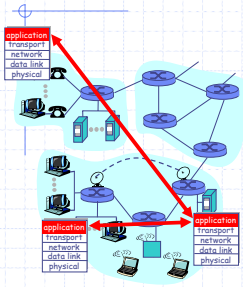
ECE453 – Introduction to Computer Networks

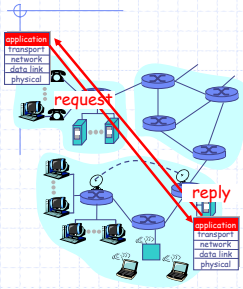
Lecture 16 – Application Layer
(DNS, E-mail, Web, FTP)

Applications and Application Level Protocols



- The three concepts
  - Service model
  - Protocol
  - Interface
- Network application is more than application level protocols
  - Client site
  - Server site
  - Application level protocol

Client/Server Paradigm



- Client
  - Initiates contact with server (speak first)
  - Typically request service from server
  - Question: identify who is/implements client in
    - Web?
    - Email?
- Server
  - Provides requested service to clients
  - Question: identify who is/implements the server counterpart in
    - Web?
    - Email?

Which Transport Service Does Application Need? - Parameters

- Data Loss
  - Loss-tolerant applications, e.g. audio/video
  - other app such as file transfer, telnet requires 100% reliable transmission
- Bandwidth
  - Bandwidth-sensitive applications, such as multimedia, require a maximum amount of bandwidth
  - Elastic applications: can use whatever bandwidth available
- Timing
  - Some apps such as internet telephone requires “low delay” to be effective

Transport Service Required By Common Applications

Application	Data loss	Bandwidth	Time Sensitive
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
Web documents	loss-tolerant	elastic	no
real-time audio/video	loss-tolerant	audio: 5Kb-1Mb video:10Kb-5Mb	yes, 100’s msec
stored audio/video	loss-tolerant	same as above	yes, few secs
interactive games	loss-tolerant	few Kbps up	yes, 100’s msec
financial apps	no loss	elastic	yes and no

Internet Apps and Their Transport Layer Protocols

Application	Application layer protocol	Underlying transport protocol
e-mail	smtp [RFC 821]	TCP
remote terminal access	telnet [RFC 854]	TCP
Web	http [RFC 2068]	TCP
file transfer	ftp [RFC 959]	TCP
streaming multimedia	proprietary (e.g. RealNetworks)	TCP or UDP
remote file server	NFS	TCP or UDP
Internet telephony	Proprietary (private) (e.g., Vocaltec)	typically UDP

DNS – Domain Name System

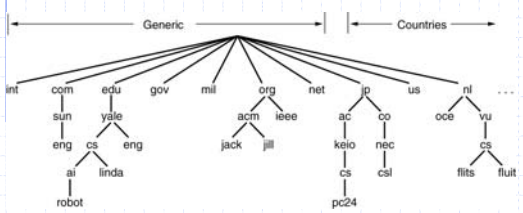
IP: 160.36.30.108

Name: panda.ece.utk.edu

DNS: Mapping Name To Address

- Name: panda.ece.utk.edu is used by human
- IP address: a 32-bit numerical value used by machine
- DNS
  - A distributed database, implemented by a hierarchy of name servers
  - Application level protocols used by hosts, routers and name servers
  - Internet intelligence is on the edge

DNS Name Space



biz, info, name, pro
aero, coop, museum

ICANN
http://www.icann.org/

DNS – Why Not Centric?

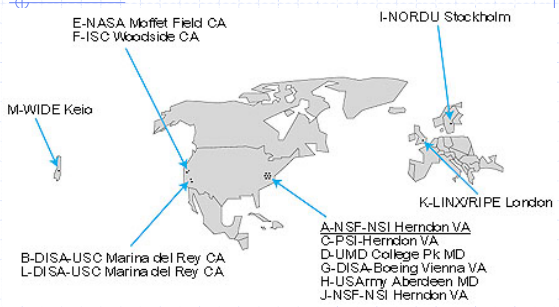
- Single point of failure
- Traffic volume
- Distant name server means slow response
- Scalability

- History: ARPANET begins with a single **`hosts.txt`**.

DNS – Hierarchical View

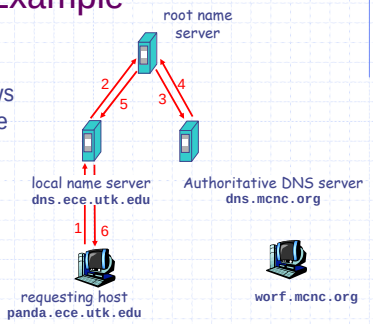
- Local DNS server
- Authoritative DNS server
- Root DNS server

DNS: Where Are Root Servers?



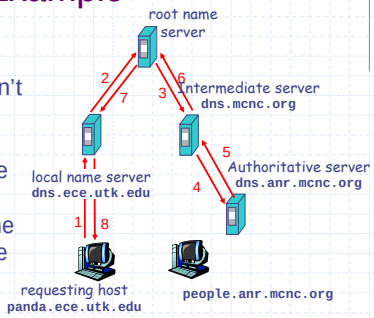
DNS: An Example

- Case: Root server knows authoritative DNS server



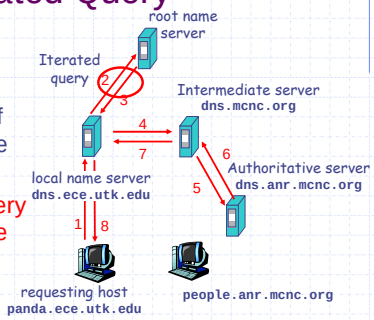
DNS: An Example

- Case: Root server doesn't know immediate authoritative DNS server, but know the intermediate one



DNS: Iterated Query

- Root server replies with the name of intermediate server
- Iterated query vs. recursive query



DNS Caching

- Once (any) server learns new mapping, it caches it
- The cache will expire after some time
- Update/notify mechanism is defined by IETF RFC 2136

DNS Server and Service

- Running on top of UDP
- Port number: 53
- Frequently used by other applications such as SMTP, FTP, HTTP
- Important services
  - Host aliasing
  - Mail server aliasing
  - Load distribution (DNS rotation)
- User utilities: `dig`, `http://www.netliner.com/dig.html`
- More DNS information: see DNS NET `http://www.dns.net/dnsrd/docs/`

*DNS Resource Record

DNS: distributed database storing resource records (RR)

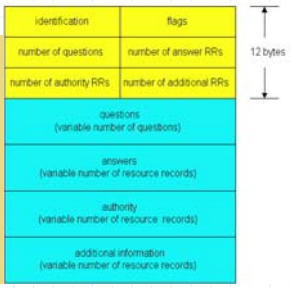
RR format: `(domain_name, ttl, class, type, value)`

Type	Meaning	Value
SOA	Start of Authority	Parameters for this zone
A	IP address of a host	32-Bit integer
MX	Mail exchange	Priority, domain willing to accept e-mail
NS	Name Server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
HINFO	Host description	CPU and OS in ASCII
TXT	Text	Uninterpreted ASCII text

*DNS: Message Format

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

message header
- identification: 16 bit # for query, reply to query uses same #
- flags:
  - query or reply
  - recursion desired
  - recursion available
  - reply is authoritative



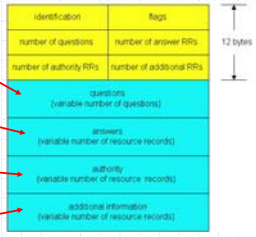
*DNS Protocol Message

Name, type fields for a query

RRs in reponse to query

records for authoritative servers

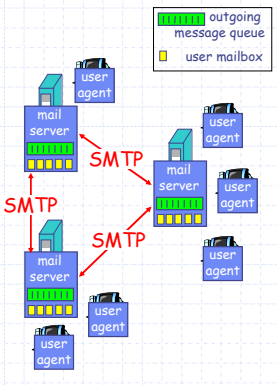
additional "helpful" info that may be used



Electronic Mail

Three major components:
- user agents
- mail servers
- simple mail transfer protocol: smtp

User Agent
- "mail reader"
- composing, editing, reading mail messages
- e.g., Eudora, Outlook, elm, Netscape Messenger
- outgoing, incoming messages stored on server



Electronic Mail: Mail Server

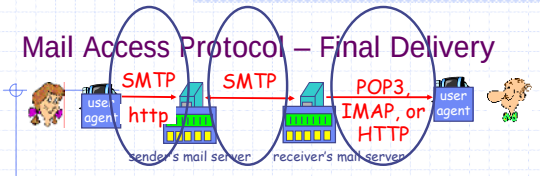
- mailbox contains incoming messages (yet to be read) for user
- message queue of outgoing (to be sent) mail messages
- Common Mail Server:
  - Sendmail
  - MS Exchange

SMTP: RFC 821

- Use TCP for reliable transfer, use port number 25
- Message must be 7-bit ASCII

```
[hqi@aicip hqi]$ telnet panda.ece.utk.edu 25
Trying 160.36.30.108...
Connected to panda.ece.utk.edu.
Escape character is '^]'.
220 panda.ece.utk.edu ESMTP Sendmail 8.11.6/8.11.6; Thu, 21 Nov
2002 09:54:04 -0500
HELO panda.ece.utk.edu
250 panda.ece.utk.edu Hello pegasus.ece.utk.edu
[160.36.30.110], pleased to meet you
MAIL FROM: <hqi@aicip.ece.utk.edu>
250 2.1.0 <hqi@aicip.ece.utk.edu>... Sender ok
RCPT TO: <hqi@panda.ece.utk.edu>
250 2.1.5 <hqi@panda.ece.utk.edu>... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
do you like ketchup?
how about pickles?
.
250 2.0.0 gALEt5U25932 Message accepted for delivery
QUIT
221 2.0.0 panda.ece.utk.edu closing connection
Connection closed by foreign host.
```

Mail Access Protocol – Final Delivery



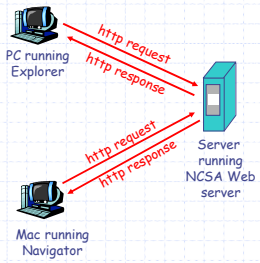
- SMTP: delivery/storage to receiver's server
- Mail access protocol: retrieval from server
  - POP: Post Office Protocol [RFC 1939] (port 110)
    - authorization (agent <-->server) and download
    - Does not maintain state across POP sessions
    - Cannot manipulate emails at the server side
  - IMAP: Internet Mail Access Protocol [RFC 1730]
    - more features (more complex)
    - manipulation of stored msgs on server
    - Maintain state for the user
  - HTTP: Hotmail , Yahoo! Mail, etc.
    - Slow

Web: Terminology

- Web page
  - Consists of "objects"
  - Addressed by "url" (universal resource locator)
- Most of web page
  - One base web page
  - Several referenced "objects"
- URL has two components
  - A host name and a path
  - http://panda.ece.utk.edu/~hqi/teaching.html
- Web client
  - Netscape communicator
  - Mozilla
  - Microsoft IE browser
- Web server
  - Apache
  - Microsoft Internet Information Server (IIS)

Web: the http Protocol

- Web application layer protocol: a hyper text transfer protocol, http
- Defined by
  - HTTP 1.0, RFC 1945
  - HTTP 1.1, RFC 2068
- Client/Server Mode
  - Client: browser asks for objects, and display it
    - Request
    - Display
  - Server: provide objects in response to requests



Web: http Operation Flow

- HTTP utilizes TCP transport services
- HTTP client initiates TCP connection (create socket) to server, at port 80
- Server accepts this connection from client
- HTTP messages (defined by HTTP protocol) are exchanged between http client and http server
- TCP connection closed

- HTTP is stateless
  - Server doesn't maintain the state of past requests
    - 'back'?

HTTP Example

Suppose user enters URL
www.someSchool.edu/someDepartment/home.index

(contains text, references to 10 jpeg images)

1a. http client initiates TCP connection to http server (process) at www.someSchool.edu. Port 80 is default for http server.

1b. http server at host www.someSchool.edu waiting for TCP connection at port 80. "accepts" connection, notifying client

2. http client sends http *request message* (containing URL) into TCP connection socket (plus another acknowledge message)

3. http server receives request message, forms *response message* containing requested object (someDepartment/home.index), sends message into socket

time

HTTP Example (cont'd)

4. http server closes TCP connection.

5. http client receives response message containing html file, displays html. Parsing html file, finds 10 referenced jpeg objects

6. Steps 1-5 repeated for each of 10 jpeg objects

time

- Two RTT (Round-trip time)
- Slow start
- Place burden on the Web server

HTTP: Persistent and Non-persistent Connection

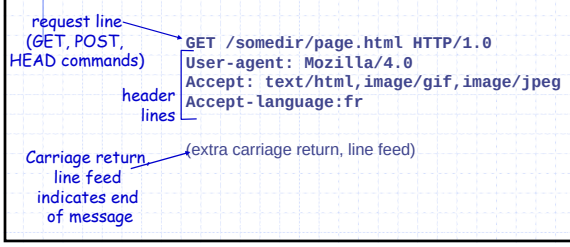
- Non-persistent
  - HTTP 1.0
  - Server parses request, responds, then closes TCP connection
  - Each object requires 2 RTT
  - Each object suffers slow start
- Persistent
  - HTTP 1.1
  - On the same TCP connection, server parses request, responds, and parses new requests

SMTP vs. HTTP

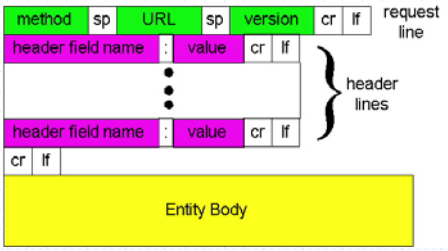
- HTTP: Direct connection, no intermediate mail servers
- Both use persistent connection
- HTTP is a pull protocol, while SMTP is a push protocol
- SMTP: 7-bit ASCII format, message ended with a line consisting of only a period

*HTTP Message Format

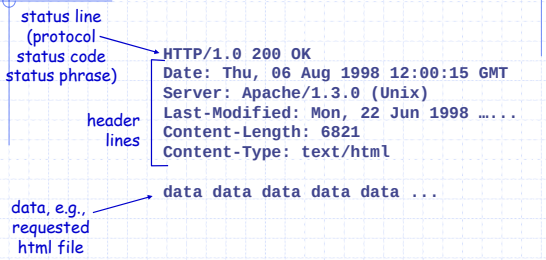
- two types of http messages: *request*, *response*
- http request message:
  - ASCII (human-readable format)



*HTTP Request: General Format



*HTTP Message Format: Response



*HTTP Response: Status Code

200 OK
- request succeeded, requested object later in this message

301 Moved Permanently
- requested object moved, new location specified later in this message (Location:)

400 Bad Request
- request message not understood by server

404 Not Found
- requested document not found on this server

505 HTTP Version Not Supported

Try Out http (client side) For Yourself

- Telnet to your favorite web site

  telnet panda.ece.utk.edu 80 — open TCP connection to panda port 80, anything you type be sent to panda port 80 socket

- Type in request, and look at the response

  GET /~hqi/index.html HTTP/1.0

[hqi@com779 hqi]$ telnet panda.ece.utk.edu 80

Trying 160.36.30.108...

Connected to panda.ece.utk.edu.

Escape character is '^]'.

get /~hqi/index.html http/1.0

HTTP/1.1 501 Method Not Implemented

Date: Sun, 02 Sep 2001 21:03:28 GMT

Server: Apache/1.3.19 (Unix) (Red-Hat/Linux) PHP/4.0.4pl1

Allow: GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, PATCH, PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK, UNLOCK, TRACE

Connection: close

Content-Type: text/html; charset=iso-8859-1

[hqi@com779 hqi]$ telnet panda.ece.utk.edu 80

Trying 160.36.30.108...

Connected to panda.ece.utk.edu.

Escape character is '^]'.

GET /~hqi/index.html http/1.0

HTTP/1.1 200 OK

Date: Sun, 02 Sep 2001 21:05:43 GMT

Server: Apache/1.3.19 (Unix) (Red-Hat/Linux) PHP/4.0.4pl1

Last-Modified: Sat, 07 Jul 2001 15:42:14 GMT

ETag: "1f222e-df9-3b472dd6"

Accept-Ranges: bytes

Content-Length: 3577

Connection: close

Content-Type: text/html

<HTML>

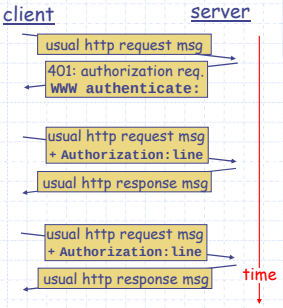
……

</HTML>

Connection closed by foreign host.

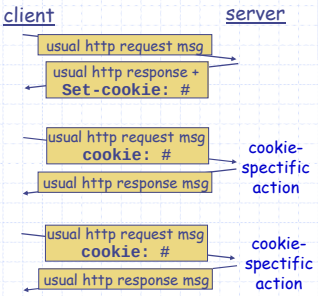
Add-on Features: Authentication

- Purpose of authentication: control access to document
- Means: user name and password
- User must present password on each request, Authorization:line
- Server asks for it by giving it the response with WWW authenticate:



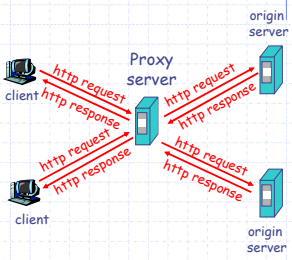
Add-on Feature: Cookies

- Server sends user cookie in response message:
  - `Set-cookie: 1678453`
- Client presents cookie in later request
  - `cookie: 1678453`
- Server matches cookies with stored information: such as user preference, password etc



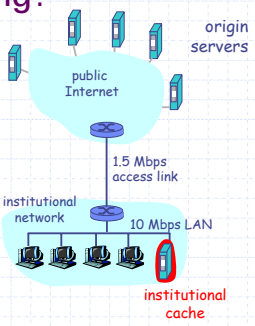
Add-on Feature: Web Caches or Proxy Servers

- Goal: to satisfy user request without invoking origin server
- User makes request, the object requested has been cached, then proxy server will reply, else proxy server request the object for client and then response

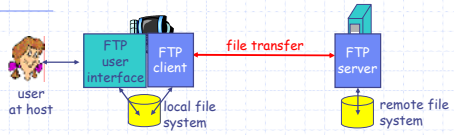


Why Web Caching?

- Cache should be closer to the clients
- Faster response
- Reduce traffic (pay less money)
- Web cache:
  - Cost is low



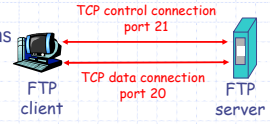
FTP: File Transfer Protocol



- transfer file to/from remote host
- client/server model
  - *client:* initiates transfer (either to/from remote)
  - *server:* remote host
- ftp: RFC 959

FTP: Operation Flow

- ftp client contacts ftp server at port 21, specifying TCP as transport protocol
- two parallel TCP connections opened:
  - control: exchange commands, responses between client, server
    "out of band control"
  - data: file data to/from server
- ftp server maintains "state": current directory, earlier authentication



Data connection is closed whenever it finished transferring one file.

FTP: Command and Response

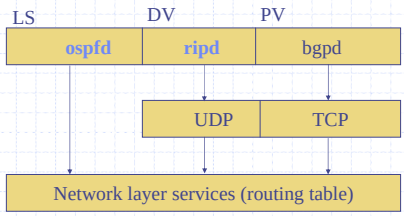
Sample commands:

- sent as ASCII text over control channel
- **USER *username***
- **PASS *password***
- **LIST** return list of file in current directory
- **RETR filename** retrieves (gets) file
- **STOR filename** stores (puts) file onto remote host

Sample return codes

- status code and phrase (as in http)
- **331 Username OK, password required**
- **125 data connection already open; transfer starting**
- **425 Can't open data connection**
- **452 Error writing file**

Routing Protocol Applications



Summary

- Application
  - Client
  - Server
  - Protocol
    - What type of service
    - Through what interface
    - Which port
- DNS
  - Aliasing vs. load distribution
  - "dig"
  - UDP
- Email
  - SMTP
  - Mail access protocol
    - POP3
    - IMAP
    - HTTP
  - TCP
- Web
  - http
  - Stateless
  - Persistent
  - Pull vs. push
  - Cookies, authentication, proxy server
- File transfer
  - ftp
  - Two connection flows
  - Maintain state
- Application protocols
  - Push vs. pull
  - State vs. stateless
  - Persistent vs. non-persistent
  - Port number
  - Service model used