

# VLSM

## Variable Length Subnet Masking.

Johan Futsæther 28.02.07

Bearbeiding av: "VLSM - Chapter 2 of CCNA Portable Command Guide"

Ved å lage subnet, kan vi få oppdelt nettverket vårt i mindre nettverk. Fordelen med dette er at vi kan utnytte det tilgjengelige adresseområdet bedre. Hvis vi ikke kunne subnette, så er eneste muligheten å bruke et rent klasse A, B eller C-nettverk. Det minste nettverket ville da være 254 hosts. Svært få har behov for et slikt nettverk. Dermed ville mange IP-adresser "gå til spille". Subnetting gir også muligheter for å kunne skille de ulike avdelinger fra hverandre i en bedrift. Nettverkstrafikken kan dermed holde seg innenfor et avgrenset område av nettverket.

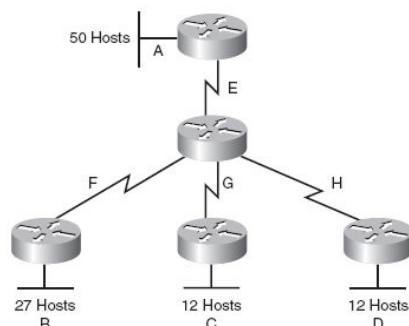
Den "klassiske" måten å lage subnet på, gjør at alle subnet får like mange hosts (H) i hvert subnet. Hvis du for eksempel låner 4 bit i en klasse C-nettverk (NNNH), ender du opp med 16 nettverk (N), med  $16-2=14$  hosts (H) i hvert av de 16 nettverk (N) [under forutsetning at "ip subnet zero" er satt]. Hvis du for eksempel bare har 8 hosts på hver subnet, så har du "mistet"  $14 - 8 = 6$  nettverksadresser for hvert subnet. I dette tilfelle  $6 * 14 = 84$  nettverksadresser. Vi mister ytterligere nettverksadresser hvis det for eksempel er serieforbindelser (serielink) mellom noen av ruterne i nettverket, siden en serieforbindelse bare trenger 2 hosts (H). Dermed mister vi  $14 - 2 = 12$  nettverksadresser i alle subnet hvor det er serieforbindelser mellom ruterne.

Eksempelet ovenfor viser at den klassiske måten å lage subnet på, bare til en viss grad utnytter det tilgjengelige adresseområdet. Det er her VLSM kommer inn. VLSM står for "Variable Length Subnet Masking" og går ut på å lage subnet hvor størrelsen på subnettet er tilpasset antall hosts (H) som skal kobles til subnetnet. Dette klare vi ved å "subnette subnet"

For å kunne benytte VLSM, er det visse betingelser. Rutingprotokollene må kunne bære "extended-nettverk-prefix"-informasjon. Forekesempel støtter ikke RIP-1 to subnettmasker. Videre må alle rutere kunne ha en "Consistent forwarding algoritim" basert på "longest match"

Eksempel på VLSM.

Se på figur 1. Nettverket består av 8 subnet merket A til H. Subnet E, F, G og H er serieforbindelser (serielinker) og trenger derfor bare 2 hostplasser for hvert subnet. Subnet A, B, C og D trenger henholdsvis 50, 27, 12 og 12 hostplasser (H). I dette eksempelet bruker vi en klasse C-nettverk for å lage subnettene. Vi har fått tildelt klasse C-nettverket 192.168.100.0/24.



Figur 1.

For å lage en IP-plan hvor vi ønsker å bruke VLSM, går vi frem på følgende måte:

1. Finn ut hvor mange bit vi må bruke for å kunne adressere det største subnetnet, deretter finne ut hvor mange subnet (N) du kan lage med de gjestående bit. (Se detaljer senere)
2. Velg ett av subnettene som du fant under punkt 1 for det største subnetnet. (Se detaljer senere)
3. Finn det nest største subnetnet. (Se detaljer senere)

4. Finn det tredje største nettverket. (Se detaljer senere)
5. Bestem nettverksnummrene for serieforbindelsene. (Se detaljer senere)

## Punkt 1.

*Finn ut hvor mange bit vi må bruke for å kunne adressere det største subnett, deretter finne ut hvor mange subnet (N) du kan lage med de gjenstående bit.*

Det største subnett er subnet A med 50 hosts (H). For å kunne adressere 50 hosts trenger vi 6 bit (b):

$$2^b - 2 \geq 50$$

$$b = 6, \text{ som gir 60 mulige hostplasser (H).}$$

Som vi ser så trenger vi 6 bit for å adressere 50 hostplasser (H). Siden vi starter med 8 bit, men trenger 6 bit til adressere 50 hosts (H), står vi igjen med  $8 - 6 = 2$  bit til subnetadresser (N).

Slik vil det da se ut:

NNHHHHHH

Den videre subnetting må nå ha dette som utgangspunkt for å tilfredsstille kravet for nettverk A.

## Punkt 2.

*Velg ett av subnettene som du fant under punkt 1 for det største subnett.*

Med utgangspunkt i hva vi har funnet i punkt 1, ser vi at vi har 2 bit for å lage subnet. Dette vil si at vi kan lage  $2^2 = 4$  subnet.

NNHHHHHH

00HHHHHH  
01HHHHHH  
10HHHHHH  
11HHHHHH

For å finne nettverksnummrene til hver subnet, setter vi alle hostbit (H) til = 0.

00000000 = .0  
01000000 = .64  
10000000 = .128  
11000000 = .192

Alle disse nettverk vil ha samme subnetmaske, akkurat som i klassisk subnetting. Vi finner nettverksmasken ved å sette alle nettverksbit inkludert subnetbittene til 1 og hostsbittene til 0.

Da blir subnetmasken:

NNNNNNNN.NNNNNNNN.NNNNNNNN.NNHHHH

11111111.11111111.11111111.11000000 = 255.255.255.192

eller prefiks /26. (Ved å bruke prefiks er det lettere å se de ulike subnetmaskene når vi bruker VLSM).

For nettverk A velger vi ett av de nettverksnummrene vi har funnet. For eksempel .64 –nettverket. Se tabellen nedenfor:

00000000 (Original)	.0	
01000000 (Original)	.64	Nettverk A

1000000 (Original)	.128	
1100000 (Original)	.192	

### Punkt 3.

*Finn det nest største subnettet.*

Nettverk B = 27 hosts (H)

Bi bestemmer hvor mange bit (b) vi trenger for å adressere 27 hosts (H)

$$2^b - 2 \geq 27$$

$$b = 5 \text{ bit}$$

Du trenger 5 bit for å adressere 27 hosts (H)

Du startet med 2 N-bits og 6 H-bit for nettverk A. Dette er fortsatt ditt utgangspunkt.

Plukk ut ett av de igjenstående /26-nettverk du har i tabellen under punkt 2. I dette eksempelet kan vi velge .128/26-subnettet.

1000000

Du trenger bare 5 bit for å kunne adressere 27 hosts. Derfor står du igjen med:

10N00000

hvor

- 10 representerer ditt utgangspunkt
- N representerer et ekstra bit til nettverk.
- 00000 representerer bittene du trenger for å kunne adressere 27 hosts.

På grunn av ditt ekstra N-bit, kan du nå lage to mindre subnet ut fra ett av de originale /26-subnet.

Slik:

$$10\underline{0}00000 = .128$$

$$10\underline{1}00000 = .160$$

**Vi har nå subnettet ett subnet!!! HURRA!**

De to nye sub-subnettene du har laget, vil få en ny subnetmaske. Den originale subnetmaske /24 ble først endret til /26 for nettverk A, deretter sub-subnettet til ett /27-nettverk.

Nettverksmasken for de to nye /27-nettverkene blir:

NNNNNNNN.NNNNNNNN.NNNNNNNN.NNNHHHHH

som er;

$$11111111.11111111.11111111.11100000 = 255.255.255.224 = /27$$

Nå kan du velge ett av disse /27-nettverk for nettverk B:

For eksempel 1000000 /27 = Nettverk B

Det igjenstående /27-nettverk kommer vi tilbake til senere.

Vi må forsikre oss om at adresser ikke overlapper hverandre, så vi går tilbake til den originale tabellen.:

00 <b>000000</b> (Original)	.0/26	
01 <b>000000</b> (Original)	.64/26	Nettverk A
10 <b>000000</b> (Original)	.128/26	
11 <b>000000</b> (Original)	.192/26	

Vi valgte ett av /26-nettverkene (128/26) og sub-subnettet dette opp i to /27-nettverk. Ny tabell blir da seende slik ut:

00 <b>000000</b> (Original)	.0/26	
01 <b>000000</b> (Original)	.64/26	Nettverk A
10 <b>000000</b> (Original)	.128/26	Kan ikke brukes fordi det har blitt subnettet
10 <b>000000</b>	.128/27	Nettverk B
10 <b>100000</b>	.160/27	
11 <b>000000</b> (Original)	.192/26	

De gjenstående nettverk kan bli brukt som de er, eller subnettet videre.

## Punkt 4.

*Finn det tredje største subnettet.*

Det tredje største nettverket er nettverk C og nettverk D som begge skal kunne adressere 12 hosts (H)

Først bestemmer vi hvor mange H-bit (b) vi trenger for å adressere 12 hosts (H):

$$2^b - 2 \geq 12$$

$$b = 4$$

Du trenger 4 bit for å kunne adressere 12 hosts (H)

Du startet med 2 N-bit og 6 H-bit for nettverk A. Dette er fortsatt ditt utgangspunkt.

Du kan nå velge hvor disse nettverkene skal plasseres. Du kan velge et annet /26-nettverk, eller du kan gå til det ledige /27-nettverket for å se om nettverk C og D "får plass" inn der.

I dette eksempelet velger vi det ledige 160/27-nettverket.

10**100000** (Det tredje bittet "1" er nå ikke markert, siden bittet er en del av N-bittene.)

Du trenger bare 4 bit for å kunne adressere 12 hosts (H). Du står derfor tilbake med:

101**N0000**

hvor;

- 10 representerer det originale subnettet
- N representerer det ekstra bittet
- **00000** representerer de 5 H-bits du trenger for nettverk B.

Siden du har ett ekstra bit, kan du lage to mindre nettverk ut fra 160/27-nettverket.

Slik:

101**0**0000 = .160  
101**1**0000 = .176

Disse sub-sub-subnet vil nå få en ny nettverksmaske. Hvert sub-sub-subnet har nå 4 N-bits og 4 H-bits, så den nye nettverksmasken vil bli:

NNNNNNNN.NNNNNNNN.NNNNNNNN.NNNNHHHH

som er;

11111111.11111111.11111111.11110000 = 255.255.255.240

eller /28.

Nettverk C og D kan nå plasseres inn i disse to nye sub-sub-subnet.

Ny tabell blir nå:

00 <b>000000</b> (Original)	.0/26	subnet	
01 <b>000000</b> (Original)	.64/26	subnet	Nettverk A
10 <b>000000</b> (Original)	.128/26	subnet	Kan ikke brukes fordi det er subnettet
10 <b>000000</b>	.128/27	sub-subnet	Nettverk B
10 <b>100000</b>	.160/27	sub-subnet	Kan ikke brukes fordi det er subnettet
101 <b>00000</b>	.160/28	sub-sub-subnet	Nettverk C
101 <b>10000</b>	.176/28	sub-sub-subnet	Nettverk D
11 <b>000000</b> (Original)	.192/26	subnet	

Du har nå brukt to av de fire originale subnet og plassert inn nettverk A, B, C og D. Det neste du skal gjøre, er å bestemme nettverksnummrene til serielinkene (E, F, G og H)

## Trinn 5.

*Bestemme nettverksnummrene for serieforbindelsene.*

Serielinkene E, F G og H trenger bare å adressere 2 hosts (H). En til hver ruter som de er tilkoblet.

Vi bestemmer hvor mange bit (b) vi trenger for å adressere 2 hosts (H):

$$2^b - 2 \geq 2$$
$$b = 2$$

Du trenger 2 H-bits for hvert av nettverkene E, F, G og H.

Siden du allerede har subnettet to av de fire originale subnet, så har du to originale subnet igjen som du kan arbeide med.

For dette eksempelet velger vi .0/26-nettverket:

00**000000**

Du trenger bare 2 H-bit, så derfor står vi igjen med:

00**NNNN00**

hvor

- 00 representerer det originale nettverket.
- NNNN representerer de ekstra bit som du har
- 00 representerer de to H-bits du trenger for serielinkene.

Fordi du har 4 N-bits, så kan du lage 16 subnet ut fra det originale subnet:

00**000000** = .0/30  
00**000100** = .4/30  
00**001000** = .8/30

00**001100** = .12/30  
 00**010000** = .16/30  
 -  
 -  
 -  
 -  
 00**111000** = .56/30  
 00**111100** = .60/30

Du trenger jo bare fire av dem. Resten kan du "spare" for senere bruk, eller lage ett større subnet av disse subnet slik som vist nedenfor:

00**010000** = .16/30  
 -  
 -  
 -  
 -  
 00**111000** = .56/30  
 00**111100** = .60/30

} **00010000 = .16/28**

Du har nå laget en plan for nettverket med klasse C-nettverket 192.168.100.0/24. en effektiv bruk av det tildelte adresseområdet

### Sammendrag.

Med utgangspunkt i den originale tabellen, får vi en ny og endelig tabell:

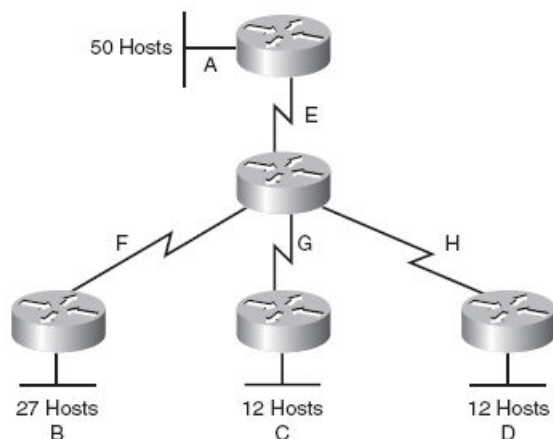
<b>00000000 (Original)</b>	.0/26	Sub-net <b>(Original)</b>	Kan ikke brukes fordi det er subnett (Orig)
00 <b>000000</b>	.0/30	sub-sub-net	Nettverk E
00 <b>000100</b>	.4/30	sub-sub-net	Nettverk F
00 <b>001000</b>	.8/30	sub-sub-net	Nettverk G
00 <b>001100</b>	.12/30	sub-sub-net	Nettverk H
00 <b>010000</b>	.16/28	sub-sub-net	Fremtidig bruk (Se egen beregning)
<b>01000000 (Original)</b>	.64/26	Sub-net <b>(Original)</b>	Nettverk A <b>(Original)</b>
<b>10000000 (Original)</b>	.128/26	Sub-net <b>(Original)</b>	Kan ikke brukes fordi det er subnett (Orig.)
10 <b>000000</b>	.128/27	sub-sub-net	Nettverk B
10 <b>100000</b>	.160/27	sub-sub-net	Kan ikke brukes fordi det er subnett
101 <b>00000</b>	.160/28	sub-sub-sub-net	Nettverk C
101 <b>10000</b>	.176/28	sub-sub-sub-net	Nettverk D
<b>11000000 (Original)</b>	.192/26	Sub-net <b>(Original)</b>	Fremtidig bruk <b>(Original)</b>

Hvis du studerer tabellen, så ser du at ingen nettverksnummer blir brukt to ganger. Ingen adresser er nå bortkastet for serielinkene.

På de neste sidene finner du en konsentrert oversikt av dette kompendiet i form av tabeller.

# Utvikling av et VLSM-adresseskjema.

Johan Futsæther 6. Mars 2007



Gitt klasse C-nettverket 192.168.100.0/24.

$$\text{Antall subnet} = 2^N - 2$$

$$\text{Antall subnet hvis "subnet zero" satt} = 2^N$$

$$\text{Antall hosts (H)} = 2^H - 2$$

Du finner nettverksID ved å sette alle hostbit (H) til "0".

Du finner nettverksmasken ved å sette alle hostbits til "0" og alle nettverksbittene (N) til "1".

Finner først det største nettverket (Nettverk A med 50 Hosts) for å finne antall hostbit (H) som er nødvendig. Deretter hvor mange subnet vi kan lage med gjenstående nettverksbit (N). I dette tilfelle blir det  $2^2 = 4$  subnet som vist i tabell 1. Plasserer

inn nett A i ett fritt valgt subnet.

0000000	(Original)	.0	(Orig)	
0100000	(Original)	.64	(Orig)	Nettverk A
1000000	(Original)	.128	(Orig)	
1100000	(Original)	.192	(Orig)	

Tabell 1

Finner deretter det nest største nettverket (Nettverk B). Sammenligner hvor mange "H"-bit jeg må bruke i forhold til det største nettverk. Får ett "H"-bit mindre som jeg kan gjøre om til "N"-bit og dermed lage to subnet.

0000000	(Original)	.0/26	(Orig)	Subnet	(Orig)	
0100000	(Original)	.64/26	(Orig)	Subnet	(Orig)	Nettverk A
1000000	(Original)	.128/26	(Orig)	Subnet	(Orig)	Kan ikke brukes fordi det er subnettet
1000000		.128/27		sub-subnet		Nettverk B
1010000		.160/27		sub-subnet		
1100000	(Original)	.192/26	(Orig)	Subnet	(Orig)	

Tabell 2

Finner deretter det tredje største nettverket (Nettverk C og D). Sammenligner hvor mange "H"-bit jeg trenger i forhold til det nest største nettverket. Får ett "H"-bit mindre som jeg kan gjøre om til "N"-bit og kan dermed lage to subnet inn i ett av de andre subnet.

0000000	(Original)	.0/26	(Orig)	Subnet	(Orig)	
0100000	(Original)	.64/26	(Orig)	Subnet	(Orig)	Nettverk A
1000000	(Original)	.128/26	(Orig)	Subnet	(Orig)	Kan ikke brukes fordi det er subnettet
1000000		.128/27		sub-subnet		Nettverk B
1010000		.160/27		sub-subnet		Kan ikke brukes fordi det er subnettet
1010000		.160/28		sub-sub-subnet		Nettverk C
1011000		.176/28		sub-sub-subnet		Nettverk D
1100000	(Original)	.192/26	(Orig)	Subnet	(Orig)	

Tabell 3

Finner det fjerde største nettverket (i dette tilfelle serielinkene E, F, G og H). Finner hvor mange "H"-bit jeg trenger for deretter finne antall "N"-bit. Kan nå lage 8 subnett som blir plassert inn på ledig plass. Fire av disse 8 nettene blir sammenfattet til ett nett (.16.28).

0000000	(Original)	.0/26	(Orig)	Sub-net	(Orig)	Kan ikke brukes fordi det er subnettet (Orig)
0000000		.0/30		sub-sub-net		Nettverk E
0000100		.4/30		sub-sub-net		Nettverk F
0001000		.8/30		sub-sub-net		Nettverk G
0001100		.12/30		sub-sub-net		Nettverk H
0010000		.16/28		sub-sub-net		Fremtidig bruk (Se egen beregning)
0100000	(Original)	.64/26	(Orig)	Sub-net	(Orig)	Nettverk A (Original)
1000000	(Original)	.128/26	(Orig)	Sub-net	(Orig)	Kan ikke brukes fordi det er subnettet (Orig.)
1000000		.128/27		sub-sub-net		Nettverk B
1010000		.160/27		sub-sub-net		Kan ikke brukes fordi det er subnettet
1010000		.160/28		sub-sub-sub-net		Nettverk C
1011000		.176/28		sub-sub-sub-net		Nettverk D
1100000	(Original)	.192/26	(Orig)	Sub-net.	(Orig)	Fremtidig bruk (Original)

Tabell 4

# VLSM- diagram

Variable Length Subnet Mask

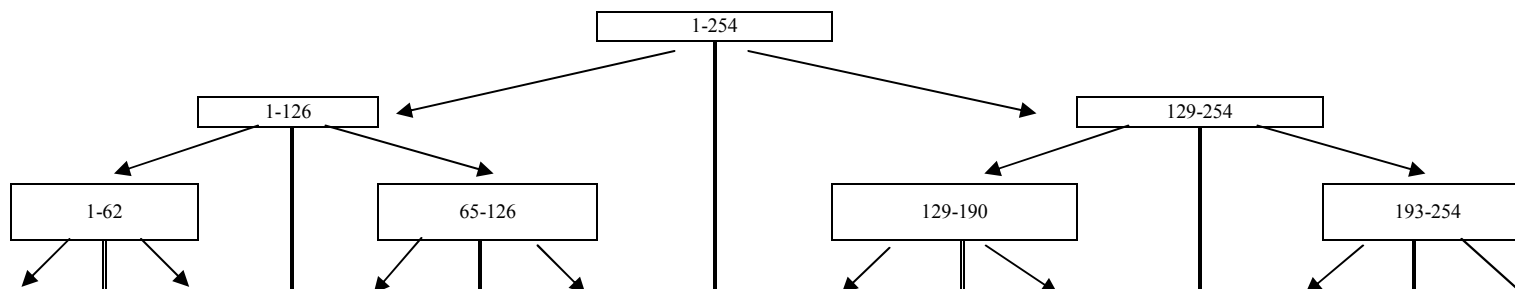
Johan Futsæther

Prefix	Mask	Hosts
--------	------	-------

/24	0	254
-----	---	-----

/25	128	126
-----	-----	-----

/26	192	62
-----	-----	----



Prefix	Mask	Hosts	1-30	33-62	65-94	97-126	129-158	161-190	193-222	225-254								
/27	224	30																
/28	240	14	1-14	17-30	33-46	49-62	65-78	81-94	97-110	113-126	129-142	145-158	161-174	177-190	193-206	209-222	225-238	241-254
/29	248	6	1-6	17-22	33-38	49-64	65-70	81-86	97-102	113-118	129-134	145-150	161-166	177-182	193-198	209-214	225-230	241-246
			9-14	25-30	41-46	57-62	73-78	89-94	105-110	121-126	137-142	153-158	169-174	185-190	201-206	217-222	233-238	249-254
/30	252	2	1-2	17-18	33-34	49-50	65-66	81-82	97-98	113-114	129-130	145-146	161-162	177-178	193-194	209-210	225-226	241-242
			5-6	21-22	37-38	53-54	69-70	85-86	101-102	117-118	133-134	149-150	165-166	181-182	197-198	213-214	229-230	245-246
			9-10	25-26	41-42	57-58	73-74	89-90	105-106	121-122	137-138	153-154	169-170	185-186	201-202	217-218	233-234	249-250
			13-14	29-30	45-46	61-62	77-78	93-94	109-110	125-126	141-142	157-158	173-174	189-190	205-206	221-222	237-238	253-254



# VLSM

Variable Length Subnet Mask  
Johan Futsæther

