CM50123 Networking: Assignment 1

djm64

November 14, 2018

0.1 Software

This chapter concerns the tools created and used to complete the coursework. It covers the work done outside the specification of the coursework and gives reasons as to why things were done the way they were.

For this coursework, I tried to use as few online tools as possible. This was in most part for two reasons

1. Online tools give little or no explanation on how they work (or if their outputs are correct)
2. I felt it was a good opportunity to try out a language I was unfamiliar with (Python 2.7)

The software pipeline has 3 stages:

- Data Collection
- Data Processing and Formatting
- Data Presentation

0.2 Data Collection

The main tool used for the Data Collection is `traceroute`. `traceroute` is a network diagnostic tool that displays the route taken by packets accross a network. Before we can use `traceroute` however we first need a list of targets. Since the purpose of this coursework is to find out how the Universities in the UK and Canada are connected to JANET and the Internet respectively, it makes sense to use UK and Canadian University websites as the targets.

0.2.1 Gathering URLs

Gathering the URLs for univeristy websites was a simple enough task. The websites https://www.uk250.co.uk/university/ and https://www.univcan.ca/universities/member-universities/ provide a list of University websites that can be used. The websites are scraped for URLs with a simple python script and stored in a text file to be used later.

The URLS generally aren't in the same format or in a format that can be used with `traceroute` so they must be modified. The standard `http://subdomain.domain.tld/category/sub-category` formatting of URL must be stripped down to `domain.tld` which is easily achieved with Python's `string.split()` method.

0.2.2 Performing traceroute

With a list of websites we can now actually use `traceroute`. `traceroute` has only has one required parameter which is the IP address or destination host. Since there over 200 websites between the UK and Canada I opted to write a bash script to perform the `traceroute`. The code below shows how this is done

```
cat country.txt | while read line; do
traceroute  $line | tail -n+2 | awk '{ print $2 "," $3 }'|\
sed -e "s/(//" -e "s/)//"> "country/traceroutes/$line.txt"
done
```

This simply goes through the text file with all the host addresses, performs `traceroute` and outputs the hostname and IP addresses of each of the hops along the way into a file. This is performed for each website in both countries resulting in a lot of .txt files containing traceroutes.

0.3 Data Processing and Formatting

With the `traceroutes` obtained, we are nearly ready to begin trying to work out how JANET and Canada's Universities are connected.

The information in the .txt files isn't very useful yet as it only contains the name of the responding router and its IP address.

0.3.1 Processing .txt Files

The first step in getting more information for each of the `traceroutes` is first deciding what information is accessible and what we actually need. There are a variety of tools online that claim to offer information about a host such as the owner, city, ISP, coordinates, etc. These tools will be discussed later and for now it will be assumed that a generic tool is being used that can give us correct information.

With this tool in hand we can now start to build a more detailed representation of each of the `traceroutes`. Since Python is being used, the `dictionary` which is simply an associative array (with key-value pairs). Considering each hop of the `traceroute` we can build a `list` of `dictionary` to represent the `traceroute`. The `dictionary` used is shown below:

```
dict = {'Hostname' : 'name',
        'IP'       : 'ip',
        'City'     : 'city',
        'ISP'      : 'isp',
        'Org'      : 'org',
        'latitude' : 'lat',
        'longitude': 'long'}
```

For each .txt file containing a `traceroute` we go through each line, use the generic tool to obtain the information that isn't known and then store this in a `list` of `dictionary`. A useful feature of `dictionary` is that they can be converted to and from `JSON` easily. For this reason they are saved as a JSON file once the `list` is built.

0.3.2 Processing .JSON Files

The JSON files now contain all the needed information in an easy to access and use format. However there are still some small adjustments that can be made to improve them.

Fixing Hostnames

The Hostnames we have can be modified to make them easier to work with. Considering only the UK, we can can consider each host as either a part of JANET or not. An example JANET host is `ae22.londpg-sbr2.ja.net` while an non-JANET host could be `ae60-0.lon04-96cbe-1a.ntwk.msn.net` (owned by Microsoft).

Since we are separating the hosts into JANET and non-JANET we can simplify the JANET host names. All JANET routers are in theory owned by Jisc so we can filter the hosts by checking if `host['Org'] == 'Jisc Services Limited'`. If it is, we can strip away some of the less important information so `ae22.londpg-sbr2.ja.net` can become `londpg-sbr2`. It might seem counterintuitive to want to have less information but in practice we don't need to know if the route is through `ae22.londpg-sbr2` or `ae25.londpg-sbr2`, we just need to know that it goes through `londpg-sbr2`. non-JANET hosts simply have "nonjanet: " prepended to the existing host name.

For the Canadian files, simlar processing is done.

Building a List of Hosts

After modifying the JSON files to our liking we can now build a list of hosts. This is done simply by going through each JSON file, checking if the host is already in the list and if not adding it.

We now have a list of every host passed through on the `traceroutes` to the Universities.

0.4 Data Presentation

With all the data gathered and processed, it can now be visualised.

0.4.1 Visualising the Data

I chose 3 different visualisations to give an indication of the topology of both sets of networks

- **Host Locations**: This shows the locations of all the hosts passed through.
- **Traceroute Paths**: This shows the individual path for each of the traceroutes to each of the universities
- **Heatmaps**: This gives a heat map for the regions which shows how often each host is visited

0.4.2 Creating the Visualisations

The visualisations were created using `gmplot` which is a `matplotlib` like library that generates the HTML and JavaScript to render data on top of Google Maps.

`gmplot` needs latitudes and longitudes to add to the map. Since latitudes and longitudes are a part of are JSON file for each traceroute we can simply iterate through them and add them to a list which can then be used by gmplot

0.5 GitHub

The code and visualisations are available at https://github.com/donalj/Networking-CW1. The code is not commented and gives no explaination on how to run it. The UK visualisations are available in the ”maps” folder and the Canadian visualisations are available in the ”Canada/maps” folder.

Chapter 1

Interpreting traceroute outputs to discover the shape of some of the Internet

This chapter is concerned with purpose (a) in the coursework specification. It will explain the structure and topology of the JANET network and the Canadian Universities

1.1 JANET

The JANET network is a high speed network for use by UK education and research communities. It is owned and operated by Jisc. The service is split into regional networks which then provide Universities, Colleges and Schools nearby.

1.1.1 Regions

The regional networks in JANET are as follows:

- Cumbria And North Lancashire
- East of England
- East Midlands
- London
- North West
- North East
- North East Scotland
- Northern Ireland
- South
- South East Scotland
- South West
- South West Scotland
- Thames Valley
- West Midlands
- Yorkshire and Humberside

1.1.2 Hosts

Each region has its own set of hosts or routers, usually named `x-y` where `x` is a 4 letter city identifier sometimes followed by a 2 letter university identifier. I am unsure of what `y` refers to but I assume it is the type of router. A few example hosts would be:

- University of Bristol, Bristol becomes `brisub-rbr1`
- Herriot Watt, Edinburgh becomes `edinhw-rbr2`
- Dundee becomes `dund-ban1`

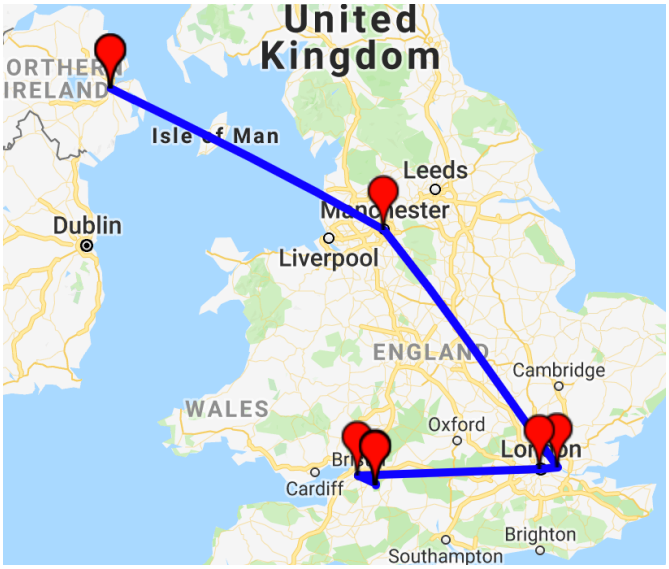
There are also hosts with more specific names such as `imperial-college` or `university-of-bradford`.

From my tests, I identified 121 unique hosts in the JANET network. These can be seen at https://github.com/donalj/Networking-CW1/blob/master/janet_servers.txt. There are undoubtedly more routers that were not found but with this list, an accurate description of the network can be seen.

1.1.3 Example traceroutes

Example visualisations of traceroutes are shown in Fig.??

(a) Traceroute visualised from Bath to https://www.abdn.ac.uk/



(b) Traceroute visualised from Bath to https://www.qub.ac.uk/

Figure 1.1: Example traceroutes

From the traceroutes shown and the others (available at https://github.com/donalj/Networking-CW1/tree/master/maps) it can be seen that from Bath, the packets must first go to Bristol before travelling elsewhere. London is almost always travelled through except in some circumstances such as travelling to Exeter seen in Fig.1.2

Figure 1.2: Traceroute from Bath to https://www.exeter.ac.uk/

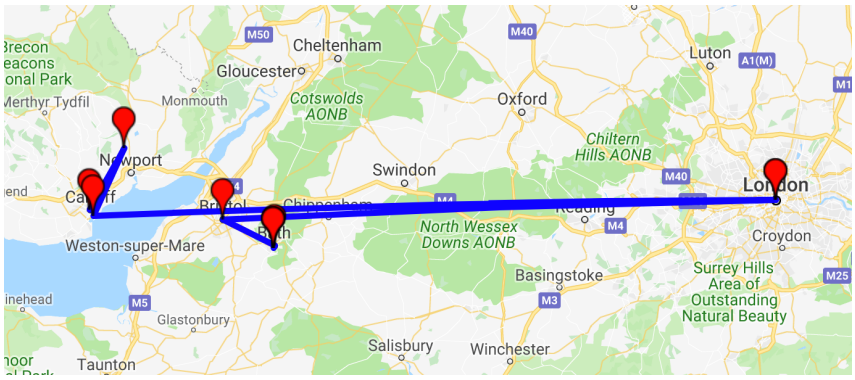


Figure 1.3: Traceroute visualised from Bath to https://www.southwales.ac.uk/

Some requests also seem to take roundabout routes to their destination such as in Fig.1.3 which goes all the way to London just to go back on itself to reach Wales.

1.1.4 Heatmap

With all the traceroute information available it is possible to create a heatmap to see which hosts are used the most. This is simple as we simply need to count how many times each host is passed through. The resulting heatmap in Fig.1.4 gives some insights into how often each of the nodes are used. Bath and Bristol are the most commonly used hosts which makes sense as the traceroute begins in Bath and must go through Bristol. London is also commonly used and is clearly the central "hub" for all the other hosts. Manchester and Liverpool are the only other notable locations.

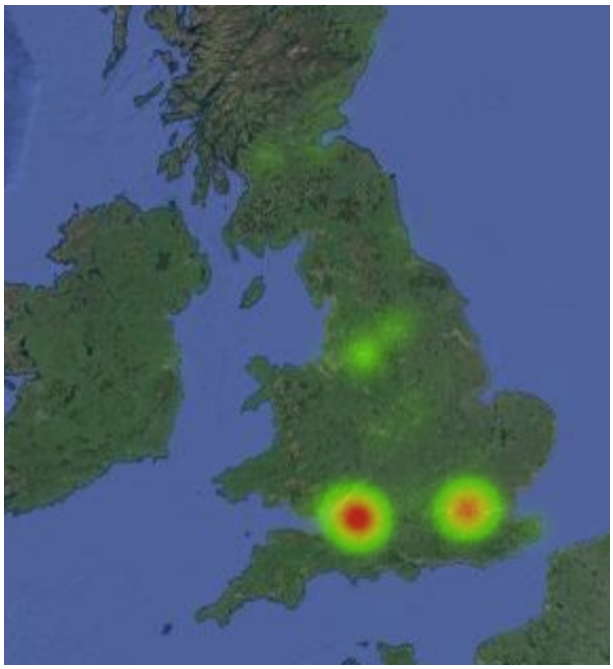


Figure 1.4: Heatmap of the UK servers

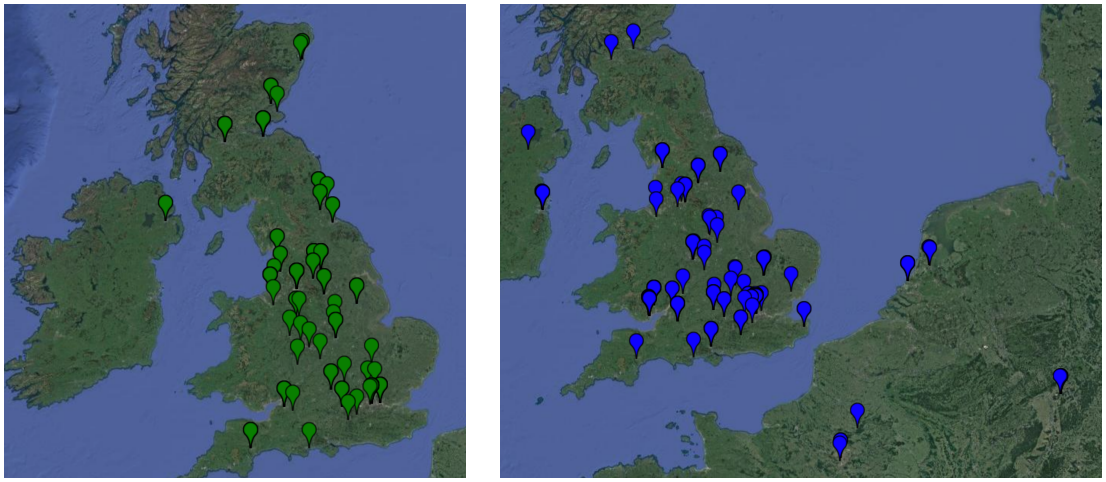
1.1.5 Topology

The location of the 121 JANET servers found are shown in Fig.1.5a:

1.1.6 Additional Comments

The host `manckh-sbr2` is interesting as in the traceroutes to Northern Irish Universities, it is the last in the mainland. This indicates the undersea connection between Great Britain and Northern Ireland goes from Manchester to Belfast.

The topology is as accurate as we could achieve with the tools available. The JANET hosts are all registered as having the same address in London. To get around this we used the first four letters as a (hopefully correct) indicator of the city the host is in. This was then used to find the lat/long of the city and update the host information with this. For the some hosts, the correct information was not easily found/wrong entirely and so these hosts were updated manually. This will be discussed in greater detail later.



(a) Map showing the location of JANET hosts in the UK

(b) Map showing the location of non-JANET hosts in the UK and Europe

Figure 1.5: Hosts in the UK

The JANET hosts are not the only hosts that are visited by the `traceroutes`. These are shown in Fig.1.5b

1.2 Canada

Before discussing how the Canadian Universities and Canarie are connected to the Internet, it is worth talking about how the UK is connected to Canada.

1.2.1 GÉANT

GÉANT connects the national research and education networks accross Europe. From the traceroutes to Canada, GÉANT appears to be the bridge between JANET and Canarie.

```
janet.mx1.lon.uk.geant.net
canarie.lon.uk.geant.net
```

The route also goes through Cambridge and Paris. The heatmap in Fig.1.6 demonstrates this.

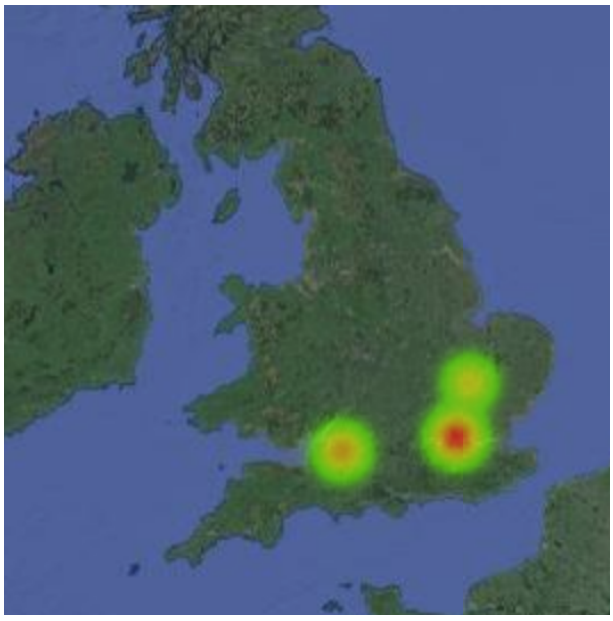


Figure 1.6: Heatmap of traces leaving UK

1.2.2 Canarie

Canarie is the Canadian equivalent of JANET. It differs in that it works in tandem with 12 partners. The process for discovering hosts was similar to JANET in that the hostname will contain ”canarie” or the organisation will be ”canarie” allowing the hosts to be identified easily.

Hosts

The hosts belonging to Canarie discovered and their assumed locations are:

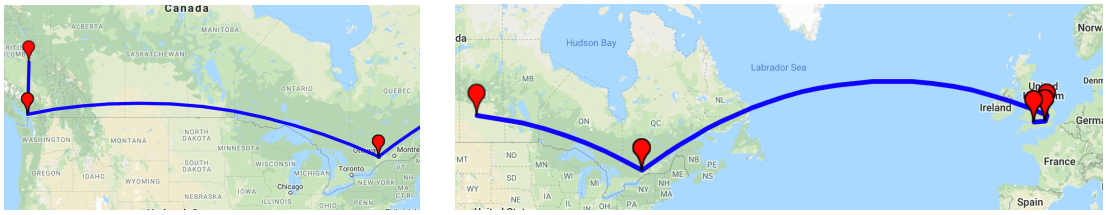
- hlfx1rtr1 located in Halifax
- clgr2rtr1 located in Calgary
- edmn1rtr1 located in Edmonton
- sask1rtr1 located in Saskatoon

- vncv1rtr1 located in Vancouver
- wnpg1rtr1 located in Winnipeg

Partners

The partners were more difficult to find. I initially went through any interesting hostnames that occurred regularly and found what organisations owned them and if they had any relation to Canarie. From the traceroutes performed I found the following partners.

- **ACORN-NS**: gigapop-gw.acorn-ns.ca
- **ACORN-NL**: mun.acorn-nl .ca
- **BCNET**: cr1-bb3900.vantx2.bc.net
- **SRnet**: srnet-reg.srnet.ca
- **MRnet**: uofm-mrrouter.mrnet.mb.ca
- **Cybera**: edm-mx-r.cybera.ca



(a) Traceroute visualised from Bath to https://unbc.ca/

(b) Traceroute visualised from Bath to https://uregina.ca/

Figure 1.7: Example traceroutes

1.2.3 Combining Partners and Host

With Canarie hosts and the partners it is possible to get a general idea of how the Canadian network is structured. Unlike with JANET, I do not have the automatic visualisation tools and the tools available online are not suitable. Instead I created a representation by hand.

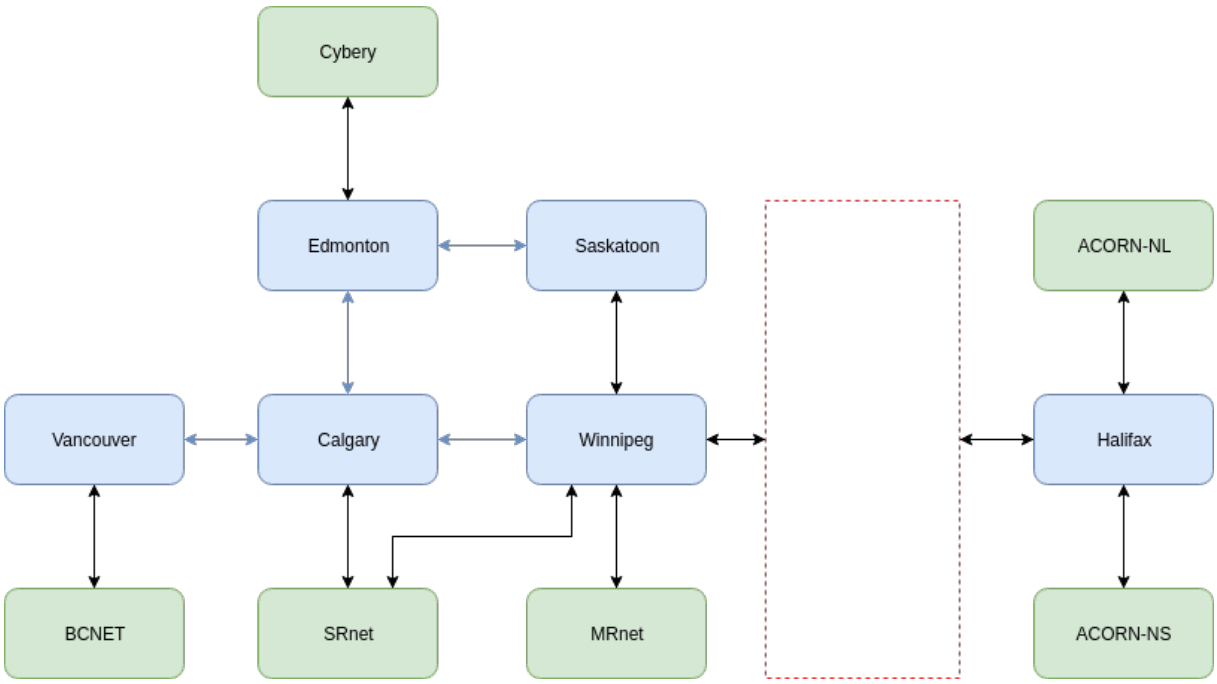


Figure 1.8: Structure of Canarie (blue) and Partners (green) network based on performed traceroutes

Chapter 2

Tools

This chapter discusses the tools that are available and their limitations.

2.1 Traceroute

For this coursework, the built in Linux `traceroute` was deemed suitable as it is easy to use and work with the outputs as they are local. Some alternatives were considered and there are numerous tools both available to use online or to download.

2.1.1 mtr

MTR (My traceroute or Matt's traceroute) is a program which combines the functionality of traceroute and ping. This program continuously probes the host and updates the traceroute. There is no need for dynamic traceroutes and doing since bulk traceroutes were done, using this tool made no sense. It was however useful in ensuring validity of the traceroutes.

2.2 Host Information

Traceroute only provides us with a Hostname and an IP address. This doesn't provide sufficient information to perform the tasks required. There are various IP databases and services which can provide more information.

2.2.1 ipstack

`ipstack.com` is one of the world leading IP to geolocation APIs. Their IP database is integrated with a series of large ISPs which provides them with up to date and accurate information.

ipstack was initially used to gather the host information through its API which returns a JSON object containing all the values needed. The main drawback of ipstack is that it is a paid service with limited free features. You can perform up to 10000 requests per month for free. This was problematic as with hundreds of universities each with its own traceroute results in thousands of requests. For this reason an alternative was found.

2.2.2 ip-api

`ip-api.com` is similar to ipstack in that it is an IP to geolocation API. It is free to use but is limited to 150 requests per minute. Since gathering Host Information would not be done regularly, doing the thousands of requests needed at 150/min is negligble.

2.2.3 keycdn

https://tools.keycdn.com/geo is another IP to geolocation API. It has a a request limit of 3/s or 180/min which is higher than ip-api. The JSON response formatting however is not as nice so this was not used.

2.2.4 Problems with gathering Host Information

The information provided by these tools while useful is not always correct. This is easily seen in JANET and Canarie hosts which are all thought to be in London. This results in incorrect visualisations as the coordinates given do not match the actual location of the hosts. This is the case for many hosts and an example can be seen in Fig.2.1.

The solution to this is to try and gather more information from the hostname which often includes the city. Doing this was only feasible for JANET due to the time constraints and the consistent naming schema of the JANET hosts.

Figure 2.1: An example of a host with incorrect information. The final destination should in London but is somewhere in Germany. The hostname xe-5-2-1.cr0-lon9.ip4.gtt.net contains lon but GTT Communications Inc. has its location as somewhere in Frankfurt, Germany

2.3 Visualisation

Visualising the results makes it easier to understand how the networks are connected together. The available tools online do not take into consideration the problems mentioned in 2.2.4

2.3.1 geotraceroute

https://geotraceroute.com/ provides a traceroute visualisation tool. It allows you to choose from a list of sources and enter your own destination. It shows the route on a 3D earth.

The main problem with this is that you cannot select the source destination.

2.3.2 Traceroute Mapper

Traceroute Mapper available at https://stefansundin.github.io/traceroute-mapper/ takes a traceroute output and draws it over Google Maps. This tool would me more than adequate if the problems in 2.2.4 did not exist.

2.3.3 My Solution

My solution is similar to Traceroute Mapper but corrects the problems in 2.2.4 and also gives the options to produce heat maps and show individual hosts.

2.4 Janet Looking Glass

http://lg.ja.net is a looking glass tool provided by Jisc. It provided useful information on the JANET hosts and could realistically used for more.

2.5 Conclusion

The tools available online are in the most part, very lacking. The online databases do not contain correct information and the visualisation tools do not provide enough freedom.

Another problem encountered was the poor information available from the JANET website. Many pages reference PDFs that do not exist, are missing information they claim to contain and a multitude of other issues.