IPv6 Technical Challenges

Joe St Sauver, Ph.D.
joe@oregon.uoregon.edu or joe@internet2.edu
Nationwide Security Programs Manager, Internet2

NCFTA Canada, Montreal, Quebec
11:30-12:15, November 18th, 2010

http://pages.uoregon.edu/joe/ipv6-technical-challenges/

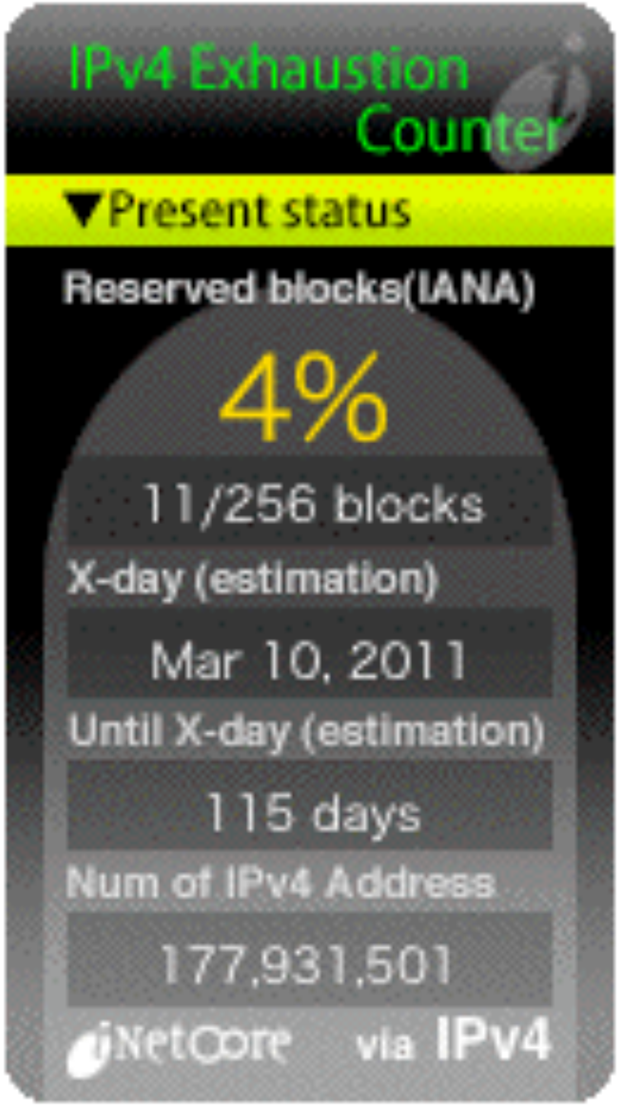
Disclaimer: all opinions expressed are solely those of the author and should not be construed as necessarily representing the opinion of any other entity.

Technical Challenge 1:
IPv4 Address Exhaustion Is Imminent

IPv4 Addrs: An Increasingly Scarce Resource

- There is a finite pool of available IPv4 addresses, and IPv4 exhaustion will occur soon.
- Based on the best available forecasts (see http://www.potaroo.net/tools/ipv4/index.html), the last IPv4 blocks will be allocated by IANA to the RIRs on 10-Mar-2011. That's 112 days from today.
- The regional internet registries (e.g., ARIN, RIPE, APNIC, LACNIC and AFRINIC) will likely begin to exhaust the address space they've received from IANA roughly six months after that, on or around 15-Sep-2011.
- These best estimates are based on current trends. Actual exhaustion might happen earlier depending on what the community does.
- From now till 15-Sep-2011 is roughly 10 months. That's really not very much time.

inetcore.com/project/ipv4ec/index_en.html



IPv4 Exhaustion Counter

▼Present status

Reserved blocks(IANA)

4%

11/256 blocks

X-day (estimation)

Mar 10, 2011

Until X-day (estimation)

115 days

Num of IPv4 Address

177,931,501

Just Ten Months…

- Ten months isn’t much time if you don’t already have an IPv6-capable infrastructure (or plans and processes underway for getting there).
- ISPs may need to do some “forklift upgrades” to at least some of their gear, they’ll need to arrange to get IPv6 address space, and they’ll need to update their provisioning systems and network monitoring systems, and they’ll need to train their staff and end users, and…
- Bottom line: there’s a lot to do, and not a whole lot of time left in which to do it.
- Moreover, there are a relatively limited number of people with IPv6 expertise available to help ISPs through any rough spots they may encounter.
- Fortunately, this is something of a slow-speed “crash.”

The Internet, Post IPv4 Run Out

- Running out of IPv4 addresses isn't like running out of water in the desert, or air while SCUBA diving -- if you already have IPv4 address space, the IPv4 address space you already have will continue to work just fine.
- People who WILL run into problems, however, include:
  -- **new ISPs** who need IPv4 addresses just to get rolling
  -- **growing ISPs** which need more IPv4 addresses
  -- **customers** of existing IPv4-based ISPs who may want to access network resources available ONLY via IPv6, or who end up behind stopgap interim kludges, and
  -- **vendors** who haven't IPv6-ified their product line.
- Surprisingly, however, **many people do NOT seem to view exhaustion of IPv4 address space as an urgent or pressing issue.** In fact, many people seem to think...

"This Whole IPv4 Exhaustion Thing Is Just A Bunch of Malarkey! Smart Internet Folks Will Figure Out *Some* Way To Stretch Out What IPv4 Space We've Got Left… What We've Got Left Has Got To Be Enough To Last Us For Years and Years and Years…"

(Sorry, no.)

Consumptive Momentum

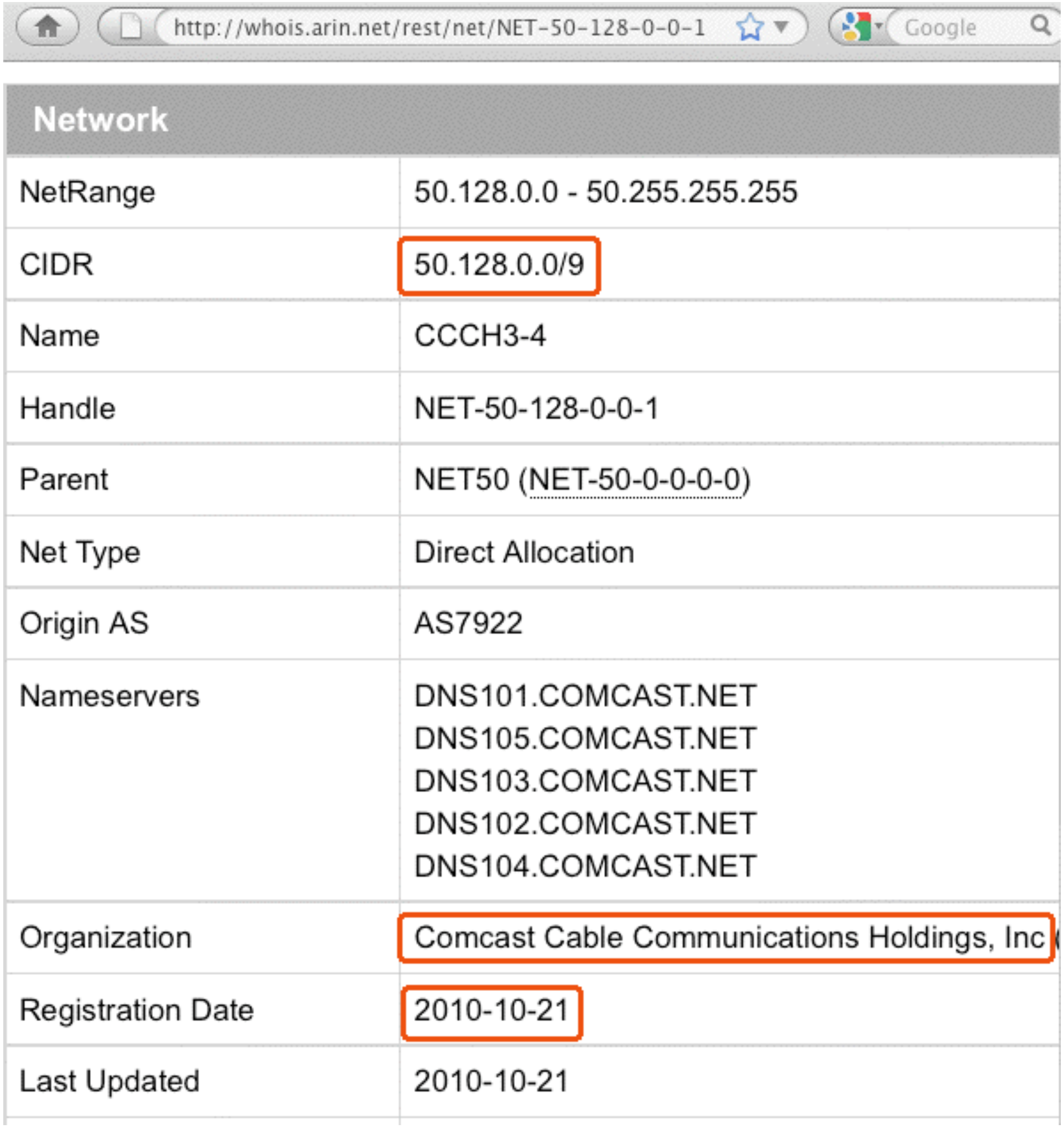
- That sort of desperate unfounded optimism, that sort of baseless hope that we're not really facing a critical point in the deployment of the Internet, may keep people from facing reality and doing what needs to be done. We need to stop clinging to the misconception that if all of us (including especially those of us in North America) would just "do our part," we'd have more than enough IPv4 addresses to last us for the foreseeable future.
- Unfortunately, clever ideas, simple address conservation, or even address reclamation, won't be enough.
- The Internet continues to grow, and that growth results in the inevitable consumption of additional addresses.
- People have had some ideas, however…

Example: "What About Using Class 'E' Space?"

- Eagle-eyed folks may notice that in addition to the space that's currently allocated, or available for allocation, there's an additional block of /8's at 240/8 through 255/8, IPv4 address space designated as "reserved for "future use." These are the addresses traditionally known as "Class E" space. Surely now, as we rapidly approach run out, the time might be ripe to begin to use that reserved block of IPv4 address space?
- Unfortunately (I tend to say that a lot in this talk, don't I?), as discussed in "What About Class E Addresses?", see http://tinyurl.com/what-about-class-e , (a) much software and hardware is hardcoded to block use of that address range, (b) we probably couldn't get everything patched to use it in a timely fashion, and (c) even if we could use that space, it would only last another ~18 mos.

“Or, Or, Some People Might Give Back Some IPv4 Address Space They’ve Got That They’re Not Using! *THAT* Would Help, Wouldn’t It?”

- There **have** been some organizations that have returned IPv4 resources (typically legacy /8 netblocks) that are larger than they’ve needed, exchanging those resources for smaller and more appropriately sized, allocations. For example, ten years ago Stanford returned 36/8, and Interop just recently returned 45/8. (Thank you both!)
- Unfortunately, at the current rate of global address consumption, that won’t delay the inevitable run out by very long – returning an unneeded /8 might delay IPv4 exhaustion by a matter of weeks at most.
- Individual national-scale ISPs can and have legitimately justified allocation of <u>large</u> amounts of additional IPv4 address space even as we come close to IPv4 exhaustion.



Network

NetRange	50.128.0.0 - 50.255.255.255
CIDR	50.128.0.0/9
Name	CCCH3-4
Handle	NET-50-128-0-0-1
Parent	NET50 (NET-50-0-0-0-0)
Net Type	Direct Allocation
Origin AS	AS7922
Nameservers	DNS101.COMCAST.NET DNS105.COMCAST.NET DNS103.COMCAST.NET DNS102.COMCAST.NET DNS104.COMCAST.NET
Organization	Comcast Cable Communications Holdings, Inc (
Registration Date	2010-10-21
Last Updated	2010-10-21

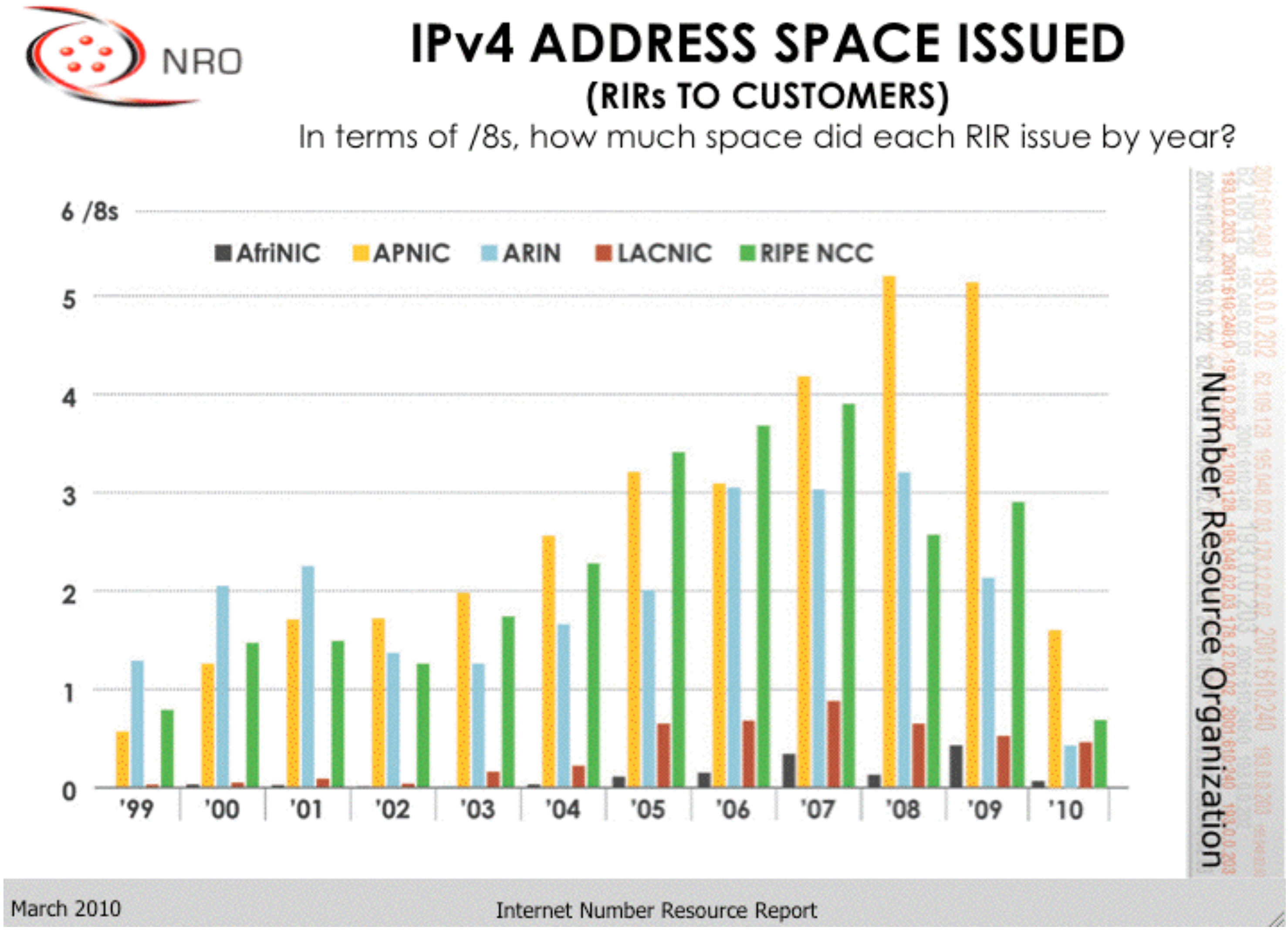
Also, Eventually, IPv4 Address Space *Will* Become an Asset Convertible Into $$$

- If you believe that assertion, and I think you should, this means that organizations that return unneeded address space are potentially being economically irrational, forgoing (potentially substantial) future revenue if/when IPv4 address space becomes a freely marketable asset.
- By implication, too, there are some companies that currently have control over large legacy IPv4 address blocks where their physical assets, or their revenues from ongoing operations, may potentially be dwarfed by the value of their legacy IPv4 address space. Watch for corporate acquisitions driven by a desire to obtain that increasingly valuable legacy IPv4 address space! See http://en.wikipedia.org/wiki/List_of_assigned_/8_IPv4_address_blocks for a list of some legacy blocks.

You Should Also Be Getting Prepared to Deal With IPv4 Address Space Hijacking

- As IPv4 address space becomes more scarce and valuable, it is reasonable to expect that at least some cyber criminals will simply take ("hijack") the IPv4 address space they'd like to have. (After all, that's what criminals do, right? They take what they want even if it doesn't belong to them – why should IP address space be any different?
- As bad as we're doing when it comes to deploying IPv6, we're doing even worse when it comes to securing the IPv4 routing environment against hijacking. Background? See "Route Injection and the Backtrackability of Cyber Misbehavior," http://pages.uoregon.edu/joe/fall2006mm/ and https://www.arin.net/resources/rpki.html

Moreover, North America Is Not The (Only) Region Driving The Address Consumption Bus!



IPv4 ADDRESS SPACE ISSUED

(RIRs TO CUSTOMERS)

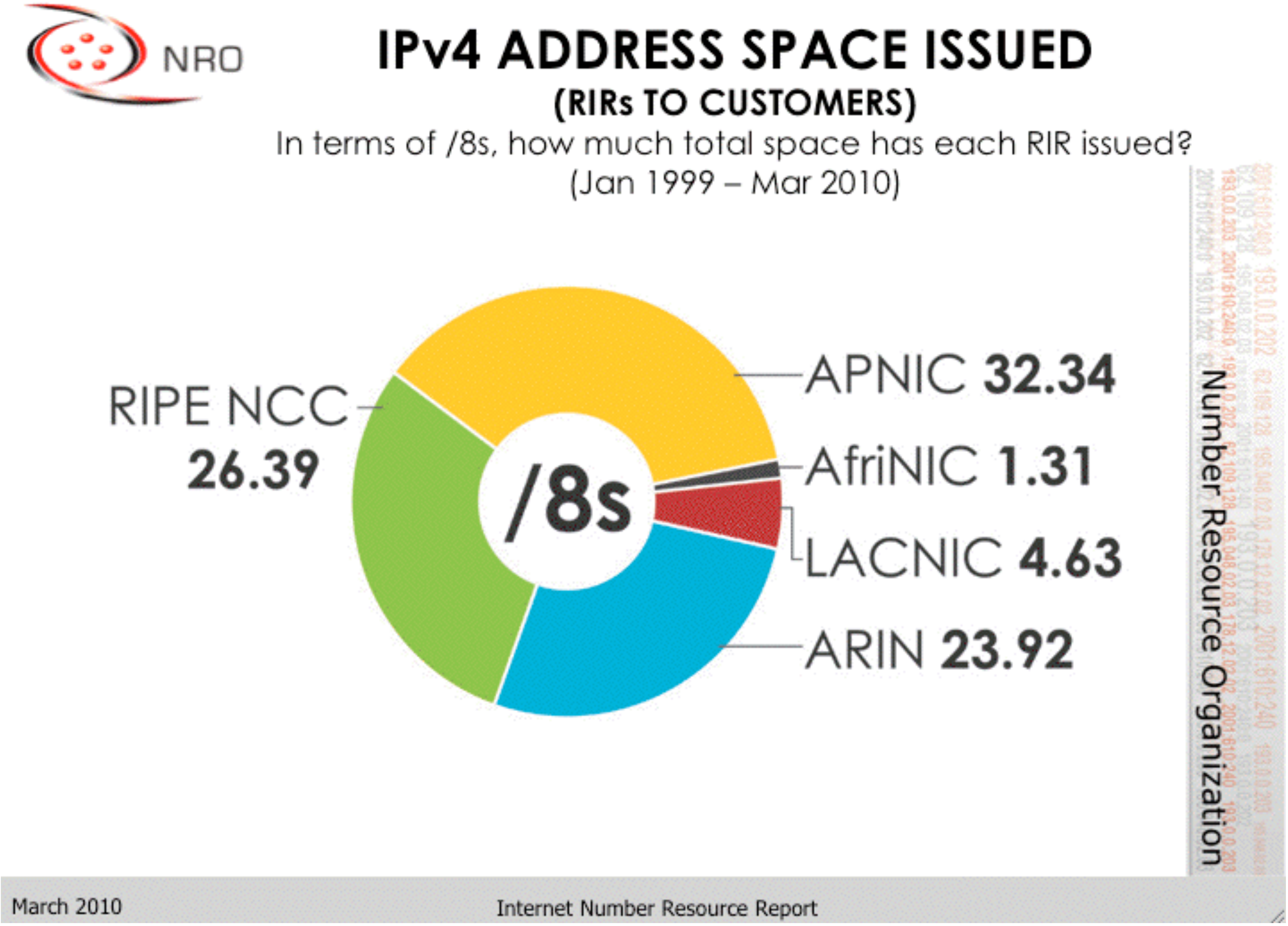
In terms of /8s, how much space did each RIR issue by year?

March 2010

Internet Number Resource Report

http://www.arin.net/participate/meetings/reports/ARIN_XXV/PDF/Monday/Nobile_NRO_joint_stats.pdf

A Cumulative View



NRO

IPv4 ADDRESS SPACE ISSUED

(RIRs TO CUSTOMERS)

In terms of /8s, how much total space has each RIR issued?
(Jan 1999 – Mar 2010)

Number Resource Organization

March 2010 Internet Number Resource Report

http://www.arin.net/participate/meetings/reports/ARIN_XXV/PDF/Monday/Nobile_NRO_joint_stats.pdf

What If IPv4 Address Usage Was Proportionate to Regional Population?

	Population	%	/8's	%	Ratio
Asia:	4,121,097	60.3%	32.34	36.5%	0.605
Africa:	1,009,893	14.7%	1.31	1.4%	0.095
Europe:	732,206	10.7%	26.39	29.7%	**2.775**
L. Amer.:	582,418	8.5%	4.63	5.2%	0.611
N. Amer.:	348,360	5.1%	23.92	27%	**5.29**
Oceania:	35,387	0.5%			
Total:	6,829,360		88.56		

Population in thousands, mid year 2009 estimates

Note: Oceania's addresses are handled by APNIC (e.g., Asia)

Note: Excludes pre-1999 (e.g., legacy) netblocks.

http://esa.un.org/unpd/wpp2008/jpg/WPP2008_Wall-Chart_Page_1.jpg

Decoding The Preceding Table

- If address space usage was proportionate to population, the ratios quoted in the far right column would all be 1.0

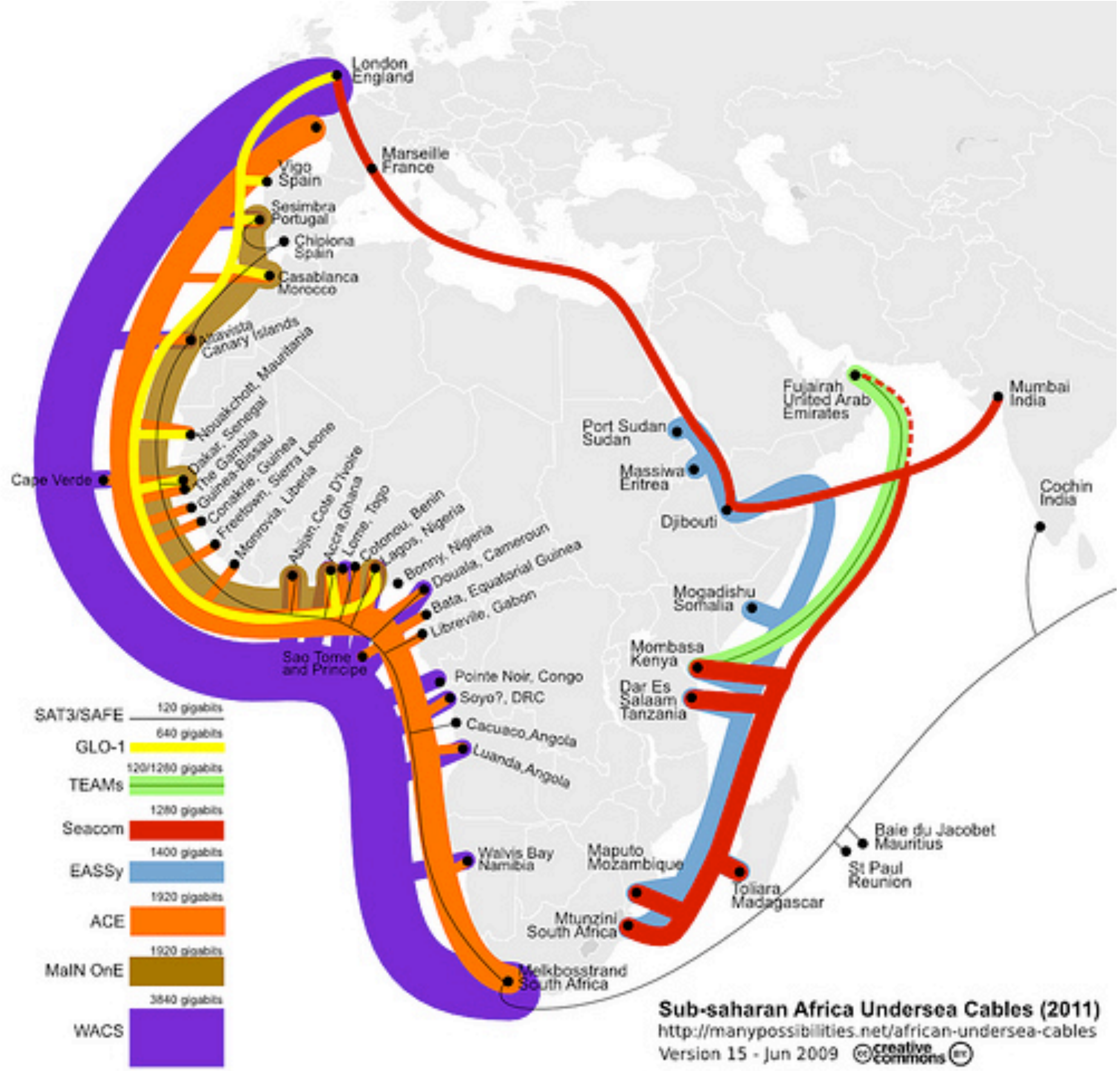
- Regions with ratios greater than one (such as North America, with a ratio of 5.29, and Europe, with a ratio of 2.775), have more IPs per capita than expected.

- Regions with ratios less than one (such as Africa at 0.095) have far fewer IPs per capita than expected.

- Over time, if IPv4 resources weren't limited, as Internet penetration improved, we'd expect those ratios to converge as all regions "caught up" with the developed world.

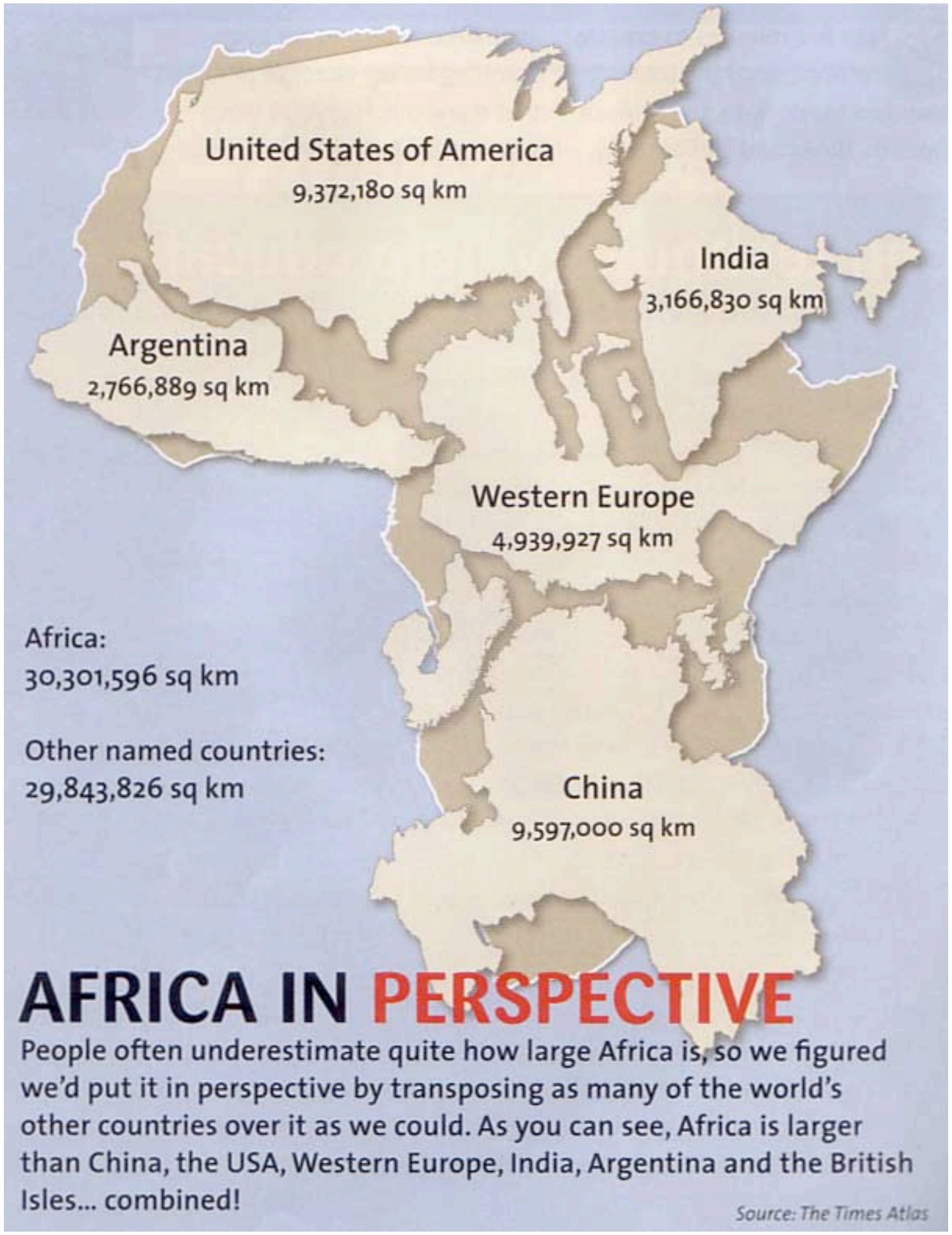
Let’s Think For A Second About “Tiny” Africa

- Historically, Africa’s non-legacy IPv4 address usage to date has been *de minimus*, less than one and a half /8s.
- This was likely due to a variety of factors, but at least one important factor was the high cost of connectivity (thousands of dollars per Mbps per month vs. just dollars per Mbps per month in the US for bulk customers).
- Another driver was widespread use of satellite Internet connectivity, with high latency, NAT’d connections and provider assigned IP address space issued by North American (or European or Asian) satellite operators.
- Improved fiber connectivity is changing all that. Some of the world’s largest and most densely populated regions in Africa and in central Asia are now coming online, and I believe the improved connectivity to those areas will result in a surge in demand for new IPv4 addresses.



Sub-saharan Africa Undersea Cables (2011)
http://manypossibilities.net/african-undersea-cables
Version 15 - Jun 2009 creative commons

http://blog.foreignpolicy.com/files/images/090618_africa_underseas_cables.jpg



AFRICA IN PERSPECTIVE

People often underestimate quite how large Africa is, so we figured we'd put it in perspective by transposing as many of the world's other countries over it as we could. As you can see, Africa is larger than China, the USA, Western Europe, India, Argentina and the British Isles... combined!

Source: The Times Atlas

http://strangemaps.files.wordpress.com/2006/11/africa_in_perspective_map.jpg

If You Still Believe We Have Enough IPv4 Addresses For The Foreseeable Future…

- … notwithstanding the preceding slides, you must also believe in miracles! :-)
- The collective populations of Europe, Asia, Latin America and Africa (and yes, North America, too!) WILL deplete any residual quantity of IPv4 addresses we manage to scrape together. There is no miraculous reclamation or conservation program that will be sufficient to save us.
- So rather than hoping for miracles, I think we need to make progress when it comes to getting IPv6 deployed. :-)

If You Do Plan to Stick with (Just) IPv4

- I recognize that some of you will, nonetheless, not plan to adopt IPv6 any time soon. If so, do YOU have all the IPv4 address space you're going to need?
- **If you have a legitimate need for more IPv4 addresses, I would strongly recommend that you do NOT procrastinate when it comes to requesting them from ARIN. If you do end up waiting, it may be too late when you finally get around to making your request. Act NOW.**
- **Note:** this slide is not meant to encourage address hoarding or requests for addresses you don't actually need. Please be responsible and only ask for what you legitimately need and can honestly justify.
- At the same time, I wouldn't shaft your own users by hesitating to request what you do legitimately need.

Technical Challenge 2: At The Same Time We’re Running Out of IPv4 Address Space, IPv6 Deployment Continues to Lag

So How *Is* IPv6 Deployment Coming?

- In a word, <u>**slowly**</u>.

- In most countries, well under 10% of all networks are announcing IPv6 (and that includes Canada, my friends).

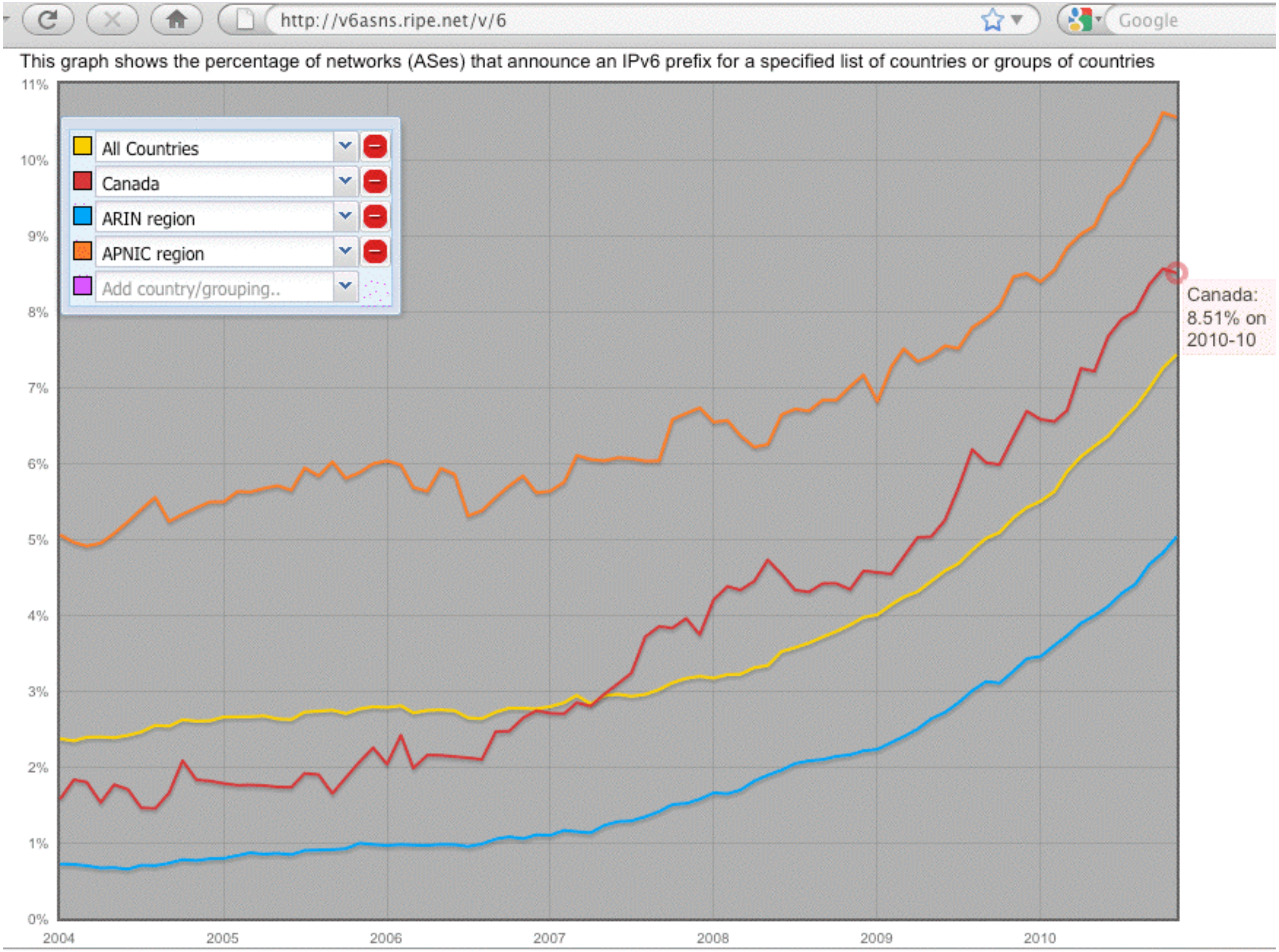
- The web sites that people care about the most are, for the most part, still IPv4 only.

- Literally 99% of all domain names are still IPv4 only, and the Internet's authoritative name server infrastructure is almost entirely still IPv4 only as well.

How Many Networks Are Routing IPv6 Blocks?

- Network engineers typically refer to networks by their associated autonomous system number, or ASN.
- An ASN is usually technically defined as a number assigned to a group of network addresses, managed by a particular network operator, sharing a common routing policy.
- Most ISPs, large corporations, and university networks have an ASN. For example, Google uses AS15169, Sprint uses AS1239, Intel uses AS4983, the University of California at Berkeley uses AS25 and so on.
- **If IPv6 deployment was perfect, and we had 100% adoption, all ASNs that routed IPv4 address space would also be routing IPv6 address space.**
- What do we empirically see if we check the global routing tables? RIPE has a tool that shows how we've been doing over time...

IPv6 Deployment Over Time



This graph shows the percentage of networks (ASes) that announce an IPv6 prefix for a specified list of countries or groups of countries

Decoding the Preceding Graph

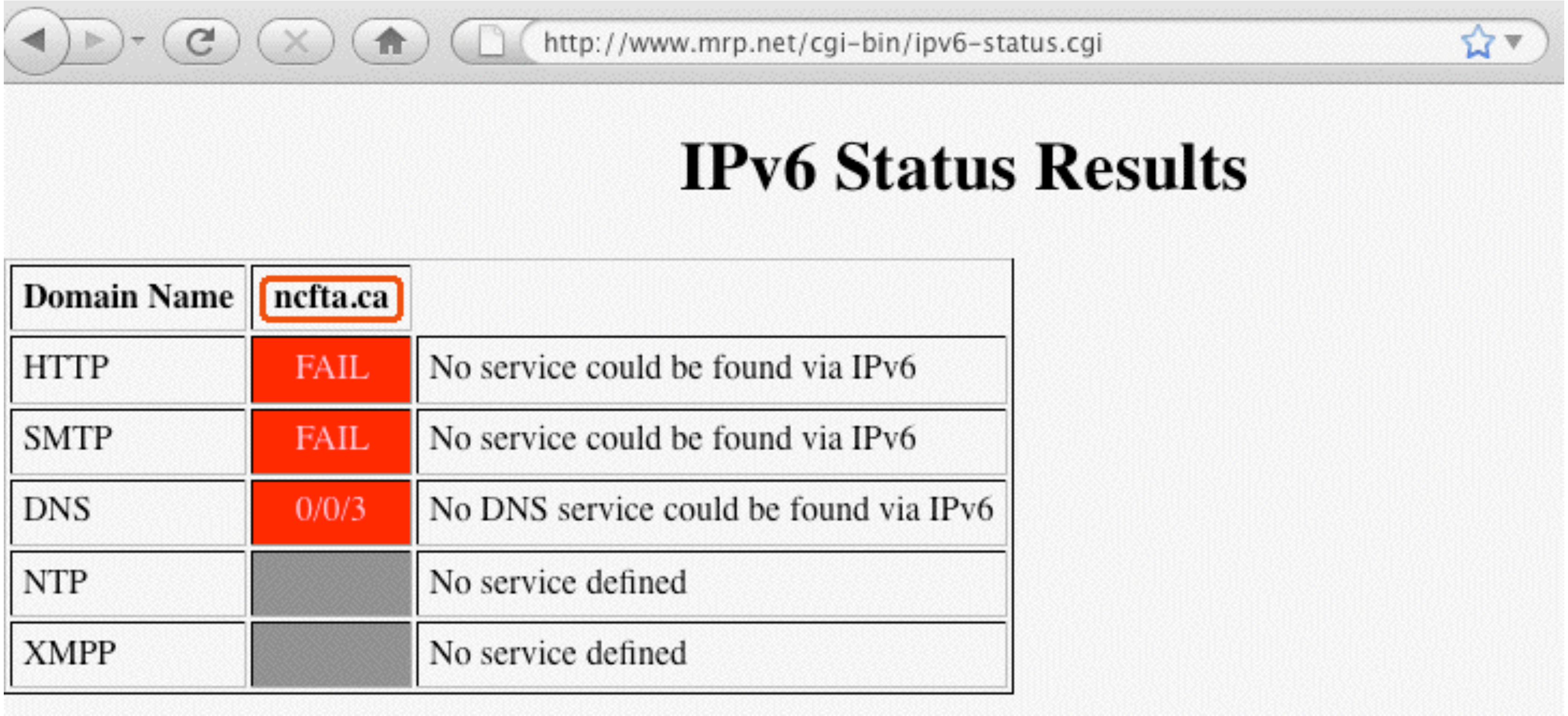
- The Y axis of that graph shows the % of all ASNs in a given country or region that are announcing an IPv6 prefix. The scale of that axis goes from 0 to 11%.
- The X axis is time, running from 2004 to 2010/10.
- The bottom line (blue) shows IPv6 uptake for ARIN (e.g., North America) as a whole. Today we're about at 5%.
- The top line (orange) shows IPv6 uptake for APNIC (e.g., the Asia Pacific region) as a whole. They're the region of the world that's doing best overall when it comes to deploying IPv6. They're at about 10.5%.
- The jaggy red line in the middle is Canadian IPv6 uptake. Canada's currently at 8.51% (that's about 1% above the smooth yellow line, representing global IPv6 uptake).
- Notice that the curves are all roughly parallel, showing approximately similar (leisurely) growth patterns.

What About Major Canadian Web Sites?

- Alexa has a list of the top 100 web sites in Canada (see http://www.alexa.com/topsites/countries/CA).
- Twenty of those web sites have "dot ca" domain names: google.ca, msn.ca, kijiji.ca, craigslist.ca, ebay.ca, sympatico.ca, cbc.ca, matchmate.ca, canoe.ca, tsn.ca, amazon.ca, realtor.ca, futureshop.ca, cyberpresse.ca, ctv.ca, canadapost.ca, yellowpages.ca, bestbuy.ca and bell.ca (there are other Canadian firms on that list with dot com domains, etc., but let's just keep this simple)
- **None** of the main web sites for those twenty dot ca domains had AAAA records (IPv6 addresses) when I tested them on 11/11/2010.
- **Given that lack of IPv6-ification, we must assume that many major dot ca domains may not be IPv6 ready by the time the world experiences IPv4 address exhaustion.**

Checking the Web Sites YOU Care About…

- http://www.mrp.net/cgi-bin/ipv6-status.cgi will let you check the IPv6 status of any arbitrary web site.
  For example:



IPv6 Status Results

Domain Name	ncfta.ca	
HTTP	FAIL	No service could be found via IPv6
SMTP	FAIL	No service could be found via IPv6
DNS	0/0/3	No DNS service could be found via IPv6
NTP		No service defined
XMPP		No service defined

Bringing Up Apache On IPv6 Isn't Very Hard…

- Get Apache 2.2.15 (or whatever's the latest stable version) from http://httpd.apache.org/
- Review httpd.apache.org/docs/2.2/bind.html#ipv6 but otherwise build, install and configure as normal
- When configuring for IPv6, in /etc/httpd/httpd.conf, bind to an appropriate static IPv6 address; EXAMPLE:

  ```
  BindAddress [2001:468:d01:d6::80df:d617]
  ```

- Check your config and start httpd; typically:

  ```
  /usr/local/apache2/bin/apachectl configtest
  /usr/local/apache2/bin/apachectl start
  ```

- Confirm that you can connect OK to your IPv6 httpd:

  ```
  % telnet 2001:468:d01:d6::80df:d617 80
  GET /
  ```

  (note: case matters, GET, not get)
- Problems? Likely a firewall thing, as "always!" :-;

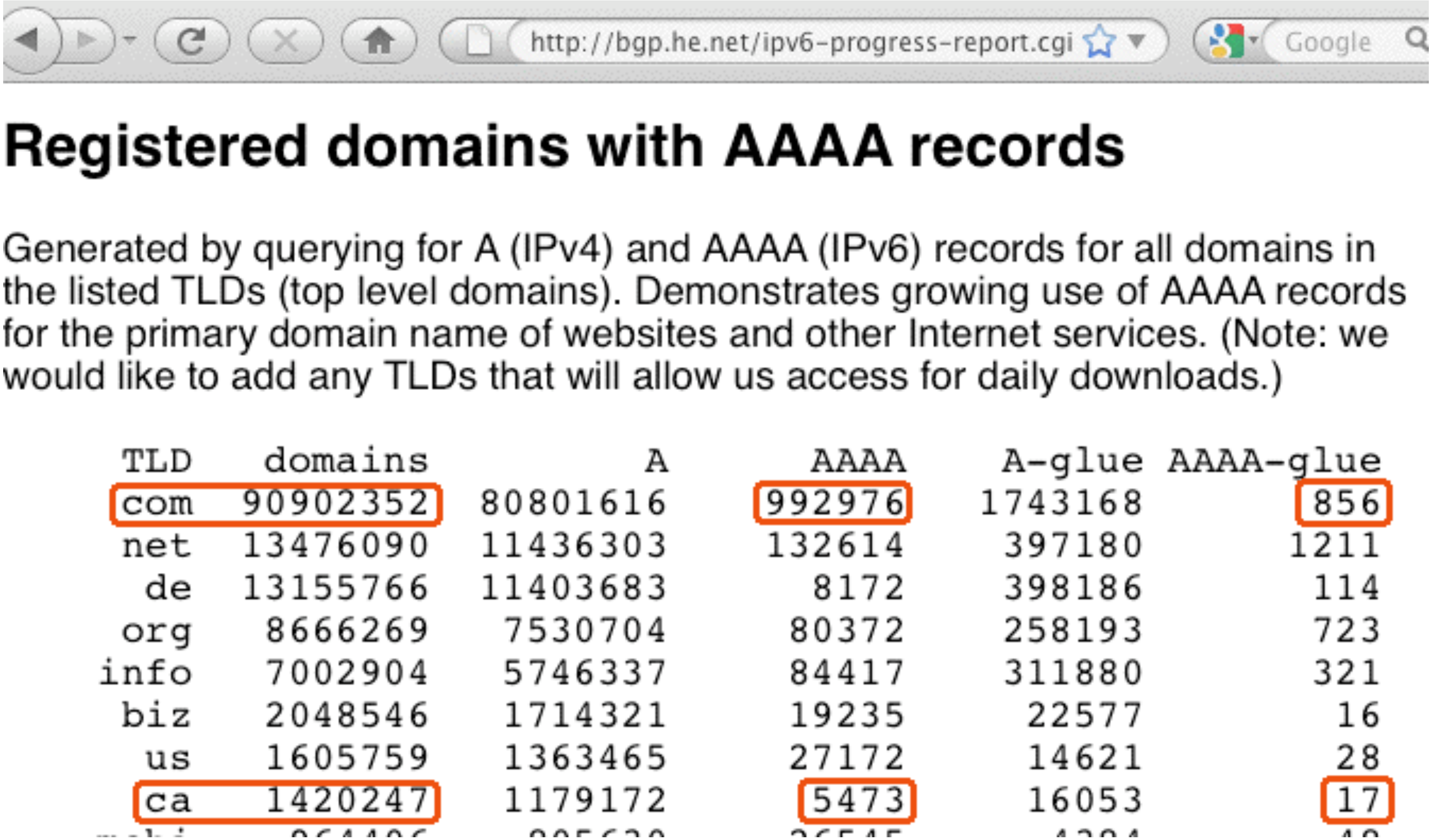
Don’t Forget About IPv6 Addrs in Log Files

```
# cd /usr/local/apache2/logs
# cat access_log
[…]
2001:468:d01:d6::80df:d617 - - [23/Apr/2010:10:20:29 -0700]
"GET / HTTP/1.1" 200 54
[etc]
```

Does your log file analyzer product support IPv6 addresses?

Some, like AWStats from http://awstats.sourceforge.net/ require a separate plugin to enable some IPv6 functionality; other functionality, like mapping addresses to geographic locations, may simply not be available for IPv6.

What About IPv6 Enabled Domain Names?



Registered domains with AAAA records

Generated by querying for A (IPv4) and AAAA (IPv6) records for all domains in the listed TLDs (top level domains). Demonstrates growing use of AAAA records for the primary domain name of websites and other Internet services. (Note: we would like to add any TLDs that will allow us access for daily downloads.)

TLD	domains	A	AAAA	A-glue	AAAA-glue
com	90902352	80801616	992976	1743168	856
net	13476090	11436303	132614	397180	1211
de	13155766	11403683	8172	398186	114
org	8666269	7530704	80372	258193	723
info	7002904	5746337	84417	311880	321
biz	2048546	1714321	19235	22577	16
us	1605759	1363465	27172	14621	28
ca	1420247	1179172	5473	16053	17

Decoding the Preceding Table of Domains

- Each line represents one top level domain, such as dot com or dot ca.
- "A" records map domain names to IPv4 addresses.
- "AAAA" ("quad A") records map domain names to IPv6 addresses.
- "Glue" records are used to define authoritative name server IP addresses
- 1.09% (992976/90902352*100=1.09) of all dot com domains have IPv6 addresses defined. Ugh, that's low.
- By comparison, only 0.38% (5473/1420247*100=0.38) of all dot ca domains have IPv6 addresses defined. Ugh*Ugh!
- Oh yes: a trivial number of IPv6 enabled authoritative name server glue records exist. (So the domain name system is <u>far</u> from being ready to be "IPv6-only.")

Bottom Line: Things Are Not Looking Good…

- **North America (including Canada) will likely not be ready to go with IPv6 when IPv4 address exhaustion occurs.**

- How could this occur in Canada (or the United States)?

- Did no one even notice? Did no one tell us about this?

ICT Standards Advisory Council of Canada, 2010

- **"IPv6 in Canada: Final Report and Recommendations of the ISACC IPv6 Task Group (IITG),"** approved at the 42nd ISACC Plenary on March 16th, 2010 (see http://isacc.ca/isacc/_doc/ArchivedPlenary/ISACC-10-42200.pdf), states [emphasis added]: "Today, Canada is clearly lagging behind its main trading partners with respect to IPv6 awareness and deployment. IPv6 expertise and awareness exists in Canada, but is concentrated in a very small number of people and organizations. [...] **IPv6 deployment into existing networks and operations can take several years. This should be a red flag for Canada, as the last IPv4 address blocks will be depleted in *less than* two years. [...] This report is a call to action. [...] IPv6 is inevitable. Not migrating to IPv6 is not an option."**

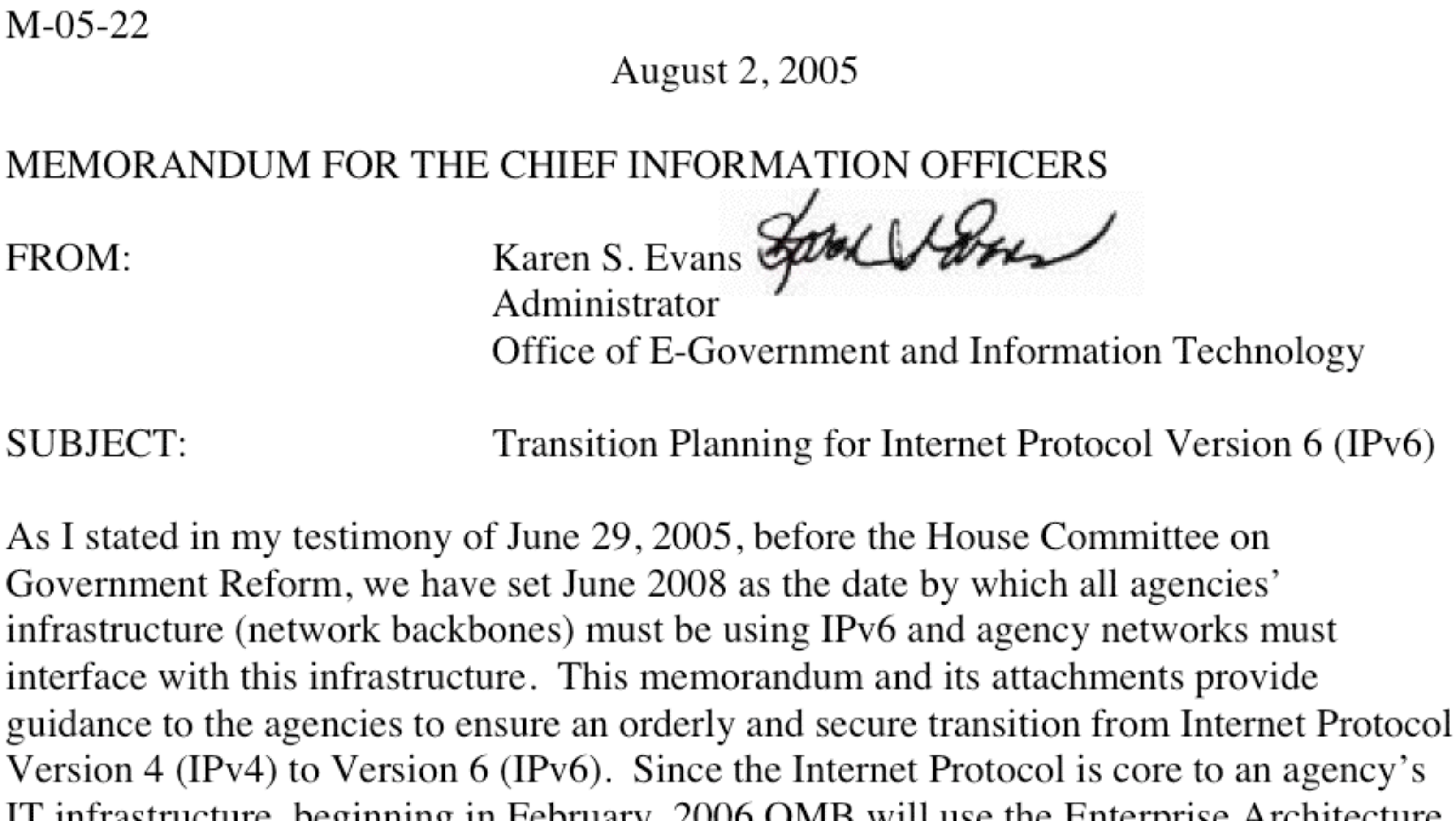
ISACC IPv6 Task Group Recommendations

- *Canadian governments of all levels (federal, provincial, territorial, regional, municipal) shall plan for IPv6 migration and specify IPv6 support in their IT procurements immediately;*
- *Canadian Internet Service Providers (ISPs) shall accelerate the deployment and the commercial availability of IPv6 services for business and consumer networks;*
- *Canadian internet content and application service providers shall make their content and applications reachable using IPv6;*
- *Canadian industries in all sectors shall intensify the support of IPv6 on all products that include a networking protocol stack;*
  [etc]

So What About The Government of Canada?

- If the Government of Canada was IPv6-ready, major Canadian government websites, such as those listed at http://canada.gc.ca/depts/major/depind-eng.html , would be accessible over IPv6 (e.g., they would have IPv6 "quad A" (AAAA) records defined).
- Testing the 228 web sites listed on that page, I don't see ANY that appear to be IPv6 enabled.
- ***Absent substantial immediate progress, we must acknowledge that the Canadian Government may NOT be ready to support access to key online government resources via IPv6 by the time IPv4 address exhaustion occurs.***
- The U.S. Government may not be in much better shape when it comes to IPv6.

U.S. Federal Networks, For Example, Are Supposed to ALREADY Be IPv6 Ready



M-05-22

August 2, 2005

MEMORANDUM FOR THE CHIEF INFORMATION OFFICERS

FROM: Karen S. Evans
Administrator
Office of E-Government and Information Technology

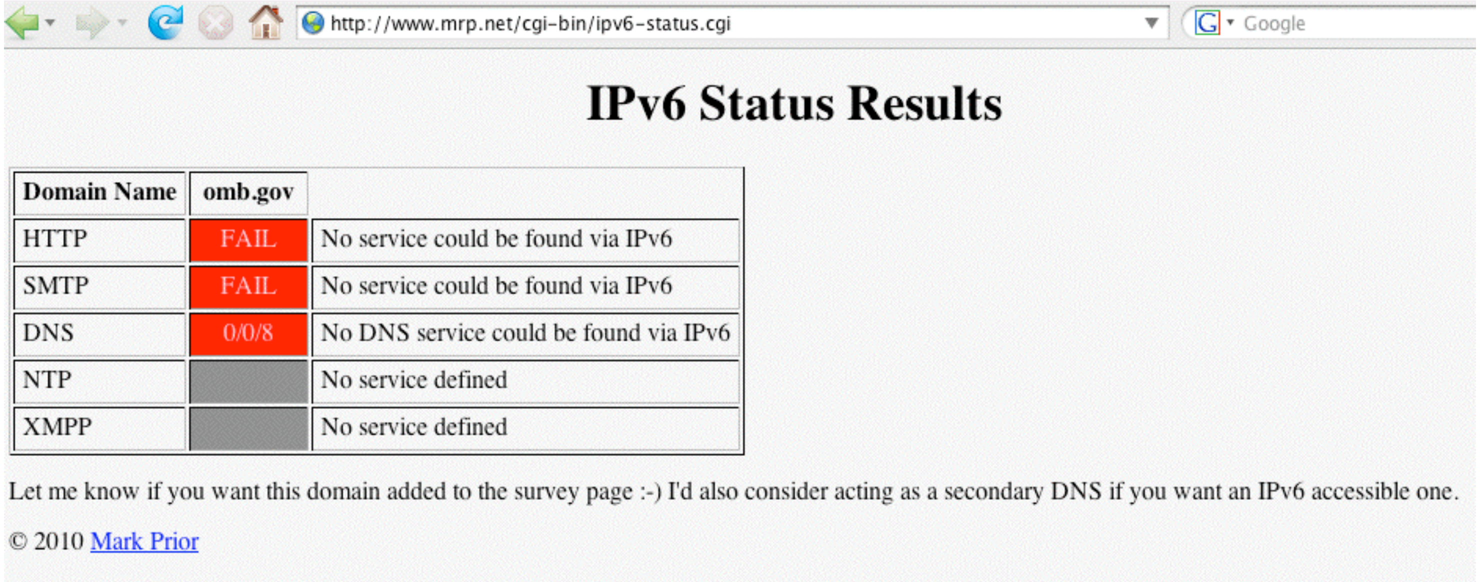
SUBJECT: Transition Planning for Internet Protocol Version 6 (IPv6)

As I stated in my testimony of June 29, 2005, before the House Committee on Government Reform, we have set June 2008 as the date by which all agencies' infrastructure (network backbones) must be using IPv6 and agency networks must interface with this infrastructure. This memorandum and its attachments provide guidance to the agencies to ensure an orderly and secure transition from Internet Protocol Version 4 (IPv4) to Version 6 (IPv6). Since the Internet Protocol is core to an agency's IT infrastructure, beginning in February, 2006 OMB will use the Enterprise Architecture

Source: www.whitehouse.gov/omb/memoranda/fy2005/m05-22.pdf

The U.S. Government Reality Today

- Reportedly many federal networks, having passed one IPv6 packet (and thus, however briefly, demonstrated that their backbones were "IPv6 capable"), promptly "re-disabled" IPv6.
- Don't believe me? Check your favorite U.S. federal sites. Are they v6 accessible?
- Even OMB itself isn't, as far as I can tell!



IPv6 Status Results

Domain Name	omb.gov	
HTTP	FAIL	No service could be found via IPv6
SMTP	FAIL	No service could be found via IPv6
DNS	0/0/8	No DNS service could be found via IPv6
NTP		No service defined
XMPP		No service defined

Let me know if you want this domain added to the survey page :-) I'd also consider acting as a secondary DNS if you want an IPv6 accessible one.

© 2010 Mark Prior

OMB Is Not Alone In Not Being IPv6 Ready

www.dhs.gov --> no
www.doc.gov --> no
www.dod.gov --> no
www.doe.gov --> no
www.dot.gov --> no
www.ed.gov --> no
www.epa.gov --> no
www.hhs.gov --> no
www.hud.gov --> no
www.doi.gov --> no
www.doj.gov --> no
www.dol.gov --> no
www.nasa.gov --> no
www.nsf.gov --> no

www.nrc.gov --> no
www.opm.gov --> no
www.sba.gov --> no
www.ssa.gov --> no
www.state.gov --> no
www.usaid.gov --> no
www.usda.gov --> no
www.ustreas.gov --> no
www.va.gov --> no

Or pick another U.S. federal agency of your choice: the pattern is pretty consistent I'm afraid…

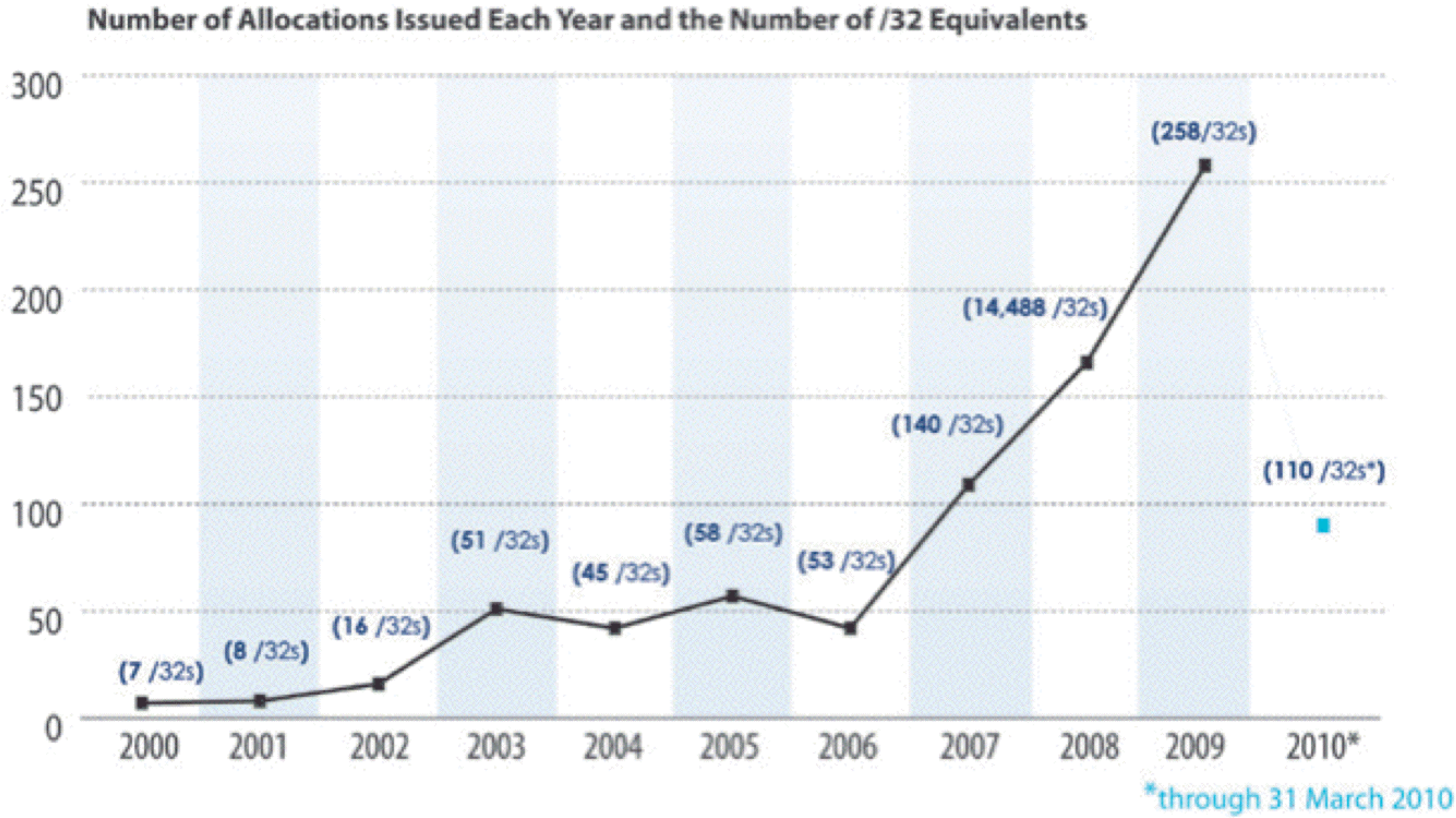
A Month Or Two Ago, The Administration in Washington Seemed To Finally Notice This…

- On Sept. 28th, 2010, the NTIA held a workshop at which Federal CIO Vivek Kundra announced a directive "requiring all U.S. government agencies to upgrade their public-facing Web sites and services by Sept. 30, 2012, to support IPv6…" *and* that access must be via native IPv6 rather than an IPv6 transition mechanism.
- A second deadline, Sept. 30th, 2014, applies for federal agencies to upgrade internal client applications that communicate with public servers to use IPv6.
- For more, see
"White House Issues IPv6 Directive,"
http://www.networkworld.com/news/2010/
092810-white-house-ipv6-directive.html?page=1

“Is There <u>Anyone</u> Who *IS* <u>Currently</u> Using IPv6?”

Yes…

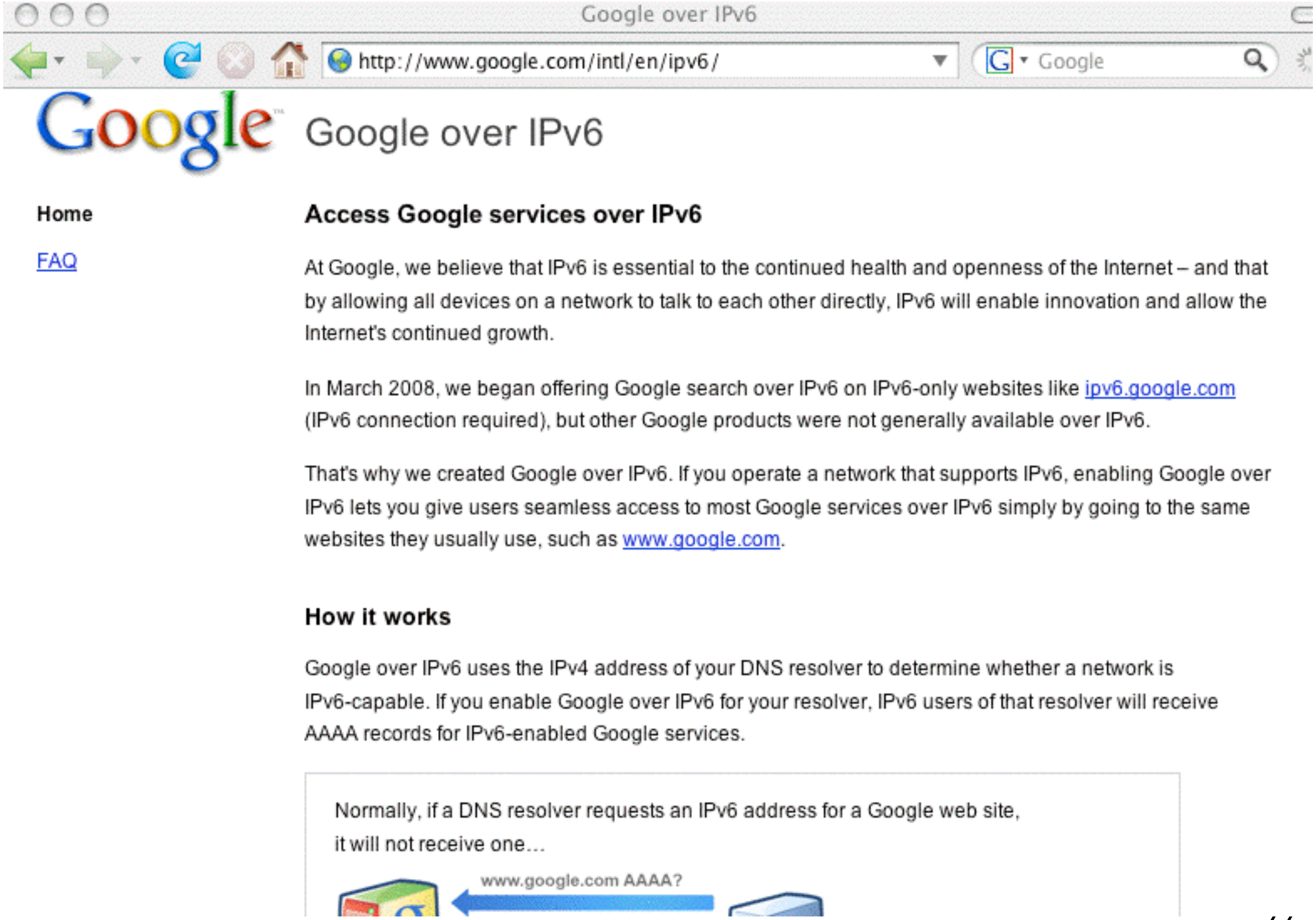
People ARE Asking for IPv6 Address Space from ARIN


Number of Allocations Issued Each Year and the Number of /32 Equivalents

*through 31 March 2010

Source: https://www.arin.net/participate/meetings/reports/ARIN_XXV/PDF/Wednesday/Nobile_RSD.pdf

Google IS Promoting Access via IPv6



Google over IPv6

Access Google services over IPv6

At Google, we believe that IPv6 is essential to the continued health and openness of the Internet – and that by allowing all devices on a network to talk to each other directly, IPv6 will enable innovation and allow the Internet's continued growth.

In March 2008, we began offering Google search over IPv6 on IPv6-only websites like ipv6.google.com (IPv6 connection required), but other Google products were not generally available over IPv6.

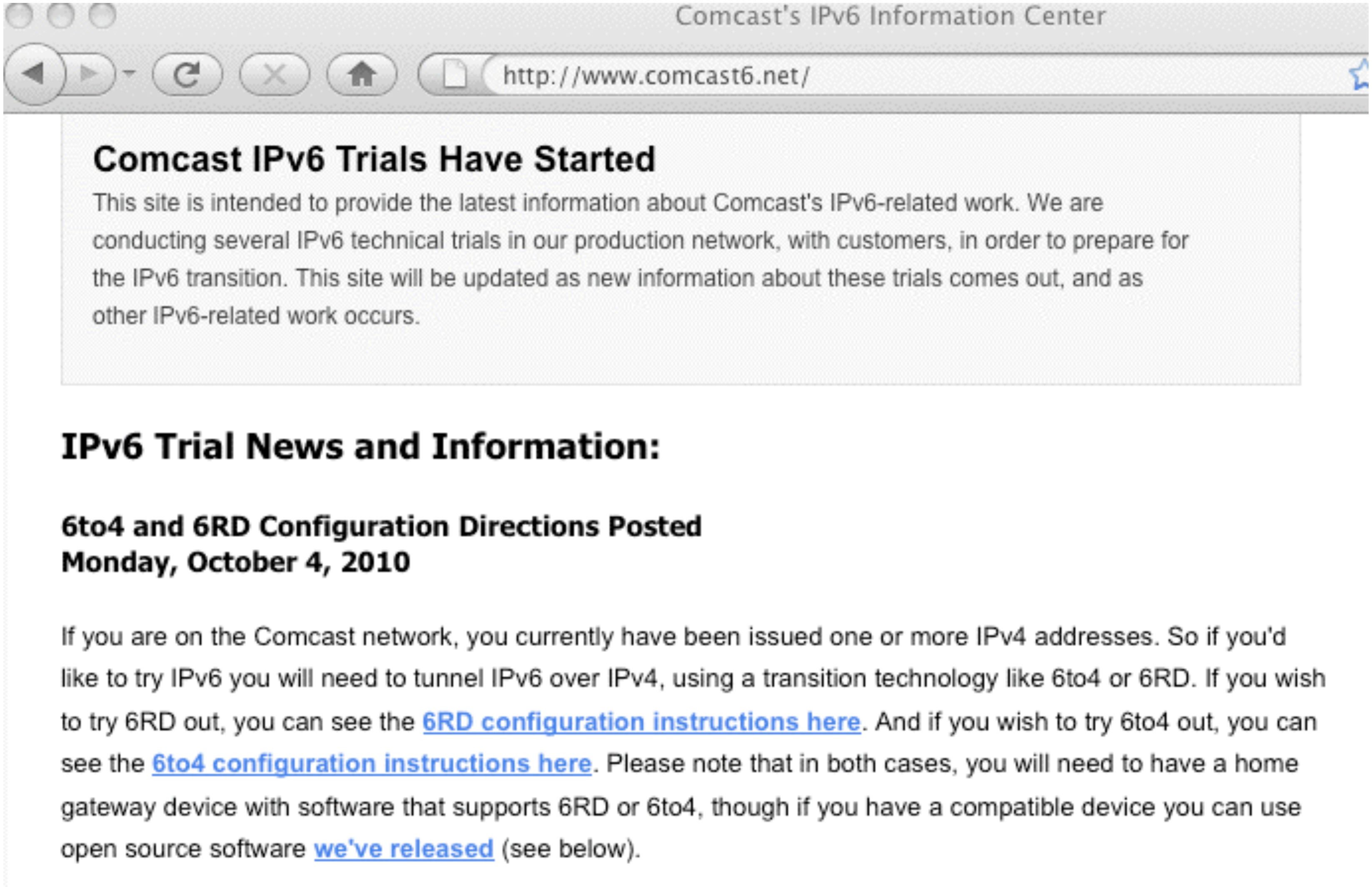
That's why we created Google over IPv6. If you operate a network that supports IPv6, enabling Google over IPv6 lets you give users seamless access to most Google services over IPv6 simply by going to the same websites they usually use, such as www.google.com.

How it works

Google over IPv6 uses the IPv4 address of your DNS resolver to determine whether a network is IPv6-capable. If you enable Google over IPv6 for your resolver, IPv6 users of that resolver will receive AAAA records for IPv6-enabled Google services.

Normally, if a DNS resolver requests an IPv6 address for a Google web site, it will not receive one…

Comcast IS Doing IPv6 Trials



Comcast IPv6 Trials Have Started

This site is intended to provide the latest information about Comcast's IPv6-related work. We are conducting several IPv6 technical trials in our production network, with customers, in order to prepare for the IPv6 transition. This site will be updated as new information about these trials comes out, and as other IPv6-related work occurs.

IPv6 Trial News and Information:

6to4 and 6RD Configuration Directions Posted
Monday, October 4, 2010

If you are on the Comcast network, you currently have been issued one or more IPv4 addresses. So if you'd like to try IPv6 you will need to tunnel IPv6 over IPv4, using a transition technology like 6to4 or 6RD. If you wish to try 6RD out, you can see the 6RD configuration instructions here. And if you wish to try 6to4 out, you can see the 6to4 configuration instructions here. Please note that in both cases, you will need to have a home gateway device with software that supports 6RD or 6to4, though if you have a compatible device you can use open source software we've released (see below).

Some Comcast IPv6 Trials Are Native IPv6, Others Are Testing A Couple of Transition Mode Technologies

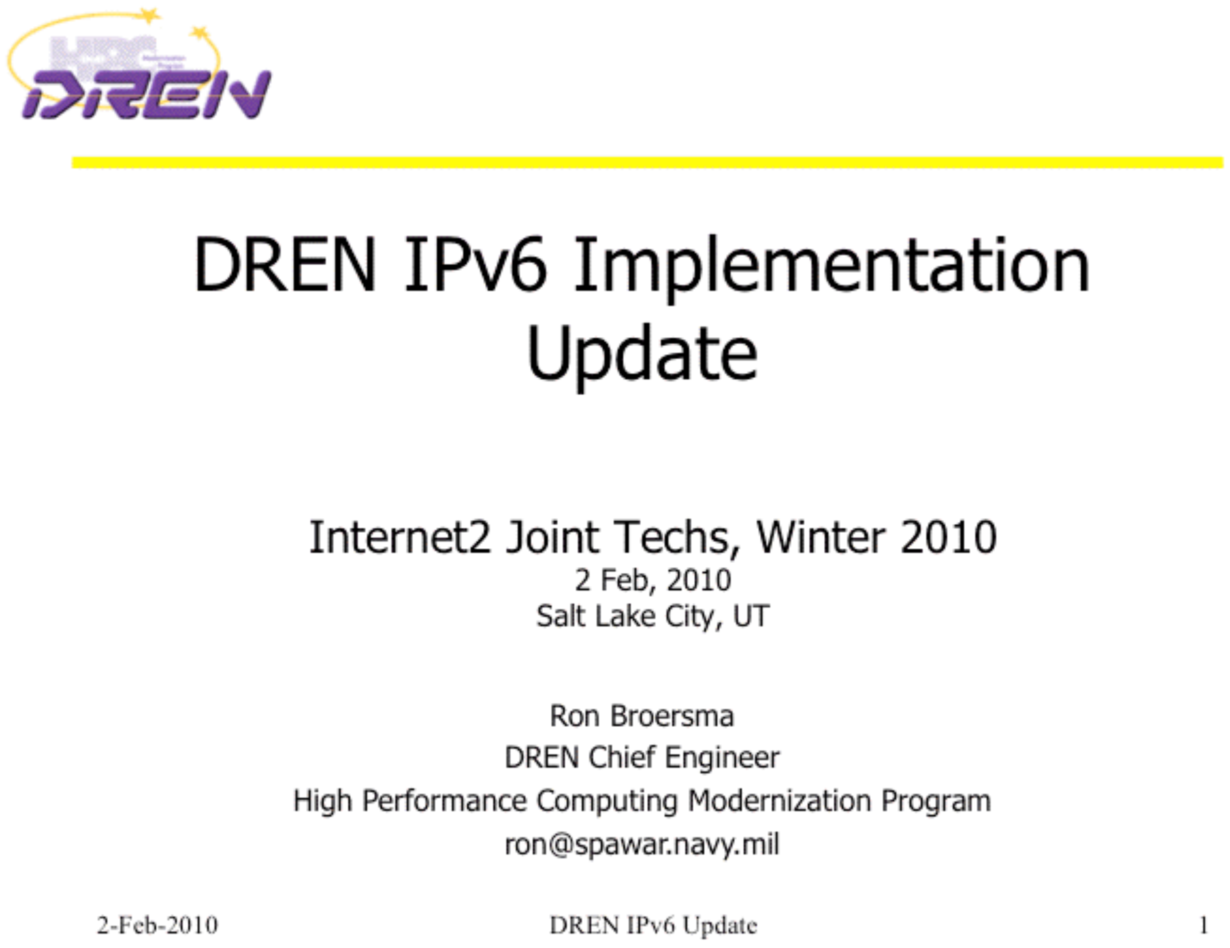
- For example, Comcast is testing both

  -- 6RD (see RFC5569 and http://en.wikipedia.org/wiki/IPv6_rapid_deployment).

  [Note that a draft policy particularly targeting IPv6 address space for 6RD was recently abandoned by the ARIN community (see https://www.arin.net/policy/proposals/2010_9.html)]

  -- Dual Stack Lite (see http://smakd.potaroo.net/ietf/idref/draft-ietf-softwire-dual-stack-lite/index.html)

The U.S. Defense Research and Engineering Network Is *Widely* Using IPv6



DREN IPv6 Implementation Update

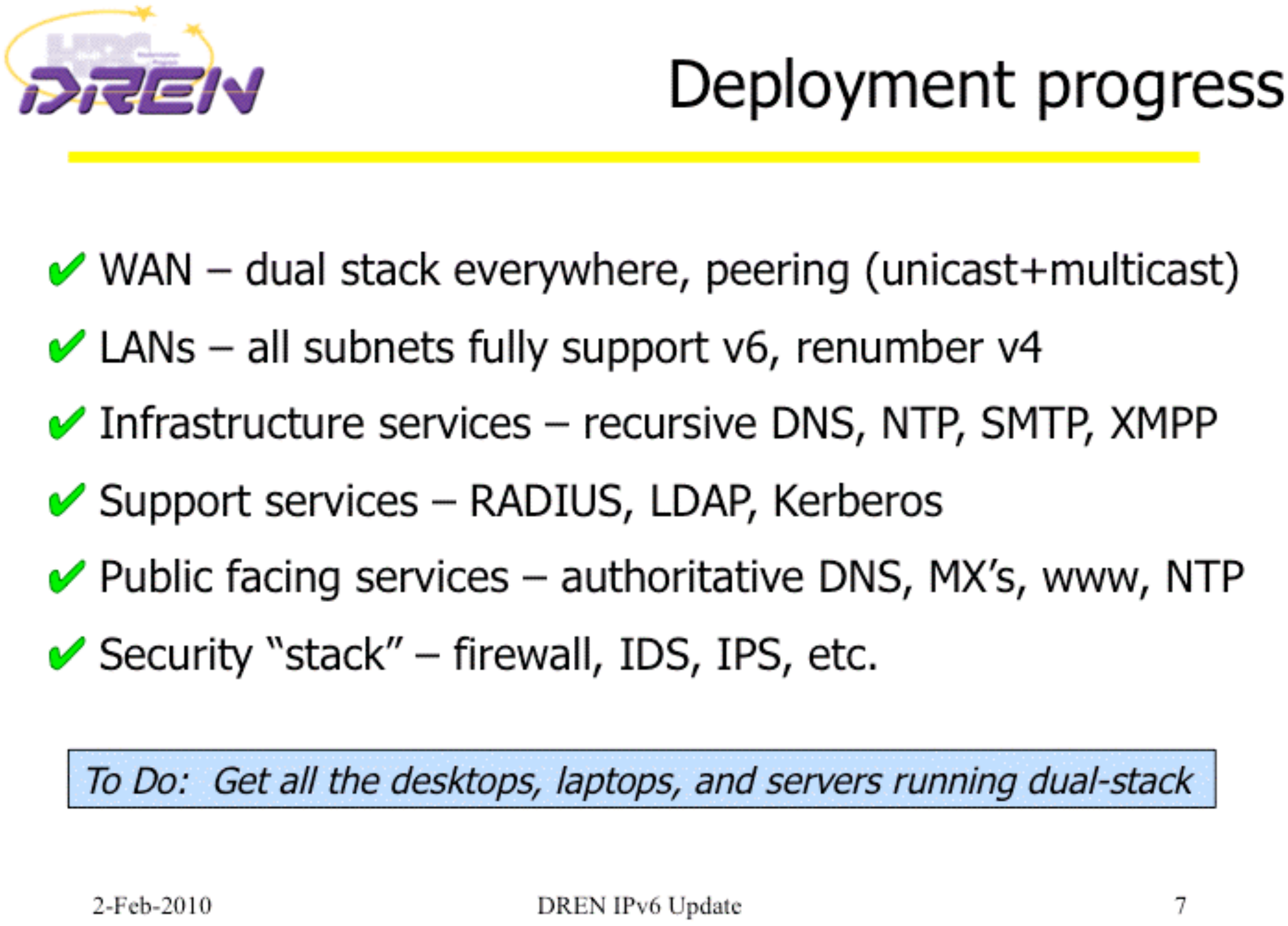
Internet2 Joint Techs, Winter 2010
2 Feb, 2010
Salt Lake City, UT

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

2-Feb-2010 DREN IPv6 Update 1

http://www.internet2.edu/presentations/jt2010feb/20100202-broersma.pdf

DREN Is *Widely* Using IPv6 (2)



Deployment progress

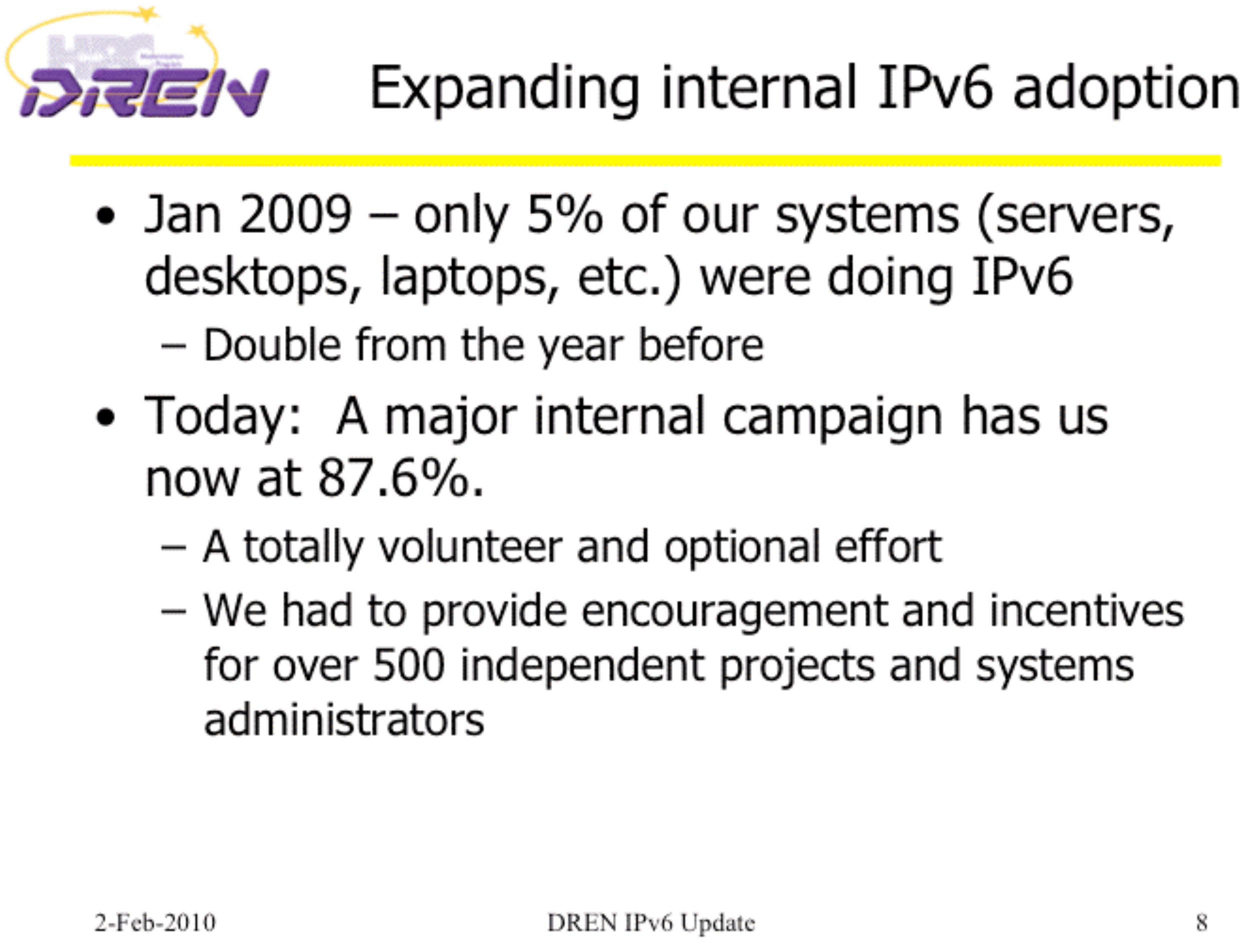
- ✓ WAN – dual stack everywhere, peering (unicast+multicast)
- ✓ LANs – all subnets fully support v6, renumber v4
- ✓ Infrastructure services – recursive DNS, NTP, SMTP, XMPP
- ✓ Support services – RADIUS, LDAP, Kerberos
- ✓ Public facing services – authoritative DNS, MX's, www, NTP
- ✓ Security "stack"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

To Do: Get all the desktops, laptops, and servers running dual-stack

2-Feb-2010 DREN IPv6 Update 7

http://www.internet2.edu/presentations/jt2010feb/20100202-broersma.pdf

DREN Is *Widely* Using IPv6 (3)



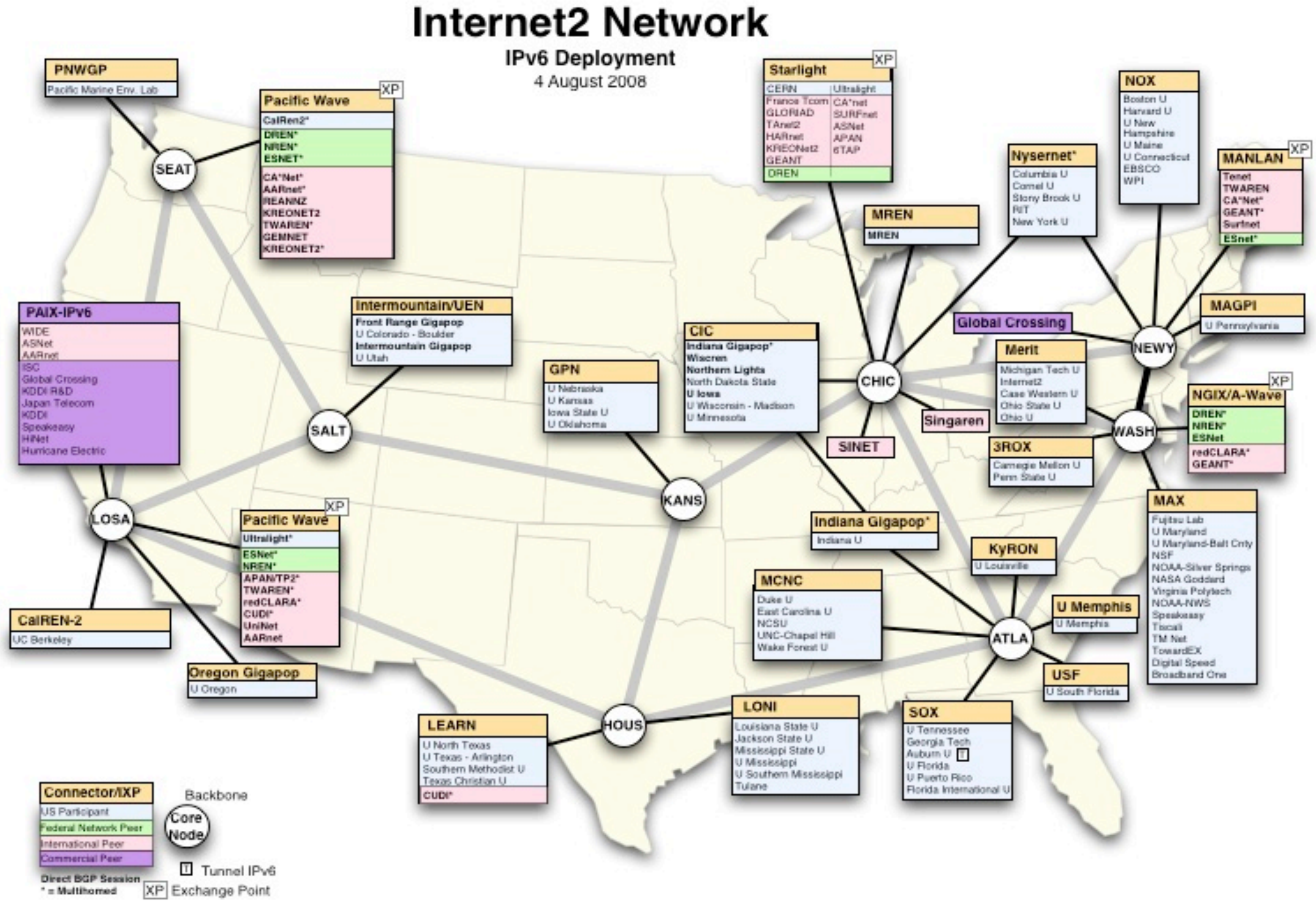
Expanding internal IPv6 adoption

- 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

2-Feb-2010 DREN IPv6 Update 8

http://www.internet2.edu/presentations/jt2010feb/20100202-broersma.pdf

Many Internet2-Connected Sites Are IPv6 Enabled

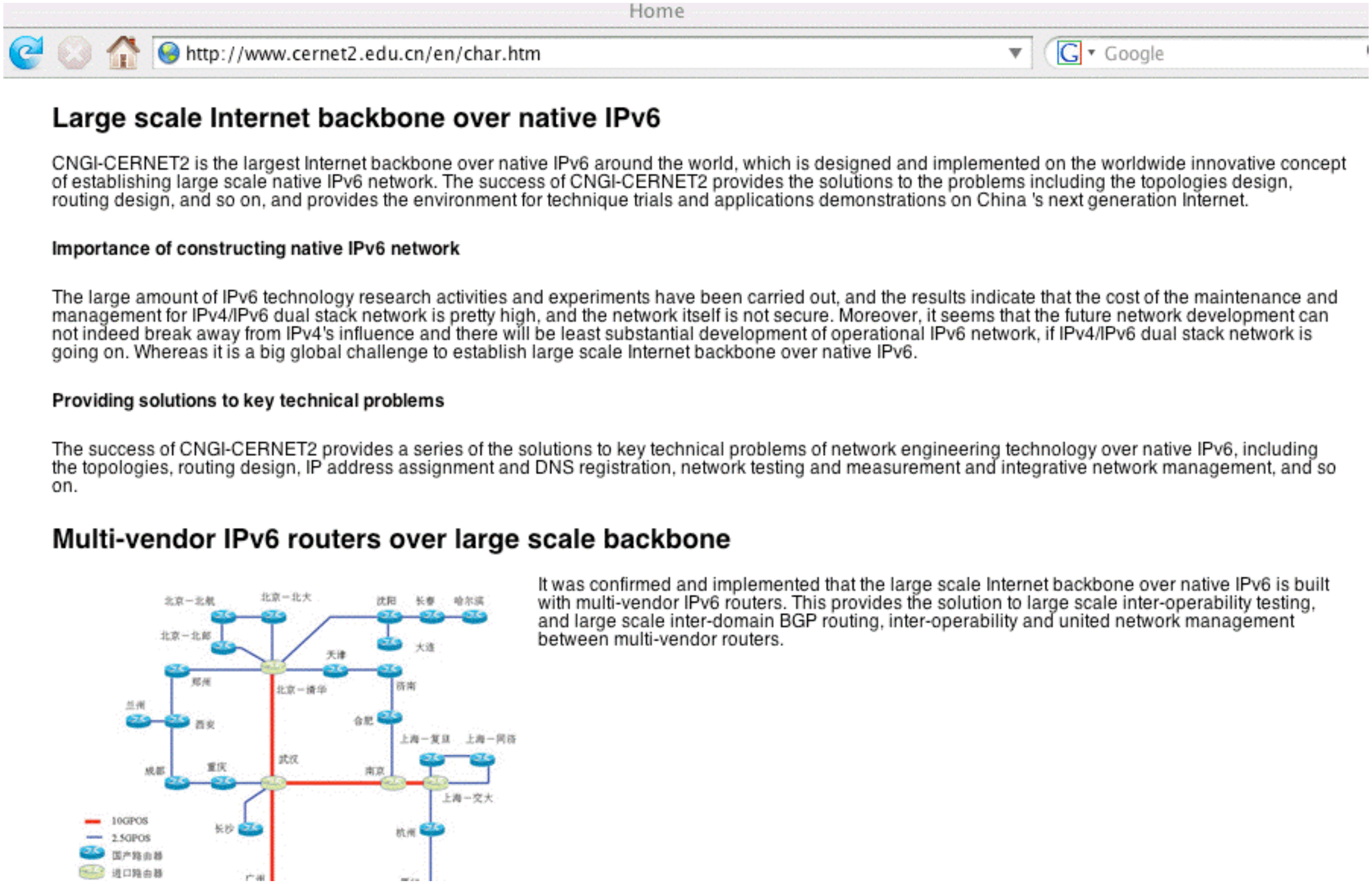


Internet2 Network

IPv6 Deployment

4 August 2008

CERNET2 (China) Is IPv6 *ONLY*



Large scale Internet backbone over native IPv6

CNGI-CERNET2 is the largest Internet backbone over native IPv6 around the world, which is designed and implemented on the worldwide innovative concept of establishing large scale native IPv6 network. The success of CNGI-CERNET2 provides the solutions to the problems including the topologies design, routing design, and so on, and provides the environment for technique trials and applications demonstrations on China 's next generation Internet.

Importance of constructing native IPv6 network

The large amount of IPv6 technology research activities and experiments have been carried out, and the results indicate that the cost of the maintenance and management for IPv4/IPv6 dual stack network is pretty high, and the network itself is not secure. Moreover, it seems that the future network development can not indeed break away from IPv4's influence and there will be least substantial development of operational IPv6 network, if IPv4/IPv6 dual stack network is going on. Whereas it is a big global challenge to establish large scale Internet backbone over native IPv6.

Providing solutions to key technical problems

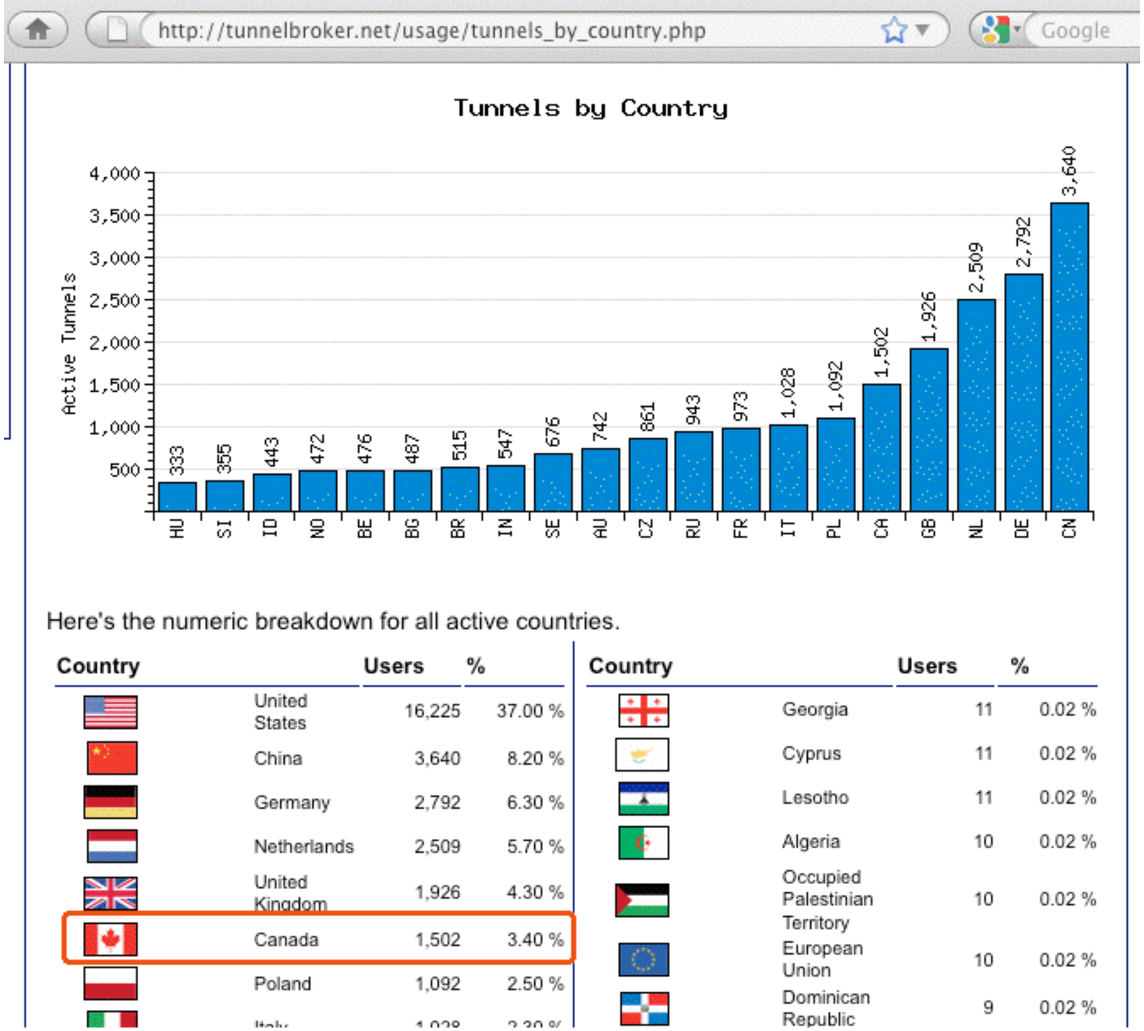
The success of CNGI-CERNET2 provides a series of the solutions to key technical problems of network engineering technology over native IPv6, including the topologies, routing design, IP address assignment and DNS registration, network testing and measurement and integrative network management, and so on.

Multi-vendor IPv6 routers over large scale backbone

It was confirmed and implemented that the large scale Internet backbone over native IPv6 is built with multi-vendor IPv6 routers. This provides the solution to large scale inter-operability testing, and large scale inter-domain BGP routing, inter-operability and united network management between multi-vendor routers.

http://www.cernet2.edu.cn/en/char.htm

Hurricane Electric Is Serving 44,383 IPv6 Tunnels Worldwide



Here's the numeric breakdown for all active countries.

Country	Users	%	Country	Users	%
United States	16,225	37.00 %	Georgia	11	0.02 %
China	3,640	8.20 %	Cyprus	11	0.02 %
Germany	2,792	6.30 %	Lesotho	11	0.02 %
Netherlands	2,509	5.70 %	Algeria	10	0.02 %
United Kingdom	1,926	4.30 %	Occupied Palestinian Territory	10	0.02 %
Canada	1,502	3.40 %	European Union	10	0.02 %
Poland	1,092	2.50 %	Dominican Republic	9	0.02 %
Italy	1,028	2.30 %			

http://tunnelbroker.net/usage/tunnels_by_country.php

The Bad Guys/Gals Are Also Interested in IPv6

- Some of the reasons why the Bad Guys/Bad Gals are interested in IPv6 is that at many sites:

  -- IPv6 network traffic isn't tracked on par with IPv4 traffic (if it is monitored at all), so IPv6 can be a great covert communications channel

  -- IPv4 security measures (such as perimeter firewalls or filter ACLs) may not be replicated for IPv6

  -- Law enforcement hasn't ramped up to deal with online badness that involves IPv6 (example: I suspect that few if any cybercrime cops have IPv6 cybercrime expertise, or even IPv6 connectivity!)

“What About IPv6 Applications Other Than HTTP?”

Email and IPv6

- While at least some people are very excited about the thought of using IPv6 for the **web**, for some reason there seems to be a lot **less** excitement about using IPv6 for **email**.

- Thus, while many mainstream mail software products support IPv6, relatively few mail administrators apparently bother to enable IPv6 support.

- But some sites ARE deploying IPv6-accessible mail servers right now. For example…

Sample Institutional IPv6 Enabled MX

```
% dig ucla.edu mx +short
5 smtp.ucla.edu.

% dig smtp.ucla.edu a +short
169.232.46.240
169.232.46.241
169.232.46.242
169.232.46.244
etc.


% dig smtp.ucla.edu aaaa +short
2607:f010:3fe:302:1013:72ff:fe5b:60c3
2607:f010:3fe:102:101c:23ff:febe:116e
2607:f010:3fe:102:101c:23ff:febf:cfa7
2607:f010:3fe:102:101c:23ff:fed0:918c
etc.
```

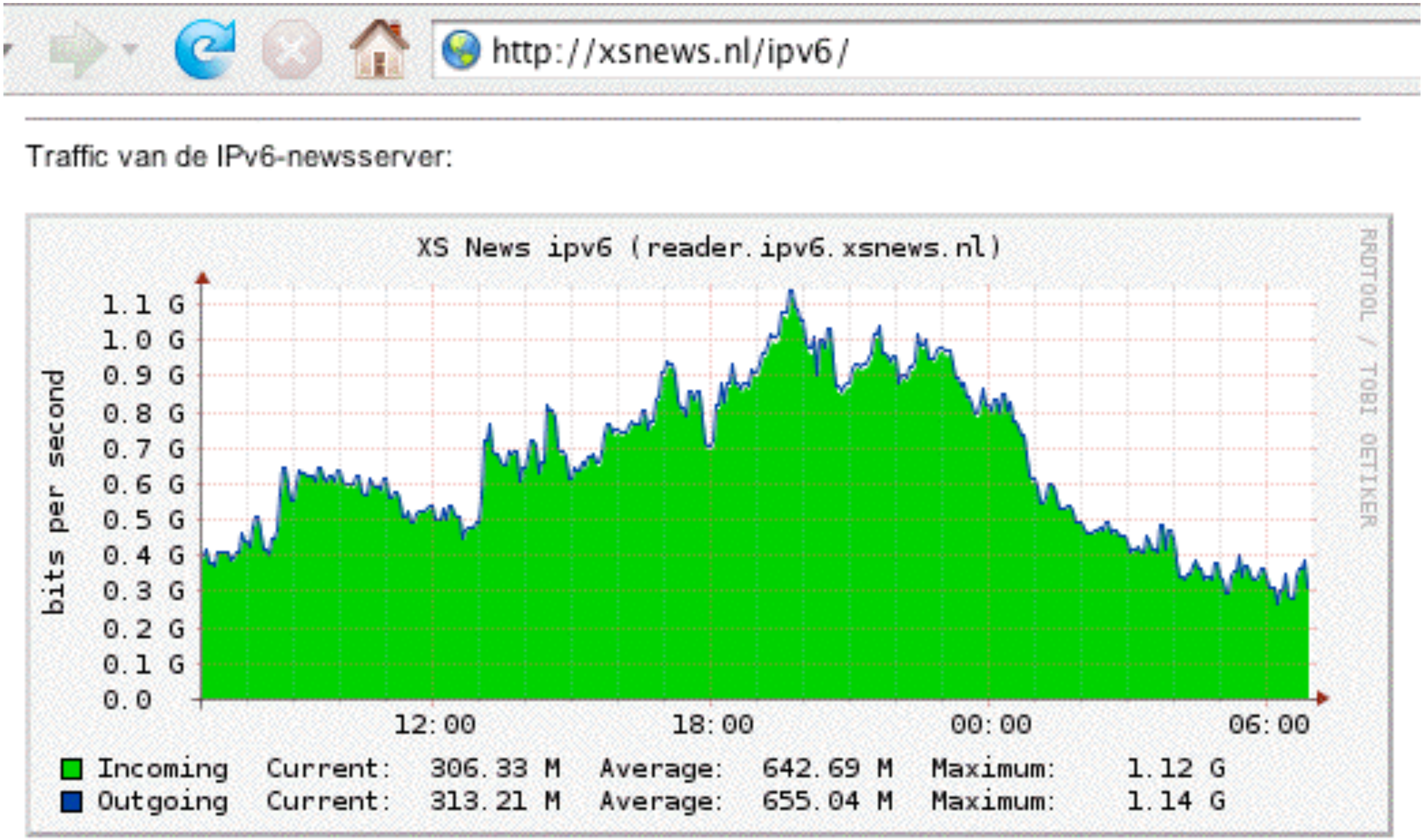
Enabling IPv6 In postfix Is Pretty Easy

- Get postfix 2.7 (or whatever's the latest stable version) from http://www.postfix.org/download.html
- Review http://www.postfix.org/IPV6_README.html
- When configuring for IPv6, in /etc/postfix/main.cf, set inet_protocols = ipv6, ipv4 (if you're dual stacking)
- In /etc/postfix/main.cf set the address you want to use for outgoing IPv6 SMTP connections; for EXAMPLE only:
  ```
  smtp_bind_address6 = 2001:468:d01:d6::80df:d617
  ```
- Check your config and start postfix; typically:
  ```
  /usr/sbin/postfix check
  /usr/sbin/postfix start
  ```
- Confirm that you can connect OK to your IPv6 smtpd:
  ```
  % telnet 2001:468:d01:d6::80df:d617 25
  quit
  ```

IPv6 and DNS Blocklists

- DNS blocklists, such as those offered by Spamhaus, are a key anti-abuse tool in today's IPv4-dominated Internet, directly blocking spam while also "encouraging" ISPs to employ sound anti-abuse practices.
- Virtually all sites that use DNS-based blocklists rely on rbldnsd (see www.corpit.ru/mjt/rbldnsd/rbldnsd.8.html). rbldnsd does NOT support IPv6 records at this time. :-(
- Spamhaus does not maintain any substantive IPv6 **blocklists**; Spamhaus has, however, just recently announced a new IPv4 **and** IPv6 **whitelist** (see http://www.spamhauswhitelist.com/en/rationale.html)
- Some mail receivers may be afraid to enable SMTP via IPv6 w/o blocklist support, but so far there has been negligible spam via IPv6 (in my experience).

IPv6 Is Also Carrying A Lot of Usenet Traffic

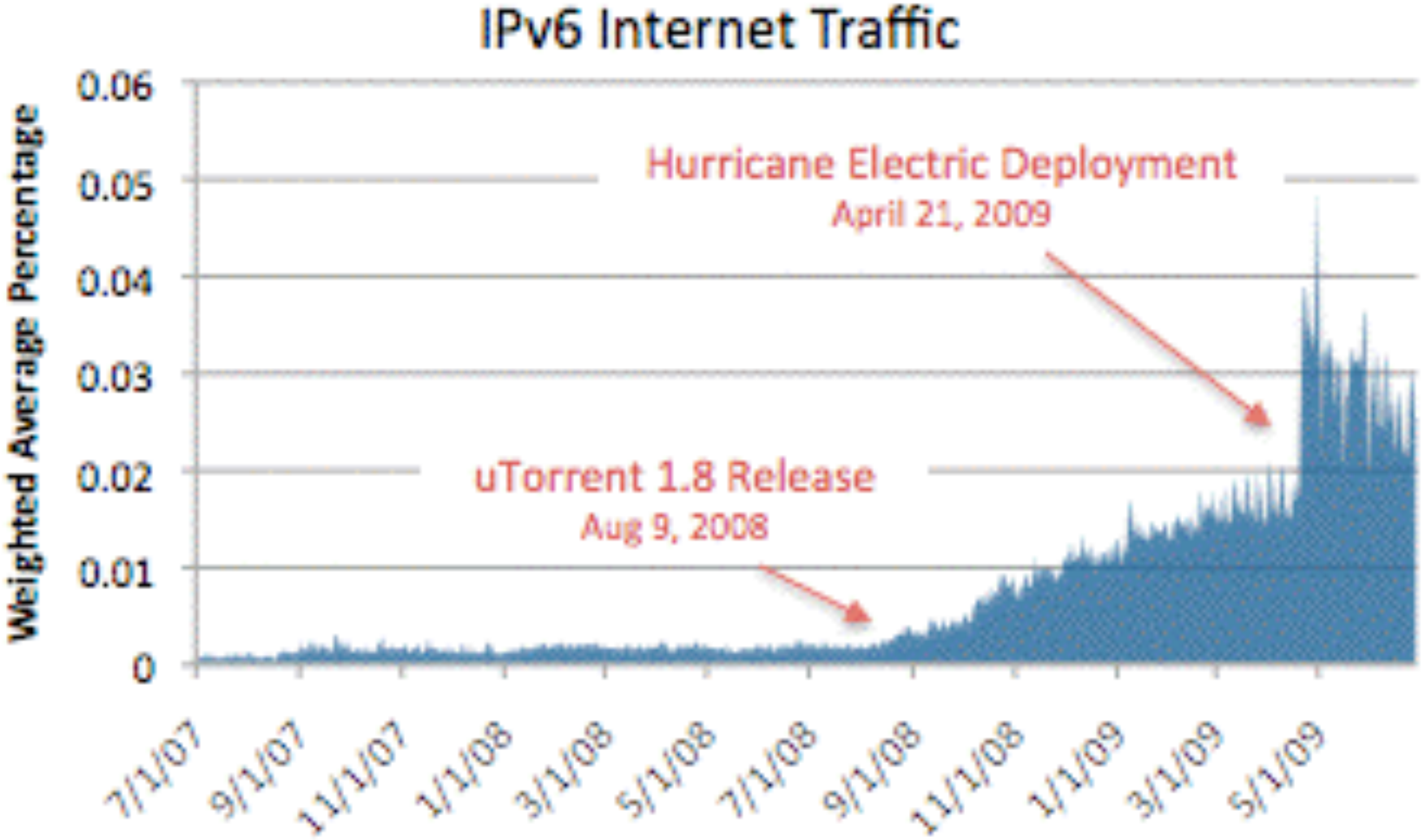


http://xsnews.nl/ipv6/

Traffic van de IPv6-newsserver:

```
🟩 Incoming   Current:   306.33 M   Average:   642.69 M   Maximum:    1.12 G
🟦 Outgoing   Current:   313.21 M   Average:   655.04 M   Maximum:    1.14 G
```

IPv6 Is Also Being Used for P2P



See http://asert.arbornetworks.com/2009/09/who-put-the-ipv6-in-my-internet/

What About YOU?
YOU Should Be Getting Ready for IPv6!

- If you're not currently deploying IPv6 locally, or at least experimenting with IPv6 in a lab setting, the time has come for you to begin to do so.

- Deployment can be incremental. You can take baby steps, you don't need to boil the ocean on day one.

- What you can't do is put off deploying IPv6 forever.

Technical Challenge 3: There *Are* Some Legitimate Potential Obstacles To Deploying IPv6 (At Some Sites)

For example, does your ISP offer native IPv6 Internet transit connectivity?

Native IPv6 Connectivity

- Your site needs IPv6 connectivity.
- Native IPv6 connectivity is strongly preferred. Native IPv6 connectivity is the IPv6 analog of normal IPv4 connectivity, and would ideally come from your current network service provider.
- **Unfortunately, some sites may currently be getting their IPv4 Internet transit from network service providers who may not yet be offering native IPv6 transit.**
- In those cases, you can add IPv6 by adding a second provider:

  If necessary, you can use one network service provider for your IPv4 Internet connectivity, and add another provider for your IPv6 Internet connectivity.

IPv6 Transit Providers (e.g., NSPs)

- There are many major network service providers which DO offer IPv6 connectivity; see the list that's at http://www.sixxs.net/faq/connectivity/?faq=ipv6transit

- That list includes most of the usual suspects, including:

  AS701 Verizon
  AS1239 Sprint
  AS2686 AT&T
  AS2914 NTT/Verio
  AS3356 Level3
  AS6939 Hurricane Electric

  plus many others…

Manually Configured IPv6 Tunnels

- Another alternative might be to arrange for a manually configured IPv6 tunnel from an IPv6 tunnel broker (although you'd **really** be better off adding native IPv6 connectivity from a second network service provider).

- Free tunneled IPv6 connectivity is available from a variety of providers, including most notably:
  -- Hurricane Electric, http://tunnelbroker.net/
  -- SixXS, https://www.sixxs.net/main/

- When establishing a manually configured IPv6 tunnel, beware of tunneling to a very distant tunnel endpoint -- all your traffic will have to make that long trip, and that will add (potentially substantial) latency. Keep tunnels as short as possible!

IPv6 and the IPv6-Readiness of Key Outsourced Service Providers

Another Major Potential Stumbling Block: Non-IPv6 Content Delivery Networks (CDNs)

- Many US dot gov web sites (and key commercial web sites) use Akamai (or another CDN) in order to handle huge online audiences and deliver good performance worldwide.
- For example, www.irs.gov is actually just a cname for www.edgeredirector.irs.akadns.net; whois confirms that akadns.net actually belongs to Akamai:

  Registrant:
  Akamai Technologies
  […]
  Domain name: AKADNS.NET

- If Akamai doesn't do IPv6, will major Akamai customers (such as Apple, Cisco, Microsoft, RedHat, the Whitehouse, etc.) be able to do so without them?

But Speaking of Akamai, Akamai *Is* Reportedly Working On IPv6…

- I'm happy to report that Akamai is now reportedly working on IPv6-ifying its CDN infrastructure. See, for example, the coverage in:

  "Akamai: Why Our IPv6 Upgrade Is Harder Than Google's,"
  http://www.networkworld.com/news/2010/091610-akamai-ipv6.html
  September 16th, 2010

The Issue Isn't Just Web CDNs…

- A growing number of sites also outsource their email operations.

- Unfortunately some email-as-a-service and some cloud-based spam filtering services don't support IPv6, thereby limiting the ability of their customers to integrate IPv6 into their existing IPv4-based services.

- CDNs and outsourced email and spam filtering services aren't the only reason why IPv6 adoption has been slow at some major Internet sites, but it is certainly an important stumbling block that will need to get resolved.

- Other issues are likely network hardware-related.

IPv6 Hardware and Software Support

Network Middleboxes Can Be A Major IPv6 PITA

- The more I talk with sites about IPv6, the more I hate network middleboxes such as firewalls or network traffic load balancers. Sometimes those devices simply do not understand IPv6 at all.
- Other times they may have a primitive or incomplete implementation of IPv6, or require users to license an expensive "enhanced" software image to support IPv4 and IPv6.
- In general, I'd recommend moving firewalls as close to the resources they're protecting as possible (e.g., down to a subnet border, or even down to the individual ethernet port level), assuming you can't get rid of them altogether
- If you need to pay extra for IPv6 support in devices, complain to your vendor or vote with your purchase orders

A Potential Major ISP Stumbling Block: Broadband Customer Premises Equipment (CPE)

- Some broadband CPE also does NOT support IPv6. Imagine having millions of customer access point devices that need to be replaced, to say nothing of customer purchased and deployed wireless access points.

- One list of products that have at least some IPv6 support can be found at http://www.getipv6.info/index.php/Broadband_CPE

- See also the work of the IETF Home Gateway Working Group (e.g., see http://www.ietf.org/proceedings/78/homegate.html)

Yet Another Potential Major ISP Stumbling Block: Uneven Native OS Support for DHCPv6

- ISPs need to be able to map complaints (reported in the form of IP addresses and time stamps with time zone information) to actual customer identities.
- For customers who are given IPv4 addresses via DHCPv4 this is readily and routinely done today.
- In an IPv6 environment, things get trickier. Support for DHCPv6 is incomplete (native support for DHCPv6 is missing in Mac OS X and Windows XP, for example).
- One could use alternative mechanisms for assigning IPv6 addresses to end user systems, such as stateless autoconfiguration ("SLAAC"), however SLAAC does not allow ISPs to map IPv6 addresses to individual customers.
- Incomplete DHCPv6 support is thus another potential major roadblock to widespread IPv6 deployment.

Accessing IPv4-Only Content Once We Run Out of Globally Routable IPv4 Addresses

IPv6 to IPv4 Gateways and/or Large Scale ("Carrier Grade") NAT

- While current IPv6 transition plans typically assume IPv6 deployment alongside IPv4 (e.g., deployment of a so-called "**dual-stack**" configuration), that model will not help us once we're completely out of globally routable IPv4 addresses.

- Once we're completely out of globally routable IPv4 addresses, new end users will still need some way to access legacy content that's still being offered only via IPv4.

- One solution would be to give those customers <u>only</u> an IPv6 address, and then use an IPv6-to-IPv4 gateway device to bridge IPv4-only content to IPv6-only users.

An Example of an IPv6 to IPv4 Gateway

- One example of an IPv6 to IPv4 gateway is IVI, see

  "CERNET IVI Translation Design and Deployment for the IPv4/IPv6 Coexistence and Transition," January 6th, 2010, http://tools.ietf.org/html/draft-xli-behave-ivi-07 and

  "Transition to IPv6: IVI in the University Campus," Nov 3rd, 2010 http://events.internet2.edu/2010/fall-mm/agenda.cfm?go=session&id=10001342&event=1159 and

  http://www.ivi2.org/ has IVI patches for Linux 2.6.18 (Yes, that *is* a relatively old Linux kernel dating from 2006-2007; the latest stable Linux kernel is now 2.6.36, available as of 2010-10-20).

Large Scale ("Carrier Grade") NAT

- Another option would be to give customers an IPv6 address and a private (RFC1918) IPv4 address that communicates with the world of globally routable IPv4 addresses via large scale ("carrier grade") NAT.
- Large scale NAT, if deployed, will likely end up being pretty miserable:
  -- some applications simply won't work from a NAT'd IP address
  -- tracking down abuse complaints will become difficult or impossible
  -- users will end up sharing their neighbor's bad reputations
  -- we'll lose Internet transparency and the flexibility and generativity that network transparency gives us

You May Already Use NAT

- NAT makes it possible for multiple workstations to all use a single shared globally routable IPv4 address, and many home users connect a home network to their broadband provider via one of those little Linksys wireless access points. That's an example of a NAT box.

- If all you do is browse the web or use a web email service such as Hotmail, or Yahoo! Mail, or Gmail, NAT may indeed work just fine for your relatively simple needs.

- On the other hand, if you want to do anything "exotic" (such as using H.323 Internet video conferencing), or if you want to run a server, NAT will typically NOT work.

Tracking Abuse

- Many of us care a great deal about tracking abusive online traffic. Tracking abuse will get much harder in a world that makes widespread use of large scale NAT.
- Most dynamic IPv4 addresses are assigned via DHCP. A single IPv4 address will often be shared by multiple customers over the span of multiple hours or days. Mapping abuse associated with a dynamic IP of that sort requires **TWO** things: an IP address and a time stamp (along with time zone information).
- If ISPs begin to deploy large scale NAT (also known as “Carrier Grade NAT”), abuse complaints will suddenly need **THREE** things: (i) the IP address, (ii) the time stamp (and time zone information), ***AND* (iii) the source port.**
- **Most complaints will not include source port information, and as such, will prove impossible to track down and fix.**

Sharing Reputation

- Or lets assume that you suddenly find that you can't access some servers or web sites -- you've been block listed! Why? "You" (or *someone else who's sharing your NAT's public IP address!*), has been bad.
- The external site blocking you has no way of knowing that it was someone else (and not you) who was bad – they only see abusive connections from an IP address. They then take what seems to be reasonable defensive steps to protect themselves: they block access from that IP.
- Regretably, when they block that IP address, while they succeed in blocking the source of the abuse they're seeing, they may ALSO block scores or even hundreds of other innocent users who happen to be sharing that large scale NAT public address, including you. Yech. :-(

End-To-End Transparency

- End-to-end transparency is the concept that networks should just dutifully deliver packets, and not filter or rewrite some of them.
- While Internet transparence is less often mentioned than imminent IPv4 address exhaustion as a reason why we need to deploy IPv6, transparency is nonetheless a very important underlying motivation for IPv6, and something that's lost in a NAT'd environment.
- If you'd like to read about the importance of end-to-end transparency, some excellent starting points are:
  -- RFC2775, "Internet Transparency," B. Carpenter, February 2000, http://tools.ietf.org/rfc/rfc2775.txt
  -- RFC4924, "Reflections on Internet Transparency," B. Aboba and E. Davies, July 2007, http://tools.ietf.org/rfc/rfc4924.txt

Things As Basic As DNS Can Also “Break” In Conjunction with IPv6

Basic IPv6 DNS Is Fairly Similar to IPv4 DNS

- In IPv4 world, servers and other hosts use “A” records to map fully qualified domain names to dotted quads:

  ```
  % dig network-services.uoregon.edu a +short
  128.223.60.21
  ```

- In IPv6 world, we use “AAAA” (“quad A”) records instead of A records to map fully qualified domain names to IPv6 addresses:

  ```
  % dig network-services.uoregon.edu aaaa +short
  2001:468:d01:3c::80df:3c15
  ```

Inverse Address Records (PTRs) Are Also Similar

- IPv4 world:

```
% dig -x 128.223.60.21 +short
network-services.uoregon.edu.
```

- IPv6 world:

```
% dig -x 2001:468:d01:3c::80df:3c15 +short
network-services.uoregon.edu.
```

- If you need a web-accessible IPv6 dig interface, try http://www.digwebinterface.com/

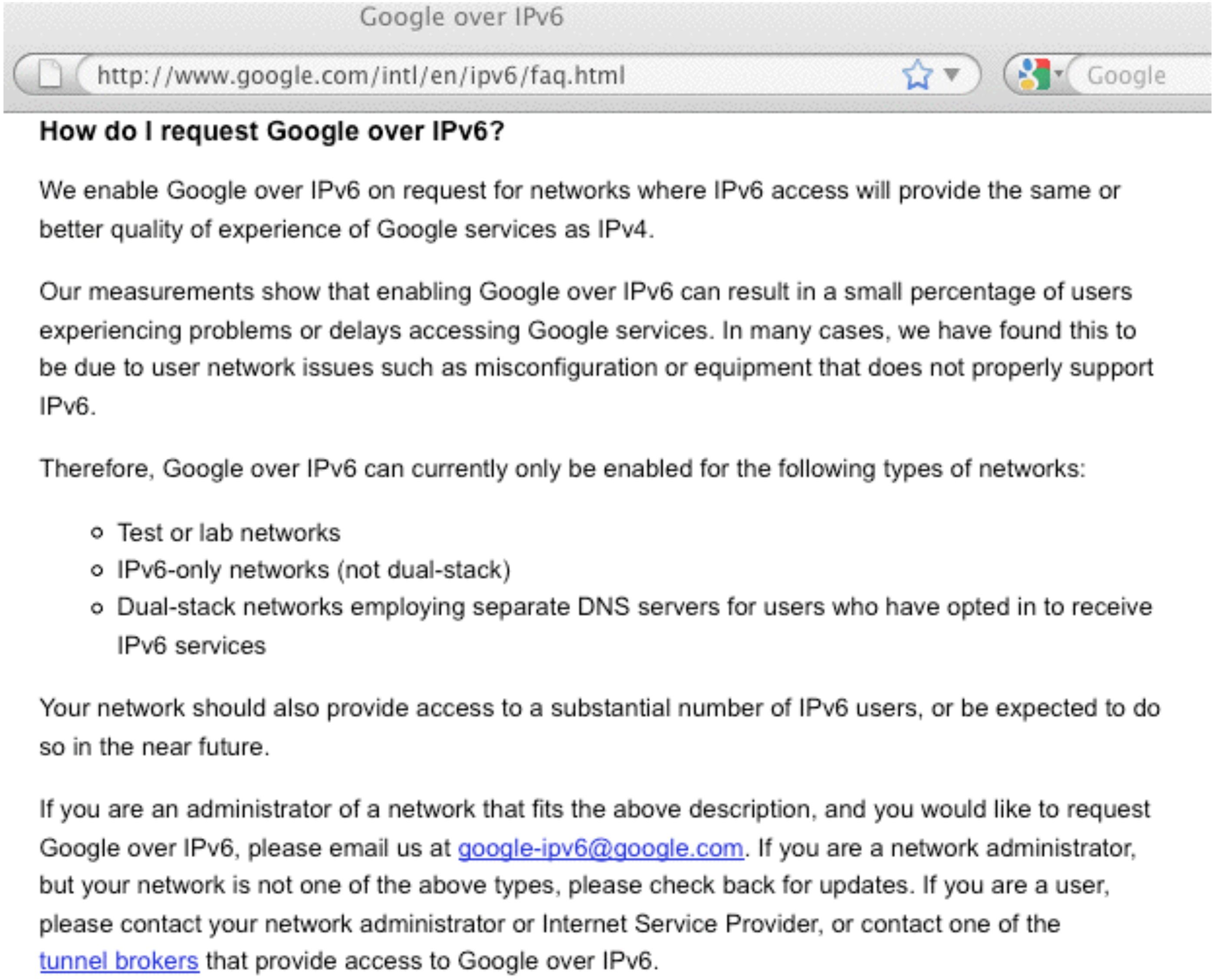
Complications: IPv6 *AND* IPv4 Domain Names

- If a fully qualified domain name (such as network-services.uoregon.edu) is bound to **both** IPv4 and IPv6 addresses, which one should gets used? **Which one** should be "preferred?" The IPv6 one or the IPv4 one?
- This may be determined by the application (e.g., it may ask for both, and then use its own internal precedence information to determine which it will use), or by the DNS server (hypothetically it might just give you an IPv6 address for a host and then stop).
- This would be a problem if you advertise an IPv6 address for a host but then don't actually offer IPv6 connectivity for that AAAA, or if the user asks for an IPv6 address but doesn't actually have IPv6 connectivity after all.
- Let's consider an example of this: Google.

Enabling IPv6 DNS For Google <u>By Default</u>

- Assume you're Google. Also assume you'd like to have http://www.google.com reachable via IPv4 *OR* IPv6. That is, you'd like IPv6-enabled users to access your site via IPv6, while allowing IPv4-only users to still use IPv4.
- When you try doing that, however, you quickly find out that there are some users that *think* they can do IPv6, while not actually being able to do so.
- When that happens, IPv6 connectivity gets tried first (only to fail). It takes time (20+ secs!) for those failures to occur. After each failure, IPv4 connectivity gets tried as a fall-back plan, but users quickly get grumpy if their browsing experience is repeatedly slowed by one failed IPv6 connection attempt after another.
- Result? Google only enables automatic IPv6 resolution of Google websites for IPv6-capable networks **<u>by request</u>**.

Enabling IPv6 Resolution <u>By Request</u>



How do I request Google over IPv6?

We enable Google over IPv6 on request for networks where IPv6 access will provide the same or better quality of experience of Google services as IPv4.

Our measurements show that enabling Google over IPv6 can result in a small percentage of users experiencing problems or delays accessing Google services. In many cases, we have found this to be due to user network issues such as misconfiguration or equipment that does not properly support IPv6.

Therefore, Google over IPv6 can currently only be enabled for the following types of networks:

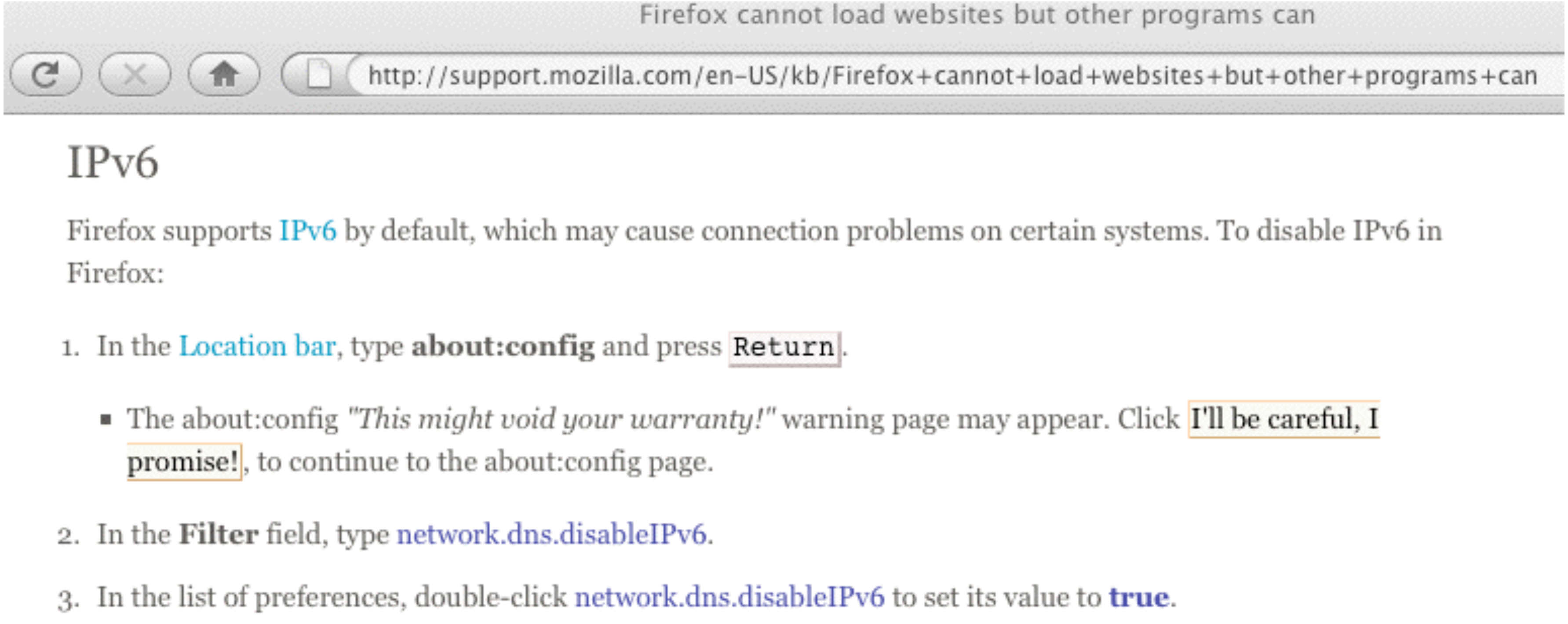
- Test or lab networks
- IPv6-only networks (not dual-stack)
- Dual-stack networks employing separate DNS servers for users who have opted in to receive IPv6 services

Your network should also provide access to a substantial number of IPv6 users, or be expected to do so in the near future.

If you are an administrator of a network that fits the above description, and you would like to request Google over IPv6, please email us at google-ipv6@google.com. If you are a network administrator, but your network is not one of the above types, please check back for updates. If you are a user, please contact your network administrator or Internet Service Provider, or contact one of the tunnel brokers that provide access to Google over IPv6.

[Of course, "by request" doesn't scale particularly well...]

Default IPv6 DNS Support Can Also Be An Issue for Some Web Browsers


Firefox cannot load websites but other programs can
http://support.mozilla.com/en-US/kb/Firefox+cannot+load+websites+but+other+programs+can

IPv6

Firefox supports IPv6 by default, which may cause connection problems on certain systems. To disable IPv6 in Firefox:

1. In the Location bar, type **about:config** and press `Return`.
   - The about:config *"This might void your warranty!"* warning page may appear. Click I'll be careful, I promise!, to continue to the about:config page.
2. In the **Filter** field, type network.dns.disableIPv6.
3. In the list of preferences, double-click network.dns.disableIPv6 to set its value to **true**.

[Take away? If you decide you're going to do IPv6, do it, don't partially do it and leave things halfway up and halfway down]

PTR Records for *Non-Static* IPv6 Addresses?

- Inverse address records ("PTRs") map IP addresses to domain names. E.G., 128.223.142.32 --> shell.uoregon.edu
- We ***can*** create static inverse address records for static IPv6 addresses assigned to servers, that's not a problem.
- Unfortunately, there's isn't community consensus around how to handle inverse address records ("PTR") records for IPv6 addresses assigned via SLAAC or DHCPv6.
- No one wants to create 18,446,744,073,709,551,616 inverse address records, one for each IP in each /64! It would take forever, and wouldn't make any sense (most of those PTRs would never even be queried!)
- Options such as dynamic DNS are sometimes suggested as a solution (yech), as well as wildcarding (yech), as well as creating inverse address records on the fly (yech).
- This is yet another unsolved IPv6 challenge.

Why Do I Care About IPv6 PTRs?

- Many cyber crime investigators will look at the PTRs of IP addresses they're interested in for clues as so who may be responsible for those IP addresses.

- Obviously PTRs can potentially be forged, so they aren't foolproof, but they still can be one additional helpful bit of information in at least some cases.

- Given the limitations of IPv6 PTR assignment processes, we may end up just needing to just rely on "whois" to map IPv6 IP addresses to responsible parties instead.

Using Whois With IPv6

- Whois for IPv6 works just as it does in IPv4.

- For example, if you wanted to know who has an IPv6 netblock in 2001:468:: and you have a Linux box or Mac, pop up a terminal window and enter…

  ```
  % whois -h whois.arin.net \> \ 2001:468::
  ```

  You can also drill down on particular objects (such as an IPv6 address or particular named IPv6 netblock):

  ```
  % whois -h whois.arin.net NET6-2001-468-D00-1
  ```

IPv6 Multihoming and Route Table Bloat

There Are Other IPv6 Issues, Too
(Even If No One Has Told You About Them)

- As daunting as the preceding issues may seem, there are other IPv6 deployment issues that have also come up over the years -- even if you've never heard of them.

- For example, IPv6 was ***supposed*** to control route table growth through the use of hierarchical and readily aggregate-able IPv6 address assignments, but that just hasn't worked out. We've never figured out how to handle IPv6 multihoming in a clean way while avoiding route table bloat.

- Since you probably don't spend much time worrying about route table growth, let me explain the pressure the community faces in that area.

Controlling Route Table Bloat

- RFC4984 (http://www.ietf.org/rfc/rfc4984.txt) states,

  "[...] routing scalability is the most important problem facing the Internet today and must be solved [...]"

What Is “Routing?”

- You may have wondered how packets know how to get from site A to site B. The answer is “routing.”
- When a server at a remote location has network traffic for a site, a series of hop-by-hop decisions get made: at each router, a packet needs to decide where to go to get closer to its ultimate destination. A packet comes in on one interface, and may have a choice of two, three, or even a dozen or more outbound interfaces for the next step in its journey. Which path should it take next?
- Each router has a table of network IP address prefixes which point at outbound router interfaces, and that table guides packets on the next step of their journey.
- After the packet traverses that link, the process is then repeated again at the next router for the next link, etc

Most Little Sites: No Impact on Table Size

- If you're a small and simple site with just a single upstream provider, your upstream ISP may aggregate the network addresses you use with other customers it also services. Thus, the global routing table might have just a single table entry servicing many customers.
- Once inbound network traffic hits the ISP, the ISP can then figure out how to deliver traffic for customer A, traffic for customer B, etc. The ISP handles that -- the Internet doesn't need to know the "gory" local details
- Similarly, outbound, if you're a small site with just a single upstream provider, your choice of where to send your outbound traffic is pretty simple: you've only <u>got</u> one place you can send it. This allows you to set a "default route," sending any non-local traffic out to your ISP for eventual delivery wherever it needs to go.

Sites With Their Own IP Address Space

- Sometimes, however, sites have their own address space.
- For example, UO has the prefix 128.223.0.0/16, the IPv4 addresses 128.223.0.0--128.223.255.255.
- That address block is not part of any ISP's existing address space.
- If UO wants to receive traffic intended for those addresses, it needs to announce (or "advertise") that network address block to the world.
- When UO's route gets announced, each router worldwide adds that route to its routers' routing tables, and thus knows how to direct any traffic it may see that's destined for UO, to UO.
- Without that route, our address space would be unreachable.

Some Sites Have Multiple Prefixes

- Sometimes sites have more than one chunk of network address space. For example, Indiana University has 129.79.0.0/16, 134.68.0.0/16, 140.182.0.0/16, 149.159.0.0/16 149.160.0.0/14, 149.165.0.0/17, 149.166.0.0/16, 156.56.0.0/16, and 198.49.177.0/24, and thus IU has nine slots in the global routing table associated with those prefixes.
- Other sites may have a range of addresses which **could** be consolidated and announced as a single route, but some sites might intentionally "deaggregate" that space, perhaps announcing a separate route for each /24 they use. For example, BellSouth announces roughly 4,000 routes globally, even though it could aggregate those routes down to less than 300 routes if they were so inclined.

“So What? Who Cares About Route Growth?”

- **Each route in the global routing table need to be carried by routers at every provider in the world.**
- Each route in the route table consumes part of a finite pool of memory in each of those routers. When routers run out of memory, "Bad Things" tend to happen.
- Some routers even have relatively small fixed limits to the maximum size routing table they can handle (see http://tinyurl.com/route-table-overflow).
- Each route in the route table will potentially change whenever routes are introduced or withdrawn, or links go up or down. The larger the route table gets, the longer it takes for the route table to reconverge following these changes, and the more CPU the router requires to handle that route processing in a timely way

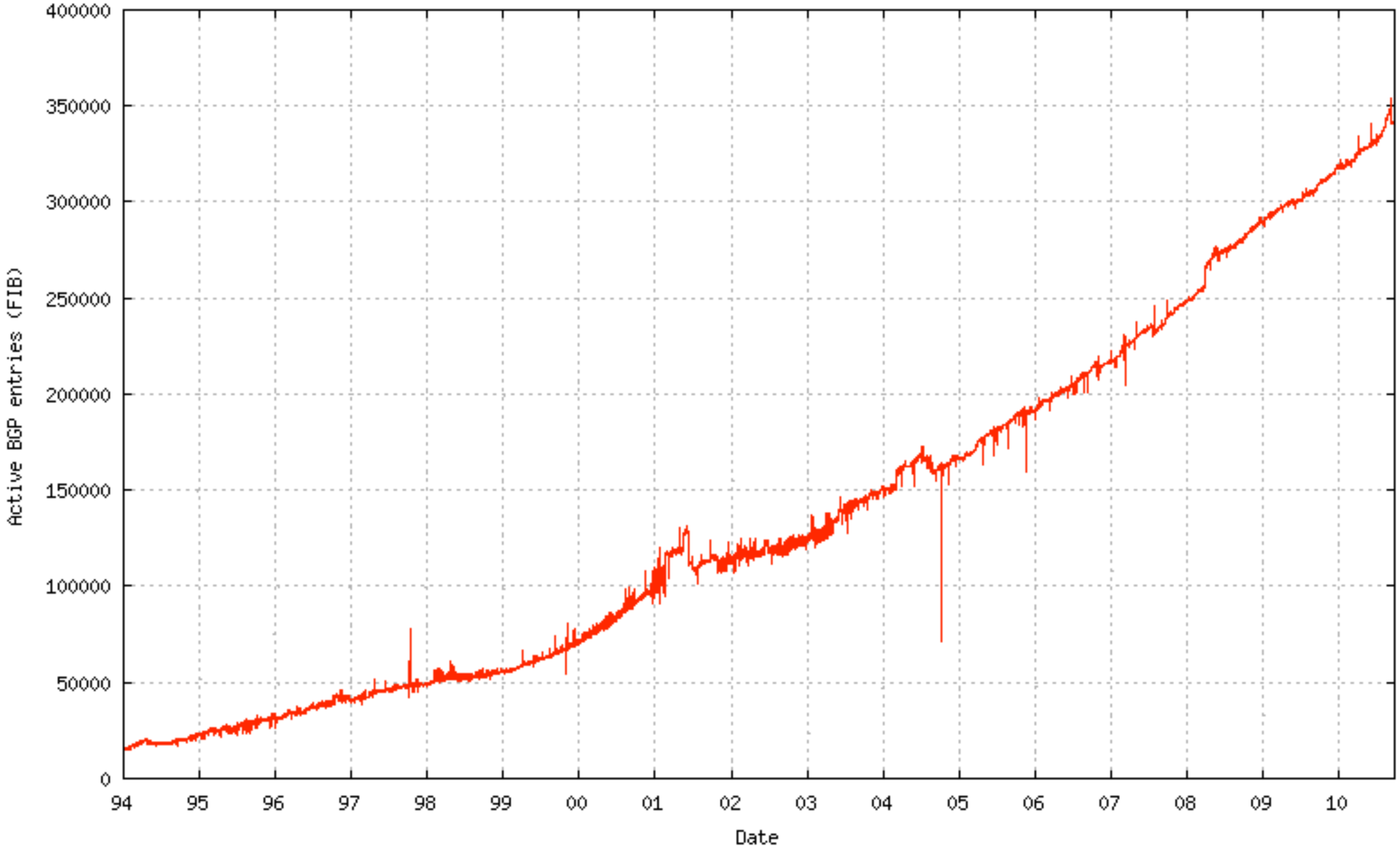
An Aside on Route Table Growth and Convergence

- There are some indications that we're getting luckier with route table performance than we might have expected; see Geoff Huston "BGP in 2009" talk from the ARIN Meeting in Toronto:

  https://www.arin.net/participate/meetings/reports/ARIN_XXV/PDF/Monday/Huston-bgp.pdf

But in Any Event,
The IPv4 Route Table Continues to Grow...

350,000



Source: http://bgp.potaroo.net/as6447/

IPv6 Was Supposed to Help Fix That

- When IPv6 was designed, address assignment was supposed to be hierarchical. That is, ISPs would be given large blocks of IPv6 address space, and they'd then use chunks of that space for each downstream customer, and only a single entry in the IPv6 routing table would be needed to cover ALL the space used by any given ISP and ALL their downstream customers (see RFC1887, "An Architecture for IPv6 Unicast Address Allocation")
- But now, let's pretend that my Internet connectivity is important to me, so I don't want to rely on just a single ISP -- I want to connect via multiple ISPs so that if one provider has problems, the other ISPs can still carry traffic for my site. This connection to multiple sites is known as "multihoming."

If I'm Multihomed, Whose Address Space Do I Use?

- When I get connectivity from sites A, B and C, whose address space would I announce? Address space from A? Address space from B? Address space from C? No...
  -- A doesn't want me to announce part of its address space via B and C
  -- B doesn't want me to announce part of its address space via A and C
  -- C doesn't want me to announce part of its address space via A and B.
- I need to <u>either</u> assign each host multiple addresses (e.g., one address from A, one from B, and one from C), <u>or</u> I need to get my own independent address space which I can use for all three ISPs, but which will then take up a slot in the global routing table.

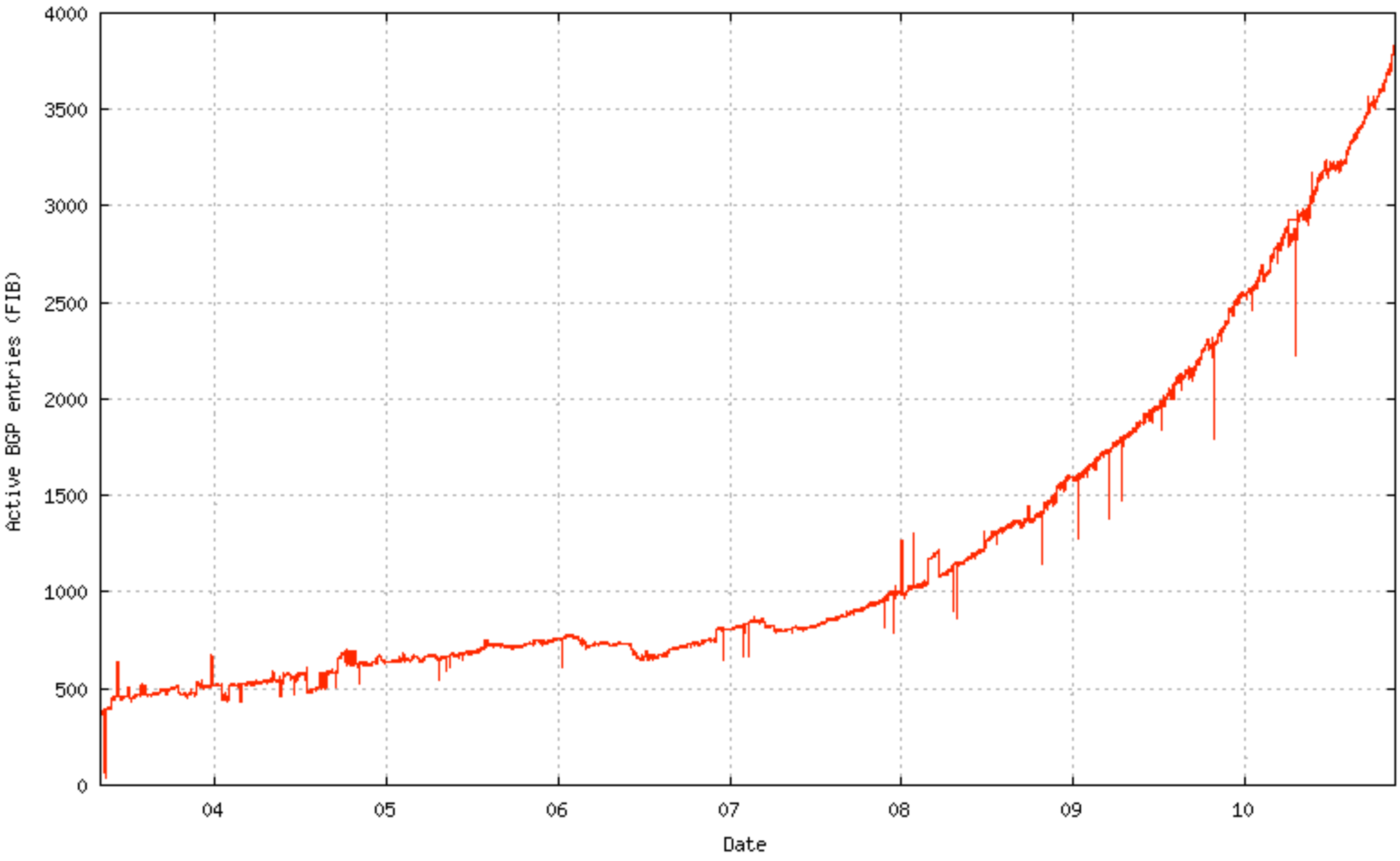
The Original Multiple IP Approach in IPv6

- The multiple IP approach was the original philosophical/ theoretical “answer” to this question in the IPv6 world.
- But if I assign multiple IPs to each host, one for each upstream ISP I connect to, how do I know which of those IP addresses I should use for outbound traffic generated by each host? Do I arbitrarily assign the address from A to some traffic? The address from B to other traffic? What about the address from C? (Hosts shouldn’t need to act like routers!)
- And which of those addresses do I map to my web site or other servers via DNS? Do I use just A’s address? Just B’s? Just C’s? All three of those addresses? What if one of my providers goes down? Will traffic failover to just the other two providers quickly enough?

The Multihoming Reality Today

- IPv6 multihoming without use of provider independent address space is one of the unsolved/open issues in the IPv6 world today. Operationally, in the real world, ISP customers who need to multihome request their own provider independent IPv6 address space, and use that, even if it adds an entry to the global routing table.
- Route table growth may be a critical issue facing the Internet in the long term, but for now, the community has "dropped back into punt formation," and we're doing what needs to be done (at least for now) to get IPv6 deployed in a robust way (e.g., with multihoming). The good news is that the IPv6 table is still small (so we still have time to solve the IPv6 routing table growth issue); the bad news is that the IPv6 table is still small (which means many people still haven't deployed IPv6!)

IPv6 Route Table Growth



4000

Source: http://bgp.potaroo.net/v6/as6447/

IPv6 Is Also Riddled with Myths and Misconceptions: For Example, Maybe You've Heard That IPv6 Is "More Secure Than IPv4" Because "IPSec Is Mandatory In IPv6"?

Tip: Support for IPSEC May Be <u>Mandatory</u>, But That Doesn't Mean It Is Getting <u>Used</u>.

A Little IPsec Backfill…

- IPsec is not new with IPv6; in fact, IPsec dates to the early 1990's. What's different when it comes to IPv6 is that support for IPsec was made "mandatory" for IPv6 (see for example "Security Architecture for IP," RFC4301, December 2005 at section 10, and "IPv6 Node Requirements," RFC4294, April 2006 at section 8.)
- **If actually used**, IPsec has the potential to provide:
  -- authentication
  -- confidentiality
  -- integrity, and
  -- replay protection
- All great and wonderful security objectives -- **IF** IPsec gets used. **Unfortunately, as we'll show you, what was supposed to be a cornerstone of the Internet's security architecture has proven in fact to be widely non-used.**

How Might IPsec Be Used?

- IPsec can be used to authenticate (using AH (the Authentication Header), RFC4302), or it can encrypt and (optionally) authenticate (using ESP (the Encapsulating Security Protocol), RFC4303)
- IPsec can be deployed in three architectures:
  -- gateway to gateway (e.g., securing a network segment from one router to another)
  -- node to node (e.g., securing a connection end-to-end, from one host to another)
  -- node to gateway (e.g., using IPsec to secure a VPN connecting from a mobile device to a VPN concentrator)
- IPsec has two main encrypting modes:
  -- tunnel mode (encrypting both payload and headers)
  -- transport mode (encrypting just the payload)
- IPsec also supports a variety of encryption algorithms (including "null" and md5 (yech)), and a variety of key exchange mechanisms
- All these alternatives obviously provide tremendous flexibility, but that flexibility also brings along a lot of potential complexity.

But, IPsec ISN'T Getting Used "Everywhere"

- IPv6 can be brought up without IPSec getting enabled, and in fact this is routinely the case -- see an example on the next slide.
- More broadly, if people are doing cryptographically secured protocols of *any* sort, they inevitably run into problems -- crypto stuff just tends to be inherently tricky and hard to learn to use. For example, how many of you routinely use PGP or GPG to cryptographically sign or encrypt your email, eh? How many of you are doing DNSSEC to cryptographically protect the integrity of your DNS traffic? Not very many, I'd wager...
- Now think about how often you see people moaning about problems they're having getting IPSec to work with IPv6 -- do you EVER see that on the mailing lists or discussion groups you're on? No? I didn't think you did. Why? That's because basically NO ONE is doing IPSec with IPv6.

Some IPv6 Traffic Statistics From A Mac OS X Host: No ipsec6 Traffic

```
# netstat -s -finet6
[snip]

ip6:
      124188 total packets received
      [snip]
      84577 packets sent from this host
      [snip]

      ipsec6:
      0 inbound packets processed successfully
      0 inbound packets violated process security policy
      [snip]
      0 outbound packets processed successfully
      0 outbound packets violated process security policy
      [snip]
```

IPsec (Even on IPv4!) Isn't Getting Much Use

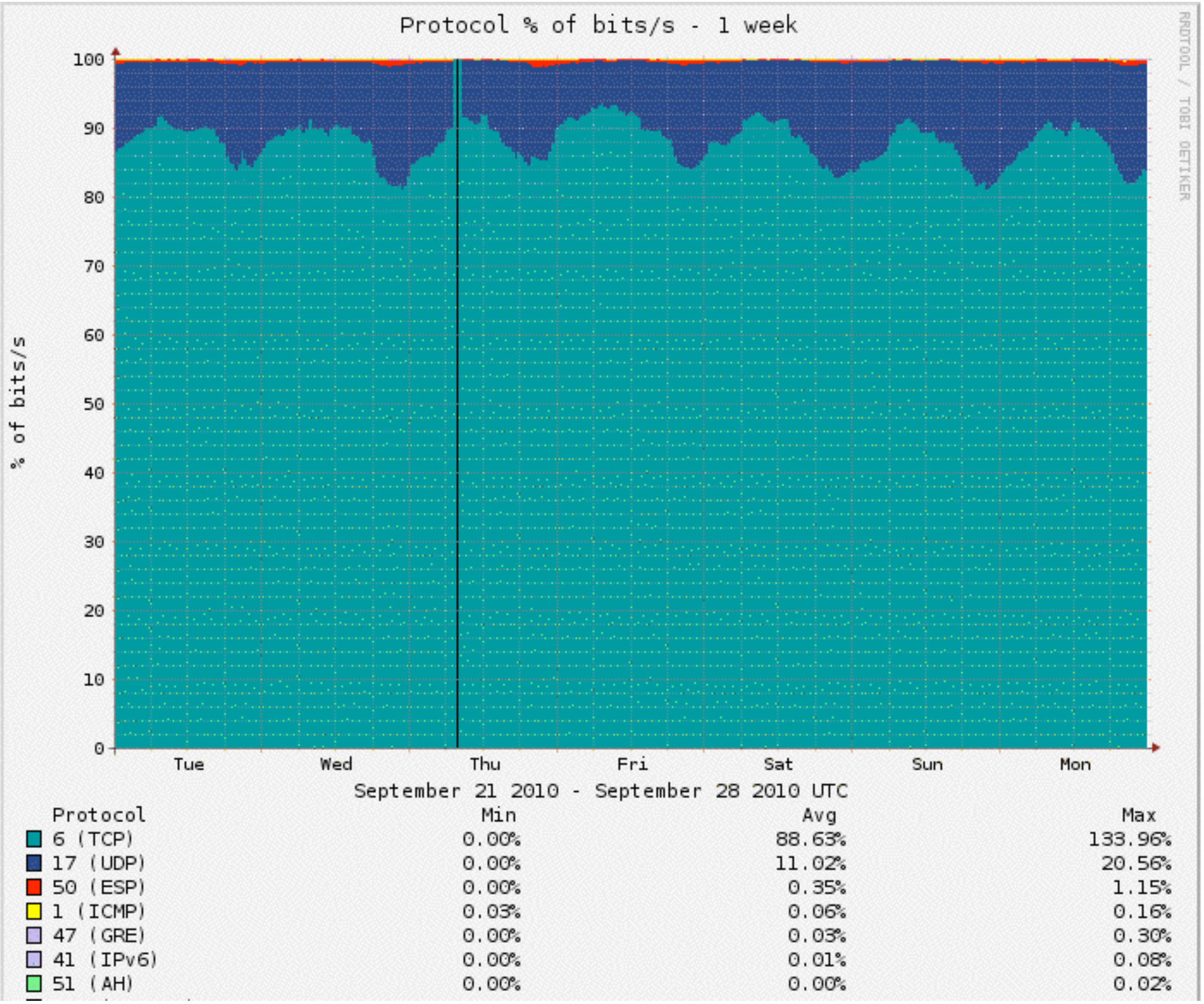
- Raw IPsec traffic (AH+ESP, protocols 50 & 51) isn't seen much on the commercial IPv4 Internet.

- For example, a year or so ago, Jose Nazario of Arbor Networks estimated IPsec traffic at 0.9% of octets (statistic courtesy the ATLAS project).

- CAIDA (thanks kc!) also has passive network monitoring data available; see http://www.caida.org/data/passive/monitors/equinix-chicago.xml

  You can see the protocol distribution from a couple of CAIDA's monitors for one recent day on the next slide. IPsec traffic is basically too small to even be seen for the most part.

Protocol Distribution From One of CAIDA's Passive Monitors



Protocol	Min	Avg	Max
🟦 6 (TCP)	0.00%	88.63%	133.96%
🟦 17 (UDP)	0.00%	11.02%	20.56%
🟥 50 (ESP)	0.00%	0.35%	1.15%
🟨 1 (ICMP)	0.03%	0.06%	0.16%
🟪 47 (GRE)	0.00%	0.03%	0.30%
🟪 41 (IPv6)	0.00%	0.01%	0.08%
🟩 51 (AH)	0.00%	0.00%	0.02%

[Not much IPv4 IPsec traffic, eh? It's the red stuff...]

Why *Aren't* We Seeing More IPSec Traffic?

- Sites may not be deploying IPsec because IPsec (like many crypto-based security solutions) has developed a reputation as:

  -- not completely baked/still too-much under development
  -- too complex
  -- hard to deploy at significant scale
  -- less than perfectly interoperable
  -- likely to cause firewall issues
  -- potentially something of a performance hit (crypto overhead issues)
  -- congestion insensitive (UDP encapsulated IPsec traffic)
  -- something which should be handled as an end-to-end matter by interested system admins (from a network engineer perspective)
  -- something to be handled at the transport layer router-to-router (from an overworked system administrator's perspective)
  -- duplicative of protection provided at the application layer (e.g., encryption is already being done using ssh or ssl)
  -- complicating maintaining/debugging the network, etc., etc., etc.

- Regardless of whether those perceptions are correct (some may be, some may **not** be), IPsec adoption hasn't happened much to date.

Non-IPSSEC IPv6 Tunneled Traffic

- Recall that I'd mentioned that Hurricane Electric has deployed tens of thousands of IPv6 tunnels to diverse locations all across the world.

- Tunneled traffic, even if not encrypted, generally has poor visibility for network traffic analysis purposes (most network traffic analysis tools do not automatically rip open tunnels to provide access to underlying protocols).
  [But see http://www.hiddenlab.net/teredont.html]

- So, even if people are NOT using IPSec, they may still be using tunnels or other technology that increases the opacity of network.

IPv6 Traffic Monitoring in General

- Ideally, for production IPv6 traffic, one would want **full IPv6 SNMP support** and **full IPv6 Netflow (V9) support**.

- Regretably, native IPv6 SNMP support and IPv6 V9 Netflow support remains elusive on many devices and networks. That's increasingly unfortunate for IPv6 as a production protocol that is, or should be, on par with IPv4.

- One way to improve IPv6 visibility on ISP backbones would be to deploy at least a limited number of dedicated, IPv6-aware, passive measurement appliances. For instance, some network measurement researchers have been pleased with the IPv6 support available from InMon Corporation's Traffic Sentinel product (e.g., see http://www.inmon.com/products/trafficsentinel.php).

Another Misconception:
IPv6 Address Space Is So Immense,
The Bad Guys Will Never Be Able To Find Me!
[Take *That*, You Dirty Abusive Scanners!]

(Well, the bad guys may not be able to successfully brute force scan for hosts in IPv6 space, but they can still find hosts to attack once they have a toehold on your network...)

Pre-Attack Network Reconnaissance

- It is common for miscreants to remotely scan IPv4 network addresses in an effort to identify active addresses, operating systems in use, open ports, etc., intelligence which may help them plan an attack against you. An increasingly common (if unfortunate) response to that threat has been to insert a firewall between the Internet and local users, thereby deflecting some scans and probes, albeit at the cost of a loss of transparency.
- Because IPv6-connected sites typically have a far larger number of addresses than IPv4-only sites, and end-to-end connectivity was another key objective of IPv6's architecture, some have suggested that it might be harder for attackers to do exhaustive scans of IPv6 sites simply because of the vastly larger number of addresses involved. That's true, as far as it goes, but that's not the whole story. If you haven't seen RFC 5157 ("IPv6 Implications for Network Scanning," March 2008), I'd urge you to look it over.
- If a miscreant can get a toehold on a local IPv6 connected host, they can obviously easily discover all the other hosts on that same subnet… :-;

Probing Magic Multicast Addresses

```
% ping6 -I eth0 ff02::1
% ping6 -I eth0 ff02::2
```

[See http://www.iana.org/assignments/ipv6-multicast-addresses/]

Magic Multicast Address Output

```
% ping6 -I en0 ff02::1
PING6(56=40+8+8 bytes) fe80::203:93ff:fecf:b638 --> ff02::1
16 bytes from fe80::203:93ff:fecf:b638, icmp_seq=0 hlim=64 time=0.248 ms
16 bytes from fe80::20f:1fff:fe98:e548, icmp_seq=0 hlim=64 time=0.761 ms(DUP!)
16 bytes from fe80::213:faff:fe01:a6a4, icmp_seq=0 hlim=64 time=0.898 ms(DUP!)
16 bytes from fe80::2e0:29ff:fe3c:9a3a, icmp_seq=0 hlim=64 time=0.951 ms(DUP!)
16 bytes from fe80::2e0:dbff:fe10:75c, icmp_seq=0 hlim=64 time=1.254 ms(DUP!)
16 bytes from fe80::2d0:1ff:fe95:e000, icmp_seq=0 hlim=64 time=1.376 ms(DUP!)
16 bytes from fe80::2e0:dbff:fe10:7c6, icmp_seq=0 hlim=64 time=1.832 ms(DUP!)
[etc]

% ping6 -I en0 ff02::2
PING6(56=40+8+8 bytes) fe80::203:93ff:fecf:b638 --> ff02::2
16 bytes from fe80::2d0:1ff:fe95:e000, icmp_seq=0 hlim=64 time=5.206 ms
[etc]
```

[Bottom line: don't assume that just because IPv6 has immense space that you'll be able to successfully "hide"]

Conclusion

In Summary, There ARE Obviously A LOT of IPv6-Related Technical Challenges…

- I'm not going to try to re-summarize them now – it's almost time for lunch, and I wanted to leave at least a few minutes for discussion.

- Let me say that I DO think that you should:

  -- pay attention to the fact that we're getting **really close** to running out of IPv4 addresses

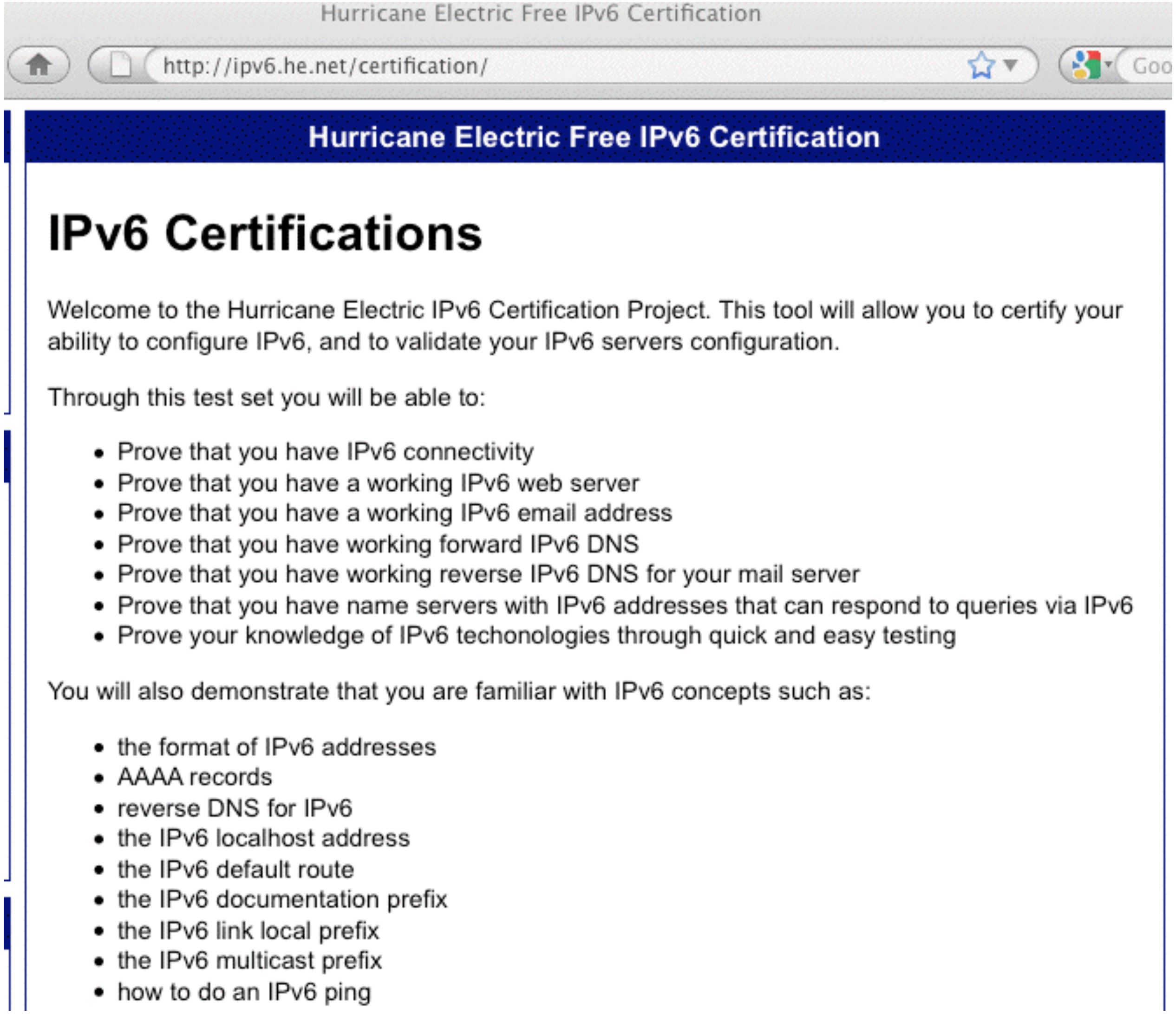
  -- start working to get IPv6 deployed for your own sites (recognizing that there are some material obstacles that folks may still need to work to overcome)

  -- learn more about IPv6.

One Way to Learn More About IPv6: Books

- A nice online reference: 6Net IPv6 Cookbook, www.6net.org/publications/deliverables/D3.1.2v2.pdf
- Traditional printed books about IPv6 are also available. If you go to Amazon's book section and search for IPv6, for example, you get 842 hits. That's a bit better than it was in the old days. :-)
- When considering which of those books might work for you, recognize that some are written for specific audiences (like programmers), and those sort of books may not meet your particular needs (unless you're a coder and you're trying to come up to speed for IPv6).
- Also recognize that IPv6 is rapidly evolving, so beware of any books that haven't been recently updated.
- Some may also find it helpful to have a concrete goal

Hurricane Electric's IPv6 Certification Program


Hurricane Electric Free IPv6 Certification
http://ipv6.he.net/certification/
Goo

Hurricane Electric Free IPv6 Certification

IPv6 Certifications

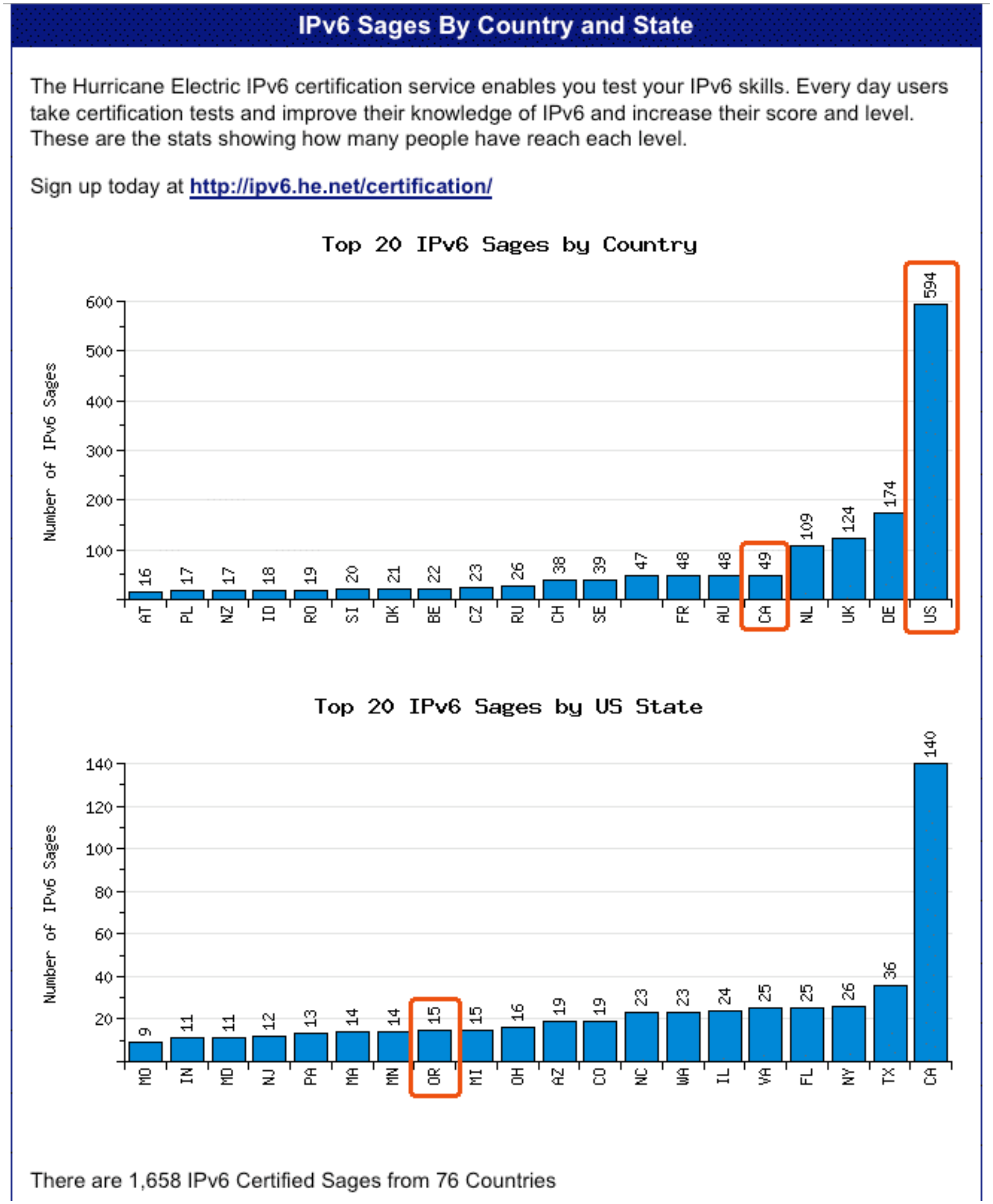
Welcome to the Hurricane Electric IPv6 Certification Project. This tool will allow you to certify your ability to configure IPv6, and to validate your IPv6 servers configuration.

Through this test set you will be able to:

- Prove that you have IPv6 connectivity
- Prove that you have a working IPv6 web server
- Prove that you have a working IPv6 email address
- Prove that you have working forward IPv6 DNS
- Prove that you have working reverse IPv6 DNS for your mail server
- Prove that you have name servers with IPv6 addresses that can respond to queries via IPv6
- Prove your knowledge of IPv6 techonologies through quick and easy testing

You will also demonstrate that you are familiar with IPv6 concepts such as:

- the format of IPv6 addresses
- AAAA records
- reverse DNS for IPv6
- the IPv6 localhost address
- the IPv6 default route
- the IPv6 documentation prefix
- the IPv6 link local prefix
- the IPv6 multicast prefix
- how to do an IPv6 ping



IPv6 Sages By Country and State

The Hurricane Electric IPv6 certification service enables you test your IPv6 skills. Every day users take certification tests and improve their knowledge of IPv6 and increase their score and level. These are the stats showing how many people have reach each level.

Sign up today at http://ipv6.he.net/certification/

There are 1,658 IPv6 Certified Sages from 76 Countries

Note!
If *you* get certified as a Hurricane Electric "IPv6 Sage" you'll have some fun, learn some new stuff, and maybe you can even help bring up your country's (or your state's) "sage ranking" at

tinyurl.com/ipv6-sages

Thanks For the Chance To Talk!

- Are there any questions?

[BTW, if you find yourself wanting a basic primer on some additional IPv6 topics, feel free to see the material that follows this slide. I didn't want to bore folks who might already be up to speed on these fundamentals.]

A Quick Intro To
Some Basic IPv6 Concepts
For Cybercrime Investigators

IPv6 Addresses and Prefixes

Starting With What We Know: IPv4

- IPv4 addresses are 32 bits long
- 2^32=4,294,967,296
- Normally represented in “dotted decimal” format:
  -- four 8 bit octets (0 to 255 decimal)
  -- each octet is separated from the next by a dot
  -- leading zeroes in each octet may be omitted
- Examples:
  -- 127.0.0.1
  -- 128.223.142.89
  -- 64.170.98.32

Something A Little Different: IPv6

- IPv4 addresses are 128 bits long
- 2^128=340,282,366,920,938,463,463,374,607,431,768,211,456 (e.g., 3.4x10^38 addresses)
- Normally represented in "colon separated" format:
  -- eight sets of four hex digits (0000 to FFFF hex)
  -- chunks are separated with colons (:)
  -- leading zeroes in each chunk may be omitted
  -- for convenience, :: (two successive colons) may replace one or more all-zero chunks, but only once in any addr
- Examples:
  2001:48a8:6880:0095:0000:0000:0000:0021
  ::1
  2001:468:d01:d6::80df:d617
  fe80::203:93ff:fecf:b638

Quick Quiz (With Answers)

- Structurally/superficially valid or invalid?
  If invalid, why?

  A) 2001:468:0d01:003c:0000:0000:80df:3c15 (valid)
  B) 2001:468:0d01:003c::80df:3c15 (ditto, compressed)
  C) 2001:468:d01:3c::80df:3c15 (ditto, compressed more)
  D) 2001:760:2e01:1::dead:beef (valid – isn't hex fun?)
  E) 2001:480:10:1048:a00:20ff:fe9a:58c1:80 (no, 9 "chunks")
  F) 2001:500::4:13::80 (no, more than one double colon)
  G) 2001:13G7:7002:4000::10 (no, G isn't a valid hex digit)
  H) 2607:f278:4101:11:209:5bff:fe8f:6609 (valid -- see http://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xml)
  I) fe80::209:3dff:fe13:fcf7 (valid, link local)
  J) :: (valid, the IPv6 "unspecified address")

Prefixes:
Starting With Something We Know:
IPv4 Prefixes

- We originally had class A, class B and class C addresses:

Name	CIDR equiv	Number	Addresses Per Block
Class A	/8	128	16,777,216
Class B	/16	16,384	65,536
Class C	/24	2,097,152	256

- Like goldilocks, some of those were too large, and some of those were too small. We needed something a little more flexible, and that's what we got from CIDR, or Classless Inter-Domain Routing.

Common IPv4 CIDR Prefix Lengths

/8	==>	16,777,216 addresses
/9	==>	8,388,608
/10	==>	4,194,304
/11	==>	2,097,152
/12	==>	1,048,576
/13	==>	524,288
/14	==>	262,144
/15	==>	131,072
/16	==>	65,536
/17	==>	32,768
/18	==>	16,384
/19	==>	8,192
/20	==>	4,096
/21	==>	2,048
/22	==>	1,024
/23	==>	512
/24	==>	256
/25	==>	128
/26	==>	64
/27	==>	32
/28	==>	16
/29	==>	8
/30	==>	4
/31	==>	2
/32	==>	1

- It's common for IPv4 subnets to be /24's (or maybe /23's or /25's)

IPv6 Uses Prefixes, Too

Prefix	Addresses
/32	2^(128-32) = 2^96 = 79,228,162,514,264,337,593,543,950,336 (e.g., 65,536 /48's or 4,294,967,296 /64's)
/48	2^(128-48) = 2^80 = 1,208,925,819,614,629,174,706,176 (256 /56's or 65,536 /64's)
/56	2^(128-56) = 2^72 = 4,722,366,482,869,645,213,696 (256 /64's)
/64	2^64 = 18,446,744,073,709,551,616

How IPv6 Allocations/Assignments/ Subneting Are Supposed to Work

- Local Internet Registries (LIRs) will get one (or more) IPv6 /32 from ARIN, RIPE, APNIC, etc.
- Large sites will get an IPv6 /48 from their LIR's /32
- Small sites needing only a few subnets over 5 years will get an IPv6 /56 from their LIR's /32
- If one and only one subnet is needed, that entity gets an IPv6 /64
- Hosts use one or more IPv6 /128s out of a /64
- ALL SUBNETS at most sites will normally be /64s (even if they have only a small handful of hosts). Do NOT try to get clever and do something exotic when it comes to subnetting.
- NOTE: All of the above (except the /64 per subnet rule) should be considered subject to change as a result of currently pending ARIN policies.
- [BTW, don't the /32, /48, /56 cut points feel a lot like the old IPv4 classful address days? They sure do to me…]

Types of IPv6 Addresses;
IPv6 Addresses and Systems

Types of IPv4 Addresses

- Most of you will be familiar with a number of different types of IPv4 addresses.

  For example:

  - Globally routable unicast addresses (e.g., regular IPv4 addresses)
  - Loopback address (127.0.0.1)
  - Private RFC1918 addresses (e.g., 10.0.0.0/8, etc.)
  - IPv4 multicast addresses

- There are similarly a variety of different types of IPv6 addresses.

Types of IPv6 Addresses

- **Global Unicast: 2000::/3**
- **Link Local Unicast: FE80::/1**
- **Loopback: ::1/128**
- 6to4: 2002::/16
- Teredo: 2001:0000::/32
- Unique Local Unicast: FC00::/7
- Multicast: FF00::/8
- IPv4-Mapped: ::ffff:128.223.214.23
- Deprecated: Site Local addrs and IPv4-Compatible addrs.
- For more on IPv6 addresses, see RFC4291.
- **For the most part, we care about IPv6 local unicast, link local unicast, and loopback addresses**

Address Type Discussion

- **Global Unicast addresses are globally unique, "real" IP addresses. These are the way you'll normally refer to most IPv6 hosts**
- **Link Local Unicast addresses are used for some purposes local to a particular link; outside the extent of that link, they don't get used. [If all you see is an FE80:: IPv6 addr, you <u>don't</u> have v6 connectivity]**
- **Loopback addresses -- just like 127.0.0.1 in IPv4 space, the IPv6 loopback address (::1/128) is an internal virtual address that the server can use to refer to itself.**
- **6to4 and Teredo** addresses are special IPv4-to-IPv6 transition mode technologies. We'll talk about them later.
- **Unique Local Unicast** addresses (RFC4193) are the IPv6 equivalent of RFC1918 IPv4 addresses. Don't use them.
- **Multicast** addresses are just like multicast in IPv4, except that in IPv6 they're used extensively on the LAN, but IPv6 multicast traffic rarely appear on the wide area Internet, unlike IPv4 where the exact opposite is largely true.
- **IPv4-Mapped** addresses allow hosts that only bind IPv6 sockets to also accept IPv4 addresses.

Expect Multiple Addresses on IPv6 Hosts

- When you look at the interfaces on an IPv6-enabled system, it will be routine for it to have multiple addresses/interface.
- Sometimes this will just be a single globally unique unicast address, plus a link local address, plus some other bits and pieces, other times you may see multiple globally unique unicast addresses.
- Do not let this shake you up.

- Speaking of multiple addresses, most hosts will have both IPv6 AND IPv4 addresses on some interfaces. This is known as being "dual stacked" and is perfectly normal and acceptable.

Looking at Addresses on Interfaces on Linux

```
% ifconfig -a
eth0     Link encap:Ethernet  HWaddr 00:09:3D:13:FC:F7
         inet addr:128.223.142.32  Bcast:128.223.143.255  Mask:255.255.254.0
         inet6 addr: 2001:468:d01:8e:209:3dff:fe13:fcf7/64 Scope:Global
         inet6 addr: fe80::209:3dff:fe13:fcf7/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:1187468996 errors:0 dropped:1805 overruns:0 frame:0
         TX packets:1338373204 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:232065679216 (216.1 GiB)  TX bytes:915094219311 (852.2 GiB)
         Interrupt:185
[snip]
lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         inet6 addr: ::1/128 Scope:Host
         UP LOOPBACK RUNNING  MTU:16436  Metric:1
         RX packets:8143461 errors:0 dropped:0 overruns:0 frame:0
         TX packets:8143461 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:4295055907 (4.0 GiB)  TX bytes:4295055907 (4.0 GiB)
```

How Addresses Get On Interfaces: IPv4

- Starting with something you know, IPv4, most **end-user workstations** get their IP addresses automatically assigned via DHCP.
- Besides IPv4 addrs, RFC2132 describes additional useful bits that a host can get from a DHCP server, including:
  -- subnet mask
  -- router address(es)
  -- time server(s)
  -- domain name server(s)
  -- the host name
  -- the domain name
  -- MTU information
  -- broadcast address
  -- plus an amazing amount of other bits and pieces (check it out if you don't believe me!)

How Addresses Get On Interfaces: IPv6

- In IPv6 world, at least some user workstations will get IPv6 addresses from state-less address auto configuration, or SLAAC. SLAAC runs on a router (not on a separate DHCP server), and as you might expect from the name, the IPv6 addresses that one gets via SLAAC are not maintained in a table anywhere (no "state" gets created when an IPv6 address is assigned via SLAAC).

- So how does the router know that it won't accidentally give you the same address as someone else (e.g., assign a duplicate address) if it doesn't keep track of who it has given an address to? Answer: it derives the address it gives you from something only you have, namely the MAC (hardware ethernet) address of your NIC

IPv6 Modified EUI-64 Format Identifier

- So, we need to take a 48 bit MAC address, and convert that into a 64 bit dynamically assigned address. What do we do?
  -- The left most 24 bits of the MAC form the left most 24 bits of the EUI-64 format identifier
  -- The right most 24 bits of the MAC form the right most 24 bits of the EUI-64 format identifier
  -- We cram the constant FFFE in the middle 16 bits
  -- We tweak bit 7 from the left from zero to one (this is the "universal/local" bit)

  The "front half" of the 128 bit address comes from the network (remember our rule that all subnets are /64s!)
- [BTW: http://mirrors.bieringer.de/www.deepspace6.net/projects/ipv6calc.html can be handy for teasing out information from IPv6 addresses]

"But Joe!"

- "Now the whole world will know my unique and unvarying hardware MAC address! Evil marketers will track and correlate my every move wherever I may connect my IPv6 device based on my MAC address! This is worse than web cookies!"

- True (especially the evil marketers bit).

- This concern spawned another type of IPv6 address, so-called RFC3041 Privacy Addresses. These addresses effectively use a random address for the low order 64 bits of the IPv6 address, instead of a value derived from the host's MAC address. Users will periodically change those random addresses from time to time.

"BUT JOE!!!!!"

- "That's NOT what I wanted either! If we give users random network addresses, how will we be able to track down abusers??? The same user may have one IPv6 address now, and another completely different IPv6 address later, and I don't see how we'd keep track of who's got what address when!!! What a pain!"

- One bit of potential happiness: NDPmon (think of this as "arpwatch for IPv6"). See ndpmon.sourceforge.net

- Another *potential* option: control network access just as you currently control network access for wireless networks. But is NAC even supported for IPv6? (Check out https://supportforums.cisco.com/message/3202862)

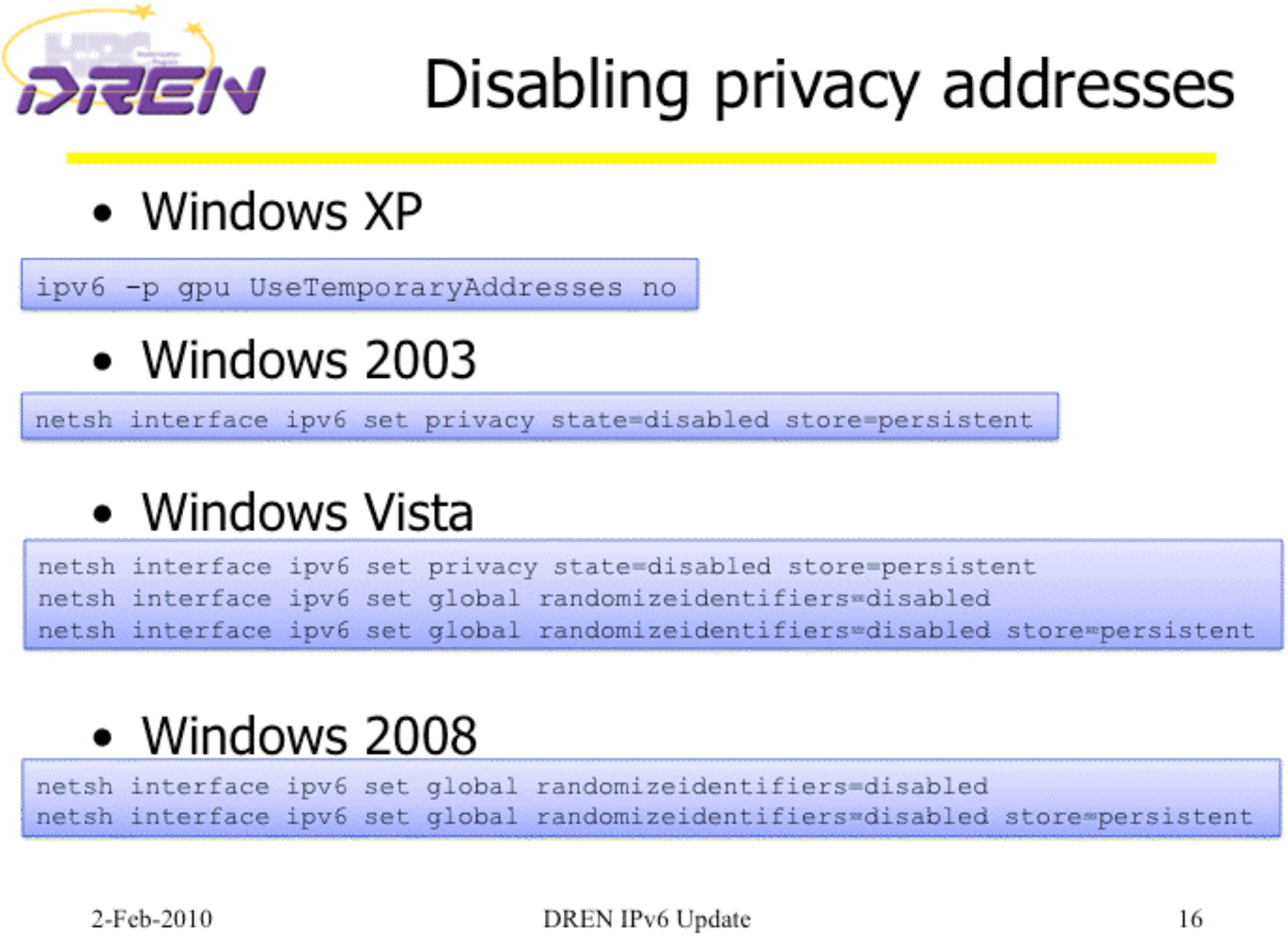
Selecting/Deselecting Privacy Addresses

- **Windows:** privacy addresses are **enabled** by default when IPv6 is enabled on Windows XP. To disable them, see the next slide.
- **Macs:** privacy addresses are **disabled** by default.
  To enable them:

  ```
  # sysctl net.inet6.ip6.use_tempaddr=1
  ```
- **Linux:** like Macs, privacy addresses are disabled by default. To enable them:

  ```
  # sysctl net.ip6.conf.all.use_tempaddr=2
  # sysctl net.ip6.conf.default.use_tempaddr=2
  ```
- Periodically recheck your assigned addresses if this is a big deal for you, and remember, this is NOT life-and-death privacy, this is just "something-to-make-life-hard(er)-for-intrusive-marketers"-grade privacy.

Disabling IPv6 Privacy Addresses



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
```

2-Feb-2010 DREN IPv6 Update 16

http://www.internet2.edu/presentations/jt2010feb/20100202-broersma.pdf

A Stateful Alternative to SLAAC: DHCPv6

- An alternative to SLAAC for workstations is DHCPv6, which feels much like DHCP for IPv4.
- One critical difference: while DHCPv6 is well supported by IPv6ified versions of Microsoft Windows, **at least some important vendors (read that as "Apple") do not support DHCPv6 at this time.**
- If you're using a Mac OS X box and you want or need to do DHCPv6, you will need to run a third party DHCPv6 client (Dibbler is a commonly suggested option, see http://klub.com.pl/dhcpv6/ , but note the comment there that "Due to work on my Ph.D, the Dibbler project is in the maintenance mode. Active development and non-critical bug fixing is on hold, until I finish my dissertation. Sorry.")

Manually Assigned Native IPv6 Addresses

- Just like in IPv4, you also have the option of manually assigning native IPv6 addresses for things like servers. Recipes for some common OS's:

  -- FreeBSD and Friends:
  http://www.cyberciti.biz/faq/freebsd-configure-ipv6-networking-static-ip-address/
  -- Redhat/CentOS:
  http://www.cyberciti.biz/faq/rhel-redhat-fedora-centos-ipv6-network-configuration/
  -- SuSE Linux:
  http://www.cyberciti.biz/faq/configuring-ipv6-in-sles10-opensuse-linux/
  -- Ubuntu Linux:
  http://www.cyberciti.biz/faq/ubuntu-ipv6-networking-configuration/
  -- Windows Server 2008/R2
  http://technet.microsoft.com/en-us/library/cc732106.aspx

  Mac user? Just set a static IP in System Preferences -> Network -> Configure -> Configure IPv6 -> Manually

Example: Enabling IPv6 On A Redhat Box Using A Static IPv6 IP

- In /etc/sysconfig/network

```
NETWORKING_IPV6=yes
```

- In /etc/sysconfig/network-scripts/ifcfg-eth0

```
IPV6INIT=yes
IPV6ADDR=your_servers_IPv6_address_here
IPV6_DEFAULTGW=your_servers_default_gw_here
```

- # service network restart

"What's My Subnet Length and Router Addr?"

- When statically configuring...
- Unless you're told otherwise, as a general rule of thumb, the subnet length will always be /64
- Unless you're told otherwise, again as a rule of thumb, the router address will always have the same first 64 bits as your host's static address, followed by ::1
- Don't forget to also define IPv6-aware name servers. If you don't have a suitable local alternative, Google's intentionally open name servers, 8.8.8.8 and 8.8.4.4 will usually work fine as long as you have IPv6 AND IPv4 connectivity to your host.
- Q. "In IPv4 I usually configure a broadcast address. What's my broadcast address for IPv6?"
  A. Broadcast isn't needed and doesn't exist in IPv6.

Automatic Tunneling Mechanisms

“Automatic” IPv6 Connectivity: 6to4

- In addition to native IPv6 connectivity or manually configured IPv6 tunnels, users may also connect via 6to4.
- 6to4 was meant as a temporary transition mechanism, to help people use IPv6 until they could get native IPv6 (or at least tunneled IPv6) deployed.
- Two issues with 6to4:
  -- it sends traffic to 192.88.99.1, part of the anycast block 192.88.99.0/24; you *will* use the “closest” 192.88.99.0/24 that’s out there, whether friend or foe :-;
  -- 6to4 cannot traverse a firewall (including those ubiquitous little blue Linksys boxes); Mac Airport Express and Airport Extreme boxes reportedly DO know how to handle 6to4 correctly, however.

Enabling 6to4 on a Mac

- *Remember:* 6to4 usually **won't work** behind a firewall.

- -- Apple Menu ==> System Preferences ==> Network ==> Show: Network Port Configuration
  -- If no 6 to 4 port already exists, click "New"
  -- Select 6 to 4 for the port from the pull down list of ports
  -- Enter "6 to 4" for the port's name
  -- Click OK
  -- Make sure "6 to 4" is checked as "On"
  -- Click "Apply Now"
  [the above details may vary on some versions of OS X]

- To disable 6to4, use System Preferences to set 6to4 to be "Off"

Teredo/Miredo

- Teredo is another automatic IPv6 tunneling protocol; this one differs from 6to4 in that it can successfully traverse network firewall boxes (unless the firewall blocks outgoing IPv4 traffic on 3544/UDP)

- Teredo ships with Microsoft Windows; if you're running Linux, you'll need to install Miredo for Teredo functionality.

- Miredo is available from http://www.remlab.net/miredo/

Enabling Teredo on a Windows XP SP2 PC

To set up IPv6 and Teredo on a Windows XP SP2 system, do:

Start ==> Accessories ==> Command Prompt

```
netsh interface ipv6 install
netsh interface ipv6 set teredo client
```

To disable it:

```
netsh interface ipv6 set teredo disabled
netsh interface ipv6 uninstall
```

Key Take Away: Even If A Site Formally "Doesn't Do" IPv6, Users May Still Be Using It…

- Some sites which rely heavily on firewalls and perimeter security may decide to forego or postpone deployment of native IPv6. Having made the decision to do so, folks may emit a big relieved sigh, believing that by "sitting this dance out," they will have foreclosed any possibility of user access to IPv6-only resources.
- Unless that policy is **very** carefully enforced on a technical basis, you may be in for a surprise or two because users may be able to easily work their way around your non-implementation or active filters.
- This is particularly important if you're relying primarily on perimeter filtering to control either the **infiltration** of malware, or the **exfiltration** of site-sensitive information.

6to4 & Teredo May Rely on "Remote Resources"

- In addition to things like 6to4 and Teredo traffic posing surprises for things like border filtering and traffic monitoring, tunneled traffic may also rely on donated **remote resources**.

- One could imagine an IPv6 transition site run by a cyber criminal, "kindly" offering free IPv6 gateway services in an effort to attract customer's traffic for surreptitious MITM-ish monitoring. [I don't think any of the current sites are malevolent, but in the future, who knows?]

- Services such as 6to4 and Teredo which do not require any sort of registration or authentication may also end up being **abused** by bad guys just as **open SMTP relays** once were.

Magic Addresses

- 6to4 uses 192.88.99.1 as a magic address, anycast via the magic prefix 192.88.99.0/24 (see RFC3068 at 2.3 and 2.4)
- Do you know where <u>your</u> 192.88.99.1 traffic is going? (simple test: traceroute to 192.88.99.1 from a machine at your home site) [Maybe you even want to *routinely* monitor the path to 192.88.99.1?]
- When I looked at some examples from public traceroute servers, (examples which I'll omit here), I've seen:
  -- large academic sites whose customers may end up using anycast 6to4 relays located clear across the country,
  -- government mission networks whose customers may rely on 6to4 anycast relays hosted on the campus of academic sites
  -- commercial providers whose customers may rely on anycast 6to4 relays hosted by some of their competitors.

Where's UO's Closest 192.88.99.1?

```
% traceroute 192.88.99.1
traceroute to 192.88.99.1 (192.88.99.1), 30 hops max, 46 byte packets
 1  vl-142.uonet2-gw.uoregon.edu (128.223.142.3)  0.341 ms  0.231 ms  0.667 ms
 2  3.xe-1-3-0.uonet10-gw.uoregon.edu (128.223.3.10)  0.180 ms  0.159 ms  0.152 ms
 3  vl-105.ge-2-0-0.core0-gw.pdx.oregon-gigapop.net (198.32.165.89)  2.805 ms  2.682 ms  2.679 ms
 4  vl-101.abilene-losa-gw.oregon-gigapop.net (198.32.165.66)  24.511 ms  24.567 ms  24.548 ms
 5  xe-0-1-0.0.rtr.hous.net.internet2.edu (64.57.28.97)  56.585 ms  56.555 ms  56.525 ms
 6  xe-2-3-0.0.rtr.atla.net.internet2.edu (64.57.28.113)  130.461 ms  79.937 ms  79.949 ms
 7  xe-0-2-0.110.rtr.ll.indiana.gigapop.net (149.165.254.20)  95.113 ms  95.063 ms  95.068 ms
 8  xe-0-0-0.1.rtr.ictc.indiana.gigapop.net (149.165.254.25)  115.358 ms  95.126 ms  95.102 ms
 9  rtr3.ul.indiana.gigapop.net (149.165.255.129)  95.359 ms *  95.302 ms
```

Teredo Also Relies on Magic Remote Resources

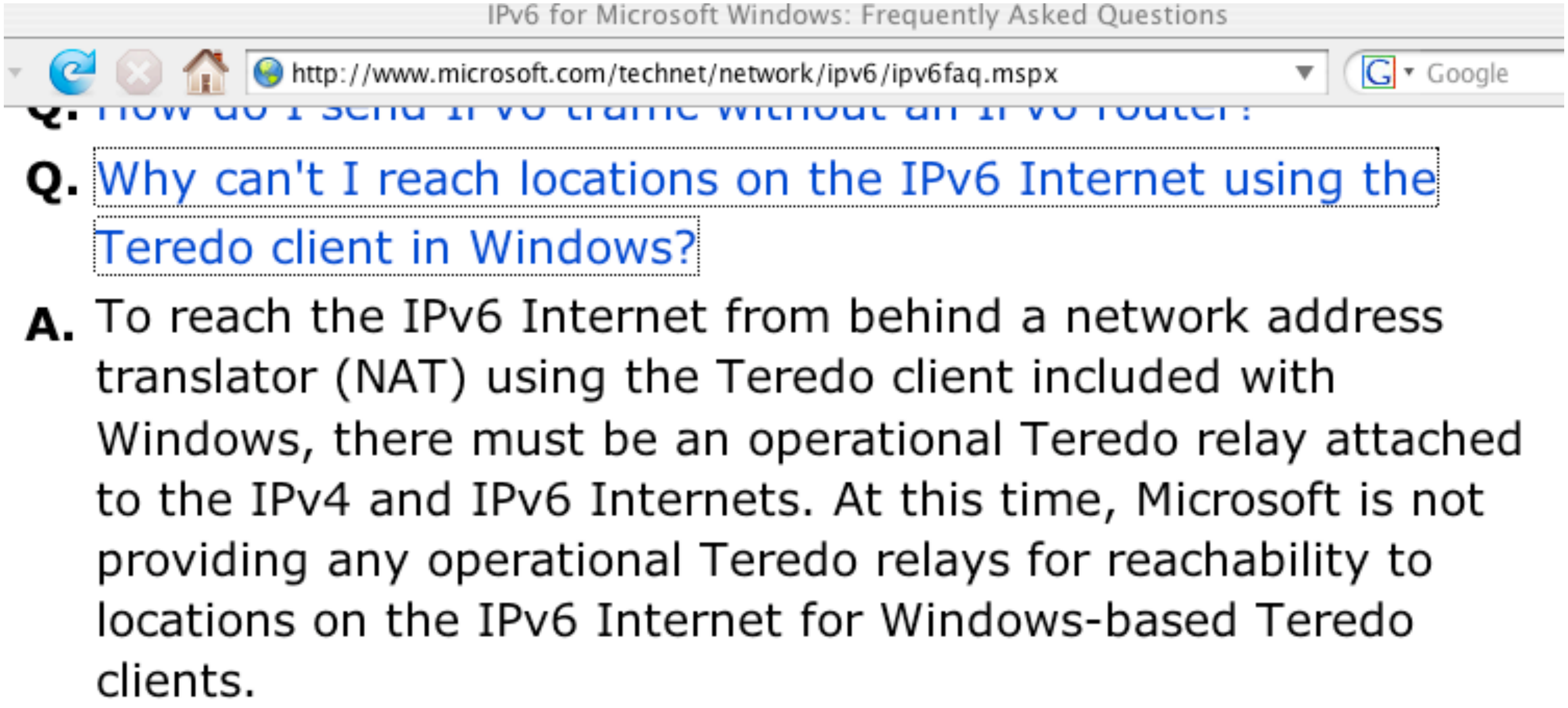
- Teredo relies on both Teredo servers and Teredo relays.

- Do you know which ones *your* folks may be using?

  When it comes to Teredo servers, http://technet.microsoft.com/en-us/library/cc722030.aspx mentions the Teredo ***server*** teredo.ipv6.microsoft.com – many of you are likely using that Teredo server by default.

- But what about Teredo relays?

Teredo *Relays*, Where the Bandwidth Intensive "Heavy Lifting" Happens...


IPv6 for Microsoft Windows: Frequently Asked Questions
http://www.microsoft.com/technet/network/ipv6/ipv6faq.mspx
Google

Q. How do I send IPv6 traffic without an IPv6 router?

Q. Why can't I reach locations on the IPv6 Internet using the Teredo client in Windows?

A. To reach the IPv6 Internet from behind a network address translator (NAT) using the Teredo client included with Windows, there must be an operational Teredo relay attached to the IPv4 and IPv6 Internets. At this time, Microsoft is not providing any operational Teredo relays for reachability to locations on the IPv6 Internet for Windows-based Teredo clients.

Sites Which *Are* Advertising 2001:0::/32 For Windows (And Other) Teredo Clients

- RFC4380 at 2.6 specifies 2001:0::/32 for the Teredo relay service. At NANOG45, Martin Levy presented "IPv6 Traffic Levels on Hurricane Electric's Backbone," (see www.nanog.org/meetings/nanog45/ presentations/Tuesday/Levy_traffic_level_hurricane_N45.pdf):

  > "**[Teredo] traffic is all eastward across the Atlantic**
  > Flows toward teredo.bit.nl AS12859 via AMS-IX
  > 2001::/32 announce by other networks including
  > AS12637 Seeweb, AS1257 Tele2, etc." [emphasis added]

- If you telnet to one of the IPv6-aware routeviews.org nodes (such as route-views.linx.routeviews.org), you can see sites advertising 2001:0::/32 by using the command "show ipv6 bgp 2001:0::/32"
- When I last checked, I saw 2001:0::/32 from AS1101 (IP-EEND/ SURFNet), AS1257 (Tele2), AS6939 (Hurricane), AS12859 (Bit.NL), AS21155 (ProServe), and AS29432 (Tampere Region Exchange, FI).
- If you are globally advertising 2001:0::/32, but for some reason your ASN isn't listed here, I'd love to hear from you.

‘So Are You Telling Me That I Should Try To "Break" or "Disable" 6to4 and/or Teredo?’

- Encountering 6to4 or Teredo is like encountering extra-terrestrial intelligence. Squelch any immediate reptilian instinct to smash/kill/eat anything which is new/different/potentially threatening. :-)
- At the same time, let's avoid philosophically overanalyzing this. We should not let "the perfect" get in the way of the "adequate." While I **really** want to see native IPv6 deployed end-to-end, 6to4 or Teredo (at least as long as it works and isn't being abused), is better for many v4 users than nothing.
- **Thus, notwithstanding some of the issues mentioned on previous slides, please refrain from breaking 6to4 or Teredo.**
- **You should consider fielding a carefully monitored version of those services, accessible only by your local users, thereby soaking up the local demand for those services (and if you do see folks using 'em, nudge them toward native IPv6 instead)**

What About ISATAP?

- ISATAP is yet another IPv6 transition mechanism, this one defined in RFC5214 (seems like there are millions of IPv6 transition mechanisms, doesn't it?)
- ISATAP violates fundamental layering principles.
- ISATAP also relies on the presence of a magic isatap.<domain> domain name. You *are* making sure that no one else has registered that magic name at your site, right? Just as you're making sure no one (except your authorized http proxy admin) registers wpad.<domain>?
- Personally, I'd explicitly urge you NOT to deploy ISATAP at your site. Nonetheless, if you want to see an example of how one can set up ISATAP, see http://technet.microsoft.com/en-us/magazine/2008.03.cableguy.aspx

Historical Trivia: What Was the "6Bone?"

- The 6bone was an IPv6 testbed using configured tunnels.
- 6bone IPv6 addresses began with 3FFE (sadly, you may still see some people trying to use 3FFE addresses today, even though the 6bone officially was decommissioned in June 2006)
- For historical information about this experiment, see http://go6.net/ipv6-6bone/

IPv6 And Some Common Cybercrime Investigative Tools

We’ve Already Covered Some Tools…

- You’ve already seen some common tools in use, such as dig and whois

- Most of the other investigative tools that you’re used to are also routinely available, albeit sometimes with an added flag or slight change in name.

- For example…

IPv6 Ping

- Ping in IPv6 works basically the same as ping in IPv4 (although you may need to use the command name "ping6" instead of just "ping" or you may need to add a flag to tell ping to use IPv6 instead of IPv4).
- For example:

```
% ping6 www.nanog.org
PING www.nanog.org(s1.nanog.org) 56 data bytes
64 bytes from s1.nanog.org: icmp_seq=0 ttl=55 time=107 ms
64 bytes from s1.nanog.org: icmp_seq=1 ttl=55 time=106 ms
64 bytes from s1.nanog.org: icmp_seq=2 ttl=55 time=107 ms
64 bytes from s1.nanog.org: icmp_seq=3 ttl=55 time=107 ms
[etc]
```

Some Notes About IPv6 Ping

- If you need to do an IPv6 ping from a Windows system, open a CMD window and use the “ping6” command
- Need a web-accessible IPv6 ping’er? Try: www.subnetonline.com/pages/ipv6-network-tools.php
- You can also ping from Route-Views. To do so, telnet to route-views.oregon-ix.net , login as rviews (no password required) then use “ping ipv6”. For example:

```
% telnet route-views.oregon-ix.net
Username: rviews
route-views>ping ipv6 www.ietf.org
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:1890:1112:1::20, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 196/197/200 ms
```

IPv6 Traceroute

- Traceroute in IPv6 works basically the same as traceroute in IPv4 (although you may need to use the command name “traceroute6” instead of just “traceroute” or you may need to add a flag to tell traceroute to use IPv6 instead of IPv4)

```
% traceroute6 www.ietf.org
[snip]
1    vl-142-gw.uoregon.edu (2001:468:d01:8e::1)  2.337 ms  1.039 ms  0.941 ms
2    vl-3.uonet9-gw.uoregon.edu (2001:468:d01:3::9)  0.3 ms  0.261 ms  0.259 ms
3    2001:468:d00:3::1 (2001:468:d00:3::1)  2.717 ms  2.725 ms  2.71 ms
4    2001:468:ffff:54d::1 (2001:468:ffff:54d::1)  24.577 ms  24.595 ms  24.569
     ms
5    pao-ipv6.gblx.net (2001:504:d::37)  33.242 ms  169.607 ms  33.134 ms
6    2001:1890:1fff:308:192:205:34:77 (2001:1890:1fff:308:192:205:34:77)
     82.432 ms  82.434 ms  82.478 ms [etc]
```

Some Notes About IPv6 Traceroutes

- Need to do an IPv6 traceroute from a Windows system? Open a CMD window and use the “tracert6” command
- Need a web-accessible IPv6 traceroute server? One’s available at http://4or6.com/
- Want a traceroute augmented with routing information? telnet to route-views.oregon-ix.net , login as rviews (no password required) then use “traceroute ipv6 *hostname*”
- Be prepared for many hops in an IPv6 traceroute to lack rDNS
- IPv6 traceroute paths will often follow bizarre paths due to poor route filtering policies (or the use of IPv6 transition technologies or IPv6 tunnel provider services)
- IPv4 and IPv6 traceroute paths may routinely follow radically different paths; make no assumptions!

curl and Retrieving IPv6 Web Pages

- Get curl from http://curl.haax.se/download.html (see also the additional libraries at http://curl.haxx.se/docs/libs.html)

- To force retrieval of a web page via IPv6, add -6:

  ```
  % curl -6 ”http://www.example.com”
  ```

- To force retrieval of a web page via IPv4, add -4:

  ```
  % curl -4 ”http://www.example.com”
  ```

- Need to specify an IPv6 hex literal? Remember brackets and -g (http://curl.haxx.se/docs/knownbugs.html at 30)

  ```
  % curl -6 -g ”http://[2001:48a8:6880:95::21]”
  ```

tcpdump and IPv6

- Build and install tcpdump (4.1.1) and libpcap (1.1.1) from http://www.tcpdump.org/

- For IPv6, add the tcpdump ip6 option. For example:

```
# tcpdump -xvs0 ip6 and tcp port 80
```

  x --> print each packet (except link layer hdr) in hex
  v --> slightly verbose
  s0 --> capture the whole packet, not just 68 bytes
  ip6 --> ipv6 traffic only
  tcp port 80 --> only tcp traffic on port 80

- If IPSec is being used on IPv6 links, obviously traffic may be obfuscated (unless you know the IPSec key)

Sample tcpdump output (ipv6.google.com)

-
  ```
  tcpdump: listening on en0, link-type EN10MB (Ethernet), capture size 65535 bytes
  09:15:39.691986 canard.ipv6.uoregon.edu.59761 >
  pv-in-x68.1e100.net.http: P [tcp sum ok] 3681843577:3681844207(630) ack 2324549898 win
  65535 <nop,nop,timestamp 365589597 2712339760> (len 662, hlim 64)
          0x0000:  6000 0000 0296 0640 2001 0468 0d01 00d6  `......@...h....
          0x0010:  0000 0000 80df d617 2001 4860 b006 0000  ..........H`....
          0x0020:  0000 0000 0000 0068 e971 0050 db74 7979  .......h.q.P.tyy
          0x0030:  8a8d d10a 8018 ffff 7235 0000 0101 080a  ........r5......
          0x0040:  15ca 745d a1ab 0530 4745 5420 2f20 4854  ..t]...0GET./.HT
          0x0050:  5450 2f31 2e31 0d0a 486f 7374 3a20 6970  TP/1.1..Host:.ip
          0x0060:  7636 2e67 6f6f 676c 652e 636f 6d0d 0a55  v6.google.com..U
          0x0070:  7365 722d 4167 656e 743a 204d 6f7a 696c  ser-Agent:.Mozil
          [snip]
  09:15:39.755859 pv-in-x68.1e100.net.http > canard.ipv6.uoregon.edu.59761: . [tcp sum ok]
  1:1209(1208) ack 630 win 285 <nop,nop,timestamp 2712558929 365589597> (len 1240, hlim 54)
          0x0000:  6000 0000 04d8 0636 2001 4860 b006 0000  `......6..H`....
          0x0010:  0000 0000 0000 0068 2001 0468 0d01 00d6  .......h...h....
          0x0020:  0000 0000 80df d617 0050 e971 8a8d d10a  .........P.q....
          0x0030:  db74 7bef 8010 011d 05d1 0000 0101 080a  .t{............
          0x0040:  a1ae 5d51 15ca 745d 4854 5450 2f31 2e31  ..]Q..t]HTTP/1.1
          0x0050:  2032 3030 204f 4b0d 0a44 6174 653a 2053  .200.OK..Date:.S
          0x0060:  6174 2c20 3031 204d 6179 2032 3031 3020  at,.01.May.2010.
          0x0070:  3136 3a31 353a 3339 2047 4d54 0d0a 4578  16:15:39.GMT..Ex
          [etc]
  ```

Watch An Amsterdam Street Cam via IPv6

- http://www.terena.org/webcam/

  (you can use tcpdump to confirm that yes, it really IS transmitting via IPv6, assuming you have IPv6 connectivity, e.g.:

  ```
  # tcpdump -i en0 -xvs0 ip6
  ```

  will show IPV6 traffic from wowza.terena.org)

Additional ipv6 tcpdump Commands

- Want more? All IPv6 traffic verbosely:
  ```
  # tcpdump -xxvvves0 ip6
  ```
- Want less? Just IPv6 traffic to/from a specific host:
  ```
  # tcpdump -xvs0 ip6 host foo.example.com
  ```
  or
  ```
  # tcpdump -xvs0 ip6 host fe80::203:93ff:fecf:b6a2
  ```
- Multiple interfaces? You can specify a specific interface (use ifconfig -a *or* tcpdump -D to see what's available to pick).

  For example, want to see if any ping6's are getting through on en0?
  ```
  # tcpdump -i en0 icmp6
  ```
- IPv6 multicast traffic can be interesting:
  ```
  # tcpdump -i en0 ip6 multicast
  ```
- Capture/replay tcpdump files:
  ```
  # tcpdump -i en0 -w sample.dump ip6
  [after a while, ^C]
  # tcpdump -r sample.dump
  ```
- Need the contents of tunneled IPv6 traffic? Check out teredont: http://www.hiddenlab.net/teredont.html