

9 Jan 2026
Tuesday

Rewatch

17 Jan 2026

6116hayat

Date _____

Page No. _____



Class 1:

Introduction to computer Networks, DNS, AI Assistance

• IP → Internet Protocol

[logical address denoted to user]

23:00 → start of class

Chair addresses → IP

room no → port

→ AI ⇒ 26:00 - 47:30

• Router → routing data

(jisme request kiya usko dena)

• Synthetic data

• Article

↳ problem

↳ solution

↳ implementation

• wifi device → (router + modem)

Break

1:27:30 - 1:47:00

→ How chaincode.com opens

47:30 - 1:27:30

→ 1:36:22

• comp + wires + rule = network

↳ protocol

20 Jan 2026
Tuesday

(1:49:00 - 2:20:30)

• many network forms internet

↳ main story (how browser)

• Internet:

(how browser reaches the)

chaincode.com website

Global connected networks

that follow some common

→ DNS Terms to search about

protocol (udp/tcp/smtp/ftp)

↳ decentralization

• Difference b/w Internet &

↳ delegation

web (www) (90)

↳ loose coupling

• How many virtual ports in computer?

DNS always says, "who should I talk to next?"

More than 65K.



① Browser

- The program you use (chrome)
- You type a website name (www.example.com)
- Browser asks "what is the IP address of this site?"

conceptual → 13 root servers

implementation → 1600+

② Recursive DNS Server

- usually provided by ISP (Google DNS/Cloudflare)
- like your personal assistant for finding website
- it does the "hard work" of asking other servers until it finds the answer
- it caches (remembers) answers to speed things up next time.

From the next slide 2

③ Root Server

- first stop for recursive server
- Root servers don't know the exact IP of the site, but they know where to look next

(to the right TOP-LEVEL DOMAIN)

• [TLD] server

like .com / .org / .net

2:06:10

④ Define

→ Browser

→ Recursive DNS Server

→ Root Server

→ TLD Server

→ Authoritative Server

- they point to the TLD servers (like .com / .org)



④ TLD Server

- Handles specific extensions
(Top-level Domains)

Example

.com servers know
about all domains ending
in .com

- They don't know final IP,
but they point to
authoritative server

⑤ Authoritative Server

- Final source of truth
- holds the actual DNS
records for domain
- It gives the recursive
server the exact IP
address

③ Root Server → points to the
right TLD server

④ TLD Server → points to the
authoritative server for
that domain.

⑤ Authoritative Server →
gives the final IP address

⑥ Recursive DNS Server →
returns the IP to browser

⑦ Browser → connects to
website using that IP.

→ Generic Meaning

Recursive → refers back to itself
or repeat a process in
steps a condition is met.

Q How they all work
together, dependent on
each other?

→ In DNS terms

① Browser asks recursive
DNS server

• A recursive DNS server,
(also called resolver) takes
your browser's request
& keeps asking other DNS
servers until it gets IP
addresses

② Recursive DNS Servers →
check cache, if not found,
ask root server.

2:10:30

Date _____

Page No. _____



→ commands of computer

① Ping

- It checks if one computer (device) can reach another over a network (internet)

- It sends small packet of data (called ICMP Echo request) to target

- If target is reachable, it replies with an Echo Reply.

useful in
www

• check connectivity :

(network/website reachable)

measure speed :

How fast is response

Troubleshoot issues:

If no reply, maybe the site is down

another command:

ping -c 2 google.com

- The command then shows you:

↳ whether the target is reachable /

↳ How long it took (latency, measured in millsec)

↳ if any packets were lost

same as before but with '-c'

-c counts the no. of sent by me google to you

↳ by mentioning 2

(you mean send me 2 packets)

Example:

ping google.com

Basic working of ping, it's
it continuously sends packet
until stopped using 'ctrl + c'

② nslookup

result code:

Reply from 142.250.190.14: bytes=32
time = 20ms TTL = 117

nslookup google.com

used to query the DNS
(Domain Name System) to

Reply from → IP add of Google serve
Time = 20ms → round trip time
(How fast response come back)

Find information about
domain "google.com"

TTL = (Time to Live) ↑ showing,

DETAILED NOTEBOOK how many hops the
packet can make before discarded

→

end.

2:23:00 - 2:36:00

Date

Page No.



nslookup → a tool that asks DNS servers for domain information

google.com → the domain you are looking up

lookup tool

dig → command

refers to root zone

(very top of DNS hierarchy)

After running we'll see:

→ DNS server your computer used to perform the lookup

④ ipconfig / ifconfig

→ IP addresses associated with google.com

ipconfig:

→ sometimes additional info like canonical names (cname)

→ Platform → Windows

→ purpose:

Shows & manages comp's network configuration.

→ It displays:

IP add/subnet mask/DNS servers

→ usage

Non-authoritative server:

It means information come from cache; not from authoritative server

ipconfig → basic network info
ipconfig/all → detailed info (MAC add/DNS)

ipconfig/release and ipconfig/renew

→ Refresh IP addresses from DHCP

③ dig command

a powerful tool, to get the detailed information about domain names.

ipconfig /flushdns → clear DNS cache

hayatzone

dig google.com

⑤ top ipconfig:

→ platform → Linux/macOS (older)

→ purpose → config & displays network interface

dig com NS queries the

→ It displays:

DNS system for name server (NS) records of .com top-level domain.

→ common use:

ifconfig → shows active network interface

ifconfig eth0 up/down

Enable/disable a network interface

① `top` : shows real-time view of system on Linux/Unix

② `uptime` : shows how long system has been running since last reboot

Date _____

Page No. _____



③ curl command

• `-o file.html` → save output to a file

⇒ **Basic** (Client URL → CURL) `curl https://example.com -o page.html`

- tool used to transfer data to or from a server using different protocols.

• `-X POST` → sends a post request

- extremely common for testing API's downloading files, or checking web connectivity.

`curl -X POST`

`https://api.example.com/data -d "name=John&age=25"`

⇒ What curl Does

- connects to a given URL (like a website or API endpoint)

• `-H` → Add custom headers

- supports many protocols (HTTP/HTTPS/FTP/SFTP) etc.

`curl -H "Authorization: Bearer Token"`
`https://api.example.com/users`

- can send requests (GET / POST / PUT / DELETE) and receive responses

• `-u` → provide username & password for authentication

- works directly from command line.

`curl -u user:password`

`ftp://example.com/file.txt`

⇒ Basic Usage

`curl google.com / https://example.com`

fetches HTML content of

⇒ Why Useful

`example.com` and prints it in the terminal.

- Developers → Test APIs quickly
- Admins → check server response
- Users → Download files or check connectivity.

⇒ Common Options

• `-I` → fetches only headers

`curl -I https://google.com`