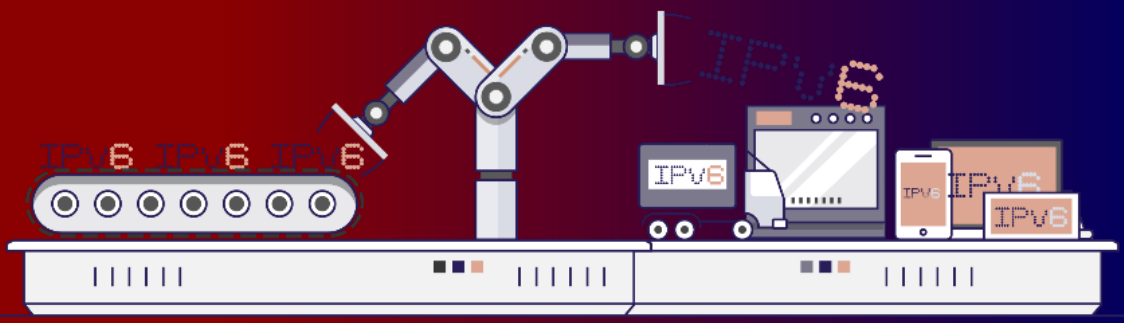
Enterprises:

How to deploy IPv6?



FROM PLANING TO DEPLOYMENT

OKTOBER 2024 // V 1.4

This document will help you to DEFINE YOUR IPv6 NEEDS, to PLAN its deployment and to ROLL IT OUT inside your organization.

It covers any type of Information System.

Table of contents

Foreword

Thanks

This guide has been originally written by Jean-Charles BISECCO, with the help of IPv6 taskforce members. It has been transfered to ASCIIDOC by Axel Schemberg.

This document is part of the work carried out by the IPv6 task force jointly led by Arcep and Internet Society France. Arcep and Internet Society France have initiated a task force dedicated to IPv6 and open to all players in the Internet ecosystem (internet service providers, hosting services, enterprises, public sector, etc.). It aims to accelerate the transition to IPv6 by allowing participants to address specific issues and share best practices.

This guide, produced by the IPv6 Task Force, aims to provide best practices for the successful implementation of the transition to IPv6. It is not intended to supersede the expertise of corporate IT teams and the authors cannot be held responsible for any difficulties encountered by a company that follows the guidelines provided.

Join our IPv6 taskforce (French form but we accept English speakers)

This document does not reflect Arcep's position, but the work of the task force contributors.

Find the latest revision of this guide here:
https://en.arcep.fr/publications/task-force-ipv6.html

Preamble

IPv6 deployment is progressing in every part of the world, and its use is no longer merely anecdotal, as it often was in the early part of the 2010s. This guide is designed to help businesses define the scale of their IPv6 implementation, explain how to implement the protocol, and provide a set of best practices.

If there is no shortage of documentation on IPv6 and its deployment, most of it tends to focus on the network layer and pertains to a horizontal view of transport, which typically applies to a carrier, a transit provider or an internet exchange point.

Providing digital services within an enterprise typically involves a more vertical model, however, with sometimes unique configurations in the upper layers, once we move towards the many production applications.

This guide aims to provide IT departments involved in corporate IPv6 transition information to ease it operationally.

If this document does refer to technical aspects, its purpose is not to teach IPv6 and will confine itself to the details required to cover the points being addressed. It will be up to you to delve deeper into the technology via existing content, purely educational material, manufacturers' documentation, blog posts, MOOC, training courses, etc.

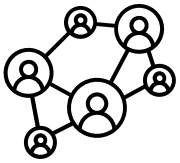
How to read

This document is primarily intended for information systems experts in charge of the transition to IPv6.

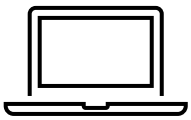
However, it contains sections and paragraphs that are aimed at different audiences.

If it is recommended that anyone who is closely involved in the project read the entire guide, the same does not apply to other readers. The logos in the margins of each section make it easy to identify the intended audience for that content.

Network



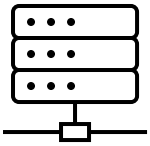
Project management



System

Telephony

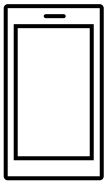
Security



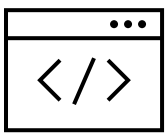
Hosting

Traceability

Home use



Mobile



Development

Some paragraphs indicate choices that will be different for larger and small structures. If it is impossible to draw a clear separation between the two, naturally the more complex and extensive a structure's information system (IS) is, the more it will fall under into the category of a large structure, even if the entity does not classify itself as such, based on the size of its staff or its geographical footprint.

Grasp of the subject

Grasp of the subject

While technology is an important part of this subject, it would be a serious mistake to reduce IPv6 to just that, even if the breadth of the subject can make for difficult project management methodology and engineering choices, which may also vary depending on the perimeters set within the organisation.

The current chapter aims to enlighten the reader on a number of methodological choices that can have a direct impact on the success of the project.

• SIMILAR PROJECTS

The subject may well be vast, but it is far from being the first of its kind. Depending on how old your company is, you may have tackled other projects that carried heavy restrictions which were similar in scope. Drawing on the lessons learned from planning and executing those projects could save you from repeating the same mistakes.

The oldest example would be the transition to the year 2000, which in some cases required profound alterations, notably to DBMS (Database Management System), as well as extensive testing for an element that was coded using two-digit values.

More recently, the introduction of TLS certificates – including for purely in-house applications that are not exposed to the internet – required changes to servers and to proxies/load balancers, etc. From middleware configuration, PKI creation or extension, adding encryption to traffic audit and monitoring systems, up to supervising the proper deployment of the entire certification chain on the different nodes, a great many elements were affected.

TLS has also been in a state of constant evolution, and requires regular changes as algorithms become obsolete, new mechanisms are added, etc.

The final example, which may be more or less common depending on a company's strategy, is the deployment of a new version of an operating system, and the migration period for customer workstations with its set of application assessments and change management.

Other large-scale projects are also common, such as deploying an ERP system, but are more business-specific and so driven by the organisation's core activity.

• UNTANGLING THE YARN

The Internet Protocol (IP) is the underlying element in the infrastructure that requires end-to-end compatibility between an information system's various constituent parts. No single analytical or reasoning method provides an efficient way to cover every impact of the change, so a combination of several will be needed.

A cartesian approach will be useful when examining elements individually – e.g.: in terms of IPv6, has my firewall supplier properly implemented a security function? Will my middleware behave the same way with an IPv4 and an IPv6 socket?

On the flipside, a systemic method of analysis is well suited to getting a big-picture view of this

mesh of interconnected equipment, each with its own system and application later. This method must be used to obtain a condensed, overall view of the IS's building blocks in the different layers.

Lastly, and because it is impossible to explore every element of the information system in detail, reductionism will sometimes be needed, notably to address the ecosystem's peripheral elements – such as supervision and log collection – allowing you to return your attention to them more efficiently during the pilot stage and not have to do the same work twice.

Trying to examine everything all at once, to inventory everything will only lead to stagnation, and is not the right method to use.

Like a good thriller writer, you need to reveal details about your characters at just the right moment. This is a long-haul project, and it is futile to take a deep dive into the entire ecosystem since it will have probably changed by the time pilots and deployments occur.

Speaking of changes in the affected areas, each one is an opportunity to deploy IPv6, so check in regularly with your different departments, to keep up with any new projects, and so not miss out on these opportunities.

Human factors

You are bound to come up against some resistance to change, so you will need to find support in each of the teams involved to be able to move forward. The steady involvement of members of the different teams needs to begin at just the right time. Too early and people's enthusiasm will wane, and the project will lose momentum; too late and it will affect the project's turnaround time. Ideally, proper planning will create the ability to increase the number of participants in a team once it needs to tackle the matter concretely.

Large, multi-department meetings should only be held to provide information, and not for brainstorming or debate. Other preliminary meetings with one or two representatives per team will be better suited to that. Specific details that concern a given team should be addressed with just that team, and possibly their neighbours in the ecosystem.

All these elements may seem par for the course, if not obvious, but because this type of project involves a very large number of players, a hierarchy needs to be established for the dialogue taking place within a large structure.

To summarise: identify one or two representatives per team and keep up to speed with future projects that could represent deployment opportunities.

Provide a high-level technical briefing of the project to all the teams at once, on a regular basis, and a less technical briefing to a larger audience, either through information meetings or the communication materials you send out.

Lastly, hold individual discussions with the teams involved when they are in the latter stages of the process, this time in a broader fashion and according to the project staging.

For instance, there is no need to solicit the team in charge of middleware on a regular basis, to ask them to get ready when the network has not yet planned a pilot trial bearing qualification servers.

• TRAINING

You will need to plan on training staff to prepare them for the introduction of IPv6.

To shepherd the transition, you will need to draft lesson plans or training courses based on the different roles, occupations and expertise that need to be taken account for a successful transition.

Before drafting a training programme, it would be wise to query staff members on the training they require to be able to perform their jobs properly with IPv6. And even in some cases, to build modules tailored to the company's particular features and needs.

At the very least, you can expect to divide the people requiring training into several different groups:

- Network ;
- Security ;
- App development.

Ideally, the project team should test out all the modules with one or two representatives of each target audience.

• MUTUAL SUPPORT

Never hesitate to contact similar size companies and organizations that are planning or carrying out an IPv6 transition, to share best practices.

IPv6 taskforce (jointly led by Arcep and Internet Society France) provides an opportunity to discuss the subject with your peers, so well worth taking advantage of. You may also join your local IPv6 forum.

We are counting on you to help us enhance this guide and document examples of products being widely used in enterprises. See the section on feedback at the end of the guide.

Needs

Earlier on, we cited examples of projects such as TLS and switching operating systems. These projects are typically driven by a need to solve a security issue or protect a technical device. Other projects are driven by legal compliance imperatives, such as ensuring the traceability of users' actions or transposing GDPR requirements onto systems. Naturally, cost effectiveness is an impetus for other projects, spurred by either business considerations or to improve the level of service, such as with orchestration projects.

It is hard to place an IPv6 project under any of these headings, and even harder to qualify IPv4 as a potential, short-term "technical debt", as might be the case with an obsolete programming language for which developers are no longuer in activity.

This section will therefore look at IPv6 use cases and lay out their degree of relevance depending on the situation.

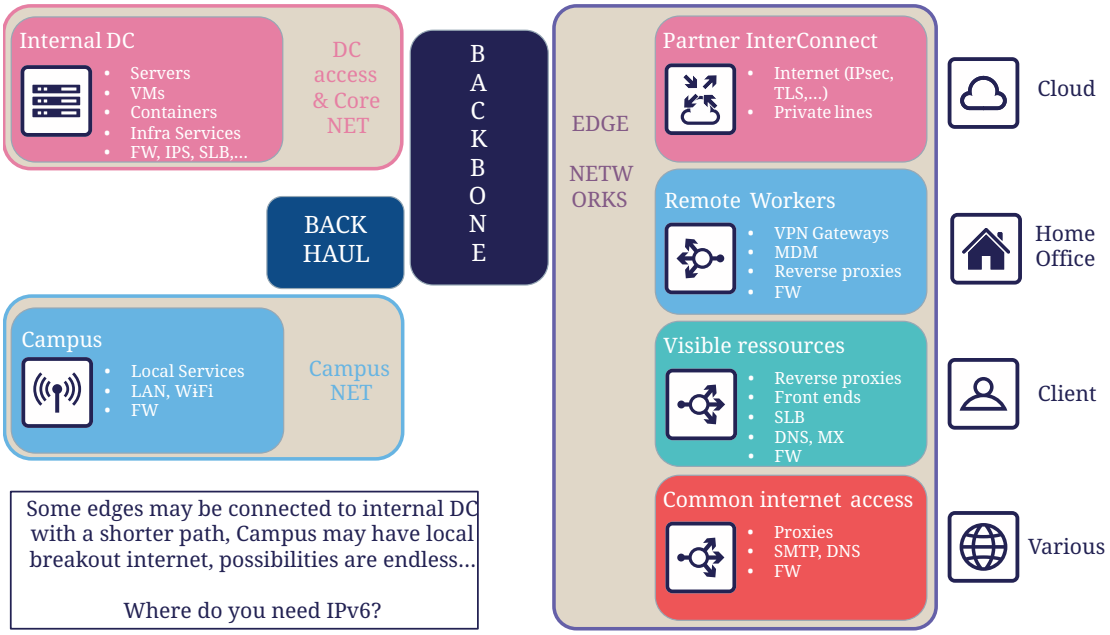


Figure 1. Where do you need IPv6?

• EXPOSURE ON THE INTERNET

Making the public face of one's network accessible in IPv6 is probably the biggest priority to be implemented.

This includes web servers but also DNS, VPN gateways...

Operators are gradually activating IPv6 on their different public networks. It began with residential connections, then mobile and finally telephone connection sharing. Most devices now operate only over IPv6 on some mobile networks, with NAT64 ensuring backwards compatibility.

So, both customers and staff are connecting to the company's infrastructures over IPv6 connections. If operators meet their deployment forecasts, it is likely that more than half the population will have a native IPv6 connection at home and on their mobile by the end of 2023. In some countries IPv6 is already offered to more than half of customers.

Despite which, if IPv4 will not disappear anytime soon, it has become such a scarce resource that in more and more cases it is being shared by multiple subscribers, using various mechanisms. If, by definition, an internet connection provides no guarantee of service, it is statistically probable that the incursion of an intermediate element in the IPv4 chain will affect its quality of service. Added to which, technical teams' lack of knowledge of these mechanisms can make solving connectivity and quality of service issues challenging, whereas an IPv6 connection is established end to end with no protocol trickery such as NAT44+PAT.

Also worth noting is that operators are starting to switch over to a world where v6 is becoming the standard on their public backbone, and where v4 is becoming a service that is transmitted more and more in an encapsulated fashion.

If your company provides services in countries with a smaller stock of IPv4 addresses, or that are simply more progressive by law, being IPv6 enabled may become necessary to align with expanding markets, or possibly to comply with a future legal requirement. Some countries like India and France encourage ISP to provide IPv6 to customers, other focus on their own internal administration like USA and Belgium, China or Germany is pushing for a full transition by 2025… Since 1 January 2021, operators in France that have acquired 5G frequencies must provide IPv6 connectivity, at the very least as an option.

In addition to these aspects, one important overall point is that IPv6 transit is now viable, and in terms of quality its maturity is nearing that of IPv4.

Pioneer adopters had long observed that IPv6 was a handicap. There were far fewer transit routes a decade ago, hence less redundancy and sub-optimal. How many of the early tutorials recommended, and rightly so, to switch off IPv6 to resolve access to a given site or public streaming service? All the more so since the Happy Eyeballs protocol did not yet exist and browsers could not save the day in a matter of milliseconds, the way they can today.

IPv6 is no longer the "handicap" that it once was: quite the opposite, thanks to veritable end to end connection.

In addition, the latency measured by Google in France, Canada and several other countries is better over IPv6, whereas the opposite was true in 2018. While IPv6 peering is becoming as good as IPv4, IPv4 now often transits through CG-NAT, and almost systematically so on mobile networks. A more minimal factor in reducing latency is the eradication of the checksum header check at each router along the IPv6 path, as well as the absence of fragmentation on routers.

It is therefore important to remember that IPv4 is being relayed more and more in a non-native fashion via CG-NAT, especially on mobile systems. Which adds an SPOF and a further element that can affect the user experience. Providing a service over IPv6 means no longer having to depend on these operators' translation infrastructures.

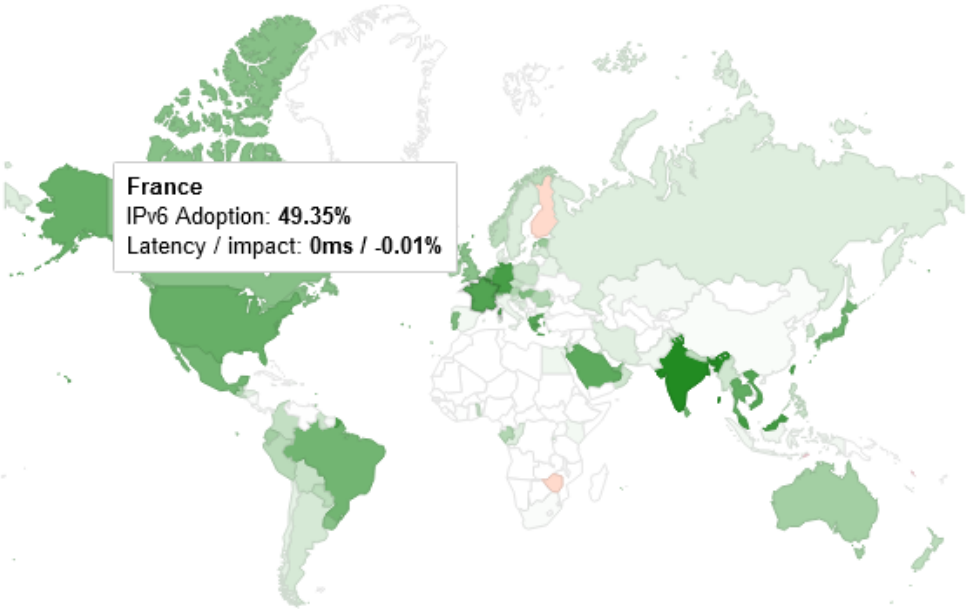


Figure 2. Statistics for access to Google services over IPv6 in France | October 2021

More and more customers gain access to IPv6 networks, in France and Germany more than the half of Google services requests are sent over IPv6.

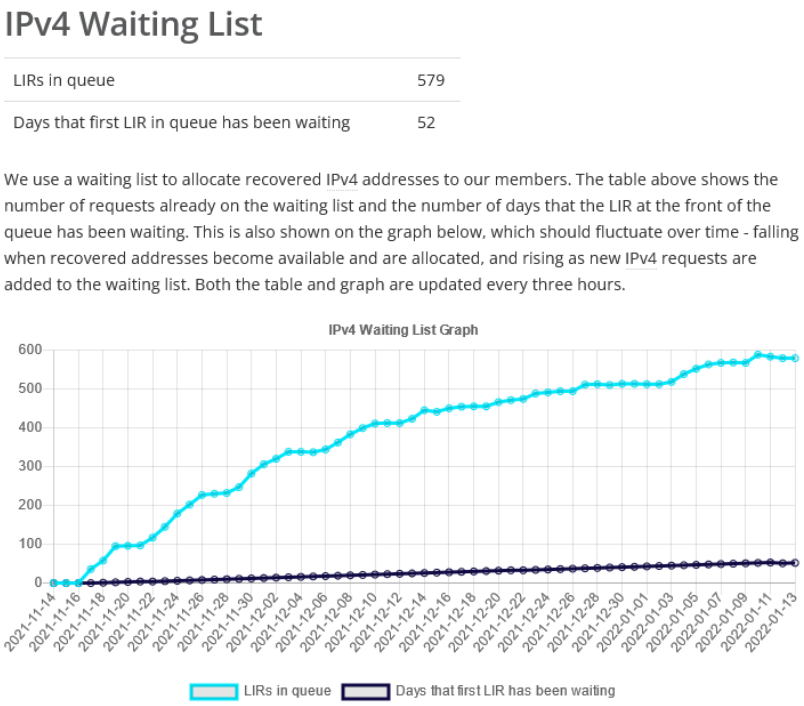


Figure 3. RIPE-NCC IPv4 waiting list January 2022

Grabing IPv4 space is getting harder every day.

• ACCESS TO EXTERNAL RESSOURCES

Regardless of size, from a protocol standpoint your company is a client of third-party resources. Here too, the number of sites and services that are IPv6-capable is rising steadily. User traffic is typically relayed through a proxy, for the purpose of filtering, protection and traceability. This proxy is usually running over IPv4 both internally and externally. And it shows. See instead:

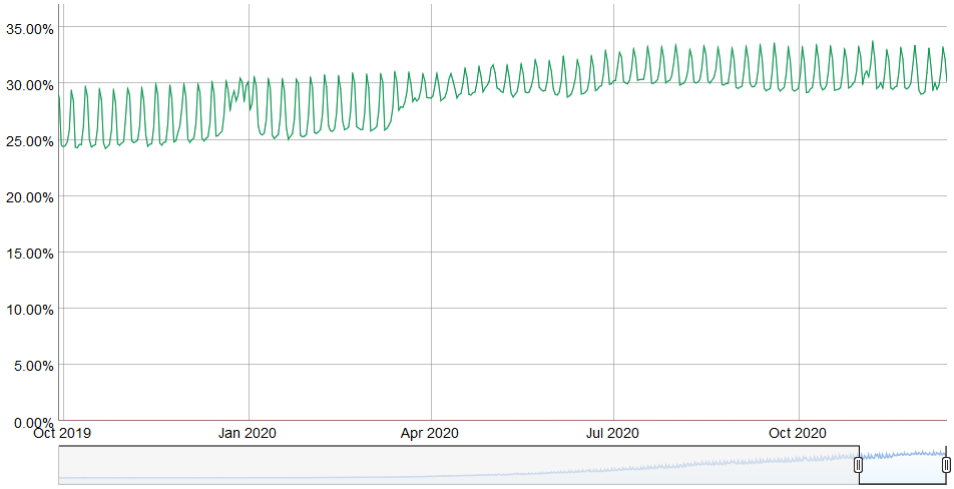


Figure 4. Percentage of global traffic accessing Google services over IPv6

Traffic patterns during the global lockdown in March-April were similar to the ones seen during the week of Christmas to New Year's. The percentage change remains in the upper bracket. Why? Simply because people connecting to the internet from home more often have access to IPv6.

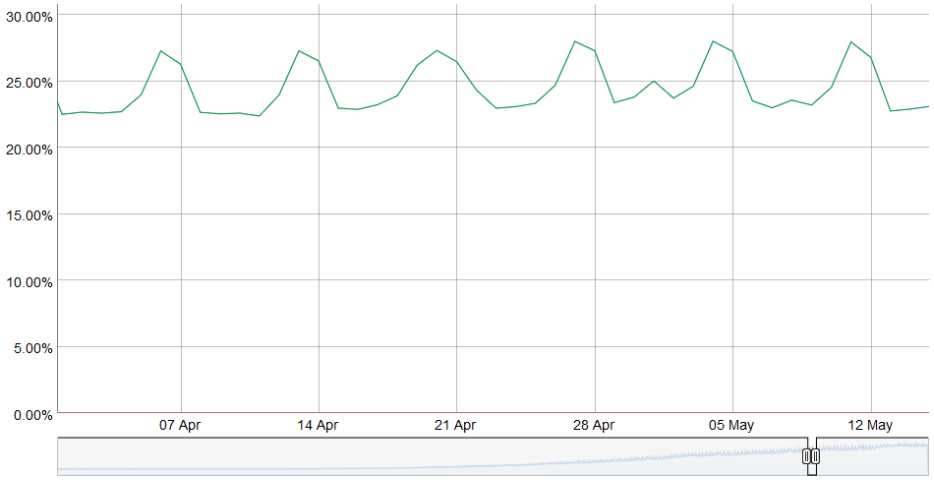


Figure 5. Percentage of global traffic accessing Google services over IPv6 in detail

This phenomenon can be seen across the course of a week: here, from the end of April to early May 2019. Peaks always occur during weekends. May 1st and 8th which are free days in many countries can be noted with hills within the two last weeks.

By Summer 2024, the average global IPv6 availability is likely to exceed 50%. Will you be part of the majority by then?

QUIC's arrival

Let's take the opportunity to address a crucial point in internet resources access and talk about the reign of the connected mode. By connected mode we are not talking about hyperconnectivity addiction, but simply about TCP.

TCP has been dominating for a long time thanks to its control mechanisms, and UDP is generally restricted to real time where retransmission is useless such as voice or online gaming. However, connectivity is always more reliable and integrity checks are done in the higher layers for an increasing number of exchanges.

Thus, if we go up in the layers of the OSI model, we find probably the biggest client of TCP, HTTP. If HTTP/1.1 has been set in stone since 1997, 20 years later HTTP/2 brought prioritization, parallelization, compression, and predictive caching. HTTP/3 (RFC 9114) brings a schism by separating from TCP to base itself on a new transport protocol, QUIC.

While wrapped in UDP to ease its deployment, QUIC is a full-fledged transport protocol that that seeks to unify the best of both worlds by offering mechanisms that considerably reduce the number of client/server exchanges, in addition on forming a symbiosis with TLS which is now directly embedded. It therefore aims to offer secure, parallelizable connections, while reducing the number of round trips.

Some vendors are already pushing UDP in enterprises, especially for communication solutions. These providers sometimes even ask their customers to announce their public IPv4 from the conference service on their internal backbone so as not to have to alter the content of the SIP message in the upper layers, to offer UDP support and to dispense with any intermediate processing. How many people have noticed during the lockdown that these solutions work better at home on their own desk or on their business workstation when they provide split-tunneling VPN?

What if tomorrow these Cloud service providers push QUIC and therefore UDP for other services? What to do?

And HTTP/3 is not the only one moving towards QUIC, the widely used network sharing protocol SMB is making the jump, with Microsoft working on implementing it in Azure Files and Windows Server.

Have a look at your flow monitoring to see what the cumulative proportion of HTTP(s) and SMB on your network is, a hint, it’s most probably high…

At the moment, firewall vendors recommend disabling QUIC, until its support is properly implemented. It will also be necessary for the equipment deciphering the traffic to adapt, as they are tuned for the TCP+TLS couple.

The redesign of egress paths to the Internet is an opportunity to deploy IPv6, which would limit any packet modification steps to proxies only.

NAT+PAT of many QUIC streams is a challenge, if the device vendor introduces Application Layer Gateways to apply specific processing to QUIC sessions, it risks compromising some of its security. The draft RFC draft-duke-quic-natsupp-01 recommends that no optimization should be attempted on NAT.

Again, an IPv6 session eliminates these issues. Is this trivial? Consider the problems you may have personally experienced on your home network with NAT and UDP for dynamic needs such as multi-player gaming, P2P, or VoIP in their earliest days. One solution is to stay with HTTP/2 over TCP, but for how long? A transitional operation could be to allow QUIC without deep packet inspection only for trusted SaaS offerings at first. And let’s not forget that QUIC can carry many other things than http.

Note that these elements are valid for access to your resources by others as well, or by your remote workers. Therefore, the way of the so-called "zero trust" solutions leads to the removal of VPNs and a more direct exposure of resources, which will also shift to QUIC.

This protocol has just been ratified in RFC 8999, 9000, 9001 and 9002.

Note about proxy: in order to benefit from its contributions, the proxyfication layer must be upgraded, both on the browser and proxy side. 2 modes exist, a tunnel mode, the most efficient and the only one able to support the initial exchange of a QUIC session (with long header). And a forward mode where the proxy keeps a protocol rupture role, but only once the session is established.

This transport protocol is expected to have a faster deployment curve than IPv6, the efforts made to support it on its proxy chain or on its web front-ends in the other direction are an opportunity to work on rolling out v6 in parallel.

• INTERNAL NETWORK

Beyond the edge of the infrastructure in contact with the Internet, what are the motivations for deployment on the internal network?

Continuing from the previous sections, end-to-end is clearly an advantage in an era of increasing outsourcing of Cloud resources. Again, vendors of certain products will likely encourage solutions that limit intermediate processing on packets. Note that the IPv6 header structure offers some delays savings by removing the checksum, using fixed size fields, and include flowlabel to offer an easier tracking of flows during QoS processing.

For large structures, IPv6 also means the removal of the problems caused by the small size of IPv4 private addressing.

RFC 1918 offers 17 891 328 IPv4, that is only 70 000 networks in /24. Many organizations have already reached the stock limit, for multiple reasons. Allocation by entity, waste and over-allocation, non-retrieval of addresses when decommissioning devices or sites, desire to aggregate routes dating back to a time when routers only supported a small number of routes, transmission to subsidiaries that have been resold but have remaining links, etc.

While NAT44 may uncomfortably accommodate connections to partners and newly acquired entities, it is often unthinkable to split one’s business into overlapping scopes; although this is also a possibility.

Others take the appropriation way and exploit on their internal network IPs belonging to others with more or less tact. There are 2 groups:

- The cautious, who deploy double NAT44 and create a real hatch partitioning routing at the edge of the Internet. Traffic is natted twice and can easily have the same source and destination IP, with the NAT masquerading another NAT, the screen is total; These cautious people find themselves at a loss when a Cloud provider recommends that they advertise the public IP of a service on their internal backbone. What if this real public IP overlaps with a spoofed LAN IP? Especially since a provider can introduce new IPs with only a few weeks’ notice. SF scenario? Not at all! A perfect example is the use of Microsoft’s communication solution, TEAMS. The editor recommends announcing their public IPs, for reasons explained earlier in this document.
- The confident ones, which leverage IPs that will never be advertised on the Internet like those

> of the US Department of Defense (DoD): 6.0.0.0/8 7.0.0.0/8 11.0.0.0/8 21.0.0.0/8 22.0.0.0/8 26.0.0.0/8 28.0.0.0/8 29.0.0.0/8 30.0.0.0/8 33.0.0.0/8 55.0.0.0/8 214.0.0.0/8 215.0.0.0/8

Well, that's purely theoretical, as the end of 2019 Section 1088 of the DoD budget bill projected that these ranges would be sold within 10 years. (See appendix) However, the bill did not pass the Senate. But what about the future?

Should these addresses end up for sale, no doubt some would end up in the hands of major Cloud providers.

Very shortly after Joe Biden's investiture, AS 8003 began advertising DoD IPs via Hurriccane Electric. Officially, the following was reported to the Washington Post:

Defense Digital Service (DDS) authorized a pilot effort advertising DoD Internet Protocol (IP) space using Border Gateway Protocol (BGP). This pilot will assess, evaluate and prevent unauthorized use of DoD IP address space. Additionally, this pilot may identify potential vulnerabilities. This is one of DoD's many efforts focused on continually improving our cyber posture and defense in response to advanced persistent threats. We are partnering throughout DoD to ensure potential vulnerabilities are mitigated.

Some people talk about traffic gathering for analysis (a honey pot), while the DoD highlights the fight against cybersquatting of its IP ranges. But what if it was simply a matter of testing the implementation of the "prudent" scenario above within the DoD itself? And to simulate that the sale and advertisement of these countless IPs would not cause any ripple effects before actually releasing them for sale?

In juin 2021, the DoD announced that all new services deployed after milestones dates should be in IPv6.

On september 7th 2021, the vaste majority of prefixes migrated to AS749, owned by the DoD, but different from its usual preduction AS (721).

If you are getting close to the end of RFC 1918, you can investigate the use of the RFC 6598 100.64/10 range reserved for carrier NAT44 as a way to share IPv4 between subscribers with a Carrier Grade NAT. However, it is recommended not to assign these addresses to carrier devices such as MPLS routers or to use them on Cloud infrastructures, except after receiving the provider's approval. On the other hand, there is no problem to use this range for user campuses, for example. Some companies are already doing it.

Beware, the 100.64/10 range is de facto used by some overlay systems such as the Zscaler Cloud proxy solution to build its tunnels.

Lastly, if you are a gambler, you can try to use the former class E (240/4). This class is located at the frontiers of IPv4, after the multicast section. Reserved for a future use that will never come and unusable by vendors who recognize that the work needed to standardize this range would take longer to reach all the deployed fleets than to migrate to IPv6. In real life, don't try it, except in the lab out of pure curiosity. Google' GCP allows to use it on VPC but mention possible OS problems: https://cloud.google.com/vpc/docs/vpc#valid-ranges . Nevertheless they don't specify you might even be unable to learn such prefixes on your on premises BGP routers, although at least 2 vendors support this space via a command.

The use of one of the "cheat" scenarios described above to extend private addressing or the short horizon of reaching the end of the RFC 1918 pool appearing to be near (less than a few years at your consumption rate) should prompt you to give serious consideration to an IPv6 deployment.

Remember the time spent on past and future NAT44 and re-addressing projects related to the incorporation of newly acquired entities. Have you ever seen an IT department decide that they would start their internal addressing with the 10.255.0.0/16 block in the downstream direction because their company would be acquired one day and hopefully the new parent entity would have started their addressing with 10.0.0.0? More seriously, IP addressing conflicts during structure integration generate costs and delays that are often significant, in addition to added complexity for long-term operation in the event that NAT44 remains in place.

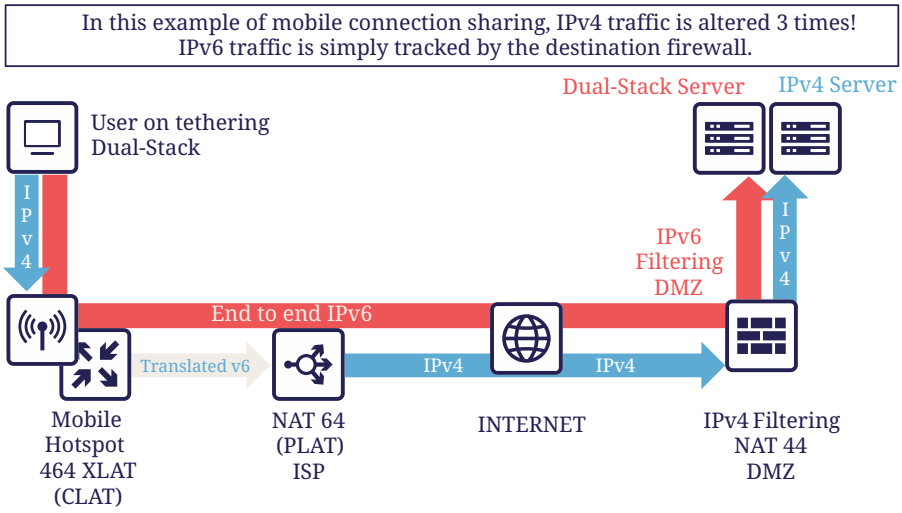


Figure 6. Mobile Connection Sharing

Transition Technologies

Transition Technologies

The Internet Engineering Task Force (IETF) is working with vendors and software editors to deliver mechanisms for handling IPv6 transition. These mechanisms are intended for carriers and/or enterprises.

First, we must divide the mechanisms into 2 usage groups:

- Those allowing to transport a protocol on a transport network of another version. They are typically used to connect IPv6 islands via IPv4 transport networks (or the opposite). There are systems based on encapsulation and others based on translation.
- Those allowing to make IPv4 host communicate with IPv6 host or the opposite. They are necessarily based on translation.

Both are complementary in most usecase.

Dual-Stack

Dual-stack is simply the simultaneous use of IPv4 and IPv6, therefore it constitutes a simple cohabitation.

However, running IPv4 and IPv6 in parallel causes overhead, not only for the network with dual routing configuration aspects or firewall rules. For example, all system deployment processes must run in parallel in v4 and v6. Quality of service monitoring requires each service to be monitored in both IPv4 and IPv6, etc.

Dual-stack is therefore rarely viable in the long term in a large organization outside the networks carrying client terminals, thus typically campuses.

During the transition phase, it enables not affecting the IPv4 legacy and therefore not causing any disruption or regression on services.

Transport mechanisms

In the early days, mechanisms were designed to reach IPv6 islands from other islands, or simply from a workstation.

These methods include ISATAP (RFC 5214), Teredo (Microsoft - RFC 4380), 6over4 (RFC 2529), 6to4 (RFC 3056), 6in4 + TB/TSB (RFC 5572), 6rd (RFC 5969), IPv6 GRE (RFC 2473 / 2784), and many more…

Most of those based on tunnels use IP protocol 41 and/or UDP encapsulation as would do a VPN.

These mechanisms are useful for structures whose migration of some or all parts of the network is impossible for technical reasons. Carriers have been using them to overcome protocols that do not support IPv6, such as cable operators before DOCSIS 3.1.

As with any tunneling technique, the main drawback is the lack of traffic visibility due to its encapsulation. In organizations, filtering requirements and quality of service management make it difficult to use tunnels because of the complexity and the small number of compatible solutions, especially firewalls.

Most of these methods bring security risks to the organization and are carrier-oriented.

Transport networks often exploit a layering of technologies on top of each other, as is the case with MPLS or VxLAN. Deploying dual-stack across all transport layers is rarely relevant. However, it is important to implement it in the highest transport layer, the one visible to the network users, the overlay.

• INTEGRATION ON AN EXISTING IPV4 UNDERLAY

The existing transport solution often allows isolation of end-to-end customer contexts within the organization, both on the backbone and in the datacenter.

While *VRFs lite* are still in predominant in campuses, the other environments massively use underlay-based technologies. It is then relatively easy to transit IPv6 in the overlay.

MPLS

MPLS is a key component often present in companies, either directly or in an outsourced way via the site interconnection offers of professional operators.

MPLS allows IPv6 to be transmitted via 2 approaches:

- 6PE (Provider Edge RFC 4798) which provides v6 in the native table (GRT) of the devices, useful only if one provides services via the GRT (like internet access or TV for a customer ISP);
- 6VPE (RFC 4659), V makes all the difference, here we simply transit VPNv6 alongside VPNv4, it is therefore the equivalent of an L3VPN, the simplest method that meets most use cases.

It is indeed possible to use an IPv6 IGP and LDPv6 to build an IPv6 underlay based MPLS, but there is little benefit to swap outside of an opportunity offered by another large project. And especially that does not provide v6 in the L3VPN of the overlay which is the main topic of giving v6 access to

users.

The 6VPE implementation is the way to go, it will be easy to deploy on current devices and will require little configuration.

If your MPLS uses the recent MP-BGP EVPN as a control plane instead of MP-BGP L3VPN, IPv6 support won't be a problem there either.

Note that you can have an IPv6 next hop for IPv4 VPN routes thanks to RFC 8950 if you go for an IPv6 underlay.

VXLAN

Mostly used in conjunction with EVPN, VxLAN solves the pitfalls of older DataCenter L2 *SPB fabrics* and has become the industry standard. More rarely it is found on backbones that have abandoned MPLS to take advantage of EVPN which was available as a *control plane* for VxLAN prior to MPLS.

Like MPLS, VxLAN encapsulates. The question of IPv6 compatibility therefore arises in the overlay that is intended to provide customer service. Configuration of an IPv6 overlay is mature with the major vendors, still, check for complete multicast's mechanisms support (PIM snooping, BiDir, etc.)

While the underlay can remain in IPv4, note that the IETF is working on the implementation of RIFT (Routing in Fat Tree), to facilitate the deployment of *CLOS fabrics* in the vein of zero touch provisioning. Targeting fabrics with iBGP underlay, it plans that loopback addresses and *route reflectors* should be in IPv6. It is difficult to say if it will be completed before the fabrics migrate to SRv6 (RIFT also provides a mechanism for exchanging Node-SIDs and SRGB global segment routing prefixes in order to facilitate the deployment). See https://datatracker.ietf.org/wg/rift/documents/.

SD-WAN

SD-WAN products generally work with DPI and flow classification *ingress* to apply QoS and possibly choose a transit path (internet/MPLS/etc.) Traffic is then often encrypted in an IPSEC tunnel specific to the client context and encapsulated to the destination router (except when an analysis requires its decapsulation on the hub for example).

The underlay is designed to leverage an existing IPv4-based network in order to limit the preparations for the implementation of this type of product.

These products mainly target large networks consisting of small and medium-sized sites with dedicated device line and/or integration with more traditional hardware lines. On the Datacenter concentrator side, we find large chassis, again from dedicated or conventional product lines.

When one wants to use some of the major market's solutions on campuses with more than 2000 users, limits of dedicated products are often reached, although manufacturers are progressing and trying to cover the last percentile of missing usages.

The fact remains that IPv6 is rarely required by customers since these solutions are intended for their internal network. As a result, the compatibility of SD-WAN solutions on the market varies greatly from one vendor to another and among different releases. It is therefore important to follow the vendor's roadmap and test the solution before a v6 deployment, but also at each new

major release, as the code can be heavily modified given the speed of evolution of these solutions and the competition.

Finally, the Local Breakout aspect of these solutions is another element also gradually integrating IPv6. Often with a whole layer of local security services commonly referred to as "SASE".

• SPECIFIC ENCAPSULATION

It is not always possible to transit IPv6 on a transport perimeter, and as seen previously, few technical solutions are exploitable on both sides on enterprise hardware series.

This leaves the possibility of tunneling IPv6 traffic. This can be done via well-known solutions such as GRE/mGRE or IPsec (the latter is however less efficient due to the encryption resources required).

Finally, you can configure 6in4 on a large portion of the routers on the market if no other solution mentioned above satisfies you. 6rd is also often available but mainly targets north/south topologies.

We do not recommend looking at 6to4 (non-configurable endpoint), 6over4 (IPv4 multicast based), ISATAP (DNS discovery based) and Teredo (UDP encapsulation) which are now very rarely used.

The availability of a given method on your devices, in conjunction with the integration with your routing, will determine your choice.

• AND IN THE OPPOSITE WAY?

As discussed at the beginning of this chapter, there are also transition technologies that enable you to dispense of IPv4 on your backbone. It is then limited to user networks, IPv4aaS

Some operators are already moving away from IPv4 on their backbone, to save addresses and even share IPs between subscribers by splitting ports. The so-called Address+Port (AP) approaches have become widespread. First DS-Lite, then *Lightweight 4over6* (lw4o6) and more recently MAP T/E and 4rd. The last two prevail in today’s deployments, thanks to their aggregation capacity, which avoids having to terminate an astronomical number of tunnels and as many routes within the ISP’s core..

Those who have not yet transitioned to an IPv6 backbone and lack of available IPv4 addresses do simple NAT44 on a CGN core platform and use the famous 100.64/10 scope of RFC 6598.

Those in IPv6 typically provide IPv4 via one of the following methods:

- 4rd (RFC 7600) which works in the opposite way than 6rd and provides an efficient stateless method. It can work in mesh or hub&spoke mode;
- MAP (T or E) (RFC 7599), available in translation and encapsulation modes, is also stateless;
- Older deployments use DS-Lite and Lw4o6.

The first two are quite similar and use common rules on a domain, edge routers (BR), EA bits to define IP sharing level, announcement of mapping rules via DHCP to end devices (CPE).

The implementation of these techniques on the client router side is done in software, they can be

found in our home routers. However, it is unlikely to find a device that can handle MAP or 4rd via its ASIC on the client side, as high-end devices only deal with the Border Router aspect.

Concerning MPLS and VxLAN, it is possible to replace IPv4 by IPv6 on the transport underlay, you should consider it on greenfield deployment and start to think about transtionning your underlay after having checked with your vendor(s).

For the particular situations where the transport cannot transit IPv4, we find the same thing as before. Specific tunnels to connect IPv4 islands together. We can thus implement GRE/mGRE, 4in6. 4rd does not seem to be very present in enterprise routers yet.

You can often easily transport IPv6 on an IPv4 underlay and might want to wait for a large backbone project, renewal,… to swap your underlay. If you're working on a greenfield deployment, consider an IPv6 underlay. Moreover, care about designing your topology and addressing plan to be ready for an SRv6 deployment. It will save you time later, if you don't directly start with it.

Translation mechanisms

The purpose of translation is to allow exchanges between clients and servers using different versions of IP.

If we stick to the dual-stack logic, we must deploy IPv6 everywhere. But this leads to a lot of duplicate operations and only works if all elements are dual-stack compatible. How to make IPv6 clients talk to IPv4 servers? (or in the opposite direction)

NAT64 and DNS64 provide a joint solution that is already widely deployed and allows IPv6 clients to contact IPv4 servers. Inversely, SIIT (Stateless IP/ICMP Translation Algorithm) lets IPv4 clients enter an IPv6-only network.

Obviously, since the IPv6 header is longer, it is technically simpler to keep the header information when sending IPv4 clients to an IPv6 server than the opposite. But the direction of deployment is a matter of need, strategy, scheduling and consistency.

• NAT64 + DNS64

NAT64 (RFC 6146) coupled with DNS64 (RFC 6147) uses the principle of "lying" DNS in concert with a translator to allow IPv6 terminals to access IPv4 resources. IETF publishes a deployment guide (RFC 7269).

When a resource does not have a DNS AAAA record, the DNS server will synthesize one from an IPv6 /96 prefix and the IPv4 /32 address returned in the DNS A record.

The terminal will then initiate a connection to an IPv6.

Somewhere on the network, (we will see locations later), a device advertising the /96 prefix will receive the connection. This NAT64 platform will remove the IPv6/96 prefix from the destination and replace the IPv6 header with an IPv4. In doing so it NATs the packet and picks a source address from its NAT pool (along with a source port for the PAT) and sends the packet. By maintaining a session table it will perform the reverse operation on the returning packet.

Note that the endpoint is at no time aware of the trickery. This results in problems on P2P protocols as well as those embedding the address in the payload like SIP, H323, IPSEC AH, SCCP, NFS older version 4 etc. features can be implemented as ALG on NAT64 platforms to solve the problem, but potentially at the cost of performance degradation.

DNSSEC validation by the host will also be prevented by this scenario. This problem could be solved if the host was aware of NAT64 (which is the case on mobile with APN configuration or when RFC 7050 is used, but the latter is not very useful with desktop OSes since they don’t support it yet. There is also a desire to be able to notify hosts via DHCPv6 and PCP of the NAT64 prefix.)

On the application side, NAT64 works as long as it can open IPv6 sockets and that it calls a hostname and not a literal IP.

Addressing

On a small network a single platform will be sufficient, it will generally use the WKP prefix (RFC 6052 Well Known Prefix) or another prefix called (Network Specific Prefix) defined within the addressing of the company with a /96.

Be aware that if you use an ULA prefix, NAT64 will always be deprioritized in comparison to IPv4.

Don't forget in your project that if 99% of the connections are initiated by client endpoint, there are special cases such as remote control by the support. And of course, P2P telephony. Those will require full IPv6 compatibility.

On a large network it is preferable to have several platforms, each with its own prefix. A range is reserved for this purpose, although not mandatory: The 64:ff9b:1::/48 (RFC 8215).

Topology

The placement of these platforms will vary according to your constraints.

Setting them up directly on the sites will avoid tromboning in the datacenter (round trip). But this will require the use of as many NSP prefixes as you have sites, in addition to adapting the DNS64 configuration each time. Through a DNS proxy configured accordingly on each site. (It can be *Bind9, Unbound* or other solution.)

It is also possible to use the same prefix on each site as long as they are dead ends and the route advertisements to the backbone filter the NSP. This makes DNS64 configuration easier.

Putting NAT64 on the sites involves in any case maintaining an IPv4 transit in backbone. Note that it will be difficult to get rid of this quickly anyway, as sites rarely contain only user stations. Creating NAT sessions on X sites also implies collecting session creation logs on all sites. Finally, it will be necessary to provision numerous NAT IPv4 pools and adapt the filtering ACLs.

On the other hand, centralizing it makes it easier to implement on all levels, but is not desirable if it generates tromboning on flows that could have remained internal to the sites.

A good trade-off is to have NAT64 gateways on the largest sites, especially those that host services locally and need these services to work even during a WAN outage. For others, centralize it in the datacenter or at the backbone edge.

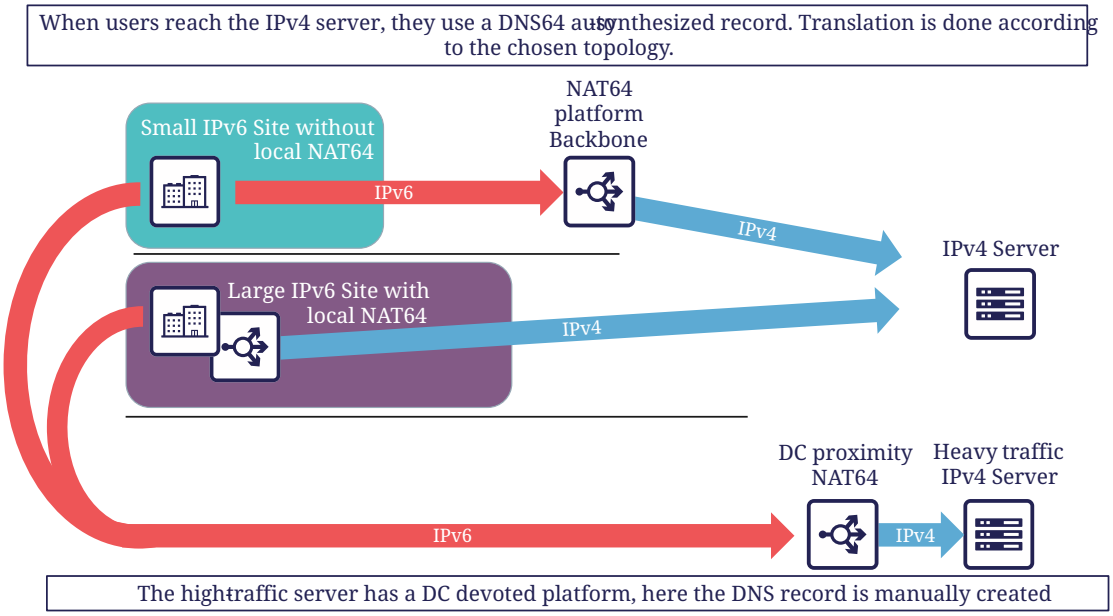


Figure 7. Using NAT64

MTU matters

IPv6 header is 20 bytes longer than IPv4, meaning that a large IPv4 packet returning to NAT64 platform might be dropped if the platform doesn't handle fragmentation properly. As fragmentation can occurs only on IPv4 side, before returning translation, you will often need to adjust a specific NAT64 MTU setting that doesn't change any real interface MTU, but just packets internal handling.

Platform might also send back an ICMP reply "Fragmentation Needed" to IPv4 server.

You may have to use the 2nd option for some traffics, and obviously for those who don't support IPv4 fragmentation, such as TFTP. See RFC 7915.

In the opposite direction, you must be sure that PMTU-D send at least a length of 1280 bytes, so always set the IPv4 side interface of your NAT64 with a MTU greater than 1260 (1260 + 20 overhead IPv6 = 1280) Without this, attackers might use your platform to perform unwanted fragmentation. See RFC 7269.

Note about filtering

Once NAT64 is crossed, how to filter the flows? If NAT64 is done close to the user, identifying a population is simple, if it is centralized it requires a lot of fine-grained ACLs in one location.

The solution lies in segmenting IPv4 NAT pools, create matching rules so that machines behind an IPv6 prefix X emerge with a dedicated IPv4 NAT pool Y, and so on. Again, the more segmentation there is, the more complex it will be to enforce on sites.

Which scope and therefore which technology?

Now that you know which technology allows a client to interact with a server that does not speak the same language and how to deal with transport, let’s see the relevance of each solution.

An ideal approach is to ask yourself what is the easiest to migrate.

What types of endpoints are present on the network?

• CAMPUS

On the user side, we generally find homogeneous workstations, with an identical ecosystem reproduced per site/geographic zone and other more centralized bricks. This ecosystem includes file storage, authentication directory, messaging and other collaborative tools such as telephony, printing, proxy, workstation management agent, protection agent, and of course business applications. The latter are now almost systematically web applications and therefore often rely on the browser on the client side.

Network equipment also often follows repeated architecture patterns, with 2 to 3 generations coexisting at the organization level. Unfortunately, campus equipment is the one that is most behind when it comes to IPv6 compatibility, especially in terms of security features.

However, it is difficult not to realize that if this perimeter is wide, it is also relatively homogeneous. This homogeneity is a strength. By deploying IPv6 in dual stack on a site of each type in pilot mode, and by implementing it on the elements of the "office/workplace" ecosystem, it becomes possible to industrialize rollout.

This can occur when replacing site equipment, moving, etc.

Eventually, it is even possible to withdraw IPv4 from the campuses in order to get rid of the dual-stack management. This is the preferred scenario if your organization lacks private IPv4 address space.

NAT64 + DNS64

If this path fits your needs, you will have to study the NAT64 and DNS64 placement. We repeat the elements of the topology section:

If your sites do not have any IPv4 compatible services and/or only rely on datacenter or Cloud servers, there is no need to have NAT64 on site, this is typically the case for banking agencies for example.

On the other hand, a large industrial site will often have on-site business servers, so that production does not depend entirely on the WAN’s reliability. And some of these systems will only work in IPv4. It is therefore necessary to be able to exchange locally in IPv4.

If few clients need to run the affected applications and if they are limited to specific networks, it

seems appropriate to preserve dual-stack. This can be done physically or logically, using a radius server for example.

On the other hand, if many workstations need to be able to reach a local IPv4 resource, the implementation of a local NAT64+DNS64 becomes interesting, and is even recommended if you encounter a lack of private IPv4.

This NAT64 will be deployed in stateful mode (with session tables and port assignment).

Although it is possible to take out IPv4 with NAT64 each time a site is migrated, one component is problematic: telephony. Indeed, while the vast majority of flows are sent to a server, telephony has the particularity of generating direct P2P UDP traffic between 2 users. Unless your equipment manufacturer offers a mechanism to segregate the IPv4 and IPv6 population automatically in order to force translation via a dual-stack media relay server when a call is established between the two domains, you will need to deploy IPv6 on all campuses before starting to remove IPv4 from some of the terminals, including those in remote access (VPN or other).

Don’t forget that some services may have to initiate an IPv6 session to a workstation, for example the helpdesk to connect to a workstation and troubleshoot. The helpdesk will therefore also need IPv6 connectivity. And if this helpdesk is outsourced, you will have to review your contracts.

This constraint linked to SIP and RTP traffic forces a global response before IPv4 is cleared.

• DATACENTER

Datacenter resources, whether on premises or Cloud-based, can be very diverse or relatively homogeneous. It all depends on your business and your history.

While GAFAM, Google-Amazon-Facebook-Apple-Microsoft, have published IPv6 transition resources, they are rarely practical in a large enterprise. To understand this, you only need to look at services in terms of volume and deployment scale. When you run fifty or so services on hundreds of thousands of servers, you are bound to be industrialized, with an orchestrator that calls for automation. It is then feasible to carry out a pilot migration to IPv6, service by service, and then to generalize. A similar approach to the one mentioned above for campuses, many machines but with a similar configuration. In an arbitrary way, let’s say a major actor of the Web has a ratio of 100 000 machines per service, what is the ratio of a company?

List your servers, VMs, containers, and divide by the number of applications your IT has. The result is likely to be between 3 and 10. Not really something to call scalable, though. But don’t get discouraged, these servers often run a much more limited number of middlewares, about ten. Their IPv6 compatibility is good, but you still have to check that each application works properly. The "applications" section will help you.

Dual-stack servers and applications

As explained in the dual-stack section of the transition technologies, keeping everything in double in the long-term leads to various additional costs. It is ideal to provide IPv6 connectivity on your server masters in order to be ready for any scenario on the system side. These aspects are discussed later in the document. Dual-stack remains recommended for critical and heavy load services (DNS,

directory, proxy, NAS, etc.)

6/4 translation

Application lifecycles can be as long as 2, 3 years or even more. It's hard to wait that long to offer access to them to clients who don't have native IPv4.

If your application is exposed on the Internet, you can just let the NAT64 do its job on the operator's side for the clients who don't have native IPv4 anymore, smartphone in most cases. However, this makes troubleshooting more complex on your side since you don't have the control, and the service is provided with a carrier dependent performance level. If latency or session drops occur, the user will blame you and your reputation will be affected. He has no idea of the intermediate treatment of his carrier.

You have 2 possibilities to expose in IPv6, NAT64 or a reverse proxy.

In order to limit the workload, you can rely on existing device to make things easier. If your presentation server is simply located behind a firewall with no other intermediary, then static NAT64 seems like a good idea. You would then match a NAT IPv6 to each server IPv4 statically, and publish the corresponding AAAA DNS record. You can even match IPv6 prefixes in /120 with IPv4 /24 networks for example, which involves even less rules. The firewall will perform NAT+PAT and track sessions.

IPv4 servers will need to track the session port in addition to the IP so they can correlate the firewall logs (see RFC 7768).

For less popular servers, classic NAT64 stateful will suffice. Always remember that it requires the implementation of DNS64 on the resolve path and the choice of a Network Specific Prefix in /96 that you will expose on the internet. Same thing for internal network.

Hybridization is a good option, static NAT64 with manually created AAAA for each heavily used front-end server, and dynamic NAT64 for everything else.

This NAT64 processing is done at low level, with high performance on recent hardware. On the other hand, it requires synchronization of the session tables to guarantee the high availability of the stateful mode. This mode is not suitable for anycast servers since there is a chance, albeit small, that the client will switch from one NAT64 platform to another during session lifetime. A break would then occur (See SIIT below).

For fine grained traffic needs, for example due to the fact that the IPv4 server to be reached internally is located in a different datacenter than the NAT64 ingress platform, you can use dedicated IPv4 SNAT pools in order to respect fine grained filtering principles (Similar to the ACL issue discussed above).

With an SLB (load balancer) at layer 4, NAT64 is also recommended, but if it works at higher layers (L7 with or without a WAF application firewall, HTTP for example) then the protocol rupture will lead to traffic reconstruction in the other version of the protocol and the question then no longer arises. Nevertheless, it is often useful to copy the client's IPv6 address into an "X-Forwarded-For" HTTP field when the latter is used. This allows the client's visibility to be traced back to the server.

Since the public entrance to the datacenter is usually made up of several of these components, remember to bring IPv6 to at least the devices with fine-grained rules.

Consider the example of Internet traffic passing through an L4 firewall and then a reverse http proxy application firewall (WAF) before reaching the server. We would be tempted to get rid of IPv6 at the network firewall and use NAT64. Consequently, certain detection of reverse proxy rules would no longer work since it would always see only the same pool of SNAT IPs from the network firewall and not the clients' IPs.

For internal access to an IPv6-incompatible application, NAT64 or reverse proxy methods can also be used. Finally, for an internal application that still does not work with these approaches, it is still possible to use an internal VPN to reach the IPv4 island from an IPv6 station. Moving all the affected customers to a VDI in a datacenter is another viable but expensive alternative.

Native IPv6 Deployment

Given the increasing proportion of IPv6 clients, why not consider providing its internet-facing services natively in IPv6 and implement a translation for IPv4 clients?

This is the principle of Stateless IP/ICMP Translation (SIIT), in its original version it is limited to a 1 to 1 two-way translation between IPv4 and IPv6. This obviously requires as much IPv4 as IPv6 on both sides and is therefore only usable on very small and specific perimeters due to the limits it imposes. For example, for some servers between them.

In its DC flavor, SIIT-DC allows for access to IPv6 servers from IPv4 clients, without maintaining a stateful table.

For this purpose, an IPv6/96 prefix will be reserved to map the IPv4 in the last 32 bits. Thus the system can be multiplied without constraint and support anycast and dissymmetry (since it does not rely on stateful). By default the prefix will be in the range 64:ff9b:1::/48 (RFC 8215).

It is of course possible to use several prefixes, for example to link the mapped packets to the IPv4 internet entry where they landed. Very useful when the internet chain has stateful controls on its side (IPS, etc.)

One must however always provide as much IPv4 as there are IPv6 servers to expose.

And when there is still a need for IPv4 on a server somewhere deep in DC, it is possible to use SIIT (Dual Translation). IPv4 internet traffic is translated to IPv6, passes through the datacenter, and is then re-translated by a device closer to the server.

Although we are talking about the Internet here, the same topology can be implemented for internal IPv4 clients.

Double mono-stack

A rarely employed but viable method on huge clusters is to deploy servers exposing their services only in IPv6 in parallel with other existing IPv4 servers. If this technique does not go in the direction of homogenizing the configuration, it has the advantage of not touching the existing. Thus IPv4 clients in production have no risk of disruption or regression.

Cloud Providers

While IPv6 is seamlessly provided in market leaders' IaaS offers, there is still a long way to go for PaaS solutions.

For example, most load balancing services are not yet compatible, and when they are (like AWS NLB since the end of 2020), it is only on the customer facing part, and not yet on the backend part. (Which is, admittedly, less urgent).

External translation

For internet facing services, you may also rely on CDN and other intermediates services which have the ability to expose in dual-stack while the backend is only in one of the IP protocol versions.

• WAN

WAN itself doesn’t directly offer services to users, it’s there to carry traffic across sites. You may come back to transport mechanisms section to see how to transport IPv6 traffic.

Regional NAT platform

Depending on the size of your sites, and your typical flowing map, you may consider setting up regional NAT64 platforms on your backbone. Remember this adds a stateful service on it, therefore enforcing the need of symmetrical flows.

Such service can also be provided by your usual carrier on your managed MPLS as long as you don’t cipher traffic between sites on your side.

Block sequencing

Block sequencing

IPv6 deployment is logically to be done starting with the bottom, the network layer. And before any deployment it is coherent to prototype the behavior of each component. Very few organizations have an end-to-end lab and qualification environment, both horizontally within the same layer and vertically between layers. For example, your campus, datacenter and security network prototypes may be managed by different teams and are not interconnected in a topology close to the production one, this is a horizontal break. If a qualification server is running on a network with production routers, we have a vertical break. This makes sense, otherwise how can you debug a problem if all the stack layers are in test, it would look like a multi-level rola-bola.

We will retain that each layer has its own test environments, and that those run on lower layers' production environments. In short, every qualification runs itself on an underlying production environment (except for the foundation that is network). This can be represented as follows:

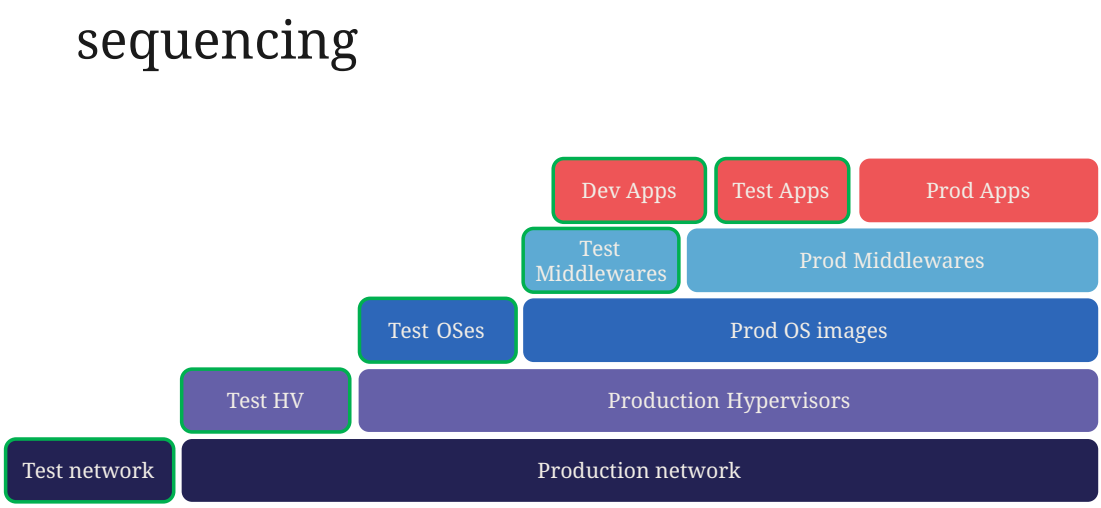


Figure 8. Sequences

Warm up

Before you even decide where to begin, start by making sure that all of your current and future specifications / RFPs / subcontractors requests include IPv6 compatibility and guarantee its proper functioning. These processes often take a long time to change, so it is advisable to start working on them right away.

This also includes build, run, life cycle processes and everything related to them.

Network

Network is the first pillar to deal with, let's start with some questions to consider about the devices:

- Does an appliance work in IPv6? (Ask the equipment manufacturer, integrator, test, etc.);
- Is there a gap in functionality / regressions with IPv6 compared to IPv4? (e.g.: does a probe have the same detection capabilities, the same rule and signature running policies);
- Is there a performance gap with IPv4 (e.g.: is the number of packets filtered per second on a firewall of the same order of magnitude? are the ASIC hardware features equivalent? such as IPSEC or TCP offloading on the OS VM/Hypervisor/Driver NIC chain or TLS on a load balancer);
- Is the device administration possible via IPv6? Its control, its supervision, etc. or does it only do IPv6 in transit data plane and not on admin side.

Implementation difficulty increases as the device's ability to climb the layers of the OSI model increases, because more and more functions must be tested, and the risk of omission/configuration heterogeneity also increases.

It is therefore easy to deal with routers, once the routing protocols have been mastered, while avoiding immediately providing dual stack to end-user networks to allow time for validating security mechanisms specific to v6 in host networks.

Then the infrastructure networks without reaching the user.

Then we move on to the filtering and flow optimization devices, where the object-based model allows most policies/ACL to be dual-stacked by acting only on the objects to reflect the v6 subnets/address in correlation with v4.

User access can only be active after the security component is rolled out to both network devices and hosts.

The rest requires more work, and concerns the advanced network services present in datacenter like load balancers, waf, probes, etc.

• Readiness

The IPv6 compatibility maturity varies according to device types. In a general manner, carrier-grade routing appliances have been free of problems for years. On the contrary, campus devices still sometimes encounter a few bugs, especially regarding security features.

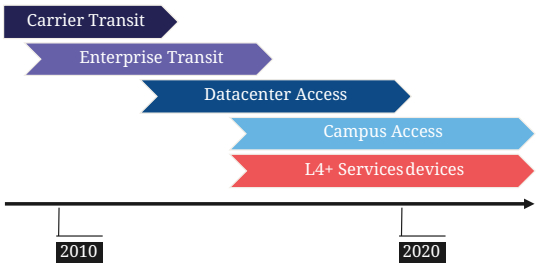


Figure 9. The IPv6 evolution

Solution maturity appears to follow the graph above, pay attention to SD-WAN and Campus SDN solutions, refer to the SD-WAN paragraph in the Transport Mechanisms section whose elements also apply to Campus SDN.

Tracking release notes and known bugs allows you to see when IPv6 support is maturing, focusing on v6 specific bugs. The evolution generally follows the probability density of a normal distribution and thus a Gaussian curve.

• HARDWARE

Consider checking your routers' ternary memory allocation, some configurations have little room for IPv6 routes to be stored. Some ASICs on the market store IPv6 /48 routes (and sometimes other frequent sizes) differently than other prefix sizes.

IPv6 full view is growing exponentially, so take some leeway when choosing devices handling public peering. If you are short on space but still need BGP full view, you can dedicate some routers to v6 peerings and others to v4, if the technical and economic study is satisfactory.

As addresses are longer, they take 4x more space in memory. Think about routing tables, ACL, stateful tables, logs. Hopefully they often consume only 2x more space as IPv4 as long as /64 are considered. That’s often the case for routing tables and routing decisions.

• LAB

Trying out features, from the simplest ones like routing to the most advanced ones like security mechanisms, can be done on a variety of environments. In stand-alone or not. Some tests, such as QoS validation, require a physical chassis and a traffic generator, whereas an ospfv3 test can, in all likelihood, be conducted on a virtual instance. The dependency on ASICs being limited.

One can imagine distributing the tests as follows, knowing that the tests can be shifted from the left columns to the right ones. However, this makes their execution more complex, to the point of

increasing the risk, the last column being in production tests.

Environnement min. Device	**Virtual Lab (vendor environment or eveNG like,…)**	**Independent physical lab**	**Pilot phase on production**
L2 Switch	• Configuration validation without real test • Some virtual L2 tests may not be very accurate depending on the vendor.	• Access security (eg: RA guard) • MLD snooping • 802.1x • QoS • ACL • Stack behavior	- In production host behavior
Wifi AP	N/A	• Previous elements (except stack…) • Controller reachability • Local routing outside tunnel • ACL	- In production host behavior
Router	• Protocols (OSPFv3, IS-IS, MP-BGP) • FHRP (HSRP, VRRP) • Multicast (PIM, MLD,…) • DHCP relay • ACL, route-map • Router / FW neighborship • DCI • PMTU Discovery	• Previous elements • Access security (RA guard, etc.) • QoS • BFD • ARP/ND inspect • Dual-Stack provisioning to access networks • Performance	• In production host behavior -Scaling
FW *(additionally to router features)*	• Previous elements • Editing objects/rules in v6 • NAT64 • v6 transit filter rules • L7 non-regression tests	• Previous elements • FW HA • transit v6 filtering rules • Vendor controller • IPsec • v6 logs + NAT64 logs	• ACL orchestration integration • Integration of v6 logs + NAT64 logs • In production host behavior
Load Balancer (SLB)	• Object/rule editing in v6 • L7 non-regression tests • NAT64	• TLS offloading • Performance • v6 logs	

Environnement min. Device	Virtual Lab (vendor environment or eveNG like,...)	Independent physical lab	Pilot phase on production
IPS/IDS	- Object/rule editing in v6	- Previous elements	- Prod SIEM processing
Wan Optimization	• Object/rule editing in v6 • L7 non-regression tests	- Previous elements	
Proxy	• Object/rule and PAC editing in v6 • Guest behavior	- Previous elements	
DNS IPAM DHCP	• DNS64 • AAAA records • Reverse PTR • IPAM v6 blocks • DHCPv6 with options	• Previous elements • Host self-registration • Service provided in IPv6	

In an effort to help you, RIPE has published RIPE-772 which is a list of compatibility points to check and ask for when issuing an RFP.

The US NIST published in 2020 the revision of its USGv6-rev1 test program.

• INTERNAL ROUTING

Depending on your network layout, the introduction of IPv6 will require deep changes in terms of routing protocols configuration.

BGP

Even if the address family v6 implementation in MP-BGP simplifies the work in BGP, it will be necessary to analyze the route classification rules of type access/prefix lists/sets so that the IPv6 addresses are taken into account to apply the route map/policy correctly and in a coherent way to IPv4. In order to limit inconsistencies, base your rules on communities when possible and mark these communities on the capillary networks rather than having to maintain lists of v4 and v6 prefixes everywhere. The rigor of a v4/v6 mapping table and automation is another valid strategy, either distributed on routers or centralized on a route server such as FreeRangeRouting, Bird, Quagga (probably facilitating other aspects of your routing engineering if you are the type of person who often tweaks BGP).

IGP

2 solutions are to be considered regarding IGP. Either use IS-IS from ISO which is IP agnostic, more flexible than OSPFv3 but rarely deployed in companies. It is the IGP that dominates today on large carrier networks, notably because of its convergence and its partial recalculation mechanism.

Moreover, the arrival of IPv6 SRv6-based Segment Routing requires IS-IS and its TLVs, even if OSPF LSAs have been created to offer an equivalence, the market and manufacturers seem to be primarily turning towards IS-IS (check with your vendors).

The other solution is to switch to OSPFv3 and, once it is stabilized, to include AddressFamilyIPv4 in order to remove OSPFv2, perimeter after perimeter if the devices are compatible with the provision of IPv4 routes in OSPFv3 RFC 5838.

Keeping the 2 versions of OSPF in parallel brings the classical problems of dual-stack (configuration homogeneity between v4 and v6, configuration overhead, monitoring equivalence, etc.).

For a large organization, IS-IS training is probably worth the cost, especially to prepare you for SRv6.

Don’t forget that only the IGPs that carry client networks are concerned, generally the capillary ones. It is useless to modify the underlay IGP of your MPLS or your VxLAN EVPN since BGP takes care of v6 in the overlay layer.

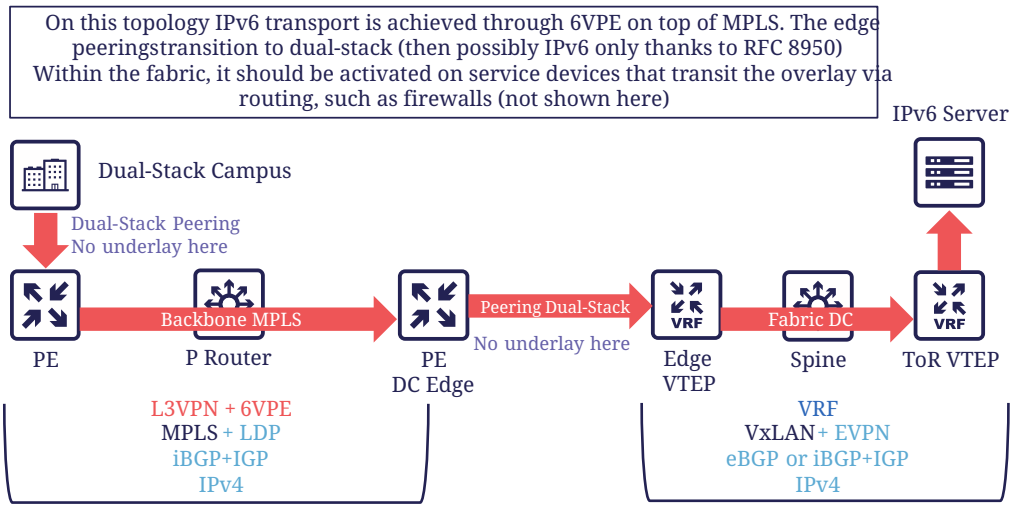


Figure 10. 6VPE Topology

• FILTERING AND TRACKING

Before transmitting flows, it will be necessary to reach the same level of security than in IPv4. The security section contains many elements on the topic. You will also find in the chapter " v4/ v6 mapping " of the addressing section some advices to facilitate the rules transcription.

Infrastructure services

Many critical services go hand in hand with the proper operation of the infrastructure. Some enable connectivity, others target security aspects, etc.

Regardless of which IPv6 deployment scenario you choose for your organization, the implementation schedule within the infrastructure services will be similar.

• SIEM

Each time a new service is migrated, logs must be collected and correlated as efficiently as with IPv4. Adapting your SIEM is therefore essential throughout the project, so in the long term, plan to have resources on the subject. The transcription of log parsing rules is rather time consuming. It would be a good idea for the main editors to offer turnkey conversion mechanisms.

Make sure that the log sources send the address between brackets followed by the port [IP]:port. Without brackets it is difficult to separate both, you can bet on the fact that the last group of numbers is the port, but some applications sometimes don’t send it when the source port is the same as the server socket and a simplification function is called when it shouldn’t be (rare case but not impossible).

Be careful with the storage of IPv6 addresses, see the applications section a few pages further.

• DNS/IPAM/DHCP

This set of services is often entrusted to the same application solution, except for specific DNS zones such as those assigned to a Microsoft Active Directory environment.

In any case, the production interfaces of such services accessible by clients are a priority to be switched to dual-stack.

The services that interact with the administration interface of the devices do not need to be provided immediately in IPv6. This is for example the case for NTP, RADIUS, TACACS, SYSLOG servers… which can wait. It is different if your scenario targets a v6 deployment on the administration networks.

• VPN, PROXY AND REVERSE PROXY

These services have the particularity of having both internal and external pointing interfaces. IPv6 provisioning can be implemented regardless of the 2 sides, since the use cases are different.

External

Probably the one to implement even if you are not aiming at an internal use of IPv6 at all, the possibility to communicate over the internet will allow your users and customers to reach you with a native IPv6 connectivity at a time when IPv4 sharing tinkering is becoming widespread. Conversely, it will allow proxy browsing to reach IPv6 sites without problems.

Thus, your VPN gateway and your reverse proxy should be exposed in dual-stack as soon as possible, avoiding you to have your flows crossing Carrier-Grade NAT and other joyful things without any possible control on your side. We remind you that the reverse proxy can also offer public IPv6 connectivity to IPv4 servers. This is another way to regain control over this translation on the Internet side.

Internal

The internal aspect goes alongside the deployment of IPv6 on its LAN. It will be necessary to focus on the correct definition of its PAC proxy files, as well as to ensure that the VPN rules are transposed, particularly those related to split tunneling.

• OS image

While OS TCP/IP stacks have been supporting IPv6 for a decade, support for some RFCs such as providing IP DNS via router advertisement (RDDNS) are more recent. For example, support on Windows 10 starts with build 1703.

Precedence

The precedence concept defines the priority given to the different types of addresses, and thus notably the prioritization of v6 over v4 or the opposite.

The order is standardized, RFC 6724 dated 2012 replaces 3484 from 2003. Here are the differences:

Address	**Prefix**	**Former Precedence (RFC 3484)**	**New Precedence (RFC 6724)**
IPv6 Loopback	::1/128	50	50
Natives IPv6	::/0	40	40
IPv4	::ffff:0:0/96	10	*35*
6to4	2002::/16	30	30
Teredo	2001::/32	05	05
ULAs	fc00::/7	40	**03**
site-local	fec0::/10	40	**01**
6bone	3ffe::/16	40	**01**
IPv4compat	::/96	20	**01**

It can be noted that between the 2 releases, IPv4 has become preferred over v6 transition mechanisms (6to4, Teredo) and that local site addresses are now deprecated. Native IPv6 still has the lead.

Beware also of private ULA addresses which are getting lower priority than IPv4, this can matter.

```
PS C:\Users\JC> netsh interface ipv6 show prefixpolicies
Recherche du statut actif...

Précédence  Libellé  Préfixe
----------  -------  ----------------------------------
        50       0  ::1/128
        40       1  ::/0
        35       4  ::ffff:0:0/96
        30       2  2002::/16
         5       5  2001::/32
         3      13  fc00::/7
         1      11  fec0::/10
         1      12  3ffe::/16
         1       3  ::/96
```

Figure 11. Precedence in Windows 10

Result of `netsh interface ipv6 show prefixpolicies` command. This behavior can be changed using the following registry key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\tcpip6\Parameters documented here

```
FILES
       /etc/gai.conf

VERSIONS
       The gai.conf file is supported by glibc since version 2.5.

EXAMPLE
       The  default  table  according  to  RFC 3484 would be specified with the following configuration
       file:

           label  ::1/128       0
           label  ::/0          1
           label  2002::/16     2
           label ::/96          3
           label ::ffff:0:0/96  4
           precedence  ::1/128       50
           precedence  ::/0          40
           precedence  2002::/16     30
           precedence ::/96          20
           precedence ::ffff:0:0/96  10

SEE ALSO
       getaddrinfo(3), RFC 3484
```

Figure 12. Man page Debian 10 (Buster) of GAI.CONF

On many GNU/Linux distributions it can be controller in the GetAddressInfo file */etc/gai.conf* .

Here an example of Debian 10 (Buster) man page, no clue of the new RFC from 2012… https://man7.org/linux/man-pages/man5/gai.conf.5.html

Changing IPv4 precedence (represented by ::ffff:0:0/96) can help you to prevent any malfunction on a production system when deploying IPv6. Indeed, unless a literal specification of an IPv6 address or a DNS record that matches only an IPv6 address is specified, the system will continue to use IPv4 for the requests it submits. Remember to normalize once a stable state is reached.

Be aware that some programs such as browsers implement their own prioritization between v6 and v4, independently of the OS stack configuration. Also, the implementation of the Happy Eyeballs 2 mechanism (RFC 8305) may vary. (Delay between DNS A and AAAA requests, time to wait for the return, timeout of the remote socket with failover…). Another example, the CURL tool nicely supports Happy Eyeballs compared to its competitors.

Software agents

OS images are usually internally shipped with preconfigured agents, more rarely these agents are deployed at first launch. In both cases, they are a part of the base and allow to ensure its

compliance, security, etc.

These agents include backup, antivirus, telemetry and monitoring, asset management, package/strategy deployment, etc.

As long as you do not plan to retire IPv4, there is no need to prioritize the transition of these services to dual-stack, it can be done at the same time as the applications.

The important point is to check that these agents do not encounter any problem linked to the simple presence of a routable IPv6 address on the host.

So don't picture an Ouroboros where you must do everything at the same time without knowing where to start.

Once the OS bases are ready for dual-stack operation, you can work on moving to IPv6 only once the ecosystem is ready, if that is your scenario, and then you can tackle the upper layers, the middleware and applications.

• WORKPLACE SERVICES

Directory

Directory Service carries LDAP and Kerberos functionalities, in addition to occasionally hosting specific DNS zones and other ancillary services. Their omnipresence within the information system makes their migration essential. The leading product on the market, Active Directory, works well in dual-stack, it has been in fact used internally by its editor in IPv6 for several years.

regarding SPN (Kerberos Service Principal Name)

In an effort to simplify the declaration of each server and its service behind a single name, some implementations by default rely on a reverse DNS query. Thus, when the user requests a service ticket for a server via a CNAME rather than its original hostname, the Kerberos server will retrieve the original FQDN via reverse DNS. The other tedious solution is to declare all possible SPNs of each server.

This behavior (canonical resolution), although discouraged by RFC 4120, is used within Active Directory for its simplicity. It is therefore necessary to make sure that the Kerberos server (KDC) will not execute a reverse DNS query with an IP retrieved through a DNS64, or at least that the DNS server knows how to lie and formulate an appropriate response to these particular queries.

Last but not least, there are still some IP-based SPNs rather than hostname-based SPNs (usually for old applications with, you have guessed, a hard-coded configuration, or simply an IP-based configuration). This is a rare case since Windows on the client side stopped to support this function between Vista and Win 10 1507, forcing to downgrade to NTLM for such services. This specific case will require the use of 2 SPN per machine and service (v4 and v6).

File shares and package repositories

Whether they are visible to users or not, servers providing files generate a heavy traffic load. If your project targets v6-only clients with NAT64 it would be a good idea to have these servers migrated to a dual-stack (or have dedicated translation platform) which would greatly relieve the centralized translation platform.

This includes SMB, NFS, WSUS, SCCM, package repositories, EDR signature repositories, CMS, Sharepoint, etc.

NFS older version 4 uses Portmapper and doesn't works with NAT64.

Communication

The e-mail infrastructure can cope with NAT64 for a long time, but the large amount of traffic this system generates makes it necessary to migrate at least the client access layer to IPv6. For the internet facing portion, the MTA, there is no rush, we are not about to see SMTP servers offering exclusively IPv6. A migration will require checking the compatibility and efficiency of your content checking and antispam solutions.

Similarly, in telephony, it is the customer-facing part of the system that needs to be quickly migrated, and much more urgently than messaging, in order to bring IPv6 compatibility to P2P communications between customers or between customers and centralized infrastructure. Urgency reinforced by the bad surprises of NAT64 with SIP, unless you trust ALGs. But RTP flows being more and more frequently encrypted, you should not rely too much on ALGs.

You should know that a growing number of SaaS providers support IPv6, with some rare exceptions such as an on-premise SBC interfacing with its SaaS counterpart, which is not very annoying.

• APPLICATIONS

Rather than specifically launching a tedious qualification campaign devoted to IPv6, it is preferable to use the opportunities offered by major upgrades of these applications to qualify them, this time in IPv6, and only in IPv6. Feedback from major editors shows that qualifying an application in IPv6 is enough and that it is useless to re-cycle everything in IPv4, as recent methods and instruction calls are backwards compatible without additional work. This is obviously not true for an application using an old programming language and/or with hard-coded addresses.

Here is a list of questions to ask yourself about each application:

- **Are there any users of the solution in v6?** (ask the editor, integrator, test…);
- **Is the programming language used compatible with IPv6?** And in a stable and reliable way? (Many implementation bugs have been corrected in different languages up to 2015);
- **Is the socket opening code agnostic of the IP protocol version?** Inet6Address and InetAddress in Java for example;
- **Does IPv4 and v6 traffic pass through the same socket?** Previous example vs use of IPv4-mapped address (still in Java);

- **Does an application handle IPv6 on the client side?** on the server front-end? on the server back-end in the case of an n-tier application? (even if this last point is less critical);
- **Does an application make calls via literal address rather than via DNS lookup?** IPv4 configuration field only for example;
- **Does an application use a protocol embedding the literal address?** Like SIP with telephony, or active FTP;
- **Does an application initiate connections to client endpoints?** Example of active FTP with its 2 concurrent control and data sessions, one in each direction. Or remote control, as well as SIP, DICOM, etc.;
- **Is there any IP address processing within your application?** For example, identifying the client by its IP rather than by its username;
- **Is RFC 8305 "Happy Eyeballs v2" correctly implemented to allow fast switching between the 2 protocols?** (The calling function used and the default language configuration should be looked at in detail, it is very easy to not implement it correctly in java for example);
- Finally, if the application is not IPv6 compatible, **will the logs retain the port in addition to the IP?** (To ensure NAT64 tracking) cf RFC 7768 of 2016, itself inspired by RFC 6302 of 2011 which initially recommended this for front-end servers on the Internet.

Various audit tools exist, some are integrated into development environments, others standalone such as Microsoft checkv4, PortToIPv6, IPv6 code checker, IPv6 care, etc. These tools can either audit the code or detect socket calls when the code is running and identify the used method.

Mobile applications published on the Google Play Store and the Apple App Store are bound to use IPv6 compliant network methods and functions since 2016, it has been a good example of fast code adaptation.

Without delay, incorporate IPv6 into your specifications and architecture requirements for new applications. Also schedule a date when upgrades of an existing application should include IPv6 implementation.

Figure 13. Example of analysis of a Web App

How to deal with a service provided through web browsers?

In n-tier architectures, priority is given to the front end, which is accessed by clients. The application back-end can remain in v4 much longer.

Ideally, you should take advantage of application obsolescence and renewal to implement IPv6.

Just to know, the famous Curl utility have now been supporting IPv6 for more than 20 years.

Case of apps handling IP

IP address is a key element found in inventories; it may include the following tools:

- Assets management / CMDB / IPAM;
- Infrastructure config Orchestrator / Deployment / Config Backup;
- Operation monitoring / metrology / incidents tracking / helpdesk;
- Information harvesting scripts;
- Log correlation (SIEM) / Audit;
- Access management Flow openings / Identity.

Usage of IPv6 implies reviewing address storage and processing for various reasons:

- IPv6 address sometimes comes in addition to IPv4 (dual-stack);
- It is longer;
- An interface can carry several IPv6 (local link, temporary routable, stable routable, etc.).

A simplification method can be to manipulate everything like IPv6, including IPv4 through the representation prefix ::ffff:0:0/96. This way eases application code cohesion and simplification.

However, see the Appendix in the section Examples of implementation problems to find a possible problem with this method.

In any case, it will be necessary to store the addresses in their canonical (shortened) shape in order to reduce their size. The code performing the canonicalization must scrupulously respect the RFC 5952 so you always end up with the exact same string to parse. Note that addresses must also be stored with lowercase characters (RFC section 4.3). For example, ab01::ffff and not AB01::FFFF. The non-fulfilment of this last recommendation can even cause problems in protocols carrying the IP in payload like SIP.

Addressing plan

Addressing plan

The constitution of this important referential that is the addressing plan is a long process to be carried out iteratively.

The addressing plan is to be built according to the organization's specificities. Beyond the elements mentioned in this section, it is necessary to exchange with many interlocutors by forecasting the result of your different plan proposals. In any case, think in the long term and keep some free space at the top of the block.

Before getting down to the plan building options, let's start by discussing the prefix choice to operate within the organization.

• PUBLIC OR PRIVATE?

Public IPv4 addressing is scarce and is therefore naturally limited to Internet exposure. Within the internal network the RFC 1918 ranges including 10.0.0.0/8 prevail.

With IPv6 the question arises to choose between the 2 options.

Depending on your size and activity, the ideal is to have your own prefix assigned by the continental manager (RIR). Similarly, a large subsidiary or one with a high degree of independence in terms of IT would be well advised to request its own block from RIR.

You can also get a PI from a sponsor LIR, that's not expensive at all.

The minimum assignment policy is /32. It is possible to obtain larger blocks if the structure's size justifies it.

• SMALL ORGANIZATION

For a small network, it is possible to use the prefix supplied by the ISP through prefix delegation. However, depending on a third-party structure for the addressing of its resources is not ideal and one quickly reaches the limits of this mode, for example for fixed addressing of internal servers. There is still the possibility of delegating prefixes with DHCPv6-PD for those who are adept at a sudden change. In an ideal world all tools could adapt to prefix renumbering, but in practice it will probably be a long time before all configurations are dynamic.

ULA

Private IPv6 addressing appears as a solution, this "equivalent" of the RFC 1918 IPv4 is called ULA for Unique Local Address and corresponds to the FC00::/7 prefix.

In order to limit the risk of conflict with another structure with which you could hypothetically exchange privately one day, for example through an IPSEC tunnel, it is recommended to randomly choose a prefix in this /7 rather than starting from the bottom (RFC4193 even imposes this randomization).

Take for example an arbitrary /48 in FC00::/7 and build your addressing on it. For a very small

structure a /56 is enough. Be careful, the size must not be too large for a reason we will see below. RFC4193 proposes a pseudo random generation algorithm to obtain a 40 bits ID, which gives a 48 bits prefix.

However, ULA has a lower priority than IPv4 in the current precedence rules. IPv4 will therefore be favored in a dual-stack environment (except for traffic between 2 ULA). You have 2 solutions:

- Modify the behavior of all hosts to give ULA priority over IPv4;
- Request an independent PI prefix and use it only internally; this last method is recommended.

Provider-Independent (PI) Prefix

Recommended solution, ask for a PI prefix in /48 at a LIR. No need to announce it on the internet etc. But it will allow you to have a LAN with a unique addressing space and to avoid the ULA vs IPv4 precedence problems.

Packets using this addressing on the LAN (non-advertised PI on the Internet) must undergo a NAT in order to exit on the Internet. If in IPv4 we use NAT44 + PAT with stateful session table, here we will use Network Prefix Translation v6 (NPTv6 | RFC 6296).

Prefix Translation swaps the first few bits of the address to match an IPv6 prefix to another of the same size. No other changes take place, everything is stateless.

It will be only needed to map its private prefix /56 (or other size) to the public one provided by the carrier to exchange on the Internet while controlling its internal addressing. It is possible to map an internal /56 to a chunk of public /48 routed, but not the reverse of course (hence the importance of not picking a too) large range on your LAN.

Thanks to NPTv6, the company can switch from one ISP to another without altering anything internally, and it doesn't break the PMTU-D maximum transmissible size discovery.

NPTv6 drawbacks

There are still negative sides, firstly protocols encapsulating the address in the payload like SIP, H323, etc. will always require the use of a corresponding Application Layer Gateway (ALG) on the equipment performing the translation. As in NAT44, ALGs can be an attack vector, see in particular the recent slipstreaming methods that have forced browser editors to block certain destination ports.

Secondly you will need to synchronize your DNS records between the internal DNS zone (non-announced PI or ULA depending of your choice) and the external zone version for exposed services. So, you don't mistakenly publish an AAAA record with an unreachable IP on the internet in one hand, and you don't use the Global Routable IP internally on the other hand as it would imply tromboning through the NPTv6 platform. For example, a LAN client requesting a DMZ server should reach it using directly towards its internal address (ULA or non-announced PI).

Oh, and don't forget to create PTR for both types of address, it's important for some services like SMTP MX as this is part of anti-spam checks. Hopefully there are mechanisms enabling to auto generate PTR.

• LARGE ORGANIZATION

Start by obtaining a public PI (Provider Independent) prefix, or several in the case of subsidiaries or a multi-continental geographical presence.

Some specificities must be taken into account before building your plan.

Your BGP public announcements cannot, by convention, be smaller than /48. (Similar situation to /24 IPv4). However, there is no need to dedicate an advertisable prefix that would correspond only to the exposed servers, we will see why.

IPv4 and the omnipresence of NAT44+PAT have brought practices that it is no longer necessary to reproduce in IPv6, in particular the false security feeling offered by NAT44 in inbound. The diode aspect is inherently present because of the session tracking requirement, so it is stateful. And while it is normal not to have upnp auto-port forwarding as found on consumer equipment, it is more difficult to guard against recent slipstreaming attacks using ALGs as mentioned above.

A stateful NAT + PAT equivalent existed in IPv6, but its use is not recommended. In fact NAT-PT (NAT Protocol Translator RFC 2766, not to be confused with NPTv6) is simply not usable and has been archived, see RFC 4966 which states the reasons why this mechanism has been discontinued.

Thus you will sometimes find security recommendations which are to have an internal network in private IPv6, to use NAT in exit in order to make its addressing plan invisible to the outside, etc.

These recommendations are just reminiscent of the IPv4 habits, as well as the fact that using private internal addressing with NPTv6 prefix translation to exit the network has no security interest for a large company, and does not hide the details of the internal plan at all, since it simply switches the first few bits of the address. It is important to remember that NAT does not protect, only a firewall with the right ACLs and possibly inspection are effective.

The whole Information System should be addressed with the global public prefix assigned to the company.

Management of direct internet access

NPTv6 prefix translation can still be used for other situations. Let’s take a company that wants to use local breakout (LBO) on its campuses in order to reach Internet resources (a SaaS application, for example) without going through its datacenter. The traffic will then have to go from an address that belongs to the company to one that is provided by the local internet carrier of the campus.

Note that this frequent use is a reason to have site prefixes based on a geographical assignment. This enables you to have only one NPTv6 rule. If your site addressing is fragmented you will have to map each local /64 to a /64 belonging to the prefix provided by the local carrier (typically a /48). This means more rules and more work.

A more subtle deviation from this local breakout use case, if the campus is very large and the local carrier allows it, it is possible for the site to advertise its own /48 (or more) via BGP directly on the Internet.

In this case, the site’s devices use the addresses of a prefix that we will call "Site" /48, this prefix is

not announced but a larger prefix "Global" /32 which includes it is announced by the datacenter. Finally, the site announces locally and directly on the Internet a prefix "LBO" /48 which also belongs to the global /32. This setup would cause a huge increment in internet BGP Fullview but would be still usable if the plan allows for a route aggregation at the carrier edge.

The local rule of NPTv6 translates the Site/48 prefix into LBO/48 at the local internet exit. The operation of the routing decisions of BGP privileging fine routes will allow the whole to function without conflict, with thus this time IPs which all belong to the company. If we have several sites in this situation with the same provider, it would be smart to ask to aggregate the announcements.

Finally, some traffic will leave the site directly via the LBO prefix, while other traffic that must undergo more advanced processing in DC will leave via the Site prefix (depending on the configuration of the workstations' proxification).

The emergence of so-called "SASE" (Secure Access Service Edge) solutions can make it possible to dispense with DC processing altogether, in which case it is no longer necessary to use two prefixes with NPTv6.

The latency gain offered by LBO can be significant, as the DC and its dependencies are no longer used. However, the same level of security must be ensured in terms of filtering, antivirus analysis, etc. The strategy will therefore vary between authorizing some of the flows (recipients with a sufficient level of trust) and all Internet traffic, depending on the level of protection that can be achieved. Whether it is provided locally via VNFs, SASE or via a Cloud solution.

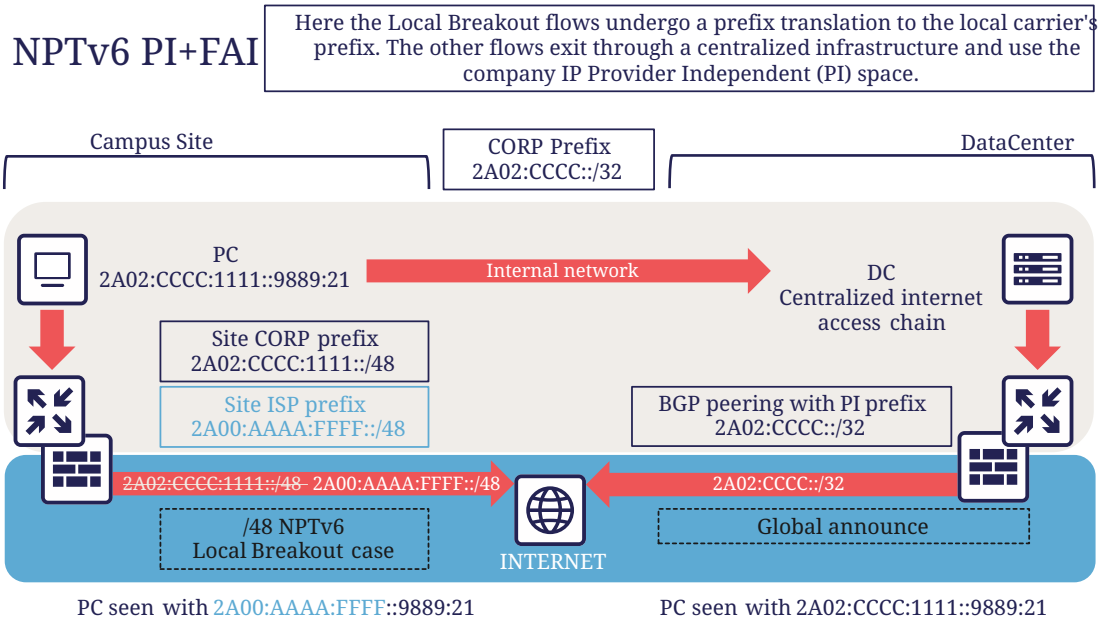


Figure 14. Schema of NPTv6 PI+FAI

For infrastructures that need to be completely isolated from the Internet and any partner (such as a SCADA network), it is possible to use ULA addresses. This does not prevent rebound attacks from another internal system, firewalls are sufficient to block traffic at the edge of these networks anyway. The contribution of ULA is therefore almost null and remains a subjective choice. Again, we recall that ULA has a lower priority than IPv4 in the current precedence rules. IPv4 will therefore be preferred in a dual-stack environment.

• LOGICAL GROUPS

Historically, IPv4 accustomed us to assign ranges by location in order to minimize the number of routes by summarizing, more recently some projects might have made the opposite assessment, i.e. assigning the geographical location from a block dedicated to a specific use as in a WiFi deployment project in agencies or IoT.

This last case is beneficial to filtering because it is focused on usage rather than location.

The choice is especially important because, unlike IPv4, you cannot use a mask, only the prefix to filter.

In IPv4, it is possible, even if rarely used, to use for example the wildcard 0.0.240.0 to select n identical hosts from different subnets. In v6 this disappears.

If devices support a large number of routes, manual rules applied to routes would become complex to implement with a usage-centric plan, and we already know that despite automation and the arrival of SDN on different perimeters, BGP will remain the way to interconnect "black boxes" between them. Nevertheless, it would still be possible to use scripts and a route server like bird or FFRouting to automatically perform classifications and apply policies or simply to make the best use of communities on announcements.

The 2 geo or type centric options have advantages and disadvantages, which can be compensated by automation (Consolidation of filtering rules VS Consolidation of routes and sites). As stated before with NPTv6 explanation, it’s easier to base addressing on location.

• BUILDING BLOCKS

The slicing can privilege the multiples of 4 bits (hexadecimal characters), /32, /48, /52 etc in order to facilitate reading, tendency which corresponds to our habit of slicing the IPv4 by octet and which leads to waste in the exact case of v4.

The grouping of 4 characters is called a Hextet, for example: A9B4:

Each hexa character can be named a nibble.

Therefore, we remind that if v6 offers a large number of addresses, this should not be an invitation to waste, we will avoid, for example, leetspeak such as ”c01d:c01a:c0fe” / “cold cola cofe” within the prefix/network ID.

One can immediately think of associating these blocks of characters with:

- Legal Entity / Sector of Activity;
- Geographic location ID;
- Network type;
- VLAN or VNI number;
- Operator;
- Device Model.

Numerical elements can be kept as they are, taking more space, or encoded in hexadecimal, eliminating human readability.

For example, to store the VLAN number, from 0 to 4094 (12 bits) we have the choice:

- 4 0 9 6 which is 4 characters so 16 bits;
- F F E or 3 characters to form 4094 in hexadecimal, with a free hexa character remaining in the hextet x F F E.

In the case where we create a new reference frame, like the network types, it would be better to write them directly in hexadecimal if the division allows it.

If we go back to the list of building blocks, some of them have a life cycle unsuitable for integration in an addressing plan. For example, the carrier may change in the meantime, as may the manufacturer and model of a level 3 device. (We know from experience that the maintenance and change would not be passed on because "it works well as it is"). We will later see an exception for interconnection.

In the datacenter, the same thing will happen with VLANs, the use of E-VPN + VxLAN technologies with a 24-bit VNI number will relegate the VLAN to the background, the same goes for proprietary segmentation technologies that integrate notions of client tenant, resource pool, etc.

From this we can deduce that the plan must only integrate elements that are relevant and static in time, which gives us:

- The division / entity at a high level in order to allow the breakdown of the structure (as is done in an Active Directory);
- The location either by continental tree / plate / site, or by the numbered site code;
- The type of network, with subcategories to facilitate the management of filtering and to delegate part of the addressing plan.

• PREFIX SIZE

Standard

From the outset, /64 appears to be the unalterable standard for a network (RFC 4291), particularly for the auto-configuration mechanism SLAAC to work.

```
The war front movements (last 20 years)
RFC 3513 - "only /64 is valid"
RFC 3627 - "don't use /127, use /126 if you must"
RFC 4291 - "reaffirming: only /64 is valid"
RFC 6164 - "a /127 is OK to use too"
RFC 6583 - "there are problems with /64"
RFC 7421 - "/64 is the best!"
RFC 7608/BCP198 - "every prefix length must be forward-able"
RFC 4291bis-07 - "fine, /64 and /127 are valid, but nothing else!"
…
RFC ???? "????"
```

Figure 15. Is a standard actually a standard?

The norm is /64, using anything else to connect hosts might cause unexpected behavior or incompatibilities.

Regarding site prefixes, the recommendations have also evolved, RFC 6177 adjusts the prefix to the real need, whereas previously /48 was mandatory.

Operators generally assign a /56 or /60 to domestic customers, and /48 to professionals. The terminal networks are always in /64, except for the interconnections.

Interconnection

Carriers seem to recommend /125 interconnections. In order to cut between 2 hexadecimal characters, it would be a good idea to provide /124 in the plan and use the 125th for failover when changing equipment or provider.

This reservation does not prevent you from setting the point-to-point interfaces to /127.

These reservations for interconnections and loopbacks can be inherited from the site addressing, or on the contrary from a /64 prefix dedicated to be divided into /124 and more for interconnections.

In the latter case you will need to advertise many fine routes on your network.

Building interconnections with link-locals works but has many drawbacks detailed in RFC 7404 (no interface ICMP return because not routable but loopback one instead, address that changes in case of hardware replacement as auto based on EUI-64 MAC, etc.) On the other hand, one of the big advantages is the lightening of the routing tables as well as the reduction of the attack surface. The path tracing aspect with link-local can be retrieved with RFC 5837. The choice will therefore generally be different between a corporate network VS a large ISP or a GIX exchange point.

You can build your interco /124 prefix with the BGP AS number of the third party, the router ID, etc. In short everything that will help you in your daily tasks.

Be careful with IPAMs, they often refuse to let you register anything else than /64, however it is not abnormal to have interconnections with long prefixes.

Apart from interconnections, /64 is the current standard and it would be a shame to venture to use something else.

Some RFC drafts aim at allowing SLAAC to provide something else than /64, see draft-mishra-v6ops-variable-slaac-problem-stmt and draft-mishra-6man-variable-slaac. These drafts try to solve the problem of subdivision of a single /64 provided for example by a mobile carrier via 3GPP link. The objective is to be able to create different networks on mobile micro-infrastructures, typically a router with multiple client networks or a connected vehicle whose different internal networks sometimes use Ethernet, sometimes BUS and cannot be bridged. It is even necessary to have direct exchange networks with neighboring vehicles (V2V). Future will tell if theses drafts become a standard or if they are abandoned in the event all carriers start supporting DHCP-PD on cell phones with /56 via 3GPP as it is often the case for home connections.

• COMMON SERCVICES ADDRESSES

For convenience, it is interesting to assign short addresses to services for which the IP must often be entered manually, of course first of all the DNS servers, but also the routers' interfaces.

Thus, the address set at the very beginning of an organization, pre:fix:0000:0000:... should be dedicated to fine allocations to facilitate the work of the operators/administrators by allowing them to retain them from the top.

On each plan level, regional plate, site... it would be good to reserve the 0 and 1 for services using shortened addresses. Again, to facilitate day-to-day tasks.

However, do not forget to not put all instances of the same service within the same prefix. Having for example all the DNS or SMTP relays in the same prefix and therefore depending on the same route is not a good practice. In case of a routing incident affecting this prefix you can have as many physical and/or logical instances of the service as possible, it will still result in a blackout.

• TIME-SCALE EVOLUTION

In order to accommodate migrations at different levels, migration bits can be implemented.

A network migration bit can facilitate changes in hardware, WAN links etc. This bit should be 64th in order to be taken into account in /63 filtering rules. It would allow transitions of subnets, VLANs, devices in a progressive way without any other modification since the ACLs in /63 would encompass the 2 usable /64.

For example, a campus changes its core and migrates to a MAN at the same time. The new networks are set with the transition bit and routed in parallel with the old ones. Business tests can take place on the new infrastructure before the migration thanks to the wide filtering including this bit. This avoids big bang migrations and limits the discovery of post migration incidents.

At the next changeover the bit is toggled. No state between 0 and 1 is preferred.

Any operation involving a change of equipment, operator, move, etc. is therefore greatly facilitated.

However, it is important to prevent the announcement of a twin network that would unintentionally take advantage of global filtering rules. Monitoring the origin of routes belonging to the same migration pair is a necessity.

In a more general manner, keep place for future frames of reference. This to allow to cope with new architectures without requiring top level new blocks.

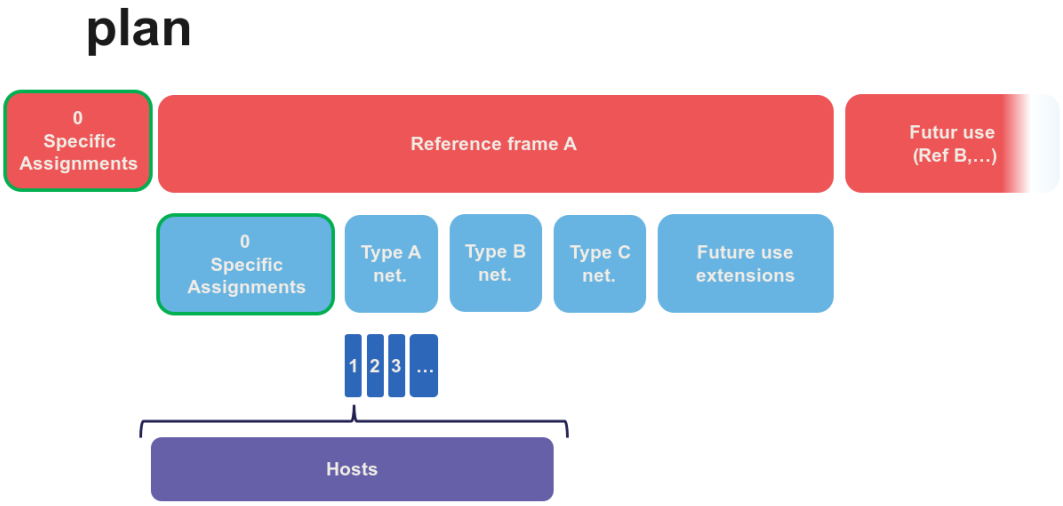


Figure 16. Example of reference frame hierarchy

• USE OF HOST NUMBER 0

In IPv6 there is no more network number address nor broadcast address, all possible addresses can be assigned to the hosts.

However, bad regex are sometimes present in configuration fields of applications. One can find bugs in systems that do not support being configured with an address ending in ::0, for example 2001:db8:abcd:1234::/64. Sometimes on their interface or on the interface of a third party element like the DNS or NTP server address.

Also, we recommend avoiding addresses with host numbers in 0 at a minimum for servers that may be configured literally in devices such as printers, cameras, and any embedded equipment.

Using DNS limits the risk, except for DNS itself. Keeping a final 1 to the addresses of your DNS servers can prevent this type of problem, even if it tends to disappear.

The use of this first available address also raises the question of the confusion risk between address and prefix. In IPv4 the network address can never be used for a host (except in the particular case of /31 RFC 3021 intercos) whereas in IPv6 it is possible to have the same address indicating a prefix and a host, the size of the prefix being then the only delimiting factor. For example a host 2001:db8:abcd:1234::/128 belonging to the network 2001:db8:abcd:1234::/64.

For this 2nd reason of human readability, it is better not to use the host address 0 at all.

• PER INTERFACE ISOLATION

Some technical or security constraints may result in instantiating several network interfaces on servers. For example, specific security strategies require dedicated management interfaces. Sometimes the interfaces used by backup agents are also compartmentalized for performance and isolation reasons.

This raises the question of the output interface choice. The IPv4 stack of a system will use a metric to choose the interface carrying the route 0.0.0.0/0, the other interface(s) being limited to only route the subnet to which they are each bound. It is then up to the server administrator to set static routes, or the network to perform NAT so that an administration flow arrives via a NAT address belonging to the same subnet as the administration interface.

What about IPv6? The short RFC 7608 states that the routing decision should be based on a bitwise comparison of the machine's interfaces with the destination address. The interface with the most common bits wins the race.

Thus, a machine with 2 cards of address 2001:db8:abba:CAFE::5 and 2001:db8:abba:1001::5 and sending a packet to 2001:db8:abba:C9D6::6 will use the first of the 2 cards.

This point should be taken into account in your addressing plan in order to reserve a high level prefix for administration or backup. This will facilitate the use of dedicated interfaces if necessary.

Does another method exist to force use of a specific interface to an off-link prefix without modifying the host configuration and without having pre-established a plan based on the RF 7608?

With IPv4 the use of the DHCP option 121 (classless static routes) allows to push fine routes to an interface (NB: this option overwrites the default route that has been advertised, which has to be copied to an option 121 if it is to be advertised).

Nothing similar in IPv6, announcing a prefix via the Router Advertisement with the L bit (on-link) at 0 will not result in learning an indirect route. As for DHCPv6, it does not include an equivalent to option 121.

The RFC 4191 suggests an extension (type 24) to the RA allowing to announce routes, written by Microsoft it works since Windows Vista, the Linux kernel also implements it since the commit 930d6ff and ebacaaa of 2006. However, the option is not necessarily enabled.

Be careful, this RFC has 2 parts, one is about the correct handling of the RA priority, the other one deals with additional routes.

If you cannot use the option, you can always try to send prefixes with the on-link option set to 1. The hosts will then add a route to the router for this prefix. However, this is a divergence from the standard.

• IP V4 / V6 MAPPING

As discussed in the dual-stack section, the parallel use of IPv4 and IPv6 leads to extra configuration and operation, and therefore to extra costs.

Good practices can facilitate the implementation of automations that reduce these efforts.

Network prefix number

It is important to have a mapping database between an IPv4 network and the corresponding IPv6 network. The ideal is to have this functionality within the IPAM, or alternatively to use a field in the IPv6 section of the IPAM to indicate the associated IPv4 network with its mask.

If the IPAM cannot store this information in any way, even by circumvention, then a third party inventory tool will have to be used. This can be another IT repository, a dedicated database, etc. The important thing is that the repository is API-enabled so that it can be requested from other systems.

Let's take the example of firewall filtering rules, recreating all existing rules in IPv6 during deployment and then double the flow opening request process would be far too cumbersome.

Instead, it is possible to implement automations that examine every night whether each object representing an IPv4 network does not have an IPv6 match, and if so, modify the object to add the associated IPv6 prefix. This prevents errors, whether they come from the firewall administrator or from applicants who might make a mistake about the IPv6 prefix when requesting to open a flow.

With a more advanced solution, it is possible to manage changes synchronously, without worrying about the dual-stack.

Host number / Interface ID

In the second half of the address are the 64 bits dedicated to host identification. Here again, good practices exist to facilitate the mapping of the IPv4 and IPv6 address of a dual-stack host.

These practices are obviously only valid with stateful DHCPv6 or manual addressing.

The easiest way is to keep the IPv4 number and carry it over to IPv6. Let's take the network 10.2.3.128/25 and a server 10.2.3.239. After IPv6 deployment, this network arbitrarily uses in the following prefix 2001:db8:abba:CAFE::/64.

Numbering the server 2001:db8:abba:CAFE::239 facilitates operation up to the human readability level. One can also use the hexadecimal 2001:db8:abba:CAFE::EF if one wants the values to keep the same strict numbering from a binary point of view. However, readability is lost.

Another option is to keep the ordinal nature instead of the number. With this same network, we see that the server uses the 89th usable IPv4 of the network 10.2.3.128/25 (239-129=110) .128 is the network number and is not assignable here in IPv4.

This ordinal pattern results in 2001:db8:abba:CAFE::110 or 2001:db8:abba:CAFE::6E in pure hexadecimal.

The most meticulous among you will have noticed that the host number ::0 is usable in IPv6 since there is no network number and broadcast address, based on this postulate one could also convert in the ordinal mode an IPv4 .1 address into IPv6 ::0. However, this is not practical in terms of reading because of the risk of confusion with a prefix, and can also cause problems on systems, for example because of poorly implemented field checks as previously said.

The choice between these 2 methods and the 2 counterparts (decimal or hexadecimal) is to be discussed. The first one in its decimal version is clearly the most practical, but other criteria may come into play as we get closer to an orchestrated world.

Here are a few examples:

IPv4 Network	IPv4 Host	IPv6 host num - Mapping	IPv6 host num - Ordinal
10.2.3.128|25	10.2.3.239	::239 dec |::EF hex	::110 dec |::6E hex | 239-129
10.2.4.0|24	10.2.4.239	::239 dec |::EF hex	::239 dec |::EF hex | 239
10.5.0.0|23	10.5.0.239	::239 dec |::EF hex	::239 dec |::EF hex | 239
10.5.2.0|23	10.5.3.239	Relative ::1239 dec |::4D7 hex Absolute ::3239 dec |::CA7 hex	::495 dec |::1EF hex|(256+239)
10.6.0.0|16	10.6.28.239	::28239 dec |::6E4F hex 28 * byte+239	::7407 dec |::1CEF hex (28x256) + 239
	10.6.28.3	::28003 dec |::6D63 hex 28 * byte+003	::7171 dec |::1C03 hex (28x256) + 3
10.8.64.0|18	10.8.72.50	Relative ::8050 dc |::1F72 hex (72-64)=8 blocs + 050 Absolute ::72050 d|::11972 hx	::2098 dec |::832 hex (8x256) + 50

We see with the example table that for an IPv4 network split at the level of the last byte (/24), the ordinal has the same value as the mapping, since the counting starts at 0 in both cases.

It becomes more complex with an IPv4 network larger than one byte, in the example a /23. What should we do here to distinguish 10.5.0.239 from 10.5.1.239 ? Adding a 1 to indicate that we are moving beyond the last byte seems to be a good method. We then count all the addresses of the /24 constituting the network, including unassignable addresses, that is 256.

But the quest for readability could also have led us to recopy the previous byte and define ::3239 rather than ::1239, thus migrating from a relative to an absolute reference. Moreover we can even copy the whole host's IPv4 into its IPv6 host number, not the most elegant solution though.

The following examples also illustrate the need to keep the 0's of the bytes in the "Mapped" mode in order not to generate duplicates. 003, 050, etc.

As you can see, the important point is to define the engineering rules clearly and to stick to them.

To sum up :

- The decimal carryover mode in absolute, i.e. copying the full byte, or even 2 bytes for networks larger than /24 (etc.) clearly wins for matching readability. However, it implies long host addresses;

- The use of hexadecimal is probably only advantageous in an automated environment;
- 2 BE or not 2 BE, the use of hexa and powers of 2 make the intellect work;
- Again, these solutions will allow the generation of ACLs, etc. without having to work twice.

The mapping can also be done via the DNS A and AAAA records of each server, which then requires another form of accuracy.

Regarding hosts, there does not seem to be any product today that allows without prior configuration to assign the same host number in IPv4 and IPv6 based on an integrated IPAM mapping.

• FOR NATIVE V6 NETWORKS

When setting up a native IPv6 network, the previous rules related to hosts do not apply. You can then use part of the 64 bits to indicate host details.

For example a letter to indicate a desktop server, another character to specify a printer. This should remind you of existing engineering rules / naming conventions for hostnames.

In a datacenter you can think of tagging the business associated with a VM, etc.

However, this remains complex and redundant with a CMDB, especially since the address cannot be changed easily if needed.

The other solution, at least on the server side, is to fix the interface ID so that it is static and does not depend on the MAC address (and therefore does not change when the card is changed, physical or virtual, and does not expose the manufacturer in the address). In conjunction with SLAAC for the provision of the prefix. This solution remains simpler than a manual setup everywhere.

In general, it is only necessary to define a range segmentation in small multi-purpose networks on small sites.

• PUBLIC ANNOUNCES

What to advertise on the internet?

To that question, some will answer "As limited resources as possible". In fact, does directly announcing its /32 instead of some /44 DMZ really represent a surface attack reduction? Will it matter when implementing stateful and IPS firewalls? The end-to-end IPv6 goal will probably end up pushing wide advertisements anyway.

Looking at the BGP IPv6 table contents, we find that the majority of advertisements are /32, /40, /44 and /48.

Adds and Wdls per Prefix Length

/28	+1	-0
/29	+41	-13
/30	+3	-2
/31	+1	-3
/32	+131	-110
/33	+49	-31
/34	+53	-58
/35	+35	-11
/36	+141	-26
/37	+4	-3
/38	+28	-3
/39	+4	-9
/40	+436	-55
/41	+8	-2
/42	+9	-9
/43	+26	-3
/44	+147	-74
/45	+42	-6
/46	+67	-126
/47	+8	-3
/48	+871	-969
/52	+2	-0
/56	+23	-27
/128	+4	-0

Weekly IPv6 announces delta https://cidr-report.org/v6/as2.0/

The /48 advertisements represent half of the advertisements in number, with 54,000 routes, but not in volume of unique addresses, since each /32 contains 65,536 (2e16) times more addresses than a /48.

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

https://bgp.potaroo.net/index-v6.html

https://www.cidr-report.org/v6/as2.0/

The last URL notably displays the following weekly report with the observed prefix additions and deletions.

*Some anti-DDoS solutions work by re-announcing the attacked prefix via a "cleaning" network. The /48 being the smallest, you will have to announce more in nominal if you run this type of solution.

Ideally, advertise in a large way according to the geographical locations of your exit nodes. IPv6 peering can also be an opportunity to experiment with RPKI route signing if you don’t already do it, or to experiment with RTBH and Flowspec.*

Security and best practices

Security and best practises

With the introduction of a new version of IP protocol comes the requirement to implement security rules at various level. Many of these rules are close to IPv4 best practices, others are specific to IPv6.

It is especially important to master IPv4 security mechanisms as well as IPv6 nuts and bolt, particularly addresses assignment modes and Neighbor Discovery Protocol (RFC 4861) operations before starting this section.

To ease the assignments of rule transposition studies on your devices, they are divided into the following scopes:

- TRANSIT for any L3 device (router, FW, etc.);
- ACCESS deals with the last mile, host bearers' devices and their particularities. And dwells specifically on NDP and interactions with OSI Layer 2 (MAC);
- HOST relate to endpoints, mainly computers and servers;
- FILTERING for Firewalls.

Access layer

• DYNAMIC ADDRESSES ASSIGNMENT

Endpoint tracking within an organization leads to prefer DHCPv6 stateful for its device tracking ability rather than SLAAC based auto-assignment. Servers may use static addressing. Some devices, mainly Android based, do not support DHCP, sometime leading to exceptions. Refer to the dedicated chapter.

Let's start by reminding that the use of Router Advertisement is mandatory, even with DHCPv6 stateful. This RA will provide at least gateway link-local address and timers. Only a fully and not recommended manual setup allows to do without RA.

Mechanisms

Several bits are used to indicate hosts how to behave:

- A – "Autonomous Address Autoconfiguration" tells the host to auto configure (SLAAC RFC 4862) ;
- M – "Managed Address Config" indicates on the opposite to use DHCPv6 stateful (RFC 3315) to obtain its address ;
- O – "Other Config" announces that other options can be fetched from a stateless DHCPv6 (RFC 3736), these options may include DNS servers, NTP, domain suffix, etc. ;
- L – "On-Link" notifies that the RA and the announced prefix are on the same L2 link. This bit gets reset to 0 if the advertisement crosses a router, to avoid side effects of misconfigured L2 extension or unintentional relaying. Unlike IPv4, an IPv6 host does not natively consider that a host located in the same prefix as it can be reached directly in L2 (except for link-local addresses). See RFC 5942.

Note that the A and L bits are sent within the prefix, not in the RA core.

Method	Bit A Auto	Bit M Managed	Bit O Other	Resulting Addresses	Options supply
SLAAC	1	0	0	-Temporary SLAAC (Ephemeral) -Link-Local	RDNSS RFC 6106
Stateless DHCPv6	1	0	1	-Temporary SLAAC (Ephemeral) -Link-Local	DHCPv6
Stateful DHCPv6	0	1	Redundant with M	-DHCPv6 -Link-Local	DHCPv6

Method	Bit A Auto	Bit M Managed	Bit O Other	Resulting Addresses	Options supply
DHCPv6 Stateful + SLAAC (host OS choice)	1	1	Redundant with M	-OS choice -Link-Local	DHCPv6 or RDNSS depending on OS

The 3rd method (Stateful DHCP) is ideal thanks to its device tracking ability.

Double check you don't accidentally fall into the last case, where things are not predictable.

Indeed, when a system is presented to an RA with A and M bits set to 1, most OS configure both and use only one of the methods for outgoing connections (mainly SLAAC temporary address + privacy extension) or let RFC 7608 deciding by comparing the longest matching host's address to the destination. Use only if you like lottery. This is a common behavior of domestic routers. RFC 4862 mentions in its section 5.6 that it is the most recently obtained information that should be used.

It is possible to provide multiple addresses in different prefixes to a client via DHCP, this case is not covered here.

Interesting point, if you ever deploy a specific infrastructure with the 2nd case (Stateless DHCP), then your servers won't have to synchronize leases but only option's configuration for each prefix.

Note that some OS like Windows will send a DHCP request even if the RA indicates to do SLAAC without flag option.

DHCP Identification

DHCPv6 does not rely on MAC address as in IPv6, instead the host provides an identifier named DUID. A section details this identifier later in the Hosts section of the security chapter.

DHCPv6 provides options that exist in IPv4 as sub-options 82 and introduce some new.

- Vendor class (Option 16) allows the client device to send its manufacturer, model, version, etc.;
- Vendor Specific (Option 17) for proprietary options;
- Interface-ID (Option 18) which allows to identify the name of an interface and the VLAN. (circuit-ID in DHCPv4);
- Remote-ID (Option 37) RFC 4649 which can retrieve the physical port, the provided user ID to a VPN, and notably the MAC;
- Subscriber-ID (Option 38) is rather used by operators for other identification information.

Due to language abuses these options are often referred to as option 82 also for DHCPv6, while option 82 is the one in DHCPv4.

It is possible to put the MAC of the client in the Remote-ID option with access devices (switchs, AP, etc.). This is important, as it will allow to gather hosts' MAC address.

Other recommendations relating to DHCPv6 to facilitate terminal identification can be found in the Hosts section.

• ICMP REDIRECT BLOCKING

Neighbor Dicovery Protocol features 5 message types:

- *Router Solicitation* and *Advertisement*;
- *Neighbor Solicitation* and *Advertisement*;
- *Redirect.*

This latest message type allows the gateway to indicate that another router is being used to reach a given destination and that the host should update its routing table accordingly.

ICMP redirect (Type 137) should be blocked, as it may allow an attacker to redirect traffic. This option should only be used when a network segment has two routers that reach different resources; a very rare case.

• IPv6 SNOOPING

Let’s start by briefly recalling the purpose of the two most frequent message types within NDP.

The Neighbor Solicitation (135) and Advertisement (136) messages are used to establish the link with layer 2 within a network segment, typically asking what the MAC address of a host based on its IP and responding. Like ARP in IPv4.

The request is done in multicast, a unicast mode also allows to check that a host is still reachable, in this last case we specify who is asking (Target Address).

When this address is not specified (::/128), the message is a DAD (Duplicate Address Detection)

The response to a NS has an "Override" O bit which is set to 1 by default to specify to overwrite any existing entry in an ND cache. The RFC indicates that setting the bit to 0 is intended for proxified responses to solicitations, or for anycast service addresses.

In practical terms the following 2 examples:

- An ND proxy (ARP proxy equivalent) will not overwrite via its response a direct response that the concerned host should have sent directly;
- 2 servers with the same anycast address in a segment will not try to overwrite the entries concerning them.

The S bit "Solicited" specifies that the response is intended for a unicast request with Target Address, i.e. a reachability request.

Finally, the R "Router" bit indicates that the host is a router. If it is set to 0, Neighbor Unreachability Detection will deduce that the host is no longer capable of routing. It will then initiate a router solicitation and will switch to any other available router (based upon priorities if several are

present).

Before we even talk about Router Solicitation and Advertisement, you'll have already noticed what an attacker can do with the NDP neighbor information. It is therefore highly recommended to implement appropriate anti-spoofing mechanisms at least on campus/user site access layer infrastructure.

NDP operates multicast groups called Solicited-Node Multicast, each host will create a multicast group for each address assigned using a standardized prefix FF02:0:0:0:1:FF00::/104 and the last 24 bits of the address to represent. These multicast addresses are used for DAD, but also to perform MAC/IP matching without disturbing everyone like the ARP broadcast does in IPv4.

The first contact between two IPv6 nodes in the same network is therefore always a multicast.

ND Fragmentation

RA messages can be large if they contain many prefixes and thus, require fragmentation, RFC 6980 states that it is then better to send several messages rather than fragmenting the packet. In any case, except for a particular configuration, there is no reason to have that many prefixes and options in an RA leading to 1280 bytes, the IPv6 minimum.

This leads to the recommendation to block NDP protocol fragments.

Binding

Security mechanisms are based on the constitution of a relationship table relation among IP, MAC and physical location, typically the switch port.

The simplest way is to use DHCP snooping, which leverages the IPv6 assignment messages returned by DHCPv6 to build a control table called binding table.

ND, DHCPv6, and other inspections are not always implemented in the most correct way. Some work perfectly with header extensions and even fragmentation. Others only operate when facing the simplest case. This discrepancy is often due to the device ASIC capabilities. On some product lines, the process even involves control plane and is incompatible with hardware optimization options.

On the configuration level, these features may be part a uniform package, otherwise the sum of several options to enable independently, and even sometimes co-exist (all-in-one + separate) and enabling one type remove the other.

Therefore, check the manufacturer's documentation carefully and test with a package forger like scapy.

Every alert/security event related to this set of rules should raise an alert in your SIEM.

Remember to setup binding table recovery so that it is immediately refilled when the switch is restarted. It is usually possible to export it periodically and/or fetch active leases from the DHCP

server (if you rely on DHCP stateful obviously).

It is possible to use part of these security features without DHCP, however loss of a secure learning source affects the level of protection (RFC 6620). Nowadays we are seeing organizations where DHCP is used in conjunction with static leased within the datacenter to provide this level of security on server hosting access layer. No more manual host configuration to do.

Without DHCP the device will build the table based on the exchanged DAD messages during the SLAAC auto-assignment.

Note that in L3 fabric-based solutions, signaling protocol carries information needed to build the table, inspection is only required for specific configurations. For example, on an EVPN+VxLAN infrastructure, EVPN type 2 routes already advertise the MAC/IP pair.

Various controls can then be derived from this table, here are the main ones:

Source

A packet with an unknown, unallocated source address will be dropped. The switch can try to ask the DHCP server and/or its neighbor via NDP if the address is known before discarding the traffic.

This check requires the presence of a binding table, it does not perform ND inspection itself.

Don’t forget to allow traffic on local link address, sometimes via an additional command and to flag trust ports for static resources like manually addressed servers.

Destination

When a packet arrives, the device will transmit it and perform ND resolution, if necessary, only if the recipient is known in the binding table. Otherwise, the packet will be discarded.

This mechanism allows to counter traffic to a malformed or non-existent address, for example for local denial of service purposes.

Move

When a host moves to a different port, physical location tracking can initiate an ND solicitation to the host on the previously known position in the binding table. If a response is obtained, then the newcomer is a spoofer.

This makes the attack ineffective to the extent that the original host is online and able to reply.

ND suppress

To optimize traffic and limit multicast, it is possible to let the access device respond to NS Neighbor Solicitation requests instead of the concerned host. This feature can be enabled at least for multicast requests, but also for unicast. ND/ARP suppress is a common feature on EVPN/VxLAN fabrics (where learning is done differently) but it can also be found on some campus products too.

However, remember that one of the uses of unicast requests, (the one with the Target Address), is to

check the reachability of a host. It is therefore not relevant to reply in the name of the host for anything else than multicast unless the device distinguishes between unicast requests with and without target address, and only acts for the last case.

In other words, a device should never have to send a Neighbor Advertisement with the S bit set to 1 instead of the host.

A possible exception is wifi, where the monitoring of the radio link with the station by the access point can authorize answering instead of the station even for a reachability test. Priority being given to a less than possible chatty underlying media, the radio channel.

Prefix

Based on information obtained from the following sources:

- Router Advertisement;
- DHCP-Prefix-Delegation;
- Manual configuration if required.

Prefix control allows you to block a packet whose source routable address does not belong to the prefix in use in the L2 segment. Thus, address spoofing is blocked at the access layer, even before using URPF later during routing, for example.

Cache poisoning

Like its predecessor ARP cache poisoning, it is possible to fulfill the ND cache of hosts leading to saturation. Especially with 2^64 possible addresses in a network, an attacker has plenty to do.

One common attack is to impersonate the router in a Neighbor Advertisement with the R bit set to 0, indicating that the route is no longer being used. The attacker can also attempt a man-in-the-middle by impersonating a host or the router.

Binding security prevents this behavior, but it is still recommended to specify a cache size limit on network devices. If you want to calculate a fine-grained limit, remember that it is not enough to count hosts but addresses. Each host has at least 2 and can have more (SLAAC with temporary addresses for example). Modern OS usually have acceptable defaults values.

For more information, see RFC 6583.

• DHCP ROGUE

Physical

Described in RFC 7610, the DHCPShield mechanism involves defining the physical ports that can receive DHCP server traffic. Generally, uplink ports. DHCP traffic from undefined ports will be discarded.

The device will have to analyze the whole content of any message coming from the DHCP server. Again, be careful depending on the ASIC and implementation.

If the device doesn't support the feature, it is still possible to use an ACL blocking traffic matching source port UDP 547 / destination UDP 546, but it won't work with a forged fragmented packet.

Logical

The extensive RFC 8415 that covers DHCPv6 includes a section about securing exchanges between the server and clients and/or relays.

IPsec can be used to authenticate or even cipher DHCP exchanges between servers and relays, RFC 8213. The cryptographic configuration can be set manually or based on a PKI.

Use of IPsec can also protect other administration traffic such as Syslog, SNMP, NTP, RADIUS, etc.

Caution, the support of IKEv2 with pre-shared secrets is not mandatory in this RFC.

The use of a simple shared key allows an attacker to replay packets. RDM limits the risks of replay but only on the client side, and not between a relay and the server.

Many obsoletes RFCs suggested other authentication mechanisms. Today, RFC 7227 dealing with the implementation of DHCP options is the foundation of many proposals. You can read about the DHCPv6Sec and Secure-DHCPv6 projects.

Last security element available in RFC 8415, RKAP (Reconfiguration Key Authentication Protocol) prevents the reconfiguration of a client by a malicious server. A unique key is sent to the client during the first response. The server then uses HMAC-MD5 to sign its messages.

However, RKAP is recent and is not yet usable in practice.

By the way, reconfiguration is a new feature that allows you to force clients to request DHCP again (without waiting for their lease to expire or for a reboot). Those of you who ever had to reboot hundreds of PoE devices to make them take into account a new option via DHCP will welcome this feature. Don't take the opportunity to DDoS yourself by trying this new feature on a too large scope…

In short, implement IPsec between your relays and the server, and let the DHCPShield component handle the security of the relay/client side only from a port of arrival point of view of authorized DHCP server messages.

Lastly, always remember that DHCPv6 can provide multiple IPv6s to the same client (DUID).

• RA GUARD

Router Advertisement messages are a key point of IPv6, it is necessary to ensure that they are issued by an authorized router.

RFC 6105 recommends to manually set one or more of the following elements in access devices in order to validate or block a RA message:

- physical port;
- MAC address of the router;

- gateway IP;
- advertised prefix;
- RA priority;
- Hop-Count limit;
- value of the M - Managed and O - Other bits.

The simplest way is generally to allow uplink interfaces, note that it is also often possible to impose a TTL limit.

The RFC also proposes a so-called stateful learning mode, during which the equipment would learn the RA source(s) for a given period. After that, it would not accept any new RA source.

This stateful mode is starting to be implemented in devices.

Note that if the router switches to a twin using an NHRP type protocol, it will be necessary to ensure that the absence of a memorized neighbor will cause the unit to fall back into a learning state, or that the controlled elements do not change (a Virtual MAC or IP for example).

If the equipment does not support RA guard you can at least block RA ingress with an ACL on the access ports.

• RA HOP LIMIT

To prevent a Router Advertisement from jumping out of the segment, section 6.1 of RFC 4861 reminds us of the basic controls to do on ND messages. Such checks as RA destruction with a hop-limit lower than 255 should work automatically, without any specific security configuration. The ND Shield draft https://tools.ietf.org/html/draft-gont-opsec-ipv6-nd-shield-00 proposes to go further.

This security may remind you of what exists in BGP with GTSM (Generalized TTL Security Mechanisms) RFC 5082. GTSM will discard a BGP message if its TTL/hop-limit is lower than 254 because this time for sure, it does not come from the neighbor (Except when having the BGP multihop option of course).

Don’t forget to adapt the configuration of intermediate devices so that they don’t voluntarily decrement the hop-limit in some particular configurations like an L2 network extension or simply if you use L3 datacenter switches in MLAG.

Be careful, some manufacturers' documentations mention editing the value of the RA hop-limit and often give a value of 64 by default. This is in fact the *current hop-limit* (CHL) field which indicates to hosts receiving the RA the hop-limit value to configure on their side.

• OTHER RA SETTINGS

After having seen the specific points on security and address assignment modes, let’s see some of the other settings of the router advertisement. These parameters must be configured on each interface.

- RA interval: delay in seconds between 2 unsolicited RA transmissions, with a minimum and

maximum value.

  - The maximum must be between 4 and 1800. The default is 600s ;
  - The minimum must be between 3s and ¾ of the maximum value. The default value is 1/3 of the max, or 3s if the max is less than 9s.

- RA lifetime: lifetime beyond which the router is considered to no longer be used. The value must lie between the MAX interval and 9000 seconds. The default is 3 x interval max.
  - A value of 0 indicates that the router is not to be used by default;
  - In the case of a point-to-point interconnection between 2 routers, e.g. BGP peering, the RA lifetime will normally be ignored, the lifetime of the neighbor being monitored via the routing protocol itself.
- MTU: it is possible to provide the MTU of the link to the hosts, the default value is 0.
  - If you encounter problems with Path-MTU-D on a site, you can temporarily set this value to handle the problem in the outbound direction while you identify the problem. This is faster than configuring each individual host.
- Prefix: the router announces one or more routable prefixes, each with:
  - *Lifetime* : route's lifetime, which can be specified in seconds since the last announcement, or via a fixed time. This last option can be used to cleanly decommission a prefix before removing it from the configuration. The default value is 2592000 seconds remaining, or 30 days. It is not recommended to use the value 0xffffffff which has the effect of making the route permanently valid, a good way to have a black hole if the router changes its local link address;
  - *On-Link* (Bit L) : already mentioned above, it indicates that the router is on the link, 1 by default.
- SLAAC
  - *Lifetime* : the preferred duration of validity of the addresses that the hosts autoconfigure, again this can be configured in seconds remaining or with a fixed date/time. The default is 7 days (604800 s). And here too, it is not recommended to use infinity (0xffffffff). Finally, note that the value must not be greater than the validity of the route of the associated prefix;
  - If you are not using DHCP stateless with SLAAC, you can specify DNS server addresses via RDDNS (Mandatory for Android).
- Priority
  - Router priority can be Low, Normal (default) or High. You may use it whenever you need to switch gateways seamlessly without even requiring keeping the same IP. It can be a good practice to keep it set to "high" to reduce an unwanted override risk.

Other fields exist in the RFC but they are not used and not configurable on most platforms (Reachable Time and Retransmit Time).

Good to know, vendors implement a status command to display all prefixes issued with the associated interface.

• seND (NOT USABLE)

Secure Neighbor Discovery is designed to authenticate NDP messages within an organization and was originally described in RFC 3971.

The protocol is based on:

- Addresses generated from an RSA cryptographic database (CGA) RFC 3972;
- PKI and anchor point;
- Pseudo random clock and nonce (anti replay).

When a host connects, the router will indicate the certification path and the “trust anchor”, this leads to a 6th type of ND message, the *Certificate Path Solicitation*. See RFC 6494 on certificate profiles and management and RFC 6495, X.509 fields.

Having certificates implies an increased message weight and new risks linked to fragmentation, see RFC 6980.

When one looks at the RFC in detail, one realizes that problems similar to those of 802.1x exist. If the RFC starts by reminding us that IPsec was not viable because NDP is the first contact with a network, there is no remediation system as there is in 802.1x.

The host must have pre-configured at least one trust anchor.

Network devices are starting to implement SeND, but there is still no support for SeND in operating systems outside of a few academic projects.

SeND is therefore unfortunately not usable at this time, and can only be used within an organization with managed workstations, like 802.1x.

• MLD

IPv6 runs naturally in multicast, whereas it is rarely used in an IPv4 network. It is often limited to discovery protocols such as mDNS, SSDP, LLMNR or even when implementing OSPF.

As a result, multicast is not always well implemented within a network segment. We are not even talking about multicast routing here, but just exchanges on the same L2 segment.

MLDv1 (RFC 2710) is the equivalent of IGMPv2 and uses 3 types of messages:

- *Listener Queries*, either general to ask all nodes if they are members of at least one multicast group, or specific to identify the members of a group based on a specific address;
- *Listener Reports* to have hosts answer requests;
- *Done* to inform that they no longer need to be part of a group.

MLDv2 (RFC 3810) builds on IGMPv3 and adds source filtering (SSM), so that sources can be included or excluded.

Hosts send reports on state changes in addition to periodic reports and the "done" message type disappears (taken over by the state change).

The messages are retransmitted to make the set robust to packet loss, a robustness variable indicates how many times messages should be retransmitted. The default value is 2, it can be useful to increase it on wifi for example.

MLDv2 is backwards compatible with MLDv1, note that it is on top of ICMPv6, unlike IGMP which is directly on top of IPv4.

MLD thus allows to know clients' needs, in particular to forward them to the PIM agent in the case of routed multicast. However, without any other mechanism, multicast traffic behaves like broadcast traffic within the network segment. It is sent to all ports.

MLD snooping optimizes multicast traffic delivery by sending it only to hosts requesting it and to routers providing the service. L2 devices will analyze the content of MLD exchanges in order to build tables matching ports and multicast addresses. In MLDv1 this association is based on the destination multicast address, in MLDv2 source address(es) are added to it, SSM is required.

It is therefore important that the MLD querier feature is active on the router (mrouter), and that the L2 devices use the MLD reports to perform snooping. Without « mrouter », state is replicated on all switchs which is unwanted.

With MLD, if multiple routers try to query, the one with the smallest link-local IP becomes the querier. This small optimization avoids the problems sometimes encountered in IPv4 with IGMP where the winner is the one that queries the most frequently.

Don’t neglect the optimization provided by snooping and check that it is working properly on the whole circuit. Take the opportunity to check IGMP on IPv4 at the same time.

In dense datacenter environments, take the time to consider the distribution of the underlying multicast trees in EVPN+VxLAN fabrics. The best practice is generally to distribute networks on at least 2 underlay trees, and to create dedicated trees for networks with intensive multicast hosts (cluster, video transmitter, etc.). This practice can also prevail on other overlay/underlay based topologies.

In summary, although MLDv2 is technically only required when using SSM, its ability to tolerate the loss of at least 1 packet is an advantage over V1 (see robustness value). Snooping is an optimization requirement that also avoids an attack via unknown multicast addresses or without client hosts.

When talking about IPv6 and multicast, we immediately think about Well-Know Multicast groups, like “all routers” (ff02::2) or “all DHCP servers” (ff02::1:2). We however forget Solicited-Node Multicast which we’ve already dealed with.

To refresh your memory, each host will create a multicast group address based upon the last 24 bits of each configured address and the F02:0:0:0:0:1:FF00::/104 prefixe. Thoses addresses must not be processed by MLD snooping, as they could fastly overload tables (with at least one group per host). This bypass is sometime enabled by default, sometime needing to

> **apply a command such as nd-workaround on MLD snooping configuration. Check with your vendor and have a glance to the content of MLD snooping content while hosts communicate.**

• STORM CONTROL

More classical and simple security, implement storm control for multicast and unknown traffic at least on access devices uplinks. The 3rd about broadcast only concerns IPv4.

Be aware that it is still better to have a high value like 30% of the link than no configuration at all, while waiting to refine it after studying the traffic.

• MULTICAST GROUPS TO BLOCK

There are some multicast addresses to block directly on access devices. You can find them in the section "Disabling auto-discovery protocols" of the Host part.

Host

Besides rare exceptions (firewall with profile), settings you apply to a host take effect regardless of the network it is connected to. Unfortunately, it is not possible to create profiles, for example disable SLAAC on host side when the prefix received in the RA is the company prefix.

Therefore, be careful especially for machines that may connect to networks outside your organization. For example, a user with a laptop at home will have a hard time doing anything if the administrator has completely disabled SLAAC.

On the other hand, you can harden the servers as much as possible.

• DHCP

DHCP DUID

DHCP Unique IDentifier allows the DHCP server to identify the client and track its lease. There are several methods of constructing this identifier, the simplest being the hardware address (MAC).

This DUID is normally persistent within a system regardless of the network interface. For example, a laptop with a DUID built from the MAC of its wired ethernet card will use the same value when making a request via the wifi card.

The possible construction sources in the initial RFC 8415 are:

- *Link-Layer Address* (DUID-LL);
- *Link-Layer Address Plus Time* (DUID-LLT);
- *Vendor Based on Enterprise Number* (DUID-EN);
- *Universally Unique Identifier* (DUID-UUID) RFC 6355.

The first one is explicit, the 2nd one adds the clock the day of the first generation, it is stored and does not change, remember this.

The 3rd is at the choice of the manufacturer.

The 4th one, UUID, tries to guarantee the persistence for a system starting from the network or in several phases. Starting a server in PXE with a light bootstrapper that then switches to a heavy OS is an interesting case:

It has several interfaces so we cannot guarantee that the DUID-LL is based on the same interface. The vendor is different between the firmware of the PXE card, the light bootstrapper and the OS.

The UUID can be tracked consistently if the whole chain is based on the same information, e.g., the system serial number known by the UEFI.

Most OS use DUID-LLT by default, there is no reason to change it.

DHCP Identity Associations

While a DUID is unique for a system, the Identity Association is unique for a given interface. No particular configuration here.

DHCP without RA

If the *Router Advertisement* indicates whether or not to use DHCPv6, what to do when there is no RA?

RFC 4862 states that in the absence of RA, a system can do DHCP. This is implemented in most OS. Good to know too that some OS send DHCPv6 requests even when told to do only SLAAC by the router.

DHCP options support in Dual-Stack

In the series of non-predictive behaviors, what happens if a dual-stack host receives specific options in both DHCPv4 and v6 and those options differ in content?

Is it precedence that prevails - the first one providing the option? It might be interesting to check this.

• SLAAC ADDRESS GENERATION METHOD

Originally it was planned that the SLAAC address would be formed from the system MAC address in the form of EUI-64. However, this raises many problems:

- Since the MAC is unique it becomes possible to track a host on the Internet regardless of the network from which it connects;
- It is easier to run an address scan on a network, as the use of EUI-64 offers a certain predictability of what can be found frequently on the first bits;
- Knowing the MAC allows you to know the vendor, so it becomes possible, for example, to guess which brand and model of device you are talking to by correlating the vendor and the protocol used during the exchange;
- Changing the network interface will change the SLAAC address.

2 RFC propose approaches to limit these problems, see:

- RFC 4941 *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*;
- RFC 7217 *A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration* (SLAAC).

Temporary address

The temporary address is an addition to the stable address (RFC 4941). It changes more or less frequently depending on the OS settings while respecting the lifetimes announced by the Router Advertisement SLAAC.

For example, some systems create a new address every 25 minutes, and completely unconfigure the previous one 5 minutes after its replacement is created and if no session exist with the oldest temporary IP. Thus, new host-initiated sessions never use an address for more than 30 minutes.

However, the host remains reachable at all times via its stable address, and only the stable address is subject to DNS self-registration.

The use of temporary addresses can cause problems because of their short life.

The RFC mentions the case of a server checking that a PTR reverse DNS record exists for the client before allowing access. But it is easy to find much more common cases:

Let's imagine authenticating on a website to access a client space while using a temporary address at its 24th minute of activity.

2 minutes later the server asks us again to authenticate while we have been browsing continuously since the connection.

This case is quite plausible, if for a security reason the server asks the client to have the same IP in addition to its cookie, it will reject the session. Similarly, if a front-end L4 load balancer starts redirecting the client to another server that does not know about the client's web session because it thinks it is dealing with a new client due to a new IP. There is currently no mechanism allowing browsers to communicate to a server for which a browser tab is active (or recently active) the IP change information.

Similarly, a P2P online game with self-hosted matchmaking could see its games interrupted after a few minutes.

In the case of a game, it would be desirable for the developer to take care of mounting sessions via the stable address, but for a browser this would completely negate the value of the temporary address, as web traffic represents the majority of the tracking possibilities.

If we take a step back, we can say that tracking (advertising for example) will be satisfied with identifying the /64, which is enough to identify a household in the same way as an IPv4 today. But it is not impossible that advertisers will start to cache IPv6s over a week to mark as stable those seen several times, thus necessarily using an EUI-64 or Stable privacy address. This finally gives them the possibility to track the single user instead of the household, and without cookies! To be thought about…

Quite recently, in February 2021, RFC 8981 made changes to temporary addresses.

In the list of changes, we find the ability to have only temporary addresses, no more stable. The RFC still does not impose a mechanism to exclude prefixes from the use of temporary addresses, but it recommends it. Microsoft's answer might not change https://social.technet.microsoft.com/Forums/azure/en-US/e36e82e9-1911-4f4d-91a2-c62f6e04c9c1/ipv6-turn-off-privacy-extensions-temporary-addresses-for-certain-prefixes-ie-ula-in-win-10?forum=win10itpronetworking

Randomized interface ID

Rather than use its MAC in EUI-64, host will generate its address based on a pseudo-random

identifier. This identifier changes on reboot, so systems that support storage persistence will base their address on the previous address in addition to the pseudorandom number.

Stable privacy address

This mechanism allows you to always get the same IPv6 address as long as you are on the same network, without keeping it when connecting to other networks. This is achieved thanks to the fact that it is derived from intrinsic constants of the host alongside the received prefix.

Specifically the following:

- Prefix received via RA;
- Interface number (as seen by the OS);
- DAD counter (0, increments if conflict);
- Secret key randomly generated the first time and stored;
- Optionally the network identifier, typically the Wifi SSID.

Thus, it is impossible to follow the machine when it moves on different networks, impossible also to find the MAC from the address. On the other hand, the stable aspect within each frequented network will facilitate the work of the administrator who wishes to avoid DHCPv6 stateful.

SLAAC synthesis

Here is a summary of the trackability by type of address. Don't forget that the global address is routable and therefore potentially visible absolutely everywhere on the internet.

SLAAC mode	**Local tracking**	**Globale tracking**	**Information about device**	**Tracking from the same network over time**
EUI-64 (MAC)	YES	YES	YES (vendor)	YES
Randomized (change on reboot)	NO	NO	NO	Over several hours/day depending on standby VS reboot
Stable Privacy (derivated from prefix)	YES	NO	NO	YES
Supplement Temporary	NO (For host-initiated session)	NO (For host-initiated session)	NO (For host-initiated session)	Usually less than a day (For host-initiated session)

Ideally you should keep the default OS behavior for machines that may connect outside the company. This behavior generally varies between Randomized or Stable Privacy, with or without Temporary.

For other machines, it is possible to completely disable SLAAC, as the use of DHCPv6 stateful and/or manual configuration (of servers for example) makes this mechanism useless. We then follow the logic of reducing the protocol attack surface and close the door.

Link-Local address generation method

Although only local in scope, the local link address also benefits from the 3 different automatic configuration modes mentioned above.

The configuration generally follows that of the global address on consumer OSes, few systems offer a specific configuration granularity according to address classes.

However, server and network-oriented systems' ones are generally based on EUI-64.

• DON'T DISABLE IPv6 STACK

If for some reason you want to avoid a host to communicate in IPv6, do not disable its IPv6 stack. Instead, use the following options:

- Change the precedence to prioritize IPv4;
- Disable SLAAC on the host and ban it from DHCP if necessary;
- Set the OS firewall to disallow all IPv6 traffic.

If you disable the IPv6 stack, you may encounter anomalies with some programs. For example, Windows has required for several years not to disable IPv6 completely at the risk of not being able to run some of its commonly used components. Under Linux the simple absence of the loopback ::1 can also bring its share of surprises. Usually recent kernel let you use ::1 loopback even with disabled stack.

• DISABLING TRANSITION MECHANISMS

Some mechanisms allow hosts to exchange in IPv6 through IPv4 networks, notably:

- TEREDO;
- ISATAP;
- 6to4.

These mechanisms are no longer of interest and the first two have even disappeared. It is therefore advisable to turn them off.

• DISABLING AUTO-DISCOVERY PROTOCOLS

It is advisable to disable auto-discovery protocols embedded in the OS. If they are useful in a domestic environment, they represent a real risk in a corporation.

This includes :

- SSDP (multi OS, ff02::c – UDP 1900) and following addresses FF0X::C, depending on the scope:
- Node-local : FF01::C (doesn't even come out…)
  - Link-local : FF02::C ;
  - Site-local : FF05::C (deprecated);
  - Organization-local : FF08::C (deprecated);
  - Global : FF0E::C.
- mDNS (multi OS, ff02:fb – UDP 5053)
- LLMNR (Windows, ff02::1:3 – UDP and TCP 5355)

Beyond attacks related to these protocols, their operation with IPv6 differs on a very particular point.

In IPv4, a machine has only one IP. If 2 machines start talking to each other after having resolved their name via one of these protocols, the IP/machine mapping is still kept via DHCP logs typically.

In IPv6, these protocols allow machines to resolve each other via their link-local address. (FE80::/10). So go and find out in a log which one was a FE80…

This behavior exists in production in organizations that have not even deployed IPv6. For example, it is enough to have an SMTP relay between 2 Microsoft Exchange servers located on the same network segment. If the above protocols are not disabled, you will see in the mail headers a delivery via FE80. Fortunately SMTP still indicates the hostname.

• BLOCKING LINK-LOCAL TRAFFIC

At home the local link address can be used to talk to your NAS, printer, chromecast/airplay receiver… after discovery via the above-mentioned protocols. The DNS auto-registration on its domestic router will make prefer the global address.

But in a corporation, a host has no reason to do anything else than ICMP (and protocols based on it like MLD) via its local link address. It is therefore recommended to block all TCP and UDP traffic in both directions within the OS firewall. But keep ICMP allowed, as said.

Beware, in the case of clustered servers it is quite possible that a software solution requiring the machines to be in the same network segment uses the local-link addresses to exchange data, or simply for the heartbeat.

Make an exception for DHCP and EAPOL 802.1x on systems that use them.

For mobile devices, it is also interesting to open NAT-PMP (RFC 6886) and its successor PCP V2 (RFC 6887) in order to allow the operation of applications that need to receive unsolicited traffic. Typically, some conference systems. These 2 protocols allow to ask the gateway to open a port, the equivalent of the NAT44 port auto-redirection in IPv4 via UPnP-IGD.

NAT-PMP initially used port 5351 on both sides, but this caused problems for machines that were both clients and servers, such as when re-sharing a connection. Therefore, the clients migrated to port 5350. PCP also uses 5350 on the client side and 5351 on the server side.

So, we will keep UDP 5350 and 5351 in listening and 5351 in destination.

For less constraint you can also choose to block only the traffic in the incoming direction.

• VPN

The introduction of IPv6 in home networks can present a risk for misconfigured VPN sessions. A company not practicing split tunneling and advertising the route 0.0.0.0/0 will be able to let the host communicate directly with the outside world if it can resolve AAAA DNS resources and the firewall does not block it.

Resolution is possible if the company's DNS server responds to AAAA requests, even over IPv4 connectivity, or if the host's stack allows resolutions to be done via IPv6 DNS locally provided to the host in VPN.

If you use split tunneling, make sure that the IPv4 and IPv6 rules match.

Many sites allow you to do an IPv6 VPN leak test.

Note for "consumer" VPNs, they rarely support IPv6 but still announce a default IPv6 route to send traffic to a blackhole and avoid a leak. You can do the same thing and advertise ::/0 on your VPN even if you don't provide real connectivity.

• DESKTOP OS CONFIGURATION

This section gives some configuration examples.

Windows

Under Windows, even if *netsh* commands still exist, it is now advised to use *powershell cmdlets*.

Most of the configuration can be found here:

https://docs.microsoft.com/en-us/powershell/module/nettcpip/set-netipv6protocol?

Some of the configurations might be done directly in the registry, such as DHCP DUID generation method, trhough key HKLM\SYSTEM\CurrentControlSet\services\TCPIP6\Parameters\Dhcpv6DUID

0001 – DUID-TTL

0002 – DUID-EN

0003 – DUID-LL

Persistent DUID is shown under the same key.

Linux

Here are some configurations for GNU/Linux.

Some are always applied at kernel level, either directly or using a third-party tool.

The rest depends on the packages in charge of the relevant features. Since the GNU ecosystem is by definition rich and open, there are many ways to do things, even within the same distribution. The official documentation of the distributions is not always aligned.

Configurations can be done via:

- Commands;
- Configuration files;
- Pseudographic tool like nmtui (for Network Manager).

Following are links to the kernel documentation :

https://www.kernel.org/doc/Documentation/networking/ipv6.txt

https://www.kernel.org/doc/Documentation/networking/ip-sysctl.txt

https://github.com/torvalds/linux/blob/master/net/ipv6/Kconfig

A more readable resume https://sysctl-explorer.net/net/ipv6/

Network Manager

Network Manager is a fairly common tool from the Gnome project used to manage networking.

https://wiki.gnome.org/Projects/NetworkManager

https://developer.gnome.org/NetworkManager/stable/settings-ipv6.html

https://developer.gnome.org/NetworkManager/stable/nm-settings-ifcfg-rh.html

https://developer.gnome.org/NetworkManager/stable/nm-settings-keyfile.html

CLI nmcli https://developer.gnome.org/NetworkManager/stable/nmcli.html

Pseudographic nmtui https://developer.gnome.org/NetworkManager/stable/nmtui.html

Systemd Networkd

systemd-networkd (network) and systemd-resolved (DNS) are omnipresent but not necessarily enabled. Be sure to disable global management (or management of certain interfaces) by another daemon such as Network-Manager to avoid conflicts with Networkd. The opposite is also true.

https://systemd.io/

https://www.freedesktop.org/software/systemd/man/resolvconf.html#

https://www.freedesktop.org/software/systemd/man/systemd-networkd.service.html#

https://www.freedesktop.org/software/systemd/man/systemd.network.html# (the most important)

netplan

Netplan is not a direct management daemon, but an abstraction tool present at canonical (Ubuntu). It then configures Network Manager or Networkd.

https://netplan.io

https://netplan.io/reference/

However, Netplan seems to lack of DHCP-PD support, which is a big downside for some uses (like when willing to provide /64 to hypervisors pods). In the meantime, you can use it with a systemd override on this element.

https://bugs.launchpad.net/netplan/+bug/1771886

wickedd

wickedd provides a service for managing network interfaces. It monitors the system's interfaces by retrieving relevant information from the kernel via netlink, sysfs and other interfaces.

It can be accessed via a DBus service, which can be used to reconfigure interfaces, bring them up or take them down.

Components

Additionally to the main wickedd daemon, wicked framework provides several helper daemons and supplicants:

Component
Description
wickedd-nanny
Event driven policy daemon responsible for hotplugging.
wickedd-dhcp6
DHCPv6 client supplicant
wickedd-dhcp4
DHCPv4 client supplicant
wickedd-auto4
IPv4 autoip supplicant

Further, it communicates also with external wpa-supplicant for Wireless (WPA) support. wickedd is used by suse distribution.

https://manpages.opensuse.org/Tumbleweed/wicked/wickedd.8.en.html

By distribution

Each distribution's documentation will instruct you which tool is in place by default. In most cases, the choice lies between systemd-networkd and Network-Manager. Conman and WICD, for example, have disappeared from the landscape.

As often, the ArchLinux documentation is very complete. Here is a link to the configuration elements for each type of network manager https://wiki.archlinux.org/title/Network_configuration#Network_managers

Also see IPv6 section https://wiki.archlinux.org/title/IPv6

Ubuntu netplan man http://manpages.ubuntu.com/manpages/jammy/man5/netplan.5.html

Lots of elements here http://mirrors.deepspace6.net/Linux+IPv6-HOWTO/

and http://www.bieringer.de/linux/IPv6/.

• MOBILE AND EMBEDDED

Mobile OSes can be found within an enterprise network in different forms:

- Embedded hardware (printer, room booking);
- Fleet of smartphones owned by the company;
- Enrolled personal smartphones (BYOD);
- Unmanaged devices on a guest network.

Android

Android is now the leading player in these segments, and it has one annoying problem, it does not support DHCPv6.

Surprising? This choice seems to be part of a trust strategy to impose the implementation of SLAAC. The reasons are given in RFC 7934, DHCP provides only one address and does not allow the use of temporary which facilitates tracing. Having only one address also prevents offering tethering/shared connections in wifi.

However, the demand is there, the problems mentioned do not make sense on a corporate network in Wifi. The problem of connection sharing only makes sense behind a 3GPP type mobile link.

But then who wrote this RFC? Engineers from Google and Apple, starting with Lorenzo Colitti.

The problem has been reported for many years:

https://www.techrepublic.com/article/androids-lack-of-dhcpv6-support-frustrates-enterprise-network-admins/

https://www.reddit.com/r/ipv6/comments/3wfpn2/i_am_getting_sick_of_lorenzos_attitude_to_ipv6/

https://www.nullzero.co.uk/android-does-not-support-dhcpv6-and-google-wont-fix-that/

https://issuetracker.google.com/issues/36949094

https://issuetracker.google.com/issues/36949085?pli=1

What to do? S ystematically ask for DHCPv6 support in your device RFPs. Whether it is a fleet of smartphones or embedded devices.

Android is enriched by the vendors well beyond the open-source OS project (AOSP), the OEMs sometimes integrate a DHCPv6 client. This is typically the case for Android printers/copiers, but rarely for phones.

How to track Android-based BYOD devices if they don't support DHCPv6? MDM (Mobile Device Management) tracking tools could provide the answer by tracing all the addresses used as long as they are part of a configured prefix list. For example, a /32 assigned by an RIR to the company. Thus, the terminal is only traced on the professional network, without using DHCPv6.

The same thing is possible for iOS, although it is easier for them to connect to an SSID without SLAAC and only DHCPv6. Not to mention forcing via MDM the use of the real MAC for this SSID and not a random MAC. Mobile OSes have recently been using random physical addresses not only when searching for SSIDs but also once connected.

Regarding guest networks, it is difficult to provide even a functional captive portal to a device using SLAAC that changes its temporary address several often.

A centralized captive portal will work with DHCPv6, too bad for Android. The implementation of an NDPmon collector could allow to follow a terminal in SLAAC, but these solutions are rare at the moment.

It is therefore delicate but not impossible to provide IPv6 SLAAC connectivity to guest networks in hotels, hospitals, airports or simply within an organization.

Other OSes

iOS supports both address assignment methods and does not present any particular problem in operation.

For other embedded devices, it will be good to ask for DHCPv6 support, but also to be able to choose the auto-address assignment mechanisms when using SLAAC. Typically, many microcontroller devices today use EUI-64 only SLAAC. This has the disadvantage of allowing an attacker to identify the brand via the MAC address, since the latter is included in IPv6. So think about asking for stable privacy IPv6 support.

Transit

• URPF

Unicast Reverse Path Forwarding (RFC 3704) prevents a packet whose source address does not match a known route in the reverse direction from traversing a router, thus limiting the risk of IP spoofing.

Several modes exist, depending on whether we focus on the match between the source interface and the best corresponding route (strict), any route that encompasses the address (Feasible) or whether we simply want to know if the router has at least one matching route regardless of the interface (loose).

RFC 8704 brings improvements based on BGP information to the feasible mode.

The implementation must be done on the edge portion of the network, where there is no risk of asymmetry. Typically, campus cores or exit routers. The configuration of uRPF is generally common to both IPv4 and IPv6.

If you are routing multicast traffic, consider multicast RPF as well.

• CONTROL PLANE PROTECTION

Packets destined to the router itself, as well as those with certain header options that cause an exception, must be forwarded to the control plane.

RFC 6192 addresses the issues. The use of the QoS engine to limit the rate of the traffic concerned to a few Mb/s makes it possible to protect the router from a denial-of-service attempt. It is of course necessary to investigate immediately if the limit is reached or about to be reached. This security does not distinguish between legitimate and illegitimate traffic.

Additionally, traffic explicitly destined to the router itself has no reason to be fragmented, you can block it if fragmented.

• OSPF SECURITY

The arrival of OSPFv3 is an opportunity to drop MD5 and use IPsec to secure exchanges. ESP must be supported, AH optionally (RFC 4552). All in transport mode.

Note about other protocols:

RIPng offers the same thing.

BGP is not specific to v6 and follows a different path through the BGPsec initiative which aims to group route origin signature and path validation (AS-Path) from end to end. This initiative focuses on public routing and does not seem to include at the moment an encryption and authentication component for corporate networks, based on a private PKI or on a manual implementation of the keys.

IS-IS sees no evolution on security side, moreover it is IP agnostic.

Filtering

Filtering recommendations are to be applied at least at the network edge, some rules can be integrated in routers and not just in firewalls, although the stateful aspect is still necessary for some of them.

• ICMP

While there is a strong trend towards restricting the ICMPv4 traffic allowed, ICMPv6 requires a more granular approach.

RFC 4890 "Border Firewall Transit Policy" reminds us of this and proposes ACLs to implement. You will find them here:

Mandatory permit :

- Destination Unreachable (Type 1) - All codes;
- Packet Too Big (Type 2) – required for PMTU discovery;
- Time Exceeded (Type 3) - Code 0 only;
- Parameter Problem (Type 4) - Codes 1 and 2 only.

Optionally:

- Time Exceeded (Type 3) - Code 1;
- Parameter Problem (Type 4) - Code 0;

To control the echo request and reply (usually blocked from the internet):

- Echo Request (Type 128);
- Echo Response (Type 129).

Except when using IPv6 mobility it is advisable to block:

- Home Agent Address Discovery Request (Type 144);
- Home Agent Address Discovery Reply (Type 145);
- Mobile Prefix Solicitation (Type 146);
- Mobile Prefix Advertisement (Type 147).

ICMPv6 error and information codes not allocated by IANA should be blocked on external filtering (internet, partner, etc.) Their internal blocking is at the discretion of the administrators.

Error code: types 5 to 99 and 102 to 126 included as well as 150 (seamoby).

Informational code: Types 154-199 and 202-254 included.

ICMPv6 foresaw mechanisms that are not used in practice, and thus to be blocked:

- Node information :
  - Node Information Query (Type 139);
  - Node Information Response (Type 140).
- Router Renumbering (Type 138) This message enables you to change the prefix of all configured interfaces of the router that receives it. Not likely to be used. Not to be confused with DHCPv6 and Prefix Delegation renumbering.
- Experimental codes (Types 100 – 101 and 200 – 201);
- Other unused types (Types 127 and 255).

When set in L3 (router) mode, the firewall should block transit (beyond gateway) of messages that exist only within the scope of the link-local address:

- All NDP including reverse.
  - Router Solicitation (Type 133);
  - Router Advertisement (Type 134);
  - Neighbor Solicitation (Type 135);
  - Neighbor Advertisement (Type 136);
  - Redirect (Type 137);
  - Inverse Neighbor Discovery Solicitation (Type 141);
  - Inverse Neighbor Discovery Advertisement (Type 142).
- Multicast NDP tied to routers:
  - Multicast Router Advertisement (Type 151);
  - Multicast Router Solicitation (Type 152);
  - Multicast Router Termination (Type 153).
- Messages related to the unusable SeND protocol:
  - Certificate Path Solicitation (Type 148);
  - Certificate Path Advertisement (Type 149).
- MLDv1 and v2 messages (must arrive via link-local and have a hop-limit of 1):
  - Listener Query (Type 130);
  - Listener Report (Type 131);
  - Listener Done (Type 132);
  - Listener Report v2 (Type 143).

On the other hand, if it works as a bridge (L2), it must authorize the above listed messages, apart from SeND (as long as it is not usable).

Should be allowed although still optional:

- Time Exceeded (Type 3) - Code 1;

- Parameter Problem (Type 4) - Code 0.

Even in L2 it is recommended to block Redirect (Type 137) for security reasons. Unless it is in fact used, for example if a segment has 2 routers (one inbound, one outbound) and a host whose routing table is not adapted.

Finally, DPI will have to analyze the payload to detect any ICMPv6 malformed or being used to exchange messages by creating a kind of tunnel. This should be done at least at the edge of the Internet.

DPI will also be able to block a PMTU-D return with a value below 1280. This is impossible and would risk making a poorly developed IP stack crash.

• TRANSITION MECHANISMS

If disabling transition mechanisms on hosts is a good practice, blocking them on filtering devices is as useful.

These rules are to be applied both on an IPv4 and on an IPv6 network, depending on the direction of the encapsulation. See RFC 7123.

It is therefore necessary to block :

- IPv4 Protocol #41 (6in4, 6to4, 6over4, 6rd, ISATAP);
- IPv4 Protocol #47 (GRE) except if used;
- Teredo:
  - UDPv4 destination port 3544;
  - If DPI is running, filter UDPv6 packets with teredo address (belonging to prefix 2001::/32) in the payload;
  - DNS requests to teredo.ipv6.microsoft.com. (via DPI and/or directly on DNS servers).
- ISATAP:
  - Filter DNS type A requests for isatap.* (via DPI and/or directly on DNS servers).
- 6to4:
  - IPv4 protocol 41 going to or coming from 192.88.99.0/24;
  - Tighter with DPI, IPv4 proto packet #41 with 6to4 address (belonging to prefix 2002::/16) in the payload.
- 6over4:
  - Packets with protocol #41 and destination 239.0.0.0/8 (block 6over4 NDP).
- Tunnel Broker / TSP (Tunnel Setup Protocol):
  - TCPv4 and UDPv4 with destination port 3653;
  - Ability to pre-screen with IP proto #41.
- AYIYA:

    - TCPv4 and UDPv4 with destination port 5072.

Be smart with DPI when possible, filter on the protocol number first before sending to the analysis engine in order to save resources.

Any triggering of one of these rules from a machine inside the network should result in an investigation to identify the cause of its misconfiguration. Especially for host-initiated mechanisms like Teredo and ISATAP.

On IPv6 you can block 4rd, 4over6, etc.

• BOGON PREFIXES AND ROUTES

In IPv4 it is abnormal to see some addresses, for example a packet with a source address of 127.0.0.5, or an IP RFC1918 coming from the internet… Same thing in IPv6.

Ideally, you should block the concerned packets on the front-end firewalls of the Internet, but also filter any BGP announcement with these prefixes from the Internet or a partner (except in special cases)

- Larges non-allocated blocks :
    - 2d00::/8
    - 2e00::/7
    - 3000::/4
    - 4000::/2
    - 8000::/1
- 2001::/23 IETF reserved;
- 0::/96 Former IPv4 compatibility prefix;
- ::ffff:0:0/96 IPv4 Mapped addresses;
- 64:ff9b::/96 NAT64 Well Known Prefix;
- 64:ff9b:1::/48 Block dedicated to NAT64 locals platforms;
- 100::/64 RTBH (Remote triggered black hole filtering);
- 2001:2::/48 Benchmarking;
- 2001:0DB8::/32 Documentation;
- 5f00::/8 6bone, dismantled;
- 2002::/16 6to4;
- 3ffe::/16 former TEREDO;
- 2001::/32 TEREDO;
- 2001:10::/28 ORCHID Overlay Routable Cryptographic Hash Identifiers RFC 4843;
- 2001:20::/28 ORCHID v2 RFC 7343;
- 2001:3::/32 AMT, used to join a multicast through a tunnel RFC 7450;

- 2001:1::1/128 PCP, allows to ask the firewall to dynamically open a port;
- ff00::/8 Multicast;
- fe00::/9 former multicast;
- fc00::/7 Unique Local Address;
- fec0::/10 former Site Local Address, deprecated;
- fe80::/10 Link-local (except for L2 bridge firewall);
- ::1/128 Loopback (Do not block on a host OS firewall);
- ::/128 (0) Address not specified;
- ::/8 Many reserved addresses included the 2 last ones;

In addition to RFCs, don't forget IANA ressources:

https://www.iana.org/assignments/iana-ipv6-special-registry/iana-ipv6-special-registry.xhtml

https://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xhtml

https://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xhtml

Automatically generated lists containing these prefixes as well as prefixes not assigned by any RIR exist. The most known is https://www.team-cymru.org/Services/Bogons/fullbogons-ipv6.txt .

You can use it directly (Bogon + unallocated) or keep only the information about routable unicast addresses 2000::/3

Note that the 2001:4:112::/48 AS112 block allows to blackhole the numerous reverse DNS (ptr) requests associated to private IPs. The AS112 project aims to relieve DNS root, conducted by ICANN it generates statistics from the requests. So you should only block this prefix if your DNS infrastructure does the blackholing itself.

• HEADER EXTENSION

IPv6 brings with it header extensions (EHs). They can be combined and must always appear on the first fragment in the case of a packet fragmented by the sender host. It is therefore necessary to destroy any 1st fragment that does not contain the full IPv6 header.

One of them is the HopByHop (proto 0) which requires to be handled by each intermediate router. This makes de facto a DDoS possible especially if the device has to forward the handling to the control plane. Rather than destroying the packet, it is better to ignore this field on organization border. It is still necessary to activate it to do multicast or jumbogram internally.

Another particular extension is the source routing, Routing Header (proto 43), which appears to be similar to the one in IPv4. However it is only appropriate to block its sub-elements RHT 0 and RHT1 which correspond to deprecated source routing and Nimrod. Others are relevant like the SRH (Segment Routing Header) of SRv6.

Don't block the extension indicating that the packet is fragmented (Proto 44), and the 2 extensions

related to IPsec : Encapsulation/ESP (Proto 50) and Authentication/AH (Proto 51).

The following RFC draft details the recommended policy (starting section 3.3)) https://datatracker.ietf.org/doc/html/draft-ietf-opsec-ipv6-eh-filtering

You should not simply reject packets because they contain extensions. Ideally you should just filter certain types between the public network and the internal network.

Make sure your ISPs don't drop packets with extensions, and internally check your routers and firewalls to identify when a packet escalates to the control plane because of extensions.

Review these rules every 2 years, some extensions may disappear, others may arrive. At the moment there are still devices that try to process extensions even if they are not in order or repeated, which can lead to crashes, see https://datatracker.ietf.org/doc/html/draft-kampanakis-6man-ipv6-eh-parsing-01

Finally, take the time to read RFC 7112 to understand what happens when extensions are chained together and fragmented. Hence the decision to force them all into the first fragment.

• BAN POLICY

Since IPv6 offers a lot of addresses, it is necessary to change the way temporary bans are handled.

Many mechanisms are triggered to block a user temporarily after a given number of unsuccessful authentication attempts, or to impose a captcha on a web site after heavy traffic from a single IP. This is typically the principle of a tool like Fail2Ban or equivalent.

An infected machine, member of a botnet, will always have the same IPv4 until its ISP decides to change it. It will however be able to use the 2^64 IPs offered by the /64 of which it is a member in a random way and with very frequent changes.

This can quickly saturate the blocking lists, or on the contrary, it can bypass them by changing the IP between each attempt.

For these reasons, it is important to always build your blocking mechanisms on the /64. And ideally, you should also trigger a malus on the parent /56, to save time in case of a malicious attempt from a neighboring /64. The latter probably belongs to the same household.

Note: This situation obviously applies to the opposite case, asking a user to re-authenticate after 20 minutes because his temporary IPv6 has changed makes no sense as long as he still resides in the same /64.

Appendix

Appendix

The appendix contains technical supplements and specific information related to domestic use.

• URL AND LINK-LOCAL IP

The link-local address, in the FE80::/10 range, has the particularity of incorporating an interface identifier. This identifier uses the name or number of the interface depending on the system.

Linux adds the name of the interface, for example %eth0, macOS similarly with %en0 typically.

Windows on the other hand adds the interface number, %1 and so on.

```
C:\Users\JC>netsh interface ipv6 show address

Interface 1 : Loopback Pseudo-Interface 1

Addr Type  État DAD    Vie valide Pers. Fav. Adresse
---------  ---------- ---------- ---------- ----------------------
Autre      Préféré       infinite   infinite ::1

Interface 3 : Ethernet

Addr Type  État DAD    Vie valide Pers. Fav. Adresse
---------  ---------- ---------- ---------- ----------------------
Public     Préféré         29m51s      9m51s 2a01:cb00:83f5:[redacted]:45e1:e8c3:a8c9:739d
Autre      Préféré       infinite   infinite fe80::45e1:e8c3:a8c9:739d%3
```

Figure 17. netsh interface ipv6 show address

Under Windows, the `netsh interface ipv6 show address` command displays all assignated IPv6. In powershell `Get-NetAdapter` and `Get-NetIPAddress` also display information.

```
PS C:\WINDOWS\system32> Get-NetIPv6Protocol


DefaultHopLimit                 : 128
NeighborCacheLimit(Entries)     : 256
RouteCacheLimit(Entries)        : 4096
ReassemblyLimit(Bytes)          : 133913120
IcmpRedirects                   : Enabled
SourceRoutingBehavior           : DontForward
DhcpMediaSense                  : Enabled
MediaSenseEventLog              : Disabled
MldLevel                        : All
MldVersion                      : Version2
MulticastForwarding             : Disabled
GroupForwardedFragments         : Disabled
RandomizeIdentifiers            : Enabled
AddressMaskReply                : Disabled
UseTemporaryAddresses           : Enabled
MaxTemporaryDadAttempts         : 3
MaxTemporaryValidLifetime       : 7.00:00:00
MaxTemporaryPreferredLifetime   : 1.00:00:00
TemporaryRegenerateTime         : 00:00:05
MaxTemporaryDesyncTime          : 00:10:00
DeadGatewayDetection            : Enabled
```

Figure 18. Get-NetIPv6Protocol

The powershell `Get-NetIPv6Protocol` command returns global host IPv6 configuration.

For development purposes, or simply for personal use, URLs are commonly formed with IP.

In IPv6, an http URL will be for example in the form http://[2001:db8:AAAA:BBBB::H]:8080

It is currently impossible to use link-local addresses within a URL under a popular web browser. RFC 6874 forces the use of the % character before the interface identifier in a URL (%2, %eth0, etc.)

However, the use of % is reserved by the HTML maintenance community, WHATWG https://github.com/whatwg/url/issues/392

So one should always use global or local addresses in a browser. Their support works normally in other contexts, like a SCP client.

Here is the ticket asking for the reimplementation in Firefox of the link-local: https://bugzilla.mozilla.org/show_bug.cgi?id=700999

And Chromium https://bugs.chromium.org/p/chromium/issues/detail?id=70762

remember, make sure that a strictly local application (such as an industrial control interface) never needs a user to connect to it via the local link address in a browser.

• MULTI-PREFIXES

IPv6 allows multiple prefixes to be used simultaneously on a network, but while the mechanism works very well at a low level on the network and hosts, it does raise some issues.

Flows will leave the host from an address chosen by default or from the one that has the longest common prefix with the destination. This uncontrolled aspect makes the configuration more complex.

Which address should be auto-registered in DNS locally?

Security systems, whether in the access layer (L2-NDP) or traffic layer (FW, ACL, etc.) must be able to adapt on the fly.

Multi-homing up to the hosts can remain interesting in a small structure to allow switching when changing providers. For intermediate size networks, it is a choice to be opposed to the combination (PI or ULA) + NPTv6. The later being stateless and easy to configure.

•=

A mechanism was designed to allow a client and a server to exchange their different addresses via a header extension and to switch in case of failure without affecting the upper layer and therefore without timeout. This was Shim6. They could even authenticate themselves via addresses generated with cryptographic mechanisms (CGA). In practice, Shim6 has been dropped, so we remain in the realm of timeout + establishment of a new session in case of loss of a path, or taken into account by a upper layer protocol. As far as the OSI model is involved, it should be noted that IP was never supposed to provide this type of mechanism anyway, it is the role of TCP and now QUIC. === •=

• CONTAINERS

•= Docker

Docker operates by default a bridge, a Docker0 interface and attaches ports to NAT44 rules pointing to published container ports. Additional bridges can be created to isolate containers from each other.

The overlay mode leverages VxLAN and allows inter-host communication without worrying about the configuration of the underlying network (in addition to the ability to encrypt, simplify SWARM administration, etc.)

It is therefore difficult to use IPv6, as Docker is designed to provide a total abstraction of the network (and the rest too).

There are several ways to get around this problem:

- Use the "macvlan" mode, which comes down to expose the containers at level 2 as if they were VMs. Each with its own MAC. Not very practical and above all difficult to integrate and operate in the ecosystem;
- The more recent IPvlan L2 mode exposes the IPs of the containers behind the same MAC than the host via a lighter mechanism than the classic bridging;
- In its L3 version, IPvlan completely eliminates loop risks and relies on IPv4 subnets and IPv6 prefixes. The corresponding routes must be implemented on the network devices, each host having one or more unique prefixes.

In 2016, a developer initiated a project bringing NAT66 in Bridge mode to Docker https://github.com/robbertkl/docker-ipv6nat

He also points out that the absence of NAT leaves all ports accessible in IPv6, and that it is therefore necessary to think about securing access upstream.

For large deployments, we recommend the IPvlan L3 mode.

Do we really need IPv6 in Docker? As indicated in the document, it is interesting to provide IPv6 support on the frontend (for example SLB containers such as traefik, hap, envoy, caddy, etc.). Beyond that the backend can stay in IPv4.

•= Kubernetes

Kubernetes exposes by default one IP per Pod (grouping of containers on a host). The host is named node. Beware of the meaning of Pod which differs here from other solutions. The address is taken from the block assigned to the node.

The addressing is thus exposed flat without overlay, facilitating inter-pod communication whether they are in the same node or not. The vision of the addressing is therefore identical whether you are inside or outside the solution.

It is therefore very similar to Docker's IPvlan 3 mode.

The management of the network is then handled by one of the many third-party solutions on the

market (open source or not).

Finally, the exposure from the outside is usually done through the Kubernetes services combo coupled with a load-balancer, the latter most often external.

IPv6 has been marked by Docker as a stable feature recently, Kubernetes followed with beta support in 1.21 and stable in 1.23. https://kubernetes.io/docs/concepts/services-networking/dual-stack/

Since these releases in late 2021, some cloud providers have already started to roll-out IPv6 on container services and on other managed services indirectly held by containers.

Remember that unless you are running Headless Services, load balancing will always perform address translation.

For outgoing traffic to the Internet, the use of public IPv6 addresses avoids the need for proxying or NAT.

• SCADA

A SCADA network is for recall a closed network, often found in industrial world. The point of migrating to IPv6 is relatively limited here. The compatibility of industrial solutions with the protocol will take time to reach full maturity. However, do not hesitate to mention this compatibility in the optional questions of RFPs and seriously consider v6 only if the whole ecosystem is compatible and tested. If your SCADA network is huge, as your business involves many points of presence, IPv6 can still save you IPv4 addressing. Implementing 6LoWPAN on embedded hardware can also be a driver. But failing that you can always operate in IPv4 addressing overlay/overlap with the rest of the IT since the very principle of SCADA is that it is isolated and not routed to other resources. This leaves the overlap treatment to be managed only on the interface elements between the general Information System and the SCADA Information System, elements which are also, for security reasons, rather few.

• NAT64 ON MOBILE CARRIERS NETWORKS

Let’s see what is involved when setting up NAT64 between smartphones and the Internet.

•= Service discovery

The NAT64 section of the document explains its implementation with workstations. Some methods are used to supply hosts with the NAT64 prefix, mainly on mobile platforms. This ensures that endpoints are aware that they are located behind a NAT64. The main benefits of this awareness are to allow the host to restore DNSSEC validation as well as to permit the operation of literal addresses not only in the IP layer but also when a payload carries it (e.g. SIP without the need for an ALG).

RFC7051 addresses this topic, as well as the following draft:

https://tools.ietf.org/id/draft-ietf-v6ops-nat64-deployment-08.html

One solution is the DNS record ipv4only.arpa which must provide a known answer based on an

RFC. In this case an A record 192.0.0.170 or 192.0.0.171.

If the response is an AAAA record, e.g. 64:ff9b::192.0.0.170 (here in decimal notation to make it easier for you to read, you who have ventured into the appendix), then a NAT64 platform using the 64:ff9b::/96 prefix is in production. For the record, Android does the same thing with the ipv4.google.com DNS record.

The PCP protocol (the one that enables you to open a port on your home router) also offers the possibility to request the existence of a NAT64 prefix.

The RFC mentions other ways, providing the information in the Router Advertisement, or via a DHCPv6 option.

Finally, the good old operator APN configuration on mobiles also allows to push the prefix to smartphones.

PC OSes unfortunately do not support any of these methods on their LAN interfaces. Leaving DNS64 in the enterprise for a long time to come.

•= Operation on mobile OSes

To ensure compatibility with the literal use of IPv4 addresses as well as support for DNSsec signatures, etc., mobile OSes need to be able to use IPv4.

While the 2 main mobile OS implement mechanisms to provide IPV4 compatibility, the implementation differs radically.

Google Android relies on the network and 464 XLAT.

The clatd.conf file provides instructions for CLAT configuration of the endpoint, an IPv6 address that is part of the /64 assigned to the endpoint is mapped (SIIT) with a virtual private IPv4 address. (Often 192.0.0.4). The IP stack intercepts any IPv4 packets and translates them into v6. In the other direction, as soon as a packet arrives on the address reserved for the CLAT it is translated into IPv4. The development can be followed here https://android-review.googlesource.com/q/project:platform%252Fexternal%252Fandroid-clat

Apple iOS takes advantage of the rather limited openness of its system to deal with the problem from the upper layers. Thus, the frameworks (CFNetwork at the lower level, Cocoa URL loading system at the higher level) as well as the WebKit mandatory browsing rendering engine directly convert any IPv4 address into the one returned by the synthesis of the NAT64 prefix with said address. Thus, no IPv4 packet is ever really created. This way is more efficient from an energetical point of view.

•= Connection sharing

Also known as hotspot or tethering, sharing involves providing dual-stack WiFi to hosts that are unaware that only IPv6 is supplied to the router, in this instance a smartphone.

464 XLAT comes to the rescue, the phone will act as a CLAT in conjunction with the NAT64 (PLAT) of

the carrier network. Same operation on Android and iOS:

Instead of performing a stateful NAT44 followed by a NAT46, it will create a stateless mapping rule (SIIT) between the hotspot's IPv4 network (/24 most often) and a piece of the /64 IPv6 it owns. Thus no need for a state table and no port change on the phone side. The traffic will then undergo the stateful NAT64 of the carrier to switch back to IPv4 on the internet.

Remember, the IPv6 header being longer, the first gateway may have to fragment traffic. So don't be surprised if uploading a file is slowed down by CLAT. ARM SoCs currently available on the market offer hardware support for all 464 XLAT operations to avoid such problems.

• IPv4 PORT SHARING

The Address + Port techniques are briefly covered in the section on transition mechanisms. (4rd and MAP-T/E for the most recent ones). Hosts behind a home router using such a mechanism are not aware that only part of the 65,535 ports is assigned to their WAN.

Nothing very worrying, except when a program requires a port to be opened (UPnP, NAT-PMP) and the router forgets that it doesn't have access to all the ports as well. It will sometimes return a port outside the range assigned to the subscriber. This is like playing Russian roulette with some P2P exchanges.

RFC 6269 discusses the problems associated with sharing, including the one mentioned here that occurs with carriers that have implemented it a bit too quickly and loosely.

An ISP should not share IPs between more than 16 customers.

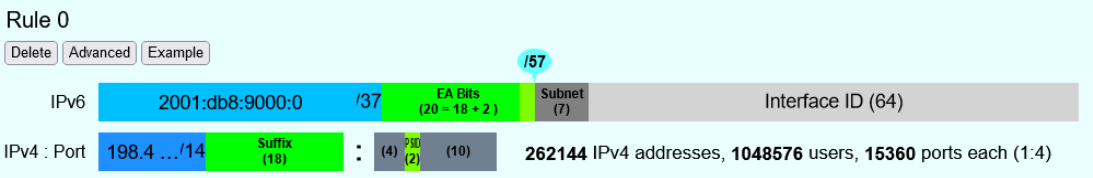


Figure 19. MAP A+P port sharing simulation

In this example, IPv4 are shared between 4 customers http://map46.cisco.com/MAP.php

• RFC DRAFTS TO SAVE IPv4

Some people are striving to extend the life of IPv4 by finding ways to increase its addressing capabilities.

Several drafts have existed, the most recent ones seem to be:

https://www.ietf.org/id/draft-schoen-intarea-unicast-0-00.html

https://www.ietf.org/id/draft-schoen-intarea-unicast-127-00.html

https://www.ietf.org/id/draft-schoen-intarea-unicast-240-00.html

Needless to say, updating all IP stacks of PC OS, smartphones, routers, etc. to support these changes

would require much more effort than switching to IPv6.

Nevertheless, 240/4 is officially supported by at least 2 major manufacturers as well as Google GCP.

On a different front, the EzIP proposal is in its ninth iteration, if you like NAT read it:

https://datatracker.ietf.org/doc/html/draft-chen-ati-adaptive-ipv4-address-space-09

• EXAMPLES OF IPV6 IMPLEMENTATION PROBLEMS

Here are some examples of implementation bugs encountered when using IPv6.

•= Non-decommissioning of routes

With IPv4, you either have connectivity or you don't. As soon as you switch to dual-stack, how can you be sure of the availability of IPv6 connectivity? Happy Eyeballs can help, but it generates a delay and is not designed to compensate a prolonged absence of IPv6 connectivity.

For example, the ISPs routers with LTE backup often have only IPv4 on the backup link. When the backup is triggered, some routers continue to send RAs to declare themselves as the default router and announce an IPv6 prefix that is no longer usable since IPv6 connectivity is completely broken.

This problem also appears during renumbering. In IPv4 NAT44 makes the local network independent of the WAN addressing. With IPv6 this is no longer the case (except when using ULA + NPTv6 combo). So on the rare occasions when a consumer ISP renumbers its network, customers may experience a temporary loss of connectivity while the old RA information is still in cache.

Section 6.3.5 of RFC 4861 states that hosts must purge the prefix if the timer expires or if the router no longer announces itself as default. But in our case the router still exists and is still reachable via its local link address. The hosts will wait for the prefix timer to expire before deleting the interface address(es) using the old prefix. The endpoints will therefore still send packets to the router, but with a source address belonging to the old prefix… It is difficult to expect a response, and without aggressive timer settings it can easily take 1800 seconds, half an hour. We can only recommend to carriers to lower the expiration times to a value below one minute.

People who want to play with IPv6 multihoming will quickly encounter similar failover problems.

•= Unexpected use of IPv4 prefix representation

In order to simplify your information system, you have decided to use only the IPv6 notation in your CMDB. So you use the prefix ::ffff:0:0/96 to indicate an IPv4 in your configuration scripts, etc.

Strangely, your script creates an ACL rule/policy, but then is unable to find it in its check and ends its execution on a failure. However, the concerned flow works.

In fact the configured system simply decided to translate the notation of an IPv4 with ::ffff:0:0/96 back to the classic IPv4 notation.

This kind of behaviour have existed on some F5 products for example: https://cdn.f5.com/product/

bugtracker/ID669888.html

Practical, but to be considered in automations.

```
PS C:\WINDOWS\system32> ping ::ffff:c0a8:1

Envoi d'une requête 'Ping'  192.168.0.1 avec 32 octets de données :
Réponse de 192.168.0.1 : octets=32 temps<1ms TTL=64
Réponse de 192.168.0.1 : octets=32 temps<1ms TTL=64
```

Figure 20. We can encounter this automatic conversion in common tools, such as Windows ping

•= Incompatible input fields

When entering an IPv6, the field checks are sometimes inadequate. One can find the following glitches in graphical environments and, more rarely, in a command line environment.

A completely incompatible field will reject an address that is not in IPv4 form, but subtleties can get through the checks. For example, sometimes the [ ] used to separate the address from the port is not taken into account. . Thus the entry of [2001:db8::2D5E]:8443 may be transformed by the software into 2001:db8::2D5E:8443 .

• WASTE OF ADDRESSING SPACE

Yes, there are plenty of IPv6 addresses! Internet is full of wise calculations to explain us that 2E128 is equal to 3,4 * 10E38 addresses, that is to say 667 sextillions by m² of terrestrial surface. Number moreover close to the constant of Avogadro point out others (~6,02*10E23).

So, of course, with sentences like "we could address each sand grain up to 2km deep" we feel that we can do anything.

However, an IPv6 address is not a license plate or a phone number. It mostly follows a construction based upon a /64 prefix. Moreover, these prefixes are part of a subset reserved for global routing and assigned by the continental manager (RIR).

Thus, a large company that gets a /29 can logically create 34 billion of networks. If we now count the number of facilities in /48, that's 524,288.

The Indian post with its 160,000 post offices is therefore quiet... Well, unless someone decides that the guest WiFi and the smart building IoT project each need their respective /48 per site, because security/policy/delegation/internal organization (strike out the irrelevant) requires it. This will make you chuckle, but look back at IPv4, this kind of reasoning is far too widespread.

• USE OF ADDRESS UNICITY FOR OTHER PURPOSES

The huge number of possible addresses has given engineers ideas for how to manipulate them based on the precise identification of the user and/or the resource to be accessed.

Here are some examples:

- Assign different IPv6 addresses to a server for each client connecting to it? In case of a DDoS we

can block only the concerned address without affecting the other clients connecting to the same machine. The future friend of RTBH?

- Include an authentication directly in the address that evolves over time? This is the principle of the IPv6 TOTP provided by this SSH server project whose IP changes every 30 seconds. https://github.com/mikroskeem/tosh
- Assigning directly data such as streaming video chunks and no longer the server that hosts them, this is for example the object of the following patent https://patents.justia.com/patent/11134052

Allocating a huge number of addresses to each server can quickly overload the NDP cache.

These uses are still possible if we assign a /64 prefix directly to the server, as described in RFC 8273. This is what we already do with containers as described above with the example of Kubernetes nodes. These /64 could as well be handled by Load Balancers.

For systems with regular change of address it implies to reassemble a session each time, but after all, it would never be more than a new use of the 0-RTT of QUIC for example.

• SRv6

Segment Routing is spreading rapidly among carriers and GAFAMs. Currently SR-MPLS leads the deployments, but forecasts show that its counterpart based on a simple IPV6 data-plane will take over within a few semesters.

Mastery of IPv6 transport and this sector-dominant IGP, IS-IS, will quickly become a must for any large network.

In addition to SR's contributions in terms of dynamic and adaptive topology, telemetry and the possibility of including service-oriented fields (security group, application identifier...) within the SRH header, it will undoubtedly be the first to replace the entirety of existing stacks of layers protocols.

Thus beyond the backbone, it will probably replace the VxLAN + EVPN pair in the datacenter, as well as the closed SDN Campus solutions. Offering a true end-to-end service without compromise.

The fields of service will then allow for a true dynamic policy application, no longer based upon address ranges etc., but rather on additional information. All this without proprietary technology, but being usable by both physical and virtual service devices (VNF).

Later on, these fields will probably be inserted by the host itself, so that information provided directly by the application can be passed on to it. The 1st hop router will still be in charge of adding the selected path. On the server side, we have seen the integration of VTEP termination (VxLAN, and sometimes GENEVE) coming down from the Top of Rack switches to the servers themselves. In the same way, we will probably witness full SRv6 processing on servers, including topology management, thanks in particular to the arrival of Network Processor Units (NPUs, not to be confused with Neural Processor Units) and IPUs (Infrastructure Processing Units).

Manufacturers are currently pushing companies to make the transition towards SR-MPLS, only to come back later with SRv6. However, we may soon start to assist to direct SRv6 transition on

corporate network and not longer only on carrier's networks.

• THREAD

Thread is a IoT oriented network protocol pushed by the Thread Group https://www.threadgroup.org/ .

Figure 21. Thread logo

Its purpose is to provide a mesh communication network between home automation devices based on 6LoWPAN. It exploits IPv6 with notions of scope, router nodes and children. Check the OpenThread open source project page https://openthread.io/guides/thread-primer/ipv6-addressing .

The smart home connectivity standard **Matter** is built with it.

• SELF-HOSTING AND RESIDENTIAL USE

The experience of implementing IPv6 on a simple home network allows to easily understand some of the differences compared to IPv4. In particular, we will see here the exposure of services to the outside world.

Although these examples can be used in a small structure, we remind you that it is essential to have a real filtering and analysis layer at the entrance of the Internet on a production system, even small.

•= Addressing and DNS publication

Most of the time, consumer carriers only provide a /64 without the possibility of using the other prefixes assigned to the router (often in a /56).

It is also impossible to ensure the stability of the prefix over time (unless there is a contractual commitment).

The address of each machine to be exposed must therefore be published independently, whereas we used to publish the WAN IPv4 address and play with the NAT44 ports.

We will start by making sure that the machines use a stable address (typically based on MAC or stable privacy, which is desirable).

We will then use a dynamic IPv6 DNS service, for example Dynu, DuckDNS, etc.

There are several methods to trace the IP/ AAAA DNS record pair directly on a machine:

- Query script with auto detection of the address by the API server of the DNS service;
- Script retrieving the public IP via a third-party API (e.g. api6.ipify.org) then forwarding to the DNS service;
- Script retrieving the IP from the system interface (be careful to use the public stable one);

- Software agent of the service.

It is also possible to rely on a router and its NDP information, but then we leave the simple use of the carrier device.

•== Flow openinng

The provisioning of a firewall in IPv6 is unevenly treated by operators. Some have implemented it very late in All or Nothing mode, others offer a granularity similar to what we find in IPv4.

Let's take the example of an Orange ISP LiveBox 4. In IPv4 the opening is done in the network section.

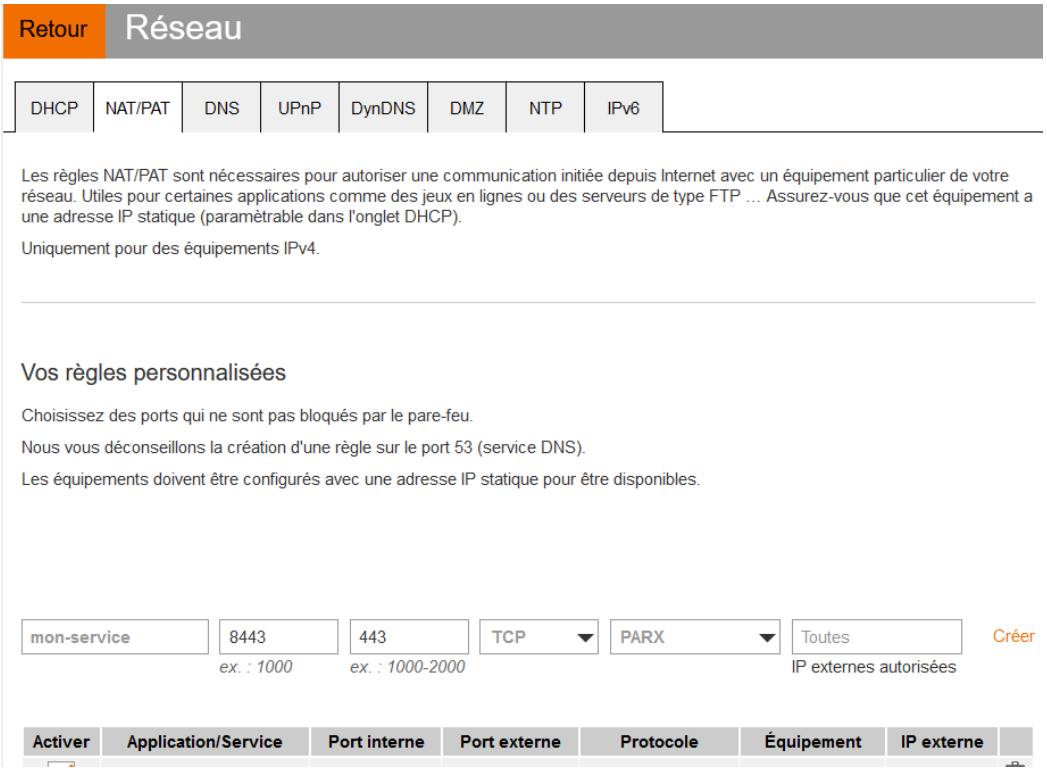


Figure 22. IPv4 Orange ISP LiveBox 4 (France)

In IPv4 we are used to have different ports between internal and external, which avoids having to change the ports on the servers, but prevents publishing several machines on the same external port (unless you go through an intermediate reverse proxy)

In IPv6 the situation is the exact opposite, each machine has its IP and therefore its 65535 ports, but one must necessarily use the same port number internally and externally because of the absence of translation (PAT).

At Orange ISP the configuration is in the firewall section.

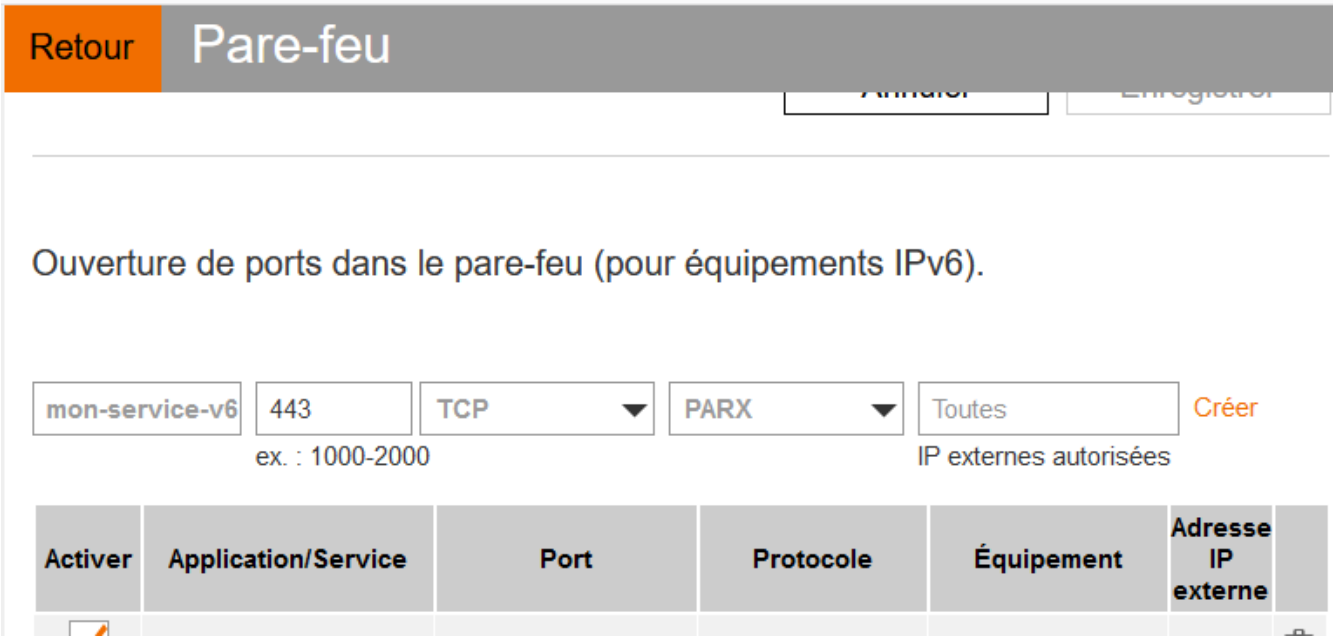


Figure 23. IPv6 Orange ISP LiveBox 4 (France)

•= Reachability test

The test can be conducted via an online port scanner such as http://www.ipv6scanner.com/

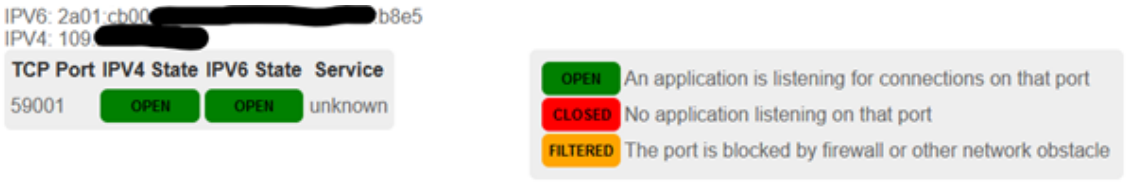


Figure 24. Port scan result

Here everything is in line, otherwise remember that Happy-Eyeballs V2 will switch the connection back to IPv4 in the absence of a v6 response.

Some providers don't offer fine firewalls, this is the case of Iliad Free which has long been hiding behind the fact that the RFC on CPE recommends, but does not impose stateful filtering. Free has only offered an IPv6 firewall since 2020 and it is very light. Many customers are asking for the implementation of a real firewall on the bugtracker https://dev.freebox.fr/bugs/index.php?string=ipv6&project=9&type%5B%5D=&sev%5B%5D=&pri%5B%5D=&due%5B%5D=&reported%5B%5D=&cat%5B%5D=&status%5B%5D=open&opened=&dev=&closed=&duedatefrom=&duedateto=&changedfrom=&changedto=&openedfrom=&openedto=&closedfrom=&closedto=&do=index

• HOST-INITIATED AUTO OPENING

Discussed earlier in the document, PCP V2 allows a port to be opened by the router on request by an application. Generally for P2P uses.

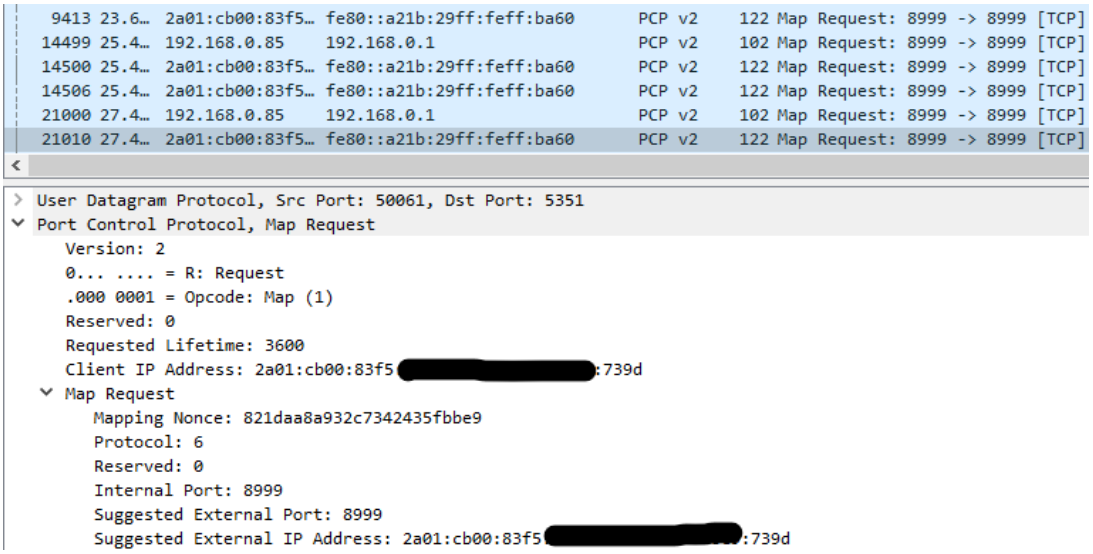


Figure 25. Wireshark PCP v2 IPv6

Example of Wireshark capture of PCP V2 with the filter "udp.port eq 5351". We notice opening requests both in IPv4 and IPv6.

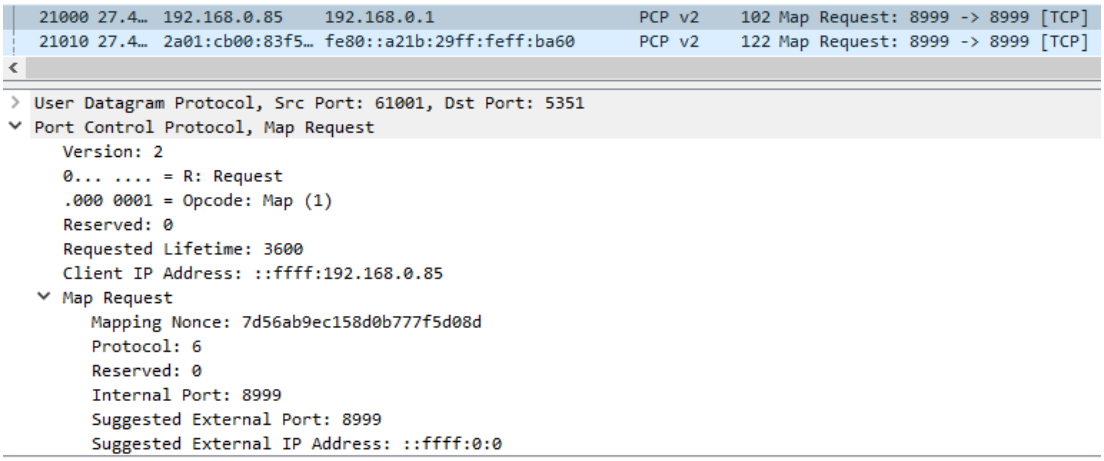


Figure 26. Wireshark PCP v2 IPv4

Observe that the IPv4 version of the request has its internal IP written as an IPv6 represented IPv4, and that the WAN address is set to 0.0.0.0 since it is the router's IPv4 WAN anyway (again in the same form with ::ffff:)

This is a world away from the heavy XML of UPnP-IGD requiring the exchange of many packets.

• EVOLUTION OF ONLINE GAMING

Currently, the gaming industry does not integrate IPv6 into its communications between players and servers. The impact of IPv4 CG-NAT and other IPv4aaS mechanisms could be avoided with an effort from the studios.

Games where the party is managed by a dedicated server should switch their server to dual stack and favor IPv6 when it is available.

For P2P games where one of the players hosts the game, it would be good to include in the host election algorithms a weighted element based on the availability of the dual-stack if for example at least 40% of the players in the game have active IPv6.

• WHAT TO EXPECT FROM CONSUMER CARRIERS?

Regulators bodies should ask carriers to implement the following mechanisms in addition to IPv6 on fixed connections (xDSL, FTTh, fixed 4/5G, Low Orbit SAT,etc.):

- A fine tunable firewall, based dynamically on the address set tracking for each host and the match with the MAC address in the NDP table;
- Provisioning of at least 2 /60 prefixes in addition to the default prefix upon a simple DHCPv6-PD request from another router. It would be convenient for carriers to also offer the possibility to implement static routes on at least one documented IPv4 RFC1918 block on their side;
- IPv6 renumbering management avoiding blackouts, typically by adjusting RA timers;
- Clear information in the modem interface about the IPv4 and IPv6 access mode, as well as the mapped port range in the case of an IPv4 A+P sharing approach (4rd, MAP-x, etc.);
- The option to use a third-party router at a time when IPv4 A+P sharing mechanisms make the provider's router even more exclusive.

On mobile connectivity, it would be relevant to support PCP v2 on the endpoints, especially on the connection sharing APN. This would allow customers to take full advantage of IPv6 end-to-end when using hotspots. DHCP-PD support would also be very handy for specific cases of multi-network sharing with multiple /64.

About this document

This document has been written within the scope of the ARCEP IPv6 task force, its main author Jean-Charles BISECCO would like to thank all its members for their contributions and corrections. The ASCIIDOC version on Github is provided by Axel Schemberg.

HELP US TO MAKE THIS DOCUMENT GROW

This document is intended to be updated over time, your feedback is precious.

You can send us your ideas to improve the guide as well as your feedback:

- Ask questions or discuss on Github: https://github.com/optimismus/HowToDeployIPv6/discussions
- File an issue on Github https://github.com/optimismus/HowToDeployIPv6/issues
- Join IPv6 task-force
- Contact the author at IPv6@arcep.fr/ IPv6@jclb.net

The latest version can be found at the following address:

https://en.arcep.fr/publications/task-force-ipv6.html

LICENSE

Schema icons sourced from https://github.com/ecceman/affinity

Chapter images come from unplash.com

> Valentin Betancur / Alex Padurariu / Andre Taissin / Erol Ahmed / Possessed Photography / Austris Augusts

TRANSLATIONS

This document is initially published in French and English. We are open to translations in other languages to facilitate the deployment of IPv6 in as many places as possible.

Translations may be added to the official list.

Access to deltas between this release and future versions will be provided to translators.

IPv6 Task -force