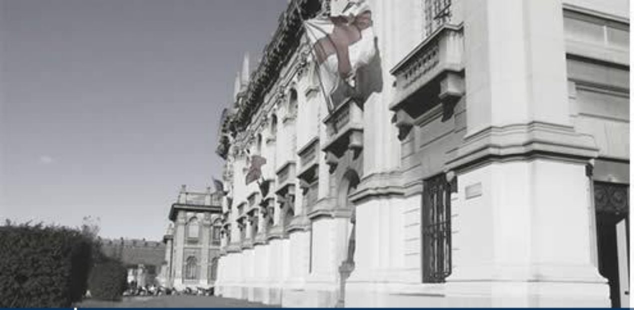


POLITECNICO DI MILANO

MPLS

Traffic Engineering with MPLS

Traffic Engineering

Know your network → measure network traffic

Optimize your network → traffic engineering (TE)

Several techniques

- IP traffic engineering
- MPLS traffic engineering
- other new techniques (e.g. ethernet)

References:

D. Medhi, K. Ramasamy. Network Routing. Algorithms, Protocols, and Architertures. Morgan Kaufman 2007

IP Traffic Engineering
MPLS Traffic Engineering

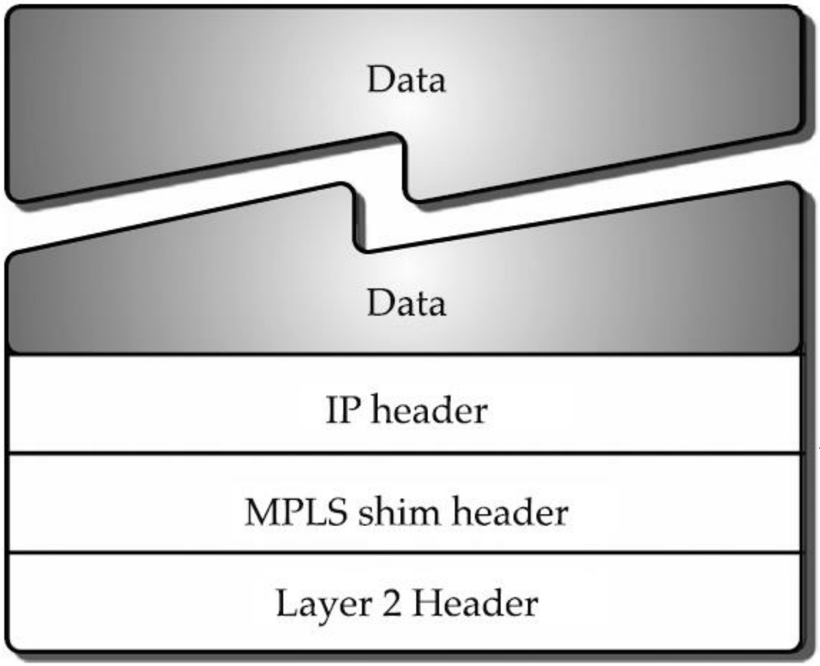
IP Traffic Engineering (IP-TE)

- Given the traffic matrix, find the optimal link weights
- Optimal according to some metrics
  - minimum delay, maximum space bandwidth, etc.
- Very few knobs, because of limitations of IP routing
  - Topology based. Current link occupation has no impact on routing
  - Destination driven. All packets going to the same destination follow the same route
- Need something to overcome these limitations

MPLS Traffic Engineering

- Allows to route flows according to
  - The “class” (remember packet classification!)
  - The current available bandwidth on a link (instead of the total bandwidth)

Multi Protocol Label Switching Header



Label (20 bit)	EXP (3 bit)	S(1)	TTL (8 bit)
L1		1	
L2		0	
...		0	
Ln		0	

The MPLS header is at layer 2.5. It is composed of a stack of elements

Label – the label, from a flat addressing space

EXP – special uses, e.g. Quality of Service class

S – if equal to 1 this is the last label

TTL – copied from the packet at ingress, to the packet at egress

Multi Protocol Label Switching
In brief

In brief.

The routers at the network edge (Label Edge Routers, LER)

- Classify the packet (according to destination and other fields)
- Apply a fixed-length label in front of the packet

The routers in the network core (Label Switching Routers, LSR)

- Switch the packet according to the label

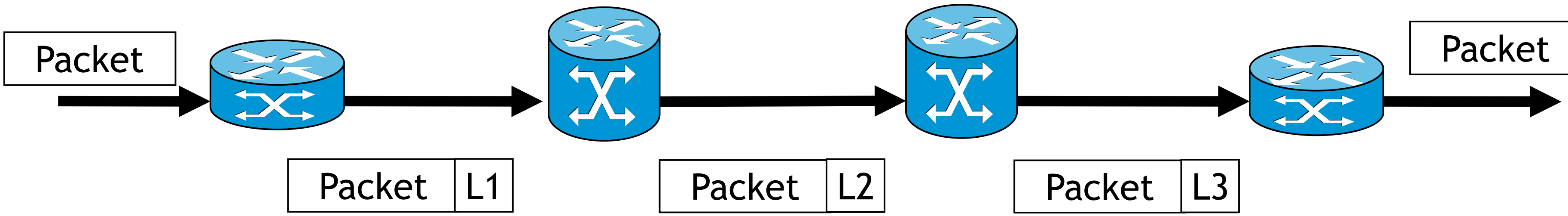
A modified routing protocol

- Collects link information (e.g. Available bandwidth)

A signalling protocol

- Fills switching tables in LSRs
- Reserves bandwidth in links

Switching in an MPLS network



1) The LER classifies the packets entering the network and assignes them to a Forwarding Equivalence Class (FEC)

2) The LER assigns a label according to the FEC (*label imposition*)

3) The LSR classifies the packet according to the label and switches the packet. The value of the label changes (*label switching*)

4) The LER at network exit removes the label (*label disposition*) and routes the packet according to the standard routing rules for the packet

In principle MPLS works for any L3 protocol and for any L2 protocol

Label Switched Path (LSP)

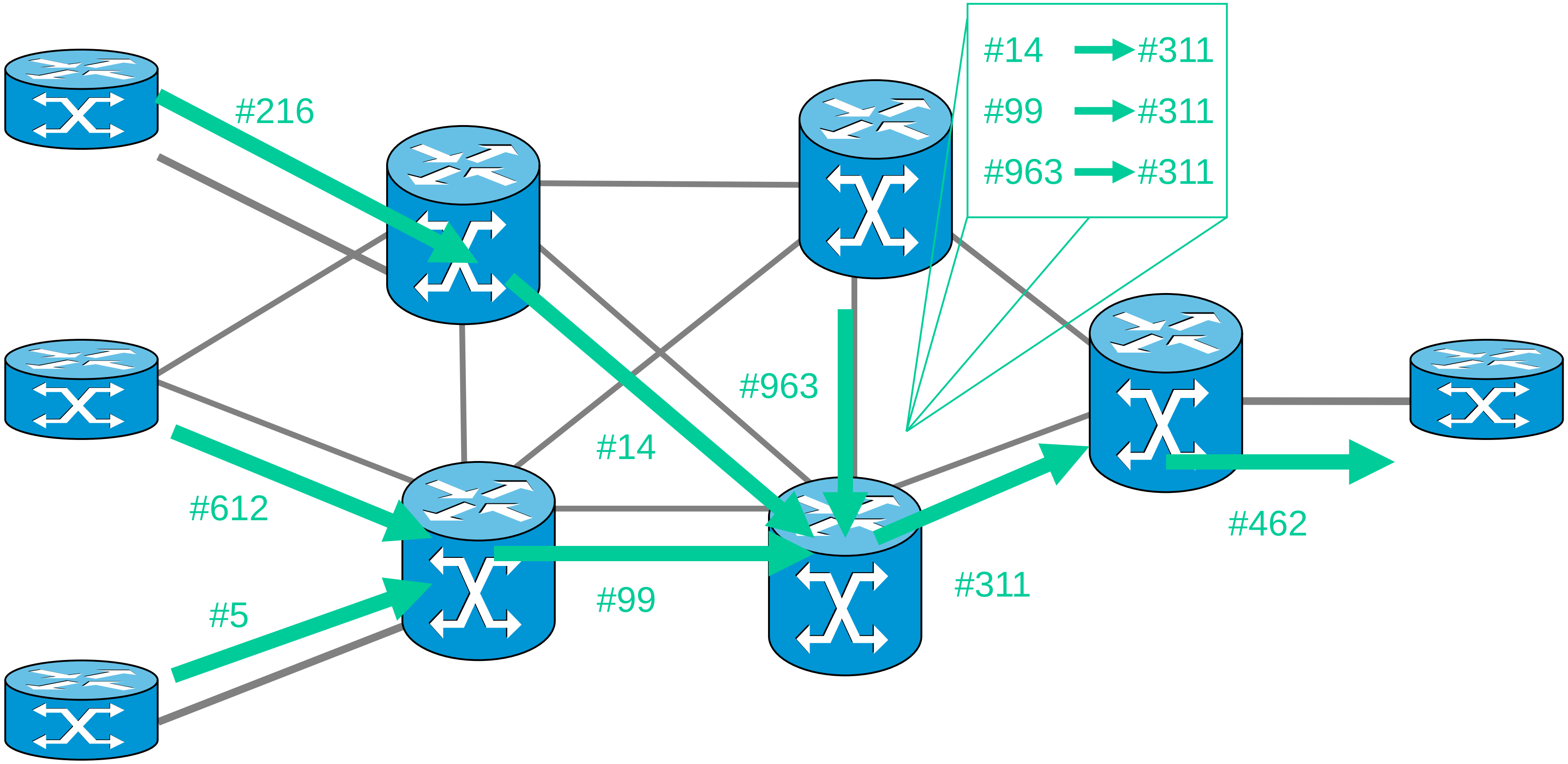
The sequence of labels and links traversed by the packets belonging to a given FEC forms a Label Switched Path (LSP)

- LSPs can be nested inside other LSPs using label stacks
- LSPs are unidirectional
- Only the outer label can be used for routing.

Types of LSP

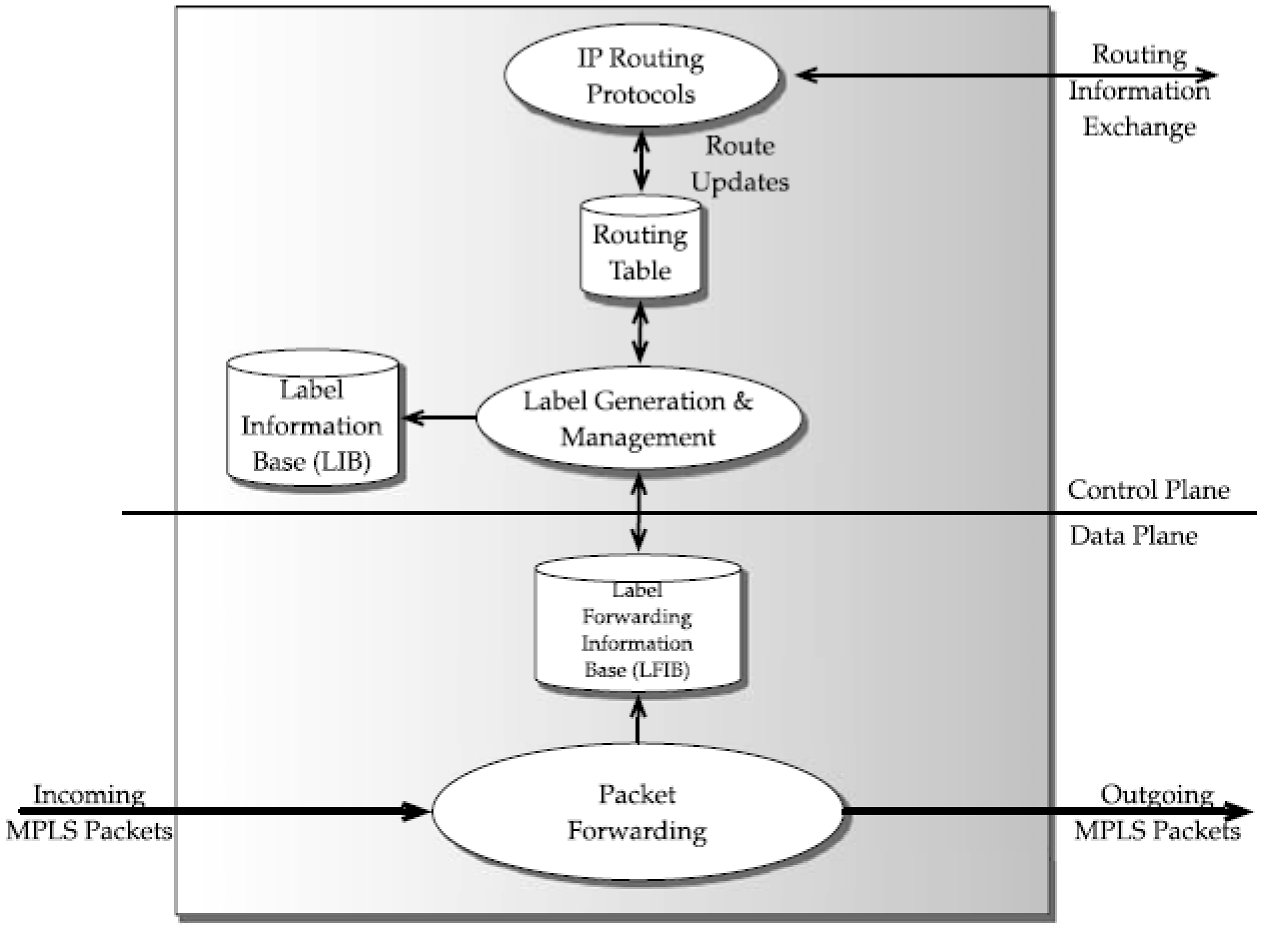
- Hop-by-hop: the next hop is obtained with the standard routing protocols and changes if topology changes
- Explict: the next hop is signalled by a Traffic Engineering protocol (RSVP-TE) and changes must be communicated by the same protocol. These are generally known as TE tunnels.

Label Switched Path



LSPs can merge along their path.

Logical Structure of LSR



Label Information Base (LIB)

The LIB contains, for each FEC, the FEC-label associations

Each entry of the LIB contains:

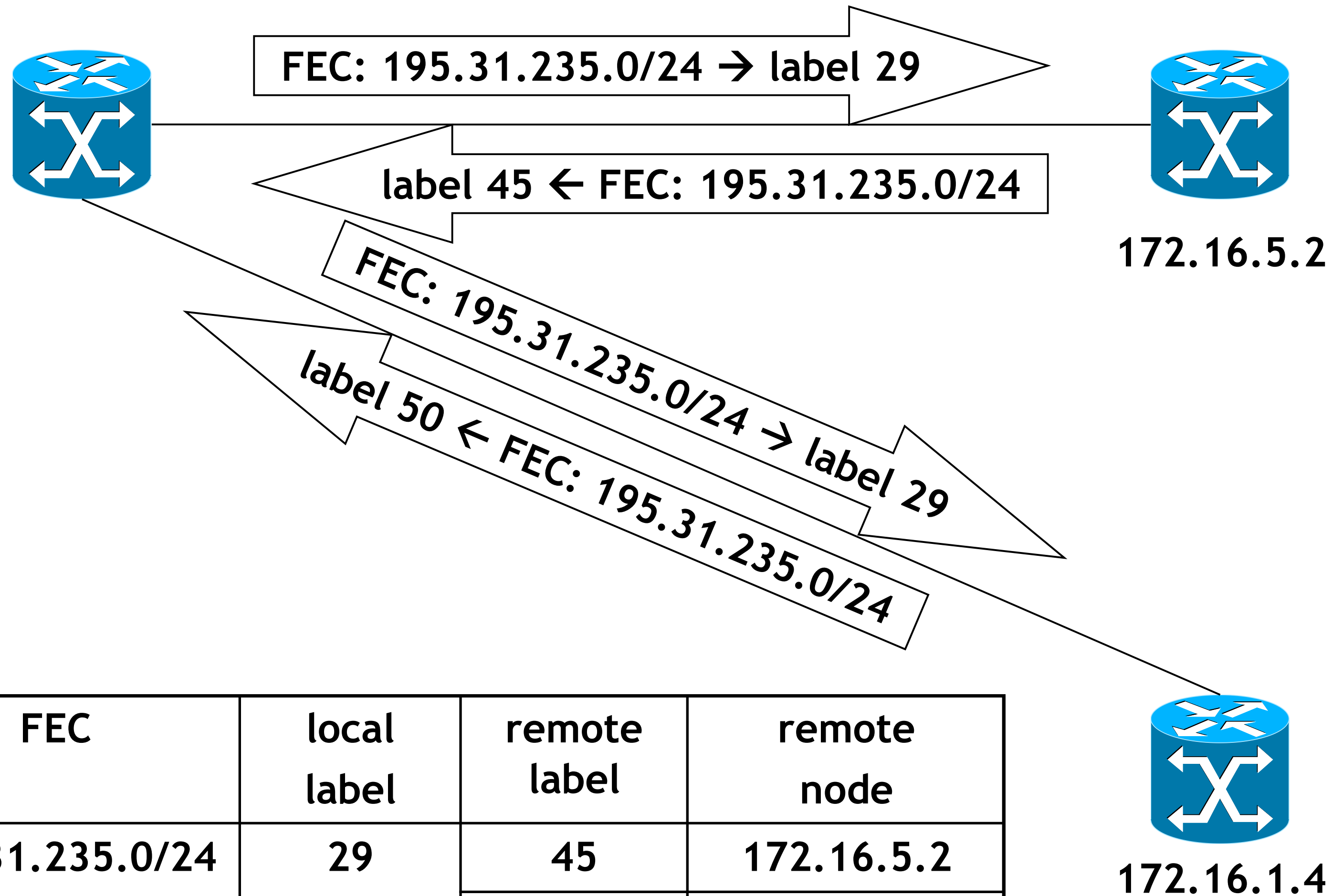
- Definition of the FEC
- Local label. This is the label that other nodes must use for sending to this node packets belonging to this FEC to this node
- Remote label and remote node. For each neighbor the node knows the label that must be used to send to that node packets belonging to this FEC

The LIB is populated by a signaling protocol

- For hop-by-hop LSPs by the Label Distribution Protocol (LDP)
- For explicit LSPs by the Reservation Protocol for Traffic Engineering (RSVP-TE)

For each FEC, the LIB contains information about several neighbors, cannot be used for fowarding packets.

Population of the LIB



FEC	local label	remote label	remote node
195.31.235.0/24	29	45	172.16.5.2
		50	172.16.1.4
...	...	...	...

Label Forwarding Information Base

Is obtained by the LIB selecting information about the LSPs currently in use.

Two versions:

- LERs use the FEC-to-NHLFE (FTN table)
- LSRs use the Incoming Label Map (ILM)

FTN and ILM Tables

FTN Table

- Used in edge nodes (LER)
- For each FEC gives the NHLFE

ILM Table

- Used in core nodes (LSR)
- For each label gives the NHLFE

The Next-Hop Label Forwarding Entry (NHLFE) contains the instructions to perform on the packet

- the next hop and the output interface
- the operations on the label (swap, push, pop)
- QoS information (e.g. priority)

Operations on Labels

Swap

- overwrites the top label

Pop

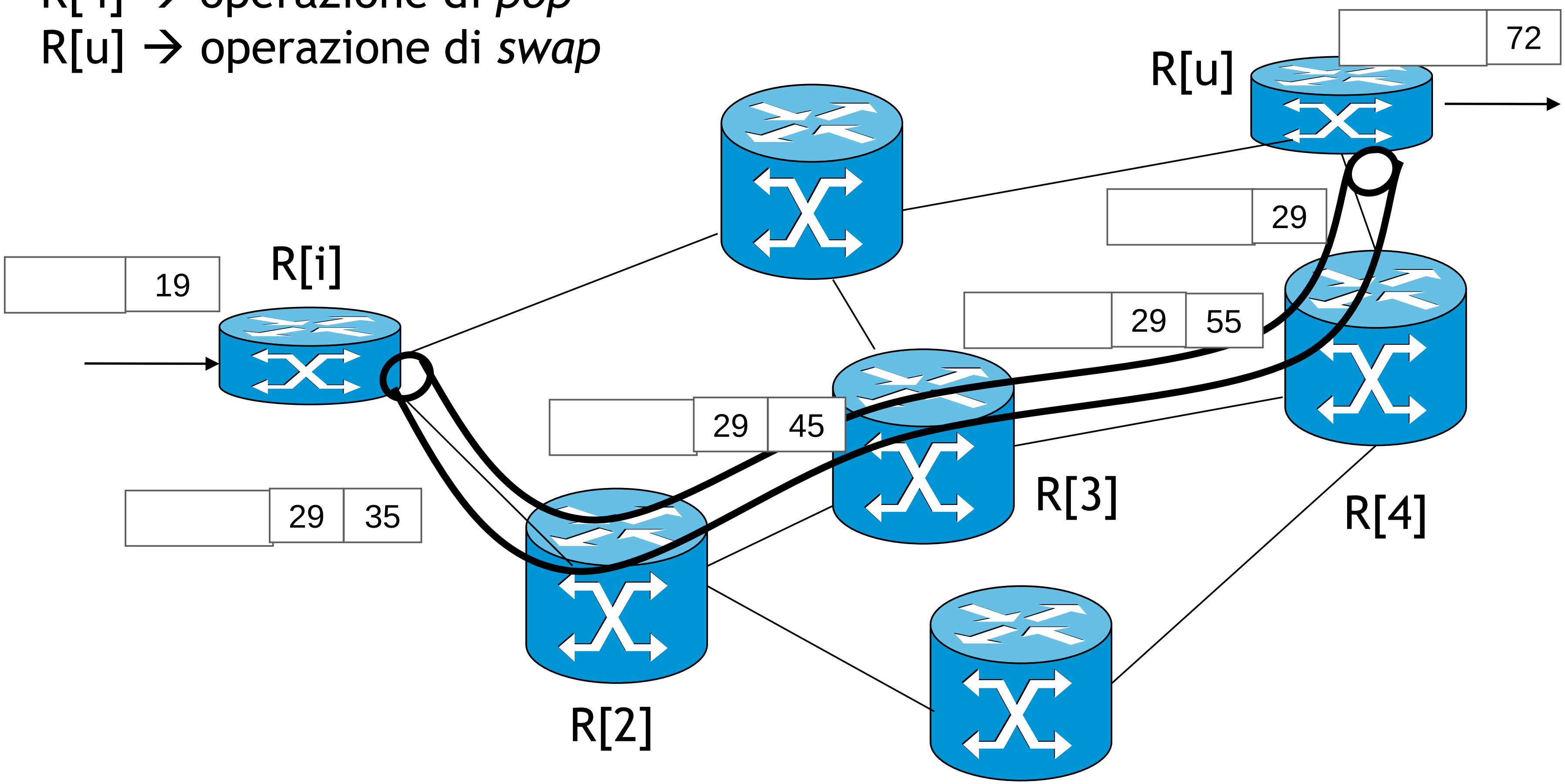
- removes the top label

Push

- if the packet is already labeled overwrites the top label
- adds a new top label

Example of LSP

R[i] → operazione di *push*
R[4] → operazione di *pop*
R[u] → operazione di *swap*



Components of MPLS Traffic Engineering

Definition of traffic flows

- Packet classification and association to a FEC

Description of the traffic flows (generally the peak bandwidth)

- Someone must tell us. For example from measurements.

Knowledge of the topology and of current usage of resources

- We get the topology from any link-state routing protocol
- Need to extend the protocol to distribute current link usage
- OSPF-TE

A mechanism for choosing where to route the TE tunnel

- Need to define an algorithm
  - Can be offline
  - Can be online (e.g. Constrained Shortest Path First, CSPF)

A protocol for configuring the new tunnel in the LIBs of the involved nodes

- RSVP-TE

Constrained Shortest Path First

Variation of the Shortest Path Algorithm

- a link is considered only if it has enough available bandwidth to meet the request
- a path must be computed only for a given destination *v*

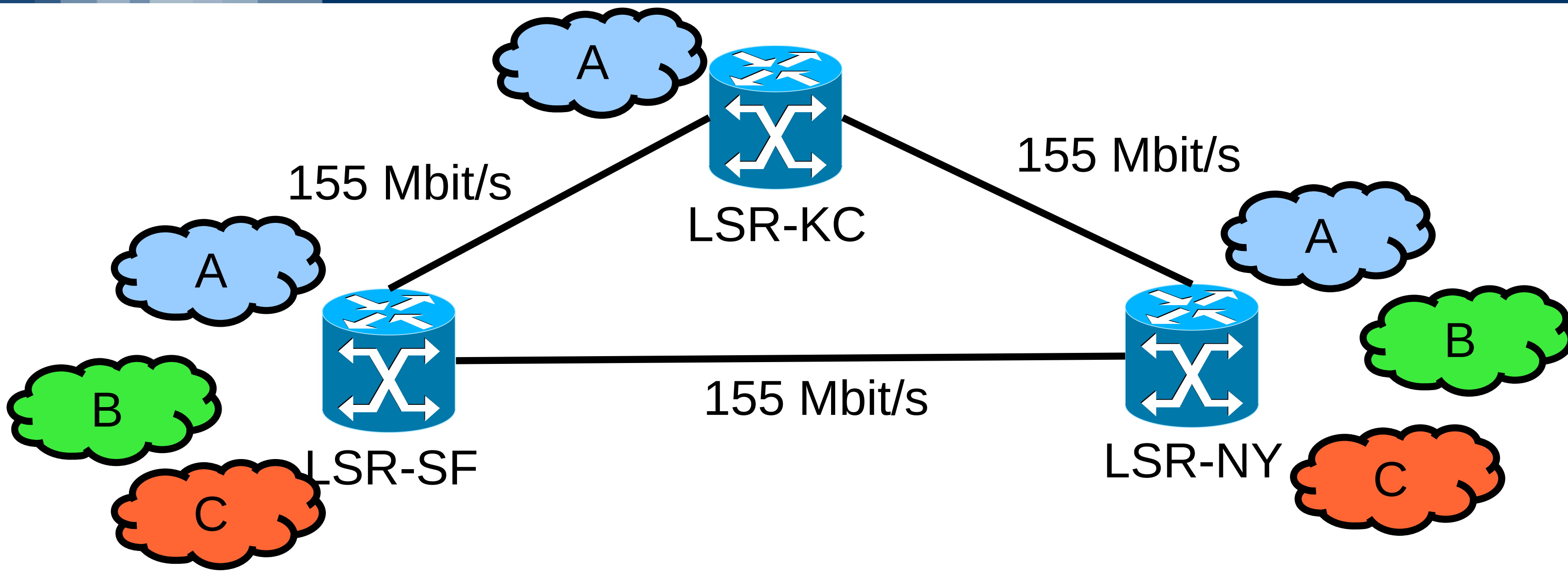
Implementation similar to Shortest Path Algorithm

- temporarily set to infinity the links that do not satisfy the constraint
- stop when node *v* is reached.

Issue: the algorithm is executed by each node starting a tunnel.

Different nodes can have different information about the available bandwidth. So, even if the algorithm finds a solution, the reservation can fail.

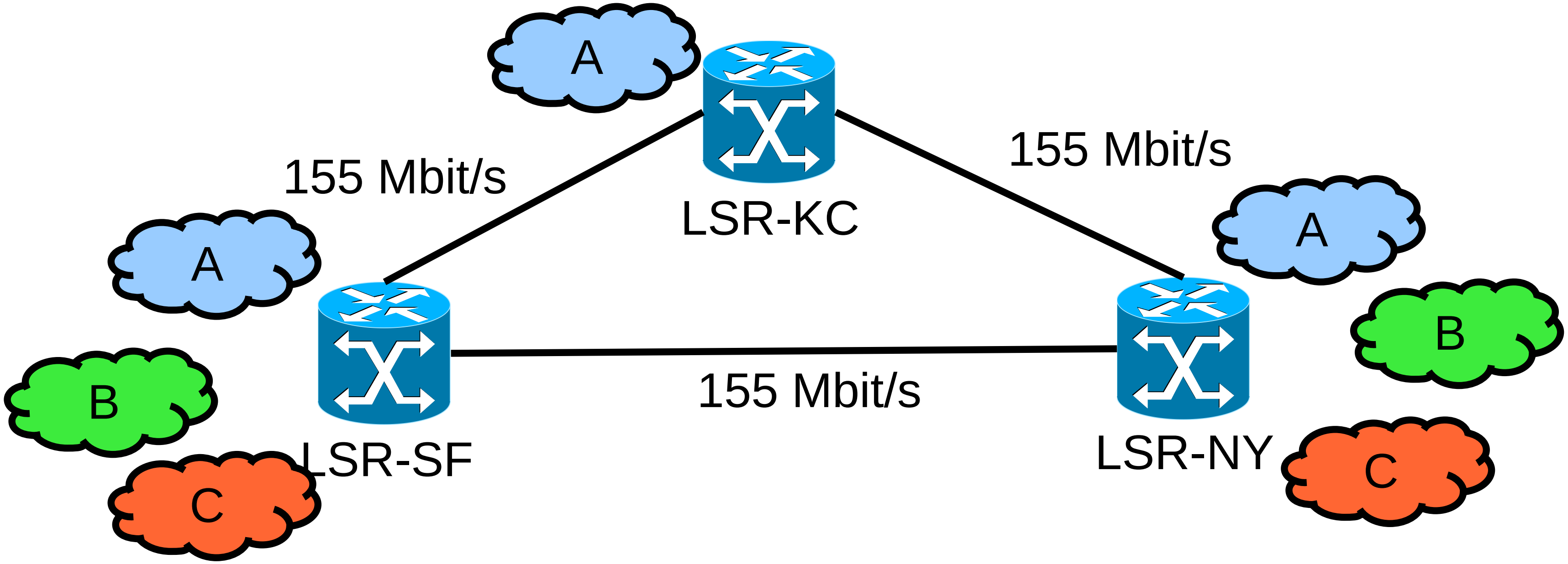
CSPF Example



Setup the following tunnels

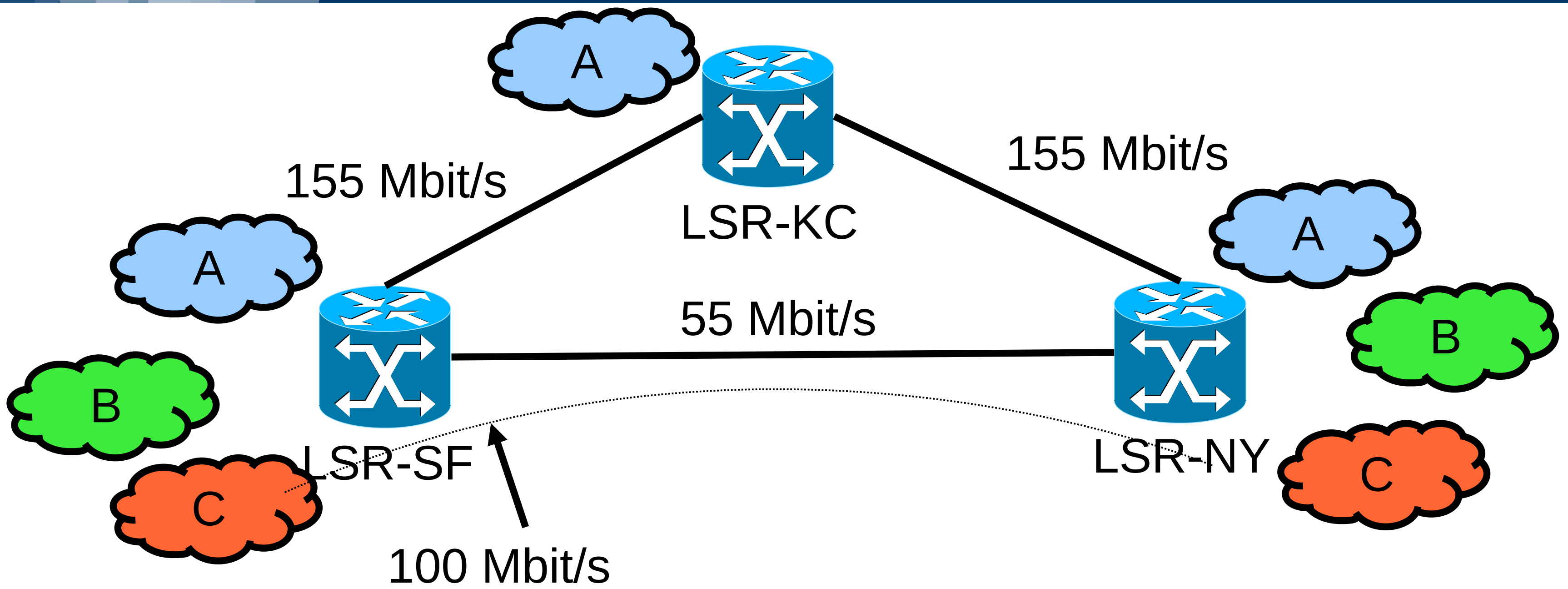
* 100 Mbit/s for customer C between SF-NY and reverse
* 80 Mbit/s for customer B between SF-NY and reverse
* 20 Mbit/s for customer A between SF-NY and reverse
* 45 Mbit/s for customer A between SF-KC and reverse
* 60 Mbit/s for customer A between KC-NY and reverse

CSPF Example



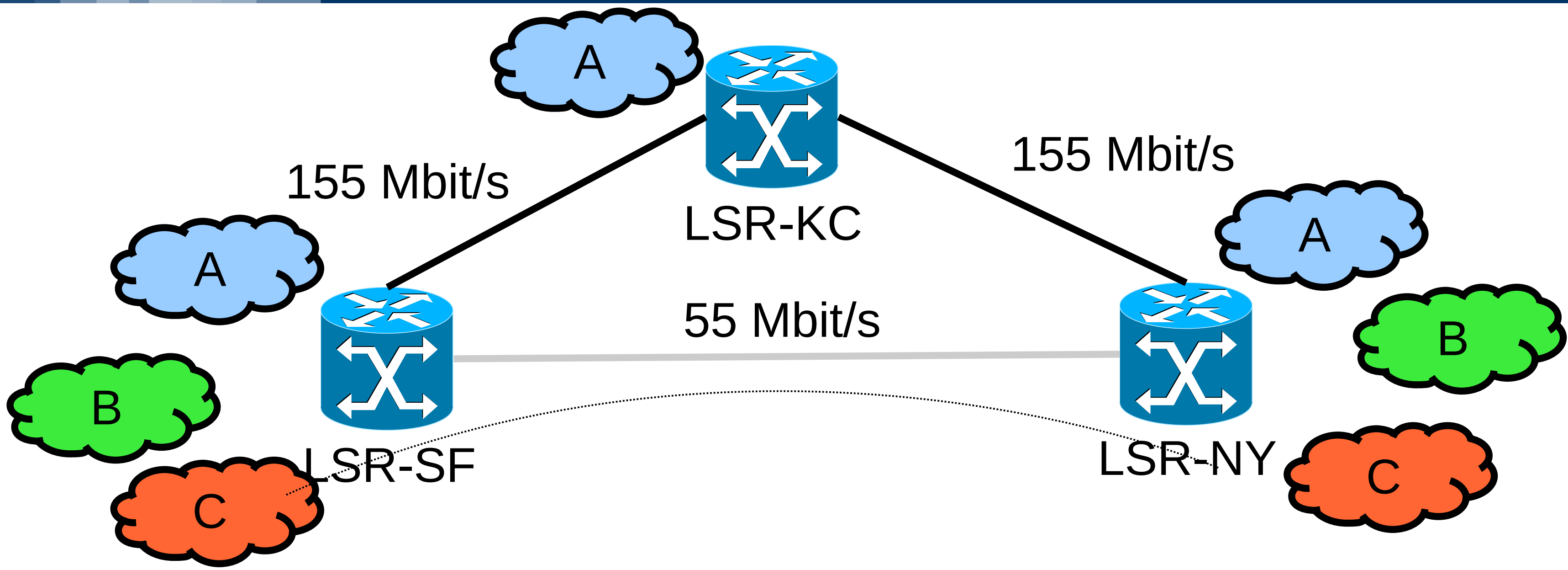
100 Mbit/s for customer C between SF-NY and reverse
All the links meet the bandwidth constraint
The shortest paths are SF-NY and NY-SF

CSPF Example



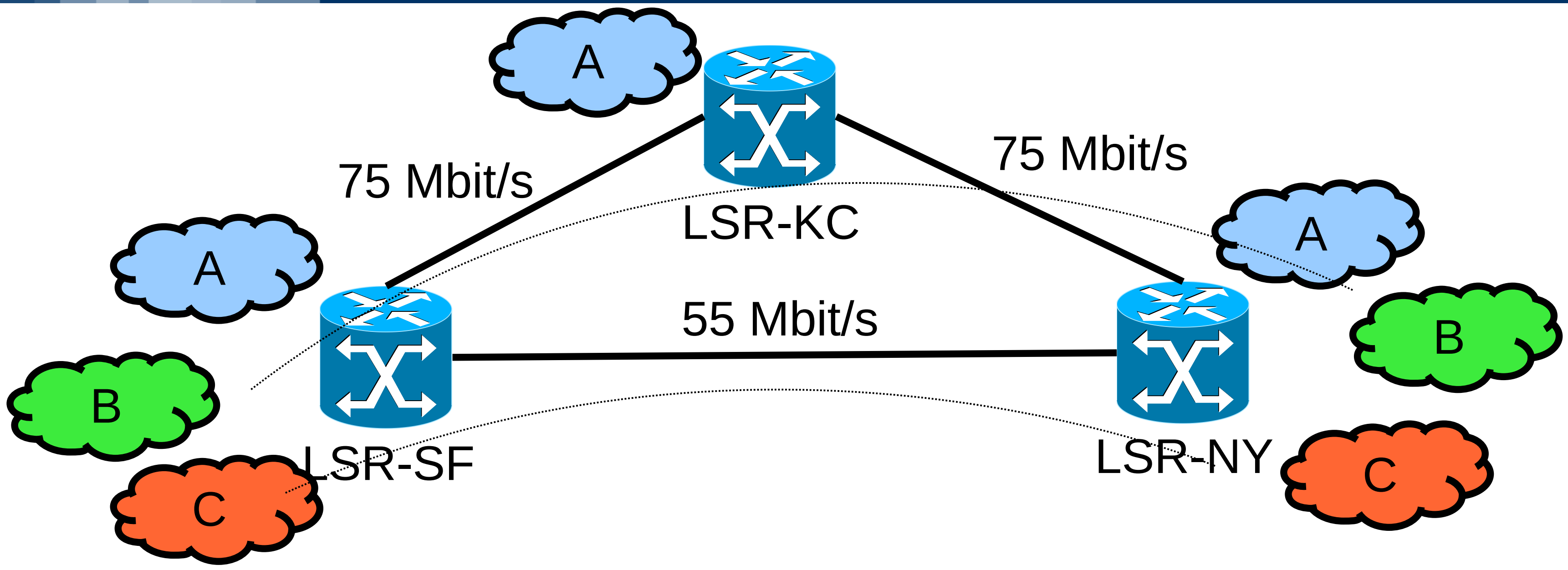
The tunnels are set up on the link between SF and NY
The available bandwidth on SF-NY is equal to 55 Mbit/s

CSPF Example



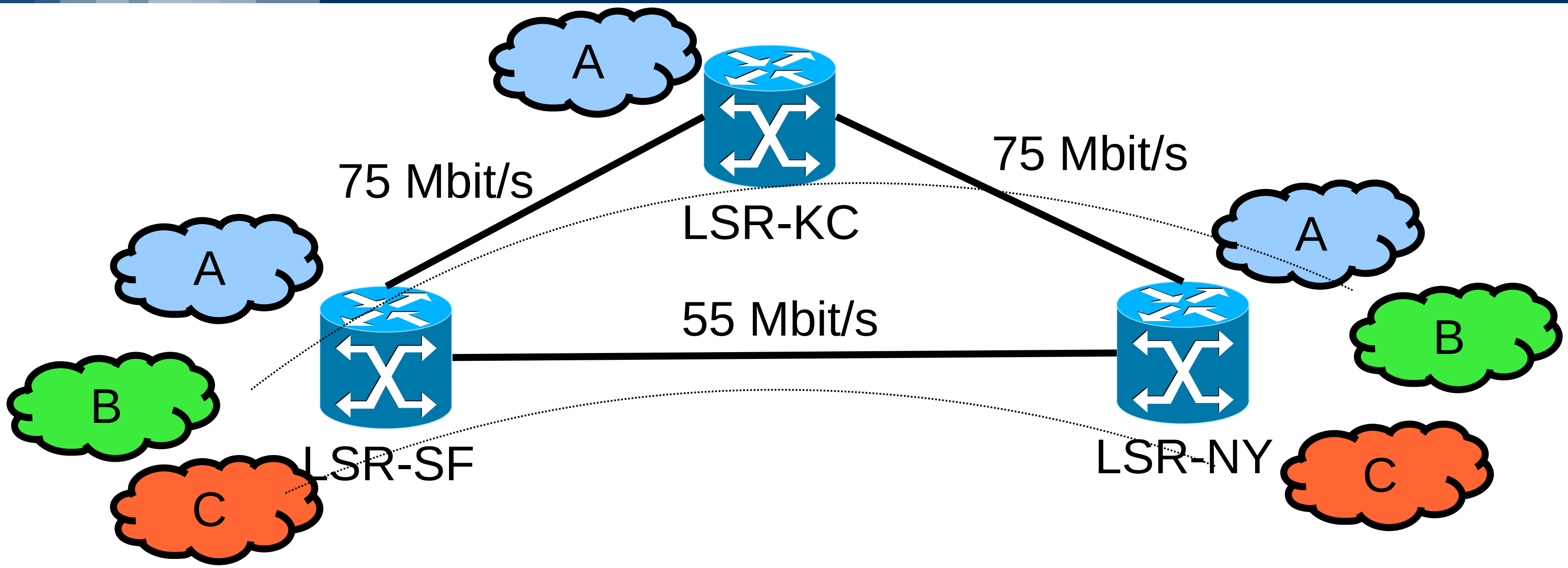
80 Mbit/s for customer B between SF-NY and reverse
The SF-NY link does not meet the bandwidth constraint
The shortest paths are SF-KC-NY and NY-KC-SF

CSPF Example



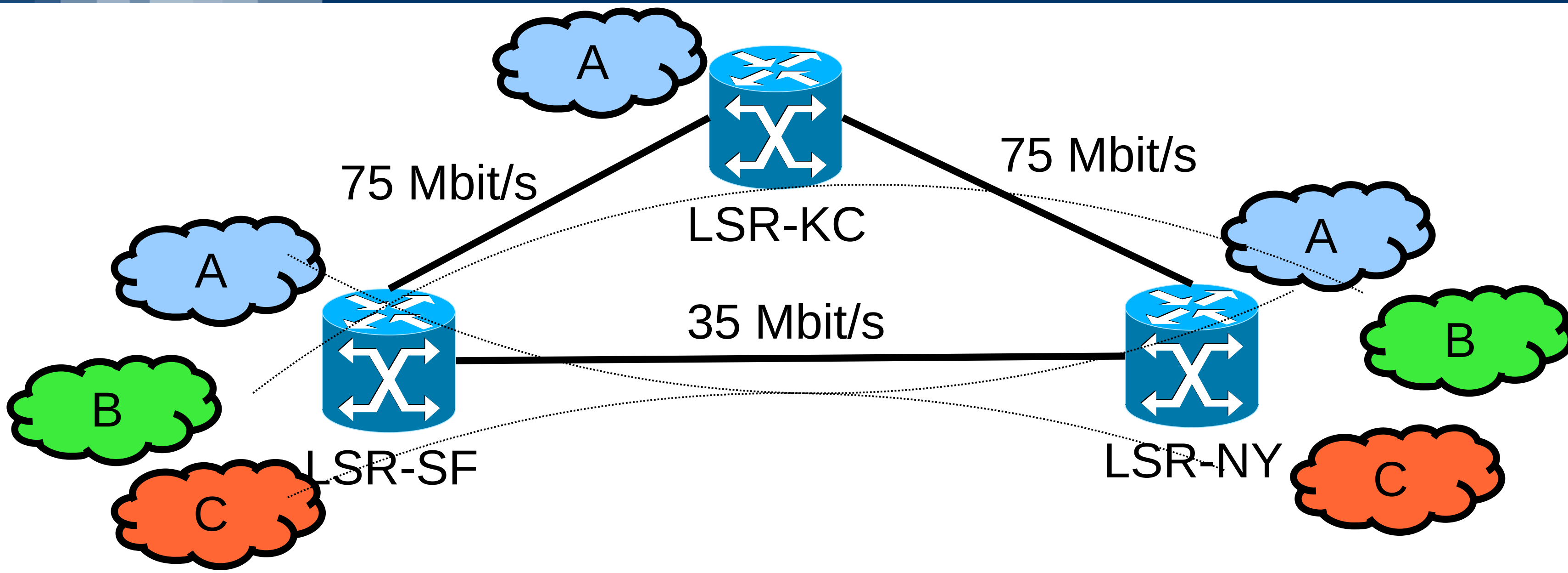
The tunnels are set up on the route SF-KC-NY
The available bandwidth of SF-KC and KC-NY becomes 75 Mbit/s

CSPF Example



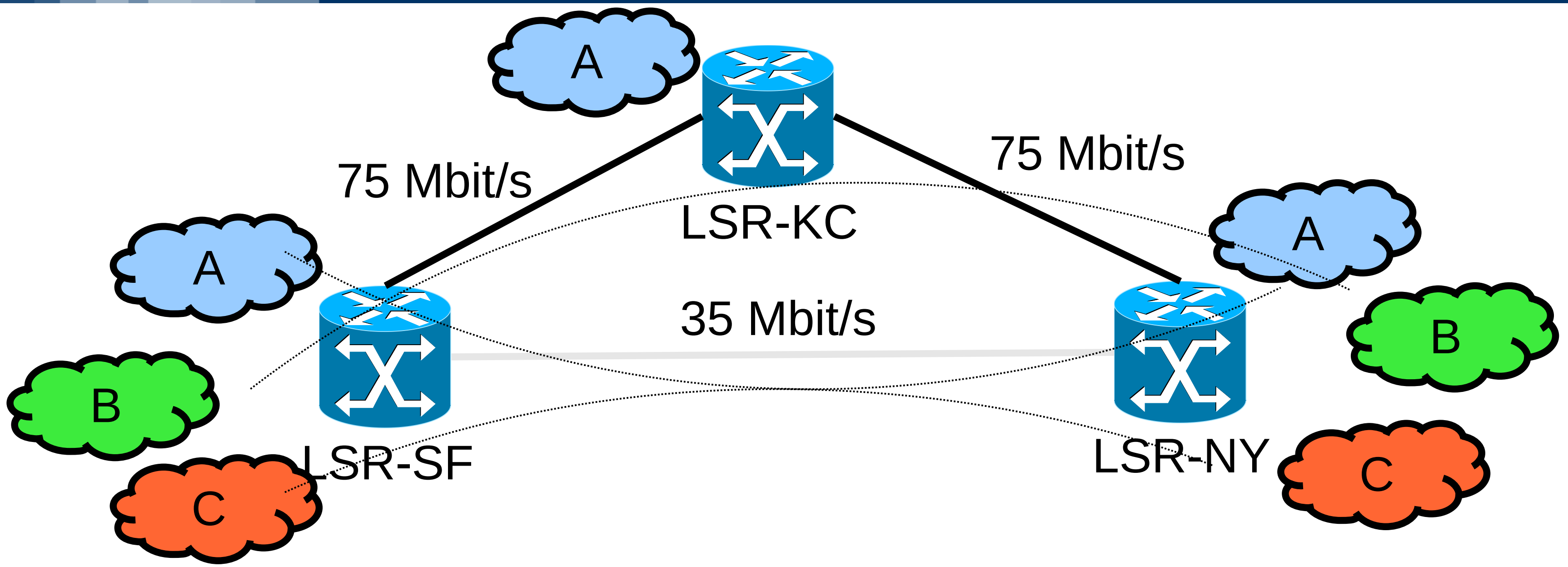
20 Mbit/s for customer A between SF-NY and reverse
All the links satisfy the requirement
Shortest Path is SF-NY

CSPF Example



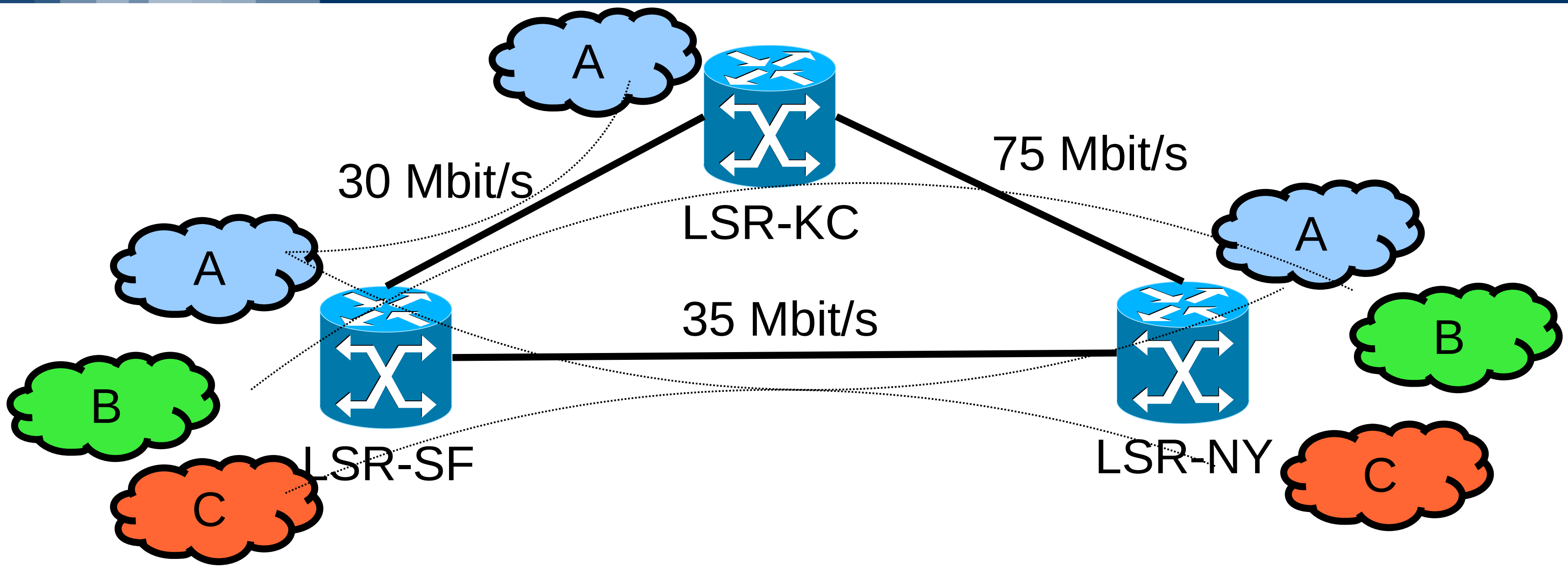
Tunnel is set up between SF-NY
Bandwidth is updated to 35 Mbit/s

CSPF Example



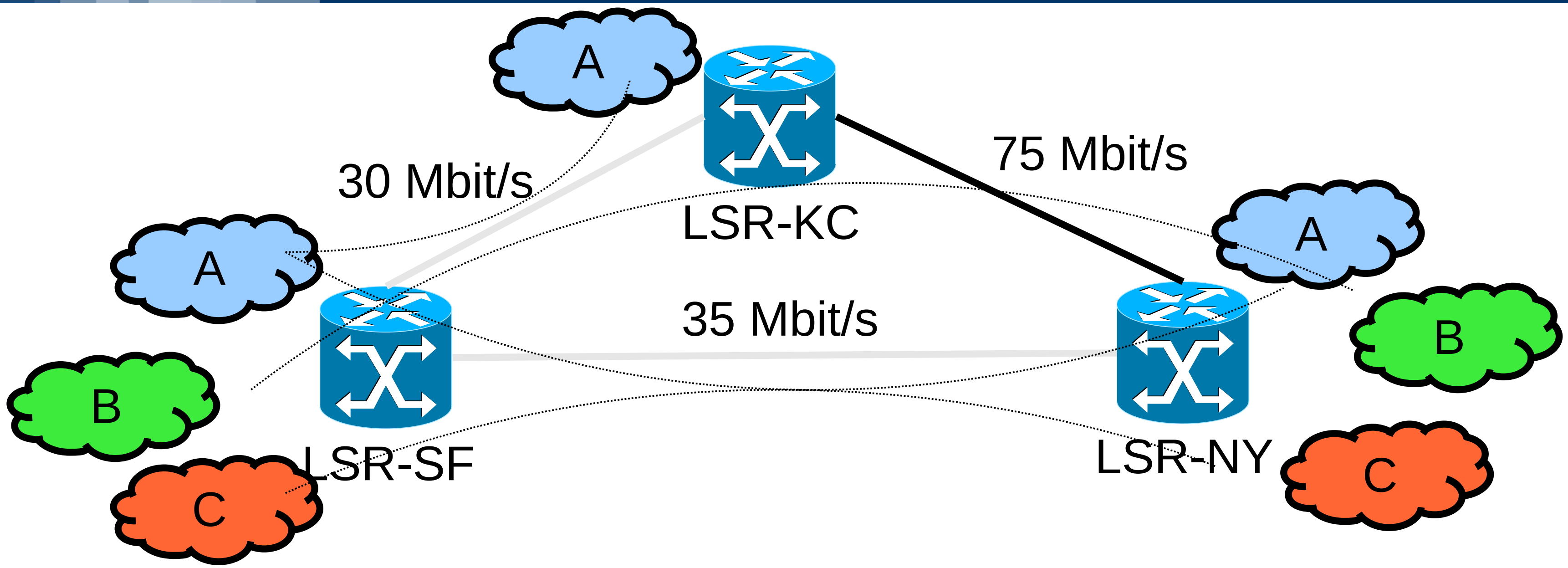
45 Mbit/s for customer A between SF-KC and reverse
Link SF-NY does not meet the requirement
Shortest Path is SF-KC

CSPF Example



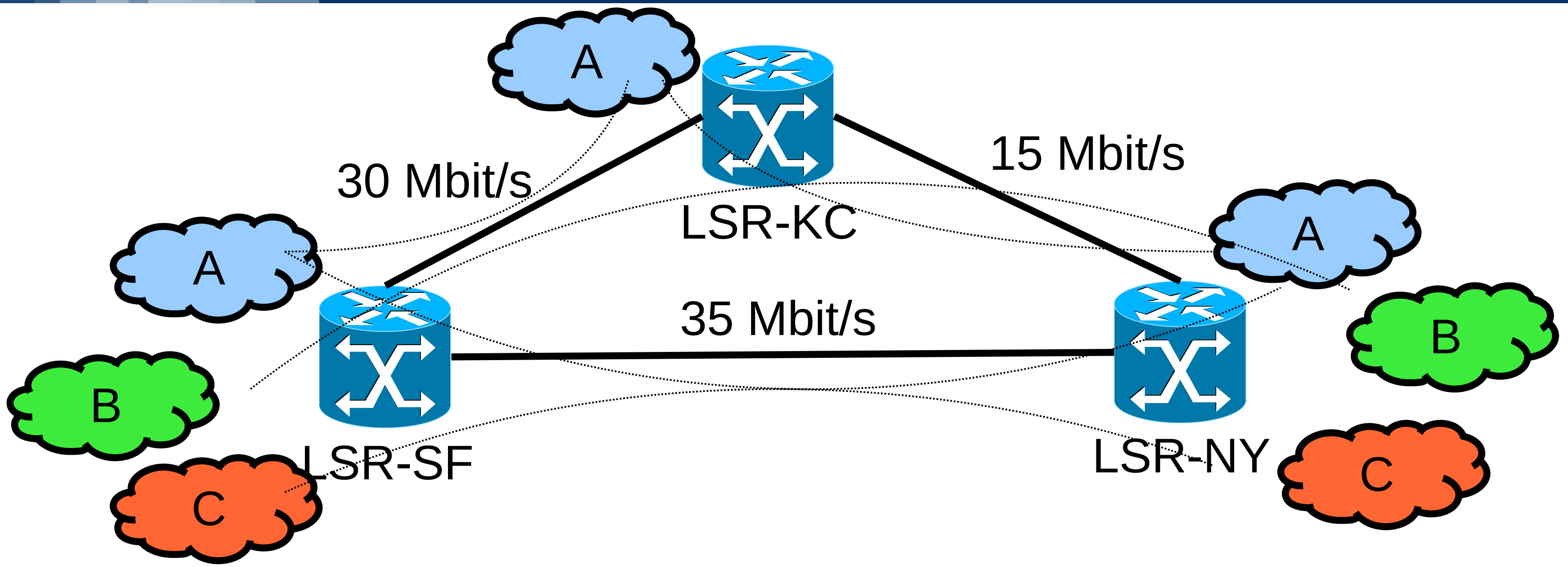
Tunnel is set up between SF-KC
Available bandwidh between SF-KC becomes 30 Mbit/s

CSPF Example



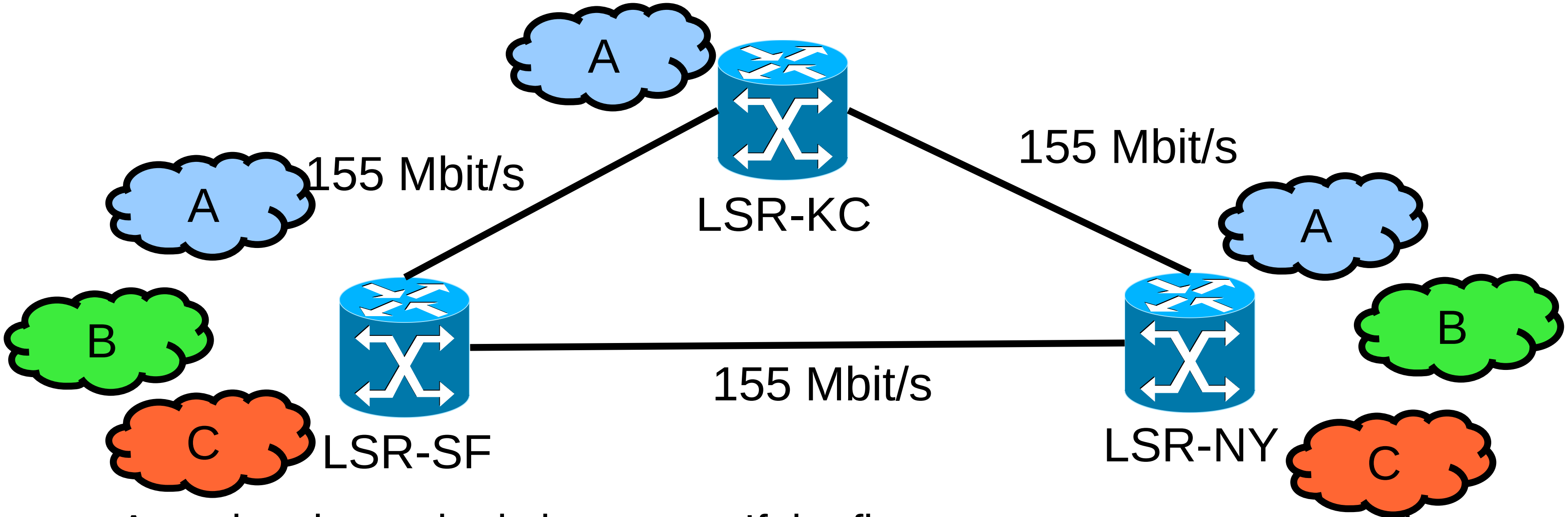
60 Mbit/s for customer A between KC-NY and reverse
Links SF-KC and SF-NY do not meet the requirement
Shortest Path is KC-NY

CSPF Example



Tunnel is set up between KC-NY
Available bandwidth between KC-NY becomes 15 Mbit/s

CSPF Example 2



Attention that order is important. If the first two request are swapped:

* 80 Mbit/s for customer B between SF-NY and reverse tunnel between SF-NY. Bandwidth of SF-NY → 75
* 100 Mbit/s for customer C between SF-NY and reverse tunnel SF-KC-NY. Bandwidth of SF-KC and KC-NY → 55
* 20 Mbit/s for customer A between SF-NY and reverse tunnel SF-NU. Bandwidth of SF-NY → 55
* 45 Mbit/s for customer A between SF-KC and reverse tunnel SF-KC. Bandwidth of SF-KC → 10
* 60 Mbit/s for customer A between KC-NY and reverse no route!