IMT2521 Network administration, Binary numbers and arithmetic IP addressing and subnet calculations

The binary system

The number system we use on a daily basis, the decimal system, uses ten numeric symbols (*), the numbers 0 through 9. The binary system on the other hand, only has two symbols: 0 and 1. These two numeric symbols are used to build larger numbers, just the same way we use the symbols 0 to 9 to build large numbers, such as 1337.

Numeric values in the binary system are determined by position, building from a base value of

$$2^n$$

* Number, Value, Numeric value and numeric symbol are fundamentally different things. A numeric value is the base magnitude a given numeric symbol represents. A number is simply a composition of numeric symbols, while a value is the magnitude represented by the number.

The decimal system, quickly

10^3	10^2	10^1	10^0
1	3	3	7

$$10^3 \cdot 1 = 1000$$

$$10^2 \cdot 3 = 300$$

$$10^1 \cdot 3 = 30$$

$$10^0 \cdot 7 = 7$$

$$1000 + 300 + 30 + 7 = 1337$$

The binary positions

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

As we commonly operate on values in the range 0 to 255 (decimal), we should be able to learn the representation and position of these common positions

Examples

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
128	64	32	16	8	4	2	1	
0	1	0	0	0	1	1	0	70
1	0	0	1	0	0	1	1	147
0	0	0	0	0	0	1	1	3
1	1	1	1	1	1	0	0	252 (255 - 3)

$$64 + 4 + 2 = 70$$

$$128 + 16 + 2 + 1 = 147$$

$$2 + 1 = 3$$

$$128 + 64 + 32 + 16 + 8 + 4 = 252$$

From decimal to binary

- Is the number greater than, or equal to 128?
  - **Yes**: put 1 at position 2^7, and subtract 128 / **No**: put 0 at position 2^7.
- Is the number greater than, or equal to 64?
  - **Yes**: put 1 at position 2^6, and subtract 64 / **No**: put 0 at position 2^6.
- Is the number greater than, or equal to 32?
  - **Yes**: put 1 at position 2^5, and subtract 32 / **No**: put 0 at position 2^5.
- Is the number greater than, or equal to 16?
  - **Yes**: put 1 at position 2^4, and subtract 16 / **No**: put 0 at position 2^4.
- Is the number greater than, or equal to 8?
  - **Yes**: put 1 at position 2^3, and subtract 8 / **No**: put 0 at position 2^3.
- Is the number greater than, or equal to 4?
  - **Yes**: put 1 at position 2^2, and subtract 4 / **No**: put 0 at position 2^2.
- Is the number greater than, or equal to 2?
  - **Yes**: put 1 at position 2^1, and subtract 2 / **No**: put 0 at position 2^1.
- Is the number equal to 1?
  - **Yes**: put 1 at position 2^0 / **No**: put 0 at position 2^0.

From decimal to binary, 193

Pos.	Value	Binary	Subtract	Remain
128	193	1	128	65
64	65	1	64	1
32	1	0		
16	.	0		
8	.	0		
4	.	0		
2	.	0		
1	1	1	1	0

Result: 1 1 0 0 0 0 0 1

From decimal to binary, 75

Pos.	Value	Binary	Subtract	Remain
128	75	0		75
64	75	1	64	11
32	11	0		
16	.	0		
8	11	1	8	3
4	3	0		
2	3	1	2	1
1	1	1	1	0

Resultat: 0 1 0 0 1 0 1 1

From decimal to binary, 10

Pos.	Value	Binary	Subtract	Remain
128	10	0		
64	.	0		
32	.	0		
16	.	0		
8	10	1	8	2
4	2	0		
2	2	1	2	0
1	0	0		

Resultat: 0 0 0 0 1 0 1 0

From decimal to binary, 235

Pos.	Value	Binary	Subtract	Remain
128	235	1	128	107
64	107	1	64	43
32	43	1	32	11
16	43	0		
8	11	1	8	3
4	11	0		
2	1	1	2	1
1	1	1	1	0

Resultat: 1 1 1 0 1 0 1 1

The buildup of an IP address

An IPv4 address is really a 32-bit value, written as four groups of eight bit.

- Makes them easier to read.
- Makes grouping easier.
- Makes calculations on parts simpler.

We can use the method we just learned, and do calculations on each octet by itself. In sub-neting and super-neting, we may ignore the octets that are not affected by a given mask.

192	.	168	.	0	.	10
11000000	.	10101000	.	00000000	.	00001010

What is a network mask?

A network mask is a 32-bit number determining what parts of an IP-address is representing a network, and what parts represent a single given host within that network.

- A network: a logical group of nodes
- A network address, the part of all node addresses that are identical
- A host/node address, the part of the address that is unique.
- The network mask is involved in determining network and host address.

How does the network mask work?

- In the network mask, all bits that represent the **network** have the value **1**, while all the bits that represent the host component have the value **0**.
- In the network mask, we supply ones (**1**) from the left until the *entire* network address is identified. After this, we suppy zeroes (**0** until we have 32 bits.
- There will **never** be a zero before a one when reading from left to right
- Simmilar; there will **never** be a one before a zero when reading from right to left.
- The number of ones in the network mask is called the mask length.

How does the network mask behave in binary?

- Network equipment use the mask to do boolean calculations
  - AND
  - NOT
- IP-address AND network mask gives network address
- IP-address AND NOT network mask gives the host address.

How does the network mask behave in binary?

	192	.	168	.	0	.	10
	11000000	.	10101000	.	00000000	.	00001010
	255	.	255	.	255	.	0
	11111111	.	11111111	.	11111111	.	00000000
	11000000	.	10101000	.	00000000	.	00001010
&	11111111	.	11111111	.	11111111	.	00000000
=	11000000	.	10101000	.	00000000	.	00000000
=	192	.	168	.	0	.	0
	11000000	.	10101000	.	00000000	.	00001010
& !	11111111	.	11111111	.	11111111	.	00000000
=	00000000	.	00000000	.	00000000	.	00001010
=	0	.	0	.	0	.	10

A simpler view on the network mask

When you see a network mask and an IP address represented in binary:

- Place a separator between the last **1** and the first **0** in the mask
- Place a separator at the same spot in the IP address.
- Anything **before** the separator is the network address.
- Anything **after** the separator is the host address.

An example of simple mask representation:

	192	.	168	.	0	.	10
	11000000	.	10101000	.	00000000	.	00001010
	255	.	255	.	255	.	0
	11111111	.	11111111	.	11111111	.	00000000
	11111111	.	11111111	.	11111111		00000000
	11000000	.	10101000	.	00000000		00001010
nett	**11000000**	.	**10101000**	.	**00000000**		00000000
host	nnnnnnnn	.	nnnnnnnn	.	nnnnnnnn		**00001010**

Another example of simple mask representation:

	192	.	168	.	20	.	10
	11000000	.	10101000	.	00010100	.	00001010
	255	.	255	.	0	.	0
	11111111	.	11111111	.	00000000	.	00000000
	11111111	.	11111111		00000000	.	00000000
	11000000	.	10101000		00010100	.	00001010
nett	**11000000**	.	**10101000**		00000000	.	00000000
host	nnnnnnnn	.	nnnnnnnn		**00010100**	.	**00001010**

Network mask cheat-sheet

Bits borrowed	0	1	2	3	4	5	6	7	8
Prefix	/24	/25	/26	/27	/28	/29	/30	/31	/32
Netmask	0	128	192	224	240	248	252	254	255
N. of nets	1	2	4	8	16	32	64	128	256
N. of values	256	128	64	32	16	8	4	2	1
N. of hosts	254	126	62	30	14	6	2	-	/

Broadcast!

A complete IP addressing is composed by:

- Network mask
  - determining what is the network- and what is the host address
- Network address
  - The part of the IP address where all netmask bits are set to 1
- Hostaddress
  - The part of the IP address where all netmask bits are 0

In addition to this, we have the **broadcast address**
This is composed of: Denne er består av:

- All parts of the network adress where the mask has a 1
- All bits set to 1 where the network mask is 0

Broadcast? Purpose

The purpose of the broadcast address is to be able to send data to **all** hosts in the same network at the same time.
Hensikten med broadcast-adressen er å kunne sende data til **alle** hosts i samme nettverk samtidig!
A computer/host listens for packets sendt to:

- Its network address, with its own host address
- Its network address, and all host-bits set to one.

IP	192	.	168	.	0	. 10
Mask	255	.	255	.	255	. 0
Net	**11000000**	.	**10101000**	.	**00000000**	00000000
Host	nnnnnnnn	.	nnnnnnnn	.	nnnnnnnn	**00001010**
B-Cast	**11000000**	.	**10101000**	.	**00000000**	**11111111**
B-Cast	**192**	.	**168**	.	**0**	**255**

IP-addresses, class

Back when the TCP model was introduced, and IP addressing was first deployed, not many needed addresses, but they were needed in diffrent scopes. So, classes were applied.
Classes were ranges of nettwork addresses that could be allocated to organizations based on an estimate of the number of hosts the organisation planned to deploy.
Classes are today an impractical idea for deployment on the Internet, but the still come up and bite us, so we need to know how they work.

IP-classes, distribution

Class	Class A	Class B	Class C	Class D	Class
Num. of bits	24	16	8		
Mask-bits	8	16	24		
Num. of hosts	16 777 214	65 534	254		
Num. of nets	128	16 384	2 097 152		
Use	Large	Medium	Small	Multicast	Res.

IP-classes, numbering

The class-division of IP-addresses is done in the leading bits.

- Class A:
  - No leading 1's before first 0
- Class B:
  - One leading 1's before first 0
- Class C:
  - Two leading 1's before first 0
- A 0 after the leading 1's

Class A	1-127	0	**0**nnn nnnn
Class B	128-191	10	**10**nn nnnn . nnnn nnnn
Class C	192-223	110	**110**n nnnn . nnnn nnnn . nnnn nnnn
Class D	224-239	1110	**1110**
Class E	240-255	11110	**1111 0**

Special network numbers

A set of special address ranges are allocated:

- One for loopback (host local) addressing and internal testing (RFC1700, 3330)
- Four for internal networks, also known as private addresses (RFC1918, RFC3330)

The four private RFC1918 addressblocks are:

- One Class A net: 10.0.0.0/8
- Sixteen Class B nets: 172.16.0.0/16 til 172.31.0.0/16
- 255 Class C nets: 192.168.0.0/24 til 192.168.255.0/24
- Link-local net, one Class B net: 169.254.0.0/16

Subnets

- Subnetting involves splitting a net into smaller parts
- Subnetting is done by "borrowing" bits from the host component
- Each bit borrowed splits the previous net in two halves.

Two formulas:

- Number of host addresses $= 2^h - 2$
- Number of subnets $= 2^s$

Here, **s** represents number of borrowed bits, and **h** the number of remaining host-address bits.

The size of a network

- The number of hosts in a network is two less than the number of valid values.
- Number of valid values is determined by raising two by the power of the number of host address bits.

Eksempel 1:

- Netmask 255.255.255.0 (24 bit, 8 bit host)
- Valid values: 256 (2^8), 0 to 255
- Number of hosts: 256 - 2 = 254

Eksempel 2:

- Netmask 255.255.255.128 (25 bit, 7 bit host)
- Valid values: 128 (2^7), 0 to 127 or 128 til 255
- Number of hosts: 128 - 2 = 126

Number of networks

- When subnetting, we always start with an original (allocated, chosen) network, with an original network mask.
- The number of identically-sized networks we and ut with when subnetting is determined by how many bits are extended to the original mask.
- Bits added/extended, are borrowed/stolen from the host-address part.

Example:

- Netmask 255.255.255.0 (24 bit, 8 bit host)
- Take two bits from host for use in new netmask
- New netmask: 255.255.255.192 (128 + 64)
- Number of nets: $2^2 = 4$
- Number of hosts in each net: $2^6 - 2 = 64 - 2 = 62$

Network adresses after subnetting

As we start subnetting with an original network address and netmask, and then extend the network mask, our resulting nettwork addresses will be the possible bit-combinations allowed by the added bits.

- 192.168.20.0 / 255.255.255.0
- 1100 0000 . 1010 1000 . 0001 1000

Extend netmask with two bits:

- — — . — — . — — . **00**00 0000 $\Longrightarrow$ 192.168.20.**0**
- — — . — — . — — . **01**00 0000 $\Longrightarrow$ 192.168.20.**64**
- — — . — — . — — . **10**00 0000 $\Longrightarrow$ 192.168.20.**128**
- — — . — — . — — . **11**00 0000 $\Longrightarrow$ 192.168.20.**192**

Supernetting!

- Supernetting does not mean building networks for Superman
- Supernetting is the opposite of subnetting
- A different name for supernetting is **summarization**

When summarizing, we move the network-mark to the left, and create fewer, but lager, networks.

- For each bit changed, we halve the number of networks, and double the number of hosts
- Two nets with identical mask, adjacent to each other on a "bit boundary" can be summarized directly by moving a bit in the netmask
- A "bit boundary" involbes having the first Bit to the left of the mask is set to zero in the first net, and one in the second.

Supernetting!

172.16.2.0/24 → 1010 1100 . 0001 0000 . 0000 001**0** . — —
172.16.3.0/24 → 1010 1100 . 0001 0000 . 0000 001**1** . — —

172.16.2.0/23 → 1010 1100 . 0001 0000 . 0000 001– . — —

172.16.0.0/24 → 1010 1100 . 0001 0000 . 0000 000**0** . — —
172.16.1.0/24 → 1010 1100 . 0001 0000 . 0000 000**1** . — —

172.16.0.0/23 → 1010 1100 . 0001 0000 . 0000 000– . — —

172.16.0.0/23 → 1010 1100 . 0001 0000 . 0000 00**0**– . — —
172.16.2.0/23 → 1010 1100 . 0001 0000 . 0000 00**1**– . — —

172.16.0.0/22 → 1010 1100 . 0001 0000 . 0000 00– – . — —

Supernetting!

The number of hosts in a net is determined by $2^h - 2$. When supernetting, we increase the number of bits in the host-part of the address

- Example, /24-nets joined to form a single /22:

$$2^8 = 256$$

$$2 \cdot 2^8 = 2^9 = 512$$

$$2 \cdot 2^9 = 2^{10} = 1024$$

We have moved from 24 mask-, 8 hostbits to 22 mask- and 10 hostbits, and end up with:

$$2^{10} - 2 = 1024 - 2 = 1022(\mathit{hosts})$$

Classfull addressing

We can change address ranges:

- From large nets to smaller nets using subnetting
- From smaller nets to lager nets using supernetting

Initially, one always started on the class boundaries, and created smaller subnets. In addition, all subnets created followed the classful principal, and were of identical size.

The same applied to supernetting, one never crossed the class boundary of the allocated classful network address when summarizing.

In classfull addressing:

- **all networks have the same size**
- class boundaries are absolute.

Inefficient addressing

Classfull addressing is inefficient, lots and lots of addresses are wasted, for instance when using link-networks. If you need 1025 hosts in a subnet, you need at leas a Class B network (65534 hosts), subnetted to 21 bits mask at the smallest. With 21 bits in the mask: $2^{11} - 2 = 2046 hosts$
Inside your organization, all nets would have to be that size, even if you were deploying a link-net, having only two hosts/nodes!

Classless addressing

Classless addressing involves not caring about the old class divisions!

- Classless addressing allows summarization of two classfull segments
  - As example, two C-class nets (startbit 110, 24 bit) can be summed into one.
- Classless addressing allows nets of different sizes within the same allocated segment.
- IP addresse is now done in a classless manner.

CIDR and VLSM

- VLSM
  In a classical network of classful nets, separated in smaller, equal parts. Communication went vie routers exchanging only network addresses. All nets had to be of identical size. Introduction of Variable Length Subnet Mask allowed networks to be separated into various different sizes.
  To accomplish this, Classless routing protocols where created, where the *network mask* was send along with the network address
- CIDR allows routing-tables to be smaller, by supernetting across classful boundaries, and using "prefix length" in place of the network mask
  - 192.168.20.0 255.255.252.0 $\Longrightarrow$ 192.168.20.0/22

CIDR compared to VLSM

- VLSM is sub- og supernetting inside a single network
- CIDR is summarization in routing between networks.
- VLSM-nets are dependant on CIDR to be able to communicate.
- CIDR applies VLSM as its absolute central concept.

Both are terms within **Classless addressing and routing**

TASKS!

- Check that you have IMT2521 in Fronter
- Verify that you can log on to Cisco Networking Academy
- IP Addressing and Subnetworking Workbook
- VLSM Workbook