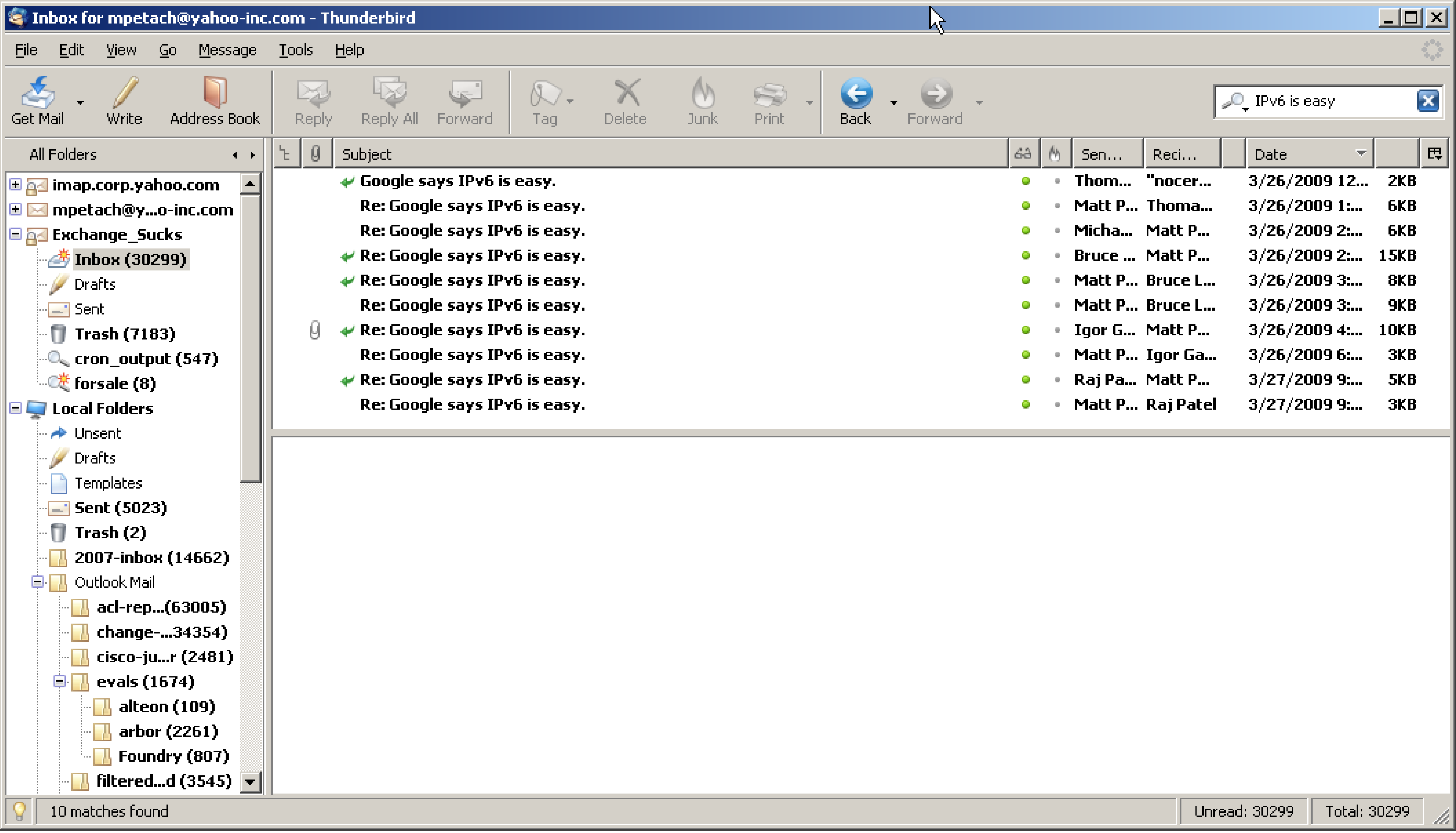
IPv6 backbone deployment @Y!

Matt Petach

Oct 20th, 2009

The challenge—the week after the 2009 IPv6 Implementer's Conference at Google, this was my inbox…



Subject	Sen...	Reci...	Date	
Google says IPv6 is easy.	Thom...	"nocer...	3/26/2009 12...	2KB
Re: Google says IPv6 is easy.	Matt P...	Thoma...	3/26/2009 1:...	6KB
Re: Google says IPv6 is easy.	Micha...	Matt P...	3/26/2009 2:...	6KB
Re: Google says IPv6 is easy.	Bruce ...	Matt P...	3/26/2009 2:...	15KB
Re: Google says IPv6 is easy.	Matt P...	Bruce L...	3/26/2009 3:...	8KB
Re: Google says IPv6 is easy.	Matt P...	Bruce L...	3/26/2009 3:...	9KB
Re: Google says IPv6 is easy.	Igor G...	Matt P...	3/26/2009 4:...	10KB
Re: Google says IPv6 is easy.	Matt P...	Igor Ga...	3/26/2009 6:...	3KB
Re: Google says IPv6 is easy.	Raj Pa...	Matt P...	3/27/2009 9:...	5KB
Re: Google says IPv6 is easy.	Matt P...	Raj Patel	3/27/2009 9:...	3KB

10 matches found

Unread: 30299

Total: 30299

Google’s done it—why haven’t we?

- At the 2009 IPv6 Implementer’s conference at Google (many thanks to Lorenzo Colitti and team for hosting it!) the chicken and egg challenge was again issued: ‘more people will deploy IPv6 when more content is available on IPv6’
- Google made it sound fairly easy to deploy, as did other presenters
- We’d already made the transition on the backbone from OSPFv2 to IS-IS, so we knew we were in good shape for supporting IPv6 on the IGP side.
- Recurring theme during the implementer’s conference was that support systems, tools, and education were the major challenges—and old hardware/software that didn’t support v6 well.

The Goal

- Get the Yahoo backbone running native IPv6 worldwide before the joint NANOG+ARIN meeting in October, 2009
- Important caveats!
  - Can’t break existing services running on IPv4!!
  - Must be on existing infrastructure—no fair deploying a parallel network
  - Native means just that—no fair cheating with 6PE or other overlays!
  - Can’t break existing toolsets or monitoring infrastructure
  - Can’t circumvent or break existing security models

The Plan

- Establish a sane, supportable, extensible long-term IPv6 address allocation plan and framework that would last at least the next two decades based on the growth rate from the first ten years of the company.
- Roll out an initial testbed to get employees comfortable with IPv6 and allow for internal training
- Write scripts to automate allocation of IPs for loopbacks, internal links, peering subnets, datacenter blocks, etc.
- Write scripts to generate config snippets for IPv6 backbone links, iBGP sessions, loopback ACLs, etc.
- Write scripts to validate IPv4 and IPv6 IGP topologies are completely congruent (we decided to run single-topology IS-IS on the backbone)
- Make it so (actually run all the scripts mentioned above!)

The Timeline

- October 2008: IPv6 testing begins with 2 routers connected via 6PE; initial code testing and validation begins.
- March-April 2009: IM conversations, late-night phone calls, and lots of email back and forth while we establish the IPv6 address allocation framework to allow us to grow the company for the next 20 years without having to tear it out and start over.
- March-July 2009: Stable router software with IPv6 support is released by vendor, passed internal certification, and is rolled out onto the backbone
- August 10th, 2009: we meet for two days to plan the actual deployment process, week by week, night by night. We start writing the support scripts, and CMRs are submitted. We start testing existing tools against test IPv6 router to make sure they won't blow up when they come face-to-face with IPv6 addresses.
- Sept 14, 2009, we bring up the first native backbone link on v6; just a single point-to-point OC192, with IS-IS on it, with new loopback ACL and IPv6 addresses on loopbacks; no IBGP yet; we run it for 24 hours to make sure nothing unexpected materializes.

The Timeline, continued

- Sept 16: new loopback ACLs and loopback addresses are pushed to half the routers worldwide.
- Sept 21: new loopback ACLs and loopback addresses pushed to the other half of the routers, worldwide
- Sept 23: all link (interface) addresses pushed to half the routers worldwide
- Sept 28: all remaining link (interface) addresses pushed to routers; scripted v6 pings done on all links to validate IPv6 reachability.
- Sept 30: IPv4/IPv6 congruency script is run; 3 link errors are found and fixed; script is re-run to ensure 100% IGP congruency.
- Oct 5: IPv6 policies and IPv6 iBGP configs pushed to half the routers
- Oct 7: v6 policies and v6 iBGP configs pushed to remaining routers worldwide.
- Oct 8, 12, 13, 14: configured IPv6 peering addresses at all major public peering points worldwide.

Conclusion

- TESTING! Don’t always trust your router vendor’s marketing claims; validate and certify the functionality *you* need within your environment. Don’t trust that the vendor tested their code in an environment that matches what your routers face every day.
- Most of the thinking time involved developing a sane, scaleable IPv6 address allocation plan
  - Don’t skimp on this part; if you do it right, you can automate the allocation of addresses throughout your network, to the point where a perl script can tell you what address will be on what router at a given site, even if the site won’t be built for the next ten years.
- When you have a sane addressing plan, most of the work can be scripted—avoids human error, speeds up deployment drastically
- Likewise, with a coherent, consistent plan, validation tests can be scripted as well to ensure there’s no errors or gaps in deployment

Conclusion

- Actual deployment time, even for a worldwide backbone spanning ¾ of the globe, is surprisingly short; took only 4 weeks to deploy, from the first trial link, to the final public peering addresses.
- From the network backbone perspective, IPv6 is easy.
- Datacenter support, application support, and the rest of the stack, not so much so…

Thanks!

- Lorenzo and the rest of the IPv6 Implementer's Conference group
- Mike, Martin, and everyone else behind TunnelBroker.Net
  - http://tunnelbroker.net/ -- if you haven't signed up yet and started working with IPv6 at home, get off your backside and just do it! This resource really rocks, and really does give you a good way to get your feet wet and learn about working with IPv6.
- All the vendors who include IPv6 support in their base OS release!
  - And a big boo-hiss to those vendors who don't—you're not helping get IPv6 out there by doing that. It really needs to be ubiquitous to help pave the way for widespread rollouts!