13. Applications and Mechanisms

Domain Name System (DNS)
WWW (World Wide Web)

Domain Name System (DNS) — Overview

- "Addresses" versus "names":
  - Addresses are fixed length
    - Tied to routing
    - Easy for computers to process
    - Often changed (*e.g.*, change of server machine, re-use as client)
    - Transport addresses 2-tupel (port, IP address)
  - Names are variable length
    - Mnemonic
    - Easy for humans to remember (*e.g.*, hosts, mail boxes, servers)
- Name Space:
  - Defines set of possible names
- DNS hides addresses from Internet applications

RFC 1034
RFC 1035

Name Space

- Consists of a set of names to value bindings
- Flat name space:
  - Centralized co-ordination to avoid overlaps
  - Not scalable for many hosts, max alphabet, max length
  - Requests addressed to a centralized directory not practicable
  - World-wide consistency of copies difficult to achieve
- Hierarchical name space:
  - Hierarchical names enable a structured name space
  - Localization of objects made simpler
  - Distribution of control (name assignments) made easier
  - Maintenance of names stored distributed

Domain Names in the Internet

- DNS defines syntax and rules for the delegation of a name authority and the implementation of a distributed system to map names onto addresses
- DNS-conforming names:
  - "Sub-domain. …. .Sub-domain.Domain": ifi.uzh.ch

Top-level domains (TLD):
uTLD, sTLD:
Various special ones
cf. next two slides

ccTLD:
cc 2 letter country code ISO 3166 (cc – country code) (*e.g.*, "CH" Switzerland, "DE" Germany, "UK" Great Britain)

Top-level Domains (Unsponsored: uTLD)

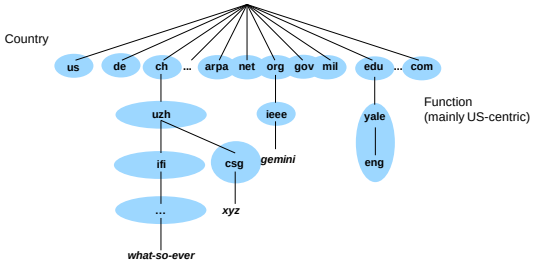
- **.arpa** arpanet Address and Routing Parameter Area
- **.biz** business for companies, open to all
- **.com** commercial for US companies, open to all
- **.info** information for information providers, open to all
- **.int** international for international organizations
- **.name** names for private persons
- **.net** network for network organization, open to all
- **.org** organization for non-commercial organizations, open to all
- **.pro** professionals for lawyer, tax accountants, doctors, engineers (only for U.S.A., Canada, Germany, U.K.)

Top-level Domains (Sponsored: sTLD)

- **.aero** aeronautics for SITA only
- **.asia** asia ICANN region Asia/Australia/Pacific
- **.cat** catalan for katalan language and culture
- **.coop** cooperatives, for .com LLC only
- **.edu** educational for "*post-secondary*" accredited US
- **.gov** government only for Government of the USA
- **.jobs** jobs only for companies with job offers
- **.mil** military for US military only
- **.mobi** mobile for services within mobile devices
- **.museum** museums, for association members only
- **.tel** telephone simplified calling of persons/companies
- **.travel** travel for travel organizations/companies
- **.post** postal for all post and logistic companies (plan)

DNS Name Space

- Hierarchical name space separates domains
- Zone represents a sub-tree of name tree

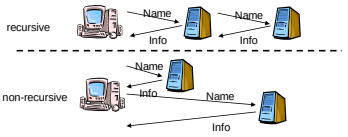


DNS Name Server (NS)

- Every zone maintains
  - One primary NS
  - Minimum two secondary NSs
- Operation:
  - Every NS knows a sub-part of the name space
  - Every NS knows those IP addresses of all NS in the directly attached sub-domains
  - Every NS caches already known addresses
  - Secondary NS maintain periodic updates (zone transfer) of local data bases (driven by the primary NS)

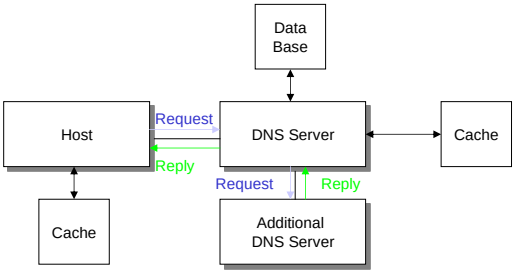
Name Resolution

- Name resolution starts at tree root
- Name resolution initiated by DNS resolver within application
- Requests:
  - Addressed to dedicated NS
  - Recursively addressing the full DNS



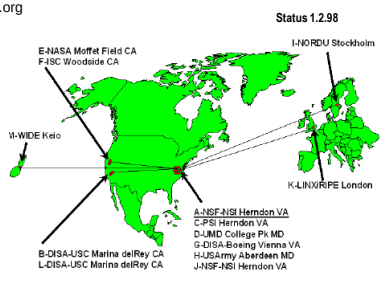
- For incoming request NS checks local sub-tree, if not able to resolve, request addressed to higher level NS
  - Resolver needs to know only "his" NS address (/etc/resolv.conf).

DNS Request



Root Server

- Location of DNS Root Servers
  - http://www.root-servers.org
  - World-wide 13 root servers
  - Root server maintain TLD (Top Level Domain) only



Name Resolution — Example

- DNS maps names onto information (*e.g.*, IP addresses)
  - **`www.ieee.org`** ⇨ **`199.172.136.14`**

MX Record

NS Records

A Records

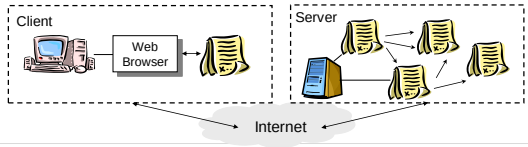
```
nslookup -q=mx ieee.org
Server:  sioux.telematik.informatik.uni-karlsruhe.de
Address:  129.13.35.73

ieee.org        preference = 0, mail exchanger = gemini.ieee.org
ieee.org        nameserver = auth01.ieee.org
ieee.org        nameserver = dns.ieee.org
ieee.org        nameserver = ns.uu.net
ieee.org        nameserver = krypton.ieee.org
ieee.org        nameserver = deepthought.ieee.org
gemini.ieee.org internet address = 199.172.136.14
auth01.ieee.org internet address = 199.172.136.2
dns.ieee.org    internet address = 199.172.136.6
ns.uu.net       internet address = 137.39.1.3
krypton.ieee.org        internet address = 199.172.136.2
deepthought.ieee.org    internet address = 199.172.136.6
```

World Wide Web (WWW)

- Goal:
  - Simple and world-wide exchange of documents
  - Proposal by Tim Berners-Lee (UK) at CERN, CH
- Implementation:
  - First prototype 1990
  - Graphical version (NEXTStep) and command-line based
- User interface:
  - WWW-client Mosaic by Marc Andreesen und Eric Bina (University of Illinois)
  - X-Windows systems
  - Source code publicly available ⇨ fast distribution
  - Marc Andreesen founded Netscape in 1995

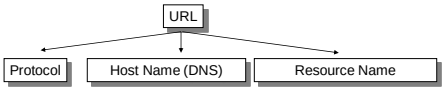
Client/Server-based WWW Architecture

- Client/Server-based architecture:
  - Web browser displaying hypertext documents/hypermedia objects
  - Hyperlinks enable navigation
- Solutions required for:
  - Addressing a web page uniquely
  - Transfer of a web page
  - Description of web page content, especially of hyperlinks



Addressing a Web Document

- A resource description identifies an object to be accessed within a server:
  - WWW: selected web page
  - FTP: to be transmitted file
  - Mail: Receiver of this message
- Uniform Resource Locator (URL):
  - Determines the resource's location as a name, the protocol to access this resource, and a name of the resource
    - E.g., http://www.ifi.uzh.ch/



Uniform Resource Locator (URL)

- Compact representation of location and access to Internet resources based on URLs:

```
<scheme>:<scheme-specific-part>
     ftp://<user>:<password>@<host>:<cwdl>..
           <cwdN>/<name>;type=<typecode>
    http://<host>:<port>/<path>?<searchpart>
  mailto:<rfc822-addr-spec>
    nntp://<host>:<port>/
           <newsgroup-name>/<article-number>
 telnet://<user>:<password>@<host>:<port>
    file://<host>/path
```

  - Web browser support a large number of protocols

WWW Application Protocol HTTP

- HTTP (Hypertext Transport Protocol):
  - Version 0.9/1.0 RFC1945, since January 1997 version 1.1 RFC2068
  - Mainly transfer of Web pages
- Characteristics:
  - ASCII-based application protocol (layer 7)
  - Utilized a reliable TCP connection at default port 80
  - Short-lived connection: HTTP server closes connection after a reply on a request of a client has been issued
- Sample commands:
  - **GET**: Request of a document
  - **HEAD**: Request of header information of this document
  - **POST**: Attachment of data to an existing document
  - **PUT**: Generation of a document

HTTP Messages

Request (Status) Line	General Header	Request (Response) Header	Entity Header	Entity Body

- Request Line:
  - Message type, requested resource, ...
- Status Line:
  - State information (error messages, HTTP version, ...)
- General Header:
  - Cache control, proxy information, ...
- Request Header:
  - Request- and client information, *e.g.*, acceptable media/coding, ...
- Response Header:
  - Response information (location, server software, retry information, ...)
- Entity Header:
  - Information on resources and body (Compression, date of modification, language, length)
- Entity Body:
  - Message body

Example of a HTTP Request/HTTP Reply

- Request:
  - <Command><URL> <Version>
  - Client requests non-cached, current version

HTTP Client

```
GET /index.html HTTP/1.1
Host: www.ifi.uzh.ch
Pragma: no-cache
....
```

- TCP connection already established:
  - Reply line
  - Reply, date, server
  - Encoding information
  - Content type

HTTP Server

```
HTTP/1.1 200 OK
Date: Fri, 24 Sep 1999 09:45:51 GMT
Server: Apache/1.3.6 (Unix)
Transfer-Encoding: chunked
Content-Type: text/html

<HTML>
Structured document (HTML-based)
</HTML>
```


Document Description Language HTML

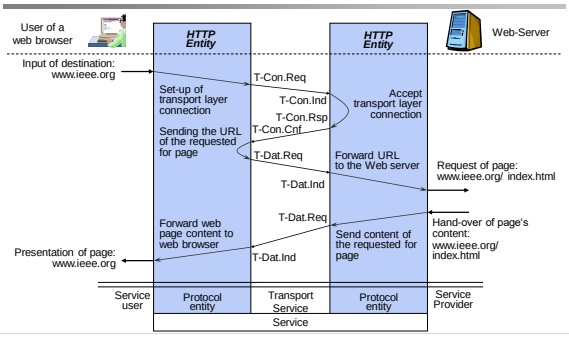
- HTML (Hypertext Markup Language):
  - HTML documents describe structured text documents
  - HTML tags describe the representation of document parts
    - HTML tags contained as clear text in document
    - *E.g.*, Bold Font
  - Documents are separated in header and body part:
    - Header contains general characteristics of document
    - Body contains "real" content:
      - *E.g.*, Title, subtitle, paragraphs, tables, items, lists
  - Text elements and further documents are concatenated by hyperlinks
  - Non-text parts, *e.g.*, graphs, figures, or video, are inter-linked
  - Advantage: Representation of client (browser) locally decided
- Standardization of version 4.0:
  - Integration of scripting languages and Cascading Style Sheets (CSS)

Example of a HTML Document

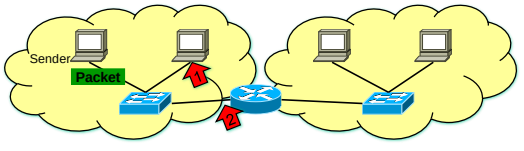
- Tags:
  - Start: <...>
  - End: </...>
- Structural information:
  - Paragraph: <p>
  - New line:
  - Header level *n*: <h*n*>
- Text options:
  - Italic <i>, bold , emphasized , strong
  - Character set ISO 8859-2 8 bit, ASCII as a subset
- References to other documents:
  - <A HREF="iis.unibw-muenchen.de/index.html">Index</A>

```
<html>
 <head>
 <title> Document Title </title>
 </head>
 <body>
 <p>This is a HTML document.</p>
 </body>
</html>
```


Surfin' … Surfin' … Surfin' the Net :-)



IP Forwarding – Hop-by-Hop



- Sending (forwarding) of an IP packet to an IP destination
  1. Destination in same subnetwork: Map IP destination address to MAC address and send directly
  2. Destination outside: Identify matching router's IP address, map to router's MAC address, and send to router
- Configuration required
  - Host IP address and network mask for each network interface
  - IP address of default router (and additional routers if required)

Forwarding/Routing Table

- Hosts and Routers maintain a single routing table
- Routing table identifies next IP address hop to be used
- Forwarding decision based on table entry with longest matching prefix of destination IP address
- Example:

Destination	Gateway	Netif
Default/0	10.5.16.254	en0
Default/0	10.5.222.254	en1
10.5.16/21	link#4	en0
10.5.222/23	link#5	en1
10.5.21.9/32	127.0.0.1	lo0
10.5.222.14/32	127.0.0.1	lo0
127/8	127.0.0.1	lo0
127.0.0.1/32	127.0.0.1	lo0
169.254/16	link#4	en0