MPLS – Multiprotocol Label Switching

Overview

The slides are based on:

- A set of slides developed by MPLS Forum.
- *MPLS Technology and Applications*, B. Davie and Y. Rekhter, Morgan Kaufman, 2001.
- *Traffic Engineering with MPLS* by E. Osborne and A. Simha, Cisco Press 2003.
- *IP Switching and Routing Essentials*, S. Thomas, Wiley, 2002.
- *Communication Networks* by & A. Leon-Garcia & I. Widjaja, McGraw-Hill, 2000.

MPLS – How It All Started

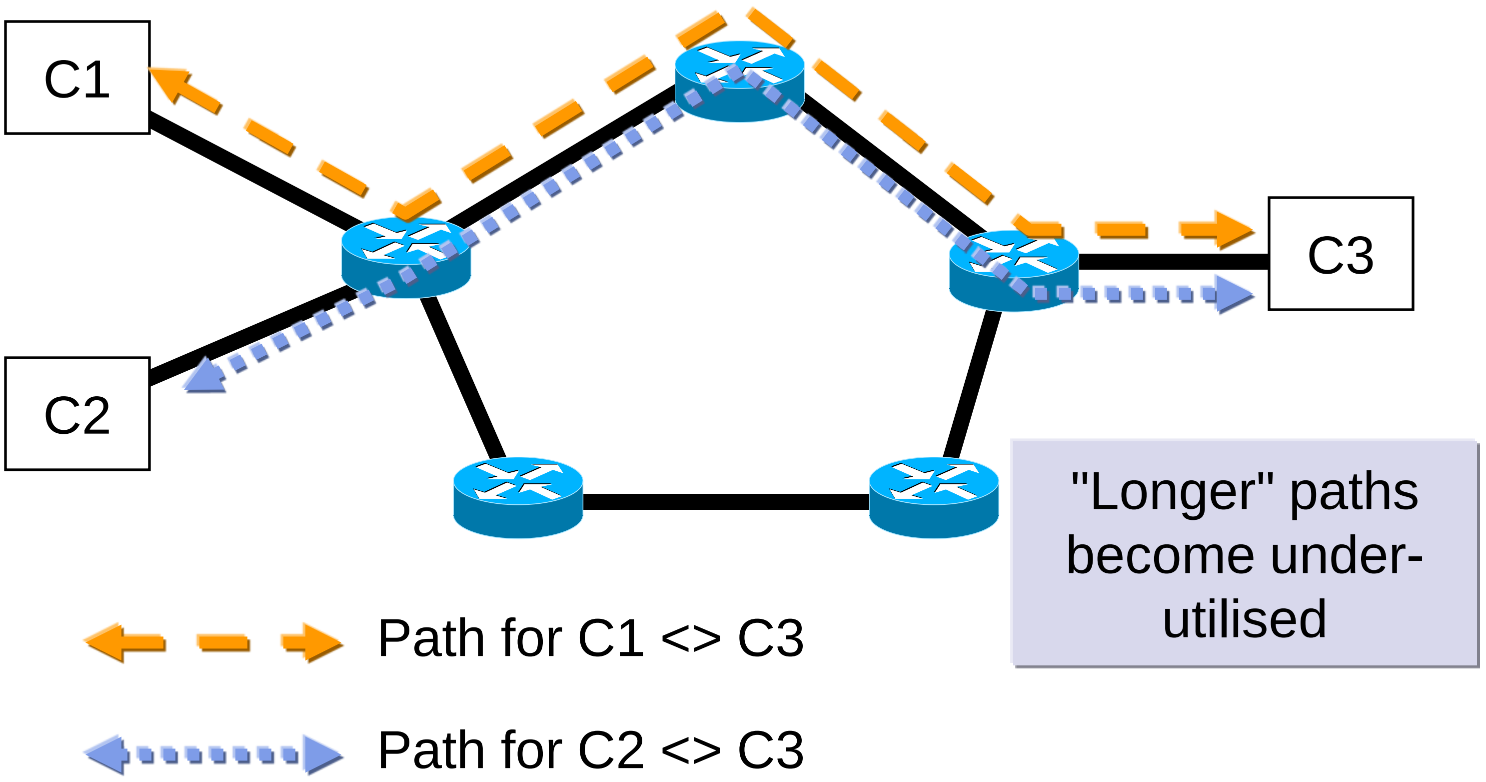
- **Early Multi-Layer Switching Initiatives**
  - **IP Switching (Ipsilon/Nokia)**
  - **Tag Switching (Cisco)**
  - **IP Navigator (Cascade/Ascend/Lucent)**
  - **ARIS (IBM)**
- **IETF Working Group chartered in spring 1997**
- **IETF Solution should address the following problems:**
  - **Enhance performance and scalability of IP routing**
  - **Facilitate explicit routing and traffic engineering**
  - **Separate control (routing) from the forwarding mechanism so each can be modified independently**
  - **Develop a single forwarding algorithm to support a wide range of routing functionality**

Drawbacks of Conventional Routing

- Performance
  - *In the past*, routing was perceived as processor-limited
  - Each forwarding decision might require ~1000 machine instructions
  - Longest prefix match was difficult to transfer to silicon
  - *Today*, it is possible to build wire-speed routing in silicon
- Connectionless IP does not support Traffic Engineering
  - The "hyper-aggregation problem"
- Difficulty of implementing QoS architectures and services (survivability, VPNs, …)

The Hyper-aggregation Problem (Fish Problem)

- Routing Protocols Create A Single "Shortest Path"



Some Terminology...

- **Network Engineering**
  - "Put the *bandwidth* where the *traffic* is"
    - Physical cable deployment
    - Virtual connection provisioning
- **Traffic Engineering**
  - "Put the *traffic* where the *bandwidth* is"
    - On-line or off-line optimisation of routes
    - Implies the ability to diversify routes

Steps in the process

- Topology determination
- Path selection/creation
- Data forwarding

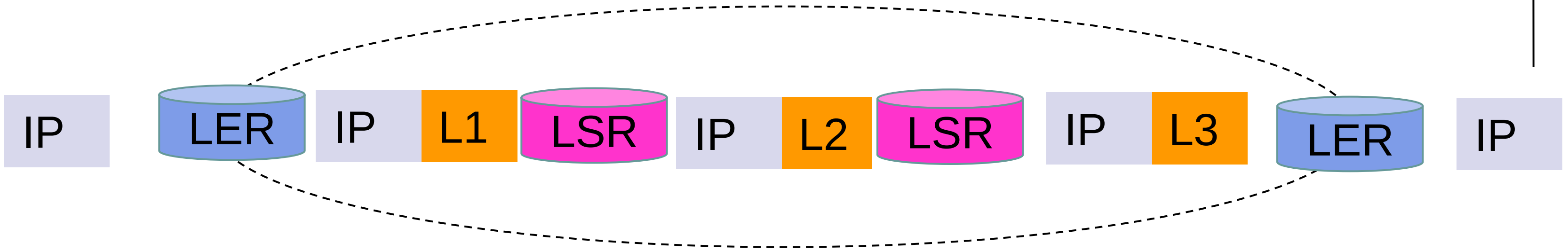
Steps in the process

- Topology determination
- Path selection/creation
- Data forwarding

Topology Determination

- Build on existing link-state routing protocols: OSPF, IS-IS
- Add traffic engineering (TE) extensions: OSPF-TE & IS-IS-TE to communicate **constraints**.
  - Two important ones:
    - Available bandwidth information, broken down by priority to allow tunnels to preempt others
    - Attribute flags
    - Example: Assuming 8-bit and a link that has attribute flags of 0x1 (0000 0001) means that the link is a satellite link.

What is MPLS?



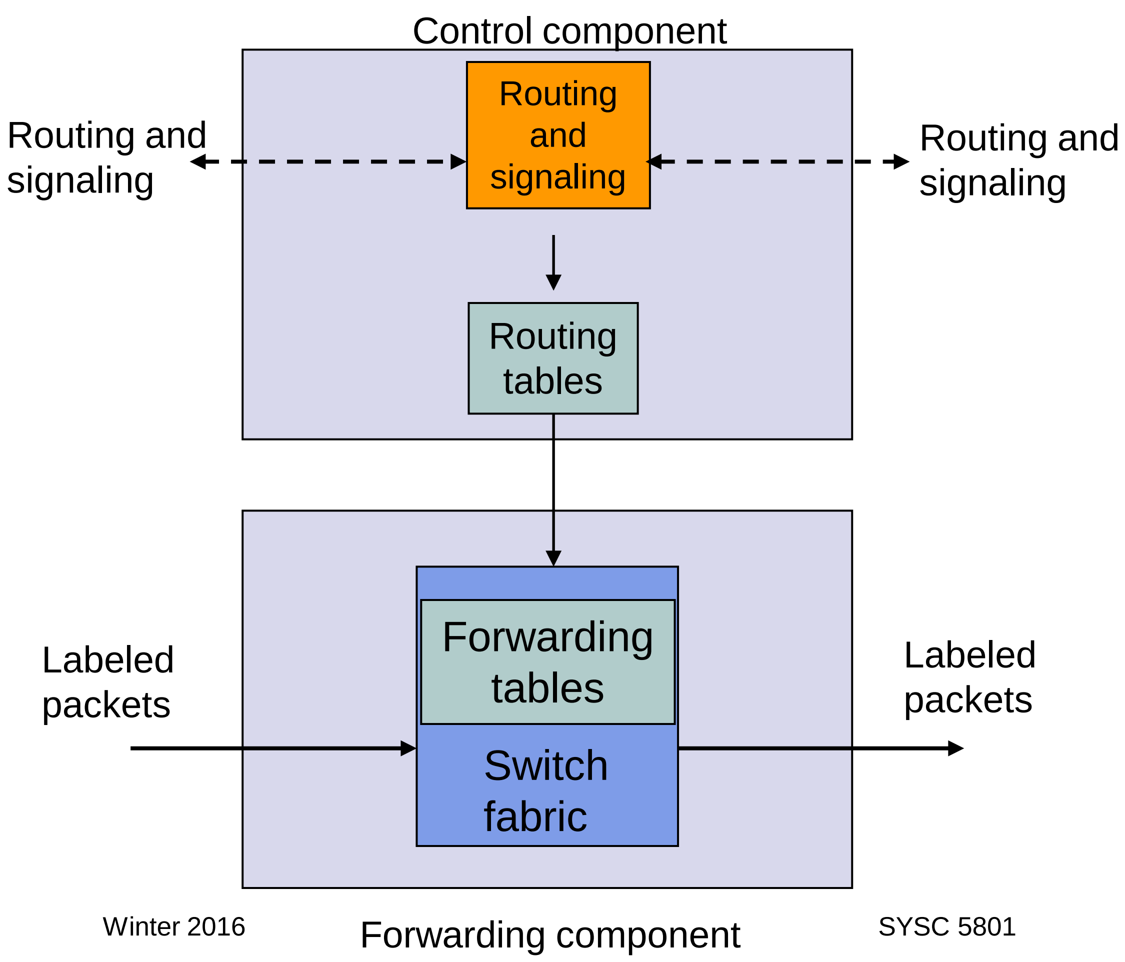
- *Multiprotocol Label Switching (MPLS)*
- A set of protocols that enable MPLS networks
  - Packets are assigned *labels* by edge routers (which perform longest-prefix match)
  - Packets are forwarded along a *Label-Switched Path (LSP)* in the MPLS network using label switching
  - LSPs can be created over *multiple layer-2 links*
    - ATM, Ethernet, PPP, frame relay
  - LSPs can support *multiple layer-3 protocols*
    - IPv4, IPv6, and in others

Why MPLS?

- Labels enable fast forwarding
  - But IP lookup is also fast for advanced core routers
  - Longest-prefix matching is expensive
- *Circuits (virtual circuits or paths) are good (sometimes)*
  - *Conventional IP routing selects a shortest path/paths, does not provide choice* of route
  - Label switching enables routing **flexibility**
  - ***Traffic engineering***: establish separate paths to meet different performance requirements or dynamic traffic demands
  - ***Fast Reroute*** in case of failures
  - ***Virtual Private Networks***: establish tunnels between user nodes
  - Other services

Separation of Forwardng & Control

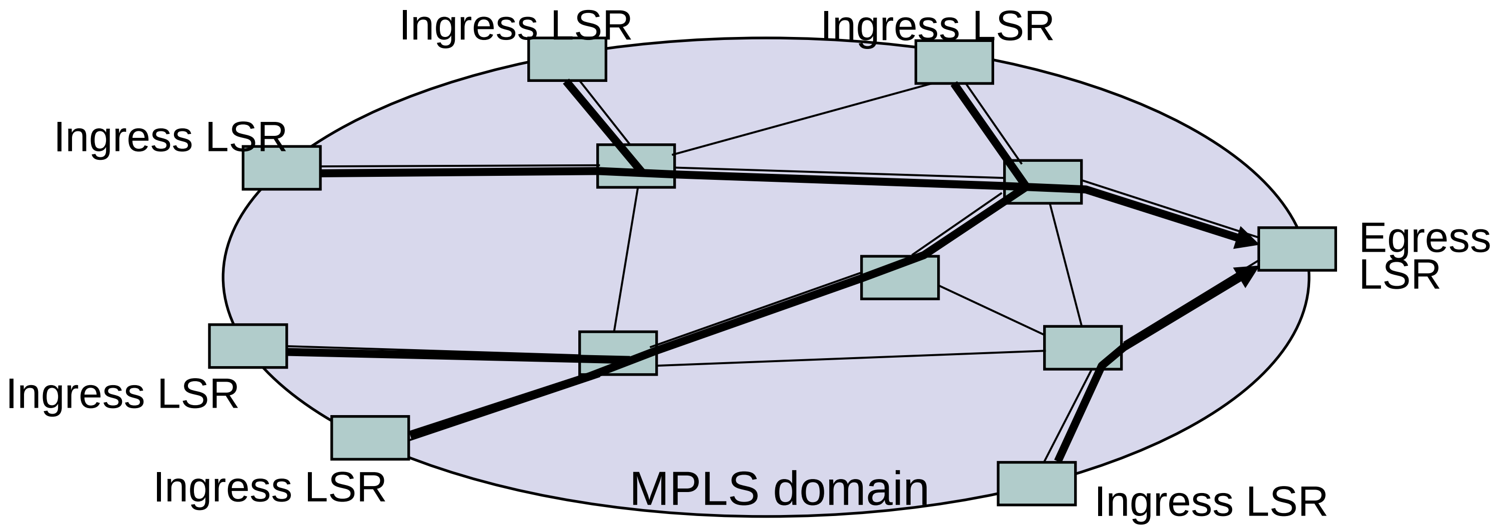
All proposals leading to MPLS separate forwarding and control



With MPLS: forwarding & control are separate

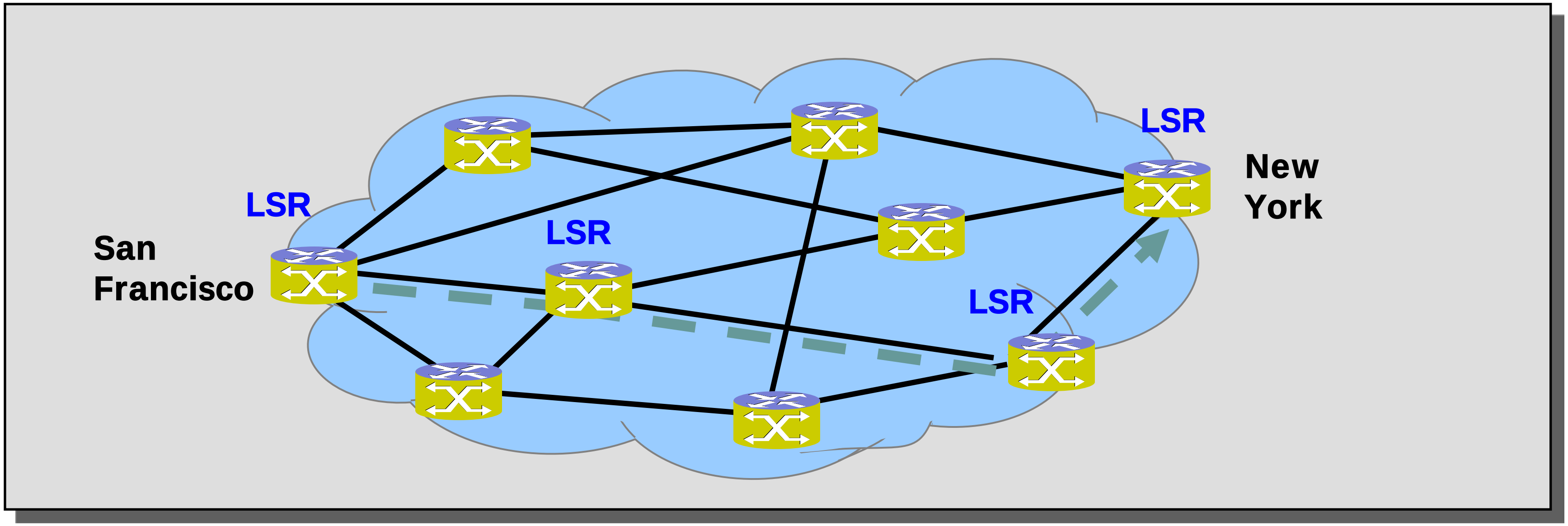
- Different control schemes dictate creation of labels & label-switched paths
- All forwarding done with label switching
- Control & forwarding can evolve independently

Labels and Paths



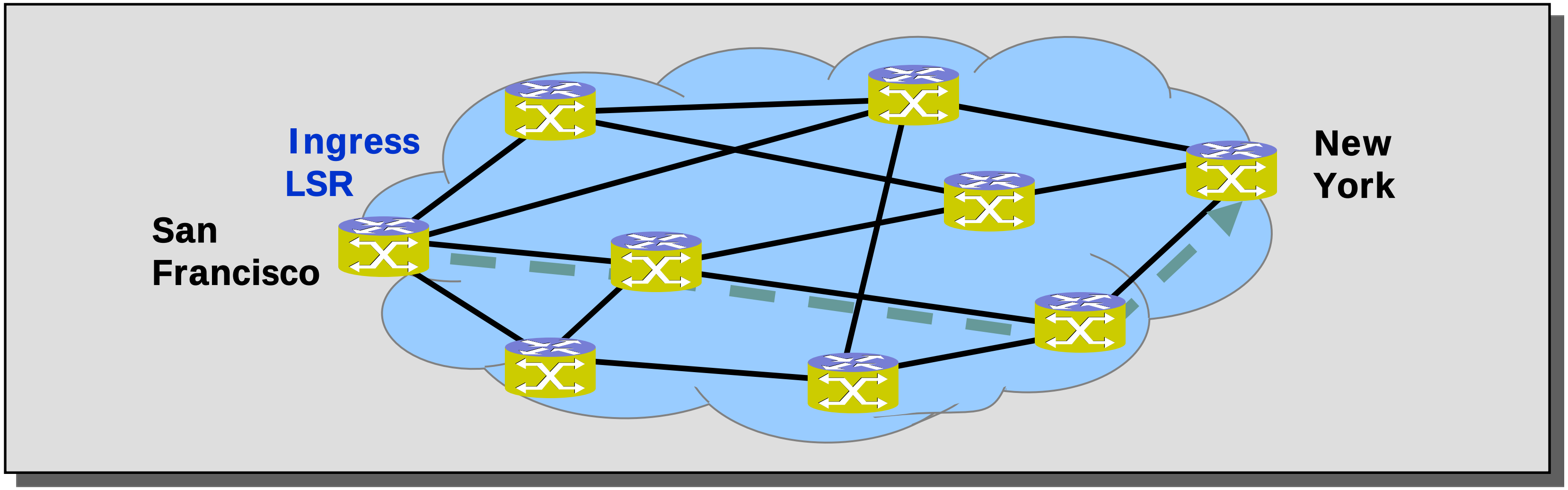
- Label-switched paths (LSPs) are *unidirectional*
- LSPs can be:
  - point-to-point
  - *tree rooted in egress node* corresponds to shortest paths leading to a destination egress router
    - Ingress: head end router of an LSP
    - Egress: tail end

Label Switching Router (LSR)



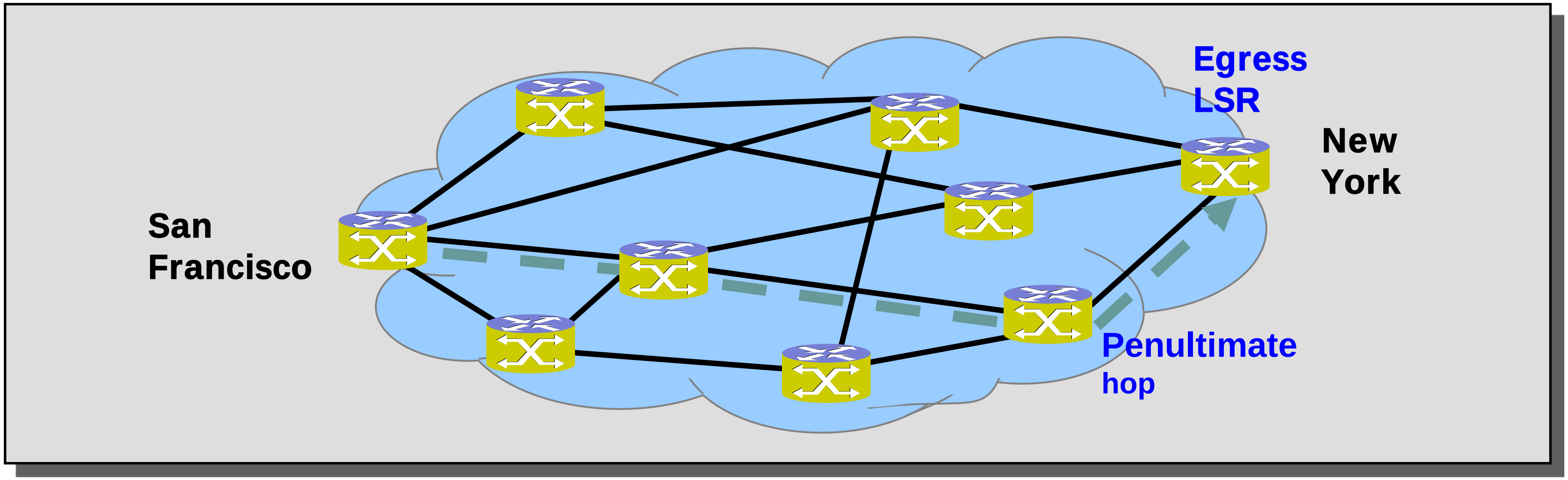
- **Label-Switching Router (LSR)**
  - ✓ **Forwards MPLS packets using label-switching**
  - ✓ **Capable of forwarding native IP packets**
  - ✓ **Executes one or more IP routing protocols**
  - ✓ **Participates in MPLS control protocols**

Ingress Router
Label Edge Router (LER)



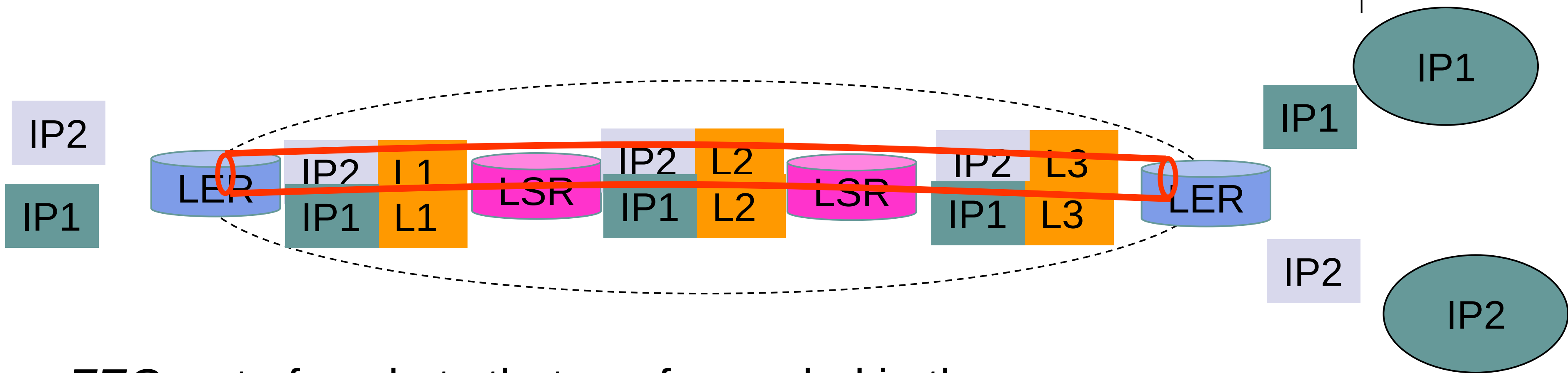
- **Ingress LSR**
  - **Examines inbound IP packets**
  - **Classifies packet to an FEC**
  - **Generates MPLS header and assigns (binds) initial label**
  - **Upstream from all other LSRs in the LSP**
  - **All other routers inside the MPLS domain look at the labels only, not at the IP address**

Egress Router
Label Edge Router (LER)



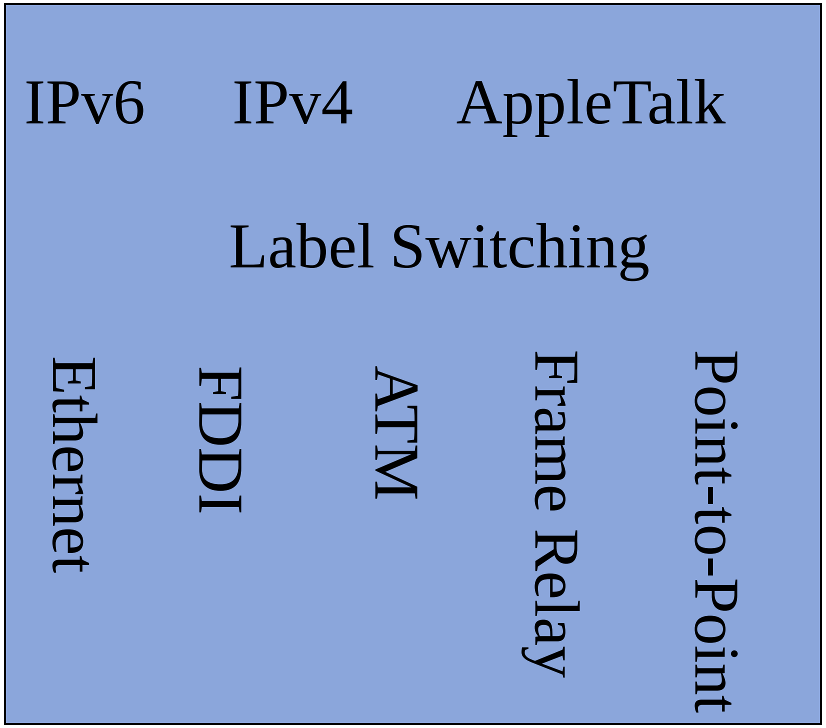
- **Egress LSR**
  - ✓ **Processes traffic as it leaves the MPLS domain – based on IP packet destination address**
  - ✓ **Removes the MPLS header – unless the "Penultimate hop" router already had removed it.**
  - ✓ **Downstream from all other LSRs in the LSP**

Forwarding Equivalence Class



- *FEC:* set of packets that are forwarded in the same manner
  - Over the same path, with the same forwarding treatment
  - Packets in an FEC have same next-hop router
  - Packets in same FEC may have different network layer header
  - Each FEC requires ***a single entry*** in the forwarding table
  - Coarse Granularity FEC: packets for all networks whose destination address matches a given address prefix
  - Fine Granularity FEC: packets that belong to a particular application running between a pair of computers

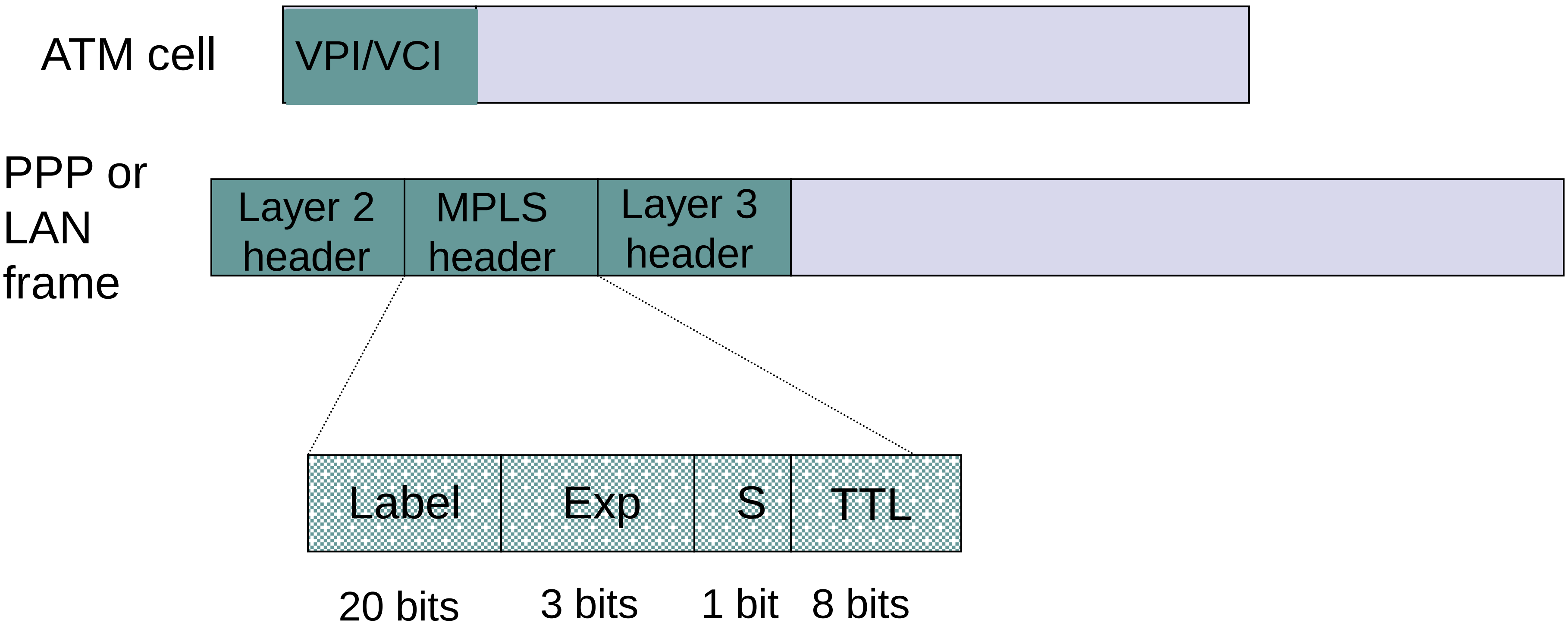
Multiprotocol: Both Above and Below



Network Layer Protocols

Link Layer Protocols

MPLS Labels



- Labels can be encoded into VPI/VCI field of ATM header
- *Shim header* between layer 2 & layer 3 header (32 bits)
  - 20-bit label + 1-bit hierarchical stack field + 8-bit TTL
  - 3-bit “experimental” field (can be used to specify 8 QoS level)

A Label by Any Other Name ….

- There are many examples of label substitution protocols already in existence:
  - **ATM:** label is called VPI/VCI and travels with cell
  - **Frame Relay:** label is called a DLCI and travels with frame
  - **Frequency substitution:** where label is a light frequency via DWDM, OXC etc.

What is a “LABEL”?

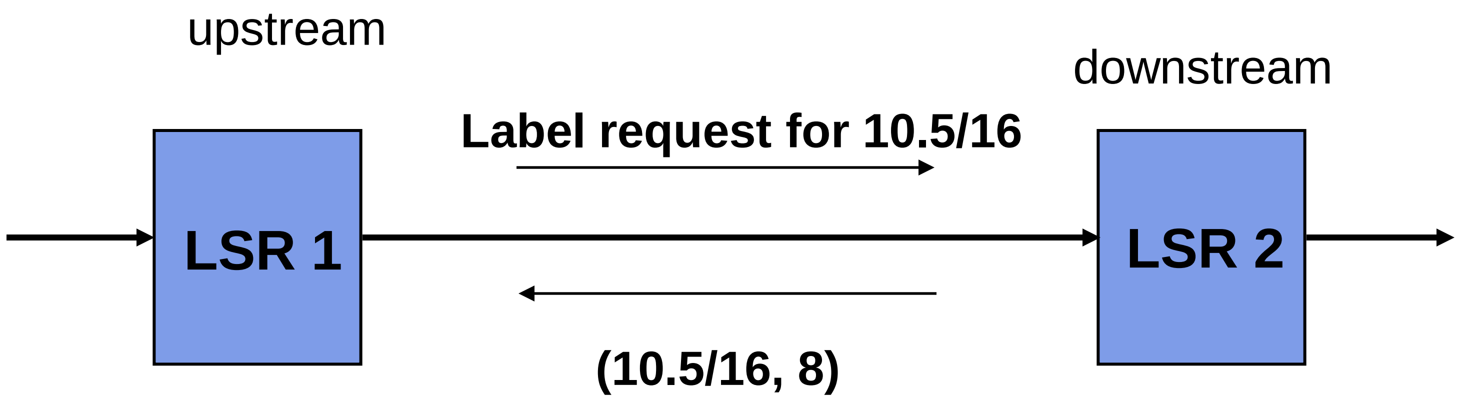
A property that uniquely identifies a flow on a logical or physical interface

- Label value mostly changes at each hop
  - Labels are local significant
- Labels can be
  - Interface-specific
    - Label 3 on interface A means something different from label 3 on interface B
  - platform-wide
    - Label 3 is label 3, no matter what interface it is received on

Label Distribution and RSVP-TE

Label Distribution

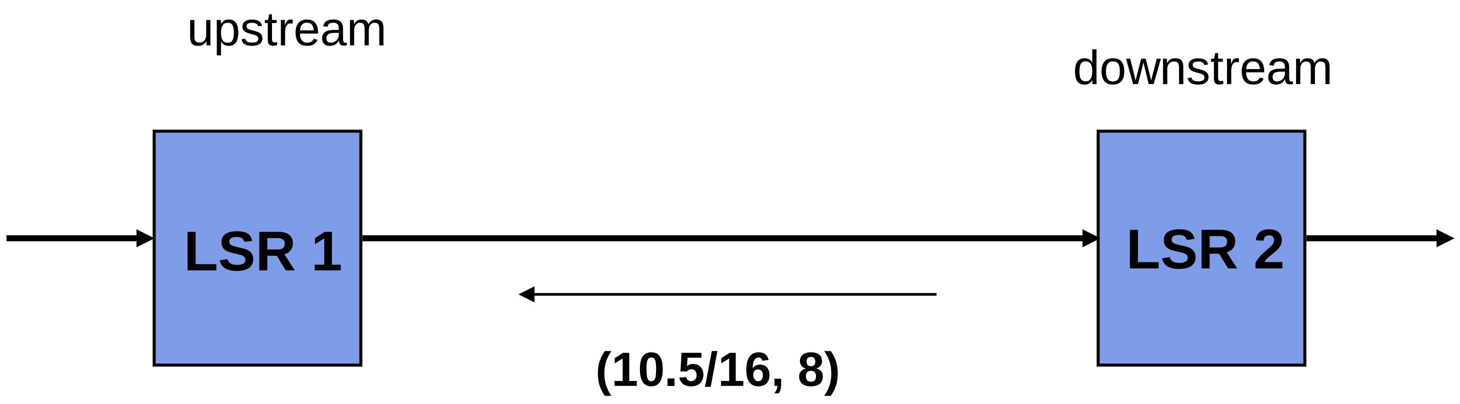
- Label Distribution Protocols distribute label bindings between LSRs



Downstream-on-Demand Mode

- LSR1 becomes aware LSR2 is next-hop in an FEC
- LSR1 requests a label from LSR2 for given FEC
- LSR2 checks that it has next-hop for FEC, responds with label

Label Distribution

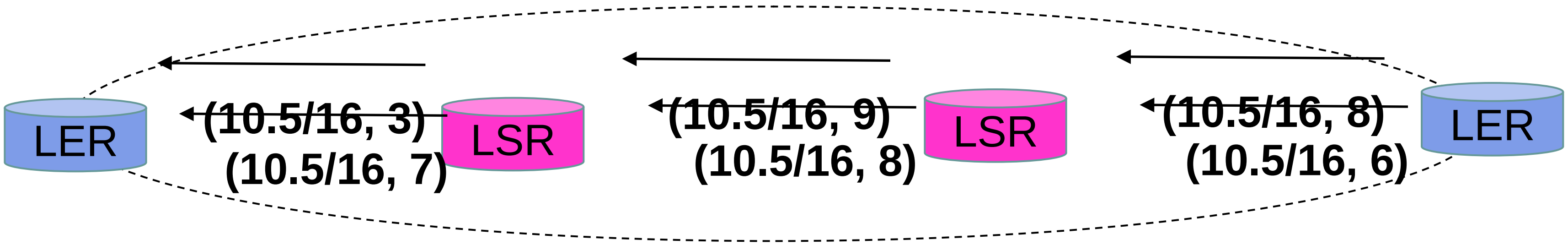


Downstream Unsolicited Mode

- LSR2 becomes aware of a next hop for an FEC
- LSR2 creates a label for the FEC and forwards it to LSR1
- LSR1 can use this label if it finds that LSR2 is next-hop for that FEC

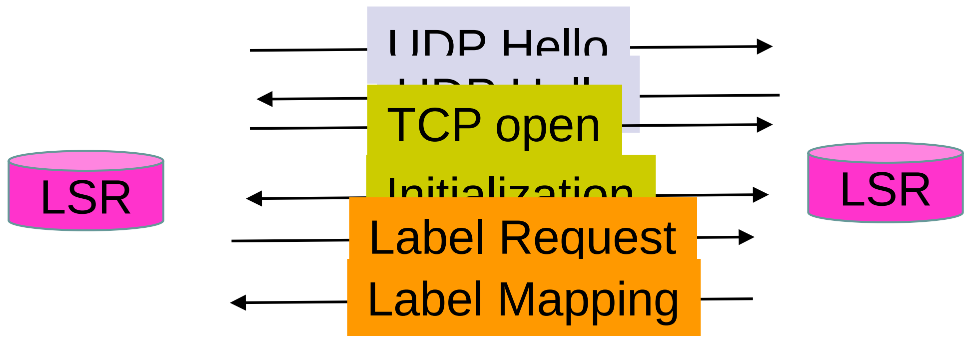
Independent vs. Order Label Distribution Control

- *Ordered Label Distribution Control*: LSR can distribute label if
  - It is an egress LSR
  - It has received FEC-label binding for that FEC from its next hop



- *Independent Label Distribution Control*: LSR independently binds FEC to label and distributes to its peers

LDP - Label Distribution Protocol



- *Label Distribution Protocol (LDP)*, RFC 3036
  - Topology-driven assignment (routes specified by routing protocol)
  - Hello messages over UDP
  - TCP connection & negotiation (session parameters & label distribution option, label ranges, valid timers)
  - Message exchange (label request/mapping/withdraw)

ReSerVation Protocol (RSVP)

- RSVP is an IP signaling protocol to setup and maintain flow-specific state in hosts and routers
- Simplex
  - Requests resources from sender to receiver
  - Sender sends PATH message that describes traffic flow
  - Bidirectional flows require separate reservations
- Receiver-oriented
  - Receivers initiate and maintain resource reservations
  - Receiver sends RESV message to reserve resources
- Soft-state at intermediate routers
  - Reservation valid for specified duration
  - Released after timeout, unless first refreshed

Steps in the process

- Topology determination

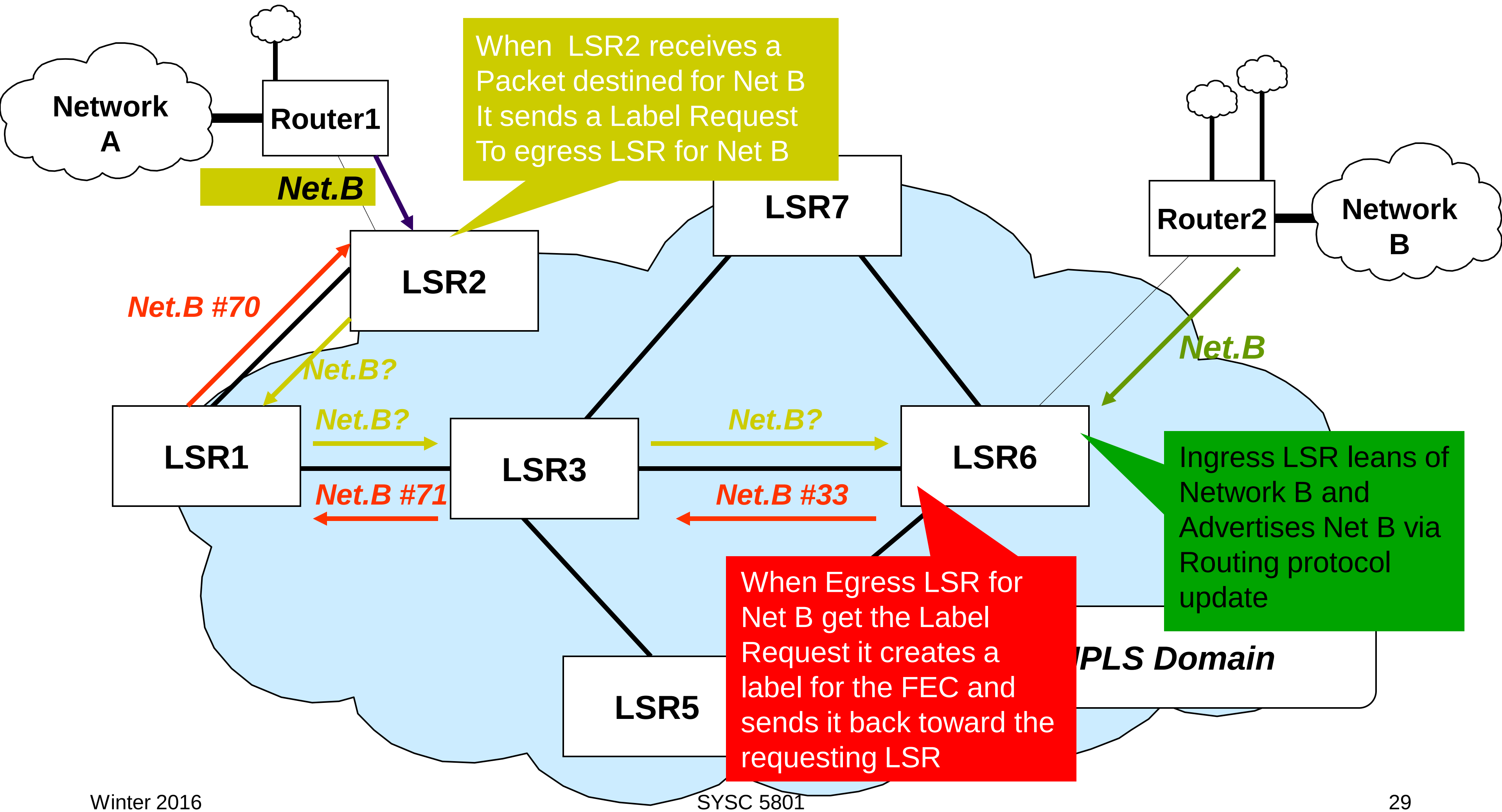
- Path selection/creation

- Data forwarding

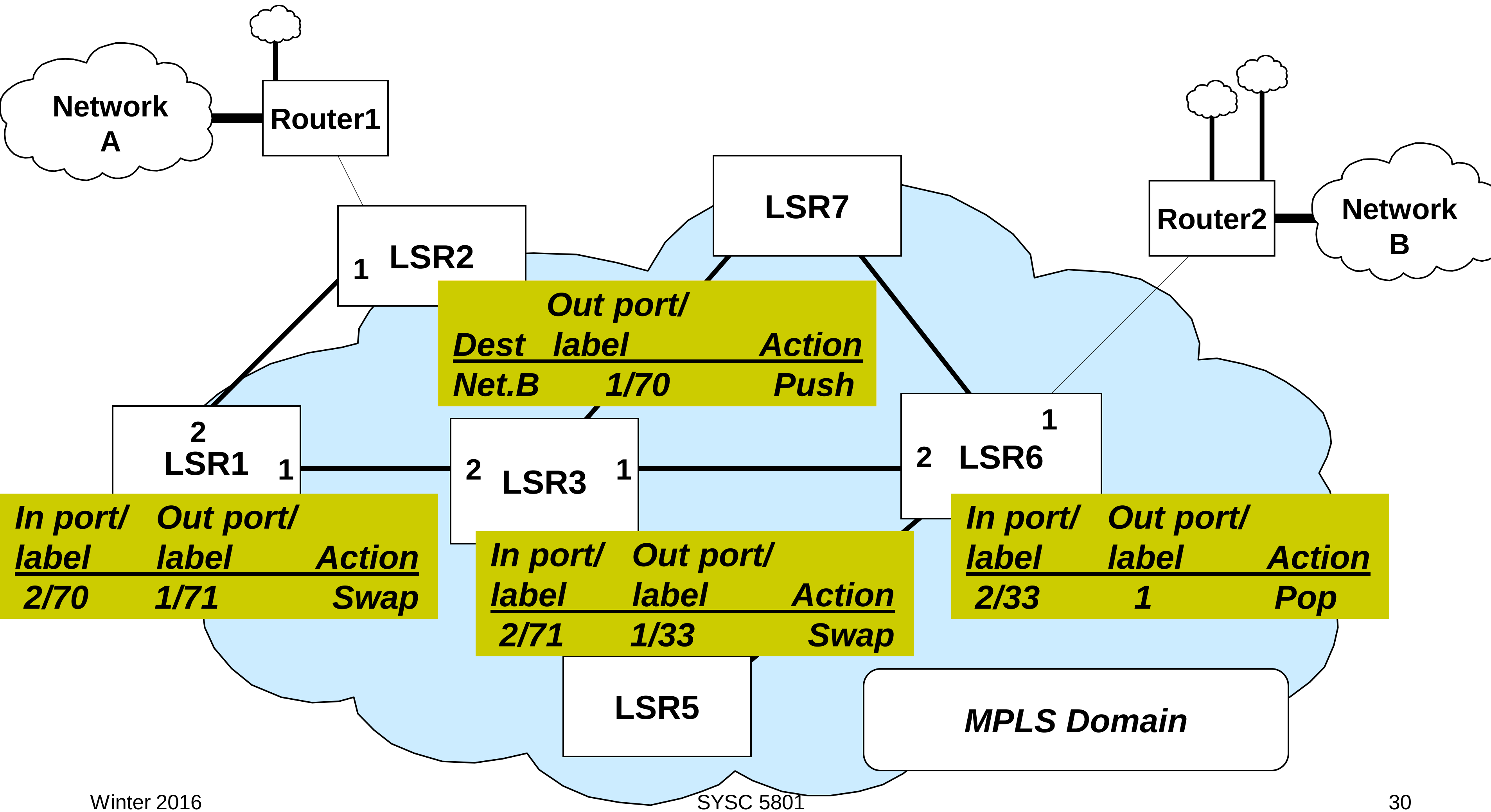
New Protocols for Path Creation and Selection

- Need extensions to existing protocols and algorithms to consider TE requirements:
  - Existing routing protocols: need to carry more link info, e.g., bandwidth, attributes
    - OSPF → OSPF-TE
    - ISIS → ISIS-TE
  - Shortest path: need to consider constraints, e.g., bandwidth, delay, ...
    - SPF → CSPF (Constraint-based SPF)
  - Label distribution protocols: need to carry more info, e.g., bandwidth, attributes
    - LDP → CR-LDP
    - RSVP → RSVP-TE

Label Distribution: Downstream On-Demand Data Driven



Label Switched Path – *created*



Dest	Out port/ label	Action
Net.B	1/70	Push

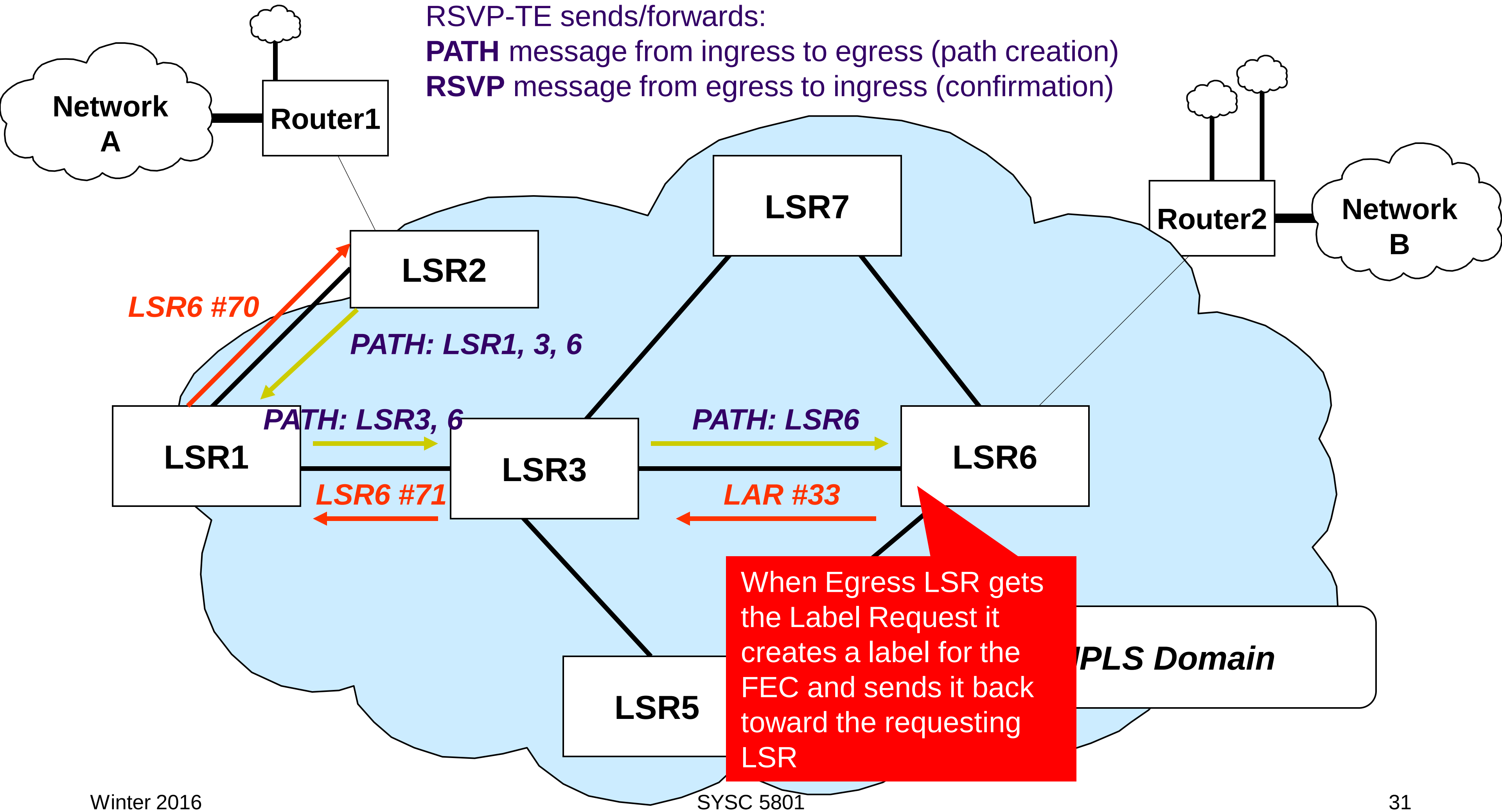
In port/ label	Out port/ label	Action
2/70	1/71	Swap

In port/ label	Out port/ label	Action
2/71	1/33	Swap

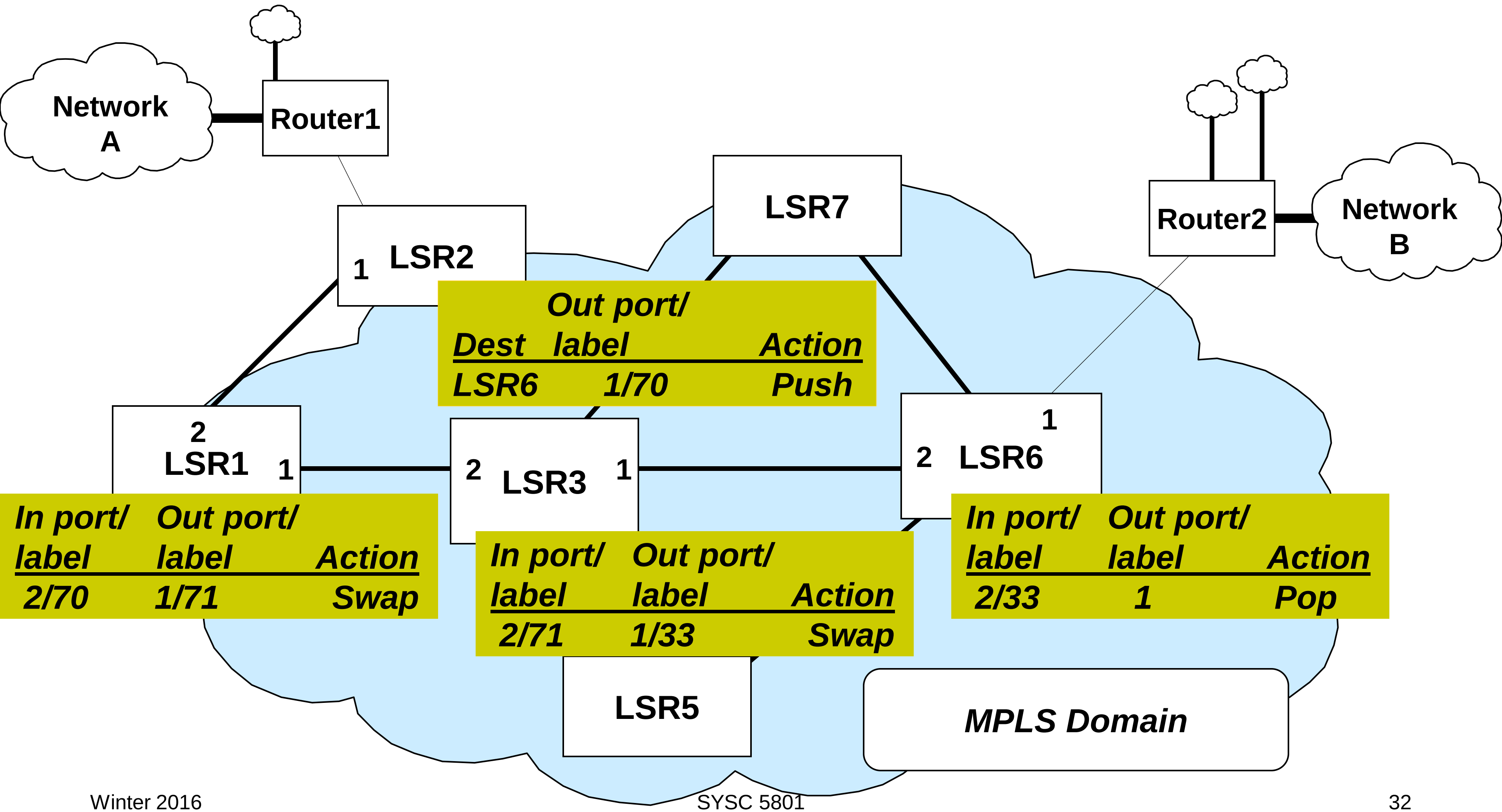
In port/ label	Out port/ label	Action
2/33	1	Pop

Label Distribution: Downstream On-Demand Explicit Route

RSVP-TE sends/forwards:
PATH message from ingress to egress (path creation)
RSVP message from egress to ingress (confirmation)



Label Switched Path – *created*



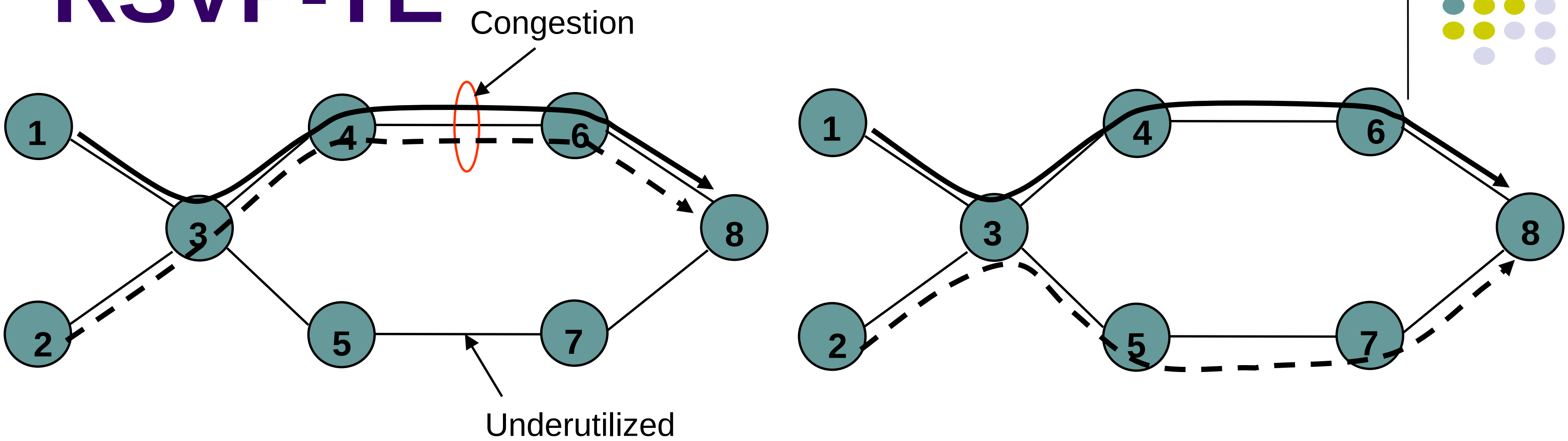
RSVP Soft State

- Reservations are valid for a timeout period
- Need to “refresh” reservation state by resending PATH & RESV messages before expiry time
- Reservation removed if not refreshed by timeout
- RSVP runs directly over IP with type=46
  - message delivery is not reliable
  - Assume 1 in 3 consecutive messages gets through
- Nominal refresh rate specified by *R* (usually 30 sec)
- Refresh period for a receiver randomized from (0.5*R*, 1.5*R*) to avoid simultaneous refresh attempts
- PathTear & ResvTear messages explicitly delete reservations

RSVP Message Objects

SESSION: IP destination address, IP protocol number, and destination port #
RSVP_HOP: IP address of RSVP-capable router that sent this message
TIME_VALUES: refresh period R.
STYLE: reservation style information not in flowspec or filterspec objects
FLOWSPEC: desired QoS in a Resv message.
FILTER-SPEC: set of packets that receive desired QoS in a Resv message.
SENDER_TEMPLATE: IP address of the sender in Path message.
SENDER_TSPEC: sender's traffic characteristics in Path message.
ADSPEC: carries end-to-end path information in Path message.
ERROR_SPEC: specifies errors in PathErr and ResvErr; confirmation in ResvConf.
POLICY_DATA: enables policy modules to determine whether request is allowed
INTEGRITY: cryptographic and authentication information to verify RSVP message
SCOPE: explicit list of senders that are to receive this message.
RESV_CONFIRM: receiver IP address that is to receive the confirmation.

RSVP-TE

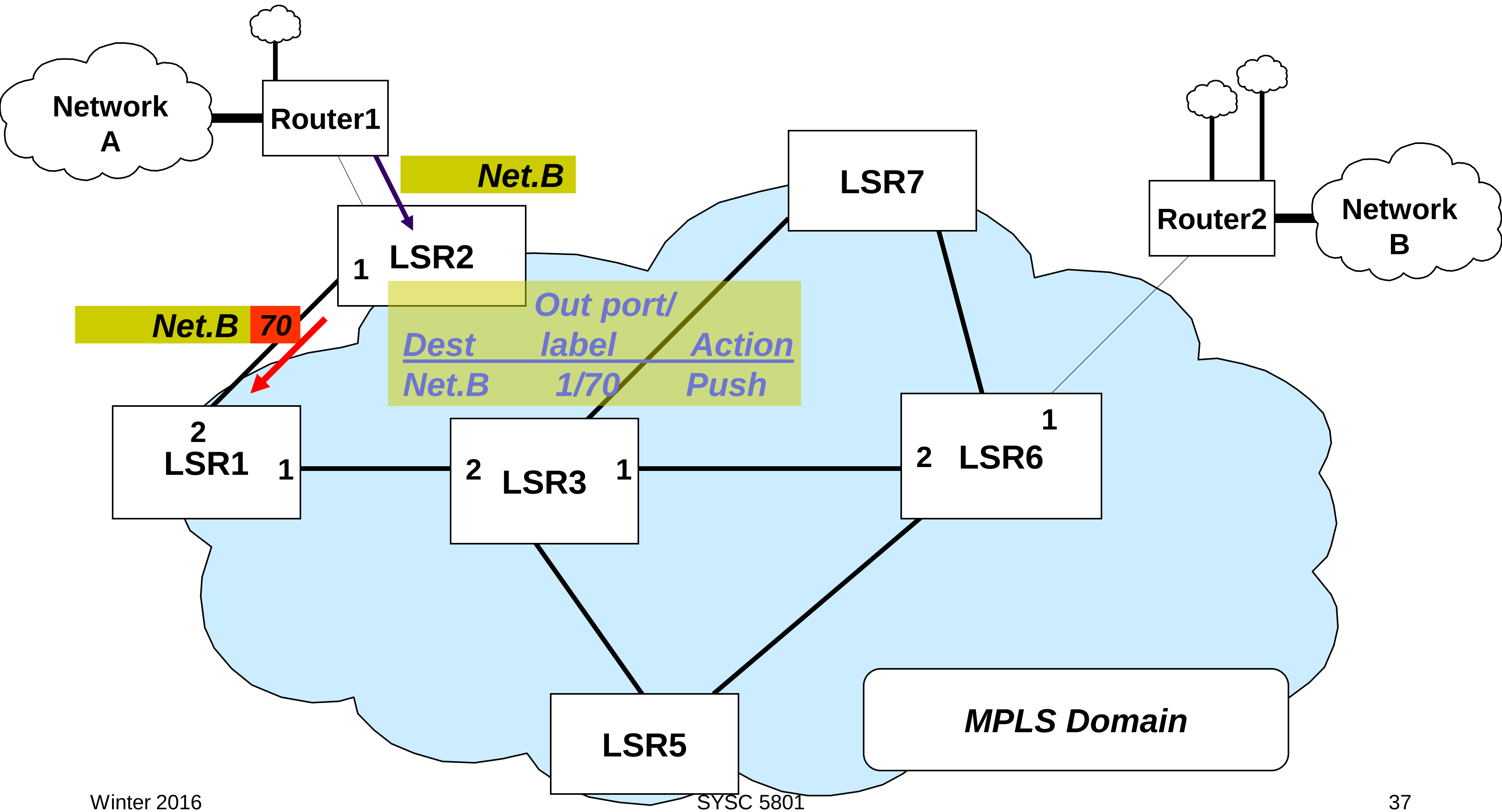


- Extensions to RSVP for *traffic-engineered LSPs*
  - Request-driven label distribution to create explicit route LSPs
  - Single node (usually ingress) determines route
  - Enables traffic engineering

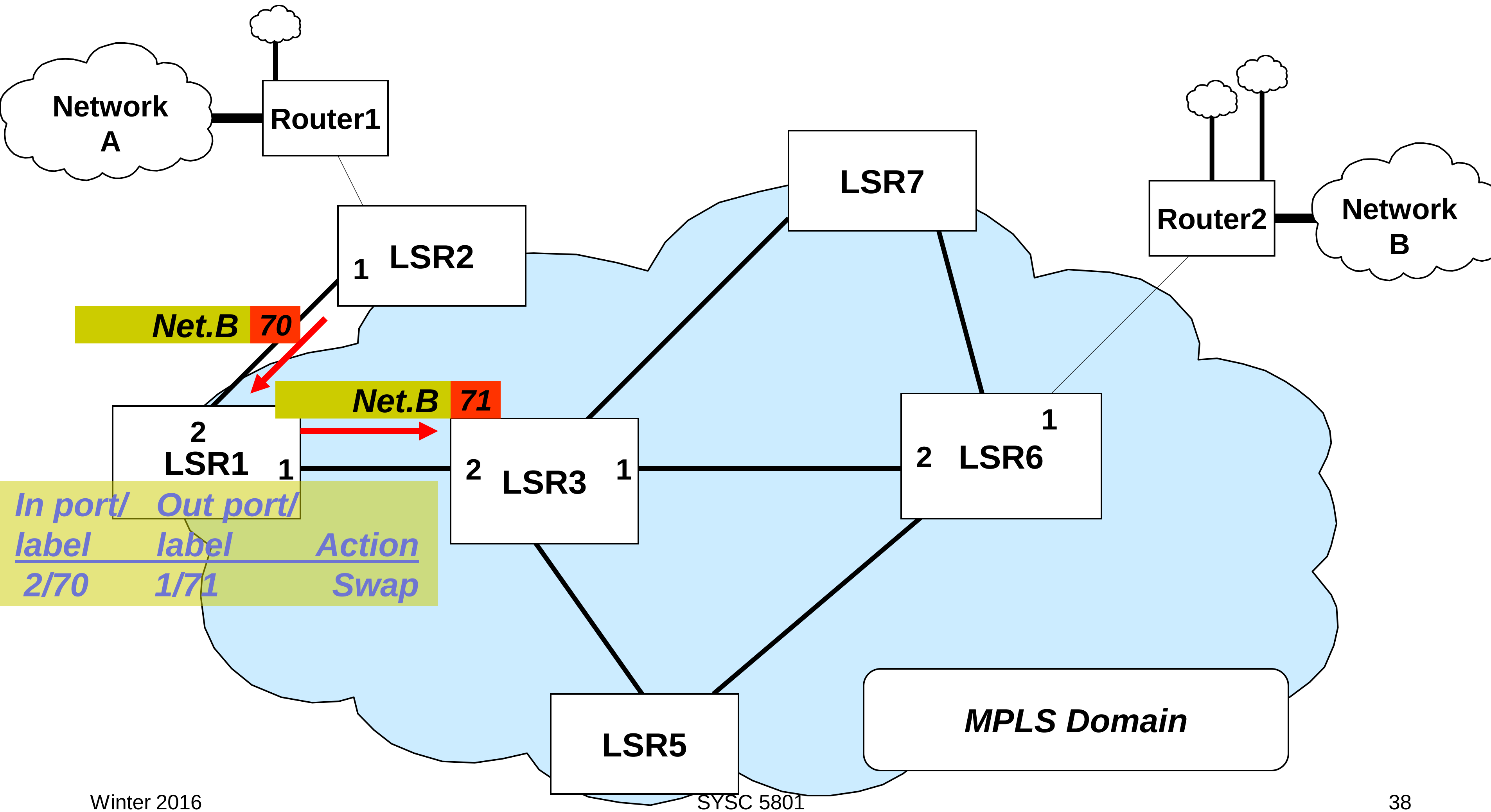
Steps in the process

- Topology determination
- Best path determination
- Data forwarding

Data Forwarding – *Unlabelled packet to Ingress*

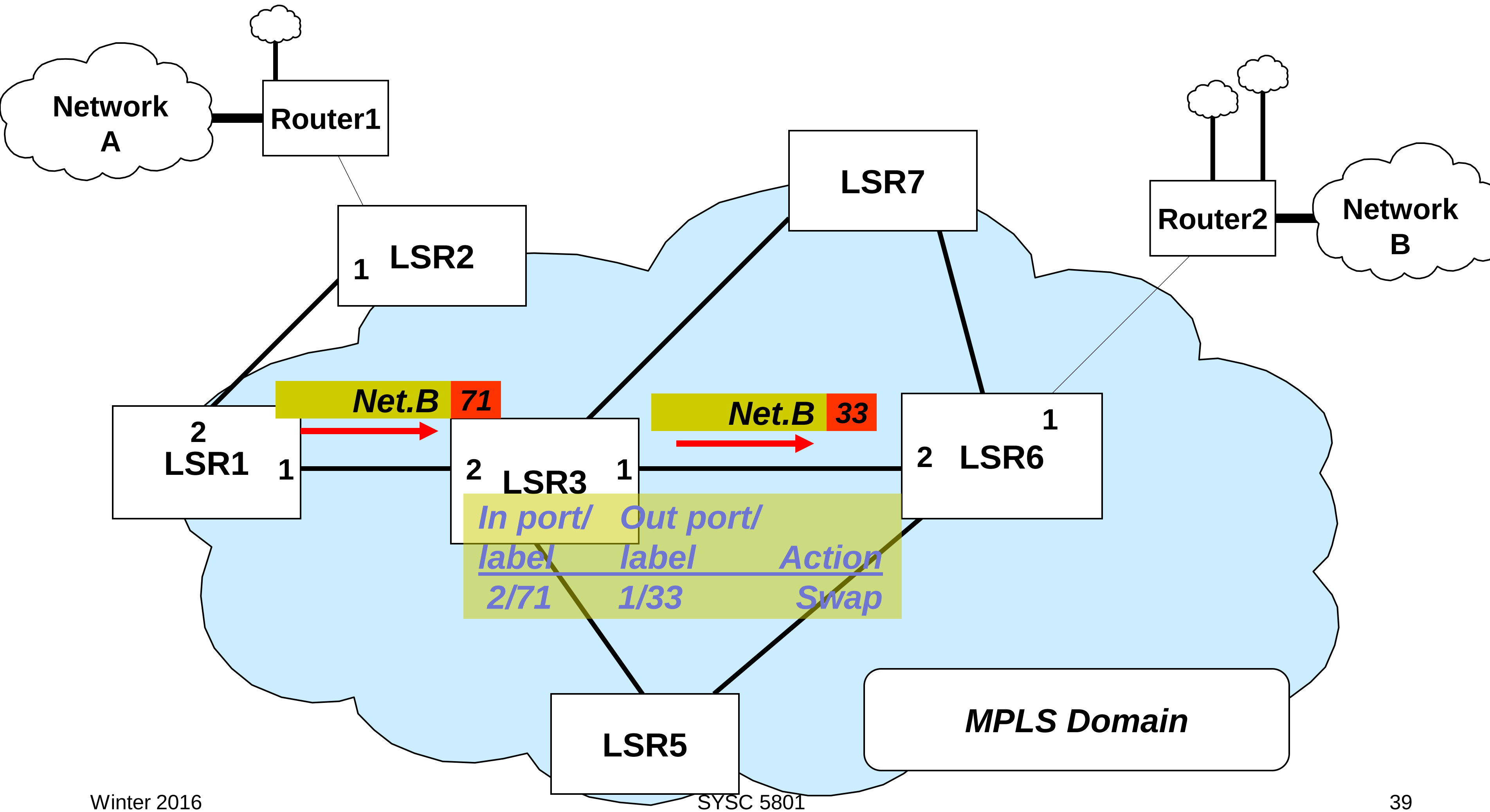


Data Forwarding – *LSR1 – LSR3*

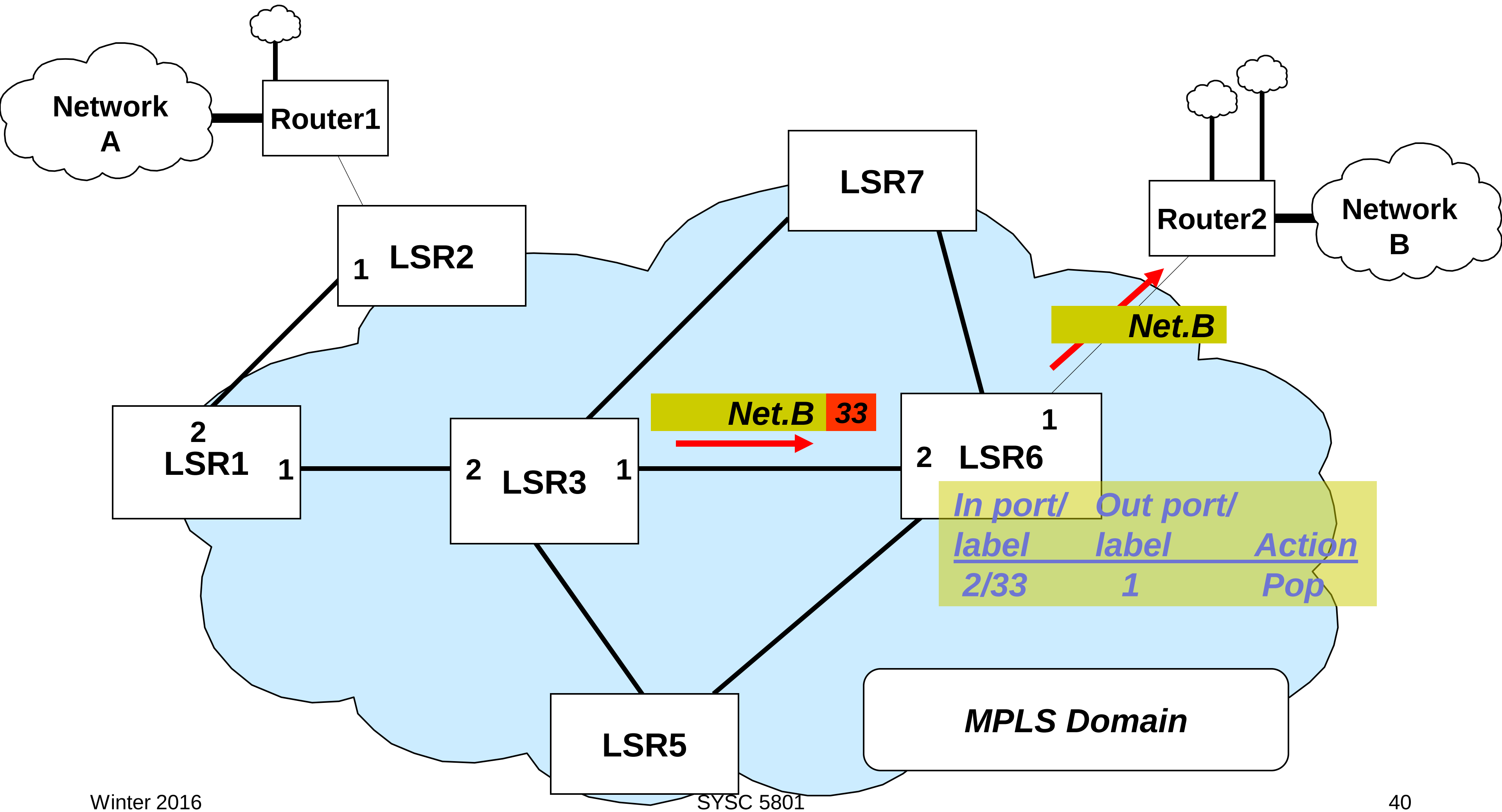


In port/ label	Out port/ label	Action
2/70	1/71	Swap

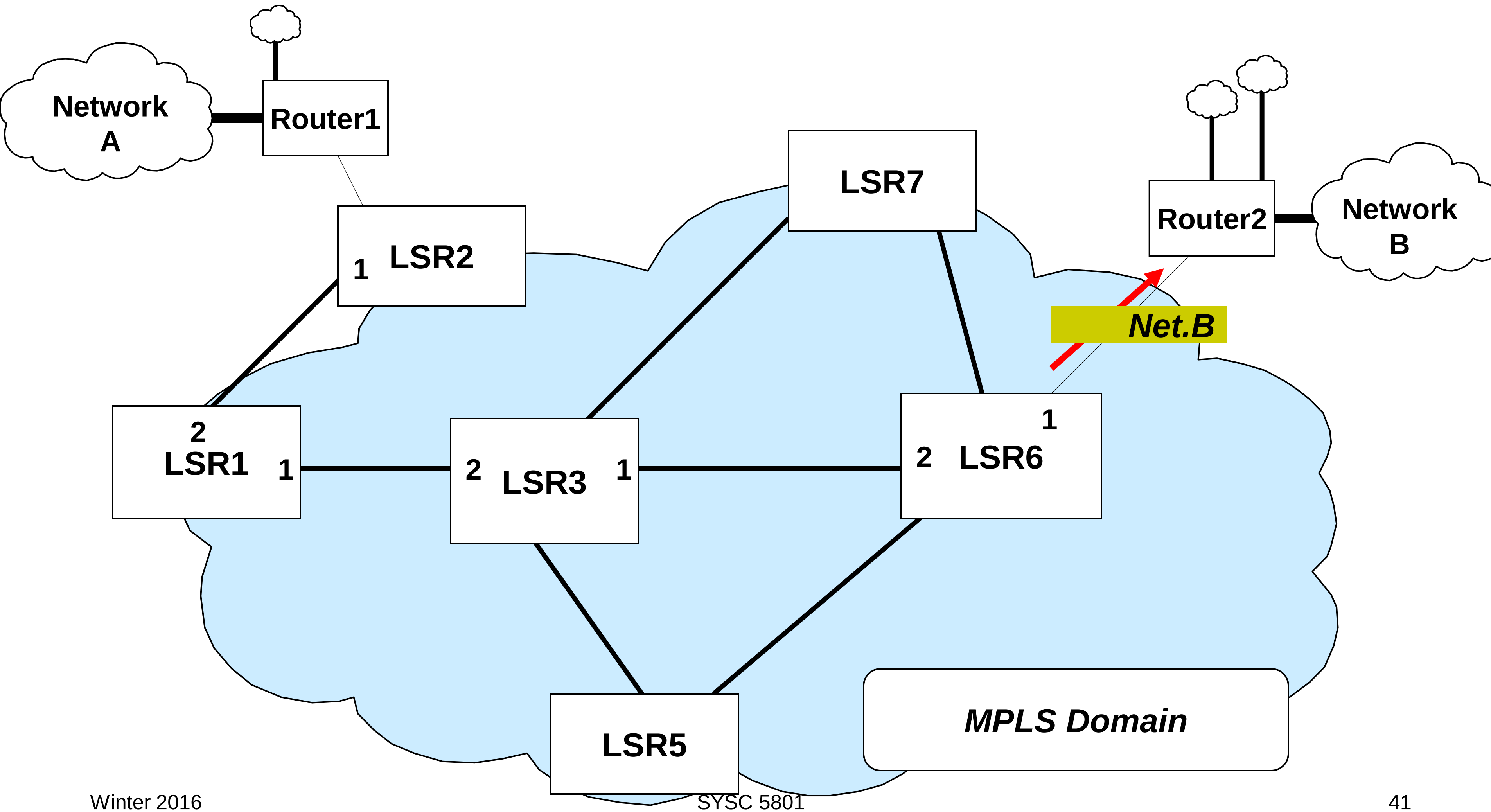
Data Forwarding – *LSR3 – LSR6*



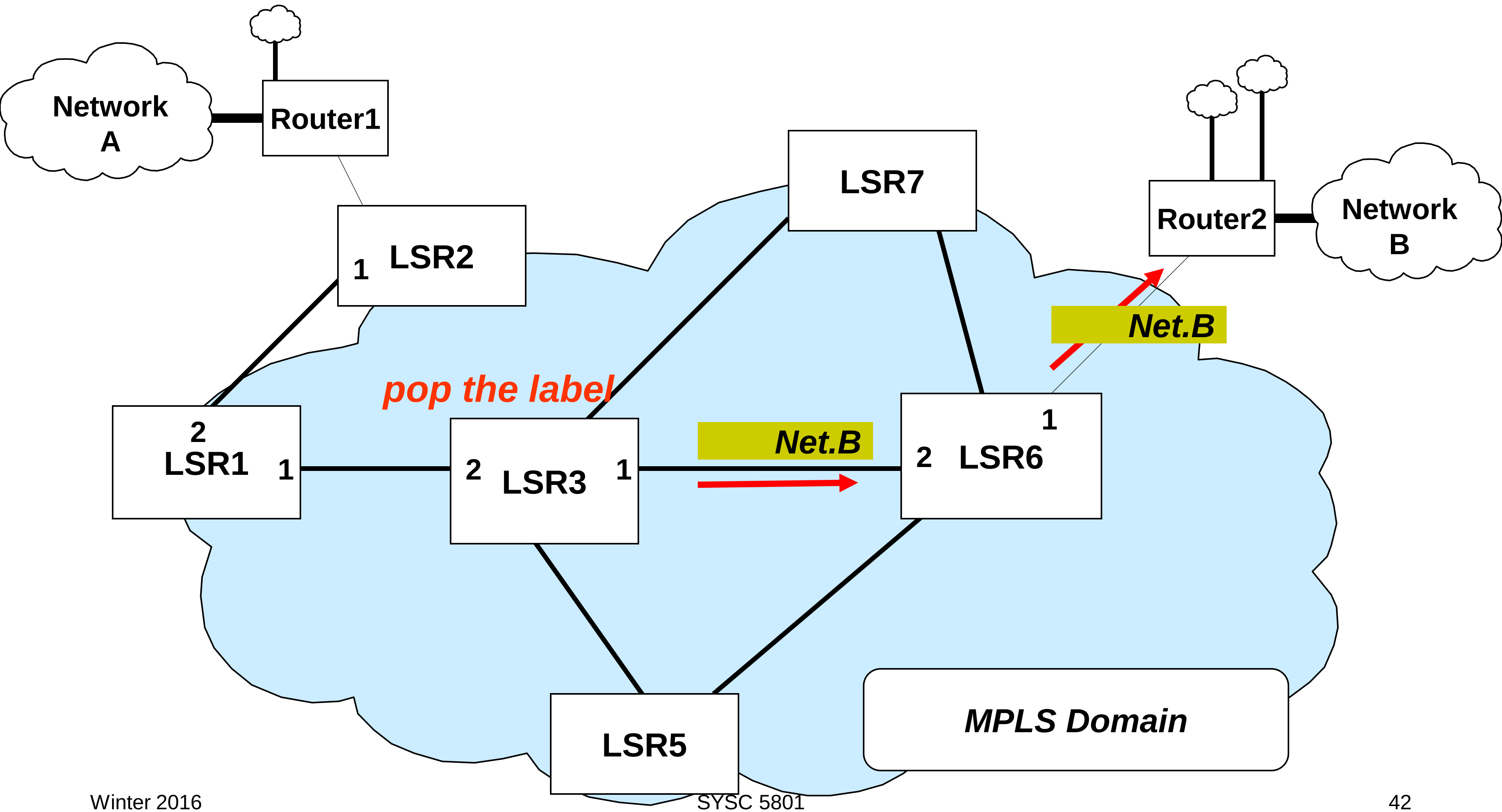
Data Forwarding – *LSR6 – Egress Router*



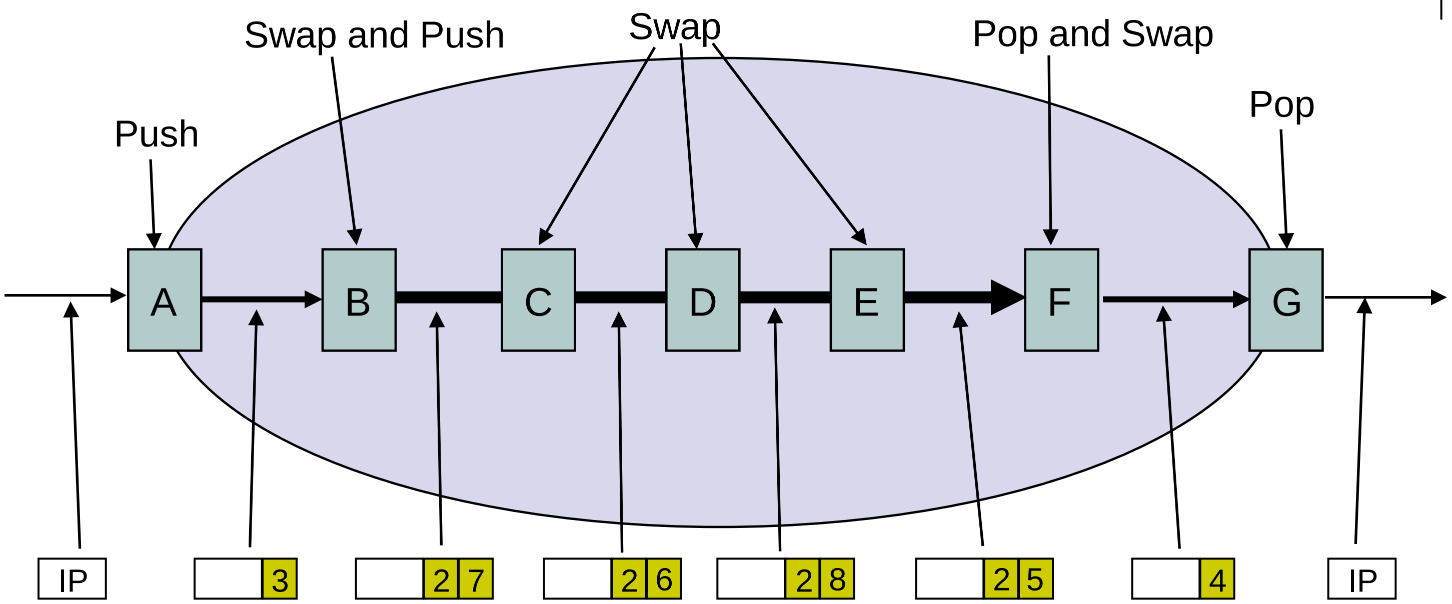
Data Forwarding – *Unlabelled packet delivered*



Data Forwarding – *Penultimate hop popping*



Label Stacking



- MPLS allows multiple labels to be stacked
  - Ingress LSR performs *label push* (S=1 in label, last level)
  - Egress LSR performs *label pop*
  - Intermediate LSRs can perform additional pushes & pops (S=0 in label) to create tunnels
  - Above figure has tunnel between A & G; tunnel between B&F
  - All flows in a tunnel share the same outer MPLS label

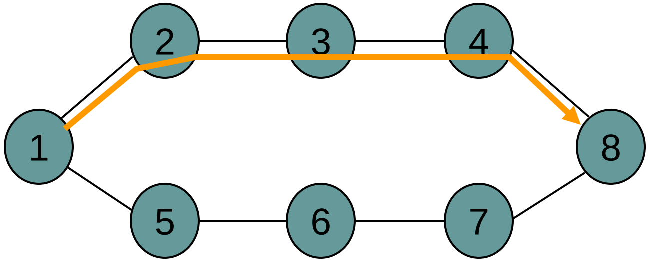
MPLS Application – Example Survivability

Protection and Restoration

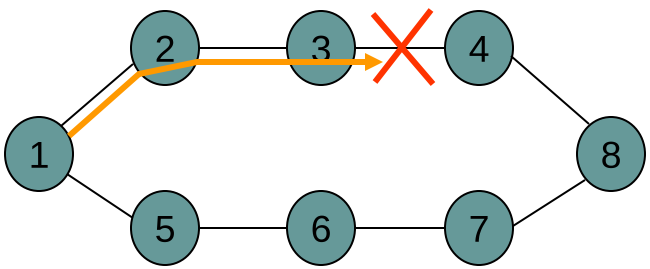
MPLS Survivability

- IP routing recovers from faults in seconds to minutes
- SONET recovers in 50 ms
- MPLS targets in-between

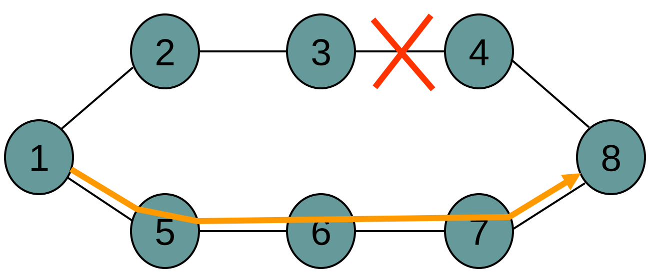
MPLS Restoration



Normal operation



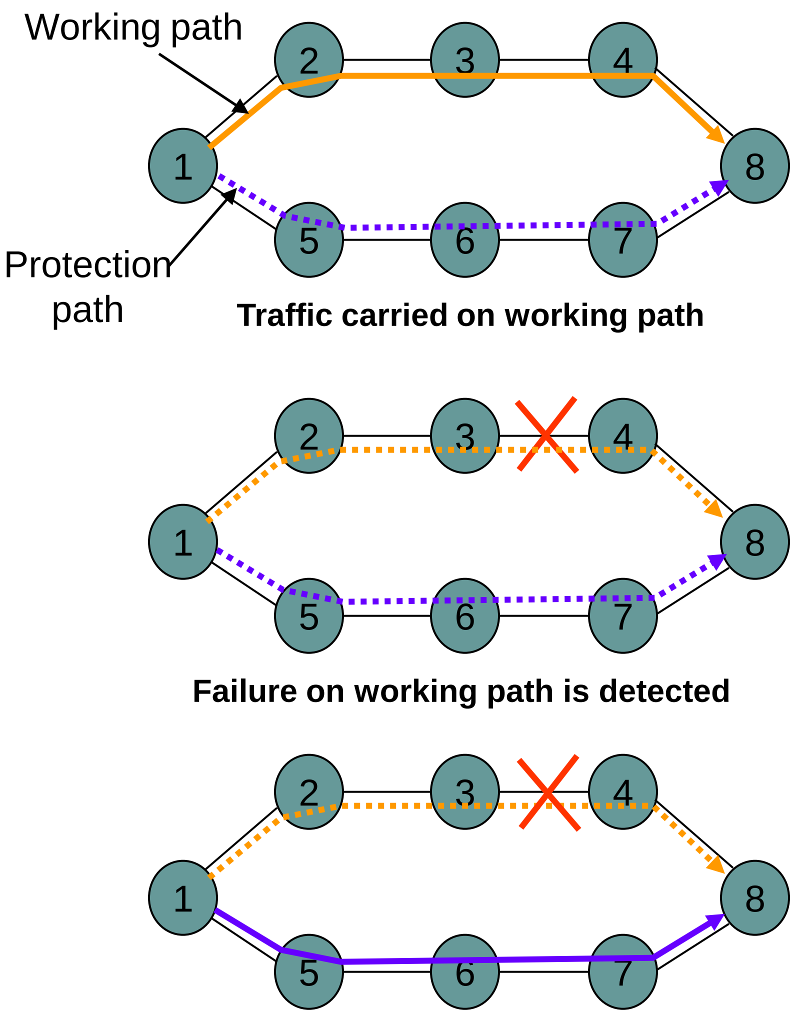
Failure occurs and is detected



Alternate path is established, and traffic is re-routed

- No protection bandwidth allocated prior to fault
- New paths are established after a failure occurs
- Traffic is rerouted onto the new paths

MPLS Protection



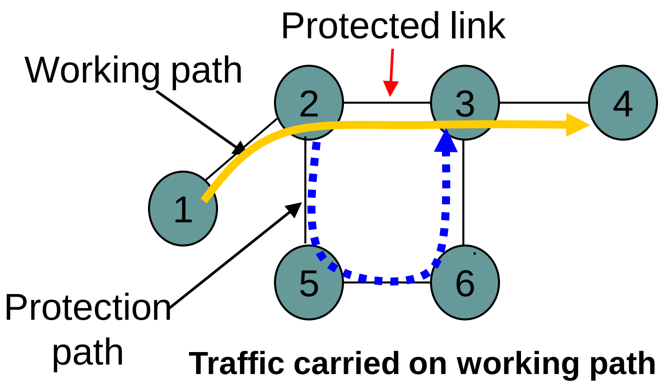
Traffic carried on working path

Failure on working path is detected

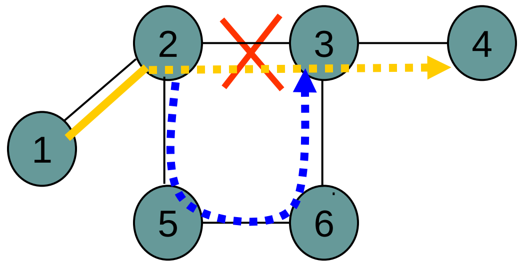
Traffic is switched to the protection path

- Protection paths are set up as backups for working paths
  - 1+1: working path has dedicated protection path
  - 1:1: working path shares protection path
- Protection paths selected so that they are disjoint from working path
- Faster recovery than restoration

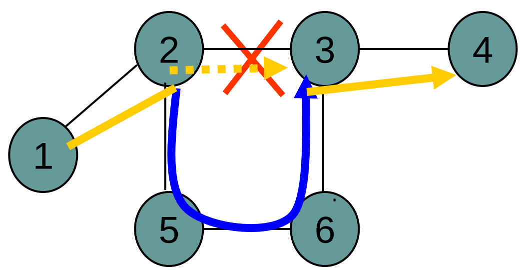
Link Protection (Local Protection)



Traffic carried on working path



Failure on working path is detected



Traffic is switched to the protection path at node 2

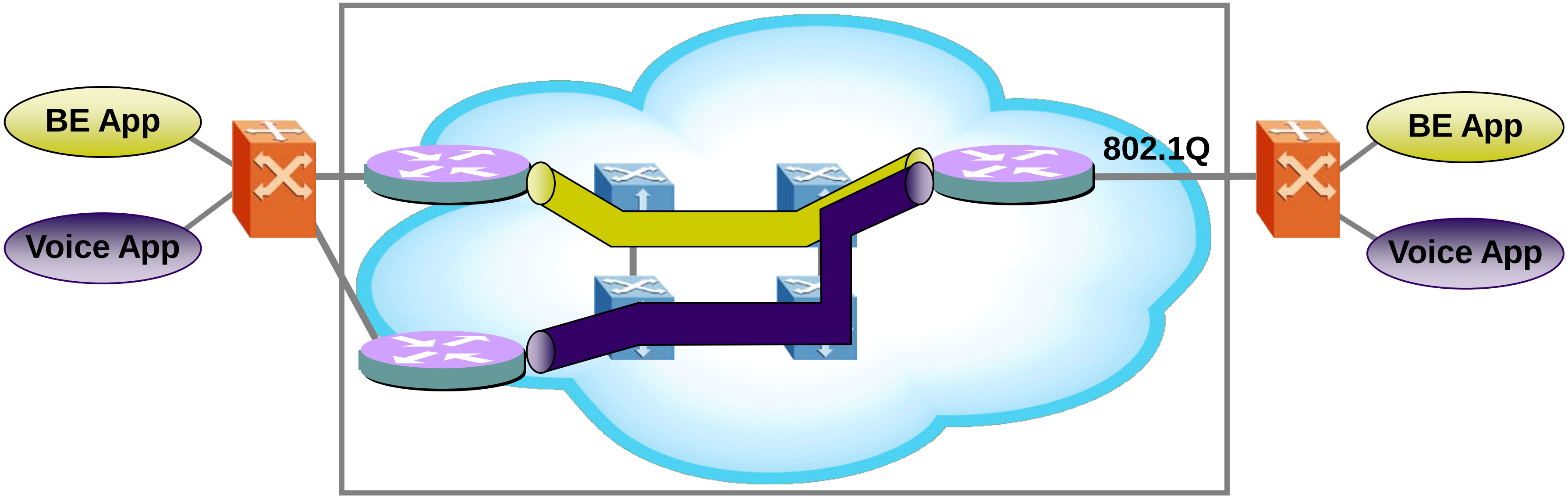
- Protection path is setup as backup for a segment of the working path (1-2-3-4)
  - 1+1: working path has dedicated protection path
  - 1:1: working path shares protection path
- Protection path (2-5-6-3) selected to support a critical link (2-3)
- Faster recovery than restoration (1-2-5-6-3-4)

MPLS and Quality-of-Service

MPLS QoS Using EXP

- QoS is specified in the Exp field which has 3 bits.
- Value copied from IP header (ToS) or others
- IP header ToS has 3 bits, but it has been extended to 6 bits for DiffServ.
- If QoS levels <= 8, no problem
- What if it is > 8?
  - QoS is inferred from label

Example of QoS Using Labels



- The Best Effort traffic (blue) and the voice traffic (red) take divergent paths on the network
- The red path is optimized through traffic engineering for low latency applications