



IPv6 at Google

Erik Kline, Lorenzo Colitti
{ek,lorenzo}@google.com

Why deploy 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...

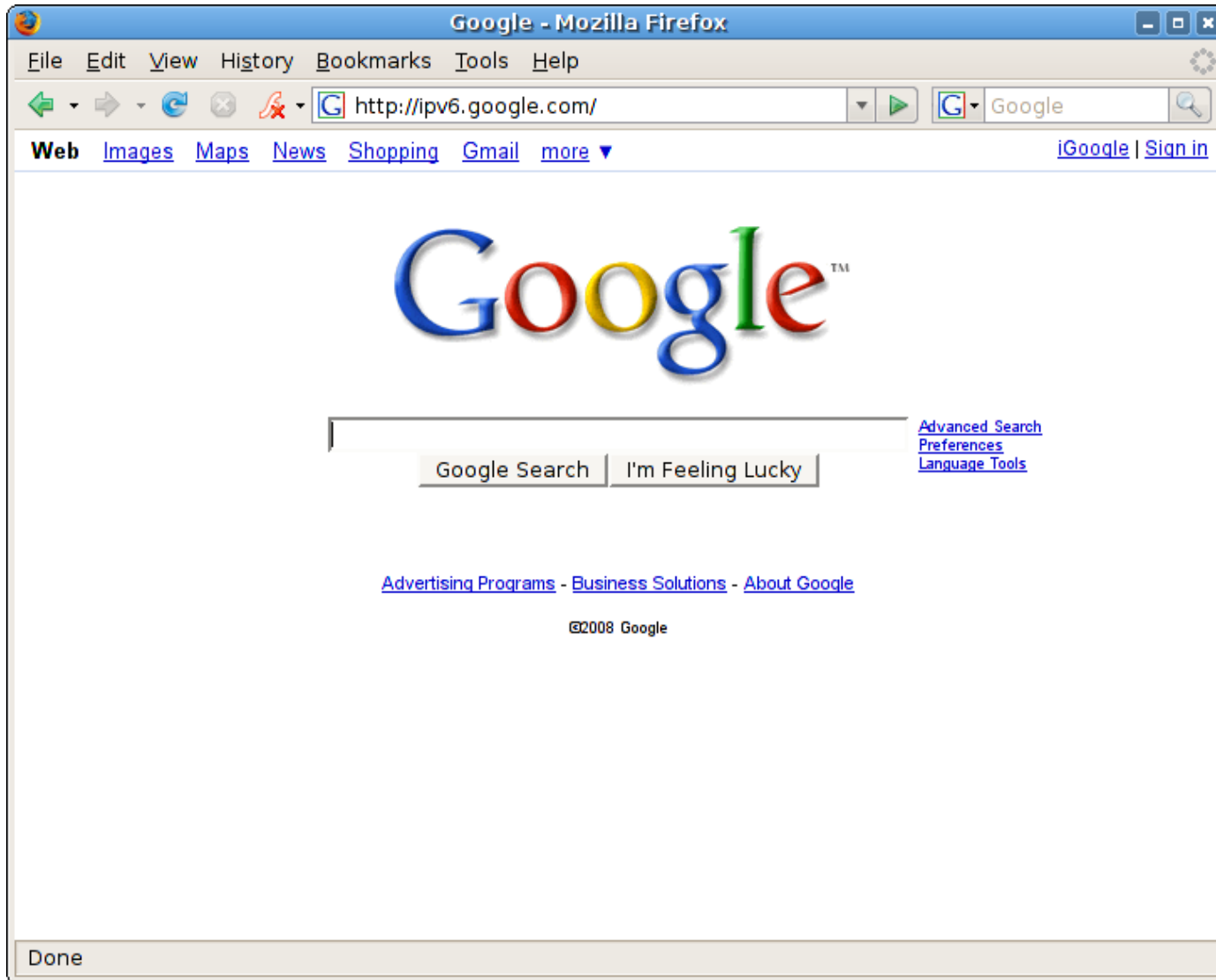
- IPv6 is going to happen
 - RIR pool exhaustion Dec 2011
 - IPv6 the only solution that really makes sense
- Not a question of if, but when
- We might as well start now
 - Early adoption critical for service quality in the future
 - Act now to save money later
 - It's not rocket science, but it takes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

Google and IPv6

Google involvement in IPv6

- Google IPv6 conference, January 2008
- Community involvement
 - IPv6 WG, IETF, RIR meetings
- IPv6-accessible websearch launched March 2008
 - Only major search engine so far
- More to come...

ipv6.google.com



(If you can see this, you're using IPv6)

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
 - Started with dedicated devices: low risk, fast rollout
 - Dual-stack only solution in long run - operational complexity too great

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 nor hard to build
 - 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

Where do we go from here?

What do we need?

- Standards work
 - /127 on point-to-point links
 - NAT64 for IPv6-only client networks
 - IPv6 VRRP for production-grade load-balancing
- Policy
 - IPv4-style multihoming for end sites (allow /48 in DFZ)
- Deployment!
 - Deploy IPv6 in *your* network
 - It's not rocket science
 - At the very least, offer 6to4/Teredo relays

IPv6 licenses included by default

- Some vendors require software licenses for IPv6
- Suppose it's \$10k per router:
 - Red tape blocks initial experimentation / deployment
 - Need to cut \$30k PO to try IPv6 on 3 routers
 - Bulk upgrade price blocks full rollouts
 - Have 100 routers? That will be \$1M, please...
- Charging separately for IPv6 support *will* hinder adoption
 - Absorb cost by raising price of base image or HW
 - The Internet will thank you

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!

A possible solution?

- Ballpark numbers (your.org, wikimedia, ...):
 - ~0.1 - 0.2 % of users have working IPv6
 - ~0.01 - 0.1 % of users have **broken** IPv6
 - Broken IPv6 much worse than no IPv6!
- Bilateral cooperation
 - If content providers interconnect with user networks:
 - QoS can be guaranteed, problems can be fixed
 - Both networks gain operational experience
 - Production-quality services can be provided
 - Hand out AAAA to networks with good connectivity



Questions?

Erik Kline, Lorenzo Colitti

{ek,lorenzo}@google.com