Topics

• Domain Naming System (DNS)
• World Wide Web
  - Web servers
  - HTTP (static content)
Internet protocol stack

Berkeley sockets interface

Unreliable best effort datagram delivery (process-process)

Unreliable best effort datagram delivery (host-host)

Reliable byte stream delivery (process-process)

Physical connection

Applications and services (DNS, WWW, email)

User datagram protocol (UDP)  Transmission control protocol (TCP)

Internet Protocol (IP)

Network interface (ethernet)

hardware
Hierarchical domain name space

Until 198x, domain name/IP address mapping maintained in HOSTS.TXT file at SRI.

Each new host manually entered and copied to backbone routers.

Explosive growth rendered HOSTS.TXT approach impractical.

Replaced by Domain Name System in 198x.
DNS

Worldwide distributed system for mapping domain names to IP addresses (and vice versa).
Implemented as a collection of cooperating servers called name servers.

Name servers perform lookups for DNS clients

- user programs
  - `gethostbyname()`, `gethostbyaddr()`
- `nslookup`
  - stand-alone client with command line interface

```
   kittyhawk> nslookup bass.cmcl
   Server:  localhost
   Address:  127.0.0.1

   Non-authoritative answer:
   Name:    bass.cmcl.cs.cmu.edu
   Address:  128.2.222.85
```
Domains are partitioned into zones.

Each zone has multiple name servers that store info about names in that zone.

- CS zone has 4 servers

One server is authoritative

- the others get copies of the authoritative server's data
Zone databases

Each name server keeps a database with information about each name in its zone.

Examples of info (type: description)

- **A**: IP address
- **NS**: name servers for zone
- **SOA**: “start of authority” indicates authoritative server
- **WKS**: well known services running on that host
- **HINFO**: host info (OS and machine type)
- **PTR**: domain name ptr (if this subdomain has its own server)
Zone transfers

Clients can inspect the contents of a zone database via a copy operation called a zone transfer.

- All info of a particular type or types (A, NS, etc) of info for each domain name in the entire zone is copied from server to client.

Servers can control which client machines are allowed to perform zone transfers.

```
Example: zone transfer of cs.cmu.edu (Types A & PTR)
(note: this is the default for nslookup)

... SAHARA.CMCL 128.2.185.40 ...
... LB             server = ALMOND.SRV.CS.CMU.EDU
LB             server = PECAN.SRV.CS.CMU.EDU
... POSTOFFICE    128.2.181.62 ...
```
Zone transfers (cont)

Example: zone transfer of cs.cmu.edu (Type HINFO)

```
... SAHARA.CMCL DEC-600-5/333 UNIX ...
... AMEFS.SRV INTEL-486 UNIX ...
```

Note: no HINFO for POSTOFFICE or LB
Mapping domain names to IP addr

Used by gethostbyname() and nslookup

(1) nslookup sahara.cmcl.cs.cmu.edu

(2) R

(3) PTR to edu name server (ns)

(4) R

(5) PTR to cmu.edu ns

(6) R

(7) PTR to cs.cmu.edu ns

(8) R

(9) 128.2.185.40

(10) 128.2.185.40

R = sahara.cmcl.cs.cmu.edu
DNS Caching

Servers cache (keep a copy of) of information they receive from other servers as part of the name resolution process. This greatly reduces the number of queries.

Example

- In our previous example, the next query for sahara.cmcl can be answered immediately because the server kept a copy of the address.

```
1. nslookup sahara.cmcl.cs.cmu.edu
```

```
client

name server

10. 128.2.185.40
```
Mapping IP addrs to domain names

A separate hierarchy exists in the in-addr.arpa domain that maps IP addresses to domain names.

Used by `gethostbyaddr()` and `nslookup`

Example:

- **IP address**: 128.2.185.40
- **Corresponding domain name**: sahara.cmcl.cs.cmu.edu stored at 40.185.2.128.in-addr.arpa

```
sahara.cmcl.cs.cmu.edu
```
Web history

1945:
    - Describes the idea of a distributed hypertext system.
    - a “memex” that mimics the “web of trails” in our minds.

1989:
  • Tim Berners-Lee (CERN) writes internal proposal to develop a distributed hypertext system.
    - connects “a web of notes with links”.
    - intended to help CERN physicists in large projects share and manage information

1990:
  • Tim BL writes graphical browser for Next machines.
Web history (cont)

1992

- NCSA server released
- 26 WWW servers worldwide

1993

- Marc Andreessen releases first version of NCSA Mosaic (Feb)
- Mosaic version released for (Windows, Mac, Unix).
- Web (port 80) traffic at 1% of NSFNET backbone traffic.
- Over 200 WWW servers worldwide.

1994

- Andreessen and colleagues leave NCSA to form "Mosaic Communications Corp" (now Netscape).
Internet Domain Survey (www.isc.org)

Internet hosts


Mosaic and Netscape
Web servers

Clients and servers communicate using the HyperText Transfer Protocol (HTTP)
- client and server establish TCP connection
- Client requests content
- Server responds with requested content
- client and server close connection (usually)

Current version is HTTP/1.1
- RFC 2616, June, 1999.
Web server statistics

(Figure showing that Apache is now the dominant web server has been deleted because it was too large; see either the Powerpoint source file or a hardcopy handout.)

source: Netcraft Web Survey
www.netcraft.com/survey
Static and dynamic content

The content returned in HTTP responses can be either static or dynamic.

Static content:

• content stored in files and retrieved in response to an HTTP request
  - HTML files
  - images
  - audio clips

Dynamic content:

• content produced on-the-fly in response to an HTTP request
  - Example: content produced by a CGI process executed by the server on behalf of the client.
URIs and URLs

network resources are identified by Universal Resource Indicators (URIs)

The most familiar is the absolute URI known as the HTTP URL:

- http-url = “http:” “//” host [“:” port] [abs_path]
- port defaults to “80”
- abs_path defaults to “/”
- abs_path ending in / defaults to .../index.html

Examples:

- http://euro.ecom.cmu.edu:80/index.html
- http://euro.ecom.cmu.edu/index.html
- http://euro.ecom.cmu.edu
HTTP/1.1 messages

An HTTP message is either a Request or a Response:

HTTP-message = Request | Response

Requests and responses have the same basic form:

generic-message = start-line
                 *message-header
                 CRLF
                 [message body]

start-line = Request-line | Status line
message-header = field-name ":" [field value] CRLF
message-body = <e.g., HTML file>
HTTP/1.1 requests

Request = Method SP Request-URI SP HTTP-VERSION CRLF
       *(general-header | request-header | entity header) CRLF
       [ message-body ]

Method: tells the server what operation to perform, e.g.,
  • GET: serve static or dynamic content
  • POST: serve dynamic content
  • OPTIONS: retrieve server and access capabilities

Request-URI: identifies the resource to manipulate
  • data file (HTML), executable file (CGI)

headers: parameterize the method
  • Accept-Language: en-us
  • User-Agent: Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)

message-body: text characters
HTTP/1.1 responses

Response = HTTP-Version SP Status-Code SP Reason-Phrase CRLF
   *(general-header | response-header | entity header)
CRLF
   [ message-body ]

Status code: 3-digit number
Reason-Phrase: explanation of status code
headers: parameterize the response
   • Date: Thu, 22 Jul 1999 23:42:18 GMT
   • Server: Apache/1.2.5 BSDI3.0-PHP/FI-2.0
   • Content-Type: text/html
message-body:
   • file
How servers interpret Request-URIs

GET / HTTP/1.1
  • resolves to home/html/index.html
  • action: retrieves index.html

GET /index.html HTTP/1.1
  • resolves to home/html/index.html
  • action: retrieves index.html

GET /foo.html HTTP/1.1
  • resolves to home/html/foo.html
  • action: retrieves foo.html

GET /cgi-bin/test.pl HTTP/1.1
  • resolves to home/cgi-bin/test.pl
  • action: runs test.pl

GET http://euro.ecom.cmu.edu/index.html HTTP/1.1
  • resolves to home/html/index.html
  • action: retrieves index.html
Example HTTP/1.1 conversation

kittyhawk> telnet euro.ecom.cmu.edu 80
Connected to euro.ecom.cmu.edu.
Escape character is '^]'.

Request sent by client
GET /test.html HTTP/1.1 ;request line
Host: euro.ecom.cmu.edu ;request hdr
CRLF

Response sent by server
HTTP/1.1 200 OK ;status line
Date: Thu, 22 Jul 1999 03:37:04 GMT ;response hdr
Server: Apache/1.3.3 Ben-SSL/1.28 (Unix)
Last-Modified: Thu, 22 Jul 1999 03:33:21 GMT
ETag: "48bb2-4f-37969101"
Accept-Ranges: bytes
Content-Length: 79
Content-Type: text/html
CRLF
<html> ;beginning of 79 byte message body (content)
<head><title>Test page</title></head>
<body><h1>Test page</h1>
</html>
OPTIONS method

Retrieves information about the server in general or resources on that server, without actually retrieving the resource.

Request URIs:

- if request URI = “*”, then the request is about the server in general
  - Is the server up?
  - Is it HTTP/1.1 compliant?
  - What brand of server?
  - What OS is it running?

- if request URI != “*”, then the request applies to the options that available when accessing that resource:
  - what methods can the client use to access the resource?
## OPTIONS (euro.ecom)

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host is a required header in HTTP/1.1 but not in HTTP/1.0</td>
<td></td>
</tr>
<tr>
<td>kittyhawk&gt; telnet euro.ecom.cmu.edu 80</td>
<td>OPTIONS * HTTP/1.1</td>
</tr>
<tr>
<td>Trying 128.2.218.2...</td>
<td>Host: euro.ecom.cmu.edu</td>
</tr>
<tr>
<td>Connected to euro.ecom.cmu.edu.</td>
<td></td>
</tr>
<tr>
<td>Escape character is '^]'.</td>
<td>CRLF</td>
</tr>
<tr>
<td>OPTIONS * HTTP/1.1</td>
<td>HTTP/1.1 200 OK</td>
</tr>
<tr>
<td>Host: euro.ecom.cmu.edu</td>
<td>Date: Thu, 22 Jul 1999 06:12:11 GMT</td>
</tr>
<tr>
<td>CRLF</td>
<td>Server: Apache/1.3.3 Ben-SSL/1.28 (Unix)</td>
</tr>
<tr>
<td>HTTP/1.1 200 OK</td>
<td>Content-Length: 0</td>
</tr>
<tr>
<td>Date: Thu, 22 Jul 1999 06:12:11 GMT</td>
<td>Allow: GET, HEAD, OPTIONS, TRACE</td>
</tr>
<tr>
<td>Server: Apache/1.3.3 Ben-SSL/1.28 (Unix)</td>
<td></td>
</tr>
<tr>
<td>Content-Length: 0</td>
<td></td>
</tr>
<tr>
<td>Allow: GET, HEAD, OPTIONS, TRACE</td>
<td></td>
</tr>
</tbody>
</table>
OPTIONS (euro.ecom)

kittyhawk> telnet euro.ecom.cmu.edu 80
Trying 128.2.218.2...
Connected to euro.ecom.cmu.edu.
Escape character is '^]'.

OPTIONS /cgi-bin/fixit.pl HTTP/1.1
Host: euro.ecom.cmu.edu
CRLF
HTTP/1.1 200 OK
Date: Thu, 22 Jul 1999 22:09:11 GMT
Server: Apache/1.3.3 Ben-SSL/1.28 (Unix)
Content-Length: 0
Allow: GET, HEAD, POST, OPTIONS, TRACE
## OPTIONS (microsoft.com)

```plaintext
kittyhawk> telnet microsoft.com 80
Trying 207.46.131.137...
Connected to microsoft.com.
Escape character is '^[a'.

<table>
<thead>
<tr>
<th>Request</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTIONS * HTTP/1.1</td>
<td></td>
</tr>
<tr>
<td>Host: microsoft.com</td>
<td></td>
</tr>
<tr>
<td>CRLF</td>
<td></td>
</tr>
<tr>
<td>HTTP/1.0 200 OK</td>
<td></td>
</tr>
<tr>
<td>Server: Microsoft-IIS/4.0</td>
<td></td>
</tr>
<tr>
<td>Date: Thu, 22 Jul 1999</td>
<td></td>
</tr>
<tr>
<td>Public: OPTIONS, TRACE, GET, HEAD, POST, PUT, DELETE</td>
<td></td>
</tr>
<tr>
<td>Content-Length: 0</td>
<td></td>
</tr>
</tbody>
</table>
```
OPTIONS (microsoft.com)

kittyhawk> telnet microsoft.com 80
Trying 207.46.130.150...
Connected to microsoft.com.
Escape character is '^[].

OPTIONS / HTTP/1.1
Host: microsoft.com
CRLF

HTTP/1.0 200 OK
Server: Microsoft-IIS/4.0
Date: Thu, 22 Jul 1999 22:13:46 GMT
Public: OPTIONS, TRACE, GET, HEAD, POST, PUT, DELETE
Allow: OPTIONS, TRACE, GET, HEAD
Content-Length: 0
OPTIONS (amazon.com)

kittyhawk> telnet amazon.com 80
Trying 208.216.182.15...
Connected to amazon.com.
Escape character is '^]'.

<table>
<thead>
<tr>
<th>OPTIONS / HTTP/1.0</th>
<th>Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRLF</td>
<td></td>
</tr>
<tr>
<td>HTTP/1.0 405 Because I felt like it.</td>
<td>Response</td>
</tr>
<tr>
<td>Server: Netscape-Commerce/1.12</td>
<td></td>
</tr>
<tr>
<td>Date: Thursday, 22-Jul-99 04:17:32 GMT</td>
<td></td>
</tr>
<tr>
<td>Allow: GET, POST</td>
<td></td>
</tr>
<tr>
<td>Content-type: text/plain</td>
<td></td>
</tr>
</tbody>
</table>
OPTIONS (etoys.com)

kittyhawk> telnet etoys.com 80
Trying 206.251.23.116...
Connected to etoys.com.
Escape character is '^]'.

OPTIONS * HTTP/1.1
Host: etoys.com
CRLF
HTTP/1.1 200 OK
Date: Thu, 22 Jul 1999 04:52:59 GMT
Server: Etoys Web server 1.0
Content-Length: 0
Allow: GET, HEAD, OPTIONS, TRACE
### OPTIONS (etoys.com)

```plaintext
kittyhawk> telnet etoys.com 80
Trying 206.251.23.116...
Connected to etoys.com.
Escape character is '^['].

OPTIONS /index.html HTTP/1.1
Host: etoys.com
CRLF

HTTP/1.1 200 OK
Date: Thu, 22 Jul 1999 22:04:43 GMT
Server: Etoys Web server 1.0
Content-Length: 0
Allow: GET, HEAD, POST, OPTIONS, TRACE
```
GET method

Retrieves the information identified by the request URI.

- static content (HTML file)
- dynamic content produced by CGI program
  - passes arguments to CGI program in URI

Can also act as a conditional retrieve when certain request headers are present:

- If-Modified-Since
- If-Unmodified-Since
- If-Match
- If-None-Match
- If-Range

Conditional GETs useful for caching
GET (euro.ecom.cmu.edu)

kittyhawk> telnet euro.ecom.cmu.edu 80
Connected to euro.ecom.cmu.edu.
Escape character is '^]'.

GET /test.html HTTP/1.1
Host: euro.ecom.cmu.edu

HTTP/1.1 200 OK
Date: Thu, 22 Jul 1999 03:37:04 GMT
Server: Apache/1.3.3 Ben-SSL/1.28 (Unix)
Last-Modified: Thu, 22 Jul 1999 03:33:21 GMT
ETag: "48bb2-4f-37969101"
Accept-Ranges: bytes
Content-Length: 79
Content-Type: text/html

<html>
<head><title>Test page</title></head>
<body><h1>Test page</h1>
</body>
</html>
GET request to euro.ecom (Internet Explorer browser)

GET /test.html HTTP/1.1
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)
Host: euro.ecom.cmu.edu
Connection: Keep-Alive
CRLF
GET response from euro.ecom

HTTP/1.1 200 OK
Date: Thu, 22 Jul 1999 04:02:15 GMT
Server: Apache/1.3.3 Ben-SSL/1.28 (Unix)
Last-Modified: Thu, 22 Jul 1999 03:33:21 GMT
ETag: "48bb2-4f-37969101"
Accept-Ranges: bytes
Content-Length: 79
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html
CRLF
<html>
<head><title>Test page</title></head>
<body>
<h1>Test page</h1>
</html>
GET request to euro.ecom (Netscape browser)

GET /test.html HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/4.06 [en] (Win98; I)
Host: euro.ecom.cmu.edu
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
       image/png, */*
Accept-Encoding: gzip
Accept-Language: en
Accept-Charset: iso-8859-1,* ,utf-8
CRLF
GET response from euro.ecom

HTTP/1.1 200 OK
Date: Thu, 22 Jul 1999 06:34:42 GMT
Server: Apache/1.3.3 Ben-SSL/1.28 (Unix)
Last-Modified: Thu, 22 Jul 1999 03:33:21 GMT
ETag: "48bb2-4f-37969101"
Accept-Ranges: bytes
Content-Length: 79
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

<html>
<head><title>Test page</title></head>
<body>
<h1>Test page</h1>
</body>
</html>
HEAD method

Returns same response header as a GET request would have...

But doesn’t actually carry out the request and returns no content
  • some servers don’t implement this properly
  • e.g., espn.com

Useful for applications that
  • check for valid and broken links in Web pages.
  • check Web pages for modifications.
HEAD (etrade.com)

kittyhawk> telnet etrade.com 80
Trying 198.93.32.75...
Connected to etrade.com.
Escape character is '^[].'

HEAD / HTTP/1.1
Host: etrade.com
CRLF

HTTP/1.0 200 OK
Server: Netscape-Enterprise/2.01-p100
Date: Fri, 23 Jul 1999 03:18:57 GMT
RequestStartUsec: 780328
RequestStartSec: 932699937
Accept-ranges: bytes
Last-modified: Tue, 20 Jul 1999 00:59:26 GMT
Content-length: 15370
Content-type: text/html
HEAD (espn.com)

kittyhawk> telnet espn.com 80
Trying 204.202.136.31...
Connected to espn.com.
Escape character is '^[].

HEAD / HTTP/1.1
Host: espn.com
CRLF

HTTP/1.1 301 Document Moved
Server: Microsoft-IIS/4.0
Date: Fri, 23 Jul 1999 03:22:32 GMT
Location: http://espn.go.com/
Content-Type: text/html
CRLF
<html>
Is now part of the http://espn.go.com service<br>
</html>

Modern browsers transparently connect to the new espn.go.com location.
POST method

Another technique for producing dynamic content.
Executes program identified in request URI (the CGI program).
Passes arguments to CGI program in the message body
  • unlike GET, which passes the arguments in the URI itself.
Responds with output of the CGI program.

Advantage over GET method:
  • unlimited argument size

Disadvantages:
  • more cumbersome
  • can’t serve static content
POST request

POST /cgi-bin/post.pl HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg,
    image/pjpeg, application/vnd.ms-excel, application/msword,
    application/vnd.ms-powerpoint, */*
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)
Host: kittyhawk.cmcl.cs.cmu.edu:8000
Content-Length: 25
CRLF
first=dave&last=ohallaron
POST response

HTTP/1.1 200 OK
Date: Fri, 23 Jul 1999 05:42:30 GMT
Server: Apache/1.3.4 (Unix)
Transfer-Encoding: chunked
Content-Type: text/html
CRLF
<p>first=dave&amp;last=ohallaron

Generated by server
Generated by CGI script
post.pl
TRACE, PUT, and DELETE methods

TRACE
- Returns contents of request header in response message body.
- HTTP’s version of an echo server.
- Useful for debugging.

PUT:
- add a URI to the server’s file system

DELETE
- delete a URI from the server’s file system