



---

# 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](mailto: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



# BLUF

- 
- 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 its 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

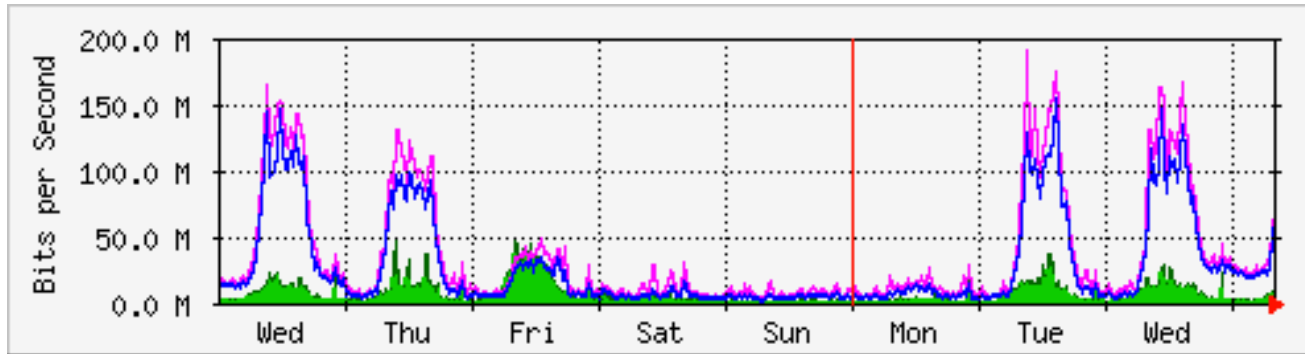
Code	IPv6 Count	Non IPv6 Count	Total Count	IPv6(%)
<u>11000</u>	<u>1</u>	<u>0</u>	<u>1</u>	100%
<u>23000</u>	<u>9</u>	<u>0</u>	<u>9</u>	100%
<u>40000</u>	<u>6</u>	<u>0</u>	<u>6</u>	100%
<u>41000</u>	<u>266</u>	<u>75</u>	<u>341</u>	78%
<u>42000</u>	<u>20</u>	<u>12</u>	<u>32</u>	62.5%
<u>43000</u>	<u>42</u>	<u>10</u>	<u>52</u>	80.8%
<u>53000</u>	<u>1432</u>	<u>129</u>	<u>1561</u>	91.7%
<u>55000</u>	<u>975</u>	<u>219</u>	<u>1194</u>	81.7%
<u>56000</u>	<u>771</u>	<u>88</u>	<u>859</u>	89.8%
<u>71000</u>	<u>560</u>	<u>112</u>	<u>672</u>	83.3%
<u>72000</u>	<u>530</u>	<u>38</u>	<u>568</u>	93.3%
<u>83000</u>	<u>11</u>	<u>0</u>	<u>11</u>	100%
<u>84000</u>	<u>2</u>	<u>0</u>	<u>2</u>	100%
<u>H0000</u>	<u>86</u>	<u>1</u>	<u>87</u>	98.9%
<u>H4000</u>	<u>4</u>	<u>0</u>	<u>4</u>	100%
<u>H5000</u>	<u>137</u>	<u>5</u>	<u>142</u>	96.5%
<b>TOTAL:</b>	<b><u>4852</u></b>	<b><u>689</u></b>	<b><u>5541</u></b>	<b>87.6%</b>

**Percentage of systems doing IPv6**

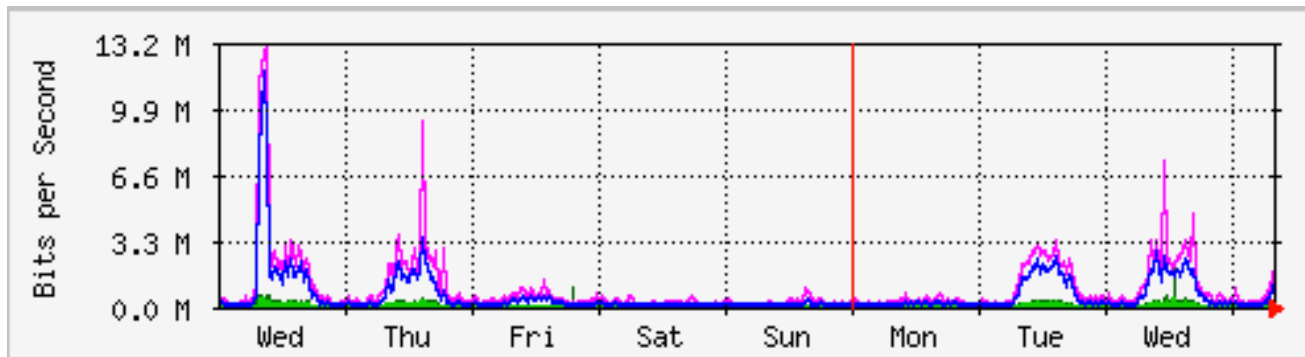


# Utilization comparison

IPv4 traffic



IPv6 traffic

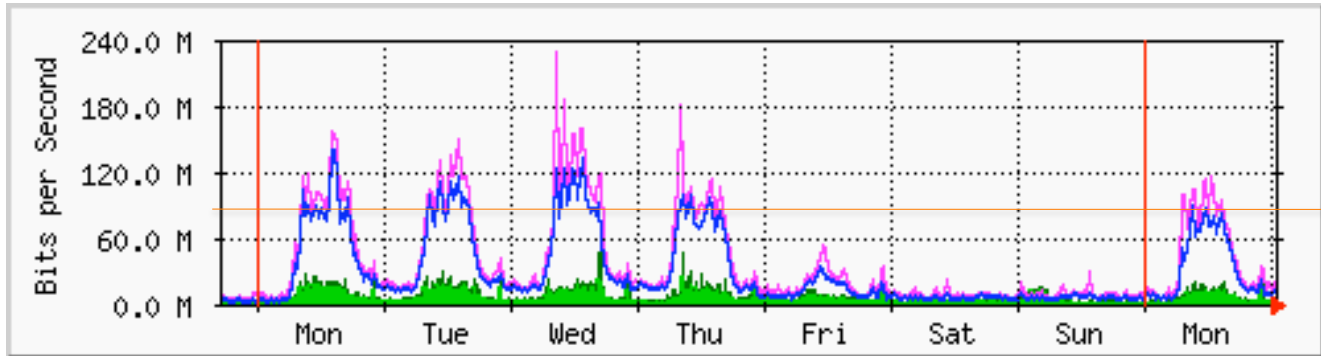


*Approx 2.5% of traffic is IPv6*



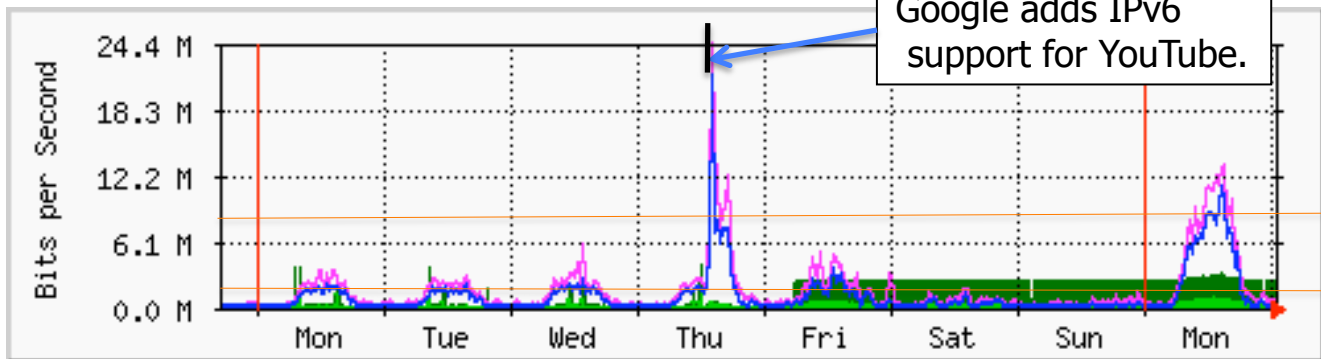
# Something changed last week

IPv4 traffic



90

IPv6 traffic



Google adds IPv6 support for YouTube.

9

2

Over 4x increase in IPv6 traffic

Now 10% of our traffic is IPv6



# 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 privacy state=disabled store=persistent  
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
```



# nga.mil

- 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.

```
09:29:59.312403 IP newiview.17577 > ins1.sd.domain: 17998+ [1au] AAAA? www.extranet.nga.mil. (49)
09:29:59.392933 IP ins1.sd.domain > newiview.17577: 17998 NXDomain 0/0/1 (49)

09:30:15.731744 IP newiview.11851 > ins1.sd.domain: 35028+ [1au] A? www.extranet.nga.mil. (49)
09:30:15.895239 IP ins1.sd.domain > newiview.11851: 35028 1/2/1 A 164.214.10.84 (105)
```

- 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.



# Mac OSX 10.6 (Snow Leopard)

---

- 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

---

```
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**