School of Computing, University of Leeds

COMP2221
Networks

Worksheet 1: Java refresher, ports, and IP addresses.

This worksheet covers the preliminary material including ports and IP addresses, right up to the Java network programming that started in the middle of Lecture 5. There is also a Java example with no network features, for those of you wishing to brush up on their Java as soon as possible.

Questions

1. This question is intended to provide a refresher in Java for anyone who has not done much Java programming for a while. It does not involve any network programming. Download and inspect the code `ParseCmdLineArgs.java` from Minerva, then modify the class to perform the following functions:

   (i) If there are no command line arguments, *i.e.* if `args.length==0`, the class should print a message to that effect, and then quit.

   (ii) Otherwise, store the command line arguments in a private string array.

   (iii) Print to `stdout` (*i.e.* using `System.out.print` and/or `System.out.println`) each command line argument on a separate line.

   (iv) In addition, if the argument contains any dots ('`.`'), the message '`(may be a hostname)`' should be output on the same line after the argument.

   (v) Furthermore, if the argument contains exactly 3 dots, then the message '`(may be an IPv4 address)`' should also be output on the same line.

   Note that `www.google.co.uk` should give messages for part (iv) and part (v), so our simple idea to identify an IPv4 address by counting the number of dots does not always work. We will see a more robust way of parsing hostnames the last half of Lecture 5.

2. *[Lecture 3]* Either log into a school machine, or use your own Unix device if you have one, open up a shell, and examine the `/etc/services` file that was covered near the middle of Lecture 3. For instance, you may like to use the `more` command,

   ```
   more /etc/services
   ```

   Why are there two port numbers for each service? (If you are using a Mac you may see a third port number for DDP, but that is not covered by this module).

   You can also search for specific port numbers by using `grep`. For instance, to see all occurrences of the number 443, type

```
grep 443 /etc/services
```

This will return the lines corresponding to port 443 for `https`, but will also return other ports that include the digits 443, *e.g.* 1443, 2443 *etc.* Try this out for all of the port numbers mentioned in Lecture 3, and confirm they correspond to the expected service.

3. *[Lecture 4]* Log into a Unix machine and open up a shell as before, then try the following command line tools related to networking.

   (i) `nslookup`, `host` and `dig`: These tools take a hostname as an argument, and query the DNS to return the IP address and other information. Try each of them on `www.comp.leeds.ac.uk` and see what information they give you. Now try the hostname `www.maths.leeds.ac.uk` – why are some of the numbers in the IPv4 address the same as the previous address? Can you find any other hostnames with a similar relationship?

   Also try out some hostnames that return IPv6 addresses, such as
   `> nslookup dns6.leeds.ac.uk`
   or
   `> nslookup www.facebook.com`
   You may notice something slightly amusing about the IPv6 address for the last example.

   (ii) `ping`: This checks if a host can be reached, and returns the Round Trip Time (RTT) for the message to travel there and back. It uses the Internet Control Message Protocol (ICMP) that we will briefly look at in Lecture 17. Try `ping` with a hostname you know exists (*e.g.* `www.leeds.ac.uk`), and one that does not exist (*e.g.* `www.wibble.com`).

   (iii) `traceroute`: Similar to `ping`, but returns reachability information for *all* routers from `localhost` to the destination, and averages over 3 packets to get some idea of fluctuations. Not all routers respond to the messages sent by `traceroute`; these are displayed as asterisks. As we will see in Lecture 17, `traceroute` also uses ICMP.

4. *[Lecture 5]* Answer the following questions about IPv4 and IPv6 that was covered in the first half of Lecture 5.

   (i) Expand the following IPv6 addresses to their full, 16-byte form.
   (a) `2000:120:a00::10`
   (b) `ff00::1`

   (ii) Contract the following IPv6 addresses to their short form.
   (a) `2001:0910:00c0:0000:0000:0000:0000:0002`
   (b) `2002:0000:0000:00c0:1234:ffff:4321:0000`

   (iii) What is the first and last address in each of the following ranges? How many IP addresses belong to each range (for (c) you can leave your answer as a power of 2)? For (a), re-write the range using the same CIDR notation as in parts (b) and (c).
   (a) `129.11.26.*`
   (b) `129.11.128.0/18`
   (c) `ff00::/8`