Engineer

IPv6

версія інтернет-протоколу зі 128-бітними адресами

Що це таке

IPv6 (англійською Internet Protocol version 6) — це версія інтернет-протоколу, створена на заміну IPv4, тієї версії, яку описує RFC 791. Чинна специфікація — RFC 8200 від липня 2017 року, інтернет-стандарт (STD 86), що замінив RFC 2460. Зміна, яку помічають усі, — розмір адреси: 128 бітів замість 32 в IPv4. Увесь IPv4 вміщує 4 294 967 296 адрес, а одна підмережа IPv6 звичайного розміру, /64, вміщує квадрат цього числа.

Решта формату пакетів теж змінилася. Згідно з RFC 8200, частину полів заголовка IPv4 вилучено або зроблено необов'язковими, щоб здешевити звичайну обробку, необов'язкові відомості винесено в окремі заголовки розширень після основного, а поле Time to Live перейменовано на Hop Limit, і кожен вузол, що пересилає пакет, зменшує його на одиницю. Контрольної суми в заголовку немає. Маршрутизатори ніколи не ділять пакет на частини: в IPv6 фрагментує лише джерело, кожен канал має переносити пакети щонайменше на 1280 октетів, а відправникам наполегливо радять застосовувати Path MTU Discovery, щоб з'ясувати, наскільки великий пакет пройде шляхом.

Адресу записують як вісім груп по 16 бітів у шістнадцятковому вигляді, розділених двокрапками. RFC 4291 дозволяє відкидати початкові нулі в групі й замінювати одну суцільну послідовність нульових груп подвійною двокрапкою, яка може трапитися лише раз, тож 2001:db8:0:0:8:800:200c:417a можна записати як 2001:db8::8:800:200c:417a. Оскільки так одна адреса отримує багато записів, RFC 5952 від серпня 2010 року обирає один із них: малі літери, жодних початкових нулів і подвійна двокрапка на місці найдовшої послідовності нульових груп (першої, якщо дві однаково довгі), але ніколи не замість однієї групи. Окрім спеціальних адрес, як-от адрес петлі

та локальних адрес каналу, усі адреси й префікси в прикладах цієї статті взято з діапазонів для документації: з 2001:db8::/32, який відвів RFC 3849, а для IPv4 — з блоків, зарезервованих RFC 5737.

RFC 4291 змінює й те, до чого прив'язана адреса. Адреси призначають інтерфейсам, а не машинам, і один інтерфейс може мати кілька адрес одночасно. Кожен інтерфейс повинен мати локальну адресу каналу (link-local) у діапазоні fe80::/10, чинну лише в його власному каналі. Адреса петлі (loopback) — ::1. Широкомовлення (broadcast) немає: цю роботу перебрала групова розсилка (multicast), і ff02::1 досягає всіх вузлів каналу. Унікальні локальні адреси з fc00::/7, визначені в RFC 4193, призначені для локального зв'язку, зазвичай у межах мережі однієї організації. Глобальні одноадресні адреси (global unicast) використовують для зв'язку за межами локальної мережі. Для більшості адрес останні 64 біти — це ідентифікатор інтерфейсу, тому звичайний сегмент мережі — це /64.

Хости здебільшого нумерують себе самі. За автоконфігурацією адрес без збереження стану (stateless address autoconfiguration, RFC 4862) маршрутизатори оголошують префікси каналу в періодичних повідомленнях Router Advertisement, кожен хост поєднує префікс з ідентифікатором власного інтерфейсу, а перш ніж узяти результат у роботу, виконує Duplicate Address Detection і надсилає повідомлення Neighbor Solicitation, щоб переконатися, що жоден інший вузол не має цієї адреси. DHCPv6 — альтернатива для випадків, коли потрібен суворіший контроль над тим, яку саме адресу отримує хост, і обидва механізми можуть працювати поруч.

Застосуйте поняття до рішення

Застосуйте «Що таке IPv6?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Wi-Fi, фаєрволи і віддалений доступ налаштовані так, щоб працювати можна було звідусіль, а заходити — лише своїм.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Керувала 30 інстансами Ubuntu Linux VPS — 30 серверів Linux під управлінням, із планом відновлення

Керувала 30 інстансами Ubuntu Linux VPS Зв'язатися з нами →

Чому це важливо

Для багатьох команд IPv6 — не проєкт, який вони вирішують розпочати. Суть RFC 4862 у тому, що нікому не треба вводити адресу на хості вручну: поставте сервер у канал, де маршрутизатор оголошує префікс, і він сам може

отримати глобальну адресу IPv6. Тоді будь-який список дозволених адрес, правило брандмауера, парсер журналів чи перевірка справності, написані з думкою лише про IPv4, мають сліпу зону завбільшки з увесь новий адресний простір.

Цю сліпу зону легко проґавити, бо дві версії — окремі протоколи, які в мережі нічого між собою не ділять. Фільтр пакетів зіставляє їхні заголовки окремо, обмеження частоти, що рахує запити за адресами IPv4, ніколи не побачить клієнта IPv6, а тест, який підключається лише до 127.0.0.1, нічого не каже про ::1. Хост, закритий в одній версії й відкритий в іншій, на ділі не має брандмауера.

IPv6 також більше покладається на свої керувальні повідомлення, ніж IPv4. Виявлення сусідів (Neighbor Discovery), оголошення маршрутизаторів і Path MTU Discovery працюють через ICMPv6, тож звичка, перенесена з IPv4, — відкидати ICMP на межі мережі — ламає IPv6 так, що це може виявитися лише тоді, коли з’єднання почне передавати дані. А оскільки одну адресу можна записати багатьма правильними способами, код, що порівнює адреси як рядки, суперечитиме сам собі.

Як це виглядає на практиці

Стандартний модуль Python `ipaddress` розбирає будь-який правильний запис і виводить коротку форму, у якій початкові нулі відкинуто, а найдовшу послідовність нульових груп згорнуто, тож це швидкий спосіб спланувати префікс і перевірити, що два записи означають одну адресу. Сеанс нижче — ілюстрація цієї статті, виконана на Python 3.15 з адресами з документаційного префікса:

```
>>> 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'
```

Префікс /48 залишає вільними 80 бітів, а поділений на сегменти /64 дає їх 65 536, пронумерованих четвертою групою. Адреса, набрана великими літерами з усіма нулями, повертається в короткій формі, 2001:db8::1:0:0:1, де подвійна двокрапка покриває першу з двох однаково довгих послідовностей нулів, і вона дорівнює 2001:db8:0:0:1::1 — іншому, але правильному запису тих самих 128 бітів. Останній рядок показує локальну адресу каналу з її зоною — тим `%eth0`, що вказує, до якого каналу вона належить.

На хості Linux з nftables друга частина ілюстрації — набір правил, що пропускає SSH з офісу через обидві версії. Сторінка посібника nft(8) описує `inet` як гібридне сімейство для таблиць, що охоплюють і IPv4, і IPv6, тоді як `ip saddr` читає заголовок IPv4, а `ip6 saddr` — заголовок 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
```

Без четвертого рядка адміністратор, що підключається з IPv6-префікса офісу, не підпадає під жодне правило accept і отримує відмову, і ця помилка принаймні дає про себе знати одразу. Тихіша помилка — протилежна: набір правил, побудований лише в сімействі `ip`, на хості, який має ще й адресу IPv6, взагалі не дивиться на трафік IPv6.

Запис у портфоліо Підготувала другий сервер із коду за вересень 2026 року розглядає ту саму вразливість з боку підготовки хостів. Першу машину створили вручну, а брандмауер додали згодом, і так лишався проміжок часу, коли щойно створений хост зі стандартною політикою паролів був доступний з інтернету. Другу створюють лише після її брандмауера: плей брандмауера виконується перед плеєм сервера, а брандмауери обох провайдерів налаштовано безпосередньо через їхні REST API. Та сторінка не згадує IPv6, тож правила вище — ілюстрація цієї статті, а не правила того хоста. Ця стаття додає ось що: брандмауер має охоплювати кожну адресу, яку матиме хост, зокрема IPv6, ще до того, як хост з’явиться.

Де виникають проблеми

Брандмауер, що знає лише IPv4. Правила в сімействі `ip` або списки дозволених адрес лише з префіксами IPv4 залишають усі адреси IPv6 хоста на розсуд того, що стоїть далі. Пишіть набір правил у сімействі `inet` або тримайте таблиці `ip` та `ip6` узгодженими, додавайте до кожного правила `ip saddr` правило `ip6 saddr` і перевіряйте кожну службу, підключаючись до її адреси IPv6 так само, як до адреси IPv4.

Відкидання всього ICMPv6. RFC 4890, присвячений фільтрації цих повідомлень у брандмауерах, каже прямо: «ICMPv6 є необхідним для функціонування IPv6». Маршрутизатор, який не може переслати пакет, бо той завеликий для наступного каналу, відкидає його й повертає повідомлення Packet Too Big, а RFC 8201 зазначає, що коли фільтрація зупиняє ці повідомлення, джерело так і не дізнається MTU шляху. Наслідок RFC описує як з'єднання, що потрапило в чорну діру (black hole): рукостискання TCP вдається, а потім передавання зависає. RFC 4890 також називає Neighbor Solicitation і Advertisement та Router Solicitation і Advertisement повідомленнями, яких вузол не повинен відкидати, якщо має брати участь у мережі IPv6. Приймайте типи, які nft називає `packet-too-big`, `nd-neighbor-solicit`, `nd-neighbor-advert`, `nd-router-solicit` і `nd-router-advert`, раніше за будь-яке ширше правило drop.

Порівняння адрес як тексту. RFC 5952 наводить приклад електронної таблиці, де пошук 2001:db8:0:0:1::1 не знаходить запису 2001:db8::1:0:0:1, і це важить, коли список переглядають перед видачею нової адреси, щоб переконатися, що її ще ніхто не використовує. Розбирайте адреси, перш ніж порівнювати, і зберігайте та виводьте одну канонічну форму, наприклад коротку, яку `ipaddress` у Python повертає з `str()`.

Адреса й порт, записані разом. 2001:db8::1:80 може бути адресою або адресою з портом 80; RFC 5952 називає таку форму неоднозначною й радить квадратні дужки, [2001:db8::1]:80, як в URL. Завжди пишіть дужки, а в коді відокремлюйте хост від порту за закривною дужкою, а не за останньою двокрапкою.

Префікс каналу, що не є /64. Автоконфігурація вимагає, щоб префікс та ідентифікатор інтерфейсу разом давали 128 бітів, і за 64-бітних ідентифікаторів RFC 4862 велить хостові ігнорувати оголошений префікс будь-якої іншої довжини, тож хости в каналі, де повідомлення Router

Advertisement несуть /72 або /56, не утворюють із цього префікса жодної адреси. Плануйте кожен канал як /64, а більші виділення діліть на блоки /64, як це робить сеанс вище.

Локальні адреси каналу без зони. Кожен інтерфейс має адресу в fe80::/10, і RFC 4291 обмежує такі адреси одним каналом, тож на хості з кількома інтерфейсами сама адреса не каже, який канал мається на увазі. Документація Python зазначає, що дві адреси, які відрізняються лише зоною, не рівні. Тримайте зону разом з адресою (fe80::1%eth0) або давайте службам глобальну чи унікальну локальну адресу, а локальні адреси каналу залишайте для самого каналу.

Що почитати далі

- RFC 8200: Internet Protocol, Version 6 (IPv6) Specification — чинний стандарт IPv6 із форматом заголовка, заголовками розширень і правилами фрагментації та розміру пакетів.
- RFC 4291: IP Version 6 Addressing Architecture — типи адрес, текстові форми, локальні адреси каналу й групові адреси, а також 64-бітні ідентифікатори інтерфейсів.
- RFC 5952: A Recommendation for IPv6 Address Text Representation — канонічний спосіб записати адресу і проблеми, яких завдає безліч записів однієї адреси.
- RFC 4862: IPv6 Stateless Address Autoconfiguration — як хости утворюють власні адреси з оголошених префіксів і перевіряють їх на дублікати.
- RFC 4890: Recommendations for Filtering ICMPv6 Messages in Firewalls — які повідомлення ICMPv6 брандмауер мусить пропускати і чому.
- ipaddress: IPv4/IPv6 manipulation library — довідка Python про розбір, порівняння й поділ адрес і мереж IPv6.

Словник

Обирайте за фактичними умовами

Опишіть шлях розгортання, експлуатаційні ризики та чергову команду. Ми можемо визначити зміну з перевірками, шляхом відновлення та чіткою відповідальністю.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке IPv6?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

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

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/ipv6/