Engineer

IPv6

Versión de IP que utiliza direcciones de 128 bits

Qué es

IPv6, cuyo nombre completo es Protocolo de Internet versión 6, es la versión del Protocolo de Internet diseñada para suceder a IPv4, la versión descrita en RFC 791. Su especificación actual es RFC 8200, de julio de 2017, un estándar de Internet (STD 86) que sustituyó a RFC 2460. El cambio más visible es el tamaño de una dirección: 128 bits en lugar de los 32 de IPv4. Todo IPv4 tiene 4.294.967.296 direcciones; una sola subred IPv6 de tamaño habitual, una /64, contiene ese número al cuadrado.

El resto del paquete también cambió. RFC 8200 elimina algunos campos de la cabecera IPv4 o los hace opcionales para reducir el coste de procesamiento, lleva la información opcional en cabeceras de extensión separadas tras la principal y renombra el campo tiempo de vida (Time to Live) como límite de saltos (Hop Limit), que cada nodo que reenvía el paquete reduce en uno. No hay suma de comprobación de la cabecera. Los routers nunca dividen un paquete: en IPv6 solo el origen fragmenta, cada enlace debe transportar paquetes de al menos 1.280 octetos y se recomienda encarecidamente que los emisores utilicen el descubrimiento de la MTU de la ruta (Path MTU Discovery) para averiguar qué tamaño de paquete admite una ruta.

Una dirección se escribe como ocho grupos de 16 bits en hexadecimal, separados por dos puntos. RFC 4291 permite omitir los ceros iniciales de cada grupo y sustituir una secuencia de grupos formados solo por ceros por un doble signo de dos puntos, que solo puede aparecer una vez, de modo que 2001:db8:0:0:8:800:200c:417a puede escribirse 2001:db8::8:800:200c:417a. Como eso permite muchas representaciones de una dirección, RFC 5952, de agosto de 2010, elige una: minúsculas, ningún cero inicial y el doble signo de dos puntos en la secuencia más larga de grupos de ceros —la primera si dos tienen la misma longitud—, nunca en un solo grupo. Salvo direcciones especiales como las

de bucle local y las locales de enlace, todas las direcciones y prefijos de ejemplo de este artículo proceden de un rango de documentación: 2001:db8::/32, reservado por RFC 3849, o, para IPv4, los bloques reservados por RFC 5737.

RFC 4291 también cambia a qué se asocia una dirección. Las direcciones pertenecen a interfaces, no a máquinas, y una interfaz puede tener varias a la vez. Cada interfaz debe tener una dirección local de enlace dentro de fe80::/10, válida solo en su propio enlace. La dirección de bucle local es ::1. No hay difusión broadcast: la multidifusión cumple esa función, y ff02::1 llega a todos los nodos del enlace. Las direcciones locales únicas de fc00::/7, definidas en RFC 4193, sirven dentro de un sitio. Las direcciones de unidifusión global se utilizan para comunicarse fuera del sitio local. En la mayoría de las direcciones, los últimos 64 bits son el identificador de interfaz, por lo que un segmento de red normal es una /64.

Los hosts suelen asignarse sus propios números. Con la autoconfiguración de direcciones sin estado de RFC 4862, los routers anuncian los prefijos de un enlace mediante anuncios de router (Router Advertisement) periódicos, cada host combina un prefijo con su propio identificador de interfaz y, antes de utilizar el resultado, ejecuta la detección de direcciones duplicadas (Duplicate Address Detection), enviando mensajes de solicitud de vecino (Neighbor Solicitation) para comprobar que ningún otro nodo tiene esa dirección. DHCPv6 es la alternativa cuando un sitio quiere controlar con más precisión las asignaciones concretas; ambos mecanismos pueden funcionar a la vez.

Utilice el concepto en una decisión

Aplique «¿Qué es IPv6?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Wi-Fi, cortafuegos y acceso remoto configurados para trabajar desde cualquier lugar y permitir el acceso solo a las personas autorizadas.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. Administración de 30 VPS Ubuntu Linux — 30 servidores Linux administrados con recuperación planificada

Administración de 30 VPS Ubuntu Linux Póngase en contacto →

Por qué importa

Para muchos equipos, IPv6 no es un proyecto que decidan iniciar. La idea de RFC 4862 es que nadie tenga que escribir una dirección en el host: coloque un servidor en un enlace cuyo router anuncie un prefijo y podrá obtener por sí solo una dirección IPv6 global. Cualquier lista de permitidos, regla de cortafuegos,

analizador de registros o comprobación de estado concebidos solo para IPv4 tendrán entonces un punto ciego exactamente del tamaño del nuevo espacio de direcciones.

Ese punto ciego pasa fácilmente inadvertido porque ambas versiones son protocolos separados que no comparten nada en la transmisión. Un filtro de paquetes comprueba sus cabeceras por separado, un límite de solicitudes basado en direcciones IPv4 nunca ve un cliente IPv6 y una prueba que solo conecta con 127.0.0.1 no dice nada sobre ::1. En la práctica, un host protegido en una versión y abierto en la otra carece de cortafuegos.

IPv6 también depende más de sus mensajes de control que IPv4. El descubrimiento de vecinos, los anuncios de routers y el descubrimiento de la MTU de la ruta utilizan ICMPv6, de modo que la costumbre heredada de IPv4 de descartar ICMP en el perímetro rompe IPv6 de maneras que quizá solo se manifiesten cuando una conexión empieza a transferir datos. Y como una dirección tiene muchas representaciones válidas, el código que compara direcciones como cadenas da resultados contradictorios.

Cómo se aplica en la práctica

El módulo estándar `ipaddress` de Python analiza todas las representaciones válidas e imprime una forma abreviada, sin ceros iniciales y con la secuencia más larga de grupos de ceros comprimida. Por eso permite planificar un prefijo y comprobar rápidamente si dos representaciones son la misma dirección. La siguiente sesión es un ejemplo de este artículo, ejecutado en Python 3.15 con direcciones del prefijo de documentación:

```
>>> import ipaddress
>>> site = ipaddress.ip_network("2001:db8:42::/48")
>>> site.num_addresses == 2**80
True
>>> lans = site.subnets(new_prefix=64)
>>> next(lans), next(lans)
(IPv6Network('2001:db8:42::/64'), IPv6Network('2001:db8:42:1::/64'))
>>> len(list(site.subnets(new_prefix=64)))
65536
>>> a = ipaddress.ip_address("2001:0DB8:0000:0000:0001:0000:0000:0001")
>>> a
IPv6Address('2001:db8::1:0:0:1')
>>> a.exploded
'2001:0db8:0000:0000:0001:0000:0000:0001'
>>> a == ipaddress.ip_address("2001:db8:0:0:1::1")
```

```
True
>>> ipaddress.ip_address("fe80::1%eth0").scope_id
'eth0'
```

Una /48 deja 80 bits libres y, dividida en segmentos /64, proporciona 65.536, numerados mediante el cuarto grupo. La dirección introducida en mayúsculas y con todos los ceros escritos se devuelve en la forma abreviada 2001:db8::1:0:0:1, donde el doble signo de dos puntos cubre la primera de dos secuencias de ceros de igual longitud, y resulta igual a 2001:db8:0:0:1::1, otra representación válida de los mismos 128 bits. La última línea muestra una dirección local de enlace con su zona, `%eth0`, que indica a qué enlace pertenece.

En un host Linux con nftables, la segunda parte de este ejemplo es un conjunto de reglas que permite SSH desde una oficina a través de ambas versiones. El manual nft(8) describe `inet` como una familia híbrida cuyas tablas ven tanto paquetes IPv4 como IPv6, mientras que `ip saddr` lee la cabecera IPv4 e `ip6 saddr` la IPv6:

```
add table inet filter
add chain inet filter input { type filter hook input priority filter; }
add rule inet filter input ip saddr 198.51.100.0/24 tcp dport 22 accept
add rule inet filter input ip6 saddr 2001:db8:42::/48 tcp dport 22 accept
add rule inet filter input tcp dport 22 drop
```

Sin la cuarta línea, una conexión del administrador desde el prefijo IPv6 de la oficina no coincide con ninguna regla de aceptación y se descarta, un fallo que al menos resulta visible. El fallo más discreto es el contrario: un conjunto de reglas construido solo en la familia `ip`, en un host que también tiene una dirección IPv6, ni siquiera examina los paquetes IPv6.

La entrada del portfolio Aprovisionamiento de un segundo servidor a partir de código, de septiembre de 2026, aborda la misma exposición desde el aprovisionamiento. La primera máquina se había creado a mano y su cortafuegos se añadió después, lo que dejó un intervalo en el que un host nuevo con una política de contraseñas predeterminada era accesible desde Internet. El segundo solo se crea después de su cortafuegos: el play del cortafuegos se ejecuta antes que el del servidor y los cortafuegos de ambos proveedores se gestionan directamente mediante sus API REST. Esa página no menciona IPv6, así que las

reglas anteriores son un ejemplo del artículo y no las de ese host. Lo que añade este artículo es que el cortafuegos debe cubrir todas las direcciones que tendrá el host, IPv6 incluida, antes de que exista.

Dónde surgen los problemas

Un cortafuegos que solo conoce IPv4. Las reglas de la familia `ip` o las listas de permitidos que solo contienen prefijos IPv4 dejan todas las direcciones IPv6 del host a merced de lo que venga después. Escriba las reglas en la familia `inet` o mantenga sincronizadas las tablas `ip` e `ip6`, acompañe cada regla `ip saddr` de otra `ip6 saddr` y pruebe cada servicio conectándose tanto a su dirección IPv6 como a la IPv4.

Descartar ICMPv6 por completo. RFC 4890, sobre el filtrado de estos mensajes en cortafuegos, lo expresa claramente: «ICMPv6 es esencial para el funcionamiento de IPv6». Un router que no puede reenviar un paquete porque es demasiado grande para el siguiente enlace lo descarta y devuelve un mensaje de paquete demasiado grande (Packet Too Big). RFC 8201 señala que, si el filtrado impide el paso de esos mensajes, el origen nunca conoce la MTU de la ruta. El resultado es lo que denomina una conexión de agujero negro: el establecimiento de la conexión TCP tiene éxito y después la transferencia se queda bloqueada. RFC 4890 también enumera la solicitud y el anuncio de vecino (Neighbor Solicitation y Neighbor Advertisement) y la solicitud y el anuncio de router (Router Solicitation y Router Advertisement) como mensajes que un nodo no debe descartar para participar en una red IPv6. Acepte los tipos que nft llama `packet-too-big`, `nd-neighbor-solicit`, `nd-neighbor-advert`, `nd-router-solicit` y `nd-router-advert` antes de cualquier regla de descarte más amplia.

Comparar direcciones como texto. RFC 5952 da el ejemplo de una búsqueda de 2001:db8:0:0:1::1 en una hoja de cálculo que no encuentra una entrada escrita como 2001:db8::1:0:0:1. Eso importa cuando se consulta la lista antes de asignar una dirección nueva para comprobar que no esté en uso. Analice las direcciones antes de compararlas y almacene e imprima una forma canónica, como la forma abreviada que el módulo `ipaddress` de Python devuelve mediante `str()`.

Juntar una dirección y un puerto. 2001:db8::1:80 podría ser una dirección o una dirección seguida del puerto 80; RFC 5952 considera ambigua esa forma y recomienda corchetes, [2001:db8::1]:80, como en las URL. Escriba siempre los corchetes y, en el código, separe host y puerto por el corchete de cierre, no por el último signo de dos puntos.

Un prefijo de enlace que no sea /64. La autoconfiguración necesita que el prefijo y el identificador de interfaz sumen 128 bits. Con identificadores de 64 bits, RFC 4862 hace que el host ignore un prefijo anunciado de cualquier otra longitud, por lo que los hosts de un enlace cuyos anuncios de router contienen una /72 o una /56 no forman ninguna dirección a partir de ese prefijo. Planifique cada enlace como una /64 y divida las asignaciones mayores en /64, como hace la sesión anterior.

Direcciones locales de enlace sin su zona. Cada interfaz tiene una dirección en fe80::/10 y RFC 4291 limita esas direcciones a un solo enlace, de modo que, en un host con varias interfaces, la dirección por sí sola no indica cuál es. La documentación de Python señala que dos direcciones que solo difieren en su zona no son iguales. Conserve la zona junto a la dirección (fe80::1%eth0), o asigne a los servicios una dirección global o local única y reserve las locales de enlace para el propio enlace.

Lecturas adicionales

- RFC 8200: especificación del Protocolo de Internet, versión 6 (IPv6) — el estándar actual de IPv6, con el formato de cabecera, las cabeceras de extensión y las reglas de fragmentación y tamaño de paquete.
- RFC 4291: arquitectura de direccionamiento de IP versión 6 — tipos de direcciones, sus representaciones textuales, direcciones locales de enlace y de multidifusión e identificadores de interfaz de 64 bits.
- RFC 5952: recomendación para representar direcciones IPv6 como texto — la forma canónica de escribir una dirección y los problemas de admitir muchas representaciones.
- RFC 4862: autoconfiguración de direcciones IPv6 sin estado — cómo forman los hosts sus propias direcciones a partir de prefijos anunciados y comprueban que no estén duplicadas.
- RFC 4890: recomendaciones para filtrar mensajes ICMPv6 en cortafuegos — qué mensajes ICMPv6 debe dejar pasar un cortafuegos y por qué.
- ipaddress: biblioteca de manipulación de IPv4 e IPv6 — la referencia de Python para analizar, comparar y dividir direcciones y redes IPv6.



Glosario

Elija según sus restricciones reales

Describa la ruta de despliegue, los riesgos operativos y quiénes se encargan de las guardias. Podemos definir el alcance de un cambio con comprobaciones, una vía de recuperación y responsabilidades claras.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es IPv6?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por **D.**

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/ipv6/