Subnetting

Josep Puigdemont

October 5, 2004

Abstract

This document is a brief proposal on namespace and address distribution for the ISP project. It has been done as a personal exercise on subnetting, but it was thought having in mind the possibility to applying it to the ISP project we should do. Since this is an exercise, I tend to write and clarify the most basic things, like what a `/16` network is, and some other stuff that you might even find insulting to your level of knowledge, but in fact this helps me a lot in remembering those basic things, and the exercise of writing it down helps this purpose.

You can take this document as a proposal, it doesn't mean that it has to be applied, although it would be cool :-)

Contents

1 Address space **2**

2 ISP network topology **2**

2.1 The ISP Core Network . . . 3

2.2 The Area Network . . . 3

2.2.1 Having fun with DNS . . . 4

2.3 The Access Network . . . 4

2.4 Black holes . . . 4

3 DNS name space **5**

List of Figures

1 Network Layout . . . 2

List of Tables

1 ISP Core Network data . . . 3

2 Area Network data . . . 3

3 Access Network Details . . . 4

4 `R4`'s Access Network . . . 5

5 Different sizes of subnets . . . 6

6 Host names . . . 7

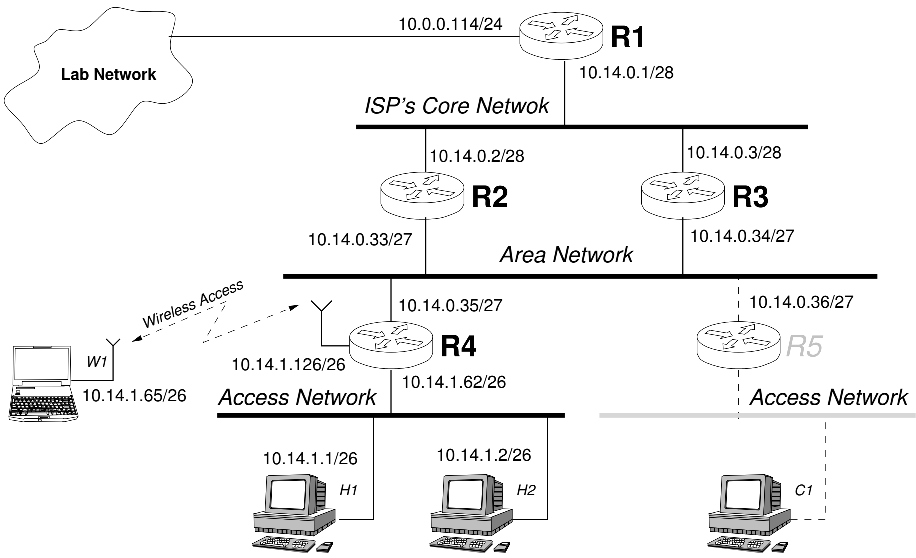

Figure 1: Network Layout

1 Address space

Our ISP has been given a `/16`[1] address space (10.14.0.0/16) which allows us to host not less than 2^{16} hosts, which is a lot as to worry about. Now the problem is how do we distribute them.

At first glance, it would seem very simple to just give each network possible a `/24` address space and go ahead with it, that would allow us to have 254 networks, with 254 hosts each, this is pretty good for our purposes, but I figured that a real ISP would like to distribute its addresses in a more efficient way (my ISP gives me a `/26` network mask, for instance). So it seem a nice exercise on subnetting which I write here.

2 ISP network topology

First I though I would think of myself as a real ISP, working in Sweden, trying to provide Internet services to the whole country. As stated in the documentation, `R1` is our Internet Exchange (IX) point. On the internal side of our IX there is another network where we will attach our own set of internal routers, that will come from several other places in the country. Although that's a bit fictitious (we wouldn't have a level 2 network that spans the whole country, but rather a set of point to point connections) we'll assume it is so, for teaching purposes. That network I called it "ISP Core Network", and in figure 1 we can see it is composed of core routers `R1`, `R2`, and `R3`.

[1]A `/16` network is a network in which the higher 16 bits of any of the network's host address are the same, that is, the network mask is 16 bits of ones followed by 16 bits of zeros: 255.255.0.0, and the lower 16 bits of the address define the host.

2.1 The ISP Core Network

The purpose of this network is to carry all traffic coming from all our ISP's routers. It will also be used as the network where all public servers (like web, DNS) will hang on. In our ISP case this will be one single computer (`R1`) having all this services, but there could be several dedicated servers for each type of service. Based on the assumption that very few computers would be on that network, I estimated that a `/28` network would suffice, giving us the possibility to have $2^4 - 2 = 14$ hosts on that network. If later we need more hosts on that net, we could opt for NAT or a [expensive] change in the topology. Table 1 shows the network data for the "ISP Core Network".

Attribute	Value
Network	10.14.0.0/28
Netmask	255.255.255.240
First host	10.14.0.1
Last host	10.14.0.14
Broadcast address	10.14.0.15
R1 IP address	10.14.0.1
R2 IP address	10.14.0.2
R3 IP address	10.14.0.3

Table 1: ISP Core Network data

2.2 The Area Network

Although with a confusing name, the "Area Network" gets its name from its function in the global schema of the ISP. Sweden has 25 *Landskap*, or regions, and it would probably be a good idea to have one Internet access node for each region, thus every user would be under a regional network. For highly populated regions there could be several access nodes, and underpopulated regions could share one same access node. For 25 regions we can do the job with a `/27` network, we would have enough access nodes for each region plus a few more as backup, or for highly populated regions, without forgetting some routers (like `R2` and `R3`). Table 2 shows the network data, and some assigned IP addresses.

Attribute	value	Router	IP address
Network	10.14.0.32	R2	10.14.0.33
Netmask	255.255.255.224	R3	10.14.0.34
First host	10.14.0.33	R4	10.14.0.35
Last host	10.14.0.62	R5	10.14.0.36
Broadcast Address	10.14.0.63		...

Table 2: Area Network data

2.2.1 Having fun with DNS

For DNS service purposes, and a bit for fun, we could provide names to each access node like `uppland01.fixme.lab`, `skane01.fixme.lab`, etc... Each access node could at the same time give specific names to each host like `h01.uppland01.fixme.lab` (fixed host), `w01.uppland01.fixme.lab` (wireless host), etc...

2.3 The Access Network

The "Access Network" is the network where the end users attach to the Internet using our services. In our case, the users connect to the Internet through an Ethernet or a wireless link, but it could access by xDSL, ISDN, POTS, etc... This network only concerns what's below `R4` (or `R5`... if we had more access nodes), and at the same time, `R4` is the selected DHCP server that will configure the IP address, network mask, default gateway, etc... for each end node or user host.

The question for the "Access Network" is how many users can each access node handle. This is a hard guess, but looking at my own ISP, as I have mentioned before, a good guess is to provide a `/26` IP address, thus each access node would serve a maximum of 61 users[2]. This is even acceptable for our humble purposes. Now, the next question is how to split that big pie that is our address space. We could reserve the `10.14.0.x` address space for "internal" use only. That is, only for routers and our own ISP infrastructure, leaving the rest (`10.14.[1...255].x`) as user or costumer IP address space.

Attribute	Ethernet IP	Wireless IP
Network	10.14.1.0	10.14.1.64
Netmask	255.255.255.192	255.255.255.192
Broadcast	10.14.1.63	19.14.1.127
First Host	10.14.1.1	10.14.1.65
Last Host	10.14.1.62	10.14.1.126

Table 3: Access Network Details

Table 3 shows the network data of the access network for which `R4` is its access node, whilst table 4 shows some proposed IP addresses for that access network.

2.4 Black holes

Notice that we have used some addresses in the range of `10.14.0.1` to `10.14.0.15` (being this last one a broadcast address), and then we move on using IP address range of `10.14.0.32` (being a network), to IP address `10.14.0.63` (being this the broadcast address). So, what happened to all the addresses in between? Well, they can still be used, don't worry, it is just that we couldn't just use them at that moment.

When we divided our address space, and made subnets, we could not take arbitrary contiguous blocks of addresses, as they are already logically divided. We can take chunks of 4, 8, 16, 32... addresses, but if we want a chunk of 64

[2]There are 64 addresses, but 1 is for the network, the other is the broadcast address, and finally another one is for the default gateway (`R4`)

Access type			
Ethernet		Wireless	
Host	IP	Host	IP
H01	10.14.1.1/26	W01	10.14.1.65/26
H02	10.14.1.2/26	W02	10.14.1.66/26
	...		...
H62	10.14.1.62/26	W62	10.14.1.126/26
R4	10.14.1.63/26	R4[a]	10.14.1.127/26

[a]Wireless access link

Table 4: `R4`'s Access Network

addresses, we need to get to the next free chunk of 64 addresses. Table 5 shows the different chunks in which the address space is divided.

As you can see, for the first network, we took a `/28` net, that is a chunk of 16 addresses (including net and broadcast). We started taking addresses from the beginning, so we got the range of addresses from `10.14.0.1` to `10.14.0.14`, plus the network address `10.14.0.0`, and the broadcast address `10.14.0.15`. Then, for the "Area Network", we wanted a `/27` net, that is a chunk of 32 addresses. This time we can not continue right after the last address we took and start from `10.14.0.16`. We can not do that because this chunk is of 16 addresses maximum (it can be chopped in chunks of 8 or 4 addresses too). We have to go to the 27 bits column, and the next chunk of 32 addresses free is the one that ends with 32, that is `10.14.0.32` (which in fact is the network, not an IP address).

At first it might seem that we're losing all those addresses, that go from `10.14.0.32` to `10.14.0.63`, but those possible networks can still be used for other purposes, like point to point links (which need a `/2` network), or we could even do a parallel structure of area networks, taking the next chunk of 32 addresses. . . , the possibilities are very flexible. Thus, this is not a waste of IP addresses, in fact, we've optimized them, if we compare with the case of using class C networks, as we would first have been tempted (see section 1 on page 2).

3 DNS name space

In some parts of the document, I've made suggestions about the names that could be used on our ISP's computers (see section 2.2.1 on page 4), table 6 on page 7 is a brief of the names and IP addresses discussed above. You have to append `.fixme.lab` at the end of the host names, to get an Internet accessible host name. Notice that it would be wise to set another DNS server on `R4` that would take care of the user host names.

Different sizes of subnets

24 bits	25 bits	26 bits	27 bits	28 bits	29 bits
0	0	0	0	0	0
					8
				16	16
					24
			32	32	32
					40
				48	48
					56
		64	64	64	64
					72
				80	80
					88
			96	96	96
					104
				112	112
					120
	128	128	128	128	128
					136
				144	144
					152
			160	160	160
					168
				176	176
					184
		192	192	192	192
					200
				208	208
					216
			224	224	224
					232
				240	240
					248

Table 5: Different sizes of subnets

Host	Proposed name(s)	IP address
R1	www, ns	10.0.0.114, 10.14.0.1
R2	hugin[a]	10.14.0.2, 10.14.0.33
R3	munin[b]	10.14.0.3, 10.14.0.34
R4	uppland01, ns.uppland01	10.14.0.35, 10.14.1.62
H1	h01.uppland01	10.14.1.1
W1	w01.uppland01	10.14.1.65

[a]A scandinavian mythical creature representing the *thought*

[b]Another scandinavian mythical creature representing the *memory*, and the name of the building where I live :)

Table 6: Host names and IP addresses proposed