A strategy for IPv6 adoption

Lorenzo Colitti
lorenzo@google.com

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, we will
  - IPv6 can have lower latency and packet loss
    - ... and we have user reports to prove it
  - AJAX applications break behind excessive NAT
    - Connections exhaust public IP port space
  - NAT traversal complicates apps like Google Talk
    - Developer time better spent elsewhere
- IPv6 is good for the Internet, and we want to help

What we have done so far

- IPv6 websites
  - ipv6.google.com (Mar 2008)
  - ipv6.google.cn (Aug 2008)
  - ipv6.google.co.jp (Oct 2008)
- IPv6 network
- IPv6 evangelism
  - Google IPv6 conference (Jan 2008)
  - IETF panels, blackout sessions, ...

The root of the problem

- Nash equilibrium for IPv6 adoption is to do nothing
  - Wait for everyone else
- Chicken and egg problem
  - ISPs say there is no content
  - Content providers say there are no users
- All the same, the writing is on the wall
- How do we break the cycle?

Creating a chicken

- If content providers offer content over IPv6, that might provide an incentive for clients
  - Even better if the content is somehow "better" than that available over IPv4
- Unfortunately, there's another problem for IPv6:
  - Low adoption causes low traffic
  - Low traffic leads to bad connectivity
  - Bad connectivity hampers adoption
- Basic problem: how do we offer IPv6 content without harming user experience?

No www.google.com AAAA

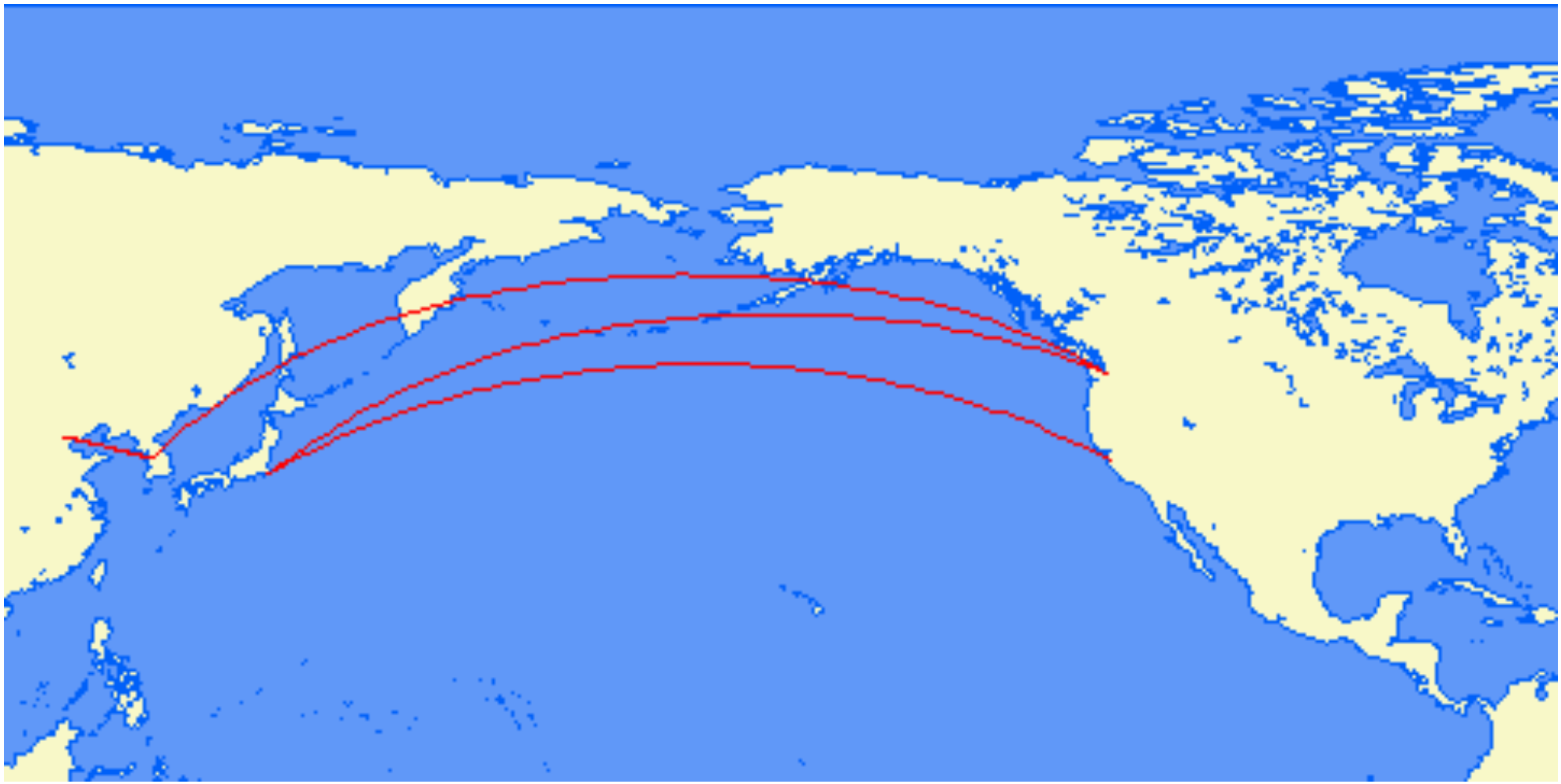
- We can't enable IPv6 for www.google.com today
  - 1 in 10000 broken users is still too many
    - Google has a lot of users
    - If you have a problem, you might want to reach Google to see how to fix it :-)
  - 150ms of RTT penalty doesn't help
    - Like going from Europe to the West coast!
- So what do we do?
- Let's look at the problems in more detail first

IPv6 connectivity problems

So what's the problem exactly?

- Symptoms:
  - Many IPv6 connections slower than IPv4
  - Some IPv6 connections fail altogether
- Not protocol problems, but deployment problems
  - IPv6 not inherently any less reliable than IPv6
- Causes:
  - Long paths
  - Non-optimal routing
  - Broken middleboxes
  - MTU issues

West coast to China, 413 ms



We don't want to do this to our users!

Long AS paths

- Long AS paths are bad
  - Slow convergence, high latency, near-impossible to debug and fix
- A couple of examples:
  - 3257 2497 4725 6939 23911 4538 23910 18011
    - 3257 peers with 2497
  - 6939 2497 4725 2500 7660 2907 11537 7539 17419
    - 6939 peers with 2497, 7660, 11537, ...
  - See Bernhard Schmidt's RIPE56 presentation for more

Long AS paths

- Causes
  - Interdomain routing over tunnels
  - Indiscriminate transit
  - Prefixes without real upstreams
- Solution: don't use these routes, and don't take transit
  - Better no connectivity than bad connectivity
    - Transit can't live with partial routing, but we can
    - For global connectivity, there's always IPv4
- If the ASes with these prefixes peer with us or take transit, we will see them again

Ashburn to Ashburn, 100ms

- Google peers with AS X in Amsterdam and Ashburn
- X sends all traffic to Google through Amsterdam
- US customers of X cross Atlantic twice
- X is unresponsive when asked to fix

```
64 bytes from 2001:504:0:2:X:X:1: icmp_seq=62 ttl=59 time=317 ms
64 bytes from 2001:504:0:2:X:X:1: icmp_seq=63 ttl=59 time=305 ms
                    <reset BGP session in Amsterdam>
64 bytes from 2001:504:0:2:X:X:1: icmp_seq=64 ttl=60 time=116 ms
64 bytes from 2001:504:0:2:X:X:1: icmp_seq=65 ttl=60 time=103 ms
```

Non-optimal routing

- Lowest-cost routing => use as few links as possible
  - But when there is no traffic, this breaks down
  - No incentive to fix non-optimal routing
- But latency matters...
  - 50ms RTT: a small HTTP load takes 100ms
  - 400ms RTT: a small HTTP load takes 800ms
- IPv6 end-user networks have more interest in low latency than large ISPs

So, what we are doing?

Peering instead of transit

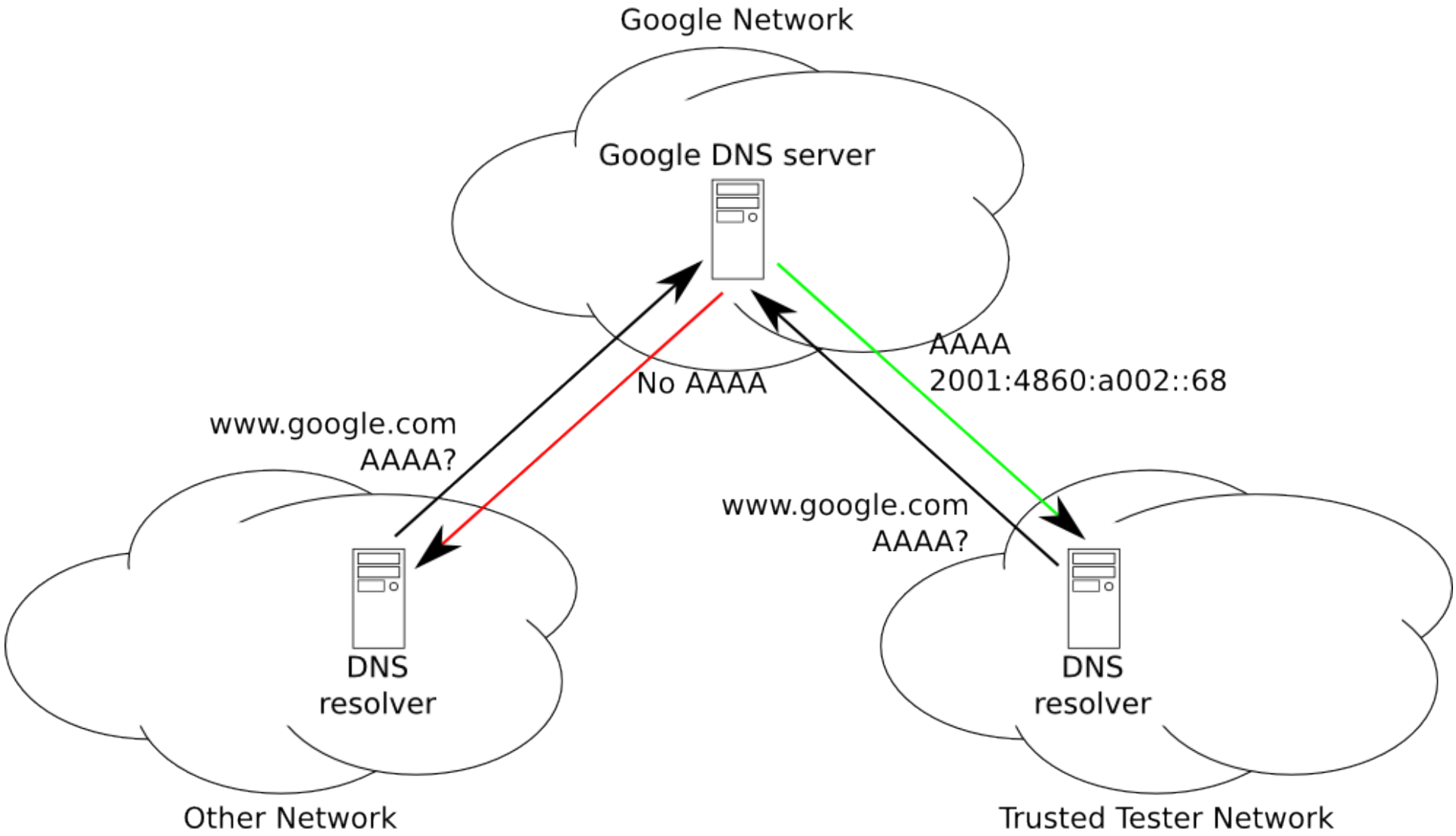
- Avoid bad routes by not taking transit
  - We don't have an IPv6 transit provider
  - But we peer with almost everybody
- Avoid suboptimal routing by peering with user networks directly
  - Guarantees better service and low latency
  - Since both networks care, IPv6 issues get fixed
- We're happy to peer with - or close - to you
  - Aggressive, user-driven rollout
  - Check peeringdb and/or email peering@

And what else?

IPv6 Trusted Tester program

- Enables IPv6 access to Google for selected networks
- IPv6 access to most Google web properties
  - www, mail, calendar, docs, ... (no youtube yet)
  - Which ones do *you* and your users want?
- Works by DNS resolver IPv4 address
  - If the user's DNS resolver is in a whitelist, it will receive AAAA answers
- Live, now, on the conference network
  - Did you notice?

IPv6 Trusted Tester program



Google

Being a Trusted Tester

- Requirements
  - Good IPv6 connectivity to Google
    - Two diverse peerings, or one peering and "good" transit
  - Production-quality IPv6 network
  - Commitment to fix user breakage and report any bugs you see
- Want to take part? Let us know!
  - Already have several networks signed up, but the more the merrier

Scaling it up

- Enabling IPv6 Trusted Testers by email doesn't scale
  - Hard to maintain 1000 networks manually
- Need a clear signal to say "we want IPv6 from you and will fix our users if they break"
- A possible signal: BGP communities
  - Tag your IPv4 resolver prefixes with a community
    - 15169:6666? IETF-standard value?
  - If IPv6 routing is good, can automatically whitelist
    - This will probably mean direct IPv6 peering
  - What do other content providers think?

A few more thoughts

On IPv6 licensing

- Some vendors charge separately for IPv6 support
- 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 *will* hinder adoption
  - Absorb cost by raising price of base image or HW
    - The Internet will thank you
  - The same goes for ISPs, exchanges, ...

On carrier-grade NAT

- Several proposals to maintain backwards compatibility with OSes that don't support IPv6
  - CGN, DS lite, A+P
- Are these really necessary?
  - Windows 98 EOL July 2006
  - Server logs say Win95/98/ME ~1% of all hits
    - Less for technical websites like RIPE NCC
    - What will it be in 3 years when IPv4 runs out?
- Are you sure you want to spend all this money on 1% of your users?

On porting applications

- Problem: many applications don't support IPv6
- Not as bad as you might think
  - IPv6 supported in all browser apps, bittorrent, ...
  - NAT-PT can take care of many of the rest
- But mostly:
  - IPv6-capable applications will only emerge when users and developers get IPv6 connections
  - If you want IPv6 support in applications, roll out IPv6 to users...

Questions?

Lorenzo Colitti
lorenzo@google.com