DREN IPv6 Implementation Update

Internet2 Joint Techs, Winter 2010
2 Feb, 2010
Salt Lake City, UT

Ron Broersma
DREN Chief Engineer
High Performance Computing Modernization Program
ron@spawar.navy.mil
Introduction

- Aggressive deployment of IPv6 to DoD’s R&E WAN (DREN) and to all campuses of one major customer (SPAWAR)
- These are production networks with 10’s of thousands of users and systems.
  - i.e., not just a testbed
- Goals
  - See what works and what’s broken
  - See what’s missing
  - Share lessons learned
Enabling IPv6 throughout your environment needs to be a cultural thing.
  - Get everyone involved

Do it as part of tech refresh.
  - Far less expensive than in crisis mode

It may seem overwhelming in the beginning, but it's really not that hard to get started.
  - Jump right in
  - Everyone gets it wrong at first, and you will too, so don't wait around trying to get it right before doing anything.

Very important that we focus on making our public facing services dual-stack.

There are still many problems and issues to be worked.
Previously discussed...

- Reported at Indianapolis meeting:
  - Google over IPv6 (for SPAWAR)
  - Lack of IPv6 support in Net::LDAP pearl module
  - Oracle – lack of IPv6 support
  - NetApp storage appliance – problems with IPv6 support
  - java defaults to IPv4 instead of IPv6, and a fix.
  - Using ISATAP to solve VPN issues
  - Wrong tunnel metrics in Windows, chooses wrong interface
  - Auto-sync for IPv6 records in DNS
  - Mac OSX failings (DHCPv6, ISATAP)
  - Broken Path MTU discovery in Juniper routers
New approach to getting people started

- Training approach is more pragmatic
  - No more “everything you wanted to know about IPv6”
  - Instead, “turn on IPv6 in 5 easy steps”
    - including templates for emails that you need to send

- Pre-configure IPv6 on all DREN customer interfaces

- Lay out some best practices
  - In very strong terms: “Read my lips”.
  - Mostly addressing guidelines.
    - forget about being conservative like in IPv4
    - subnets are /64.
    - don’t encode v4 subnet values into bottom 64 bits.
Google over IPv6

- Feb 3, 2009 – added all of SPAWAR
- July 28, 2009 – DREN and ALL customers added
- Any DREN user that is IPv6-enabled will get to Google services over IPv6
  - Faster (over non-congested links)
    - DREN private peering with Google is IPv6-only
    - Helps to quickly identify IPv6 connectivity problems
- As incentive, we block IPv4 to Google
Deployment progress

- ✔ WAN – dual stack everywhere, peering (unicast+multicast)
- ✔ LANs – all subnets fully support v6, renumber v4
- ✔ Infrastructure services – recursive DNS, NTP, SMTP, XMPP
- ✔ Support services – RADIUS, LDAP, Kerberos
- ✔ Public facing services – authoritative DNS, MX’s, www, NTP
- ✔ Security “stack” – firewall, IDS, IPS, etc.

To Do: Get all the desktops, laptops, and servers running dual-stack
Expanding internal IPv6 adoption

- Jan 2009 – only 5% of our systems (servers, desktops, laptops, etc.) were doing IPv6
  - Double from the year before
- Today: A major internal campaign has us now at 87.6%.
  - A totally volunteer and optional effort
  - We had to provide encouragement and incentives for over 500 independent projects and systems administrators
Making progress visible within organizations – another incentive

<table>
<thead>
<tr>
<th>Code</th>
<th>IPv6 Count</th>
<th>Non IPv6 Count</th>
<th>Total Count</th>
<th>IPv6(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>11000</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>100%</td>
</tr>
<tr>
<td>23000</td>
<td>9</td>
<td>0</td>
<td>9</td>
<td>100%</td>
</tr>
<tr>
<td>40000</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>100%</td>
</tr>
<tr>
<td>41000</td>
<td>266</td>
<td>75</td>
<td>341</td>
<td>78%</td>
</tr>
<tr>
<td>42000</td>
<td>20</td>
<td>12</td>
<td>32</td>
<td>62.5%</td>
</tr>
<tr>
<td>43000</td>
<td>42</td>
<td>10</td>
<td>52</td>
<td>80.8%</td>
</tr>
<tr>
<td>53000</td>
<td>1432</td>
<td>129</td>
<td>1561</td>
<td>91.7%</td>
</tr>
<tr>
<td>55000</td>
<td>975</td>
<td>219</td>
<td>1194</td>
<td>81.7%</td>
</tr>
<tr>
<td>56000</td>
<td>771</td>
<td>88</td>
<td>859</td>
<td>89.8%</td>
</tr>
<tr>
<td>71000</td>
<td>560</td>
<td>112</td>
<td>672</td>
<td>83.3%</td>
</tr>
<tr>
<td>72000</td>
<td>530</td>
<td>38</td>
<td>568</td>
<td>93.3%</td>
</tr>
<tr>
<td>83000</td>
<td>11</td>
<td>0</td>
<td>11</td>
<td>100%</td>
</tr>
<tr>
<td>84000</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>100%</td>
</tr>
<tr>
<td>H0000</td>
<td>86</td>
<td>1</td>
<td>87</td>
<td>98.9%</td>
</tr>
<tr>
<td>H4000</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>100%</td>
</tr>
<tr>
<td>H5000</td>
<td>137</td>
<td>5</td>
<td>142</td>
<td>96.5%</td>
</tr>
</tbody>
</table>

TOTAL: 4852, 689, 5541, 87.6%

Percentage of systems doing IPv6
Utilization comparison

Approx 2.5% of traffic is IPv6
Something changed last week

Over 4x increase in IPv6 traffic
Now 10% of our traffic is IPv6

Google adds IPv6 support for YouTube.
New problem areas

- Windows 2000 systems
  - Don’t bother with v6. Just upgrade them.
- Printers
  - Most lack IPv6 support.
  - We’ve started to upgrade the Jet Direct cards in our HP printers.
- Maintaining all the new IPv6 addresses in DNS
- Large groups of systems that are under “configuration control”, and can’t be modified.
- Sys admins that are too busy with other priorities.
- Rogue 6to4 relays sending RAs
  - Windows systems with ICS enabled.
- Symantec Endpoint Protection (SEP) breaks IPv6
- Broken external DNS servers prevent some of our clients from running IPv6
- Vmware ESX 3.x systems – need upgrade to 4.x
- Blackberry Enterprise Services (BES) on IPv6-enabled Windows server will crash.
Keeping DNS updated

• Need to get all PTRs and some AAAA’s in DNS for all devices doing IPv6

• Manual editing of zone files?
  – Much more painful than IPv4
  – How do you know when some device starts doing IPv6 and gets a SLAAC address?

• DHCPv6?
  – Use DHCPv6 to provide addresses, and use dynamic DNS update
  – Problem: too many clients do not yet support DHCPv6 (Windows XP, MAC OSX, others)
DNS auto-update

• Basic scheme
  – Use SNMP to poll the routers
    • Grab the ARP cache and the ND table
  – For all MAC addresses in the ND table with global unicast addresses matching the site IPv6 prefix:
    • Find the corresponding IPv4 address from the ARP cache
    • Find the FQDN for the IPv4 address in DNS (PTR lookup)
    • Build a PTR record for the IPv6 address, using FQDN from IPv4 address
    • Push to DNS dynamically
  – Works very well
  – Yes, there are some additional complexities, and optimizations required, like garbage collection of temporary and privacy addresses.
  – Hoping to release tool tool as open-source.

• Lingering problems with IPv6 objects in the IP-MIB and IPV6-MIB
  – We really need all routers supporting RFC 4293 (version independent IP-MIB)
Privacy addresses

- See RFC 4941
- Windows systems do this by default (and we don’t like it!)
- Breaks many things in our environment
  - Forensics
  - Stable DNS entries
  - Automated management tools
- Could fix with DHCPv6, but client not available in important OS’s
  - Windows XP, Mac OSX
- Would be nice if RA’s could say “don’t do this”
- So we have to visit every Windows machine to disable this.
  - Breaks the “plug and play” goal of IPv6 for clients.
- How To: (next slide)
Disabling privacy addresses

- **Windows XP**
  
  `
  ipv6 -p gpu UseTemporaryAddresses no
  `

- **Windows 2003**
  
  `
  netsh interface ipv6 set privacy state=disabled store=persistent
  `

- **Windows Vista**
  
  `
  netsh interface ipv6 set global randomizeidentifiers=disabled
  netsh interface ipv6 set global randomizeidentifiers=disabled store=persistent
  `

- **Windows 2008**
  
  `
  netsh interface ipv6 set global randomizeidentifiers=disabled
  netsh interface ipv6 set global randomizeidentifiers=disabled store=persistent
  `
• Query to resolve www.extranet.nga.mil with AAAA returns RCODE 3, “no such name” (NXDOMAIN).
  – Windows XP will never do the “A” query
• If the name exists, even if no RRs for it, it should not return NXDomain.

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Source</th>
<th>Destination</th>
<th>Type</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:29:59.312403</td>
<td>IP newiview.17577 &gt; ins1.sd.domain</td>
<td>17998+ [1au] AAAA? <a href="http://www.extranet.nga.mil">www.extranet.nga.mil</a></td>
<td>(49)</td>
<td></td>
</tr>
<tr>
<td>09:29:59.392933</td>
<td>IP ins1.sd.domain &gt; newiview.17577</td>
<td>17998 NXDomain 0/0/1</td>
<td>(49)</td>
<td></td>
</tr>
<tr>
<td>09:30:15.731744</td>
<td>IP newiview.11851 &gt; ins1.sd.domain</td>
<td>35028+ [1au] A? <a href="http://www.extranet.nga.mil">www.extranet.nga.mil</a></td>
<td>(49)</td>
<td></td>
</tr>
<tr>
<td>09:30:15.895239</td>
<td>IP ins1.sd.domain &gt; newiview.11851</td>
<td>35028 1/2/1 A 164.214.10.84</td>
<td>(105)</td>
<td></td>
</tr>
</tbody>
</table>

• Due to faulty behavior of Cisco CSS load balancer doing DNS functions
• Windows XP machines that are IPv6-enabled can’t get to web site.
• See 4.2 of RFC 4074.
After upgrade to Snow Leopard, web browsing and other apps no longer seemed to prefer IPv6 over IPv4.

Behavior is that only the first DNS answer to any query is accepted, and the others are dropped.
- if you get the A before the AAAA, the AAAA will get dropped

In 10.6, mDNSResponder is now used for all unicast DNS queries, not just for multicast as was the case in earlier releases.

mDNSResponder will query for “A” and “AAAA”, but will immediately stop listening after the first reply.
- the application never receives the other responses

References:
- http://support.apple.com/kb/HT3789
- http://openradar.appspot.com/7333104
java on Mac OS X

- java defaults to IPv4 instead of IPv6
  - reported earlier
- You can change the behavior by setting a preference
  - -Djava.net.preferIPv6Addresses=true
- This preference setting has no effect in Mac OS X
  - can’t override the bad default
- Reference:
  - http://openradar.appspot.com/7100919
Windows patching

- We upgraded to Windows Software Update Service (WSUS) 3.0
  - supports IPv6
- All of our Windows patching now happens over IPv6
Mac OS X and IPv6 printers

- You can’t configure an IPv6 address for a printer
- It has to find the printer using Bonjour, or you have to specify a DNS name.
  - an explicit IPv6 address will not work.
  - Apple says: “this is expected behavior”

- Reference:
  - http://openradar.appspot.com/7100507
A note on Freeradius 2

• Freeradius 2 supports IPv6
• Documentation and discussion would lead you to believe that it can’t do IPv4 and IPv6 at the same time
  – see notes in radiusd.conf
  – see discussion on various web forums
• Actually, all you need to do is add another “listen” clause...
Freeradius 2 example

```plaintext
listen {
    type = auth
    ipaddr = *
    port = 0
    clients = clients-ipv4
}

# Listen on the IPv6 address too
listen {
    type = auth
    ipv6addr = ::
    port = 0
    clients = clients-ipv6
}
```

- clients config file for all your IPv4 clients
- IPv6 clients config file
END