

Unit 1

Introduction

The Scope of System Administration

The task of system administration is a balancing act. It requires patience, understanding, knowledge and experience

Being a system administrator is as much a state of mind as it is about being knowledgeable. It is the sound of one hand tapping (on the keyboard) while the other is holding the phone, talking to a user and there is a queue of people waiting for help

We must be ready for the unexpected, resigned to the uncertain, and we need to be able to plan for the future. It requires organization and the ability to be systematic. **Risk management**

If you have installed Windows, DOS or GNU/Linux on a PC, you might think that you already know a lot about system administration, but in fact you know only the very beginning.

The Goals of System Administration

System administration is about putting together a network of computers (workstations, PCs and supercomputers), getting them running and then *keeping them running in spite of the activities of users who tend to cause the systems to fail.*

System administration is a service profession, but it is far more than that. System administrators are also mechanics, sociologists and research scientists.

A system administrator works for users, so that they can complete work which is unrelated to the upkeep of the computer system itself. However, a system administrator should not just cater to one or two selfish needs, but also work for the benefit of a whole community

The Challenges of System Administration

System administration is not just about installing operating systems. It is about planning and designing an efficient *community of computers so that real users will be able to get their jobs done*.

Designing a network which is logical and efficient.

- Deploying large numbers of machines which can be easily upgraded later.
- Deciding what services are needed.
- Planning and implementing adequate security.
- Providing a comfortable environment for users.
- Developing ways of fixing errors and problems which occur

Some system administrators are responsible for both the hardware of the network and the computers which it connects, i.e. the cables as well as the computers.

An understanding of how data flow from machine to machine is essential, as is an understanding of how each machine affects every other.

Bugs

Operating systems and programs are full of bugs. Learning to tolerate bugs is a matter of survival for system administrators

Bugs can be caused by many things. They may come from

- Shoddy software.
- Little known problems in the operating system.
- Unfortunate clashes between incompatible software, i.e. one software package destroys the operation of another.
- Totally unexplainable phenomena, cosmic rays and invasions by digital life-forms.

Information Sources for Sysadms

Information can be found from many sources:

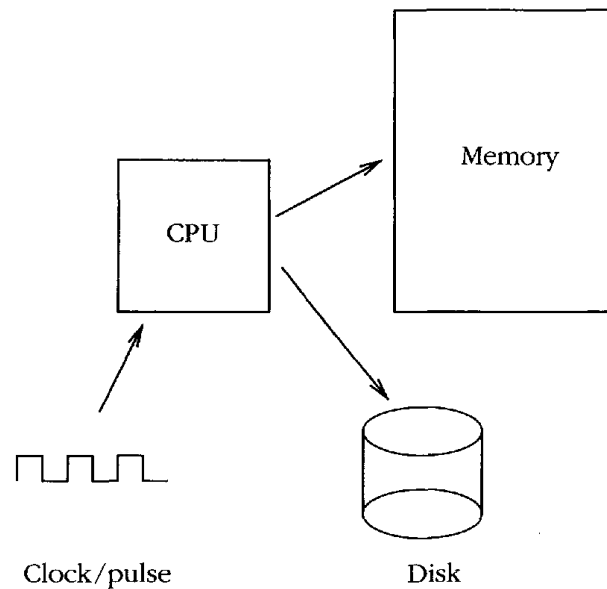
- Printed manuals.
- Unix manual pages (man and apropos commands).
- The World Wide Web.
- RFCs (Requests for comment), available on the web.
- News groups and discussions.
- Papers from the SAGE/Usenix LISA conferences.
- More specialized books.

The System Components

What is The System'?

System a lot to refer both to the operating system of a computer and often, collectively the set of computers on a network.

All contemporary computers are based on the Eckert-Mauchly-von Neumann architecture



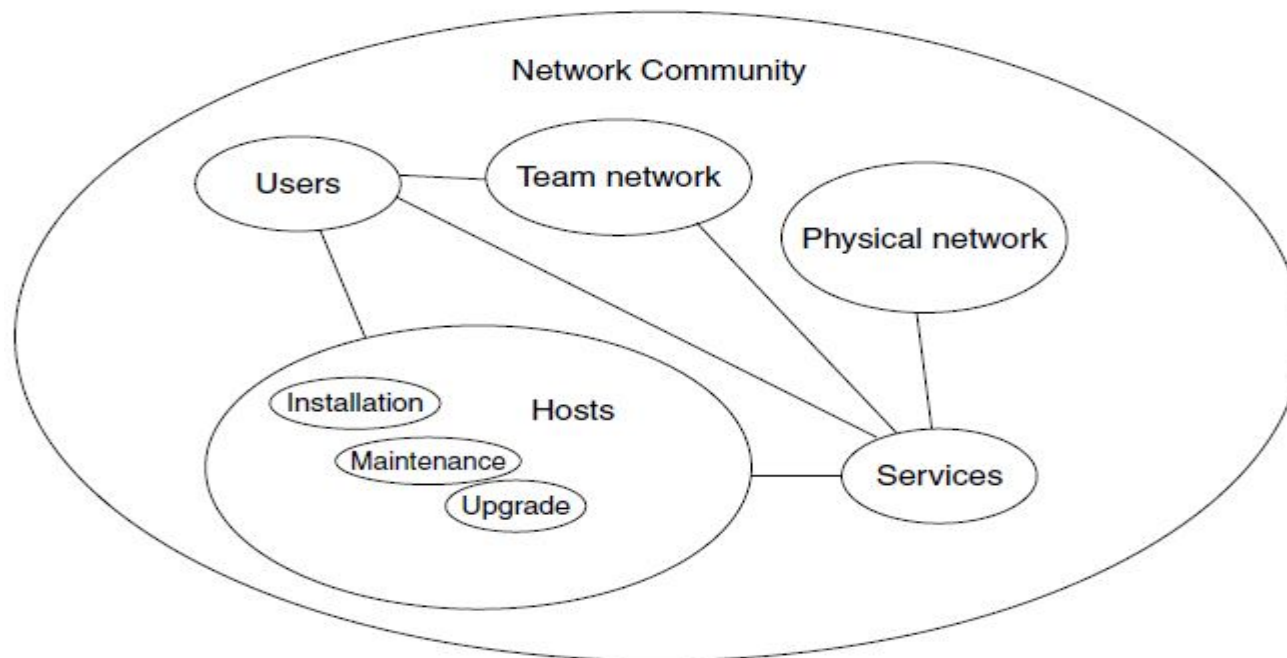
The basic elements of the von Neumann architecture

Each computer has a clock which drives a *central processor unit*(CPU), *random access memory (RAM)* and an array of other devices, such as disk drives

In order to make these parts work together, the CPU is designed to run programs which can read and write to hardware devices.

The most important program is the **operating system kernel**.

- **Network Infrastructure**
 - **Human, Host computers and Network hardware**



Operating Systems

An operating system is the software which shares and controls the hardware resources of a computer.

It shields the user of the machine from the low-level details of the machine's operation and provides frequently needed facilities

Normally the operating system has a number of key elements: (i) a technical layer of software for driving the hardware of the computer, like disk drives, the keyboard and the screen; (ii) a filesystem which provides a way of organizing files logically; and (iii) a simple user interface which enables users to run their own programs and to manipulate their files in a simple way.

Of central importance to an operating system is a core software system or *kernel* *which is* responsible for allocating and sharing the resources of the system between several running programs or *processes*

It is supplemented by a number of supporting *services* (*paging*, *RPC*, FTP, WWW, etc.) which either assist the kernel or extend its resource sharing to the network domain

The operating system can be responsible for sharing the resources of a single computer, but increasingly we are seeing *distributed operating systems* in which the execution of programs and sharing of resources happens without regard for hardware boundaries; or network operating systems in which a central server adds functionality to relatively dumb workstations

For an operating system to be managed consistently it has to be possible to prevent its destruction by restricting the privileges of its users

Operating systems may be classified both by how many tasks they can perform 'simultaneously' and by how many users can be using the system 'simultaneously', i.e. single-user or multi-user and single-task or multi-tasking. A multi-user system must clearly be multi-tasking. The table below shows some examples.

OM- Single user Quasi Multitasking

OS	Users	Tasks	Processors
MS/PC DOS	S	S	1
Windows 3x	S	QM	1
Macintosh System 7*	S	QM	1
Windows 9x	S	M*	1
AmigaDOS	S	M	1
MTS	M	M	1
Unix-like	M	M	<i>n</i>
VMS	M	M	1
NT	S/M	M	<i>n</i>

The Legacy of Insecure Operating Systems

The home computer revolution preceded the network revolution by a number of years, and home computer operating systems did not address security issues.

There is a large number of insecure computers in use, and many of them are now connected to the network. This should be a major concern for a system administrator

Such machines should not be allowed to hold important data, and they should not be allowed any privileged access to network services.

Secure Operating Systems

To distinguish them from insecure operating systems, we shall refer to operating systems like Unix and NT as *secure operating systems*

The most fundamental tenet of security is the ability to restrict access to certain system resources. The main reason why DOS, Windows 9x and the Macintosh are so susceptible to virus attacks is because any user can change the operating system's files

To restrict access to the system we require a notion of *ownership and permission*. Ordinary users should not have access to the hardware devices of a secure operating system's files, only their own files, for then they will not be able to do anything to compromise the security of the system.

Secure operating systems are usually multi-user systems, i.e. operating systems where files and processes can be owned by a particular user, and access is restricted on the basis of user identity.

Unix-like OS	NT
chmod	CACLS
chown	CACLS
chgrp	<i>No direct equivalent.</i>
emacs	<i>Wordpad</i> or emacs in GNU tools
kill	kill command in Resource Kit
ifconfig	ipconfig
lpq	lpq
lpr	lpr
mkfs/newfs	format and label
mount	net use
netstat	netstat
nslookup	nslookup
ps	pstat in Resource Kit
route	route
setenv	set
su	su in resource kit
tar	tar command in cygnus tools
traceroute	tracer

Unix-like OS	NT
Standard libraries	WIN32 API
Unix libraries	Posix compatibility library
Symbolic/hard Links	Hard links (short cuts)
Processes	Processes
Threads	Threads
Long filenames	Long filenames on NTFS
Mount disk on directory	Mount drive A: B: etc
endl is LF	endl is CR LF
UID (User ID)	SID (Subject ID)
groups	groups
ACLs (non standard)	ACLs
Permission bits	(Only in ACLs or with cygwin)
Shared libraries	DLL's
Environment variables	Environment variables

File Systems

Files and file systems are the very basis of what system administration is about

Most file systems (e.g. NT, GNU Linux) are 32-bit addressable and therefore support a maximum file size of 2GB or 4GB, depending on their implementation details, or that newer file systems like Solaris and Netware 5 are 64-bit addressable, and therefore have essentially no storage limits

Unix File Model

Unix has a hierarchical file system which makes use of directories and sub-directories to form a tree.

All file systems on Unix-like operating systems are based on a system of *index nodes*, or *inodes*, in which every file has an index entry stored in a special part of the file system

The top or start of the Unix file tree is called the root file system or '/'.

The File Hierarchy

The main sub-directories of the root directory together with the most important file are shown below.

- **/b** in Executable (binary) programs. On most systems this is a separate directory to **/usr/bin**. In SunOS, this is a pointer (link) to **/usr/bin**.
 - **/etc** Miscellaneous programs and configuration files. This directory has become very messy over the history of Unix and has become a dumping ground for almost anything. Recent versions of unix have begun to tidy up this directory by creating subdirectories **/etc/mail**, **/etc/services**, etc!
 - **/usr** This contains the main meat of Unix. This is where application software lives, together with all of the basic libraries used by the OS.
 - **/usr/bin** More executables from the OS.
 - **/usr/local** This is where users' custom software is normally added.
- /sbin** A special area for statically linked system binaries. They are placed here to distinguish commands used solely by the system administrator from user commands, and so that they lie on the system root partition where they are guaranteed to be accessible during booting.
- **/sys** This holds the configuration data which go to build the system kernel

- **/export** Network servers only use this. This contains the disk space set aside for client machines which do not have their own disks. It is like a 'virtual disk' for diskless clients.
- **/dev and /devices** A place where all the 'logical devices' are collected. These are called 'device nodes' in Unix and are created by mknod. Logical devices are Unix's official entry points for writing to devices. For instance, /dev/console is a route to the system console, while /dev/kmem is a route for reading kernel memory. Device nodes enable devices to be treated as though they were files.
- **/home** (called /users on some systems.) Each user has a separate login directory where files can be kept. These are normally stored under /home by some convention decided by the system administrator.
- **/root** On newer Unix-like systems, root has been given a home-directory which is no longer the root of the file system '/. The name root then loses its logic.

Every unix directory contains two 'virtual' directories marked by a single dot and two dots:

ls -a

• ..

The single dot represents the directory one is already in (the current directory). The double dots mean the directory one level up the tree from the current location

Symbolic Links

A symbolic link is a pointer or an alias to another file. The command

ln -s fromfile /other/directory/tolink

makes the file fromfile appear to exist at /other/directory/tolink simultaneously.

File Access Control

To restrict privilege to files on the system, and create the illusion of a virtual host for every logged-on user, Unix records information about *who creates files and also who is allowed to access them later*.

A file's contents are classified by *magic numbers which are codes kept in the file's inode and defined in the magic number file for the system*

Each user has a unique *username or loginname, together with a unique user id or uid..* A file belongs to user A if it is *owned by user A*. User A then decides whether or not other users can read, write or execute the file by setting the *protection bits or the permission of the file using the command **chmod***.

\$ chmod [*options*] *mode[,mode]* *file1* [*file2 ...*]

In addition to user identities, there are groups of users. The idea of a group is that several named users might want to be able to read and work on a file, without other users being able to access it

Changing File Permissions

Let's take a closer look at the contents of a sample directory by typing the command [ls -l](#) (the “l” stands for “long”).)

total 42

-rwxr-xr-x -		joe acctg		Feb 26	archive.sh
rw-rw-r-- -		joe acctg	23068	2004 Jul 24	orgchart.gi
rw-rw-r-- -	1 1 1 1 2 1	joe acctg	12878	21:58 Jun	f
rw-r--r--	1	joe acctg	2645 168	30 08:48	personnel.t
drwxrwxr-x		joe acctg	1024 512	Jul 17	xt
-rw-r-----		joe acctg	2645	11:51 Mar	publicity.ht
rw-r-xr-x		joe acctg		18 16:27	ml sales
				Sep 1	topsecret.i
				07:00 Aug	nf
				4 11:03	wordmatic

-rwx r-xr-x joe acctg archive.sh
- rw- rw-r-- joe acctg orgchart.gif
- rw- rw-r-- joe acctg
personnel.txt
- rw- r--r-- joe acctg
publicity.html
d rwx r-xr-x joe acctg sales
-rw- r----- joe acctg topsecret.inf
-- rwx r-xr-x joe acctg wordmatic

The first set three letters after the file type tell what you, the owner of the file, have permission to do.

An **r** in the first position means you are permitted to read the file.

A **w** in the second position means you may write the file.

An **x** in the third position means you may execute the file.

A hyphen in any position means that you don't have that particular permission.

Every user is a member of at least one group, called the *login group*, and each group has both a textual name and a number (*group id*). The *uid* and *gid* of each user is recorded in the file `/etc/passwd`

Making Programs Executable

A Unix program is normally executed by typing its pathname. If the x execute bit is not set on the file, this will generate a 'Permission denied' error

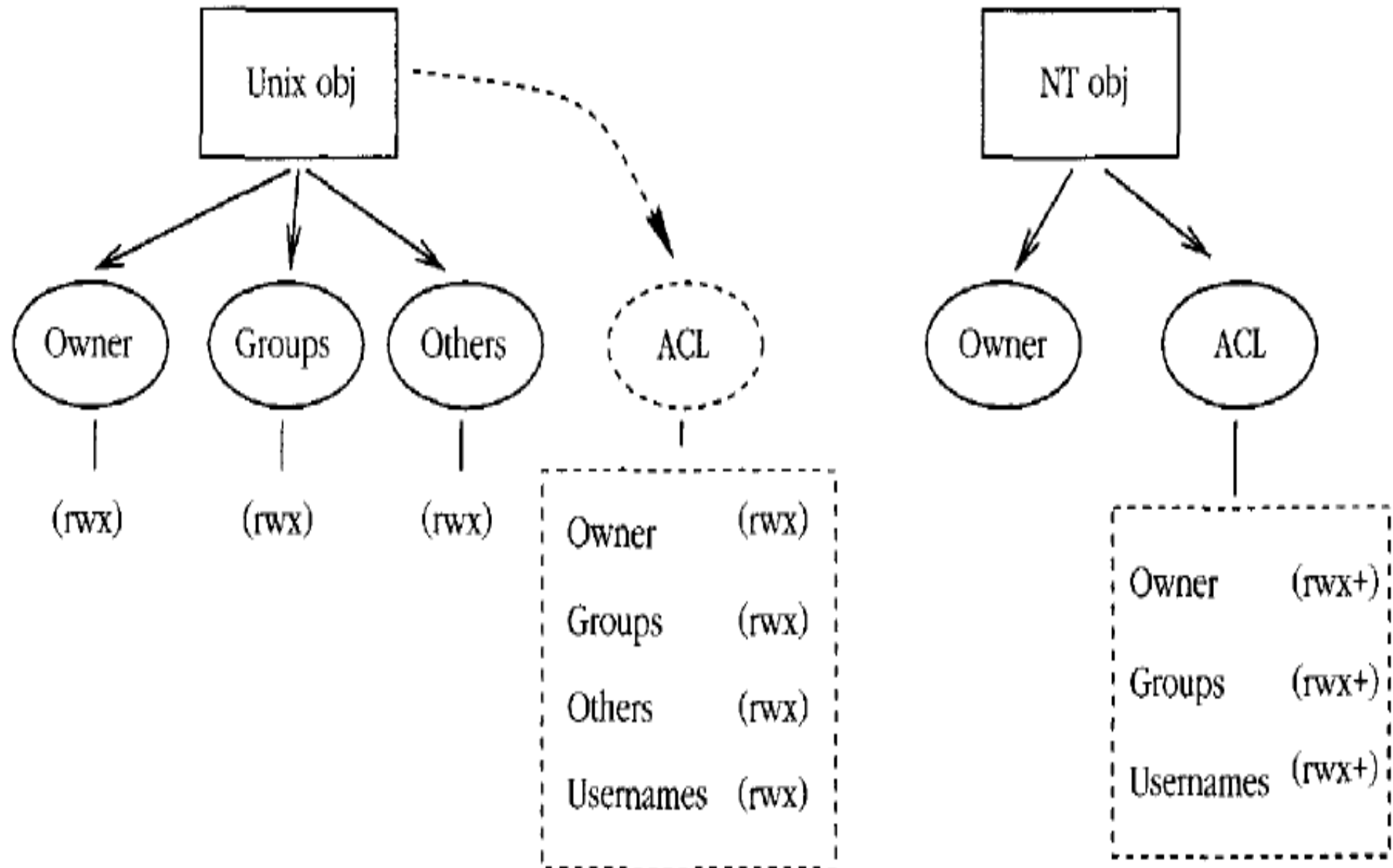
`chmod u+x filename`

This command would set execute permissions for the owner of the file;

`chmod ug+x filename`

would set execute permissions for the owner and for any users in the same group as the file.

Access Control Lists



NT File Model

The file system NTFS was introduced with NT in restricting access to files.

NTFS, like the Unix file system, is a hierarchical file system with files and directories. Each file or directory has an owner, but no group membership

File System Layout

Drawing on its DOS legacy, NT treats different disk partitions as independent disks, labeled by a letter of the alphabet:

C: D: ...

C : is the primary hard disk partition.

The system root is usually stored in

C : \WinNT, and is generally referred to by the system environment variable
%System-Root%:

- **C : \I386** This directory contains binary code and data for the NT operating system. This should normally be left alone.
- **C:\Program Files** This is NT's official location for new software. Program packages which you buy should install themselves in subdirectories of this directory. More often than not, they choose their own locations, often with a distressing lack of discipline.
- **C : \Temp** Temporary scratch space, like Unix's /tmp.
- **C: \WinNT** This is the root directory for the NT system. This is mainly for operating system files, so you should not place new files under this directory yourself unless you really know what you are doing. Some software packages might install themselves here.
- **C:\WinNT\config** Configuration information for programs. These are generally binary files, so the contents of NT configuration files is not very interesting.
- **C: \WinNT\system32** This is the so-called system root. This is where most system applications and data files are kept.

Access Control Lists

NT files and directories have the following attributes. Access control lists are composed of Access Control Entries (ACEs), which consist of these:

Permission bit	Files	Directories
R (Read)	See file contents	See directory contents
W (Write)	Modify file contents	Modify directory contents
X (Execute)	Executable program	Can cd to directory
D (Delete)	Deletable	Deletable
P (Permission)	Permissions changeable	Permissions changeable
O (Ownership)	Ownership changeable	Ownership changeable

/G	Grant access to user
/E	Edit ACE instead of replacing
/T	Act on all files and subdirectories
/R	Revoke (remove) access rights to a user
/D	Deny access rights to a given user

Logs and Audits

Operating system kernels share resources and offer services. They can be asked to keep lists of transactions which have taken place so that one can later go back and see exactly what happened at a given time. This is called logging or auditing.

The use for auditing is so-called *non-repudiation, or non-denial*. If everything on a system is logged, then users cannot back away and claim that they did not do something: it's all there in the log. Non-repudiation is a security feature which encourages users to be responsible for their actions.

Privileged Accounts

Operating systems which restrict user privileges need an account which can be used to configure and maintain the system. Such an account must have access to the whole system, without regard for restrictions. It is therefore called a privileged account.

Principle 1 (Privilege) *Restriction of unnecessary privilege protects a system from accidental and malicious damage, and infection by viruses, and prevents users from concealing their actions with false identities. It is desirable to restrict users' privileges for the greater good of everyone on the network.*

Corollary 2 (Privilege) *No one should use a privileged root/Administrator account as a user account. To do so is to place the system in jeopardy*

One of the major threats to Internet security has been the fact that everyone can now be root/Administrator on their own host. Many security mechanisms associated with trusted ports, TCP/IP spoofing, etc., are now broken, since all of the security of these systems lies in the outdated assumption that ordinary users will not have privileged access to network hardware and the kernel.

Hardware Awareness

To be a system administrator it is not absolutely essential to know much about hardware, but it is very useful to have a basic appreciation of hardware installation procedures and how to nurse-maid hardware later

- *Read instructions: when dealing with hardware, one should always look for and read instructions in a manual.*
- *Interfaces and connectors: hardware is often connected to an interface by a cable or connector. Obtaining the correct cable is of vital importance. Many manufacturers use cables which look similar, superficially, but which actually are different. An incorrect cable can result in damage to an interface.*

Network interfaces are often built-in; they can always be added with expansion cards.

- *Handling components: modern day CMOS chips work at low voltages (typically 5 volts or lower). Standing on the floor with insulating shoes, you can pick up a static electric charge of several thousand volts. Such a charge can instantly destroy computer chips.*

Disks: the most common disk types are IDE (integrated drive electronics) and SCSI (small computer software interface). IDE disks are usually cheaper than SCSI disks, but SCSI disks are more efficient at handling multiple accesses, and are therefore better in multitasking systems.

IDE	2.5
ESDI	3
SCSI-2	5
Fast SCSI-2	10
Fast wide SCSI-2	20
Ultra SCSI	40
Ultra-2 SCSI	80
Ultra-3 SCSI	160

Nominal disk drive speeds in Mbytes/sec [70] for various standards

• **Memory:** *memory chips are sold on small circuit boards called SIMMs. These SIMMs are sold in different sizes, and with different speeds. When buying and installing RAM, remember that*

- The physical size of SIMMs is important. Most have 72 pins and some older SIMMs have 30 pins.

- SIMMs are sold in 1MB, 4MB, 16MB, 64MB sizes, etc. Find out what size you can use in your system. In most cases you are not allowed to mix different sizes.

- Do not buy slower RAM than that which is recommended for your computer, or it will not work.

- There are several incompatible kinds of RAM, FP RAM, EDO RAM, SDRAM, such as which work in different ways. ECC/SDRAM RAM (error correcting code, synchronous dynamic RAM) is tolerant to error from external noise sources like cosmic rays, etc. It can be recommended for important servers.

- On some computers you need to fill up RAM slots in a particular order, otherwise the system will not be able to find them.

System Uniformity

Principle 3 (Uniformity) *A uniform configuration minimizes the number of differences and exceptions one has to take into account later. This applies to hardware and software alike*

PC networks are often a melange of random parts from different manufacturers. If possible, one should standardize graphics and network interfaces, disk sizes, mice and any other devices which have to be configured.

This means that, not only will it be easier to configure and maintain, but also that it 'will be easier to buy extra parts or cannibalize systems for parts later.

With software, the same principle applies: a uniform software base is easier to install and maintain than one in which special software needs to be configured in special ways.

Networked Communities

Communities

System administration is not just about machines and individuals, it is about communities. There is the local community of users on multi-user machines; then there is the local area network community of machines at a site; finally, there is the global community of all machines and networks in the world.

Principle (Communities) *What one member of a cooperative community does affects every other member, and vice versa. Each member of the community therefore has a responsibility to consider the well-being of the other members of the community.*

Principle (Multi-user communities) *A multi-user computer system does not belong to any one user. All users must share the resources of the system. Each user has a responsibility to consider the effect of his/her actions on all the other users.*

Principle 6 (Network communities) *A computer which is plugged into the network is no longer just ours. It is part of a society of machines which shares resources and communicates with the whole. What that machine does affects other machines. What other machines do affects that machine.*

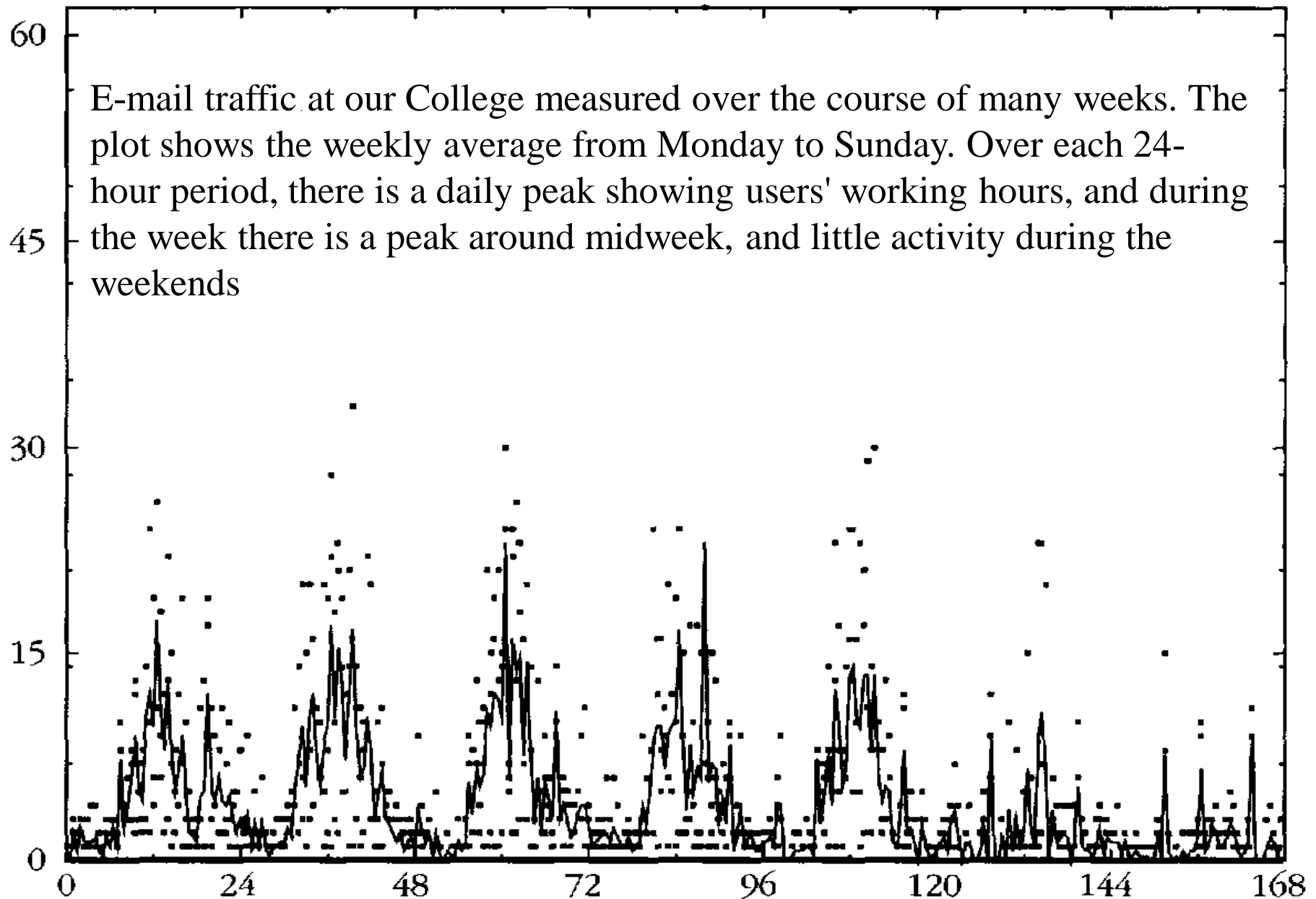
User Sociology

A newly installed machine does not usually require attention until it is first used, but as soon as a user starts running programs and storing data, the reliability and efficiency of the system are tested. This is where the challenge of system administration lies.

The load on computers and on networks is a social phenomenon: it peaks in response to patterns of human behavior. For example, at universities and colleges network traffic usually peaks during lunch breaks, when students rush to the terminal rooms to surf on the web or to read e-mail.

smtp

nexus



Client-Server Cooperation

At the heart of all cooperation in a community is a system of *centralization and delegation*. No program or entity can do everything alone, nor is it expected to do so. It makes sense for certain groups to specialize in performing certain jobs. That is the function of a society

Principle (Delegation I) *Leave experts to do their jobs. Assigning responsibility for a task to a body which specializes in that task is an efficient use of resources*

The client-server nomenclature has been confused by history. A server is not a host, but a program or process which runs on a host. A client is any process which requires the services of a server.

Host Identities and Name Services

A host can have all of the following:

- *Host ID: circuit board identity number. Often used in software licensing.*
- *Install name: configured at install time. This is often compiled into the kernel, or placed in a file like /etc/hostname. Solaris adds to the confusion by also maintaining the install name in /etc/hostname. le0 or an equivalent file for the appropriate network interface, together with several files in /etc/net/*/hosts.*
- *Application level name: any name used by application software when talking to other hosts.*
- *Local file mapping: originally the Unix /etc/hosts file was used to map IP addresses to names, and vice versa. Other systems have similar local files, to avoid looking up on network services.*
- *Network Information Service: a local area network database service developed by Oracle. This was originally called Yellow Pages, and many of its components still bear the 'yp' prefix.*

- *Network level address(es): each network interface (Ethernet/FDDI, etc.) has a hardware address burned into it at the factory, also called its MAC address, or Media Access Control address. Some services (e.g. RARP) will turn this into a name or an IP address through a secondary naming service like DNS.*

- *DNS name(s): the name returned by a domain name server (DNS/BIND) based on an IP address key.*

- *WINS name(s): the name returned by a WINS server (Microsoft's name server) based on the IP address.*

At boot-time, then, each host needs to obtain an Internet identity. It has two choices:

- Ask for an address to be provided from a list of free addresses (DHCP or BOOTP protocols).

- Always use the same IP address, stored on its system configuration files (requires correct information on the disk).

The only worldwide service in common use today is DNS (the Domain Name Service) whose common implementation is called BIND (Berkeley Internet Name Domain). This associates IP addresses with a list of names. Every host in the DNS has a *canonical name*, or official name, and any number of *aliases*. *For instance, a host which runs several important services* might have the canonical name

DNS binds a local network to the worldwide Internet in several important ways. It makes it possible for data to organizations to be spread across the surface of the planet at any location, and yet still maintain a transparent naming structure

Under NT, each system has an alphanumeric name which is chosen during the installation. A domain server will provide an SID (Security ID) for the name which helps prevent spoofing.

When NT boots it broadcasts the name across the network to see whether it is already in use. If the name is in use, the user of the workstation is prompted for a new name(!)

The OSI Model

The International Standards Organization (ISO) has defined a standard model for describing communications across a network, called the OSI model, for *Open Systems Interconnect (reference model)*.

7	Application layer	Program which sends data
6	Presentation layer	XDR or user routines
5	Session layer	RPC / sockets
4	Transport layer	TCP or UDP
3	Network layer	IP Internet protocol
2	Data link layer	Ethernet (protocols)
1	Physical layer	Ethernet (electronics)

Cables and Interface Technologies

Different vendors have invested in different networking technologies, with different Media Access Control (MAC) specifications

- *Bus/Ethernet approach: In the Ethernet bus approach, every host is connected to a common cable or bus. Only one host can be using a given network cable at a given instant.*

It is like a conference telephone call: what goes out onto a network reaches all hosts on that network (more or less) simultaneously, so everyone has to share the line by waiting for a suitable moment to say something.

Ethernet is defined in the IEEE 802.3 standard documents. An Ethernet network is available to any host at any time, provided the line isn't busy. This is called CSMA/CD, or Carrier Sense Multiple Access/Collision Detect

- *Token ring/FDDI approach: in the token ring approach, hosts are coupled to hubs or nodes, each of which has two network interfaces, and the hosts are connected in a unidirectional ring.*

The token ring is described in IEEE 802.5.

A common token ring based interface in use today is the optical FDDI (Fiber Distributed Data Interface). Token rings can pass 16 Mbits/sec, with packet sizes of 18 kilobytes. The larger packet sizes are possible, since there is no risk of collisions.

- *ATM: Asynchronous Transfer Mode technology [18] is a high capacity transmission technology developed by telephone companies to exploit existing copper telephone networks.*

Current ATM implementations use 48-byte cells with 5 bytes of routing information. The cells are of a fixed length, allowing predictable switching times, and each cell has enough information to route it to its destination

ATM is believed to be able to reach transfer rates as high as 10 Gbits/sec.

Connectivity

The cables of the network are joined together in segments by hardware which makes sure that messages are transmitted from cable segment to cable segment in the right direction to reach their destinations.

A host which is coupled to several network segments and which forwards data, from one network to another is called **a router**. *Routers not only forward data, but they prevent the spread of network messages which other network segments do not need to know about.*

A bridge is a hardware device which acts like a filter on busy networks. A bridge separates two segments of the same cable.

A repeater is an amplifier which strengthens the network signal over long stretches of cable.

Switched networks are immune to spies, net-sniffing or network listening devices. A switch performs many of the tasks of a router, and *vice versa*.

Protocols and Encapsulation

On a network, protocols are required to make sure that data are understood, not only by the receiver, but by all the network hardware which carry them between source and destination.

The data are wrapped up in envelope information which contains the address of the destination. Wrapping data inside envelope information is called *encapsulation*

NT supports three network protocols, running on top of Ethernet:

- *NETBEUI: NETBIOS Extended User Interface, Microsoft's own network protocol. This was designed for small networks and is not routable. It has a maximum limit of 20 simultaneous users, and is thus hardly usable.*
- *NWLink/IPX: Novell/Xerox's IPX/SPX protocol suite. Routable. Maximum limit of 400 simultaneous users.*
- *TCP/IP: Standard Internet protocols. The default for NT 4 and Unix-like systems. Novell Netware and Apple Macintosh systems also support TCP/IP. There is no in-built limit to the number of simultaneous users.*

Data Formats

There are many problems which arise in networking when hardware and software from different manufacturers have to exist and work together.

Big endian

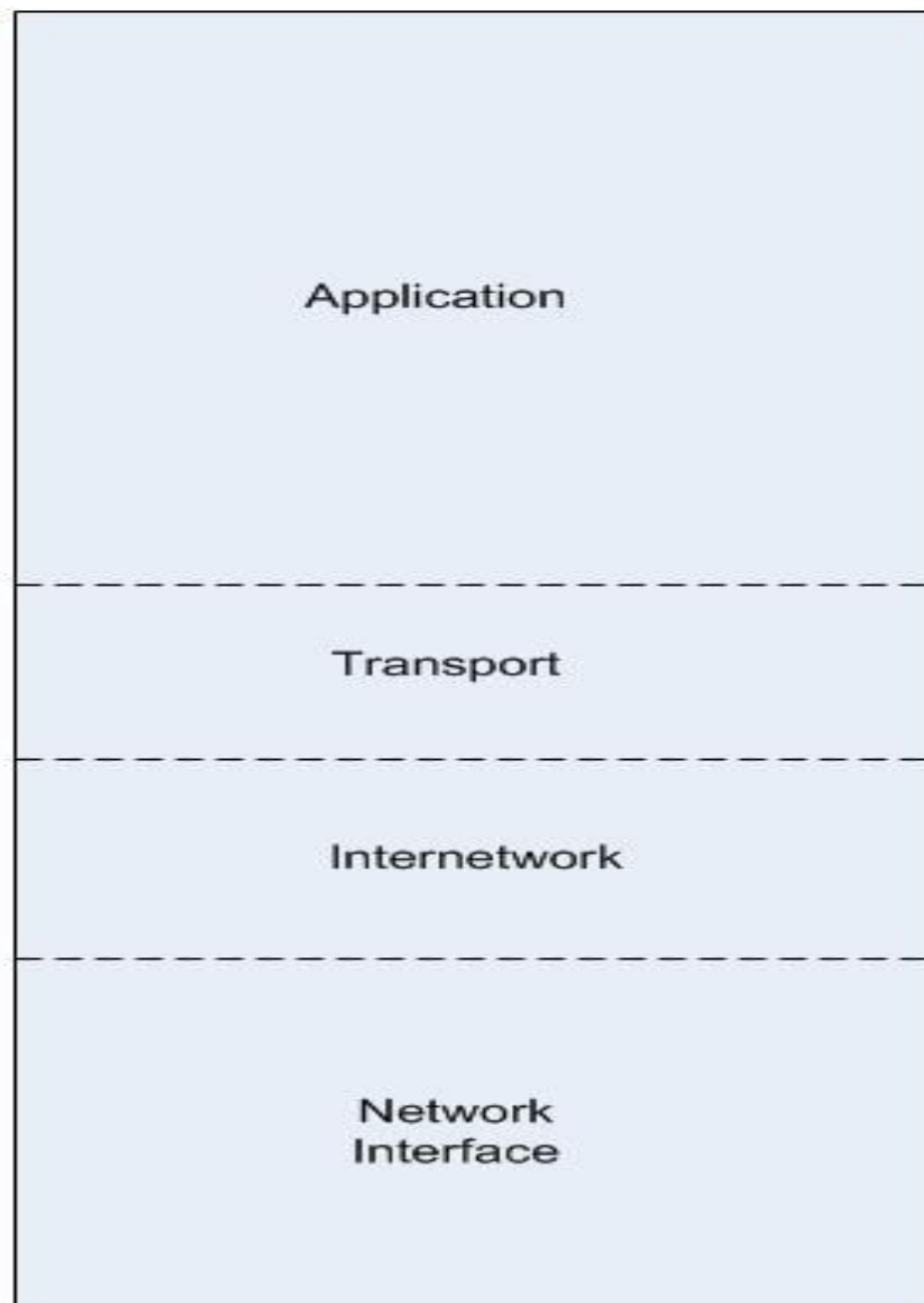
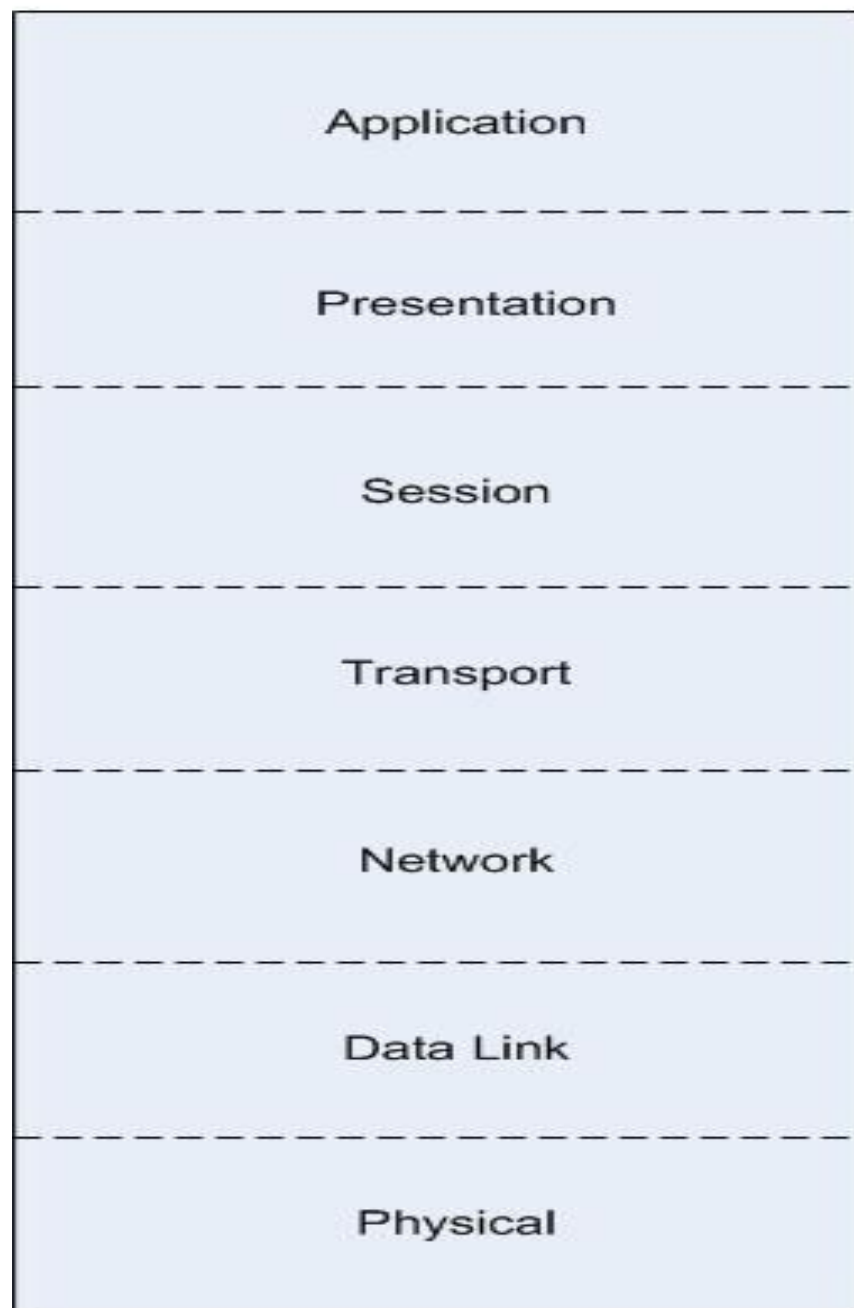
2	17	34	126
---	----	----	-----

Little endian

126	34	17	2
-----	----	----	---

Byte ordering sometimes has to be specified when compiling software. The representation of the number 34,677,374 has either of these forms

Comparing The OSI Model And TCP / IP Architecture.



Internet Protocol

The IP in TCP/IP

- IP is the network layer
 - packet delivery service (host-to-host).
 - translation between different data-link protocols.

An IP packet is called a *datagram*


IP Datagrams

- IP provides connectionless, unreliable delivery of *IP datagrams*.
 - Connectionless: each datagram is independent of all others.
 - Unreliable: there is no guarantee that datagrams are delivered correctly or even delivered at all.

IP Addresses

- IP is a network layer - it must be capable of providing communication between hosts on different kinds of networks (different data-link implementations).
- The address must include information about what *network* the receiving host is on. This is what makes routing feasible.

IP Addresses

- IP addresses are *logical* addresses (not physical)
- 32 bits.  IPv4 (*version 4*)
- Includes a network ID and a host ID.
- Every host must have a unique IP address.
- IP addresses are assigned by a central authority (*American Registry for Internet Numbers* for North America).

The *four* formats of IP Addresses

Class



8 bits

8 bits

8 bits

8 bits

● Class A

- ❑ 128 possible network IDs
- ❑ over 4 million host IDs per network ID

Class B

- ❑ 16K possible network IDs
- ❑ 64K host IDs per network ID

Class C

- ❑ over 2 million possible network IDs
- ❑ about 256 host IDs per network ID

Network and Host IDs

- A Network ID is assigned to an organization by a global authority.
- Host IDs are assigned locally by a system administrator.
- Both the Network ID and the Host ID are used for routing.

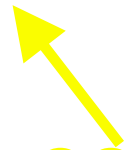
IP Addresses

- IP Addresses are usually shown in *dotted decimal* notation:

1.2.3.4 00000001 00000010 00000011 00000100

- cs.rpi.edu is 128.213.1.1

10000000 11010101 00000001 00000001



CS has a class B network

Host and Network Addresses

- A single network interface is assigned a single IP address called the *host* address.
- A host may have multiple interfaces, and therefore multiple *host* addresses.
- Hosts that share a network all have the same IP *network* address (the network ID).

IP Broadcast and Network Addresses

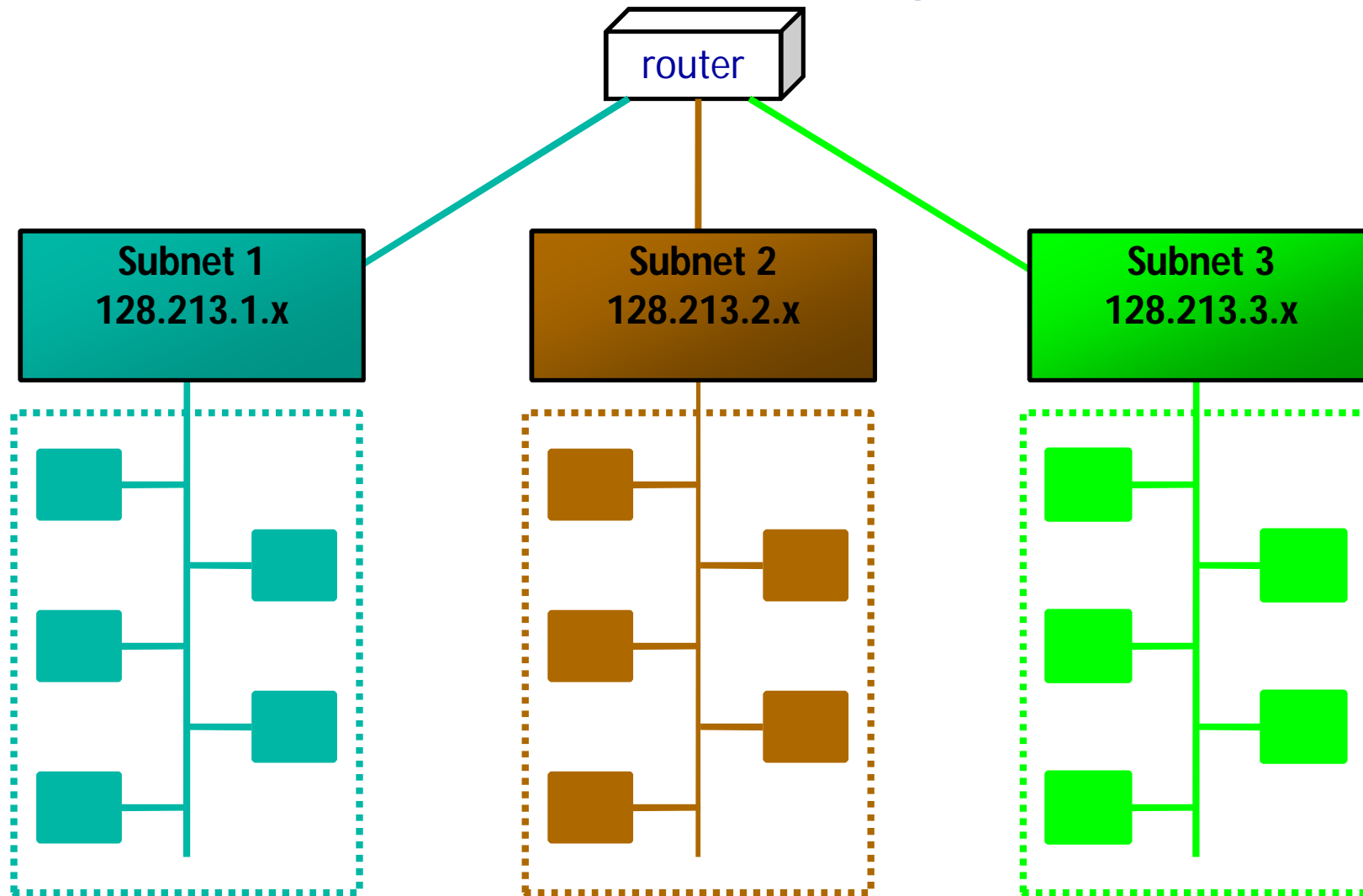
- An IP broadcast address has a host ID of all 1s.
- IP broadcasting is not necessarily a true broadcast, it relies on the underlying hardware technology.
- An IP address that has a host ID of all 0s is called a *network address* and refers to an entire network.

Subnet Addresses

- An organization can subdivide it's host address space into groups called subnets.
- The subnet ID is generally used to group hosts based on the physical network topology.

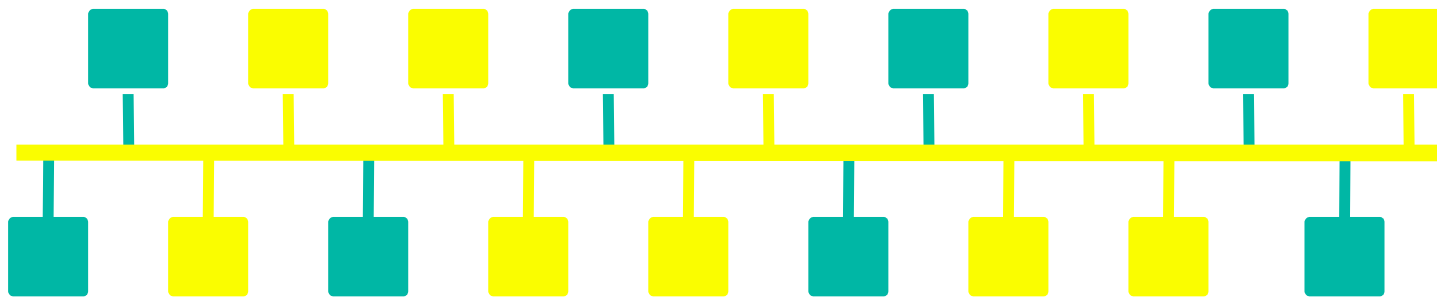


Subnetting



Subnetting

- Subnets can simplify routing.
- IP subnet broadcasts have a hostID of all 1s.
- It is possible to have a single wire network with multiple subnets.



Mapping IP Addresses to Hardware Addresses

- IP Addresses are not recognized by hardware.
- If we know the IP address of a host, how do we find out the hardware address ?
- The process of finding the hardware address of a host given the IP address is called

Address Resolution

Reverse Address Resolution

- The process of finding out the IP address of a host given a hardware address is called

Reverse Address Resolution

- Reverse address resolution is needed by diskless workstations when booting (which used to be quite common).

ARP

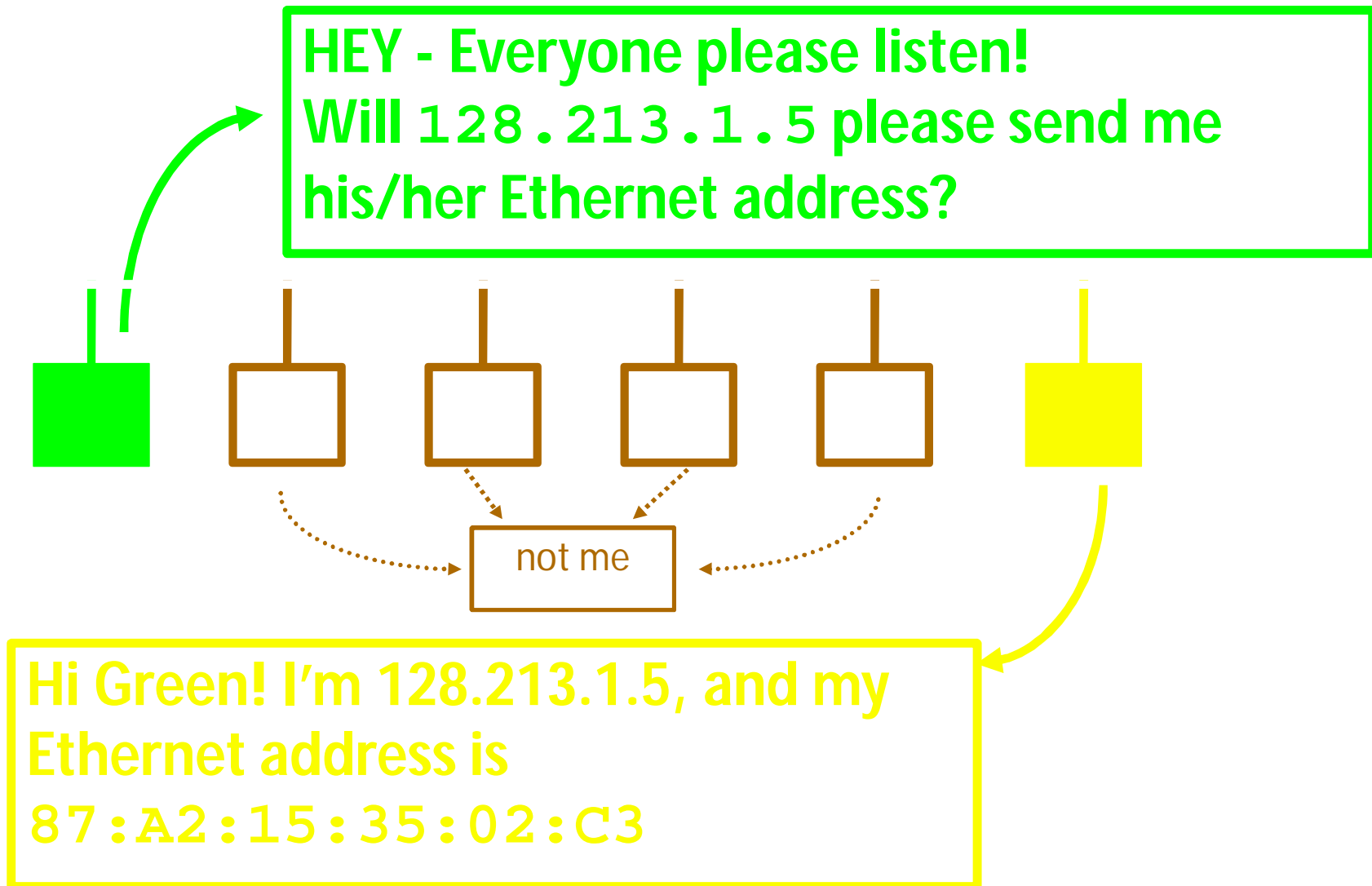
Arp Arp!

- The *Address Resolution Protocol* is used by a sending host when it knows the IP address of the destination but needs the Ethernet (or whatever) address.
- ARP is a broadcast protocol - every host on the network receives the request.
- Each host checks the request against its IP address - the right one responds.

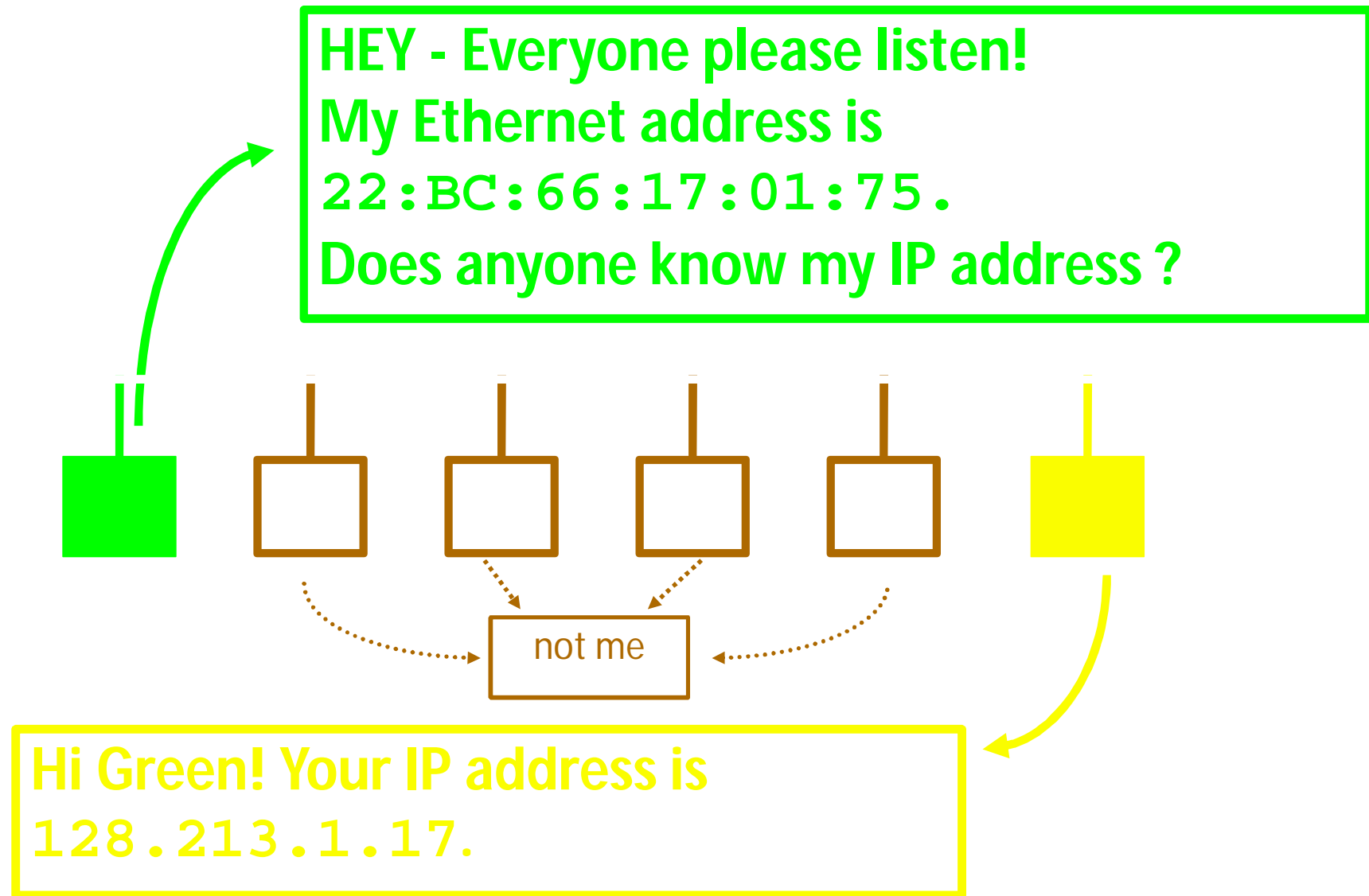
ARP (cont.)

- ARP does not need to be done every time an IP datagram is sent - hosts *remember* the hardware addresses of each other.
- Part of the ARP protocol specifies that the receiving host should also remember the IP and hardware addresses of the sending host.

ARP conversation



RARP conversation



Services provided by IP

- Connectionless Delivery (each datagram is treated individually).
- Unreliable (delivery is not guaranteed).
- Fragmentation / Reassembly (based on hardware MTU).
- Routing.
- Error detection.

0.0.0.0	<i>Default route</i>
0.*.*.*	<i>Not used</i>
127.0.0.1	<i>Loopback address</i>
127.*.*.*	<i>Loopback network</i>
..*.0	<i>Network addresses (or old broadcast)</i>
..*.255	<i>Broadcast addresses</i>
..*.1	<i>Router or gateway (conventionally)</i>
224.*.*.*	<i>Multicast addresses</i>

Class B address

Net	Net	Host	Host
-----	-----	------	------

Subnet mask 255.255.254.0

1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 0	0 0 0 0 0 0 0 0
-----------------	-----------------	-----------------	-----------------

Interpretation

Net id	Net id	Subnet	Host
--------	--------	--------	------

Example of how the subnet mask can be used to double up the number of hosts per subnet by pairing host parts. The boundary between host and subnet parts of the address is moved one bit to the left, doubling the number of hosts on the subnets which have this mask

C:\WINDOWS\system32\cmd.exe

(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\admin>ipconfig -all

Windows IP Configuration

Host Name : SITEYAJ1
Primary Dns Suffix :
Node Type : Unknown
IP Routing Enabled. : No
WINS Proxy Enabled. : No

Ethernet adapter Wireless Network Connection:

Connection-specific DNS Suffix . :
Description : Intel(R) PRO/Wireless 3945ABG Network
k Connection
Physical Address. : 00-1C-BF-10-CF-69
Dhcp Enabled. : No
IP Address. : 192.168.146.1
Subnet Mask : 255.255.240.0
Default Gateway : 192.168.144.1
DNS Servers : 192.168.64.4
192.168.64.240

```
C:\Documents and Settings\admin>netstat -r
```

Route Table

Interface List

```
0x1 ..... MS TCP Loopback interface
0x2 ...00 1c bf 10 cf 69 ..... Intel(R) PRO/Wireless 3945ABG Network Connection
- Packet Scheduler Miniport
```

Active Routes:

Network Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.144.1	192.168.146.1	25
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.144.0	255.255.240.0	192.168.146.1	192.168.146.1	25
192.168.146.1	255.255.255.255	127.0.0.1	127.0.0.1	25
192.168.146.255	255.255.255.255	192.168.146.1	192.168.146.1	25
224.0.0.0	240.0.0.0	192.168.146.1	192.168.146.1	25
255.255.255.255	255.255.255.255	192.168.146.1	192.168.146.1	1

Default Gateway: 192.168.144.1

Persistent Routes:

None

Network Analysis

A top down approach is useful for understanding network inter-relationships

- How does the network fit together? (What is its topology?)
- How many different subnets does the network have?
- What are their network addresses?
- Find the router addresses (the default routes) on each segment.
- What is the netmask?
- What hardware is there in the network? (Hosts, printers, etc.)
- Which function does each host/machine have on the network?
- Where are the key network services located?

Most newer network hardware supports some kind of querying using SNMP protocols

An overview of network services can sometimes be obtained using port-scanning software, such as nmap

We need to know hosts both from the viewpoint of hardware and software:

- What kind of machines are on the network? What are their names and addresses and where are they? Do they have disks. How big? How much memory do they have? If they are PCs, which screen cards do they have?
- How many CPUs do the hosts have?
- What operating systems are running on the network? MS-DOS, Novell, NT or Unix (if so, which Unix? GNU/Linux, Solaris, HP-UX?)
- What kind of network cables are used? Is it thin/thick Ethernet? Is it a star net (hubs/twisted pair), or fibre optic FDDI net?
- Where are hubs/repeaters/the router or other network control boxes located? Who is responsible for maintaining them?
- What is the hierarchy of responsibility?

There is information about the local environment:

- What is the local time zone?
- What broadcast address convention is used? 255 or the older 0?
- Find the key servers on these networks:
 - Where are the NFS network disks located? Which machine are they attached to?
 - Which name service is in use (DNS, NIS or NIS plus)?
 - Where is the inevitable WWW/http service? Who is running pirate servers?

Network Orientation

Familiarizing oneself with an organization's network involves analyzing the network's hosts and all of their inter-relationships.

The most obvious way to view an organization is by its logical structure.

This is usually reflected in the names of different machines and domains

Who do we call if the Internet connection is broken?

What service contracts exist on hardware

what upgrade possibilities are there on software?

What system is in use for making backups?

How does one obtain a backup should the need arise?

The Domain Name Service (DNS) is the Internet's primary naming service. It not only allows us to name hosts, but also whole organizations, placing many different IP addresses under a common umbrella.

The DNS is thus a hierarchical organization of machine names and addresses.

Organizations are represented by *domains* and a domain is maintained either by or on behalf of each organization.

DNS – the Domain Name System

- Which is easier to remember:
 - www.swan.ac.uk or 137.44.1.20?
- Names also provide transparency:
 - eg can move www.swan.ac.uk to 137.44.1.21
- Early days: `hosts.txt`
- Lists all hosts and their IP addresses
- Every night, each host fetches new copy from central location
- OK for a few hundred machines
 - More: bottleneck, single point of failure, clashes, ...

DNS Servers

- Primary only one
- Many secondary

DNS essentials

- RFC 1034, 1035, others
- Maps hostnames to IP addresses (and back)
- Also says where to deliver mail for a host
- 1. A hierarchical, domain-based naming scheme
 - Domains responsible for managing subdomains
- 2. Distributed database system for implementing the scheme
 - Basic data type is the Resource Record
- 3. A protocol for querying the database system
 - Binary messages, UDP transport

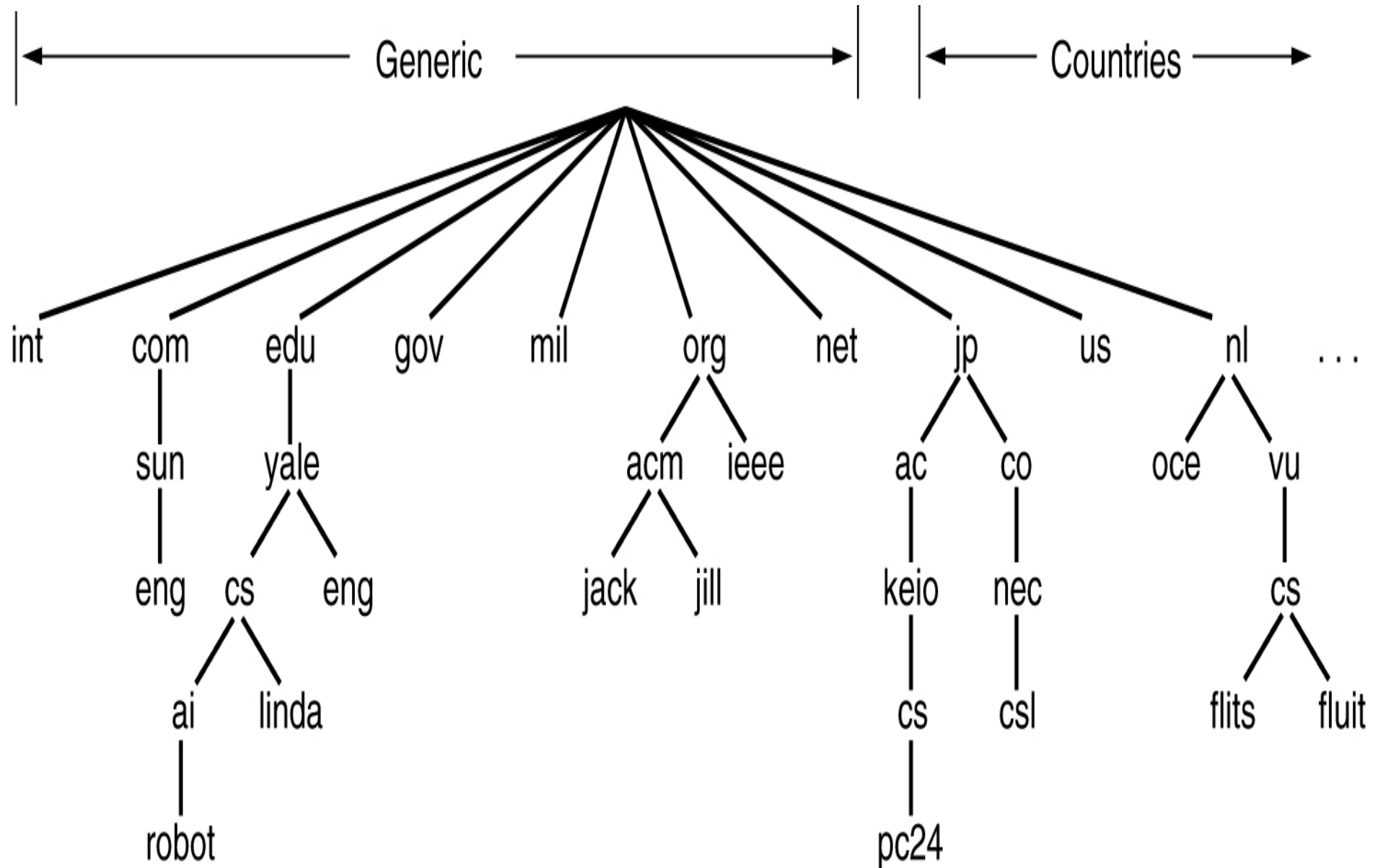
Example: DNS as used by HTTP

- Surf: `http://www.swan.ac.uk/index.html`
- Browser extracts `www.swan.ac.uk` from URL, passes it to DNS client code
 - (eg `gethostbyname()` call)
- DNS client code sends query to DNS server
- DNS server replies: `137.44.1.7`
- Web browser establishes TCP connection to `137.44.1.7` port 80
- Every single time! Adds time to operation, sometimes long time...

The DNS Name Space

- Over 200 Top Level Domains
 - com, org, edu, net, ..., uk, nl, jp, cn, ...
- Each domain controlled by a registrar, who assigns ownership of subdomains
 - eg want hatsforcats.com ? Pay .com registrar \$\$\$
- Owner of a domain can then divide it further/sell subdomains/etc
 - eg fluffy.hatsforcats.com
 - eg ac.uk -> swan.ac.uk, -> cs.swan.ac.uk
- Organisational boundaries, not physical networks

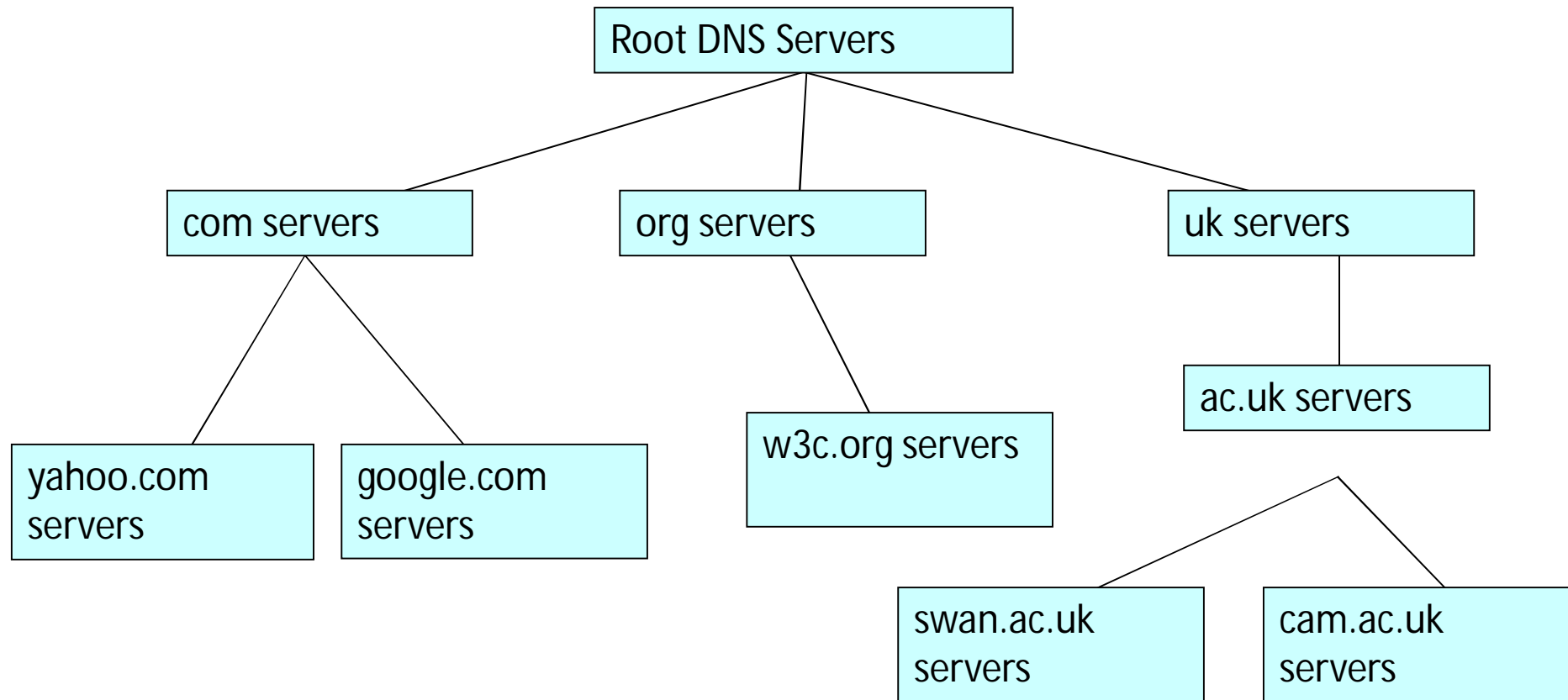
Part of the DNS Name Space



Services provided by DNS

- Hostname IP resolution (and reverse)
- Host aliasing
 - eg, `www.swan.ac.uk` might *really* be called `funky.swan.ac.uk`
 - then `www.swan.ac.uk` -> `funky.swan.ac.uk` -> `137.44.1.7`
- Mail server aliasing
 - Where does email for `webmaster@www.swan.ac.uk` go?
 - Might be `funky.swan.ac.uk`, but doesn't have to be
- Load distribution
 - Some sites use multiple redundant servers, return many IP addresses for one hostname
 - Typically only first is used: DNS server rotates the list, so have round robin

Part of the DNS Server Hierarchy



NSLOOKUP

Special Information

The domain name service identifies certain special hosts which perform services like the name service itself and mail-handlers (called mail exchangers).

These servers are identified by special records so that people outside of a given domain can find out about them. After all, the mail service in one domain needs to know how to send mail to a neighboring domain.

```
> set q=mx
```

```
> otherdomain.org
```

```
Server: mother.example.org
```

```
Address: 192.0.2.10
```

```
Non-authoritative answer:
```

```
otherdomain.org preference = 0, mail exchanger =  
mercury.otherdomain.org
```

```
Authoritative answers can be found from:
```

```
otherdomain.org nameserver =mercury.otherdomain.org
```

```
otherdomain.org nameserver = delilah.otherdomain.org
```

```
mercury.otherdomain.org internet address = 158.36.85.10
```

```
delilah.otherdomain.org internet address = 129.241.1.99
```

Listing Hosts Belonging to a Domain

To list every registered Internet address and hostname for a given domain one can use the `Is`

command inside `nslookup` . For instance

```
> Is example.org
```

```
[mother.example.org]
```

```
example.org. server =mother.example.org
```

```
example.org. server =mercury.otherdomain.org
```

```
pc61 192.0.2.61
```

```
pc59 192.0.2.59
```

```
pc59 192.0.2.59
```

```
pc196 192.0.2.196
```

Changing to a Different Server

If we know the name of a server which contains authoritative information for a domain, we can tell ns lookup to use that server instead. In that way it might be possible to list the hosts in a remote domain and find out detailed information about it. At the very least, it is possible to find out about key records, like name servers and mail exchangers (MX). To change the server we simply type

> server *new-server*

Planning Network Resources

Mapping Out Services

Existing network services have to be analyzed, so that we know where we are starting from, and new networks need to be planned or extended.

Location can be performed by a visual inspection of the process tables, or from configuration files. There are tools for port scanning networks in order to locate services, e.g. the nmap program.

- How to choose the right hardware for the right job.
- Which hosts should be servers and for which services.
- How to make disks available to the network.
- How to share tasks between machines.
- How clock/time synchronization will work.

The efficiency of a network can be improved greatly by planning carefully how key networks services are organized: particularly file servers and name services, which form the basic infrastructure of a network. Here is a partial checklist:

- Which hosts keep the physical disks for NFS disk servers? It makes sense to keep all file services which use those disks on that same host. If the source data are on host A, then we run all file services for those data on host A, otherwise data will first have to be copied from A to B, over the network, in order to be served back over the network to host C, i.e. there will be an unnecessary doubling of traffic.
- Normally we shall want to use a powerful system for the servers which provide key disk and WWW services, since these are at the heart of network infrastructure. Other hosts depend upon these.

However, if resources are limited we might need to reserve the fastest host for running some especially heavy software. This has to be a site dependent calculation.

Uniform Resource Locators (URLs)

In a network the name of a partition ought to be a URL, i.e. contain the name of the host

Principle (One name for one object I) *Each unique resource should have a unique name which labels it and describes its function.*

Choosing Server-Hosts

Choosing the best host for a service is an issue with several themes. The main principles have to do with efficiency and security, and can be summarized by the following questions:

- Does traffic have to cross subnet boundaries?
- Do we avoid unnecessary network traffic?
- Have we placed insecure services on unimportant hosts?

Principle 11 (Inter-dependency) *Avoid making one service reliant on another. The more independent a service is, the more efficient it will be, and the fewer possibilities there will be for its failure*

Place all file servers which serve the same data on a common host, e.g. WWW, FTP and NFS serving user files. Place them on the host which physically has the disks attached. This will save an unnecessary doubling of network traffic and will speed up services. A fast host with a lot of memory and perhaps several CPUs should be used for this

Distributed File Systems and Mirroring

The purpose of a network is to share resources amongst many hosts. Making files available to all hosts from a common source is one of the most important issues in setting up a network community. There are three types of data which we have to consider separately:

- Users' home directories.
- Software or binary data (architecture specific).
- Other common data (architecture unspecific).

How are network data shared? There are two strategies:

- Use of a shared file system (e.g. NFS or Novell Netware).
- Disk mirroring.

Host Management

Choices

We can make life easy or difficult for ourselves by the decisions we make at the outset of host installation. The first step in setting up hosts is to make some basic choices. Should we:

- Follow the OS designer's recommended setup? (Often not good enough.)
- Create our own setup?
- Make all machines alike?
- Make all machines different?

Booting and Shutting Down NT

The NT boot procedure on a PC begins with the BIOS, or PC hardware. This performs a memory check and looks for a bootable disk. A bootable disk is one which contains a Master Boot Record (MBR).

The boot block is located in the first sector of the bootable drive. It identifies which partition is to be used to continue with the boot procedure.

On each primary partition of a bootable disk, there is a boot program which 'knows' how to load the operating system it finds there. NT has a menu-driven boot manager program which makes it possible for several OSes to coexist on different partitions

Once the disk partition containing NT has been located, the program NTLDR is called to load the kernel. The file BOOT. INI configures the defaults for the boot manager

In organizing disk space, we can make the best use of resources, and separate:

- Space for the operating system.
- Space which can be shared and made available for all hosts.
- Space which can be used to optimize local work, e.g. temporary scratch space, space which can be used to optimize local performance (avoid slow networking).
- Space can be used to make distributed backups, for multiple redundancy.

Partitioning

Disks can be divided up into partitions. Partitions physically divide the disk surface into separate areas which do not overlap.

The disk controller makes sure that partitions behave as independent, logical disks. The main difference between two partitions on one disk and two separate disks is that partitions can only be accessed one at time, whereas multiple disks can be accessed in parallel.

Here are some practical points to consider when partitioning disks:

- Size partitions appropriately for the jobs they will perform. Bear in mind that operating system upgrades are almost always bigger than previous versions, and that there is a general tendency for everything to grow.
- Bear in mind that RISC (e.g. Sun Spare) compiled code is much larger than CISC compiled code (e.g. GNU/Linux), so software will take up more space on a RISC system.
- If we are sharing partitions, there might be limits on partition sizes. Some older implementations of NFS could handle file systems larger than 2GB, owing to the 32-bit limitation. This is not normally a problem now, though individual files cannot exceed 4GB on a 32-bit OS.
- Consider how backups will be made of the partitions. It might save many complications if disk partitions are small enough to be backed up in one go with a single tape, or other backup device.

Formatting and Building File Systems

Disk formatting is a way of organizing and finding a way around the surface of a disk

On a disk surface, it makes sense to divide up the available space into sectors or blocks.

Regrouping and labeling procedure is called *formatting in PC culture*, and is called *making a file system in Unix culture*

Making a file system also involves setting up an infrastructure for creating and naming files and directories.

A file system is not just a labeling scheme, it also provides functionality.

Swap Space

In Windows operating systems, virtual memory uses file system space for saving data to disk.

In Unix-like operating systems, a preferred method is to use a whole, unformatted partition for virtual memory storage

File System Layout

A working computer system has several facets:

- The operating system software distribution.
- Third party software.
- Users' files.
- Information databases.
- Temporary scratch space.

These are logically separate because

- They have different functions.
- They are maintained by different sources.
- They change at different rates.
- A different policy of backup is required for each.

Principle (Separation I) *Data which are separate from the operating system should be kept in a separate directory tree, preferably on a separate disk partition. If they are mixed with the operating system file-tree it makes re-installation or upgrade of the operating system unnecessarily difficult*

Diskless Clients

Diskless workstations are, as per the name, workstations which have no disk at all. Diskless workstations know absolutely nothing other than the MAC address of their network interface (Ethernet address).

Dual Homed Host

A host with two network interfaces, both of which is coupled to a network, is called a dualhomed host. Dual homed hosts are important in building *firewalls for network security*

Cloning Systems

We are almost never interested in installing every machine separately. A system administrator usually has to install ten, twenty or even a hundred machines at a time

- A few Unix-like operating systems provide a solution to this using package templates so that the installation procedure becomes standardized.
- The hard disks of one machine can be physically copied, and then the host name and IP address can be edited afterwards.
- All software can be placed on one host and shared using NFS, or another shared file system.

User Management

User Registration

One of the first issues on a new host is to issue accounts for users

The need for centralization is often in conflict with the need for delegation of responsibility. It is convenient for autonomous departments to be able to register their own users, but it is also important for all users to be registered under the umbrella of the organization, to ensure unique identities for the users and flexibility of access to different parts of the organization

Registration

of single users under NT can be performed remotely from a workstation, using the ***net user username password /ADD /domain***

Local and Network Accounts

Most organizations need a system for centralizing passwords, so that each user will have the same password on each host on the network

With a local account, a user has permission to use only the local host.

With a network account, the user can use any host which belongs to a network *domain*.

In NT the Security Accounts Manager (SAM) is used to add local accounts to a given workstation.

For network accounts, Unix-like systems have widely adopted Sun Microsystems' **Network Information Service (NIS)**

An NT domain server involves not only shared databases, but also shared administrative policies and shared security models. A host can subscribe to one or more domains, and one domain can be associated with one another by a trust relationship

Principle (Distributed accounts) *Users move around from host to host, share data and collaborate. They need easy access to data and workstations all over an organization.*

Suggestion 4 (Passwords) *Give users a common username on all hosts, of no more than eight characters. Give them a common password on all hosts, unless there is a special reason not to do so. Some users never change their passwords unless forced to, and some users never even log in, so it is important to assign good passwords initially. Never assign a simple password and assume that it will be changed.*

Unix Accounts

To add a new user to a Unix-like host we have to

- Find a unique user name, user-id (uid) number and password for the new user.
- Update the system database of user accounts, e.g. add a line to the file `/etc/passwd` for Unix (or on the centralized password server of a network) for the new user.
- Create a login directory (home directory) for the user.
- Choose a shell for the user (if appropriate).
- Copy some configuration files like `.cshrc` or `.profile` into the new user's directory, or update the system registry.

NT Accounts

Single NT accounts are added with the command `net user username password /ADD /domain`

Groups of Users

Both Unix and NT allow users to belong to multiple groups. A group is an association of user names which can be referred to collectively by a single name. File and process permissions can be granted to a group of users.

Groups are created by command, rather than by file editing, using

```
net group groupname /ADD
```

Users may then be added with the syntax

```
net group groupname username1 username2. . . /ADD
```

Some standard groups are defined by the system, e.g.

Administrators

Users

Guest

The Administrators group has privileged access to the system.

Account Policy

Most organizations need a strict policy for assigning accounts and opening the system for users.

Closing Unix accounts can be achieved simply by changing their default shell in /etc/passwd with a script such as

```
#!/bin/sh
```

```
echo "/local/bin/blocked.passwd was run" | mail sysadm
```

```
/usr/bin/last -10 | mail sysadm
```

```
message='
```

You account has been closed because your password was found to be vulnerable to attack. To reopen your account, visit the admin office , carrying some form of personal identification.

```
echo "$message"
```

```
sleep 10
```

```
exit 0
```

Unix Environment

Here is a simple checklist for configuring a user environment.

- `.cshrc` If the default shell for users is a C shell or derivative, then we need to supply a default 'read commands' file for this shell. This should set a path which searches for commands, a terminal type and any environment variables which a local system requires.
- `.profile` If the default shell is a Bourne-again shell like bash or ksh, then we need to supply this file to set a PATH variable, terminal type and environment variables which the system requires.
- `.xsession` This file is read by the Unix xdm login service. It specifies what windows and what window manager will be used when the X-windows system is started.
 - `.mwmrc` This file configures the default menus, etc., for the mwm window manager.
 - `.fvwmrc` This file customizes the behaviour of the fvwm window manager.

Principle 19 (Environment) *It should always be clear to users which host they are using and what operating system they are working with. Default environments should be kept simple both in appearance (prompts, etc.,) and in functionality (specially programmed keys, etc.). Simple environments are easy to understand.*

Suggestion 5 (Clear prompts) *Try to give users a command prompt which includes the name of the host they are working on. This is important, since different hosts might have different operating systems, or different files. Including the current directory in the prompt, like DOS, is not always a good idea. It uses up half the width of the terminal and can seem confusing. If users want the name of the current directory in the prompt, let them choose that. Don't assign it as a default.*

The Superuser's Environment

What kind of user environment should the superuser have? As we know, a privileged account has potentially dangerous consequences for the system. From this account, we have the power to destroy the system, or sabotage it. In short, the superuser's account should be configured to avoid as many casual mistakes as possible.

Disk Space

Disks fill up at an alarming rate. Users almost never throw away files unless they have to

Suggestion 7 (Problem users) *Keep a separate partition for problem users' home directories, so that they only cause trouble for one another, not for more considerate users.*

Quotas and Limits

One way of protecting operating systems from users and from faulty software is to place quotas on the amount of system resources which they are allowed:

- *Disk quotas: place fixed limits on the amount of disk space which can be used per user.* The advantage of this is that the user cannot use more storage than this limit; the disadvantage is that many software systems need to generate/cache large temporary files (e.g. compilers or web browsers), and a fixed limit means that these systems will fail to work as a user approaches his/her quota.

- *CPU time limit: some faulty software packages leave processes running which consume valuable CPU cycles to no use. Users of multi-user computer systems occasionally steal CPU time by running huge programs which make the system unusable for others. The C-shell limit cpu-time function can be globally configured to help prevent accidents.*
- *Policy decisions: users collect garbage. To limit the amount of it, one can specify a system policy which includes items of the form: 'Users may not have mp3, wav, mpeg, etc., files on the system for more than one day*

Killing Old Processes

Processes sometimes do not get terminated when they should.

One way to clean up processes in a work environment is to look for user processes which have run for more than a day.

Cfengine can also be used to clean up old processes. Cfengine's processes commands are used to match processes in the process table (which can be seen by running `ps ax` on Unix).

Moving Users

When disk partitions become full, it is necessary to move users from old partitions to new ones

```
cd /site/host/home-old
```

```
du -s *
```

Having chosen a user, with username *user*, we copy the directory to its new location,

```
tar cf - user \ (cd/site/host/home-new; tar xpvf - )
```

edit the new location in the password file,

```
emacs /etc/passwd
```