



Domain Name System (DNS)



Motivation

- IP addresses hard to remember
- Meaningful names easier to use
 - Assign names to IP addresses
- Name resolution – map names to IP addresses when needed
- Namespace – set of all names
 - Flat
 - Hierarchical



Flat Namespace

- Each host given a name
- Special file to keep name-address mapping (ex. /etc/hosts file in Linux)
- All hosts must know the current mapping for all other hosts with which they want to communicate
- Central authority to maintain authoritative host file with which all other hosts sync (ex. HOSTS.TXT at NIC in the old days)
- Makes the hostname file too large and the entire scheme unmanageable and impractical in any large network (ex., Internet)



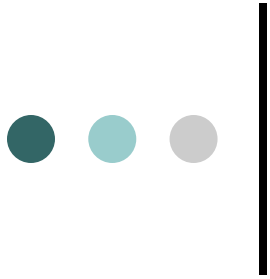
Hierarchical Namespace

- Break complete namespace into *domains*
- Domains broken up recursively into *subdomains* to create any level of hierarchy
- Delegate task of name allocation / resolution
 - Name allocation for any subdomain left to subdomain authority
 - Name resolution done by name server for subdomain



DNS

- Naming system for the internet
- Specifies
 - A hierarchical naming scheme
 - Name resolution mechanism
- Can handle multiple object types within one system
 - “Type” associated with each name to distinguish different types of entities
 - Ex. the name “cse.iitkgp.ernet.in” can be a domain name, a simple host name, an email server name etc.



- RFC 1034/1035, many other related ones (See <http://www.dns.net/dnsrd/rfc/> for a list of DNS related RFCs (incl. ones that updates 1034/1035))
- DNS-relevant important websites
 - www.iana.org
 - www.icann.org
 - www.internic.net

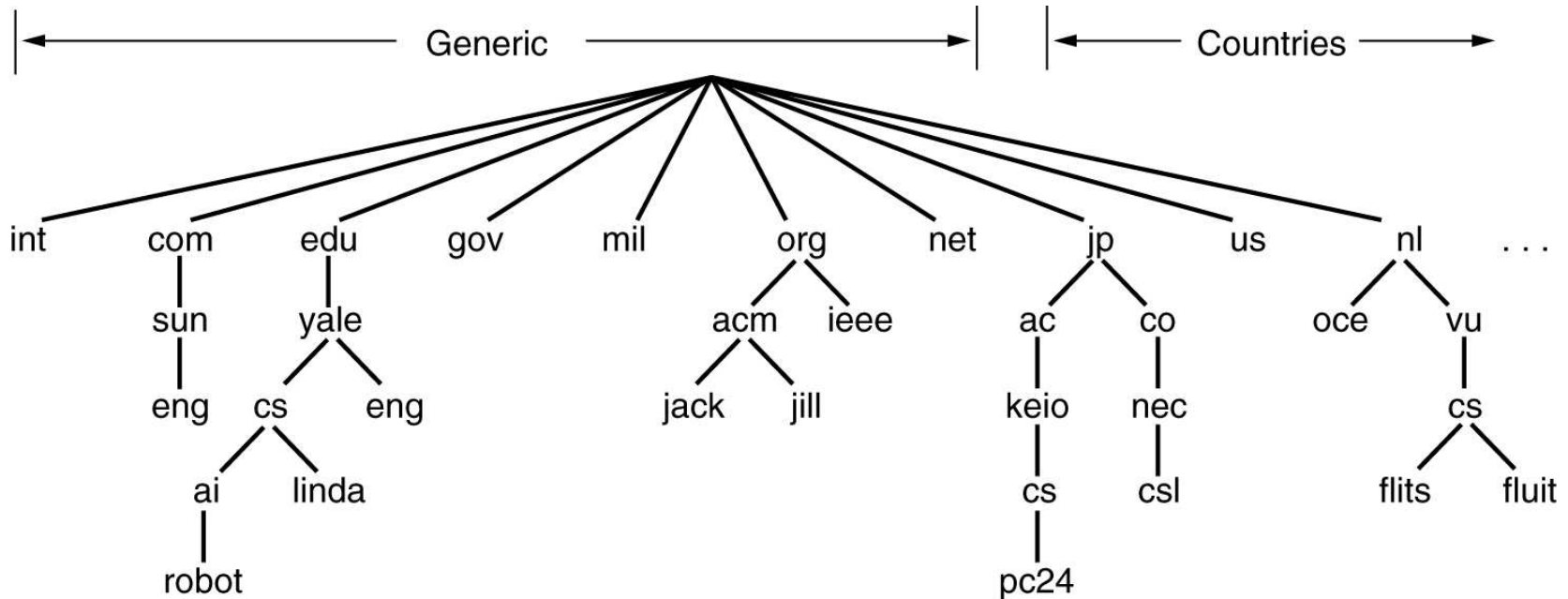


DNS Namespace

- Complete namespace is a tree of domains
- Root is a special domain (no name)
- Top level domains – domains at second level of tree
 - *com, edu, gov, net, mil, int, org, arpa, in*, country specific domains (*us, in, kr* etc.)
 - Managed by delegated authorities
 - Ex. for [.in](#) domain, NCST/National Internet Exchange of India
- Domains from third level
 - Managed by local authorities



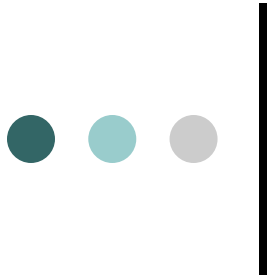
DNS Namespace (contd)





DNS Names

- Every node in the tree has a label (max 63 bytes, case insensitive)
- Sibling nodes must have different labels
- DNS name of a node = sequence of labels from that node to root, separated by ‘.’
- Absolute names – names that end with ‘.’
- Relative names – names that does not end with ‘.’, meaning they will be completed by appending something
- Nodes can be domains or hosts
- Arbitrary hierarchy allowed (but implementations usually limit name length to 255 bytes)



- Domain : subtree of the DNS namespace tree
- Zone : part of the tree for which the naming authority has been delegated to some name server
- Domain $x.y$ and Zone $x.y$ may not be same, as part of $x.y$ domain may have its own naming authority and is not part of $x.y$ zone



Name Servers

- Contains mapping information for one or more zones (*zone files* - text files in standard format)
- Maps names to IPs (*forward lookup*, mandatory) or IPs to names (*reverse lookup*, optional)
- Primary/Master name server : gets mapping data for zone from zone file on the host it runs on
- Secondary name server: pulls zone file data from primary name server (*zone transfer*)
- Authoritative server for a zone: either a primary or secondary server for that zone
- A host can be primary for some zones and secondary for others at the same time

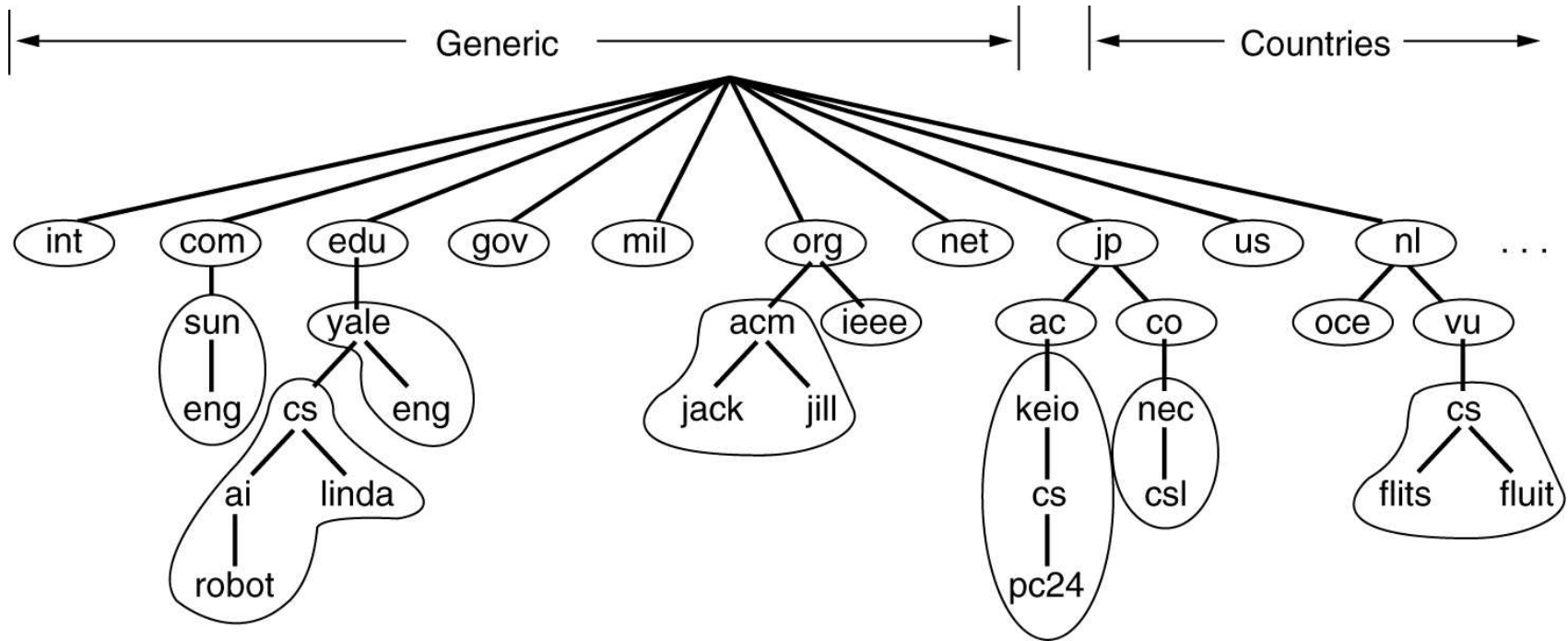


Root Servers

- Name servers for root zone
- Contains name server for all top level domains
- Currently 13 root servers spread all over the world (all are secondaries of a hidden primary, *a.root-servers.net* through *m.root-servers.net*)
- All DNS name servers knows at least one root server



Name Servers





Domain Name Resolution: Overall Steps

- User program issues the request
- Query to name server is formulated
- The name server checks if name in database.
- If not, ask the higher level name server
- Finally the user program gets IP Address or error



Name Resolution

- Resolver
 - Accesses name server for name resolution
 - Knows the address of at least one name server
 - Sends a DNS request to the name server
 - Standard access routine: `gethostbyname()`
- Name server
 - Gets request from resolver
 - Looks up the name and sends back response



Name Resolution Basics

- Contact root server for name server of top level domain
- Name server for top level domain gives name server for next level domain
- Process continues until mapping is found or error



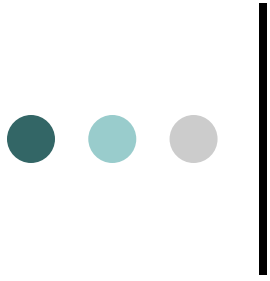
Example

- To resolve `www.yahoo.com`, first contact root server to get name server for *com*
- Querying name server for *com* gives name server for *yahoo.com*
- Querying name server for *yahoo.com* gives IP address of *www.yahoo.com*
- Three queries needed to resolve the name in the worst case



Recursive/Iterative Query

- Recursive Query
 - DNS server either gives the mapping, or forwards the request to the name server that may have it
 - Original requestor finally gets either the mapping or an error
 - May be ok for lower level domains with less request volumes
 - Not suitable for higher level domains with high request volumes



- Iterative
 - If DNS server does not have mapping, it gives the address of the name server that may have it (*referral*)
 - Original requestor contacts the new name server
 - Repeated until mapping is found or no referral is obtained (error)
- Servers must implement iterative query, may implement recursive query



Caching

- Caching employed at both client and server for efficiency
 - Lookup results in cache (both final IP address, or name server addresses for intermediate domains)
 - Refreshed at regular intervals
- Caching Name Servers: not authoritative for any zone, only caches entries for other zones



Resource Records (RR)

- Each zone file contains a set of resource records for that zone
- Each RR has: name, type, TTL, Rdata, plus some other fields
- RR Types (16 bit value):
 - SOA : Start of authority
 - NS : authoritative name server for the domain
 - A : hostname
 - MX : mail server
 - CNAME : alias name
 - HINFO : CPU and OS Info
 - PTR : pointer to another part of namespace
 - SRV : Service name (RFC 2782)
 - Others.....



- TTL : indicates how long the RR can be cached (32 bit integer in seconds)
- RDATA : a type specific value (for ex., an IP address for A type etc.)
- Some other fields in RR not of interest to us



\$TTL 3D

@ IN

SOA mc1.land-5.com. root.land-5.com. (

199609206 ; serial, todays date + todays serial #

8H ; zone file refresh period for secondaries

2H ; retry period for secondaries if primary is unreachable

4W ; expiry time if zone file cannot be refreshed

1D) ; minimum TTL of any RR

NS mc1.land-5.com.

NS ns2.psi.net.

MX 10 mailsrv.land-5.com. ; Primary Mail Exchanger

MX 20 backupmail.land5.com.

TXT "LAND-5 Corporation"

router A 206.6.177.1

mc1.land-5.com. A 206.6.177.2

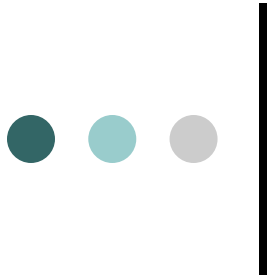
mc2.land-5.com. A 206.6.177.3

mailsrv A 206.6.177.4

ftp CNAME mc1.land-5.com.

news CNAME mc1.land-5.com.

funn A 206.6.177.2



**www CNAME mc1.land-5.com.
CNAME mc2.land-5.com.**

**telnet.tcp SRV 10 1 23 mc2.land-5.com.
SRV 10 3 23 mc1.land-5.com.**

**subdomain1.land-5.com. NS ns1.subdomain1.land-5.com.
subdomain2.land-5.com. NS ns2.subdomain2.land-5.com.**

**ns1.subdomain1 A 202.122.132.7
ns2.subdomain2 A 202.122.136.9**



Forwarders

- A DNS server X to which DNS queries can be sent by another DNS server Y if it cannot resolve it
- X resolves it and sends back the result to Y. X also caches.
- Motivation:
 - No internet connection for Y
 - Forwarder cache builds up over time
 - Forwarder may be able to resolve most queries
- X may or may not be authoritative for any zone
- Y does not need to know root servers

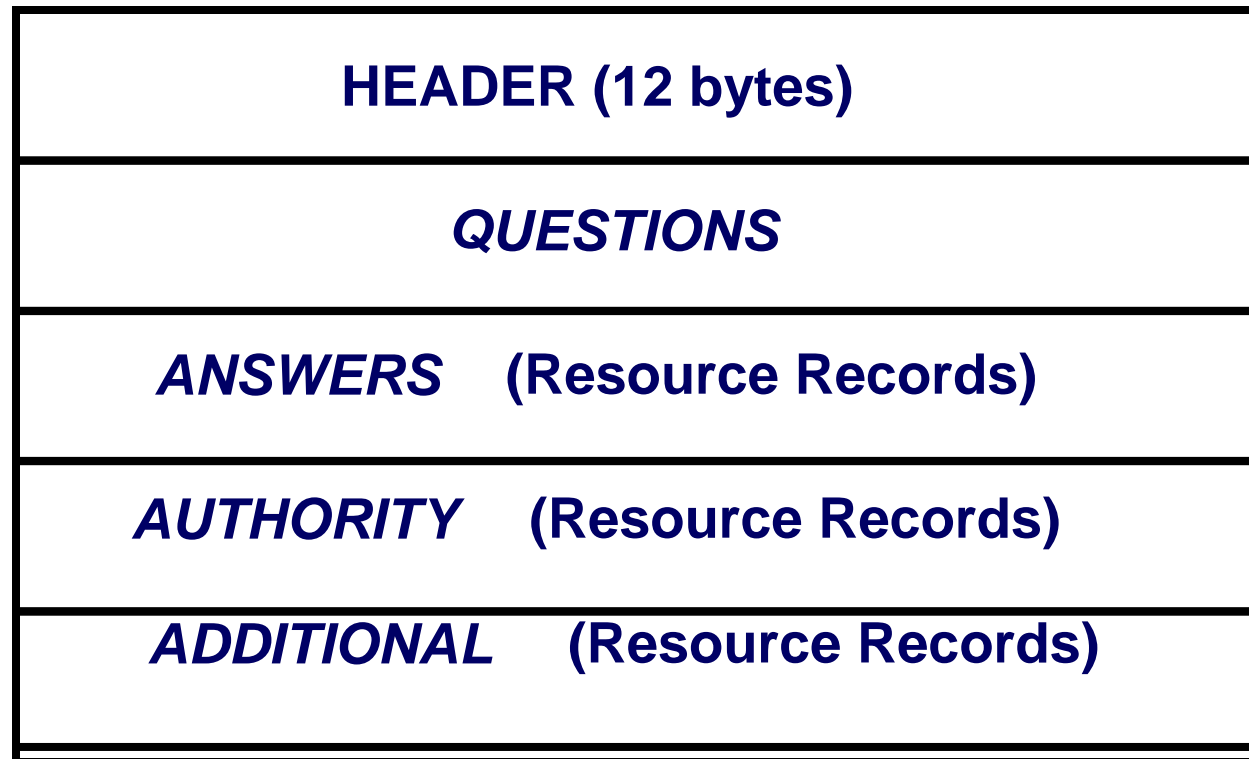


Protocol Details

- Usually runs on UDP port 53
- Uses TCP for zone transfers (and some large responses)
- TCP can also be used for normal operation, though not used normally
- Same message format for query and response



DNS Message Format





Header Format

Identification	Flags
# of questions	# of answer RRs
# of authority recs.	# of additional RRs

DNS Header FLAGS field



QR: 0 means message is query, 1 means response.

OPCODE: 0 is *standard query* (use 0).

AA: 1 means authoritative answer (set by server).

TC: 1 means response was truncated (set by server).

RD: 1 means recursion desired (set by client).

RA: 1 means recursion available (set by server).

000: must be three zero bits.

RCODE: return code. 0 is no error, 3 is name error, etc.

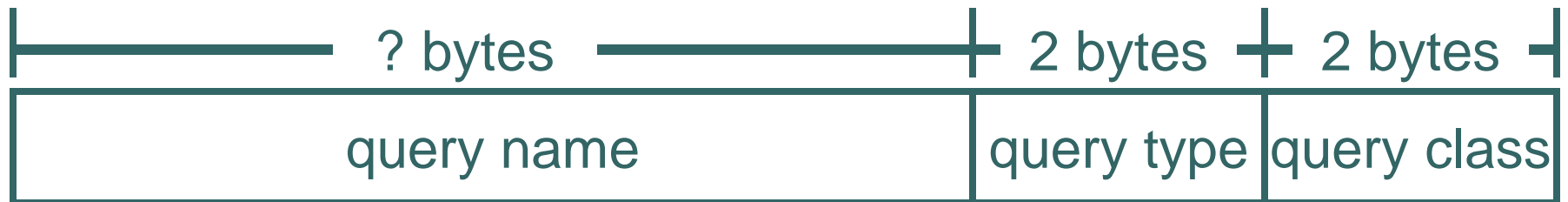


Question Format

- Each *question* includes a variable length *query name* that specifies a hostname
- Each question also includes
 - *query type* (what type of RRs are asked for, ex., A, SRV etc.)
 - *query class* is 1 for Internet Addresses



Question Format



Query Name is a sequence of one or more *labels*.

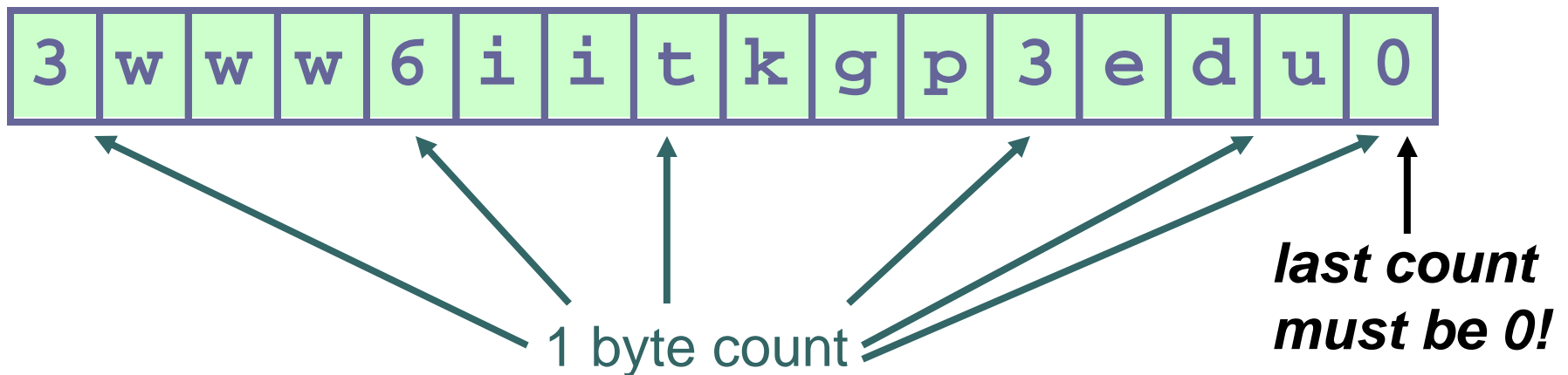
Each *label* is a single byte count, followed by that many characters.

The last label must have a count of 0.



Query Name Example

The name www.iitkgp.edu would be sent like this:



Each count byte is a binary value in the range 0-63
count bytes are not ASCII !



Some Query Type Field Values

A	1	IP Address
NS	2	Name Server
CNAME	5	Canonical Name
PTR	12	Pointer
HINFO	13	Host Info
MX	15	Mail Exchanger
ANY	255	<i>everything</i>

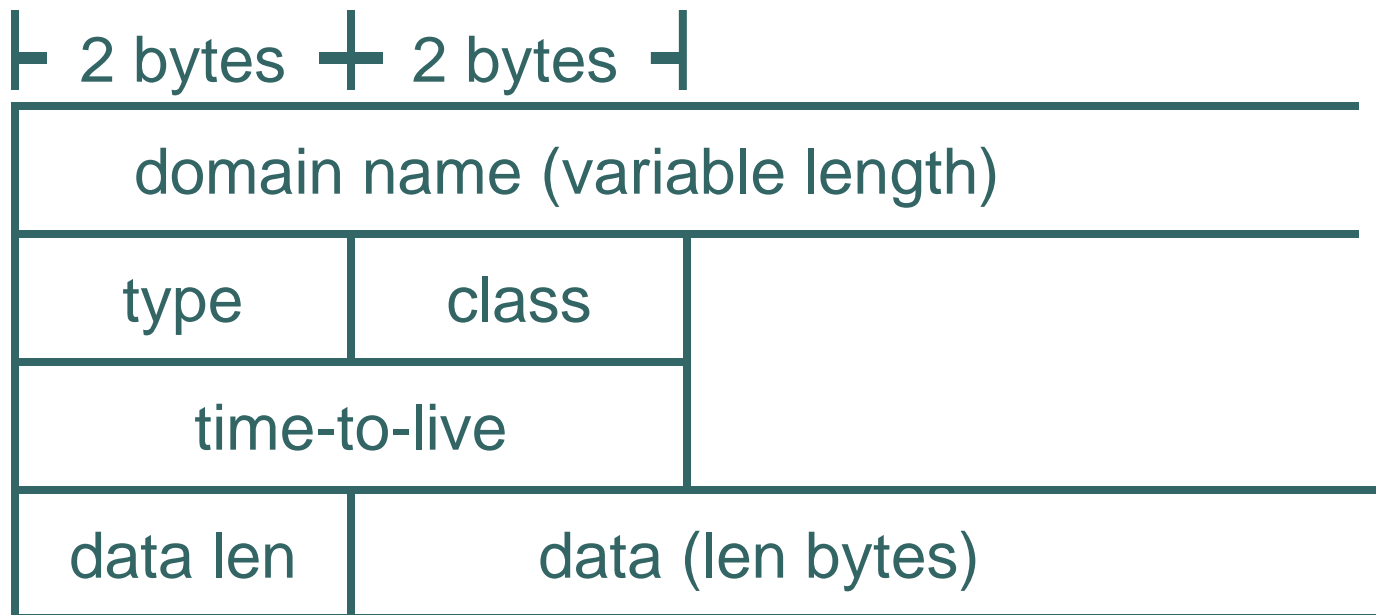


Answer Format

- The *answers, authority and additional information* parts of a response are all provided via the same format –a Resource Record (RR).
- Each Resource Record specifies the value of a single resource along with information about the resource (what kind it is, how long the information is valid, etc.)



Resource Record Format





Reverse Lookup

- IP to name mapping
- Not mandatory to implement, but most DNS servers support
- All IP addresses are part of the special zone in-addr.arpa
 - Ex. 10.5.17.2 will map to the name 2.17.5.10.in-addr.arpa
 - PTR type RR kept to map this to a name
 - Lookup similar otherwise



Dynamic Update

- Simple DNS requires the primary name server to be updated manually when a mapping changes – not good for working with protocols like DHCP
- Dynamic DNS updates allow dynamic updates to zone files by messages
- For more details, see RFC 2136