MPLS RSVP-TE Auto-Bandwidth: Practical Lessons Learned

Richard A Steenbergen <ras@nlayer.net> nLayer Communications, Inc.

MPLS RSVP-TE Quick Review

- MPLS Traffic Engineering 101
  - Classically, IGPs used only link cost to select a best path.
    - And a “Shortest Path First” (SPF) algorithm finds the “lowest cost” path.
  - Traffic Engineering takes this, and adds additional constraints.
    - For example, “find the lowest cost path *that also has available bandwidth*”.
  - RSVP-TE accomplishes this by doing the following:
    - Measure the amount of bandwidth used between two points in the network.
    - Track which circuits the bandwidth is used on, using a reservation system.
    - Deny additional reservations when the remaining bandwidth is insufficient.
    - And hopefully find a different path which DOES have sufficient bandwidth.
  - Bandwidth is measured at the MPLS Label Switched Path (LSP).
    - Each LSP is configured with the amount of bandwidth traveling across it.
    - The RSVP protocol maps each individual MPLS LSP onto RSVP speaking circuist, and reserves the amount of bandwidth configured for those LSPs.

MPLS LSP Bandwidth Measurement

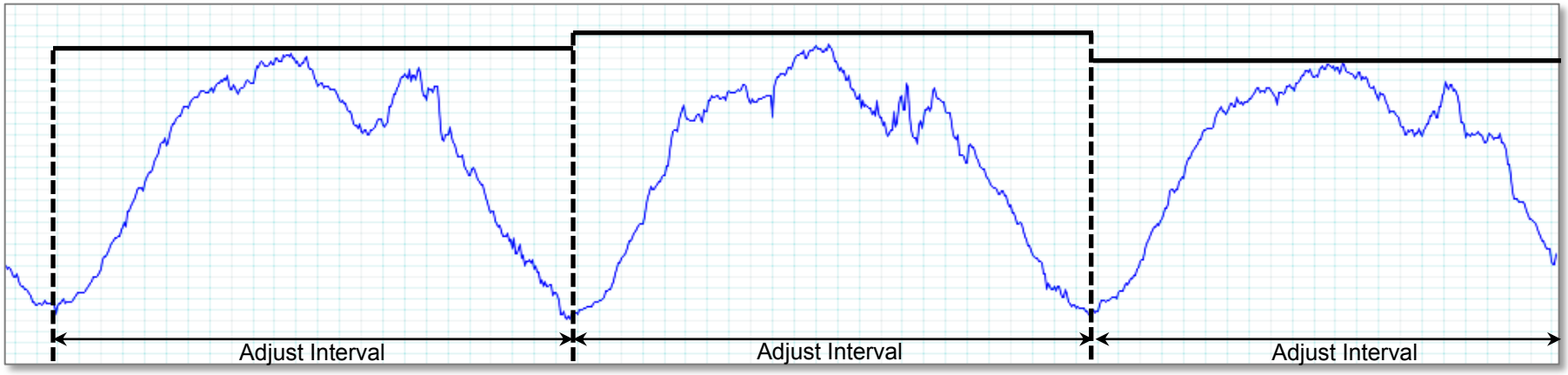
- So how do you configure the bandwidth on a particular LSP?
  - After all, IP networks are dynamic and packet switched.
  - Bandwidth usage can change in and instant, and be unpredictable.
- There are two main ways to accomplish this:
  - Offline Calculation
    - Calculation which occurs outside of the router, typically based on some type of “bandwidth modeling”, and often using a third party script or tool.
    - This is how RSVP-TE was first implemented, and is still commonly used today by many large networks and early MPLS adopters.
  - Auto-Bandwidth
    - The bandwidth value is calculated on the routers themselves, by periodically measuring how much traffic is actually forwarding over the LSPs.

Offline Calculation vs. Auto-Bandwidth

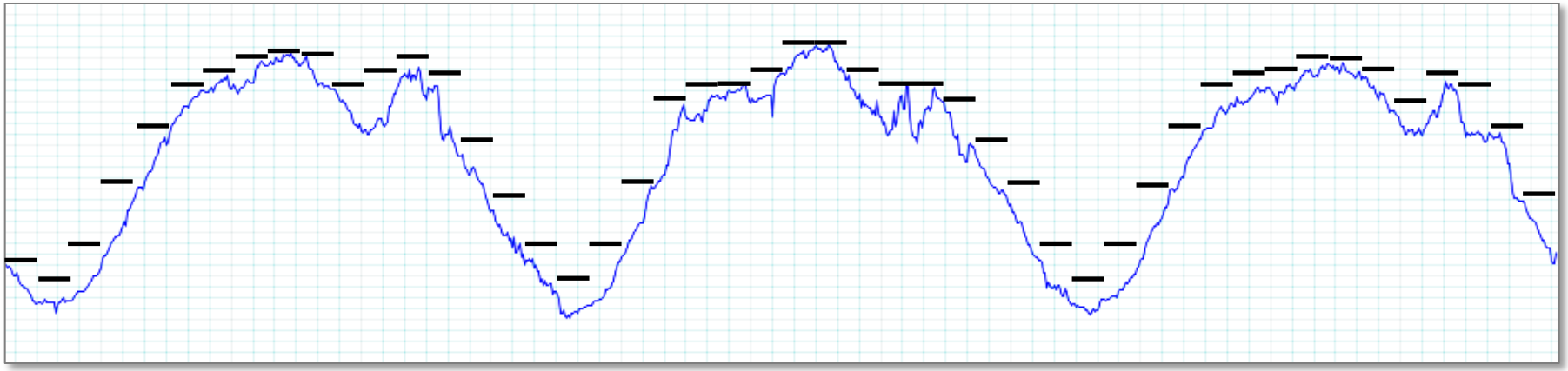
- **Offline Calculation**
  - You can implement any modeling algorithm you'd like.
  - Some extremely complex LSP modeling software is available from a variety of vendors, allowing you to do very detailed LSP planning.
  - But you have to either write the software yourself, or buy it.
- **Auto-Bandwidth**
  - Because it runs directly on the routers, it can respond to changing traffic conditions much more rapidly, and with less overhead.
    - Most offline calculation systems expect very stable traffic patterns.
    - Unusual traffic spikes or shifts can cause congestion, or inefficient bandwidth use.
  - Easier to implement (just turn the knob on your router, it's free).
  - But you're constrained to the algorithms provided by your vendor.

LSP Bandwidth Measurement Examples

24 Hour Adjust Interval



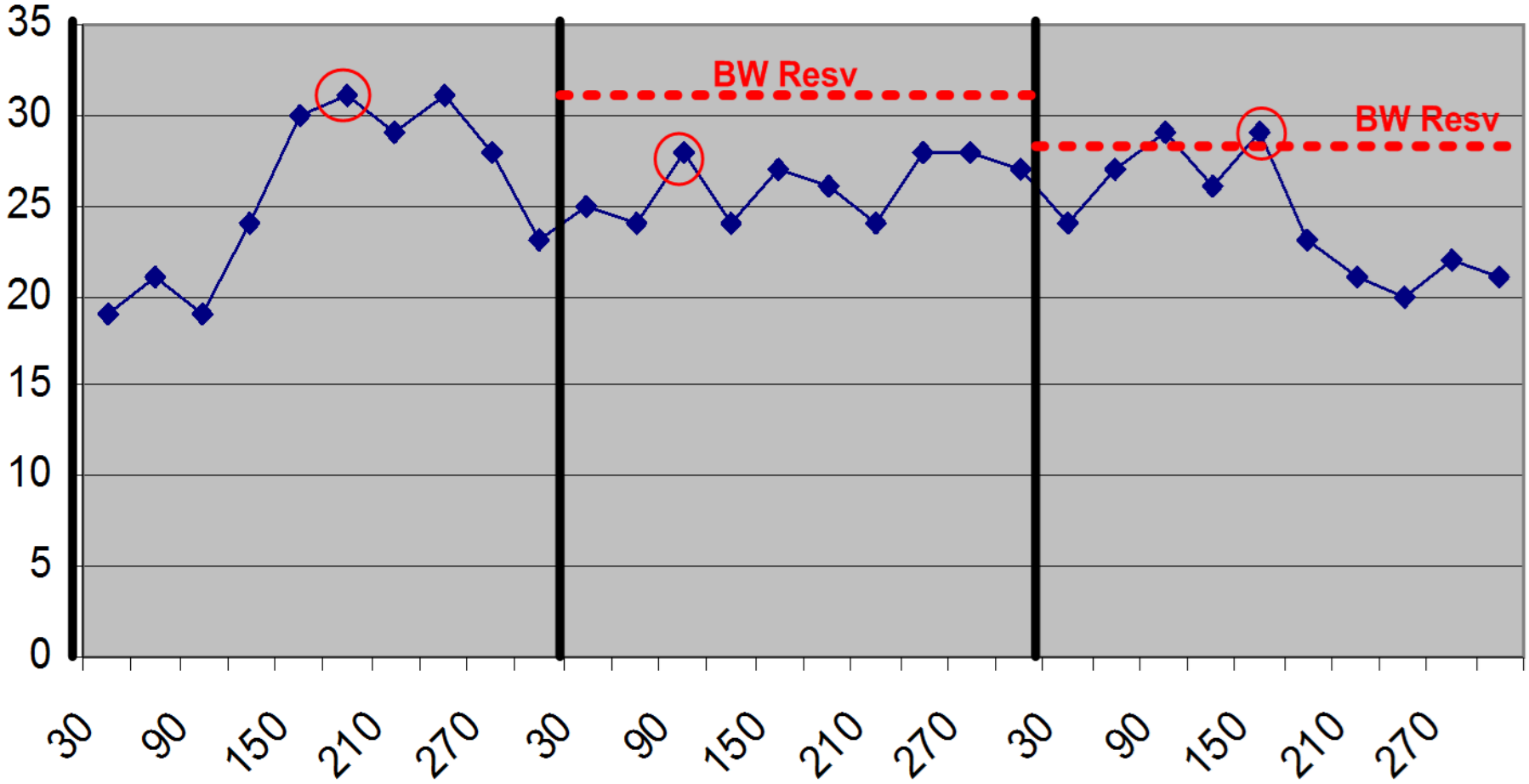
1.5 Hour Adjust Interval – Leads to more efficient bandwidth use



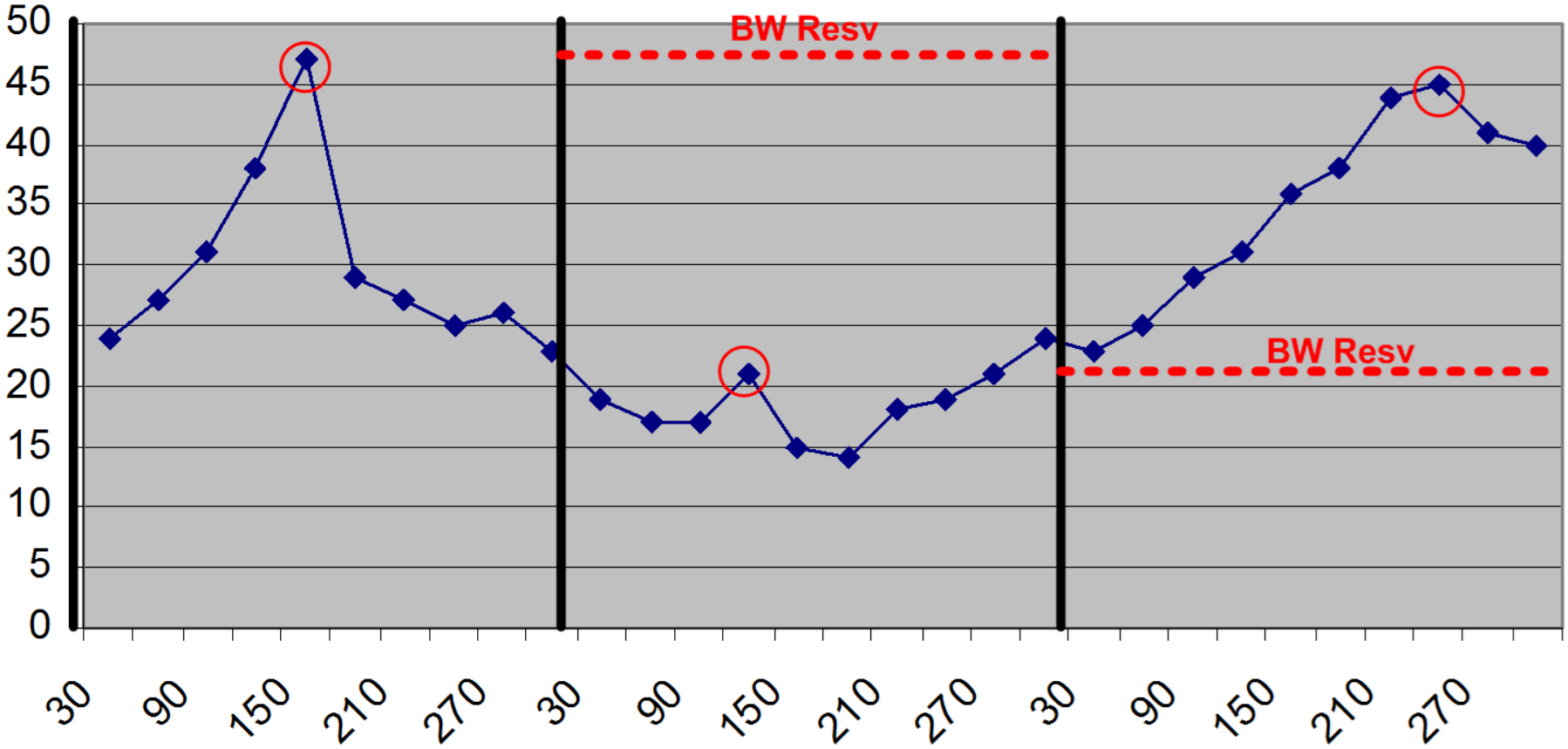
How Does Auto-Bandwidth Work?

- Auto-Bandwidth is an entirely router-specific behavior.
  - Thus the algorithms can be completely different between vendors.
  - It just so happens that both Cisco and Juniper implement it in much the same way.
- Auto-Bandwidth performs the following basic steps:
  1. Every “Statistics Interval”, bandwidth over an LSP is measured.
     - For example, you might configure this to every 60 seconds.
  2. Every “Adjust Interval”, the largest sample from the process above is used to calculate the new LSP bandwidth.
     - For example, you might use 5 samples and adjust every 300 seconds.
  3. If the change is larger than a user configured minimum “Adjust Threshold”, the new bandwidth value is re-signaled across RSVP.
     - Ideally using a make-before-break configuration, to signal the new LSP with the new bandwidth value before tearing down the old one.

When Auto-Bandwidth Works Well



When Auto-Bandwidth Doesn't Work Well



Overflow and Underflow

- Another common feature is called “Overflow” and “Underflow”.
  - If the difference between the signaled and measured LSP bandwidths exceeds a configurable percentage for a configurable number of Statistics Samples, kick off an Adjust event before the normal Adjust Interval timer would have done so.
  - “Overflow” detects increases in traffic rates, “Underflow” detects decreases in traffic rates.
  - The goal is to allow operators to configure a longer Adjust Interval (to reduce re-signaling), yet still react to rapidly changing traffic conditions.
- WARNING: Some vendors don’t support Underflow.
  - Support in IOS XR and JUNOS as of 11.4+

Where It All Goes Wrong, A.K.A.
You Knew It Wasn't Going To Be That Easy

RSVP and Changing Traffic Patterns

- RSVP-TE is very good at adapting to changing network conditions, such as circuit failures or other capacity changes.
- But it is very bad at adapting to changing traffic ***destinations***.
  - That is, traffic which was previously destined for router A is now destined for router B.
  - On an IP network, this is typically the result of a change in the BGP routing. For example:
    - An eBGP neighbor flaps on router A, and the traffic moves to router B.
    - A circuit flaps, and the resulting IGP cost change modifies the BGP best path selection, causing traffic to prefer the exit learned from router B.
  - Every time this happens, the old LSP must be resized down, and the new LSP must be resized up. This can take up to an Adjust Interval amount of time to happen, with potential congestion in the mean time.

The Problem With Overflow/Underflow

- Overflow/Underflow can actually cause many problems.
  - Consider what happens during a traffic destination shift, where LSP A must be sized down, and LSP B must be sized up.
  - If your router doesn't support Underflow (e.g. Juniper, Cisco IOS, etc), you can only react to the increasing LSP, but can't reclaim the bandwidth from the newly decreased LSP.
  - This creates inefficient routing (or worse) for an entire Adjust Interval period, which also makes a high Adjust Interval a bad idea.
- They also often don't create any detection optimization at all.
  - If you weren't going to exceed the minimum threshold of change in the first place, an RSVP re-signaling would never have occurred.
  - You could have just set your Adjust Interval to the lower value to begin with, for the same results in bandwidth change detection.

MPLS LSPs Don't Just Create Themselves

- Unlike some other protocols, MPLS isn't entirely automatic.
  - There are no protocols to auto-discover MPLS speaking nodes.
    - The MPLS "protocols" just exchange label values for the configured LSPs.
    - But they have no involvement in the creation of the LSPs themselves.
  - Building the full mesh of LSPs is an exercise left to the operator.
    - Essentially this means operator supplied scripts are a necessity.
    - Or else an operator purchased commercial software solution.
      - Examples include WANDL, Cariden, etc.
- Some vendors offer some ***very*** basic Auto-Mesh capabilities.
  - For example, Cisco IOS can auto-create a mesh of LSPs from a template, using a list of router IPs supplied via an access-list.
  - But this leaves you no way to control a specific LSP configuration.
    - Oh, and if you want to remove a node from the mesh, you have to remove the entire ACL, bringing down every dynamic auto-mesh LSP on the box.

Large LSPs Can't Fit Down Small Pipes

- An LSP can only be moved as an atomic unit.
  - So if you have large LSPs relative to the size of the circuits they're traveling, efficient packing becomes a serious problem.
  - For example, imagine you have 3 x 6 Gbps LSPs and 2 x 10G circuits.
    - 3 x 6 Gbps = 18 Gbps down 20 Gbps of pipe, it should fit right?
    - But you'll actually only be able to fit 2 of the 3 LSPs above.
    - The third LSP will have to find another longer path, if one exists at all.
    - And your two circuits above will be left with 4 Gbps of unfilled capacity.
  - Another example, say you have a mix of 10G and 2.5G circuits:
    - A 3 Gbps LSP will never be able to fit down a 2.5G circuit.
- A workaround is to load balance over multiple parallel LSPs
  - Instead of having 3 x 6 Gbps LSPs, you could have 9 x 2 Gbps LSPs.
    - But so far no router vendor auto-mesh system supports parallel LSPs.
    - You can also run into a maximum number of ECMP'd paths limitation.

Auto-Bandwidth Behavior Under Stress

- Another issue is the behavior of auto-bandwidth systems when RSVP cannot find sufficient bandwidth on any link.
  - For example, consider the previous example of poor LSP packing.
  - Sometimes, when a new bandwidth reservation cannot be met, the LSP value is simple not updated even though traffic has increased.
    - This leads to non-RSVP accounted traffic on the network, which often causes silent congestion where RSVP thinks there should be none.
  - Under other conditions, an updated bandwidth reservation which cannot be met causes an existing LSP to be torn down completely.
    - In a parallel-LSP scenario, traffic immediately shifts to the remaining LSPs.
    - In the short term, non-accounted traffic is now traveling over these links.
    - But even worse, the remaining LSPs have now become larger, and are also likely to fail to find sufficient bandwidth. This leads to a pathological behavior where all of the parallel LSPs fail, resulting in a single large LSP which can never find bandwidth, without manual human intervention.

What About An Intelligent Auto-Mesh Script?

- Imagine you have to implement your own LSP Auto-Mesh script anyways, to support configuring multiple parallel LSPs.
- Could you make it automatically “fork” LSPs which get too big?
  - For example, this could be done automatically with a Juniper Event Script.
  - When a sufficiently large LSP is detected, the router could automatically configure a new parallel LSP.
- But not so fast, there are gotchas here too:
  - Newly created auto-bandwidth LSPs initially signal with a bandwidth value of 0, even though traffic is immediately load-balanced over them.
    - And there is no way to override this with current Cisco/Juniper implementations.
  - This leads to incorrect RSVP bandwidth measures, non-RSVP accounted traffic on the network, and congestion often results.
  - And under the previously explained “pathological LSP collapse” scenario, creating a pile of new LSPs would make the situation even worse.

Auto-Bandwidth and Congestion

- Auto-Bandwidth doesn't know anything about congestion.
  - Imagine that for some reason (such as the many examples provided) a link becomes congested, even though RSVP is unaware of it.
  - Packet loss causes TCP to throttle back, and the IP traffic goes down.
  - Auto-Bandwidth adapts to this new rate, and thinks everything is fine.
  - This can lead to sustained congestion, requiring manual intervention.
- Also be careful of routers which can't "see" layer 2 overhead.
  - For example, Juniper M/T/MX series routers can't measure any of it.
    - But every vendor misses at least some of the L2 per-packet overhead.
  - A 28-byte UDP packet consumes 84 bytes over the wire on Ethernet.
  - But if your router can't measure this, a Denial of Service attack (or even a shift of traffic with small packets, such as TCP ACKs) can cause a link to become congested even though RSVP thinks it's fine.

Some Practical Suggestions

- Well designed RSVP-TE Auto-Bandwidth can work well.
  - In fact, most of the time it works great.
  - But a wide variety of conditions exist where human intervention is required to resolve pathological issues.
- Don’t bother using Overflow
  - Especially if you don’t also have Underflow.
- Beware of vendor bugs.
  - We’ve seen MANY conditions under which auto-bandwidth measurements can be affected by router vendor bugs.
- Careful monitoring of your network is still required.

Nag Your Vendors For

- Don’t forget to nag your vendors for:
  - Underflow as well as Overflow support, if you really want to use it.
  - An adjust-threshold minimum in bytes as well as percent.
    - So you don’t have to re-signal every LSP that changes from 256Kbps to 512Kbps every adjust-interval.
  - Better built-in LSP Auto-Mesh capabilities.
  - A feature to automatically “fork” large LSPs past a certain size.
  - A better way to display these forked LSPs in your “show route” output so you don’t end up with a screen full of LSP next-hops on every route when using parallel LSPs.
  - The ability to set an “initial bandwidth reservation” for new LSPs.
  - An operational mode command to manually adjust the bandwidth values of an LSP, if an operator supplied auto-mesh script is required to implement LSP forking.

Send questions, comments, complaints to:

Richard A Steenbergen <ras@nlayer.net>