C4.

Adressering (§5.2 behalve §5.2.5)

a. Bespreek de *structuur* en *numerieke voorstelling* van IPv6 adressen.
b. Geef en bespreek de verschillende *types* IPv6 adressen. Geef onder andere van elk van deze types de structuur, hun interpretatie, relevante voorbeelden en eventuele subtypes.

C4 a)

IPv6 adressen zijn samengesteld uit 128 bits (16bytes), door de enorme adresruimte zijn lapmiddelen zoals NAT niet meer aan te bevelen. De basisrepresentatie van een IPv6 adres, heeft de vorm P:Q:R:S:T:U:V:W waarbij elk segment een hexadecimaal geheel getal van vier cijfers voorstelt. Voorloopnullen zijn niet vereist. Elk hexadecimaal cijfer representeert ½ adresbyte, elk segment 2 adresbytes. Ter vereenvoudiging kunnen 2 regels toegepast worden:

- Sommige adressen bestaan uit lange reeksen nullen. Één groep van opeenvolgende :0000-strings of 0000:-strings kan hierbij vervangen worden door ::-string

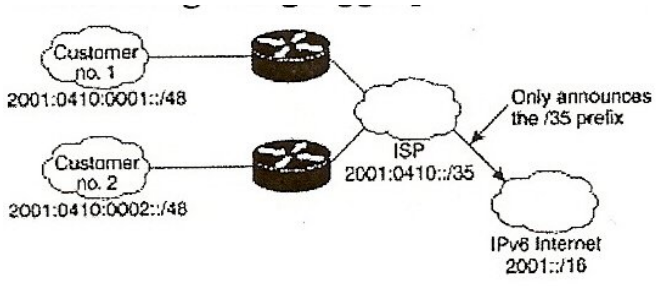
  Bijv 2001:410:1:: ←→ *2001:0410:0001:0000:0000:0000:0000:0000*

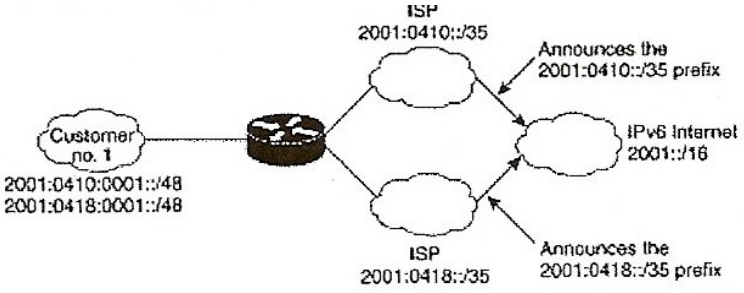
  Deze methode vervangt nullen alleen als ze een volledig segment vormen, de ::-string mag slechts 1 keer in een adres voorkomen

- Een willekeurig segment van 2 adresbytes mag ook voorgesteld worden door de overeenkomstige *dotted-decimale* notatie, waarbij elke adresbyte door een decimaal geheel getal wordt gerepresenteerd

  Bijv FE80::5EFE:192.168.41.30 ←→ *FE80:0:0:0:0:0:5EFE:C0A8:291E*

Net zoals bij IPv4 identificeert het eerste deel van een IPv6 adres, de *netwerkprefix*, het subnetwerk waarop het toestel is aangesloten, terwijl het tweede deel, *de interface-id* , een specifieke computer op dit subnetwerk aanduidt. Een IP-adres waarvan de *interface-id* uit allemaal 0-bits bestaat, wordt beschouwt als de identitficatie van het volledige subnetwerk(netwerkadres), en mag niet toegekend worden aan een individuele interface. In tegenstelling tot IPv4 kan de grootte van de netwerkprefix enkel door middel van de prefixlengte syntax aangeduid worden : *bijv: 2001:410:1::/48* stelt een subnetwerk voor met potentieel 2^{80}-1 knooppunten. De *subnetmasker syntax* kan niet langer gebruikt worden. De *prefixlengte syntax* vereenvoudigt ondermeer de voorstelling van geaggregeerde netwerkadressen.



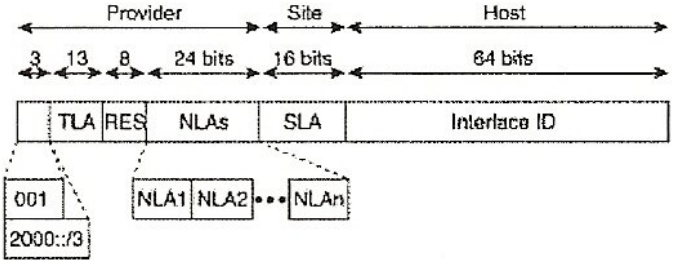


C4 b)

Er zijn 3 types IPv6 adressen: ***unicast, multicast*** en ***anycast:***

Globale Unicast adressen:

Het concept van unicast adressen, die elke 1 enkele interface specifieren, blijft nagenoeg ongewijzigd tov IPv4. Met elke interface moet minstens 1 unicast adres geassocieerd zijn. Unicast adressen die mondiaal uniek moeten zijn, worden globale unicast (Aggregatable Global Unicast) adressen genoemt. Meerdere interfaces kunnen 1 globaal unicast adres delen. Globale unicast adressen worden gekenmerkt door het patroon *001* als eerste 3 bits. Dit wordt het *formaatprefix* (FP) genoemd. De verzameling van alle globale unicast adressen kan bijgevolg genoteerd worden als 2000::/3



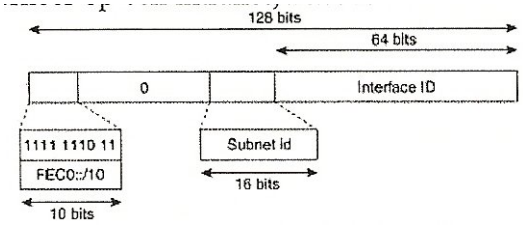
IPv6 adressen zijn zo ruim dat ze in meerdere vaste velden verdeeld worden. Elk van de velden komt overeen met een ander niveau in de hiërarchische adrestoewijzing:

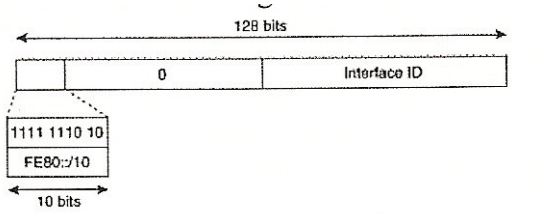
1. De Top-*Level Aggregation Identifier* (TLA) identificeert het hoogste niveau in de routing hiërarchie. TLAs worden toegewezen aan tier-1 ISPs. Aangezien TLA een veld is van 13 bits lang, is het de bedoeling dat de default-free routers van het IPv6 gebaseerde Internet ten hoogste 8912 routes bevatten.
2. De *Next-Level Aggregation Identifiers* (NLAs) worden consequent provider-based toegewezen: netwerken nemen aggregeerbare adressen op basis van de hiërarchie van ISPs bij wie de netwerkdiensten aangekocht worden. De TLA en NLA velden samen worden bepaald door de publieke topologie van het Internet. De netwerkstructuur corresponderend met de NLAs is niet zichtbaar voor de default-free routers.
3. De *Site-Level Aggregation Identifier* (SLA), het 4de segment van de colon-hexadecimal notatie, is voor de klant beschikbaar om de adressering zoveel mogelijk aft e stemmen op de topologie van zijn privaat netwerk. Met de beschikbare 16 bits kan elke organisatie zijn eigen interne netwerkstructuur creëren met behulp van hiërarchie van subnetten.
4. De *interface-id* (laatste 4 segmenten van de colon-hexadecimal notatie) kan willekeurig ingesteld worden, maar kan ook eenduidig bepaald worden door het MAC adres dat in de hardware van netwerkkaarten is gecodeerd.
   Om bijv. een 8byte lang IPv6 *interface-id* te vormen uit het 6-byte grote MAC adres van een Ethernet kaart neemt men achtereenvolgens de eerste 3 bytes van het MAC adres, de string FF:FE, en de laatste 3 bytes van het MAC adres. Deze representatie noemt men het IEEE EUI-64 formaat van een MAC adres. Tenslotte complementeert men de op 1 na laagste bit van de 1ste byte.

Lokale unicast adressen:

IPv6 biedt in zijn adressering ruimte voor unicast adressen die enkel intern in organisatie gebruikte kunnen worden, en niet doorgestuurd worden door routers buiten de organisatie, dit worden lokale unicast adressen. Behalve op basis van hun formaatprefix is er niets dat ze van andere geldige IPv6 adressen onderscheidt. Er zijn de *sitelokale* en de linklokale unicast adressen.

- *Sitelokale* netwerkadressen zijn gekenmerkt door een FEC0:0:0/48, deze bezitten net als de globale unicast adressen een SLA veld van 16 bits dat willekeurige subnetting toelaat. *Sitelokale* unicast adressen zijn dan ook te beschouwen als de opvolgers van de private adresblokken 10/8, 172.16/12 en 192.168/16 van IPv4. *Sitelokale* unicast adressen kunnen gebruikt worden voor het verwerken van verkeer op intranet, zonder dat direct versturen naar het Internet toegestaan wordt.

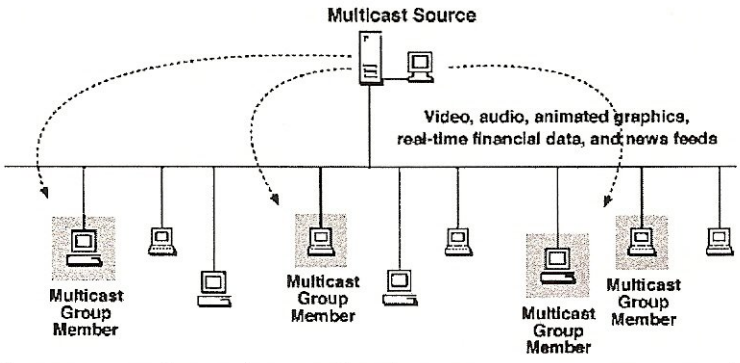




- *Linklokale* unicast adressen zijn gekenmerkt door een FE80:0:0:0/64 prefix en laten geen subnetting toe. Deze adressen zijn bedoeld om tot 2^{64}-1 knooppunten op 1 enkel netwerkverbinding te nummeren.

Multicast adressen:

Het concept van multicast adressen blijven ongewijzigd tov IPv4. Deze kunnen enkel als bestemmingsadres gebruikt worden. Ethernet multicast adressen worden gekenmerkt door een op 1 ingestelde laagste bit van de 1ste byte. IPv6 multicast adressen worden gekenmerkt door een 1ste segment in de colon-hexadecimal notatie van de vorm FFvs: . Het *vlag* veld bevat 4 bits waarvan enkel de laagste momenteel gedefinieerd is.



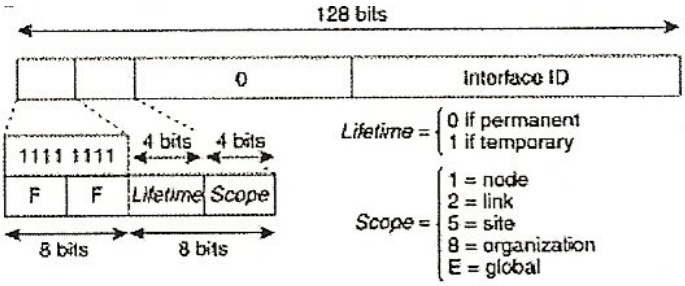
- Staat het *vlag* veld ingesteld op 0, dan is het multicast adres een standaard adres dat door de centrale Internet autoriteit (IANA) permanent is toegewezen, permanent toegewezen multicast adressen hechten aan de resterende 7 bytes bijzondere betekenissen zoals bijv:

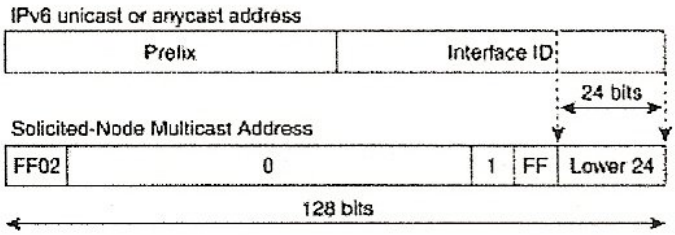
```
          FF0s::1 | alle knooppunten
          FF0s::2 | alle routers
          FF02::5 | alle OSPF routers
          FF02::6 | alle OSPF designated routers
          FF02::9 | alle RIP routers
          FF02::A | alle EIGRP routers
      FF02::1:2   | alle DHCP cliënts
      FF05::1:3   | alle DHCP servers
      FF05::1:4   | alle DHCP relay-agents
FF02::1:255.x.y.z | solicited-node multicast address
```

Het *all-nodes* multicast adres FF02::1 vervangt de broadcast adressen van IPv4.

- Staat het *vlag* veld op 1 dan gaat het om een *transient* (tijdelijk) adres, dat ad hoc kan worden gebruikt, bijv voor videoconferenties, netwerkspelletjes of de verspreiding van financieel nieuws.

Het *scope* veld geeft aan of het multicast adres het globale Internet bestrijkt, of alleen knooppunten kan omvatten die zich op hetzelfde lokale subnetwerk, op dezelfde site, of in dezelfde organisatie bevinden. Multicast adressen hebben geen betekenis buiten hun eigen bereik.

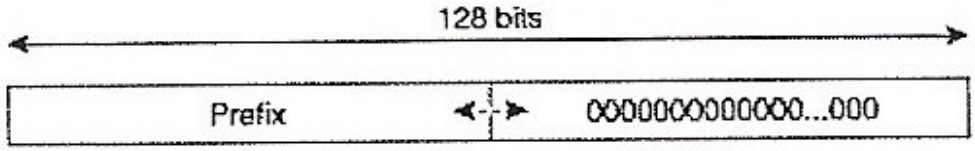




Aan elke interface wordt niet alleen automatisch een linklokaal unicast adres toegewezen, maar ook het *solicited-node* multicast adres. Dit multicast adres wordt gebruikt bij het adres resolutie proces, dat het op broadcast ARP mechanisme van IPv4 vervangt. Het *solicited-node* multicast adres heeft de vorm FF02::1:255.x.y.z, waarbij de laagste 3 bytes bepaald worden door de laatste 3 bytes van het unicast IP adres.

Anycast adressen:

Een *anycast* adres kan door meerdere knooppunten gedeeld worden. In tegenstelling tot multicast doeladressen ontvangt slechts 1 enkel knooppunt een datagram dat naar een *anycast* adres is gestuurd. *Anycast*ing is vooral nuttig voor het verlenen van diensten zoals nameservers en timeservers. Als de cliënt software een informatieaanvraag naar een *anycast* adres stuurt, dan wordt de aanvraag beantwoord door de server die zich het dichtst bij de aanvrager bevindt. *Anycast* adressen worden toegewezen uit de unicast adresruimte, en kunnen niet van unicast adressen onderscheiden worden. Elk lid van een *anycast* adres moet bijgevolg expliciet zo geconfigureerd worden dat het *anycast* adres als dusdanig herkend wordt. Ook elke router infrastructuur moet op de hoogte zijn van elke locatie van ieder *anycast* adres. Elke interface van een router wordt automatisch geconfigureerd met het *subnet-router anycast* adres, bekomen door het netwerkadres aan te vullen met een *interface-id* dat uit allemaal 0-bits bestaat.



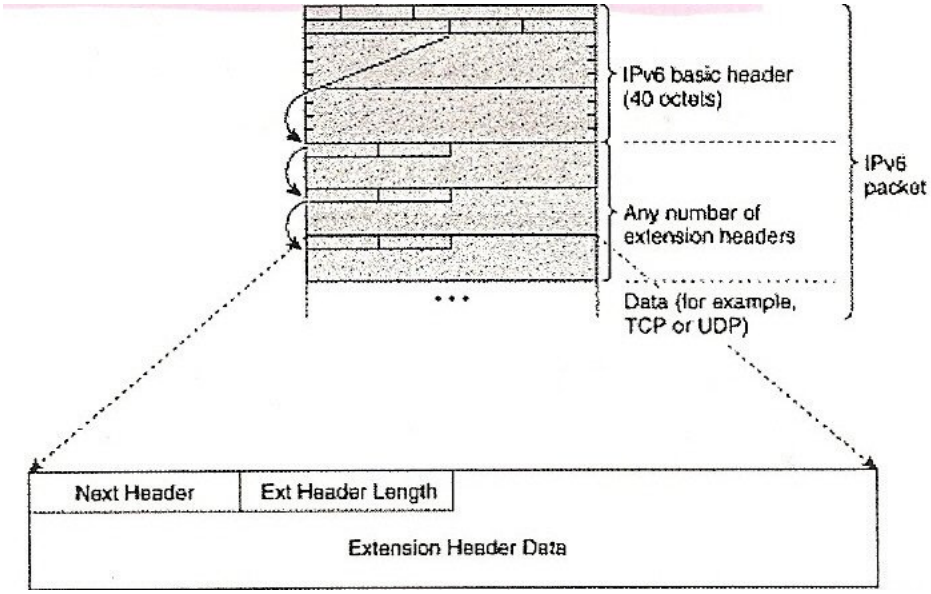
C5.

IPv6 berichtstructuur (§5.3)

a. Bespreek in detail de *structuur* van IPv6 berichten.
b. Geef en bespreek de verschillende soorten *extensieheaders* die momenteel voor IPv6 gedefinieerd zijn.

C5 a)

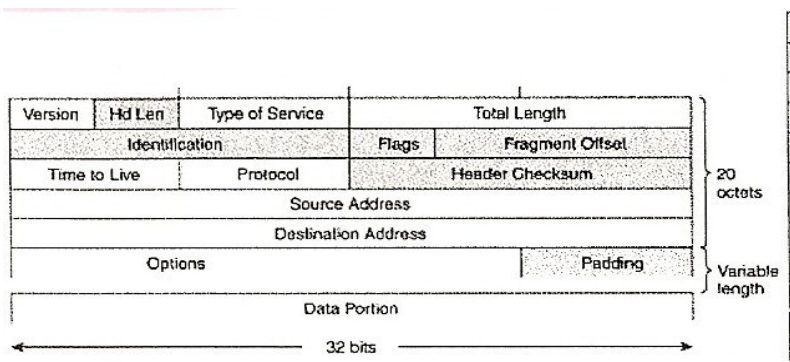
In IPv6 is efficiënte routing gemakkelijker omdat men een *header* van uniforme grootte gebruikt, met minder velden om te onderzoeken en te verwerken. Ook voegt IPv6 opties toe in een gelinkte lijst van aparte *extensieheaders*. Op deze manier hoeven de opties alleen te worden onderzocht als dat werkelijk nodig is.

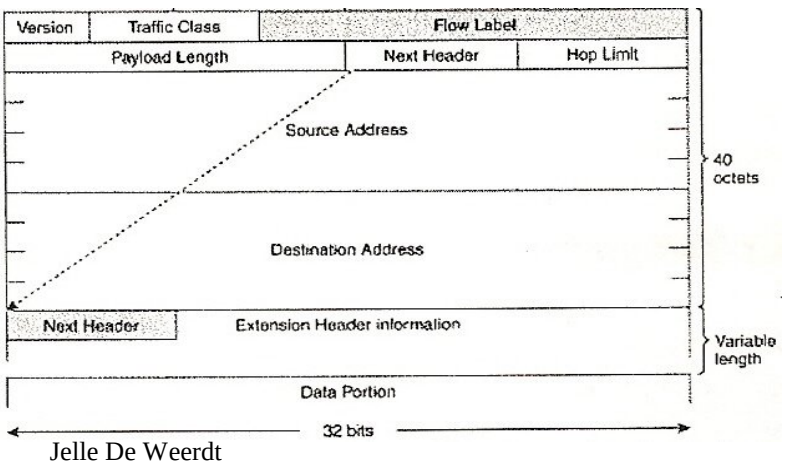


Zowel de *header* als de *extensieheaders* bevatten een *Next Header* veld, dat aangeeft welke type data er na de (*extensie*)*header* komt. Het geheel van de *extensieheaders* en de ingekapselde gegevens wordt de *payload* van een IPv6 datagram genoemd.

IPv6 header:

Tov IPv4 kunnen een aantal velden(grijs gekleurd in figuur) uit de *header* weggelaten worden. Zo vervalt de behoefte aan een veld voor de *headerlengte*, omdat IPv6 *header* een constante grootte heeft van 40 bytes. Vervolgens kunnen verschillende velden uit de *header* verwijderd worden omdat de regels voor datagramfragmentatie veranderd zijn. Tenslotte heeft het behoud van een *checksum* weinig zin, aangezien de foutcontrole toch door de transportlaag uitgevoerd wordt.





De meeste velden van de IPv6 *header* zijn analoog aan hun IPv4 equivalent: het *Version* veld bijv moet ingevuld zijn met 6. Het *Traffic Class* veld is net als het TOS veld van IPv4 bedoeld om een meer gedifferentieerde dienst aan te bieden. De standaardwaarde voor dit veld bestaat uit allemaal 0-bits. Het *Payload Length* veld telt het aantal bytes in het datagram waarbij de *headerextensies* wel, maar de *header* zelf niet in berekening opgenomen worden. Het *Hop Limit* veld stelt een bovengrens aan de levensduur van datagrammen, en moet op de bron, anders dan in IPv4, door de hogere protocollagen telkens opnieuw ingesteld worden. Velden voor de bron- en bestemmingsadressen zijn uiteraard 16 bytes lang.
Behalve het *Next Header* veld is ook het 20 bits lange *Flow Label* veld nieuw. Sommige realtime toepassingen, zoals audio en video zijn voor een goede werking afhankelijk van verkeerscontrole. Zij moeten met een betrouwbare en regelmatige snelheid op hun bestemming aankomen, dus zonder latentie of *jitter*. Daarom wordt in IPv6 van routers verwacht dat ze voor specifieke *flows*, geïdentificeerd door het *Flow* Label veld, in een cache informatie bijhouden die voor elk datagram in een *flow* constante blijft, zonder telkens volledige *header* opnieuw te moeten verwerken. Hierdoor kunnen datagrammen in een *flow* sneller afgehandeld worden dan andere. Een andere toepassing van *flows* is het toewijzen van bepaald verkeer, zoals e-mail, aan goedkopere verbindingen, zodat de duurdere meer beschikbaar blijven. Een *flow* is specifiek voor een bepaalde bron en eindbestemming.

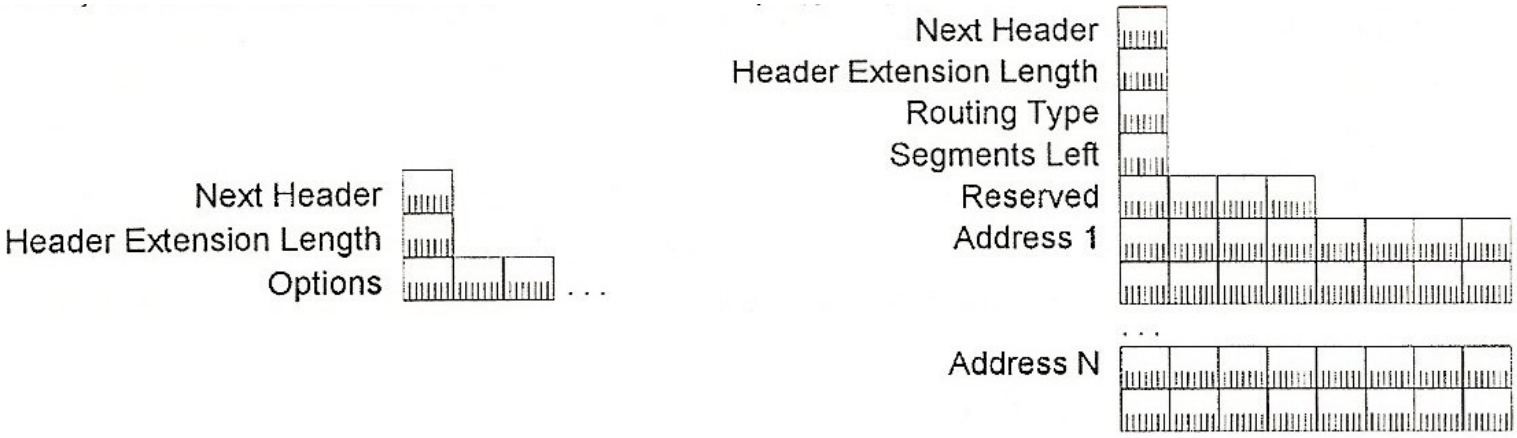
C5 b)

Alle *extensieheaders* (behalve ESP) hebben dezelfde basisstructuur: 2 bytes die respectievelijk het type en de grootte van de *header* aangeven, gevolgd door extensie specifieke gegevens. Volgende bestaan: *Hop-by-Hop Options header, Routing header, Fragment header, Authentication header, Encapsulating Security Payload header en Destination Options header.*

Hop-by-Hop Options header: verzamelt alle opties die ook door tussenliggende routers op weg naar de eindbestemming moeten worden verwerkt? Deze extensie *header* moet altijd onmiddellijk na de IPv6 *header* komen. Net zoals in IPv4 is deze steeds gecodeerd als een *Type-Length-Value* (TLV) triplet. De *type* en *length* velden hebben een vaste grootte van 1 byte. De 2 hoogste bits van het type veld geven aan wat er moet gebeuren als het knooppunt de optie niet begrijpt. De inhoud en lengte van het *value* veld is optieafhankelijk. Opvulopties vullen de *extensieheader* eventueel op met 0-bytes tot een grens van een veelvoud van 8 bytes. Momenteel zijn er slechts 2 nuttige opties:

- De *Jumbo Payloads* maakt het afhandelen mogelijk van datagrammen groter dan 65535 bytes, *jumbogrammen* genoemd. Het 4 byte lange *value* veld van deze optie vervangt het 2 byte lange *Payload Length* veld in de IPv6 *header*.

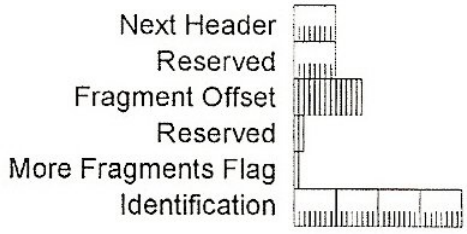
- De *Router Alert* optie licht routers in dat er in het datagebied van het datagram informatie staat die ook door de router zelf moet verwerkt worden.

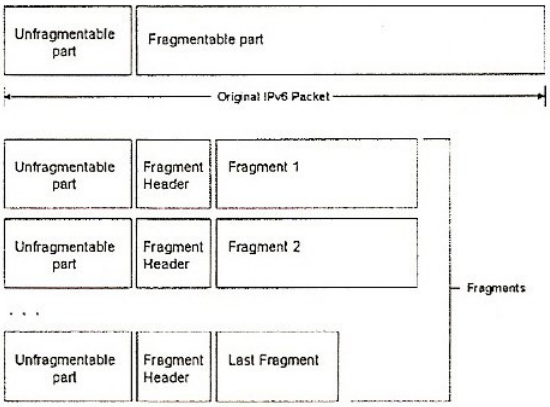


Routing header (RH): kan de route specificeren die het pakket moet afleggen: de RH bevat een lijst adressen die op het pad van het datagram moeten worden bezocht. Als deze extensie gebruikt wordt, dan kan het doeladres in de *header* 1 van de tussenliggende knooppunten zijn in plaats van de uiteindelijke eindbestemming. Indien het *Routing Type* 0 is, dan is het initiële bestemmingsadres in de *header* het 1ste adres in de lijst van de RH. Als dit knooppunt het datagram ontvangt dan stuurt die het bericht door naar het 2de adres. Dit proces gaat door tot het datagram de uiteindelijke bestemming bereikt. Het *Segment Left* veld in de RH geeft aan hoeveel van de routers in de RH nog bezocht moeten worden vooraleer de eindbestemming bereikt wordt. Routing Type 0 implementeert *loose source routing*: tussenliggend routers tussen routers in de RH zijn toegestaan.

Fragment header (FH): Dit gebeurt alleen door de afzender. Het bestemmingsknooppunt gebruikt de FH om het oorspronkelijke datagram te reconstrueren. Tussenliggende routers kunnen het datagram niet meer splitsen, en hoeven zich dan ook niet te bekommeren om de FH. De FH bevat dezelfde velden als deze in de IPv4 *header* voor fragmentatie ingevoerd zijn. Wel is het *Identification* veld 2 keer zo groot. Door de beperkte lengte van het Fragment Offset veld kunnen jumbogrammen niet gefragmenteerd worden. Slechts een deel van het

IPv6 datagram kan gefragmenteerd worden: zowel de *header* als de *Hop-by-Hop Options* en RH extensies moeten in elk fragment herhaald worden.





Authentication header (AH): is de 1ste van 2 beveiligingsextensies die het IPsec mechanisme implementeren. De AH laat authenticatie van inhoud van het datagram toe. Op de zender berekenen functies voor *message digest* (MD5) op basis van het datagram een veilige en betrouwbare digitale handtekening, en voegen het resultaat toe in de AH. De ontvanger berekent de MD5 van het ontvangen pakket opnieuw en vergelijkt deze met de waarde in de AH. Als de 2 waarden identiek zijn, dan kan de ontvanger er zeker van zijn dat het datagram tijdens het transport niet gewijzigd werd.

Encapsulating Security Payload header (ESP): staat knooppunten toe met elkaar te communiceren via datagrammen waarvan de inhoud geëncrypteerd is. Tenzij de ESP zelf voor authenticatie zorgt, wordt aangeraden de ESP steeds in combinatie met de AH *header* te gebruiken. De ESP is altijd de laatste, niet-versleutelde *header* van een datagram. De ESP geeft de bestemming voldoende informatie om dit te kunnen ontcijferen. Net als AH kan ESP op 2 manieren gebruikt worden. In de *transparante* of *transport mode* wordt enkel de ingekapselde data van het pakket versleuteld, terwijl *headers* en extensies ongecodeerde getransportteerd worden. Onderscheppers kunnen zo ondermeer de bron en eindbestemming identificere en andere informatie van het datagram achterhalen.
In de *tunnel mode* wordt het volledige datagram versleuteld en dan in een ander datagram ingepakt, door een knooppunt dat als *security gateway* werkt. Een *security gateway* aan de andere kant van de tunnel pakt de ingekapselde datagrammen uit, en stuurt ze door naar de eindbestemming. Op deze manier worden Virtual Private Networks (VPN's) gecreëerd waardoor ondernemingen het Internet als hun eigen private backbone kunnen gebruiken, zonder openbaring van informatie.

Destination Options header: verzamelt opties die enkel door de eindbestemming of door 1 van de expliciet vermelde routers in de RH, moet worden verwerkt. Deze opties hebben hetzelfde formaat als in de *Hop-by-Hop Options* header.

C6.

Tunneling (subsecties §5.7)

a. Wat is *IPv6 over IPv4 tunneling*, en *waarvoor* zal men deze techniek aanwenden ?
b. Bespreek in detail de diverse, ook de meest recente, *tunnel mechanismen* waarop men een beroep kan doen. Bespreek onder andere hun relatieve voor- en nadelen, en de gebruikte addresseringsschema's.

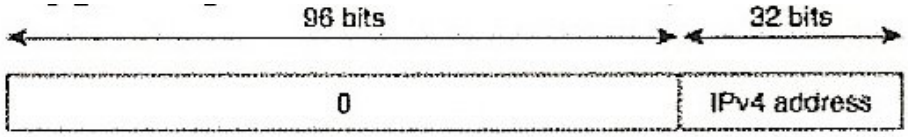
C6 a)

IPv6 over IPv4 *tunneling* kapselt IPv6 datagrammen, die van IPv6 eilanden door IPv4 oceanen verzonden moeten worden, in IPv4 datagrammen in, net alsof het data van een hogere protocollaag zijn. Het Protocol vel van de IPv4 *header* wordt ingevuld met de waarde 41. Het IPv6/IPv4 eindpunt van de tunnel verwijdert de IPv4 *header* en verwerkt de IPv6 *header*, ondermeer om de uiteindelijke IPv6 bestemming te achterhalen.
Tunneling is nodig voor de overgang van IPv4 naar IPv6 te bewerkstelligen.
Deze techniek is essentieel voor de 1ste fasen van de overschakelingsperiode, omdat het een *end-to-end* IPv6 dienst kan verzekeren, zonder een essentiële upgrade van de IPv4 infrastructuur, en zonder de werking van IPv4 diensten te verstoren. Ook in de latere fasen zal *tunneling* voor de connectiviteit blijven zorgen doorheen backbones die alleen IPv4 kunnen verwerken. *Tunneling* vereist dat de beide eindpunten van de tunnel *dual-stack* zijn. Tunnels moeten niet beschouwt worden als permanente verbindingen: ze bestaan enkel indien een specifieke transactie communicatie tussen de eindpunten vereist.

C6 b)

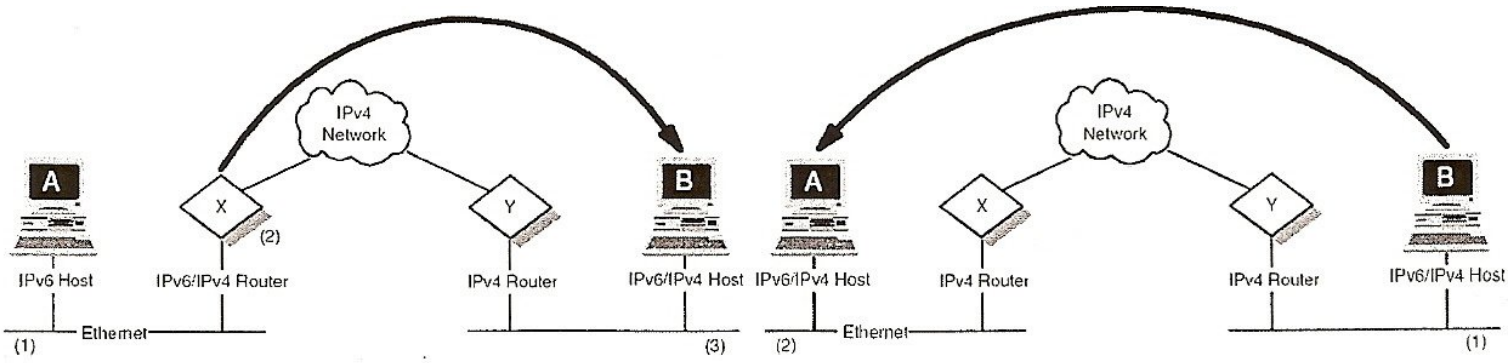
1. automatische IPv6 over IPv4 tunneling:

Automatisch *tunneling* laat IPv6 knooppunten toe om dynamisch, afhankelijk van de eindbestemming, tunnels te creëren doorheen een bestaande IPv4 netwerkinfrastructuur. Automatische tunnels kunnen enkel gecreëerd worden voor IPv4-compatible doeladressen. Deze worden bekomen door een 32-bit IPv4 adres vooral te laten gaan door een prefix van 96 0-bits, en zijn bijgevolg van de vorm ::w.x.y.z .



Of een IPv6/IPv4 bron van automatische *tunneling* gebruik maakt, hangt af van het adres en van de locatie van de eindbestemming, waarbij achtereenvolgens volgende criteria gehanteerd worden:

- Indien het doeladres IPv4 is, dan wordt van IPv4 gebruik gemaakt.
- Indien de eindbestemming zich op hetzelfde subnetwerk bevindt, dan wordt IPv6 gebruikt.
- Indien er een route is naar de eindbestemming, met een IPv6 router als 1ste hop, dan wordt het bericht aan deze router afgeleverd. De router past vervolgens dezelfde criteria toe, om ast te stellen of hijzelf het bericht moet afleveren, doorsturen aan een andere router, of automatisch tunnelen. Indien een IPv6 router het bericht tunnelt, dan noemt men dit *router-to-host* automatische *tunneling*.



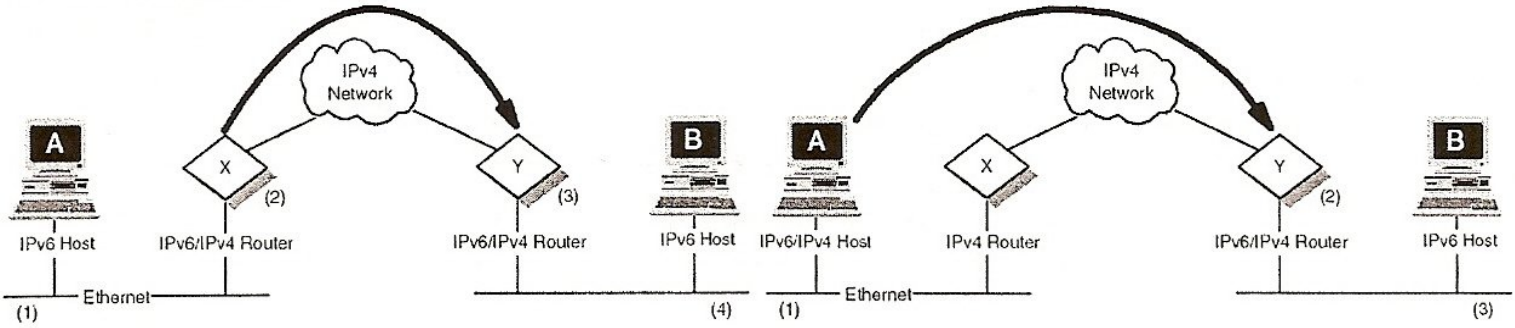
- Enkel indien de vorige 3 criteria niet vervuld zijn, èn het doeladres is bovendien IPv4-compatibel, dan wordt automatische tunneling toegepast.

Nadelen:
Het subnetwerk van de eindbestemming moet IPv4 compatibel zijn.
Automatische *tunneling* kan geen gebruik maken van 128-bit adresruimte en blijft beperkt tot kleine netwerkomgevingen.

2. Geconfigureerde IPv6 naar IPv4 tunneling:

Geconfigureerde tunnels worden manueel ingesteld, en gelden slechts in 1 enkel richting. Men kan uiteraard de configuratie in beide richtingen uitvoeren. Aan het knooppunt of de router aan het begin van de tunnel wordt een *pseudo-interface* geassocieerd. Deze tunnelinterface moet een IPv6 adres krijgen, en kan net als andere interfaces in de routingtabel van het toestel opgenomen worden. Zowel het beginpunt als het eindpunt van de tunnel moeten manueel geconfigureerd worden met een IPv4 adres.

Een enkele geconfigureerde tunnel, met een IPv6/IPv4 router als eindpunt, kan voor connectiviteit zorgen met een volledig IPv6 internetwerk: de router injecteert het ingekapselde IPv6 datagram opnieuw in zijn IPv6 stack, waarna het kan doorgestuurd worden naar om het even welke eindbestemming van het IPv6 internetwerk. Op zijn weg naar de eindbestemming kan het datagram hierbij zowel gerout als opnieuw getunneld worden. Een pad tussen 2 IPv6 knooppunten kan bijgevolg uit meerdere geconfigureerde, maar uit slechts 1 enkele automatische tunnel bestaan.

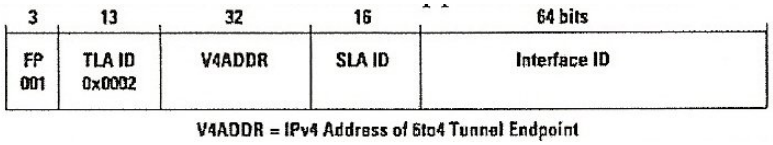


Indien men het beginpunt van de tunnel op een border router van een IPv6 internetwerk configureert, en in diens routingtabel exact definieert welke routes via de tunnelinterface bereikt kunnen worden, dan bekomt men een *router-to-router* geconfigureerde tunnel configuratie. Men kan eventueel ook een *standalone* IPv6 knooppunt als beginpunt instellen, met een *host-to-router* geconfigureerde tunnel als resultaat. Voor elk koppel IPv6 sites, dat men met elkaar wil laten communiceren over een IPv4 infrastructuur moet men minstens 1 *router-to-router* tunnel configureren.

Voordelen:
Het subnetwerk van de eindbestemming moet niet IPv4 compatibel zijn.
Nadelen:
Manuele configuratie.

3. 6to4 tunneling:

6to4 tunneling biedt een eenvoudige oplossing voor de interconnectie van IPv6 eilanden behorend tot verschillende sites. *6to4* steunt volledig op de voorwaarde dat aan elk eiland een unieke fractie van de publieke IPv4 adresruimte toegekend kan worden. Daarom kan 6to4 niet steeds toegepast worden voor *tunneling* tussen subnetten van eenzelfde private netwerkinfrastuctuur.
Het *6to4* mechanisme is een bijzondere vorm van geconfigureerde IPv6 over IPv4 *tunneling*, waarbij een specifieke adresseringsschema van de IPv6 knooppunten ervoor zorgt dat de *router-to-router* tunnels automatisch en dynamisch geconfigureerd worden. Het IPv4 adres van het eindpunt van de tunnel kan immers eenduidig uit de bijzondere vorm van de IPv6 bestemmingsadressen afgeleid worden. Elke organisatie die over een publieke IPv4 adresruimte beschikt, w.x.y.z/p, met om het even welke prefixlengte p, kan 6to4 implementeren. IPv6 knooppunten die met deze adressen geconfigureerd worden, noemt men 6to4 knooppunten. *6to4* knooppunten beschikken over de volledige IPv6 functionaliteit. Het SLA segment is bijv voor de organisatie volledig beschikbaar om de adressering zoveel mogelijk af te stemmen op de topologie van het private internetwerk. *6to4* knooppunten hoeven niet *dual-stack* te zijn.



V4ADDR = IPv4 Address of 6to4 Tunnel Endpoint

De *6to4* router van een organisatie is een *dual-stack* border router, met een publieke IPv4 adres voor de tunnelinterface, die de toegewezen 6to4 adresruimte in het Internet injecteert. Dezelfde 6to4 router adverteert de globale 6to4 adresruimte, 2002::/16, naar het intranet van de organisatie toe.
Om verkeer mogelijk te maken tussen *6to4* knooppunten en andere IPv6 knooppunten moeten in het Internet *6to4 relay routers* geconfigureerd worden. Een *6to4 relay router* is een IPv6/IPv4 *dual-stack* router die zowel over een *6to4* adres als over een regulier IPv6 adres beschikt.

Voordelen:
Het 6to4 principe is door zijn eenvoud waarschijnlijk de belangrijkste techniek die bij de overschakeling naar IPv6 zal gebruikt worden.
Nadelen:
Zo is het aan elke organisatie toegewezen *6to4* blok beperkt in grootte, en zal na de globale migratie moeten hernummert worden naar een regulier IPv6 adresseringsschema. *6to4* kan niet gecombineerd worden met NAT. Zolang massaal van 6to4 gebruik gemaakt wordt, blijven de routingtabellen van de *default-free routers* hun huidige grootte behouden.

4. ISATAP tunneling:

Het *Intra-Site Automatic Tunnel Addressing Protocol* (ISATAP) is zeer gelijkaardig aan 6to4 *tunneling* en laat eveneens toe om IPv6 over IPv4 tunnels automatisch te configureren, ook binnen dezelfde site. In tegenstelling to *6to4* wordt het IPv4 adres van het eindpunt van de tunnel niet afgeleid uit de prefix, maar uit de laagste 32 bits van een ISATAP adres. ISATAP adressen zijn samengesteld uit een willekeurige globale unicast prefix(64-bit) en een *interface-id* van de vorm 0:5EFE:w.x.y.z, waarbij w.x.y.z een willekeurig IPv4 adres is dat aan de interface is toegekend. ISATAP knooppunten zijn steeds *dual-stack.* 2 knooppunten van dezelfde site, maar behorend tot verschillende IPv6 eilanden, kunnen IPv6 datagrammen *host-to-host* naar elkaar tunnelen, indien ze gebruik maken van ISATAP adressen op basis van de *linklokale* prefix FE80::/64

Voordelen:
ISATAP technieken kun je gemakkelijk combineren met *6to4*.