IPv6 transition experiences

Lorenzo Colitti

What works

Timeline

April 2005	Obtain and announce address space
...	...
July 2007	Network architecture and software engineering begin (20%)
January 2008	First pilot router. Google IPv6 conference, Google over IPv6 for attendees
March 2008	ipv6.google.com (IETF 72)
November 2008	First Google over IPv6 networks enabled. Google over IPv6 at RIPE / IETF / ...
January 2009	Google over IPv6 publicly available
March 2009	Google maps available over IPv6, 3x increase in traffic
August 2009	IPv6 enabled in Android (Droid and Nexus One)
February 2010	Youtube available over IPv6, 10x increase in traffic
March 2010	Backbone fully dual-stack. IPv6 in AppEngine
June 2010	Googlebot starts crawling IPv6

And all this with a small core team

Development strategy

- Gradual approach
  - Work from the outside, move in
  - First the load-balancer, then the frontend, then...
- "Address coercion" protects IPv4-only code from IPv6
  - Take IPv6 address
  - Remove user-modifiable bits
  - Hash into 224.0.0.0/3
- Sometimes not perfect
  - "Your last login was from 238.1.2.3"

Network design principles

- Make design as similar to IPv4 as possible
  - Principle of least surprise for NOC, other engineers, ...
- Dual stack everything
  - Scales better, no added maintenance / support load
  - Using IS-IS for IPv6? Might want to use it for IPv4
  - Using OSPFv3? Make sure implementation is proven
- Use IPv4 to carry IPv4 routes, IPv6 to carry IPv6 routes
  - Don't block convergence of one protocol on another
  - Avoid ::ffff:10.0.0.1 and ::10.0.0.1 as IPv6 next-hops

Testing and iteration

- Implementations mostly work, but will have bugs
  - Nobody has really kicked the tyres
- Don't expect something to work just because it's supported
- If you find a bug in the lab:
  - Report it, and keep testing!
  - There are many more bugs to find
  - We don't have time to fix them one by one
- Work around it in the design
  - If you get to something that is supportable, trial it
  - That will help you find the hard bugs

For example...

- If a firewall filter term has a 1-bit match in bits 32-64, and then term with a 2-bit match on bits 64-96, the second term will not match on hardware X on version Y
- In particular circumstances, FIB and RIB may get out of sync due to race conditions in pushing updates
- If DAD triggers due to an interface loop, it requires removing config from the interface and putting it back
- If a linux gets a packet too big on a receive-only interface with no route, it ignores it
- Are you going to find these in the lab?
    - We only saw the race condition after months in production in a fair number of datacenters

What's not working

Broken IPv6

- Clients try IPv6 first, but IPv6 not as reliable as IPv4
- Host-local errors
  - No IPv6 address, no default route, ...
  - Fast, no problem if application falls back (e.g., not Java)
- Network errors
  - Router replies to SYN packets with unreachables
  - Network spoofs RST packets
- Blackholing, MTU holes
  - Misbehaving router, packet loss in core
  - Misconfigured firewalls dropping ICMP

What's the damage?

- Local failure, RST: fast
- Unreachables: OS-dependent timeout
  - Windows: 20 seconds
  - Mac: 4 seconds
  - Linux: instant
- Blackholing similar (but Linux timeout is ~3 minutes)
- MTU holes: only some TCP stacks recover (in seconds)
- Even if failure is fast applications may have other limits
  - e.g., MSIE >= 7 gives up completely after 5 attempts

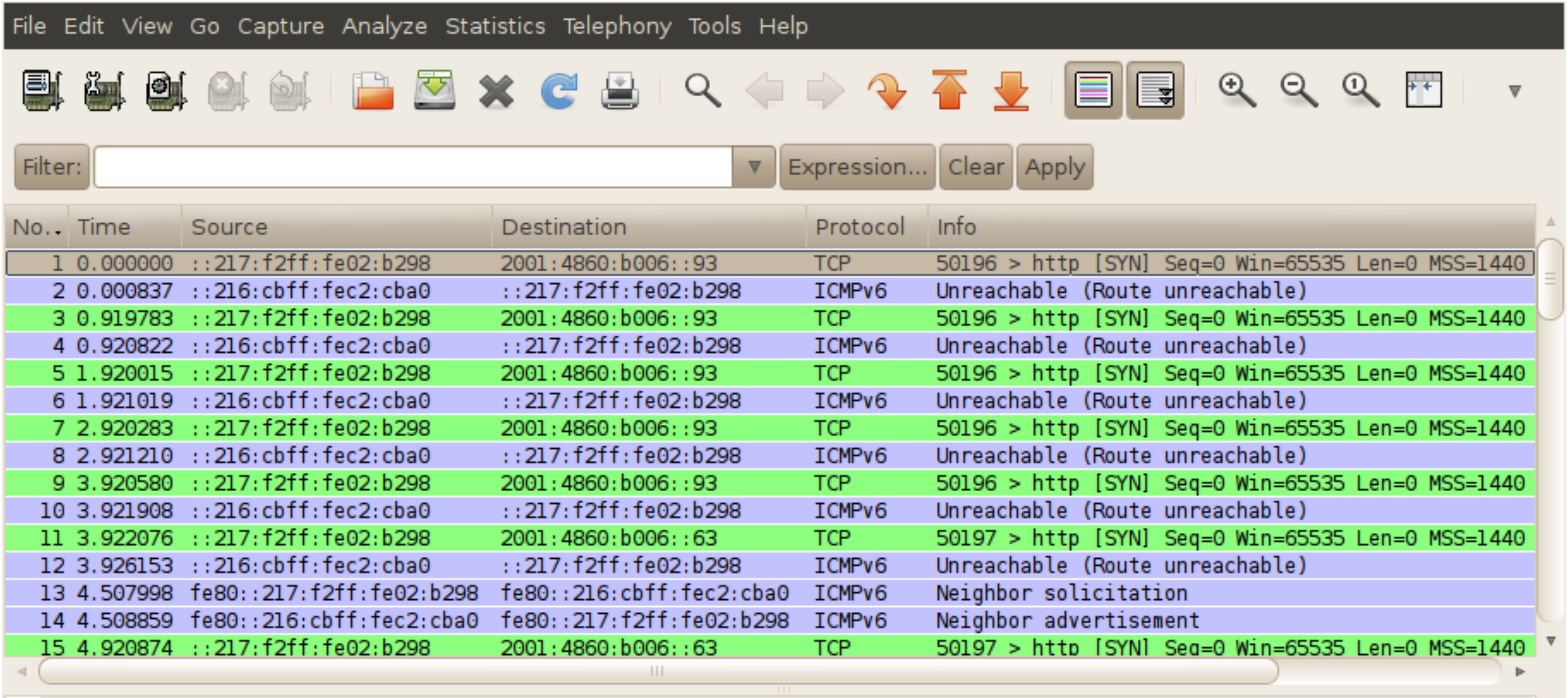
Home gateway behaviour

- Routers may turn on 6to4 and go through broken relays
  - At best, it will cause a latency increase
  - Relay may introduce packet loss or refuse to route packets not originating from 2002::/16
  - This will break things even if there is real IPv6 connectivity!
- Routers may turn on 6to4 with private addresses
  - This will never work
  - ... but some implementations do it anyway

Host behaviour

- Hosts may prefer 6to4 router over native IPv6 router
  - e.g., if 6to4 router sends RAs more frequently
- Host may prefer 6to4 address over IPv4 address
  - Not using RFC3484-compliant getaddrinfo()
  - Using private addresses
    - Known issue in RFC 3484
- Similar considerations for Teredo
  - High setup times, uncertain reliability
  - Most implementations know better than this
- Firewalls may block or break IPv6 (e.g., blocking ICMPv6)

My favourite



No.	Time	Source	Destination	Protocol	Info
1	0.000000	::217:f2ff:fe02:b298	2001:4860:b006::93	TCP	50196 > http [SYN] Seq=0 Win=65535 Len=0 MSS=1440
2	0.000837	::216:cbff:fec2:cba0	::217:f2ff:fe02:b298	ICMPv6	Unreachable (Route unreachable)
3	0.919783	::217:f2ff:fe02:b298	2001:4860:b006::93	TCP	50196 > http [SYN] Seq=0 Win=65535 Len=0 MSS=1440
4	0.920822	::216:cbff:fec2:cba0	::217:f2ff:fe02:b298	ICMPv6	Unreachable (Route unreachable)
5	1.920015	::217:f2ff:fe02:b298	2001:4860:b006::93	TCP	50196 > http [SYN] Seq=0 Win=65535 Len=0 MSS=1440
6	1.921019	::216:cbff:fec2:cba0	::217:f2ff:fe02:b298	ICMPv6	Unreachable (Route unreachable)
7	2.920283	::217:f2ff:fe02:b298	2001:4860:b006::93	TCP	50196 > http [SYN] Seq=0 Win=65535 Len=0 MSS=1440
8	2.921210	::216:cbff:fec2:cba0	::217:f2ff:fe02:b298	ICMPv6	Unreachable (Route unreachable)
9	3.920580	::217:f2ff:fe02:b298	2001:4860:b006::93	TCP	50196 > http [SYN] Seq=0 Win=65535 Len=0 MSS=1440
10	3.921908	::216:cbff:fec2:cba0	::217:f2ff:fe02:b298	ICMPv6	Unreachable (Route unreachable)
11	3.922076	::217:f2ff:fe02:b298	2001:4860:b006::63	TCP	50197 > http [SYN] Seq=0 Win=65535 Len=0 MSS=1440
12	3.926153	::216:cbff:fec2:cba0	::217:f2ff:fe02:b298	ICMPv6	Unreachable (Route unreachable)
13	4.507998	fe80::217:f2ff:fe02:b298	fe80::216:cbff:fec2:cba0	ICMPv6	Neighbor solicitation
14	4.508859	fe80::216:cbff:fec2:cba0	fe80::217:f2ff:fe02:b298	ICMPv6	Neighbor advertisement
15	4.920874	::217:f2ff:fe02:b298	2001:4860:b006::63	TCP	50197 > http [SYN] Seq=0 Win=65535 Len=0 MSS=1440

- Home gateway sending out an RA of ::/64
- Host ignoring the unreachables
- 24-second timeout

Brokenness numbers (not final!)

- For all clients:
  - Internet: 0.082% breakage (was 0.09% in June)
  - ISP A: 0.058% (was 0.064%)
  - Whitelisted ISP: 0.014% (was 0.03%)
    - Spread with IPv4 is less significant than above
    - Whitelisting masks brokenness
    - Returning only one AAAA helped
- Without OS X, numbers are in four nines territory

	1 week	1 month
Internet	0.039%	0.041%
ISP A	0.0080%	0.0090%
Whitelisted ISP	0.0097%	0.0082%

How do we fix this?

- Router problems
  - Need router upgrade
  - Home gateways not upgraded, often not upgradable
  - Hard to figure out what the problem is
- Host problems
  - Workarounds in individual applications (e.g., Chrome)
  - To fix all apps, need OS upgrade
  - OS upgrade can also work around router problems
- Only real fix can happen in the hosts
  - Don't use 6to4 or Teredo
    - There is hope: Apple already fixing OS X, Airport