TE tunnel basics

APNIC

MPLS Traffic Engineering

How it works ? This can be broken down into three pieces:

- Information distribution
  - How routers know what the network looks like and what resources are available
- Path calculation and setup
  - How routers decide to build TE tunnels, and how these TE tunnels are actually built and maintained
- Forwarding traffic down a tunnel
  - After a tunnel is built, how is it used?

What Information Is Distributed?

Information distribution is again broken down into three pieces:

- What information is distributed and how you configure it?
- When information is distributed and how you control when flooding takes place?
- How information is distributed (protocol-specific details)?

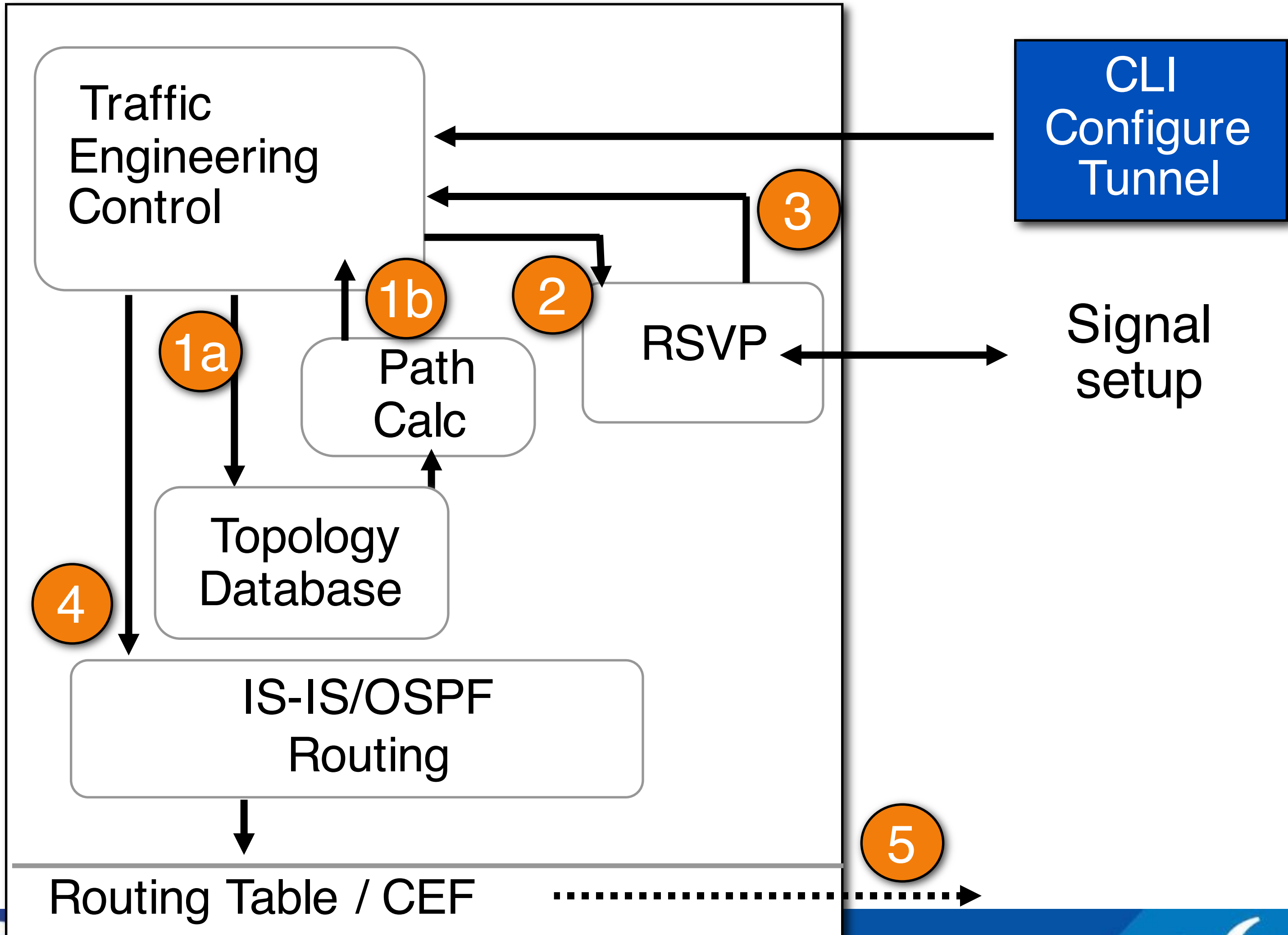
How MPLS TE works

- By using OSPF or IS-IS to distribute information about available resources in your network
- Three major pieces of information are distributed:
  - **Available bandwidth information per interface**, broken out by priority to allow some tunnels to preempt others
  - **Attribute flags per interface**
  - **Administrative weight** per interface

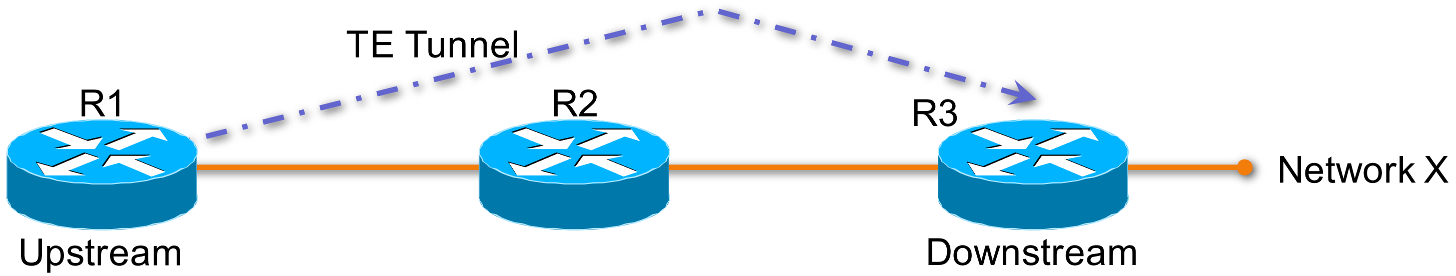
How MPLS TE works

- Tunnel attributes:
  - Bandwidth
  - Priority
  - Metric selection
  - Affinity

Tunnel Setup



A Terminology Slide—Head, Tail, LSP, etc.



- Head-End is a router on which a TE tunnel is configured (R1)
- Tail-End is the router on which TE tunnel terminates (R3)
- Mid-point is a router thru which the TE tunnel passes (R2)
- LSP is the Label Switched Path taken by the TE tunnel, here R1-R2-R3
- Downstream router is a router closer to the tunnel tail
- Upstream router is farther from the tunnel tail (so R2 is upstream to R3's downstream, R1 is upstream from R2's downstream)

Tunnel Attributes

- Tunnel attributes are characteristics the tunnel requires to have on the links along the LSP.
- Configured at the head-end of the trunk
- These are:
  - Bandwidth
  - Priority
  - Affinity

```
interface Tunnel0
  tunnel mpls traffic-eng bandwidth Kbps
  tunnel mpls traffic-eng priority pri [hold-pri]
  tunnel mpls traffic-eng affinity properties [mask]
```

Tunnel Bandwidth

```
tunnel mpls traffic-eng bandwidth Kbps
```

- Bandwidth required by the tunnel across the network
- Not a mandatory command
- If not configured, tunnel is requested with zero bandwidth.

Priority

```
tunnel mpls traffic-eng <S> {H}
```

- Configured on tunnel interface
- S = setup priority (0–7)
- H = holding priority (0–7)
- Lower number means higher priority

Priority

- Setup priority of new tunnel on a link is compared to the hold priority of an existing tunnel
- New tunnel with better setup priority will force preemption of already established tunnel with lower holding priority
- Preempted tunnel will be torn down and will experience traffic black holing. It will have to be re-signaled
- Recommended that S=H; if a tunnel can setup at priority “X”, then it should be able to hold at priority “X” too!
- Configuring S > H is illegal; tunnel will most likely be preempted
- Default is S = 7, H = 7

Tunnel Affinity

- Tunnel is characterized by a
  - Tunnel Affinity: 32-bit resource-class affinity
  - Tunnel Mask: 32-bit resource-class mask (0= don't care, 1= care)
- – Link is characterized by a 32-bit resource-class attribute string called Link Affinity
- – Default-value of tunnel/link bits is 0
- – Default value of the tunnel mask = 0x0000FFFF

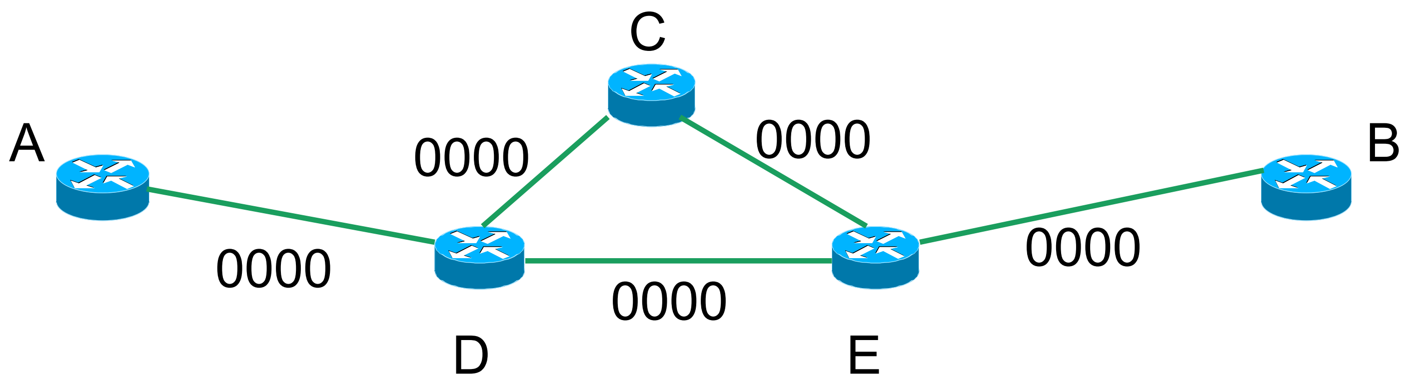
Tunnel Affinity (Cont.)

- Affinity helps select which tunnels will go over which links
- A network with OC-12 and Satellite links will use affinities to prevent tunnels with VoIP traffic from taking the satellite links

Tunnel can only go over a link if

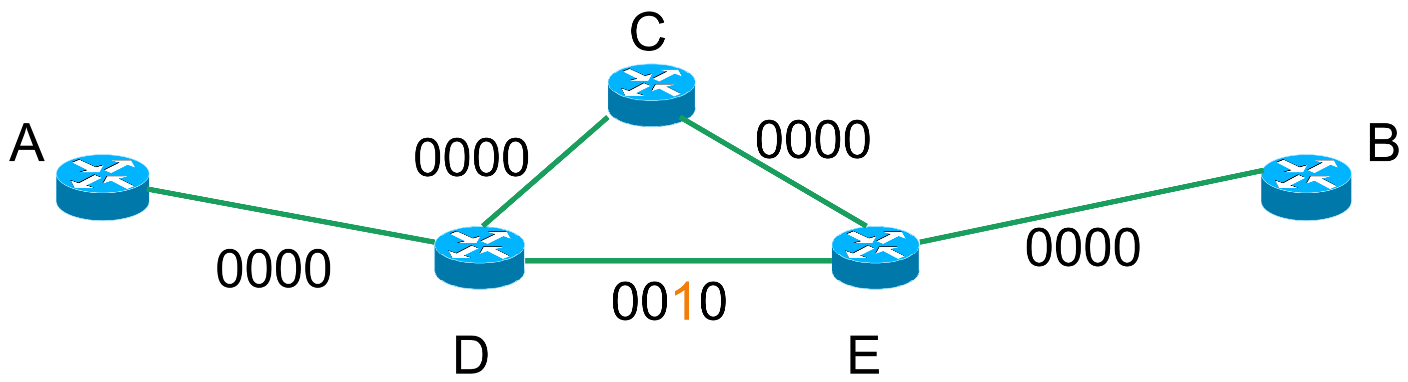
$$\boxed{\text{(Tunnel Mask) AND (Link Affinity) == Tunnel Affinity}}$$

Example0: 4-bit string, default



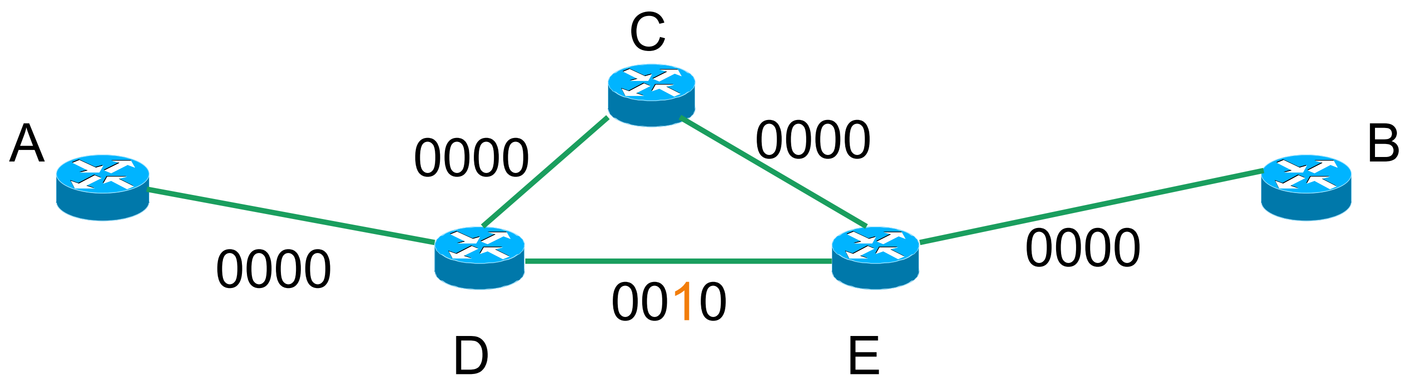
- Trunk A to B:
  - tunnel = 0000, t-mask = 0011
- ADEB and ADCEB are possible

Example1a: 4-bit string



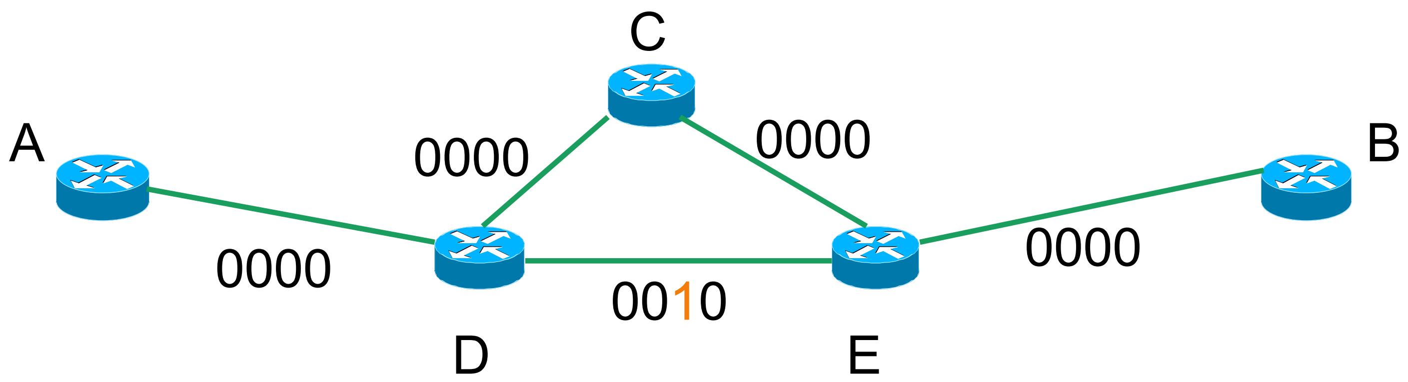
- Setting a link bit in the lower half drives all tunnels off the link, except those specially configured
- Trunk A to B:
  - tunnel = 0000, t-mask = 0011
- Only ADCEB is possible

Example1b: 4-bit string



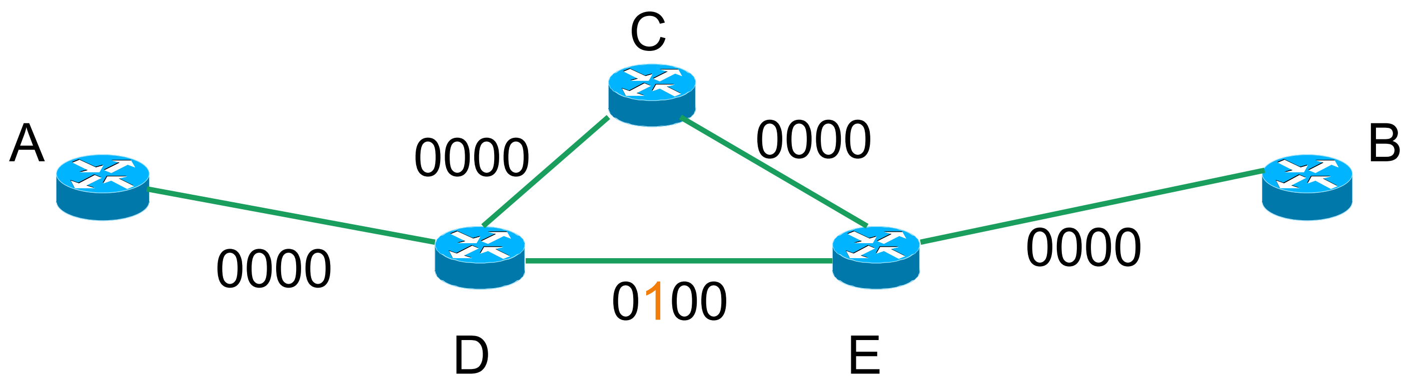
- A specific tunnel can then be configured to allow such links by clearing the bit in its affinity attribute mask
- Trunk A to B:
  - tunnel = 0000, t-mask = 0001
- Again, ADEB and ADCEB are possible

Example1c: 4-bit string



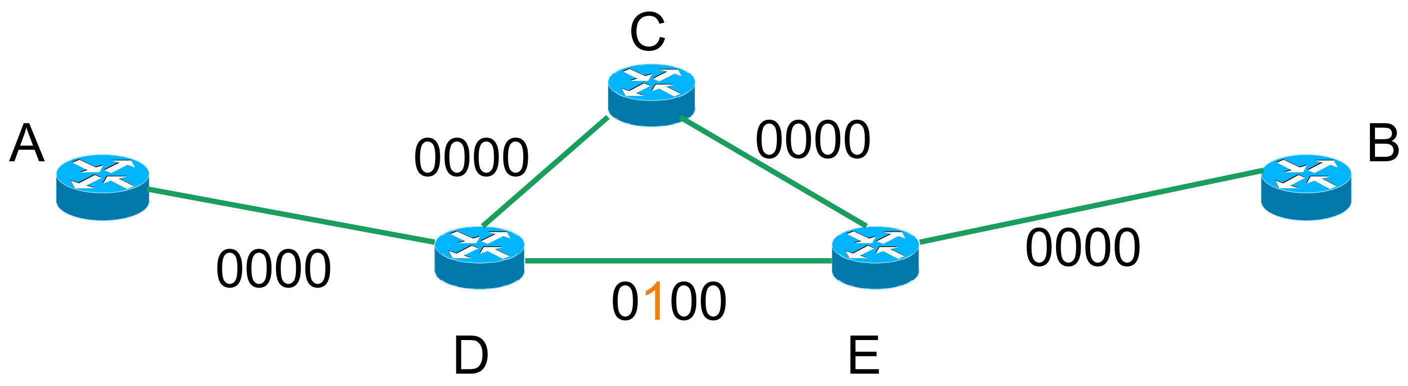
- A specific tunnel can be restricted to only such links by instead turning on the bit in its affinity attribute bits
- Trunk A to B:
  - tunnel = 0010, t-mask = 0011
- No path is possible

Example2a: 4-bit string



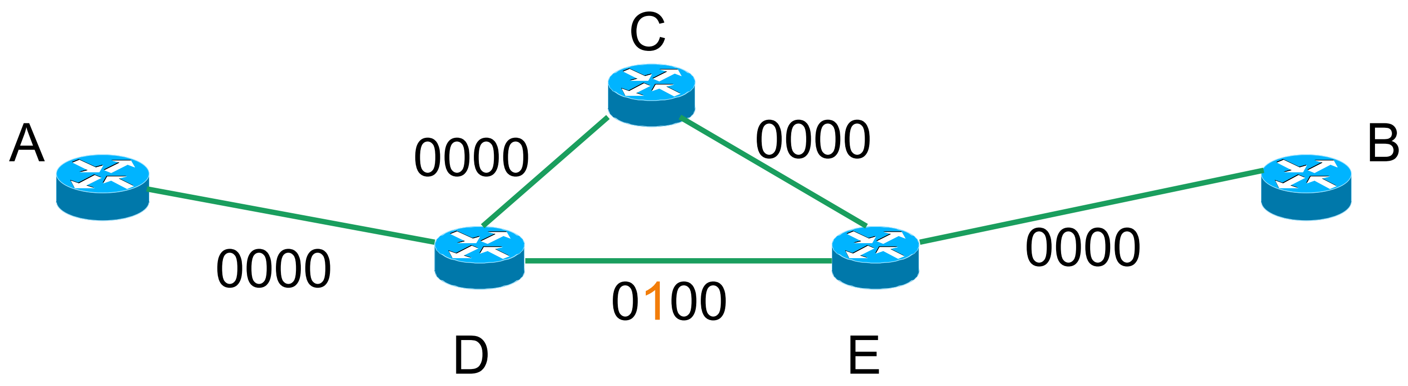
- Setting a link bit in the upper half drives has no immediate effect
- Trunk A to B:
  - tunnel = 0000, t-mask = 0011
- ADEB and ADCEB are both possible

Example2b: 4-bit string



- A specific tunnel can be driven off the link by setting the bit in its mask
- Trunk A to B:
  - tunnel = 0000, t-mask = 0111
- Only ADCEB is possible

Example2c: 4-bit string



- A specific tunnel can be restricted to only such links
- Trunk A to B:
  – tunnel = 0100, t-mask = 0111
- No path is possible

Tunnel Path Selection

- Tunnel has two path options
  1. Dynamic
  2. Explicit
- Path is a set of next-hop addresses (physical or loopbacks) to destination
- This set of next-hops is called Explicit Route Address (ERO)

Dynamic Path Option

```
tunnel mpls traffic-eng path-option <prio>
dynamic
```

- dynamic = router calculates path using TE topology database
- Router will take best IGP path that meets BW requirements

Explicit Path Option

```
tunnel mpls traffic-eng path-option
<prio> explicit <id|name> [ID|NAME]>
```

- explicit = take specified path

Explicit Path Option (Cont.)

```
ip explicit-path <id|name> [ID|NAME]
     next-address 192.168.1.1
     next-address 192.168.2.1 {loose}
     . . .
```

- explicit = take specified path
- Router sets up path you specify
- Strict source-routing of IP traffic
- Each hop is a physical interface or loop back

MPLS-TE: Link attributes, IGP enhancements, CSPF

APNIC

Agenda

- Link Attributes
- Information flooding
- IGP Enhancements for MPLS-TE
- Path Computation (C-SPF)

Link Attributes

- Link attributes
  - Bandwidth per priority (0-7)
  - Link Affinity
  - TE-specific link metric

Bandwidth

```
ip rsvp bandwidth <x> <y>
```

- Per-physical-interface command
- X = amount of reservable BW, in K
- Y = not used by MPLS-TE

Link Affinity

```
mpls traffic-eng attribute-flags <0x0-0xFFFFFFFF>
```

- Per-physical-interface command

Administrative Weight

```
mpls traffic-eng administrative-
          weight <X>
```

- Per-physical-interface command
- X = 0–4,294,967,295
- Gives a metric that be considered for use instead of the IGP metric
- This can be used as a per-tunnel delay-sensitive metric for doing VoIP TE
- By default TE metric is used. However, when no TE metric is configured,
  - IGP metric => TE metric

Information Distribution

- TE LSPs can (optionally) reserve bandwidth across the network
- Reserving bandwidth is one of the ways to find more optimal paths to a destination
- This is a control-plane reservation only
- Need to flood available bandwidth information across the network
- IGP extensions flood this information
  - OSPF uses Type 10 (area-local) Opaque LSAs
  - ISIS uses new TLVs

Information Distribution

- A link-state protocol has to be used as the IGP (IS-IS or OSPF)
- A Link-state protocol is not a requirement for other MPLS applications (e.g. VPNs)

Need for a Link-State Protocol

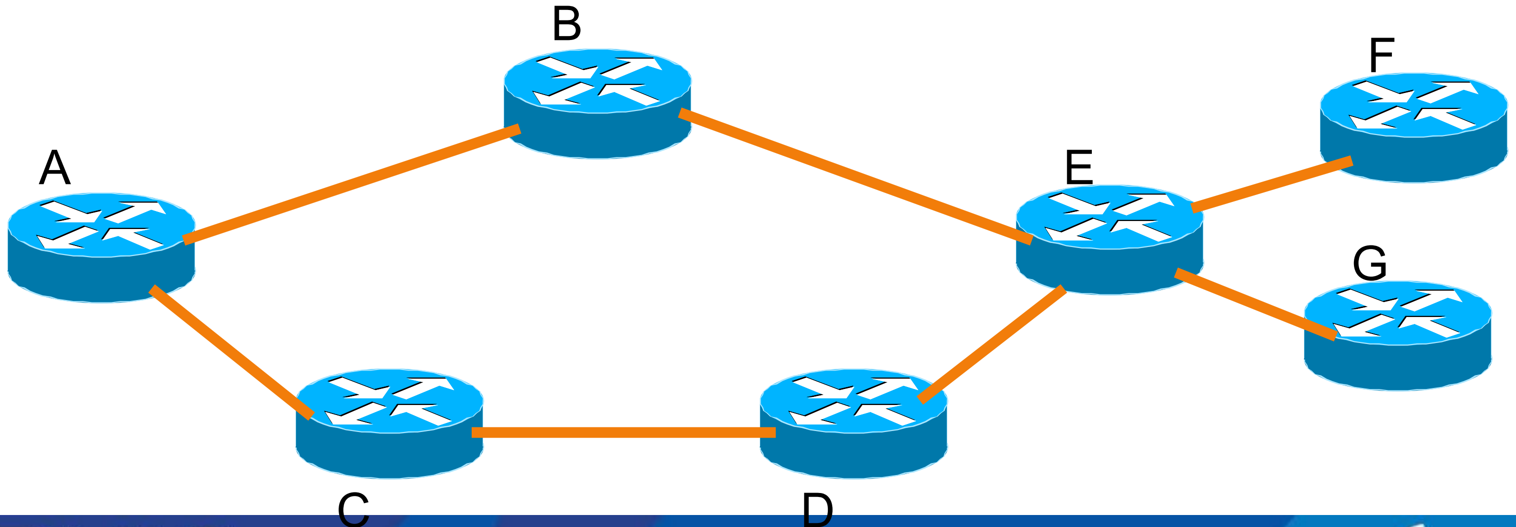
Why is a link-state protocol required?

- Path is computed at the source
- Source needs entire picture (topology) of the network to make routing decision
- Only link-state protocols flood link information to build a complete network topology

Need for a Link-State Protocol

Consider the following network:

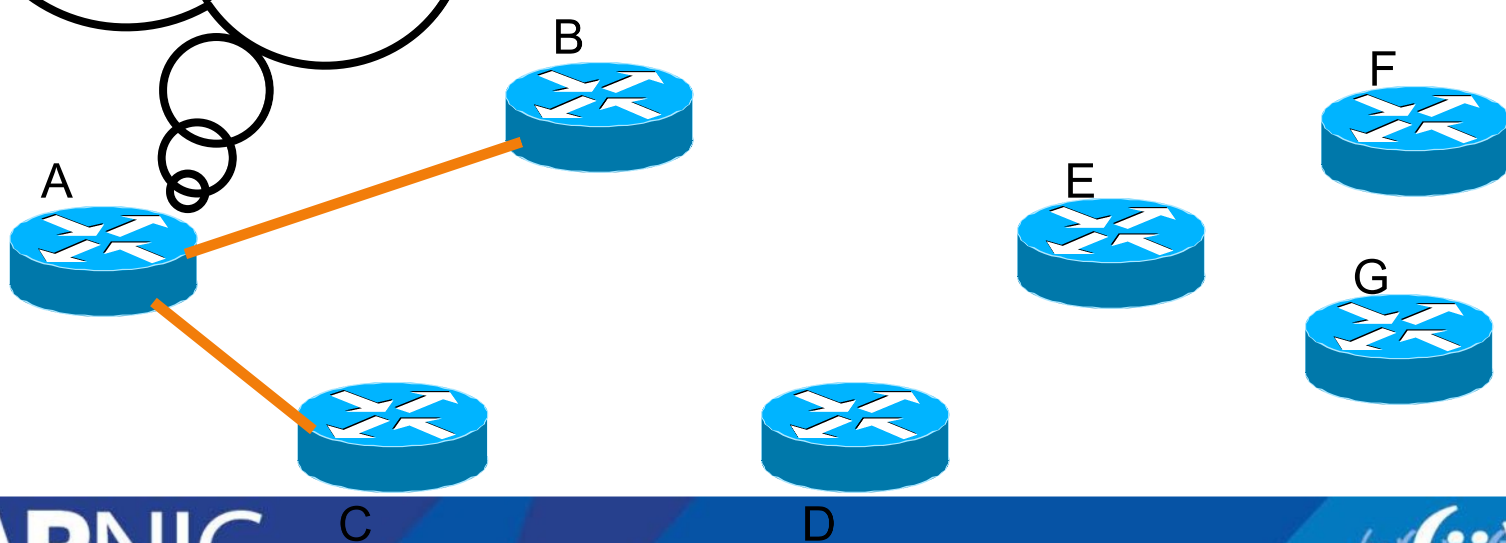
- All links have a cost of 10
- Path from “A” to “E” is A->B->E, cost 20
- All traffic from “A” to {E,F,G} goes A->B->E



What a Distance Vector Protocol Sees

Node	Next-Hop	Cost
B	B	10
C	C	10
D	C	20
E	B	20
F	B	30
G	B	30

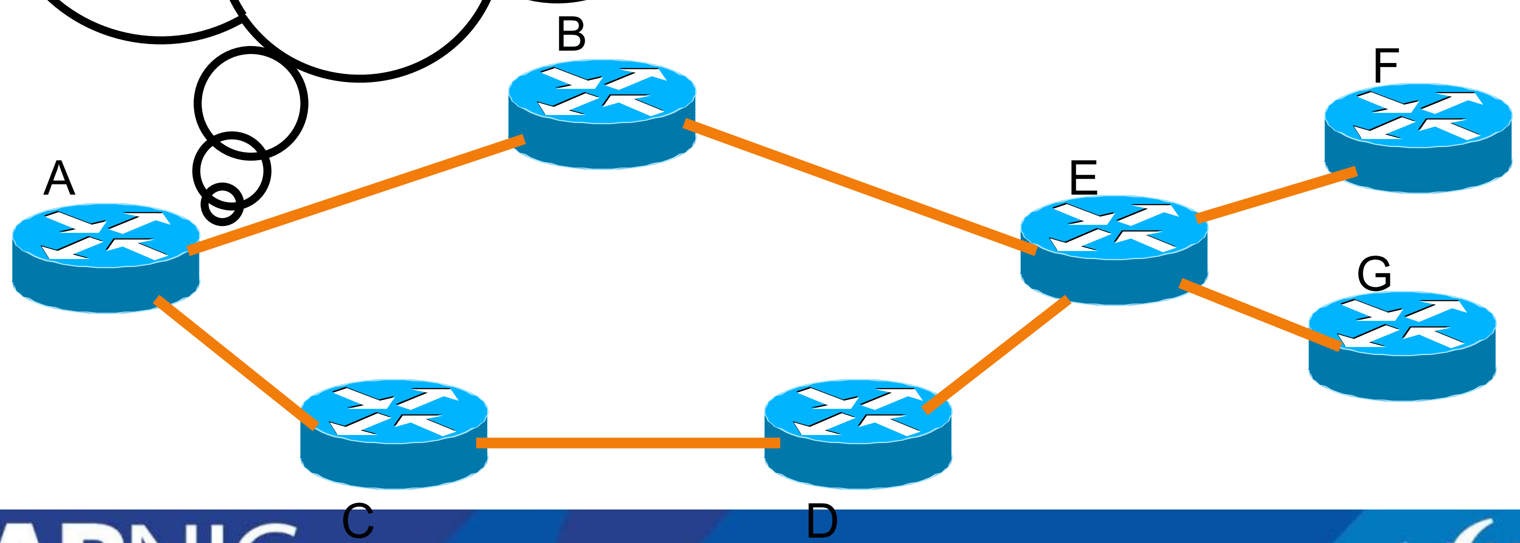
- “A” doesn’t see all the links
- “A” *knows* about the shortest path
- Protocol limitation by design



What a Link-State Protocol Sees

Node	Next-Hop	Cost
B	B	10
C	C	10
D	C	20
E	B	20
F	B	30
G	B	30

- “A” sees all links
- “A” *computes* the shortest path



Link-State Protocol Extensions/ IGP Flooding

- TE finds paths other than shortest-cost
- To do this, TE must have more info than just per-link cost
- OSPF and IS-IS have been extended to carry additional information
  - Physical bandwidth
  - RSVP configured bandwidth
  - RSVP Available bandwidth
  - Link TE metric
  - Link affinity

OSPF Extensions

- OSPF
  - Uses Type 10 (Opaque Area-Local) LSAs
  - See draft-katz-yeung-ospf-traffic

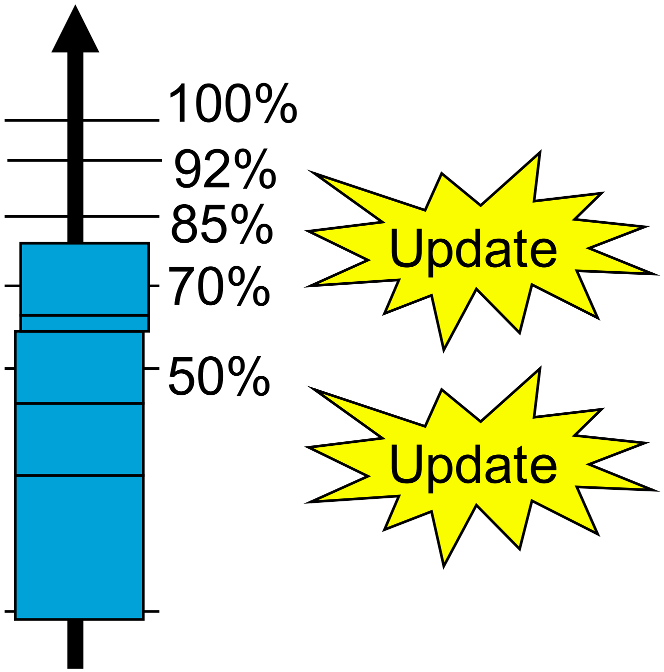
IS-IS Extensions

- IS-IS
  - Uses Type 22 TLVs
  - See draft-ietf-isis-traffic

Information Distribution

- Dynamics of ISIS and OSPF are unchanged
  - Periodic flooding
  - Hold-down timer to constrain the frequency of advertisements
- Current constraint information sent when IGP decides to re-flood
- TE admission control requests re-flooding on significant changes
  - *significant* is determined by a configurable set of thresholds
  - On link configuration changes
  - On link state changes
  - On LSP Setup failure
  - TE refresh timer expires (180 seconds default)

Significant Change



- Each time a threshold is crossed, an update is sent
- Denser population as utilization increases
- Different thresholds for UP and Down

```
router#sh mpls traffic-eng link bandwidth-allocation pos4/0
 .............<snip>...................
     Up Thresholds:          15 30 45 60 75 80 85 90 95 96 97 98 99 100 (default)
    Down Thresholds:        100 99 98 97 96 95 90 85 80 75 60 45 30 15 (default)
.............<snip>....................
```

Constrained-based Path Computation (C-SPF)

APNIC

Path Calculation

- Modified Dijkstra
- Often referred to as CSPF
  – Constrained SPF
- …or PCALC (path calculation)
- Final result is explicit route meeting desired constrain

Path Calculation (C-SPF)

- Shortest-cost path is found that meets administrative constraints
- These constraints can be
  - bandwidth
  - link attribute (aka color, resource group)
  - priority
- The addition of constraints is what allows MPLS-TE to use paths other than *just* the shortest one

Path Computation

"On demand" by the trunk's head-end:

- for a new trunk
- for an existing trunk whose (current) LSP failed
- for an existing trunk when doing re-optimization

Path Computation

Input:

- configured attributes of traffic trunks originated at this router
- attributes associated with resources
    - available from IS-IS or OSPF
- topology state information
    - available from IS-IS or OSPF

Path Computation

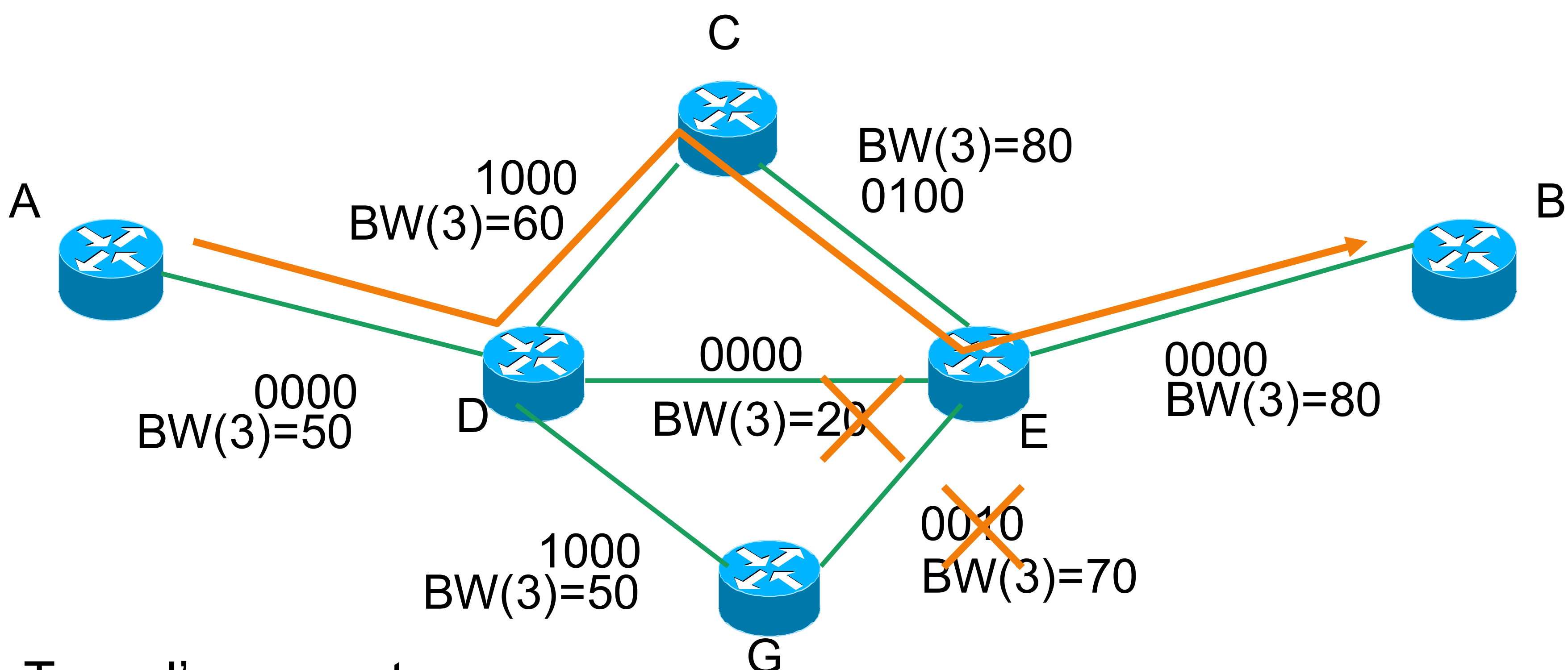
- Prune links if:
  - insufficient resources (e.g., bandwidth)
  - violates policy constraints
- Compute shortest distance path
  - TE uses its own metric
- Tie-break:
  1. Path with the highest available bandwidth
  2. Path with the smallest hop-count
  3. Path found first in TE topology database

Path Computation

Output:

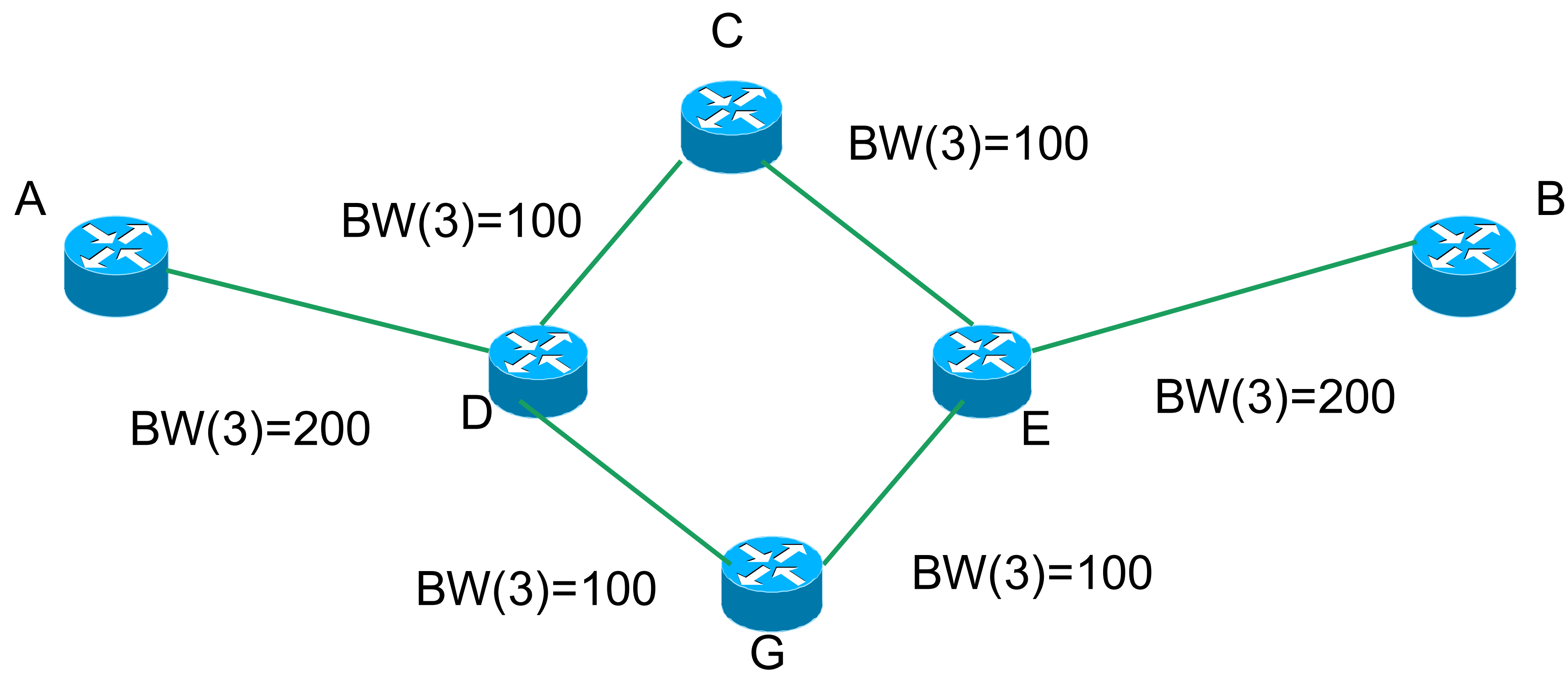
- explicit route - expressed as a sequence of router IP addresses
  - interface addresses
  - loopback address
- used as an input to the path setup component

BW/Policy Example



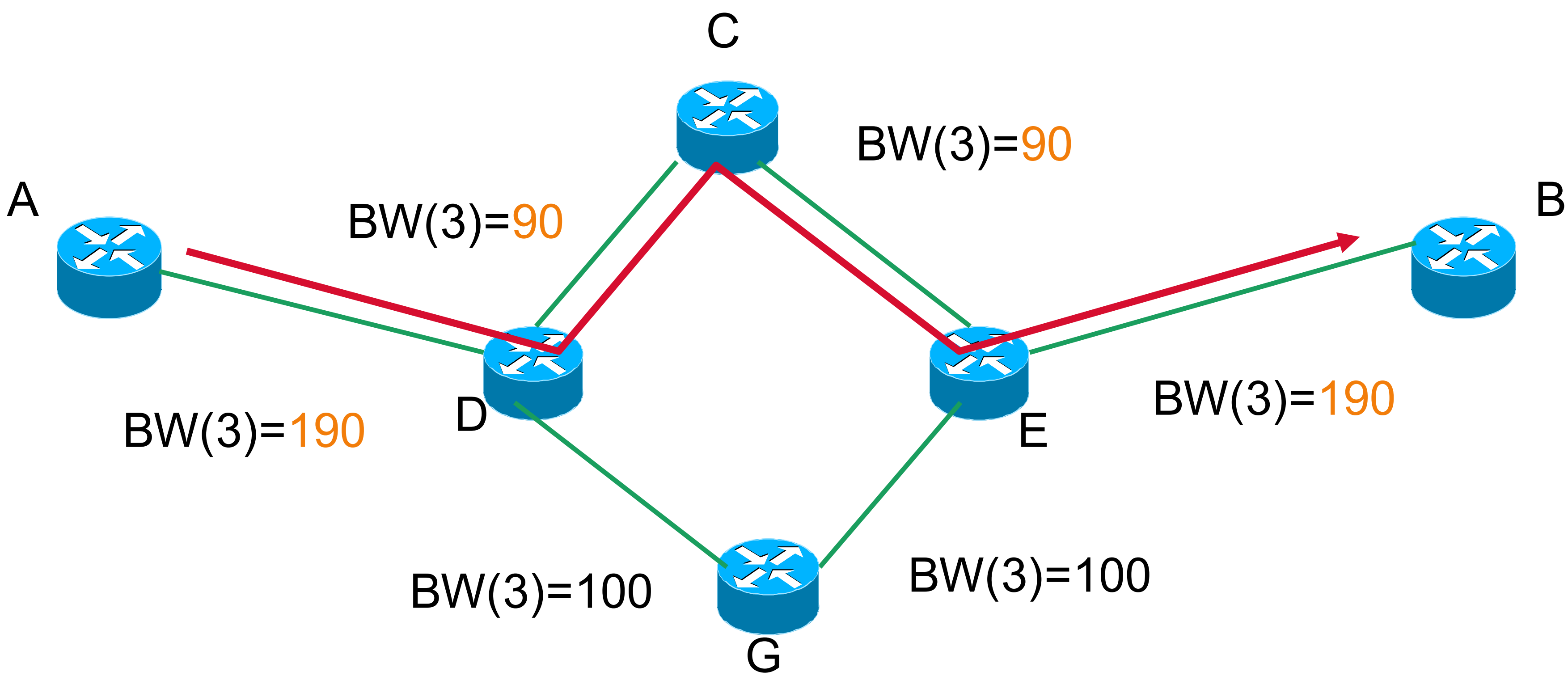
- Tunnel's request:
  - Priority 3, BW = 30 units,
  - Policy string: 0000, mask: 0011

Load-Balancing tunnels



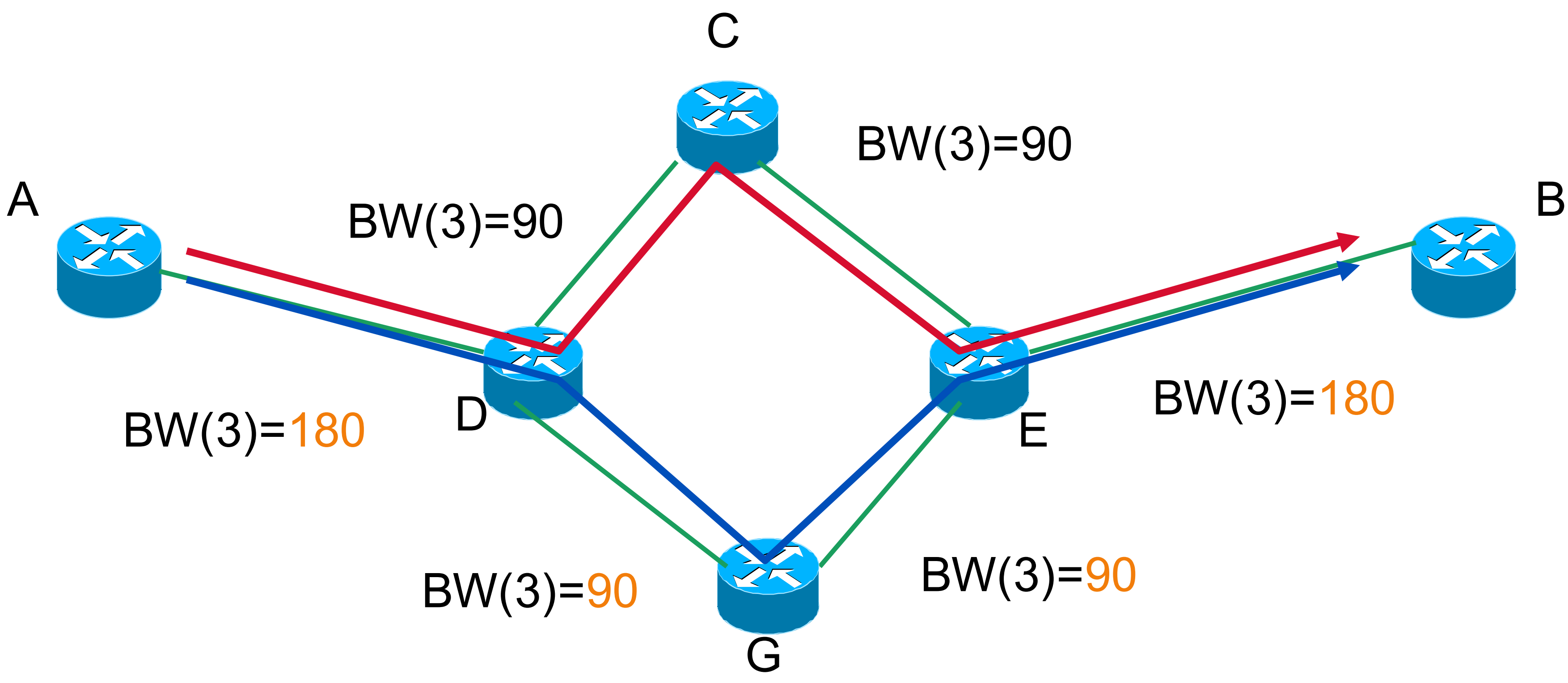
- all tunnels require 10

Load-Balancing tunnels



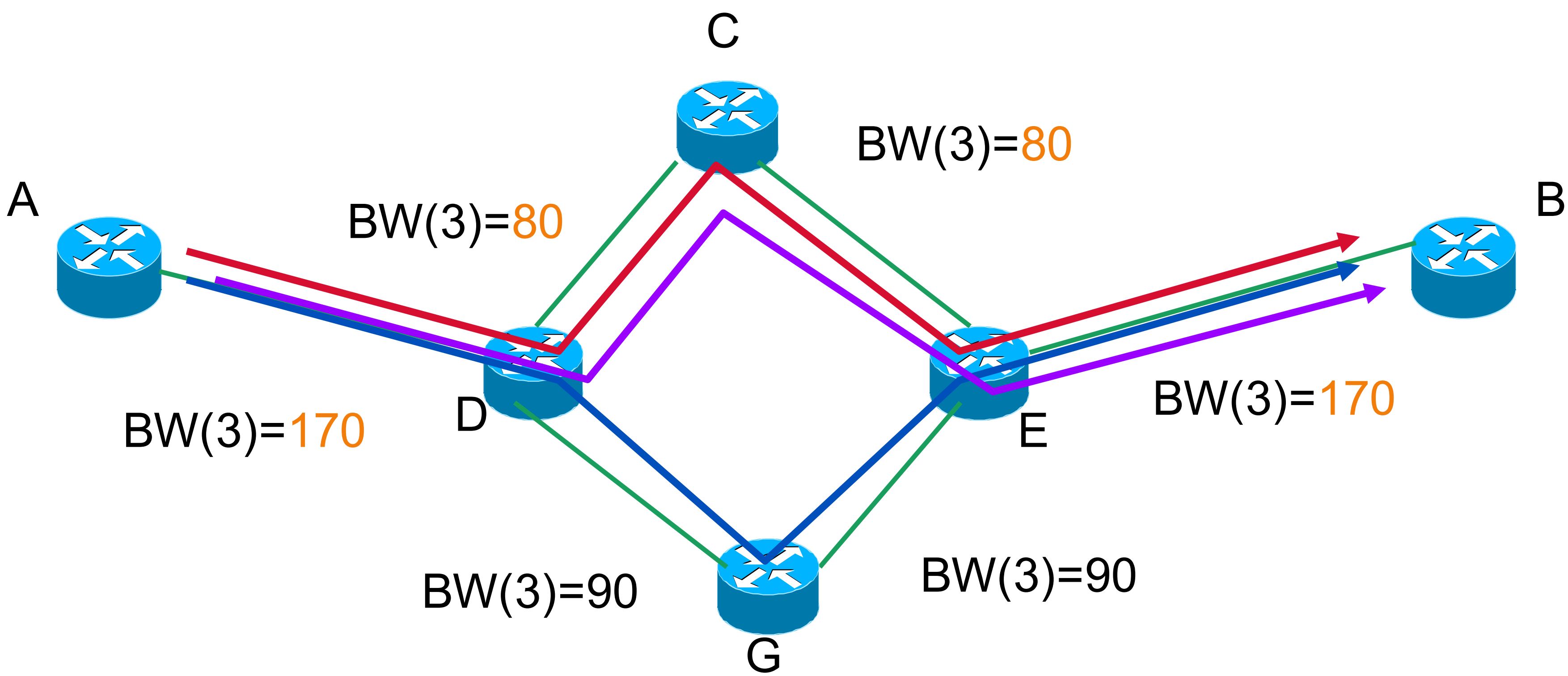
- all tunnels require 10

Load-Balancing tunnels



- all tunnels require 10

Load-Balancing tunnels



- all tunnels require 10

Mapping Traffic to Path

Routing Traffic Down a Tunnel

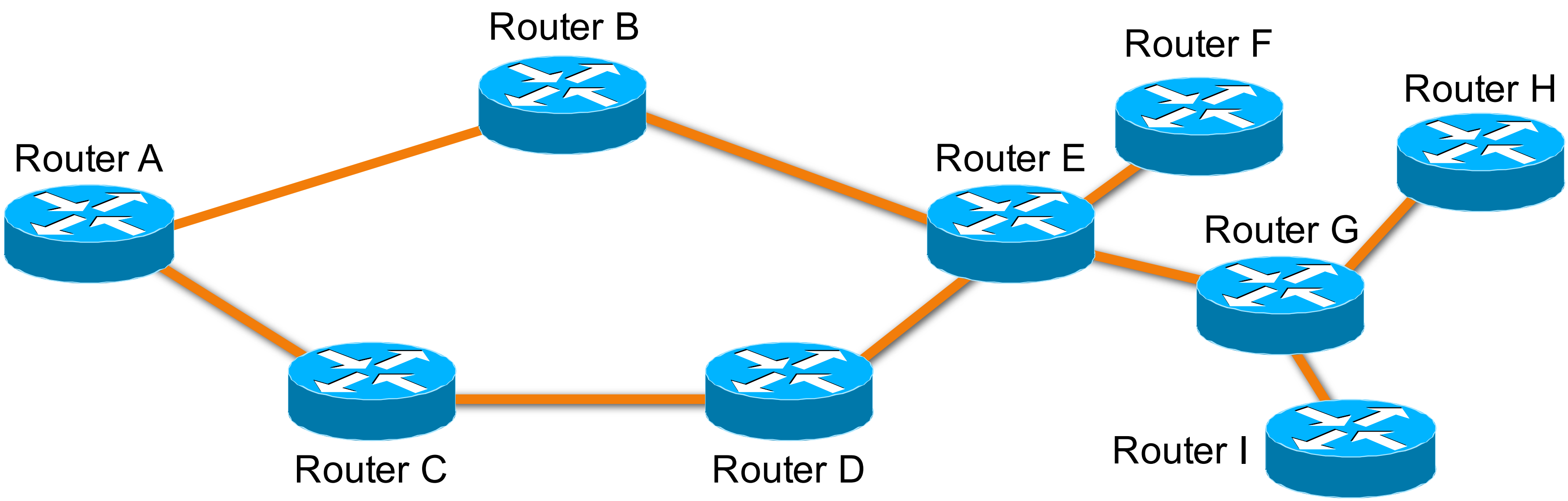
- Once RESV reaches headend, tunnel interface comes up
- How to get traffic down the tunnel?
  1. Autoroute
  2. Forwarding adjacency
  3. Static routes
  4. Policy routing

Autoroute

- Tunnel is treated as a directly connected link to the tail
- IGP adjacency is NOT run over the tunnel!
  - Unlike an ATM/FR VC
- Autoroute limited to single area/level only

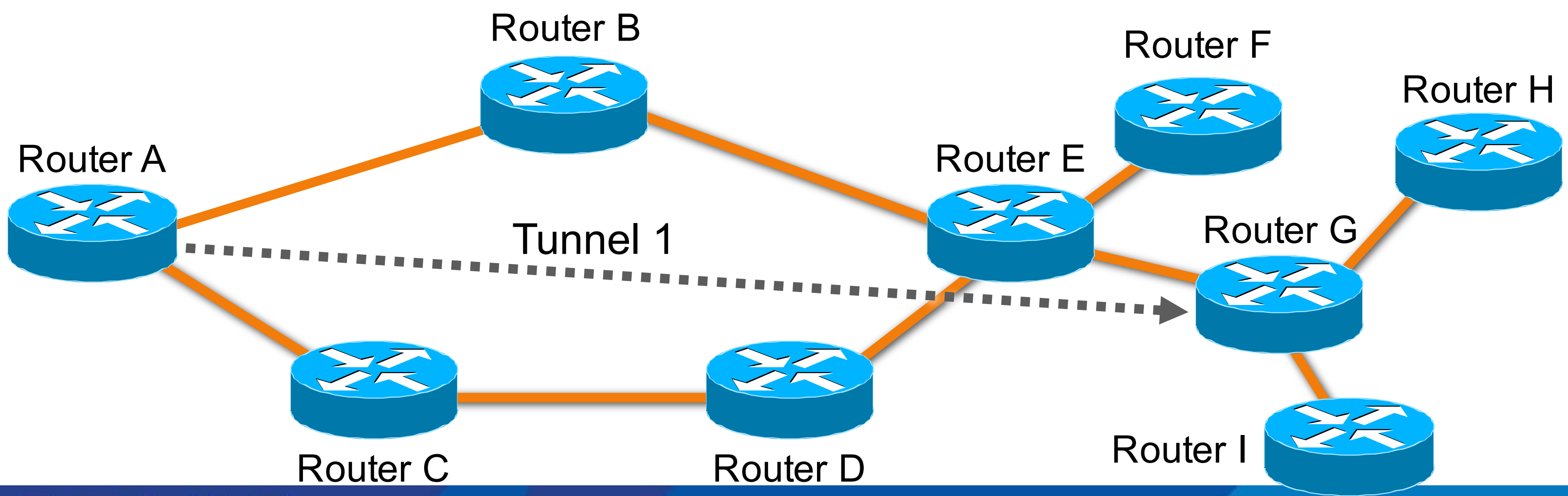
Autoroute

This Is the Physical Topology



Autoroute

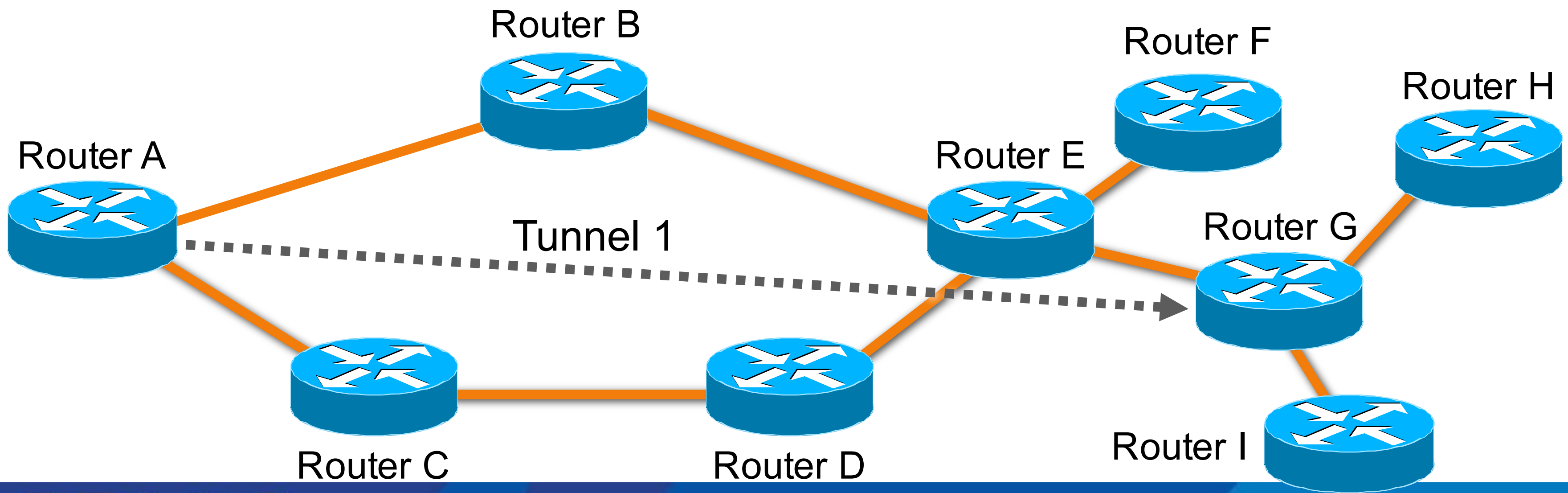
- This is Router A's logical topology
- By default, other routers don't see the tunnel!



Autoroute

Node	Next-Hop	Cost
B	B	10
C	C	10
D	C	20
E	B	20
F	B	30
G	Tunnel 1	30
H	Tunnel 1	40
I	Tunnel 1	40

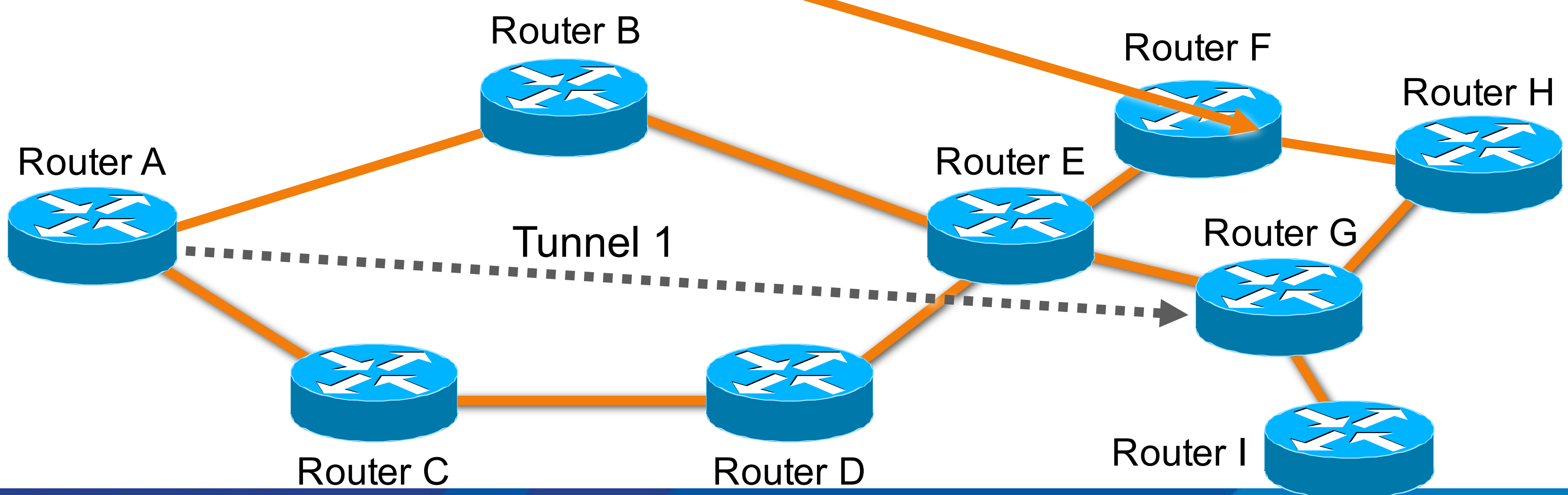
- Router A's routing table, built via auto-route
- Everything "behind" the tunnel is routed via the tunnel



Autoroute

Node	Next-Hop	Cost
B	B	10
C	C	10
D	C	20
E	B	20
F	B	30
G	Tunnel 1	30
H	Tunnel 1 & B	40
I	Tunnel 1	40

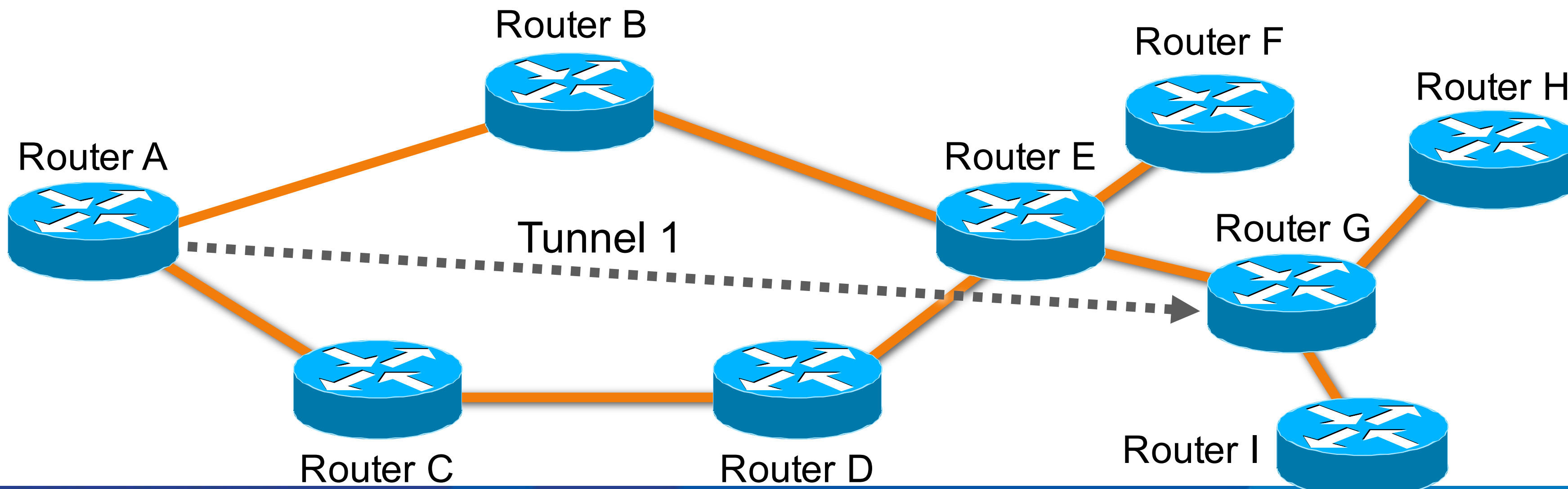
- If there was a link from F to H, Router A would have 2 paths to H (A->G->H and A->B->E->F->H)
- Nothing else changes



Autoroute

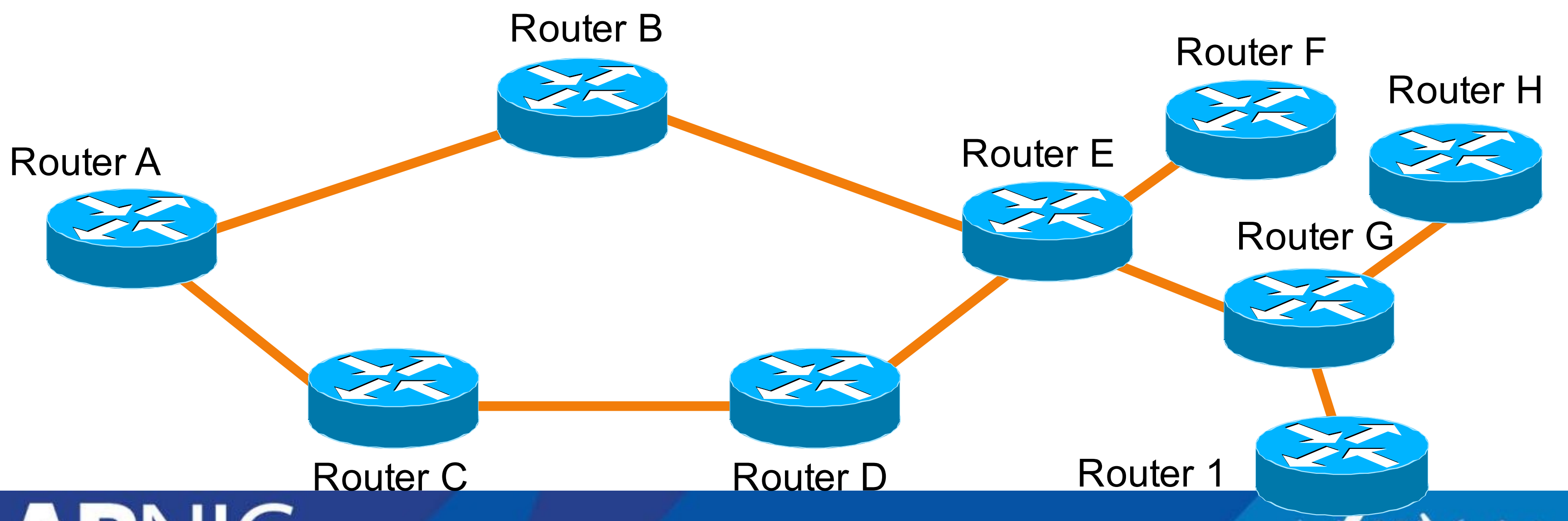
Node	Next-Hop	Cost
B	B	10
C	C	10
D	C	20
E	B	20
F	B	30
G	Tunnel 1	30
H	Tunnel 1	40
I	Tunnel 1	40

```
interface Tunnel1
 tunnel mpls traffic-eng autoroute announce
```



Static Routing

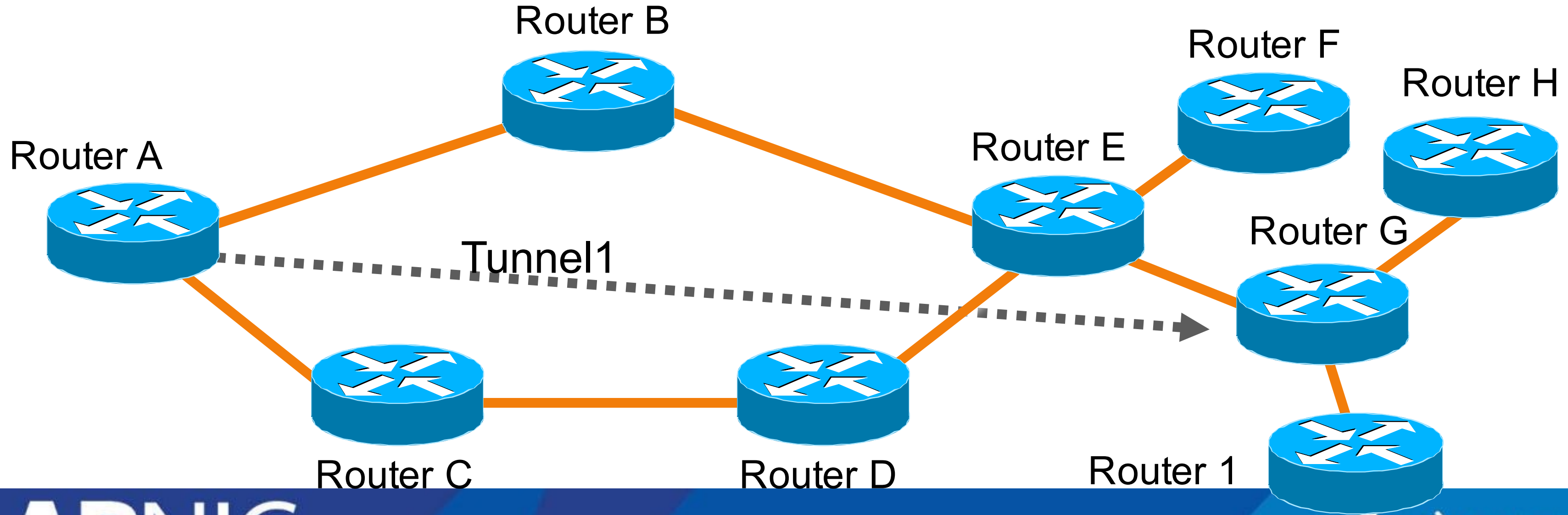
```
RtrA(config)#ip route H.H.H.H 255.255.255.255 Tunnel1
```



Static Routing

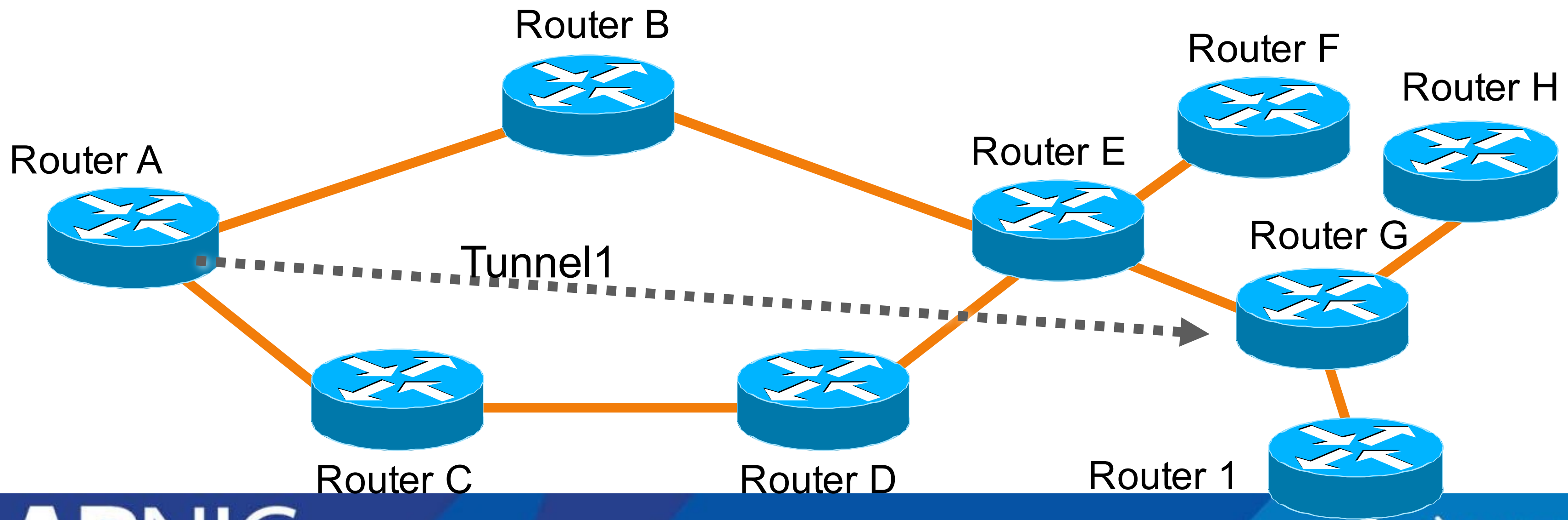
Node	Next-Hop	Cost
B	B	10
C	C	10
D	C	20
E	B	20
F	B	30
G	B	30
H	Tunnel 1	40
I	B	40

- Router H is known via the tunnel
- Router G is not routed to over the tunnel, even though it's the tunnel tail!



Policy Routing

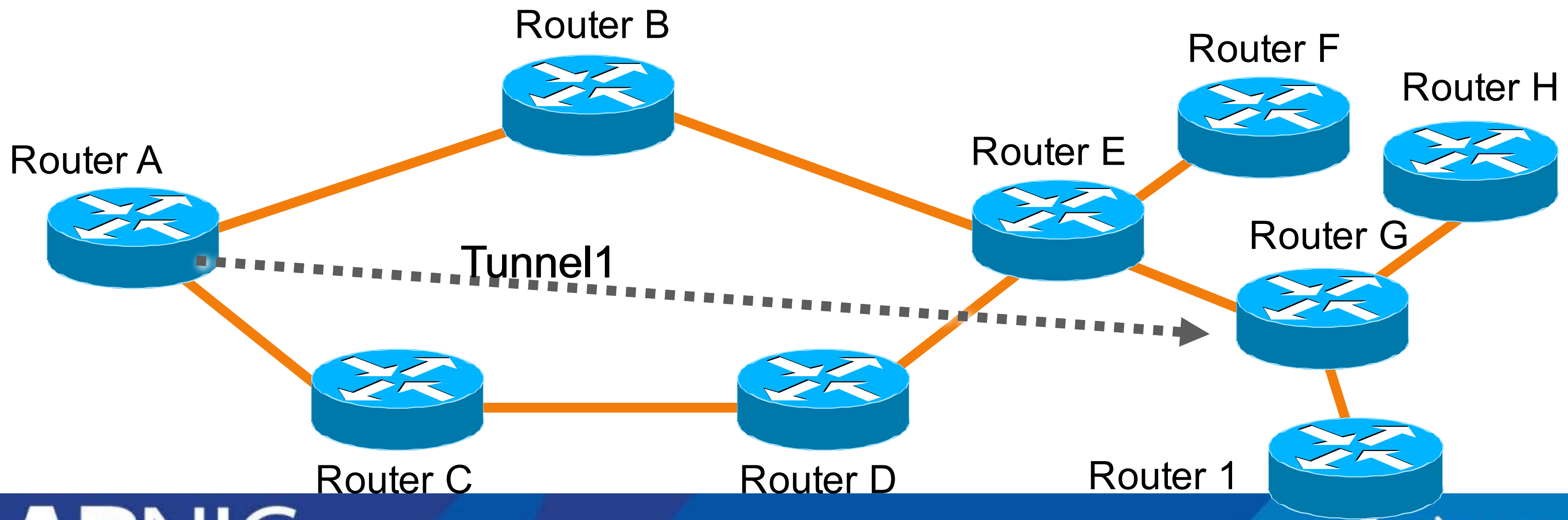
```
RtrA(config-if)#ip policy route-map set-tunnel
RtrA(config)#route-map set-tunnel
RtrA(config-route-map)#match ip address 101
RtrA(config-route-map)#set interface Tunnel1
```



Policy Routing

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

- Routing table isn't affected by policy routing
- Need (12.0(23)S or 12.2T) or higher for 'set interface tunnel' to work



MPLS-TE: Configuring and monitoring

APNIC

Prerequisite Configuration (Global)

```
ip cef [distributed]

mpls traffic-eng tunnels
```

Information Distribution

- OSPF

```
mpls traffic-eng tunnels
mpls traffic-eng router-id loopback0
mpls traffic-eng area ospf-area
```

- ISIS

```
mpls traffic-eng tunnels
mpls traffic-eng router-id loopback0
mpls traffic-eng level-x
metric-style wide
```

Information Distribution

- On each physical interface

```
interface pos0/0
  mpls traffic-eng tunnels
  ip rsvp bandwidth Kbps (Optional)
  mpls traffic-eng attribute-flags
 attributes (Opt)
```

Build a Tunnel Interface (Headend)

```
interface Tunnel0
  ip unnumbered loopback0
  tunnel destination RID-of-tail
  tunnel mode mpls traffic-eng
   tunnel mpls traffic-eng bandwidth  10
```

Tunnel Attributes

```
interface Tunnel0
  tunnel mpls traffic-eng bandwidth Kbps
  tunnel mpls traffic-eng priority pri [hold-pri]
  tunnel mpls traffic-eng affinity properties [mask]
  tunnel mpls traffic-eng autoroute announce
```

Path Calculation

- Dynamic path calculation

```
int Tunnel0
   tunnel mpls traffic-eng path-option #
  dynamic
```

- Explicit path calculation

```
int Tunnel0
  tunnel mpls traffic path-opt # explicit
  name foo

ip explicit-path name foo
   next-address 1.2.3.4 [loose]
   next-address 1.2.3.8 [loose]
```

Multiple Path Calculations

- A tunnel interface can have several path options, to be

```
tunnel mpls traffic-eng path-option 10 explicit name foo
tunnel mpls traffic-eng path-option 20 explicit name bar
tunnel mpls traffic-eng path-option 30 dynamic
```

- Path-options can each have their own

```
tunnel mpls traffic-eng path-option 10 explicit name foo
     bandwidth 100
tunnel mpls traffic-eng path-option 20 explicit name bar
     bandwidth 50
tunnel mpls traffic-eng path-option 30 dynamic
     bandwidth 0
```

LSP Attributes

Configure on Tunnel:

```
tunnel mpls traffic-eng
  path-option 10 dynamic
  attributes foo
```

Attribute list 'foo' is defined at:

```
mpls traffic-eng lsp
  attributes foo
 bandwidth 25
 priority 2 2
```

- Attribute list options

```
affinity
auto-bw
bandwidth
lockdown
priority
protection
record-route
```

Static and Policy Routing Down a Tunnel

- Static routing

```
ip route prefix mask Tunnel0
```

- Policy routing (Global Table)

```
access-list 101 permit tcp any any eq
  www

interface Serial0
  ip policy route-map foo

route-map foo
  match ip address 101
  set interface Tunnel0
```

Autoroute and Forwarding Adjacency

```
interface Tunnel0
  tunnel mpls traffic-eng autoroute
 announce
```

Questions?

APNIC

Post Training Engineering Assistance

- **What it is?**
  - APNIC training team often receives requests for personalized assistance after training/workshop events
  - These requests are typically from individual trainees of APNIC members/non-member organizations
  - Asking APNIC trainers to visit and provide advice and guidance on the next stages of a particular technology specification or deployment
  - In the past, training team made efforts to provide such assistance on an ad hoc basis
  - The Engineering Assistance Program formalizes this support
  - Allowing APNIC to provide structured assistance for APNIC member organizations that request extra support

Post Training Engineering Assistance

- **Program Objective?**
  - APNIC's ongoing commitment to directly supporting regional infrastructure development.
  - Engineering Assistance Program bridges the gap between APNIC Training courses and the services of a consultancy organization that would provide advanced assistance with major infrastructure deployments.
  - APNIC Training aims to assist organizations requesting advice on specification, deployment, and operational technologies that were covered in APNIC Training sessions attended by that APNIC member organization's staff.

Post Training Engineering Assistance

- **How much it cost?**
  - Engineering Assistance is run on a cost recovery basis.
  - Need to pay the trainers for travel cost or additional days after training
  - With the daily fee structured into the same three categories used for Tutorials and Workshops fees
    - **Category A** Developed Economies.
    - **Category B** Developing Economies.
    - **Category C** Least Developed Economies.

Post Training Engineering Assistance

- **Engineering Assistance program covers following topic:**
  - Routing Protocols e.g. BGP, OSPF, ISIS
  - IPv6 technology and deployment
  - Small scale MPLS deployment
  - IXP establishment and deployment
  - DNS and DNSSEC deployment
  - Newer technologies such as BGP Security (RPKI)

Post Training Engineering Assistance

- **How to know more about Engineering Assistance?**
  - https://training.apnic.net/engineering-assistance
  - training@apnic.net

Thank you

APNIC