

NETWORK MANAGEMENT AND ADMINISTRATION

Chapter 2

NETWORK SERVICES

Detailed Notes with Diagrams, 15 MCQs and 10 Broad Questions

For Engineering / Diploma Students

PolyNotesHub

Contents

2.1 Managing the X Window System	3
2.2 Configuring Printer	6
2.3 TCP/IP Networking	7
2.4 The Network File System (NFS)	11
2.5 Network Related Jobs	13
2.6 Directory Services	15
2.7 Active Directory Architecture	19
2.8 Remote Network Access	21
2.9 Virtual Private Network (VPN)	24
Chapter Review Questions	27

Topics covered in this chapter

- X Window System, X server configuration (tool and manual)
- Printer configuration (CUPS)
- TCP/IP networking: network classes, network configuration, directory services, remote access
- Network File System: server, client and security
- Network related jobs
- Directory services: NDS, NT domains, Active Directory, X.500, LDAP, forests and trees, Samba
- Active Directory architecture: objects, naming, GUID, UPN, domains, trees and forests
- Remote network access: PSTN, ISDN, DSL, CATV
- Virtual Private Networks: protocols, types, clients and SSL VPNs

2.1 Managing the X Window System

The **X Window System** (also called **X11** or simply **X**) is the graphical windowing system used by Linux and UNIX. It was developed at MIT in 1984 and is currently maintained by the X.Org Foundation. X provides the basic framework for a graphical user interface (GUI): drawing windows on the screen, moving them, and handling keyboard and mouse input. X does *not* decide how windows look; that job is done by a separate program called the **window manager** (for example GNOME, KDE, Xfce) and by the desktop environment.

The most important feature of X is that it is **network transparent**. A program (the X client) can run on one computer while its window is displayed on another computer (the X server). Hence, a network administrator can run graphical tools on a remote server and see them on a local desktop.

Architecture of X (client-server model)

The words *client* and *server* are used in a slightly unusual way in X. The **X server** is the program that runs on the user's machine and controls the display hardware. The **X clients** are the applications that request the server to draw things. They communicate using the **X protocol**.

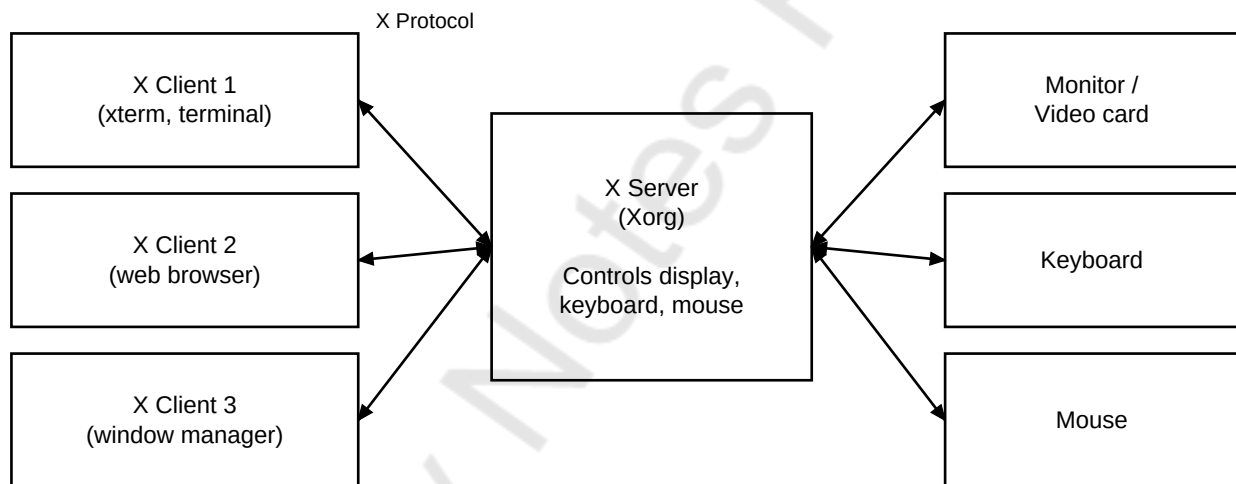


Fig 2.1: Client-server architecture of the X Window System

Component	Role
X Server	Runs on the machine with the screen, keyboard and mouse. Draws graphics, takes input from the user and sends it to clients (Xorg is the modern implementation).
X Client	Any graphical application (xterm, browser, editor). Sends drawing requests to the server and receives events such as key presses.
Window Manager	A special X client that draws window borders, title bars, buttons and handles moving, resizing and minimising windows.
Display Manager	Provides the graphical login screen (gdm, kdm, lightdm, xdm) and starts the X server at boot time.
X Protocol	The language in which clients and server talk to each other. Works over a local socket or over TCP (port 6000 + display number).

Starting X and the DISPLAY variable

- **startx** starts X from a text console (runlevel 3). Graphical login (runlevel 5) is handled automatically by the display manager.
- The environment variable **DISPLAY** tells a client which server to use. Format: `hostname:display.screen`, for example `:0.0` for the local first display.
- **xhost** controls which hosts may connect to your X server (for example `xhost +192.168.1.20`). Using `xhost +` allows everyone and is a security risk.
- The safer way of remote graphical access is **SSH X11 forwarding**: `ssh -X user@server`.

```
$ echo $DISPLAY
:0.0
$ export DISPLAY=192.168.1.10:0.0      # send windows to another machine
$ xterm &                             # opens on 192.168.1.10
$ ssh -X admin@server "firefox"       # secure remote GUI
```

Configuring the X Server with the X Configuration Tool

Older Red Hat / Fedora systems provide a graphical program, the **X Configuration Tool** (`system-config-display`, earlier `redhat-config-xfree86`), which allows an administrator to change the display settings without editing text files. It writes the result into the X configuration file. The tool is started from the menu (*System, Administration, Display*) or from a terminal as root.

The tool has three main tabs:

Tab	What can be configured
Settings	Screen resolution (for example 1024 x 768) and colour depth (8, 16, 24 or 32 bit colours).
Hardware	Selection of the monitor type (manufacturer and model, horizontal and vertical sync ranges) and the video card type (driver).
Dual Head	Settings for using two monitors: second video card, second monitor, resolution and whether the two screens are used as one desktop or as separate desktops.

After the changes are saved, the X server must be restarted (log out and log in again, or press **Ctrl + Alt + Backspace**) for the new settings to take effect. The tool is simple, but it can only set the common options; for fine control the file must be edited manually.

Manually Configuring the X Server

The main configuration file of the X server is `/etc/X11/xorg.conf` (the older XFree86 used `/etc/X11/XF86Config`). Many modern distributions detect hardware automatically and may not even need this file, but it can still be created and edited by hand. A basic file can be generated by running `Xorg -configure`, which probes the hardware and writes `xorg.conf.new` in the current directory.

The file is divided into **sections**. Each section begins with the keyword `Section` and ends with `EndSection`.

Section	Purpose
Files	Paths of font directories and module directories used by the server.
ServerFlags	Global options of the server (for example, disable <code>Ctrl+Alt+Backspace</code>).
Module	Lists dynamically loadable modules (glx, dri, freetype and so on).

Section	Purpose
InputDevice	Defines the keyboard and the mouse: driver, layout, protocol, device file.
Monitor	Describes the monitor: identifier, horizontal sync (HorizSync) and vertical refresh (VertRefresh) ranges, and modelines.
Device	Describes the video card: identifier, driver (intel, nvidia, vesa), bus ID, video memory.
Screen	Binds a Device and a Monitor together and sets the colour depth and the list of modes (resolutions).
ServerLayout	Top level section. Combines Screen and InputDevice sections into one working setup.

```

Section "Monitor"
    Identifier      "Monitor0"
    HorizSync      31.5 - 64.0
    VertRefresh     50.0 - 75.0
EndSection

Section "Device"
    Identifier      "Card0"
    Driver          "vesa"
EndSection

Section "Screen"
    Identifier      "Screen0"
    Device          "Card0"
    Monitor         "Monitor0"
    DefaultDepth    24
    SubSection "Display"
        Depth       24
        Modes        "1024x768" "800x600"
    EndSubSection
EndSection

```

Important: Always take a backup copy of `xorg.conf` before editing (for example `cp xorg.conf xorg.conf.bak`). Wrong sync values can give a blank screen, and a wrong monitor range could even damage an old CRT monitor. If X fails to start, boot into text mode (runlevel 3) and restore the backup.

Useful commands for troubleshooting X

- `X -version` : shows the installed X server version.
- `xdpyinfo` : prints information about the display, such as resolution and extensions.
- `xrandr` : changes the resolution and rotation of the screen while X is running.
- `/var/log/Xorg.0.log` : the log file which should be read first when X does not start.

2.2 Configuring Printer

Printing in Linux is handled by a **print spooler**. When a user prints a document, the file is not sent directly to the printer; it is first stored as a **print job** in a **queue** (spool directory, normally `/var/spool/cups`) and then sent to the printer when it is free. This lets many users share one printer on the network and lets the user continue working while the printing goes on in the background.

Early UNIX used **LPD** (Line Printer Daemon, configured through `/etc/printcap`). Today most Linux systems use **CUPS** (Common UNIX Printing System), which uses the **Internet Printing Protocol (IPP)** on **TCP port 631**.

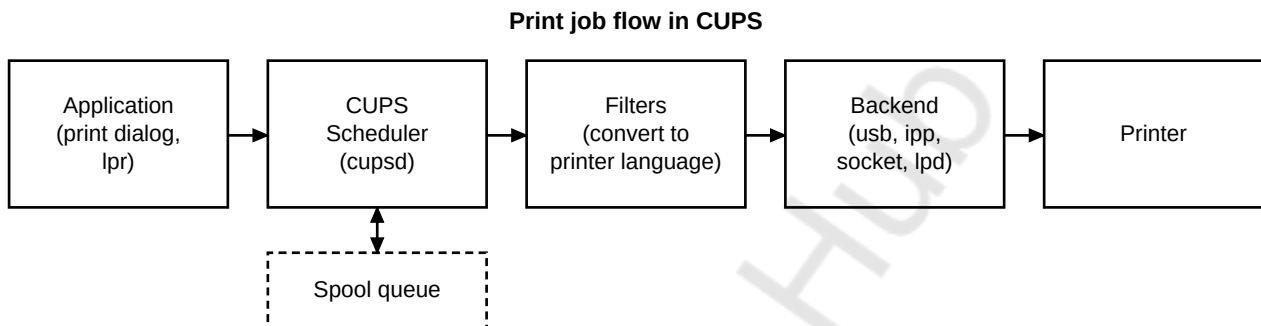


Fig 2.2: Flow of a print job through CUPS

Types of printer connection

Type	Description
Local printer	Connected directly to the computer through a USB or parallel port.
Network printer (IPP/CUPS)	Printer shared by another CUPS server. Address looks like <code>ipp://server:631/printers/name</code> .
Network printer (LPD/LPR)	Printer or server which supports the older LPD protocol (TCP port 515).
Socket / JetDirect	Printer with its own network card; raw data is sent to port 9100.
Windows (SMB) printer	Printer shared by a Windows computer or Samba server, accessed using the SMB protocol.

Configuring a printer

A printer can be added graphically with **system-config-printer** (Printer Configuration tool) or through the **CUPS web interface** by opening `http://localhost:631` in a browser. The usual steps are:

- 1 Start the CUPS service: `service cups start` (or `systemctl start cups`).
- 2 Open the Printer Configuration tool and click **New Printer**.
- 3 Select the connection type (USB port, network printer, Windows printer and so on) and give the device URI.
- 4 Select the printer **manufacturer and model**. This chooses the right driver, which is stored as a **PPD** (PostScript Printer Description) file.
- 5 Give the printer a name, a description and a location, and finish.
- 6 Print a **test page** to check that the printer works. Set it as the default printer if required.

Command line printing tools

Command	Function
lpr file / lp file	Submits a file for printing. Option -P (lpr) or -d (lp) selects the printer.
lpq	Displays the jobs waiting in the print queue.
lprm job-id / cancel job-id	Removes a job from the queue.
lpstat -p -d	Shows the status of printers and the default destination.
lpadmin	Administration command to add, modify, delete printers and set the default (for example <code>lpadmin -p laser -E -v socket://192.168.1.50 -m drv:///...</code>).
cupsenable / cupsdisable	Starts or stops a printer.
cupsaccept / cupsreject	Makes the queue accept or refuse new jobs.

Important CUPS files

File	Use
/etc/cups/cupsd.conf	Main server configuration: listening port, access control, logging, browsing.
/etc/cups/printers.conf	List of configured printers and their device URIs.
/etc/cups/ppd/	Directory containing the PPD (driver) file of each printer.
/var/log/cups/	Access, error and page logs, used to trace printing problems.
/var/spool/cups/	Spool directory holding the queued jobs.

Sharing a printer. To make a local printer available to the network, the option *Share published printers* is enabled and *Listen / Allow* lines in `cupsd.conf` are adjusted. Windows computers can reach the same printer through Samba. The firewall must allow port 631.

2.3 TCP/IP Networking

TCP/IP (Transmission Control Protocol / Internet Protocol) is a suite of protocols which is the basis of the Internet and of almost all LANs today. Every Linux system has TCP/IP built into the kernel. A network administrator must understand addressing, configuration files and the commands used to set up and test a network connection.

TCP/IP model

Layer	Function	Examples
Application	Gives network services to user programs.	HTTP, FTP, SMTP, DNS, DHCP, SSH, NFS
Transport	End-to-end delivery, error control, port numbers.	TCP (reliable), UDP (fast, connectionless)
Internet	Logical addressing and routing of packets.	IP, ICMP, ARP, IGMP
Network Access	Physical transmission of frames in the local network.	Ethernet, Wi-Fi, PPP

Understanding Network Class

An **IPv4 address** is 32 bits long and is written as four decimal numbers (octets) separated by dots, for example 192.168.10.25. Each address has two parts: the **network ID** (identifies the network) and the **host ID** (identifies the computer inside that network). In the classful addressing scheme, the first few bits of the address decide where the division falls. Based on this, addresses are divided into five **classes**.

32-bit IPv4 address (4 octets)

Class A	0 + 7 bit Network	Host (24 bits)	1 - 126
Class B	10 + Network (14 bits)	Host (16 bits)	128 - 191
Class C	110 + Network (21 bits)	Host (8)	192 - 223

Fig 2.3: Network and host parts of Class A, B and C addresses

Class	First octet range	Default mask	Networks	Hosts per network	Use
A	1 - 126	255.0.0.0 (/8)	126	16,777,214	Very large organisations
B	128 - 191	255.255.0.0 (/16)	16,384	65,534	Medium sized organisations
C	192 - 223	255.255.255.0 (/24)	2,097,152	254	Small networks, LANs
D	224 - 239	Not applicable	Not applicable	Not applicable	Multicast groups
E	240 - 255	Not applicable	Not applicable	Not applicable	Reserved, experimental

- **127.0.0.0/8** is reserved for **loopback** (127.0.0.1 = localhost), used to test the TCP/IP software of the machine itself.
- The first address of a network (host bits all 0) is the **network address** and the last (host bits all 1) is the **broadcast address**; these cannot be given to hosts. That is why the number of hosts is $2^n - 2$.
- **Private addresses** (RFC 1918) are not routed on the Internet: 10.0.0.0/8, 172.16.0.0 to 172.31.255.255 (/12) and 192.168.0.0/16.
- The **subnet mask** shows which bits belong to the network part. **Subnetting** borrows host bits to create smaller networks, and **CIDR** (Classless Inter-Domain Routing) writes the mask as a prefix length such as /26.
- **IPv6** uses 128-bit addresses written in hexadecimal (for example 2001:db8::1) and removes the shortage of IPv4 addresses.

Configuring the Network

To make a Linux computer work on a network, the administrator must set at least: the **IP address**, **subnet mask**, **default gateway**, **DNS server** and **hostname**. These can be assigned **statically** (typed in manually) or **dynamically** from a DHCP server.

Network configuration files (Red Hat style)

File	Contents
/etc/sysconfig/network-scripts/ifcfg-eth0	Settings of interface eth0: DEVICE, BOOTPROTO (static or dhcp), IPADDR, NETMASK, GATEWAY, ONBOOT.
/etc/sysconfig/network	Global settings such as HOSTNAME and default GATEWAY.
/etc/hosts	Local table mapping IP addresses to host names.
/etc/resolv.conf	Addresses of DNS servers (nameserver lines) and the search domain.
/etc/nsswitch.conf	Order in which name sources are searched (files, dns, nis, ldap).
/etc/services	Mapping between service names and port numbers.

```
# /etc/sysconfig/network-scripts/ifcfg-eth0 (static address)
DEVICE=eth0
BOOTPROTO=static
IPADDR=192.168.10.25
NETMASK=255.255.255.0
GATEWAY=192.168.10.1
ONBOOT=yes

# /etc/resolv.conf
search example.com
nameserver 192.168.10.2
```

Network commands

Command	Purpose
ifconfig / ip addr	Display or set the IP address of an interface (ifconfig eth0 192.168.10.25 netmask 255.255.255.0 up).
route / ip route	Show or change the routing table; route add default gw 192.168.10.1.
ping	Tests whether a remote host is reachable (uses ICMP echo).
traceroute	Shows the path (hops) taken by packets to a destination.
netstat / ss	Lists open connections, listening ports and routing table.
nslookup / dig / host	Queries the DNS server for name to address translation.
hostname	Shows or sets the name of the computer.
system-config-network	Graphical Network Configuration tool.
service network restart	Restarts the network service so new settings are applied.

DHCP (Dynamic Host Configuration Protocol) assigns IP address, mask, gateway and DNS automatically. In the ifcfg file the line **BOOTPROTO=dhcp** is used and the client program **dhclient** contacts the server. DHCP reduces the administrator's work and avoids duplicate addresses.

Troubleshooting order: (1) check cable and link light; (2) ping 127.0.0.1 to test the TCP/IP stack; (3) ping own IP address; (4) ping the default gateway; (5) ping an outside IP address; (6) ping a host name to test DNS. The first failure shows where the fault lies.

Exploring Directory Services

A **directory service** stores information about network resources (hosts, users, printers and so on) in a central place and lets clients look it up by name. At the TCP/IP level the most common directory services are:

Service	Description
DNS (Domain Name System)	Distributed database which converts host names such as www.example.com into IP addresses and back. Uses UDP/TCP port 53. Zones are managed by BIND (named).
NIS (Network Information Service)	Developed by Sun (earlier called Yellow Pages). Shares /etc/passwd, /etc/group and /etc/hosts maps among many UNIX computers so that one login works everywhere.
LDAP	Lightweight Directory Access Protocol. A modern, standard directory which can store users, groups, hosts, and so on (see section 2.6).
WINS / NetBIOS	Name resolution used in older Windows networks.

Remote Network Access

An administrator seldom sits in front of a server. Linux provides several tools for logging in and running commands on remote machines:

Tool	Port	Description
telnet	23	Old remote login program. Sends everything, even the password, in plain text, so it is insecure and avoided.
rlogin / rsh	513 / 514	Remote login and remote shell of the r-commands family. Trust based, insecure.
SSH (Secure Shell)	22	Encrypted remote login. Replaces telnet and rsh. Includes scp and sftp for secure file copy and supports X11 forwarding and tunnelling.
VNC	5900+	Shows the whole remote graphical desktop. Often tunnelled through SSH.
FTP / SFTP	21 / 22	File transfer. FTP is clear text; SFTP works over SSH.

```
$ ssh admin@192.168.10.5          # login
$ scp report.txt admin@192.168.10.5:/tmp/  # copy file
$ ssh -L 5901:localhost:5901 admin@server  # tunnel VNC
```

2.4 The Network File System (NFS)

NFS overview

NFS was developed by Sun Microsystems in 1984. It lets a computer **mount a directory located on a remote server** and use it as if it were part of the local file system. Users and programs open, read and write the remote files with the normal commands. NFS is used mostly between UNIX and Linux machines to share home directories, software and data from one central server.

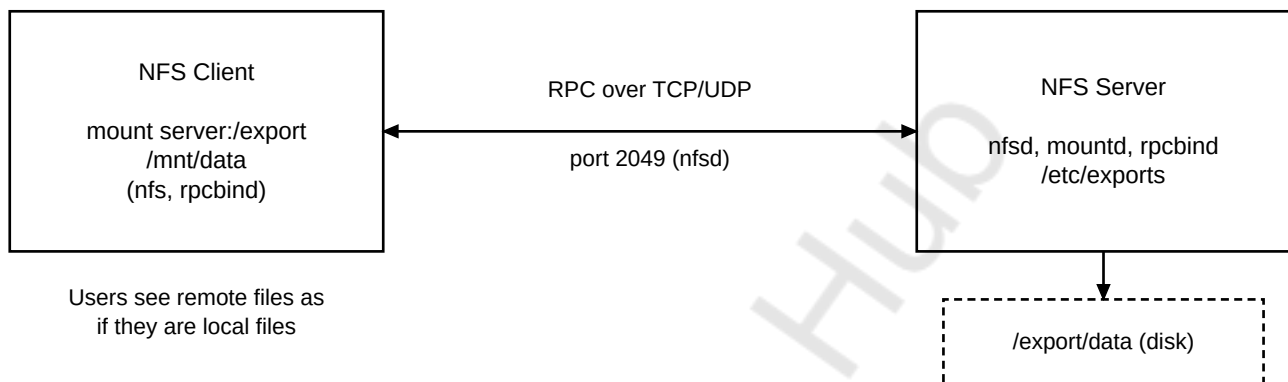


Fig 2.4: NFS client and NFS server

- NFS works on the **client-server** model. The server **exports** directories, and the client **mounts** them.
- It is built over **RPC** (Remote Procedure Call), so the **portmapper / rpcbind** service must be running. The server itself listens on port **2049**.
- Versions: **NFSv2** (1989, UDP, 32-bit file sizes), **NFSv3** (1995, large files, TCP, asynchronous writes) and **NFSv4** (2000, stateful, one port 2049, strong security with Kerberos, access control lists).
- Advantages: central storage, easy backup, reduced disk usage on clients and consistent files from every machine.
- Disadvantages: depends on the network and the server, traditional versions have weak security, and performance may fall when the network is busy.

NFS daemons

Daemon / service	Task
rpcbind (portmap)	Maps RPC program numbers to port numbers. Must run on both server and client (NFSv2/v3).
nfsd (rpc.nfsd)	Serves the actual file requests from clients. Several threads can run at the same time.
mountd (rpc.mountd)	Handles the mount requests; checks /etc/exports.
statd (rpc.statd)	Network Status Monitor, tells clients when the server reboots.
lockd (rpc.lockd)	Gives file locking over the network.

Configuring an NFS Server

- 1 **Install the packages:** nfs-utils (and rpcbind).
- 2 **Create the directory to share**, for example `mkdir /export/data`, and set its permissions.

- 3 **Edit /etc/exports.** Each line has the directory, followed by the clients allowed and the options in brackets.
- 4 **Start the services:** `service rpcbind start; service nfs start` (on systemd: `systemctl enable --now nfs-server`) and make them start at boot with `chkconfig`.
- 5 **Apply changes** with `exportfs -ra` and check with `exportfs -v`.
- 6 **Open the firewall** for NFS (port 2049 and rpcbind port 111).

```
# /etc/exports
/export/data 192.168.10.0/24(rw,sync,root_squash)
/export/docs client1.example.com(ro,sync)
/export/pub *(ro,all_squash)
```

Option	Meaning
rw / ro	Clients may read and write / read only.
sync / async	Server replies only after the data is safely on disk / replies earlier (faster but risk of data loss on crash).
root_squash	Maps the remote root user to the anonymous user nfsnobody (default). Protects the server.
no_root_squash	Remote root keeps root powers on the share. Dangerous.
all_squash	Maps every remote user to the anonymous user.
anonuid / anongid	Specifies the UID / GID used for the anonymous user.
subtree_check / no_subtree_check	Controls checking that the file is inside the exported tree.

Management commands: `exportfs -a` exports all entries, `exportfs -u` un-exports, `showmount -e server` lists the exports of a server and `rpcinfo -p` lists the registered RPC services.

Configuring an NFS Client

- 1 Install `nfs-utils` and start `rpcbind` (needed for v2/v3).
- 2 Check what the server offers: `showmount -e 192.168.10.5`.
- 3 Create a **mount point**: `mkdir /mnt/data`.
- 4 Mount the share: `mount -t nfs 192.168.10.5:/export/data /mnt/data`.
- 5 To mount automatically at boot, add a line to **/etc/fstab**.
- 6 To unmount use `umount /mnt/data`.

```
# /etc/fstab
192.168.10.5:/export/data /mnt/data nfs rw,hard,intr,rsize=8192,wsiz=8192 0 0
```

- **hard** mount: the client keeps retrying if the server stops responding (safe for data). **soft** mount: gives an error after a timeout.
- **intr** lets the user interrupt a stuck NFS operation. **rsiz** / **wsiz** set the read and write block size.
- **autofs** mounts a share only when it is accessed and unmounts it after a period of inactivity, which saves resources.

NFS Security

Plain NFS (v2 and v3) was designed for trusted networks. It trusts the user ID sent by the client machine, so the following measures are important:

Measure	Explanation
Restrict clients	Export only to specific hosts or networks in /etc/exports, never to * for sensitive data.
Squashing	Keep root_squash on. Use all_squash for public shares so that remote users get no special rights.
Read-only exports	Export with ro when users need not write.
TCP Wrappers	Use /etc/hosts.allow and /etc/hosts.deny to allow only listed hosts to access rpcbind, mountd and so on.
Firewall	Allow ports 111 and 2049 only from trusted networks. Fix the ports of mountd and statd so they can be filtered.
Mount options	Use nosuid, noexec and nodev on the client to stop clients from running dangerous files from the share.
Kerberos (NFSv4)	sec=krb5 (authentication), krb5i (adds integrity) and krb5p (adds encryption of the data) give strong security.
Matching UIDs	User IDs must be the same on server and client (use NIS or LDAP) otherwise files can be opened by the wrong user.

2.5 Network Related Jobs

As organisations depend more and more on computer networks, a number of specialised careers have been created. For an engineering student who is studying network management, it is useful to know the roles, their duties and the skills required. The titles are sometimes used in different ways by different companies, but the basic division is as follows.

Network Administrator

The **network administrator** looks after the daily working of an existing network. He or she keeps the servers, user accounts, printers, shared files and network services running smoothly, and is the first person to be called when something fails.

- Creating, modifying and deleting **user accounts**, groups and permissions.
- Installing, configuring and updating servers, operating systems and network software.
- Monitoring performance and availability of the network, and solving faults (**troubleshooting**).
- Taking regular **backups** and testing recovery; keeping a disaster-recovery plan.
- Applying **security** patches, managing antivirus and firewall rules, and checking logs.
- Maintaining documentation, IP address lists and an inventory of equipment; supporting users.

Network Engineer

The **network engineer** is more technical and works on the planning, implementation and support of network equipment: routers, switches, firewalls, wireless access points and links between sites. He or she installs and configures the devices, optimises performance and handles higher-level faults which the administrator cannot solve.

- Installing and configuring routers, switches, VLANs, routing protocols (RIP, OSPF, BGP) and WAN links.
- Capacity planning, traffic monitoring and quality of service (QoS).
- Implementing VPNs, firewalls and wireless networks; upgrading hardware and firmware.
- Working with vendors and service providers to solve problems and to commission new links.

Network Architect / Designer

The **network architect** (or network designer) is a senior role. The architect studies the business needs of the organisation and designs the whole network: topology, devices, addressing plan, security model, redundancy and future growth. The design is then handed over to engineers for implementation.

- Preparing the **network design** and diagrams for new buildings, campuses or data centres.
- Selecting technologies, vendors and equipment according to cost, performance and scalability.
- Planning **security architecture**, high availability and disaster recovery.
- Setting standards and policies for the organisation; evaluating new technologies such as cloud and SDN.

Other Network Related Jobs

Job	Main responsibility
Network Security Specialist	Protects the network from attacks: firewalls, IDS/IPS, VPN, security audits and penetration tests.
Systems Administrator	Manages servers and operating systems (Linux, Windows) which provide the network services.
Help Desk / Technical Support	First line support to users; records and solves simple problems or passes them upwards.
Network Technician	Physical work: laying cables, crimping, installing and testing equipment, hardware repair.
Wireless Network Engineer	Designs and maintains Wi-Fi and cellular networks, performs site surveys.
Database / Web Administrator	Maintains database servers or web servers and their connectivity.
Network Manager / IT Manager	Manages the team and the budget, and links IT work to organisational targets.
Cloud / DevOps Network Engineer	Builds and automates virtual networks in cloud platforms.

Comparison of the three main roles

Point	Administrator	Engineer	Architect
Focus	Day-to-day operation	Implementation and optimisation	Planning and design
Level	Entry to mid	Mid	Senior
Typical work	Users, servers, backup	Routers, switches, links	Topology, standards, strategy
Useful certifications	CompTIA Network+, RHCSA, MCSA	Cisco CCNA / CCNP, JNCIA	CCIE, CCDE, TOGAF

Skills required: knowledge of TCP/IP, routing and switching, operating systems (Linux and Windows), network security, scripting (Bash, Python), documentation, problem solving and good communication.

2.6 Directory Services

Definition of Directory Services

A **directory service** is a database and a set of services that **stores, organises and provides information about network resources** such as users, computers, groups, printers, shared folders and applications, and makes it available to authorised users and programs. It is similar to a telephone directory: if the name is known, the details can be found quickly. A directory is read much more often than it is written, so it is optimised for fast searching.

- **Single point of administration:** one account is enough for logging into many computers.
- **Authentication and authorisation:** checks who the user is and what he or she is allowed to do.
- **Location transparency:** resources are found by name, not by physical address.
- **Hierarchical, replicated and scalable:** data is arranged as a tree and copied to several servers.

Novell Directory (NDS / eDirectory)

Novell Directory Services (NDS) was introduced by Novell with NetWare 4.0 in 1993, and later renamed **eDirectory**. It was one of the first commercial, full-fledged directory services. It is hierarchical and replicated, and treats every resource in the network as an **object** in a global tree. In earlier NetWare (3.x) every server had its own separate user list (the *bindery*); NDS replaced this with a network-wide database.

- **Tree:** the whole directory. **Root:** the top of the tree.
- **Container objects** hold other objects: Country (C), Organization (O) and Organizational Unit (OU).
- **Leaf objects** are the end resources: User, Group, Printer, Server, Volume.
- The directory database is divided into **partitions** and copied to several servers as **replicas**.
- eDirectory is now available also on Linux and uses LDAP for access.

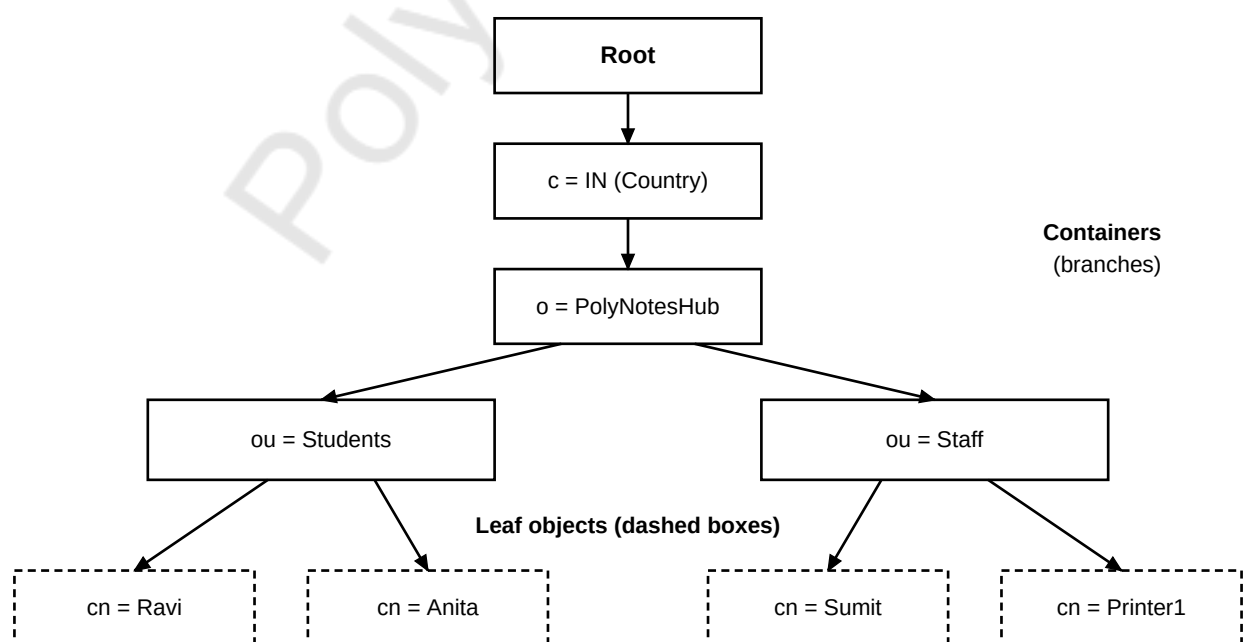


Fig 2.5: A directory tree - root, containers and leaf objects

Windows NT Domains

In **Windows NT 4.0** (1996) Microsoft used the concept of a **domain**, a logical group of computers sharing a common user account database and security policy. Account information is kept in the **SAM** (Security Accounts Manager) database on the **Primary Domain Controller (PDC)**, and copies are kept on one or more **Backup Domain Controllers (BDC)**. Computer and user names are resolved with **NetBIOS / WINS**.

- There is **one master copy** (on the PDC), so changes can be made only on the PDC and then replicated to BDCs.
- The structure is **flat**: there are no containers or OUs, so large organisations need many domains.
- Domains are connected by manually created, **one-way, non-transitive trust relationships**; the number of trusts grows very fast.
- Limited scalability (practical limit of around 40,000 objects per domain) and administration cannot be delegated easily.

Microsoft's Active Directory

Active Directory (AD) is the directory service of Microsoft, introduced with **Windows 2000 Server** and now part of Active Directory Domain Services (AD DS) in Windows Server. It replaced the NT domain model and is built on **DNS** for naming and on **LDAP** for access, with **Kerberos** for authentication.

Feature	Windows NT Domain	Active Directory
Directory structure	Flat	Hierarchical (forests, trees, domains, OUs)
Domain controllers	One PDC, many BDC	Multi-master: every DC accepts changes
Naming	NetBIOS names	DNS names, LDAP, UPN
Trust	One-way, non-transitive, manual	Two-way, transitive, automatic within a forest
Authentication	NTLM	Kerberos (NTLM for compatibility)
Scalability	About 40,000 objects	Millions of objects
Administration	Central only	Delegation through OUs and group policy

X.500 Directory Access Protocol

X.500 is a series of standards of the ITU-T (1988) which defines a global, distributed directory for the OSI model. Information is stored in the **Directory Information Base (DIB)** and arranged as a tree called the **Directory Information Tree (DIT)**. Every entry has a **Distinguished Name** built from attribute-value pairs such as country (C), organisation (O), organisational unit (OU) and common name (CN).

Clients talk to the directory through **DAP (Directory Access Protocol)**, a heavy protocol which needs the complete OSI protocol stack. Because of this complexity, DAP was never widely used on TCP/IP networks, but the X.500 *data model* (entries, attributes, DIT, DN) became the base of LDAP, Novell eDirectory and Active Directory.

Lightweight Directory Access Protocol (LDAP)

LDAP was designed at the University of Michigan as a simple "lightweight" front end to X.500, running directly over **TCP/IP**. It is now an Internet standard (RFC 4511) and the common language for directory access. The standard ports are **389** (plain or STARTTLS) and **636** (LDAPS - LDAP over SSL/TLS). A popular open source implementation is **OpenLDAP** (server daemon `slapd`).

LDAP term	Meaning
Entry	One record in the directory; a collection of attributes with a unique DN.
Attribute	A type and its value(s), for example mail: ravi@example.com.
objectClass	Defines which attributes an entry must / may have (for example person, inetOrgPerson).
DN (Distinguished Name)	Full unique path of an entry, for example cn=Ravi,ou=Students,o=PolyNotesHub,c=IN.
RDN (Relative DN)	The leftmost part of a DN which identifies the entry inside its parent (cn=Ravi).
Base DN	The starting point (top of the sub-tree) of a search.
LDIF	LDAP Data Interchange Format, a text format for exporting and importing entries.

LDAP operations: *Bind* (authenticate), *Search*, *Compare*, *Add*, *Modify*, *Delete*, *Modify DN* and *Unbind* (close). Search filters look like `(&(objectClass=person)(cn=Ra*))`.

```
dn: cn=Ravi Kumar,ou=Students,o=PolyNotesHub,c=IN
objectClass: inetOrgPerson
cn: Ravi Kumar
sn: Kumar
mail: ravi@example.com
```

Forests, Trees, Roots and Leaves

These words describe how a directory is organised. In a general sense, a directory is an upside-down tree: the **root** is at the top, **branches (containers)** spread downwards and the **leaves** are at the bottom. In Active Directory the same terms get a more particular meaning.

Term	Meaning
Root	The topmost point of a directory tree. In AD, the first domain created in a tree or forest is the root domain.
Leaf	An object which cannot contain other objects, for example a user, computer or printer.
Container / branch	An object which can hold other objects: a domain or an organisational unit (OU).
Tree	A group of domains in a contiguous DNS namespace , joined by automatic two-way transitive trusts (example: polynoteshub.co.in and north.polynoteshub.co.in).
Forest	A collection of one or more trees which do not share a contiguous namespace but share a common schema, configuration and global catalog. It is the outer security boundary of AD.

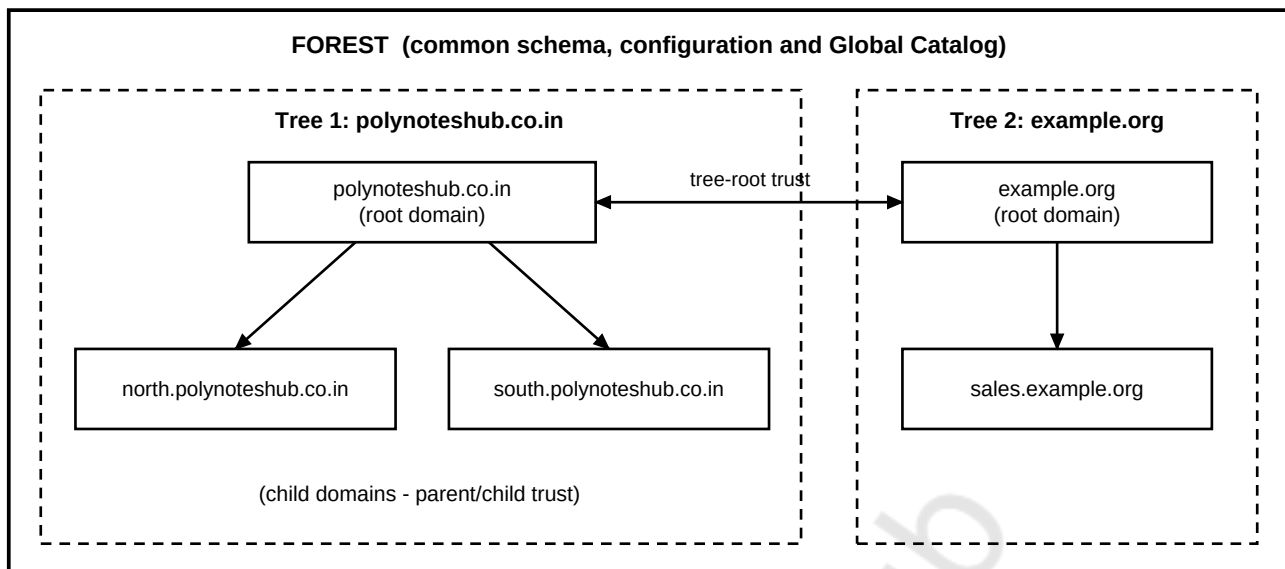


Fig 2.6: A forest containing two trees

Configuring Samba Server

Samba is a free software suite which implements the Microsoft **SMB** / **CIFS** (Server Message Block / Common Internet File System) protocol on Linux and UNIX. With Samba, a Linux server can share files and printers with Windows clients, act as a file server, a print server, a domain member or even a domain controller. Samba is made of two main daemons:

Daemon	Function
smbd	Provides file and print sharing and user authentication (TCP ports 139 and 445).
nmbd	Provides NetBIOS name service and browsing (UDP ports 137, 138).
winbindd	Connects Linux with Windows domains / Active Directory (user and group mapping).

Steps to configure a basic Samba file server:

- 1 Install the packages: `samba, samba-client, samba-common`.
- 2 Create the directory to share and set permissions: `mkdir /srv/share; chmod 777 /srv/share`.
- 3 Edit the main file `/etc/samba/smb.conf` (set workgroup, security mode and the share).
- 4 Check the file for mistakes with `testparm`.
- 5 Create a Samba password for each user, who must already exist as a Linux user: `smbpasswd -a ravi`.
- 6 Start and enable the services: `service smb start; service nmb start`. Open the firewall and set the SELinux boolean if needed.
- 7 Test from Linux with `smbclient -L //server -U ravi`, and from Windows with `\\server\share`.

```
# /etc/samba/smb.conf
[global]
    workgroup = WORKGROUP
    server string = Samba Server
    security = user
    map to guest = Bad User

[homes]
    comment = Home Directories
    browseable = no
    writable = yes

[public]
    comment = Public Share
    path = /srv/share
    writable = yes
    guest ok = yes
```

- **[global]** section: settings for the whole server. **[homes]**: automatically shares each user's home directory. **[printers]**: shares the CUPS printers. Other sections create ordinary shares.
- Security modes: **user** (each user needs a Samba account), **ads** (member of an Active Directory domain), **domain** (NT domain member).
- A Linux client can mount a Windows or Samba share with `mount -t cifs //server/share /mnt -o username=ravi`.

2.7 Active Directory Architecture

Active Directory stores its data in a database file (NTDS.dit) on every **domain controller (DC)**. The database is divided into logical **partitions**: the *schema*, the *configuration* and the *domain* partitions. The architecture decides how objects are described, named, found and secured.

Object Types

Everything stored in AD is an **object**: a set of attributes that describes a resource. The **schema** defines the object *classes* (types) and *attributes* which can exist in the directory.

Object type	Description
User	Person who logs on; has attributes like name, password, e-mail, group membership.
Computer	A workstation or server which has joined the domain and has its own account.
Group	Collection of users, computers or other groups used to give permissions easily (security groups and distribution groups).
Contact	External person with an e-mail address; cannot log in.
Organizational Unit (OU)	A container used to organise objects, delegate administration and apply Group Policy.
Printer	Published printer which users can search and connect to.
Shared Folder	Published network share.
Domain	Container for all objects; a security and replication boundary.

Objects are of two kinds: **container objects** (domain, OU) which can hold other objects, and **leaf objects** (user, computer, printer) which cannot.

Object Naming

Every object has a unique name by which it can be found. AD supports several ways of writing an object name, which are explained below with one example: user *Ravi Kumar* in OU *Students* of the domain *polynoteshub.co.in*.

Name type	Example	Explanation
Distinguished Name (DN)	CN=Ravi Kumar,OU=Students,DC=polynoteshub,DC=co,DC=in	Full path of the object in the directory; unique in the forest.
Relative Distinguished Name (RDN)	CN=Ravi Kumar	Part of the DN which identifies the object inside its container.
Canonical Name	polynoteshub.co.in/Students/Ravi Kumar	DN written in a readable top-down form, starting from the domain.
User Principal Name (UPN)	ravi@polynoteshub.co.in	E-mail style logon name; short and unique in the forest.
SAM Account Name	POLY\ravi	Older style logon name (domain\user) for earlier Windows versions.
GUID	6F9619FF-8B86-D011-B42D-00C04FC964FF	128-bit number which never changes during the life of the object.

Canonical names

The **canonical name** shows the object like a file path. The domain name is first, then the OUs separated by "/", and finally the object name. It is easier for administrators to read than the DN and is shown in the *Active Directory Users and Computers* console.

LDAP notation

Because AD is accessed using LDAP, objects are identified by their LDAP **Distinguished Name**. The DN is written **from the object up to the root** (right to left in hierarchy) using these attribute abbreviations:

Abbreviation	Meaning	Used for
DC	Domain Component	Each part of the DNS name of the domain (polynoteshub, co, in).
OU	Organizational Unit	Container objects below the domain.
CN	Common Name	Leaf objects and built-in containers (Users, Computers).
O / C	Organization / Country	Used in X.500 style names.

Globally Unique Identifiers (GUID)

A **GUID** is a 128-bit number which is assigned to every object when it is created (stored in the objectGUID attribute). Unlike the DN, which changes if the object is renamed or moved, the GUID **never changes**, so programs use it as a permanent reference. Users and groups also have a **SID** (Security Identifier) used for access control, which is a different number from the GUID.

User Principal Names (UPN)

The **UPN** has the form `username@domain`, similar to an e-mail address. It is the logon name for users in modern Windows. The UPN is unique in the forest and does not change if the user account is moved to another domain or OU, which makes the logon easy for the user.

Domain, Trees and Forests

- **Domain:** the basic unit of AD. All objects of a domain share one database partition, one security policy and one DNS name. A domain is a replication and security boundary and has at least one domain controller.
- **Tree:** one or more domains in a continuous DNS namespace (child domains are named by adding a prefix to the parent: `north.polynoteshub.co.in`). Parent and child domains have automatic **two-way transitive trust**.
- **Forest:** one or more trees which share the same schema, configuration and **Global Catalog**. The first domain created is the **forest root domain**.
- **Global Catalog (GC):** a server holding a partial copy of all objects of the forest so that searches can be done across domains quickly, and also used for UPN logons.
- **Multi-master replication:** any domain controller can accept changes and copies them to the others. Some special tasks are held by only one DC (FSMO roles: schema master, domain naming master, PDC emulator, RID master, infrastructure master).
- **Sites:** a physical grouping of IP subnets that are connected by fast links, used to control replication and logon traffic.

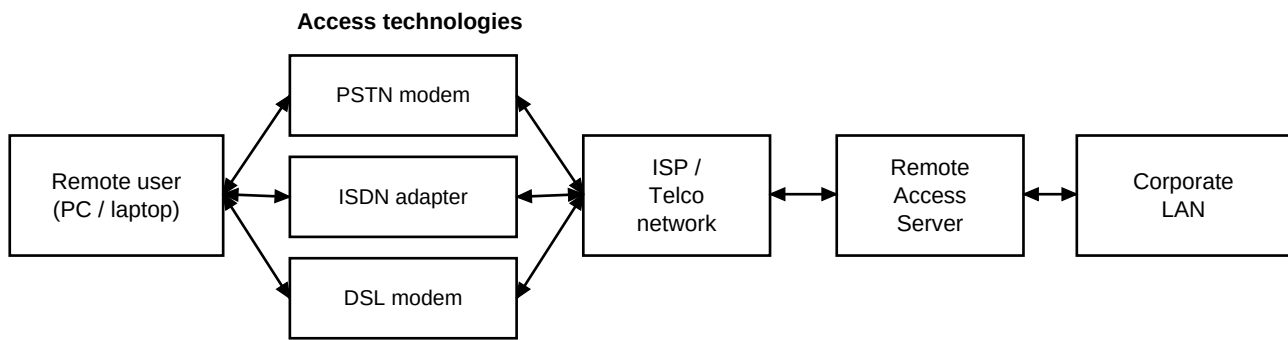
Level	Boundary / role	Example
Forest	Security boundary; shared schema and global catalog.	PolyNotesHub forest
Tree	Continuous DNS namespace.	polynoteshub.co.in
Domain	Administrative, policy and replication boundary.	north.polynoteshub.co.in
OU	Delegation of control and Group Policy.	OU=Students
Object	The resource: user, computer, printer.	CN=Ravi Kumar

2.8 Remote Network Access

Need of Remote Network Access

Remote network access means connecting to a network from a distant place using a communication link, so that the user can use the resources of the network as if he or she were sitting in the office. It is needed because:

- Employees work from **home, travel or branch offices** and still need e-mail, files and applications.
- Administrators must manage servers and fix faults **without going to the site**, which saves time and cost.
- Organisations with several branches need a way to link the **head office and branches**.
- Customers and partners may need limited access to some company services.
- It supports business continuity, for example work during a natural disaster or pandemic.
- But remote access also opens the network to outside attacks, so **authentication, encryption and access control** are essential.



CATV (cable modem) is another access path to the ISP

Fig 2.7: Remote access to a corporate network

The main connection technologies are described below.

Public Switched Telephone Network (PSTN)

The **PSTN** is the ordinary analog telephone network. A computer is connected through a **modem** (modulator - demodulator), which converts digital data into audio tones that can travel on the voice line, and a call is made to the remote access server. It is also called **dial-up** access.

- Maximum speed is **56 kbps** (V.90 / V.92 standards); the actual rate is usually less because of line noise.
- It is a **circuit-switched** connection: the line is occupied during the call, so the telephone cannot be used for voice at the same time.
- Protocol used is **PPP** (Point-to-Point Protocol) (older: SLIP).
- Advantages: available almost anywhere and very cheap. Disadvantages: very slow, long connection time and unsuitable for modern applications.

Integrated Services Digital Network (ISDN)

ISDN is a **digital** telephone service which carries voice, data and video over the same ordinary copper wires, from end to end. The line is divided into channels: **B channels** (Bearer, 64 kbps each) carry the user data and a **D channel** (Delta) carries signalling and control information.

Interface	Channels	Speed	Use
BRI (Basic Rate Interface)	2B + D (D = 16 kbps)	144 kbps (128 kbps user data)	Homes and small offices
PRI (Primary Rate Interface)	23B + D (T1, USA / Japan); 30B + D (E1, India / Europe)	1.544 Mbps / 2.048 Mbps	Large offices, PBX connection

- Faster call set-up (about one second) and better quality than PSTN; two channels can be joined to give 128 kbps.
- Needs special equipment, an ISDN terminal adapter (TA) and a network termination (NT1). Mostly replaced by DSL and broadband.

Digital Subscriber Line (DSL)

DSL gives high-speed digital data on the **same copper telephone wire**, by using the high frequencies above the voice band (above 4 kHz) which the telephone does not use. A **splitter** separates voice and data so that the telephone and the Internet can be used at the same time. At the exchange a **DSLAM** (DSL Access Multiplexer) collects the lines of many subscribers and connects them to the ISP.

Type	Meaning	Speed (approx.)	Remark
ADSL	Asymmetric DSL	Down 8 Mbps / Up 1 Mbps (ADSL2+: 24 Mbps down)	Most common home service; download faster than upload
SDSL	Symmetric DSL	Same speed both directions, 1.5 to 2 Mbps	Business use; no voice on the line
HDSL	High bit-rate DSL	1.544 / 2.048 Mbps	Replacement of T1 / E1 lines
VDSL / VDSL2	Very high bit-rate DSL	52 Mbps down (VDSL2 up to 100 Mbps)	Works over short distance only

- Always-on connection (no dialling) with a dedicated line to the exchange.
- Speed falls with the **distance** from the exchange (service is normally limited to about 5 km).

CATV (Cable TV)

The **cable television** network, which uses **coaxial cable** and optical fibre, can also be used for data. A **cable modem** in the house connects the computer to the cable. The network is called a **Hybrid Fibre-Coaxial (HFC)** network; the data standard is **DOCSIS** (Data Over Cable Service Interface Specification). At the operator's end a **CMTS** (Cable Modem Termination System) connects the cable network to the Internet.

- TV channels use some frequency ranges, while other ranges are kept for downstream and upstream data.
- The cable is a **shared medium** among the neighbourhood, so speed may fall when many users are active, and security needs more attention.
- Speeds range from a few Mbps (early) to hundreds of Mbps (DOCSIS 3.0) and gigabit (DOCSIS 3.1).

Comparison of remote access technologies

Point	PSTN	ISDN	DSL	CATV
Signal	Analog	Digital	Digital (over copper)	Digital (over coax / fibre)
Max. speed	56 kbps	128 kbps (BRI)	Several Mbps to 100 Mbps	Tens to hundreds of Mbps
Connection	Dial-up	Dial-up (fast)	Always on	Always on
Medium	Telephone line	Telephone line	Telephone line	Cable TV line
Equipment	Modem	ISDN adapter	DSL modem, splitter	Cable modem
Sharing	Dedicated	Dedicated	Dedicated line	Shared with neighbours
Cost	Very low	Medium	Low to medium	Low to medium

2.9 Virtual Private Network (VPN)

A **Virtual Private Network (VPN)** creates a **secure, private connection over a public network** such as the Internet. It is called *virtual* because no dedicated physical line is used, and *private* because the data is encrypted and protected so that outsiders cannot read it. VPN is a cheap alternative to leased lines and to dial-up servers.

Two basic techniques are used by every VPN:

- **Tunnelling:** the original packet is wrapped (encapsulated) inside another packet and sent across the Internet. The path between the two ends is called a **tunnel**.
- **Encryption and authentication:** the data is encrypted for confidentiality, checked for **integrity**, and both ends prove their identity (password, certificate, token).

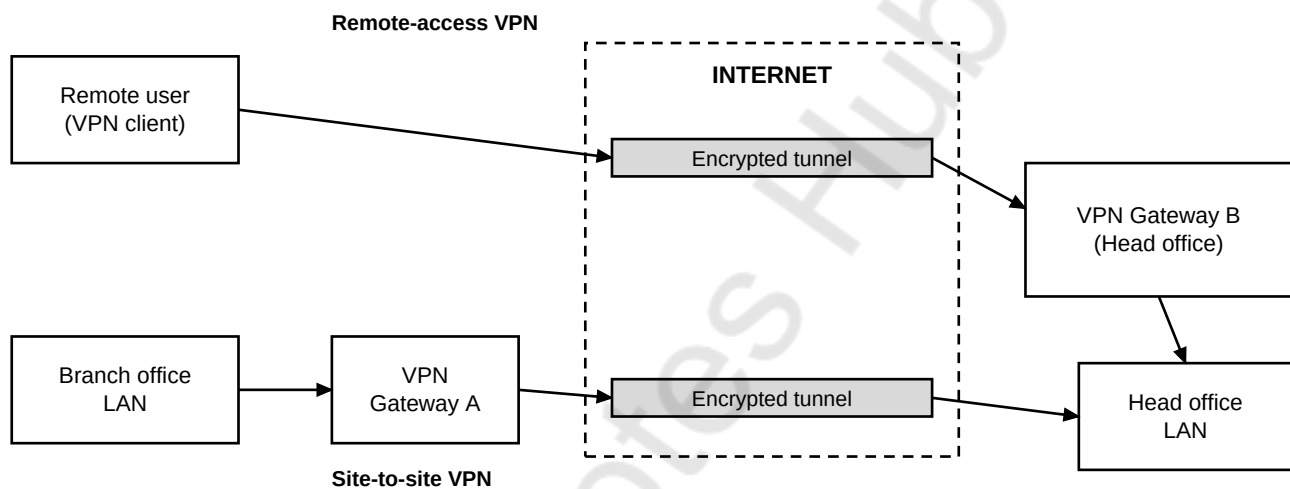


Fig 2.8: Remote-access VPN and site-to-site VPN through the Internet

VPN Protocols

Protocol	Layer / Ports	Features
PPTP (Point-to-Point Tunneling Protocol)	Layer 2; TCP 1723 + GRE (protocol 47)	Developed by Microsoft; easy to set up and fast; uses MPPE encryption. Now considered weak and not recommended.
L2F / L2TP (Layer 2 Tunneling Protocol)	Layer 2; UDP 1701	Combination of PPTP and L2F. Gives tunnelling only, so it is normally used together with IPsec (L2TP/IPsec) for encryption.
IPsec (IP Security)	Layer 3; IKE UDP 500 / 4500; ESP (protocol 50), AH (51)	Suite of protocols for authentication (AH), encryption (ESP) and key exchange (IKE). Works in transport mode (only the payload is protected) or tunnel mode (the whole packet is protected).
SSTP (Secure Socket Tunneling Protocol)	Application; TCP 443	Microsoft; sends PPP traffic over SSL/TLS. Passes most firewalls.
OpenVPN	TCP / UDP 1194 (configurable)	Open source; uses SSL/TLS; very flexible and secure; runs on many systems.
IKEv2/IPsec	UDP 500 / 4500	Modern, fast and good at reconnecting when the mobile user changes network.
WireGuard	UDP (default 51820)	Modern, simple code, high speed, strong modern cryptography.

Types of VPNs

Type	Description
Remote-access VPN (client to site)	A single user connects to the company network through VPN software. Used by travelling staff and people working from home.
Site-to-site VPN (gateway to gateway)	Two networks, for example the head office and a branch, are joined permanently through VPN gateways (routers or firewalls). Users do not need any software.
Intranet VPN	Site-to-site VPN that connects the different offices of the same organisation.
Extranet VPN	Site-to-site VPN between the company and business partners, suppliers or customers with limited access.
Host-to-host VPN	Secure tunnel directly between two computers.

From the ownership point of view, VPNs are also grouped as **provider-provisioned** (for example MPLS VPN built by a service provider) and **customer-provisioned** (the organisation builds it using the Internet).

VPN Clients

A **VPN client** is the software (or built-in feature) on the user's device which sets up the tunnel to the **VPN server / concentrator**. It authenticates the user, negotiates the encryption and sends the traffic through the tunnel; a **virtual network adapter** is created and the computer gets an internal IP address of the company network.

- **Built-in clients:** Windows, macOS, Android and iOS have inbuilt support for L2TP/IPsec, IKEv2 and others; Linux uses NetworkManager, strongSwan, OpenVPN.
- **Vendor clients:** Cisco AnyConnect, FortiClient, GlobalProtect, OpenVPN Connect, and so on.
- **Split tunnelling:** only company traffic goes through the VPN and other Internet traffic goes directly. Gives better speed but lower security. **Full tunnelling:** all the traffic passes through the VPN.
- Authentication methods: user name and password, digital certificates, smart cards and **multi-factor authentication** (OTP).

SSL VPNs

An **SSL VPN** (more precisely a TLS VPN) uses the **SSL/TLS** protocol, the same one used by HTTPS websites, to give remote access. Because it works over **TCP port 443**, it passes through almost all firewalls and needs no special client software in many cases; a web browser is sufficient.

SSL VPN mode	How it works
Clientless (web portal)	User logs in to a web page using a browser; the gateway gives access to web applications, file shares and e-mail. No software installation.
Thin client (port forwarding)	A small Java or ActiveX applet is downloaded and forwards the traffic of certain TCP applications (for example e-mail or remote desktop) through the gateway.
Full tunnel mode	A complete client (often downloaded from the portal) creates a virtual adapter and gives full network-layer access similar to IPsec.

IPsec VPN versus SSL VPN

Point	IPsec VPN	SSL VPN
Layer	Network layer (Layer 3)	Application / session layer (SSL/TLS)
Client software	Required on each device	Often only a web browser
Access given	Full network access	Can be restricted to selected applications
Firewall / NAT passing	May need special settings (ESP, UDP 500/4500)	Easy, uses port 443
Best suited for	Site-to-site links and managed devices	Remote users, home and public computers
Configuration	More complex	Simpler for the user

Advantages and disadvantages of VPN

Advantages	Disadvantages
Lower cost than leased lines or dial-up servers	Performance depends on the Internet speed and on the encryption load
Strong security through encryption and authentication	Misconfiguration may create security holes
Easy to extend to new users and sites	Different vendors may have compatibility problems
Allows access to internal resources from anywhere	A compromised remote device can bring threats into the network

Chapter Review Questions

Part A: Multiple Choice Questions (15)

Choose the correct answer for each question. The answer key is given at the end of this part.

1. Which file is the main configuration file of the modern X server?
 - (a) /etc/X11/xorg.conf
 - (b) /etc/xorg.sh
 - (c) /etc/sysconfig/display
 - (d) /etc/startx.conf
2. In the X Window System, the X server is the program that
 - (a) runs on the remote application host
 - (b) controls the display, keyboard and mouse of the user's machine
 - (c) draws window borders and title bars
 - (d) stores user passwords
3. The CUPS printing system uses which protocol and TCP port?
 - (a) LPD, port 515
 - (b) SMB, port 445
 - (c) IPP, port 631
 - (d) SNMP, port 161
4. Which command displays the jobs waiting in the print queue?
 - (a) lpq
 - (b) lpadmin
 - (c) lpstat -d
 - (d) cupsenable
5. The default subnet mask of a Class B network is
 - (a) 255.0.0.0
 - (b) 255.255.0.0
 - (c) 255.255.255.0
 - (d) 255.255.255.128
6. Which of the following is a private IP address?
 - (a) 172.32.5.4
 - (b) 192.169.1.1
 - (c) 172.20.10.5
 - (d) 11.0.0.1
7. Which file contains the addresses of the DNS servers on a Linux system?
 - (a) /etc/hosts
 - (b) /etc/resolv.conf
 - (c) /etc/nsswitch.conf
 - (d) /etc/services

8. In /etc/exports, the option that maps the remote root user to an anonymous user is
- (a) no_root_squash
 - (b) all_access
 - (c) root_squash
 - (d) async
9. Which command lists the directories exported by an NFS server?
- (a) mount -t nfs
 - (b) showmount -e
 - (c) ifconfig -a
 - (d) testparm
10. Which job mainly plans and designs the overall structure of a network for an organisation?
- (a) Help desk technician
 - (b) Network administrator
 - (c) Network architect
 - (d) Network technician
11. LDAP uses which standard TCP port for ordinary (non-SSL) connections?
- (a) 389
 - (b) 636
 - (c) 53
 - (d) 443
12. The distinguished name CN=Ravi,OU=Students,DC=poly,DC=in is an example of
- (a) UPN
 - (b) LDAP notation
 - (c) GUID
 - (d) NetBIOS name
13. An Active Directory forest is best described as
- (a) a single organisational unit
 - (b) a group of trees sharing a common schema and global catalog
 - (c) a printer shared on a network
 - (d) a backup domain controller
14. How many channels does an ISDN Basic Rate Interface (BRI) have, and what is its total speed?
- (a) 2B + D, 144 kbps
 - (b) 23B + D, 1.544 Mbps
 - (c) 30B + D, 2.048 Mbps
 - (d) 1B + D, 80 kbps
15. Which VPN type is best suited for a web browser based access that works over port 443?
- (a) PPTP
 - (b) L2TP
 - (c) SSL VPN
 - (d) GRE tunnel

Answer Key

1 - (a)	2 - (b)	3 - (c)	4 - (a)	5 - (b)
6 - (c)	7 - (b)	8 - (c)	9 - (b)	10 - (c)
11 - (a)	12 - (b)	13 - (b)	14 - (a)	15 - (c)

Part B: Broad Questions (10)

Answer the following questions in detail, with neat diagrams and examples wherever possible.

Q1. Describe the client-server architecture of the X Window System. Explain how the X server can be configured using the X Configuration Tool and manually through `xorg.conf`.

Q2. Explain the steps involved in configuring a printer in Linux using CUPS. List the important commands and configuration files.

Q3. What is an IP address class? Explain Class A, B, C, D and E addresses with their ranges, default masks and the number of networks and hosts. Mention the private address ranges.

Q4. Explain the steps for configuring the network on a Linux machine (static IP and DHCP). Name the important configuration files and commands.

Q5. What is NFS? Explain how an NFS server and an NFS client are configured. Discuss the measures used for NFS security.

Q6. Describe the different network related jobs (network administrator, network engineer, network architect and others) with their duties and required skills.

Q7. Define directory service. Compare Novell Directory, Windows NT domains and Microsoft Active Directory. Write a short note on X.500 and LDAP.

Q8. Explain the architecture of Active Directory: object types, object naming (DN, canonical name, UPN, GUID), domains, trees and forests.

Q9. Why is remote network access needed? Compare PSTN, ISDN, DSL and CATV technologies in detail.

Q10. What is a VPN? Explain the VPN protocols, the types of VPN, VPN clients and SSL VPNs. Compare SSL VPN and IPsec VPN.