The UNIX/Linux Operating System

Networking/Internet

Claude Cantin (claude.cantin@nrc.ca)
http://www.nrc.ca/rpso/scsg

Research Computing Support Group
Information Management Services Branch
National Research Council

May 27, 2018

This page intentionally left blank.

This document was produced by Claude Cantin of the National Research Council of Canada. Reproductions are permitted for non-profit purposes provided the origin of the document is acknowledged.

Claude Cantin
National Research Council of Canada

History of printing:

Date	Copies
March 2003	200
March 2001	200
June 1999	200
November 1997	200
July 1996	200
November 1995	150
March 1995	150
February 1994	150
October 1993	100
August 1993	75
February 1993	75
November 1992	35
September 1992	40
February 1992	50
December 1991	50
April 1991	50
September 1990	40
January 1990	40

Table 0.1: Printings.

Contents

List of Tables

List of Figures

Chapter 1

Networking/Internet

This section is devoted to network communication between users. It will start by introducing the largest network in the world, the `Internet`.

It will then introduce a number of tools used to navigate the local network as well as the larger Internet.

1.1 The Internet

1.1.1 A Network of Networks

The Internet network is a hierarchical network of thousands of networks involving millions of host computers and tens of millions of users from over 100 countries. In March 1997, the one-millionth `domain name` was registered (from Internet World, October 1997 issue) – `nrc.ca` is considered as one registration.

ONet, BITS (Bell Internet Transit Services – formerly known as CA*net), CA*net 2 and CA*net 4 are three such networks.

NRCnet

NRC has a number of local networks. Most buildings are individually networked internally, then linked to other buidings within the same campus. The 3 Ottawa campuses and other locations across the country are in turn linked together.

Figure 1.1 shows the topology of NRC's Montreal Road network, and how it connects to the other two Ottawa campuses (Sussex Drive, and Uplands) as well as both the general Internet, and CA*net 4.

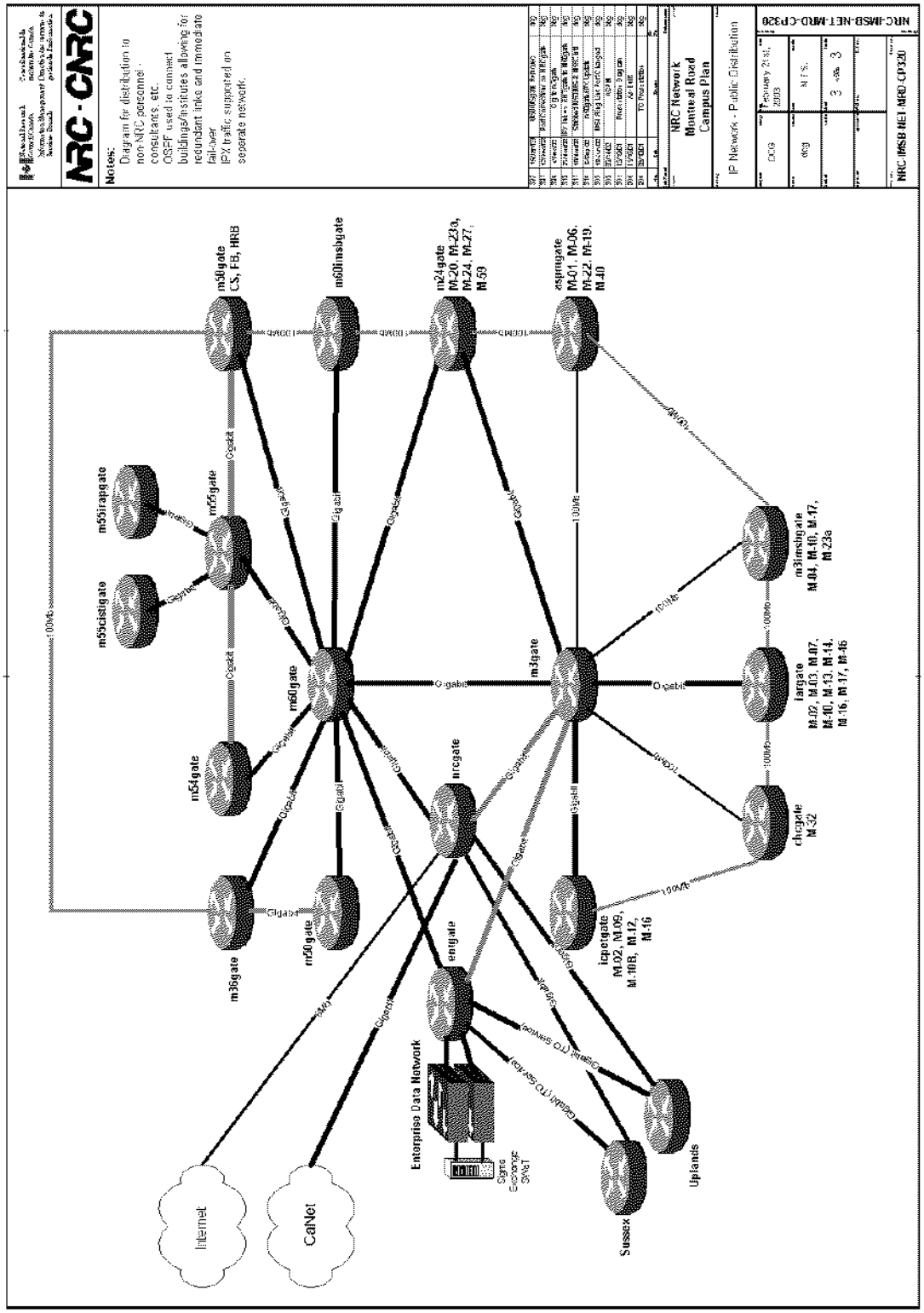

Figure 1.1: Network Connectivity of NRC/National Capital Region

Figure 1.2 shows how NRC connects all the regional Institutes together.

ONet

ONet Networking is the non-profit ISP (Internet Service Provider) mostly made out of Ontario educational and research institutions. It was the first public Ontario-based regional network (1988).

Prior to 2003, NRC's WAN was connected to ONet. For direct access to CA*net 2 and CA*net 4, ONet has a GigaPOP in Ottawa. This provides Ottawa ONet members direct access to CA*net 2/4 members.

Many Ottawa Internet sites are connected to ONet using that same router/GigaPOP.

More specific information on ONet may be found at

`http://www.onet.on.ca`

CA*net/BITS

In the early 1990s, all provinces had their own regional networks, the same way Ontario had ONet. The connections between those networks was handled by `CA*net` (regional networks are listed in Table 1.1).

On April 1, 1997, BITS (Bell Internet Transit Service, managed by Bell Advanced Communications) took over CA*net's functionality.

Soon after, a new project, CA*net 2 was officially launched (more information on CA*net 2 below).

Figure 1.3 (courtesy of the CA*net staff, 1995) shows the topology of CA*net, as it was before it became BITS.

As with ONet, more information on BITS may be found at URL

`http://www.bacits.bell.ca`

BITS is in turn connected to the American national backbone, through various connections, giving us access to international networks.

CA*net 2

The second main CANARIE (Canadian Network for the Advancement of Research, Industry and Education) initiative, CA*net 2, soon followed as the replacement Research and Development national backbone network. It was based on the ATM (Asynchronous Transfer Mode) and SONET facilities operated by Canada's Telecommunications carriers. Average

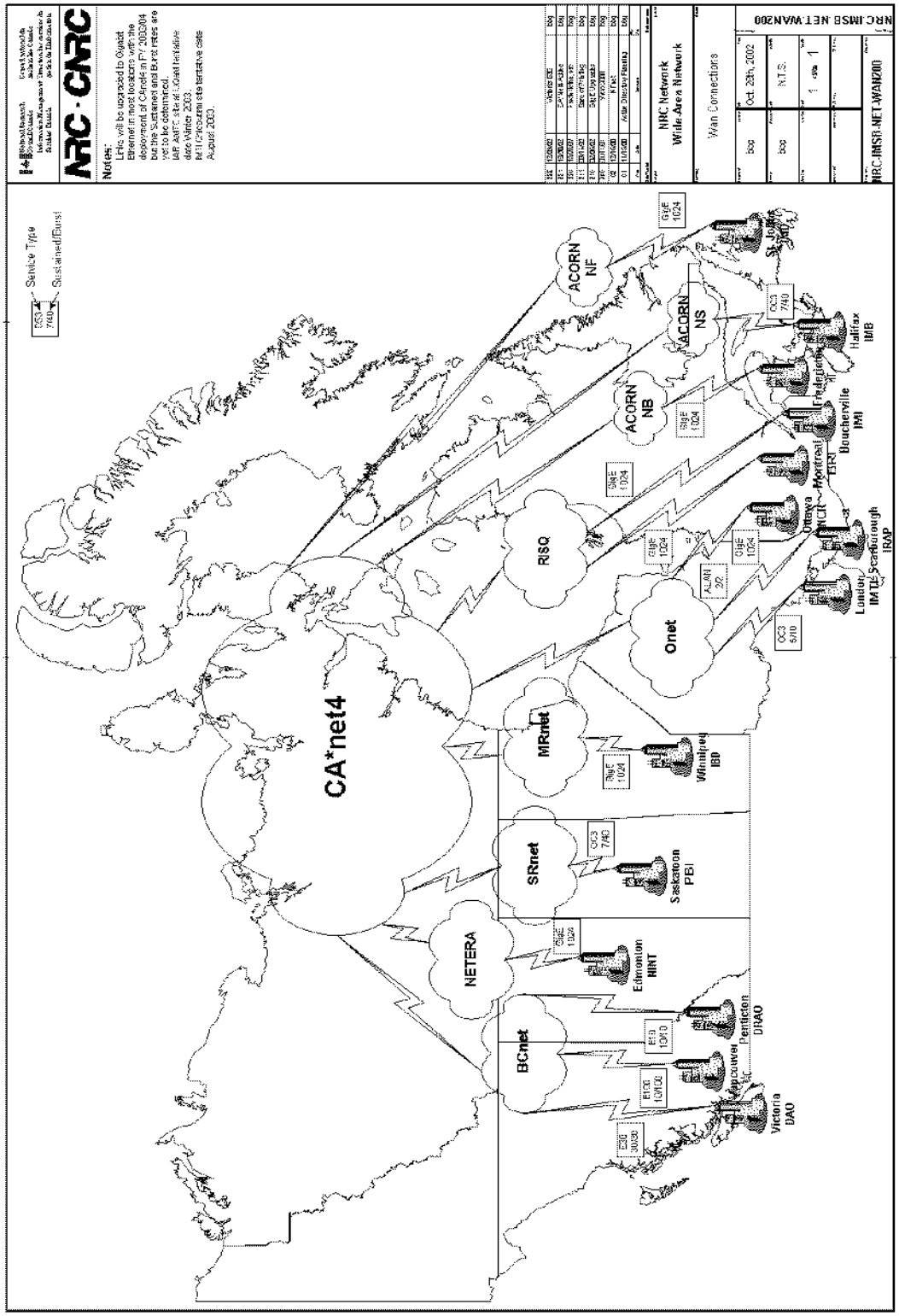

Figure 1.2: Network Connectivity of NRC/all Institutes

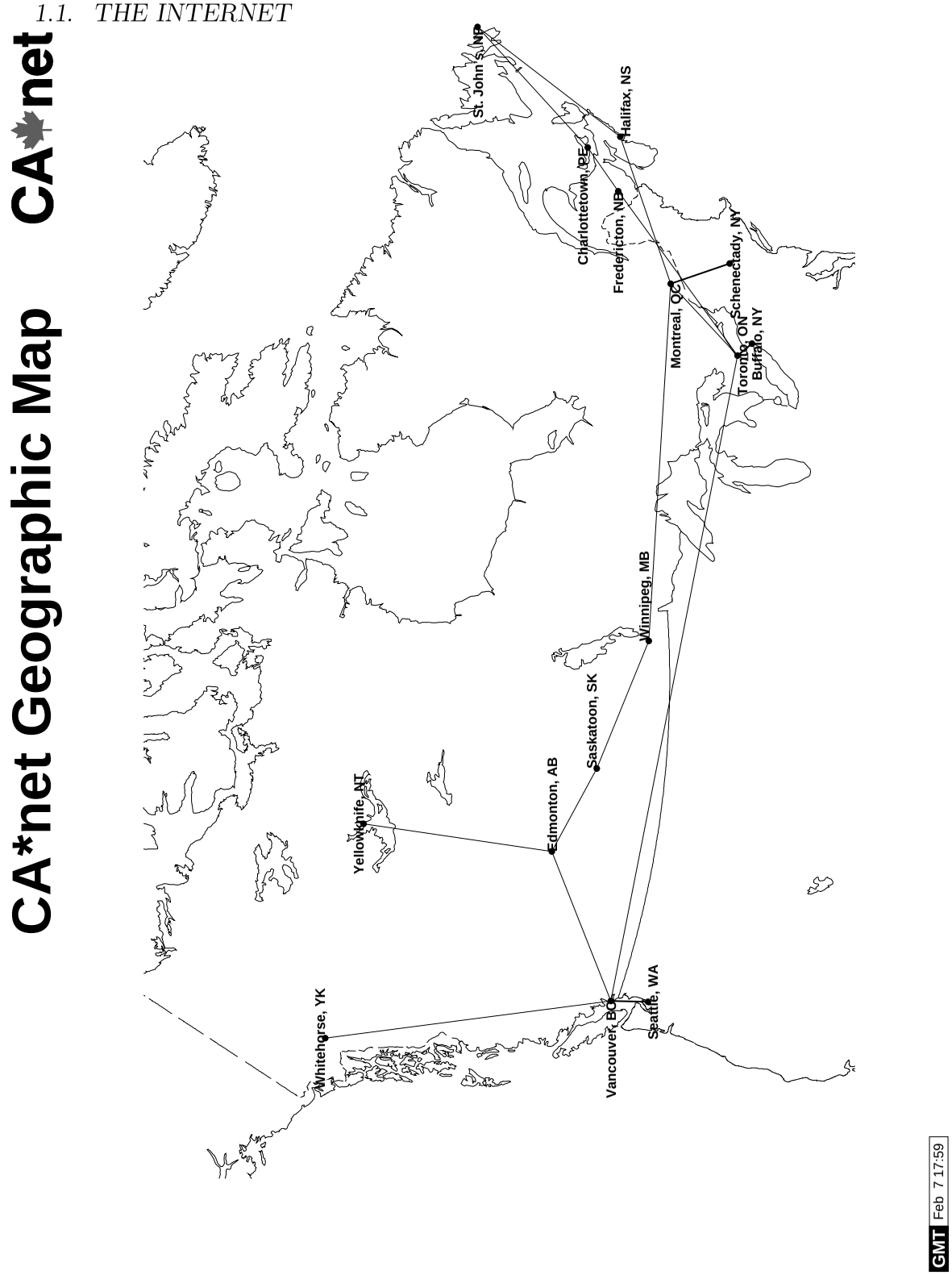

Figure 1.3: Structure of CA*net (before it became BITS)

Province/Territory	Regional Network
Yukon	YKnet
Northwest Territories	NTnet
Nunavut	???
British Columbia	BCnet
Alberta	NETERA
Saskatchewan	SRnet
Manitoba	MRnet
Ontario	ONet
Québec	RISQ
New Brunswick	ACORN NB
Nova Scotia	ACORN NS
Newfoundland	ACORN NF
Prince Edward Island	ACORN PE

Table 1.1: Canadian Regional Networks.

speeds were to be 155 Mbps, the fastest and largest national network in the world (at start-up time).

CA*net 2 is connected in each province to Regional Advanced Networks (some provinces have more than one connection to their RAN), via GigaPoPs (Gigabit Point of Presence). A list of Regional Advanced Networks is shown in Table 1.2, based on October 1997 data.

Figure 1.4 (courtesy of CANARIE Inc.) shows the topology of CA*net 2, as of October 1997. As mentionned above, this topology is constantly changing.

CA*net 2 would connect to the US "vBNS" (very high-speed Broadband Network Service) and "Internet 2" initiatives. CA*net 2's European connection was done via the Halifax GigaPoP.

NRC was connected to CA*net 2 via an ONet GigaPoP located in building M-60 of the Ottawa Montreal Road Campus. The Canadian Bioinfomatics Resource (CBR) based in the Institute for Marine Biology in Halifax, was the driving force behind that connection.

CA*net 2 would be replaced by CA*net 3.

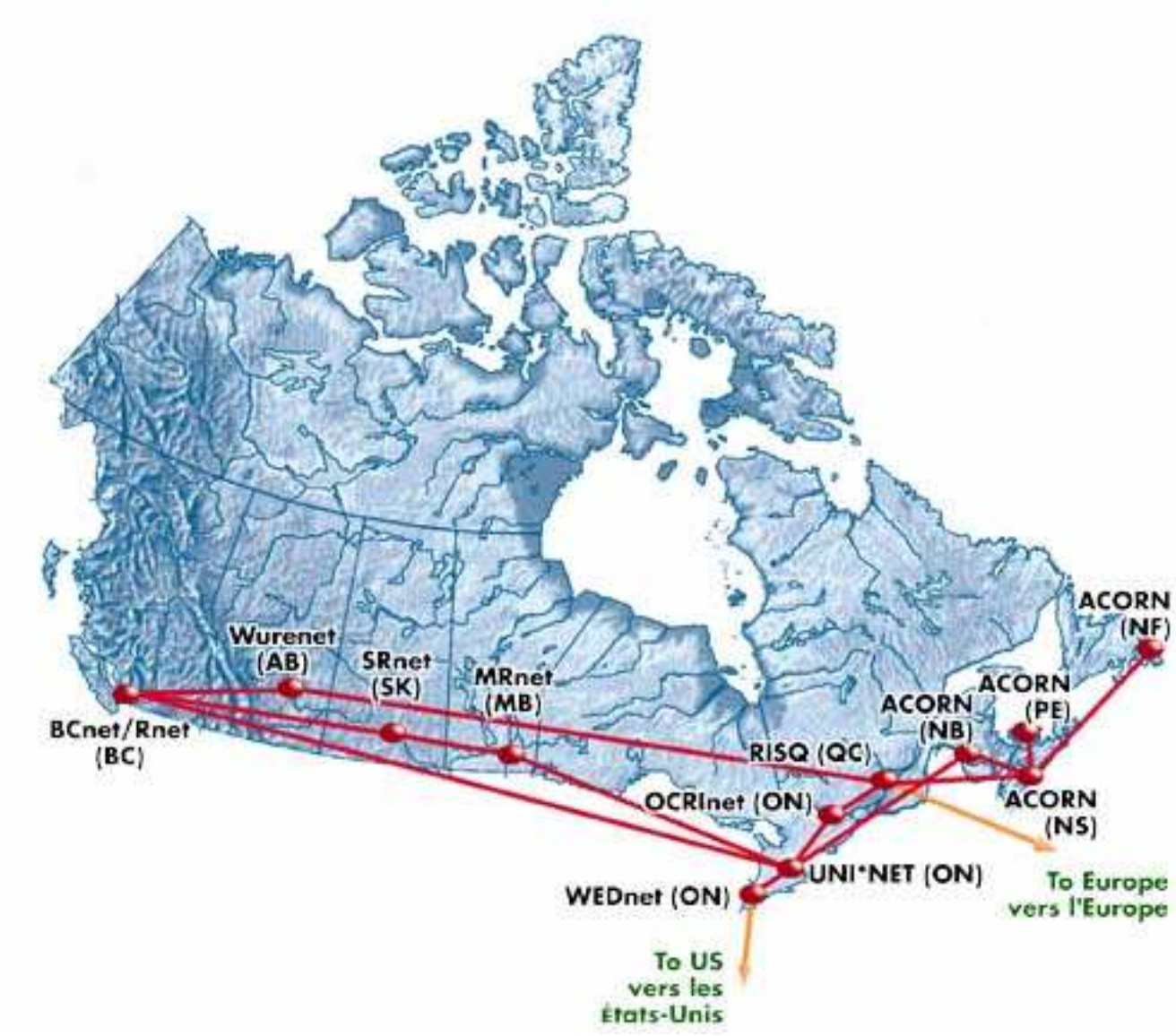

Figure 1.4: Structure of CA*net 2

Province	Regional Advanced Network
British Columbia	BCnet/Rnet
Alberta	Wurenet
Saskatchewan	SRnet
Manitoba	MRnet
Ontario	LargeNet
Ontario	ONet
Ontario	UNI*NET
Ontario	WEDnet
Québec	RISQ
New Brunswick	ACORN
Nova Scotia	ACORN
Newfoundland	ACORN
Prince Edward Island	ACORN

Table 1.2: CA*net 2 Connectivity (October 1997).

CA*net 3

In February 1998, CA*net 3 was announced. This fibre optic network was the first of its kind in the world, as it was developped first and foremost as a fibre optic/data based network. It could deliver up to 40-gigabit capability (that is 250 times faster than CA*net 2). Imagine: the 2.5 hour movie Titanic downloaded in .5 seconds.

CA*net 3 was the world's first national optical network. Instead of using 1 light within a strand of the fibre optic cable, it uses 32 lights of different colours, thus providing for a much larger bandwidth. This is known as Dense Wavelength Division Multiplexed (DWDM) technology.

CA*net 3, as was the case when CA*net 2 was first developed, was only available to researchers at universities, and government laboratories engaged in research and applications development relating to high-performance networks.

The March 1999 CA*net 3 (proposed) topology is presented in 1.5.

Access points to CA*net 3 are shown in 1.3.

More up to date information may be accessed at

`http://www.canarie.ca`

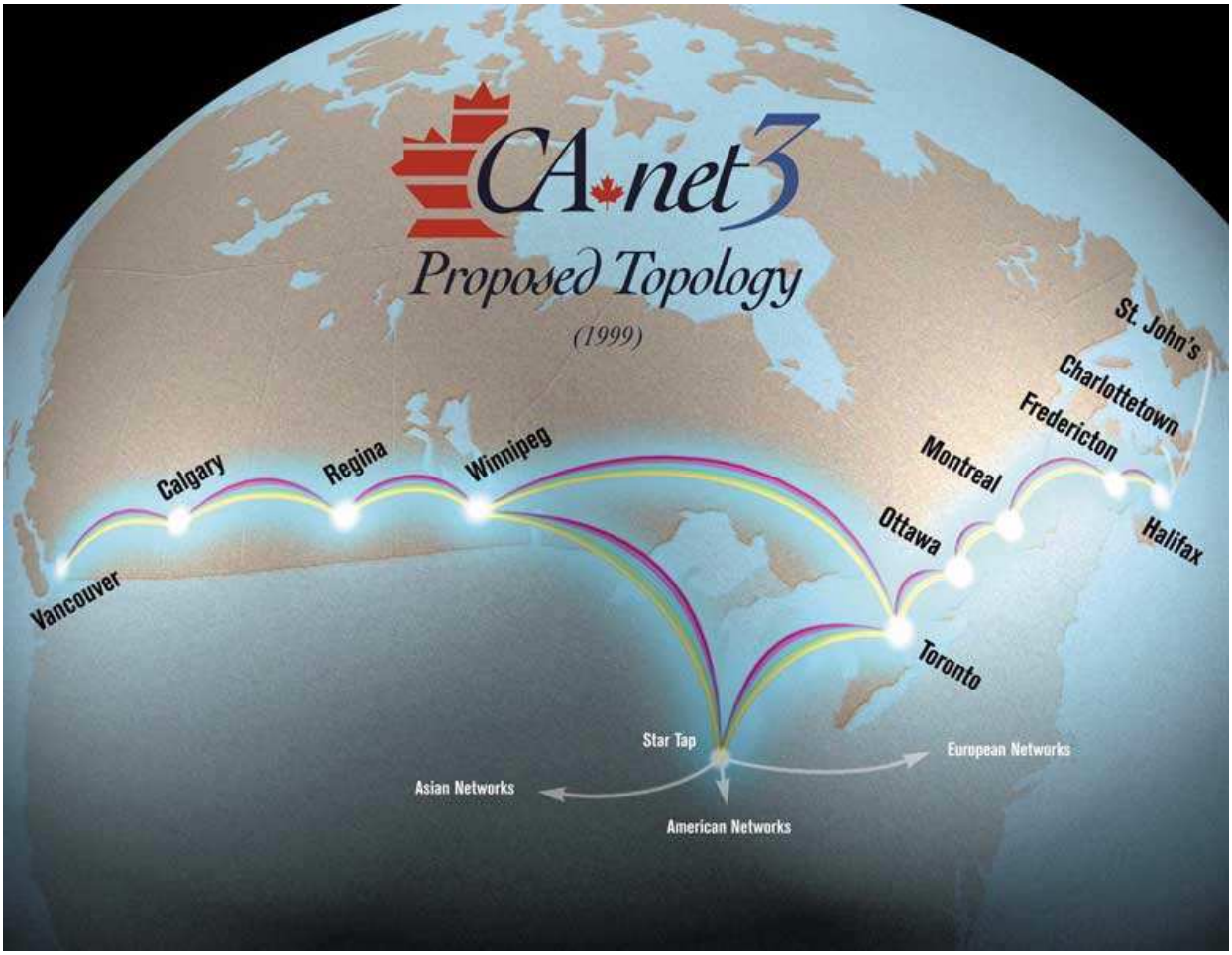

Figure 1.5: Topology of CA*net 3

Province	gigaPOP location	backbone node
British Columbia	BCnet	Vancouver
Alberta	WURCnet	Calgary
Saskatchewan	SRnet	Regina
Manitoba	MRnet	Winnipeg
Ontario	ONet	Toronto
Ontario	ARDNOC	Ottawa
Ontario	NRC	Ottawa
Ontario	CRC	Ottawa
Québec	RISQ	Montreal
New Brunswick	UNB	Fredericton
Nova Scotia	DAL	Halifax
Nova Scotia	UPEI	Halifax
Nova Scotia	Memorial	Halifax
US	Chicago	

Table 1.3: CA*net 3 Connectivity.

`http://www.canet3.net` (this will bring you to the CA*net 4 page!)

CA*net 4

As has happened with CA*net 2 and CA*net 3, a newer, faster, better, more flexible network followed.

During 2002, CA*net 3 was therefore replaced with CA*net 4. As with its predecessors, CA*net 4 connects the provinces research networks using state-of-the-art network technology. Initial connections to the sites is at OC-192 (10Gbps).

A Map of CA*net 4 is shown in 1.6.

Summary

In summary, smaller networks (NRC's) are connected to bigger networks (ONet) which in turn are connected to still bigger networks (BITS, CA*net 2, CA*net 4), inter-connected with other similar networks (NSFnet, Internet 2).

Each `node` on the network is assigned a name (and, transparently, a numeric IP address).

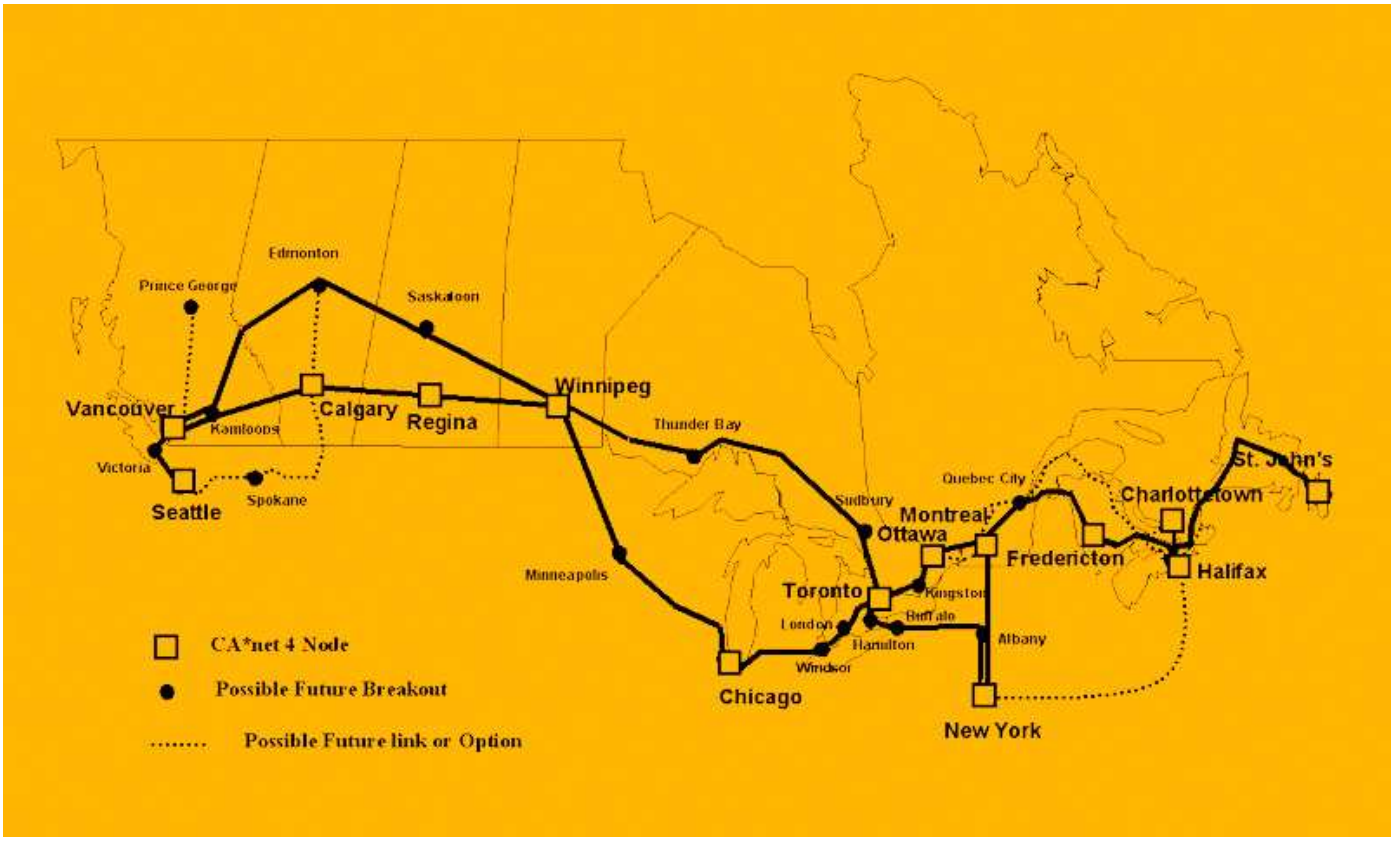

Figure 1.6: Topology of CA*net 4

Tools to access the information distributed on the Internet are gaining in popularity. Traditional tools like `gopher`, `USENET News Readers` have been replaced with web browsers such as `Mozilla`, `Netscape` and `Internet Explorer`. The goal of these tools is to provide assistance to users in searching and collecting the wide range of information available on the Internet.

1.1.2 Cost

Each member of the network has to pay their share of the cost of the lines. NRC's cost for the 2000/2001 fiscal year for connecting to the Internet, BITS, CA*net 2 and CA*net 3 was $263,000.

NRC's allowed bandwidth on CA*net 2/3 is currently (March 2001) set to 5 Mbps with bursts (if the bandwidth is available) of up to 12 Mbps. The non CA*net 2/3 brandwidth available to NRC is 6 Mpbs with 12 Mpbs bursts.

1.1.3 Basic Internet Protocols

IP: Internet Protocol

The Internet set of networks are all based on `IP`, the `Internet Protocol`. "The Internet Protocol (IP) takes care of addressing, or making sure that the routers know what to do with your data when it arrives." [5, p. 25]

Data is transmitted in a series of small chunks, called *packets*, each approximately 1200 characters in length.

The header of each *packet* includes the destination address of the system to receive the packet. The address of that system is the *IP address*.

All IP addresses consist of four fields of numbers ieach one less than 256.

Any packet containing IP addresses in the form

`132.246.`*xxx.xxx*

are for systems owned by NRC (actually, it is more precise to say that they are destined for systems behind the main NRC National Capital Region router). The routers in the network use this number to decide where to forward each packet.

Some addresses, however, are used for internal networks only. Three sets of addresses were set aside for that specific purpose:

- 10.0.0.0 – 10.255.255.255

- 172.16.0.0 – 172.16.255.255
- 192.168.0.0 – 192.168.255.255

Those sets of addresses should never be routed through the Internet.

TCP: Transmission Control Protocol

The Transmission Control Protocol describes how those IP packets are to be transmitted. The TCP software ensures all packets are received and that all packets are placed in the proper order. It then "extracts the data, and puts it in the proper order." [5, p. 27]

UDP: User Datagram Protocol

The User Datagram Protocol is typically used by applications requiring small amounts of data transfer (less than one packet at one time). UDP may be viewed as "TCP's small brother" since it handles one packet at a time instead of a string of them.

1.1.4 DNS: Domain Name Service

The Domain Name Service is also know as `name server`. It translates IP addresses to system names and system names to IP addresses.

The DNS is responsible for translating

```
nickel.sao.nrc.ca
```

which humans understand, to the address

```
132.246.10.3
```

which routers (the equipment responsible to route the packets through the network) and computers understand.

NRC's main name server (and thus, DNS) in Ottawa is

```
dns1.nrc.ca
```

1.1.5 Node/Machine Names

By convention, the name of systems follow the style

machine_name.division.organization.country (outside USA)
or
machine_name.division.organization.org_type (inside USA)

where

- *machine_name* is the personal name of the system.
- *division* is the division, institute or group which owns that workstation.
- *organization* is the name of the company which owns the computer.
- *country* is a two letter country code, such as `ca` for Canada, `fr` for France, `uk` for United Kingdom, etc.

  The official list of two letter accronyms may be found at `http://www.iana.org/cctld/cctld-whois.htm`.
- *org_type* is the type or organisation, as shown in Table 1.4.
- In November 2000, ICANN (the Internet Corporation for Assigned Names and Numbers) approved a list of seven (6) new Top Level Domains. Shown in 1.5, those new Top Level Domain names were not expected to be in use before the Fall of 2001. In fact (Winter 2003), they have yet to be noticed by the casual web surfer.

An example is

```
nickel.sao.nrc.ca
```

where `nickel` is the *machine_name* owned by the group/branches `sao` (Science Affairs Office – now called Research Program Support Group) of the `nrc` (National Research Council) , in `ca` (Canada).

org_type	Description
arpa	Old Style Arpanet
com	Commercial Organisation (mostly US)
edu	US Educational
gov	US Government
int	International
mil	US Military
nato	Nato field
net	Major Network gateway, ISP
org	Non-profit Organisation (mostly US)

Table 1.4: Main Domain Names (non-country).

1.1.6 `NIC`, `ICANN`: Internet Management

If every system on the network must have a unique IP address, who is ultimately assigning those addresses to institution?

The answer this question is the Internet `Network Information Center`, also known as the `InterNIC`.

They may be contacted at

```
www.internic.net
```

Some of the `InterNIC`'s previous responsabilities have been transferred to `ICANN` (The

org_type	Description
aero	Air-transport industry
biz	Businesses
coop	Cooperatives
info	Unrestricted use
museum	Museums
name	Registration by individuals

Table 1.5: New Top Domain Names (2001)

Internet Corporation for Assigned Names and Numbers). `ICANN`'s responsibilities include IP address space and domain name service (DNS) management (including root DNSs).

`ICANN` information is at

`www.icann.net`

1.2 Finding Users and Communicating with them

There is no central place on the network where one could look to find out the electronic mail address for a specific person. It would be like asking if there was one place in the world where you could get the phone number (or address) of any individual.

The Internet is a network of networks. Individual networks may have a database of their own users and their addresses. Some do not.

There is no standard by which one can get (or be) registered. There is also the "privacy issues" to consider (many would argue the telephone directories are no different than email addresses. The main differences is that the phone companies have had 80 years to get together and collaborate. There are also a lot fewer telephone companies than there are Internet Service Providers + Organisations).

There are a number of ways, within UNIX/Linux, one can use to try to find out information about users. The most often used are the `finger` and `rusers` commands. But because of security issues, many systems disable the use of those services.

On the systems maintained by the Research Computing Support Group, the `phone` command is available. That command is mapped to

`whois -h gold.sao.nrc.ca` *person_name*

One may also use Web Browsers to find information about a specific person. Use a search engine, and enter the person's name in quotes. Results may amaze you!

1.2.1 `finger`: Point Finger To

finger [*user*[@*node*]]

where *user* is the logon name of the user whose information is to be displayed, and *node* is that user's machine name.

With no arguments, `finger` displays all users presently logged on, who they are, and the time each of them logged on.

When information concerning a specific user is sought, the information displayed is more detailed: the logon name, and the real name are displayed , the `home` directory of the user is displayed, as well as the logon shell, and the last logon time.

user does not need to be logged on for `finger` to work.

```
prompt> finger cantin

Login name: cantin                      In real life: Claude Cantin
Office:  2025 Sx, 993-0822
Directory: /usr2/people/cantin           Shell: /bin/tcsh
On since Jul 22 08:49:29 on ttyq0       2 hours 59 minutes Idle Time
On since Jul 22 08:49:34 on ttyq1       2 hours 33 minutes Idle Time
On since Jul 22 10:11:35 on ttyq4 from dp1.phy.nrc.ca
On since Jul 22 09:11:02 on ttyq2       1 hour 53 minutes Idle Time
On since Jul 22 11:00:38 on ttyq5 from dp1.phy.nrc.ca
48 seconds Idle Time
Plan:

Progresser, progresser, progresser...
```

In the above example, `cantin` is presently logged on. His real name is Claude Cantin. His home directory is `/usr2/people/cantin`. He is logged on to his account five times, two of which are from a remote location (`dp1.phy.nrc.ca`)

Furthermore, he has a file named `.plan` in his home directory, whose content is

> Progresser, progresser, progresser...

If he did not have the `.plan` file, the string

> No Plan.

would have been displayed instead.

In the next example, Ratilal Haria, on *node* `nrcnet0`, last logged on Friday December 8th, at 10:52, from the node whose I.P. (Internet Protocol) address is 132.246.20.10. That node was using a windowing interface (we know this because it was logged on to `ttyp0` on that machine – not the console).

```
prompt> finger ratilal@nrcnet0
```

```
[nrcnet0.nrc.ca]
Login name: ratilal                 In real life: Ratilal Haria
Directory: /home/nrcnet0/ratilal    Shell: /bin/csh
Last login Fri Dec  8 10:52 on ttyp0 from 132.246.20.10
No Plan.
```

It should also be pointed out that some systems may not allow local users to be `finger`ed by remote systems. In that that case, and error message would be displayed:

```
prompt> finger cantin@cansnd.cisti.nrc.ca
finger:  cansnd.cisti.nrc.ca:  connection refused
```

1.2.2 `rusers`: Remote Users

To find out if other users on the same network are logged on,

rusers [*node...*]

`rusers` by itself simply gives all the logged on users on the given network. If *node* is specified, then only that node(s) is checked.

1.2.3 `talk`: Talk To

If user_a wants to talk to user_b and both are logged on, he/she may want to use the `talk` command:

talk *user_b*[@*node*]

When `talk` is invoked, the screen of the originator is split into two equal parts (a horizontal line of hyphens separates the two parts). *user_b* receives a request to answer:

```
Message from Talk_Daemon@neon at 9:59 ...
talk:  connection requested by smith@neon.sao.nrc.ca.
talk:  respond with:  talk smith@nickel.sao.nrc.ca
```

Once *user_b*, in this case `smith@neon`, responds to `cantin@nickel`, his screen will be split in two: the top portion of the screen always displays the local user's messages, and the bottom portion the remote user's message. Characters are displayed as they are typed.

To close the connection, press <CTL-C>.

As with `finger`, many systems do not allow non-local systems to use the `talk` facility on the local system.

1.2.4 `write`: Write To

For line mode terminals,

write *user_b*

is used. As with `talk`, `write` is used to communicate between logged-on users. BUT unlike `talk`, `write` is not "conversational".

When `write` is invoked, the user types in the message to be sent: every time a carriage return is typed, that line is sent to the remote user. <CTL-D> is used to get out of line mode.

1.3 Connecting to other Systems

`Traditionally, UNIX/Linux systems have used telnet to connect to remote systems. But if someone were to use a` *network sniffer* `(a program which allows the reading of all information flowing through the network), not only could they eavesdrop on your connection, but they could see your login AND password as you type them!`
`telnet connections are therefore considered` *non-secure*.

One way to avoid this is to ensure all data exchanged is encrypted. This is what SSH – Secure SHell – does: the initial connection between two system allows for the transfer of information, including public keys. A session ID is calculated by both parties. That session ID is then used as part of every piece of information exchanged, to ensure the connection is encrypted, and thus considered much more *secure*.

`ssh` is quickly becomming the prefered method for connecting to remote systems.

`ssh` is first introduced (because it is considered *secure*), even though it is still not found on all UNIX/Linux installations. `telnet`, on the other hand, is found on all UNIX/Linux platforms, as well as on the MS Windows 95/98/2000/NT systems.

1.4 SSH: Connecting Securely to other Systems

SSH itself is a protocol. Older NRC/RCSG supported systems use mostly version 1.5 of the protocol (version 1.2.`xx` of the `ssh` tool). All newer systems (from year 2000+) use openSSH, which supports both versions 1 and 2 of the protocol.

Version 2 of the SSH protocol is not compatible with version 1. Although it is considered more secure, and may provide more flexibility, not many public domain client and server packages are available for it.

The following commands are tools based on the SSH protocol to ensure encrypted sessions between systems.

1.4.1 `slogin`, `ssh`: Secure `telnet`

ssh [**-l** *login*] *node*
or
ssh [*login***@**]*node*

allows user *login* to initiate a session with system *node*.

`slogin` is another name for the `ssh` command. Both work the same way – only the name differs.

If there is not a proper `.rhosts` or `.shosts` file entry, a password will be required. That password will NOT be sent in plain text. It will be sent in encryption form.

The default encryption method uses RSA encryption and decryption, using the public/private key scheme: during the initial handshake between the two systems, they exchange their public keys (encryption keys). When any message is sent to the other party, it is first encrypted using that party's public key. When the message is received, it is read using the host's own private (or decryption) key.

The private key is NEVER given to anyone. Only the public key is.

The public key encrypts messages, while the private key decrypts them. Even is the message is intercepted by a third party, they cannot decrypt it because they have no access to the private key of the destination host.

The session is therefore *secure*.

Most RCSG supported SGI/IRIX NRC systems configured to use SSH have had their `telnet` and `rlogin` commands modified so that they use the SSH protocol if the remote host will accept it.

Escape sequences

During your SSH session, you may want to come back to your original shell, without breaking your connection. A series of escape sequences allow you to perform that function:

Supported escape sequences:
`~.` - terminate connection

`~^Z` - suspend ssh
`~#` - list forwarded connections
`~&` - background ssh (when waiting for connections to terminate)
`~?` - this message
`~~` - send the escape character by typing it twice
(Note that escapes are only recognized immediately after a newline.)

For example, to suspend a current SSH session:

```
~^Z
```

(that's *tilde*, followed by CTRL-Z). To resume execution of the session:

```
fg
```

If, for some reason, there are network problems, the current SSH session "hangs", and you want to disconnect from it:

```
~.
```

1.4.2 `ssh-keygen`: password-less SSH login

SSH is often used to login from one system to another without requiring passwords.

A number of methods may be used for that to work properly, one of which is to setup a `.rhosts` file (permission 600) with its content being the name of the remote system you trust, followed by the username your trust:

```
nickel.sao.nrc.ca cantin
```

would mean you trust user `cantin` from `nickel.sao.nrc.ca` to connect to your account, without requiring a password.

But for that to work, SSH itself must be configured to trust `.rhosts` files (which it does not for most OpenSSH installations – but we do on most systems RCSG maintains), and the private/public key pair of each system must be properly set in the system-wide `ssh_known_hosts` public key file.

This, of course, requires help from the local systems administrator.

The second method does not require any help from the systems administrator. And it does not require modifications to the `.rhosts` file. Instead, it requires you generate your own personal set of private/public pair.

`ssh-keygen` is used to generate that key pair for you. Here is a session where your own personal private/public key pair is created:

```
cantin@sodium:~> ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/cantin/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/cantin/.ssh/id_rsa.
Your public key has been saved in /home/cantin/.ssh/id_rsa.pub.
The key fingerprint is:
f6:61:a8:27:35:cf:4c:6d:13:22:70:cf:4c:c8:a0:23 cantin@sodium
```

The command `ssh-keygen -t rsa` initiated the creation of the key pair.

No passphrase was entered (`Enter` key was pressed instead).

The private key was saved in `.ssh/id_rsa`. This file is read-only and only for you. No one else must see the content of that file, as it is used to decrypt all correspondence encrypted with the public key.

The public key is save in `.ssh/id_rsa.pub`.

In this case, the content of file `id_rsa.pub` is

```
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEArkwv9X8eTVK4F7pMlSt45pWoiakFkZMw
G9BjydOJPGHORFNAy1QqIWBGWv7vS5K2tr+EEO+F8WL2Y/jK4ZkUoQgoi+n7DWQVOHsR
ijcS3LvtO+5ONp4yjXYWJKh29JL6GHcp8o7+YKEyVUMB2CSDOP99eF9g5Q0d+1U2WVdB
WQM= cantin@sodium
```

It is one line in length.

Its content is then copied in file `.ssh/authorized_keys` of the system you wish to SSH to without being prompted for a password.

The example shown here generated keys on `sodium` by user `cantin`. If the public key generated, file `.ssh/id_rsa.pub`, was copied to your account, file `.ssh/authorized_keys` on `nickel.sao.nrc.ca`, then user `cantin@sodium` is allowed to SSH into your own account on `nickel.sao.nrc.ca` without the use of a password.

To summarize, a personal private/public key pair is generated using the `ssh-keygen` command. The public key is then copied onto a remote systems' `.ssh/authorized_keys` file. And you can now SSH to the remote systems's account without the use of a password.

1.4.3 `telnet`: Connecting to Remote Systems – (*non-secure*)

telnet *node*

allows a connection to *node*, from any other machine connected to the Internet. *node* itself will prompt for a logon name, then the password.

Logging on remotely allows users to use the UNIX/Linux commands, but not the Graphics User Interface of the remote machine.

Logging out of the remote machine will close the connection between the host and the remote system.

You cannot use `telnet` to connect to any of the systems mananged by the Research Computing Support Group: it was completely disabled in 2000.

1.4.4 `rlogin`: Bypassing Password – (*non-secure*)

rlogin *node*

allows a connection to *node* without the use of a password.

For this to be true, two things must be properly set.

1. You must have an account (of the same name) on the remote system.
2. File `.rhosts` on the remote system must contain the name of the local system, followed by your account name.

`rlogin` is essentially the same as a `telnet` but assumes the same account name, and looks for the `.rhosts` file on the remote machine. You can therefore not use `rlogin` to connect to any of the Research Computing Support Group's managed systems.

1.5 Logging on to NRC from home

1.5.1 Cisco Server/M-60 dial-in access

NRC no longer maintains its M-60 dial-in access. That service was discontinued on April 1, 2001.

NRC contracted a local ISP, EduNET to replace that service.

1.5.2 EduNET dial-in server

This service replaces the "M-60/Cisco Server Dial-up facility".

Since the summer 2000, NRC personnel may get an account with `EduNET`, a (Ottawa) local Internet Service Provider (ISP). NRC has an agreement with that ISP, which allows

approved NRC personnel to have an account with them. Once connected to the EduNET ISP, the NRC employee becomes part of the NRC internal network.

This allows them to access NRC-only services such as CISTI services and the NRC/zone web pages. It also allows users to connect to internal NRC UNIX/Linux systems.

There is a minimum monthly cost for each account. That cost is paid by the individual institutes (in some intitutes, by the individual groups). EduNET registration information is available either from your institute corporate office, or from your computing support personnel.

1.6 Connecting to UNIX/Linux from MS Windows-based Systems

Although these course notes are for UNIX/Linux, many people use PCs running a Microsoft-based operating system such as Windows 95/98/2000/NT to access their UNIX/Linux servers. Traditionally they have relied on "stock" programs like `telnet` and `ftp` to access their systems. They have also used tools like Eudora or Outlook to read their UNIX mail.

Since the spring of 2001, all communication done with UNIX/Linux must be done through a secure channel. Between UNIX/Linux systems, that secure channel is created when using `ssh` and `scp`.

The Research Computing Support Group (RCSG) has put together a series of tools people can install on their PCs, to access the UNIX/Linux systems both within NRC, and from the NRC dial-up access.

The tools covered include `putty` and `WinSCP`:

- `putty` is a SSH-based telnet-like client. It allows for secure communication between Windows and UNIX/linux, much the same way `ssh` does on the UNIX/linux platforms.

  It has a wide range of configuration for fonts, colours, behaviour. If you run X on your PC, `putty` allows the tunnelling of X applications (option must be enabled within `putty`).

  Its basic installation requirement is the download of one executable `.exe` file, but the full package includes command and batch capable utilities.

- `WinSCP` is the UNIX/linux equivalent of `scp`. Graphical-based, it allows for the safe/encrypted transfer of files to/from Windows and UNIX/linux platforms.

More details about those tools, as well as downloadeable modules may be found at

`http://www.nrc.ca/imsb/rcsg/ras/ssh-clients.html`

That web page also explains how GUI-based PC FTP tools, how mail tools like Eudora and Outlook, may be safely used, through the secure channel created by SSH.

1.7 Transferring Files Between Systems

1.7.1 `scp`: Secure Copy

scp *filename(s) node:filename*
or
scp *node:filename(s) filename*
or
scp *filename(s) user*`@`*node*`:`*filename*
or
scp *user*`@`*node*`:`*filename(s) filename*

where *node* is the name of the remote system, and *user* is the login name to use on the remote system.

`scp` allows the copy of file(s) between systems. The files are transferred in encryption form.

If proper `.rhosts` or `.shosts` files exist, no password will be required. If they do not exist, or the proper entry does not exist, a password will be required.

The encryption method is the same as with `ssh`.

1.7.2 WinSCP: Windows Secure Copy

Although the purpose of these notes are for linux/UNIX users, many people require moving files to/from the PC/Windows environment from/to linux/UNIX. The recommended MS Windows package to perform that operation is `WinSCP`, an open source package, available primarily at

`http://sourceforge.net/projects/winscp/`

but also made available at NRC on our web site at

`http://www.sao.nrc.ca/imsb/rcsg/ras/ssh-clients.html`

1.7.3 FTP: File Transfer Protocol – (*non-secure*)

`This is the traditional way of transferring files with UNIX/Linux. Since it involves using plain-text login/password combinations, it is considered non-secure. Many systems, including most systems managed by the Research Computing Support Group at NRC, do not allow connections to their FTP servers using that tool.`

`The exception to that is the` *anonymous* *FTP* `servers the Research Computing Services Group maintains.`

ftp *node*

A connection to *node* will take place, and a logon name will be required, as well as a password.

Within the `ftp` tool, a restricted number of commands are available:

?: Help

?

will display the list of available commands.

? *command*

will display a short message explaining what *command* does.

`put/send`: Transfer to Other

put [*local_file* [*remote_file*]]

If invoked by itself, `put` will prompt for *local_file*, then for *remote_file*. If only one parameter is used, *remote_file* will be assumed to be the same as *local_file*.

`mput`: Multiple Transfer to Other

Used the same way as `put`, this command also accepts metacharacters. It will prompt for every file to be transferred, unless the `-i` flag was used on the `ftp` command line, or the `prompt` command is issued.

`mput` is very useful when moving a number of files.

`get/recv`: Transfer From Other

The syntax for `get` is the same as for `put`:

get [*remote_file* [*local_file*]]

If invoked by itself, `get` will prompt for *remote_file*, then for *local_file*. If only one parameter is used, *local_file* will be assumed to be the same as *remote_file*.

If *local_file* is set to `-`, it will be taken as standard output (i.e. the screen). For example,

get README -

will display the content of `README` on the screen.

get *remote_file* "| *command*"

will run the local *command* using *remote_file* as input.

Example:

get README "| more"

`mget`: Multiple Transfer From Other

Used the same way as `get`, this command can use metacharacters, but will prompt the user before every file transfer unless the `-i` flag was used with the `ftp` command, or the `prompt` command is issued.

As with `mput`, this command is very useful when moving a group of files.

`bin`: Binary

bin

is used when transferring binary files. Files containing binary code include compiled programs, commands, tar files (*file*.`tar`), compressed files.

The response to this command is

Type set to I.

`as`: **ASCII**

as

is used when transferring ASCII files (text, source code, etc.).

The response to this command is

Type set to A.

`cd` : **Change Directory**

cd *directory*

is used to change directory in the remote machine.

`ls` : **Listing**

ls [**-lsa**]

is used to produce a listing of the files on the remote machine. The meaning of the flags is the same as for UNIX's `ls`.

If the form

ls *argument filename*

where *argument* may be either a list of options (such as `-l`) or a filename(s) on the remote site, the output from the command would be put in *filename* on the local system.

Another form,

ls *remote_file* "| *command*"

would pipe the output of the remote `ls` directive into the local *command*.

A typical *command* may be `more` or `grep`, such as

ls * "| more"

`quit`:

quit

exits `ftp`.

1.7.4 `.netrc`: FTP in batch

File

`.netrc`

in your $HOME directory allows file transfers in batch mode. Permission on that file should be `600` (`rw-------`); it will likely not work otherwise.

Each record has the format:

`machine` *machine* `login` *login* `password` *passwd*

where each of *machine*, *login* and *passwd* refer to a system name with the login and password for that account on the machine.

Typically, this is used to do anonymous FTP transfers, so an entry could look like:

```
machine nrcnet0 login anonymous passwd cantin@nrc.ca
```

FTP commands to be used would then be put in some file, and the command to be executed would be similar to

ftp *machine* < *filename*

where the content of *filename* could look like

```
bin
cd /pub
get README
quit
```

1.7.5 sftp: secure FTP

There are two incompatible versions of the `sftp` package:

- SSHv2 companion program.
- an independant client/server version

Systems with SSHv2 (ex: openSSH) use the SSH-compatible package. The older IRIX systems at NRC use the other version (if it was installed).

Both are used in a very similar way.

`sftp [`*user*`@]`*node*

`sftp` allows the transfer of file, using interactive commands very similar to `ftp`. The entire `sftp` session is encoded, as are the login/password tokens.

The basic commands are the same as those of the `ftp` command: `cd`, `get`/`mget`, `put`/`mput`, `help`, `quit` are all used the same way as their `ftp` counterpart.

`sftp` ignores `.netrc`.

1.7.6 Anonymous FTP

An `anonymous FTP` account is a special account for which no password is necessary. The login id is either "`ftp`" or "`anonymous`" and the password is typically your E-mail address.

`anonymous FTP` accounts are typically used to provide unrestricted access to freely available files on various sites on the Internet.

Public Domain packages are usually put in such accounts (although many sites now put them within their web server area, for added access control/security).

1.7.7 `archie`/`xarchie`: accessing FTP databases.

`archie` is no longer used, but kept here for historical reasons.

There are thousands of anonymous FTP systems world-wide. They all contain different files and programs.

Prior to the web and its browsers, the best way to find out which application was located where, `archie` was the tool of choice. It was created circa 1990.

The `Archie` database was constructed bit by bit over the period of one month. During that month, each night, `Archie` visited 1/30 (approximatly) of the anonymous FTP sites registered with it, and collected all file/program names stored at each and every location.

`archie` was then used to interrogate the `Archie` database to find the location of a specific program. `xarchie` is a graphical version of the tool.

Although `archie`/`xarchie` is still in use, web searches through search engines is currently a more efficient measure in finding tools on the Internet.

1.8 The Network News

The `Network News`, also know as `NetNews`, is the largest Bulletin Board Service in existence. It may be considered as the "first-generation bulletin boards".

1.8.1 NetNews terminology

`category`: top level name of the organisational structure of newsgroups.
`newsgroup`: is a special interest group.
`article`: is a post to a specific `newsgroup`.
`thread`: a set of articles with the same subject.
`newsfeed`: workstation used to propagate the news to `newsservers`.
`newsserver`: a workstation storing news articles.
`newsreader`: a program used to read the `NetNews` off the `newsserver`.

1.8.2 newsgroup Organisation

The `newsgroups` are organised in a hierarchical tree, each level separated by a dot. The top level is called a `category`.

The [original] `USENET News` consisted of 7 categories/groups (based on [5, p. 154]). Table 1.6 displays those categories.

Category	Description
comp	computer science related.
news	network news related.
rec	hobbies; recreational activities; arts.
sci	scientific research and applications.
soc	social issues; politics.
talk	forum for debate on controversial topics.
misc	other topics.

Table 1.6: USENET News Categories.

But since any `newsgroup` may be created locally and be distributed to all other `news servers`, a number of other categories are now accessible from most sites.

The most important other `categories` are (based on [5, p. 155-6]) shown in Table 1.7.

Any `newsserver` may chose which groups are allowed on the system. NRC has only a subset of the `Network News` groups on the main `newsserver`, `news.nrc.ca`, in addition to many other categories, some of which are seen in Table 1.8.

Category	Description
alt	alternate groups that don't quite fit anywhere.
bionet	for biologists.
bit	BITNET listserv discussion groups.
biz	for business; one of the only categories where advertisement and marketing are allowed.
de	different groups, in the German language.
fj	different groups, in Japanese.
hepnet	High Energy Physics.
ieee	related to the Institute of Electrical and Electronic Engineers.
info	mailing lists on wide variety of topics.
gnu	Free Software Foundation and GNU project.
k12	for teachers and students of kindergarten to grade 12.
relcom	groups originating in the former USSR.
u3b	AT&T 3B computer series.
vmsnet	Digital VAX/VMS and DECnet groups.

Table 1.7: Other important Network News Categories.

1.8.3 FAQ: Frequently Asked Questions

Most `newsgroups` maintain a `FAQ`, regularly posted to the group.

A `FAQ` is a list of `F`requently `A`sked `Q`uestions (with answers) typically asked in a specific `newsgroup`.

The easiest way of accessing the news FAQs is to use the web at URL

```
http://www.faqs.org
```

That site contains FAQs from a large percentage of the USENET newgroups.

1.8.4 Newsreaders

`newsreaders` are programs used to access and read the news. They usually fetch the news from the `newsserver`.

All `newsreaders` are capable of performing at least the following tasks:

Category	Description
nrc	NRC related; eg: `nrc.test`, `nrc.events`.
uw	University of Waterloo; eg: `uw.unix`, `uw.outers`.
ut	University of Toronto; eg `ut.chinese`; `ut.dcs.graphics`.
ott	Ottawa related; eg: `ott.general`, `ott.forsale`.
tor	Toronto specific; eg: `tor.news`, `tor.test`.
ont	Ontario-wide; eg: `ont.jobs`, `ont.uucp`.
can	Canadian groups; eg: `can.newprod`, `can.domain`.

Table 1.8: More Local Network News Categories.

- Read an article.
- Post a `follow-up` article (i.e. post an article within the same thread/subject).
- Reply to the author of the article.
- Save an article.
- Print an article.
- Mail an article.
- Subscribe/Unsubscribe to newsgroup(s).

`newsreaders` use the configuration file

```
.newsrc
```

to find out which newsgroups you subscribe to. It is created the first time the `newsreader` is invoked.

That file may be viewed/edited with any text editor to find out which `newsgroups` are available to be read on that system.

`tin`: threaded Internet newsreader.

`tin` is a user-friendly menu-driven interface to the News. It is said to be `threaded` because it keeps all articles from the same thread together (a more typical newsreader lists the articles in the order they came in).

To invoke it, issue

```
tin
```

When `tin` is started for the first time, it will display a screenful of helpful advice. Pressing any key will then subscribe the user to all newsgroups distributed by the newsserver.

At this point, it is better to `unsubscribe` to all groups by pressing

U

`tin` will then ask for a `regex` (regular expression) to enter. Enter

```
*
```

to specify "all newsgroups".

You are now unsubscribed from all groups. You may then scan the group names using the arrow keys and issue

s

to subscribe to any single group you wish to.

Whenever `tin` is started, a new newsgroup may be added by pressing

g

(`goto` newsgroup) and enter the newsgroup's name. That newsgroup will be automatically subscribed to and its name added to the `.newsrc` file.

Next time you start `tin`, only groups to which you subscribe will be displayed.

There are three main levels:

1. `newsgroups`: this level shows you the newsgroups for which you are currently subscribed. Some possible operations are "`s/u`" (subscribe/unsubscribe) and "`g`" (goto a specific newsgroup).

   The numbers on the left side of the group names tell you how many unread articles are in each of the newsgroups.

2. `threads`: this level shows the list of threads in a newsgroup. The plus (+) sign on the left shows you which one have unread articles.

3. `articles`: you are now looking at the postings to the newsgroup. You may now reply, mail, print, etc.

The TAB key automatically displays (if at the articles level) the next unread article. If not at the articles level, it invokes the next sublevel.

q

will bring back the next higher level, until tin is exited.

One last option worthy of mention is

y

This toggle option (used at the newsgroups level) brings in all unsubscribed newsgroups, allowing you to scan them at your leisure. Issuing the option another time will "yank" out all unsubscribed groups.

There are numerous other options within tin. ‘‘h" within tin will display two help screens of commands to use.

Once in tin, the screen layout will appear as in Figure 1.7.

xrn: X-based news reader.

xrn is a package used to read the News, using the X window system as its interface. ASCII terminals cannot use this package; tin should be used with such terminals.

xrn is user friendly. The mouse is used to click on boxes to read, forward, reply, print, save, delete, and post articles. Customization instructions are found in the man page.

1.9 gopher: go fer it!

gopher is "a lookup tool that lets you prowl through the Internet by selecting resources from menus" [5, p. 233]. The Gopher system is a set of interconnected ASCII based menus. It was developed at the University of Minnesota. To cycle through a menu list, use the arrow keys. To select a menu, simply press the ENTER key.

Using gopher (also the name of a program to access the Gopher system), users can access files, copy them, print them, mail them. They can also FTP programs to their local machine, or access databases, phone books.

Gopher was developed in the late 80s early 90s. Many view it as "the first generation web browser".

It has since been superseeded by web browsers like Netscape and Internet Explorer.

```
Group Selection (nrcnet0.nrc.ca  15)

   1    54  comp.sys.hp                Discussion about Hewlett-Packard
   2    11  ott.general
   3    24  comp.sys.sgi               Silicon Graphics's Iris workstations
   4    45  comp.sys.sun.admin         Sun system administration issues and
   5    11  comp.sys.sun.apps          Software applications for Sun comput
   6    31  comp.sys.sun.hardware      Sun Microsystems hardware.
   7        sci.aeronautics.airliners
   8     6  comp.sys.sgi.admin
   9        comp.sys.sgi.announce
  10     7  comp.sys.sgi.apps
  11     1  comp.sys.sgi.bugs
  12    11  comp.sys.sgi.graphics
  13     6  comp.sys.sgi.hardware
  14     8  comp.sys.sgi.misc
  15        alt.gopher                 Discussion of the gopher information

       <n>=set current to n, TAB=next unread, /=search pattern, c)atchup,
    g)oto, j=line down, k=line up, h)elp, m)ove, q)uit, r=toggle all/unread,
      s)ubscribe, S)ub pattern, u)nsubscribe, U)nsub pattern, y)ank in/out

*** End of Groups ***
```

Figure 1.7: `tin` layout, `newsgroup` level.

1.9.1 Veronica: Searching Gophers

Veronica is to Gopher what Archie is to anonymous FTP. Veronica keeps a collection of all known/registered Gopher menus. People can then search the database to gather a collection of menu items pertaining to (hopefully) a specific domain.

Those menu items are automatically put into a gopher-like menu.

As with archie, veronica is now part of Internet history.

1.10 WWW: World Wide Web

1.10.1 What is it?

The World Wide Web is one of the newer information gathering services on the Internet (early to mid 90s). It is also the most successful tool, by far, used to access information. It is based on the concept of hypertext, first developed at CERN (Centre Européen de Recherche Nucléaire), the European particle physics lab in Geneva, Switzerland..

A hypertext is based on the concept that a word, phrase, or object is actually a link to another object. The hyperlink object is usually either underlined, highlighted, or of a different colour.

The link could represent many types of objects. When the link is activated (by clicking the mouse of a graphical browser, or pressing the ENTER key on an ASCII browser), a number of different scenarios may happen.

Table 1.9 displays the different actions the browser may take, depending on the type of link.

It is assumed the example browser being used is graphical in nature (eg: netscape).

The language used to write WWW documents is HTML (HyperText Markup Language).

http (hypertext transport protocol) is the protocol used to access/service HTML documents.

1.10.2 netscape, internet explorer, lynx, mozilla: WWW browsers

There are two styles of browsers. One is graphical, the other is ASCII.

The ASCII browser was developed at the University of Kansas and is called lynx. It shares many of the same properties as its graphical counterparts, Netscape and Internet Explorer.

linked object	action or program used
Local text file	File is displayed.
Remote text file	File downloaded, then displayed.
Remote `HTML` file	File downloaded, then interpreted.
Remote PostScript file	File downloaded; `ghostview` invoked.
Remote `GIF` file	File downloaded; `xv` invoked.
Remote audio file	File downloaded; `sfplay` invoked.
Remote video file	File downloaded; `mpeg_play` invoked.
Remote `DVI` file	File downloaded; `xdvi` invoked.
Telnet session to *host*	`xterm` invoked with `telnet` *host*.
Gopher site	Gopher site contacted; `gopher` menu started.
FTP site	FTP connection done; file may be downloaded.

Table 1.9: Some actions taken by WWW browsers.

The first graphical browser, `xmosaic`, was developed, as a research project, at the National Center for Supercomputing Applications (NCSA) at the University of Illinois. Version 1.0 of the tool was released in April 1993. After a few versions of the browser, it was decided the project had exceeded its goal, so it was terminated.

The last full version ever released was 2.6 (July 1995) and the last beta version released was 2.7b5 (July 1996).

`Netscape` was developed at Netscape Communications Inc. (formerly Mosaic Communications, Inc), a private software company, founded by Marc Andreesen (the original `xmosaic` developer) and Jim Clark (one of the original Silicon Graphics founders).

Netscape Communications Inc. was purchased by AOL (America On Line) in 1999.

Figure 1.8 is an example of what `netscape` looks like.

Browsers, ASCII-based or graphical in nature, offer much of the same functionalities.

Some of the functionalities include

- starting a new browser window.
- send/receive mail.
- access web servers.
- access news servers.
- save documents as HTML, text or PS format.
- print documents.

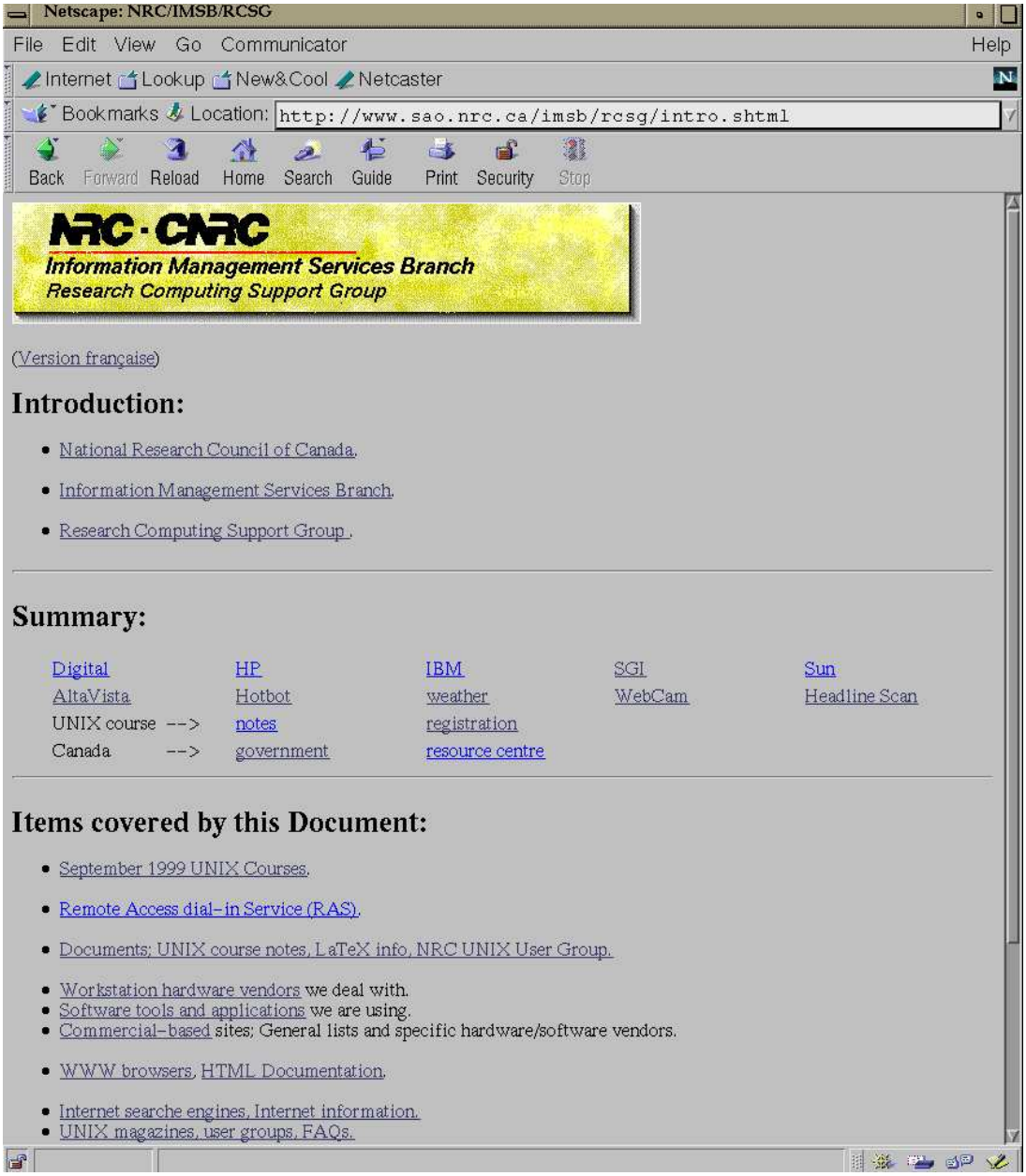

Figure 1.8: RPSO/Research Computing Support Group home page using `netscape`

- find a strings within documents.
- web searches.
- look at the HTML source for the current document.
- navigate through recently accessed documents.
- tag (bookmark) specific documents; manage that list of tags.

1.10.3 URL: Uniform Resource Locator

The action done on the `hypertext link` depends on both the type of `link` it is (the `URL`), and the file name (its extension) itself.

The general syntax of a `URL` is

URL-type`://`*system*`/`*path*

where

- *URL-type* is defined in table 1.10.
- *system* is the name of the system on the network (may also include additional information such as account name, password, and/or port number of server)
- *path* is the path name of the document or directory to be accessed.

An example would be

`http://www.nrc.ca/imsb/rcsg/unix_course`

1.10.4 Miscellaneous Comments on WWW browsers

Here are a number of comments on the use of browsers, but also directed towards people writting HTML documents.

- People don't realize how much information is transferred using these tools. An image file may easily be 1 MB in size. Over a slow or busy link, it may take a long time to transfer.

  Also, each icon has to be downloaded individually; that also adds to the download time.

  In writing the document, use small images; if the user wants to see a more detailed version of it, allow the user to click on it for a high resolution display.

URL type	Description
ftp	allows connection to anonymous FTP servers
http	accesses web server with HTML documents
gopher	accesses gopher servers
telnet	opens telnet session
file	opens local HTML file
mailto	opens builtin mail tool to send mail
news	connects to news server

Table 1.10: URL types and descriptions.

- When documents are read with `lynx` or `netscape` with graphics turned off, information normally given by the images is omitted. That information should be made available using substitute text (HTML has the constructs to do that; unfortunately too many people don't use them in created web pages).

  In `netscape`, unselect `Auto Load Images` (set on by default) in the `options` menu. It will reduce the network bandwidth and the time required to load the documents.

- Large (or many) pictures use up a lot of bandwidth. As a result, it takes a long time to display the document.

  A good balance between graphics and text should be maintained.

- There are converters available to translate from some file formats (such as `WordPerfect` or LaTeX documents) to HTML. Some are better than others.

  In fact, these course notes are available on

  `http://www.nrc.ca/imsb/rcsg/documents`

  in HTML format (translated from the LaTeX source using `latex2html`).

- If possible, always test new documents with several systems and browsers (on PC, MAC, UNIX colour terminal, UNIX b&w terminal, X terminal; using `xmosaic`, `netscape`, `lynx`) to ensure that it is readable from all.

  If a document looks shabby, it may be because it was written on a different platform than you are using to read it.

- Try to access a default local home page. The `home page` is the startup document.

  Most of the SGI Science Institute servers in Ottawa access the IMSB/RCSG home page (see Figure 1.8).

  Any user may write his/her own home page and access it using

  netscape *file.html*

  where *file.html* is local, and written in `HTML`.

- If the server you are using is running an `http server`, you can make your own documents available to anyone with a browser by putting your document in directory

  `$HOME/public_html/`*file.html*

  where *file.html* is the document you would like others to access.

  People could then access your file by using URL

  http://*server_name*/~*login*/*file.html*

  where *server_name* is the name of the UNIX workstation or server running the `http` server, *login* is your `logon` account, and *file.html* is the document opene to the public.

  Some institutes are restricting the use of the `public_html` directory to people within NRC only. Outside users have to access to it.

- When writing documents, make use of the directory structure; every document should probably have its own directory along with the images it needs.

- When writing documents, make use of `relative` path names as much as possible. Relying on `absolute` file names is asking for disaster.

  One never knows when the top level directory will be moved somewhere else!

- Writers of `HTML` documents: plan ahead. Know what the general look of the documents will be; how it will be presented.

  Planning ahead will also incorporate ideas/concepts addressed during the few points above. It will also make it much easier for the next person taking over to understand not only the relationships between the documents, but also *where* the source for all documents is (including images, etc.).

- There are many HTML editors available, both in the public and commercial domains. As with other software, you often get what you pay for. You may also find very good public domain editors, both for UNIX and DOS.

  Popular work processing packages also often offer HTML output for documents.

- Again, for developers: ensure each and every directory has an `index.html` file. That way, users accessing the `http` server will never see the layout of your directory structure.

  It also increases the security of the system by not allowing them to see how the information is stored.

1.11 Navigating the Web

Finding information on the Web is like looking for a needle in a hay stack. Fortunately, there are starting points one can use. Tools are also available to help perform searches for specific data.

The following two sections address these two points.

1.11.1 Where to Start

More and more technical journals now publish `URL`s. More and more organisations have a home page. Fortunately, the format of the addresses is relatively consistent: more than often, the name looks like

`www.`*organisation*`.`*org_type*
or
`www.`*organisation*`.`*country_code*

For example

```
www.nrc.ca
```

represents NRC's own home page.

Most high technology organisations have a home page.

A few examples include `www.sgi.com`, `www.nt.com`, `www.ams.org`.

But where does one start when one wants to explore what is out there? The easiest way is to use someone else's work (we all do for driving; that's what maps are).

That "someone else's work" is a list of lists (also known as `portals`), or a home page filled with interesting starting points. That's probably all you need to get started!

One of the more popular portal is

`http://www.yahoo.com`
or
`http://www.yahoo.ca`

Governments are increasingly creating portals for their own services.

`http://www.canada.gc.ca`

is an entry point for the federal government services.

1.11.2 Search Engines

Portals also include search engines. They allow for searches (either within the portal, or on the web in general) of specific items.

Search engines access databases constructed "and updated by programs called 'robots' or 'spiders' that continually 'travel' the Net finding documents. As they encounter documents, they record information about each document, which is then used to update the database of documents." [2, p. 70].

The databases are built over time, as the "robots" and "spider" travel the Internet. Some date the entries and "drop them" after a period of time (not all do!). As a result some of the information gathered may be out of date.

But this is not a reason not to use theses services.

The complexity of the search engine, and of the interface, varies from tool to tool. But the output from the searches, in all cases, results in a list of `URLs` that match the search string.

There are a number of good search engines. Some of them include

`http://www.google.com`
`http://www.altavista.com`
`http://www.yahoo.com`
`http://www.hotbot.com`

1.12 Exercises

1. What is the identifying name of the UNIX system you are using?

2. You have an account on a remote machine. How can you access it? Do it (if you don't have an account on a remote machine, assume the remote machine is the one you are now using).

3. On that same remote system, you have a file you wish to transfer to yourself. How would you do it?

Appendix A

Bibliography

Bibliography

[1] Stephen R. Bourne. *The UNIX System V environment*. Addison-Wesley Publishing Company. Don Mills, Ontario. 1987.

[2] D. Dougherty, R. Koman, and P. Ferguson. *The Mosaic Handbook for the X Window System*. O'Reilly & Associates, Inc. Sebastopol, California. 1994.

[3] E. Foxley. *UNIX for Super-Users*. Addison-Wesley Publishing Company. Don Mills, Ontario. 1985.

[4] Æleen Frisch. *Essential System Administration*. O'Reilly & Associates, Inc. Sebastopol, California. 1992.

[5] Ed Krol. *The Whole INTERNET User's Guide & Catalogue*. O'Reilly & Associates, Inc. Sebastopol, California. 1994.

[6] Jerry Peek, Tim O'Reilly, and Mike Loukides. *UNIX Power Tools*. O'Reilly & Associates, Inc. Sebastopol, California. 1993.

[7] H. McGilton and R. Morgan. *Introducing the UNIX SYSTEM*. McGraw-Hill Software Series for Computer Professionals. Toronto. 1983.

[8] R. Thomas, and R. Farrow. *UNIX Administration Guide for System V*. Prentice Hall. Englewood Cliffs, New Jersey. 1989.

[9] Silicon Graphics Inc. *IRIS-4D User's Guide*, man pages.

[10] Sun Microsystems. *SunOS 4.0, 4.1 Reference Manuals*.

[11] SuSE Linux LTD. *SuSE Linux 7.3 Reference Manual*.

[12] UNIX International. *The UNIX Operating System: A Commercial Success Story*. Nov 1, 1989. Parsippany, NJ.

[13] http://www.canarie.ca; November 1997 version.