

Java and Networking

David Malone

August 1996

1 URL's

Universal Resource Locators, the way of saying where to get information on the net. They are most commonly of the form:

```
proto://host.domain.name:port/file/name.blah
```

The parts are as follows :

proto Is the protocol used to get the data eg `http` or `ftp`.

host.domain.name Is the name of the machine to get it from, this can be a name (eg. `salmon.maths.tcd.ie`) or a IP address (eg `134.226.81.11`). The Java class for dealing with hostnames is `InetAddress`. You can get a `InetAddress` for a hostname by saying `InetAddress = InetAddress.getByName("salmon.maths.tcd.ie");`

port Is the number of the port to connect to on the remote machine. Ports aren't little connectors on the back of a computer in this case, think of them like the frequency on a radio.

file/name.blah Is the name of the file on the other machine.

2 Java Applet Security

Java applets are supposed to live in a cage, so if you load an applet from across the net it can't do anything bad to your computer. This has two major effects on networking :

- You may not open a network connection to any machine, except the machine you (the applet) came from.
- You may not listen for or accept network connections.

For a more full list of applet security restrictions see Java in a Nutshell.

3 The URL Classes

The `URL` class provide the basic interface which you would most commonly use in writing applets. You can create an instance of a `URL` by saying :

```
URL my_url = new URL("http://www.maths.tcd.ie/pub/images/schoola.gif");
```

You can then use the information by saying :

```
Image school = (Image) my_url.getContent();
```

The protocol you specify in your `URL` and the content of the `URL` must be understood by Java. By default Java understands `http` and `file` protocols and GIF's, JPG's and plain text contents. You can define new 'handlers' for these so it understands more types. See Exploring Java's networking section.

4 The URLConnection Class

The `URLConnection` class provides you with more control than the `URL` class. It can provide you with a Stream type connection to the machine you are getting the `URL` from. It can also provide the following information about what you have asked for :

- Content Type
- Content Length
- Expiration Date
- Content Encoding

It also allows you to control the use of a caching proxy, and do conditional gets etc. The `get.java` program provides a simple example of using the `URL` and `URLConnection` classes.

5 Lower Level Networking

Lower level networking is available in Java. The ‘sockets’ of TCP/IP are available to you. We have a choice of reliable and unreliable communication. If you send information reliably then you know it will get where its going in the order you sent it. If you send things unreliable then you don’t know this. Also we have a choice of stream or packet based communication. If you choose packet based, then you send out little chunks of information - each separately from the others. If you choose stream based you get something that looks like a file that you can write to or read from.

The `DatagramSocket` and `DatagramPacket` classes provide us with a packet based, unreliable connection. The `Socket` class provides with reliable or unreliable stream connections.

The `finger.java` program provides a simple example of using the `Socket` class. Note Java does not provide a way of turning common service names into port numbers - hence the 79 in the socket constructor.

6 Blocking IO and Threads

Usually if you try to do input or output there is a delay before the computer can do it — it may be waiting for data from the net, or for disks to finish reading. While this is happening your program is ‘Blocked’ and can do nothing.

To get around this you can have a program with several threads, so while one is blocked waiting for disk or information from the net another thread can run. This is one of Java’s strengths.

The `Chat` and `ChatServer` java programs demonstrate both the use of threads in networking and the unreliable packet based networking.

7 Other Stuff

The `ChatServer` uses Java’s `Vector` class, which provides an interesting alternative to linked lists. Java’s event driven GUI environment could be extended to include event driven networking — that would be cool.