Domain Names

- Naming of Resources
- Problems of Internet's IP focus
  - IP addresses (123.25.33.44) difficult to remember
  - Event worse for IPv6
  - IPs may change
- Name resolution
  - Host name (www.myserver.com) -> IP
- Back resolution / reverse lookup
  - IP -> Host name
- Additional information about hosts

Domain Name System

- Internationalized Domain Names

Agenda

- Domain Names
- Domain Name System
- Internationalized Domain Names
HOSTS.TXT

- Original naming facility
- RFC 810, later 952
- Maintained by SRI NIC (Network Information Center)
- Stores address mappings
- IP to Domains

Disadvantage:
- Load on central server
- Bandwidth for distribution proportional to $N^2$
  - $N$=Number of hosts
- Name clashes
- Simultaneous updates

HOSTS.TXT - Example

NETWORK: 10.0.0.0 : ARPANET :
HOST: 10.2.0.11: SU-TIP, FELT-TIP :::

Today different format:

ipAddress localhost aliases
127.0.0.1 localhost
192.168.0.1 bar.mydomain.org bar

Domain Name System (DNS) - Design Goals

- Consistent name space
- Distributed by design
  - Multiple servers
  - Hierarchically
    - Tree structure
      - Organizations may maintain their own servers
- Names used to get
  - Host addresses
  - Mailbox Data
- Other, yet undefined information
- Access to data critical
- Instantaneous updates less important

Domain Name System - Structure

Root Domain

Generic TopLevelDomains
- gTLDs

Country-Code TLDs
- ccTLDs

Organizations
- Introduce additional Inofficial TLDs

Alternative Root Domains

Root domain has always an empty label!

Disadvantage:
- Not usable for normal Internet users
- Conflicts

Family, .med, .ngo, .men, .chem
DNS - Elements

- Resolvers
  - Programs/Routines that extract information from Name Servers
- Name Servers
  - Hold information about the domain tree's structure
  - May cache any information of the whole domain tree
  - In general holds information about a subset
  - Name server is an AUTHORITY for this subset
  - Authoritative information organized as ZONES

Resolver

- Client part of DNS
  - triggers DNS queries
  - Parts of the OS (or libraries)
    - Convert names to IP addresses
    - IP addresses to names

Resolv.conf

- Unix OS
  - In Lab environment in /etc/resolv.conf
- Configuration how to build a name
- Configuration options
  - nameserver ip-address
    - Which nameservers (max 3) shall be used
  - domain localdomainname
  - search domainname1...
    - extends names without . with names in searchlist
    - Mutual exclusive to domain keyword

Resolv.conf - Example

```
domain mydomain.org
nameserver 128.131.171.77
nameserver 128.131.171.212
```

or

```
search infosys.org dslab.org
nameserver 128.131.171.77
```
**Resolver**
- **Iterativ**
  - Queries the first (top-level) nameserver
  - Based on the result the next nameserver is queried
- **Rekursiv**
  - Asks the nameserver to do the whole query for the resolver
- Resolvers located at both client and server
- (Verteilte Systeme, VO)

**Name Server Configuration**
- **Domain**
  - Contains whole DNS subspace under a treenode
- **Zone**
  - Subdomains may be in their own zones
  - Primary/Master DNS servers have authority
  - Secondary/Slaves servers have copies of information
  - Zone files contain info about zone in resource records

**Domain vs Zone**

**Name Servers**
- Repositories that make up the domain database
- Primary task of name servers
  - Answer queries using data in its zones
  - Answer created using only local data
  - Or Referral to other name servers
  - Answers are resource records
- Name server typically supports one/more zones
- Allows partitioning at points where an organization wants control
- Root.hint already installed
  - Points to root nameservers
DNS Resource Records

- Resource Record (RR)
  - Different types

Syntax

```
Owner-domain Type class TTL RDATA
```

<table>
<thead>
<tr>
<th>Domain Name</th>
<th>Where RR is found</th>
<th>May be omitted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IN ... Internet</th>
<th>Time To Live (caching)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SOA Resource Record

- Defines Start Of an Authority for a zone

Domain IN SOA primmastersrv contactemail {
  serialnumber; Serial number
  refreshtime; how often try to refresh
  retrytime; when to retry
  expiretime; when to abandon zone info
  negativecaching; how long cache negative answers
}

Zone Transfers

- Multiple name servers
- More robust
- Additional servers usually slave name servers
- Where is zone information?
  - Master server zone files
  - Slave server gets from another server
- When is data updated?
  - Controlled with numbers in SOA record
    - After Refreshtime; slave checks if serial number has changed
    - If no connection was possible after refreshtime
      - wait refreshtime and try refresh again
    - If no connection was possible after expiretime
      - Declare zone as invalid
      - negativecaching; how long cache negative answers
- DNS Notify
  - RFC 1996
  - Master server triggers update to slaves when serial number has changed

SOA Example

```
mydom.org. IN SOA mastersrv.mydom.org. dnsadmin.mail.mydom.org. {
  2006033001; serial number
  3h; refresh
  1h; retry
  1w; expire
  1h; negative Caching TTL
}
```
<table>
<thead>
<tr>
<th>RR Type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>NS RR - Nameserver</td>
<td>Defines a nameserver for this zone</td>
<td>Zone IN A Nameserver subdom1.org. IN NS namesrv.myorg.org.</td>
</tr>
<tr>
<td>A RR - Address</td>
<td>Maps a name to an IPv4 address</td>
<td>Hostname IN A ipAddress MyHost IN A 1.2.3.4</td>
</tr>
<tr>
<td>CNAME RR - Assign alias</td>
<td>Each host has a canonical name defined with an A record. CNAME allows definition of alias without introducing additional host with same IP.</td>
<td>Aliasname IN CNAME canonicalName www IN CNAME myHost. Some applications/resolvers do not work correctly with Aliases!</td>
</tr>
<tr>
<td>PTR RR - Point to</td>
<td>Points to another part of the domain space.</td>
<td>Hostname IN PTR Hostname anotherHost.org. IN PTR myHost.org.</td>
</tr>
</tbody>
</table>
PTR RR

How to do a Reverse lookup

Parallel name space
- .arpa tld
- 4.3.2.1.in-addr.arpa.
- For 1.2.3.4

Example
4.3.2.1.in-addr.arpa. IN PTR myHost

Why is order of IP Address reversed?
- Hierarchical structure of IP addresses

AAAA – ipv6 Address

- Maps a name to an IPv6 address
  RFC 1886
- Example:
  Hostname IN AAAA ipv6 Address
  ip6Host.x.y. IN AAAA 0123::ab:1234

IPv6 Reverse Mapping

- New domain ip6.int
- Like .in-addr.arpa.
- Each subdomain represents 4 bits of IPv6 address, ie 1 hex character
  - No shortcut for 0s
- Example
  4.3.2.1.b.a.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.3.2.1.0.ip6.int. IN PTR ip6host
A6 – IPv6

- RFC 2874
- Another form of specifying IPv6 addresses
- Chain of A6 records
- Only those parts need to be specified that are controlled by the nameserver
- For the remaining bits another A6 entry is consulted, probably at another nameserver
- Supports chaining of IPv6 addresses
- **Example**
  ip6Host IN A6 64 ::ab:1234 parentnet.org.

DNAME – IPv6 Reverse Mapping

- **Bitstring-Label**
  - Parts of IPv6 Addresses
  - |0x1230000000000000000000000ab1234|
  - |0x123/16| means bitstring with 16 significant bit
- **DNAME**
  - |0x1234/16| IN DNAME ip6.m.net.
  - At each step part of the bitstring will be replaced

DNAME – Nameserver entries

- **Root nameserver:**
  - |0x123/16| IN DNAME ip6.m.net.
- **ip6.m.net:**
  - |0x00000000/32| IN DNAME x.y.
  - x.y:
    - |0x0000000000000000ab1234/80| IN PTR ip6Host.x.y.

DNAME – Resolving Example

- Resolving |0x1230000000000000000000000ab1234|
- Query at root nameserver:
  - |0x1230000000000000000000000ab1234| ip6.arpa. IN CNAME
- **Returns:**
  - |0x1230000000000000000000000ab1234| ip6.m.net.
- Query to ip6.m.net:
  - |0x0000000000000000ab1234| IN CNAME
- **Returns:**
  - |0x00000000000000ab1234| x.y.
- Query to x.y:
  - Finds |0x00000000000000ab1234| and returns searched name.
Applications for Reverse Lookup

- Spam Prevention
  - Almost all spam emails contain forged sender addresses
  - Email sender address may easily be forged
    - It's just text!
  - Reverse Lookup off sender mail address and server mail address

DNS Security / TSIG

- Transaction Signatures (TSIG)
  - Secret Key Transaction Authentication for DNS (TSIG)
  - RFC 2845
  - Authentication of DNS partners
  - Data Integrity
  - Secret Key
    - Known by involved DNS Servers
  - Used in zone transfers, dynamic updates
  - Principle
    - MD5 hash of each DNS packet
    - Stored in a TSIG Resource Record
    - NO corresponding RR in any zone file!
    - Hash verified by receiver

DNS Security / DNSSEC

- Authentication of DNS partners
- Data integrity
- Public key cryptosystem
  - KEY resource record for public key
- Private key used to digitally sign RRs
  - Creates SIG RR
  - SIG-RR is delivered in each DNS transaction

Dynamic DNS

- DNS based on static database
- Dynamic Update
  - RFC 2136
  - Allows updates of DNS from outside
    - Without intervention of administrator
- Allows Dynamic DNS (DDNS)
  - Clients have not a static IP
    - But require static name
    - After startup Dynamic Update is done on DNS server
  - Updates incrementally stored in journal files
  - Requires either Access Control Lists or TSIG
Server & Clients

- DNS Server
  - Bind 8 & 9
  - Berkely Internet Name Demon
  - Djbdns
  - Daniel J. Bernstein DNS
  - More secure than Bind
  - MS DNS included in Windows Server OS
- Client tools
  - nslookup
  - dig

DNS Protocol

- UDP & TCP port 53
- Header identical for query and answer
- Flags: query/response, authoritative answer recursion desired, recursion available

<table>
<thead>
<tr>
<th>Identification - 16 bit</th>
<th>flags - 16 bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of questions - 16 bit</td>
<td>Number answers - 16 bit</td>
</tr>
<tr>
<td>Number of authority RRs - 16 bit</td>
<td>Number of additional RRs - 16 bit</td>
</tr>
</tbody>
</table>

Internationalized Domain Names

- Internationalized Domain Name
  - Contains potentially non-ASCII characters
  - Eg. Österreich.at
  - Allows country-specific domain names
  - Umlaute: ö,ü,ä
  - Greek, Cyrilic, Japanese, Chinese Symbols
- Internationalizing Domain Names in Applications
  - RFC 3490
  - Based on Unicode
  - Conversion done by the application
  - DNS not involved

Internationalized Domain Names Example

- Example
  - www.Österreich.at
    - Split into individual labels
    - Österreich (has non-US ASCII characters)
    - Perform Nameprep algorithm
      - RFC 3491 based on Stringprep 3454
    - Normalizes string
    - Österreich
    - Perform Punycode algorithm (RFC 3492)
      - Removes Special Characters
      - Encodes symbol and position
      - sterreich-z7a
    - Prepend ACE label (ASCII Compatible Encoding): xn--
- Result: www.xn--sterreich-z7a.at
Internationalized Domain Names

- Browser support
  - Mozilla > 1.4
  - Netscape 7.1
  - Opera 7
  - IE < 7 only with plugin
  - IE 7
- Conversion Tools
  - Search for Punycode or IDN Converter