3. BOOTP, DHCP, Domain Name Space

DATE / /

DHCP

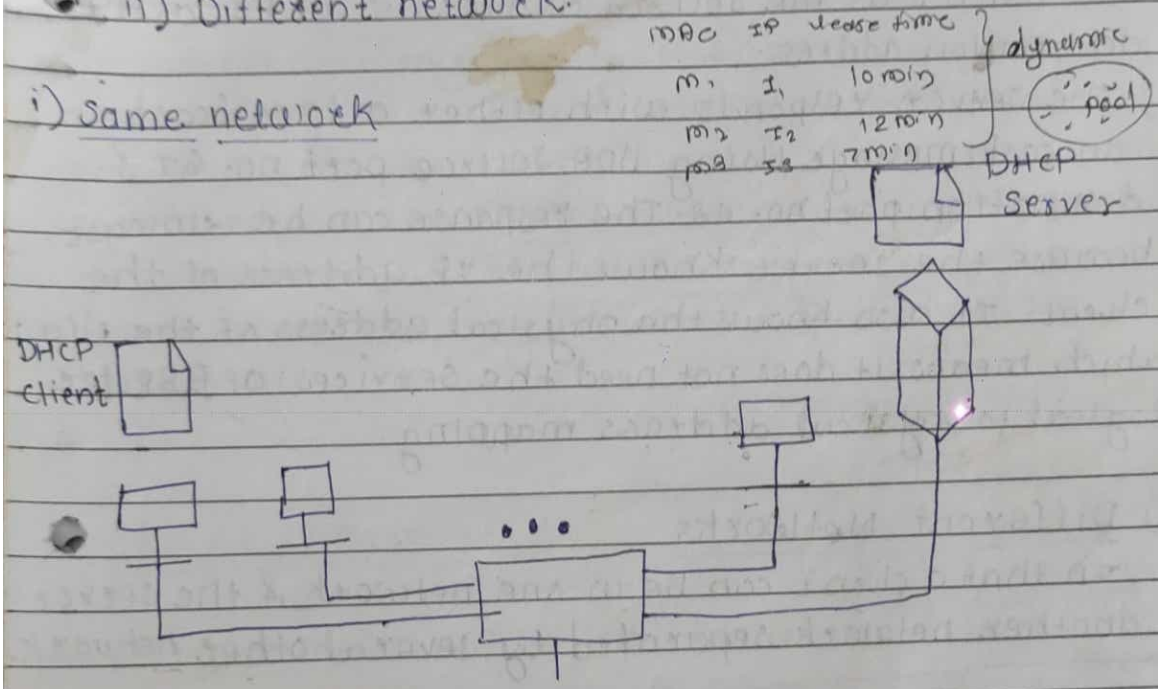
DHCP → Dynamic Host configuration Protocol.
It is a client/server protocol designed to provide the four pieces of information for a diskless computer or computer that is booted for the first time. It is successor to BOOTP & is backward compatible with it.

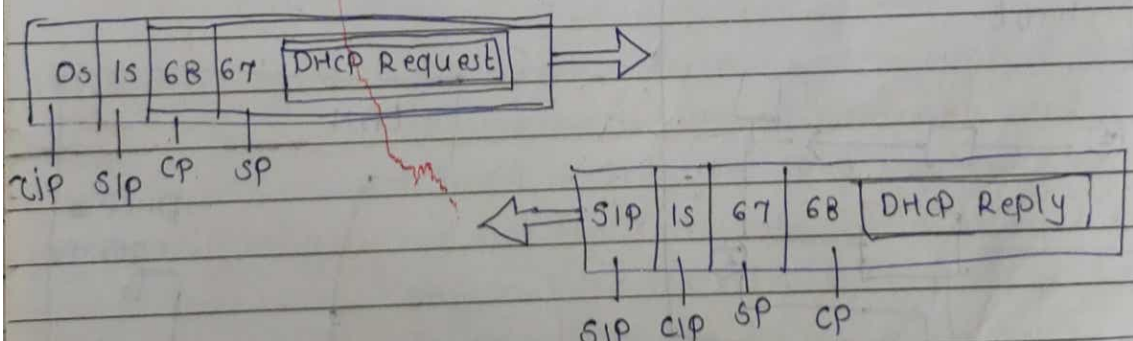
DHCP operation — DHCP - means uniquely

The DHCP client & server can either be on the same network or on different networks.

i) same network
ii) Different network.

i) Same network





CP - Client Port No.
SP - Server Port No.
CIP - Client Port IP address
SIP - server IP address

Fig- client & server on the same network.

1. The DHCP server issues a passive open command on UDP port no. 67 & waits for a client.
2) A booted client issues an active open command on port number 68. The message is encapsulated in a UDP user datagram using the destination port no. & the source port no. 68.
The UDP user datagram, in turn is encapsulated in an IP datagram. The reader may ask how client can send an IP datagram when it knows neither it's own IP address nor the server's IP address. The client uses all 0's as the source address & all 1's as the destination address.
3) The server responds with either a broadcast or unicast message using UDP source port no. 67 & destination port no. 68. The response can be unicast because the server knows the IP address of the client. It also knows the physical address of the client which means it does not need the services of ARP for logical to physical address mapping.

ii) Different Networks
In that a client can be in one network & the server in another network separated by several other network

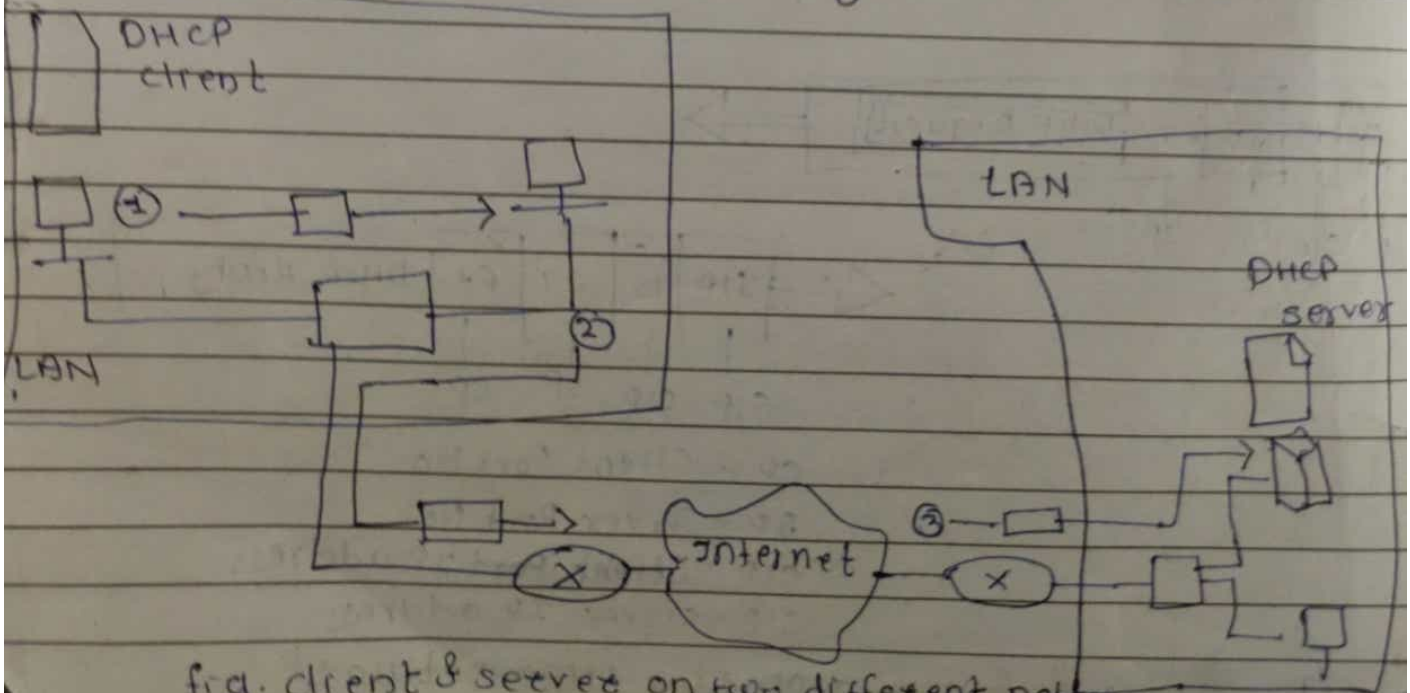
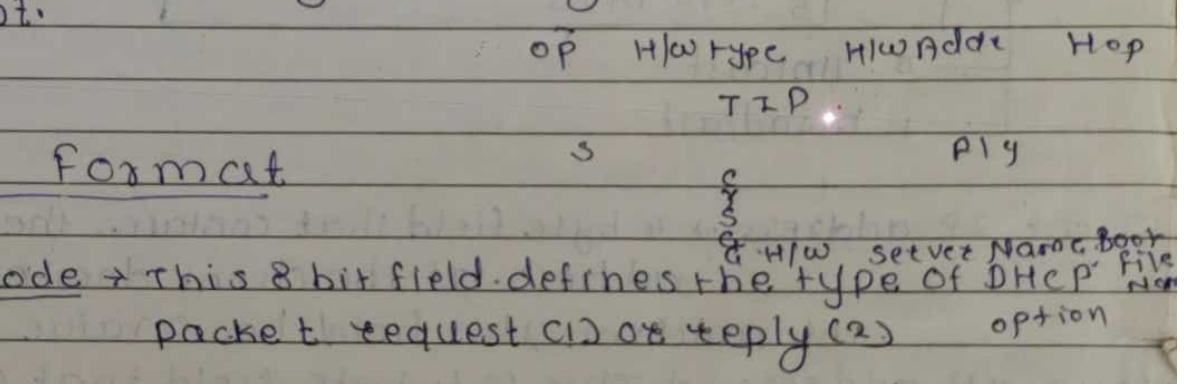


fig. client & server on two different network

DHCP request is broadcast because the client does not know the IP address of the server. A broadcast IP datagram cannot pass through any router. A router receiving such packet discard it. Recall that IP address of all 1's in is a limited broadcast address.

To solve the problem, there is need for an intermediary one of the hosts can be used as relay. The host in this case is called a relay agent. The relay agent knows the unicast address of a DHCP server & listens for broadcast messages on port 67. When it receives this type of packet, it encapsulates the message in a unicast datagram & sends the request to the DHCP server. The packet carrying a unicast destination address is routed by any router & reaches the DHCP server. The DHCP server knows the message comes from a relay agent because one of the fields in the request message defines the IP address of the relay agent. The relay agent, after receiving the reply, sends it to the DHCP client.

Packet Format



Operation code → This 8 bit field defines the type of DHCP packet request (1) or reply (2)

Hardware type → 8 bit field defining the type of physical network. each type of network has been assigned an integer. e.g Ethernet value 1

Hardware length → 8 bit field defining the length of the physical address in bytes. e.g for Ethernet value is 6.

Hop count → 8 bit field defining the maximum no. of hops the packet can travel.

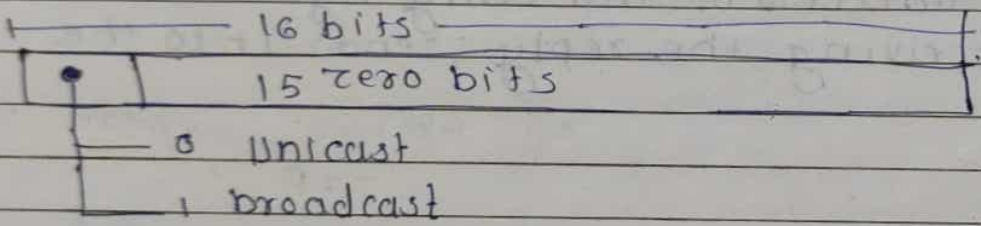
Transaction ID → This is 4 byte field carrying an integer. The transaction identification is set by the client & is used to match a reply with the

request. The server returns same value in it's reply.

Number of seconds → This is a 16 bit field that indicates the no. of seconds elapsed since the time the client started boot.

Flag → This is a 16 bit field in which only the leftmost bit is used & the rest of the bits should be set to 0s. A leftmost bit specifies a forced broadcast reply from the server. If the reply were to be unicast to the client, the destination IP address of the IP packet is the address assigned to the client; the client does not know it's IP address, it may discard the packet. if the IP datagram is broadcast, every host will receive & process the broadcast message.

Flag format



Client IP address → 4 byte field that contains the client IP address. If the client does not have this information, this field has a value of 0

Your IP address → This is 4 byte field that contains the client IP address. It is filled by the server at the request of the client.

Server IP address → This is a 4 byte field containing the server IP address of a router. It is filled by the server in a reply message

Gateway IP address → This is a 4 byte field containing the IP address of a router. It is filled by the server in a reply message.

Client hardware address → This is the physical address of the ~~router~~ client. Although the server can retrieve this address from the frame sent

DHCP Configuration

DHCP provide static & dynamic address allocation
1) static address allocation
2) Dynamic address allocation

1) Static address allocation

DHCP server has a database that statically binds physical addresses to IP addresses. ~~when~~ DHCP is backward compatible with the depreciated protocol BOOTP

2) Dynamic Address Allocation

It has 2nd database with pool of available IP addresses. This 2nd DB makes DHCP dynamic. when a DHSP client requests a temporary IP address, the DHCP server goes to the pool of available IP addresses & assigns an IP address for negotiable period of time.

When a DHCP client send a request to a DHCP server the server 1st checks it's static database. If an entry with the requested physical address exists in the static database, the permanent IP address of the client is returned. If the entry does not exist in the static database the server selects an IP address from the available pool assigns the address to the client & adds the entry to the dynamic DB.

The dynamic aspect of DHCP ~~provid~~ is needed when a host moves from network to network or is connected & disconnected from a network. it provides temporary IP address for limited period of time. It issues a lease for specific period of time. when the lease expires, the client must either stop using the IP address or renew the lease. The server has choice to agree or disagree with the renewal. If the server disagree, client stops using address.

Transition states

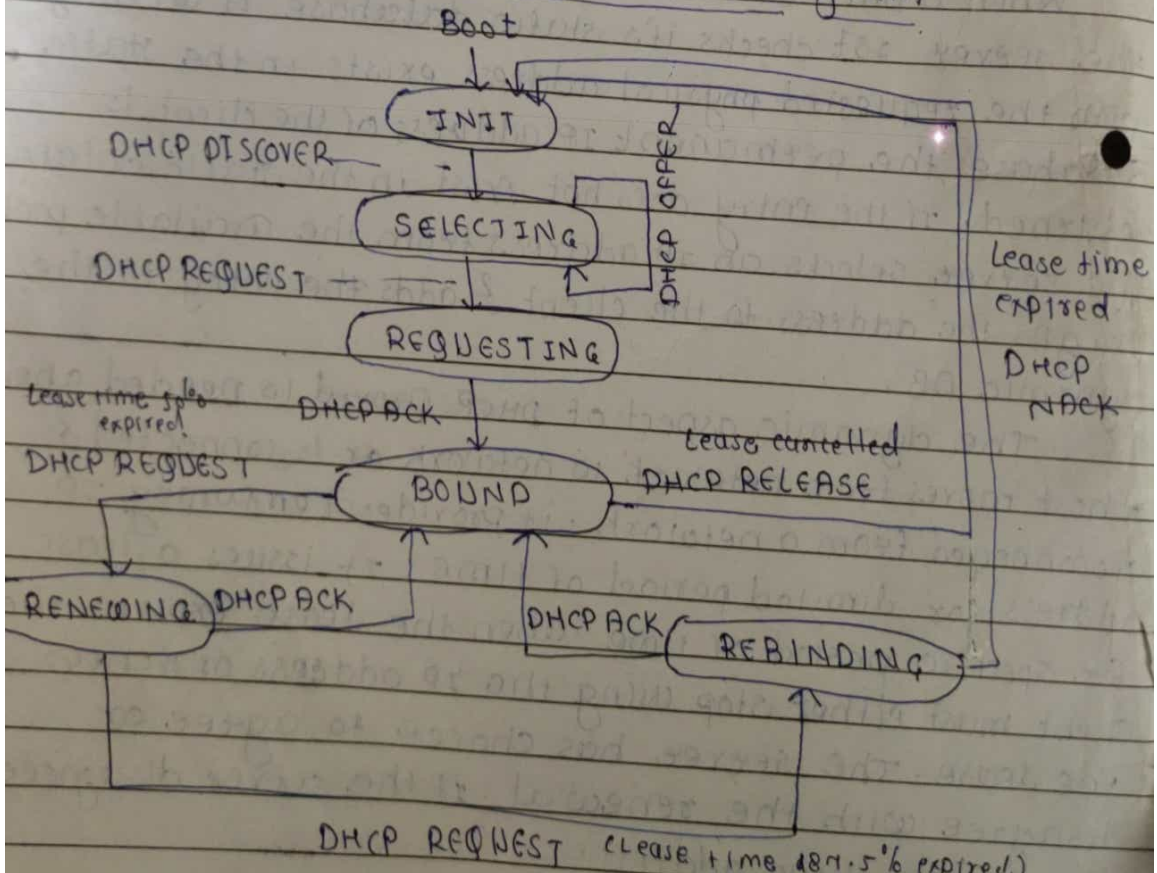
To provide dynamic address allocation, the DHCP client acts as a state machine that performs transition from one state to another depending on the msg it receives or sends. The type of the msg is defined by the option with tag 53 i.e included in the DHCP packet.

option with tag 53

Tag	length	value
53	1	•

- 1 DHCP DISCOVER
- 2 DHCP OFFER
- DHCP REQUEST
- DHCP DECLINE
- DHCP ACK
- DHCP NACK
- DHCP RELEASE
- DHCP INFORM

Figure → DHCP client transition diagram



the client, it is more efficient if the address is supplied explicitly by the client in the request message.

Server name → 64 byte field that is optionally filled by the server in a reply packet. It contains a null terminated string consisting of the domain name of the server. If the server does not want to fill this field with data, the server must fill it with all 0's

Boot filename

This is 128 byte field that can be optionally filled by the server in reply packet. It contains a null-terminated string consisting of the full pathname of the boot file. The client can use this path to retrieve other booting information. If the server does not want to fill this field with data, server must fill it with all 0's.

options :

This is 64 byte field with dual purpose. It can carry either additional information (network mask or default router address) or some specific vendor information. This field is used only in a reply message, the server uses number, called a magic cookie in the format of an IP address with the value of 99.130.83.99. When the client finishes reading the msg, it looks for this magic cookie. If present, the next 60 bytes are options. An is composed of 3 field 1-byte tag field, 1 byte length field & variable length value field, the length field defines the length of the value field, not whole option.

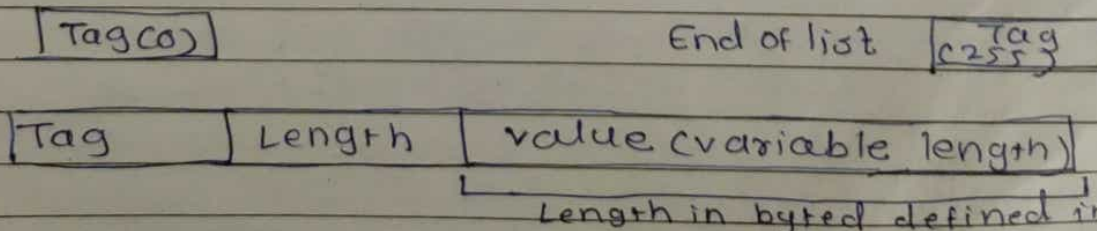


fig → option format

List of options

Tag	Length	value	Description
0			Padding
1	4	subnet mask	subnet mask
2	4	Time of day	Time offset
3	variable	IP address	Default router
4	1	IP —11—	Time server
5	1	IP 11	IEN 16 server
6	1	IP 11	DNS server
7	1	11	Log server
8	1	11	Quote server
9	1	11	Print server
10	1	11	Impress
11	1	11	RLP server
12	1	DNS Name	Host name
13	2	Integer	Boot file size
53	1	-	used for dynamic conf.
128-254	variable	specific info	vendor specific
255			End of list

The lengths of the field that contain IP addresses are multiples of 4 bytes. The padding option, which is only 1 byte long, is used only for alignment. The end of list option which is also only 1 byte long, indicates the end of the option field. vendors can use option tags 128 to 254 to supply extra info. In a reply msg.

1) INIT state

When DHCP client first starts, it is in the INIT state means initializing state. The client broadcasts a DHCPDISCOVER message using port 67.

2) SELECTING state

After sending the DHCPDISCOVER message, the client goes to the selecting state those servers that can provide this type of service respond with a DHCPOFFER message. In these messages, the servers offer an IP address. They can also offer the lease duration. The default is 1 hour. The server that sends a DHCPOFFER locks the offered IP address so that it is not available to any other clients. The client chooses one of the offers & sends a DHCPREQUEST message to the selected server. It then goes to the requesting state. However, if the client receives no DHCPOFFER message, it tries four more times each with a span of 2 seconds. If there is no reply to any of these DHCPDISCOVERs, the client sleeps for 5 minutes before trying again.

REQUESTING state

The client remains in the requesting state until it receives a DHCP PACK message from the server that creates the binding between the client physical address & it's IP address. After receipt of the DHCP ACK, the client goes to the bound state.

BOUND state

In this state, the client can use the IP address until lease expires, when 50% of lease period is reached the client sends another DHCPREQUEST to ask for renewal It then goes to the renewing state. When in the bound state, the client can also cancel the lease & go to the INIT state

RENEWING state

The client remains in the renewing state until one of two events happens. It can receive a DHCP ACK, which renews the lease agreement. client resets it's timer & goes back to the bound state. If DHCP ACK is not received the client & 87.5 % of lease time expires, the client goes to the rebinding state.

REBINDING State

The client remains in the rebinding state until one of 3 events happens. If the client receives a DHCPNACK or the lease expires, it goes back to the initializing state & tries to get another IP address. If the client receives DHCPACK, it goes to the bound state & reset timer.

Other Issues

Early Release

A DHCP client that has been assigned an address for a period of time release the address before the expiration time. The client may send a DHCPRELEASE message to tell the server that the address is no longer needed. This helps the server to assign the address to another client waiting for it.

Timers

The above discussion requires that the client uses three times: renewal timer, rebinding timer & expiration timer. If the server does not specify the time-out values for these timers when the address is allocated the client needs to use the default value.

value for timer is

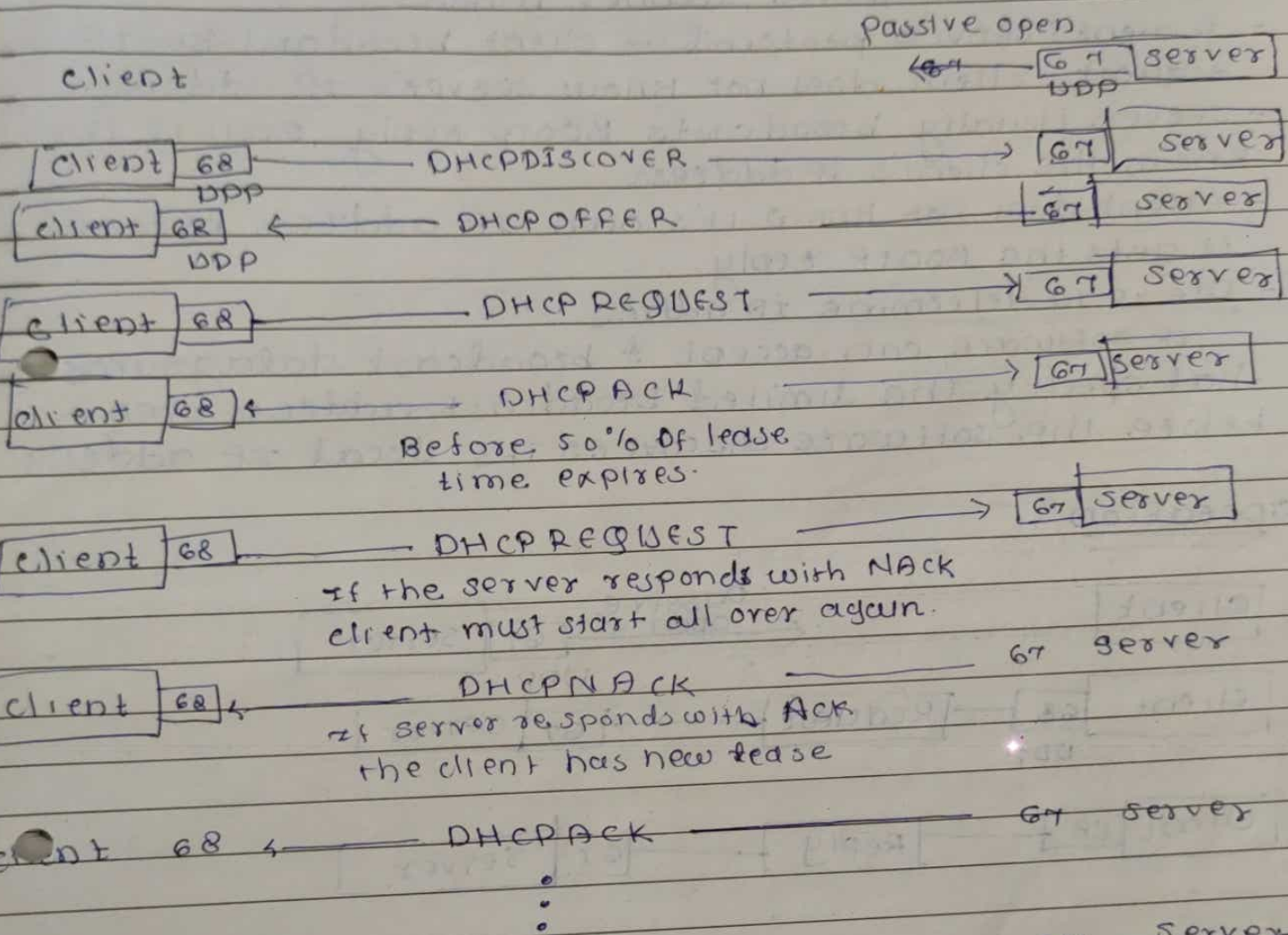
Renewal timer → 50% of lease time

Rebinding timer → 87.5%

Expiration timer → 100%

Exchanging Messages

Fig. shows exchange of messages related to the transition dia.

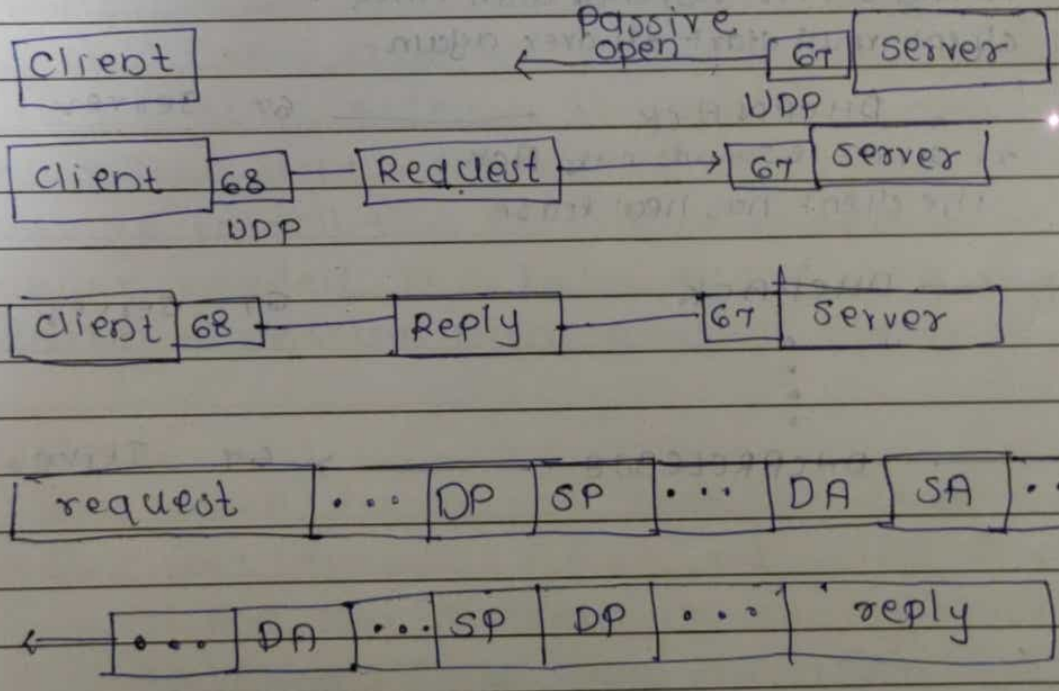


BOOTP (Bootstrap Protocol)

- It provides information about IP address of the requester, name server, subnet mask.
- A client / server protocol → client broadcast BOOTP request, client does not know server's IP address.
- server usually broadcasts BOOTP reply, even if it knows the client's IP address.
- client does not know it's own IP address before it gets the BOOTP reply.
- Use IP to determine IP address

IP software can accept & broadcast datagrams that specify the limited broadcast address even before the software discovers it's local IP address.

Operation



To.

Transmission policy

- Require UDP to use checksums
- Request & reply should be sent with the do not fragment bit set
- Multiple replies are allowed, the 1st reply is accepted & processed
- client is responsible for communication reliability
- Handles datagram loss with conventional technique of timeout & retransmission
- Random retransmission timer is recommended
- Double timer after each retransmission before it reaches 60s

Relay agent

- Use a remote BOOTP server to serve several LANs
- Broadcast address can not reach the server outside of the local network
- A relay agent is a router that can help send local requests to remote servers.
- when the relay agent receives broadcast request from a client, it forwards it to the remote server.
- The remote server sends the reply to the relay agent which is then forwarded to the client.

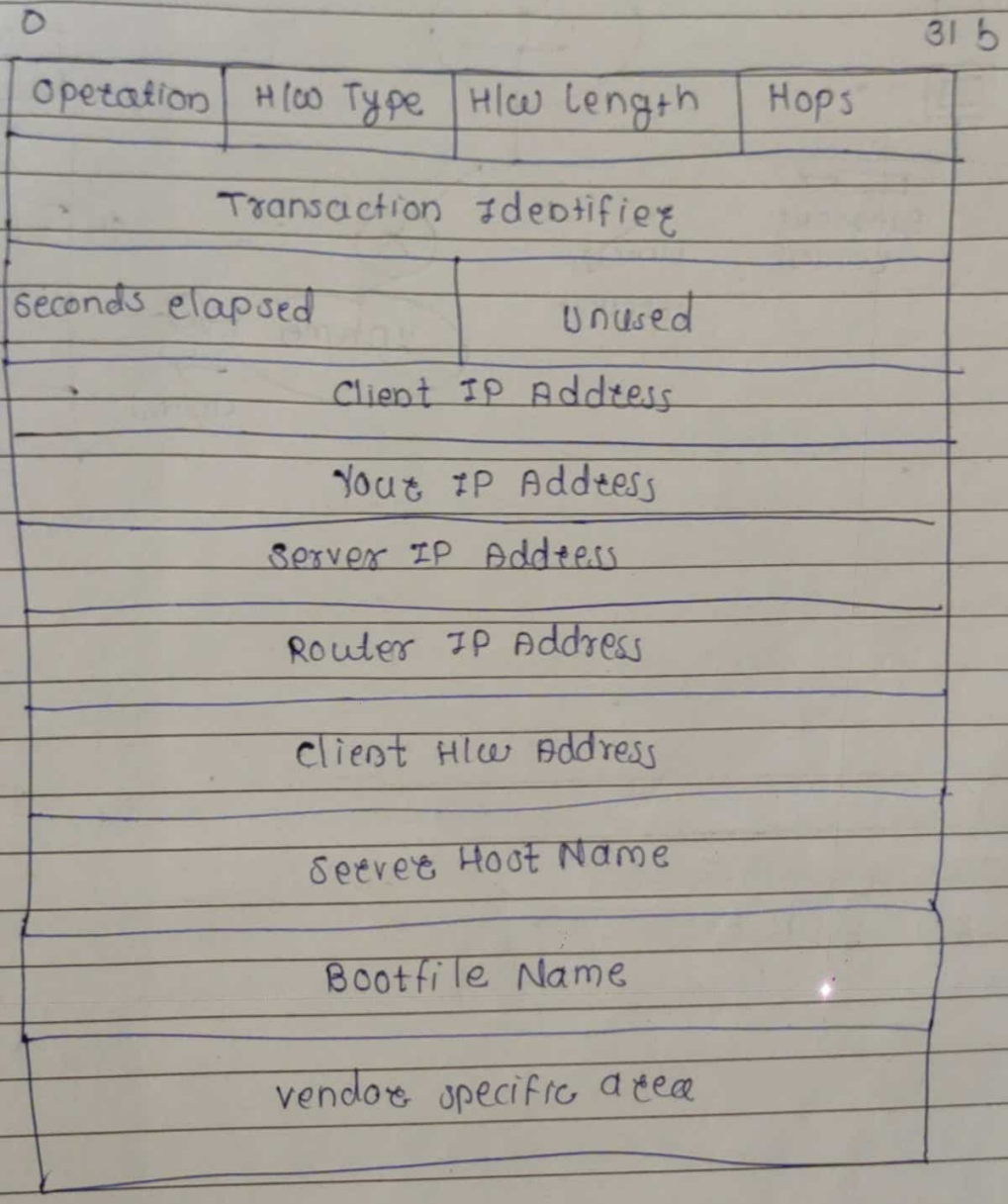
Message format

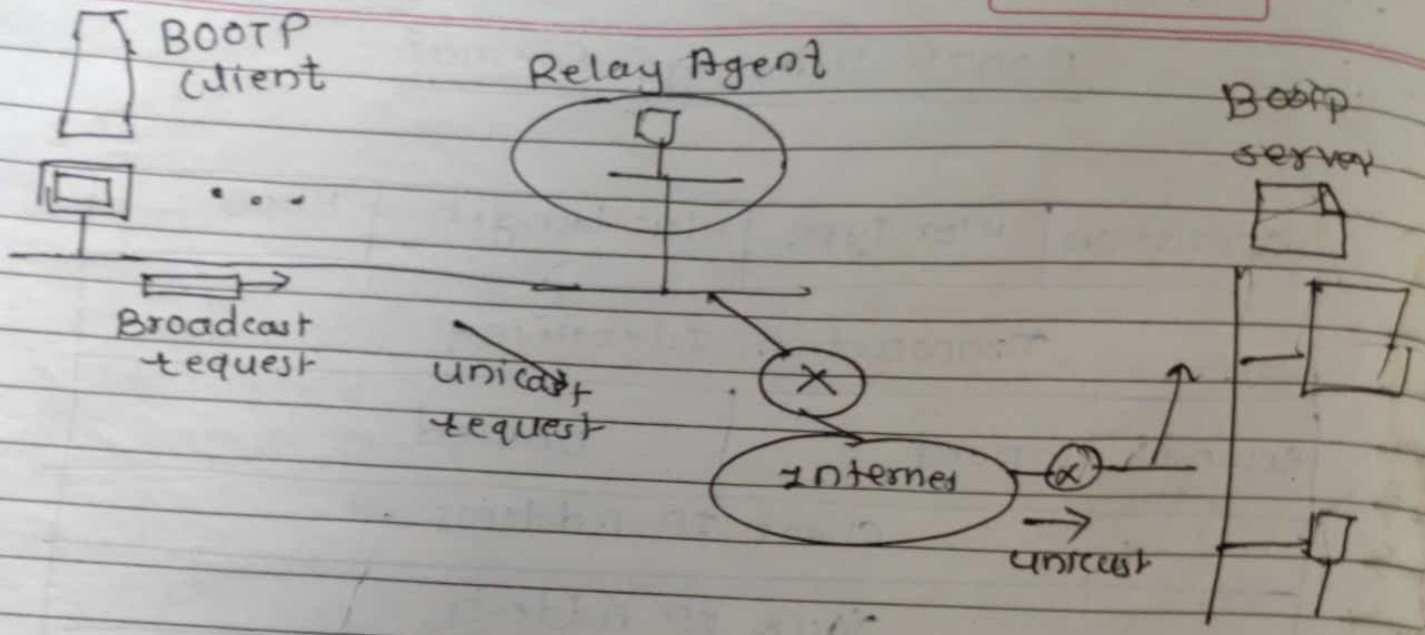
- All fields have fixed length
- To keep implementation small enough to fit in ROM
- Request & reply have the same format
- client fills in as much information as it knows & leaves remaining fields to zero.
- OP → operation code, request → 1, reply → 2
- HYPE → hardware type, ethernet = 1
- HLEN → hardware address length, for ethernet HA:6

Hops - maximum # of hops the message can travel →
Transaction ID → an integer to match reply with request
Seconds → # of seconds since the client started to boot
- client IP address & your IP address
If client knows it's IP address, place it in the client IP address field.
Boot file name - Generic name like "Unix" in the request.
Full name in response
vendor specific area : misnomer. Also used for general purpose information.
magic cookie : First four octets = 99.130.48.44
type length value

Item	code	Length
Padding	0	-
subnet mask	1	4
Time of day	2	4
End	255	-

BOOTP Message Format




BOOTP
Client
Relay Agent
Bootp
server
Broadcast
request
unicast
request
Internet
unicast

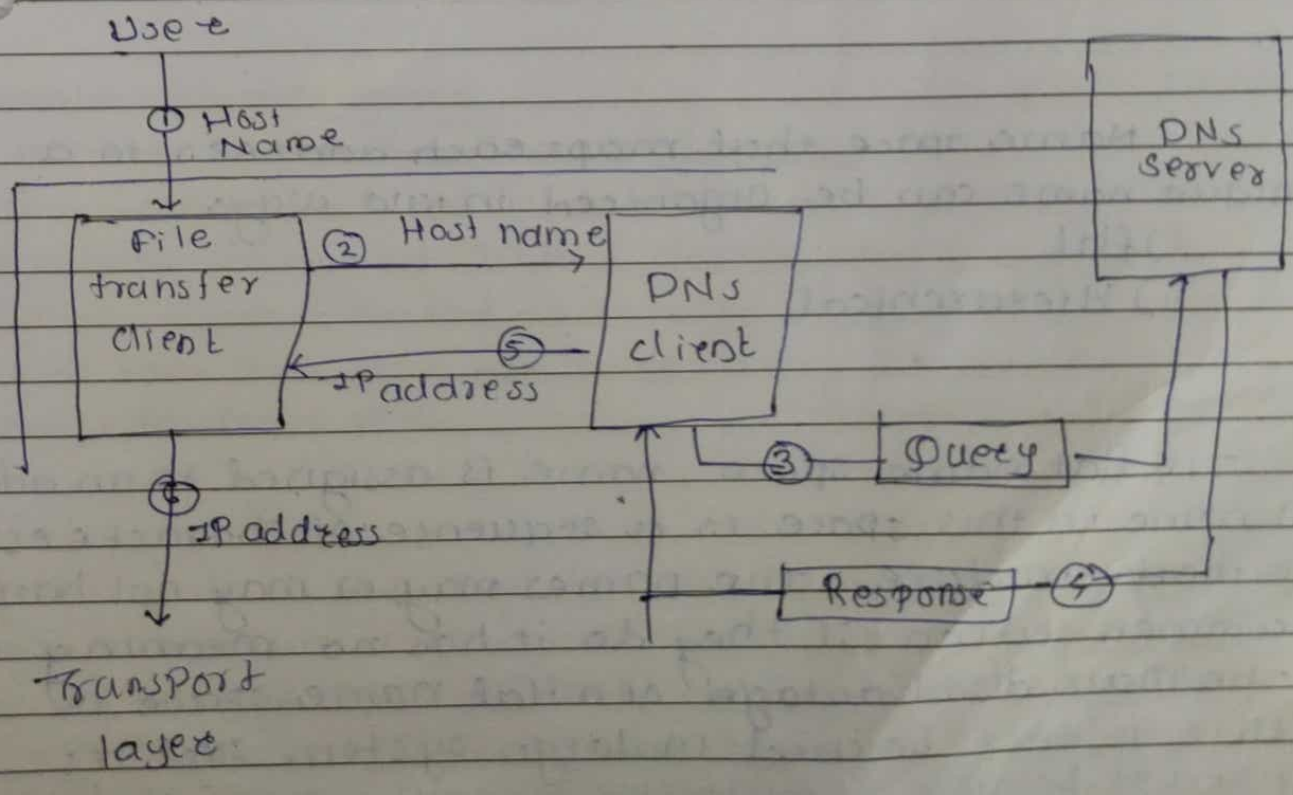
DNS - Domain Name Space

Need for DNS

When the Internet was small, mapping was done using a host file. The host file had only two columns, name & address. Every host could store the host file on disk & update it periodically from a master host file. When a program or user wanted to map a name to an address, the host consulted the host file & found the mapping. The host file would be too large to store in every host, it would be impossible to have update the host files every time there is a change

Solution is, the one used today, is to divide this huge amount of information into smaller parts & store each part on a different computer. In this method, host that needs mapping can contact the closest computer holding the needed info. This method is used by the DNS.

Figure shows how TCP/IP uses a DNS client & DNS server to map name to an address, the reverse mapping is similar



In figure user wants to use file transfer client to access the corresponding file transfer server running on a remote host. The user knows only the file transfer server name such as forouzan.com. However the TCP/IP suite needs the address of the file transfer server to make the connection. Six steps map the host name to an IP address.

1) The user passes the host name to the file transfer client.
2) The file transfer client passes the host name to the DNS client.
3) Each computer, after beeing booted, knows the address of one DNS server. The DNS client sends a message to a DNS server with a query that gives the file transfer server name, using the known IP address of the DNS server.
4) The DNS server respond with the IP address of the desired file transfer server.
5) The DNS client passes the IP address of the desired file transfer server.
6) The file transfer client now uses the received IP address to access the file transfer server.

We need two connections, the 1st is for mapping the name to an IP address, the second is for transferring files.

Name space

Name space that maps each address to a unique name can be organized in two ways

i) flat
ii) Hierarchical

i) Flat name space

In flat name space, name is assigned to an address. A name in this space is a sequence of characters without structure. The names may or may not have a common section, if they do it has no meaning. The main disadvantage of a flat name space is that it can't be used in large system such as Internet because it must be centrally controlled to avoid ambiguity & duplication.

2) Hierarchical Name space

In hierarchical name space, each name is made of several parts. 1st part can define the nature of the organization, 2nd part can define name of organization 3rd part can define departments in the organization & so on. In this case, authority to assign & control name spaces can be decentralized. A central authority can assign the part of the name that defines the nature of org. & name of org. The responsibility of the rest of name can be given to the org. itself.

Example → Assume 2 colleges & company can call one of their comp. challenger. The 1st college is given a name by central authority such as fhda.edu, the 2nd college is given the name berkeley.edu & the company is given name smart.com. When each of these org. adds the nam challenger to the name they have already been given the end result is 3 distinguishable name: challenger.fhda.edu, challenger.berkely.edu & challenger.smart.com. The name are unique without need for assignment by a central authority. The central authority controls only part of name not the whole.

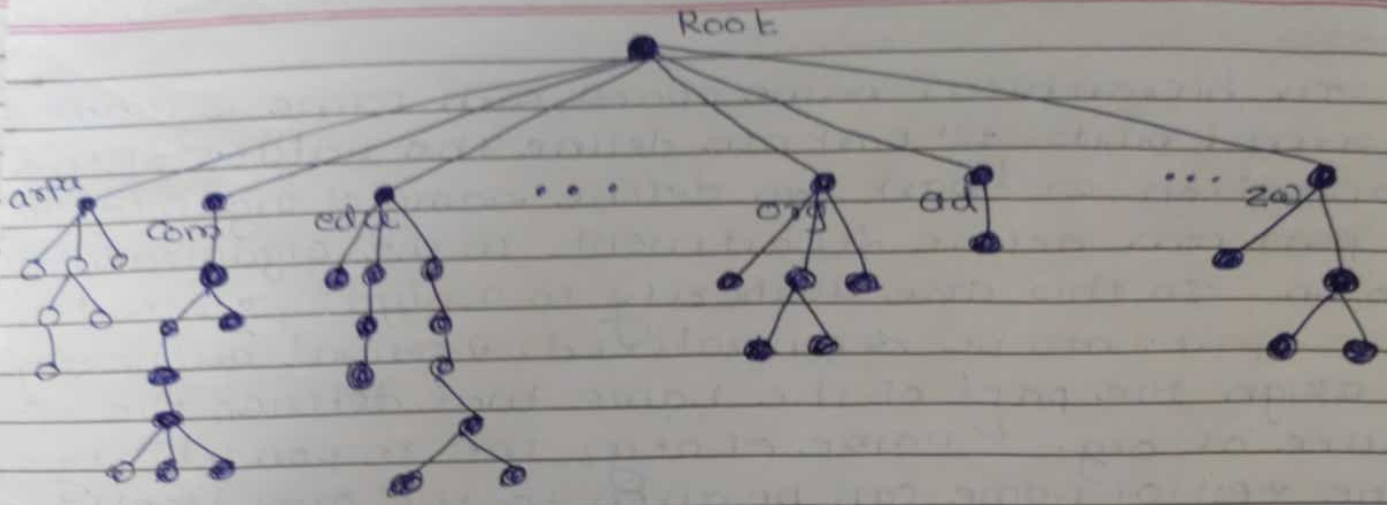
~~Domain Name space~~

~~To have hierarchical name space~~

To have hierarchical name space, a domain name space was designed. In this design the name are defined in an inverted-tree structure with the root at the top. The tree can have only 128 levels: level 0 (root) to level 127.

Label:

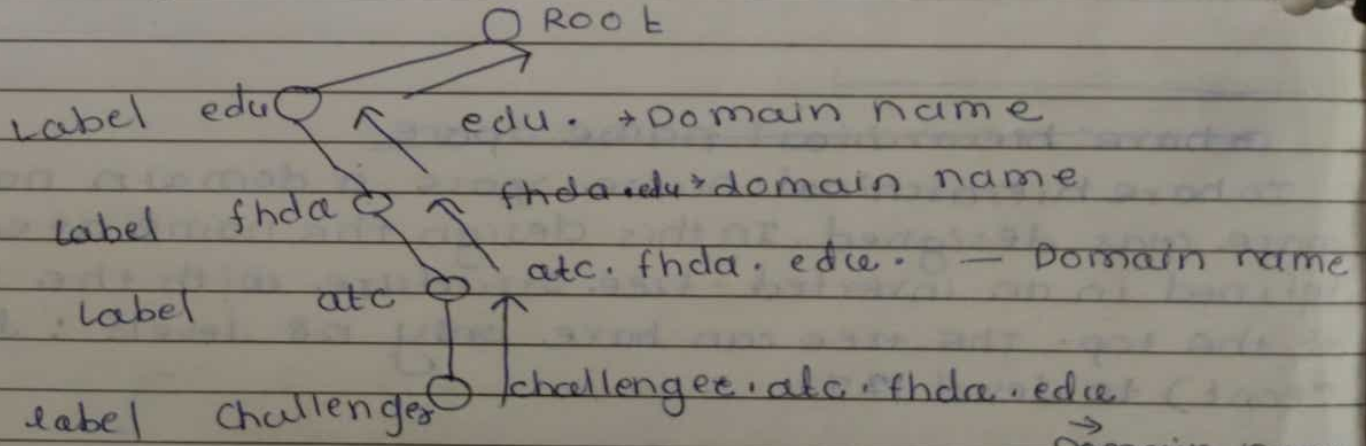
Each node in the tree has a label which is string with maximum of 63 characters. The root label is null (string (empty string) DNS requires that children of a node have different labels which guarantees the uniqueness of the domain names.



Domain Name

Each node in the tree has domain name. A full domain name is a sequence of labels separated by dot (.). The domain names are always read from the node up to the root. The last label is the label of the root (null). This means that a full domain name always ends in a null label, which means the last character is dot because the null string is nothing.

Fig → Domain names & labels



Fully Qualified Domain Name (FQDN)

If a label is terminated by null string, it is called fully qualified domain name (FQDN). It is domain name that contains full name of host. It contains all labels

from the most specific to the most general that uniquely define the name of the host

e.g → Domain name is FQDN of computer named challenger installed at ATC at fhda college so that the name must end with null label, label ends with a dot (.)

challenger.atc.fhda.edu.

Partially Qualified Domain Name

If label is not terminated by a null string it is called a partially qualified domain name (PQDN). A PQDN starts from a node but it does not reach the root. It is used when the name to be resolved belongs to the same site as the client. Here the resolver can supply the missing part called the suffix to create an FQDN.

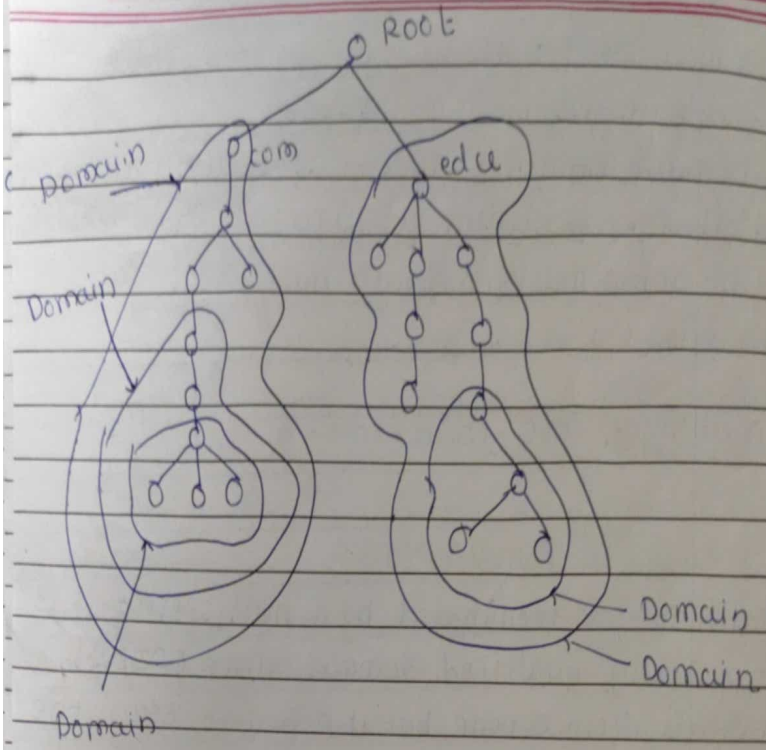
e.g → if user at fhda.edu site wants to get the IP address of the challenger computer, he can define partial name.

challenger.

FQDN	PQDN
challenger.atc.fhda.edu.	challenger.atc.fhda.edu
cs.hmme.com.	cs.hmme
www.funny.int.	www

Domain

A domain is a subtree of the domain name space. The name of the domain is the name of the node at the top of the subtree. figure shows domain. Domain may itself be divided into domains / subdomains.



Hierarchy of Name servers

To distribute the info among many computers called DNS servers. The root stand alone & create as many domains (subtree) as there are 1st level nodes. Because domain created this way could be very large. DNS allows domains to be divided further into smaller domains. Each server can be responsible for either large or small domain. We have hierarchy of servers in the same way that we have hierarchy of names.

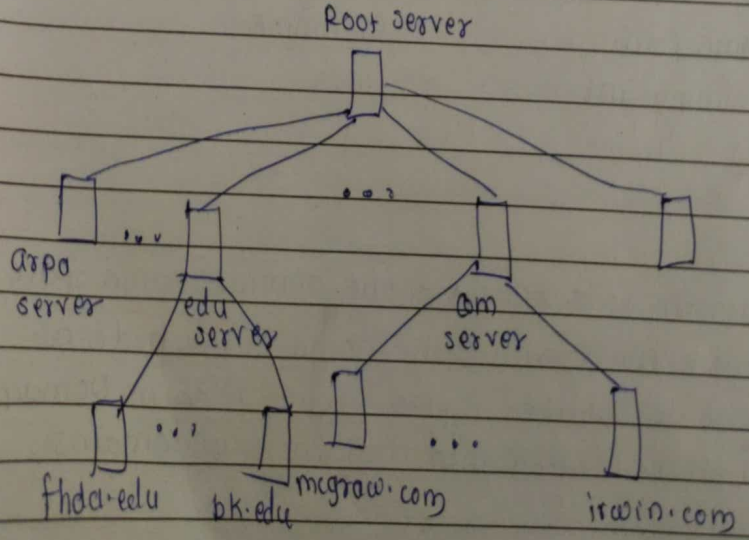


Fig:- Hierarchy of name servers

Zone

Since the complete domain name hierarchy can't be stored on single server, it is divided among many servers. What a server is responsible for or has authority over is called a zone. Define zone as contiguous part of the entire tree. If a server accepts the "domain" & "zone" refer to same thing. The server makes a database called a zone file & keeps all the info. for every node under that domain.

However if a server divides its domain into subdomains & delegates part of it's authority to other servers. domain & zone refer to different things. The info. about the nodes in the subdomains is stored in the servers at the lower levels, with original server keeping some sort of reference to these lower-level servers. The original server does not free itself from responsibility totally. It still has a zone, but the detailed info. is kept by the lower-level servers.

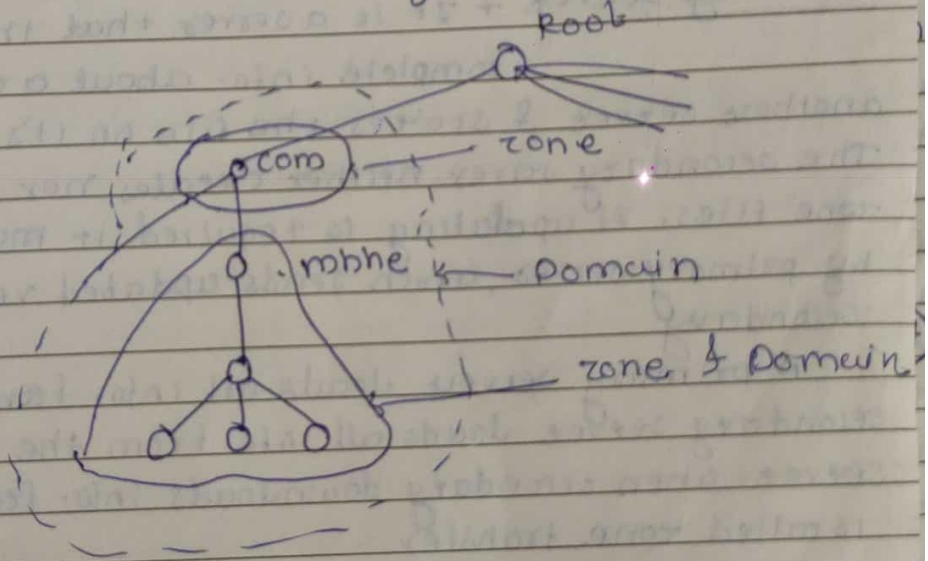


Figure - Domains & zones

24 DATE / /

Root server

A root server is a server whose zone consists of the whole tree. A root server usually does not store any info. about domains but delegates its authority to other servers, keeping references to those servers. There are several root servers, each covering the whole domain name space. The root servers are distributed all around the world.

Types of server

i) Primary server
ii) Secondary server

Primary server → Is a server that stores a file about the zone for which it is an authority. It is responsible for creating, maintaining & updating the zone file. It stores the zone file on local disk.

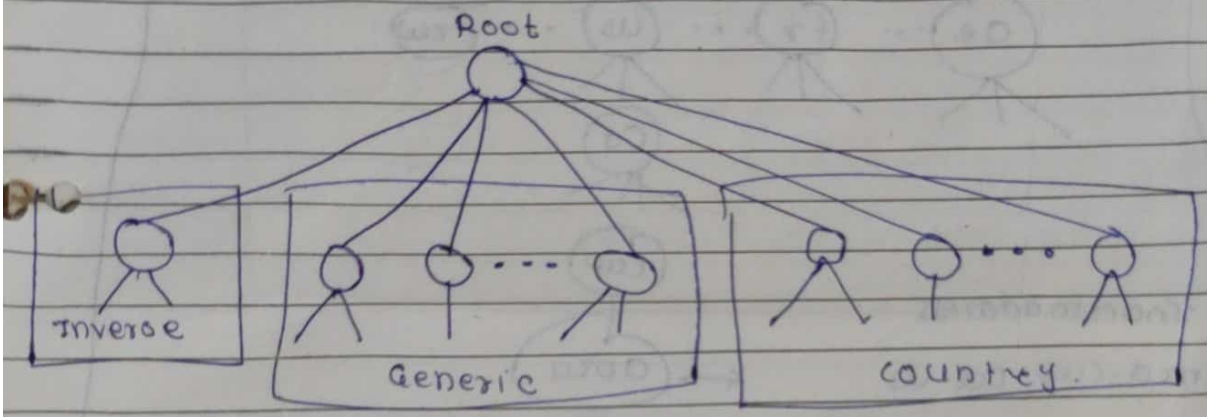
Secondary Server → It is a server that transfers the complete info. about a zone from another server & stores the file on it's local disk. The secondary server neither creates nor updates the zone files. If updating is required, it must be done by primary server, which sends updated version to secondary.

A primary server loads all info. from disk, the secondary server loads all info. from the primary server. When secondary downloads info. from primary is called zone transfer

DNS in the Internet

DNS is a protocol that can be used in different platforms.

i) generic Domains
ii) country Domains
iii) Inverse Domains.



① Generic Domains

It defines registered hosts according to their generic behavior. Each node in the tree defines domain, which is an index to the domain name space database

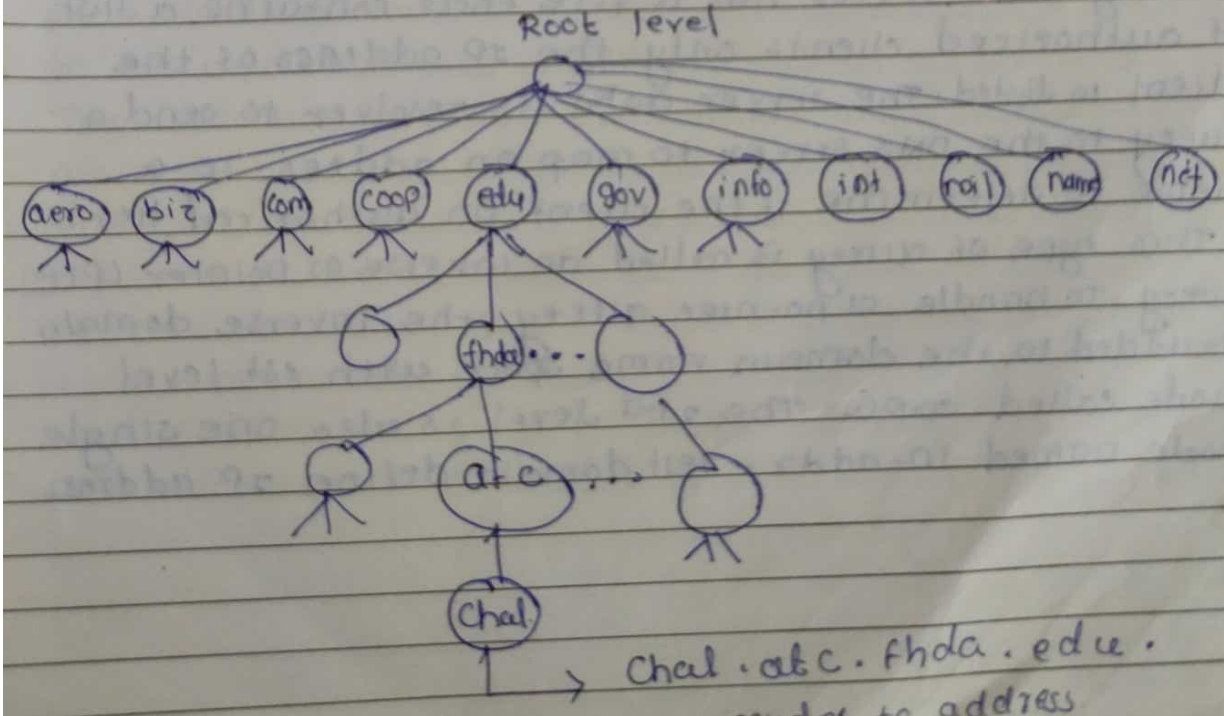
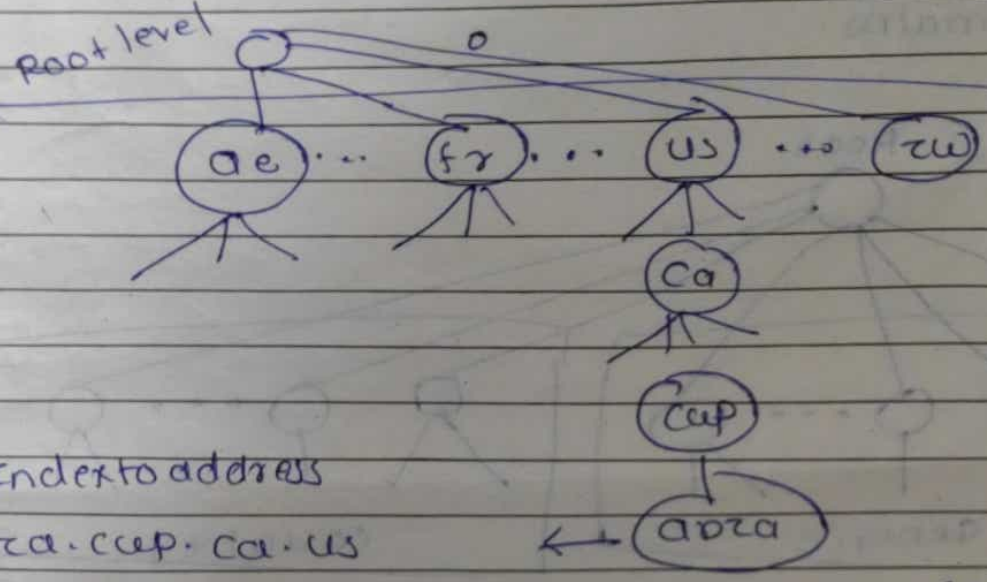


fig → Generic domain

→ Chal.atc.fhda.edu.
Index to address

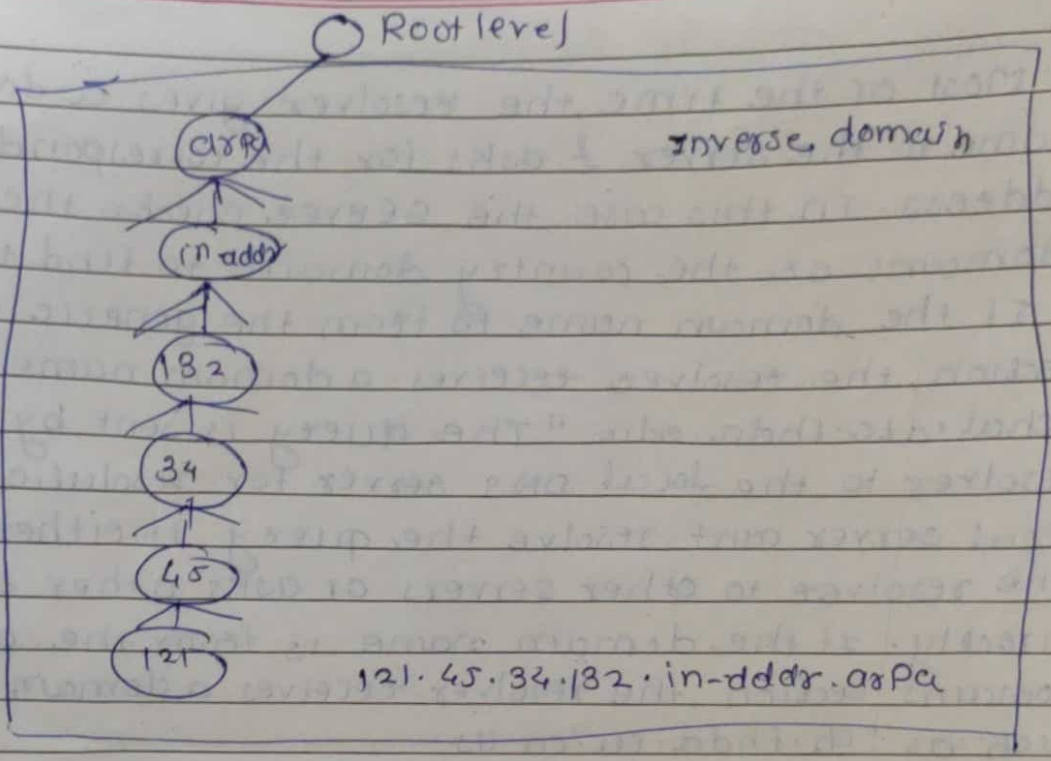
② Country Domains

The country domains section uses two country abbreviations. second labels can be organizational or they can be more specific national designations.



③ Inverse Domain

It is used to map an address to a name. when server has received a request from a client to do ask. Although the server has a file that contains a list of authorized clients, only the IP address of the client is listed. The server asks its resolver to send a query to the DNS server to map an address to a name to determine if the client on authorized list. This type of query is called an inverse or pointer (PTR) query. To handle a pointer query, the inverse domain is added to the domain name space with 1st level node called arpa. The 2nd level is also one single node named in-addr. rest domain defines IP address



Resolution

Mapping name to an address or an address to a name is called name-address resolution.

Resolver

DNS is designed as client server application. A host that needs to map an address to a name or a name to an address calls DNS client called a resolver. The resolver accesses the closest DNS server with a mapping request. If the server has the info. it satisfies the resolver, otherwise, it either refers the resolver to other servers or asks other severers to provide the info. After resolver receives the mapping it interprets the response to see if it is a real resolution or an error & finally delivers the result to the process that requested it.

Mapping Names to Addresses

Most of the time, the resolver gives a domain name to the server & asks for the corresponding address. In this case, the server checks the generic domains or the country domains to find the mapping.

If the domain name is from the generic domain section, the resolver receives a domain name such as "chal.atc.fhda.edu." The query is sent by the resolver to the local DNS server for resolution. If the local server can't resolve the query, it either refers the resolver to other servers or asks other servers directly. If the domain name is from the country domains section, the resolver receives a domain name such as "ch.fhda.cc.ca.us."

Mapping Addresses to Names

A client can send an IP address to a server to be mapped to domain name. However in the request, the IP address is reversed & two labels, in-addr & arpa are appended to create a domain acceptable by the inverse domain section. For example, if the resolver receives the IP address 132.34.65.121, the resolver 1st inverts the address & then adds the two labels before sending. The domain name sent is "121.89.34.132 in-addr.arpa", which is received by the local DNS & resolved.

Recursive Resolution

The client (resolver) can ask for a recursive answer from a name server. This means that the resolver expects the server to supply the final answer. If the server is the authority for the domain name, it checks its database & responds. If the server is not the authority, it sends the request to another server (parent usually) & waits for the response. If the parent is the authority it responds otherwise it sends query to yet another server

When the query is finally resolved, the response travels back until it finally reaches the requesting client.

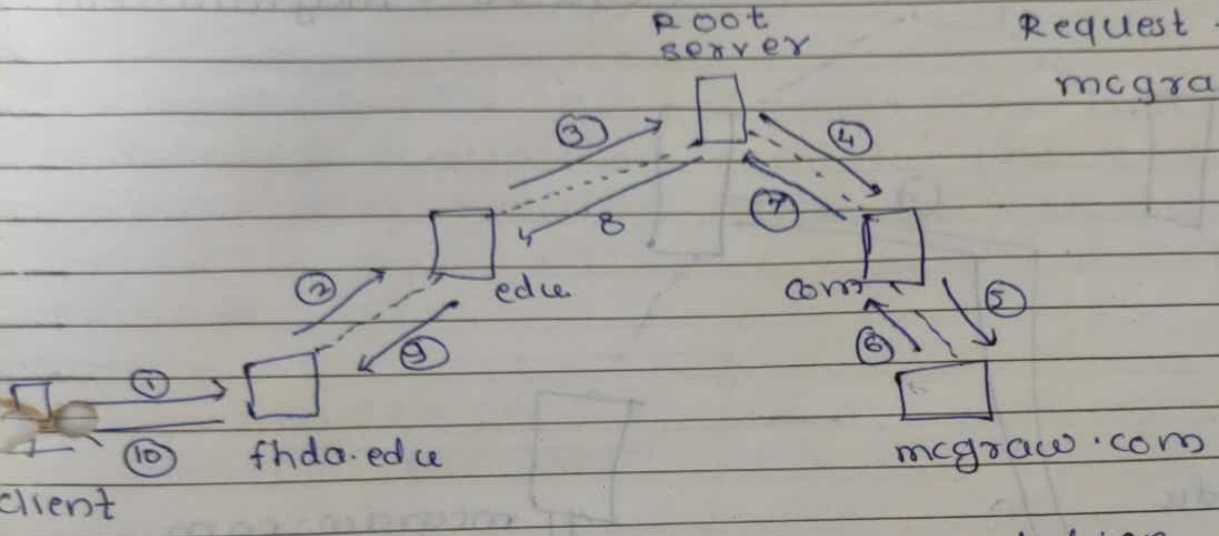


Figure → Recursive resolution.

Iterative Resolution

If the client does not ask for recursive answer, the mapping can be done iteratively. If the server is an authority for the name, it sends the answer. If it is not, it returns the IP address of the server that it thinks can resolve the query. The client is responsible for repeating the query to this second server. If the newly addressed server can resolve the problem, it answers the query with IP address; otherwise, it returns the IP address of new servers to the client. Now the client must repeat the query to the 3rd servers. In figure client queries five servers before it gets an answer from the mcgraw.com server.

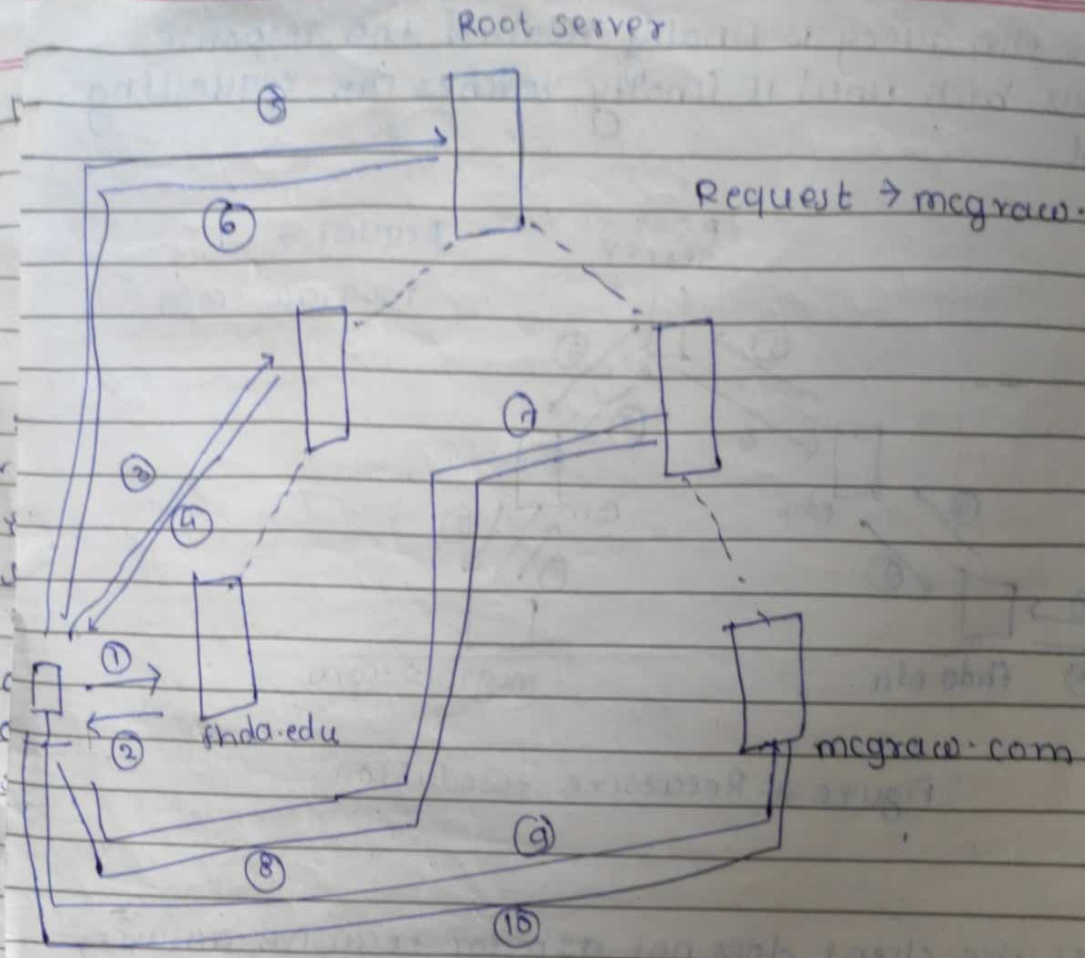


Figure → Iterative resolution

Caching

Each time server receives a query for a name that is not in its domain, it needs to search its database for a server IP address. Reduction of this search time would increase efficiency. DNS handles this with mechanism called caching. When server asks for a mapping from another server & receives the response, it stores this info. in it's cache memory before sending it to the client. If the same or another client asks for the same mapping, it can check its cache memory & resolve the problem. However to inform the client that the response is coming from the cache memory & not from an authoritative source, the server marks the response as unauthoritative.

DNS Messages

① Query
② Response

Both have the same format. The query msg consist of header & question records, the response message consist of a header, question records, answer records, authoratative records & additional records.

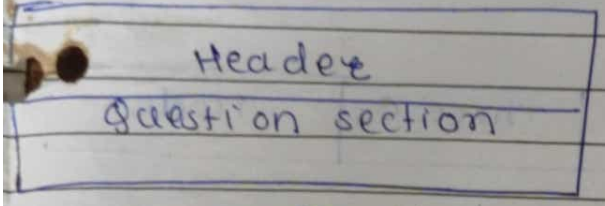


fig ① Query

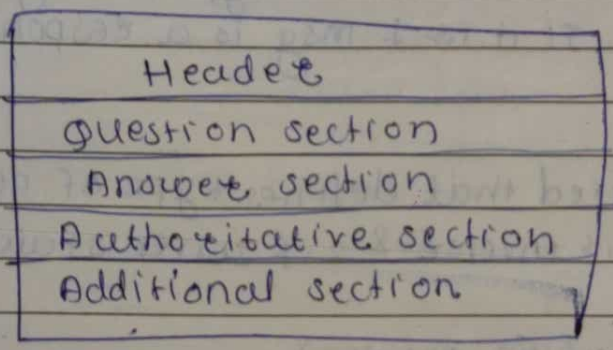


fig. ② Response

Header :

Both query & response messages have the same header format with some fields set to zero for the query messages. The header is 12 bytes & its format shown in figure.

Identification	Flags
No. of question records	No. of ans records (All 0's in query msg)
No. of Authoratative records (All 0s in query msg	No. of additional records (All 0s in query message)

Identification →
This is 16-bit field used by the client to match the response with the query. The client uses a different identification no. each time it sends a query. The server duplicates this no. in the corresponding response.)

Flags →
16 bit field consisting of subfield shown in table

Flags field

QR	opcode	AA	TC	RD	RA	Three 0s	rcode

QR (Query / Response)
1 bit subfield that defines type of msg. If it is 0 msg is query. If it is 1 msg is a response.

opcode
4 bit subfied that defines type of query or response 0 if standard, 1 if inverse & 2 if server status request

AA (Authoritative answer)
This is 1-bit subfield. When it is set it means that the name server is an authoritative server. It is used only in a response msg.

TC (Truncated)
1 bit subfield. When it is set it means that the response was more than 512 bytes & truncated to 512. It is used when DNS uses the services of UDP.

RD (recursion desired)
1 bit subfield when it is set (1), it means client desire a recursive answer. It is set in the query msg & repeated in the response msg.

if response otherwise 0 server

RA (recursion available) – This is 1 bit subfield. when it is set in the response, it means that a recursive response is available. It is set only in the response message.

Reserved – 3 bit subfield set to 000

rcode – 4 bit field that shows status of the error in the response.

values for rcode

value	Meaning	value	Meaning
0	No error	4	Query type not supported
1	Format error	5	Administratively prohibited
2	Problem at name server	6-15	Reserved
3	Domain reference problem		

No. of question Records
This is 16 bit field containing the no of queries in the question section of the msg.

Number of answer records → This is 16 bit field containing the no. of answer records in the answer section of the response msg. Its value is zero in query message.

Number of authoritive records → This is 16 bit field containing the no. of authoritative records in the authoritative section of response msg. It's value is zero in the query message.

Number of additional records → This is 16 bit field containing the no. of additional records in the additional records in the additional section of a response message. It's value is zero in the query message.

19 APR	20 APR
3, 20	3, 20
39, 38	39, 38
24, x	24, 35
21, 9	1, 21, 9
18, 19, 25	18, 25, 19
37, 16, 33, 36	16, 33, 13
26, 30	36, 37, 26
17, 15, 7	30, 11, 17
	10, 7, 8

17/4/21

Types of Records

1) Question Record
2) Resource Record

1) Question Records

i) A question record is used by the client to get information from a server.

ii) This contains the domain name

iii) Format of question record

Query name	
Query Type	Query class

Query name → This is a variable length field containing a domain name.
- The count field refers to the number of character in each section.

Query name format

count						count				count					count				count
5	a	d	m	i	n	3	a	t	c	4	f	h	d	a	3	e	d	u	0

Query type - This is a 16 bit field defining the type of query.

1	A	Address
2	NS	Name server
5	CNAME	Canonical Name
6	SOA	start of authority
11	WKS	well known service
12	PTR	Pointer
13	HINFO	Host Info.
15	MX	Mail Exchange
28	AAAA	Address
252	AXFR	Request for transfer of entire zone.
255	ANY	Request for all Records

Page No :
Date :

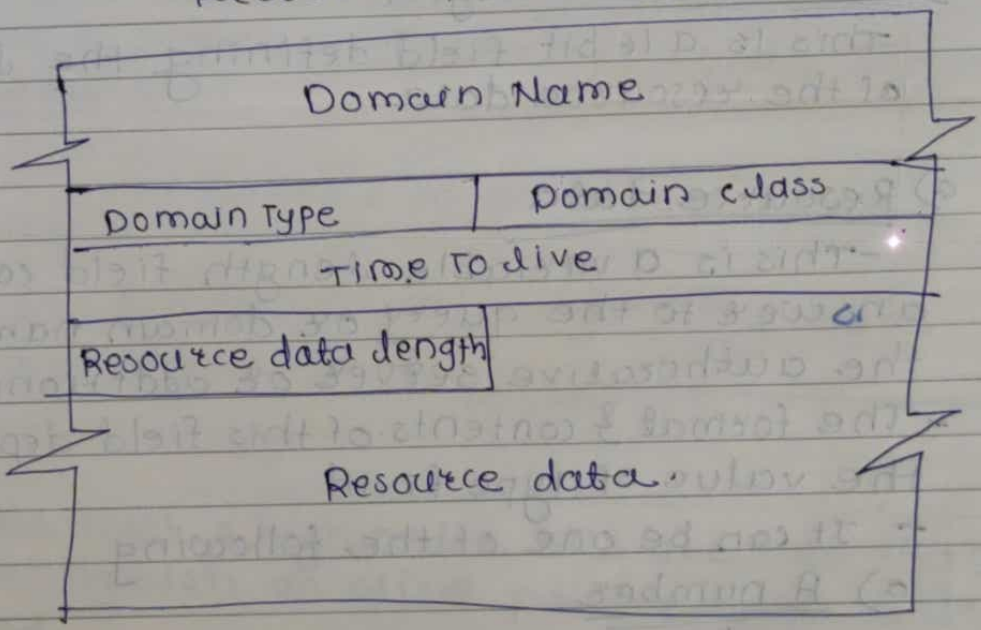
Query class → 16 bit field defining the specific protocol using DNS.

class	Mnemonic	Description
1	IN	Internet
2	CSNET	CSNET Network (obsolete)
3	CS	The CHAS network
4	HS	The Hesiod server

2) Resource Record

- i) Each domain name is associated with a record called the resource record
- ii) The server Database consists of resource records.
- iii) Resource records are also what is returned by the server to the client.

Resource Record format



i) Domain Name

- This is variable length field containing the domain name.
- It is a duplication of the domain name in the question record.
- since DNS ~~records~~ requires the use of compression everywhere a name is repeated, this field is a pointer offset to the corresponding domain name field in the question record.

2) Domain Name Type

1) This field is same as the query type field in the question record except the last two types are not allowed.

3) Domain class

1) This field is the same as the query class field in the question record.

4) Time to live

- This is 32-bit field that defines the no. of seconds the answer is valid.
- The receiver can cache the answer for this period of time. A zero value means that the resource record is used only in a single transaction & is not cached.

5) Resource data length

This is a 16 bit field defining the length of the resource data.

6) Resource data

- This is a variable length field containing answer to the query or domain name of the authorative server or additional info.
- The format & contents of this field depend on the value of type field.
- It can be one of the following

a) A number

- This is written in octets.

e.g - IPV4 address is a 4 octet integer & IPV6 address is 16 octet integer.

b) A domain name

- Domain names are expressed as sequence of labels.
- Each labled is preceded by 1-byte length field that defines the no. of character in label

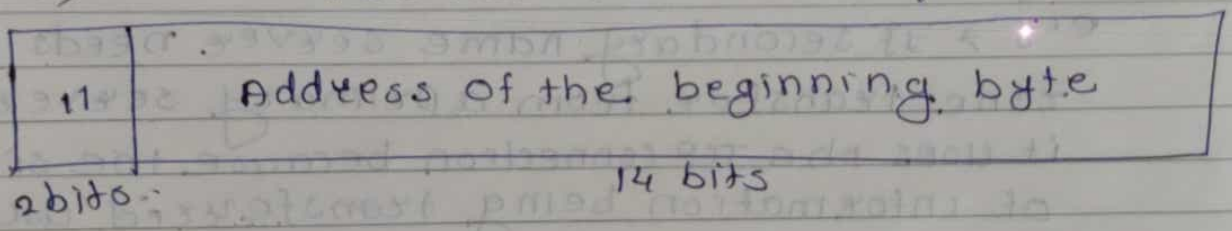
c) An offset pointer
- Domain name can be replaced with an offset pointer is a 2 byte field with each of the 2 high order bits set to 1 (11)

d) character string
- 1 -byte length field followed by the no. of characters defined in the length field.

Compression

i) DNS requires that domain name be replaced by an offset pointer if it is repeated.
ii) e.g → In a resource record the domain name is usually a repetition of the domain name in the question record.
iii) DNS defines 2 byte offset pointer that points to a previous occurrence of the domain or part of it

iv) Format of an offset pointer



v) The first 2 high order bits are two 1s to distinguish an offset pointer from a length field.
vi) The other 14 bits represent a number that points to the corresponding byte no. in the message.
vii) The bytes in a message are counted from the beginning of the msg. with the 1st byte counted as byte 0.
e.g → if an offset pointer refers to byte 12 of the msg., value should be 1100000000001100

Here the leftmost bits define the field as an offset pointer & the other bits define the decimal number 12.

Encapsulation

i) DNS can use either UDP & TCP. In both cases the well known port used by the server is port 53

ii) UDP is used when the size of the response msg is less than 512 bytes because most UDP packages have a 512-byte packet size limit.

iii) If the size of response msg. is more than 512 bytes, TCP connection is used.

In that case one of two scenarios can occur

a) If the resolver has prior knowledge that the size of response msg. is more than 512 bytes it uses the TCP connection.

e.g → If secondary name server needs a zone transfer from a primary server, it uses the TCP connection because the size of information being transferred usually exceed 512 bytes.

b) If the resolver does not know the size of response msg. It can use the UDP port, However if the size of the response msg. is more than 512 bytes, the server truncates the message & turns on the TC bit. The resolver now opens a TCP connection & repeats the request to get full response from server.