

Ask Ars: How should my organization approach the IPv6 transition?

By [Clint Ecker](#) | Last updated February 19, 2011 3:03 PM



Whenever Ars runs an article about the increasing global scarcity of IPv4 addresses or an IPv6-related topic, we inevitably hear from some readers that they would like to see Ars available over IPv6. We thought we'd explain why we haven't made that move yet.

Why should *you* care?

First though, we want to help your organization or business decide if it should be pursuing the goal of making your websites or applications available on IPv6. There are so many kinds of businesses and applications out there that it's hard to generalize, but the first question you should ask yourself is whether making this transition even makes sense right now.

If your company is an ISP, datacenter, network provider, or similar, this article is probably not for you. You've likely dealt with all your IPv6 issues long ago or are working through a transition right now. The level at which this article becomes more worthwhile is for SaaS providers (think 37signals' products, Amazon's AWS platform, Heroku, Google Docs, and so forth), and content-heavy organizations (like *Ars Technica*, *The New York Times*, Government portals, University Websites, your company's website, or an online store like Amazon.com).

First, identify your goals for making your site available over IPv6. Are you losing customers and money being IPv4-only? This could start happening very soon in Asian nations where customers will be handed IPv6 addresses when the current pools run dry (although the impact is debatable, as their ISP will likely tunnel them to IPv4 sites for a long time). Do you have philosophical ambitions and want to set a good example? Do you

want geek cred with your community, or do you want to make sure you and your company are ready when that fateful day comes?

Most are not going to take action until they can show some kind of financial benefit. As we'll show, auditing and transitioning even a small operation will take a good chunk of time, research, and implementation. This all costs money in labor, but it could also cost money in hardware upgrades, support time, and software licenses. In addition, it could lead to downtime, which could result in lost profits. All of these will have to be accounted for and weighed against the potential financial benefit. Will your signups increase 5% by being available over IPv6? Can you leverage the transition as positive press for your product and bring in new customers on geek cred alone? Can you establish yourself amongst your peers as an expert on IPv6 transitions, which could lead to more contracts and projects in the future?

In most cases, the financial benefit will be near \$0 at this point—most likely you will lose money—so you'll probably be searching for a promotional or preventative angle. In those cases, or in cases where you can identify a financial benefit, you'll need to start exploring and detailing the process. We've tried to lay out some concrete steps and guidelines below to get you started. Beyond that, we'll detail exactly how we've managed this process for Ars Technica and how we'll approach IPv6 in the future.

The eight circles of IPv6 transition agony

You'll need to start at the bottom and work your way up. You host your software somewhere, and that somewhere is generally a hosting provider. That hosting provider may manage its own network or live on top of a different network provider. Next up the stack will be your organization's networking equipment. These live between your hosting company's network and your servers. Then you have the operating systems themselves, then the Web servers, your home-grown applications, your third-party applications, glue code, and at the very end, the place where you store IP addresses for later analysis (like a database or logging system).

1. Your network provider

This is either where your servers are colocated or hosted. (If you've got your own datacenter, you would've probably figured this IPv6 transition out by now.) If you're colocating or leasing equipment in someone else's datacenter, they've probably already worked out the logistics of providing IPv6 connectivity to their customers (you). You can generally find out what level they support by poking around on their website. If not, a quick email to your support contact or opening a ticket should suffice. If you're hosting your applications on a non-dedicated or VPS hosting provider, it's less likely they've personally done the legwork on IPv6. They may be rehosting service from a bigger provider, or they could be a budget provider of shared-hosting products.

The good news is that even budget hosting providers like Dreamhost are getting on board with IPv6 these days. So it's likely that you're covered.

2. Your networking equipment

Depending on your setup, you may have your own dedicated networking equipment that lives between your servers and your network provider. All of this equipment will need to be checked out to make sure it can support IPv6. Anything purchased in the past few years probably has the capability built in—it'll be ready to go with a little configuration. It's also possible that you may need to update the firmware to get support. Also not a huge deal, but you may have to account for the downtime. Then there is cheap and old equipment. Cheap stuff like switches may not have support at all. Similarly, equipment bought long enough ago may have no upgrade path. In either case, this hardware will need to be replaced with modern equivalents.

3. Operating systems

Your next task will be to confirm that your server operating systems actually have IPv6 support and that support is enabled. [OS X](#) has been [IPv6 ready](#) since at least 10.4.8, with various applications and services gaining IPv6 support in 10.5; Apple ships Macs with IPv6 installed and enabled. Windows XP has had support since [SP2](#), but it is not installed and enabled by default. Windows Vista and 7 have IPv6 support installed and active right out of the box. Windows 2000 has experimental IPv6 support, but there doesn't seem to be a consensus on whether you should be using it or not. Windows Server [2003](#) and [2008](#) both come with IPv6 support. Linux, BSD, and UNIX have [long supported IPv6](#), so you're most likely okay there. Even Haiku (née BeOS) [received IPv6 support](#) recently.

In all likelihood, your OS has long had IPv6 support. The exceptions will probably be those servers that have been plodding away in a server closet since the '90s—the ones that no one thinks about anymore. Hopefully you're not in that situation. If you are, the best of luck to you!

4. Software networking applications

More and more these days, businesses opt to run software-based networking servers like proxies and load-balancers. Old versions may or may not fully support IPv6, but luckily these are extremely easy to update. Moving from very old versions to new editions that support IPv6 may cause issues if other backwards-incompatible changes have wormed their way into the code since then. You'll need to make sure all your configuration settings and tweaks still work the way you thought they did before rolling the update out to production.

5. Web servers

Next up the stack are your Web servers and application code. Most modern Web servers have had IPv6 support for quite a while. Apache since 2.0.14 (but with IPv6 virtual hosts broken until 2.0.28), IIS since 6.0 (but only without certain functionality like bandwidth throttling enabled, and with IP address restrictions until 7.0), lighttpd has always had it, and nginx has had it since 0.7.36. You can find a huge [list of IPv6 compatible applications](#) on Wikipedia.

6. Application Code

Once you've verified that your Web server can handle it, you'll need to step up into your application code. This is probably the meat and potatoes of your business. This step is going to be highly unique to what your company does, and it will be hard for us to recommend any specific course of action, but we can give you some ideas of places to look for problems.

You'll need to look at all your homegrown code, your third-party products, and any home-grown plugins and glue code that ties all this together. In Ars Technica's case, for example, we have a fairly limited set of stakeholders and systems. We have the front-end framework that templates out the site and displays content like this article. We have a CMS (Moveable Type) that our authors and editors spend most of their day in. We have our discussion boards that host our forums and also power our commenting system and user database system. Finally, we've written a number of plugins and ancillary services to make all of this work together.

7. Are you a bad enough dude(tte) to handle IPv6 addresses?

The key to any IPv6 transition will be determining where and how your applications deal with IP addresses at all. Most of the hard work is below this layer in networking equipment and Web servers. Most of our systems have no idea about IP addresses and will never have to deal with one. Our forums/comments/user database, however, deals with lots of IP addressees for a number of reasons.

The product phpBB 3 that powers all of this stores the IP address of newly registered users, stores the IP address of the user alongside every post, and lets our moderators place IP bans on abusive users. There are several points where throwing an IPv6 address into the mix could mess things up, but the two big ones are when IP checking or calculations take place and when those IP addresses are stored in the database.

What do I mean about IP address calculations, matching, and verification? phpBB, for example, allows an administrator to block a single IP address, a range of IP addresses, or use a wildcard syntax. Your application could allow for hostmasks (/24, /16) to be stored alongside an IP address. What you'll need to do is verify that your code can accept an IPv6 address and do all the same work. You'll need to make sure your code can handle abbreviated and full IPv6 addresses. You'll also need to make sure that any IPv4-specific paradigms can be applied to IPv6. Do you really need to let people specify host masks alongside IPv6 addresses? Are they specified in the same way? These are all things you'll need to research and determine for your own specific application.

The second place you might run into problems is when you store IP addresses in a database. You might have configured those columns over-restrictively to perfectly fit an address' four octets and decimals. You might store the IP address in binary. You'll need to make sure that you can store an IPv6 address in either its abbreviated or full format.

Making these changes in your own, home-grown software will probably be fairly straightforward. There are plenty of IPv6 libraries and IPv6 support in your programming language of choice, but you'll need to [understand their idiosyncrasies](#) and be prepared to do some more research.

8. Third-party software

When it comes to software that is out of your hands, you'll need to either wait or fork the code. phpBB 3 is both the most extensive third-party product we use and the one most affected by IPv6 concerns (as explained above). Luckily for us, the phpBB team and community have been nailing down and cleaning up IPv6 bugs for a while. Most of these are slated to be rolled out in the next major release. You'll need to make a list of all the software you maintain that is out of your hands and see if it deals with IP addresses in any capacity, and then you'll have to determine if there's support already baked-in or if it's in the release pipeline. You could theoretically fork the project if it's open-source, but that's a pretty big commitment. In most cases, you'll just have to wait it out. These delays will be important factors in determining the timeline for your transition.

And that's just about it. Once you've nailed down all the problem points, you're probably ready to go. The only job left is to enable the DNS records and see what happens. You'll undoubtedly run into problems that will be worked out over time. For a (really) long time you're going to see both IPv4 and IPv6 clients, so take that into account as you keep moving forward.

How is Ars handling all of this?

Now that you have an understanding of all the potential pain points, we thought we'd run down our specific situation, and try to stick to the levels we outlined above.

First up, at the bottom is our hosting and network provider, ServerCentral. Their network is IPv6-capable and they can provide support up to our racks. Our local networking equipment is where the issues begin to crop up. Not all of this equipment is IPv6-ready or can be made IPv6-ready. Our routers, load balancers, and firewalls will either need to be replaced or get software updates. This will mean downtime and money on our part.

Our operating systems (Linux) and webservers (lighttpd), are all fine, and the same goes for our caching proxy, Varnish. The software that powers our Web front-end is oblivious to anything to do with network-level stuff, but the software that powers our commenting system and forums is not currently able to cope with IPv6-style addresses. This software—phpBB3—won't be fully IPv6 compatible until the next major revision. Even when this release comes out, we'll need to test it before it's deployed, and probably even setup some kind of IPv6 staging area where we can make sure it handles our use-case in a real-world environment. This will all take time away from regular projects to support the site, and it will take money (our labor and the costs for the staging infrastructure).

It's also currently unclear what would happen if Ars were to enable IPv6 tomorrow. We know from past experiments by Google and others that a percentage of our customers would find Ars to be unreachable due to poorly configured IPv6 support in the chain between their home and our servers. A good primer on a lot of the issues and challenges involved can be found in our four-page article on the subject: [Why the IPv4 to IPv6 transition will be ugly](#).

This is why [World IPv6 Day](#) and the results gleaned from it will be so important. These mega-sites will be looking to see what effects this has on their traffic, their services, and most importantly, their revenue. If these sites turn on IPv6 and see even a 0.1% failure rate, it would be catastrophic in terms of the impact on their bottom line.

Ars will be watching for the results of these trials to help guide us to the right decision on this issue. We know what steps we need to take to make the site available over IPv6, but we just don't quite know when it will be a worthwhile investment on our part.

It's worth noting that Ars is an existing service, so we have no particular need for more IPs at the moment, and likely won't for a long time. If we were contemplating the launch of a new site, we would probably enable IPv6 from the beginning, though we might skip the AAAA record for the time being.

To round this out, we've thought about a number of auxiliary issues that we and our partners will have to deal with in the upcoming months and years:

- Analytics packages may rely on each user, or a significant number of users, having a unique IP address. Google Analytics, which we use, uses cookie-based tracking to get around this. However there are certain parts of the package, namely geographical tracking down to the state and city level, that require the use of IP tracking. This could break in subtle and unexpected ways when large, nationwide carriers begin using NAT for large swaths of customers.
- On occasion, we must rely on IP-based bans to prevent abusive users from accessing our comments and forums. For pretty much every case, this is a successful strategy to keep these spammers and ill-adjusted individuals out. With the upcoming surge in NAT, however, our options in these scenarios are drastically reduced. We (or our forums vendor) will need to find a new solution to this problem.

- Even though phpBB will be mostly IPv6-compatible in its next major release, it's not clear if there will have been extensive testing done across the system. Furthermore, we've done a number of modifications to the source and written our own plugins. All of this will need to be tested to ensure nothing breaks if and when IPv6 users begin accessing the boards.

We hope this article has given you a good overview of the basics involved in transitioning to IPv6, along with a little context as to why Ars hasn't made the switch and when we might. If you have any corrections or additional information our readers might find useful on this topic, please leave them in the comments below.

<http://arstechnica.com/business/news/2011/02/ask-ars-how-should-my-organization-approach-the-ipv6-transition.ars?comments=1&start=40#comments-bar>

[brainee28](#) | 4 days ago | [permalink](#)

I appreciate you posting this article; however it didn't really answer the questions that I needed answered.

For an organization of my type, which is using a connection to an ISP's hardware (not a hosting provider or ISP), using a standard firewall/router solution, is there any benefit for me to go to IPv6?

Most of my users won't be affected by this, as everything is NATted now. The only thing for me is that I use IPv4 addresses for hosting servers (email, remote access, etc). It seems like to me the ISP's should change their hardware over, and then use a 6to4 connection to translate to my network.

Is there really a need for me to transition every piece of network gear I have internally to IPv6? Internally, we'll never run out of addresses. It seems like this doesn't really affect my organization (medium sized at 200 users), does it?

[ajmas](#) | 3 days ago | [permalink](#)

Your network strategy is likely to be two-fold: one for your public servers and one for internal computers wishing to access the net.

As the IPv4 pool runs dry ISPs and new companies will have to turn to IPv6. This will mean that if you wish these customers to talk to your servers then you will need IPv6 support. You could limit the IPv6 requirements to a simple proxy that you put in front of your current infrastructure to start with. The question to ask here is whether you can afford not to have these customers?

As new services come online from new organisations, they are likely to be IPv6 only since they won't have access to IPv4 addresses. If you wish to be able to access these services then you will need to provide IPv6 accessibility to your client computers. You could temporarily buy a bit of time by using an IPv6 proxy, but you may be limiting yourself to web access.

If you feel that you aren't going to be burnt by waiting for IPv6 you can do so, but I would encourage you to ensure any new hardware you buy to be IPv6 ready. This will reduce your costs the day your actual transition happens. At the same time I would invest a bit of time trying to understand how to do the transition and what the implications are, in case a sudden business need comes up. You don't want to be dealing with too many unknowns.

[brainee28](#) | 3 days ago | [permalink](#)

I should clarify. I'm hosting my company's email and remote access. I do not have customers other than my users.

[ajmas](#) | 3 days ago | [permalink](#)

If access is purely through an intranet or VPN, then you can probably hold off IPv6 for now. Where it matters is anything that requires interaction with the internet in general.

Even if you do hold off at work, I would encourage you to install IPv6 at home if only to get acquainted with it. In fact I would encourage anyone who is responsible for a network to get acquainted with it in a practical form since you in order to learn about the issues and the solutions.