



S u b n e t t e n

t h e t r u e s t o r y

1100 0000 . 1010 1000 . 0000 0001 . 0000 0001
1111 1111 . 1111 1111 . 1111 1111 . 0000 0000

Door: Jeroen Claes

Inhoud

Wat is een subnet?	2
Waarom?	3
Het begin....	3
Een voorbeeld	3
Het nut van subnetmaskers.....	4
Wat we nog moeten weten voor het echte werk begint....De classes.....	5
Een voorbeeld	5
Oefeningen.	5
ClassFul -- ClassLess ?	5
En het echte werk begint.....	6
Subnetten op het aantal hosts.	6
Oefeningen	9
Subnetten op het aantal Subnetten.	10
Oefeningen	14
VLSM.....	14
<i>Waarom VLSM?</i>	15
<i>Hoe werk VLSM?</i>	15
Stap 1	15
Stap 2	16
Stap 3	16
Stap 4.....	17
Oefeningen	20
Opmerkingen.....	23

Wat is een subnet?

Een subnet is een deelnetwerk in een netwerk, dat werkt volgens het Internet Protocol. Het werkwoord subnetten betekent het opdelen van een set opeenvolgende IP-adressen (een IP-range) voor adressering op gescheiden fysieke netwerken. Het subnetten vindt zijn uitdrukking in het zogenaamde subnetmasker. Met dit masker wordt door de netwerkbeheerder bepaald welk gedeelte van een IP-adres het netwerk-ID is, en welk deel het host-ID. Door de subnetmaskers weten de routers in het netwerk op welk fysiek segment zich een bepaald adres bevindt, en dus waarheen de gegevens voor dat bepaalde adres moeten worden gerouteerd.

Waarom?

- Beveiliging: hosts op een ander subnet zijn alleen via een router of firewall te bereiken. Deze router kan daardoor de verbindingen analyseren, filteren en loggen.
- Bandbreedte: alle hosts op een fysiek subnet delen de beschikbare bandbreedte van het medium. Door een aantal hosts te 'verplaatsen' naar een ander subnet kan de op beide subnets de gemiddeld beschikbare bandbreedte meer evenredig worden verdeeld.

In een grote organisatie zou de hoeveelheid gegevens die wordt verstuurd onbeheersbaar worden als elke computer de door elke andere computer verstuurde gegevens via broadcast zou ontvangen. Via subnetten kan de aan een organisatie toegewezen set van opeenvolgende IP-adressen

Het begin....

Elk IP-adres in een subnet start in de binaire notatie met hetzelfde binaire getal. Dit getal geeft het netwerk-ID weer. Soms wordt dit ook wel aangeduid als netwerkadres of subnet-ID. Omdat de term netwerkadres ook vaak wordt gebruikt om het volledige IP-adres aan te duiden, wordt hier verder steeds de term netwerk-ID gebruikt.

De overblijvende bits in het IP-adres vormen het host-ID of hostadres.

Om van een IP-adres te weten welk deel behoort tot het netwerk-ID en welk deel tot het host-ID maakt men gebruik van een subnetmasker. In binaire notatie bestaat een subnetmasker uit een rij enen, één voor elke bit van het netwerk-ID deel van het IP-adres, gevolgd door een rij nullen, één voor elke bit van het host-ID deel van het IP-adres.

Een subnet wordt geïdentificeerd aan de hand van zijn algemeen IP-adres samen met zijn subnetmasker. Een alternatieve methode voor het subnetmasker is de CIDR notatie, waarbij men het getal gebruikt dat het aantal bits van het netwerk-ID aangeeft.

Bv.: 192.168.1.1 255.255.255.0 kan je afkorten naar 192.168.1.1 /24

Een voorbeeld

Een voorbeeld: stel een netwerkje bestaat uit 2 computers A en B, welke zijn verbonden met een router, die op zijn beurt via een kabelmodem gekoppeld is aan het Internet. Dit netwerkje is opgezet

als een subnet, waarbij IP-adres 192.168.1.10 toegewezen is aan computer A en 192.168.1.11 aan computer B. Het subnet werd zo ingesteld op de router dat de eerste 24 bits, oftewel de eerste 3 bytes, het netwerk-ID vormen: 192.168.1.

Dit wordt aangeduid via het subnetmasker 255.255.255.0 (binaire notatie: 11111111 11111111 11111111 00000000). De router kan vervolgens zien of een te contacteren IP-adres binnen of buiten het subnet ligt.

Het algemene adres waarmee we bij dit voorbeeld het subnetwerk aanduiden is: 192.168.1.0 Als we, in plaats van het opgeven van het subnetmasker, gebruikmaken van de CIDR notatie wordt het: 192.168.1.0/24, waarbij de 24 in deze notatie erop duidt dat de eerste 24 bits het netwerk-ID vormen en de overblijvende 8 bits het host-ID.

Het nut van subnetmaskers...

Subnetmaskers worden vaak voorgesteld in decimale vorm met punten tussen de bytes. Het gebruik van subnetmaskers is echter duidelijker als we de binaire notatie gebruiken. Om een IP-adres op te splitsen in een netwerk-ID en een host-ID gebruiken apparaten de bitsgewijze binaire AND operatie.

Bijvoorbeeld IP 192.168.1.1 SM 255.255.255.0 zegt ons niet gek veel buiten dat dit een IP adres en een subnetmask zijn, echter als we dit in binair gaan bekijken kunnen we er al iets meer uitafliden.

```
192.168.1.1
1100 0000 . 1010 1000 . 0000 0001 . 0000 0001
255.255.255.0
1111 1111 . 1111 1111 . 1111 1111 . 0000 0000
```

Als we deze netjes boven elkaar schrijven kunnen we meteen het netwerk en het host adres van elkaar gaan onderscheiden.

Vb1:

```
IP:   1100 0000 . 1010 1000 . 0000 0001 . 0000 0001
SM:   1111 1111 . 1111 1111 . 1111 1111 . 0000 0000
```

Vb2:

```
IP:   1100 0000 . 1010 1000 . 0000 0101 . 0100 1010
SM:   1111 1111 . 1111 1111 . 1111 1111 . 1100 0000
```

Subnetmaskers bestaan net als IP-adressen uit 32 bits: een reeks enen gevolgd door een reeks nullen, waarbij de reeks enen het netwerk-ID deel, en de reeks nullen het host-ID deel aanduidt. Zoals het voorbeeld 2 laat zien, hoeft het subnetmasker de splitsing tussen netwerk-ID en host-ID niet per hele byte te maken: het kan op bit niveau gebeuren. Een classful netwerk is een netwerk met een subnetmasker 255.0.0.0, 255.255.0.0 of 255.255.255.0 in overeenstemming met de netwerkklassie van het IP-adres, waarbij dus wel een verdeling gemaakt wordt op hele bytes.

Wat we nog moeten weten voor het echte werk begint....De classes

Er zijn meerdere IP klassen die we moeten kennen en waar we soms rekening mee dienen te houden, op deze reden komen we in een later deel nog op terug.

In principe bestaat een IP-adres in IPv4 uit een netwerkgedeelte, gevolgd door een hostgedeelte. Het netwerkgedeelte geeft aan welk netwerk bedoeld is en het hostgedeelte geeft de host (bv. een pc of een router) aan binnen het netwerk. Voor een 'klasse A'-netwerk geldt dat de eerste 8 bits netwerkbits zijn, en de overige 24 zijn hostbits. In een 'klasse B'-netwerk bestaat het IP-adres uit 16 netwerkbits gevolgd door 16 hostbits. Een adres in een 'klasse C'-netwerk bestaat uit 24 netwerkbits gevolgd door 8 hostbits. Of we te maken hebben met een klasse A, B of C netwerk is op te maken uit de eerste byte van het IP-adres. Onderstaande tabel geeft steeds een waardebereik binnen de eerste byte, gevolgd door de bijbehorende netwerkklassie.

Class	Eerste bit's	Aantal netwerk bit's	Aantal host bit's	Aantal mogelijke netwerken	Aantal mogelijke host's	Start address	End address
Class A	0000	8	24	128	16,777,216	0.0.0.0	127.255.255.255
Class B	1000	16	16	16,384	65,536	128.0.0.0	191.255.255.255
Class C	1100	24	8	2,097,152	256	192.0.0.0	223.255.255.255
Class D	1110	ND	ND	ND	ND	224.0.0.0	239.255.255.255
Class E	1111	ND	ND	ND	ND	240.0.0.0	255.255.255.255

Een voorbeeld; 192.168.1.1 = **1100** 0000 . 1010 1000 . 0000 0001 . 0000 0001 = Class C
 172.16.230.1 = 1010 1100 . 0001 0000 . 1110 0110 . 0000 0001 = Class B
 127.0.0.0 = 0111 1111 . 0000 0000 . 0000 0000 . 0000 0000 = Class A

Oefeningen.

Probeer van volgende IP adressen de Classes te bepalen.

10.18.20.33		1.0.0.1	
127.1.0.10		8.8.8.8	
178.16.22.10		265.167.30.10	
193.16.15.11		78.16.77.66	

ClassFul -- ClassLess ?

We spreken van Classful IP Subnetting wanneer gebruik gemaakt wordt van de default subnetmask. Sinds 1992 werd Classless IP Subnetting (CIDR = Classless InterDomain Routing) ingevoerd om het gebrek aan IP-adressen te voorkomen.

Vb. Indien je bij classful subnetting iets meer dan 254 hosts nodig had, kreeg je een klasse-B netwerk toegewezen, wat overeenstemt met een adresblok van 65.534 IP-adressen. Dit had een enorme verspilling van IP-adressen voor gevolg. Bij classless IP-adressing wordt het default subnet mask uitgebreid, zodat één groot netwerk wordt opgesplitst in verschillende kleinere subnetwerken.

Voorbeeld:

Een enkel klasse-C netwerk 200.1.1.0 moet opgesplitst worden in twee subnetten.

Deze subnetten zullen met een router gekoppeld worden.

Voordeel : Hierdoor worden kleinere onafhankelijke broadcast domeinen gemaakt (routers geven namelijk geen broadcasts door), waardoor de performantie en de veiligheid verbeteren.

Daar je slechts één klasse-C netwerk hebt, heb je maar 8 bits in het vierde octet beschikbaar voor een totaal van 254 hosts. We moeten een aangepast subnet mask maken, waarbij we een aantal bits gaan lenen in het host deel van het adres.

Hieronder een voorbeeld.

We hebben twee klas lokalen en we willen het verkeer tussen deze lokalen filteren, we gaan deze dus verdelen in twee subnets.

Netwerk adres (1) 192.168.1.0 /25
(2) 192.168.1.128 /25

Zoals je allicht opgevallen is hebben we nu als SM nu 255.255.255.128 terwijl dit normaal 255.255.255.0 is voor een klasse C netwerk.

En het echte werk begint....

We gaan de drie meest voorkomende methodes toepassen, namelijk subnetten naar het aantal hosts die we nodig hebben, subnetten naar het aantal subnetten die we nodig hebben en VLSM of Variable Length Subnet Masking de meest optimale manier, deze komt in deel twee van dit boek.

Subnetten op het aantal hosts.

Maar we beginnen uiteraard bij het begin, met subnetten naar het aantal benodigde hosts .

Opdracht we hebben een school als klant met 5 computer lokalen van ieder 16 pc's, wij moeten deze netwerken gaan splitsen in subnet's . het gebruikte ip is 192.168.1.0/24

Subnetten kan in kortweg een 4 tal stappen...

Stap 1

We zetten ons IP en onze SM om naar binair en we trekken een streep achter het netwerk gedeelte, dit zou er moeten uitzien als onderstaand voorbeeld...

IP: 1100 0000 . 1010 1000 . 0000 0001 . 0000 0000

SM: 1111 1111 . 1111 1111 . 1111 1111 . 0000 0000

Stap 2

Het bepalen van het aantal benodigde bits om onze 16 hosts kunnen vormen, hierbij moeten we steeds 2 adressen reserveren voor ons netwerkadres en ons broadcastadres die we uiteraard niet voor onze hosts kunnen gebruiken. Bv we hebben 16 hosts nodig dus we houden rekening met 18 adressen.

Hoe berekenen we het aantal bits dat we nodig hebben? We gaan kijken naar onze macht van twee.

We nemen steeds het getal waar binnen onze behoefte valt, toch ook niet nodeloos groter.

In ons geval past onze 18 binnen 32 en $32 = 2^5$.

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1
18	18	18	18	18	18	18	18	18

Zoals we uit onze tabel kunnen afleiden hebben we dus 5 bits nodig.

We trekken nog een streep na onze 5 bits, we tellen vanaf achter.

Als we dit op onze oefening toepassen krijgen we dit;

IP: 1100 0000 . 1010 1000 . 0000 0001 . 000 0 0000
SM: 1111 1111 . 1111 1111 . 1111 1111 . 000 0 0000

Als je nu naar bovenstaande uitwerking kijkt zie je dat je 5 bits (Groen) hebt om hosts mee te vormen en 3 bits (Blauw) om 8 subnets te creëren.

Stap 3

In deze stap gaan we eerst en vooral ons subnet mask aanpassen aan de grote van ons nieuw netwerk gedeelte.

IP: 1100 0000 . 1010 1000 . 0000 0001 . 000 0 0000
SM: 1111 1111 . 1111 1111 . 1111 1111 . 111 0 0000

Het subnet mask of SM genaamd blijft na deze bewerking steeds hetzelfde.

SM: 1111 1111 . 1111 1111 . 1111 1111 . 111 0 0000 in decimaal : 255.255.255.224

Als dit gebeurt is kunnen we ons eerste subnet gaan berekenen.

Het eerste wat we steeds gaan zoeken is ons netwerkadres, hierbij staan steeds alle host bits op nul.

In ons geval is dit dus,

NA: 1100 0000 . 1010 1000 . 0000 0001 . 000 0 0000 in decimaal: 192.168.1.0

Het tweede dat we zoeken is ons laagst mogelijke host adres hierna afgekort tot LH.

Ons LH is steeds ons netwerkadres +1, dit word duidelijk in onderstaand voorbeeld.

LH: 1100 0000 . 1010 1000 . 0000 0001 . 000 0 0001 in decimaal: 192.168.1.1

Hierna gaan we opzoek naar ons hoogst mogelijke host hierna kortweg HH, dit is alle host bits op 1 zetten behalve de laatste bit, die blijft gewoon op nul staan.

HH: 1100 0000 . 1010 1000 . 0000 0001 . 000 1 1110 in decimaal: 192.168.1.30

Als laatste moeten we enkel ons broadcast adres of BA nog zien te vinden, dit is alle host bits op 1.

BA: 1100 0000 . 1010 1000 . 0000 0001 . 000 1 1111 in decimaal: 192.168.1.31

Als we dit netjes op een rijtje zetten krijg je dit,

NA	: 192.168.1.0
LH	: 192.168.1.1
HH	: 192.168.1.30
BA	: 192.168.1.31
SM	: 255.255.255.224

Stap 4

Herhaal stap 3 zo vaak tot je voldoende subneten hebt, om je volgend subnet te berekenen tel je bij je netwerk gedeelte steeds 1 bij. BV: 1100 0000 . 1010 1000 . 0000 0001 . 001 0 0000 is ons volgende netwerk adres.

We door lopen nog even de berekening voor het volgende subnet.

Het subnet mask of SM genaamd blijft na deze bewerking steeds hetzelfde.

SM: 1111 1111 . 1111 1111 . 1111 1111 . 111 0 0000 in decimaal : 255.255.255.224

Om ons nieuwe netwerk adres te berekenen tellen we bij het netwerkgedeelte een beetje bij.

NA: 1100 0000 . 1010 1000 . 0000 0001 . 001 0 0000 in decimaal: 192.168.1.32

Ons LH....

LH: 1100 0000 . 1010 1000 . 0000 0001 . 001 0 0001 in decimaal: 192.168.1.33

Onze HH...

HH: 1100 0000 . 1010 1000 . 0000 0001 . 001 1 1110 in decimaal: 192.168.1.62

En tenslotte onze BA.

HH: 1100 0000 . 1010 1000 . 0000 0001 . 001 1 1111 in decimaal: 192.168.1.63

Als we dit netjes op een rijtje zetten krijg je dit,

NA	: 192.168.1.32
LH	: 192.168.1.33
HH	: 192.168.1.62
BA	: 192.168.1.63
SM	: 255.255.255.224

Vul nu zelf de volgende subnetten verder aan.

NA:

LH:

HH:

BA:

SM

NA:

LH:

HH:

BA:

SM

NA:

LH:

HH:

BA:

SM

Oefeningen

Maak de onderstaande oefeningen.

1. Het te gebruiken ip adres is 172.16.0.0 en we hebben 200 hosts nodig in 10 subnetten.
2. We hebben 142 hosts nodig is 30 subnetten, kies hiervoor zelf welk ip je nodig hebt om te beginnen.

Subnetten op het aantal Subnetten.

Dit is in wezen bijna identiek aan subnetten op het aantal hosts, enkel ga je hier het aantal benodigde netwerken bereken ipv het aantal benodigde hosts, toch dien je ook nog steeds er rekening mee te houden dat je een voldoende aantal host adressen overhoud.

Opdracht we hebben weer een school als klant met 10 computer lokalen van ieder 50 pc's, wij moeten deze netwerken gaan splitsen in subnet's . het gebruikte ip is 172.16.0.0/16

Dit kan ook weer in een 4 tal stappen...

Stap 1

We zetten ons IP en onze SM om naar binair en we trekken een streep achter het netwerk gedeelte, dit zou er moeten uitzien als onderstaand voorbeeld...

```
IP:   1010 1100 . 0001 0000 | 0000 0000 . 0000 0000
SM:   1111 1111 . 1111 1111 | 0000 0000 . 0000 0000
```

Stap 2

Het bepalen van het aantal benodigde bits om onze 10 netwerkjes kunnen vormen.

Hoe berekenen we het aantal bits dat we nodig hebben? We gaan kijken naar onze macht van twee.

We nemen steeds het getal waar binnen onze behoefte valt, toch ook niet nodeloos groter.
In ons geval past onze 10 binnen 16 en $16 = 2^4$.

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1
10	10	10	10	10	10	10	10	10

Zoals we uit onze tabel kunnen afleiden hebben we dus 4 bits nodig.

We trekken nog een streep na onze 4 bits, te tellen vanaf het einde van het netwerkgedeelte .
Als we dit op onze oefening toepassen krijgen we dit;

```
IP:   1010 1100 . 0001 0000 | 0000 0000 . 0000 0000
SM:   1111 1111 . 1111 1111 | 0000 0000 . 0000 0000
```

Als je nu naar bovenstaande uitwerking kijkt zie je dat je 12 bits (Groen) hebt om hosts mee te vormen en 4 bits (Blauw) om 10 subnets te creëren.

Stap 3

In deze stap gaan we eerst en vooral ons subnet mask aanpassen aan de grote van ons nieuw netwerk gedeelte.

IP: 1010 1100 . 0001 0000 . 0000 0000 . 0000 0000
SM: 1111 1111 . 1111 1111 . 1111 0000 . 0000 0000

Het subnet mask of SM genaamd blijft na deze bewerking steeds hetzelfde.

SM: 1111 1111 . 1111 1111 . 1111 0000 . 0000 0000 in decimaal : 255.255.240.0

Als dit gebeurt is kunnen we ons eerste subnet gaan berekenen.
Het eerste wat we steeds gaan zoeken is ons netwerkadres, hierbij staan steeds alle host bits op nul.

In ons geval is dit dus,

NA: 1010 1100 . 0001 0000 . 0000 0000 . 0000 0000 in decimaal: 172.16.0.0

Het tweede dat we zoeken is ons laagst mogelijke host adres hierna afgekort tot LH.
Ons LH is steeds ons netwerkadres +1, dit word duidelijk in onderstaand voorbeeld.

LH: 1010 1100 . 0001 0000 . 0000 0000 . 0000 0001 in decimaal: 172.16.0.1

Hierna gaan we opzoek naar ons hoogst mogelijke host hierna kortweg HH, dit is alle host bits op 1
zetten behalve de laatste bit, die blijft gewoon op nul staan.

HH: 1010 1100 . 0001 0000 . 0000 1111 . 1111 1110 in decimaal: 172.16.15.254

Als laatste moeten we enkel ons broadcast adres of BA nog zien te vinden, dit is alle host bits op 1.

BA: 1010 1100 . 0001 0000 . 0000 1111 . 1111 1111 in decimaal: 172.16.15.255

Als we dit netjes op een rijtje zetten krijg je dit,

NA	: 172.16.0.0
LH	: 172.16.0.1
HH	: 172.16.15.254
BA	: 172.16.15.255
SM	: 255.255.240.0

Stap 4

Herhaal stap 3 zo vaak tot je voldoende subnetten hebt, om je volgend subnet te berekenen tel je bij je netwerk gedeelte steeds 1 bij. BV: 1010 1100 . 0001 0000 | 0001 | 0000 . 0000 0000 is ons volgende netwerk adres.

We door lopen nog even de berekening voor het volgende subnet.

Het subnet mask of SM genaamd blijft na deze bewerking steeds hetzelfde.

SM: 1111 1111 . 1111 1111 | 1111 | 0000 . 0000 0000 in decimaal : 255.255.240.0

Om ons nieuwe netwerk adres te berekenen tellen we bij het netwerkgedeelte een beetje bij.

NA: 1010 1100 . 0001 0000 | 0001 | 0000 . 0000 0000 in decimaal: 172.16.16.0

Ons LH....

LH: 1010 1100 . 0001 0000 | 0001 | 0000 . 0000 0001 in decimaal: 172.16.16.1

Onze HH...

HH: 1010 1100 . 0001 0000 | 0001 | 1111 . 1111 1110 in decimaal: 172.16.31.254

En tenslotte onze BA.

HH: 1010 1100 . 0001 0000 | 0001 | 1111 . 1111 1111 in decimaal: 172.16.31.255

Als we dit netjes op een rijtje zetten krijg je dit,

NA	: 172.16.16.0
LH	: 172.16.16.1
HH	: 172.16.31.254
BA	: 172.16.31.255
SM	: 255.255.240.0

Vul nu zelf de volgende subnetten verder aan.

NA:

LH:

HH:

BA:

SM

NA:

LH:

HH:

BA:

SM

NA:

LH:

HH:

BA:

SM

NA:

LH:

HH:

BA:

SM

NA:

LH:

HH:

BA:

SM

NA:

LH:

HH:

BA:

SM

NA:

LH:

HH:

BA:

SM

NA:
LH:
HH:
BA:
SM

NA:
LH:
HH:
BA:
SM

NA:
LH:
HH:
BA:
SM

Oefeningen

Maak de onderstaande oefeningen.

1. Het te gebruiken ip adres is 172.16.0.0 en we hebben 200 hosts nodig in 10 subnetten.
2. We hebben 142 hosts nodig is 30 subnetten, kies hiervoor zelf welk ip je nodig hebt om te beginnen.

VLSM

VLSM is een uitbreiding op subnetten, in dit deel gaan we het namelijk hebben over Variable-Length Subnet Masking of kort weg VLSM genoemd.

Nu we goed op weg zijn met het traditionele subnetten kunnen we dus een stap verder gaan.

Het deel dat we kennen mogen we uiteraard niet verloren laten gaan, en zullen we ook blijven gebruiken zolang je opleiding loopt.

Waarom VLSM?

Het lijkt misschien raar om nu ineens alles te gaan om gooien en een heel andere manier van subnetten te gaan bekijken en toe te passen.

Maar we zouden dit uiteraard niet doen zonder een deftige reden, "het zo nuttig mogelijk benutten van de beschikbare adressen".

Dit lijkt misschien een rare uitspraak maar we gaan je zo goed mogelijk proberen duidelijk te maken waarom we dit nu net geen doen .

Stel we hebben een classe C netwerk: 192.168.1.0/24 en hebben 1 netwerk van 120 computers nodig en 2 netwerken van 60 computers nodig.

Bij de vorige manieren van subnetten hebben we gezien dat we dan gewoon 3 netwerken van ieder 128 hosts zouden gaan maken.

Maar stel dat we een meer logische manier zouden hebben, een manier die het ons toelaat om niet 3 keer een net werk van 128 hosts te maken maar een manier waar we 1 maal 128 hosts hebben en 2 keer 64 hosts.

Die manier hebben we, VLSM.

Hoe werk VLSM?

VLSM werkt kort weg ook in een **X** aantal stappen, het is niet echt moeilijker dan het "gewone" subnetten, maar het begin vergt wat inzet en voor al telt hier ook weer, oefening baart kunst.

Opdracht:

we hebben een school als klant met 3 computer lokalen een van 120 pc's en twee van 60 PC's, wij moeten deze netwerken gaan splitsen in subnet's . het gebruikte ip is 192.168.1.0/24

Stap 1

De eerste stap is nog steeds identiek aan de eerste stap van het gewone subnetten.

We zetten ons IP en onze SM om naar binair en we trekken een streep achter het netwerk gedeelte, dit zou er moeten uitzien als onderstaand voorbeeld...

IP: 1100 0000 . 1010 1000 . 0000 0001 . 0000 0000
 SM: 1111 1111 . 1111 1111 . 1111 1111 . 0000 0000

Stap 2

Bij VLSM gaan we steeds eerst opzoek naar het grootste subnet dat we moeten vormen, in ons geval is dat dus 120 PC's

Het bepalen van het aantal benodigde bits om onze 120 hosts kunnen vormen, hierbij moeten we steeds 2 adressen reserveren voor ons netwerkadres en ons broadcastadres die we uiteraard niet voor onze hosts kunnen gebruiken. Bv we hebben 120 hosts nodig dus we houden rekening met 122 adressen.

Hoe berekenen we het aantal bits dat we nodig hebben? We gaan kijken naar onze macht van twee.

We nemen steeds het getal waar binnen onze behoefte valt, toch ook niet nodeloos groter.
 In ons geval past onze 122 binnen 128 en $128 = 2^7$.

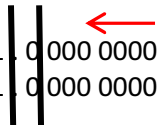
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1
122	<u>122</u>	122	122	122	122	122	122	122

Zoals we uit onze tabel kunnen afleiden hebben we dus 7 bits nodig.

We trekken nog een streep na onze 7 bits, we tellen vanaf achter.

Als we dit op onze oefening toepassen krijgen we dit;

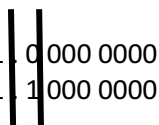
IP: 1100 0000 . 1010 1000 . 0000 0001 . 0 000 0000
 SM: 1111 1111 . 1111 1111 . 1111 1111 . 0 000 0000



Stap 3

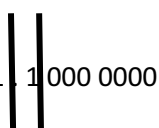
In deze stap gaan we eerst en vooral ons subnet mask aanpassen aan de grote van ons nieuw netwerk gedeelte.

IP: 1100 0000 . 1010 1000 . 0000 0001 . 0 000 0000
 SM: 1111 1111 . 1111 1111 . 1111 1111 . 1 000 0000



In tegen stelling tot het gewone subnetten gaat de lengte van de subnetmask bij VLSM wel groter worden naar mate we in de berekening vorderen.

SM: 1111 1111 . 1111 1111 . 1111 1111 . 1 000 0000 in decimaal : 255.255.255.128



Als dit gebeurt is kunnen we ons eerste subnet gaan berekenen.

Het eerste wat we steeds gaan zoeken is ons netwerkadres, hierbij staan steeds alle host bits op nul.

In ons geval is dit dus,

NA: 1100 0000 . 1010 1000 . 0000 0001 . 0 000 0000 in decimaal: 192.168.1.0

Het tweede dat we zoeken is ons laagst mogelijke host adres hierna afgekort tot LH.
Ons LH is steeds ons netwerkadres +1, dit word duidelijk in onderstaand voorbeeld.

LH: 1100 0000 . 1010 1000 . 0000 0001 . 0 000 0001 in decimaal: 192.168.1.1

Hierna gaan we opzoek naar ons hoogst mogelijke host hierna kortweg HH, dit is alle host bits op 1
zetten behalve de laatste bit, die blijft gewoon op nul staan.

HH: 1100 0000 . 1010 1000 . 0000 0001 . 0 111 1110 in decimaal: 192.168.1.126

Als laatste moeten we enkel ons broadcast adres of BA nog zien te vinden, dit is alle host bits op 1.

BA: 1100 0000 . 1010 1000 . 0000 0001 . 0 111 1111 in decimaal: 192.168.1.127

Als we dit netjes op een rijtje zetten krijg je dit,

NA	: 192.168.1.0
LH	: 192.168.1.1
HH	: 192.168.1.126
BA	: 192.168.1.127
SM	: 255.255.255.128

Stap 4

Stap 4 is misschien wel de meest ingewikkelde stap, maar in feite is dit louter schijn en is dit even simpel als bovenstaande 3 stappen.

Eigenlijk is dit ook exact wat we gaan doen, onze stappen herhalen...

Ons laatste adres was 192.168.1.127 /25 of in binair,

1100 0000 . 1010 1000 . 0000 0001 . 0 111 1111

Voor we hier iets mee kunnen doen gaan we bij dit binaire getal 1 bij tellen om ons volgende subnet te maken dus krijgen we dit,

1100 0000 . 1010 1000 . 0000 0001 . 1 000 0000

Vanaf nu zij de stappen weer identiek aan de stappen van het gewone subnetten.

Dus wat we nu moeten doen is terug gaan kijken hoeveel bits we nodig hebben om ons volgende subnet te gaan berekenen.

We hebben twee subnets van 60 pc's nodig, $60 + 2 = 62$.

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1
62	62	62	62	62	62	62	62	62

Zoals we uit onze tabel kunnen afleiden hebben we nu dus 6 bits nodig.

We trekken nog een streep na onze 6 bits, we tellen vanaf achter.

Als we dit op onze oefening toepassen krijgen we dit;

IP: 1100 0000 . 1010 1000 . 0000 0001 . 1000 0000
 SM: 1111 1111 . 1111 1111 . 1111 1111 . 1000 0000

In deze stap gaan we eerst en vooral ons subnet mask aanpassen aan de grote van ons nieuw netwerk gedeelte.

IP: 1100 0000 . 1010 1000 . 0000 0001 . 1000 0000
 SM: 1111 1111 . 1111 1111 . 1111 1111 . 1100 0000

In tegen stelling tot het gewone subnetten gaat de lengte van de subnetmask bij VLSM wel groter worden naar mate we in de berekening vorderen.

SM: 1111 1111 . 1111 1111 . 1111 1111 . 1100 0000 in decimaal : 255.255.255.192

Als dit gebeurt is kunnen we ons eerste subnet gaan berekenen.

Het eerste wat we steeds gaan zoeken is ons netwerkadres, hierbij staan steeds alle host bits op nul.

In ons geval is dit dus,

NA: 1100 0000 . 1010 1000 . 0000 0001 . 1000 0000 in decimaal: 192.168.1.128

Het tweede dat we zoeken is ons laagst mogelijke host adres hierna afgekort tot LH.

Ons LH is steeds ons netwerkadres +1, dit word duidelijk in onderstaand voorbeeld.

LH: 1100 0000 . 1010 1000 . 0000 0001 . 1000 0001 in decimaal: 192.168.1.129

Hierna gaan we opzoek naar ons hoogst mogelijke host hierna kortweg HH, dit is alle host bits op 1 zetten behalve de laatste bit, die blijft gewoon op nul staan.

HH: 1100 0000 . 1010 1000 . 0000 0001 . 1 0 11 1110 in decimaal: 192.168.1.190

Als laatste moeten we enkel ons broadcast adres of BA nog zien te vinden, dit is alle host bits op 1.

BA: 1100 0000 . 1010 1000 . 0000 0001 . 1 0 11 1111 in decimaal: 192.168.1.191

Als we dit netjes op een rijtje zetten krijg je dit,

NA	: 192.168.1.128
LH	: 192.168.1.129
HH	: 192.168.1.190
BA	: 192.168.1.191
SM	: 255.255.255.192

Om nog duidelijk te maken wat we doen met twee netwerken van de zelfde grote gaan we de laatste ook nog volledig uitwerken.

Dit is ook weer de zelfde werk wijze als bij het gewone subnetten.

Omdat we met de zelfde aantal hosts blijven werken blijft ons subnetmask hetzelfde,

SM: 1111 1111 . 1111 1111 . 1111 1111 . 1 1 00 0000 in decimaal : 255.255.255.192

Om je volgend subnet te berekenen tel je bij je netwerk gedeelte steeds 1 bij. BV: 1100 0000 . 1010 1000 . 0000 0001 . 1100 0000 is ons volgende netwerk adres.

We door lopen nog even de berekening voor het volgende subnet.

NA: 1100 0000 . 1010 1000 . 0000 0001 . 1 1 00 0000 in decimaal: 192.168.1.192

Ons LH....

LH: 1100 0000 . 1010 1000 . 0000 0001 . 1 1 00 0001 in decimaal: 192.168.1.193

Onze HH...

HH: 1100 0000 . 1010 1000 . 0000 0001 . 1 1 11 1110 in decimaal: 192.168.1.254

En tenslotte onze BA.

HH: 1100 0000 . 1010 1000 . 0000 0001 . 1 1 11 1111 in decimaal: 192.168.1.255

Als we dit netjes op een rijtje zetten krijg je dit,

NA	: 192.168.1.192
LH	: 192.168.1.193
HH	: 192.168.1.254
BA	: 192.168.1.255
SM	: 255.255.255.192

En daar mee is onze eerste VLSM oefening klaar.

Oefeningen

Maak de onderstaande oefeningen.

1) Ik heb 4 subnetten van 20 pc's, 2 subnetten van 70 pc's, en 2 subnetten van 252 pc's

IP 150.18.0.0 /16

NA	NA
LH	LH
HH	HH
BA	BA
SM	SM

NA	NA
LH	LH
HH	HH
BA	BA
SM	SM

NA	NA
LH	LH
HH	HH

BA	BA
SM	SM
NA	NA
LH	LH
HH	HH
BA	BA
SM	SM

2) Ik heb 3 subnetten van 30 pc's, 2 subnetten van 125 pc's, en 3 subnetten van 254 pc's

IP 10.18.0.0 /8

NA	NA
LH	LH
HH	HH
BA	BA
SM	SM

NA	NA
LH	LH
HH	HH
BA	BA
SM	SM

NA	NA
LH	LH
HH	HH
BA	BA
SM	SM

NA	NA
LH	LH
HH	HH
BA	BA
SM	SM

3) Ik heb 2 subnetten van 70 pc's, 1 subnetten van 30 pc's, en 5 subnetten van 22 pc's

IP 172.16.0.0 /16

NA	NA
LH	LH
HH	HH
BA	BA
SM	SM

NA	NA
LH	LH
HH	HH
BA	BA
SM	SM

NA	NA
LH	LH
HH	HH
BA	BA
SM	SM

NA	NA
LH	LH
HH	HH
BA	BA
SM	SM

Opmerkingen

Dit boek mag enkel gebruikt worden met de vermelding van de naam van de auteur.

In dit boek kunnen helaas nog steeds enkele fouten staan, voor fouten te rapporteren en opmerkingen kan je steeds mailen naar jeroen@claes-it.be.

Let wel dit is geen boek over de Nederlandse taal !!!