Going Native

A backbone transition to extensive native IPv6 peering

By Mike Leber

Abstract

Hurricane Electric (AS6939) completed a core router and backbone circuit upgrade in early 2007, which enabled Hurricane to move it's existing IPv6 overlay network into the core and go dual stack IPv4 + IPv6 at 18 different exchange points in the US and Europe. This paper compares network measurements before and after as well as IPv6 to IPv4

History

Hurricane Electric has operated IPv6 network elements since 2000. Hurricane setup IPv6 peering with 6bone participants in 2000, and received 6bone and ARIN IPv6 address allocations in 2001. Hurricane's network had a mix of native and tunneled IPv6 BGP connections for customers and peers. This network was operated as an overlay network using dedicated IPv6 routers to avoid stability issues with buggy IPv6 router software and because IPv6 on Hurricane's core routers at that time was process switched, meaning that OC48 backbone circuits could not handle IPv6 at wirespeed (which was cause concern for both security (denial of service) and performance reasons). Native IPv6 peering was established at PAIX Palo Alto, MAE-WEST ATM, Equi6IX Ashburn, and NY6IX. Hurricane's IPv6 user base consists of transit customers, colocation customers, and tunnel broker users (Hurricane operates a free IPv6 tunnel broker).

In early 2007 Hurricane Electric replaced every core router in the network and replaced every OC48 backbone circuit in the US and Europe with 10 gigabit wavelengths. The new core router platform routes IPv6 at wirespeed, and enabled the IPv6 network to be moved in to the core. In the following weeks extensive native IPv6 peering was setup. This paper documents the some interesting network metrics of the resulting network.

Comparison By Path Length

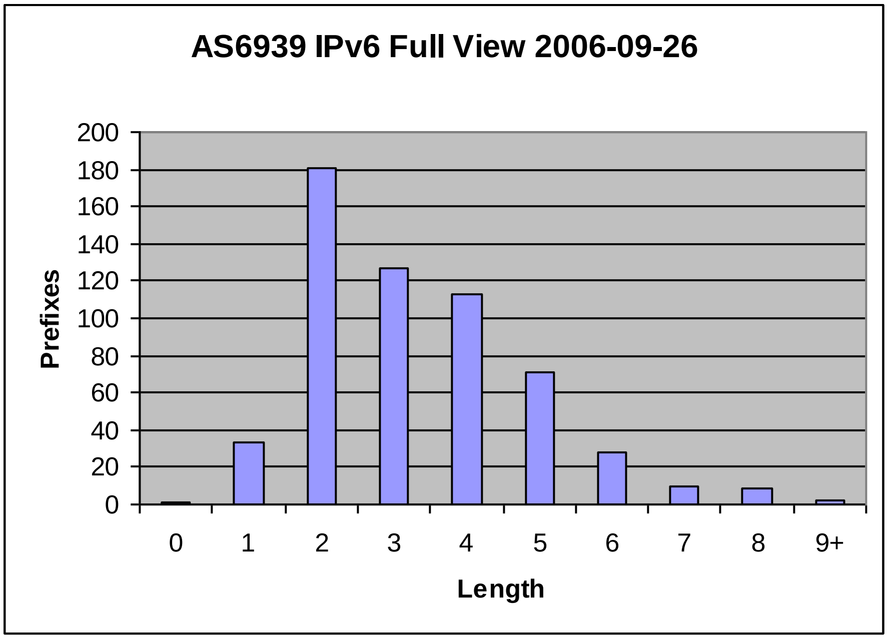
Path Length as measured by the number of ASes that show up in a BGP path indicates the number of autonomous networks that must be crossed to reach a destination prefix, and is a very rough indicator of the amount of network equipment and circuits that must be crossed to reach the destination. A shorter path length is usually better.

Public BGP data collectors such as University of Oregon Route Views Project (routeviews) and RIPE Routing Information Service (RIS) are an invaluable aid for the analysis of historical IPv4 and IPv6 BGP announcements.

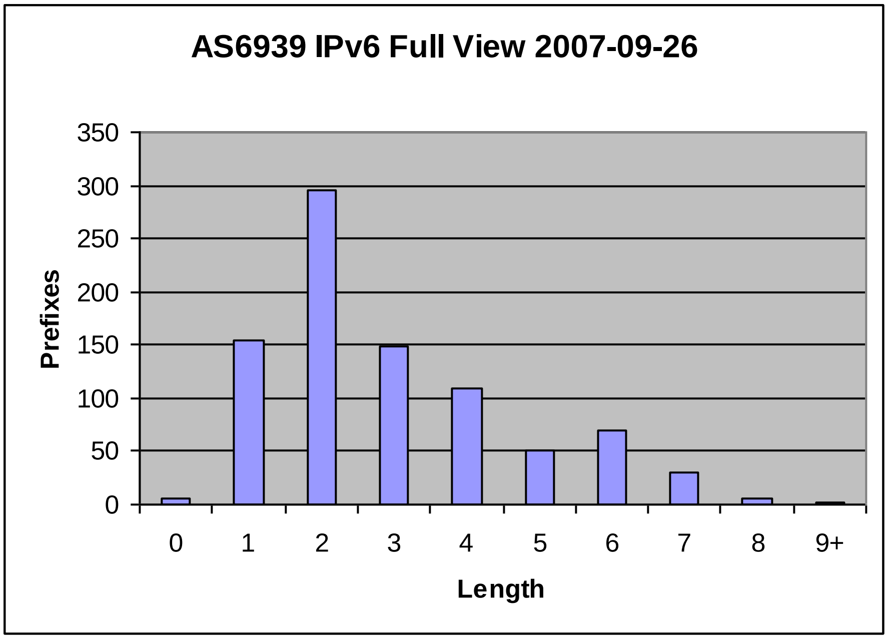
Using Marco d'Itri's zebra table parser and some perl scripts I was able to calculate the average path length of the IPv6 full view for AS6939 as well as generate a histogram of path lengths as of 2007-09-26 (the date this paper is being prepared) and 2006-09-26 (a year prior, and before the core router and backbone upgrade). A full view is the set of routes that are the best path for each prefix in the BGP routing table on a specific router.

Properly filtered, prefixes roughly correspond to a unique destination network per prefix. IPv6 prefixes /32 in length and shorter are allocated directly by the RIRs. Additionally, /48 prefixes may be allocated by a RIR or may be issued by a provider that received a /32. Customer /48s show up in the global IPv6 routing table due to all the various reasons networks multihome and run BGP. Hurricane Electric filters routes heard from peers and customers using Gert Döring’s IPv6 BGP filter recommendations (section 5.1).

BEFORE (2006-09-26)
Length: Number of Prefixes
0: 1
1: 33
2: 181
3: 127
4: 113
5: 71
6: 28
7: 10
8: 9
9+: 2
Total Prefixes 575
Average Path Length 3.33



AFTER (2007-09-26)
Length: Number of Prefixes
0: 6
1: 155
2: 295
3: 149
4: 109
5: 50
6: 69
7: 30
8: 6
9+: 1
Total Prefixes 870
Average Path Length 2.94



Note that the shape of the histogram has shifted towards the left, in this case zero, which visually gives the impression that the average path length has become shorter. Calculating the average path length reveals this to be true.

Path Length

BEFORE (2006-09-26)	3.33
AFTER (2007-09-26)	2.94
Decrease in average path length:	22 percent

Due to nascent deployment, many IPv6 prefixes are not thoroughly announced. This is a concern when providing IPv6 transit as a production service. So it is worth noting that the total number of IPv6 prefixes seen by AS6939 has also increased significantly after setting up extensive native IPv6 BGP sessions. Note that this total is after applying the recommended IPV6 BGP prefix filters.

Prefixes

BEFORE (2006-09-26)	575
AFTER (2007-09-26)	870
Increase in number of prefixes seen:	51 percent

Comparison By Latency

To truly measure the performance of IPv6 as a production service, rather than comparing it to itself at some other point in time, it should be compared to IPv4, since IPv4 is currently dominant.

Accordingly it would be convenient to have a set of servers that ran both IPv4 and IPv6, so that the relative performance to them over both IPv4 and IPv6 could be measured.

In an imaginary world all such servers would be connected to networks that had identical IPv4 and IPv6 deployment throughout the intervening network infrastructure, and all the intervening networks would have identical IPv6 and IPv4 peering/customer/provider relationships. In real life, this is not the case. IPv4 and IPv6 performance differ not just due to deployment issues, they also differ due to peering and transit relationship differences. However, unlike the common perception of IPv6 being uniformly slower than IPv4, sometimes IPv6 is *faster* than IPv4, as our data shows. This is surprising because while it is easy to inadvertently increase latency due to tunnels or inadequate IPv6 deployment, one would imagine economic incentive to reduce physical cable plant size or equipment would normally ensure a mature IPv4 network be near optimal from a cost/performance basis.

Identifying servers that run both IPv4 and IPv6 in other networks posed an interesting challenge, since we wanted relatively good coverage of the IPv6 prefixes in the global IPv6 routing table. We solved this problem by noting that all IPv6 prefixes should have working reverse DNS servers, and properly configured IPv6 reverse DNS nameservers ought to have both working IPv4 and IPv6 addresses. Through the use of pings and UDP DNS requests we measured the IPv4 and IPv6 latency to IPv6 reverse DNS servers. There are quite a few caveats about this technique, such as the IPv4 DNS address might be anycasted, however since this would likely skew the results towards IPv4 because this pratice is more common with production IPv4 services under load and IPv6 if separate would be likely limited to an early test server, it would make our results conservative with regards IPv6.

The tests were performed on 2007-09-04 from a server on Hurricane Electric's IPv4 and IPv6 backbone in Fremont, California. 10 pings and 10 UDP DNS requests were sent to each listed server and the minimum (best) time used to reduce variance due to jitter. For the totals listed below, latency within 1ms was considered the same. 1ms of RTT (round trip time) corresponds to approximately 60 miles in fiber. The test data results should be considered preliminary and incomplete, primarily useful from the point of provoking thought and further research.

The dataset used is available online at http://bgp.he.net/going-native.cgi

Total IPv6 Prefixes
843

Total IPv6 rDNS Nameservers in DB
969

IPv6 rDNS Nameservers with an IPv4 address
951

IPv6 rDNS Nameservers with an IPv6 address
361

IPv6 rDNS Nameservers reachable via IPv4 (ping or dns)
810

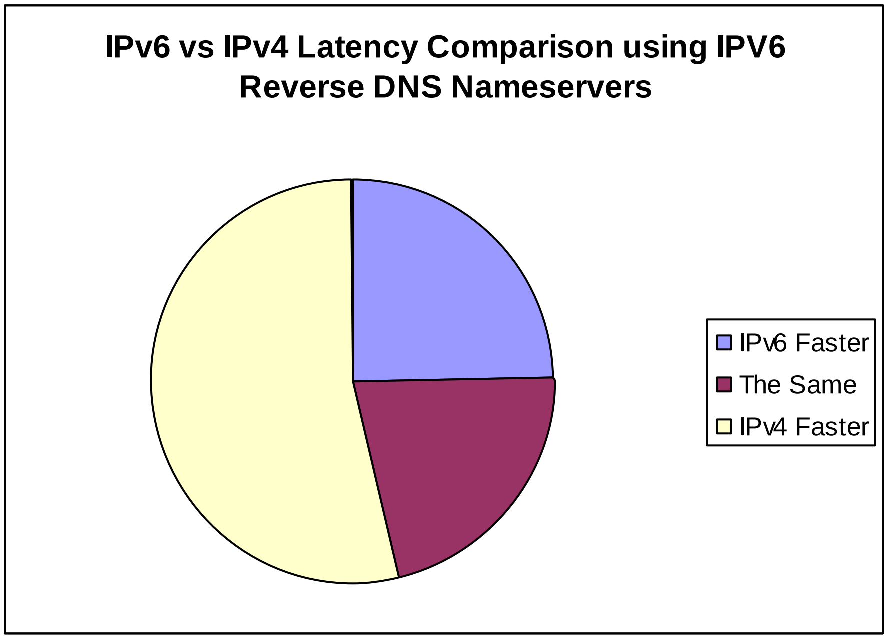
IPv6 rDNS Nameservers reachable via IPv6 (ping or dns)
284

IPv6 rDNS Nameservers reachable via both IPv4 and IPv6
256

IPv6 rDNS Nameservers where IPv6 is faster than IPv4 (by more than 1ms)
63

IPv6 rDNS Nameservers where IPv4 and IPv6 are the same speed (within 1ms)
56

IPv6 rDNS Nameservers where IPv4 is faster than IPv6 (by more than 1ms)
137



IPv6 vs IPv4 Latency Comparison using IPV6 Reverse DNS Nameservers

Networks with a higher IPv6 than IPv4 latency should consider fixing it by natively IPv6 peering with Hurricane at any one of 18 different exchange points in the US and Europe. http://he.net/adm/peering.html

Notes

Going Native Dataset
http://bgp.he.net/going-native.cgi

University of Oregon Route Views Project
http://www.routeviews.org/

RIPE Routing Information Service (RIS)
http://www.ripe.net/ris/

Marco d'Itri's zebra table parser
http://www.linux.it/~md/software/zebra-dump-parser.tgz

Gert Döring’s IPv6 BGP filter recommendations
http://www.space.net/~gert/RIPE/ipv6-filters.html

Latency in Fiber
http://www.onlamp.com/pub/a/onlamp/excerpt/bgp_ch06/index.html

Hurricane Electric (AS6939) Exchange Point Connections (as of 2007-09-26)

```
NAP             Status  Speed   IPv4           IPv6
--------------- ------- ------- -------------- ------------------------
EQUINIX-ASH     UP      10GigE  206.223.115.37 2001:504:0:2::6939:1
EQUINIX-CHI     UP      GigE    206.223.119.37 2001:504:0:4::6939:1
EQUINIX-DAL     UP      GigE    206.223.118.37 2001:504:0:5::6939:1
EQUINIX-LAX     UP      GigE    206.223.123.37 2001:504:0:3::6939:1
EQUINIX-SJC     UP      10GigE  206.223.116.37 2001:504:0:1::6939:1
LINX            UP      10GigE  195.66.224.21  2001:7f8:4:0::1b1b:1
LoNAP           UP      GigE    193.203.5.128  2001:7f8:17::1b1b:1
AMS-IX          UP      10GigE  195.69.145.150 2001:7f8:1::a500:6939:1
NL-IX           UP      GigE    193.239.116.14 2001:7f8:13::a500:6939:1
PAIX Palo Alto  UP      10GigE  198.32.176.20  2001:504:d::10
PAIX New York   UP      10GigE  198.32.118.57  2001:504:f::39
NYIIX           UP      10GigE  198.32.160.61  2001:504:1::a500:6939:1
LAIIX           UP      GigE    198.32.146.50  2001:504:a::a500:6939:1
NYCX            UP      GigE    198.32.229.22
BIGEAPE         UP      100BT                  2001:458:26:2::500
SIX             UP      10GigE  198.32.180.40  2001:478:180::40
PaNAP           UP      10GigE  62.35.254.111  2001:860:0:6::6939:1
DE-CIX          UP      10GigE  80.81.192.172  2001:7f8::1b1b:0:1
NOTA            UP      10GigE  198.32.124.176 2001:478:124::176
```