Broken IPv6 clients

Lorenzo Colitti
lorenzo@google.com

What's the problem?

What's the problem?

- The canonical behaviour for dual-stack applications is itojun's "Implementing AF-independent application"
  - Use getaddrinfo() to resolve all addresses
  - Connect to them in order, taking the first that works
- getaddrinfo() usually returns IPv6 first
  - Application tries IPv6, falls back to IPv4
  - AAAA records fail / timeout one by one, then try IPv4
- How bad can it get?

Failure modes

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

OS behaviour

- 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

What's the impact?

- www.google.com can have up to 6 AAAA records
  - Mac: 24 seconds
  - Windows: 2 minutes
  - Linux: either instant or > 18 minutes
  - MSIE >= 7 won't work at all (gives up after 5 attempts)
- Needless to say, this is unacceptable
- Mitigate the damage by publishing only one AAAA
  - Still a 20-second timeout on Windows
  - Would you like to wait 20 seconds every time you want to do a Google search?

What's going wrong?

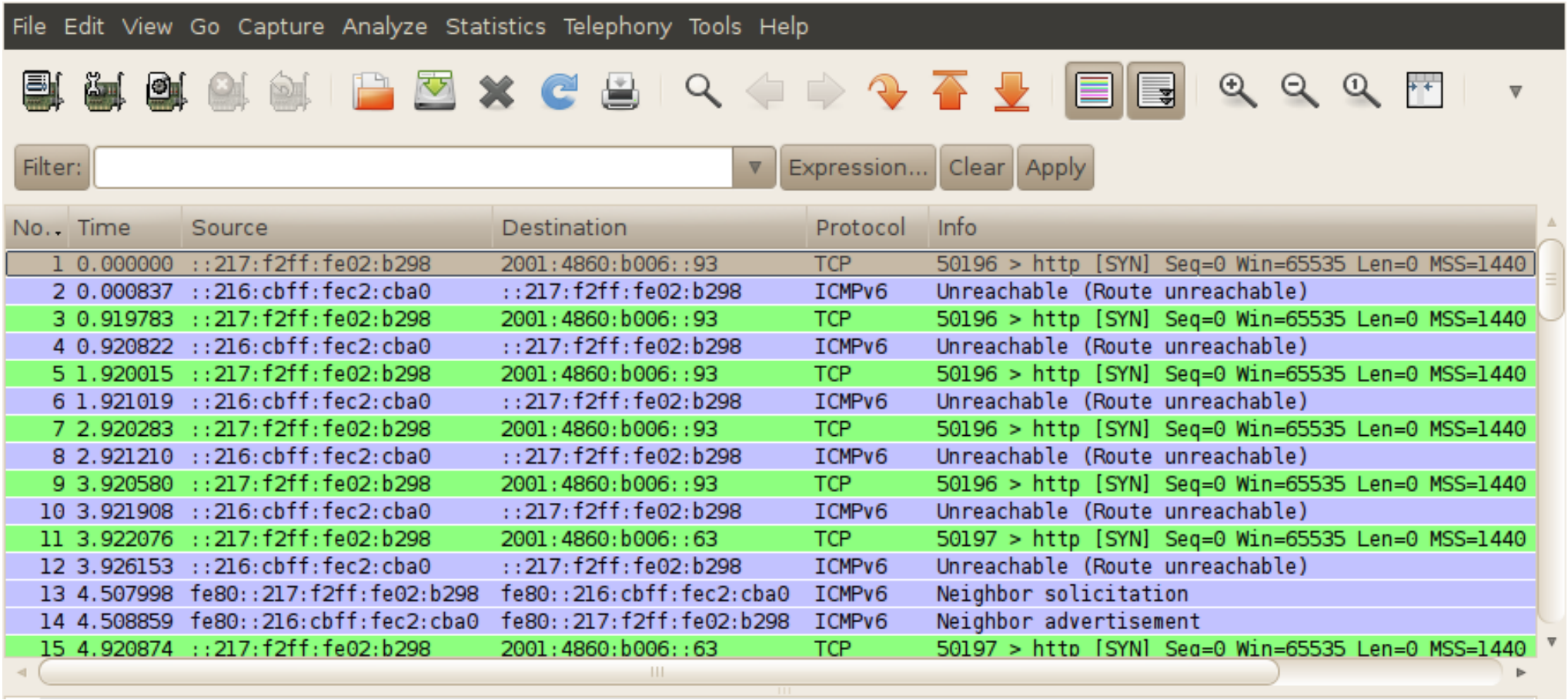
Home gateway behaviour

- 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

Measuring brokenness

Methodology

- Similar to what we and others have done before
  - Ask browser to connect to IPv4 and dual-stack hosts using invisible element on web page
- A few tweaks:
  - Use long-lived websites (e.g., YouTube, gmail)
  - Use Javascript to make multiple requests in session
    - Allows other measurements: MTU, glue, ...
  - Have a sentinel request after a given time
    - Good if user disconnects between two requests
  - Use one-time hostnames
    - Uniquely identifies / associates measurements
    - Finds out if browser asked for AAAA, A and when
    - Prevents browsers caching Javascript

Data set

- Reasonable data set:
  - Currently about 10M samples per day
  - Web requests only, no DNS yet
- No statistical analysis yet, but daily numbers are stable
- IPv4 also has non-zero failure
  - But difference between dual-stack and IPv4 is clearly visible
- Entire Internet: 0.09% breakage

Results per network and OS

- Large ISP A: 0.064%
- Large whitelisted ISP: 0.03%
  - Spread with IPv4 is less significant than above
  - Whitelisting masks brokenness
- Different OSes have different numbers. For large ISP A:
  - All clients: 0.064%
  - Excluding Mac: 0.014%
    - Mac prefers 6to4 over IPv4

How do we fix this?

How do we fix this?

- Router problems
  - Need router upgrade
  - Users don't typically upgrade home gateways
  - Firmware not upgradable
  - Even if they did, hard to know 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

Host fixes

- draft-wing-http-new-tech-00 ("Happy eyeballs")
  - More general, perhaps more complex solution
  - Needs to be implemented in every application
- Simultaneous parallel connections
  - Mac OS X plan of record
  - Can't fix MTU holes
- Probing on attach
  - Fetch > 1280-byte object over HTTP over IPv6
  - Warn users or even disable IPv6 OS-wide on failure
    - Similar to what recent versions of Windows do to detect captive portals

Questions?

Lorenzo Colitti
lorenzo@google.com