



IPv6 software porting

Bringing IPv6 to the Google code base

Steinar H. Gunderson
IPv6 mercenary

A bit too much to do in one night



- Google internal code base is pretty large
 - Tens of millions of lines of C++
 - Tens of millions of lines of Java
 - Somewhat less Python and JavaScript
 - ...that is not counting Android, Chrome, etc.
- Beginnings: Limited resources (mostly 20% time)
- Need to get visible results and build enthusiasm fast
 - Start at bare minimum, expand from there

“No magic, just 96 more bits”



- IPv4 and IPv6 have lots of differences...
 - Multicast replacing broadcast
 - Link-local addresses
 - Flow labels
 - etc.
- ...that ***don't really matter to most code***
 - Most back-end systems don't even need to speak IPv6 on the wire
- By far the most important difference is simply ***different address format***

Address handling strategies

2001:700:oops:32:bit:overflow

#0: First, a few non-starters



- Will vary from language to language, but a few from the C world:
 - `sockaddr_in`
 - `in_addr`
 - `int`

- The use of `int` is especially bad, since it's hard to grep for

#1: “The whole world is IPv6”



- Just store everything as IPv6 addresses
- IPv4 addresses are stored as “*mapped-v4*”
 - 1.2.3.4 → ::ffff:1.2.3.4 (also written ::ffff:102:304)
- Bonus points: Your OS understands it already
 - Unless you have Windows XP, or OpenBSD
- Completely viable strategy, but
 - Cumbersome way to deal with IPv4 addresses
 - Hard to interface with non-IPv6 code

#2: “These things look pretty similar”



- Build some sort of ***address abstraction***
 - IPv4 and IPv6 can largely be treated the same way
 - Decide what to support and what to leave
- Exact strategy will depend on your implementation language and general situation (like almost everything else said here)
 - Java has `net.util.InetAddress`
 - Python has `ipaddr` module (Google)
 - C and C++ has, well, nothing good
 - Often, people simply use strings, which works surprisingly well
- Google's trio of C++ classes: `IPAddress`, `SocketAddress`, `IPRange`
 - Pretty obvious implementations; look at Squid 3.1's `IPAddress` class for something similar
 - After a while, ingrained enough that even non-IPv6-conscious teams started using them

Observation

Code often needs to *store* and *pass around* addresses,
but only occasionally *understand* them

#3: “The whole world is still IPv4” (!)

- Replace IPv6 addresses by an IPv4 identifier (*IPv6 address coercion*)

2001:700:300:1800:230:48ff:fe98:f8d



Hash 64 → 24 bits

224.148.180.31

- Actually all of 224.0.0.0/3 (class D + E space) is fine for this purpose
 - Gives you 29 bits (probably minus 255.255.255.255)

- Not a permanent or perfect solution, but allows you to *integrate rapidly and easily* with almost all of your IPv4 infrastructure
- Essential part of Google's IPv6 software rollout
- Can even happen implicitly in certain cases
 - IPAddress class will automatically coerce (and log a warning) if needed to avoid crash in production
 - Logs and HTTP headers store both v4-coerced and real (IPv6) addresses, v6-aware applications just use the latter

Practical considerations

```
sed -i s/ipv4/ipv6/ src/*
```

Google

90% done; that leaves only the last 90%



- Most porting is straightforward, but there are some harder problems as well
- E.g., multihoming
 - Not an IPv6 issue per se, but client software will be pretty much forced to deal with it if it didn't before
- Other typical case: Highly variable subnetting
 - You may need to think carefully about DoS and spam handling
- No magic bullet; you'll need to review such problems on a case-by-case basis

- Pretty obvious: Start listening on IPv6, then send IPv6 data
 - Watch it crash
 - Fix, repeat until it looks OK
- Variant of the above: Change some type from `in_addr` to `IPAddress`, fix the compile errors
- Finding pain points quickly: Grep for IPv4-only types
 - If you want to get really fancy, you can do semantic greps with Dehydra/Clang (also lets you look at the call graph)
- Once it's working, keep people from breaking it
 - Unit tests, regression tests, API deprecations
- The sooner you get something up, the easier it is to ***build and keep enthusiasm***
 - None or few bonus points for style; coerce, coerce, coerce
 - Announce your AAAAs, collect cake

Questions?
