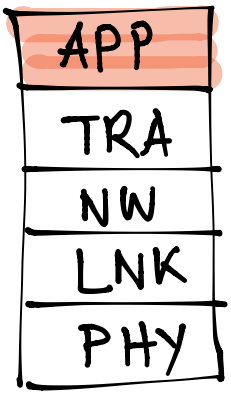
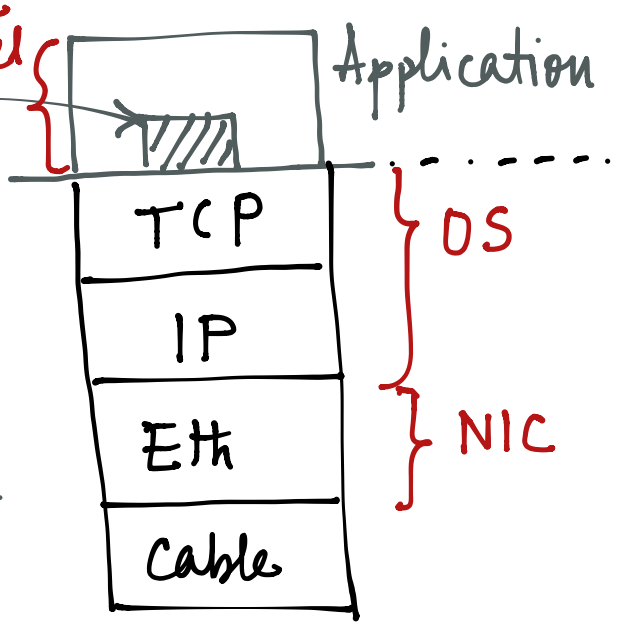
Application Layer



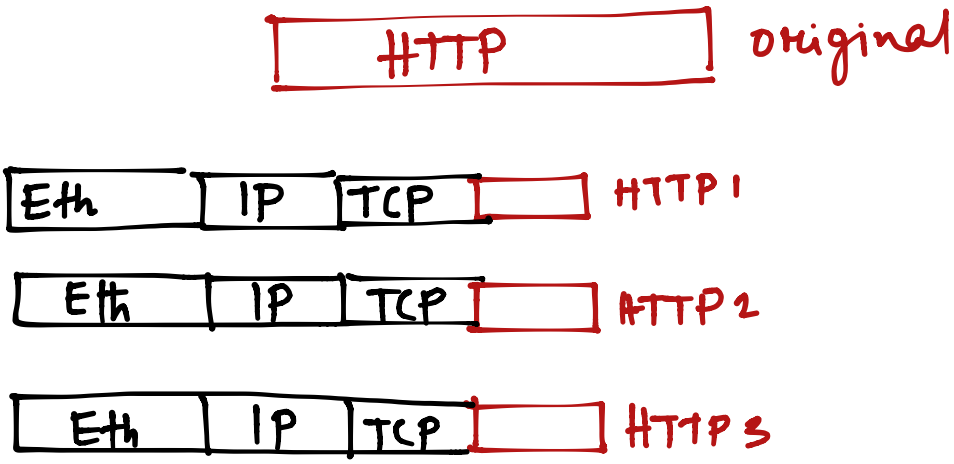
Plethora of applications
- all use the services provided by the transport layer
- in particular, Socket API to access the network

Our focus: few widely used applications

every application or "app" has an implementation of an application layer protocol



App layer messages are large



Need of Applications

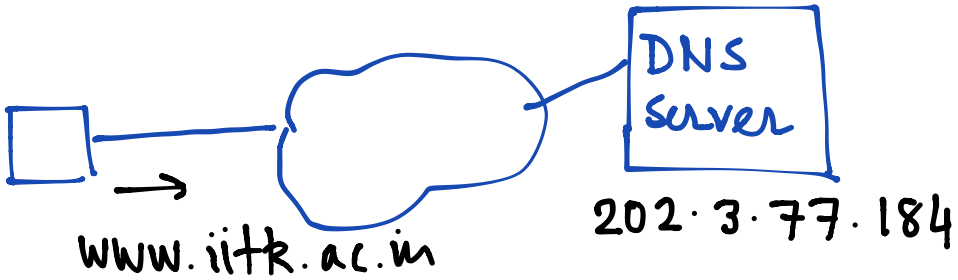
App type	Message Size	Reliability	TRA Layer Service Used
Web	Large	yes, bytestream	TCP
Real-time video/audio	Small	no, message based	UDP
Domain Name System	Small	partly, message based	UDP

not provided by any TRA protocol
app layer has to design by itself.

Some prominent APP layer protocols

Purpose	Protocol	↓ order of development
Remote access	Telnet	
	SSH (secure shell)	
File transfer	FTP	
Email	SMTP	
	POP	
News	NTTP	(now obsolete)
Web	HTTP	
Web	DNS	
Web	CDNs	(popular content)
Peer to peer	Bit Torrent	
Web (video)	RTSP	(real time streaming protocol)
Web (conferencing)	VOIP	(voice over internet protocol)

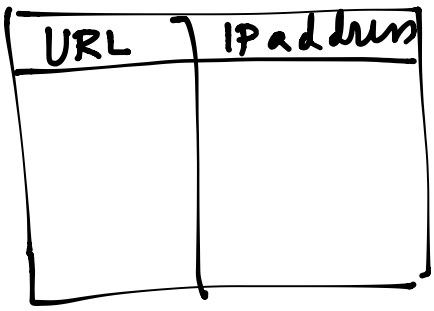
Domain Name System (DNS)



Names – high level identifier (URL)

Address – low level identifier (IP address)

Resolution – mapping from Name to Address



Phonebook

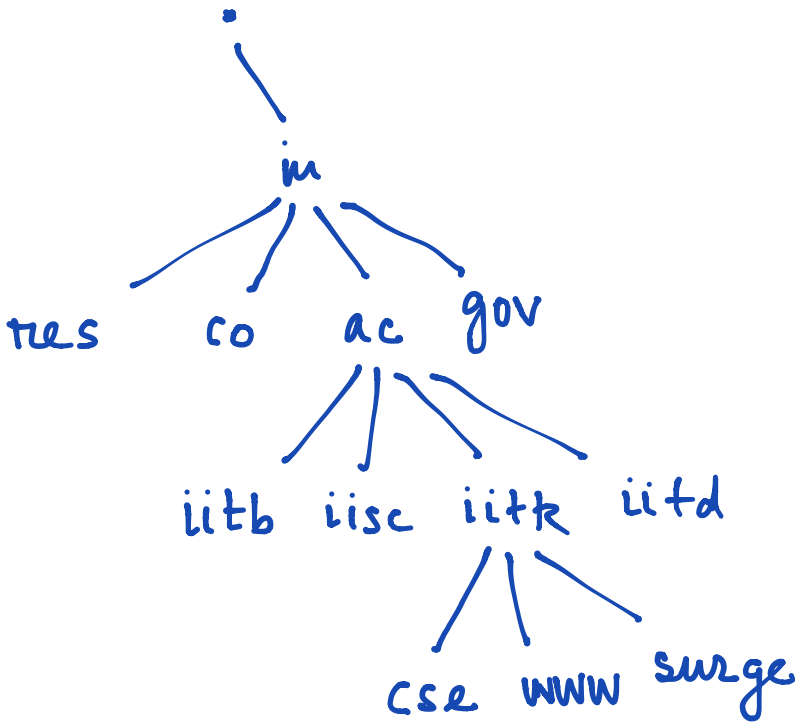
Before DNS

- mapping was via a central file (master list) called HOSTS.TXT
- copied in each machine and regularly updated – not scalable

DNS - General Overview

- distributed directory based on hierarchical namespace

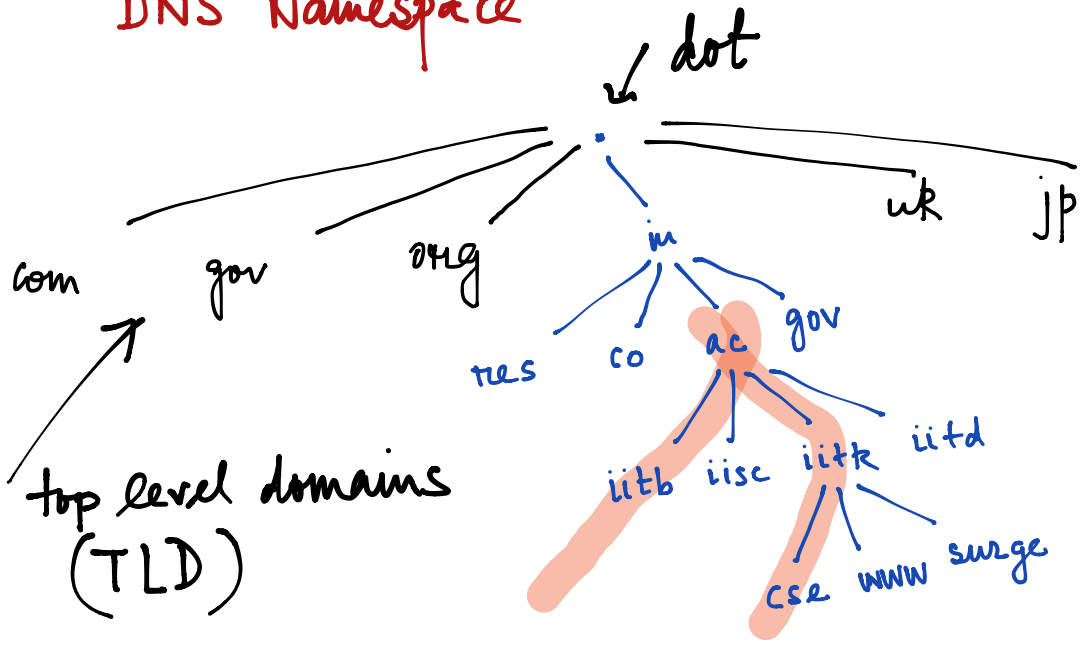
cse . iitk . ac . in



- Automated protocols to tie them together

Goal: • Easy to manage - distributed
• Efficient - quick and few resource

DNS Namespace



- The namespace tree is administered by ICANN (Internet Corp. for Assigned Names & Numbers)

- 22+ generic TLDs .com, .edu, .org ...

- 250+ country code TLDs - .in, .uk, .jp, ...

[domain names are critical for financial reasons]

this TLDs have usage policies

A ZONE is a contiguous portion of the namespace

Zones are the basis of distributed control

- .ac.in registrar administers this domain
- IITK administers iitk.ac.in
- CSE administers cse.iitk.ac.in

Each zone has a **name server** to contact for information about it

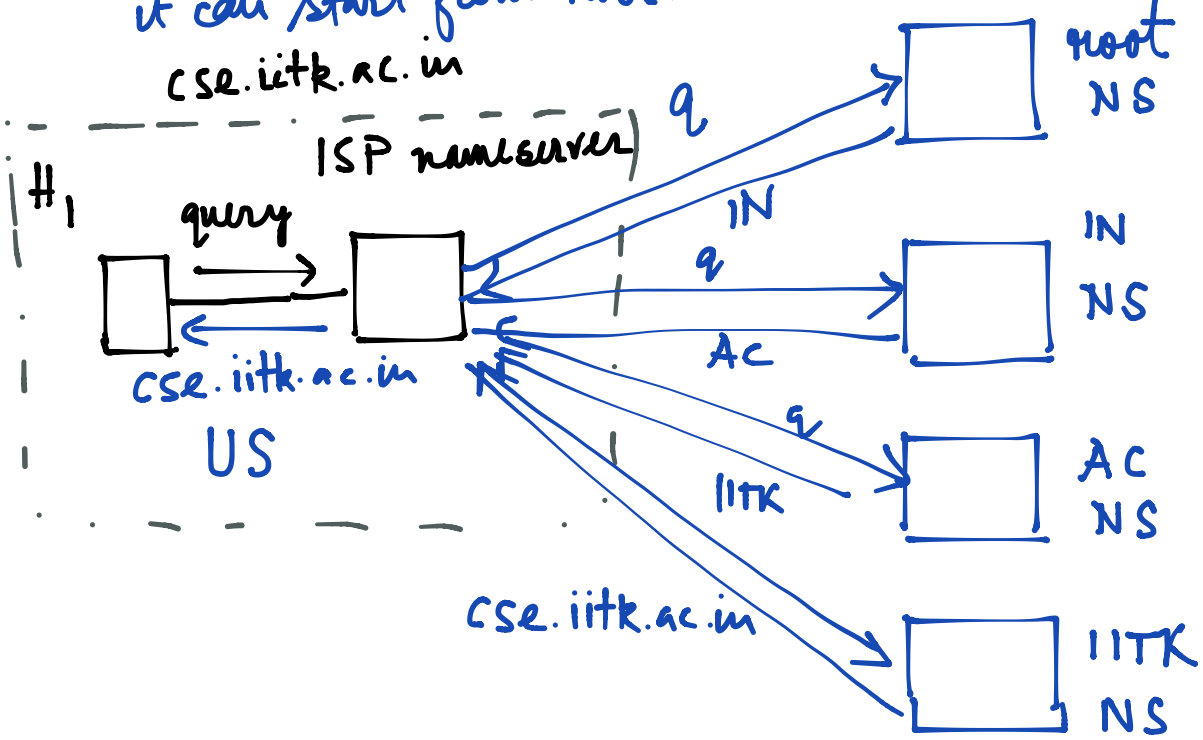
- Zones must include contacts for **delegations** (the next level in the namespace tree)

e.g., .ac.in knows the nameserver for iitk.ac.in

- finding the address of cse.iitk.ac.in is delegated to iitk.ac.in

DNS – Address Resolution

– If the address is unknown, it can start from root.



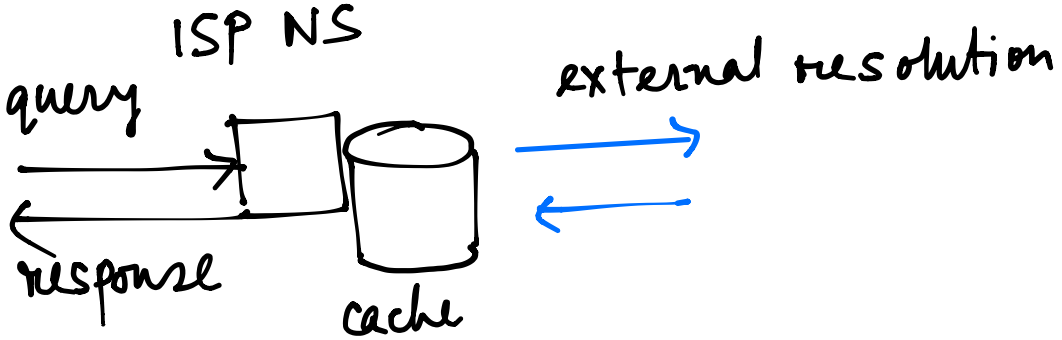
Iterative vs Recursive query

Recursive – The server takes the load of resolving the address and not the client, e.g., the ISP NS

Iterative – The receiver responds with the forwarding information and forgets the query

Lightweight & good for high load servers

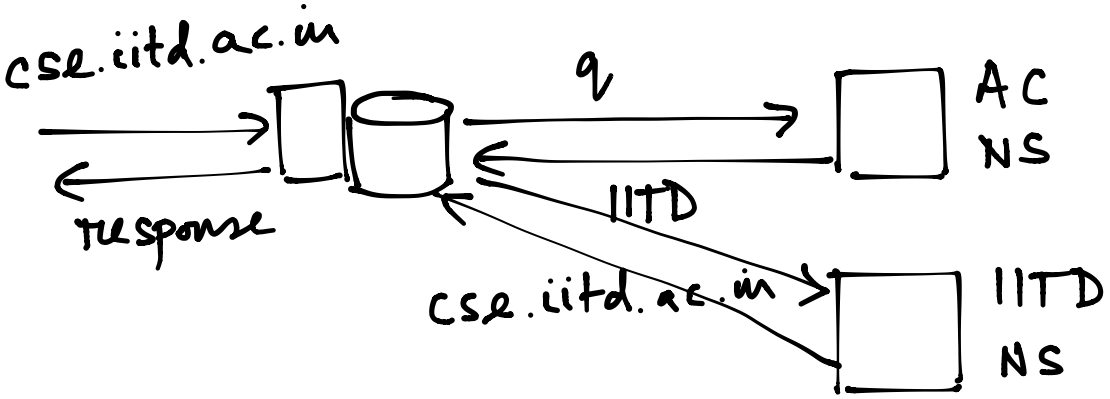
Natural Extension – Resolution Caching



Advantages

- Response time is smaller
- cache includes partial answers

iitk.ac.in NS
.ac.in NS
.in NS



Local nameservers are administered by the ISP/organizations
- or public NSs - e.g. Google NS.

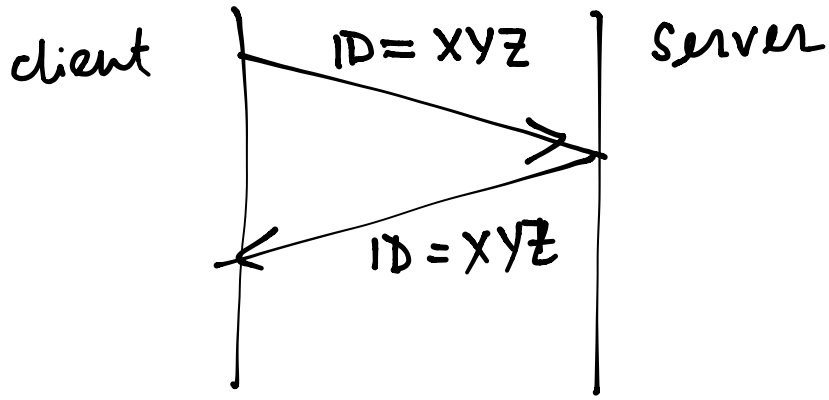
How does a host know the local NS
- part of the DHCP response.

Root Nameservers

- 13 server names
- > 250 distributed server instances located all over the world
  - see them in www.root-servers.org
- use the same IP address anycast DNS queries to the nearest root NS.

DNS protocol

Query and Response



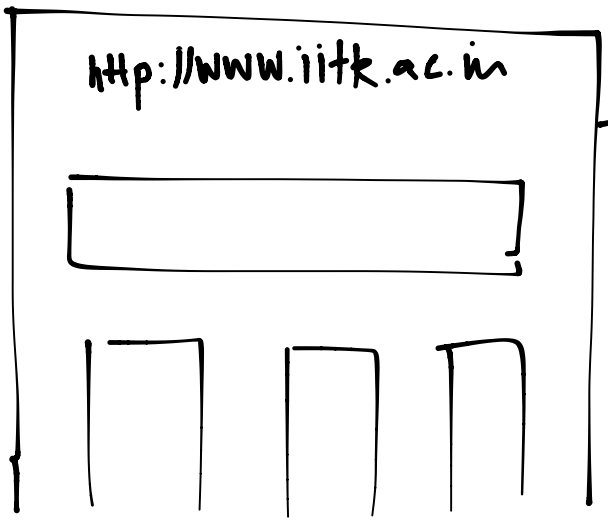
- 16 bit ID uniquely identifies the query.
- ARQ for reliability - server is stateless
- DNS built on UDP, port 53

DNS endnotes

- Security is a major issue
- Servers are prime targets of attacks
- DNSSEC (security extension)

HyperText Transfer Protocol (HTTP)

Web is primarily driven by HTTP.



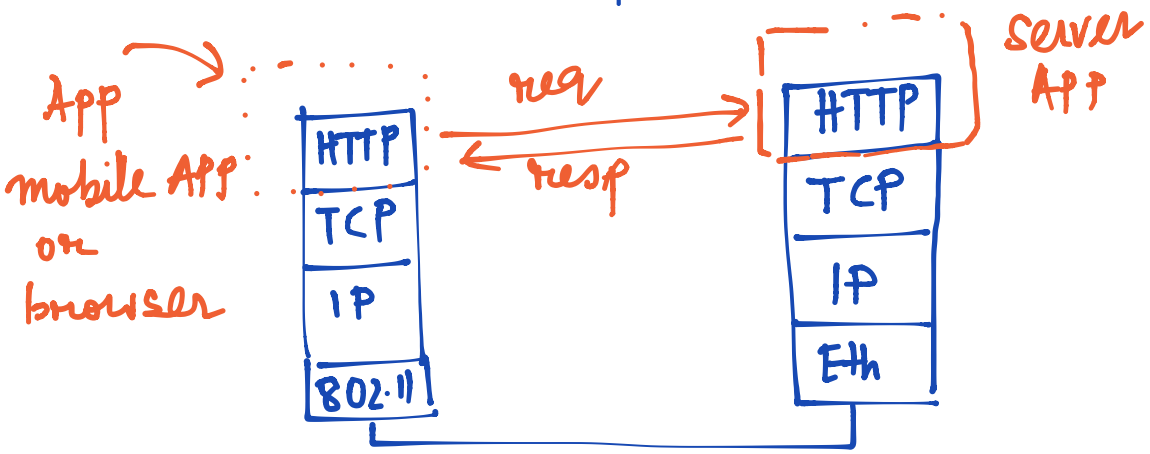
→ The webpage is a collection of linked resources

Text, images, videos hyperlinks to other pages, etc.

The browser decides how to display them and the markup language specifies it.

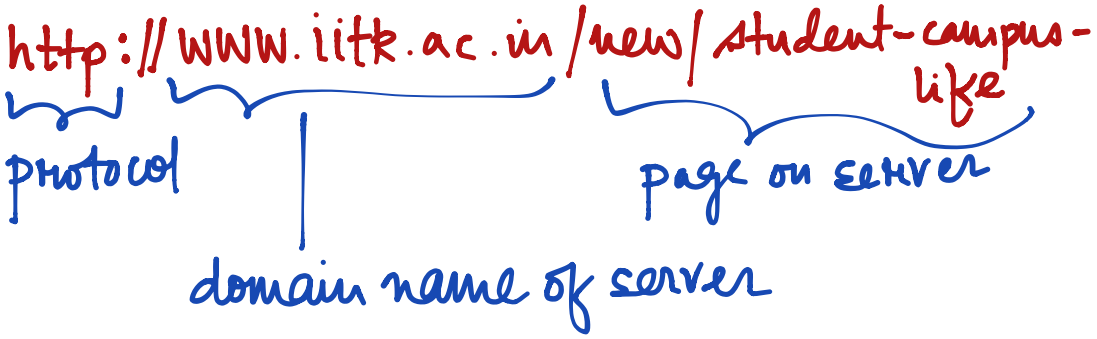
HTTP is a request/response protocol

- Runs on TCP, port 80



Fetching a webpage using HTTP

- URL:

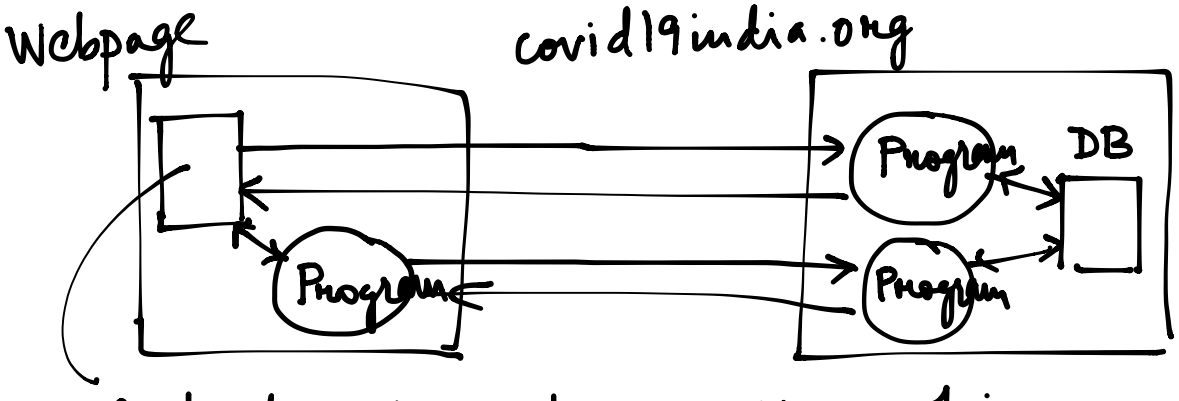


Steps (from client's view)

1. Resolve the server to IP (DNS)
2. Setup the TCP connection
3. Send HTTP request
   - await HTTP response
   - Execute / fetch embedded resources / render
     find the page
n. teardown the TCP connection

Static & Dynamic Webpages
↳ result of program execution

- languages: javascript / flash on client,
  PHP on server, Perl / Python / Ruby



Content evolving dynamically evolving.

HTTP details

REQUEST

Method	Purpose
GET	Read a webpage
POST	Append to a webpage
⋮	⋮

RESPONSE

1XX	information
2XX	success
3XX	redirection
4XX	client error
5XX	server error

Performance of HTTP

What is performance that is measurable?

- direct measure – Page Load Time (PLT)
- $t_{\text{view content}} - t_{\text{click}} = \text{PLT}$
- Very important measure, e.g., for user engagement in e-commerce.

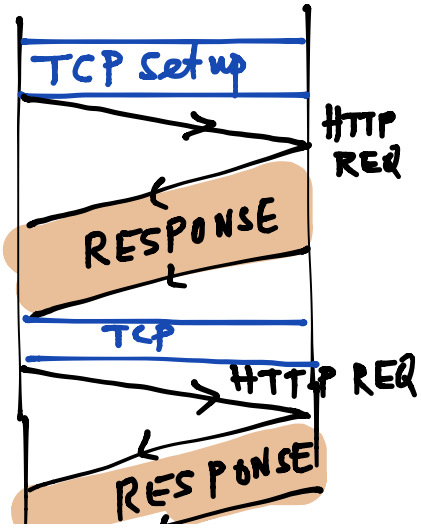
Depends on various factors

1. Structure of page/content
2. HTTP & TCP
3. Network RTT & bandwidth

Early HTTP: Sequential load

Webpage having multiple embedded objects/ web resource, e.g., images

Easy to implement but poor user experience



multiple TCP connection setup with same server – multiple slow-start phases

not using the network effectively.

Improvement 1 : Persistent Connection

– Natural to consider the connections made in parallel

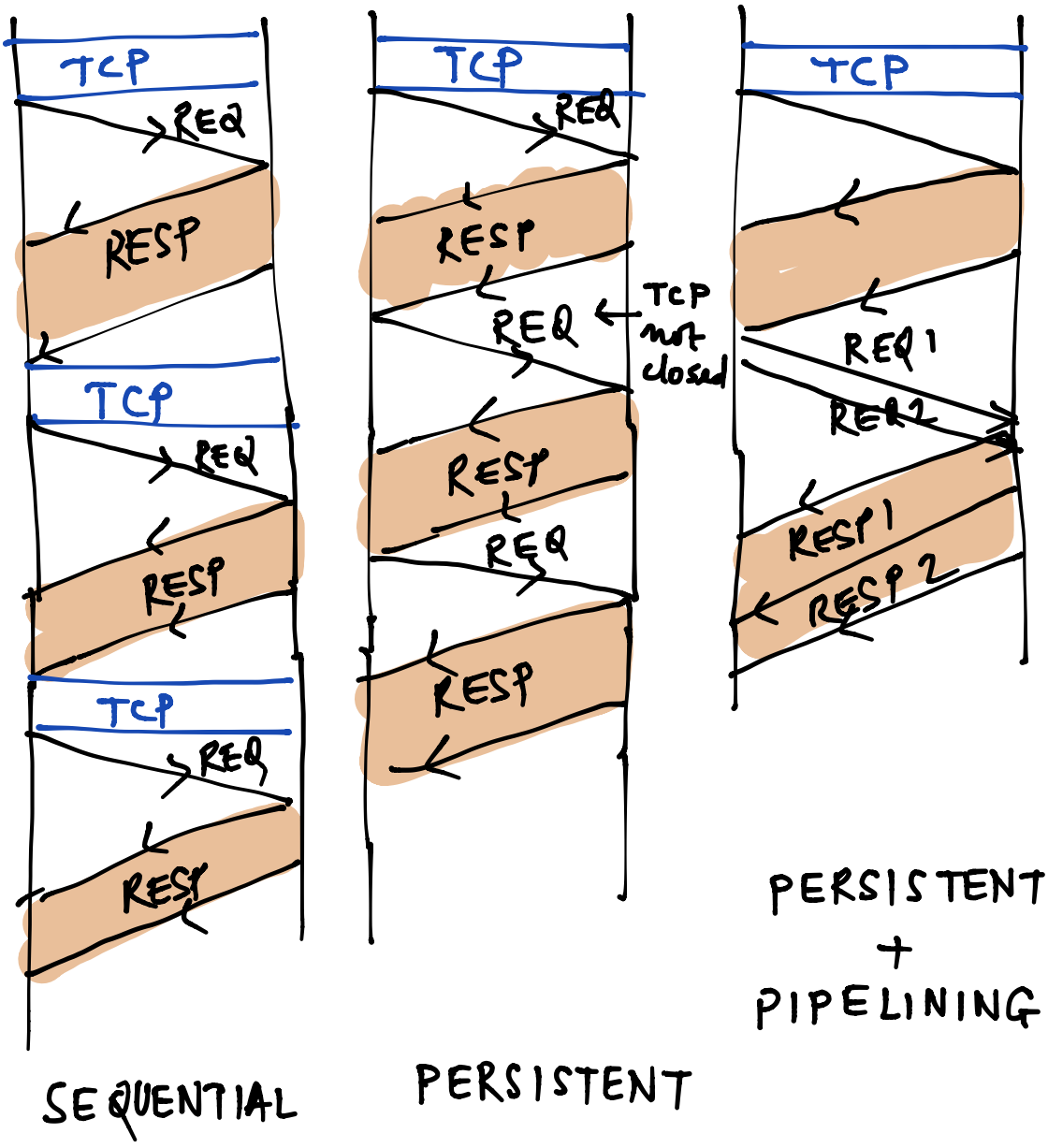
– The parallel connections will still have multiple TCP overhead and each of them will slow start – not ideal

– also they are competing for the same bandwidth

– Persistent connection is one TCP connection which is not torn down before an explicit signal

  – Single TCP connection
  – One slow start phase
  – more bandwidth for data

Ex.

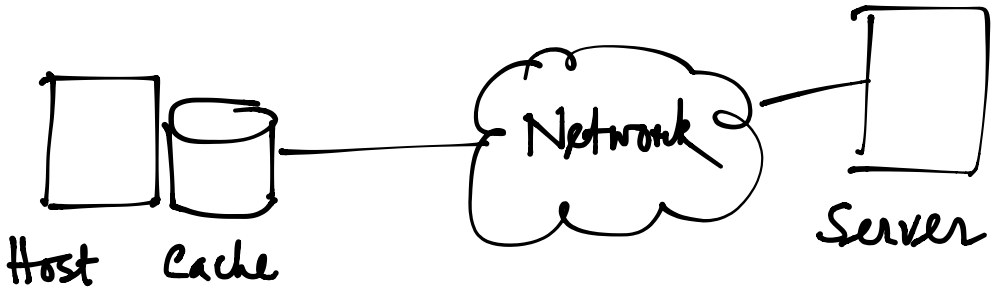


Persistent connection needs to decide how long to keep persisting.

- typically short to keep the load on server low.

Improvement2: Caching & Proxy

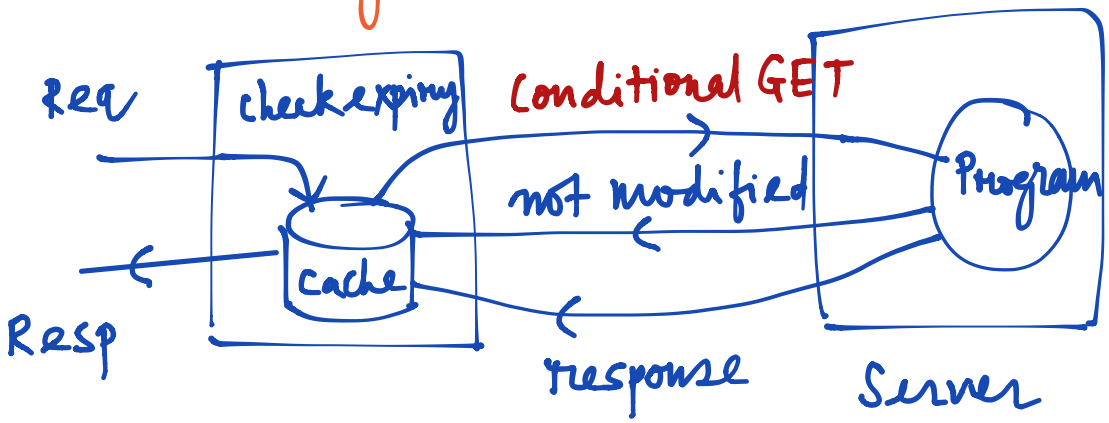
Motivation: Users typically revisit webpages



Q: When the cached copy is okay to use?

- Local determination of the copy
  1. check metadata, e.g., "expires" in the header
  2. heuristics to guess - cacheable, e.g., not dynamic page, freshly valid, e.g., time between the cached copy is not very old.
- Revalidate the copy with the remote server
  1. via the metadata, e.g., modified date and time with the server
  2. header of the content, e.g., Etag entity tag validate.

Typically local & remote validation are done together



These are still done at individual level

- typically there is large overlap of the content a group of users

Improvement: place an intermediary between a pool of clients and external web servers

Proxy caching

- Benefits: proxies can ensure better security and caching (shared)
  - organizational access policies

Ex.