Deploying IPv6 in the Google Enterprise Network. Lessons learned.

Haythum Babiker <*haythum@google.com*>
Irena Nikolova <*iren@google.com*>
Kiran Kumar Chittimaneni <*kk@google.com*>

Abstract

This paper describes how we deployed IPv6 in our corporate network in a relatively short time with a small core team that carried most of the work, the challenges we faced during the different implementation phases, and the network design used for IPv6 connectivity.

The scope of this document is the Google enterprise network. That is, the internal corporate network that involves desktops, offices and so on. It is not the network or machines used to provide search and other Google public services.

Our enterprise network consists of heterogeneous vendors, equipment, devices, and hundreds of in-house developed applications and setups; not only different OSes like Linux, Mac OS X, and Microsoft Windows, but also different networking vendors and device models including Cisco, Juniper, Aruba, and Silverpeak. These devices are deployed globally in hundreds of offices, corporate data centers and other locations around the world. They support tens of thousands of employees, using a variety of network topologies and access mechanisms to provide connectivity.

Tags: IPv6, deployment, enterprise, early adoption, case study.

1. Introduction

The need to move to IPv6 is well-documented and well-known - the most obvious motivation being IANA IPv4 exhaustion in Feb 2011. Compared to alternatives like Carrier-Grade NAT, IPv6 is the only strategy that makes sense for the long term since only IPv6 can assure the continuous growth of the Internet, improved openness, and the simplicity and innovation that comes with end-to-end connectivity.

There were also a number of internal factors that helped motivate the design and implementation process. The most important was to break the chicken-or-egg problem, both internally and as an industry. Historically, different sectors of the Internet have pointed the finger at other sectors for the lack of IPv6 demand, either for not delivering IPv6 access to users to motivate content or not delivering IPv6 content to motivate the migration of user networks. To help end this public stalemate, we knew we had to enable IPv6 access to Google engineers to launch IPv6-ready products and services.

Google has always had a strong culture of innovation and we strongly believed that IPv6 will allow us to build for the future. And when it comes to universal access to information we want to provide it to all users, regardless of whether they connect using IPv4 or IPv6.

We needed to innovate and act promptly. We knew that the sooner we started working with networking equipment vendors and with our transit service providers to improve the new protocol support, the earlier we could adopt the new technology and shake the bugs out. Another interesting problem we were trying to solve in our enterprise organization was the fact that we are running tight on private RFC1918 addresses - we wanted to evaluate techniques like Dual-Stack Lite, i.e to make hosts IPv6-only and run DS-Lite on the hosts to provide IPv4 connectivity to the rest of the world if needed.

2. Methodology

Our project started as a grass-roots activity undertaken by enthusiastic volunteers who followed the Google practice of contributing 20% of their time to internal projects that fascinate them. The first volunteers had to learn about the new protocol from books and then plan labs to start building practical experience. Our essential first step was to enable IPv6 on our corporate network, so that internal services and applications could follow.

Our methodology was driven by four principles:

1. Think globally and try to enable IPv6 everywhere: in every office, on every host and every service and application we run or use inside our corporate network.
2. Work iteratively: plan, implement, and iterate launching small pieces rather than try to complete everything at once.
3. Implement reliably: Every IPv6 implementation had to be as reliable and capable as the IPv4 ones, or else no one would use and rely on the new protocol connectivity.
4. Don't add downtime: Fold the IPv6 deployments into our normal upgrade cycles, to avoid additional network outages.

3. Planning and early deployment phases

First, we started creating a comprehensive addressing plan for the different sized offices, campus buildings, and data centers. Our initial IPv6 addressing scheme followed the guidelines specified in RFC5375 (IPv6 Unicast address assignment):

- Assign /64 for each VLAN
- Assign /56 for each building
- Assign /48 for each campus or office

We decided to use the Stateless Address Auto-Configuration capability (SLAAC) for IPv6 address assignments to end hosts. This stateless mechanism allows a host to generate its own addresses using a combination of locally available information and information advertised by routers, thus no manual address assignment is required.

As manually configuring IP addresses has never really been an option, this approach addressed various operating systems DHCPv6 client support limitations and therefore sped the rollout of IPv6. It also provides a seamless method to re-number and provide address privacy via the privacy extension feature (RFC 4941). Meanwhile, we also requested various sized IPv6 space assignments from the Regional Internet Registries. Having PI (Provider Independent) IPv6 space was required to solve any potential multihoming issues with our multiple service providers.

Next, we had to design the IPv6 network connectivity itself. We obviously had several choices here; we preferred dual-stack if possible, but if not then we had to build different types of tunnels (as a 6-to-4 transitioning mechanism) on top of the existing IPv4 infrastructure or to create a separate IPv6 infrastructure. The latter was not our preferred choice since this would have meant the need for additional time and resources to order data circuits and to build a separate infrastructure for IPv6 connectivity.

We also tried to design a scalable IPv6 backbone to accommodate all existing WAN clouds (MPLS, Internet Transit and the Google Production network, which we use as our service provider for some of the locations). Along with the decision to build the IPv6 network on top of the existing physical one we tried to keep the IPv6 network design as close to the IPv4 network in terms of routing and traffic flows as possible. The principle of changing only the minimum amount necessary was applied here.

By keeping the IPv6 design simple, we wanted to ensure scalability and manageability; also it is much easier for the network operations team to support it. In order to comply with this policy we decided to use the following routing protocols and policies:

- HSRPv2 - First hop redundancy
- OSPFv3 - Interior gateway protocol
- MP-BGP - Exterior gateway protocol
- SLAAC - for IP addresses assignments for the end hosts.

Our proposed routing policy consist of the following rules: we advertise the office aggregate routes to the providers, while only accept the default route from the transit provider.

We also aggressively started testing and certifying code for the various hardware vendors' platforms and working on building or deploying IPv6 support into our in-house built network management tools.

In 2008 we got our first ARIN-assigned /40 IPv6 space for GOOGLE IT and we deployed a single test router having a dual-stacked link with our upstream transit provider. The reason for having a separate device was to be able to experiment with non-standard IOS versions and also to avoid the danger of having higher resource usage (like CPU power).

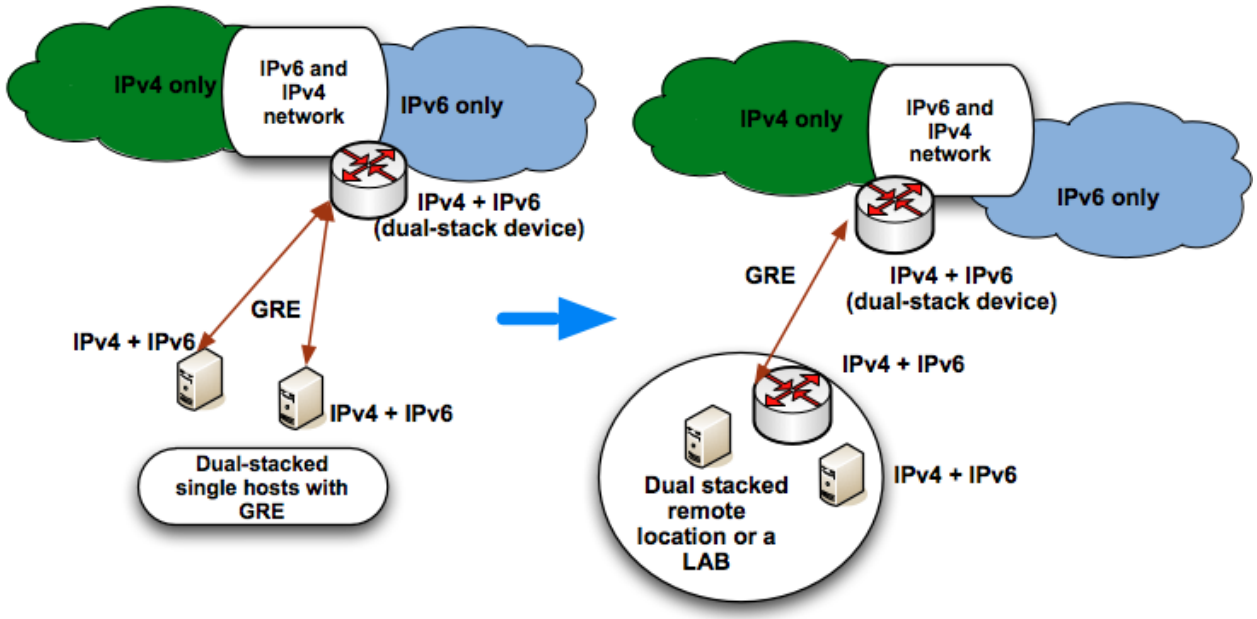

Figure 1: phase I - dual-stack separate hosts and labs

The early enthusiasts and volunteers to test the IPv6 protocol had one GRE tunnel each running from their workstations to this only IPv6 capable router, which was sometimes giving around 200ms latency, due to reaching relatively closely located IPv6 sites via a bro-

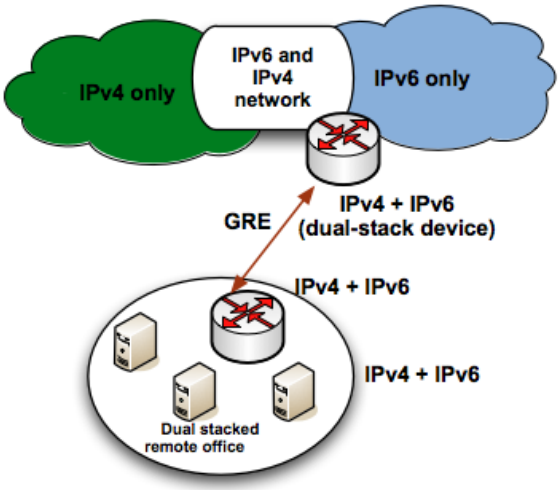

Figure 2: phase II - dual-stack offices

ker device on the other side of the world.

The next steps during this initial implementation phase were to create several fully dual-stacked labs (Figure 1) and connect them to the dual-stacked router using the same GRE tunnels, but instead of at certain hosts, these GRE tunnels were now terminated at the lab routers. In the next phase we started dual-stacking entire offices and campus buildings (Figure 2) and then building a GRE tunnel from the WAN Border router at each location to the egress IPv6 peering router.

In the third phase we started dual-stacking entire offices, while trying to prioritize deployment in offices with immediate need for IPv6 (Figure 3), e.g. engineers working on developing or supporting applications for IPv6.

Using this phased approach allowed us to gradually gain skills and confidence and also to confirm that IPv6 is stable and manageable enough to be deployed in our network globally.

4. Challenges

We faced numerous challenges during the planning and deployment phases, not only technical, but also administrative and organizational such as resource assignment, project prioritization and the most important - education, training and gaining experience.

4.1 Networking challenges

The most important technical issue we faced was the fact that the major networking vendors lack enterprise IPv6 features, especially on some of the mid-range devices and platforms. Also certain hardware platforms support IPv6 in software only, which causes high CPU usage when the packets are handled by the software. This has a severe performance impact when using access control lists (ACLs). In another example of limitations with some of our routing platfors vendors, the only IPv6 tunneling mechanism available is Generic Routing Encapsulation (GRE). The main reason for this partial IPv6 implementation in the networking devices is that most vendors are not even running IPv6 in their own

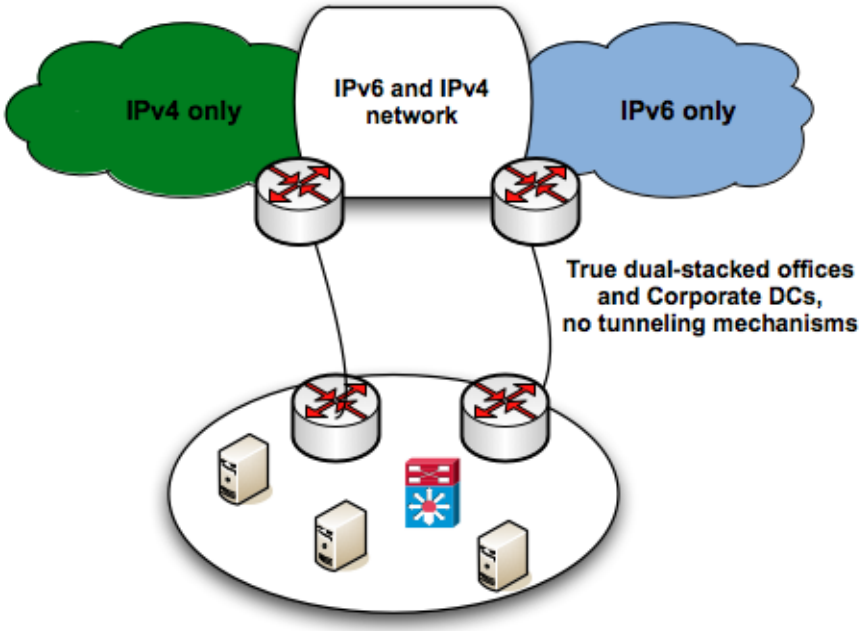

Figure 3: phase III - dual-stack the upstream WAN connections to the transit and MPLS VPN providers

corporate networks. Also the TCAM table in one of the switch platforms we use is limited when you enable an IPv6 SDM routing template. Another example of a network challenge is the software only routing support of IPv6 in the platforms we deploy as wireless core switches.

Our wireless equipment vendor did not have support for IPv6 ACLs and currently lacks support for IPv6 routing. We also faced the problem with VLAN pooling on the wireless controllers - in that mechanism, the wireless controller assigns IP addresses from the different VLANs (subnets) on a round-robin basis as each wireless client logs in. We wanted to utilize multiple VLANs using this technique to provide easy address management and scalability. However, the VLANs pooling implementation on our specific vendor leaked IPv6 neighbor discovery and multicast Router Advertisements (RAs) between the VLANs. This introduced IPv6 connectivity issues as the clients were able to see multiple RAs from outside the client VLANs. The solution provided by the vendor in a later software release was to implement IPv6 firewalling to restrict the neighbor discovery and Routers Announcement multicast traffic leaking across VLANs.

One more example is the WAN Acceleration devices we use in our corporate network - we cannot encrypt or accelerate IPv6 traffic using WCCP (Web Cache Control Protocol), since the current protocol standard (WCCPv2) does not even support IPv6 and thus is not implemented on the devices. Currently we are evaluating workarounds like PBR (Policy Based Routing) to overcome this.

A related problem is that we lacked good test tools that support IPv6 and thus we could not do real stress testing with IPv6 traffic. One interesting unexpected challenge with the dual-stack infrastructure is getting a feel how much traffic on the links is IPv4 and how much IPv6. We still needed to work on collecting, parsing, and properly displaying Netflow stats for IPv6 traffic. The problem that we have here is due to a specific routing platform vendor that is no longer developing the OS branch for the specific hardware model we use, while the current OS versions do not support NetFlow v9.

We also faced some big challenges when working with various service providers. The SLA that they support is very different than the SLA for IPv4, and, in our experience, the implementation time for turning up IPv6 peering sessions takes much longer than IPv4 ones. In addition, our internal network monitoring tools were unable to alert on base monitoring for IPv6 connectivity until recently.

4.2 Application and client software

The main problem was that the many application whitelists we use for multiple internal applications were initially not developed to support IPv6, so when we first started implementing IPv6 the users on the IPv6 enabled VLANs and offices were not able to reach lots of our internal online tools. We even got some false positive security reports saying that some unknown addresses were trying to access restricted online applications.

In order to fight this problem, we aimed at phasing out old end-host OSes and applications that do not support IPv6 or where IPv6 is disabled by default. Although we no longer support obsolete host OSes in our corporate network, there are still some IPv6 related issues with some of the supported ones. For example, some of them use ISATAP tunneling as their default IPv6 connectivity method, which means that very often the IPv6

connectivity might be broken due to problems with the remote ISATAP router and infrastructure.

We also still have not fully solved the printer problem, an most do not support IPv6 at all or just for management.

Unfortunately large groups of systems and applications exist that cannot be easily modified, even to enable IPv6 - for example heavy databases and some of the billing applications due to the critical service they offer. And on top of that, the systems administrators are often too busy with other priorities and do not have the cycles to work on IPv6 related problems.

5. Lessons Learned

We learned a lot of valuable lessons during the deployment process. Unfortunately, the majority of the problems we've faced were unexpected.

Since lots of providers still do not offer dual-stack support to the CPE (customer-premises equipment), we had to use manually built GRE over IPSec tunnels to provide IPv6 connectivity for our distributed offices and locations.

Creating tunnels causes changes in the maximum transmission unit (MTU) of the packets. This often causes extra load on the router's CPU and memory, and all possible fragmentation and reassembly adds extra latency. Since we often do not have full control over the network connectivity from end to end (e.g. between the different office locations) we had to lower the IPv6 path MTU to 1416 to avoid possible packets being lost due to lost ICMPv6 messages on the way to the destination.

Another big problem we had to deal with was the end host OSes immature IPv6 support. For example, some of them still prefer IPv4 over IPv6 connectivity by default. Some others do not even have IPv6 connectivity turned on by default, which makes the users of this OS incapable of testing and providing feedback for the IPv6 deployment. It also turned out that another popular host OS does not have client support for DHCPv6 and thus we were forced to go with SLAAC for assigning IPv6 addresses to the end hosts.

We ran into countless applications problems too: No WCCP support for IPv6, no proxy, no VoIP call managers, and many more. When trying to talk to the vendors they were always saying - *if there is a demand for IPv6 support at all, we've never heard it before.*

In summary, when it comes to technical problems we can confirm that there is a lot of new, unproven and therefore buggy code, and getting our vendors aligned so that everything supports IPv6 has been a challenge.

Regarding the organizational lessons we learned - the most important one is that IPv6 migration potentially touches everything, and so migrating just the network or just a single service or application or platform does not make sense by itself. This project also turned out to be a much longer term project than originally intended. We've been working on this project for 4 years already and we are still probably only half way to completion. Still, the biggest challenge is not deploying IPv6 itself, but integrating the new protocol in all management procedures and applying all IPv4 current practice concepts for it too - for example the demand for redundancy, reliability and security.

6. Summary

The migration to IPv6 is not an L3 problem. It is more of an L7-9 problem: resources, vendor relationship/management, and organizational buy-in. The networking vendors' implementations mostly work, but they do have bugs: we should not expect something to work just because it is declared supported.

Because of that we had to test every single IPv6 related feature, then if a bug was found in the lab we reported it and kept on testing!

7. Current status and future work

Around 95% of the engineers accessing our corporate network have IPv6 access on their desks and are whitelisted for accessing Google public services (search, Gmail, Youtube etc.) over IPv6. This way they can work on creating, testing and improving IPv6 aware applications and Google products. At the same time internally we keep on working on enabling IPv6 support on all our internal tools and applications used in the corporate network.

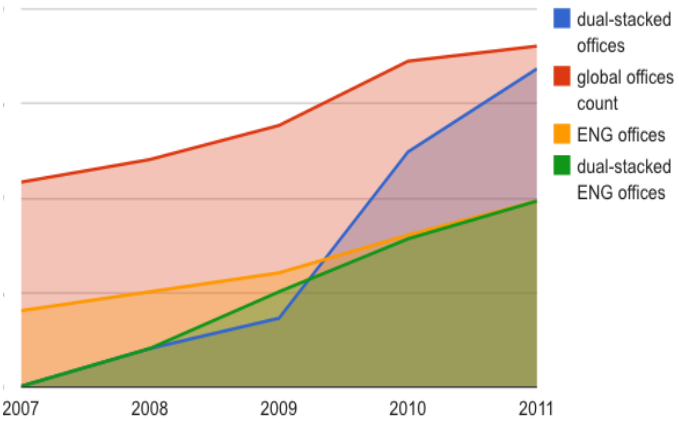

Figure 4: Timeline for dual-stacking Google corporate locations

In the long run, the potential of introducing DHCPv6 (state-full auto-configuration) can be investigated given the advantages of DHCP flexibility and better management. However enabling this functionality still depends on the support of the end hosts DHCPv6 client on the desktop platforms.

We also want to revisit the IP addressing allocation of /64 to every subnet on the corporate network, since a new RFC 6164 has been published that recommends assigning /127 addresses on P2P links.

Since the highest priority for all organizations is to IPv6-enable their public-facing services, following our experience we can confirm - dual-stack works well today as a transition mechanism!

There is still quite a lot of work before IPv4 can be turned off anywhere, but we are working hard towards it. The ultimate goal is to successfully support employees working on an IPv6-only network.