IPV4 consists of a quartet of 8 bits ranging from 0 to 255

[00000000 – 11111111]

Depending on which class of network [ A B C ] will determine how many networks and how many hosts you can utilize.

[A] consists of 28 or 256 networks and 224 or 16,777,216 hosts per network

[N.H.H.H] 🡪 default subnet 255.0.0.0

[B] consists of 216 or 65,536 networks and 216 or 65,536 hosts per network

[N.N.H.H.] 🡪 default subnet 255.255.0.0

[C] consists of 224 or 16,777,216 networks and 28 hosts per network

[N.N.N.H] 🡪 default subnet 255.255.255.0

The term subnetting is defined as the process of breaking a network (mostly large networks) into multiple smaller networks.

This is achieved by “borrowing” bits from the hosts to expand the network range (why subnetting is mostly done on type A networks).

Let us take the example of Class A network 10.0.0.0. This one network supports 16,777,216 hosts. For most businesses this is too many potential hosts and devices that would be necessary, so you can subnet this Class A into parts and utilize the parts you need and lease the rest out.

Let us split this network into 6 parts. Since I can only divide the network into powers of two [2x], I would need to borrow 3 bits from the host [23] = 8. This will give me the 6 parts I need but 2 in reserve. **Remember you must borrow in powers of 2**.

So my default subnet would look like this: 11111111.00000000.00000000.00000000

So I am going to borrow 3 bits from the host from my 2nd quartet. [Lets mark them in Blue]

11111111.11100000.00000000.00000000

These 3 bits represent the decimal value of 128+64+32 = 224

The new subnet of our borrowed network will now change from:

255.0.0.0 to 255.224.0.0! This is now a classless network!

So what would be the address ranges for my new subnets?

**Binary place holders: 128.64.32.16.8.4.2.1**

**10.0.0.1** would still be my first but it will stop at **10.31.255.254**

(why 31? Because the next subnet will start at 32 [001]) 00100000

So the next would be **10.32.0.1** to **10.63.255.254**

(again we would stop at 63 since 64 is the next network [010]) 01000000

And the next would be **10.64.0.1** to **10.95.255.254**

(again we would stop at 95 since 96 is the next network [011]) 01100000

And the next would be **10.96.0.1** to **10.127.255.254**

(again we would stop at 127 since 128 is the next network [100]) 10000000

And the next would be **10.128.0.1** to **10.159.255.254**

(again we would stop at 159 since 160 is the next network [101]) 10100000

And the next would be **10.160.0.1** to **10.191.255.254**

(again we would stop at 191 since 192 is the next network [110]) 11000000

And the next would have **10.192.0.1** to **10.223.255.254**

(again we would stop at 223 since 224 is the next network [111] 11100000

And last we would have **10.224.0.1** to **10.255.255.254**

So the useable addresses for our 255.224.0.0 network would be:

|  |  |
| --- | --- |
| 10.0.0.1 | 10.31.255.254 |
| 10.32.0.1 | 10.63.255.254 |
| 10.64.0.1 | 10.95.255.254 |
| 10.96.0.1 | 10.127.255.254 |
| 10.128.0.1 | 10.159.255.254 |
| 10.160.0.1 | 10.191.255.254 |
| 10.192.0.1 | 10.223.255.254 |
| 10.224.0.1 | 10.255.255.254 |

So now I have 211 (2048) networks and 221 (2,097,152) hosts

So to satisfy my network needs I can use any 6 of these 8 groups and keep 2 in reserve.

In the real world, you would most like borrow 8 or more bits.

If I was to borrow 8 bits, my Class A address would appear to have a Class B subnet. That would break your network into 256 pieces (unlike the 8 we used in this example)

For Class A you can technically borrow up to 31 bits if you like, but it would only support 2 devices per network.