Universiteit van Amsterdam

Networking and System Security

Lab Exercise 7

IP Addressing and BGP Routing

David van Erkelens, 10264019

Department of Computer Science
University of Amsterdam

November 26, 2012

Contents

1 Introduction

In this document, I'll note my findings regarding lab exercise 7 of the Networking and System Security course. This document will also be a playground for testing my LaTeX skills, hence the fancy layout of this document. I hope you'll enjoy the formatting of this document and, after all, the answers to the questions!

2 Task 1

IPv4 addressing

Using IPv4 address space `5.22.0.0/15`

1. (a) There are $2^{23-15} = 256$ subnets
   (b) Each subnet has $2^{32-23} = 512$ IP addresses
   (c) Since the network and broadcast IP addresses are reserved, each subnet can handle 510 hosts.

2. (a) The IPv4 address space `5.22.0.0/15` with a subnet space of `/23` gives us the following translation into bits:

   `00000101 0001011X XXXXXXXY YYYYYYYY`

   In which each X stands for a bit of the subnet-part of the address, and each Y stands for a bit in the host address part.

   Adding the zeros for the subnet part (the first subnet has number 0), gives us the following result:

   `00000101 00010110 0000000Y YYYYYYYY`

   The network IP address is the address in which each Y is a zero, giving us the following IP address in bits:

   `00000101 00010110 00000000 00000000`

This translates into the following decimal IP address:

`5.22.0.0`

(b) The broadcast IP address of a subnet is the address in which each Y from the explanation at (a) is an one instead of a zero. This gives us the following IP address in bits:

`00000101 00010110 00000001 11111111`

This translates into the following decimal IP address:

`5.22.1.255`

3. (a) The binary value of 129 (we start counting at 0, so we need 129 to calculate the IP address of the 130^{th} subnet) is

`10000001`

Putting this at the place of the X's of the IP address which we saw in (2a), gives us the following subnet space:

`00000101 00010111 0000001Y YYYYYYYY`

Again, we'll replace every Y with a zero to find the network IP address of the 130^{th} subnet:

`00000101 00010111 00000010 00000000`

Giving us a decimal IP address:

`5.23.2.0`

(b) Again, we'll replace every Y from (a) with an one to find the broadcast IP address of the 130^{th} subnet:

`00000101 00010111 00000011 11111111`

Giving us a decimal IP address:

`5.23.3.255`

3 Task 2

IPv6 addressing

Using IPv6 address space `2001:0db8:ff00:1100::/58`

4. (a) The total address range starts at

   `2001:0db8:ff00:1100::`

   And it ends at

   `2001:0db8:ff00:113f:ffff:ffff:ffff:ffff`

   (b) There are $2^{64-58} = 64$ `/64` subnets

   (c) Each subnet has $2^{128-64} = 2^{64}$ addresses.

5. The address range starts at

   `2001:0db8:ff00:1100::`

   Considering the `/58` range, numbering subnets starts at the `1100`, the 4^{th} hexadecimal number group. However, 58 is not dividable by 4 so creating the IP address of a subnet has to be done on a binary level instead of a hexadecimal level. Transforming the `1100` into binary gives us:

   `0001 0001 0000 0000`

   This is the part from the 49^{th} to the 64^{th} bit of the IP address. Keeping the `/58` into mind, we can start numbering our subnets from the 59^{th} to the 64^{th} bit of the IP address. So all the subnet numbering happens within the 4^{th} hexadecimal number group. Marking the space for subnet numbering, gives us the following binary view of this number group:

   `0001 0001 00XX XXXX`

The binary notation of 1 is `00 0001`. When we add this value to the number group above, the number group will look like this:

```
0001 0001 0000 0001
```

Which translates back to a hexadecimal value of `1101`. Putting this back into the original IPv6 address, gives us the IP address of the 2^{nd} `/64` subnet:

```
2001:0db8:ff00:1101::
```

6. With all the work done at (5), the only thing left to do to calculate the IP address of the 42^{th} is calculating the binary value of 41, which is

   ```
   10 1001
   ```

   Adding this to the binary view of the 4^{th} number group gives:

   ```
   0001 0001 0010 1001
   ```

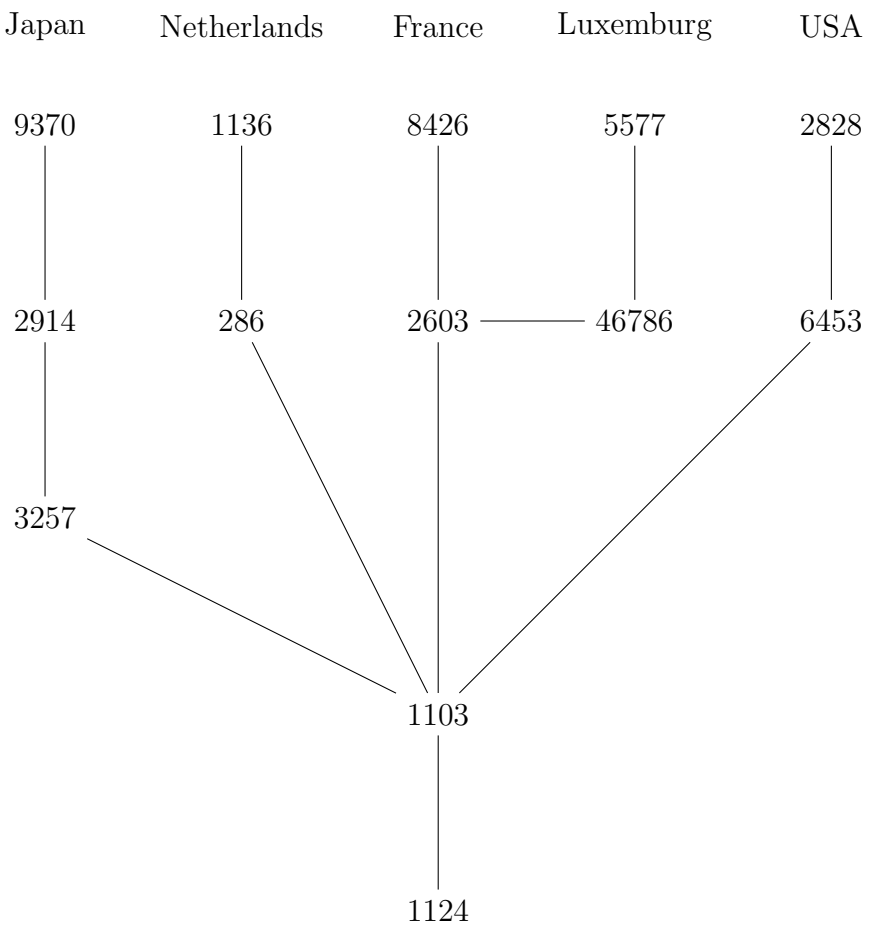
   Which translates back to a hexadecimal value of `1129`. Adding this to the original IP address gives us the IP address of the 42^{th} `/64` subnet:

   ```
   2001:0db8:ff00:1129::
   ```

4 Task 3

Find the path

7. LG Server BGP paths:



Used LG servers:
Sakura Japan (9370)
`http://as9370.bgp4.jp/`
KPN Netherlands (1136)
`http://netcollect.kpn.net/looking-glass/`
Claranet France (8426)
`http://noc.eu.clara.net/lg.php`
Root SA Luxemburg (5577)
`http://lg.as5577.net/`
XO Washington DC (2828)
`http://xostats.xo.com/cgi-bin/xostats/diagtool-pub`

8. AS name overview:

	AS Number	Country	AS Name
1	9370	Japan	SAKURA-B SAKURA Internet Inc.
2	2914	United States	NTT-COMMUNICATIONS-2914 - NTT America, Inc.
3	3257	Germany	TINET-BACKBONE Tinet SpA
4	1136	Netherlands	KPN KPN Internet Solutions
5	286	Europe	KPN KPN Internet Backbone
6	8426	Great Britain	CLARANET-AS ClaraNET LTD
7	2603	Norway	CLARANET-AS ClaraNET LTD
8	5577	Luxemburg	ROOT root SA
9	46786	United States	IPTRANSIT - IP Transit Inc.
10	2828	United States	XO-AS15 - XO Communications
11	6453	Canada	GLOBEINTERNET TATA Communications
12	1103	Netherlands	SURFNET-NL SURFnet, The Netherlands
13	1124	Netherlands	UVA-NL Universiteit van Amsterdam