IPv6 deployment at Google

Lorenzo Colitti, Angus Lees
{lorenzo,alees}@google.com

Why?

Why IPv6?

- When the day comes that users only have IPv6, Google needs to be there

- If we can serve our users better over IPv6, then we will
  - IPv6 can have lower latency and packet loss
    - ... and we have user reports to prove it
  - AJAX applications break behind excessive NAT
    - Too many connections exhaust public IP port space
  - NAT traversal complicates apps like Google Talk
    - Developer time better spent elsewhere

The reasoning is simple

- IPv6 is going to happen
  - RIR pool exhaustion Dec 2011
  - IPv6 the only solution that really makes sense
- Not a question of if, but when
- We might as well start now
  - Early adoption critical for service quality in the future
  - Act now to save money later
  - It's not rocket science, but it takes time!

IPv6 is Not Rocket Science™

What worked for us

- IPv6 at Google started as a 20% project
  - Like gmail and news :-)
- Built a pilot network
  - Lab testing, engineering, pilot deployment
  - Proved architecture at internal IPv6 conference
  - Once network was up, applications followed
- Scaled up architecture, productionized
  - Monitoring, documentation, support, ...

You can do it too

- Tap enthusiasm
  - 20% project had incredible influx of contributors
- Make it easy for contributors to get initial results
  - A pilot network is not expensive
  - Once network is up, internal applications will follow
- Do it in stages
  - v6 doesn't have to be as capable as v4 on day one!
  - Make slow, steady progress: operators are cautious
- Remember: it's not rocket science. It just takes time

Lessons Learned

Operations: be consistent

- Dispel notion that IPv6 is "experimental"
- IPv6 must be a production service
  - Monitored
  - Supported
  - Designed to the same quality standards as IPv4
- How to achieve this?
  - Make NOC aware of IPv6
  - Scale down, but don't skimp
  - Design as closely to IPv4 as possible
    - Make the principle of least surprise work for you

Device support: adequate

- Feature parity not there yet
  - No MPLS TE for IPv6
  - No extension header filtering in hardware
  - NAT-PT temperamental
  - No 6to4/Teredo in hardware
  - Load-balancing not mature yet
- Reliability not quite ironed out
  - Load balancer memory leaks
  - Router crashes (fixed on same day)
  - None of these are showstoppers
  - But might want to start with dedicated devices :-)

Internetworking: patchy

- Tunnels increase latency and complicate debugging
  - Avoid them, especially for interdomain traffic!
- IPv6 interdomain routing patchy
  - Indiscriminate transit
    - Slows convergence, increases RTT
  - Blackholing, incomplete visibility, ...
- Peering, peering, peering
  - Quality of deployed IPv6 highly variable
  - Interconnecting production-ready networks creates production-ready Internet

The real challenge

- How do we adopt IPv6 while maintaining Google quality of service?
- `www.google.com IN AAAA` not the solution today
  - Lower reliability and higher latency for many users
  - Partial/total breakage for small percentage of users
    - Our users rely on us
    - Breakage is unacceptable!
    - Bad IPv6 worse than no IPv6 (timeouts, ...)
- Bilateral relationships the way forward?
  - Directly connect IPv6 clients to IPv6 content

The way forward?

IPv6-only networks with NAT-PT

- Client goes IPv6-only to ease address exhaustion
  - NAT-PT provides connectivity to IPv4 content
- Solves chicken and egg problem
  - Clients can move to v6 without waiting for content
  - When other end gets v6, NAT goes away
- Transforms the content deployment problem into an application porting problem
  - More and more applications run inside the browser
  - Enterprises might only need a few applications
  - You might not need v4 on the client at all!

Undeprecate RFC 2766?

- NAT-PT is deprecated by RFC 4966
  - All the drawbacks in 4966 are present in v4 NAT too
    - But this is how the IPv4 Internet works today
- We need a bare-bones NAT-PT standard
  - The standard should not require host support
  - It's a little late to change host stacks now
- NAT breaks end-to-end. But:
  - Better IPv6 end-to-end+NAT-PT than just IPv4 NAT
  - Post IPv4 runout, new hosts will be NATed anyway

IPv4-style multihoming

- Multiple-address multihoming doesn't work
  - Failover breaks TCP connections
- HIP/SHIM6 not equivalent to IPv4-style multihoming
  - Failover works, but new connections see timeouts
  - No load-balancing or traffic engineering
  - Doesn't sound convincing for mission-critical applications
- IPv6 deployment must not block on IPv6 multihoming!
- /48 needs to be accepted in DFZ

Deployment-friendly licensing

- Some vendors require software licenses for IPv6
  - Price is trivial, but paperwork and approvals are a barrier to entry
    - Turning a lab into a real deployment will require hardware upgrades anyway
  - Spending $500k in licenses to roll out IPv6 in a large network all at once likely to be hard sell

- Charging extra for IPv6 support *will* hinder adoption
  - OS manufacturers don't charge extra for IPv6...

How can the IETF help?

- Bare-bones NAT-PT for IPv6-only networks
  - No change in host stacks
  - Minimalistic, simple to standardize / implement
- Allow /127s on point-to-point links
  - Currently forbidden by RFC 4291
- Finalize VRRP for IPv6
  - Current NUD not fast enough for production failover
  - Last VRRP draft Feb 2006
- Allow multihoming using /48 prefixes

Questions?

Lorenzo Colitti, Angus Lees
{lorenzo,alees}@google.com