BGP Traffic Engineering Using RSVP-TE?

Tom Scholl

AT&T Labs

<tom.scholl@att.com>

Richard Steenbergen

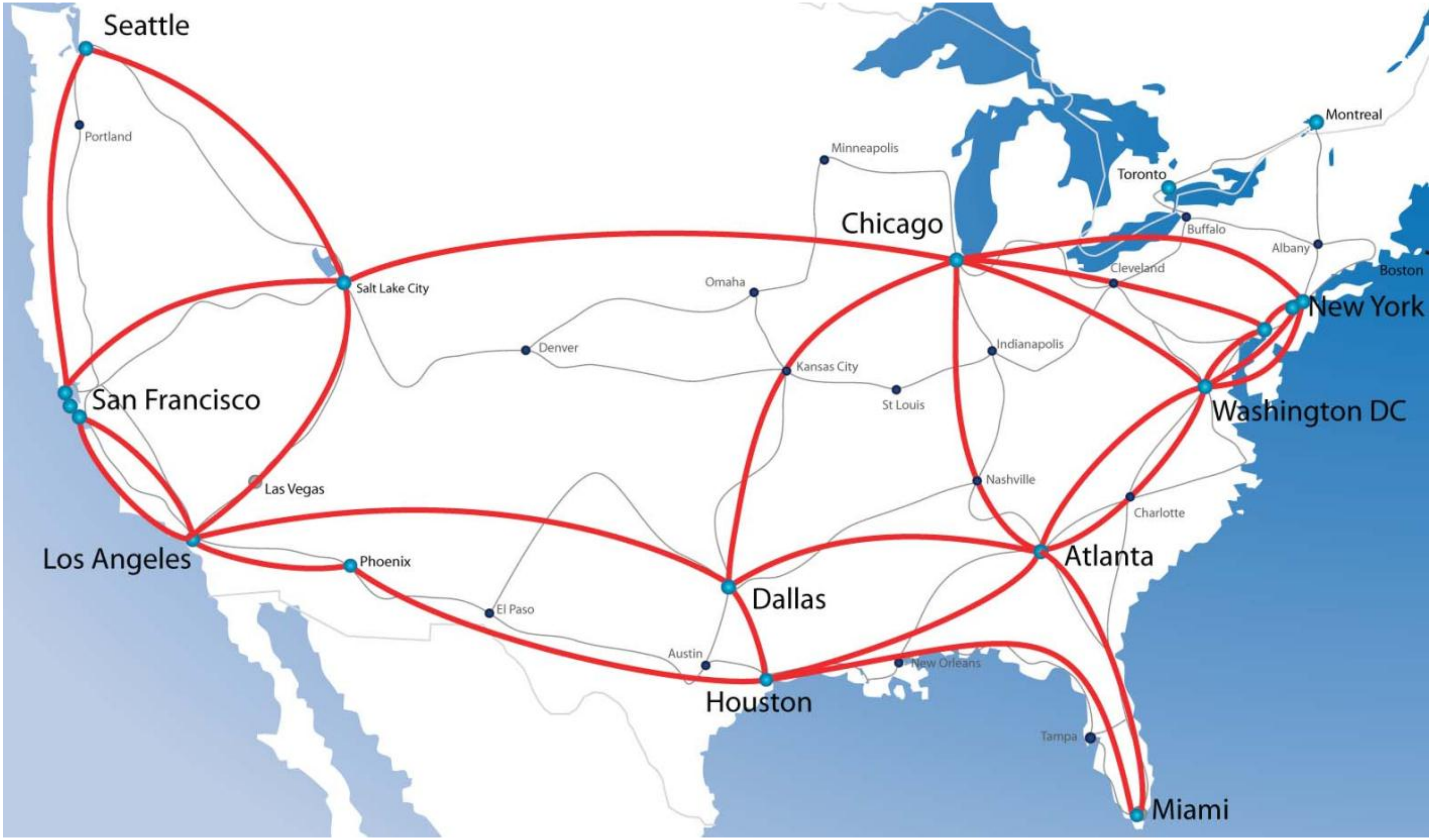
nLayer Communications

<ras@nlayer.net>

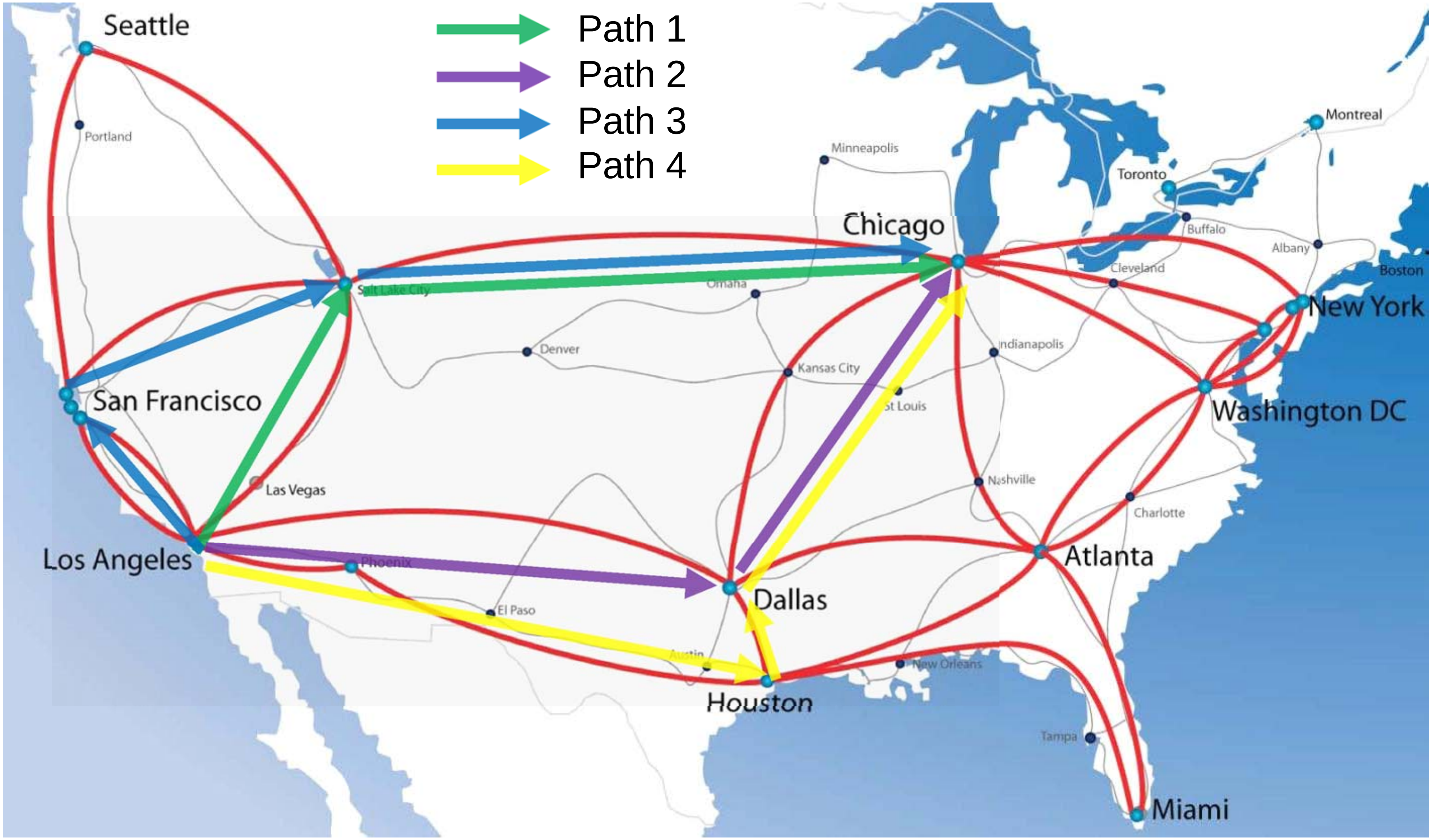
Traffic Engineering For Dummies

- Today, operators have great software mechanisms for doing Traffic Engineering (TE) within our network.
  - Using MPLS and RSVP-TE protocols.
  - Traffic on MPLS LSPs between routers is measured.
  - Capacity across our network is reserved using RSVP.
  - If capacity for an LSP isn't available on the shortest path, we pull it from the next shortest path to avoid congestion.
  - Traffic flows can be kept up to date with auto-bandwidth.
  - And react quickly to events like fiber cuts or outages.
  - A well designed MPLS backbone can largely run itself.

Generic Random US Backbone Network



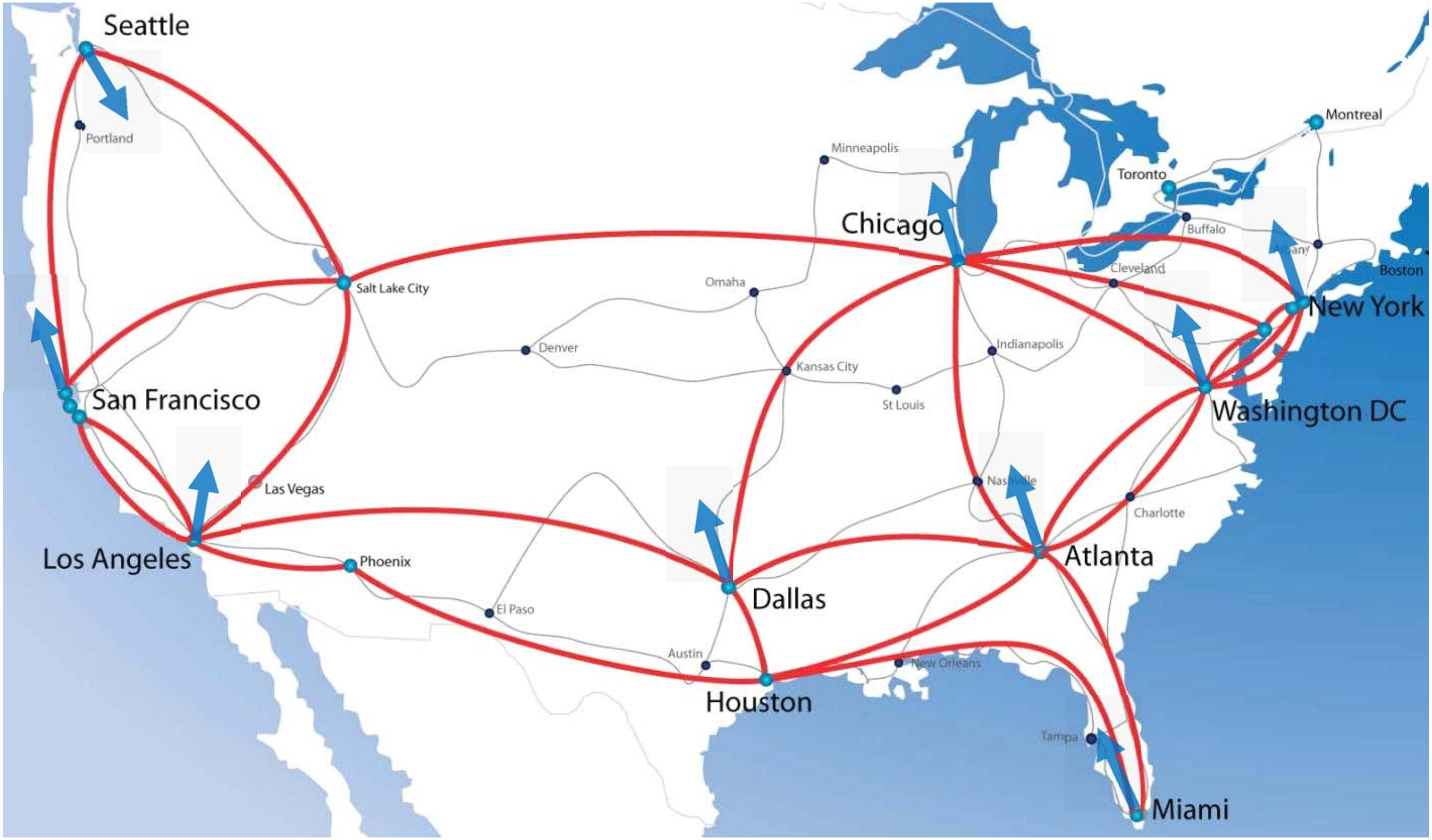
MPLS with RSVP-TE Traffic Flow



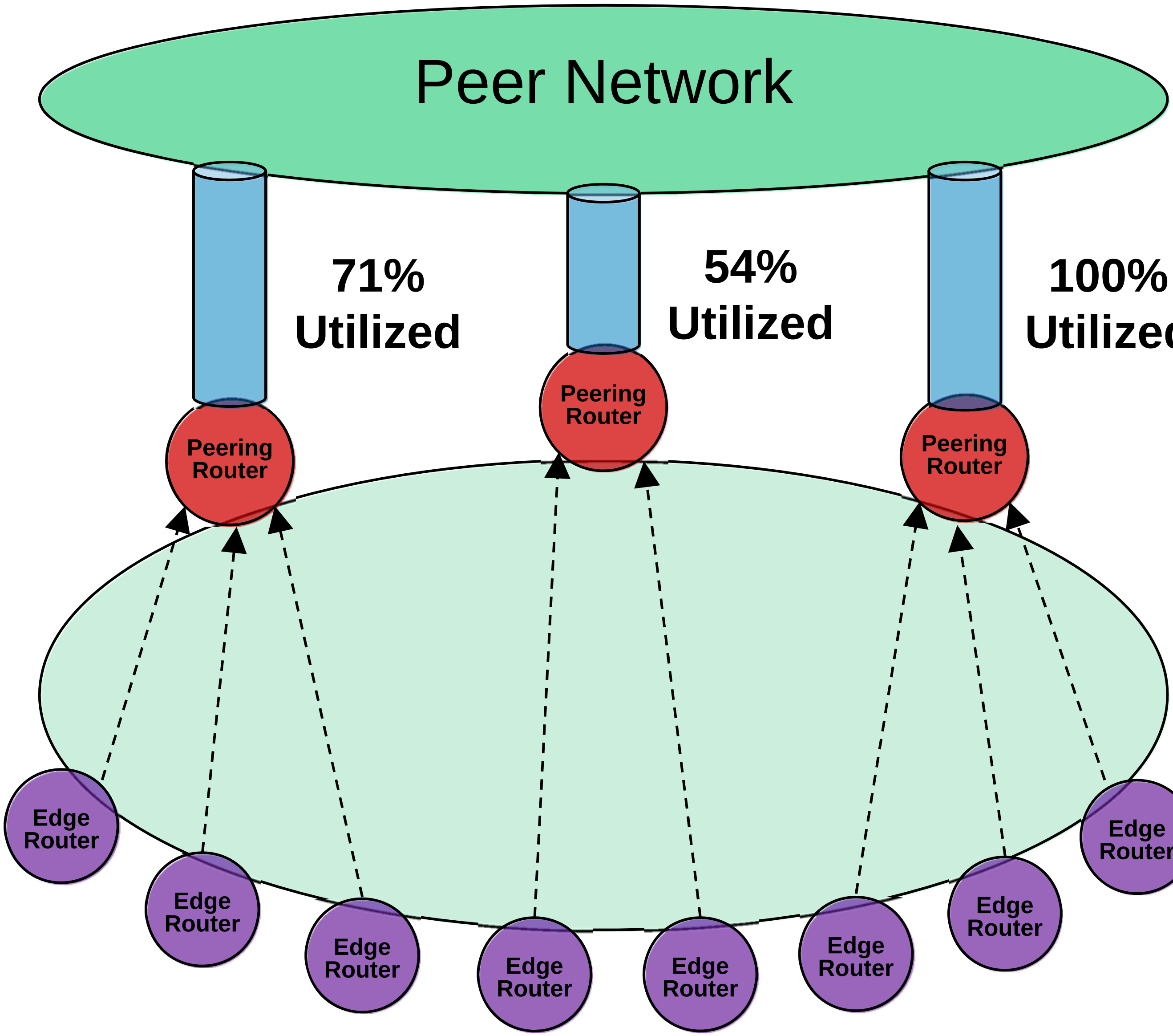
Traffic Engineering and the Internet

- But there is another kind of traffic engineering beyond managing capacity on our own network.
  - Traffic that we send to other networks on the Internet.
  - You know, to those pesky routes we learn via BGP.
- And there are no router based mechanisms to automatically manage or balance the traffic today.
  - At best we have crude, manually operated mechanisms like policies for local-preference, MEDs, etc.
  - And much like the original problems of managing complex networks that motivated MPLS-TE, we are limited in the number of interconnections points humans can manage.

External Traffic Flow To The Internet



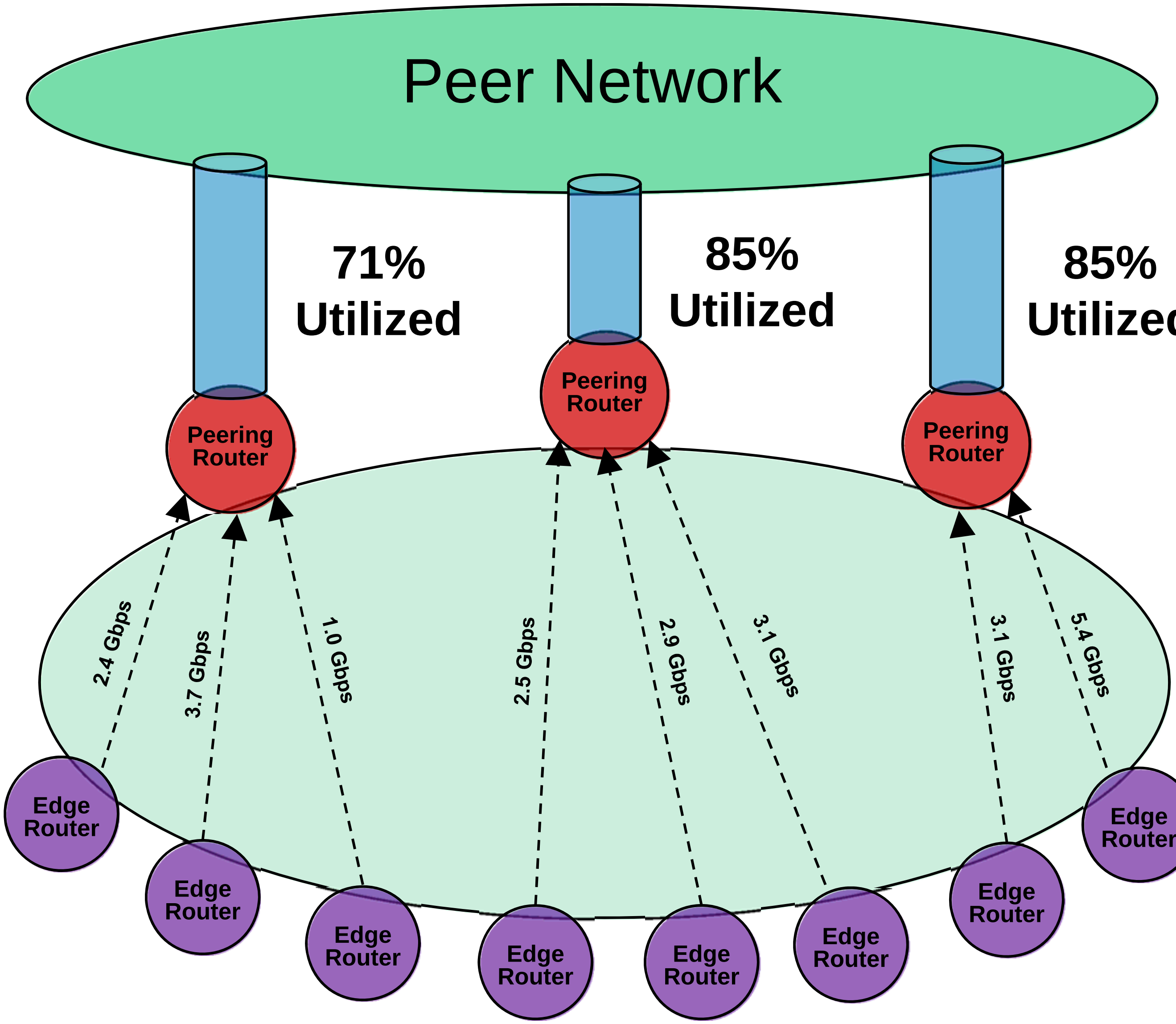
Common Traffic Engineering Dilemma



BGP Best Path Decision

- The BGP best path decision is lacking information
  - It doesn't know anything about capacity.
  - It barely knows anything about where traffic should go.
    - And every network who has tried wide-scale MEDs knows that for every one destination it improves, it breaks two others.
    - Large networks are mostly stuck doing closest exit.
- So operators must handle all of this manually.
- This got us thinking, "is there a better way"?
  - Can we adapt the TE mechanisms we already have?
  - Could we use this to let BGP make better decisions?

The End Goal



Peer Network
71% Utilized
85% Utilized
85% Utilized
Peering Router
Peering Router
Peering Router
2.4 Gbps
3.7 Gbps
1.0 Gbps
2.5 Gbps
2.9 Gbps
3.1 Gbps
3.1 Gbps
5.4 Gbps
Edge Router
Edge Router
Edge Router
Edge Router
Edge Router
Edge Router
Edge Router
Edge Router

A Possible Solution?

- Use MPLS/RSVP-TE to build LSPs to the edge
  - All the way out to the destination interface.
- Why MPLS?
  - One constraint is that we can't create routing loops.
  - With multiple routers making independent and maybe inconsistent decisions about best path using capacity, we could easily create IP forwarding loops.
  - MPLS allows up to avoid this by deciding the complete forwarding path once at the ingress of the IP packet.
  - This is a key reason why MPLS-TE works today.

What Information Are We Missing?

- To do BGP Traffic Engineering, we'd need:
  - To know the available bandwidth at each exit:
    - BGP itself knows absolutely nothing about available capacity.
    - And while it might revolutionize routing on the Internet if it did, this is not an easy fix and is beyond the scope of what we could do.
    - Even if it did, it would need to be dynamic to reflect utilization.
  - To know about all available exits, on every edge router:
    - This is a problem, because BGP hides non-best-path information.
    - Passing along only the best route is inherent in BGP's design.
    - Sending a second path is an implicit withdrawal of the first path.

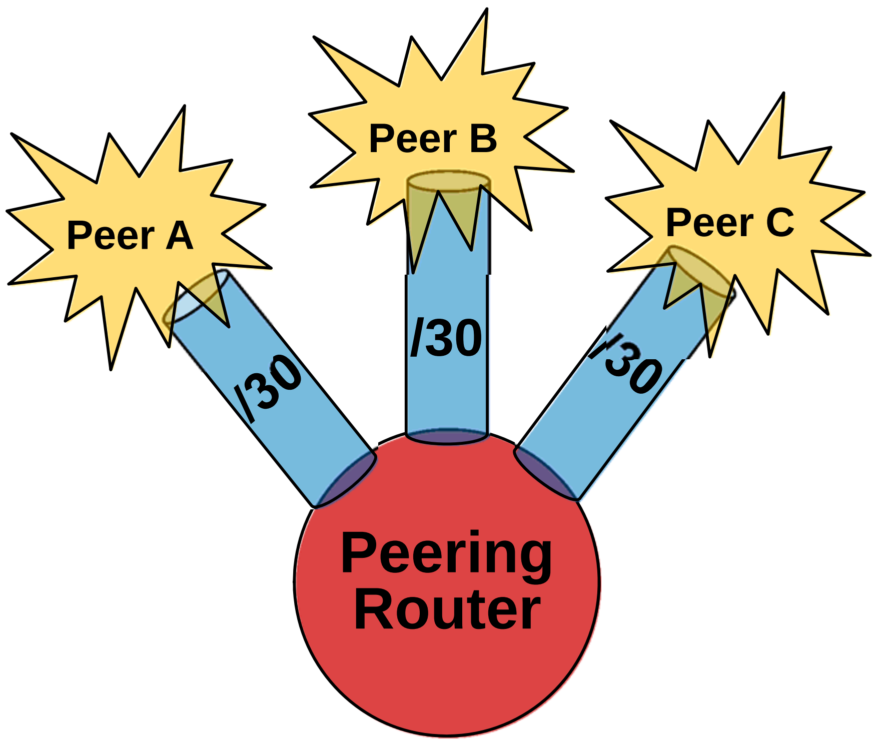
So How Could We Do It?

- How could we track capacity to each peer?
  - Without adding anything to the BGP protocol?
  - Why not track capacity to the interface next-hop instead?
- This imposes the following design requirements:
  - Your eBGP next-hops need to be carried in your IGP.
  - They need to utilize IGP (OSPF/ISIS) TE extensions.
  - You can’t reset the next-hops to self (lo0) in your iBGP.
- Does this create scaling issues?
  - Depends - Even a very large network is probably only talking about hundreds of next-hops to edge networks.
  - This would probably only be used on peers/transits.

So How Could We Do It?

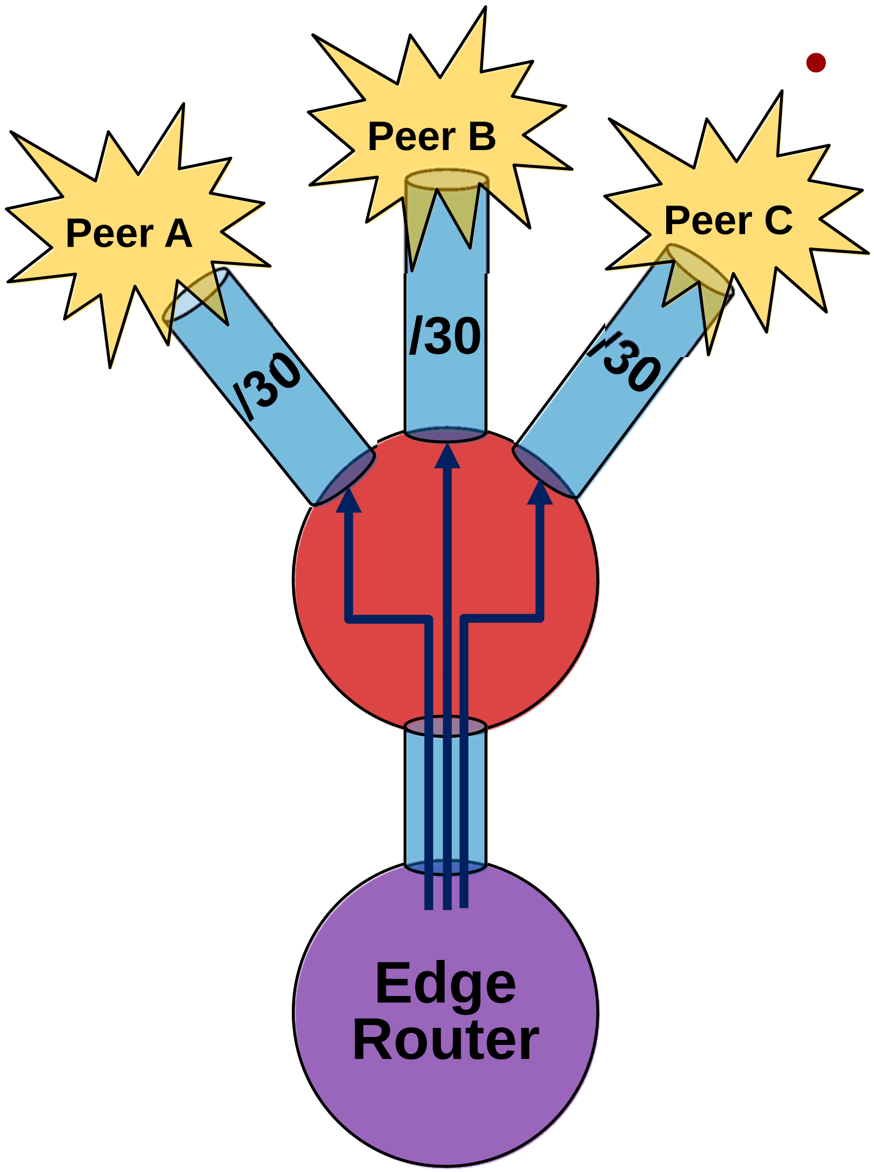
- But what about learning every possible exit?
  - If we don’t know our other options, we can’t use them.
  - BGP does an excellent job of hiding inactive paths.
  - Even more so if you use route-reflectors.
- One solution is BGP ADD-PATHS
  - A draft currently under consideration in the IETF.
  - It allows BGP to pass more than just a single best path.

How Would This Work



- The edge interfaces need to be in IGP w/TE extensions.
- You can't quite do this today
  - Setting the interfaces to passive injects the route but not TE info.
  - But it would be a pretty simple sw modification to let you inject these /30s without actually speaking your IGP over them.
  - We only care about egress utilization, participation on the other side is not required.

How Would This Work



- Build LSPs to the Edge
  - There are a couple ways to do this
    - Build a RSVP-TE LSP to the egress interface across your core.
    - Build a nested RSVP-TE LSP through an existing LSP to a peering router
      - Eliminates LSP state within the Core.
    - Use BGP Unicast-Label with RSVP
      - Redistribute peer /30 connected interfaces into BGP.
      - Allows you to bypass Peering Router IP lookup.

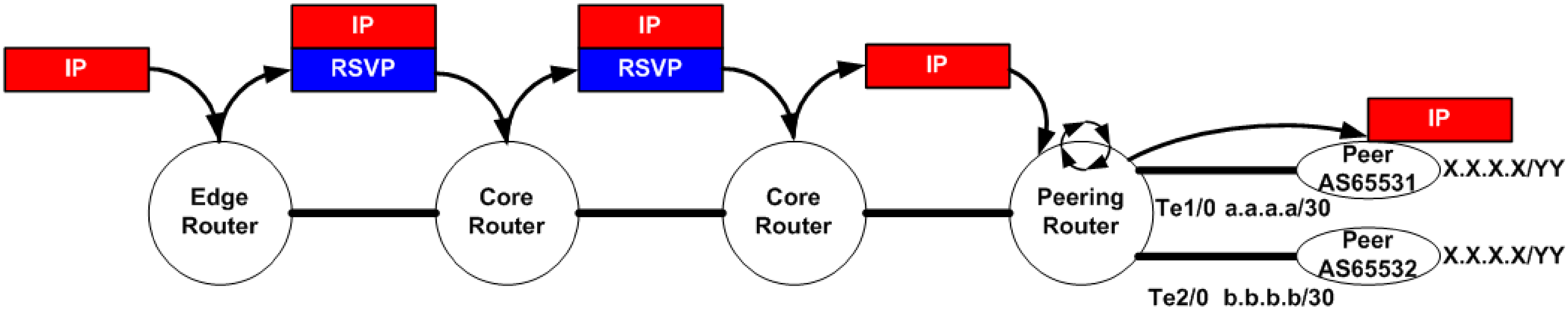
BGP Unicast Label?

Can be used as a way to bypass IP routing lookup on the peering router

- RSVP-TE LSPs built to an interface generally are treated like a LSP built to the loopback.
- Packets are popped on the peering router and then have an IP lookup and a routing decision is made.
- RSVP could possibly be used to bypass IP lookup, but not currently.

With BGP Unicast-Label, an additional label in the stack can have traffic transit through a router without an IP lookup directly to the egress (peer) interface.

BGP Unicast Label?

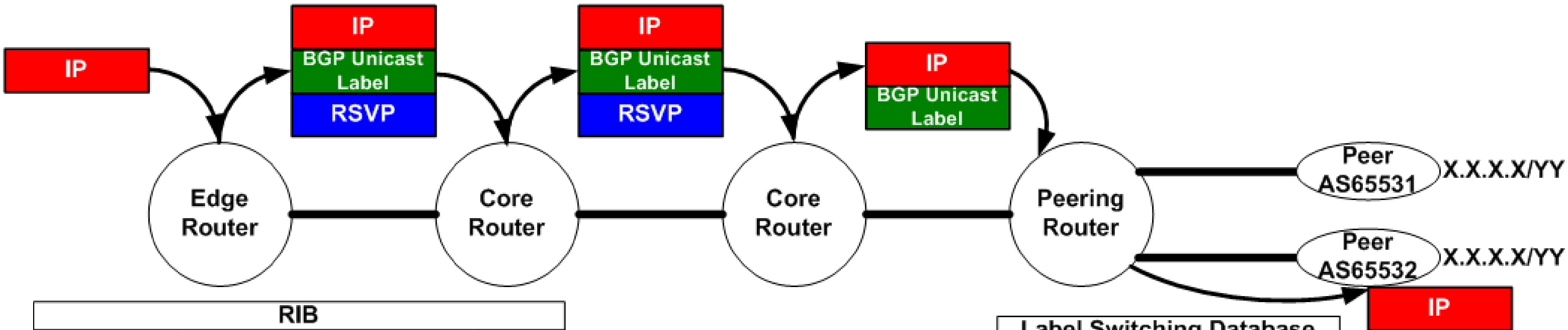


RIB		
Prefix	AS-Path	Next-Hop
X.X.X.X/YY	AS65531	a.a.a.a/30
X.X.X.X/YY	AS65532	b.b.b.b/30

RIB		
Prefix	AS-Path	Best Path?
X.X.X.X/YY	AS65531	Best due to RID
X.X.X.X/YY	AS65532	

You are still restricted to the peering router performing the IP lookup and making a decision within the RIB on where to forward you.

Our LSP is Online!



RIB			
Prefix	AS-Path	Next-Hop	Label
X.X.X.X/YY	AS65531	a.a.a.a/30	123
X.X.X.X/YY	AS65532	b.b.b.b/30	321

Label Switching Database	
Prefix	Next-Interface
123	Te1/0
321	Te2/0

With BGP Unicast-Label, your can stack another label and bypass the routing lookup on the peering router. This would allow for Edge Routers to have per-interface granularity on remote devices within the network.

Ok Now What?

- Assuming everything so far works, we now:
  - Know every BGP exit available to us, on every router.
  - Know the bandwidth available to each next-hop.
  - Have a way to reserve capacity to each next-hop.
  - Have a way to deliver packets to each next-hop.
- Now the $64,000 question:
  - How could we use this data to implement TE?

One Idea We Tossed Around

- When capacity to a particular next-hop is exceeded:
  - Make that next-hop ineligible for use in BGP on this router.
  - Would immediately make BGP re-evaluate it's choices, and select the next best path for the affected routes.
- Problems:
  - All traffic from the source router would be shifted away.
    - Potentially shifting far more traffic than we actually want.
  - We have no way to know what traffic was shifted where.
    - We know how much traffic went to this next-hop before our actions.
    - But we don't know where this traffic is going to go afterwards.
    - Thus we can't reserve BW for it, so our reservations will be wrong.
      - Leads to chaos, probable route oscillations, undetected congestion, etc.

Some More Ideas That Didn't Work

- Could we shift traffic to the next closest next-hop?
  - As in the case of a multi-exit peer.
  - No, we can't risk blackholing traffic.
    - There is no guarantee that we're getting consistent advertisements.
    - We have to look at each route individually, to make sure our backup path is capable of handling this specific route.
- Could we just alter the BGP best path algorithm?
  - To have it check for capacity on a per-route basis?
  - This really risks having something very unpredictable.
    - We couldn't know if the route we just moved was a big one, or not.
    - We couldn't know when to stop moving traffic. Again, chaos.

The Big Gotcha

- We really need to know the traffic to each prefix.
  - Traffic to a next-hop is fine for detecting the congestion.
  - But it doesn't completely help with fixing it.
- Why do we need to know the traffic to each prefix?
  - We don't want to kill the entire next-hop for a whole router.
    - Say a particular source router is sending 4 Gbps of traffic.
    - It wants to move to 5 Gbps, but that won't fit via the best path.
    - Is moving the existing 4 Gbps of traffic somewhere else really the best move? Probably not, it's too big a hammer.
  - We'd need to know how much traffic we moved, and where.
    - The traffic probably won't all move to one single new next-hop.
    - We'd need to assign the correct reservations based on new paths.

Measure Traffic to Each Prefix

- So could we track traffic to each prefix?
  - Maybe. This is potentially something a router could keep track of, at least internally, on a 1-5 minute basis.
- And could we do reservations with it?
  - 300k+ reservations per router is probably a bad idea.
  - But maybe we could simply use this data to do our next-hop reservations, by aggregating the total traffic value.
  - This could let us move just the right number of routes to solve our capacity limitations, and still have correct data when we do.

Problems

This is far from a final solution.

- Remember, LSP’s are make-before-break.
  - You may not have enough capacity to “grow” your LSP.
  - Every time you resignal bandwidth you have to m-b-b anyways.
- The right answer may be to not use an LSP for TE at all.
  - You still need the LSP to deliver your packets to the final dest.
  - But the reservation process is still pretty different from RSVP-TE.
  - It might be better to just implement a new reservation scheme, (perhaps built on the existing RSVP software components), which talks between the SRC and DST routers directly without involving the MPLS bandwidth components.
  - This might reduce overhead, and completely eliminate the need to make before break with every bandwidth update.

Send questions, comments, to:

Tom Scholl
AT&T Labs
<tom.scholl@att.com>

Richard Steenbergen
nLayer Communications
<ras@nlayer.net>