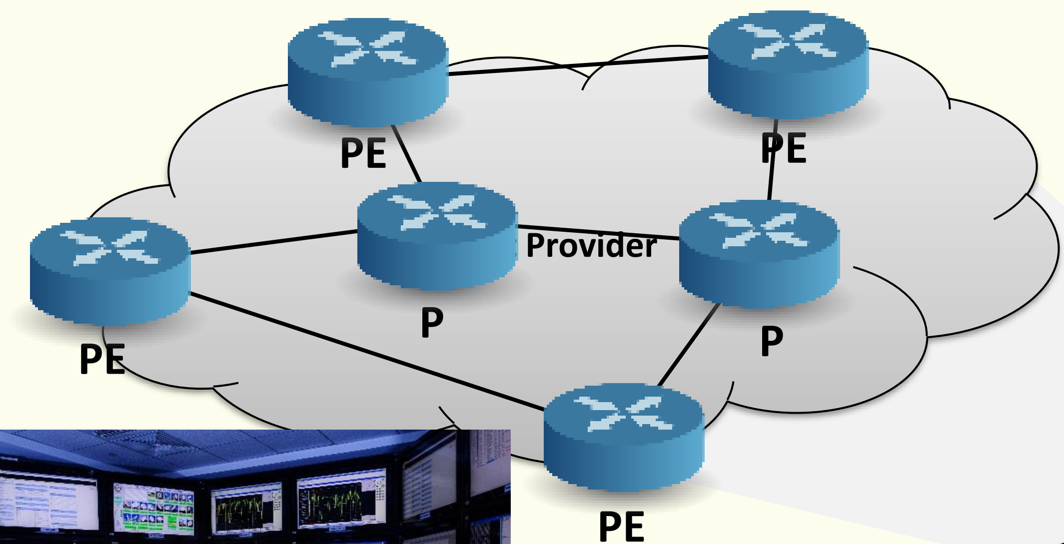
Core network protocols and architectures

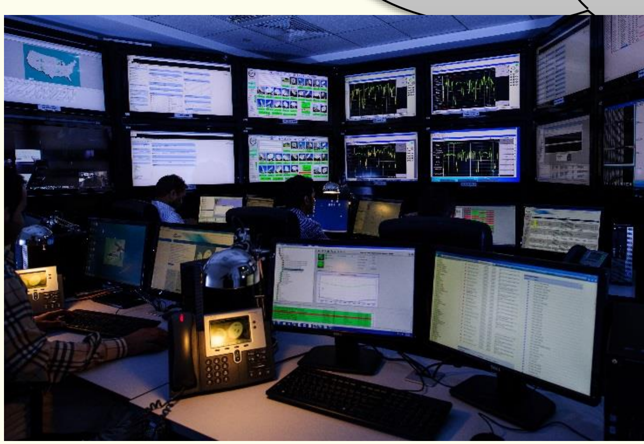
MPLS-based Traffic Engineering

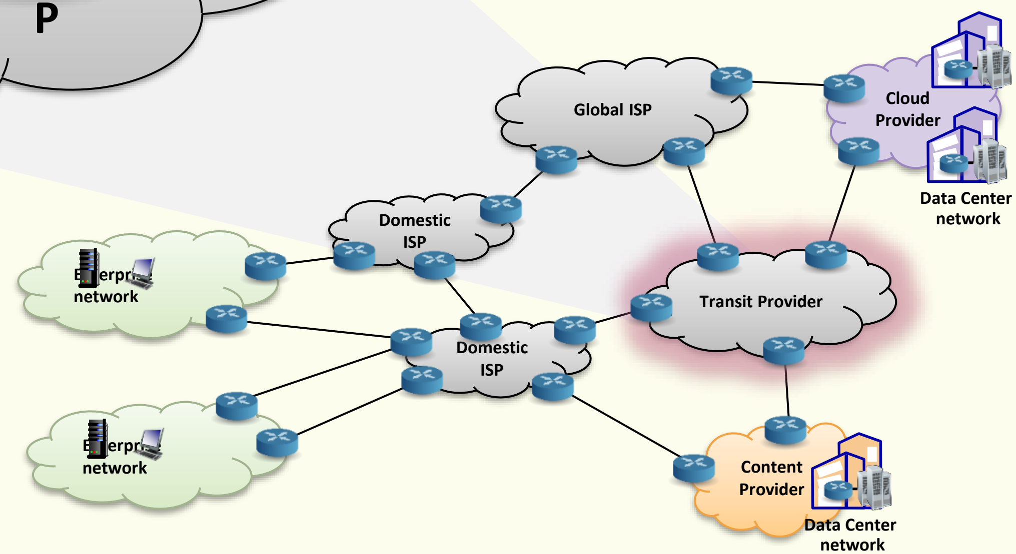
Enzo Mingozzi
Professor @ University of Pisa
enzo.mingozzi@unipi.it

MPLS-based Traffic Engineering

- Routing flexibility







Traffic Engineering

- **Controlling** the **path** taken by traffic **through a network**
- What is the purpose of influencing paths?
  - **Improve utilization** of network resources
    - Avoid congestion (and/or load unbalancing across the network)
  - Ensure the path has **certain characteristics**
    - E.g., not using high-latency links
  - Determine which traffic gets **priority** at a time of resource **crunch**

Traffic Engineering

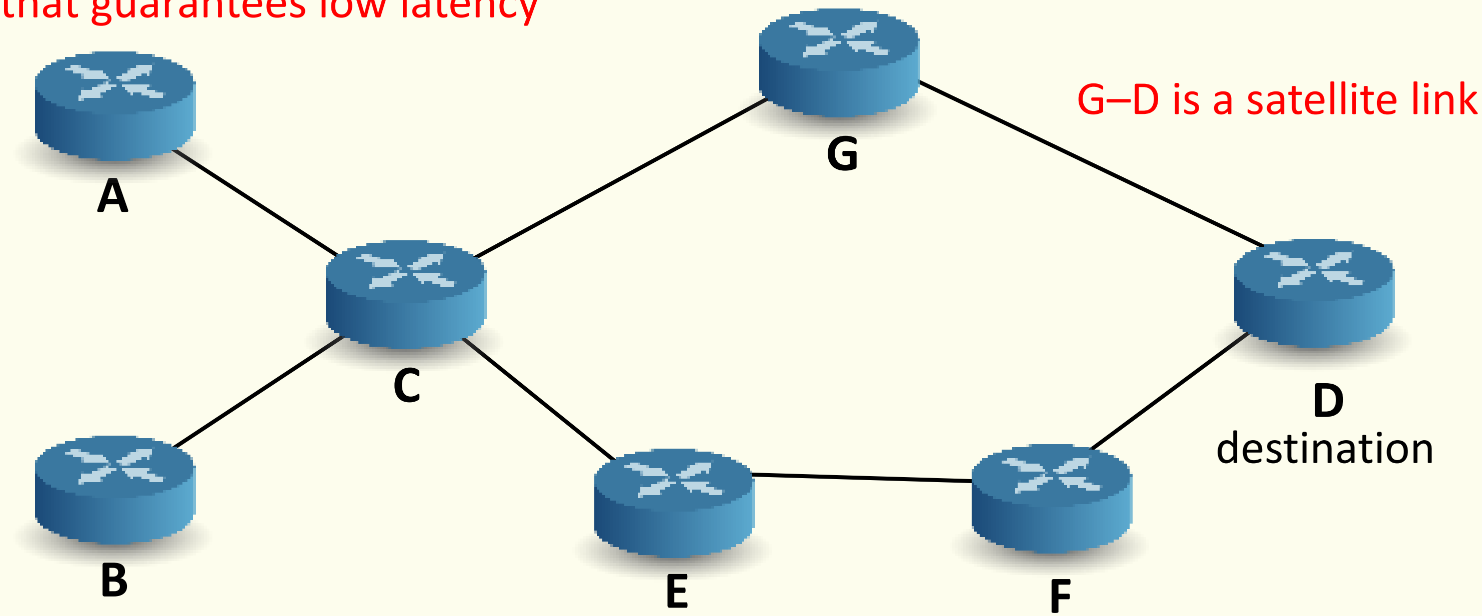
- Why is that **relevant** for a network operator?
  Increase revenues
  - **Offering new services** with extra guarantees
    - Extra guarantee -> extra money charged
    - E.g. guaranteed bandwidth
  - **Lowering capex** in new resources **by improving utilization** of existing ones
    - Cost savings by delaying link upgrades
    - E.g. by increasing average % of link utilization

Traffic Engineering

- TE is **not always required**
  - Not all MPLS deployments are used for TE
  - Not all MPLS networks can indeed provide TE
- TE entails operating a more complex network (i.e., **additional cost**)
  - The means by which TE is implemented must be simple enough to deploy and maintain
  - **MPLS** provides **operational simplicity** along with **flexibility** to support complex TE policies

Application scenario [1]

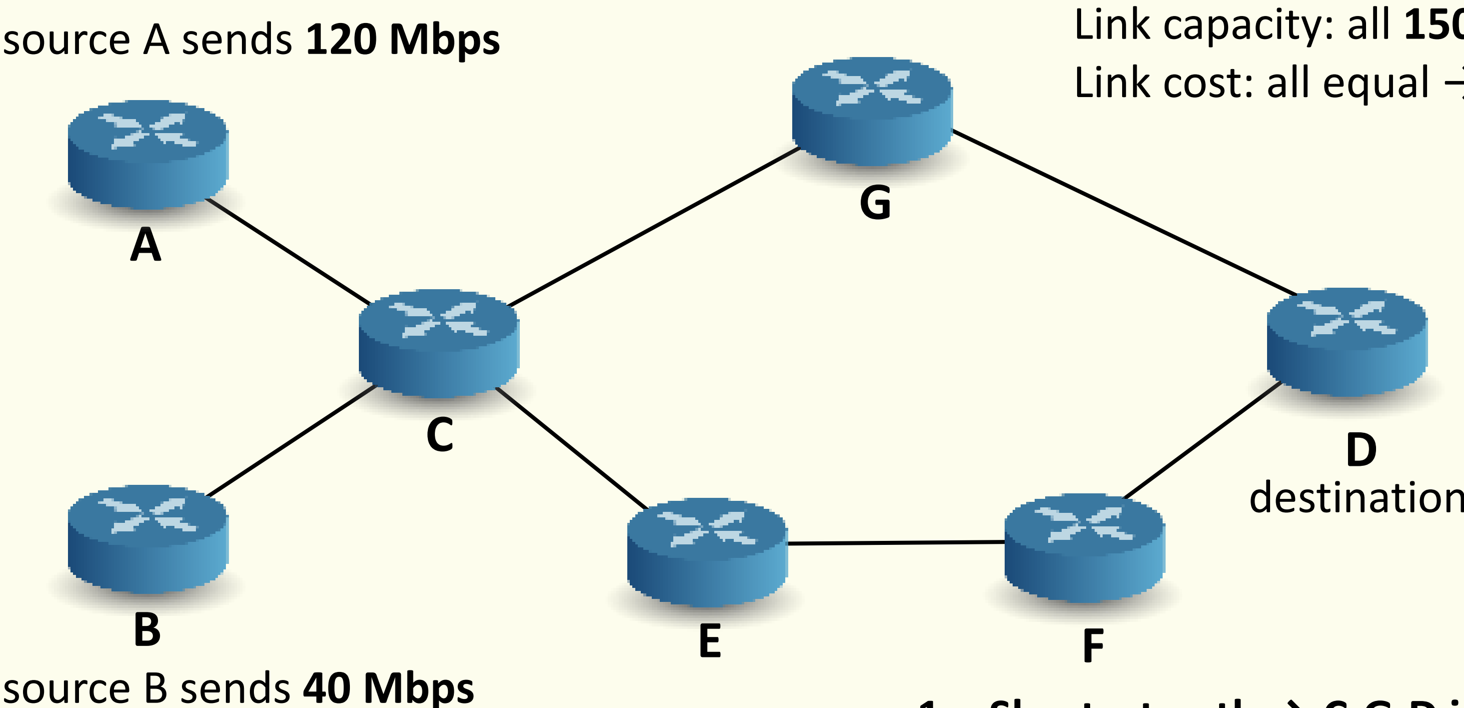
Customer A buys a service that guarantees low latency



traffic originating at A should avoid the high-latency link G–D

Ability to forward traffic along a path specified by the source, i.e., **explicit routing**

Application scenario [2]



Link capacity: all **150 Mbps**
Link cost: all equal → Path cost = #hops

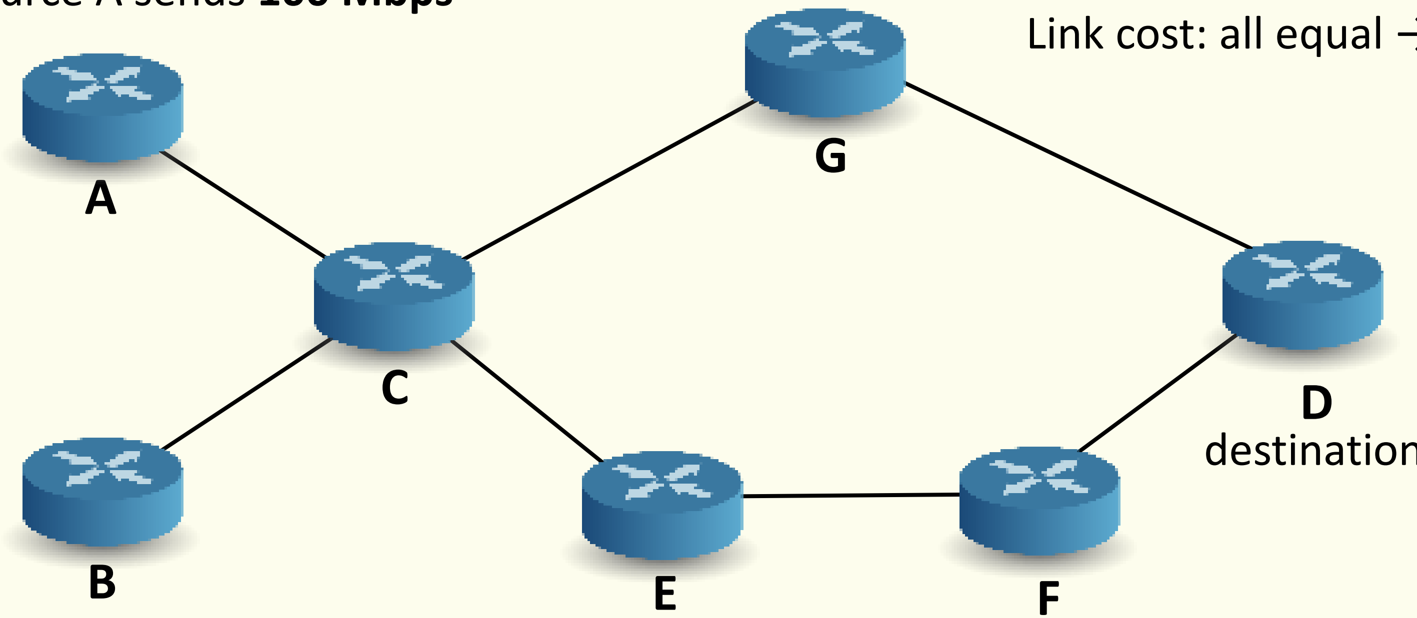
1. **Shortest path → C-G-D is congested**
2. **ECMP → ok, needs manipulating link costs**
3. **What if link E-F has capacity of 50 Mbps?**

Specify **bandwidth requirements** between each source/destination pair, **find a path** that **satisfies** these requirements, **forward** the traffic **along this path**

Application scenario [3]

source A sends **100 Mbps**

All link **150 Mbps**, but **E-F** (**50 Mbps**)
Link cost: all equal → Path cost = #hops

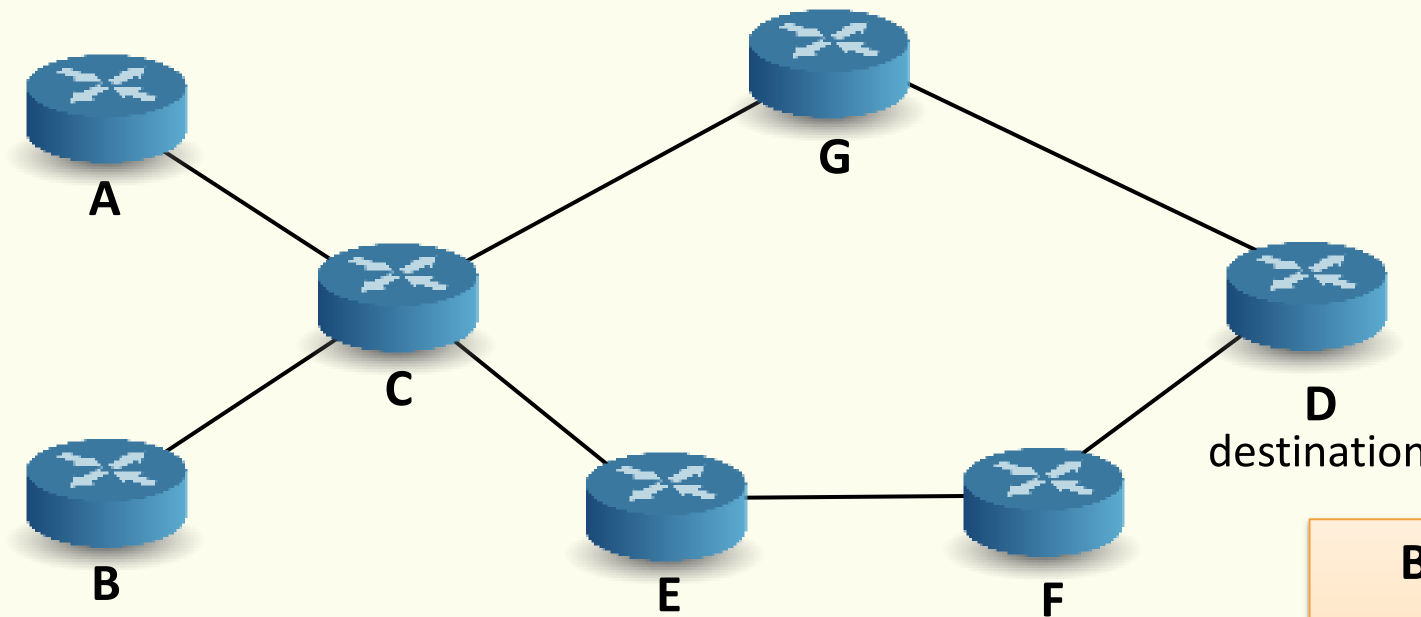


source B sends **40 Mbps**
Customer B buys a service with strict guarantees

1. **Shortest path → ok in normal conditions**
2. **What if link G-D fails?**

find paths between source/destination pairs that **comply with bandwidth constraints**, enforce the **priority of the path** sourced at B over that sourced at A

Requirements for TE



Computing paths that comply with a set of constraints

Enforcing traffic to be forwarded along these paths

By decoupling service from transport, MPLS is fundamental to support TE requirements