

IPv6-foredrag

Pent brukt 19-åring

Trond Endrestøl

Fagskolen Innlandet, IT-avdelingen

5. mars 2015

- Filene til foredraget er tilgjengelig gjennom:
 - Subversion: `svn co svn://svn.ximalas.info/ipv6-foredrag`
 - Web: svnweb.ximalas.info/ipv6-foredrag
 - Begge metodene er tilgjengelig med både IPv4 og IPv6
- [ipv6-foredrag.foredrag.pdf](#) vises på lerretet
- [ipv6-foredrag.handout.pdf](#) er mye bedre for publikum å se på egenhånd
- [ipv6-foredrag.handout.2on1.pdf](#) og [ipv6-foredrag.handout.4on1.pdf](#) er begge velegnet til utskrift
- *.169.pdf-filene er i 16:9-format
- *.1610.pdf-filene er i 16:10-format

Foredragets filer II

- Foredraget er mekka ved hjelp av [GNU Emacs](#), [AUCTEX](#), [pdfTEX](#) fra [MiKTeX](#), [LATEX](#)-dokumentklassa [beamer](#), [Dia](#), [GIMP](#), [Inkscape](#), [Wireshark](#), [Subversion](#), [TortoiseSVN](#) og [Adobe Reader](#)
- Hovedfila bærer denne identifikasjonen:
`$Ximalas: trunk/ipv6-foredrag.tex 139 2015-03-05 07:14:09Z trond $`
- Driverfila for denne PDF-fila bærer denne identifikasjonen:
`$Ximalas: trunk/ipv6-foredrag.handout.1610.tex 78 2013-12-04 09:53:24Z trond $`
- Copyright © 2015 Trond Endrestøl
- Dette verket er lisensiert med: [Creative Commons](#), [Navngivelse-DelPåSammeVilkår 3.0 Norge](#) (CC BY-SA 3.0)



Oversikt av hele foredraget

Del 1: Kort om IPv6

- 1 Hva er IPv6?
- 2 Antall adresser
- 3 Hvorfor trenger vi IPv6?
- 4 Hvorfor brukes ikke IPv6?
- 5 Andre nyttige ting ved IPv6
- 6 IPv6 ved Fagskolen Innlandet
- 7 IPv6 andre steder i Norge
- 8 IPv6 i utlandet
- 9 Google Chrome og IPvFoo
- 10 Mozilla Firefox og IPvFox

11 IPv6-header

- Flow Label

12 Utvidelsesheadere

- Hop-by-hop Options Header
- Destination Options Header
- Routing Header
- Fragment Header
- Authentication Header
- Encapsulating Security Payload
- Mobility Header

Oversikt av hele foredraget

Del 3: IPv6 over Ethernet

13 IPv6 over Ethernet

14 IPv6 over andre lag-2-typer

Oversikt av hele foredraget

Del 4: Grunnleggende om adresser

- 15 Grunnleggende om adresser
- 16 Adressedemo
- 17 MAC-48-adresser
- 18 Modda IEEE EUI-64-format
- 19 Manuell grensesnittidentifikator
- 20 Tilfeldig grensesnittidentifikator
- 21 Spesialadresser
- 22 Duplicate Address Detection — DAD

Oversikt av hele foredraget

Del 5: Adresstyper

- 23 Adresstyper
- 24 Link-local-adresser
- 25 Site-local-adresser
- 26 Offentlige unicast-adresser
- 27 Unike, lokale, aggregerbare adresser
- 28 Anycast-adresser
- 29 Multicast-adresser

Oversikt av hele foredraget

Del 6: DNS

30 AAAA og PTR

31 A6

Oversikt av hele foredraget

Del 7: ICMPv6

- 32 ICMPv6
- 33 Multicast Listener Discovery
- 34 Neighbor Discovery
- 35 Router Renumbering
- 36 Node Information
- 37 Inverse Neighbor Discovery
- 38 Version 2 Multicast Listener Report
- 39 Mobile IPv6
- 40 SEcure Neighbor Discovery (SEND)
- 41 Experimental Mobility Type
- 42 Multicast Router Discovery
- 43 FMIPv6
- 44 RPL Control Message
- 45 ILNPv6 Locator Update Message
- 46 Duplicate Address

Oversikt av hele foredraget

Del 8: Neighbor Discovery

- 47 Router Solicitation
- 48 Router Advertisement
- 49 Neighbor Solicitation
- 50 Neighbor Advertisement
- 51 Redirect

Oversikt av hele foredraget

Del 9: DHCPv6

52 DHCPv6

53 Meldinger

54 DHCP Unique Identifier

55 Identity association

56 Identity association identifier

Oversikt av hele foredraget

Del 10: Avansert multicast

- 57 Multicastflaggene
- 58 Når T er satt til 1
- 59 Når PT er satt til 11
- 60 Når RPT er satt til 111

61 Cisco IOS

- IPv6-unicast-routing
- IPv6-multicast-routing
- ACL-er
- DHCPv6
- Sperre for fremmed routerannonsering
- Sperre for falske DHCPv6-servere
- Kombinert ACL for kantporter

62 OS-konfig

Oversikt av hele foredraget

Del 12: Noen RFC-er om IPv6

63 Noen RFC-er om IPv6

Kort om IPv6

Oversikt over del 1: Kort om IPv6

- 1 Hva er IPv6?
- 2 Antall adresser
- 3 Hvorfor trenger vi IPv6?
- 4 Hvorfor brukes ikke IPv6?
- 5 Andre nyttige ting ved IPv6
- 6 IPv6 ved Fagskolen Innlandet
- 7 IPv6 andre steder i Norge
- 8 IPv6 i utlandet
- 9 Google Chrome og IPvFoo
- 10 Mozilla Firefox og IPvFox

Kort om IPv6

Hva er IPv6?

- En lag-3-protokoll ment å erstatte IPv4
- Har eksistert siden desember 1995, først spesifisert i [RFC 1883](#)
- Enkel grunnheader med fast lengde
- Flere utvidelsesheadere, riktig rekkefølge er viktig
- 128-bit adresser
- Ny versjon av ICMP: ICMPv6
- ARP og RARP for IPv6 er en del av ICMPv6
 - Ikke nødvendig med ekstra lim for adressene i lagene 2 og 3
- Ny versjon av DHCP: DHCPv6
- Automatisk adressekonfigurasjon *uten* bruk av DHCPv6

- Totalt antall IPv6-adresser:
 - $2^{128} = 340.282.366.920.938.463.463.374.607.431.768.211.456$
- Bare 1/8 kan brukes til offentlige unicast-adresser:
 - $2^{125} = 42.535.295.865.117.307.932.921.825.928.971.026.432$
- Fortsatt er det mange flere IPv6-unicast-adresser enn det er IPv4-adresser:
 - $2^{32} = 4.294.967.296$
- Mindre enn **3.702.258.688** IPv4-adresser kan bli brukt som offentlige IPv4-unicast-adresser
- Se Tronds utregning fra juli 2012:
<http://ximalas.info/2012/07/20/how-many-ipv4-addresses-are-there/>

Kort om IPv6

Hvorfor trenger vi IPv6?

- Mobilmarkedet viser en enorm vekst: smarttelefoner, nettbrett m.m.
- Verden går tom for offentlige IPv4-adresser
- «[IPokalypsen](#)» er her!
- [IANA](#) gikk tom [3. februar 2011](#)
 - [APNIC](#) gikk tom [19. april 2011](#)
 - [RIPE](#) gikk tom [14. september 2012](#)
 - [LACNIC](#) gikk tom [10. juni 2014](#)
- Dersom disse [RIR](#)-ene oppfører seg pent:
 - [ARIN](#) kan holde på til [20. mai 2015](#)
 - [AFRINIC](#) kan holde på til [13. januar 2019\(!\)](#)

Kort om IPv6

Hvorfor brukes ikke IPv6?

- Markedskreftene bestemmer
- «Vente-og-se»-holdning
- Stikker hodet ned i sanda
- Store selskaper:
 - Kjøper opp små selskaper og hamstrer IPv4-blokker
 - Kjøper IPv4-blokker på ettermarkedet/konkursbo:
 - Microsoft → \$7,5 mill. → Nortel → 666.624 IPv4-adresser → Microsoft
 - Altibox → \$1,3 mill. → ? → 130.000 IPv4-adresser → Altibox
 - Prisen for brukte IPv4-adresser har gått ned fra \$11,25/adresse til \$10/adresse

Kort om IPv6

Hvorfor brukes ikke IPv6?

- Telebransjen satser fortsatt hardt på IPv4:
 - (Edge) NAT i CPE (RFC 1631)
 - Carrier-Grade NAT i stamnett (RFC 6264)
 - Shared Address Space etter behov i stamnett (100.64.0.0/10) (RFC 6598)
- Glem det!
- Ende-til-ende-konnektivitet oppnås best uten noen former for adresseoversettelse
- Før eller siden blir CGN for kostbart og komplisert å vedlikeholde
- 3G og 4G/LTE klarer kanskje å øke IPv6-presset (RFC 6459)
- IPv6 er det eneste tilgjengelige og realistiske alternativet til IPv4

- Hierarkisk adressestruktur
- Enklere planlegging av subnett sammenlignet med IPv4
 - De fleste IPv6-subnett bruker et 64-bit prefiks
 - Autokonfigurasjon *krever* et 64-bit prefiks
 - Fast prefikslengde på 64 bit er *ikke* et absolutt krav
 - DHCPv6 eller manuell konfigurasjon brukes når prefikslengda er ulik 64 bit

- Kortere rutingtabeller

- Uninett annonserer disse IPv4-subnettene med BGP:

- 78.91.0.0/16, 128.39.0.0/16, 129.177.0.0/16,
129.240.0.0/15, 129.242.0.0/16, 144.164.0.0/16,
151.157.0.0/16, 152.94.0.0/16, 156.116.0.0/16,
157.249.0.0/16, 158.36.0.0/14, 161.4.0.0/16,
193.156.0.0/15, 192.111.33.0/24, 192.133.32.0/24,
192.146.238.0/23

- Uninett trenger bare å annonsere dette IPv6-prefikset:
• 2001:700::/32

- Sjekksm er overlatt til høyere og lavere lag
- Fragmentering skal gjøres hos avsender, og ikke underveis
 - Avsender må sjekke veien lengre fremme og måle smaleste krøtteri
 - Path Maximum Transmission Unit Discovery (Path MTU, PMTUD)
- IPsec ble spesifisert som en del av IPv6
 - Finnes også for IPv4
 - Må konfigureres før den begynner å virke
 - Tilbyr:
 - Kryptert overføring (ESP), og/eller
 - Bekreftelse av avsenders identitet og beskyttelse mot gjentakelse («replay») (AH)
 - Ble omgjort fra krav til anbefaling for IPv6 av [RFC 6434](#)

- 1994: Tildelt 128.39.174.0/24 av Uninett
- 1. juni 2005: Ny IT-ansvarlig, yours truly
- Høsten 2005: Fikk reservert IPv4-serien 128.39.172.0/23
- Påska 2006: Fikk reservert IPv6-serien 2001:700:1100::/48
- Før og etter pinsehelga 2006: Fiberlinjer fra serverrommet og til sentrale punkter i hver etasje i hovedbygningen
- Sommeren 2006: Nytt Cisco-gear som Catalyst 3560G og 2960 ([Cisco IOS 12.2\(25\)SEB4](#))
 - 128.39.46.8/30 ble linknettet mellom HiG/Uninett og FSI
 - 128.39.46.9 ble brukt ved HiG
 - 128.39.46.10 ble brukt ved FSI
 - 128.39.174.0/24 ble delt opp i flere subnett og satt opp som servernett og ansattnett, m.m.
 - 128.39.172.0/24 ble delt opp i flere subnett og satt opp som nett for datalab
 - 128.39.173.0/24 ble satt opp for inntil 252 IPv4-klienter på trådløst studentnett

- 6. september 2006: IPv6-linknettet 2001:700:0:11D::/64 ble aktivert mellom HiG/Uninett og FSI
 - 2001:700:0:11D::1 ble brukt ved HiG
 - 2001:700:0:11D::2 ble brukt ved FSI
- Samme dag ble IPv6 innført for FSI-VLAN-ene 20, 30, 70 og 80:
 - FSI-VLAN 20: 2001:700:1100:1::/64 (ytre servernett)
 - FSI-VLAN 30: 2001:700:1100:2::/64 (indre servernett)
 - FSI-VLAN 70: 2001:700:1100:3::/64 (IT-kontornett)
 - FSI-VLAN 80: 2001:700:1100:4::/64 (IT-lekenett)
- Andre FSI-VLAN fikk IPv6 i ukene og månedene etterpå
- Sommeren 2007: [Genererte](#) og frivillig [registrerte](#) ULA-serien [FD5C:14CF:C300::/48](#)
 - Brukes i FSI-VLAN for internt bruk
 - Fikk første HP-skriver med IPv6-støtte og ville bruke IPv6
 - Noen år senere: IPv6-adresser på kantswitchene med [Cisco IOS 12.2\(40\)SE](#)

- Høsten 2010: Enda en IPv4-serie ble innført: 128.39.194.0/24
 - 128.39.194.0/24 brukes til datalab med samme subnetting (inndeling) som den gamle 128.39.172.0/24-serien hadde i 2006
 - 128.39.172.0/23 brukes nå for inntil 508 IPv4-klienter på trådløst studentnett
- Våren 2014: Tok i bruk nye linknett fordi fig-gsw.fig.ol.no ble tilkoblet gjovik-gw1.uninett.no
 - IPv4-linknett: 128.39.70.168/30
 - 128.39.70.169 brukes ved HiG
 - 128.39.70.170 brukes ved FSI
 - IPv6-linknett: 2001:700:0:8074::/64
 - 2001:700:0:8074::1 brukes ved HiG
 - 2001:700:0:8074::2 brukes ved FSI
- Vinteren 2015: La om datalabseriene, siden antallet av datalab er skikkelig knøttete

- I dag er de fleste brukere ved FSI kasta over i nettet til Oppland fylkeskommune (OFK)
- Dette skjedde etter ombygginga av skolen i 2011–2012
- Andreklasse data er velsigna med å kunne velge mellom FSI- og OFK-nettene
- Andreklasse data velger som regel det førstnevnte, vanligvis FSI-VLAN 40 som tilbyr 128.39.194.0/26 og 2001:700:1100:8001::/64
- Førsteklasse data ønsker det samme tilbudet; så vi får se ...

- Alle FSI-VLAN har både IPv4- og IPv6-adresser (dual-stack)
- FSI-VLAN med offentlige IPv4-adresser, bruker offentlige IPv6-adresser fra 2001:700:1100::/48-serien
- FSI-VLAN med private IPv4-adresser ([RFC 1918](#)), bruker private IPv6-adresser fra FD5C:14CF:C300::/48-serien
- Private adresser brukes for alt utstyr som ikke har behov for internettforbindelse:
 - Switcher
 - Med unntak av kjerneswitchen som er L3-router for nettverket ved FSI
 - Gammel WLAN-kontroller (AIR-WLC4402-25-K9) og gamle basestasjoner (AIR-LAP1231G-E-K9)
 - Den nyeste WLAN-kontrolleren (AIR-CT5508-K9) og de nyere basestasjonene (AIR-LAP1242AG-E-K9) er dytta inn i OFK-nettet
 - UPS-er
 - Skrivere
 - VPN-klienter

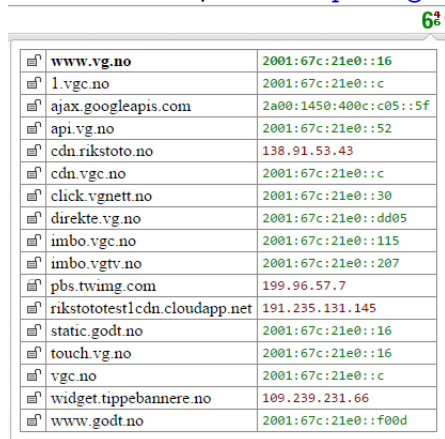
Kort om IPv6

IPv6 andre steder i Norge

- Mesteparten av Uninett og deres kunder bruker IPv6
- [Oppland FK](#) har ingen planer om å innføre IPv6
- [Hordaland FK](#) har satt en IPv6-adresse på webserveren deres, 2a02:20a0:0:3::81:130
- [Vest-Agder FK](#) har også satt en IPv6-adresse på webserveren deres, 2001:67c:28ac:1::2
- [Nasjonal kommunikasjonsmyndighet](#) har satt en IPv6-adresse på webserveren deres, 2a02:228:105:d000::10
- [VG](#) tok IPv6 i bruk i 2010, 2001:67c:21e0::16
- [Amedia AS'](#) (tidl. A-pressen) mange (nett)aviser ble tilgjengelig med IPv6 samtidig med VG

- Facebook er tilgjengelig med IPv6, 2a03:2880:2130:cf05:face:b00c:0:1 og 2a03:2880:2110:df07:face:b00c:0:1
- Google er tilgjengelig med IPv6, 2a00:1450:400c:c00::5e, 2a00:1450:400c:c00::8a og 2a00:1450:4010:c04::63
- Snapchat er tilgjengelig med IPv6, 2a00:1450:400c:c00::79
- LinkedIn er tilgjengelig med IPv6, 2620:109:c007:102::5be1:f881

- **IPv6** for Google Chrome lar deg se hvilke IP-adresser som innholdet ble hentet fra
- Her er et eksempel fra <http://vg.no/>:



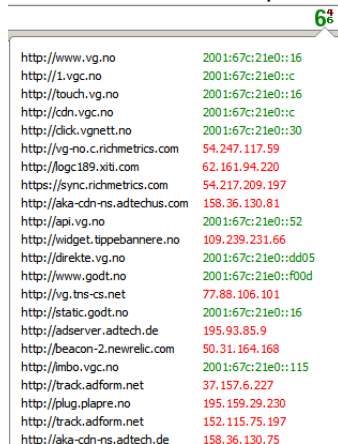
A screenshot of a web browser window showing a list of IPv6 addresses for the domain vg.no. The list is displayed in a table with two columns: the domain name and the corresponding IPv6 address. The addresses are color-coded: green for IPv6 and red for IPv4. The list includes various subdomains and the main domain, as well as CDNs and static content providers.

www.vg.no	2001:67c:21e0::16
l.vgc.no	2001:67c:21e0::c
ajax.googleapis.com	2a00:1450:400c:c05::5f
api.vg.no	2001:67c:21e0::52
cdn.rikstoto.no	138.91.53.43
cdn.vgc.no	2001:67c:21e0::c
click.vgnett.no	2001:67c:21e0::30
direkte.vg.no	2001:67c:21e0::dd05
imbo.vgc.no	2001:67c:21e0::115
imbo.vgtv.no	2001:67c:21e0::207
pbs.twimg.com	199.96.57.7
rikstototest1cdn.cloudapp.net	191.235.131.145
static.godt.no	2001:67c:21e0::16
touch.vg.no	2001:67c:21e0::16
vgc.no	2001:67c:21e0::c
widget.tippebannere.no	109.239.231.66
www.godt.no	2001:67c:21e0::f00d

Kort om IPv6

Mozilla Firefox og IPv6

- **IPv6** gjør det samme for Mozilla Firefox som IPv4 gjør for Google Chrome
- Her er enda et eksempel fra <http://vg.no/>:



A screenshot of a network packet capture tool, likely Wireshark, showing a list of network connections. The connections are listed in two columns: the domain name and the corresponding IPv6 address. The addresses are color-coded: green for standard IPv6 addresses and red for specific IPv4-mapped IPv6 addresses. A small green icon with the number '6' is visible in the top right corner of the screenshot.

http://www.vg.no	2001:67c:21e0::16
http://1.vgc.no	2001:67c:21e0::c
http://touch.vg.no	2001:67c:21e0::16
http://cdn.vgc.no	2001:67c:21e0::c
http://click.vgnett.no	2001:67c:21e0::30
http://vg-no.c.richmetrics.com	54.247.117.59
http://logc189.xiti.com	62.161.94.220
https://sync.richmetrics.com	54.217.209.197
http://aka-cdn-ns.adtechus.com	158.36.130.81
http://api.vg.no	2001:67c:21e0::52
http://widget.tippebannere.no	109.239.231.66
http://direkte.vg.no	2001:67c:21e0::dd05
http://www.godt.no	2001:67c:21e0::f00d
http://vg.tns-cs.net	77.88.106.101
http://static.godt.no	2001:67c:21e0::16
http://adserver.adtech.de	195.93.85.9
http://beacon-2.newrelic.com	50.31.164.168
http://imbo.vgc.no	2001:67c:21e0::115
http://track.adform.net	37.157.6.227
http://plug.plapre.no	195.159.29.230
http://track.adform.net	152.115.75.197
http://aka-cdn-ns.adtech.de	158.36.130.75

IPv6-header

11 IPv6-header

- Flow Label

12 Utvidelsesheadere

- Hop-by-hop Options Header
- Destination Options Header
- Routing Header
- Fragment Header
- Authentication Header
- Encapsulating Security Payload
- Mobility Header

IPv4-header

Version	IHL	Type of Service	Total Length	
Identification			Flags	Fragment Offset
Time To Live		Protocol	Header Checksum	
Source Address				
Destination Address				
Options & Padding				

	Felter som er beholdt i IPv6
	Felter som er fjernet fra IPv6
	Navn og plassering er forskjellig for IPv6
	Nytt felt i IPv6

IPv6-header

Version	Traffic Class	Flow Label	
Payload Length		Next Header	Hop Limit
Source Address			
Destination Address			

\$Ximalas: trunk/IPv4-vs-IPv6.svg 97 2014-09-29 13:04:08Z trond \$

IPv4-header

Version	IHL	Type of Service	Total Length				
Identification			Flags	Fragment Offset			
Time To Live	Protocol	Header Checksum					
Source Address							
Destination Address							
Options & Padding							

	Felter som er beholdt i IPv6
	Felter som er fjernet fra IPv6
	Navn og plassering er forskjellig for IPv6
	Nytt felt i IPv6

IPv6-header

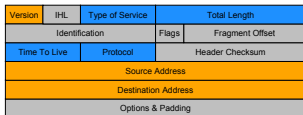
Version	Traffic Class	Flow Label		
Payload Length		Next Header	Hop Limit	
Source Address				
Destination Address				





\$Ximalas: trunk/IPv4-vs-IPv6.svg 97 2014-09-29 13:04:08Z trond \$

- IPv6-headeren er dobbelt så stor som IPv4-headeren (40/20 oktetter)
- IPv6-headeren har færre felter enn IPv4-headeren
- De utelatte feltene er i stor grad flyttet over til egne utvidelsesheadere

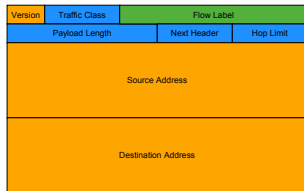
IPv6-header

IPv4-header



-  Felter som er beholdt i IPv6
-  Felter som er fjernet fra IPv6
-  Navn og plassering er forskjellig for IPv6
-  Nytt felt i IPv6

IPv6-header



\$X\$maltes: trunk/IPv4-vs-IPv6.svg 97 2014-09-29 13:04:06Z trondd \$

- Versjonsfeltet (4 bit) settes til 0110
- Traffic Class (8 bit) er det samme som Type of Service i IPv4
- Flow Label (20 bit) er et nytt felt, se neste slide
- Payload Length (16 bit) er det samme som Total Length i IPv4

- Next Header (8 bit) er det samme som Protocol i IPv4
- Hop Limit (8 bit) er det samme som Time To Live i IPv4
- Avsender og mottaker er 128-bit IPv6-adresser
- IPv4-feltene Internet Header Length (IHL), Identification, Flags, Fragment Offset, Header Checksum, Options og Padding, er enten fjernet for godt eller flyttet til egne utvidelsesheadere

- Flow Label-feltet kan brukes av sanntidsapplikasjoner
- Flow Label-verdien angir pakker som tilhører samme sesjon
- Routere bør videresende pakker med samme verdi i Flow Label-feltet fra samme avsender på samme grensesnitt, slik at rekkefølgen bevares
- Verdien 0 (null) brukes for individuelle pakker
- Routere bør videresende pakker med 0 i Flow Label-feltet fra samme avsender på samme grensesnitt, slik at rekkefølgen bevares
- Tilfeldig valgte verdier brukes for pakker som hører sammen
- Flow Label-feltet kan også brukes til å smugle data sammen med legitim trafikk, eller merke slik trafikk, se avsnitt 6.1 i [RFC 6437](#)
- Se [RFC 2460](#), [RFC 3595](#), [RFC 6294](#), [RFC 6436](#) og [RFC 6437](#)

- Utvidelsesheaderne finnes i stort antall:
 - ① Hop-by-hop Options Header
 - ② Destination Options Header
 - ③ Routing Header
 - ④ Fragment Header
 - ⑤ Authentication Header
 - ⑥ Encapsulating Security Payload
 - ⑦ Mobility Header
- Se [RFC 2460](#), [RFC 4302](#), [RFC 4303](#), [RFC 6275](#) og [RFC 7045](#)

Utvidelsesheadere

Hop-by-hop Options Header

```
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Next Header | Hdr Ext Len |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|
.
.
Options
.
.
|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

- Protokollnummer: 0
- Hop-by-hop Options Header må komme før andre Options Headere og før payload
- Alle ledd bør undersøke Hop-by-hop Options Header og dens innhold
- Høyhastighetsroutere vil enten ignorere H-b-H eller la en saktegående routingprosess ta seg av slike pakker

- Valgene Pad1 og PadN er definert i [RFC 2460](#)
- Andre valg: Jumbo Payload ([RFC 2675](#)), RPL Option ([RFC 6553](#)), Tunnel Encapsulation Limit ([RFC 2473](#)), Router Alert ([RFC 2711](#)), Quick-Start ([RFC 4782](#)), CALIPSO ([RFC 5570](#)), SMF_DPD ([RFC 6621](#)), Home Address ([RFC 6275](#)), ILNP nonce ([RFC 6744](#)), Line-Identification Option ([RFC 6788](#)), IP_DFF ([RFC 6971](#))
- Ref.:
<http://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml>

Utvidelsesheadere

Destination Options Header

```
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Next Header | Hdr Ext Len |                                         |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                                                                                   |
.                                                                                   .
.                                     Options                                     .
.                                                                                   .
|                                                                                   |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

- Protokollnummer: 60

Utvidelsesheadere

Routing Header

```
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Next Header | Hdr Ext Len | Routing Type | Segments Left |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|
.
.           type-specific data
.
|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

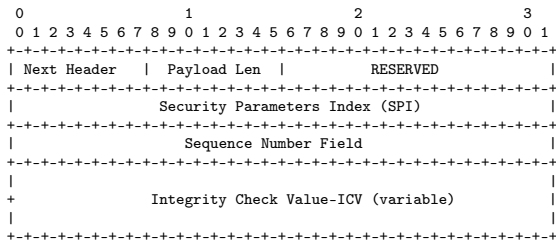
- Protokollnummer: 43

```
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Next Header | Reserved | Fragment Offset | Res|M|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|
| Identification |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

- Protokollnummer: 44

Utvidelsesheadere

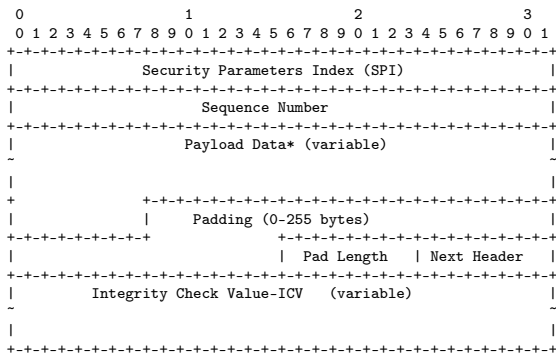
Authentication Header



- Protokollnummer: 51

Utvidelsesheadere

Encapsulating Security Payload



- Protokollnummer: 50

Utvidelsesheadere

Mobility Header

```
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Payload Proto | Header Len  | MH Type   | Reserved  |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                Checksum      |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|
.
.
.
|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

- Protokollnummer: 135

IPv6 over Ethernet

13 IPv6 over Ethernet

14 IPv6 over andre lag-2-typer

- [RFC 2464](#) definerer frameformatet for IPv6-datagrammer over Ethernet
- IPv6-datagrammer fraktes i standard Ethernetformat, [RFC 894](#)
 - Først angis mottakerens MAC-48-adresse
 - Deretter angis avsenders MAC-48-adresse
 - Frametypen settes til 86DD (heksadesimalt)
 - Deretter følger IPv6-header og resten av datagrammet
- Standard MTU for IPv6 over Ethernet er 1500 oktetter
- Minste tillatte MTU for IPv6 er 1280 oktetter
- Er største tilgjengelige MTU mindre enn 1280 oktetter, så må lagene under IPv6 sørge for fragmentering og sammensetting av IPv6-datagrammene ([RFC 2460](#))

Programmet **Wireshark** fremstilte følgende lag-2-informasjon om en utsendt IPv6-pakke:

```
Ethernet II, Src: AsustekC_f2:72:40 (00:26:18:f2:72:40), Dst: Cisco_77:14:57 (00:17:e0:77:14:57)
  Destination: Cisco_77:14:57 (00:17:e0:77:14:57)
    Address: Cisco_77:14:57 (00:17:e0:77:14:57)
      .... ..0. .... = LG bit: Globally unique address (factory default)
      .... ..0. .... = IG bit: Individual address (unicast)
  Source: AsustekC_f2:72:40 (00:26:18:f2:72:40)
    Address: AsustekC_f2:72:40 (00:26:18:f2:72:40)
      .... ..0. .... = LG bit: Globally unique address (factory default)
      .... ..0. .... = IG bit: Individual address (unicast)
Type: IPv6 (0x86dd)
```

- Presentert som heksadesimale oktetter/byter:
- 00 17 E0 77 14 57 00 26 18 F2 72 40 86 DD
 - 00 17 E0 77 14 57 er MAC-48-adressa til mottakeren, routeren
 - 00 26 18 F2 72 40 er MAC-48-adressa til avsenderen, klienten
 - 86 DD angir at et IPv6-datagram følger etter i lag 3

IPv6 over andre lag-2-typer

- FDDI: [RFC 2467](#)
- Token Ring: [RFC 2470](#)
- Non-Broadcast Multiple Access (NBMA) networks: [RFC 2491](#)
- ATM: [RFC 2492](#)
- ARCnet: [RFC 2497](#)
- Frame Relay: [RFC 2590](#)
- IEEE 1394 (FireWire): [RFC 3146](#)
- Low-Power Wireless Personal Area Networks (6LoWPAN): [RFC 4919](#)
- Point-to-point protocol (PPP): [RFC 5072](#)
- Brevduer: [RFC 6214](#), basert på [RFC 1149](#)

Grunnleggende om adresser

Oversikt over del 4: Grunnleggende om adresser I

- 15 Grunnleggende om adresser
- 16 Adressedemo
- 17 MAC-48-adresser
- 18 Modda IEEE EUI-64-format
- 19 Manuell grensesnittidentifikator
- 20 Tilfeldig grensesnittidentifikator
- 21 Spesialadresser
- 22 Duplicate Address Detection — DAD

- 128 bit
- Heksadesimal notasjon
- 16 og 16 bit grupperes og adskilles med kolon
- Ledende nuller kan sløyfes
- To eller flere *sammenhengende* 16-bitblokker med nuller kan slås sammen til :: (dobbelkolon), bare én gang pr. adresse
- Prefikslengde angis ved å sette på en skråstrek og oppgi riktig antall av signifikante bit fra venstre mot høyre i adressa
 - Dette er helt likt CIDR-notasjon for IPv4 ([RFC 4632](#))

Grunnleggende om adresser

Adressedemo

- Uninett:
2001:0700:0000:0000:0000:0000:0000:0000
- FSI:
2001:0700:1100:0000:0000:0000:0000:0000
- IT-avdelingen@FSI:
2001:0700:1100:0003:0000:0000:0000:0000
- Tronds D531 i IT-avdelingen@FSI:
2001:0700:1100:0003:0221:70FF:FE73:686E

Grunnleggende om adresser

Adressedemo: Hierarkisk struktur

- Uninett:
`2001:0700:0000:0000:0000:0000:0000`
- FSI:
`2001:0700:1100:0000:0000:0000:0000`
- IT-avdelingen@FSI:
`2001:0700:1100:0003:0000:0000:0000`
- Tronds D531 i IT-avdelingen@FSI:
`2001:0700:1100:0003:0221:70FF:FE73:686E`

Grunnleggende om adresser

Adressedemo: La oss forenkle adressene

- Uninett:
2001:0700:0000:0000:0000:0000:0000:0000
- FSI:
2001:0700:1100:0000:0000:0000:0000:0000
- IT-avdelingen@FSI:
2001:0700:1100:0003:0000:0000:0000:0000
- Tronds D531 i IT-avdelingen@FSI:
2001:0700:1100:0003:0221:70FF:FE73:686E

Grunnleggende om adresser

Adressedemo: Ledende nuller

- Uninett:
2001:0700:0000:0000:0000:0000:0000:0000
- FSI:
2001:0700:1100:0000:0000:0000:0000:0000
- IT-avdelingen@FSI:
2001:0700:1100:0003:0000:0000:0000:0000
- Tronds D531 i IT-avdelingen@FSI:
2001:0700:1100:0003:0221:70FF:FE73:686E

Grunnleggende om adresser

Adressedemo: Fjernet ledende nuller

- Uninett:
2001:700:0:0:0:0:0:0
- FSI:
2001:700:1100:0:0:0:0:0
- IT-avdelingen@FSI:
2001:700:1100:3:0:0:0:0
- Tronds D531 i IT-avdelingen@FSI:
2001:700:1100:3:221:70FF:FE73:686E

Grunnleggende om adresser

Adressedemo: La oss forenkle litt til

- Uninett:
2001:700:0:0:0:0:0:0
- FSI:
2001:700:1100:0:0:0:0:0
- IT-avdelingen@FSI:
2001:700:1100:3:0:0:0:0
- Tronds D531 i IT-avdelingen@FSI:
2001:700:1100:3:221:70FF:FE73:686E

Grunnleggende om adresser

Adressedemo: To eller flere sammenhengende 16-bitblokker med bare 0

- Uninett:
2001:700:0:0:0:0:0:0
- FSI:
2001:700:1100:0:0:0:0:0
- IT-avdelingen@FSI:
2001:700:1100:3:0:0:0:0
- Tronds D531 i IT-avdelingen@FSI:
2001:700:1100:3:221:70FF:FE73:686E

Grunnleggende om adresser

Adressedemo: Erstattet med dobbelkolon

- Uninett:
2001:700::
- FSI:
2001:700:1100::
- IT-avdelingen@FSI:
2001:700:1100:3::
- Tronds D531 i IT-avdelingen@FSI:
2001:700:1100:3:221:70FF:FE73:686E

Grunnleggende om adresser

Adressedemo: Kompakt form

- Uninett:
2001:700::
- FSI:
2001:700:1100::
- IT-avdelingen@FSI:
2001:700:1100:3::
- Tronds D531 i IT-avdelingen@FSI:
2001:700:1100:3:221:70FF:FE73:686E

Grunnleggende om adresser

Adressedemo: Vis prefikslengde

- Uninett:
2001:700::/32
- FSI:
2001:700:1100::/48
- IT-avdelingen@FSI:
2001:700:1100:3::/64
- Tronds D531 i IT-avdelingen@FSI:
2001:700:1100:3:221:70FF:FE73:686E/128

Grunnleggende om adresser

Adressedemo: Kompakte adresser med prefikslengde

- Uninett:
2001:700::/32
- FSI:
2001:700:1100::/48
- IT-avdelingen@FSI:
2001:700:1100:3::/64
- Tronds D531 i IT-avdelingen@FSI:
2001:700:1100:3:221:70FF:FE73:686E/128

- MAC-48-adresser har følgende oppbygging, gitt av [IEEE 802-2001](#):
 - CC:cc:cc:nn:nn:nn (heksadesimalt)
 - Den første halvparten er produsentnummer: CC:cc:cc
 - Den andre halvparten er løpenummer: nn:nn:nn
- Den første oktetten i produsentnummeret, CC, har en spesiell oppbygging: (binært)
 - CCCCCug (heksadesimalt)
 - Når u-bitet er satt til 0 (null), så gjelder formatet som er oppgitt her, altså CC:cc:cc:nn:nn:nn (heksadesimalt)
 - Når u-bitet er satt til 1, så er alle C- og c-sifrene løpenummer, mens u- og g-bitene beholder sine spesielle betydninger
 - Når g-bitet er 0 så angir adressa en individuell node, og når g-bitet er 1 så er adressa en multicastgruppe

Grunnleggende om adresser

MAC-48-adresser

- Gitt denne MAC-48-adressa: 00:21:70:73:68:6E
- CC-oktetten har verdien 00 (heksadesimalt)
- På binær form er dette 00000000 (CCCCCug)
- Vi ser at både u- og g-bitene er satt til 0
- Dette er en MAC-48-adresse som:
 - følger det vanlige mønsteret med produsent- og løpenummer
 - angir en individuell node
 - er produsert av «Dell Inc» ifølge OUI-lista hos IEEE (søk i fila etter 00-21-70)

- Unicast-adresser består av 2 ting:
 - 1 Prefiks
 - 2 Grensesnittidentifikator
- Bestemt av [RFC 4941](#)
- Grensesnittidentifikatorer er alltid på 64 bit
 - Dette gjelder ikke for adresser som starter på 000 (binært)
- Grensesnittidentifikatorer kan lages automatisk fra MAC-48-adresser
- Grensesnittidentifikatorer kan også angis manuelt eller velges tilfeldig
- Angis grensesnittidentifikatoren manuelt, så angis som regel en fullstendig IPv6-adresse
- Grensesnittidentifikatorer følger [IEEE EUI-64](#)-formatet med to unntak:
 - 1 Universal/local-bitet brukes med *invertert* betydning/verdi
 - Gruppebitet mister sin vanlige betydning i forbindelse med grensesnittidentifikatorer
 - 2 Oktettene på midten skal være FF:FE ved automatisk konvertering fra MAC-48 til EUI-64

- Grensesnittidentifikatorer lages fra MAC-48-adresser etter oppskriften i [RFC 4291](#):
 - Gitt denne MAC-48-adressa: 00:21:70:73:68:6E
 - Invertér universal/local-bitet: 02:21:70:73:68:6E
 - Før: 00 (heksadesimalt) = 00000000 (binært)
 - Etter: 00000010 (binært) = 02 (heksadesimalt)
 - Sett inn FF:FE på midten: 02:21:70:FF:FE:73:68:6E
 - Ta bort overflødig kolon og nuller: 221:70FF:FE73:686E
 - Høyreskift hele stasen: ::221:70FF:FE73:686E
 - Nå er grensesnittidentifikatoren klar til å bli kombinert med ønsket prefiks
 - Prefiks annonsert av router: 2001:700:1100:3::/64
 - Fullstendig adresse: 2001:700:1100:3:221:70FF:FE73:686E

- OBS! Arbeidsuhell!
- Det skulle egentlig ha vært FF:FF i stedet for FF:FE
 - MAC-48 → EUI-64 skal bruke FF:FF
 - EUI-48 → EUI-64 skal bruke FF:FE
- Se <http://standards.ieee.org/develop/regauth/tut/eui.pdf>
- Fordi IPv6 bruker universal/local-bitet med invertert betydning/verdi, så er arbeidsuhellet akseptert
- Se RFC 4291
- IEEE 802.15 WPAN, IEEE 1394 FireWire, og ZigBee bruker EUI-64-adresser i lag 2

Grunnleggende om adresser

Manuell grensesnittidentifikator

- Manuell grensesnittidentifikator innebærer at universal/local-bitet som regel er satt til 0
- De øvrige 63 bitene kan være hva som helst, bare verdien ikke skaper adressekollisjon i samme VLAN
- Normalt bruker man manuelle grensesnittidentifikatorer satt til lave verdier
- For eksempel ::53 (DNS-tjener, kanskje)
- Samme eksempel, men med et vilkårlig prefiks: 2001:db8:1234:8::53

Grunnleggende om adresser

Manuell grensesnittidentifikator

- Lav verdi for grensesnittidentifikatorer gjør at universal/local-bitet blir satt til null:
 - ::53 (heksadesimalt)
 - ::0:0:0:53 (heksadesimalt)
 - ::0000000000000000:00 ... 00:0000000001010011 (binært)
 - Veldig praktisk for lokalgitte adresser, ikke sant?
- *Uten* invertering av universal/local-bitet, måtte vi bruke manuelle grensesnittidentifikatorer på denne måten:
 - ::0200:0:0:53 (heksadesimalt)
 - ::0000000100000000:00 ... 00:0000000001010011 (binært)
 - Tungvint og upraktisk, ikke sant?
- Se her:
 - 2001:db8:1234:1:0200:0:0:53 vs
 - 2001:db8:1234:1::53
 - Ja til den siste, nei til den forrige

Grunnleggende om adresser

Manuell grensesnittidentifikator

- Det er ingenting i veien for å «kode» IPv4-adressa inn i IPv6-adressa:
- 2001:700:1100:3:**128:39:174:67** (excelsior.fig.ol.no)
- Man må bare passe på verdien til universal/local-bitet
- $128 = 0 \ 1 \ 2 \ 8 = 0000 \ 00**0**1 \ 0010 \ 1000$ (heks, heks, bin)
- u-bitet er 0, altså en lokalgitt adresse
- Dette gikk bra!

- Verdiene

- 0 = 0000,
- 1 = 0001,
- 4 = 0100,
- 5 = 0101,
- 8 = 1000,
- 9 = 1001,
- C = 1100, og
- D = 1101,

medfører 0 i u-bitet

Grunnleggende om adresser

Tilfeldig grensesnittidentifikator

- Konstant grensesnittidentifikator truer personvernet
- Eksempel med Tronds D531-læppis:
 - 2001:700:1100:3:221:70FF:FE73:686E (IT-avdelingen@FSI)
 - 2001:700:1D00:8:221:70FF:FE73:686E (public-nettet@HiG)
- RFC 4941 beskriver bruk av tilfeldig grensesnittidentifikator
- Med tilfeldig grensesnittidentifikator:
 - 2001:700:1100:3:B9D9:B729:6CDD:4E5 (IT-avdelingen@FSI)
 - 2001:700:1D00:8:B9D9:B729:6CDD:4E5 (public-nettet@HiG)
- Disse byttes ut typisk hver dag:
 - 2001:700:1100:3:F503:1E6F:5F2F:F5F2 (IT-avdelingen@FSI)
 - 2001:700:1D00:8:F503:1E6F:5F2F:F5F2 (public-nettet@HiG)
- Man må bare passe på u/l-bitet og passe seg for adressekollisjon

- [RFC 4941](#) angir en metode for generering av tilfeldig grensesnittidentifikator:
 - 1 Sett sammen historisk verdi fra forrige runde (eller et tilfeldig 64-bit heltall) med den konstante grensesnittidentifikatoren til et 128-bit heltall
 - 2 Beregn MD5-hash av resultatet fra trinn 1
 - 3 Bruk de 64 *mest* signifikante bitene og sett det sjuende mest signifikante bitet til null (dette indikerer en lokalgitt grensesnittidentifikator)
 - 4 Sammenlign den nye tilfeldige grensesnittidentifikatoren med lista over reserverte identifikatorer; oppdages en uakseptabel identifikator, gå til trinn 1 og bruk de 64 *minst* signifikante bitene fra trinn 2 som historisk verdi
 - 5 Ta i bruk den nye tilfeldige grensesnittidentifikatoren
 - 6 Lagre de 64 *minst* signifikante bitene fra trinn 2 som historisk verdi for bruk den neste gangen denne algoritmen brukes

- Nulladressa:

- 0:0:0:0:0:0:0:0/128 eller ::/128
 - Brukes av klienter som ennå ikke vet sin egen adresse (DHCPv6)
 - Brukes av tjenester som godtar forespørsler fra alle grensesnitt (sjekk ut [bind\(2\)](#)-systemkallet i «Juniks»)
- 0:0:0:0:0:0:0:0/0 eller ::/0
 - Brukes for å angi default route
- Tilsvarende 0.0.0.0/32 og 0/32, og 0.0.0.0/0 og 0/0 i IPv4

- Loopbackadressa: 0:0:0:0:0:0:0:1/128 eller ::1/128
 - Velkjent adresse for å snakke med tjenester i samme node
 - Tilsvarende 127.0.0.1/32 i IPv4

- Dokumentasjonsprefiks: 2001:db8::/32
 - Brukes for beskrivelse av IPv6-oppsett i lærebøker og annen generell dokumentasjon ([RFC 3849](#))
 - Forbudt å bruke på det offentlige internettet
 - Bør blokkeres i *inngående* og *utgående* ACL-er for internettgrensesnittet til routere

- IPv4-mapped IPv6 addresses: `::FFFF:w.x.y.z`
 - Hvor *w.x.y.z* er den opprinnelige IPv4-adressa skrevet på vanlige måte for IPv4-adresser
 - Eksempel: `::FFFF:128.39.174.1`
 - Brukes i systemer som har både IPv4- og IPv6-adresser, men hvor den enkelte tjeneste bare bruker IPv6-socketer og har slått av `IPV6_V6ONLY` med `setsockopt(2)` for lyttesocketen
 - Forbudt av sikkerhetshensyn i enkelte OS-er som [OpenBSD](#), se OpenBSDs [ip6\(4\)](#)
 - Tjenestene må da åpne separate lyttesocketer for IPv4 og IPv6
- [RFC 6890](#) inneholder en oversikt over alle spesialadresser for både IPv4 og IPv6

- Når en unicast-adresse er generert skal man alltid sjekke at ingen andre bruker den samme adressa (RFC 4862)
- Dette gjøres ved å sende en «ICMPv6 Neighbor Solicitation-melding» til den genererte adressas «Solicited-node multicast address»
- ICMPv6-meldinga inneholder den genererte adressa i feltet for «Target Address» (RFC 4861)
- En «Solicited-node multicast address» er på formen FF02::1:FF $aa:bbcc$, hvor $aabbcc$ er de 24 minst signifikante bitene fra den opprinnelige adressa (RFC 4291)
- Sett at den genererte adressa er 2001:700:1100:3:221:70FF:FE73:686E
- «Solicited-node multicast address» vil da være FF02::1:FF73:686E
- Vanligvis kommer det ikke noe svar på slike ICMPv6-meldinger ...

- ... trodde vi ...
- «[Danger, Will Robinson!](#)»
- Det er et stort potensiale for Denial of Service — DoS (RFC 3756)
- En «slabbedask» kan velge å svare på DAD og nekte oss å bruke *enhver* adresse
- Svaret kommer i form av en «ICMPv6 Neighbor Advertisement»-melding som forteller oss at en annen node bruker den samme adressa (RFC 4862)
- Resultat: «slabbedasken» kan bruke nettverket uforstyrret
- Dersom det er 2 eller flere «slabbedasker» i samme nettverk, hva da?
- Problemet kan løses med «SEcure Neighbor Discovery» (SEND), [RFC 3971](#)

Adresstyper

Oversikt over del 5: Adresstyper

- 23 Adresstyper
- 24 Link-local-adresser
- 25 Site-local-adresser
- 26 Offentlige unicast-adresser
- 27 Unike, lokale, aggregerbare adresser
- 28 Anycast-adresser
- 29 Multicast-adresser

- Det finnes flere adresstyper med forskjellige bruksområder:
 - Unicast-adresser:
 - Link-local-adresser
 - Site-local-adresser
 - Offentlige unicast-adresser
 - Unike, lokale, aggregerbare adresser
 - Anycast-adresser
 - Multicast-adresser
- Merk at broadcast er avskaffa og er i stor grad erstatta med link-local-multicast

- Definert: [RFC 4291](#)
- Bruksområde:
 - Lokal kommunikasjon internt i VLAN-et
 - Sentral for autokonfigurasjon (av unicastadresser)
 - Blir ikke videresendt av routere til andre VLAN eller til internett
 - Kan brukes i ad-hoc-nett
- Prefiks: FE80::/10
- De neste 54 bitene skal settes til null
- De siste 64 bitene er grensesnittidentifikator i modda EUI-64-format
- Eksempel: FE80::221:70FF:FE73:686E

- Definert: [RFC 3513](#)
- Bruksområde: private adresser på lik linje med [RFC 1918](#)
- Prefiks: FEC0::/10
- De neste 54 bitene brukes til subnet-ID
- De siste 64 bitene er grensesnittidentifikator i modda EUI-64-format
- Eksempel: FEC0::DEAD:BEEF:1337
- Ikke bruk site-local-adresser ([RFC 3879](#))
- Site-local-adresser er erstatta med ULA ([RFC 4193](#))

- Definert: [RFC 4291](#) og [RFC 3587](#)
- Bruksområde: ende-til-ende-kommunikasjon på det offentlige internett
- Prefiks: 2000::/3
- De neste bitene allokeres hierarkisk, minimum i 4-bitblokker, men gjerne i 8- eller 16-bitblokker
- De siste 64 bitene er grensesnittidentifikator i modda EUI-64-format
- Det er vanlig at kundene blir tildelt /48-, /56- eller /62-bits prefiks av ISP-ene:
 - /48-bits prefiks gir $128 - 64 - 48 = 16$ subnetbit $\rightarrow 2^{16} = 65536$ subnett
 - /56-bits prefiks gir $128 - 64 - 56 = 8$ subnetbit $\rightarrow 2^8 = 256$ subnett
 - /62-bits prefiks gir $128 - 64 - 62 = 2$ subnetbit $\rightarrow 2^2 = 4$ subnett
- Eksempel: 2001:700:1100:1::1/128

- Definert: [RFC 4193](#)
- Bruksområde: ende-til-ende-kommunikasjon internt i nettverket
- Veldig praktisk å ha faste, interne adresser uavhengig av offentlig prefiks tildelt av ISP
- Prefiks: `FC00::/7`
- Det åttende mest signifikante bitet skal settes til 1 inntil videre
- Det reelle prefikset er dermed `FD00::/8`
- Prefikset `FC00::/8` er reservert inntil videre

Adressetyper

Unike, lokale, aggregerbare adresser

- Reelt prefiks: FD00::/8
- De neste 40 bitene genereres tilfeldig, gjerne som beskrevet i [RFC 4193](#)
- De neste 16 bitene brukes til subnett-ID
- De siste 64 bitene er grensesnittidentifikator i modda EUI-64-format
- Eksempel: FD5C:14CF:C300:31::1/128

Adresstyper

Unike, lokale, aggregerbare adresser

- SixXS tilbyr bl.a.:
 - Generering av ULA-prefiks: <http://www.sixxs.net/tools/grh/ula/>
 - Registrering av ULA-prefiks: <http://www.sixxs.net/tools/grh/ula/list/>
- George Michaelson, seniorforsker ved APNIC, har oppdaget ULA-adresser i fri dressur ute på internett:
 - Tydeligvis klarer ikke folk å lese RFC-ene og holde seg til de fastsatte reglene
 - http://www.sixxs.net/archive/docs/IEPG2013_ULA_in_the_wild.pdf

- Her er algoritmen fra [RFC 4193](#) for å generere de 40 tilfeldige bitene:
 - ① Uttrykk nåværende øyeblikk som et 64-bit heltall i NTP-format ([RFC 5905](#))
 - ② Bruk en EUI-64-identifikator fra systemet som kjører denne algoritmen
 - Mangler du en EUI-64-identifikator, så kan du lage en fra en 48-bit MAC-adresse som angitt i [RFC 4291](#)
 - Kan du ikke lage en EUI-64-identifikator, så bruk en annen unik verdi som serienummeret til systemet
 - ③ Sett sammen de to 64-bit heltallene til et 128-bit heltall
 - ④ Beregn en SHA-1-hash som beskrevet i [RFC 3174](#). Resultatet er et heltall på 160 bit
 - ⑤ Bruk de 40 minst signifikante bitene som global identifikator
- Har man tilgang på tilfeldige tall av god kvalitet, så kan man bruke de i stedet for metoden over

- Definert: [RFC 4291](#)
- Bruksområde: felles adresse for distribuerte tjenester, routerne bestemmer hvilken server som er nærmest og sender trafikken dit
- Prefiks: ingen, allokeres fra dine egne unicast-adresser og markeres som en anycast-adresse hos routerne og serverne
- Alle IPv6-adresser hvor alle bit i grensesnittidentifikatoren satt til null, er reservert som «Subnet-Router anycast address»
- Denne anycast-adressa brukes når man vil kontakte én av potensielt flere routere i subnettet der du er
- Eksempel: 2001:700:1100:1::/128 **anycast**
- Se også [RFC 2526](#)

- Definert: [RFC 4291](#)
- Bruksområde: én-til-mange-kommunikasjon
- Prefiks: $FF::/8$
- Flagg f og rekkevidde r er innebygget i adressa: $FFfr::/16$
- Eksempel: $FF0E::101/128$ (global multicast-adresse for NTP)

- Flaggene heter ORPT (null, err, pe, te)
- Flagget T angir med 0 at adressa er velkjent (definert av [IANA](#)), og med 1 at adressa er midlertidig (lokalt definert)
- Flagget P angir med 1 at adressa inneholder et unicast-prefiks og skal følge reglene i [RFC 3306](#)
- Flagget R angir med 1 at adressa også inneholder et møtepunkt («rendezvous point») og skal følge reglene i [RFC 3956](#)
- Flaggene P og R gjør det enkelt å lage egne multicast-adresser for internt bruk i organisasjonen
- Bruk av flaggene R, P og T gjennomgås i detalj i del 10

- Følgende rekkevidder er definert i [RFC 4921](#):
 - 0: reservert
 - 1: interface-local
 - 2: link-local
 - 3: reservert
 - 4: admin-local
 - 5: site-local
 - 6: ikke definert
 - 7: ikke definert
 - 8: organization-local
 - 9: ikke definert
 - A: ikke definert, brukt av Uninett til å [begrense](#) trafikken innenfor «Uninettet»
 - B: ikke definert
 - C: ikke definert
 - D: ikke definert
 - E: global
 - F: reservert

- Noen kjente IPv6-multicastadresser:
 - FF02::1 All nodes on the local network segment
 - FF02::2 All routers on the local network segment
 - FF02::5 OSPFv3 All SPF routers
 - FF02::6 OSPFv3 All DR routers
 - FF02::8 IS-IS for IPv6 routers
 - FF02::9 RIP routers
 - FF02::A EIGRP routers
 - FF02::D PIM routers
 - FF02::16 MLDv2 reports
 - FF02::1:2 All DHCP servers and relay agents on the local network segment
 - FF02::1:3 All LLMNR hosts on the local network segment
 - FF05::1:3 All DHCP servers on the local network site
 - FF0x::C Simple Service Discovery Protocol
 - FF0x::FB Multicast DNS
 - FF0x::101 Network Time Protocol
 - FF0x::108 Network Information Service
 - FF0x::114 Used for experiments

- Kobling av multicast-adresser til lag-2-adresser:
 - Eksempel:
 - IPv6: FF02::1 = FF02::0000:0001
 - MAC-48: 33:33:00:00:00:01
 - De 32 minst signifikante bitene kopieres fra IPv6-adressa og til MAC-48-adressa
 - Dette gir en viss overlapp for de multicast-adresser som tilfeldigvis slutter på de samme 32 bitene
 - Det går ganske bra i praksis
 - Se [RFC 2464](#) og [RFC 6085](#)

DNS

30 AAAA og PTR

31 A6

- Navn-til-IPv6-adresser bruker AAAA-poster
 - Eksempel:
`$ORIGIN fig.ol.no.`
`svabu IN AAAA 2001:700:1100:1::4`
- IPv6-adresser-til-navn bruker PTR-poster plassert i ip6.arpa.
 - Eksempel:
`$ORIGIN 1.0.0.0.0.0.1.1.0.0.7.0.1.0.0.2.ip6.arpa.`
`4.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN PTR svabu.fig.ol.no.`
- Se [RFC 3596](#)

- A6-poster var foreslått som erstatning for AAAA-poster av [RFC 2874](#), men er endret til eksperimentell av [RFC 3363](#)
- [RFC 3364](#) diskuterer fordeler og ulemper med AAAA og A6
- En A6-post består av 2–3 ting:
 - 1 Prefikslengde fra og med 0 til og med 128
 - 2 Utdrag av IPv6-adressa
 - 3 Navn som henviser til resten av adressa
- Settes prefikslengda til:
 - 0, så er det **ikke** lov å oppgi noen henvisning, fordi dette navnet er det øverste eller det eneste nivået i en kjede
 - 128, så er det **ikke** lov å oppgi noen IPv6-adresse, fordi man henviser til et helt annet navn, tydeligvis et overflødig alternativ til CNAME

- Avsnittene 3.1.1 og 3.1.3 i [RFC 2874](#) er ikke enige med hverandre når prefikslengda settes til 128
 - Avsnitt 3.1.1:
The address suffix component SHALL NOT be present if the prefix length is 128.
 - Avsnitt 3.1.3:
The IPv6 address MAY be be[sic] absent if the prefix length is 128.
- Med andre ord, avsnitt 3.1.1 forbyr IPv6-adresse når prefikslengda er 128, mens avsnitt 3.1.3 sier at IPv6-adresse *kan utelates* i det samme tilfellet
- Er det noe rart at noen av oss kan bli forvirra?
- Vil du leke med A6 i et lukket miljø, så sjekk ut BIND 9.2.x

- Et tenkt eksempel med A6:
- \$ORIGIN ip6.uninett.no.
uninett IN A6 0 2001:700::
fig IN A6 32 0:0:1100:: uninett

\$ORIGIN fig.ol.no.
ext-servere.ip6 IN A6 48 0:0:0:1:: fig.ip6.uninett.no.
svabu IN A6 64 ::4 ext-servere.ip6
- Vi vil vite IPv6-adressa for svabu.fig.ol.no. og vi vil bruke A6-poster for å finne svaret

- Et tenkt eksempel med A6:
- \$ORIGIN fig.ol.no.
svabu IN A6 64 ::4 ext-servere.ip6
- Forklaring:
 - svabu.fig.ol.no. oppgir ::4, mangler de 64 mest signifikante bitene og henviser til ext-servere.ip6.fig.ol.no.

- Et tenkt eksempel med A6:
- \$ORIGIN fig.ol.no.
svabu IN A6 64 ::4 ext-servere.ip6
ext-servere.ip6 IN A6 48 0:0:0:1:: fig.ip6.uninett.no.
- Forklaring:
 - ext-servere.ip6.fig.ol.no. oppgir 0:0:0:1::, mangler de 48 mest signifikante bitene og henviser til fig.ip6.uninett.no.

- Et tenkt eksempel med A6:

- \$ORIGIN fig.ol.no.

```
svabu          IN A6 64          ::4 ext-servere.ip6  
ext-servere.ip6 IN A6 48 0:0:0:1:: fig.ip6.uninett.no.
```

```
$ORIGIN ip6.uninett.no.
```

```
fig           IN A6 32 0:0:1100:: uninett
```

- Forklaring:
 - fig.ip6.uninett.no. oppgir 0:0:1100::, mangler de 32 mest signifikante bitene og henviser til uninett.ip6.uninett.no.

- Et tenkt eksempel med A6:

- \$ORIGIN fig.ol.no.

```
svabu          IN A6 64          ::4 ext-servere.ip6  
ext-servere.ip6 IN A6 48 0:0:0:1:: fig.ip6.uninett.no.
```

```
$ORIGIN ip6.uninett.no.
```

```
fig          IN A6 32 0:0:1100:: uninett  
uninett IN A6  0 2001:700::
```

- Forklaring:

- Kjeden slutter med uninett.ip6.uninett.no. og her angis de 32 mest signifikante bitene, 2001:700::

- Et tenkt eksempel med A6:

- \$ORIGIN fig.ol.no.

```
svabu          IN A6 64          ::4 ext-servere.ip6
ext-servere.ip6 IN A6 48 0:0:0:1:: fig.ip6.uninett.no.
```

```
$ORIGIN ip6.uninett.no.
```

```
fig      IN A6 32 0:0:1100:: uninett
uninett IN A6  0 2001:700::
```

- Vi har påvist følgende adressekjede:

- 0000:0000:0000:0000::**4**
- 0000:0000:0000:000**1**::
- 0000:0000:**1100**:0000::
- **2001**:0**700**:0000:0000::

```
svabu.fig.ol.no.
ext-servere.ip6.fig.ol.no.
fig.ip6.uninett.no.
uninett.ip6.uninett.no.
```

- Bitvis-OR gir den fullstendige adressa 2001:700:1100:1::4

Del VII

ICMPv6

Oversikt over del 7: ICMPv6 I

- 32 ICMPv6
- 33 Multicast Listener Discovery
- 34 Neighbor Discovery
- 35 Router Renumbering
- 36 Node Information
- 37 Inverse Neighbor Discovery
- 38 Version 2 Multicast Listener Report
- 39 Mobile IPv6
- 40 SEcure Neighbor Discovery (SEND)
- 41 Experimental Mobility Type
- 42 Multicast Router Discovery
- 43 FMIPv6
- 44 RPL Control Message
- 45 ILNPv6 Locator Update Message
- 46 Duplicate Address

- Fra [RFC 4443](#)
- Feilmeldinger:
 - 1: Destination Unreachable
 - 2: Packet Too Big
 - 3: Time Exceeded
 - 4: Parameter Problem
 - 100: Private eksperimenter
 - 101: Private eksperimenter
 - 127: Reservert for utvidelse av feilmeldingene
- Informative meldinger:
 - 128: Echo request (ping)
 - 129: Echo reply (pong)
 - 200: Private eksperimenter
 - 201: Private eksperimenter
 - 255: Reservert for utvidelse av informative meldinger

- Definert: [RFC 2710](#)
- Angir tre nye ICMPv6-meldinger:
 - 130: Multicast Listener Query
 - 131: Multicast Listener Report
 - 132: Multicast Listener Done
- Brukes for å fortelle routere hvilke multicastadresser man vil motta trafikk for

- Definert: [RFC 4861](#)
- Angir fem nye ICMPv6-meldinger:
 - 133: Router Solicitation
 - 134: Router Advertisement
 - 135: Neighbor Solicitation
 - 136: Neighbor Advertisement
 - 137: Redirect
- Sentral ved autokonfigurering av adresser
- Brukes for å bekrefte at nodene er oppegående og bestemme lag-2-adressene til mottakere
- Neighbor Discovery gjennomgås i detalj i del 8

- Definert: [RFC 2894](#)
- Angir én ny ICMPv6-melding:
 - 138: Router Renumbering
- [RFC 2894](#) angir følgende underkoder:
 - 0: Router Renumbering Command
 - 1: Router Renumbering Result
 - 255: Sequence Number Reset

- Definert: [RFC 4620](#)
- Angir to nye ICMPv6-meldinger:
 - 139: Node Information Query
 - 140: Node Information Reply
- [RFC 4620](#) angir følgende underkoder for type 139:
 - 0: Datafeltet inneholder en IPv6-adresse
 - 1: Datafeltet inneholder et navn
 - 2: Datafeltet inneholder en IPv4-adresse
- [RFC 4620](#) angir følgende underkoder for type 140:
 - 0: Vellykket svar
 - 1: Svaret vil ikke bli avslørt
 - 2: Underkoden i forespørselen er ukjent

- Definert: [RFC 3122](#)
- Angir to nye ICMPv6-meldinger:
 - 141: Inverse Neighbor Discovery Solicitation
 - 142: Inverse Neighbor Discovery Advertisement
- Gjør det mulig for én node å lære IPv6-adressen(e) til en annen node i samme VLAN, når man bare vet lag-2-adressa til den andre noden

- Definert: [RFC 3810](#)
- Angir én ny ICMPv6-melding:
 - 143: Version 2 Multicast Listener Report
- Utvider MLDv1 ([RFC 2710](#)) med slik at bare bestemte avsendere er interessante (Source-Specific Multicast, [RFC 3569](#))

- Definert: [RFC 6275](#)
- Angir fire nye ICMPv6-meldinger:
 - 144: Home Agent Address Discovery Request
 - 145: Home Agent Address Discovery Reply
 - 146: Mobile Prefix Solicitation
 - 147: Mobile Prefix Advertisement
- Brukes for å tilrettelegge for digitale nomader

- Definert: [RFC 3971](#)
- Angir to nye ICMPv6-meldinger:
 - 148: Certification Path Solicitation
 - 149: Certification Path Advertisement
- Med SEND unngås DoS-problemene til Neighbor Discovery
- Routerne deler ut kryptografisk genererte adresser [RFC 3972](#)
- Dette krever sertifikatstruktur (RPKI, [RFC 6494](#)) i routere og i klienter
- Ikke implementert i Cisco IOS 12.2(55)SE for Catalyst 3560G
- Ikke spesielt aktuelt for FSI, for annet enn ansattnett, på grunn av den administrative byrden

- Definert: [RFC 4065](#)
- Angir én ny ICMPv6-melding:
 - 150: Experimental Mobility Type
- «The Seamoby Candidate Access Router Discovery (CARD) protocol [[RFC 4066](#)] and the Context Transfer Protocol (CXTCP) [[RFC 4067](#)] are experimental protocols designed to accelerate IP handover between wireless access routers»

- Definert: [RFC 4286](#)
- Angir tre nye ICMPv6-meldinger:
 - 151: Multicast Router Advertisement
 - 152: Multicast Router Solicitation
 - 153: Multicast Router Termination
- Catalyst 3560G har ikke støtte for annet enn IPv4-multicast
- Ved FSI har vi ikke fått testet IPv6-multicast

- Definert: [RFC 5568](#)
- Angir én ny ICMPv6-melding:
 - 154: FMIPv6, Fast handovers, Mobile IPv6

- Definert: [RFC 6550](#)
- Angir én ny ICMPv6-melding:
 - 155: RPL Control Message
- IPv6 Routing Protocol for Low-Power and Lossy Networks

- Definert: [RFC 6743](#)
- Angir én ny ICMPv6-melding:
 - 156: ILNPv6 Locator Update Message
- Identifier-Locator Network Protocol
- En eksperimentell måte å håndtere digitale nomader

- Definert: [RFC 6775](#)
- Angir to nye ICMPv6-meldinger:
 - 157: Duplicate Address Request
 - 158: Duplicate Address Confirmation
- Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)

Neighbor Discovery

Oversikt over del 8: Neighbor Discovery I

- 47 Router Solicitation
- 48 Router Advertisement
- 49 Neighbor Solicitation
- 50 Neighbor Advertisement
- 51 Redirect

- Definert: [RFC 4861](#)
- Angir fem nye ICMPv6-meldinger:
 - 133: Router Solicitation
 - 134: Router Advertisement
 - 135: Neighbor Solicitation
 - 136: Neighbor Advertisement
 - 137: Redirect
- Sentral ved autokonfigurering av adresser
- Brukes for å bekrefte at nodene er oppegående og bestemme lag-2-adressene til mottakere

Neighbor Discovery

Router Solicitation

Internet Control Message Protocol v6

Type: Router Solicitation (133)

Code: 0

Checksum: 0xc065 [correct]

Reserved: 00000000

ICMPv6 Option (Source link-layer address : 00:21:70:73:68:6e)

Type: Source link-layer address (1)

Length: 1 (8 bytes)

Link-layer address: Dell_73:68:6e (00:21:70:73:68:6e)

- Avsenders IPv6-adresse er enten `::/0` eller en av utgående grensesnitts IPv6-adresser
- Mottakers IPv6-adresse er vanligvis `FF02::2`
- «Hop Limit» i IPv6-headeren skal settes til 255
- Det er god sedvane å angi sin egen lag-2-adresse i ICMPv6-meldinga

Neighbor Discovery

Router Advertisement

```
Internet Control Message Protocol v6
  Type: Router Advertisement (134)
  Code: 0
  Checksum: 0xfa8c [correct]
  Cur hop limit: 64
  Flags: 0x48
    0... .... = Managed address configuration: Not set
    .1... .... = Other configuration: Set
    ..0. .... = Home Agent: Not set
    ...0 1... = Prf (Default Router Preference): High (1)
    ....0... = Proxy: Not set
    ....0.0. = Reserved: 0
  Router lifetime (s): 1800
  Reachable time (ms): 0
  Retrans timer (ms): 0
  ICMPv6 Option (Source link-layer address : 00:17:e0:77:14:57)
    Type: Source link-layer address (1)
    Length: 1 (8 bytes)
    Link-layer address: Cisco_77:14:57 (00:17:e0:77:14:57)
  ICMPv6 Option (MTU : 1500)
    Type: MTU (5)
    Length: 1 (8 bytes)
    Reserved
    MTU: 1500
```

- Avsenders IPv6-adresse må være routerens link-local-adresse for utgående grensesnitt
- Mottakers IPv6-adresse er enten adressa til den noden som sendte «Router Solicitation» eller til FF02::1 for generell annonsering
- «Hop Limit» i IPv6-headeren skal settes til 255

Neighbor Discovery

Router Advertisement

```
Internet Control Message Protocol v6
  Type: Router Advertisement (134)
  Code: 0
  Checksum: 0xfa8c [correct]
  Cur hop limit: 64
  Flags: 0x48
    0... .... = Managed address configuration: Not set
    .1... .... = Other configuration: Set
    ..0. .... = Home Agent: Not set
    ...0 1... = Prf (Default Router Preference): High (1)
    ....0... = Proxy: Not set
    .... ..0. = Reserved: 0
  Router lifetime (s): 1800
  Reachable time (ms): 0
  Retrans timer (ms): 0
  ICMPv6 Option (Source link-layer address : 00:17:e0:77:14:57)
    Type: Source link-layer address (1)
    Length: 1 (8 bytes)
    Link-layer address: Cisco_77:14:57 (00:17:e0:77:14:57)
  ICMPv6 Option (MTU : 1500)
    Type: MTU (5)
    Length: 1 (8 bytes)
    Reserved
    MTU: 1500
```

- Routeren er snill og oppgir:
 - Autokonfigurasjon av adresser skal utføres
 - Andre opplysninger er tilgjengelig med DHCPv6
 - Dette er ingen «Home Agent»
 - Routerens preferansenivå er «High»
 - Annonseringens levetid er 1800 s = 30 min
 - Routerens lag-2-adresse
 - Linkens MTU-verdi

Neighbor Discovery

Router Advertisement

```
ICMPv6 Option (Prefix information : 2001:700:1100:3::/64)
  Type: Prefix information (3)
  Length: 4 (32 bytes)
  Prefix Length: 64
  Flag: 0xc0
    1... .... = On-link flag(L): Set
    .1... .... = Autonomous address-configuration flag(A): Set
    ..0. .... = Router address flag(R): Not set
    ...0 0000 = Reserved: 0
  Valid Lifetime: 2592000
  Preferred Lifetime: 604800
  Reserved
  Prefix: it.ip6.fig.ol.no (2001:700:1100:3::)
```

- Routeren oppgir følgende om 2001:700:1100:3::/64
 - Prefikset er direkte tilgjengelig
 - Autokonfigurasjon er tillatt
 - Genererte adresser er gyldige i 30 dager, med foretrukket levetid på 7 dager

Neighbor Discovery

Neighbor Solicitation

Internet Protocol Version 6, Src: 2001:700:1100:3:226:18ff:fe2:7240, Dst: ff02::1:ff52:67e2

0110 = Version: 6
.... 0000 0000 = Traffic class: 0x00000000
.... 0000 0000 0000 0000 = Flowlabel: 0x00000000

Payload length: 32

Next header: ICMPv6 (58)

Hop limit: 255

Source: pc226-02-w7.fig.ol.no (2001:700:1100:3:226:18ff:fe2:7240)

Destination: ff02::1:ff52:67e2

Internet Control Message Protocol v6

Type: Neighbor Solicitation (135)

Code: 0

Checksum: 0x4571 [correct]

Reserved: 00000000

Target Address: monitor2.fig.ol.no (2001:700:1100:3:20b:dbff:fe52:67e2)

ICMPv6 Option (Source link-layer address : 00:26:18:f2:72:40)

Type: Source link-layer address (1)

Length: 1 (8 bytes)

Link-layer address: AsustekC_f2:72:40 (00:26:18:f2:72:40)

- I dette tilfellet ville

- ① 2001:700:1100:3:226:18FF:FEF2:7240 sjekke om
- ② 2001:700:1100:3:20B:DBFF:FE52:67E2 fortsatt var i live

- Forespørselen ble sendt til «Solicited-node multicast-adressa» FF02::1:FF52:67E2

Neighbor Discovery

Neighbor Advertisement

```
Internet Protocol Version 6, Src: 2001:700:1100:3:20b:dbff:fe52:67e2, Dst: 2001:700:1100:3:226:18ff:fe2:7240
  0110 .... = Version: 6
  .... 0000 0000 .... = Traffic class: 0x00000000
  .... 0000 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
  Payload length: 32
  Next header: ICMPv6 (58)
  Hop limit: 255
  Source: monitor2.fig.ol.no (2001:700:1100:3:20b:dbff:fe52:67e2)
  Destination: pc226-02-w7.fig.ol.no (2001:700:1100:3:226:18ff:fe2:7240)
Internet Control Message Protocol v6
  Type: Neighbor Advertisement (136)
  Code: 0
  Checksum: 0x157e [correct]
  Flags: 0x60000000
    0... .. = Router: Not set
    .1.. .. = Solicited: Set
    ..1. .... = Override: Set
    ...0 0000 0000 0000 0000 0000 0000 = Reserved: 0
  Target Address: monitor2.fig.ol.no (2001:700:1100:3:20b:dbff:fe52:67e2)
  ICMPv6 Option (Target link-layer address : 00:0b:db:52:67:e2)
    Type: Target link-layer address (2)
    Length: 1 (8 bytes)
    Link-layer address: DellEsgP_52:67:e2 (00:0b:db:52:67:e2)
```

Neighbor Discovery

Neighbor Advertisement

```
Internet Control Message Protocol v6
Type: Neighbor Advertisement (136)
Code: 0
Checksum: 0x157e [correct]
Flags: 0x60000000
 0... .. = Router: Not set
 .1.. .. = Solicited: Set
 ..1. .... = Override: Set
 ...0 0000 0000 0000 0000 0000 0000 = Reserved: 0
Target Address: monitor2.fig.ol.no (2001:700:1100:3:20b:dbff:fe52:67e2)
ICMPv6 Option (Target link-layer address : 00:0b:db:52:67:e2)
  Type: Target link-layer address (2)
  Length: 1 (8 bytes)
  Link-layer address: DellEsgP_52:67:e2 (00:0b:db:52:67:e2)
```

- 2001:700:1100:3:20B:DBFF:FE52:67E2 sendte svar tilbake til 2001:700:1100:3:226:18FF:FEF2:7240 med klar beskjed om at
 - Den er ikke en router
 - Dette er et svar på en forespørsel og ikke en tilfeldig annonsering
 - Gamle opplysninger om 2001:700:1100:3:20B:DBFF:FE52:67E2 skal slettes
 - Lag-2-adressa er stadig 00:0B:DB:52:67:E2

DHCPv6

52 DHCPv6

53 Meldinger

54 DHCP Unique Identifier

55 Identity association

56 Identity association identifier

- DHCPv6 er definert i [RFC 3315](#) med oppdateringer fra [RFC 3319](#), [RFC 3633](#), [RFC 3646](#), [RFC 3736](#), [RFC 4361](#), [RFC 5007](#), [RFC 5494](#), [RFC 6221](#), [RFC 6422](#), [RFC 6603](#), [RFC 6644](#) og [RFC 7083](#)
- Kommunikasjonen foregår først med multicast og UDP, og kan senere bytte til unicast og UDP
- Klientene bruker port 546, mens serverne og relay-agentene bruker port 547
- Klientene bruker sin egen link-local-adresse som avsender og multicast-adressa FF02::1:2 som mottaker
- Relay-agentene videresender til multicast-adressa FF05::1:3, med mindre de kjenner og vil bruke unicast-adressa til serveren
- Serverne svarer med sin link-local-adresser som avsender og klientens link-local-adresse som mottaker

- Solicit
 - Fra klient til server/relay
 - Brukes for å oppdage servere
- Advertise
 - Fra server/relay til klient
 - Brukes for å varsle klienten om tjenestetilbudet
- Request
 - Fra klient til spesifikk server
 - Bruker for å etterspørre om adresser og andre opplysninger fra en bestemt server
- Confirm
 - Fra server/relay til klient
 - Brukes for å bestemme om tidligere oppgitt adresse fortsatt er gyldig

- Renew

- Fra klient til server/relay
- Brukes for å fornye leieavtalen og oppdatere andre opplysninger

- Rebind

- Fra klient til server/relay
- Brukes til annonsering i etterkant av en renew-melding, dersom det ikke kom noe svar på fornyelsen

- Reply
 - Fra server til klient
 - Serveren sender tildelt adresse og andre opplysninger i en reply-melding som svar på solicit-, request-, renew- og rebind-meldinger
 - Serveren sender konfigurasjonsparametre i en reply-melding som svar på en information-request-melding
 - Serveren sender en reply-melding som svar på en confirm-melding for å bekrefte eller avkrefte at adressa tilordnet klienten er gyldig eller ikke
 - Serveren sender en reply-melding for å kvittere for mottatt release- eller decline-meldinger
- Release
 - Fra klient til server/relay
 - Brukes for å frigjøre en utleid adresse

- Decline
 - Fra klient til server/relay
 - Brukes for å fortelle at en eller flere utdelte adresser allerede er tatt i bruk i nabolaget til klienten
- Reconfigure
 - Fra server til klient
 - Brukes for å gjøre klienten oppmerksom på nye opplysninger og at klienten må gjennomføre renew/reply- eller information-request/reply-transaksjoner for å få de nye opplysningene
- Information-request
 - Fra klient til server/relay
 - Brukes for å be om konfigurasjonsparametre uten å bli tildelt en adresse

- Relay-forward
 - Fra relay til relay/server
 - Brukes av relay for å videresende forespørsler fra klienter eller andre relay til en ny relay eller server
- Relay-reply
 - Fra server/relay til relay
 - Brukes av server for å videresende svar tilbake til klienter gjennom relay(kjeden)

- Klientene identifiseres med DHCP Unique Identifier, DUID, som har variabel lengde og format
- Klientene kan ha flere nettverksgrensesnitt
- Hvert grensesnitt har i tillegg sin Identity Association Identifier, IAID, lengde 32 bit
- Klientene oppgir aktuell DUID og IAID i forespørslene
- DHCPv6-serverne har sine egne DUID og IAID, og oppgir disse i svarene

- DUID finnes i tre varianter:
 - Type 1: Linklagsadresse med tidspunkt for generering, DUID-LLT
 - Type 2: Unik identifikator basert på Enterprise-nummer utdelt av IANA, DUID-EN
 - Type 3: Linklagsadresse, DUID-LL

- Type 1 kan se slik ut:

00 01 00 01 13 10 43 9B 00 26 18 F2 72 40

- 00 01 angir at dette er DUID type 1.
- 00 01 angir at det kommer en MAC-48-adresse til slutt
- 13 10 43 9B angir klokkeslettet målt i sekunder siden 1. januar 2000 UTC
 - I dette tilfellet: 0x1310439B s, 319832987 s, 10.1351038909 år etter 1. januar 2000 UTC, altså 18. februar 2010, kl. 18:29:47 UTC
- 00 26 18 F2 72 40 er MAC-48-adressa for systemet som dette eksempelet er hentet fra

- Type 3 kan se slik ut:

00 03 00 01 00 26 18 F2 72 40

- 00 03 angir at dette er DUID type 3.
- 00 01 angir at det kommer en MAC-48-adresse til slutt
- 00 26 18 F2 72 40 er MAC-48-adressa for systemet som dette eksempelet er hentet fra

- Type 1 er vanlig i Windows, og lagres i Dhcpv6DUID i `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\TCPIP6\Parameters`
- Denne verdien må slettes før man lager et image av oppsettet, ellers vil alle maskinene identifisere seg som den samme klienten
- Type 3 er enklere og mer forutsigbart, og er det beste valget for statisk tildeling av IPv6-adresse via DHCPv6, spesielt med tanke på reinstallasjon av OS
- Jeg har ikke funnet noen måte å tvinge en bestemt DUID-type i Windows, annet enn å sette Dhcpv6DUID manuelt eller gjennom skript, og naturlig nok restarte Windows etterpå
- **Dibbler** og Unix-systemer er tradisjonelt langt snillere, og lar oss angi i konfigurasjonen de gangene vi ønsker DUID-LL istedet for DUID-LLT

- RFC 3315
- Bla, bla, bla

- RFC 3315
- Bla, bla, bla

Avansert multicast

57 Multicastflaggene

58 Når T er satt til 1

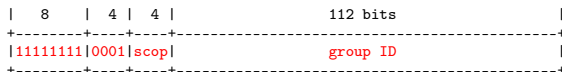
59 Når PT er satt til 11

60 Når RPT er satt til 111

- Flaggene heter ORPT (null, err, pe, te)
- Flagget T angir med 0 at adressa er velkjent (definert av [IANA](#)), og med 1 at adressa er midlertidig (lokalt definert)
- Flagget P angir med 1 at adressa inneholder et unicast-prefiks og skal følge reglene i [RFC 3306](#)
- Flagget R angir med 1 at adressa også inneholder et møtepunkt («rendezvous point») og skal følge reglene i [RFC 3956](#)
- Flaggene P og R gjør det enkelt å lage egne multicast-adresser for internt bruk i organisasjonen

Avansert multicast

Når T er satt til 1



- Adresseformatet er gitt av [RFC 4291](#)
- De 12 mest signifikante bitene må beholdes som vist
- Rekkevidden settes til ønsket, lovlig verdi
- De 112 øvrige bitene kan settes fritt
- Eksempel:
 - FF12:DEAD:BEEF:CAFE:0:FACE:B00C:1
 - En midlertidig, link-local multicast-adresse

Avansert multicast

Når PT er satt til 11

	8		4		4		8		8		64		32	
+-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+
	11111111		0011		scop		reserved		plen		network prefix		group ID	
+-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+	-----+

- Adresseformatet er gitt av [RFC 3306](#)
- De 12 mest signifikante bitene må beholdes som vist
- Rekkevidden settes til ønsket, lovlig verdi, og rekkevidden skal ikke overskride *utbredelsen* av det angitte nettverksprefikset
- Feltet «plen» settes til **prefikslengden til nettverksprefikset for subnettet ditt**,
 $0 < plen \leq 64$
- **Nettverksprefikset** er unicast-prefikset for subnettet ditt
- «Group ID» settes i tråd med retningslinjene til [RFC 3307](#)

Avansert multicast

Når PT er satt til 11

	8		4		4		8		8		64		32	
+	-----	+	-----	+	-----	+	-----	+	-----	+	-----	+	-----	+
	11111111		0011		scop		reserved		plen		network prefix		group ID	
+	-----	+	-----	+	-----	+	-----	+	-----	+	-----	+	-----	+

- Eksempler:

- FF3E:0030:2001:700:1100:0:1337:1337
- Den første adressa er begrenset til internett (global, 48-bit)
- FF38:0030:2001:700:1100:0:1337:1337
- Den andre adressa er begrenset til FSI, gitt at FSI er utgangspunktet (organizational-local, 48-bit)
- FF32:0040:2001:700:1100:3:1337:1337
- Den tredje adressa er begrenset til IT-avdelingen ved FSI, gitt at IT-avdelingen er utgangspunktet (link-local, 64-bit)

Avansert multicast

Når RPT er satt til 111

8	4	4	4	4	8	64	32								
+-----+-----+-----+-----+-----+-----+-----+-----+															
11111111		0111		scop		rsvd		RIID		plen		network prefix		group ID	
+-----+-----+-----+-----+-----+-----+-----+-----+															

- Adresseformatet er gitt av [RFC 3956](#)
- De 12 mest signifikante bitene må beholdes som vist
- Rekkevidden settes til ønsket, lovlig verdi, og rekkevidden skal ikke overskride *utbredelsen* av det angitte nettverksprefikset
- Feltet «RIID» settes til *møtepunktets grensesnittidentifikator*
 - Feltet «RIID» kan ikke være 0, for dette skaper konflikt med «Subnet-Router Anycast Address» fra [RFC 3513](#)
- Feltet «plen» settes til *prefikslengden til nettverksprefikset for subnettet ditt*, $0 < plen \leq 64$
- *Nettverksprefikset* er unicast-prefikset for subnettet ditt
- «Group ID» settes i tråd med retningslinjene til [RFC 3307](#)

Avansert multicast

Når RPT er satt til 111

8	4	4	4	4	8	64	32
11111111 0111 scop rsvd RIID plen network prefix group ID							

- Eksempel:

- FF78:0130:2001:700:1100:0:1337:1337

- Denne adressa er begrenset til organization-local
- Nettverksprefikset er 2001:700:1100::/48
- Møtepunktets adresse er 2001:700:1100::1
- Møtepunktets adresse må konfigureres på et loopbackgrensesnitt i Fagskolens ytterste IPv6-multicast-router
- interface Loopback1
ipv6 address 2001:700:1100::1/128

Konfigurasjon av IPv6

61 Cisco IOS

- IPv6-unicast-routing
- IPv6-multicast-routing
- ACL-er
- DHCPv6
- Sperre for fremmed routerannonsering
- Sperre for falske DHCPv6-servere
- Kombinert ACL for kantporter

62 OS-konfig

Konfigurasjon av IPv6

Cisco IOS: IPv6-unicast-routing

- ❶ `configure terminal`
- ❷ `sdm prefer dual-ipv4-and-ipv6 default`
- ❸ `end`
- ❹ `reload`
- ❺ `configure terminal`
- ❻ `ip routing`
- ❼ `ipv6 unicast-routing`
- ❽ `no ipv6 source-route`
- ❾ `end`

(Rekonfigurere TCAM)

(Nødvendig for IP-routing i det hele tatt)

(Er unødvendig i nyere IOS)

Konfigurasjon av IPv6

Cisco IOS: IPv6-unicast-routing

- ❶ `interface GigabitEthernet0/49`
- ❷ `description Linknett mellom FiG og Uninett/HiG`
- ❸ `no switchport`
- ❹ `ip address 128.39.70.170 255.255.255.252`
- ❺ `ip access-group InetIPv4Inn in`
- ❻ `ip access-group InetIPv4Ut out`
- ❼ `ip pim sparse-mode`
- ❽ `ip igmp version 3`
- ❾ `ipv6 address 2001:700:0:8074::2/64`
- ❿ `ipv6 nd ra suppress`
- ⓫ `ipv6 traffic-filter InetIPv6Inn in`
- ⓬ `ipv6 traffic-filter InetIPv6Ut out`

Konfigurasjon av IPv6

Cisco IOS: IPv6-unicast-routing

① Default route:

```
ipv6 route ::/0 GigabitEthernet0/49 2001:700:0:8074::1
```

② Nullroute linknettet, og offisielle og private adresser:

```
ipv6 route 2001:700:0:8074::/64 Null0
```

```
ipv6 route 2001:700:1100::/48 Null0
```

```
ipv6 route FD5C:14CF:C300::/48 Null0
```

③ Statisk routing av returtrafikk til VPN-klientene:

```
ipv6 route FD5C:14CF:C300:A000::/52 Vlan29 2001:700:1100:F002::2
```


Konfigurasjon av IPv6

Cisco IOS: IPv6-unicast-routing

- ❶ `interface Vlan40`
- ❷ `description Klasserom 100`
- ❸ `ip address 128.39.194.1 255.255.255.192`
- ❹ `ip access-group Vlan40IPv4InnFra in`
- ❺ `ip access-group Vlan40IPv4UtTil out`
- ❻ `ip helper-address 128.39.174.42`
- ❼ `ip pim sparse-mode`
- ❽ `ip igmp version 3`
- ❾ `ipv6 address 2001:700:1100:8001::1/64`
- ❿ `ipv6 nd other-config-flag`
- ⓫ `ipv6 nd router-preference High`
- ⓬ `ipv6 dhcp server offisiell`
- ⓭ `ipv6 traffic-filter Vlan40IPv6Infra in`
- ⓮ `ipv6 traffic-filter Vlan40IPv6UtTil out`

- 1 Global konfigurasjon:
`ipv6 multicast-routing`
- 2 Begrense utbredelse av intern multicasttrafikk
`interface GigabitEthernet0/49`
`ipv6 multicast boundary scope 8`
Bare trafikk med rekkevidde større enn 8 slipper ut på, og inn fra, internett
- 3 Kan du ikke bruke `ipv6 multicast boundary scope`, så må du bruke ACL-er og sperre for uaktuelle rekkevidder og *alle* mulige kombinasjoner av flagg! (Bare for å være føre var.)
- 4 Derfor burde flagg og rekkevidde ha omvendt rekkefølge i multicastadressene, men det toget har forlengst gått ...

- Alle flagg og rekkevidde lik 3

```
deny ipv6 any FF03::/16
deny ipv6 any FF13::/16
deny ipv6 any FF23::/16
deny ipv6 any FF33::/16
deny ipv6 any FF43::/16
deny ipv6 any FF53::/16
deny ipv6 any FF63::/16
deny ipv6 any FF73::/16
deny ipv6 any FF83::/16
deny ipv6 any FF93::/16
deny ipv6 any FFA3::/16
deny ipv6 any FFB3::/16
deny ipv6 any FFC3::/16
deny ipv6 any FFD3::/16
deny ipv6 any FFE3::/16
deny ipv6 any FFF3::/16
```

- Alle flagg og rekkevidde lik 4

```
deny ipv6 any FF04::/16
deny ipv6 any FF14::/16
deny ipv6 any FF24::/16
deny ipv6 any FF34::/16
deny ipv6 any FF44::/16
deny ipv6 any FF54::/16
deny ipv6 any FF64::/16
deny ipv6 any FF74::/16
deny ipv6 any FF84::/16
deny ipv6 any FF94::/16
deny ipv6 any FFA4::/16
deny ipv6 any FFB4::/16
deny ipv6 any FFC4::/16
deny ipv6 any FFD4::/16
deny ipv6 any FFE4::/16
deny ipv6 any FFF4::/16
```

- Alle flagg og rekkevidde lik 5:

```
deny ipv6 any FF05::/16
deny ipv6 any FF15::/16
deny ipv6 any FF25::/16
deny ipv6 any FF35::/16
deny ipv6 any FF45::/16
deny ipv6 any FF55::/16
deny ipv6 any FF65::/16
deny ipv6 any FF75::/16
deny ipv6 any FF85::/16
deny ipv6 any FF95::/16
deny ipv6 any FFA5::/16
deny ipv6 any FFB5::/16
deny ipv6 any FFC5::/16
deny ipv6 any FFD5::/16
deny ipv6 any FFE5::/16
deny ipv6 any FFF5::/16
```

- Alle flagg og rekkevidde lik 6:

```
deny ipv6 any FF06::/16
deny ipv6 any FF16::/16
deny ipv6 any FF26::/16
deny ipv6 any FF36::/16
deny ipv6 any FF46::/16
deny ipv6 any FF56::/16
deny ipv6 any FF66::/16
deny ipv6 any FF76::/16
deny ipv6 any FF86::/16
deny ipv6 any FF96::/16
deny ipv6 any FFA6::/16
deny ipv6 any FFB6::/16
deny ipv6 any FFC6::/16
deny ipv6 any FFD6::/16
deny ipv6 any FFE6::/16
deny ipv6 any FFF6::/16
```

- Alle flagg og rekkevidde lik 7:

```
deny ipv6 any FF07::/16
deny ipv6 any FF17::/16
deny ipv6 any FF27::/16
deny ipv6 any FF37::/16
deny ipv6 any FF47::/16
deny ipv6 any FF57::/16
deny ipv6 any FF67::/16
deny ipv6 any FF77::/16
deny ipv6 any FF87::/16
deny ipv6 any FF97::/16
deny ipv6 any FFA7::/16
deny ipv6 any FFB7::/16
deny ipv6 any FFC7::/16
deny ipv6 any FFD7::/16
deny ipv6 any FFE7::/16
deny ipv6 any FFF7::/16
```

- Alle flagg og rekkevidde lik 8:

```
deny ipv6 any FF08::/16
deny ipv6 any FF18::/16
deny ipv6 any FF28::/16
deny ipv6 any FF38::/16
deny ipv6 any FF48::/16
deny ipv6 any FF58::/16
deny ipv6 any FF68::/16
deny ipv6 any FF78::/16
deny ipv6 any FF88::/16
deny ipv6 any FF98::/16
deny ipv6 any FFA8::/16
deny ipv6 any FFB8::/16
deny ipv6 any FFC8::/16
deny ipv6 any FFD8::/16
deny ipv6 any FFE8::/16
deny ipv6 any FFF8::/16
```

- Hadde bare flagg og rekkevidde byttet plass i spesifikasjonen:

```
deny ipv6 any FF30::/12
```

```
deny ipv6 any FF40::/12
```

```
deny ipv6 any FF50::/12
```

```
deny ipv6 any FF60::/12
```

```
deny ipv6 any FF70::/12
```

```
deny ipv6 any FF80::/12
```

- Dette ville bare gitt 6 regler i ACL-ene
- Det er en sterk kontrast til de 96 reglene som vi må bruke i ACL-ene når vi ikke kan bruke `ipv6 multicast boundary scope 8`

Konfigurasjon av IPv6 I

Cisco IOS: ACL-er

- ❶ `configure terminal`
- ❷ `ipv6 access-list access-list-name`
- ❸ `deny | permit protocol {source-ipv6-prefix/prefix-length | any |
host source-ipv6-address} [operator port-number]
{destination-ipv6-prefix/prefix-length | any |
host destination-ipv6-address} [operator port-number] [dest-option]
[dest-option-type value] [dscp value] [flow-label value] [fragments] [hbh]
[log] [log-input] [mobility] [mobility-type value] [reflect access-list-name]
[routing] [routing-type value] [sequence value] [time-range name]
[undetermined-transport]`

Konfigurasjon av IPv6 II

Cisco IOS: ACL-er

- ❹ deny | permit tcp {*source-ipv6-prefix/prefix-length* | any |
host *source-ipv6-address*} [*operator port-number*]
{*destination-ipv6-prefix/prefix-length* | any |
host *destination-ipv6-address*} [*operator port-number*] [ack] [dest-option]
[dest-option-type *value*] [dscp *value*] [established] [fin] [flow-label *value*]
[hbh] [log] [log-input] [mobility] [mobility-type *value*] [psh]
[reflect *access-list-name*] [routing] [routing-type *value*] [rst]
[sequence *value*] [syn] [time-range *name*] [urg]
- ❺ deny | permit udp {*source-ipv6-prefix/prefix-length* | any |
host *source-ipv6-address*} [*operator port-number*]
{*destination-ipv6-prefix/prefix-length* | any |
host *destination-ipv6-address*} [*operator port-number*] [dest-option]
[dest-option-type *value*] [dscp *value*] [flow-label *value*] [hbh] [log]
[log-input] [mobility] [mobility-type *value*] [reflect *access-list-name*]
[routing] [routing-type *value*] [sequence *value*] [time-range *name*]

Konfigurasjon av IPv6 III

Cisco IOS: ACL-er

- ⑥ `deny | permit icmp {source-ipv6-prefix/prefix-length | any |
host source-ipv6-address} {destination-ipv6-prefix/prefix-length | any |
host destination-ipv6-address} [{icmp-type [icmp-code]} | icmp-message]
[dest-option] [dest-option-type value] [dscp value] [flow-label value] [log]
[log-input] [mobility] [mobility-type value] [reflect access-list-name]
[routing] [routing-type value] [sequence value] [time-range name]`
- ⑦ `evaluate reflexive-access-list-name [sequence value]`
- ⑧ `remark comment`
- ⑨ `exit`
Husk:
 $operator \in \{gt | lt | neq | eq | range\}$
reflect er bare gyldig for permit-regler

Konfigurasjon av IPv6 IV

Cisco IOS: ACL-er

- 10 interface *interface-id*
- 11 ipv6 traffic-filter *access-list-name* {in | out}
- 12 end

- Alle IPv6-ACL-er har følgende 5 regler innebygget (eng. implicit) på slutten:
 - 1 permit icmp any any nd-na
 - 2 permit icmp any any nd-ns
 - 3 permit icmp any any router-advertisement
 - 4 permit icmp any any router-solicitation
 - 5 deny ipv6 any any
- Disse reglene tillater Neighbor Discovery, og blokkerer all annen IPv6-trafikk
- Dine egne regler kommer *alltid* før de 5 reglene over, og kanskje må du kopiere de innebygde reglene og gjøre dine egne justeringer, for eksempel slå på logging av blokkert trafikk

- Ønsker du logging av blokkert trafikk, men vil samtidig ikke blokkere Neighbor Discovery, så må du gjøre slik:

- 1 remark Øvrige regler kommer før denne linja
- 2 permit icmp any any nd-na
- 3 permit icmp any any nd-ns
- 4 permit icmp any any router-advertisement
- 5 permit icmp any any router-solicitation
- 6 deny ipv6 any any **log**
- 7 remark Her kommer de skjulte, implisitte reglene
- 8 *permit icmp any any nd-na*
- 9 *permit icmp any any nd-ns*
- 10 *permit icmp any any router-advertisement*
- 11 *permit icmp any any router-solicitation*
- 12 *deny ipv6 any any*

Konfigurasjon av IPv6

Cisco IOS: DHCPv6

- `ipv6 dhcp pool offisiell`
 - `dns-server 2001:700:1100:1::3`
 - `dns-server 2001:700:1100:1::2`
 - `domain-name fig.ol.no`
 - `sntp address 2001:700:1100:1::2`
 - `sntp address 2001:700:1100:1::3`
 - `sntp address 2001:700:1100:1::4`
 - `information refresh 0 2`
- `interface Vlan40`
 - `ipv6 dhcp server offisiell`

Konfigurasjon av IPv6

Cisco IOS: DHCPv6

- `ipv6 dhcp pool ULA`
 - `dns-server 2001:700:1100:1::3`
 - `dns-server 2001:700:1100:1::2`
 - `domain-name fig.netlocal`
 - `sntp address 2001:700:1100:1::2`
 - `sntp address 2001:700:1100:1::3`
 - `sntp address 2001:700:1100:1::4`
 - `information refresh 0 2`
- `interface Vlan31`
 - `ipv6 dhcp server ULA`

- `ipv6 dhcp pool dynamisk-utdeling-vlan60`
 - `address prefix 2001:700:1100:6::/64`
 - `dns-server 2001:700:1100:1::3`
 - `dns-server 2001:700:1100:1::2`
 - `domain-name fig.ol.no`
 - `sntp address 2001:700:1100:1::2`
 - `sntp address 2001:700:1100:1::3`
 - `sntp address 2001:700:1100:1::4`
 - `information refresh 0 2`
- `interface Vlan60`
 - `ipv6 address 2001:700:1100:6::1/64`
 - `ipv6 nd managed-config-flag`
 - `ipv6 nd other-config-flag`
 - `ipv6 nd router-preference High`
 - `ipv6 dhcp server dynamisk-utdeling-vlan60`

Konfigurasjon av IPv6

Cisco IOS: Sperre for fremmed routerannonsering

- Fremmed routerannonsering må sperres i inngående retning på kantporter
- Nyere IOS har egne kommandoer for dette:
 - `interface range GigabitEthernet0/1 - 48`
 - `ipv6 nd rguard`
- Eldre IOS må bruke port-ACL-er for å oppnå det samme:
 - `ipv6 access-list sperre-fremmed-RA`
 - 1 `deny icmp any any router-advertisement`
 - 2 `permit ipv6 any any`
 - `interface range GigabitEthernet0/1 - 48`
 - `ipv6 traffic-filter sperre-fremmed-RA in`

Konfigurasjon av IPv6

Cisco IOS: Sperre for falske DHCPv6-servere

- Falske DHCPv6-servere må sperres i kantportene, og det beste er å bruke port-ACL-er:
 - `ipv6 access-list sperre-falske-dhcpv6-servere`
 - ❶ `deny udp any eq 547 any`
 - ❷ `permit ipv6 any any`
 - `interface range GigabitEthernet0/1 - 48`
 - `ipv6 traffic-filter sperre-falske-dhcpv6-servere in`

- Kombinert ACL for kantporter
 - ipv6 access-list kantporter
 - ❶ deny icmp any any router-advertisement
 - ❷ deny udp any eq 547 any
 - ❸ permit ipv6 any any
 - interface range GigabitEthernet0/1 - 48
 - ipv6 traffic-filter kantporter in

- De fleste moderne operativsystemer har IPv6-støtte
- Windows 2000 har en eksperimentell IPv6-protokoll, men mangler DNS-oppslag for AAAA
- IPv6 må installeres manuelt i Windows XP og Server 2003
 - DNS-oppslag sendes alltid over IPv4
 - Noe av AD-trafikken sendes alltid over IPv4
 - RDP-server i XP og Server 2003 kan bare bruke IPv4
- IPv6 er påskrudd i Windows Vista, Server 2008 og nyere versjoner
 - DNS-oppslag kan nå sendes over IPv6
 - Nyere Windows kan fint fungere med bare IPv6
- Linux og *BSD har hatt IPv6-støtte i lang tid
- Autokonfig med tilfeldig grensesnittidentifikator er det mest vanlige for skrivebordssystemer
- Manuell konfigurasjon er mest vanlig for serversystemer

- Windows:

- `netsh interface ipv6 set address "navn-på-grensesnitt" IPv6-adresse`
- `netsh interface ipv6 set route ::/0 "navn-på-grensesnitt" routerens-IPv6-adresse`

- Eksempel:

- `netsh interface ipv6 set address "Lokal tilkobling" 2001:700:1100:8001::1337`
- `netsh interface ipv6 set route ::/0 "Lokal tilkobling" 2001:700:1100:8001::1`

- Konfigurasjon gjennom grafisk grensesnitt i «Kontrollpanelet» er også mulig

- *BSD:

- `ifconfig navn-på-grensesnitt inet6 IPv6-adresse prefixlen prefikslengde`
- `route add -inet6 default routerens-IPv6-adresse`

- Eksempel:

- `ifconfig em0 inet6 2001:700:1100:8001::1337 prefixlen 64`
- `route add -inet6 default 2001:700:1100:8001::1`

- Vanligvis lagres slike innstillinger permanent, for eksempel i `/etc/rc.conf`

- `ifconfig_em0_ipv6="inet6 2001:700:1100:8001::1337 prefixlen 64"`
- `ipv6_defaultrouter="2001:700:1100:8001::1"`

Noen RFC-er om IPv6

63 Noen RFC-er om IPv6

- IPv6-spesifikasjon: [RFC 2460](#), [RFC 5095](#), [RFC 5722](#), [RFC 5871](#), [RFC 6437](#), [RFC 6564](#), [RFC 6935](#) og [RFC 6946](#)
- ICMPv6: [RFC 4443](#) og [RFC 4884](#)
- Neighbor Discovery: [RFC 4861](#), [RFC 5942](#) og [RFC 6980](#)
- Krav til IPv6-noder: [RFC 6434](#)
- Path MTU: [RFC 1981](#)
- DHCPv6: [RFC 3315](#), [RFC 3319](#), [RFC 3633](#), [RFC 3646](#), [RFC 3736](#), [RFC 4361](#), [RFC 5494](#), [RFC 6221](#), [RFC 6422](#), [RFC 6644](#) og [RFC 7083](#)
- Overføring av IPv6-pakker over Ethernet: [RFC 2464](#) og [RFC 6085](#)
- Adressearkitektur: [RFC 4291](#), [RFC 5952](#) og [RFC 6052](#)
- Unicastadresser: [RFC 3587](#)
- ULA: [RFC 4193](#)

- Autokonfigurering av adresser: [RFC 4862](#)
- Tilfeldig grensesnittidentifikator: [RFC 4941](#)
- Prefiks-baserte multicastadresser: [RFC 3306](#), [RFC 3956](#) og [RFC 4489](#)
- IPsec: [RFC 4301](#), [RFC 4302](#), [RFC 4303](#), [RFC 4304](#), [RFC 4307](#), [RFC 4308](#), [RFC 4309](#), [RFC 4312](#), [RFC 4835](#) og [RFC 5996](#)
- For programmerere av nettverksprogrammer: [RFC 3493](#), [RFC 3542](#) og [RFC 4038](#)
- Grunnleggende krav til IPv6-routere hos sluttbrukere (CER): [RFC 7084](#)