

HEP grids face IPv6: a readiness study

Matt Crawford
Fermilab

IPv6 Promises

More addresses

~~Better security~~

~~Manageable routing tables~~ ?

~~Better QoS~~

~~True mobility~~ - - ?

Will IPv6 “happen”?

Evidence for the affirmative:

- If sites are using NAT, effectively IPv4 addresses have already run out.



Evidence for the negative:

- The US government has mandated support for IPv6 on agency networks.

NATs (and firewalls) are ruining the internet. Two LHC T-2 centers have requested IPv6 access to FNAL T-1.



China's **CERNET2** is IPv6-only, carrying some v4-over-v6 for transition purposes.

The global BGP routing tables have 230,000 entries, with 400,000 updates/day – seems to be outstripping progress in chips and memory.



But When?

IANA's last block of addresses is estimated to go 19 Mar 2010.

Regional registries' last blocks:
10 Oct 2010 – "10/10/10."



Will IPv4 end then? Of course not.

Readiness Roll-Call

Backbone Networks	...	<i>Ready!</i>
Operating Systems	...	<i>Ready!</i>
Site Networks	...	<i>Partly ready</i>
Site Infrastructure	...	<i>Not ready</i>
Application Software	...	<i>Partly ready</i>

Site Networks

Common network hardware is fully IPv6-ready.

Your site needs to obtain or prepare:

- Address prefix(es) from network provider(s)
- Addressing plan
- Addressing infrastructure: Static, DHCPv6, and/or auto-configuration, and DNS connection to assignment mechanism.

Site Infrastructure

The biggest problem
is security.

(Isn't it always?)



- Border/firewall ACLs.
- Internal network partitioning.
- Scanning, monitoring, logging.
- Address-based application access controls.
- “Host firewalls” – ipchains and the like.

Site Infrastructure

Some things are *not* on the critical path to IPv6 deployment:

- Duplicating or porting every infrastructure service – email, printing, file servers.

IPv4 will not go away soon ... perhaps never.



Applications

To write a v4/v6 application in C, *forget* many of the socket library functions you mastered in 1990. For one example:

```
inet_ntoa( )      ... out
```

```
addr2ascii( )    ... in
```

Or use java – SDK 1.5 is fully v6-ready, and programs can be ignorant* of IP versions in use.

Address Dependencies

Applications may be IPv4-specific in these ways:

- Configuration and output files, notably ACLs and logs
 - ★ Generalize – handle both forms
- Inside the code, manipulating `sockaddr_in`
 - ★ Generalize – handle *neither* form.
- On the wire, inside application messages
 - ★ This is the tough case

Grid Software

GridFTP works.

GT4 is v6-compatible, with perhaps a residue of small bugs.

Storage systems dCache and BeSTMan, based on java, seem to be v6-ready, up to possible configuration file and ftp PORT/PASV issues.

Storage system DPM, in C, is rife with casts between network address and `int`.

Deployment Tip

Do *not* use a different set of host names for the IPv6 addresses — use a single name for each host.

This prevents a great many authentication and connectivity problems.

If you accept this advice, you'll never know how much pain you avoided.