Alan Bateman

« Meanwhile, back at... | Main | file paths and file... »

IPv6

TUESDAY NOV 09, 2010

I've been nabbed recently by several people asking about IPv6. I'm not sure where the sudden interest is coming from, and maybe it's just that the cost of offshore drilling for IPv4 addresses has got too prohibitive.

As background, the Java SE platform has had support for IPv6 since 1.4. When JDK 1.4 was released then we initially only supported IPv6 on Solaris and Linux. The Solaris support worked really well. Linux was a bit ropey, mostly because it was experimental and wasn't compiled into the kernel by many distributions at the time. Windows didn't have any IPv6 support back then.

When Microsoft released Windows XP SP1, about six months after we shipped JDK 1.4 as I recall, they included their experimental/research IPv6 stack. It wasn't completely integrated, and most importantly for the JDK, didn't support dual-stack sockets that we need to interoperate with IPv4 hosts.

During the development of JDK 5 there were a couple of requests to add support for this IPv6 stack and in the end, **Michael** updated classic networking (java.net.Socket and friends) to use it. With this solution, a java.net.ServerSocket uses two underlying sockets (one IPv4, one IPv6) when bound to the wildcard address and this allows it to accept connections from both IPv4 and IPv6 hosts. Similarly, a java.net.Socket is initially created with two underlying sockets and then the appropriate socket is used once the protocol family of the local or remote address is known. The solution was a bit messy and didn't work for New I/O (NIO), but at least provided something for developers that wanted to play with IPv6 on Windows.

When Microsoft released Windows Vista then it came with a completely new and integrated IPv6 stack. Critically, it brought support for dual-stack sockets making it feasible for the JDK to finally support IPv6 on Windows. Unfortunately it came too late for JDK 6, and so had to wait until JDK 7. So early in JDK 7 we added support for the new stack. For NIO it was done as part of the socket-channel completion work that we did in **NIO2**, and for classic networking, the existing code was re-wacked by **Chris** to use the new stack when on Windows Vista or newer.

Aside from the Windows support, there are a few other bug fixes and improvements. I mentioned above that the Linux support was initially a bit ropey and some of that ropiness continued for several releases with several guilty parties involved. Some of those issues impacted applications that didn't care about IPv6, which is why some developers were forced to run with the java.net.preferIPv4Stack property set. The only other IPv6 update in JDK 7 that comes to mind is with NIO's DatagramChannel. That finally gets multicast support, including IPv6 multicasting, and source filtering, which for IPv6 means **MLDv2**.

[3] Comments

Posted at **12:34PM Nov 09, 2010** by alanb in Java
Tags:

Comments:

RE: "Sudden Interest"
The complete depletion of IPv4 addresses is near.
It is estimated by some to occur within the next 200 odd days.
Others say early 2011.

http://www.nro.net/media/remaining-ipv4-address-below-5.html

There is a strong push by folks that plan ahead to start embracing and supporting IPv6 now, to learn it, to use it.

Posted by **joakime** on November 09, 2010 at 01:01 PM PST **#**

There is a script that predicts the depletion of IPv4 addresses:
http://www.potaroo.net/tools/ipv4/index.html

Posted by **Rémi Forax** on November 10, 2010 at 03:01 AM PST **#**

IPv6 gaining traction also mentioned here: **http://asert.arbornetworks.com/2010/10/ipv6-momentum/**

Posted by **Vladimir Kotal** on November 10, 2010 at 03:03 AM PST **#**

Post a Comment:

Comments are closed for this entry.

Tags

jdk7

Subscribe

Search

Search

» Browse Archives

Link

Navigation

blogs.sun.com

Weblog

About

Login