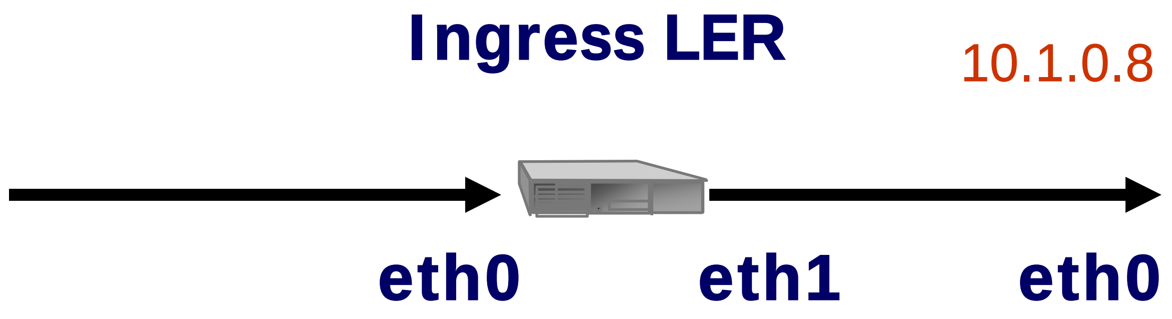
Forwarding Traffic Down Tunnels

Sources:
MPLS Forum, Cisco
V. Alwayn, *Advanced MPLS Design and Implementation*, Cisco Press
E. W. Gray, *MPLS Implementing the Technology*, Addison Wesley
B. Davie and Y. Rekhter, *MPLS Technology and Applications*, Morgan Kaufmann
E. Osborne and A. Simha, *Traffic Engineering with MPLS*, CiscoPress

Traffic Engineering

- TE encompasses many aspects of network performance
  - ✓ Improving the utilization of network resources by distributing traffic evenly across network links
    - ➢ Information distribution
    - ➢ Path calculation and setup
  - ✓ Provisioning of a guaranteed hard QoS
  - ✓ Providing for quick recovery when a node or link fails
- After a tunnel is up, what's the next?

Example on MPLS-Enabled Linux



```
mpls nhlfe add key 0 instructions push gen 1000 nexthop eth1 ip4 10.1.0.8
```

```
ip route add 10.1.0.8/32 via 10.1.0.8 spec_nh 0x8847 0x2
```

(routing table management)

Forwarding Traffic Down a Tunnel Interface

- Three methods can be used:
  - ✓ Static routes
  - ✓ Policy routing
  - ✓ Autoroute

- Also
  - ✓ Load sharing
    - ➢ Main attractive property: unequal-cost load sharing

Forwarding with Static Routes

- Simple
- Configure a route pointing down a tunnel interface
  - Example:
    - Configure a route in regular IP
      - ip route 10.0.0.0 255.0.0.0 eth4
      - Send traffic for 10.0.0.0/8 down the interface eth4
    - Configure a route pointing down a tunnel interface
      - ip route 10.0.0.0 255.0.0.0 **Tunnel0**
      - Send all traffic for 10.0.0.0/8 down Tunnel0

Forwarding with Policy-Based Routing

- Policy-based routing (PBR) is enabled using policy route maps applied to the incoming interface.
- Configure the **policy** and the **tunnel interface**
- Can be used to send specific types of traffic down a tunnel interface **without modifying a router's routing table**
- Example

```
interface Eth0                              // incoming interface
   ip policy route-map foo

route-map foo                          // define the policy
   match …  (e.g. match ip address …)

set interface Tunnel0              // outgoing interface
```

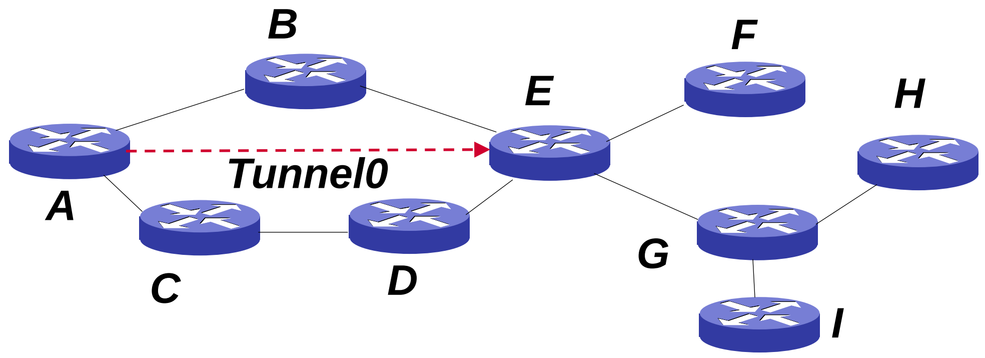
Forwarding with Autoroute

- Most types of interfaces need IGP enabled on them in order to form routing adjacency, learn routes, and build a routing table involving the interfaces.

- How about **enabling IGP on a TE tunnel interface**?
  - ✓ Usually IGP is not run over an MPLS TE tunnel
    - ➢ TE tunnels are unidirectional and thus can never receive any packets.
    - ➢ Don't need it. Because often the full link-state topology is already available.
    - ➢ Better flexibility and scalability for TE

- Instead, need to inform the tunnel headend to treat this interface like the tunnel is directly connected to the tunnel tail, and send any packets down the tunnel that are destined for either the tunnel's tail or anything behind that tunnel tail.

Example



All links have a cost of 10. Before TE tunnels are built, router A's routing table:

Node	Next Hop	Cost
A	self	0
B	B	10
C	C	10
D	C	20
E	B	20
F	B	30
G	B	30
H	B	40
I	B	40

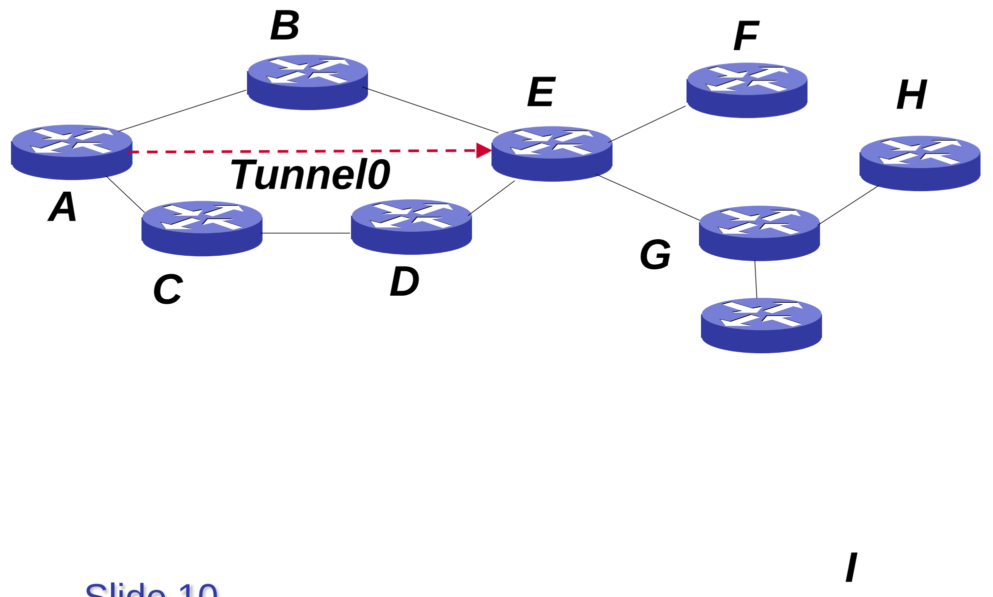
Example (cont'd)

- After the tunnel from router A to router E is up, need to map traffic to router E to Tunnel0.
- Configure a static route for router G pointing down the tunnel:
  - ✓ ip route *router G's RID* 255.255.255.255 Tunnel0
  - ✓ Router A's routing table for router G will change to
    - ➢ G Tunnel0 0 // cost is always 0 for static routes
- Policy routing is even simpler, cause it doesn't change the routing table. Packet forwarding decisions are made based on the configured policy and interface, not the routing table.

Example (cont’d – autoroute)

- Autoroute tells a router to build its routing table so that anything behind the TE tunnel tailend is routed down that tunnel.
- How does it work?
  - ✓ IGP runs SPF
  - ✓ If a node is either tunnel tail or behind the tunnel tailend, the TE tunnel will be added to that node instead of the IGP path in the routing table.

Node	Next Hop	Cost
A	self	0
B	B	10
C	C	10
D	C	20
E	Tunnel0	20
F	Tunnel0	30
G	TUnnel0	30
H	Tunnel0	40
I	Tunnel0	40



More on Autoroute

- With autoroute enabled, the tunnel tail is always routed through the tunnel.
- The tunnel tail can be reached only through the tunnel interface because of the replacement of the physical next hop with the tunnel interface during IGP SPF.
- Node behind the tunnel tail can generally be reached through the tunnel, although you can get to the a node through both an IGP route and the TE tunnel route in some cases.
- How about load sharing?

Load Sharing

- In terms of paths:
  - Load sharing between a TE tunnel path and an IGP path
  - Load sharing between two or multiple TE tunnels
  - Changing the metric used for the TE tunnel
- In terms of cost:
  - Equal-cost load sharing
    - Per-flow/per-destinaiton/per-src-dest load sharing: Packet’s source & destination addresses
      - Can be done with traditional IP
      - For MPLS, how to find out src/dest addresses in the label header?
    - Per-packet: round-robin, need packet reordering
  - Unequal-cost load sharing
    - With IP: Need to guarantee loop-free
    - MPLS is useful

Load Sharing – Equal Cost Multipath

- Supported in OSPFv2
- Principle
  - ✓ SPF distributes the network topology info to all routers
  - ✓ Based on the topology, each router computes the routes towards all destinations
  - ✓ If a router finds multiple equal cost paths to the same destination, it stores those paths in the routing table. It then balances its traffic over these paths
  - ✓ Load sharing is done at the router level – local sharing
    - ➢ Loops will not occur if the network is stable

Limitations of ECMP

- Drawbacks:
  - ✓ Load sharing/balancing works for ***exactly equal*** costs paths, but few paths are exactly equal
  - ✓ Local decision made by individual router without knowing the actual load of the network and coordinating with other routers
    - ➢ Traffic may be balanced to the same destination, but TE at the network level generally not realized
    - ➢ Example
  - ✓ If a link cost is changed, other parts will often be affected in unanticipated ways
- How can it be improved?
  - ✓ Support almost equal costs paths
  - ✓ Router should know the current work load
  - ✓ Need to be careful to avoid loops

Load Sharing for Tunnels – Equal Cost

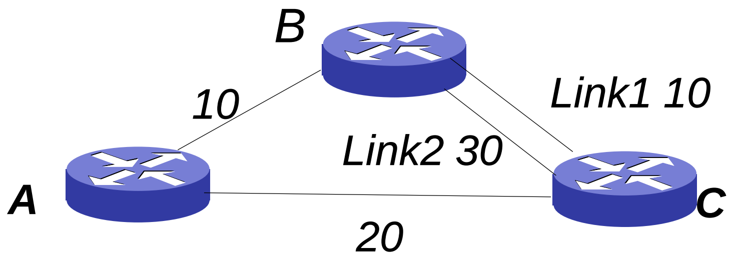
- Between the TE tunnel and the IGP path
  - ✓ Never load share between an IGP route and a TE route for the tunnel tail
    - ➢ Lose the ability to explicitly route traffic down a tunnel that takes a suboptimal path.
    - ➢ Much harder to traffic-engineer the network, cause don't have the complete control over all the traffic.
- Between two or more TE tunnels
  - ✓ Build > 1 tunnels to the tail for load sharing
- To nodes behind the tunnel tail
  - ✓ Rule is the same for equal-cost forwarding with IP or MPLS

Load Sharing to Nodes Behind the Tunnel Tail

- Sometimes you may want to share between a TE tunnel path and an IGP path to get to the destinations ***downstream*** of the tunnel tail.
- Example
- Load sharing is equal-cost
  - ✓ Not flexible, has to be equal cost
  - ✓ Difficult to guarantee loop-free

Unequal-Cost Load Sharing

- Difficult to do while guaranteeing a loop-free topology with IP



Assume that unequal-cost paths are calculated based on path cost, with the amount of traffic forwarded down a particular path being *inversely proportional to the cost* of the path.

How many paths exist from A to C?

- A->C, cost 20
- A->B(link1)->C, cost 20
- A->B(link2)->C), cost 40

So, traffic is shared between these three paths in a 40:40:20 ratio.

Unequal-Cost Load Sharing

- What are router A and B's routing tables?
- If router A has 100Mbqs of traffic to send to router C. What will happen?
  - ✓ Loop – router A to router B to router A …
- Reason: router A couldn't tell router B what to do with the packet.
- Router A needs to identify a path that traffic needs to follow. Router A needs to be able to tell router B which traffic should be forwarded across link1 and which should go across link2 – some sort of label is needed to indicate the direction in which the traffic should flow.
- MPLE TE is beneficial to **unequal-cost load sharing**.

How Unequal-Cost Load Sharing Works?

- MPLS-TE load sharing works between multiple tunnels to the same destination. Two parts needed:
  - ✓ Setting up the load-sharing ratios
    - ➢ Bandwidth
    - ➢ Manual configuration of load-share value
  - ✓ Adding these ratios to the forwarding table
  - ✓ Example
- Keys for UCLS:
  - ✓ All paths to a destination must be TE tunnels
  - ✓ All paths must have a nonzero bandwidth (or nonzero load-share metric)

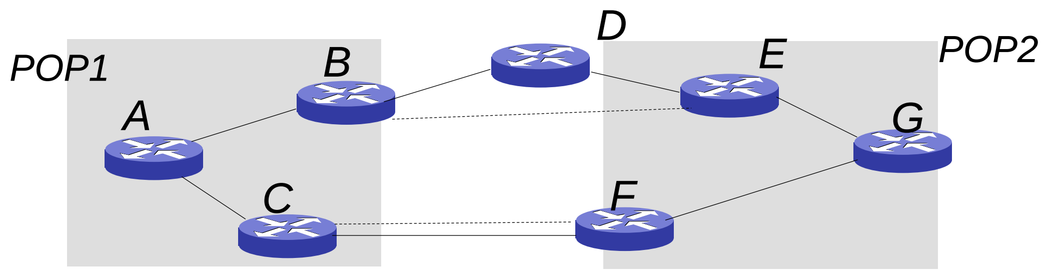
Changing TE Tunnel Metric

- Changing TE tunnel metric influences **only the tunnel headend**. Other routers don't know about the change, unless the change is explicitly advertised.
- Several mechanisms:
  - ✓ Autoroute
  - ✓ Absolute
  - ✓ Relative
- How does it work?
  - ✓ Example
  - ✓ Key: metrics are changed **after** SPF run is complete.
    - ➢ Example
    - ➢ Changing the tunnel metric doesn't influence what routes are installed through the tunnel, only the cost to get to those routes.
    - ➢ It may not work as expected. Example

Forwarding Adjacency

- Sometimes, changing the metric sometimes isn't enough for TE.
- TE tunnels are not advertised in IGP, i.e., if you change the metric on a TE tunnel, other routers will not see it and can't make use of it.
- One solution to support TE is to build two TE tunnels for each pair of source and destination.
- Another issue: extending TE tunnels all the way to the edge works fine in a small network, but not suitable to large networks. Why?
- To scale better, we can move the TE tunnels up one level in the network hierarchy, toward the core.

Forwarding Adjacency



- If we want to send A->G traffic across both A->C->F->G and A->B->D->E->G, we can build two tunnels toward the core (to reduce the number of tunnels).
- Now, we have two tunnels, so the problem is solved, right?
- Unfortunately, router A doesn't know about those TE tunnels. So, **router A makes its SPF decision based on the IPG metrics alone**. That means traffic is sent to router C.
- How to solve it? (Currently, only IS-IS supports forwarding adjacency.)
  - ✓ Need a way to **advertise the TE tunnels into the IGP** so that router A and other routers can see them as **regular links**. (It's not a link that TE tunnels can be signalled across, but it is available for regular IGP traffic.)
  - ✓

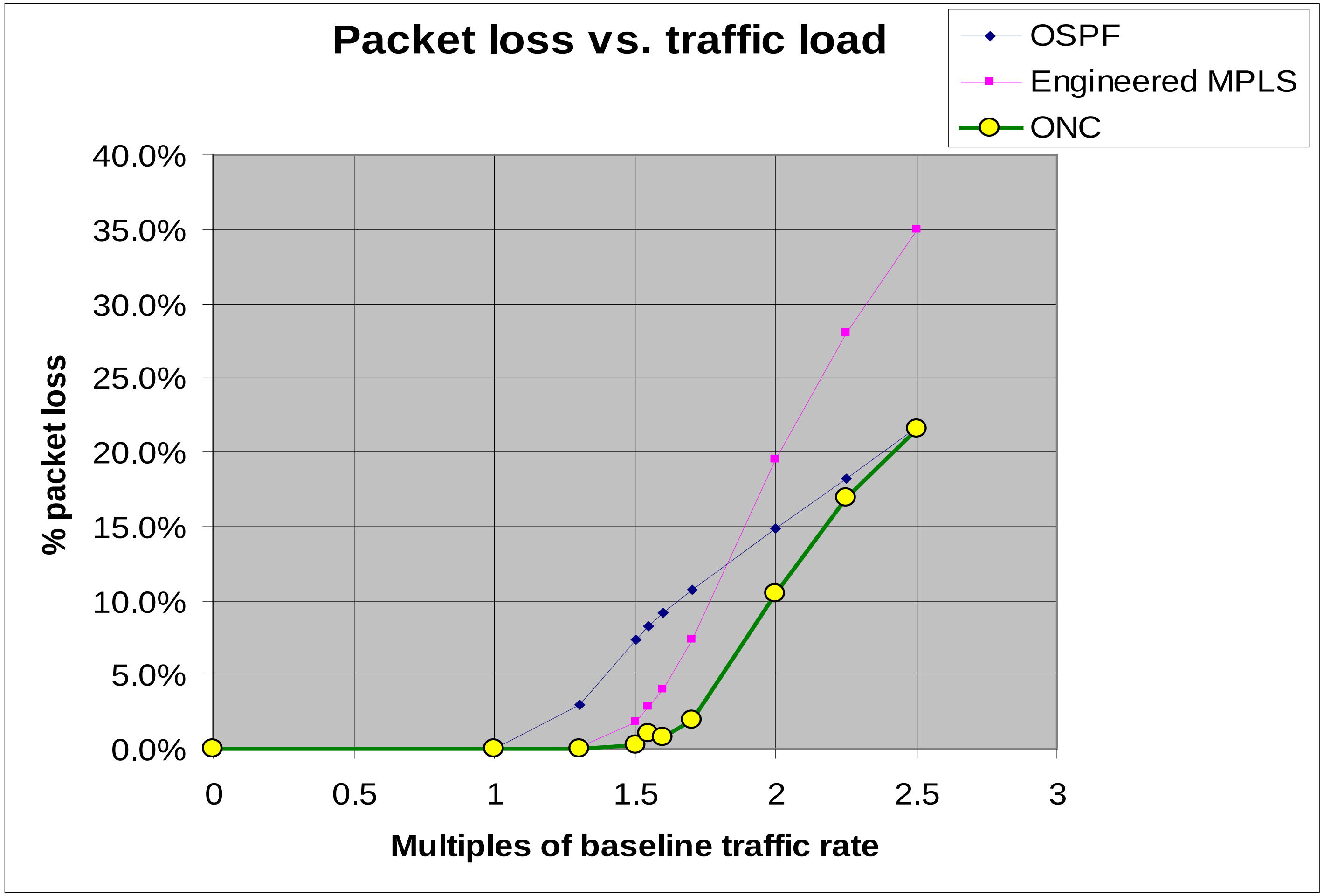
    Example:

    ```
      Interface Tunnel1
      …
      tunnel mpls traffic-eng forwarding-adjacency
      Is-is metric 9 level-2
    ```
- Forwarding adjacency is **bi-directional** and is **treated as a IGP link**, not as a TE link. Tunnel headend and tail must be in the same area.

Automatic Bandwidth Adjustment

- MPLE TE tunnels can be configured to reserve bandwidth. So far, reservations require manual work. What if traffic patterns change?
  - ✓ Offline tool to calculate how much bandwidth is needed for each tunnel, calculate paths, and send new configurations to routers.
    - May be more efficient in bandwidth usage
    - Lots of work
  - ✓ Online automatic bandwidth adjustment
    - Concept is simple
    - Monitor traffic for each tunnel and periodically, the headend/ingress looks at the tunnel utilization
    - Lots of details:
      - application frequency (A), tunnel bandwidth (B), collection frequency (C), highest collected bandwidth (H) or average, delta (D=H-B): What is the relationship between A, B, C, H, and D?
      - Where to put the tunnels, when to change, how much to change, competitions of bandwidth between tunnels, available resources, congestion management …

An Example of Automatic Bandwidth Adjustment



MPLS TE and QoS

DiffServ Aware TE
Explicit Congestion Notification

DiffServ-Aware TE (DS-TE)

- DiffServ with MPLS packets is basically the same thing as with IP packets.
  - ✓ EXP setting vs. IP Precedence setting
- Why MPLS?
  - ✓ Make a headend resource-aware, so that it can intelligently pick paths through the network for its traffic to take.
  - ✓ Can steer IP traffic away from the IGP shortest path or congested links.
- However, we can’t steer traffic per QoS.
  - ✓ If there is traffic destined for a router, all that traffic follows the same path (per-src-dest), regardless of the DSCP/EXP settings.
  - ✓ Routing is limited by the routing table and how it decides to forward traffic.
  - ✓ As it’s been discussed so far, TE doesn’t do admission control on a per-QoS class basis.

DS-TE

- What's the problem?
  - ✓ If there is a congested link at a downstream node along the forwarding path, the congestion knowledge is localized at the downstream node and is not propagated back to the edge devices that send traffic down that path.
    - ➢ Gold traffic might be dropped.
  - ✓ Edges continue to send traffic to the same downstream router.
    - ➢ Gold traffic might continue to be dropped.
- Need per-class admission control.
- Combine DiffServ and TE (DS-TE).

DS-TE (more)

- TE offers call admission control in addition to the PHB offered by DiffServ.
  - ✓ If more traffic is sent down a certain path than there is available bandwidth, queue higher-priority traffic ahead of low-priority traffic.
- How about the possible contention between different high-priority traffic streams?
  - ✓ Two voice pipes from customers, both with a low-latency requirement, if you forward both streams down the same congested paths, both streams might be affected.
- DS-TE allows to advertise more than one pool of available resources for a given link – a global pool and subpools.

Subpools

- A subpool is a subset of link bandwidth that is available for a specific purpose.
  - ✓ A pool with which you can advertise resources for a separate queue.
  - ✓ Currently, DS-TE allows to advertise one subpool.
  - ✓ Recommended for low-latency queue
  - ✓ The actual queuing behavior at every hop is still controlled by the regular DiffServ mechanisms such as LLQ.
  - ✓ DS-TE has the ability to reserve queue bandwidth, rather than just link bandwidth in the control plane.
  - ✓ Let you build TE-LSPs that specifically reserve subpool bandwidth and carry only the specified traffic (e.g. LLQ).

How to Make Use of Subpool?

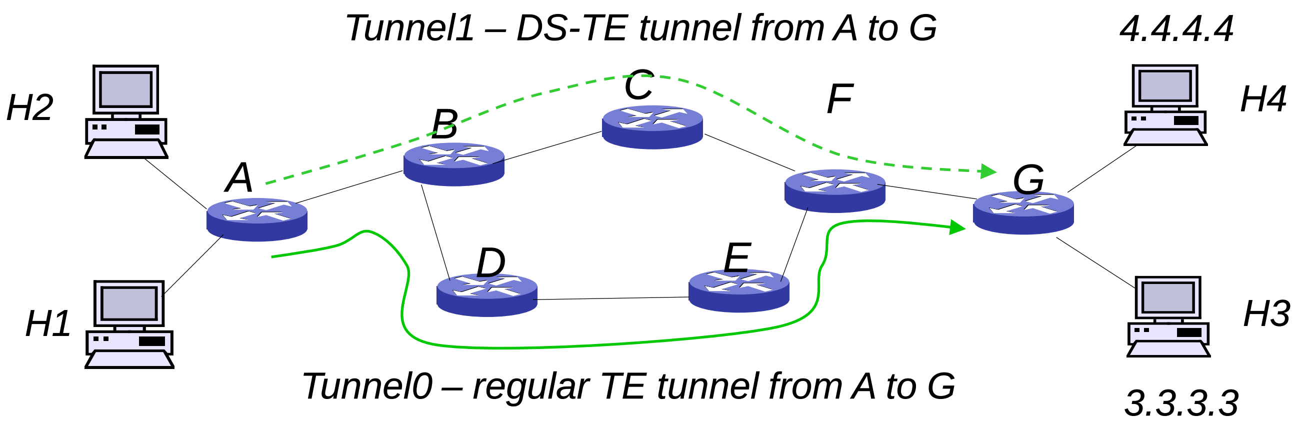
- Five steps involved:
  - ✓ Advertise a per-link subpool & its bandwidth availability
    - ➢ ip rsvp bandwidth 150000 sub-pool 45000
  - ✓ Specify per-link scheduling, LLQ

```
Class-map match-all voice
              match mpls experimental 5
policy-map llq
              class voice
                                    priority percent 30
interface POS3/0
              service-policy output llq
```

  - ✓ Tell the headEnd subpool bandwidth requirement for path calculation and bandwidth reservation
    - ➢ tunnel mpls traffic-eng bandwidth sub-pool *kbps*
  - ✓ Perform headend tunnel admission control
    - ➢ Make sure that the only traffic to enter the DS-TE tunnel is traffic that belongs there
  - ✓ Enable tunnel preemption
- Example

Forwarding DS-TE Traffic Down a Tunnel

- Forwarding DS-TE traffic down a tunnel
  - ✓ Static routes
  - ✓ Policy-based routing
  - ✓ Autoroute
    - ➢ Easiest. Requires only one command on the headend, and all the traffic destined for or behind the tail is sent down the tunnel.
    - ➢ But, if have both TE and DS-TE tunnels to the same destination, it may not do what you want.


Tunnel1 – DS-TE tunnel from A to G
4.4.4.4
H2
C
F
H4
B
A
G
D
E
H1
H3
Tunnel0 – regular TE tunnel from A to G
3.3.3.3

Forwarding DS-TE Traffic Down a Tunnel

- H2 has voice traffic destined for H4, and H1 has regular IP traffic destined for H3.
- If enable autoroute on both tunnels, what will happen?
- Need to use static route, so that H3 is only reachable over Tunnel0
  - ✓ Example: ip route 3.3.3.3 255.255.255.255 Tunnel0
- What if there are many hosts that receive voice traffic?
  - ✓ Static routes are reasonable for a small-scale problem
  - ✓ Need to aggregate devices into subnets.

Explicit Congestion Notification

- ECN is classified as an “experimental” protocol by the IETF, has been specified but not standardized until some experience is gained with it.
- Congestion control in today’s IP networks (implicit)
  - ✓ Congestion avoidance mechanisms of TCP: packet losses are an indication of congestion.
  - ✓ TCP senders reduce their sending rates when they experience packet loss and slowly increase rates during periods when no packets have been lost.
- Drawbacks:
  - ✓ Dropped packets need to be retransmitted and will arrive later, degradation of the response time and quality
  - ✓ Dropped packets still consume resources, better not to send the packet at all.

ECN Overview

- ECN introduces a way to explicitly signal congestion to the sender without dropping a packet.
- How to know congestion?
  - ✓ Need some form of queue management such as RED to monitor congestion rather than just dropping packets when the queue becomes full.
- What to do?
  - ✓ A router sets a bit (congestion experienced CE) in a packet header when it detects congestion, and then forwards the packet rather than dropping it.
- How does the sender know it and respond?
  - ✓ When a packet with the CE bit arrives at its destination, the receiver sends a signal back to the sender to reduce rate
  - ✓ The way the sender responds to it is dependent on end-to-end protocol used.
  - ✓ For TCP, ECN-echo bit in the TCP header is set and sent to the sender via ACK packet. When the sender receives it, it responds exactly as if a packet had bee dropped.
- Compatibility and deployment issue
  - ✓ Some routers are ECN-capable; some, non-ECN-capable. Sender may not reduce traffic.
  - ✓ ECN defines two new bits (2 unused bits in the ToS byte) to be carried in the IP header: CE bit and ECT bit (ECN-capable transport). If congestion:
    - ➢ If ECT bit is set, set CE bit
    - ➢ If ECT bit is not set, drop packets

MPLS Support of ECN

- ECN should be supported in the MPLS header. Where in the MPLS header?
  - ✓ Use one bit in the Exp field
- Is it enough? How to represent ECN states?
  - ✓ Not ECN capable
  - ✓ ECN capable AND not CE
  - ✓ ECN capable AND CE
- Rules for setting the ECN bit in the MPLS header
  - ✓ When we add the MPLS header to IP header
    - ➢ 0 ECN capable AND not CE
    - ➢ 1 Not ECN capable OR CE
  - ✓ When we remove the MPLS header

IP ECT bit on input	MPLS ECN bit value	IP ECN bits on output
Not ECN capable (ECT=0)	1	ECT=0, CE=0
ECN capable (ECT=1)	0	ECT=1,CE=0
ECN capable (ECT=1)	1	ECT=1,CE=1

- Why wait until it reaches the destination and send a notification? BECN vs. FECN