

Calculating IP VLSM Subnet Addresses

A. P. Robson

`adrian.robson@nepsweb.co.uk`

29 October 2011

Contents

1	Introduction	1
2	Method	2
2.1	Gather Data	2
2.2	Calculate Subnet Sizes	2
2.3	Assign Subsubnet Addresses	3
2.4	Calculate Standard Addresses	4
2.5	Complete the Addressing Scheme	5
2.5.1	Single Octet Subnet	5
2.5.2	Multi-Octet Subnets	6
2.5.3	Part-Octet Subnets	6
3	Example	6
3.1	Gather Data	6
3.2	Calculate Subnet Sizes	7
3.3	Assign Subsubnet Addresses	7
3.4	Calculate Standard Addresses	7
3.5	Complete the Addressing Scheme	7
4	Useful Data	8
4.1	Binary Values	8
4.2	Classful Addressing	8

1 Introduction

This paper presents a detailed method for calculating IP subnet address schemes by hand. It is easier to perform if a calculator that supports binary is available, but it can be done using only mental arithmetic. The binary value table given in §4.1 is invaluable in this case, and this table can be quickly constructed if required.

2 Method

The method given here generates the required addressing scheme in stages using a number of tables with duplicated data. However, these table can be merged into a single large table for convenience if required. The stages are:

1. Gather Data – §2.1
2. Calculate Subnet Sizes– §2.2
3. Assign Subsubnet Addresses – §2.3
4. Calculate Standard Addresses – §2.4
5. Complete the Addressing Scheme – §2.5

2.1 Gather Data

Get the major IP address and mask. If the major address is given as a classful address consult §4.2 for the mask. Note the total number of bits in its host part. These are the bits available for sub-netting, or the length of the subnet. Identify the required networks and how many hosts they have. Remember that WAN links are networks with two hosts.

2.2 Calculate Subnet Sizes

Create a table with the following columns:

Name	Hosts	HostB	NetB	MaskB
------	-------	-------	------	-------

Name The specified name of the subnet.

Hosts The specified number of hosts in the subnet.

HostB The calculated number of bits required for the hosts.

NetB The calculated number bits in the net part of the subnet address.

MasB The calculated number bits in the full mask.

Fill the table by:

1. Listing the networks and the number of required hosts in descending order of number of hosts.
2. Calculate the number of host bits that are needed for the given hosts. The maximum number of hosts that can be specified by n bits is $2^n - 2$, and this gives the formula:

$$hostB_n = \lceil \log_2(hosts_n + 2) \rceil$$

If a calculator that handles \log_2 is not available, this conversion can be used:

$$\log_2 x = \frac{\lg x}{\lg 2}$$

The following table can be constructed and consulted if there is no calculator:

Number of bits	1	2	3	4	5	6	7	8
2^{bits}	2	4	8	16	32	64	128	256
Maximum hosts	0	2	6	14	30	62	126	254

- For each row calculate the number of network and mask bits as follows:

$$netB_n = subnetB - hostB_n$$

$$maskB_n = 32 - hostB_n$$

2.3 Assign Subsubnet Addresses

At this stage work with just the number of bits in the subnet. Create a table with the following columns:

Name	Net Bits	Host Bits	Subnet Bit Grid
------	----------	-----------	-----------------

Name The name of the subnet – from the previous stage.

Net Bits The number of bits in the net part of the subnet address – from the previous stage.

Host Bits The number of bits in the host part of the subnet address – from the previous stage.

Subnet Bit Grid A grid for the calculation of the network address. There should be a location in the grid for each bit available for subnetting.

Fill the table by

- Transferring Name, Net Bits and Host Bits from the previous stage.
- The process for assigning subsubnet addresses is as follows:
 - Starting with the smallest net (the most hosts) at the top of the table, give it an address of zero.
 - Fill the remaining bit locations, which are the host, with H's.
 - For the next net, use the previous address incremented by one.
 - If this net is wider than the previous one, add zeros to the required width.
 - Repeat from **2b** until there are no more networks

2.4 Calculate Standard Addresses

At this stage work with just the number of bits in the subnet, and calculate decimal values. Create a table with the following columns:

Name	Binary Addr	Host Bits	Inverse Mask	Net Addr	Bcast Addr	Mask	First Host	Last Host
------	-------------	-----------	--------------	----------	------------	------	------------	-----------

Name The name of the subnet – from the previous table.

Binary Addr The subsubnet addresses in binary – from the Subnet Bit Grid column of the previous table.

Host Bits The number of host bits in the address – from the previous table.

Inverse Mask The calculated ‘inverse mask’, which is used to calculate the other addresses in the table.

Net Addr The calculated network address.

Bcast Addr The calculated broadcast address.

Mask The calculated mask.

First Host The calculated address of the first host in the subsubnet.

Last Host The calculated address of the last host in the subsubnet.

Fill the table by

1. Transferring Name, Binary Addr and Host Bits columns from the previous table.
2. For each row in the table, use the number of host bits to calculate the inverse masks using the formula

$$invMask_n = 2^{hostBits_n} - 1$$

Without a calculator construct and use the following table:

Number of host bits	1	2	3	4	5	6	7	8
2^{bits}	2	4	8	16	32	64	128	256
Inverse Mask	1	3	7	15	31	63	127	255

3. The network addresses are obtained by replacing the H’s in the bits grid with zeros; and the broadcast addresses by replacing the H’s with ones. these binary values are then converted to decimal. However, this is rather tedious without a calculator, so the following method can be used:
 - (a) For the first row, make the network address zero and broadcast address the same as the inverse mask:

$$\begin{aligned} networkA_1 &= 0 \\ broadcastA_1 &= maskInv_1 \end{aligned}$$

- (b) Now for each subsequent row, make the network address one more than the previous row's broadcast address, and the broadcast address the network address plus the inverse mask.

$$\begin{aligned} netAddr_n &= bcastAddr_{n-1} + 1 \\ bcastAddr_n &= netAddr_n + invMask_n \end{aligned}$$

4. Fill the mask column by subtracting the inverse masks from 255:

$$mask_n = 255 - invMask_n$$

5. Fill in the first host column with one more than the network address:

$$firstHost_n = netAddr_n + 1$$

6. Fill in the last host column with one less than the broadcast address:

$$lastHost_n = bcastAddr_n - 1$$

2.5 Complete the Addressing Scheme

The final stage is to draw the above information into a single table that gives full 32 bit IP addresses in decimal four octet notation. The table should have the following headings:

Name	Req Hosts	Avail Hosts	Network Addr	Mask	First Host	Last Host	Broadcast Addr
------	-----------	-------------	--------------	------	------------	-----------	----------------

Name The name of the subnet.

Req Hosts The number of hosts required in the subnet.

Avail Hosts The maximum number of hosts available in the subnet.

Network Addr The network address of the subnet.

Mask The addresses mask in suffix notation and decimal notation.

First Host The first usable host address.

Last Host The last usable host address.

Broadcast Addr The broadcast address for the subnet.

2.5.1 Single Octet Subnet

If the subnet is exactly 8 bits long (a single octet) then composing the above addresses is simple. Just replace the last octet in the major IP address with the required value. Fortunately, this format is typical for test or examination problems.

2.5.2 Multi-Octet Subnets

If subnet is longer than 8 bits, the value has to be split into octets. Do this by converting the the address to binary; splitting it into 8 bit segments, counting left to right; and converting the segments to decimal. For example:

$$1700 = 11010100100 \rightarrow 110.10100100 = 6.164$$

Alternatively, if a suitable calculator is available the following computations can be used:

$$\begin{aligned} \text{Octet}_4 &= \text{value} \bmod 256 \\ \text{Octet}_3 &= \lfloor \text{value}/256 \rfloor \bmod 256 \\ \text{Octet}_2 &= \lfloor \text{value}/256^2 \rfloor \end{aligned}$$

2.5.3 Part-Octet Subnets

If the subnet split is not on an octet boundary, just add the left most subnet octet to the right most octet from the major IP address's network part to form the split octet. For example if we have the network address 128.10.160.0/20 and the subnet 6.164, then

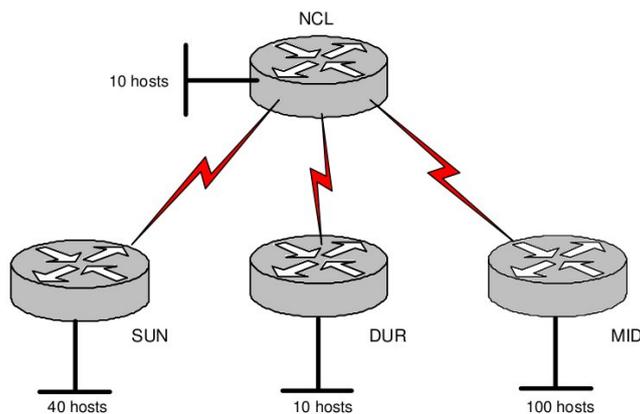
128.10.160.0	10000000.00001010.1010 0000.00000000
6.164	110.10100100
128.10.166.164	10000000.00001010.1010 0110.10100100

3 Example¹

3.1 Gather Data

Master network address: 212.10.10.0/24 (class C)

Total subnet size: 8 bits.



¹Thanks to David Kendall, Northumbria University, for this example.

3.2 Calculate Subnet Sizes

Name	Hosts	Host Bits	Subsubnet Bits	Mask bits
MID	100	7	1	25
SUN	40	6	2	26
DUR	10	4	4	28
NCL	10	4	4	28
WAN1	2	2	6	30
WAN2	2	2	6	30
WAN3	2	2	6	30

3.3 Assign Subsubnet Addresses

Name	Net Bits	Host Bits	Subnet Bit Grid
MID	1	7	0 H H H H H H H
SUN	2	6	1 0 H H H H H H
DUR	4	4	1 1 0 0 H H H H
NCL	4	4	1 1 0 1 H H H H
WAN1	6	2	1 1 1 0 0 0 H H
WAN2	6	2	1 1 1 0 0 1 H H
WAN3	6	2	1 1 1 0 1 0 H H

3.4 Calculate Standard Addresses

Name	Binary Addr	Host Bits	Inverse Mask	Net Addr	Bcast Addr	Mask	First Host	Last Host
MID	0HHHHHHH	7	127	0	127	128	1	126
SUN	10HHHHHH	6	63	128	191	192	129	190
DUR	1100HHHH	4	15	192	207	240	193	206
NCL	1101HHHH	4	15	208	223	240	209	222
WAN1	111000HH	2	3	224	227	252	225	226
WAN2	111001HH	2	3	228	231	252	229	230
WAN3	111010HH	2	3	232	235	252	233	234

3.5 Complete the Addressing Scheme

Name	Requ Hosts	Avail Hosts	Network Addr	Mask
MID	100	126	212.10.10.0	255.255.255.128 /25
SUN	40	62	212.10.10.128	255.255.255.192 /26
DUR	10	14	212.10.10.192	255.255.255.240 /28
NCL	10	14	212.10.10.208	255.255.255.240 /28
WAN1	2	2	212.10.10.224	255.255.255.252 /30
WAN2	2	2	212.10.10.228	255.255.255.252 /30
WAN3	2	2	212.10.10.232	255.255.255.252 /30

Name	First Host	Last Host	Addr
MID	212.10.10.1	212.10.10.126	212.10.10.127
SUN	212.10.10.129	212.10.10.190	212.10.10.191
DUR	212.10.10.193	212.10.10.206	212.10.10.207
NCL	212.10.10.209	212.10.10.222	212.10.10.223
WAN1	212.10.10.225	212.10.10.226	212.10.10.227
WAN2	212.10.10.229	212.10.10.230	212.10.10.231
WAN3	212.10.10.233	212.10.10.234	212.10.10.235

4 Useful Data

4.1 Binary Values

Number of host bits		1	2	3	4	5	6	7	8
	2^{bits}	2	4	8	16	32	64	128	256
Inverse Mask	$2^{bits} - 1$	1	3	7	15	31	63	127	255
Maximum hosts	$2^{bits} - 2$	0	2	6	14	30	62	126	254

4.2 Classful Addressing

Class	1st Octet Binary	1st Octet Range	Net ID	Host ID	Mask	
A	0XXXXXXX	1-127	a	b.c.d	255.0.0.0	/8
B	10XXXXXX	128-191	a.b	c.d	255.255.0.0	/16
C	110XXXXX	192-223	a.b.c	d	255.255.255.0	/24