Engineer

IPv6

versionen af internetprotokollen med adresser på 128 bit

Hvad det er

IPv6, skrevet helt ud Internet Protocol version 6, er den version af internetprotokollen, der blev udformet som afløser for IPv4, den version, som RFC 791 beskriver. Den gældende specifikation er RFC 8200 fra juli 2017, en internetstandard (STD 86), der erstattede RFC 2460. Den ændring, alle lægger mærke til, er adressens størrelse: 128 bit i stedet for IPv4's 32. Hele IPv4 rummer 4.294.967.296 adresser; et enkelt IPv6-subnet af den sædvanlige størrelse, en /64, rummer det tal i anden potens.

Også resten af pakkeformatet er lavet om. Ifølge RFC 8200 er nogle af IPv4's headerfelter fjernet eller gjort valgfri for at gøre den almindelige behandling billigere, valgfri oplysninger ligger i separate udvidelsesheadere efter selve headeren, og feltet Time to Live hedder nu Hop Limit og tælles én ned af hver node, der videresender pakker. Der er ingen kontrolsum i headeren. Routere deler aldrig en pakke op: I IPv6 fragmenteres der kun hos afsenderen, hvert link skal kunne bære pakker på mindst 1280 oktetter, og afsendere opfordres kraftigt til at bruge Path MTU Discovery til at finde ud af, hvor store pakker en sti kan bære.

En adresse skrives som otte grupper à 16 bit i hexadecimal, adskilt af koloner. RFC 4291 tillader, at man udelader foranstillede nuller i en gruppe og erstatter én sammenhængende række af nulgrupper med et dobbelt kolon, som kun må optræde én gang, så 2001:db8:0:0:8:800:200c:417a kan skrives 2001:db8::8:800:200c:417a. Fordi det giver mange måder at skrive den samme adresse på, vælger RFC 5952 fra august 2010 én af dem: små bogstaver, ingen foranstillede nuller og det dobbelte kolon på den længste række af nulgrupper (den første, hvis to er lige lange), aldrig på en enkelt gruppe. Bortset fra særlige adresser som loopback-adresserne og de link-lokale kommer alle

eksempeladresser og -præfikser i denne artikel fra et dokumentationsområde: 2001:db8::/32, som RFC 3849 har sat af, eller for IPv4 de blokke, som RFC 5737 reserverer.

RFC 4291 ændrer også, hvad en adresse sidder på. Adresser tildeles grænseflader, ikke maskiner, og én grænseflade kan have flere ad gangen. Hver grænseflade skal have en link-lokal adresse under fe80::/10, som kun gælder på dens eget link. Loopback-adressen er ::1. Der findes ingen broadcast; multicast har overtaget den opgave, og ff02::1 når alle noder på linket. Unikke lokale adresser under fc00::/7, defineret i RFC 4193, er beregnet til lokal kommunikation, typisk inden for én organisations net. Globale unicast-adresser bruges til kommunikation ud over det lokale net. For de fleste adresser er de sidste 64 bit grænsefladens identifikator (interface identifier), og derfor er et almindeligt netværkssegment en /64.

Værter nummererer som regel sig selv. Med tilstandsløs autokonfiguration af adresser (stateless address autoconfiguration, RFC 4862) annoncerer routere et links præfikser i periodiske Router Advertisements, hver vært kombinerer et præfiks med identifikatoren for sin egen grænseflade, og før den tager resultatet i brug, kører den Duplicate Address Detection og sender Neighbor Solicitation-beskeder for at tjekke, at ingen anden node har adressen. DHCPv6 er alternativet, når man har brug for strammere kontrol med de præcise adresser, og de to kan køre side om side.

Brug begrebet i en beslutning

Brug »Hvad er IPv6?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Wi-Fi, firewalls og fjernadgang sat op, så medarbejdere kan arbejde hvor som helst, og kun de rette kommer ind.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har administreret 30 Ubuntu Linux VPS-instanser — 30 Linux-servere driftet, med en plan for gendannelse

Har administreret 30 Ubuntu Linux VPS-instanser Kontakt os →

Hvorfor det betyder noget

For mange teams er IPv6 ikke et projekt, de beslutter sig for at gå i gang med. Pointen med RFC 4862 er, at ingen behøver at taste en adresse ind på en vært: Sæt en server på et link, hvor en router annoncerer et præfiks, og den kan selv få

en global IPv6-adresse. Enhver tilladelsesliste, firewallregel, logparser eller ethvert sundhedstjek, der er skrevet med kun IPv4 i tankerne, har så en blind vinkel, der er præcis lige så stor som det nye adresserum.

Den blinde vinkel er let at overse, fordi de to versioner er separate protokoller, der intet deler på ledningen. Et pakkefilter matcher deres headere hver for sig, en hastighedsbegrænsning, der tæller pr. IPv4-adresse, ser aldrig en IPv6-klient, og en test, der kun forbinder til 127.0.0.1, siger intet om ::1. En vært, der er låst af på den ene version og åben på den anden, har i praksis ingen firewall.

IPv6 læner sig også mere op ad sine kontrolbeskeder, end IPv4 gør. Neighbor Discovery, routerannonceringer og Path MTU Discovery går alle over ICMPv6, så en vane, man tager med fra IPv4, nemlig at droppe ICMP i kanten af nettet, ødelægger IPv6 på måder, der måske først viser sig, når en forbindelse begynder at flytte data. Og fordi én adresse kan staves på mange gyldige måder, bliver kode, der sammenligner adresser som tekst, uenig med sig selv.

Sådan ser det ud i praksis

Pythons standardmodul `ipaddress` fortolker enhver gyldig stavemåde og udskriver en kort form, hvor foranstillede nuller er fjernet og den længste række af nulgrupper er slået sammen, så det er en hurtig måde at planlægge et præfiks på og at tjekke, at to stavemåder er den samme adresse. Sessionen herunder er denne artikels illustration, kørt på Python 3.15 med adresser fra dokumentationspræfikset:

```
>>> 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'
```

En /48 efterlader 80 bit, og delt op i /64-segmenter giver den 65.536 af dem, nummereret i den fjerde gruppe. Adressen, der er tastet med store bogstaver og alle nuller skrevet ud, kommer tilbage i den korte form, 2001:db8::1:0:0:1, hvor det dobbelte kolon dækker den første af to lige lange rækker af nuller, og den er lig med 2001:db8:0:0:1::1, en anden, men gyldig stavemåde for de samme 128 bit. Den sidste linje viser en link-lokal adresse med sin zone, det `%eth0`, der siger, hvilket link den hører til.

På en Linux-vært med nftables er anden del af illustrationen et regelsæt, der lukker SSH ind fra et kontor over begge versioner. Manualen nft(8) beskriver `inet` som en hybrid familie til tabeller, der rummer både IPv4 og IPv6, mens `ip saddr` læser IPv4-headeren og `ip6 saddr` IPv6-headeren:

```
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
```

Uden den fjerde linje matcher en administrator, der forbinder fra kontorets IPv6-præfiks, ingen af accept-reglerne og bliver afvist, og den fejl melder sig i det mindste med det samme. Den stille fejl er den omvendte: Et regelsæt, der kun er bygget i familien `ip`, på en vært, der også har en IPv6-adresse, kigger slet ikke på IPv6-trafikken.

Porteføljeposten Har provisioneret en ekstra server ud fra kode fra september 2026 tager den samme eksponering op fra provisioneringssiden. Den første maskine var oprettet i hånden og havde fået sin firewall bagefter, hvilket efterlod et tidsvindue, hvor en helt ny vært med en standardpolitik for adgangskoder kunne nås fra internettet. Den anden oprettes først efter sin firewall: Firewall-playet kører før server-playet, og begge udbyderes firewalls skrives direkte mod deres REST-API'er. Siden nævner ikke IPv6, så reglerne ovenfor er denne artikels illustration og ikke den værts regler. Det, denne artikel lægger til, er, at firewallen skal dække alle de adresser, værten kommer til at have, IPv6 medregnet, før værten overhovedet findes.

Hvor det går galt

En firewall, der kun kender IPv4. Regler skrevet i familien `ip` eller tilladelseslister med kun IPv4-præfikser overlader alle værtens IPv6-adresser til det, der kommer bagefter. Skriv regelsættet i familien `inet`, eller hold `ip` - og `ip6` -tabeller i takt, sæt en `ip6 saddr` -regel ved siden af hver `ip saddr` -regel, og test hver tjeneste ved at forbinde til dens IPv6-adresse såvel som til dens IPv4-adresse.

At droppe al ICMPv6. RFC 4890, der handler om at filtrere disse beskeder i firewalls, siger det ligeud: »ICMPv6 er afgørende for, at IPv6 fungerer«. En router, der ikke kan videresende en pakke, fordi den er for stor til det næste link, kasserer den og sender en Packet Too Big-besked tilbage, og RFC 8201 bemærker, at hvis filtrering stopper de beskeder, får afsenderen aldrig stiens MTU at vide. Resultatet er det, RFC'en kalder en black hole-forbindelse: TCP-håndtrykket lykkes, og så hænger overførslen. RFC 4890 nævner også Neighbor Solicitation og Advertisement og Router Solicitation og Advertisement som beskeder, en node ikke må droppe, hvis den skal kunne deltage i et IPv6-netværk. Accepter de typer, som nft kalder `packet-too-big`, `nd-neighbor-solicit`, `nd-neighbor-advert`, `nd-router-solicit` og `nd-router-advert`, før en bredere drop-regel.

At sammenligne adresser som tekst. RFC 5952 giver et eksempel med et regneark, hvor en søgning efter 2001:db8:0:0:1::1 ikke finder en post, der er skrevet 2001:db8::1:0:0:1, og det betyder noget, når listen gennemsøges, før en ny adresse tildeles, for at tjekke, at den ikke allerede er i brug. Fortolk adresserne, før du sammenligner dem, og gem og udskriv én kanonisk form, for eksempel den korte form, som Pythons `ipaddress` returnerer fra `str()`.

En adresse og en port skrevet i ét. 2001:db8::1:80 kan være en adresse eller en adresse efterfulgt af port 80; RFC 5952 kalder den form tvetydig og anbefaler firkantede parenteser, [2001:db8::1]:80, som i URL'er. Skriv altid parenteserne, og del i kode vært og port ved den afsluttende parentes i stedet for ved det sidste kolon.

Et linkpræfiks, der ikke er en /64. Autokonfiguration kræver, at præfikset og grænsefladens identifikator tilsammen giver 128 bit, og med identifikatorer på 64 bit ignorerer en vært ifølge RFC 4862 et annonceret præfiks af enhver anden

længde, så værter på et link, hvis Router Advertisements bærer en /72 eller en /56, danner ingen adresse ud fra det præfiks. Planlæg hvert link som en /64, og del større tildelinger op i /64-blokke, som sessionen ovenfor gør.

Link-lokale adresser uden deres zone. Hver grænseflade har en adresse i fe80::/10, og RFC 4291 begrænser de adresser til et enkelt link, så på en vært med flere grænseflader siger adressen alene ikke, hvilket link der menes. Pythons dokumentation bemærker, at to adresser, der kun adskiller sig i deres zone, ikke er ens. Hold zonen sammen med adressen (fe80::1%eth0), eller giv tjenester en global eller unik lokal adresse, og lad de link-lokale blive på linket.

Læs mere

- RFC 8200: Internet Protocol, Version 6 (IPv6) Specification — den gældende IPv6-standard med headerformatet, udvidelsesheaderne og reglerne for fragmentering og pakkestørrelse.
- RFC 4291: IP Version 6 Addressing Architecture — adressetyperne, tekstformerne, link-lokale adresser og multicastadresser samt identifikatorer på 64 bit for grænseflader.
- RFC 5952: A Recommendation for IPv6 Address Text Representation — den kanoniske måde at skrive en adresse på og de problemer, som mange stavemåder skaber.
- RFC 4862: IPv6 Stateless Address Autoconfiguration — hvordan værter danner deres egne adresser ud fra annoncerede præfikser og tjekker dem for dubletter.
- RFC 4890: Recommendations for Filtering ICMPv6 Messages in Firewalls — hvilke ICMPv6-beskeder en firewall skal lukke igennem, og hvorfor.
- ipaddress: IPv4/IPv6 manipulation library — Pythons opslagsværk for at fortolke, sammenligne og opdele IPv6-adresser og -netværk.

Ordliste

Vælg ud fra jeres faktiske vilkår

Medbring udrulningsforløbet, driftsrisiciene og dem, der har vagten. Vi kan afgrænse en ændring med kontroller, en vej tilbage og tydeligt ansvar.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er IPv6?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

welcome@engineer.company +45 50 10 61 84 København, Danmark

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/ipv6/