

NETWORK MANAGEMENT AND ADMINISTRATION

Chapter 4

Implementation of Network

Detailed Study Notes for Engineering Students

- 4.1 Designing Network
- 4.2 Configuring a Database Server
- 4.3 Creating VNC Server
- 4.4 Providing Additional Network Services
- 4.5 Optimizing Network Services

Includes diagrams, tables, 15 MCQs with answer key and 10 broad questions

Contents

Chapter Overview	3
4.1 Designing Network	4
4.1.1 Assessing Network Needs	4
4.1.2 Applications	5
4.1.3 Users	5
4.1.4 Network Services	5
4.1.5 Security and Safety	6
4.1.6 Growth and Capacity Planning	7
4.1.7 Meeting Network Needs	7
4.2 Configuring a Database Server	12
4.2.1 Planning Before Installation	12
4.2.2 Installation and Initial Setup	12
4.2.3 Creating Databases, Users and Privileges	13
4.2.4 Allowing Remote Access	13
4.2.5 Backup and Restore	14
4.2.6 Security and Performance	14
4.3 Creating VNC Server	15
4.3.1 Key Points About VNC	15
4.3.2 Steps to Create a VNC Server (TigerVNC on Ubuntu with XFCE)	15
4.3.3 Security Measures for VNC	17
4.3.4 Troubleshooting	17
4.4 Providing Additional Network Services	18
4.4.1 Configuring a Time Server	18
4.4.2 Providing a Caching Proxy Server	20
4.5 Optimizing Network Services	23
4.5.1 Performance Measures	23
4.5.2 Monitoring and Finding Bottlenecks	23
4.5.3 Optimization Techniques	24
4.5.4 Typical Problems and Remedies	25
Chapter Summary (Quick Revision)	26
Multiple Choice Questions	27
Answer Key for MCQs	29
Broad Questions	30

Chapter Overview

Implementation of a network is the stage at which planning turns into a working system. A network that is merely connected is not necessarily a good network: it must be designed around real requirements, equipped with the right servers and services, configured securely and then tuned so that it keeps performing as demand grows. This chapter follows that natural order. It begins with the design of a network, moves on to building common services (a database server, a remote desktop (VNC) server, a time server and a caching proxy server) and ends with the techniques used to optimize services once they are running.

After studying this chapter, a student should be able to:

- assess the needs of users and applications and translate them into a network design;
- choose a suitable network type, structure (topology) and servers for a given requirement;
- install and configure a database server and manage its databases and users;
- create and secure a VNC server for remote graphical access;
- configure a time server (NTP) and a caching proxy server (Squid);
- identify bottlenecks and apply optimization techniques to network services.

Note on examples

Commands in this chapter are shown for a Linux server (Ubuntu/Debian style, with notes for RHEL-family systems) because Linux is the most common platform for teaching network administration. The concepts are identical on other operating systems, but package names, file paths and service names can differ between distributions and versions, so always confirm them in the documentation of the system you use. IP addresses such as 192.168.1.10 are sample private addresses used only for illustration.

4.1 Designing Network

Network design is the process of planning the structure, devices, media, addressing scheme, services and security of a network so that it satisfies the present needs of an organization and can be expanded in future. A design made without studying requirements leads to expensive rework: slow links, overloaded servers, unsafe access and a network that cannot grow. Good design therefore starts with **questions**, not with equipment.

The overall design life cycle is shown in Fig. 4.1. Note that it is a cycle: monitoring a running network produces new information that feeds back into the next round of assessment.

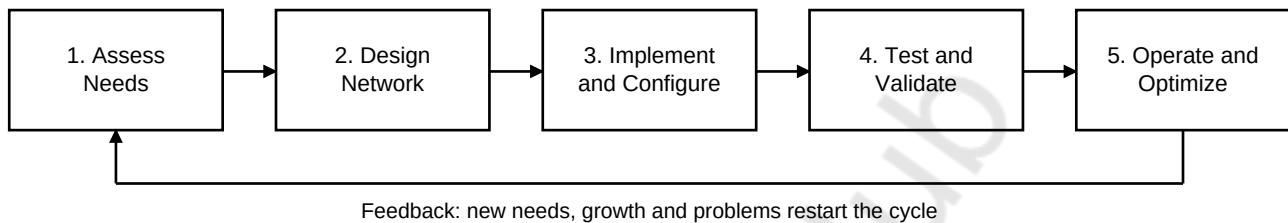


Fig. 4.1: Network design and implementation life cycle

4.1.1 Assessing Network Needs

Assessing (analyzing) network needs means collecting and documenting what the network must do. The designer talks to managers, teachers or staff, surveys the building and studies existing equipment. The result is a written requirements document that is later used to judge whether the finished network is successful.

Main activities in needs assessment:

- **Identify stakeholders** and interview them about their work and expectations.
- **Take an inventory** of existing computers, printers, switches, cabling and software.
- **Survey the site:** building layout, floors, distances, cable routes, power and cooling.
- **List applications and services** that run and how critical each one is.
- **Estimate users and traffic:** number of users, locations, peak hours.
- **Define constraints:** budget, time, skills available, legal and safety rules.
- **Set measurable goals:** for example 99.9 percent availability or response time below 2 seconds.

Category	Questions to ask	Example answer
Users	How many users? Where are they? Who needs remote access?	120 students and 15 staff in 2 buildings
Applications	Which programs and data must be shared?	Library software, e-mail, lab software
Performance	What speed and response time are acceptable?	1 Gbps to desktops, 10 Gbps backbone
Reliability	How much downtime is tolerable?	Less than 1 hour per month
Security	What data is confidential? Who may access it?	Examination data limited to staff
Budget	What are the capital and running costs allowed?	Fixed yearly budget, spare funds limited

4.1.2 Applications

Applications decide how much bandwidth the network needs and how sensitive it is to delay. Two networks with the same number of users may need very different designs if one mostly sends e-mail and the other carries video conferencing. Applications are therefore classified by their bandwidth demand, their sensitivity to **latency** (delay) and **jitter** (variation in delay), and their importance to the business.

Application	Bandwidth need	Delay sensitivity	Remarks
E-mail	Low	Low	Store-and-forward; delay of minutes is acceptable
Web browsing	Medium	Medium	Benefits from caching (Section 4.4)
File transfer / backup	Very high	Low	Schedule outside working hours
VoIP (voice calls)	Low	Very high	One-way delay should stay below about 150 ms
Video conferencing	High	Very high	Needs QoS and stable links
Database / ERP	Medium	High	Many small queries; latency matters
Remote desktop (VNC/RDP)	Low to medium	High	Screen updates must feel responsive

Applications are also divided into **mission-critical** (must always be available, such as the examination or billing system) and **non-critical** (such as general browsing). Critical applications receive redundant links and servers and higher priority in traffic management.

4.1.3 Users

Users are the reason a network exists, so their number, location and behavior directly shape the design. The designer must know how many users are present now, how many will be added, how many use the network at the same time (**concurrency**) and what level of technical skill they have. A network for a few trained staff is administered differently from a campus network with thousands of students.

User group	Typical needs	Access level
Administrators	Full control of servers and devices, remote management	Highest; strictly limited number
Staff / faculty	File sharing, e-mail, printing, internal software	Read/write to departmental data
Students	Internet, lab systems, learning portal	Restricted; no access to staff data
Guests	Internet only	Separate guest network or VLAN
Remote users	Secure access from home or other branches	Through VPN with authentication

Each user should have a unique account, and permissions should follow the **principle of least privilege**: a user receives only the access required for his or her work, nothing more.

4.1.4 Network Services

A network service is a function provided by a server or device to other computers on the network. During design, the required services are listed first, because each service needs a server (or virtual machine), software, a port through which clients reach it and a place in the security plan.

Service	Purpose	Protocol / Port
DHCP	Automatically assigns IP addresses to clients	UDP 67, 68
DNS	Translates names to IP addresses	UDP/TCP 53
Web (HTTP/HTTPS)	Hosts websites and portals	TCP 80, 443
E-mail	SMTP sends mail; POP3/IMAP retrieve mail	TCP 25, 110, 143
File transfer (FTP / SFTP)	Uploads and downloads files	TCP 21 / 22
File and print sharing	Shared folders and printers (SMB)	TCP 445
Directory (LDAP)	Central user accounts and authentication	TCP 389
Remote login (SSH)	Secure command-line administration	TCP 22
Database	Stores and serves structured data (MySQL)	TCP 3306
Time (NTP)	Keeps clocks synchronized	UDP 123
Proxy (Squid)	Caches and filters web access	TCP 3128
Remote desktop (VNC)	Graphical remote access	TCP 5900 + display no.

4.1.5 Security and Safety

Security protects data and services from misuse. It is built around the **CIA triad: Confidentiality** (only authorized persons read data), **Integrity** (data is not changed illegally) and **Availability** (services are usable when needed). **Safety** protects people and equipment from physical danger and loss, such as fire, electric shock, overheating and power failure. Both must be planned during design; adding them later is costly and incomplete.

Threat or risk	Countermeasure
Unauthorized access	Strong passwords, multi-factor authentication, least privilege, account audit
Attacks from the Internet	Firewall at the boundary, intrusion detection/prevention, closing unused ports
Eavesdropping on data	Encryption: HTTPS, SSH, VPN, WPA2/WPA3 for Wi-Fi
Malware and unpatched software	Antivirus, regular updates and patch management
Internal misuse	Network segmentation (VLANs), logging and monitoring
Data loss	Regular backups stored off-site, tested restore procedures, RAID
Power failure	UPS and generator; surge protectors
Fire, heat, humidity	Fire suppression, air conditioning, temperature monitoring in server room
Physical theft or tampering	Locked server room and racks, access log, CCTV
Disaster	Written disaster recovery plan and backup site

A written **security policy** that states acceptable use, password rules, backup schedule and incident response is part of the design and must be known to all users.

4.1.6 Growth and Capacity Planning

Networks almost never stay the same size. New users, new applications and heavier files increase traffic every year. **Capacity planning** is the practice of estimating future demand and making sure that bandwidth, server resources, storage, IP addresses and physical ports will be sufficient when the demand arrives. A common guideline is to design for a growth period of three to five years and to keep normal utilization well below the limit (many administrators aim for about 60 to 70 percent at peak) so that there is headroom for bursts.

Steps of capacity planning:

- 1 **Baseline:** measure current utilization of links, CPU, memory, disk and users.
- 2 **Forecast:** estimate growth rate from history and from business plans.
- 3 **Identify bottlenecks:** find the resource that will run out first.
- 4 **Plan upgrades:** decide what to add, when and at what cost.
- 5 **Review regularly:** compare forecast with reality and correct the plan.

Worked example (assumed figures)

A department has 200 users and expects 15 percent growth every year. Users after 3 years = $200 \times (1.15)^3 = 200 \times 1.5209 =$ about 304. If each user generates about 2 Mbps at peak, the required peak bandwidth is $304 \times 2 = 608$ Mbps. A 1 Gbps backbone would then run at about 61 percent, which is acceptable, whereas a 100 Mbps backbone would be hopelessly overloaded.

Design practices that support growth:

- leave spare ports in switches and spare conduits and capacity in cable trays;
- prefer modular or stackable switches and rack space for new servers;
- plan the IP address scheme with unused subnets (for example, allocate a /22 where a /24 is needed today);
- use server virtualization so that new servers need no new hardware immediately;
- choose cabling and backbone links (fibre, Cat6/Cat6a) that support higher speeds later.

4.1.7 Meeting Network Needs

When the requirements are known, the designer selects the network type, the structure and the servers that best satisfy them. These three choices are described below.

A. Choosing Network Type

Networks are first classified by the area they cover. The designer decides which type (or combination) is needed.

Type	Full form	Coverage	Example
PAN	Personal Area Network	A few metres	Bluetooth devices around one person
LAN	Local Area Network	Room, building or campus	College laboratory network
MAN	Metropolitan Area Network	City	Cable TV or municipal fibre network
WAN	Wide Area Network	Country or world	The Internet, bank branch network

Next, the designer chooses the networking model (how computers share resources) and the medium.

Feature	Peer-to-Peer	Client-Server
Control	No central control; every computer is both client and server	Central servers control resources and users
Suitable size	Very small (about 10 computers or fewer)	Medium to large networks
Security	Weak; each user sets own sharing	Strong; central accounts and permissions
Backup / management	Difficult, done on each machine	Centralized and easy
Cost	Low, no dedicated server	Higher; server and administrator needed
Scalability	Poor	Good

Feature	Wired (Ethernet)	Wireless (Wi-Fi)
Speed and stability	High and consistent	Shared medium; varies with distance and interference
Mobility	Fixed locations only	Free movement inside coverage area
Security	Physical access required	Needs strong encryption (WPA2/WPA3)
Installation	Cabling work required	Quick, fewer cables
Typical use	Servers, desktops, backbone	Laptops, phones, guest access

In practice, most organizations use a **client-server wired LAN** for servers and desktops with a **wireless LAN** added for mobility, and connect to other sites through a WAN or VPN.

B. Choosing Network Structure

Network structure has two parts: the **physical topology** (how devices are connected) and the **logical design** (how the network is organized into layers, subnets and VLANs). Fig. 4.2 shows three common basic topologies.

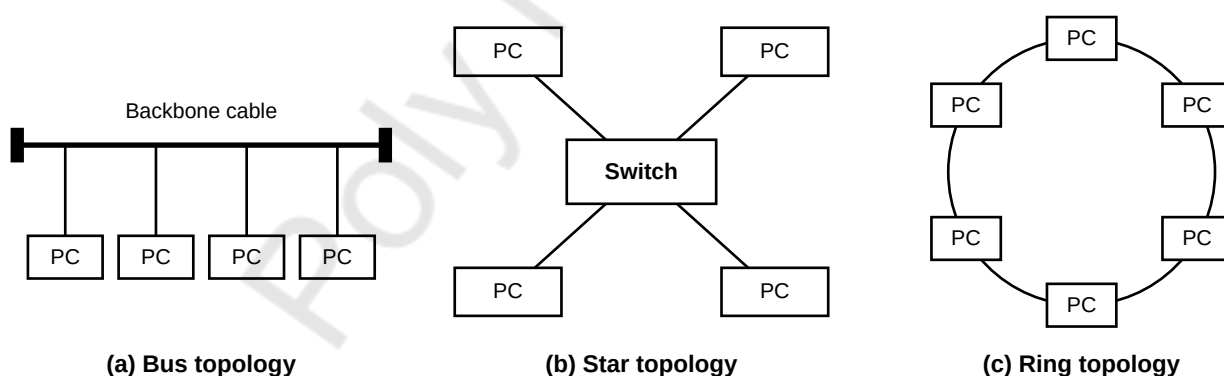


Fig. 4.2: Bus, star and ring topologies

Topology	Description	Advantages	Disadvantages
Bus	All devices share one backbone cable	Cheap, little cable	Cable fault stops everything; poor performance with load
Star	Every device connects to a central switch	Easy to add devices; one fault affects only one device	Central switch is a single point of failure

Topology	Description	Advantages	Disadvantages
Ring	Each device connects to two neighbours in a loop	Orderly access to medium	One break can disturb the ring; hard to extend
Mesh	Devices have multiple links to others	Very reliable; many alternative paths	Expensive, complex cabling
Tree / hybrid	Stars joined to a backbone in a hierarchy	Scalable, matches buildings and floors	Backbone failure splits the tree

Modern Ethernet LANs use a **star** (or extended star) topology based on switches, because it is easy to manage and a failure remains local. Mesh links are added between important switches and routers to provide redundancy.

Hierarchical three-layer model

For a campus network, a hierarchical design divides the network into three layers (Fig. 4.3). Each layer has a clear job, which makes the network easier to scale, troubleshoot and secure.

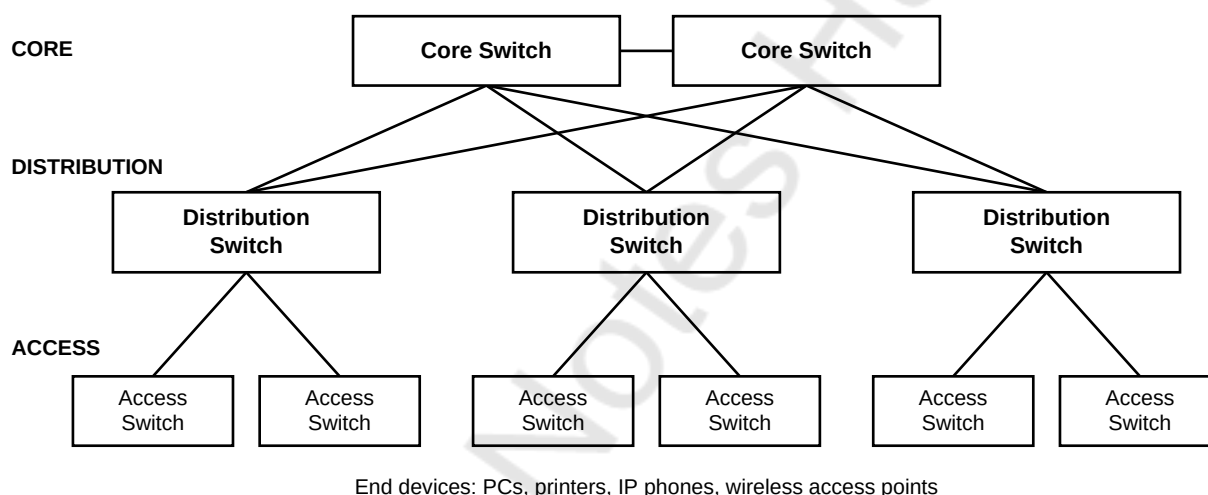


Fig. 4.3: Hierarchical three-layer network model

- **Core layer:** the high-speed backbone that moves traffic quickly between distribution blocks. It should be fast and redundant, with no complex filtering.
- **Distribution layer:** connects access switches, applies policies (routing between VLANs, access lists, QoS) and separates departments or buildings.
- **Access layer:** connects end devices (computers, printers, access points) to the network and provides port-level security.

Small networks may use a **collapsed core** (two-tier) design in which core and distribution functions are combined in the same switches. **VLANs** (Virtual LANs) and subnets further divide the network logically, for example separate VLANs for staff, students, servers and guests, to reduce broadcast traffic and improve security.

Choice of cabling

Medium	Typical speed / distance	Typical use
UTP Cat5e	1 Gbps up to 100 m	Basic desktop connections
UTP Cat6	1 Gbps up to 100 m; 10 Gbps up to about 55 m	Desktop and floor cabling
UTP Cat6a	10 Gbps up to 100 m	High-speed floor and server connections
Optical fibre	10 Gbps and above over long distances (hundreds of metres to kilometres)	Backbone between buildings and floors; immune to electrical noise

C. Choosing Servers

A server is a computer that provides services to other computers. The designer decides which servers are needed, whether each service gets its own server or several services share one, and what hardware and operating system to use.

Server role	Function
File server	Central storage and sharing of files with permissions
Print server	Manages shared printers and print queues
Web server	Hosts websites (Apache, Nginx, IIS)
Mail server	Sends, receives and stores e-mail
Database server	Runs the DBMS for applications (Section 4.2)
DNS / DHCP server	Provides names and automatic IP configuration
Authentication server	Central logins using LDAP / Active Directory / RADIUS
Proxy server	Caches and filters web traffic (Section 4.4)
Backup server	Stores copies of important data

Hardware selection factors

Component	Consideration
Processor (CPU)	Number of cores and speed according to the number of simultaneous requests; database and virtualization servers need more cores
Memory (RAM)	Enough to avoid swapping; database and cache servers benefit greatly from extra RAM; ECC memory improves reliability
Storage	Capacity, speed (SSD for databases) and protection using RAID
Network card	Gigabit or faster; two or more NICs for redundancy or separate networks
Power supply	Redundant (dual) power supplies and connection to UPS
Form factor	Tower (small office), rack-mounted (data room), blade (high density)
Support	Vendor warranty, spare parts, availability of drivers
Operating system	Linux (low cost, stable, flexible) or Windows Server (strong Active Directory integration)

RAID levels (Redundant Array of Independent Disks)

Level	Minimum disks	Method	Fault tolerance
RAID 0	2	Striping only; fastest, no copy of data	None; one disk failure loses all data
RAID 1	2	Mirroring; same data on both disks	Survives one disk failure
RAID 5	3	Striping with distributed parity	Survives one disk failure
RAID 6	4	Striping with two parity blocks	Survives two disk failures
RAID 10	4	Mirrored pairs that are striped	Survives one failure in each mirrored pair

RAID protects against disk failure but is **not a backup**; deletion, corruption or fire affects all disks, so regular backups are still necessary.

Dedicated or shared servers?

- **Dedicated server:** one main service per server. Better performance, security isolation and easier troubleshooting, but higher cost.
- **Shared (multi-role) server:** several light services on one server. Cheaper, but one fault or attack can affect all services.
- **Virtualization:** several virtual servers on one physical host (KVM, VMware, Hyper-V). Gives dedicated-style isolation with better hardware use and easy migration, but the host becomes a critical point that needs redundancy.

4.2 Configuring a Database Server

A **database** is an organized collection of data and a **Database Management System (DBMS)** is the software that creates, stores, retrieves and protects that data. A **database server** runs the DBMS on a networked computer so that many clients (applications, web sites, users) can use the same data at the same time using the client-server model (Fig. 4.4). Relational databases store data in tables and are accessed with **SQL** (Structured Query Language).

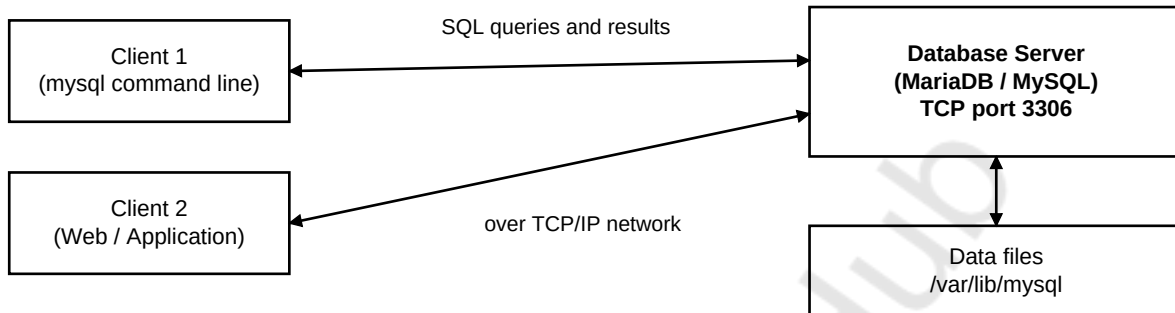


Fig. 4.4: Client-server database architecture

DBMS	Type	Default port	Remarks
MySQL	Relational, open source (Oracle)	3306	Very popular with web applications
MariaDB	Relational, fork of MySQL	3306	Default MySQL-compatible server in many Linux distributions
PostgreSQL	Object-relational, open source	5432	Rich features, strong standards support
Microsoft SQL Server	Relational, commercial	1433	Common in Windows environments
Oracle Database	Relational, commercial	1521	Large enterprise systems

In this section MariaDB/MySQL on Ubuntu or Debian is used as the working example.

4.2.1 Planning Before Installation

- Decide the **purpose and size** of the data to estimate CPU, RAM and disk needs.
- Use **SSD storage with RAID** for data files and plan a separate disk for backups.
- Give the server a **static IP address** and a DNS name.
- Decide which clients may connect and from which addresses.
- Plan a **backup schedule** and a security policy before storing real data.

4.2.2 Installation and Initial Setup

Step 1: Update the package list and install the server.

```
sudo apt update
sudo apt install mariadb-server
# RHEL / Fedora family: sudo dnf install mariadb-server
```

Step 2: Start the service and enable it at boot.

```
sudo systemctl enable --now mariadb
sudo systemctl status mariadb
```

Step 3: Secure the installation.

Run the security script. It guides the administrator to set or confirm the root authentication, remove anonymous users, disallow remote root login, remove the sample *test* database and reload the privilege tables.

```
sudo mysql_secure_installation
```

4.2.3 Creating Databases, Users and Privileges

Administration is done with SQL statements entered in the client program. Log in as the administrator and create a database, a user and the permissions for that user.

```
sudo mysql

CREATE DATABASE college_db;
CREATE USER 'student'@'192.168.1.%' IDENTIFIED BY 'Str0ng#Pass123';
GRANT SELECT, INSERT, UPDATE, DELETE ON college_db.* TO 'student'@'192.168.1.%';
FLUSH PRIVILEGES;
SHOW DATABASES;
```

The part after the @ sign is the **host** from which the user may connect. 192.168.1.% means any address in the 192.168.1.0 network. Using a specific host or network instead of % (any host) is a basic security measure. Privileges should be limited to the necessary database and operations (least privilege).

Creating a table and storing data:

```
USE college_db;
CREATE TABLE students (
    roll_no INT PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    branch VARCHAR(20)
);
INSERT INTO students VALUES (1, 'Rahul', 'CSE');
SELECT * FROM students;
```

4.2.4 Allowing Remote Access

By default MariaDB on Debian/Ubuntu listens only on the local machine (127.0.0.1). To serve other computers, change the **bind-address** and open the port in the firewall for the trusted network only.

- 1 Edit the configuration file, for example `/etc/mysql/mariadb.conf.d/50-server.cnf`, and set `bind-address = 192.168.1.10` (the server IP) or 0.0.0.0 for all interfaces.
- 2 Restart the service: `sudo systemctl restart mariadb`.
- 3 Allow only the LAN to reach port 3306 and test from a client.

```
sudo ufw allow from 192.168.1.0/24 to any port 3306 proto tcp
mysql -h 192.168.1.10 -u student -p college_db # run on the client
```

Item	Debian / Ubuntu	RHEL / Fedora family
Main configuration	/etc/mysql/ (mariadb.conf.d/)	/etc/my.cnf and /etc/my.cnf.d/
Data directory	/var/lib/mysql	/var/lib/mysql
Log files	/var/log/mysql/ and the system journal	/var/log/mariadb/
Service name	mariadb (or mysql)	mariadb

4.2.5 Backup and Restore

A database server is incomplete without a tested backup plan. A **logical backup** exports the database as SQL statements, which can later be replayed to rebuild it.

```
mysqldump -u root -p college_db > college_db.sql      # backup
mysql -u root -p college_db < college_db.sql          # restore
```

Backups can be automated with **cron** and copied to another machine. Binary logs allow point-in-time recovery for important systems, and a regular restore test proves that the backup is usable.

4.2.6 Security and Performance

- Never use the database root account for applications; create one account per application.
- Use strong passwords and, where possible, encrypted connections (TLS) for remote clients.
- Disable remote root login and remove anonymous users and the test database.
- Restrict port 3306 with a firewall and keep the DBMS updated.
- Enable the slow query log to find poorly written queries.
- Tune key settings such as `innodb_buffer_pool_size` (memory used to cache data and indexes) and `max_connections` according to the available RAM and load.
- Create **indexes** on columns used often in searches and joins.

Troubleshooting guide

Problem	Likely cause	Action
Cannot connect from a client	Service stopped, bind-address is 127.0.0.1 or firewall blocks 3306	Check service status, bind-address and firewall rule
Access denied for user	Wrong password or user not allowed from that host	Check CREATE USER host part and GRANT statements
Server will not start	Configuration error or disk full	Read the log file and journalctl output
Queries are slow	Missing indexes or too little memory	Use slow query log and EXPLAIN; tune buffer sizes

PostgreSQL in brief

For PostgreSQL the main steps are similar: install the package (`sudo apt install postgresql`), open the administrator shell with `sudo -u postgres psql`, create roles and databases, set `listen_addresses` in `postgresql.conf` and allow client networks in the `pg_hba.conf` file. The default port is 5432.

4.3 Creating VNC Server

VNC (Virtual Network Computing) is a graphical desktop sharing system that lets a user see and control the desktop of a remote computer over a network. The remote machine runs a **VNC server**; the user runs a **VNC viewer** (client). The viewer sends keyboard and mouse events to the server, and the server sends back the changes in the screen image. VNC uses the **Remote Framebuffer (RFB)** protocol and is platform independent, so a Windows viewer can control a Linux desktop and the reverse.

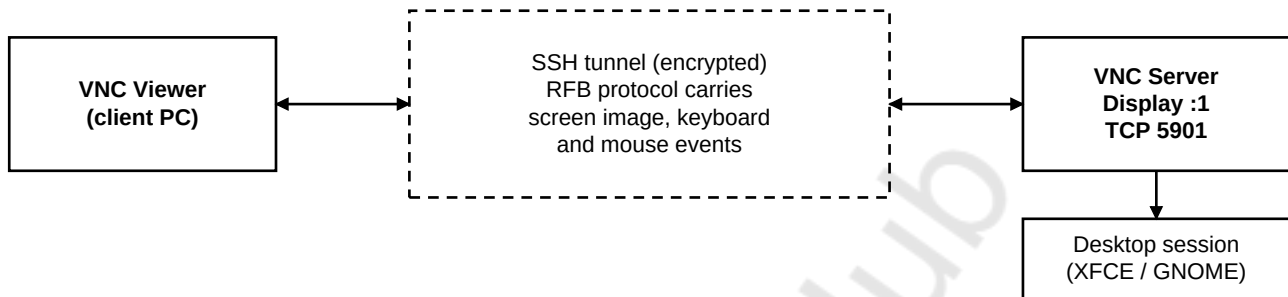


Fig. 4.5: Working of VNC with an SSH tunnel

4.3.1 Key Points About VNC

- VNC server for display number N listens on TCP port **5900 + N**: display :1 uses port 5901, display :2 uses 5902.
- A **virtual desktop** server (such as TigerVNC *vncserver*) creates a new desktop session independent of the physical screen. A **shared desktop** server (such as *x11vnc*) mirrors the screen that is already displayed.
- The classic VNC password is not strong (only the first 8 characters are used) and the traffic is not encrypted by default, so VNC should always be protected with an SSH tunnel or VPN.
- Popular implementations: TigerVNC, TightVNC, RealVNC, x11vnc.

Feature	VNC	RDP (Windows)	SSH
Mode	Graphical desktop	Graphical desktop	Text command line (X forwarding possible)
Protocol	RFB	Remote Desktop Protocol	SSH
Default port	5900 + N	TCP 3389	TCP 22
Platform	Any	Mostly Windows server	Any
Encryption	Not by default; use tunnel	Supported	Always encrypted
Bandwidth use	Medium to high	Medium (optimized)	Very low

4.3.2 Steps to Create a VNC Server (TigerVNC on Ubuntu with XFCE)

Step 1: Install a light desktop environment and the VNC server.

```
sudo apt update
sudo apt install xfce4 xfce4-goodies tigervnc-standalone-server tigervnc-common
```

Step 2: Set the VNC password (run as the normal user who will use the desktop).

```
vncpasswd
```

The password is stored in `~/.vnc/passwd`. The command may also ask for an optional view-only password.

Step 3: Tell the server which desktop to start.

Create or edit `~/.vnc/xstartup` and make it executable (`chmod +x ~/.vnc/xstartup`):

```
#!/bin/sh
unset SESSION_MANAGER
unset DBUS_SESSION_BUS_ADDRESS
exec startxfce4
```

Step 4: Start the server on display :1.

```
vncserver :1 -geometry 1280x800 -depth 24 -localhost yes
vncserver -list # show running servers
ss -tlnp | grep 5901 # confirm the port is listening
vncserver -kill :1 # stop the server
```

The option `-geometry` sets the screen size, `-depth` sets colour depth and `-localhost yes` makes the server accept connections only from the machine itself, which forces users to connect through a secure tunnel.

Step 5: Connect securely from the client.

```
ssh -L 5901:localhost:5901 student@192.168.1.10 # open tunnel (client PC)
vncviewer localhost:5901 # in a second window
```

On Windows the same tunnel can be created in PuTTY (Connection, SSH, Tunnels: source port 5901, destination localhost:5901) and any VNC viewer then connects to `localhost:5901`.

Step 6: Start the VNC server automatically at boot (optional).

Create a systemd template unit such as `/etc/systemd/system/vncserver@.service` (the user name must be replaced). On RHEL-family systems the packaged service reads the user list from `/etc/tigervnc/vncserver.users` (for example a line `:1=student`), so follow the documentation of the installed version.

```
[Unit]
Description=TigerVNC server for display %i
After=network.target

[Service]
Type=simple
User=student
ExecStart=/usr/bin/vncserver %i -fg -geometry 1280x800 -depth 24 -localhost yes
ExecStop=/usr/bin/vncserver -kill %i

[Install]
WantedBy=multi-user.target
```

```
sudo systemctl daemon-reload
sudo systemctl enable --now vncserver@:1.service
```

4.3.3 Security Measures for VNC

- Always tunnel VNC through **SSH** (or use a VPN); never expose ports 5900 and above to the Internet.
- Bind the server to localhost (*-localhost yes*) and block VNC ports in the firewall.
- Use a dedicated low-privilege user for VNC sessions, not root.
- Use strong SSH authentication (key-based login) because it protects the whole VNC session.
- Lock the session when idle and close sessions that are no longer needed.
- Keep the VNC software updated.

4.3.4 Troubleshooting

Symptom	Possible cause	Remedy
Connection refused	Server not running, wrong display/port, or tunnel not established	Run <code>vncserver -list</code> , check ss output and the SSH tunnel
Grey screen with cross cursor	xstartup missing or desktop not installed	Fix <code>~/.vnc/xstartup</code> and install the desktop packages
Black screen	Desktop crashed or session bus problem	Read the log in <code>~/.vnc/</code> and restart the server
Authentication failed	Wrong VNC password	Run <code>vncpasswd</code> again
Slow display	Low bandwidth or high colour depth	Reduce resolution and depth; use compression options

4.4 Providing Additional Network Services

Beyond the basic services, a network usually offers supporting services that make it more accurate, faster and easier to control. Two important examples are a **time server**, which keeps all clocks consistent, and a **caching proxy server**, which speeds up and controls web access.

4.4.1 Configuring a Time Server

Accurate and consistent time is essential in a network. Log files from different machines can be compared only if their clocks agree; authentication systems such as Kerberos reject requests when clocks differ by more than a few minutes; digital certificates have validity periods; scheduled jobs and database transactions depend on time; and forensic investigations need trustworthy timestamps.

Network Time Protocol (NTP)

NTP is the protocol used to synchronize clocks over a network. It works on **UDP port 123** and can keep computers within a few milliseconds of each other on a LAN. NTP arranges time sources in levels called **strata** (Fig. 4.6). Stratum 0 devices are reference clocks such as atomic clocks or GPS receivers; a server directly connected to them is stratum 1; each further hop adds one to the stratum number. A college network usually synchronizes one internal server with public servers and lets all other devices use that internal server.

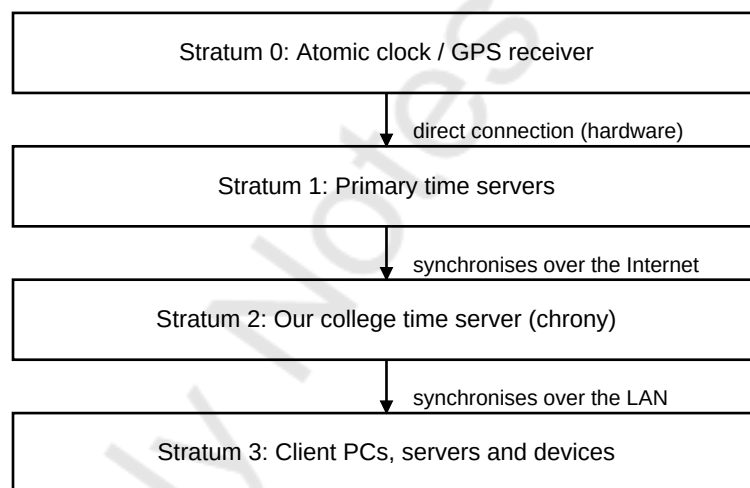


Fig. 4.6: NTP stratum hierarchy

Term	Meaning
Stratum	Distance (in hops) from the reference clock; lower is closer to the source
Offset	Difference between the local clock and the reference time
Delay	Round-trip time of the NTP request to the server
Jitter	Variation in repeated offset measurements
Drift	Natural speed error of the local hardware clock; recorded in the drift file
Poll interval	How often the client queries the server

Software	Role	Remarks
chrony	NTP client and server	Modern default on many Linux systems; fast sync, good for virtual machines

Software	Role	Remarks
ntpd	NTP client and server	The reference implementation; long established
systemd-timesyncd	Simple SNTP client only	Cannot act as a time server for others
OpenNTPD	NTP client and server	Simple, security-focused implementation

Steps to configure a time server using chrony

Step 1: Install chrony (this normally replaces timesyncd) and set the time zone.

```
sudo apt install chrony
sudo timedatectl set-timezone Asia/Kolkata
```

Step 2: Edit the configuration file (/etc/chrony/chrony.conf on Debian/Ubuntu, /etc/chrony.conf on RHEL family).

```
pool 2.pool.ntp.org iburst          # upstream public servers
driftfile /var/lib/chrony/chrony.drift
makestep 1.0 3                     # step the clock if off by more than 1 s
rtcsync                            # keep hardware clock in sync
allow 192.168.1.0/24               # let LAN clients use this server
local stratum 10                   # serve time even if Internet is down
```

Step 3: Start the service and allow NTP through the firewall.

```
sudo systemctl enable --now chronyd # service is chronyd on RHEL family
sudo ufw allow from 192.168.1.0/24 to any port 123 proto udp
```

Step 4: Verify operation.

```
chronyc sources -v          # list upstream servers and their state
chronyc tracking             # show current offset and stratum
chronyc clients              # list LAN clients served (run as root)
```

Configuring clients

- **Linux client:** in chrony.conf replace the pool line with server 192.168.1.10 iburst and restart chrony.
- **Windows client:** in an administrator command prompt use w32tm /config /manualpeerlist:192.168.1.10 /syncfromflags:manual /update and then w32tm /resync.
- **Network devices** (switches, routers, access points) can be pointed to the same internal time server through their NTP settings.

chrony directive	Purpose
pool / server	Defines upstream time sources; <i>iburst</i> speeds up the first synchronization
allow	Permits clients from the given network to query the server
local stratum	Allows the server to act as a time source when it has lost its upstream servers
makestep	Allows a quick step instead of slow adjustment when the offset is large at start-up
driftfile	File where the computed clock drift is stored

chrony directive	Purpose
rtcsync	Copies system time to the hardware clock regularly

Security of the time service: allow only the internal network, use several upstream servers so one wrong source cannot mislead the clock, and consider NTS (Network Time Security) where supported.

4.4.2 Providing a Caching Proxy Server

A **proxy server** sits between clients and the Internet and makes requests on their behalf. A **caching proxy** also stores copies of the web objects it has fetched (pages, images, files). When another user requests the same object, the proxy can deliver it from its own storage instead of downloading it again, as shown in Fig. 4.7.

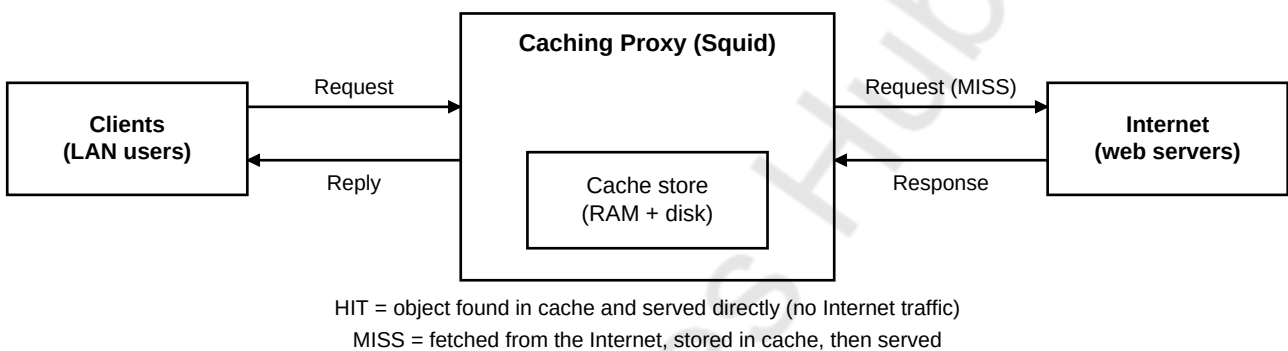


Fig. 4.7: Working of a caching proxy server

Advantages of a caching proxy

- **Saves bandwidth** because repeated downloads come from the local cache.
- **Faster response** for popular content, since the LAN is faster than the Internet link.
- **Access control** using rules: block sites, allow only certain users or time periods.
- **Logging** of web usage for monitoring and policy enforcement.
- **Privacy and security**: internal clients are hidden behind the proxy address and content can be filtered.

How the cache behaves

When a request arrives the proxy checks whether a fresh copy exists. If yes it is a **cache hit** and the object is returned immediately. If not (or if the copy has expired) it is a **cache miss**: the proxy fetches the object, stores it according to the rules of the cache and sends it to the client. For expired objects the proxy can ask the web server whether the object has changed, so that unchanged objects need not be downloaded again.

Steps to configure Squid, a widely used caching proxy

Step 1: Install Squid.

```
sudo apt install squid           # RHEL family: sudo dnf install squid
```

Step 2: Edit the main configuration file `/etc/squid/squid.conf`. A simplified working example is shown below. Keep a backup copy of the original file first.

```

http_port 3128
visible_hostname proxy.college.local

acl localnet src 192.168.1.0/24
acl blocked_sites dstdomain .facebook.com .example-ads.com

http_access deny blocked_sites
http_access allow localnet
http_access deny all

cache_mem 256 MB
maximum_object_size 50 MB
cache_dir ufs /var/spool/squid 10000 16 256
access_log /var/log/squid/access.log squid

```

Directive	Meaning
http_port	Port on which Squid accepts client requests (default 3128)
acl	Defines a named condition, for example a source network or destination domain
http_access	Allows or denies requests; rules are checked from top to bottom and the first match wins, so the final rule is normally deny all
cache_mem	Amount of RAM used for hot objects
cache_dir	Disk cache: type, directory, size in MB, then first- and second-level directory counts (10000 MB with 16 and 256 here)
maximum_object_size	Largest object that will be stored on disk
access_log	Location and format of the log of all requests

Step 3: Check the file, create the cache directories and start the service.

```

sudo squid -k parse           # check configuration syntax
sudo squid -z                 # create cache directories (if needed)
sudo systemctl enable --now squid
sudo ufw allow from 192.168.1.0/24 to any port 3128 proto tcp

```

Step 4: Configure clients and test.

In the browser (or system) proxy settings enter the proxy IP address 192.168.1.10 and port 3128. A command-line test and log monitoring can confirm that the proxy works:

```

curl -x http://192.168.1.10:3128 -I http://example.com
sudo tail -f /var/log/squid/access.log

```

Log result code	Meaning
TCP_MISS	Object was not in cache; fetched from the origin server
TCP_HIT	Object served from the disk cache
TCP_MEM_HIT	Object served from the memory cache
TCP_REFRESH_UNMODIFIED	Cached copy was checked and found still valid
TCP_DENIED	Request was rejected by an access rule

Transparent (intercepting) proxy

In a transparent proxy the clients need no browser configuration: the gateway redirects web traffic to the proxy. In Squid this uses the `intercept` option on the port together with a firewall redirection rule, for example `iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j REDIRECT --to-port 3128`. Interception works easily only for plain HTTP because HTTPS traffic is encrypted end-to-end and is normally just tunnelled through the proxy.

Limitations and related ideas

- Most websites now use HTTPS, which reduces the amount of cacheable content; inspecting it requires special certificates and has privacy and legal implications.
- Dynamic or personalized pages and very large streaming files are not suitable for caching.
- A **forward proxy** serves internal clients going out to the Internet (Squid); a **reverse proxy** (Nginx, Varnish) sits in front of web servers to cache and balance load for outside users.

4.5 Optimizing Network Services

Optimization means improving the performance, reliability and efficiency of the network and its services without unnecessary expense. Even a well-designed network slows down as usage grows, hardware ages or configuration changes. Optimization is therefore not a one-time job but a continuous cycle of measurement, analysis, correction and verification (Fig. 4.8).

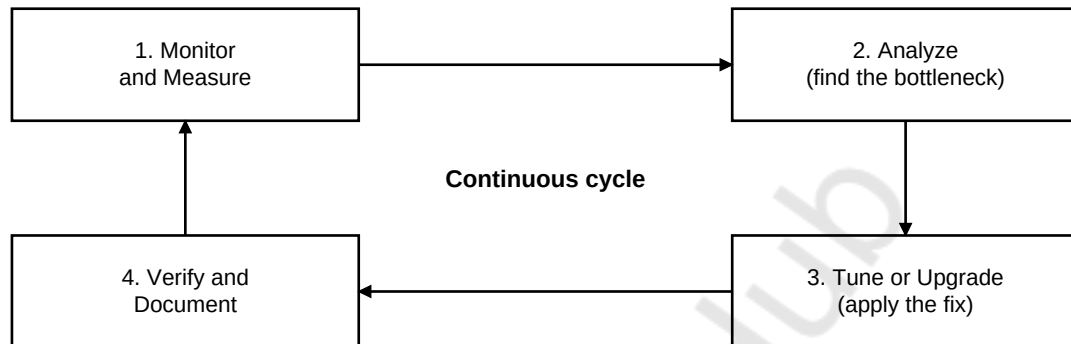


Fig. 4.8: Network optimization cycle

4.5.1 Performance Measures

Measure	Meaning	Notes
Bandwidth	Maximum data rate of a link (for example 1 Gbps)	Capacity, not actual speed
Throughput	Actual data rate achieved	Always lower than bandwidth because of overheads and load
Latency	Time taken by data to travel from source to destination	Measured in milliseconds; ping shows round-trip time
Jitter	Variation in latency	Harms voice and video
Packet loss	Percentage of packets that do not arrive	Causes retransmission and slow transfers
Utilization	Percentage of capacity in use	Sustained high values show congestion
Availability	Percentage of time a service is usable	For example 99.9 percent allows about 8.8 hours of downtime per year

4.5.2 Monitoring and Finding Bottlenecks

Optimization starts with measurement. A **baseline** records normal behavior so that abnormal behavior can be recognized. A **bottleneck** is the slowest component that limits the whole system (an overloaded link, a busy CPU, a full disk or a misconfigured service). Improving anything other than the bottleneck gives little benefit.

Tool	Use
ping	Tests reachability and measures latency and packet loss
tracert / traceroute	Shows the path and the delay at each hop

Tool	Use
iperf3	Measures achievable throughput between two hosts (iperf3 -s on server, iperf3 -c server-ip on client)
ss / netstat	Lists open ports, listening services and connections
tcpdump / Wireshark	Captures and analyzes packets for detailed troubleshooting
iftop / nload	Shows live bandwidth use per connection or interface
top / htop / vmstat / iostat	Show CPU, memory and disk load on servers
Nagios, Zabbix, MRTG, PRTG	Continuous monitoring, graphs and alerts for the whole network

4.5.3 Optimization Techniques

1. Traffic management and Quality of Service (QoS)

QoS classifies traffic and gives priority to important or delay-sensitive packets such as voice, video and critical applications. **Traffic shaping** smooths bursts by buffering excess packets, while **policing** drops or marks packets above an agreed rate. Bandwidth-hungry but non-urgent traffic such as backups or software updates can be limited or scheduled for off-peak hours.

2. Caching

Storing frequently used data closer to the user reduces repeated work: the Squid proxy for web content, a local caching DNS resolver for name lookups, database buffer caches and content delivery networks (CDN) for public websites.

3. Load balancing and clustering

A load balancer spreads requests across several servers so that no single server is overloaded. Methods include round robin, least connections and weighted distribution. Health checks remove a failed server from the pool automatically, so load balancing also improves availability.

4. Network segmentation

Dividing a large network into smaller VLANs and subnets reduces broadcast traffic, limits the effect of faults and attacks and lets different departments receive different policies. Servers with heavy mutual traffic should be placed on fast links or in the same segment.

5. Link and hardware upgrades

When measurements show sustained high utilization, the remedy may be to upgrade a link (for example 1 Gbps to 10 Gbps), use **link aggregation** (combining several physical links into one logical link), replace old switches, add RAM or move databases to SSD storage. Upgrades should follow measurement so that money is spent on the real bottleneck.

6. Server and service tuning

- Disable unnecessary services and daemons to free resources and reduce attack surface.
- Tune operating system parameters such as TCP buffer sizes, file descriptor limits and connection limits.
- Tune each service: worker processes in web servers, buffer pool and indexes in databases, cache sizes in Squid.
- Use fast storage, correct RAID level and enough RAM to avoid swapping.
- Keep software patched, since updates often contain performance as well as security fixes.

7. Reducing data volume

Compression of web content, files and backups, as well as incremental backups and removal of duplicated data, lowers the amount of data crossing the network.

8. Redundancy and high availability

Duplicate critical components so that a single failure does not stop the service: redundant links and switches, NIC teaming, gateway redundancy protocols (VRRP/HSRP), server clustering, RAID, dual power supplies and UPS. Redundancy improves availability, which is also a part of service quality.

9. Wireless optimization

Place access points to give even coverage, use non-overlapping channels (1, 6 and 11 in the 2.4 GHz band), prefer the 5 GHz band for capacity, limit the number of clients per access point and use strong encryption.

10. Documentation and change control

Maintain network diagrams, IP address records and configuration backups, and record every change with its date and reason. Good documentation shortens troubleshooting time and prevents configuration mistakes.

4.5.4 Typical Problems and Remedies

Problem	Possible cause	Remedy
Slow file transfers	Congested link, duplex mismatch, faulty cable	Check utilization with monitoring tool; correct duplex setting; replace cable; upgrade link
Voice or video breaks up	Jitter and packet loss under load	Apply QoS; separate voice VLAN; reduce competing traffic
Web browsing is slow	High Internet usage, no caching	Install caching proxy; limit bandwidth for non-essential sites
Application server overloaded	Too many requests for one server	Add servers and a load balancer; tune and cache
Database queries slow	No indexes; small buffer pool	Add indexes; tune buffer size; use SSD storage
Excessive broadcast traffic	One large flat network	Create VLANs and subnets
Frequent service failure	Single point of failure	Introduce redundancy and monitoring alerts
Logs from servers do not match in time	Unsynchronized clocks	Deploy an internal NTP time server

Chapter Summary (Quick Revision)

- **Network design** begins with assessing needs: applications, users, services, security and safety, and growth. Capacity planning prepares the network for future demand (typically 3 to 5 years).
- **Meeting needs** means choosing the network type (LAN/MAN/WAN, peer-to-peer or client-server, wired or wireless), the structure (star and hierarchical core-distribution-access) and suitable servers (CPU, RAM, RAID storage, redundancy).
- **Database server:** install MariaDB/MySQL, run `mysql_secure_installation`, create databases and users with GRANT, change bind-address for remote access, open port 3306 only for the LAN and take regular backups with `mysqldump`.
- **VNC server:** uses the RFB protocol on port 5900 + display number; install TigerVNC and a desktop, set password with `vncpasswd`, configure `xstartup`, start with `vncserver :1` and connect through an SSH tunnel.
- **Time server:** NTP on UDP 123 with stratum hierarchy; configure chrony with upstream pool, allow LAN network, verify with `chronyc sources` and `chronyc tracking`.
- **Caching proxy server:** Squid on port 3128 with ACLs, `cache_mem` and `cache_dir`; saves bandwidth, speeds up access and controls browsing. Check TCP_HIT and TCP_MISS in `access.log`.
- **Optimization** is a continuous cycle: monitor, analyze, tune and verify, using QoS, caching, load balancing, segmentation, upgrades, redundancy and service tuning.

Multiple Choice Questions

Choose the correct option for each question. The answer key is given after the questions.

Q1. Which activity should be done first when designing a network?

- (a) Buying servers
- (b) Assessing network needs
- (c) Installing cables
- (d) Configuring the firewall

Q2. Growth and capacity planning is mainly concerned with:

- (a) Selecting the colour of cables
- (b) Removing old users
- (c) Estimating future users, traffic and storage so the network can expand
- (d) Disabling security features

Q3. In which topology does the failure of the central device stop the whole network?

- (a) Star
- (b) Bus
- (c) Ring
- (d) Mesh

Q4. The three layers of the hierarchical network model are:

- (a) Core, Distribution and Access
- (b) Physical, Data link and Network
- (c) Client, Server and Proxy
- (d) Input, Process and Output

Q5. Which RAID level uses mirroring of two disks?

- (a) RAID 0
- (b) RAID 5
- (c) RAID 6
- (d) RAID 1

Q6. The default port used by MySQL/MariaDB is:

- (a) 1433
- (b) 5432
- (c) 3306
- (d) 1521

Q7. Which SQL statement is used to create a new database?

- (a) MAKE DATABASE college_db;
- (b) NEW DATABASE college_db;
- (c) CREATE DATABASE college_db;
- (d) ADD DATABASE college_db;

Q8. The GRANT statement in a database server is used to:

- (a) Give privileges to a user
- (b) Delete a table
- (c) Take a backup
- (d) Restart the server

Q9. VNC uses which protocol to transfer the remote screen?

- (a) RDP
- (b) SSH
- (c) Telnet
- (d) Remote Framebuffer (RFB)

Q10. A VNC server running on display :1 normally listens on TCP port:

- (a) 5900
- (b) 5901
- (c) 5001
- (d) 5801

Q11. The safest way to use VNC over an untrusted network is to:

- (a) Disable the VNC password
- (b) Open all ports in the firewall
- (c) Tunnel the connection through SSH
- (d) Use Telnet for login

Q12. NTP uses which transport protocol and port?

- (a) UDP 123
- (b) TCP 80
- (c) UDP 53
- (d) TCP 25

Q13. Stratum 0 in the NTP hierarchy refers to:

- (a) Client computers
- (b) Stratum 2 servers
- (c) Network routers
- (d) Reference clocks such as atomic clocks or GPS

Q14. The default port on which the Squid proxy server listens is:

- (a) 8080
- (b) 3128
- (c) 3306
- (d) 5900

Q15. Giving higher priority to voice traffic over file downloads is an example of:

- (a) Caching
- (b) NAT
- (c) Mirroring
- (d) Quality of Service (QoS)

Answer Key for MCQs

Q. No.	Answer	Explanation
1	(b)	Design starts with requirements; equipment is chosen afterwards.
2	(c)	It forecasts future demand and plans resources to meet it.
3	(a)	In a star topology all devices depend on the central switch.
4	(a)	Core, distribution and access layers form the standard campus design.
5	(d)	RAID 1 stores identical data on two disks and survives one disk failure.
6	(c)	3306 is the standard MySQL/MariaDB port (1433 SQL Server, 5432 PostgreSQL, 1521 Oracle).
7	(c)	CREATE DATABASE is the standard SQL command.
8	(a)	GRANT assigns permissions such as SELECT or INSERT to a user.
9	(d)	VNC is based on the RFB protocol.
10	(b)	Port = 5900 + display number, so :1 uses 5901.
11	(c)	VNC traffic is not encrypted by default, so SSH tunnelling is used.
12	(a)	NTP works on UDP port 123.
13	(d)	Stratum 0 devices are the highly accurate reference clocks.
14	(b)	Squid uses port 3128 unless changed with http_port.
15	(d)	QoS prioritizes delay-sensitive traffic.

Quick key: 1-b, 2-c, 3-a, 4-a, 5-d, 6-c, 7-c, 8-a, 9-d, 10-b, 11-c, 12-a, 13-d, 14-b, 15-d

Broad Questions

Answer the following questions in detail.

1. Explain the steps involved in assessing network needs for designing a new network. Describe how applications, users and network services influence the design.
2. Discuss security and safety considerations in network design. Explain growth and capacity planning with a suitable example.
3. Compare the different network types and network topologies. How would you choose the type and structure of a network for a college campus? Draw and explain the hierarchical three-layer model.
4. What factors are considered while choosing servers for a network? Explain server hardware requirements, RAID levels and the difference between dedicated, shared and virtualized servers.
5. Describe the procedure to install and configure a database server (MariaDB/MySQL). Explain creation of a database and a user, remote access, and backup and restore.
6. What is VNC? Explain its working with a diagram and describe the steps to create and secure a VNC server on Linux.
7. Explain the need for a time server. Describe the NTP stratum hierarchy and the steps to configure a time server and its clients using chrony.
8. What is a caching proxy server? Explain its advantages and describe how to configure Squid with access control lists, including cache hits and misses.
9. Explain the techniques used to optimize network services. Describe the optimization cycle and the tools used to find bottlenecks.
10. Write short notes on any three: (a) VLAN and network segmentation, (b) Quality of Service, (c) Load balancing, (d) Transparent proxy, (e) Database security and performance tuning.