COMP202-16B - Longest Prefix Matching

September 1, 2016

1 Introduction

This practical will reinforce concepts learned in the course around determining the longest matching prefix. You are supplied with some code that parses a file of prefix to AS mappings derived from publicly available BGP data.

You are required to complete the code so that it can do a longest prefix match lookup for addresses supplied on the command line, as well as print out the corresponding name of the AS. The code is commented in the places where it needs to be completed. The reference implementation is about 200 lines of code, and you are supplied with 100 of those lines.

Due to limitations of Java, the implementation you complete will be far from optimal. The easiest way to implement the prefix::match method is to break the address into four integers, and then mask the appropriate number of bits from each address integer and compare with the corresponding network address integer. I have supplied you an array of mask values for this purpose. This is roughly how you approach answering test questions on which routing entry is chosen, so should be reasonably natural to you.

2 Academic Integrity

The files you submit **must** be your own work. You may discuss the assignment with others but the actual code you submit must be your own. You must fully also understand your code and be capable of reproducing and modifying it. If there is anything in your code that you don't understand, seek help until you do.

You **must** submit your files to Moodle in order to receive any marks recorded on your verification page.

This assignment is due September 12th at 11am and worth 6% of your final grade. If the assignment is verified after the 12th, it is worth 1% less for each lab session after the 12th where it could have been verified.

3 Materials

You are provided with an IP2AS mapping file that was derived from public BGP data. Briefly, I processed archives of BGP raw data files that contained routes (prefixes and paths), i.e.:

```
3303|7575|38022|681 130.217.0.0/16
2914|7575|38022|681 130.217.0.0/16
62567|2914|7575|38022|681 130.217.0.0/16
```

and extracted the unique *origin ASes* belonging to each prefix. For example, in the above example, the origin AS (the last AS in the path, likely to be the AS that announced the route) is 681. I condensed the raw paths into a file where each line contains a unique prefix and all origin ASes associated with the prefix. Where there is more than one origin AS, ASes are separated with an underscore character.

```
130.217.0.0/17 681
130.217.0.0/16 681
130.217.128.0/17 681
192.107.171.0/24 681
192.107.172.0/24 681
205.204.15.0/24 3356_3549
```

You are also supplied with an asnames file, which associates an AS number with a name.

```
1 LVLT-1 - Level 3 Communications, Inc.,US
2 UDEL-DCN - University of Delaware,US
3 MIT-GATEWAYS - Massachusetts Institute of Technology,US
```

The format is the ASN, followed by free-form text naming or describing the ASN.

4 Requirements

1. Do not load any prefix less specific than a /8, or more specific than a /24.
2. Ensure you handle the case where there is no prefix covering an IP address (i.e. no mapping from an IP to an ASN).
3. Ensure you handle the case where there is no name for a given ASN.

5 Correctness

If your program produces output like the following, you've probably implemented it correctly for answering the verification questions.

```
$ java ip2as ../20150701.ip2as ../asnames.txt 128.30.2.155 10.110.8.71 1.0.192.1 205.204.15.1
128.30.2.155 128.30.0.0/15 3 MIT-GATEWAYS - Massachusetts Institute of Technology,US
10.110.8.71 : no prefix
1.0.192.1 1.0.192.0/21 23969 TOT-NET TOT Public Company Limited,TH
205.204.15.1 205.204.15.0/24 3356 LEVEL3 - Level 3 Communications, Inc.,US
205.204.15.1 205.204.15.0/24 3549 LVLT-3549 - Level 3 Communications, Inc.,US
```

VERIFICATION PAGE

Name: _____

Id: _____

Date: _____

Note: this practical is due Monday 12th of September at 11am, and the code submitted to the Moodle assignment page. It will be marked out of 6 and is worth up to 6% of your final grade. If the assignment is verified after the 12th, it is worth 1% less for each lab session after the 12th where it could have been verified.

1. Explain to the tutor or demo the structure of your program, and allow them to read and test your code.
2. Show the demo or the tutor the output of your program when you run java ip2as 127.0.0.1
3. What IPv4 address does www.amazon.com resolve to? Which AS announces the corresponding prefix?
4. What IPv4 address does www.netflix.com resolve to? Which AS announces the corresponding prefix? What do you think is happening here?
5. Which AS announces 130.217.0.0/16 in the ip2as file?
6. Which prefixes are the longest matching prefixes for i.waikato.ac.nz and sorcerer.cms.waikato.ac.nz? Is it possible for any address to match the 130.217.0.0/16 prefix? Explain your answer.
7. For three different NZ ISPs, (e.g. spark, vodafone, lightwire, snap, orcon, callplus) identify an IP address (using DNS) advertised by each. For each, list the longest matching prefix, ASN, and AS name involved.