

IPv6-foredrag

Grunnleggende og viderekomne

Trond Endrestøl

Fagskolen Innlandet, IT-avdelingen

12. november 2013

- Filene til foredraget er tilgjengelig gjennom:
 - Subversion: `svn co`
`svn://svn.ximalas.info/ipv6-foredrag-grunnleggende`
 - Web: `svnweb.ximalas.info/ipv6-foredrag-grunnleggende`
- `ipv6-foredrag-grunnleggende.foredrag.pdf` vises på lerretet
- `ipv6-foredrag-grunnleggende.handout.pdf` er mye bedre for publikum å se på
- `ipv6-foredrag-grunnleggende.handout.2on1.pdf` og `ipv6-foredrag-grunnleggende.handout.4on1.pdf` er begge velegnet til utskrift
- `*.169.pdf`-filene er i 16:9-format
- `*.1610.pdf`-filene er i 16:10-format

- Foredraget er mekket ved hjelp av [GNU Emacs](#), [AUCTEX](#), [pdfTEX](#) fra [MiKTEX](#), [L^AT_EX](#)-dokumentklassa [beamer](#), [Subversion](#), [TortoiseSVN](#) og [Adobe Reader](#)
- Hovedfila bærer denne identifikasjonen:
`$Ximalas: trunk/ipv6-foredrag-grunnleggende.tex 64
2013-11-12 20:51:38Z trond $`
- Copyright © 2013 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 Hvorfor trenger vi IPv6?
- 3 Andre nyttige ting ved IPv6
- 4 Hvorfor brukes ikke IPv6?
- 5 IPv6 ved Fagskolen Innlandet

Oversikt av hele foredraget

Del 2: IPv6-header

6 IPv6-header

Oversikt av hele foredraget

Del 3: IPv6 over Ethernet

7 IPv6 over Ethernet

Oversikt av hele foredraget

Del 4: Grunnleggende om adresser

- 8 Grunnleggende om adresser
- 9 Adressedemo
- 10 MAC-48-adresser
- 11 Modda IEEE EUI-64-format
- 12 Manuell grensesnittidentifikator
- 13 Tilfeldig grensesnittidentifikator
- 14 Duplicate Address Detection — DAD
- 15 Livsløpet til en adresse
- 16 Spesialadresser

Oversikt av hele foredraget

Del 5: Adresstyper

- 17 Adresstyper
- 18 Link-local-adresser
- 19 Site-local-adresser
- 20 Offentlige unicast-adresser
- 21 Unike, lokale, aggregerbare adresser
- 22 Anycast-adresser
- 23 Multicast-adresser

Oversikt av hele foredraget

Del 6: DNS

24 AAAA og PTR

25 A6

Oversikt av hele foredraget

Del 7: ICMPv6

- 26 ICMPv6
- 27 Multicast Listener Discovery
- 28 Neighbor Discovery
- 29 Router Renumbering
- 30 Inverse Neighbor Discovery
- 31 Version 2 Multicast Listener Report
- 32 Mobile IPv6
- 33 SEcure Neighbor Discovery (SEND)
- 34 Multicast Router Discovery
- 35 FMIPv6
- 36 RPL Control Message
- 37 ILNPv6 Locator Update Message
- 38 Duplicate Address

Oversikt av hele foredraget

Del 8: Neighbor Discovery

- 39 Router Solicitation
- 40 Router Advertisement
- 41 Neighbor Solicitation
- 42 Neighbor Solicitation
- 43 Redirect

Oversikt av hele foredraget

Del 9: DHCPv6

44 DHCPv6

45 Meldinger

46 DHCP Unique Identifier

Oversikt av hele foredraget

Del 10: Avansert multicast

- 47 Multicastflaggene
- 48 Når T er satt til 1
- 49 Når PT er satt til 11
- 50 Når RPT er satt til 111

Oversikt av hele foredraget

Del 11: Konfigurasjon av IPv6

- 51 Cisco IOS
 - IPv6-routing
 - ACL-er
 - DHCPv6

- 52 OS-konfig

- 53 Tunneloppsett

Oversikt av hele foredraget

Del 12: Noen RFC-er om IPv6

54 Noen RFC-er om IPv6

Del I

Kort om IPv6

Oversikt over del 1: Kort om IPv6

- 1 Hva er IPv6?
- 2 Hvorfor trenger vi IPv6?
- 3 Andre nyttige ting ved IPv6
- 4 Hvorfor brukes ikke IPv6?
- 5 IPv6 ved Fagskolen Innlandet

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**

Kort om IPv6

Hva er IPv6?

- 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 mange adresser enn det fullstendige IPv4-adresserommet:
 - $2^{32} = 4.294.967.296$
- Bare 3.702.258.688 IPv4-adresser kan bli brukt som offentlige IPv4-unicast-adresser
- Se Tronds utregning fra 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 m.m.
- Verden går tom for offentlige IPv4-adresser
- «IPokalypsen» er her!
- IANA gikk tom i februar 2011
 - APNIC gikk tom i april 2011
 - RIPE gikk tom i september 2012
- Dersom disse RIR-ene oppfører seg pent:
 - LACNIC kan holde på til juni 2014
 - ARIN kan holde på til desember 2014
 - AFRINIC kan holde på til oktober 2020
- Network Address Translation, Carrier-Grade NAT og Shared Address Space er bare støttebandasje med kort utløpstid
 - Glem det
 - Ende-til-ende-konnektivitet oppnås best uten noen former for adresseoversettelse

Kort om IPv6

Andre nyttige ting ved IPv6

- 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 (kan) brukes når prefikslengda er ulik 64 bit
- Sjekksum er overlatt til høyere og lavere lag
- Fragmentering skal *gjøres* hos avsender, ikke underveis
 - Avsender må sjekke veien lengre fremme og måle smaleste krøttersti
 - Path Maximum Transmission Unit (Path MTU)

- 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](#)

Kort om IPv6

Hvorfor brukes ikke IPv6?

- Markedskreftene råder
- «Vente-og-se»-holdning
- Store selskaper:
 - Kjøper opp små selskaper og hamstrer IPv4-blokker
 - Kjøper IPv4-blokker på ettermarkedet:
 - Microsoft → \$7,5 mill. → Nortel → 666.624 IPv4-adresser
 - Stikker hodet ned i sanda
- 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 (RFC 6598)
 - HTTP/S-tunnelering av rubb og stubb
- Før eller siden blir CGN for kostbar og komplisert å vedlikeholde
- 3G og 4G/LTE klarer kanskje å øke IPv6-presset (RFC 6459)
- IPv6 er det eneste tilgjengelige og realistiske alternativet

- 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
 - 128.39.46.8/30 ble linknett mellom HiG/Uninett og FSI
 - 128.39.46.9 brukes ved HiG
 - 128.39.46.10 brukes ved FSI
 - 128.39.174.0/24 ble subnett og satt opp som servernett og ansattnett, m.m.
 - 128.39.172.0/24 ble subnett og satt opp som nett for datalab
 - 128.39.173.0/24 ble satt opp som 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::1/64` brukes hos HiG
 - `2001:700:0:11D::2/64` brukes hos FSI
- Samme dag ble IPv6 innført for FSI-VLAN-ene 20, 30, 70 og 80:
 - FSI-VLAN 20: `2001:700:1100:1::/64`
 - FSI-VLAN 30: `2001:700:1100:2::/64`
 - FSI-VLAN 70: `2001:700:1100:3::/64`
 - FSI-VLAN 80: `2001:700:1100:4::/64`
- Sommeren 2007: **Genererte** og frivillig **registrerte** ULA-serien `FD5C:14CF:C300::/48` for FSI-VLAN som tidligere bare brukte **RFC 1918**-adresser
 - Fikk første HP-skriver med IPv6 og ville bruke IPv6
 - 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
 - 128.39.172.0/23 brukes nå til klienter på trådløst studentnett
 - 128.39.194.0/24 brukes nå til datalab etter samme mønster som for den gamle 128.39.172.0/24
- Oppland FK (OFK) har ingen planer om å innføre IPv6
- Hordaland FK har satt en IPv6-adresse på webserveren deres, 2a02:20a0:0:3::81:130
- I dag er de fleste brukere ved FSI kasta over til OFK-nettene
- Dette skjedde etter ombygginga 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 48, 128.39.194.192/27 og 2001:700:1100:8008::/64

- Alle FSI-VLAN har både IPv4- og IPv6-adresser
- 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)
 - Basestasjoner og WLAN-kontroller (Før omlegginga)
 - UPS-er
 - Skrivere
 - VPN-klienter

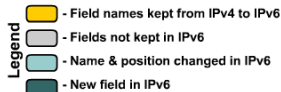
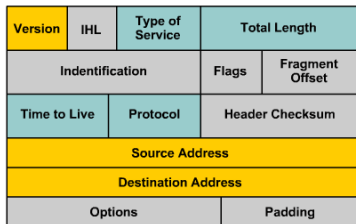
Del II

IPv6-header

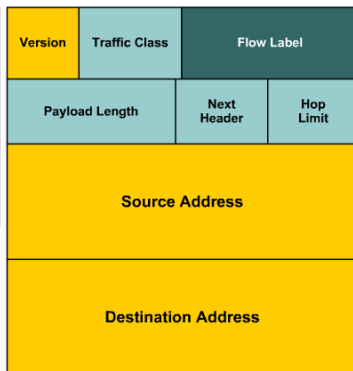
6 IPv6-header

IPv6-header

IPv4 Header

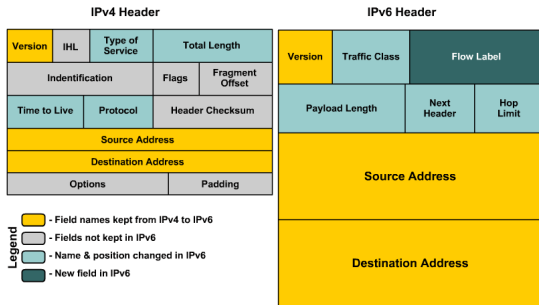


IPv6 Header



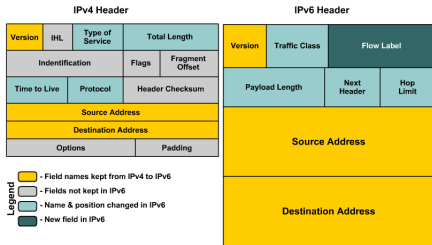
Hentet fra

<http://www.tekkom.dk/mediawiki/images/5/5e/CCNP-108.png> FAGSKOLEN INNLANDET



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

IPv6-header



- 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 og foreløpig eksperimentell

- 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

- Utvidelsesheaderne finnes i stort antall:
 - Hop-by-hop options
 - Destination options
 - Routing
 - Fragment
 - Authentication Header
 - Encapsulating Security Payload
 - Mobility
- Se [RFC 2460](#), [RFC 4302](#), [RFC 4303](#) og [RFC 6275](#)

Del III

IPv6 over Ethernet

7 IPv6 over Ethernet

- [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
- Overføring av hode og hale er vanligvis en oppgave for lag 1
- 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 i lag 3

Del IV

Grunnleggende om adresser

Oversikt over del 4: Grunnleggende om adresser I

- 8 Grunnleggende om adresser
- 9 Adressedemo
- 10 MAC-48-adresser
- 11 Modda IEEE EUI-64-format
- 12 Manuell grensesnittidentifikator
- 13 Tilfeldig grensesnittidentifikator
- 14 Duplicate Address Detection — DAD
- 15 Livsløpet til en adresse
- 16 Spesialadresser

- 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

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:
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:

2001:0700:1100:0003:0000:0000:0000:0000

- Tronds D531:

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:
2001:0700:1100:0003:0221:70FF:FE73:686E

Grunnleggende om adresser

Adressedemo: Ledende nuller

- 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:

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:

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:
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:
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:
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:
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:
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:
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:
 - CCCCCCug (binært)
 - 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
 - Når g-bitet settes lik 1, så blir også u-bitet satt lik 1
 - Kombinasjonen ug = 01 er høyst uvanlig

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)

Grunnleggende om adresser

Modda IEEE EUI-64-format

- Unicast-adresser består av 2 ting:
 - Prefiks
 - Grensesnittidentifikator
- Grensesnittidentifikatorer er på 64 bit
- Grensesnittidentifikatorer kan lages automatisk fra MAC-48-adresser
- Grensesnittidentifikatorer kan også angis manuelt eller velges tilfeldig ([RFC 4941](#))
- Angis grensesnittidentifikatoren manuelt, så angis som regel hele IPv6-adressa manuelt
- Grensesnittidentifikatorer følger [IEEE EUI-64](#)-formatet med to unntak:
 - 1 Universal/local-bitet brukes med *invertert* betydning/verdi
 - Gruppebitet mister sin vanlige betydning i denne sammenhengen
 - 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

Grunnleggende om adresser

Modda IEEE EUI-64-format

- 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
- Fordi IPv6 bruker universal/local-bitet med invertert betydning/verdi, så er arbeidsuhellet akseptert
- Se [RFC 4291](#)

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 med lave verdier
- For eksempel ::53 (DNS-tjener, kanskje)
- Det er ingenting i veien for å «kode» IPv4-adressa inn i IPv6-adressa:
2001:700:1100:3:128:39:174:67
- Man må bare passe på verdien til universal/local-bitet

Grunnleggende om adresser

Manuell grensesnittidentifikator

- Lav verdi for grensesnittidentifikatorer medfører at universal/local-bitet er satt til null:
 - `::53` (heksadesimalt)
 - `::0:0:0:53` (heksadesimalt)
 - `::0000000000000000:00...00:0000000001010011` (binært)
- Uten *invertering* av universal/local-bitet, måtte vi bruke manuelle grensesnittidentifikatorer på denne måten:
 - `::0200:0:0:53` (heksadesimalt)
 - `::0000000100000000:00...00:0000000001010011` (binært)
- Se her:
2001:700:1100:1:0200:0:0:53 vs
2001:700:1100:1::53

Grunnleggende om adresser

Tilfeldig grensesnittidentifikator

- Konstant grensesnittidentifikator truer personvernet
- Eksempel med Tronds lappis:
 - 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 tilfeldig grensesnittidentifikator
- Generér to autokonfigurerte IPv6-adresser:
 - 1 Konstant grensesnittidentifikator (RFC 4291)
 - 2 Tilfeldig grensesnittidentifikator (RFC 4941)
- Velg å bruke adressa med tilfeldig grensesnittidentifikator i størst mulig grad
- Aksepter også innkommende trafikk for adressa med konstant grensesnittidentifikator
- Send svarene tilbake med riktig adresse

- **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

- 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
- 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», hva da?

Grunnleggende om adresser

Livsløpet til en adresse

- En generert adresse vil først være i tilstanden «tentative» ([RFC 4862](#))
- Deretter gjennomføres DAD
- Går alt bra, så sjekker man om én eller flere routere er tilstede
- «ICMPv6 Router Solicitation»
- «ICMPv6 Router Advertisement»
- Routeren annonserer:
 - IPv6-prefiks
 - «Preferred lifetime»
 - «Valid lifetime»
 - Routerens lag-2-adresse

Grunnleggende om adresser

Spesialadresser

- Nulladressa: $0:0:0:0:0:0:0:0/0$ eller $::/0$
 - 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»)
 - Tilsvarende $0.0.0.0/0$ eller $0/0$ 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 generell 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 lyttesoketer for IPv4 og IPv6

Del V

Adresstyper

Oversikt over del 5: Adresstyper

- 17 Adresstyper
- 18 Link-local-adresser
- 19 Site-local-adresser
- 20 Offentlige unicast-adresser
- 21 Unike, lokale, aggregerbare adresser
- 22 Anycast-adresser
- 23 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

- Definerert: [RFC 4291](#)
- Bruksområde:
 - Lokal kommunikasjon internt i VLAN-et
 - Sentral for autokonfigurasjon
 - 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- eller /56-bits prefiks av ISP-ene:
 - /48-bits prefiks gir $64 - 48 = 16$ subnetbit $\rightarrow 2^{16} = 65536$ subnett
 - /56-bits prefiks gir $64 - 56 = 8$ subnetbit $\rightarrow 2^8 = 256$ subnett
- Eksempel: $2001:700:1100:1::1/128$

Adresstyper

Unike, lokale, aggregerbare adresser

- Defineret: **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

Adresstyper

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/>
- **Geoff Huston**, 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

Adresstyper

Unike, lokale, aggregerbare adresser

- 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

- Definerert: [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 foredraget for de viderekomne

- 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

Del VI

DNS

24 AAAA og PTR

25 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.
4.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 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 seg selv når prefikslengda settes til 128

- 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. 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. 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. 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

- 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 får bygd opp følgende adressekjede:

- ::4 svabu.fig.ol.no.
- 0:0:0:1:: ext-servere.ip6.fig.ol.no.
- 0:0:1100:: fig.ip6.uninett.no.
- 2001:700:: uninett.ip6.uninett.no.

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

Del VII

ICMPv6

Oversikt over del 7: ICMPv6 I

- 26 ICMPv6
- 27 Multicast Listener Discovery
- 28 Neighbor Discovery
- 29 Router Renumbering
- 30 Inverse Neighbor Discovery
- 31 Version 2 Multicast Listener Report
- 32 Mobile IPv6
- 33 SEcure Neighbor Discovery (SEND)
- 34 Multicast Router Discovery
- 35 FMIPv6
- 36 RPL Control Message
- 37 ILNIPv6 Locator Update Message
- 38 Duplicate Address

- Feilrapportering- og feilsøkingstjeneste for IPv6
- Definert: [RFC 4443](#) og [RFC 4844](#)
- ICMPv6-meldinger inneholder to tall som forteller noe om budskapets mening og innhold:
 - Type: hovednummer
 - Code: undernummer, settes til 0 når det ikke er definert noen undernummer
- I tillegg er det felter for sjekksum og andre opplysninger som er unike for hver type (og undertype) av meldingene

- 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
- 129: Echo reply
- 200: Private eksperimenter
- 201: Private eksperimenter
- 255: Reservert for utvidelse av informative meldinger

(ping)

(pong)

- Definert: [RFC 2710](#)
- Angir tre nye ICMPv6-meldinger:
 - 130: Multicast Listener Query
 - 131: Multicast Listener Report
 - 132: Multicast Listener Done

- 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 foredraget for de viderekomne

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

- 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

- 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

- Definert: [RFC 3971](#)
- Angir to nye ICMPv6-meldinger:
 - 148: Certification Path Solicitation
 - 149: Certification Path Advertisement

- Definert: [RFC 4286](#)
- Angir tre nye ICMPv6-meldinger:
 - 151: Multicast Router Advertisement
 - 152: Multicast Router Solicitation
 - 153: Multicast Router Termination

- Definert: [RFC 5568](#)
- Angir én ny ICMPv6-melding:
 - 154: FMIPv6

- Definert: [RFC 6550](#)
- Angir én ny ICMPv6-melding:
 - 155: RPL Control Message

- Definert: [RFC 6743](#)
- Angir én ny ICMPv6-melding:
 - 156: ILNPv6 Locator Update Message

- Definert: [RFC 6775](#)
- Angir to nye ICMPv6-meldinger:
 - 157: Duplicate Address Request
 - 158: Duplicate Address Confirmation

Del VIII

Neighbor Discovery

Oversikt over del 8: Neighbor Discovery I

- 39 Router Solicitation
- 40 Router Advertisement
- 41 Neighbor Solicitation
- 42 Neighbor Solicitation
- 43 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 Solititation

- Bla, bla, bla

Neighbor Discovery

Router Advertisement

- Bla, bla, bla

Neighbor Discovery

Neighbor Solicitation

- Bla, bla, bla

Neighbor Discovery

Neighbor Solicitation

- Bla, bla, bla

- Bla, bla, bla

Del IX

DHCPv6

44 DHCPv6

45 Meldinger

46 DHCP Unique Identifier

- DHCPv6 er definert i [RFC 3315](#) med oppdateringer fra [RFC 4361](#), [RFC 5494](#), [RFC 6221](#), [RFC 6422](#) og [RFC 6644](#)
- Kommunikasjonen foregår først med multicast og UDP, senere unicast og UDP
- Klientene bruker port 546 og serverne/relay bruker 547
- Klientene bruker sin egen link-local-adresse som avsender og multicast-adressa FF02::1:2 som mottaker
- Serverne svarer med sin link-local-adresser som avsender og klientens link-local-adresse som mottaker

- Solicit
 - Fra klient til server/relay
- Advertise
 - Fra server/relay til klient
- Request
 - Fra klient til spesifikk server
- Confirm
 - Fra server/relay til klient
- Renew
 - Fra klient til server/relay
- Rebind
 - Fra klient til server/relay

- Reply
 - Fra server til klient
- Release
 - Fra klient til server/relay
- Decline
 - Fra klient til server/relay
- Reconfigure
 - Fra server til klient
- Information-request
 - Fra klient til server/relay
- Relay-forward
 - Fra relay til relay/server

- Relay-reply
 - Fra server/relay til relay

- 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, og oppgir, sine egne DUID og IAID 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
 - 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`
- Type 3 er enklere og mer forutsigbart, og det beste valget for statisk tildeling av IPv6-adresse 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 restartet 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

Del X

Avansert multicast

47 Multicastflaggene

48 Når T er satt til 1

49 Når PT er satt til 11

50 Når RPT er satt til 111

- Bla, bla, bla

- Bla, bla, bla

- Bla, bla, bla

- Bla, bla, bla

Konfigurasjon av IPv6

- 51 Cisco IOS
 - IPv6-routing
 - ACL-er
 - DHCPv6

- 52 OS-konfig

- 53 Tunneloppsett

Konfigurasjon av IPv6

Cisco IOS: IPv6-routing

- 1 configure terminal
- 2 `sdm prefer dual-ipv4-and-ipv6 default` (Rekonfigurere TCAM)
- 3 end
- 4 reload
- 5 configure terminal
- 6 interface *interface-id*
- 7 no switchport (Aktuelt for routergrensesnitt)
- 8 `ipv6 address ipv6-address`
- 9 `ipv6 nd ra suppress` (Aktuelt for routergrensesnitt)
- 10 exit
- 11 `ip routing` (Nødvendig for IP-routing i det hele tatt)
- 12 `ipv6 unicast-routing`
- 13 `no ipv6 source-route`
- 14 end

Konfigurasjon av IPv6

Cisco IOS: IPv6-routing

- 1 interface GigabitEthernet0/49
- 2 description Linknett mellom FiG og Uninett/HiG
- 3 no switchport
- 4 ip address 128.39.46.10 255.255.255.252
- 5 ip access-group InetIPv4Inn in
- 6 ip access-group InetIPv4Ut out
- 7 ip pim sparse-mode
- 8 ip igmp version 3
- 9 ipv6 address 2001:700:0:11D::2/64
- 10 ipv6 nd ra suppress
- 11 ipv6 traffic-filter InetIPv6Inn in
- 12 ipv6 traffic-filter InetIPv6Ut out

- 1 configure terminal
- 2 ipv6 access-list *access-list-name*
- 3 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

```
④ 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]
```


Konfigurasjon av IPv6 III

Cisco IOS: ACL-er

```
5 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 IV

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 ∈ {gt | lt | neq | eq | range}

reflect er bare gyldig for permit-regler

Konfigurasjon av IPv6 V

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

- Bla, bla, bla

- 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/2003, men DNS-oppslag sendes over IPv4(!)
- IPv6 er påskrudd i Windows Vista/2008 og nyere versjoner
- Autokonfig med tilfeldig grensesnittidentifikator er mest vanlig

- Bla, bla, bla

Del XII

Noen RFC-er om IPv6

54 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 4361, RFC 5494, RFC 6221, RFC 6422 og RFC 6644
- 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 4038](#)