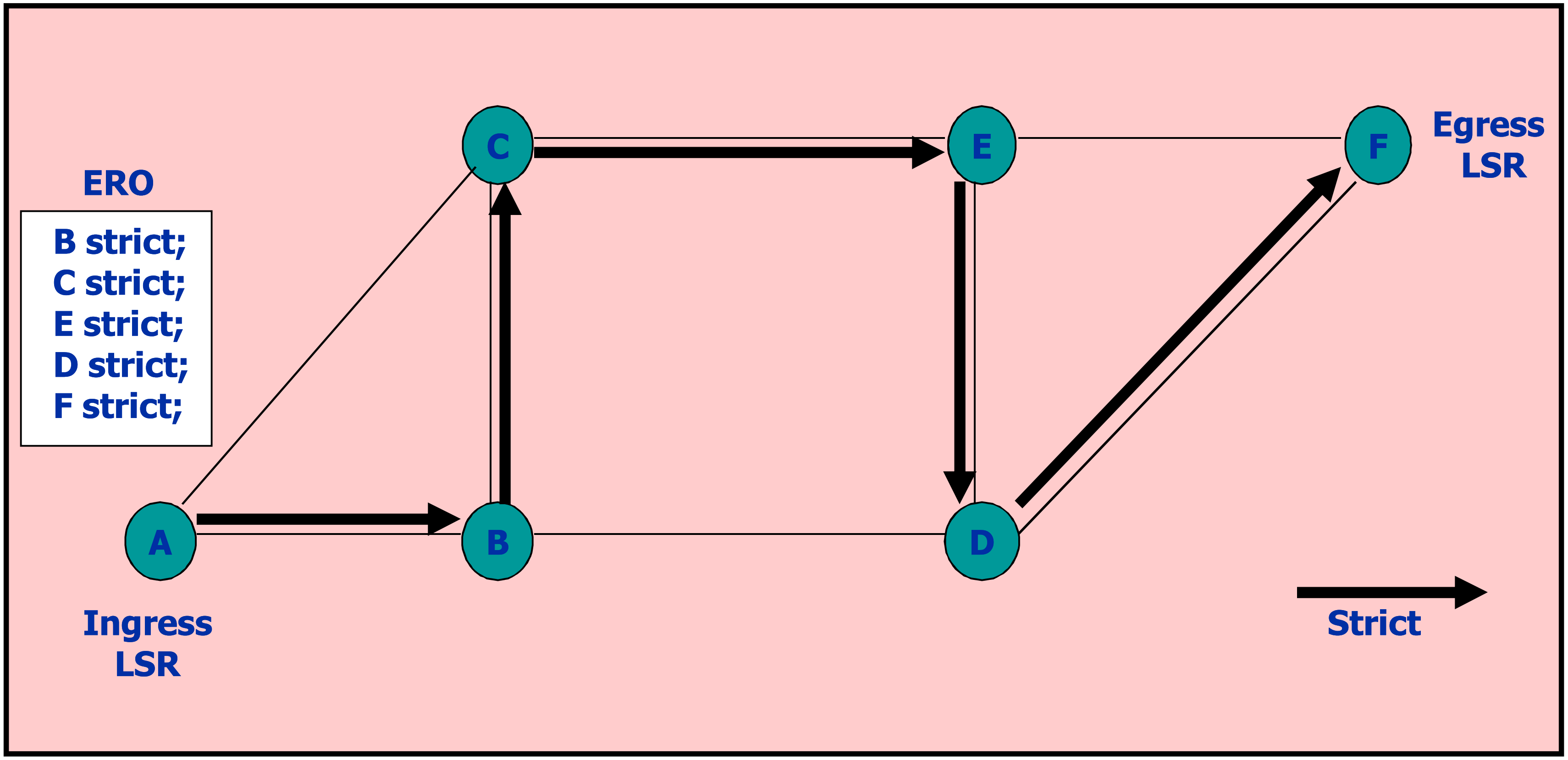
Explicit Route Object

- Used to specify the explicit route (list of LSRouters between ingress to egress endpoints) RSVP Path messages take for setting up LSP
- Can specify loose or strict routes
  - Loose routes rely on routing table to find destination
  - Strict routes specify the directly-connected next router
- A route can have both loose and strict components

The *Explicit Route Object* (ERO) is added to an RSVP Path message by the ingress LSR to specify an explicit route for the message, independent of conventional IP routing. The ERO is only to be used when all routers along the explicit route support RSVP and the ERO. The ERO is also only intended to be used for unicast situations.

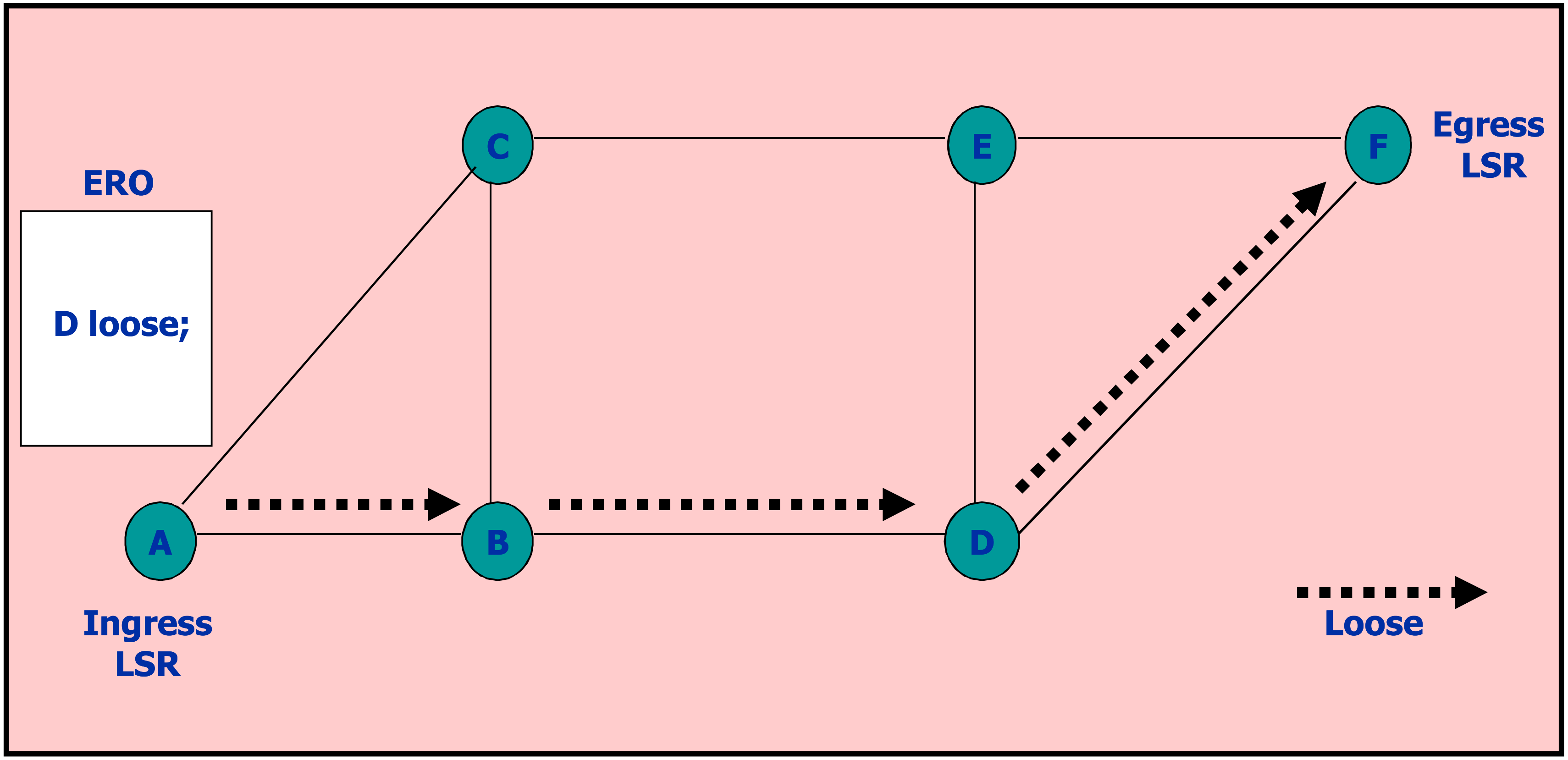
ERO: Strict Route

- Next hop must be directly connected to previous hop



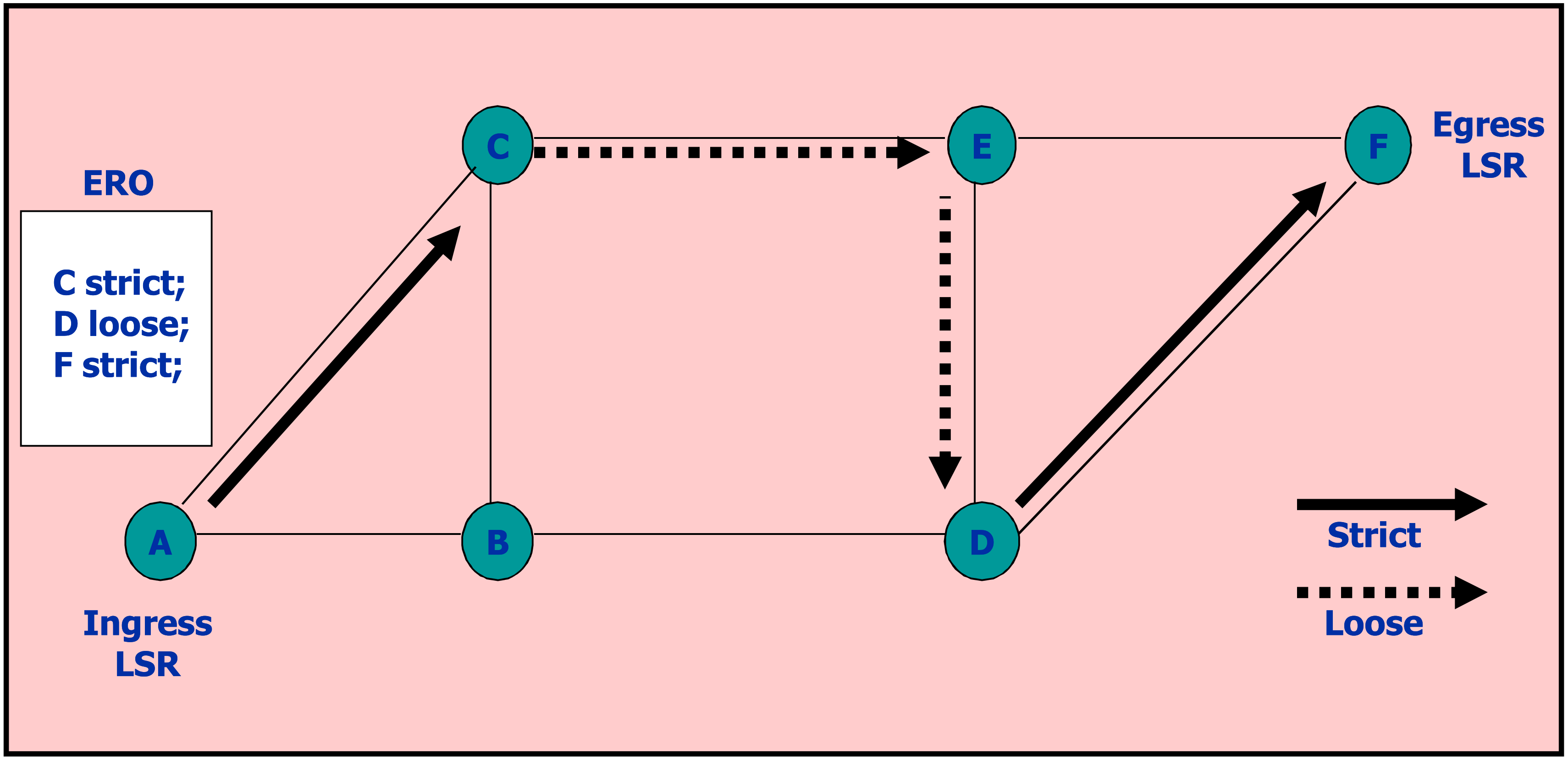
ERO: Loose Route

- Consult the routing table at each hop to determine the best path: similar to IP routing option concept



ERO: Strict/Loose Path

- Strict and loose routes can be mixed



Label Objects

- Label Request Object
  - Added to PATH message at ingress LSR
  - Requests that each LSR provide label to upstream LSR

- Label Object
  - Carried in RESV messages along return path upstream
  - Provides label to upstream LSR

More text…

The *Label Request Object* can be added to the PATH message by the ingress LSR to request that intermediate routers provide a label binding for the path. The object provides an indication of the network layer protocol that is to be carried over the path, permitting non-IP network layer protocols to be sent down the path. **When a PATH message containing a Label Request Object arrives at an LSR, the LSR allocates a label for upstream propagation and stores it as part of the path state. When the corresponding RESV message arrives, the label is placed in its Label Object.**

The *Label Object* is carried in RESV messages. The Label Object carries a label, and when an LSR receives a RESV message it uses the label as the outgoing label associated with the sender. The LSR allocates a new label, or uses the label allocated and stored in path state as a result of the Label Request Object, and places it in the Label Object of the RESV message to be sent to the previous hop. In this way, the Label Object supports the distribution of labels from downstream nodes to upstream nodes.

Record Route Object— PATH Message

- Added to PATH message by ingress LSR
- Adds outgoing IP address of each hop in the path
  - In downstream direction

- Loop detection mechanism
  - Sends “Routing problem, loop detected” PathErr message
  - Drops PATH message

The *Record Route Object* can be added to Path messages to allow the sender to receive information about the actual path the LSP traverses. Each node along the path records its IP address in the RRO, and the RRO is returned to the sender in Resv messages.

Session Attribute Object

- Added to PATH message by ingress router
- Controls LSP
  - Priority
  - Preemption
  - Fast-reroute
- Identifies session
  - ASCII character string for LSP name

Adjacency Maintenance—Hello Message

- New RSVP extension: improved RSVP for hellos!
  - – Hello messages
    - Hello Request
    - Hello Acknowledge

- Rapid node to node failure detection
  - – Asynchronous updates
  - – 3 second default update timer
  - – 12 second default dead timer

Path Maintenance — Refresh Messages

- Maintains reservation of each LSP
- Sent every 30 seconds by default
- Consists of PATH and RESV messages

RSVP Message Aggregation

- Bundles up to 30 RSVP messages within single PDU
- Controls
  - Flooding of PathTear or PathErr messages
  - Periodic refresh messages (PATH and RESV)
- Enhances protocol efficiency and reliability
- Disabled by default

Traffic Engineering: Constrained Routing

Signaled vs Constrained LSPs

- Common Features
  - Signaled by RSVP
  - MPLS labels automatically assigned
  - Configured on ingress router only
- Signaled LSPs
  - CSPF not used (i.e. normal IP routing is used)
  - User configured ERO handed to RSVP for signaling
  - RSVP consults routing table to make next hop decision
- Constrained LSPs
  - Constrained Shortest Path First (CSPF) used
  - Full path computed by CSPF at ingress router
  - Complete ERO handed to RSVP for signaling

Constrained Shortest Path First Algorithm

- Modified “shortest path first” algorithm
- Finds shortest path based on IGP metric while satisfying additional QoS constraints
- Integrates TED (Traffic Engineering Database)
  - IGP topology information
  - Available bandwidth
  - Link color (the administrative groups to which the interface belongs; an administrative group allows the formation of policies that dictate what links an individual LSP can or cannot traverse)
- Modified by administrative constraints
  - Maximum hop count
  - Bandwidth
  - Strict or loose routing
  - Administrative groups

CSPF algorithm – more text!

A link state protocol (OSPF, IS-IS) can be easily extended to include other local information in the protocol data unit it floods. So to support MPLS traffic engineering, both OSPF and IS-IS have extensions that enable each router to flood extra information about each of its interfaces:

- Maximum bandwidth
- Maximum reservable bandwidth (the portion of the maximum bandwidth that can be reserved for exclusive use by an individual LSP)
- Unreserved bandwidth (the percentage of the maximum reservable bandwidth not yet reserved by any LSP)
- An interface metric that can be used separately from the IGP interface metric
- The administrative groups to which the interface belongs (Commonly called “link color,” an administrative group allows the formation of policies that dictate what links an individual LSP can or cannot traverse)

When this information is flooded, each LSR stores the information in a database called **the traffic engineering database**. When you configure an LSP at an ingress router, you can specify constraints based on any or all of that flooded information: the amount of bandwidth the LSP requires, the cost of the path, and the link “colors” the LSP must or must not use.

The ingress LSR then runs a special version of SPF called Constrained Shortest Path First (CSPF) that takes as its input both the information in the traffic engineering database and the constraints you configure.

Where the results of the SPF calculation are used to make entries in the unicast routing table, RSVP-TE takes the ERO resulting from the CSPF calculation and sends PATH messages to the egress to reserve resources for the LSP.

The egress LSP sends RESV messages back to the ingress to distribute the labels; this is what actually sets up the LSP.

Once this process is complete, RSVP can make entries into the unicast routing table that indicates the LSP as a virtual link to the egress LSR.

Computing the ERO

- Ingress LSR passes user defined restrictions to CSPF
  - Strict and loose hops
  - Bandwidth constraints
  - Admin Groups
- CSPF algorithm
  - Factors in user defined restrictions
  - Runs computation against the TED
  - Determines the shortest path
- CSPF hands full ERO to RSVP for signaling

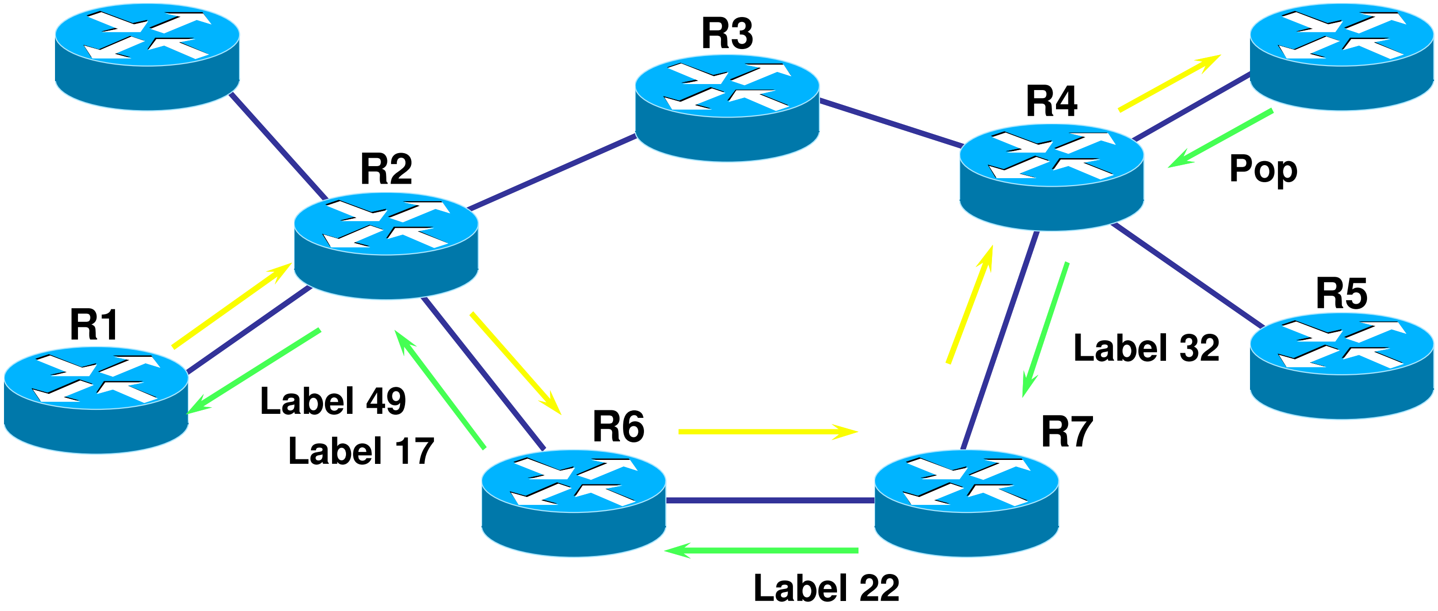
Computing the ERO (more text)

The ingress router determines the physical path for each LSP by applying a Constrained Shortest Path First (CSPF) algorithm to the information in the TED. CSPF is a shortest-path-first algorithm that has been modified to take into account specific restrictions when calculating the shortest path across the network. Input into the CSPF algorithm includes:

- Topology link-state information learned from the IGP and maintained in the TED
- Attributes associated with the state of network resources (such as total link bandwidth, reserved link bandwidth, available link bandwidth, and link color) that are carried by IGP extensions and stored in the TED
- Administrative attributes required to support traffic traversing the proposed LSP (such as bandwidth requirements, maximum hop count, and administrative policy requirements) that are obtained from user configuration

As CSPF considers each candidate node and link for a new LSP, it either accepts or rejects a specific path component based on resource availability or whether selecting the component violates user policy constraints. The output of the CSPF calculation is an **explicit route** consisting of a sequence of router addresses that provides the shortest path through the network that meets the constraints. This explicit route (ERO) is then passed to the signaling component (MPLS), which **establishes forwarding state in the routers along the LSP.**

Path Setup - Example



→

Setup: Path (ERO = R1->R2->R6->R7->R4->R9)

←

Reply: Resv communicates labels and
reserves bandwidth on each link