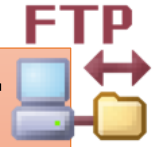
Application Layer

Application layer interfacing

Name Service (DNS)

Web (HTTP)

Email (SMTP, POP, IMAP)



File Transfer (FTP)

End to end packet delivery

UDP

Connection Establishment

Reliable Data Delivery

Flow and Congestion Control

Ordered Packet Delivery

TCP

Transport

Network

Data Link

Domain Name System

- Assign **Unique names** to an IP address (machine interface)

- ARPANET – a file *hosts.txt* listed all the computer names and their IP addresses.

• To map a name onto an IP address, an application program calls a library procedure called the resolver, passing it the name as a parameter.

• The resolver sends a query containing the name to a local DNS server, which looks up the name and returns a response containing the IP address to the resolver, which then returns it to the caller.

• The query and response messages are sent as UDP packets.
• Armed with the IP address, the program can then establish a TCP connection with the host or send it UDP packets.

DNS Services

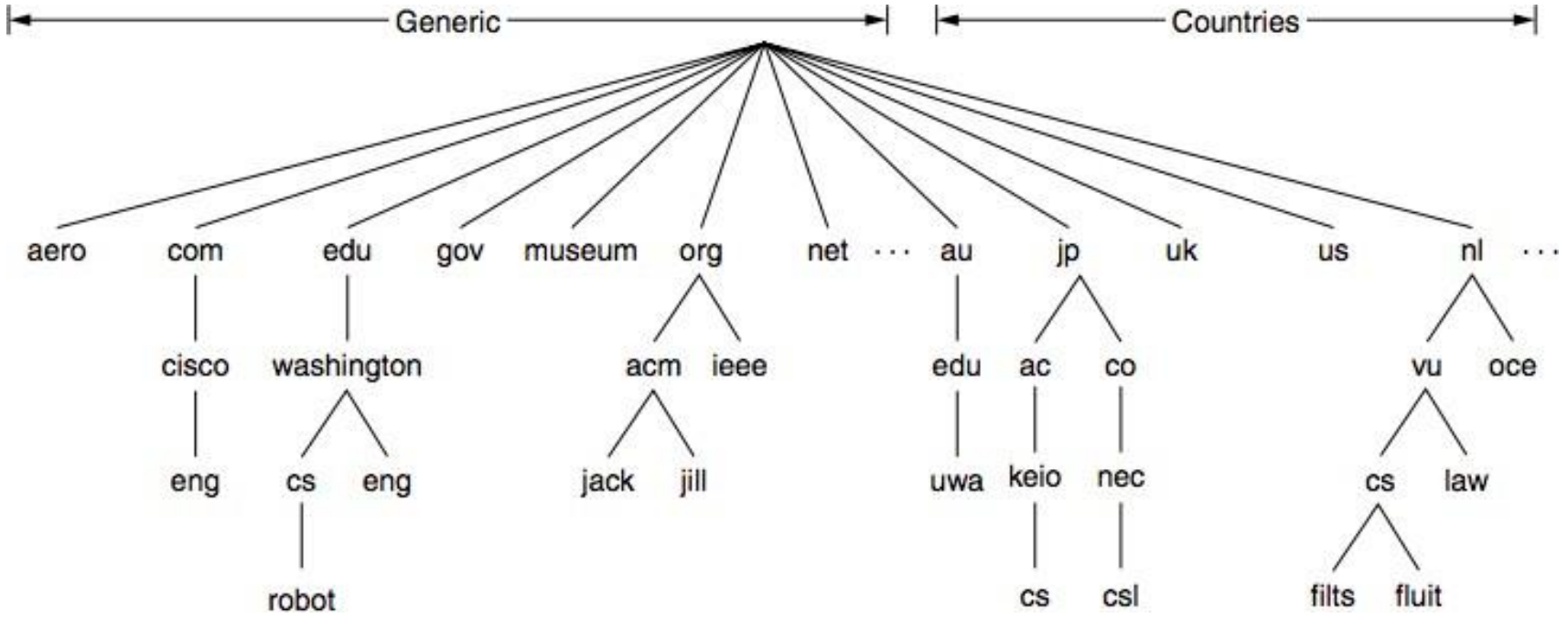
- Hostname to IP address translation
- Host aliasing
  - Canonical and alias names
- Mail server aliasing
- Load distribution
  - Replicated Web servers: set of IP addresses for one canonical name

Why not centralized DNS?

- single point of failure
- traffic volume
- distant centralized database
- maintenance

The DNS Name Space

The Domain Name Space refers to a hierarchy in the Internet naming structure.



The top level domains are run by **registrars** appointed by ICANN
Name registrar for India (.in domain): **registry.in** (National Internet Exchange of India – **NIXI**)

Elements of DNS (RFC1034)

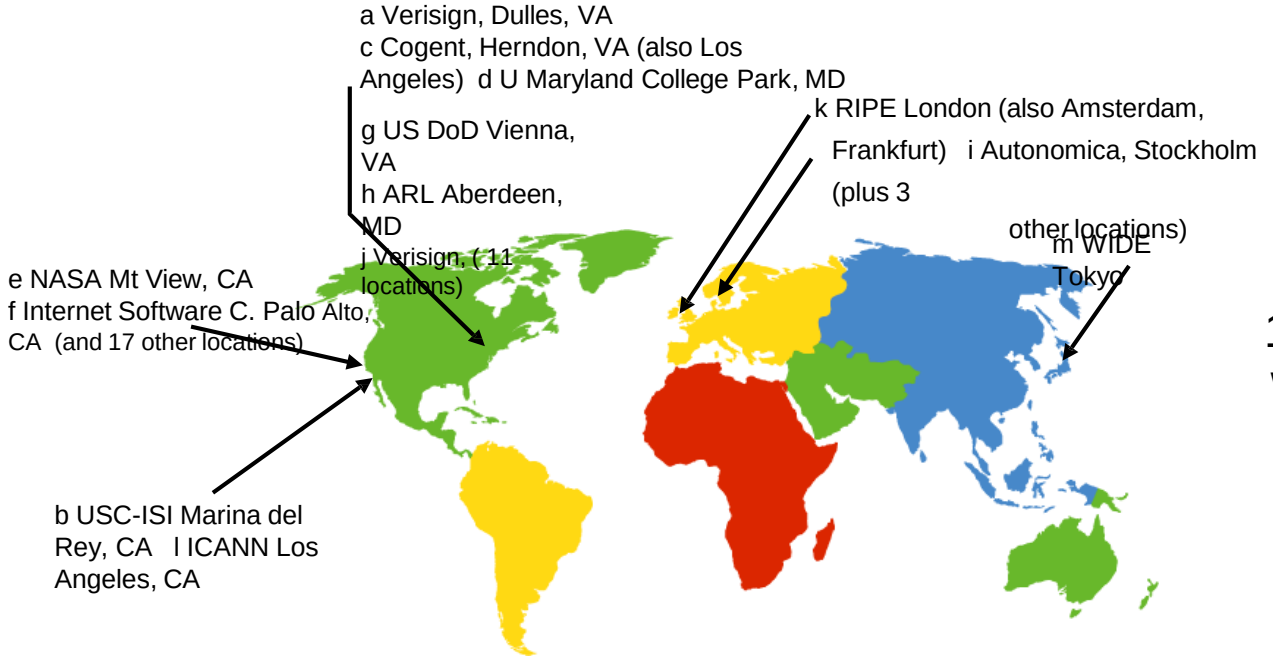
- **The Domain Name Space and Resource Records:** Specifications for a tree-structured namespace and data associated with names.

- **Name Servers:** Server programs which hold information about the domain tree's structure and set information
  - A particular name server has complete information about a subset of the domain space
  - Name servers know the parts of the domain tree for which they have complete information -- a name server is said to be an **AUTHORITY** for this parts of the namespace

- **Resolvers:** Program that extracts information from name servers in response to client requests

Root Name Servers

- contacted by local name server that can not resolve name
- root name server:
  - contacts authoritative name server if name mapping not known
  - gets mapping
  - returns mapping to local name server



13 root name servers worldwide

TLD and Authoritative Servers

- Top-level domain (TLD) servers: These servers are responsible for top-level domains such as com, org, net, edu, and gov, and all of the country top-level domains such as uk, fr, ca, and jp.

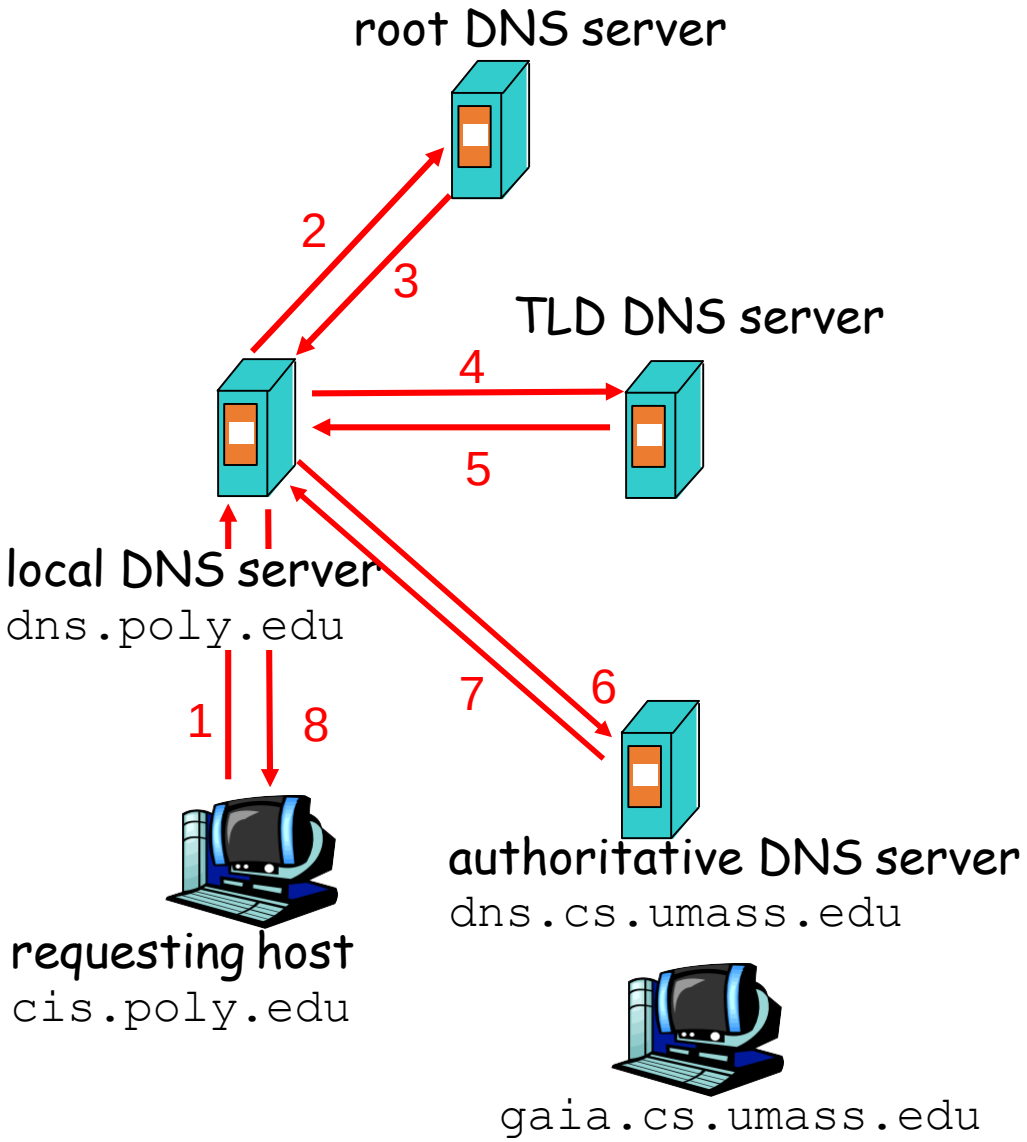
- Authoritative DNS servers: organization's DNS servers, providing authoritative hostname to IP mappings for organization's servers (e.g., Web and mail).
  - Can be maintained by organization or service provider

Local Name Server

- Does not strictly belong to hierarchy
- Each ISP (residential ISP, company, university) has one.
  - Also called “default name server”
- When a host makes a DNS query, query is sent to its local DNS server
  - Acts as a proxy, forwards query into hierarchy.

Example

- Host at cis.poly.edu wants IP address for gaia.cs.umass.edu



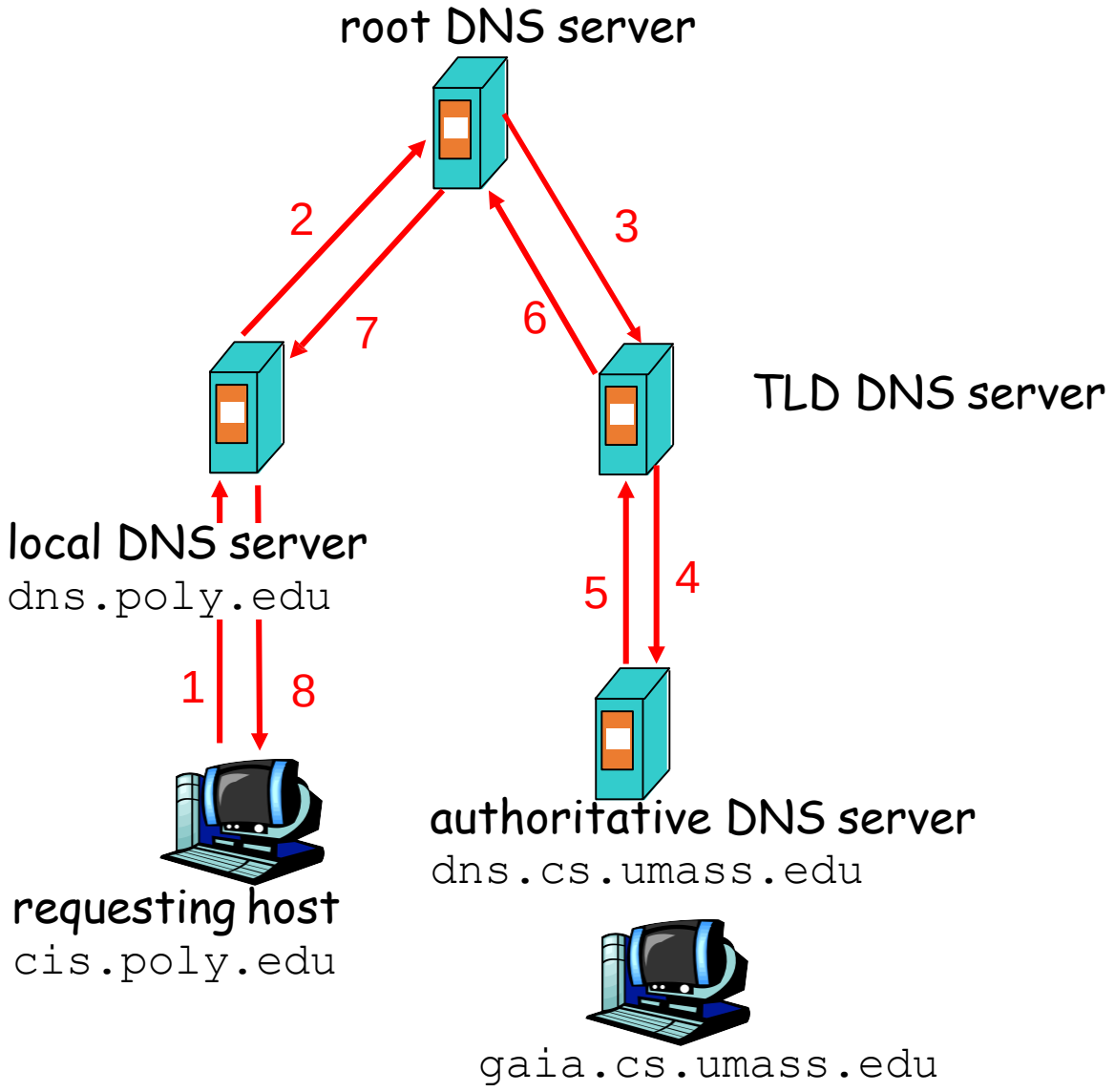
Recursive Queries

recursive query:

- puts burden of name resolution on contacted name server
- heavy load?

iterated query:

- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"



DNS Caching and Updating Records

- once (any) name server learns mapping, it *caches* mapping
  - cache entries timeout (disappear) after some time
  - TLD servers typically cached in local name servers
    - Thus root name servers not often visited
- update/notify mechanisms under design by IETF
  - RFC 2136
  - http://www.ietf.org/html.charters/dnsind-charter.html

DNS Records

DNS: distributed db storing resource records (RR)

RR format: `(name, value, type, ttl)`

Type=A

`name` is hostname

`value` is IP address

Type=NS

`name` is domain (e.g. foo.com)

`value` is IP address of authoritative name server for this domain

Type=CNAME

`name` is alias name for some "cannonical" (the real) name

`www.ibm.com` is really `servereast.backup2.ibm.com`

`value` is cannonical name

Type=MX

`value` is name of mailserver associated with `name`

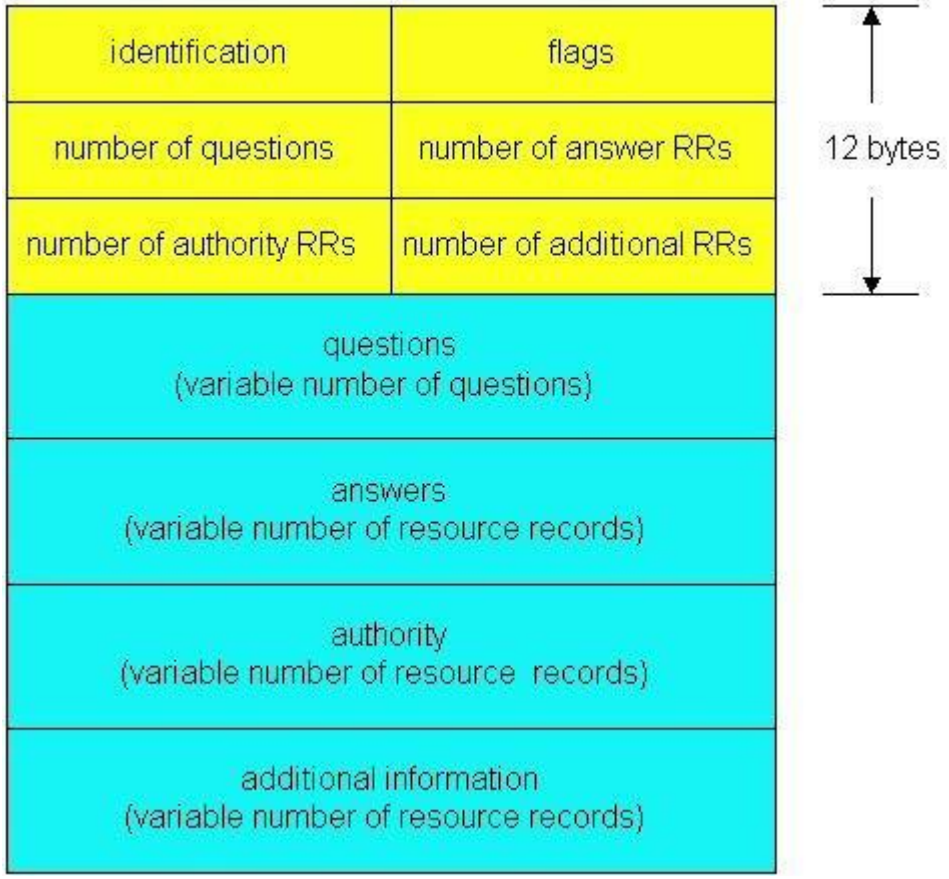
Domain Resource Records

Every domain has a set of **resource records** associated with it – DNS database

Type	Meaning	Value
SOA	Start of authority	Parameters for this zone
A	IPv4 address of a host	32-Bit integer
AAAA	IPv6 address of a host	128-Bit integer
MX	Mail exchange	Priority, domain willing to accept email
NS	Name server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
SPF	Sender policy framework	Text encoding of mail sending policy
SRV	Service	Host that provides it
TXT	Text	Descriptive ASCII text

DNS protocol, messages

DNS protocol: *query* and *reply* messages, both use same *message format*



msg header

identification: 16 bit number for query, reply to query uses same number

flags:

- query or reply
- recursion desired
- recursion available
- reply is authoritative

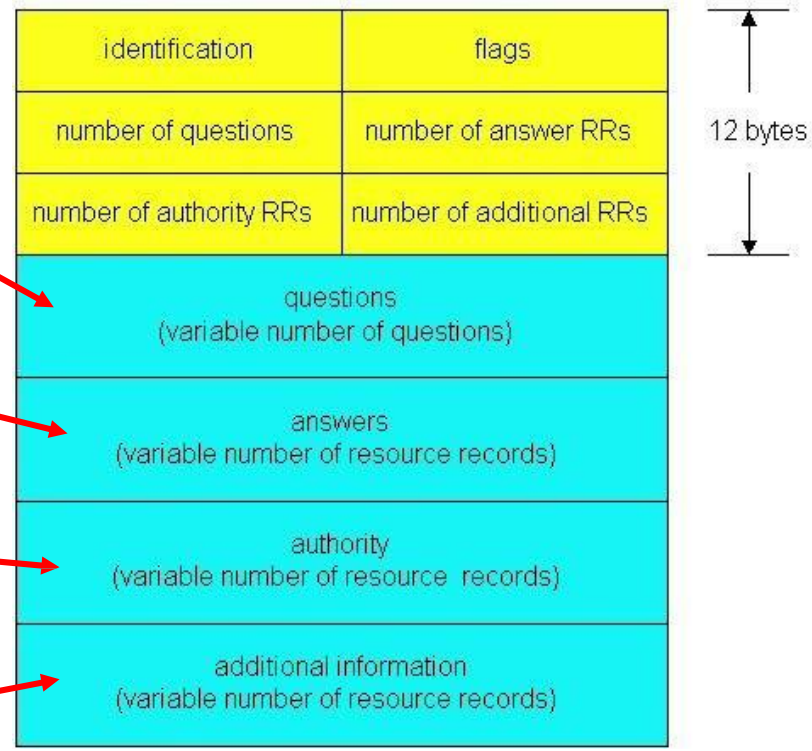
DNS Protocol, message

Name, type fields for a query

RRs in reponse to query

records for authoritative servers

additional "helpful" info that may be used



Inserting records into DNS

- Example: just created startup “Network Utopia”
- Register name networkuptopia.com at a registrar (e.g., Network Solutions)
  - Need to provide registrar with names and IP addresses of your authoritative name server (primary and secondary)
  - Registrar inserts two RRs into the com TLD server:

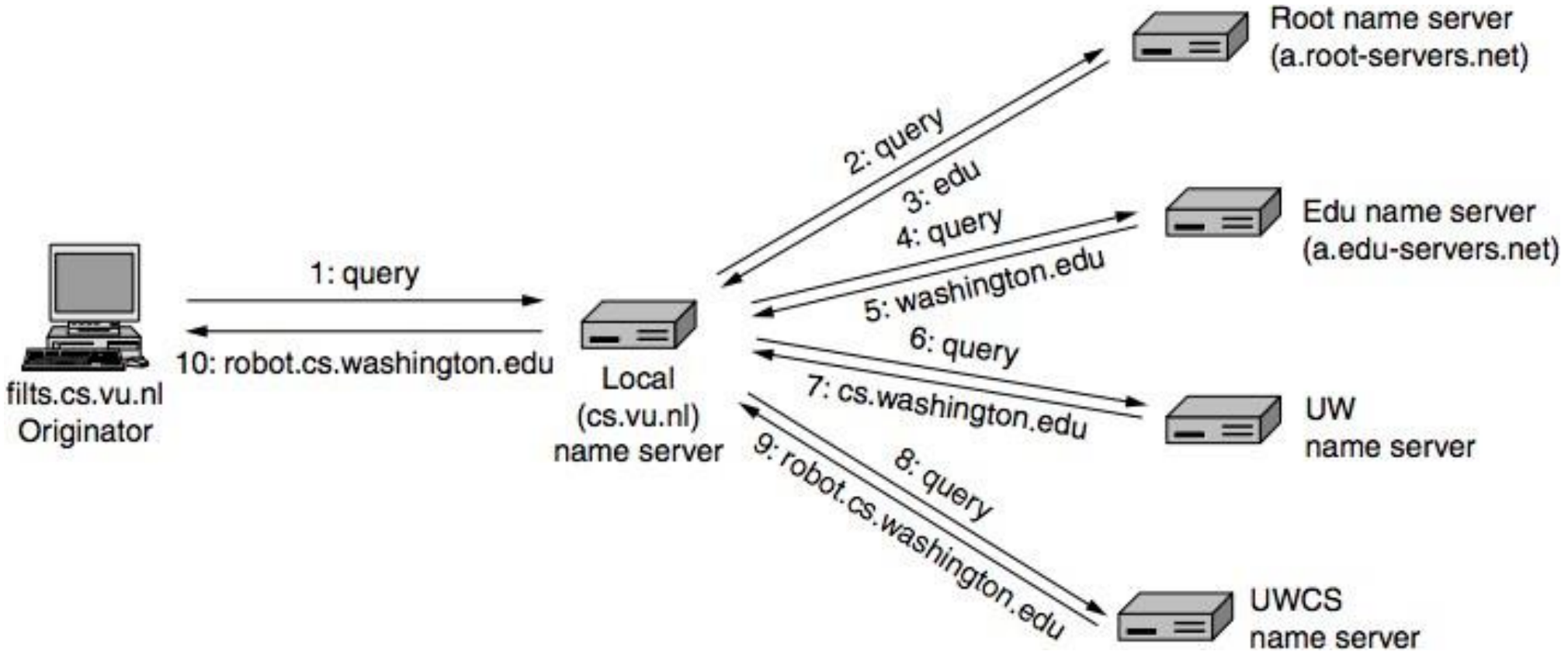
```
(networkutopia.com, dns1.networkutopia.com, NS)
(dns1.networkutopia.com, 212.212.212.1, A)
```

- Insert in authoritative server Type A record for www.networkuptopia.com and Type MX record for networkutopia.com
- How do people get the IP address of your Web site?

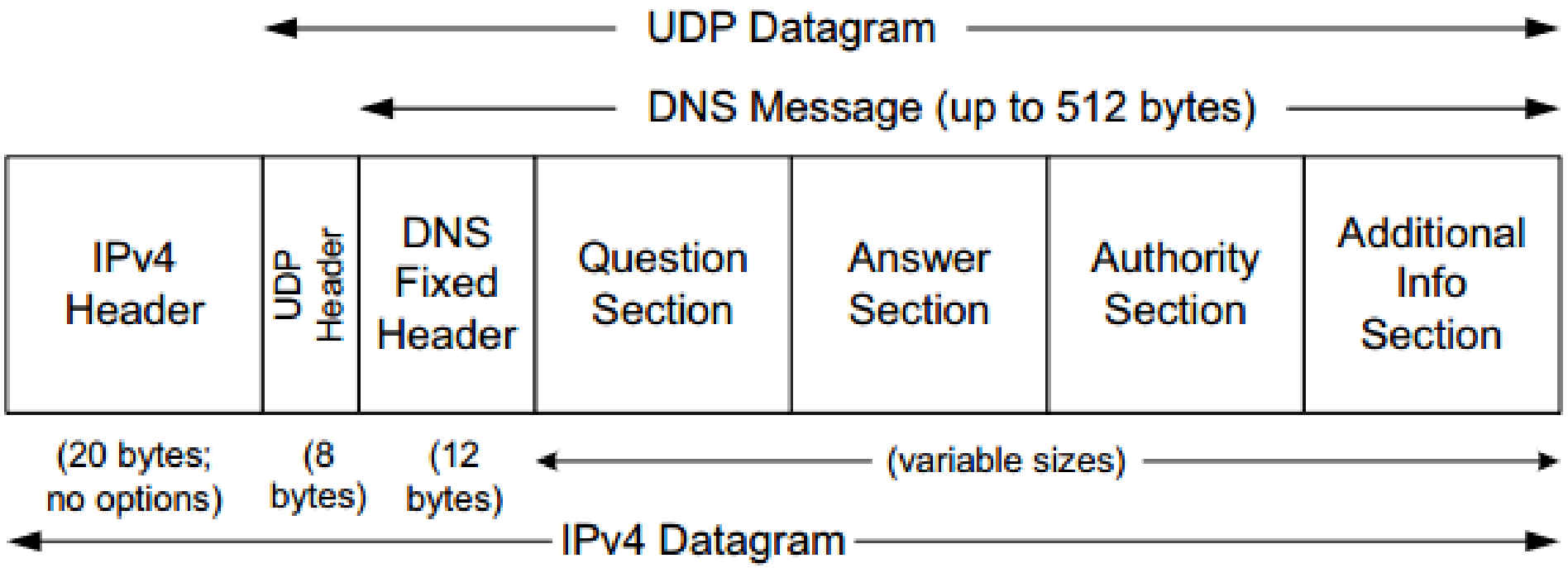
Sample DNS Database

```
; Authoritative data for cs.vu.nl
cs.vu.nl.       86400   IN   SOA     star boss (9527,7200,7200,241920,86400)
cs.vu.nl.       86400   IN   MX      1 zephyr
cs.vu.nl.       86400   IN   MX      2 top
cs.vu.nl.       86400   IN   NS      star

star            86400   IN   A       130.37.56.205
zephyr          86400   IN   A       130.37.20.10
top             86400   IN   A       130.37.20.11
www             86400   IN   CNAME   star.cs.vu.nl
ftp             86400   IN   CNAME   zephyr.cs.vu.nl

flits           86400   IN   A       130.37.16.112
flits           86400   IN   A       192.31.231.165
flits           86400   IN   MX      1 flits
flits           86400   IN   MX      2 zephyr
flits           86400   IN   MX      3 top

rowboat                 IN   A       130.37.56.201
                        IN   MX      1 rowboat
                        IN   MX      2 zephyr

little-sister           IN   A       130.37.62.23

laserjet                IN   A       192.31.231.216
```

Name Resolution – Looking Up for the Names



DNS Packet Structure



Why DNS uses UDP?

- UDP is much faster. TCP requires handshake time. DNS uses a cascading approach for name resolution. With TCP, for every message, a connection setup is required.

- DNS requests and responses are generally very small, and fits well within one UDP segment.

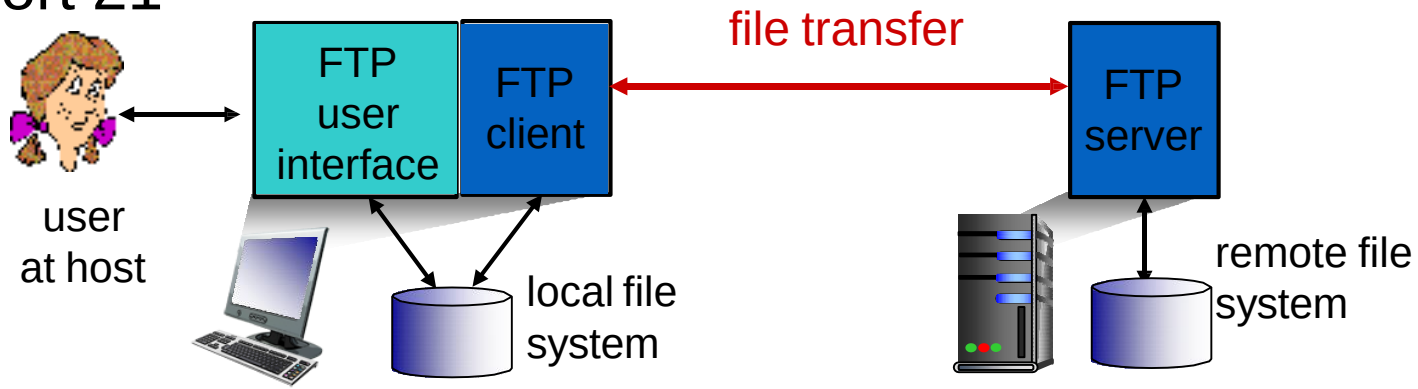
- **UDP is not reliable.** In DNS, reliability is ensured at the application layer. After timeout, the DNS client sends back the requests. After few consecutive timeouts (can be set at the client), the request is aborted with an error.

File Transfer Service

- Goal
  1. to promote sharing of files (computer programs and/or data),
  2. to encourage indirect or implicit (via programs) use of remote computers,
  3. to shield a user from variations in file storage systems among hosts, and
  4. to transfer data reliably and efficiently
- Use client-server model based on TCP/IP
- Authenticated and anonymous accesses

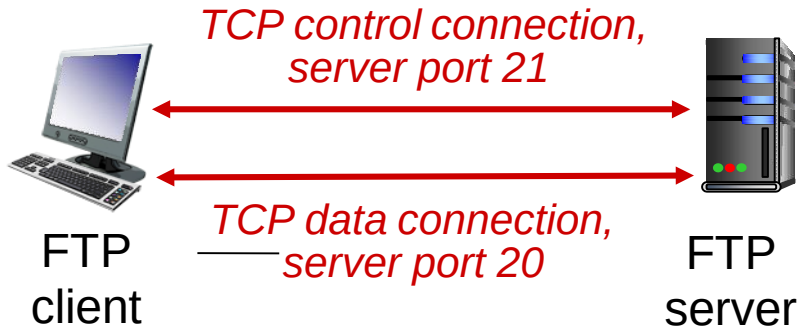
FTP: the file transfer protocol

- Transfer file to/from remote host
- Client/server model
  - *client:* side that initiates transfer (either to/from remote)
  - *server:* remote host
- FTP: RFC 959
- FTP server: port 21



FTP: separate control, data connections

- FTP client contacts FTP server at port 21, using TCP
  - client is authorized over control connection
  - client browses remote directory, sends commands over control connection
  - when server receives file transfer command, *server* opens *2nd* TCP data connection (for file) *to* client
  - after transferring one file, server closes data connection



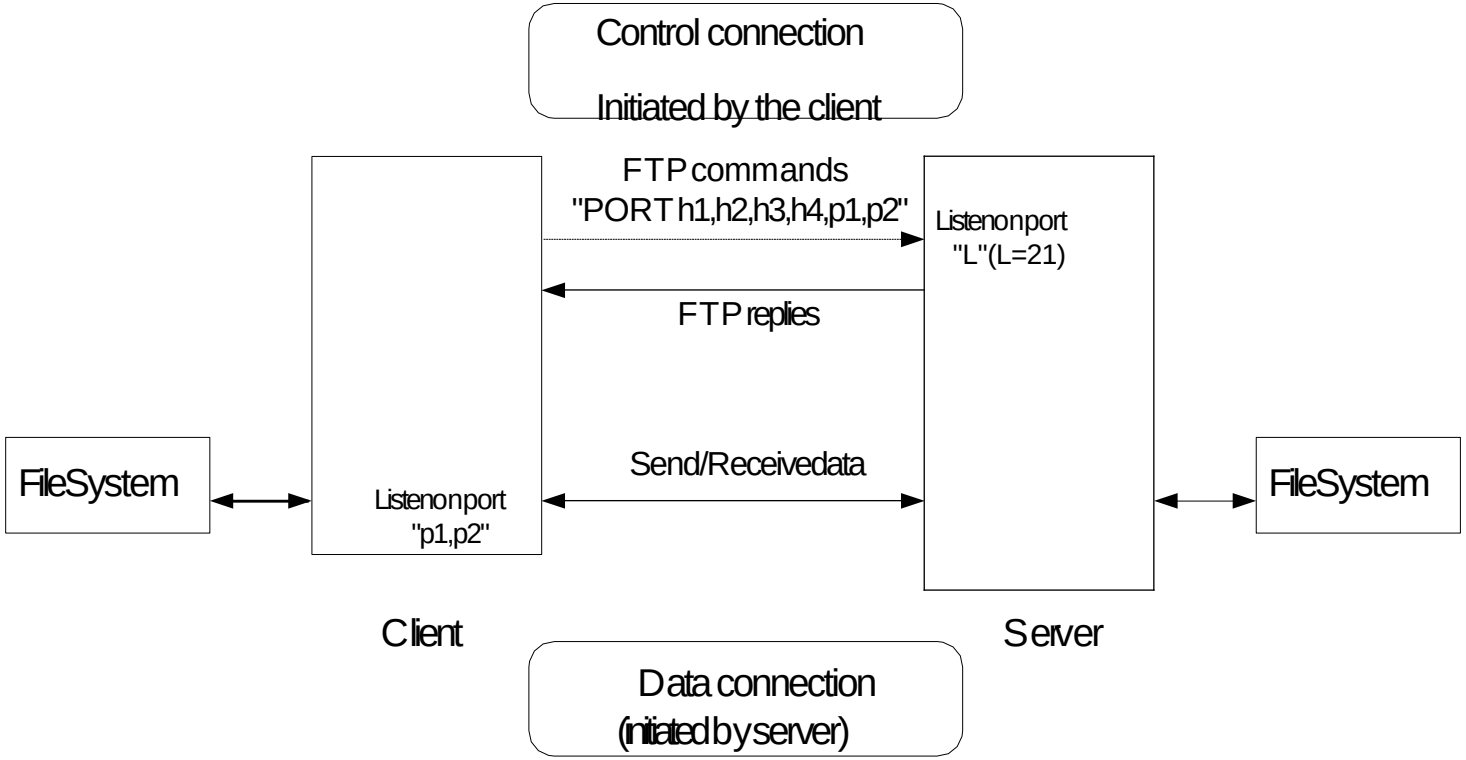
- server opens another TCP data connection to transfer another file
- control connection: *"out of band"*
- FTP server maintains "state": current directory, earlier authentication

Some Application FTP Commands

Command	Description
OPEN	Connect to a remote host
CAT	View a file in a remote host
GET	Retrieve files in a remote host
RENAME	Change the name of a file in a remote host
RM	Delete a file in a remote host
QUIT	Terminate an FTP session

Operational Model

PORT: Send the IP and port of the client to which the data is retrieved

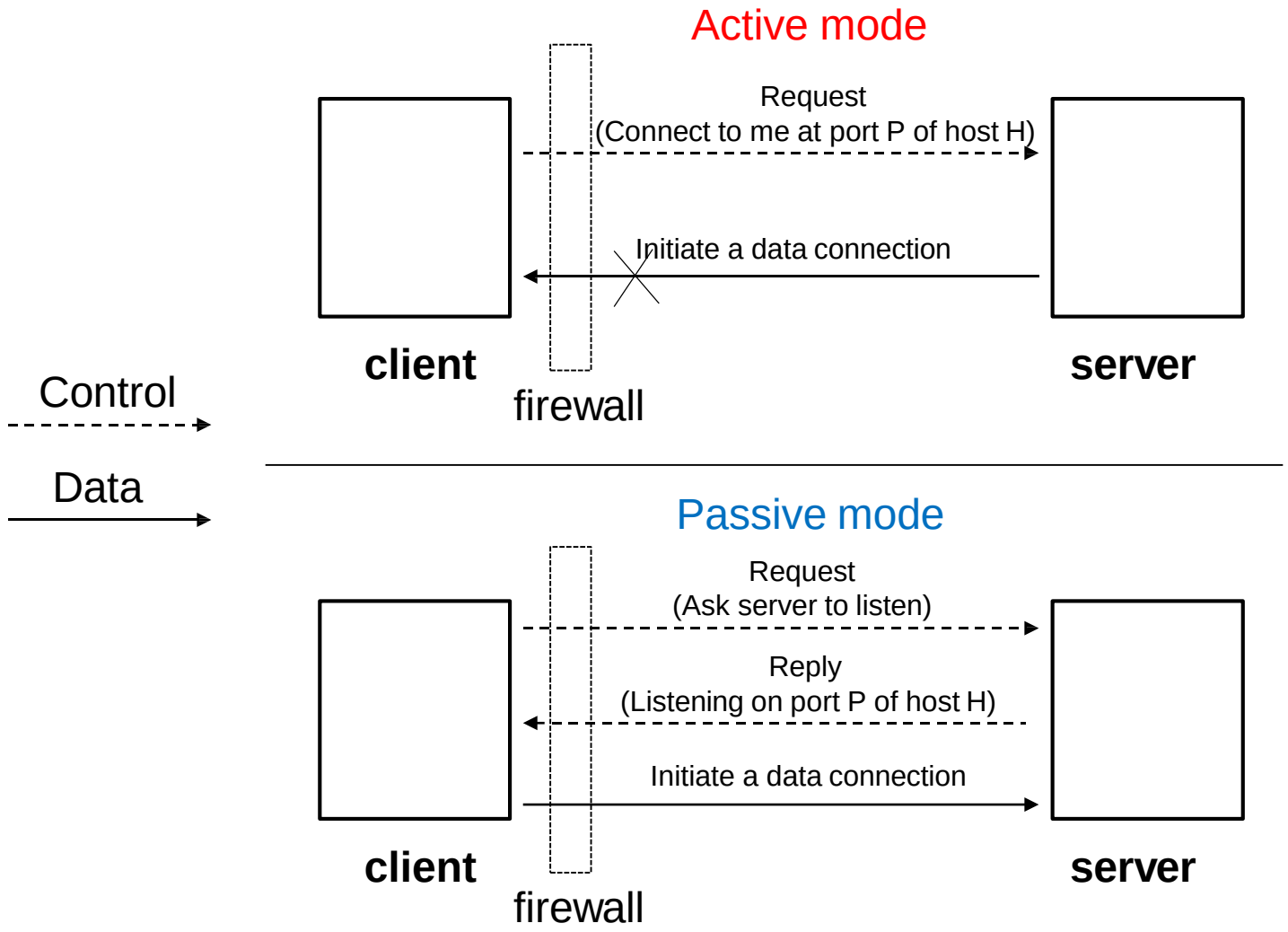


Client as both control host and receiver

Inside the Connections

- Establishing control/data connections
  - Active Mode
    - Control connection initiated by client
    - Data connection initiated by server
  - Passive Mode
    - When client is behind a firewall / NAT server
    - Both control/data connections are initiated by client
- FTP Reply
- Error Recovery

Active/Passive Mode



Few FTP Commands

Command	Description	Type
USER	Send the user name	Access Control
PASS	Send the password	Access Control
PORT	Send the IP and port of the client to which the data is retrieved	Transfer Parameter
PASV	Tell the server to listen on a data port rather than initiate a data connection	Transfer Parameter
RETR	Ask server to transfer a copy of the requested file to the client	File service
STOR	Cause the server to accept and receive the data and store it as a file	File service
RNFR	Specify the path of a source file to rename from	File service
RNTO	Specify the path of a destination file to rename to	File service
ABOR	Tell the server to abort the previous command and the corresponding data transfer	File service

FTP Reply

Reply	Description	Type
1yz	The requested action is being initiated; expect another reply before proceeding with a new command.	Positive Preliminary reply
2yz	The requested action has been successfully completed.	Positive Complete reply
3yz	The command has been accepted, but the requested action is being held, waiting for further information from another command.	Positive Intermediate reply
4yz	The command is not accepted the and the requested action did not take place. The action can be requested again.	Transient Negative Completion reply
5yz	Similar with 4yz, except that the error condition is permanent so that the action cannot be requested again.	Permanent Negative Completion reply

Error Recovery

- The restart mechanism
  - Sender inserts ‘marker’ (used to identify the checkpoint) in the data stream
  - Receiver marks the position of the marker and reply the latest marker position of both sender and receiver to user
  - When error, user issues ‘restart’ with the position of the marker to the sender

    * User (control host) and receiver may/may not exist in the same machine

Example of an FTP Session

```
STATUS:>    Connecting to www.cis.nctu.edu.tw (ip = 140.113.166.122)
STATUS:>    Socket connected. Waiting for welcome message...
            220 www.cis.nctu.edu.tw FTP server (Version wu-2.6.0(1) Mon Feb 28 10:30:36 EST 2000) ready.
COMMAND:>           USER www
            331 Password required for www.
COMMAND:>           PASS ********
            230 User www logged in.
COMMAND:>           TYPE I
            200 Type set to I.
COMMAND:>           REST 100
            350 Restarting at 100. Send STORE or RETRIEVE to initiate transfer.
COMMAND:>           REST 0
            350 Restarting at 0. Send STORE or RETRIEVE to initiate transfer.
COMMAND:>           pwd
            257 "/home/www" is current directory.
COMMAND:>           TYPE A
            200 Type set to A.
COMMAND:>           PORT 140,113,189,29,10,27 ← tell the server where to connect to
            200 PORT command successful.
COMMAND:>           LIST                      ← retrieve directory listing
            150 Opening ASCII mode data connection for /bin/ls. ← File status okay; about to open data connection

…….list of files….

COMMAND:>           TYPE I
            200 Type set to I.
COMMAND:>           PORT 140,113,189,29,10,31
            200 PORT command successful.
COMMAND:>           RETR test                 ← retrieve the file "test"
            150 Opening BINARY mode data connection for test (5112 bytes).
```