Domain Name System

Lenuta Alboaie (adria@info.uaic.ro)
Andrei Panu (andrei.panu@info.uaic.ro)

Content

- Domain Name System (DNS)
  - Characterization
  - Organization
  - Configuration
  - Commands, Primitives
  - IDN

DNS

- IP Addresses (e.g., 85.122.23.145, 2001:0db8:0001:0000:0000:0ab9:C0A8:0102) are difficult to remember
- A **domain name system** is used to map IP addresses to domain names and vice versa
- Domain names are organized in hierarchies

- RFC 1034, 1035, 1123, 2181

DNS| organization

- Initial: **/etc/hosts** – pairs (name, IP)
  - Scalability problems

- Actual: DNS consists of a hierarchical domain scheme and a distributed database system to implement this naming scheme

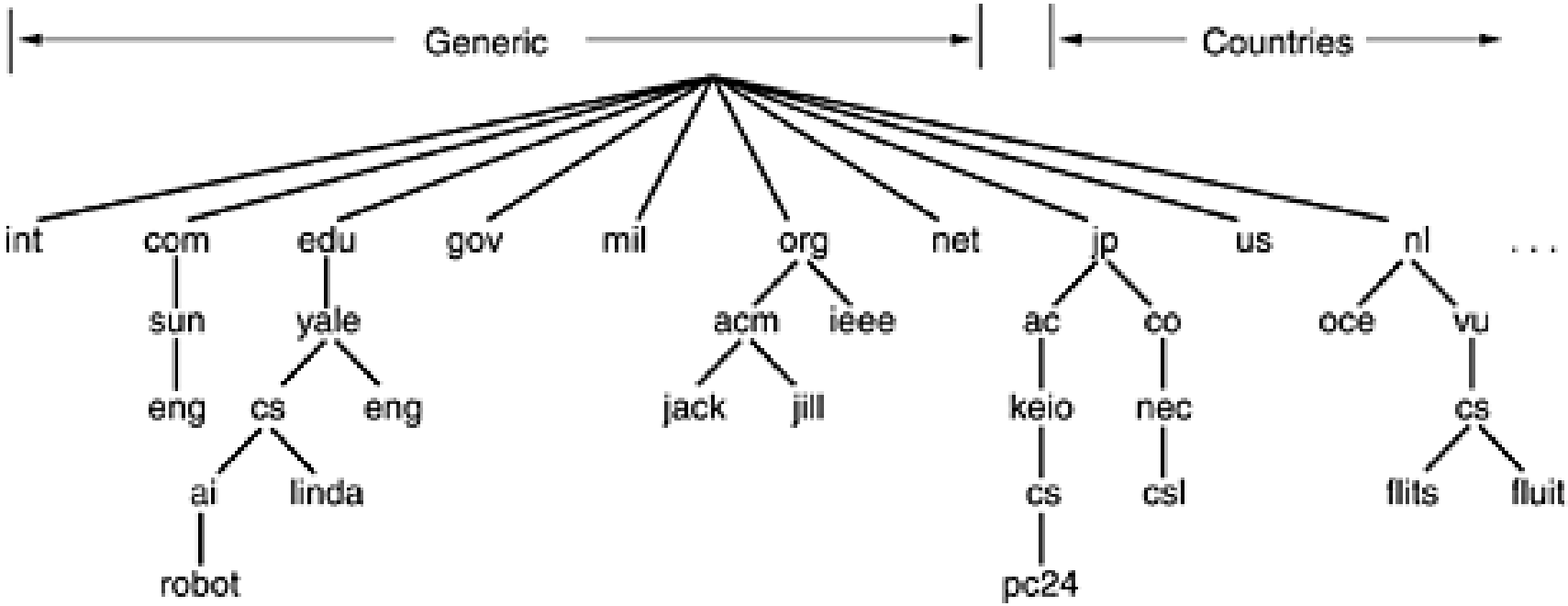


Figure. A portion of the domain names space on the Internet

[Computer Networks, 2003
Andrew S. Tanenbaum]

DNS| domain types

- **Primary** (*Top Level Domains* – TLD)
  - For Internet Infrastructure – one domain: **.arpa** ARPA (Address and Routing Parameter Area)
    - *"Changes to the .arpa zone must be coordinated manually with IANA"*
  - State (*ccTLD*) – states code: .ro, .fr, .jp, …
  - IDN ccTLD (*Internationalized Country Code Top-Level Domains*)

    http://例子.测试 http://*example.test*
  - Generics: .biz, .com, .info, .name, .net, .org, .pro
  - Sponsored: .aero, .edu, .gov, .int, .jobs, .mil, .tel
  - Reserved: .example, .invalid, .localhost, .test
  - Pseudo-domains: .bitnet, .local, .root, .uucp etc.

https://www.iana.org/domains/root/db/

DNS| domain types

https://www.iana.org/domains/root/db

Domain Names

- Overview
- **Root Zone Management**
  - Overview
  - **Root Database**
  - Hint and Zone Files
  - Change Requests
  - Instructions & Guides
  - Root Servers
- .INT Registry
- .ARPA Registry
- IDN Practices Repository
- Root Key Signing Key (DNSSEC)
- Reserved Domains

Root Zone Database

The Root Zone Database represents the delegation details of top-level domains, including gTLDs such as .com, and country-code TLDs such as .uk. As the manager of the DNS root zone, we are responsible for coordinating these delegations in accordance with our policies and procedures.

Much of this data is also available via the WHOIS protocol at whois.iana.org.

DOMAIN	TYPE	TLD MANAGER
.aaa	generic	American Automobile Association, Inc.
.aarp	generic	AARP
.abarth	generic	Fiat Chrysler Automobiles N.V.
.abb	generic	ABB Ltd
.abbott	generic	Abbott Laboratories, Inc.
.abbvie	generic	AbbVie Inc.
.abc	generic	Disney Enterprises, Inc.
.able	generic	Able Inc.
.abogado	generic	Minds + Machines Group Limited
.abudhabi	generic	Abu Dhabi Systems and Information Centre
.ac	country-code	Network Information Center (AC Domain Registry) c/o Cable and Wireless (Ascension Island)
.academy	generic	Binky Moon, LLC
.accenture	generic	Accenture plc

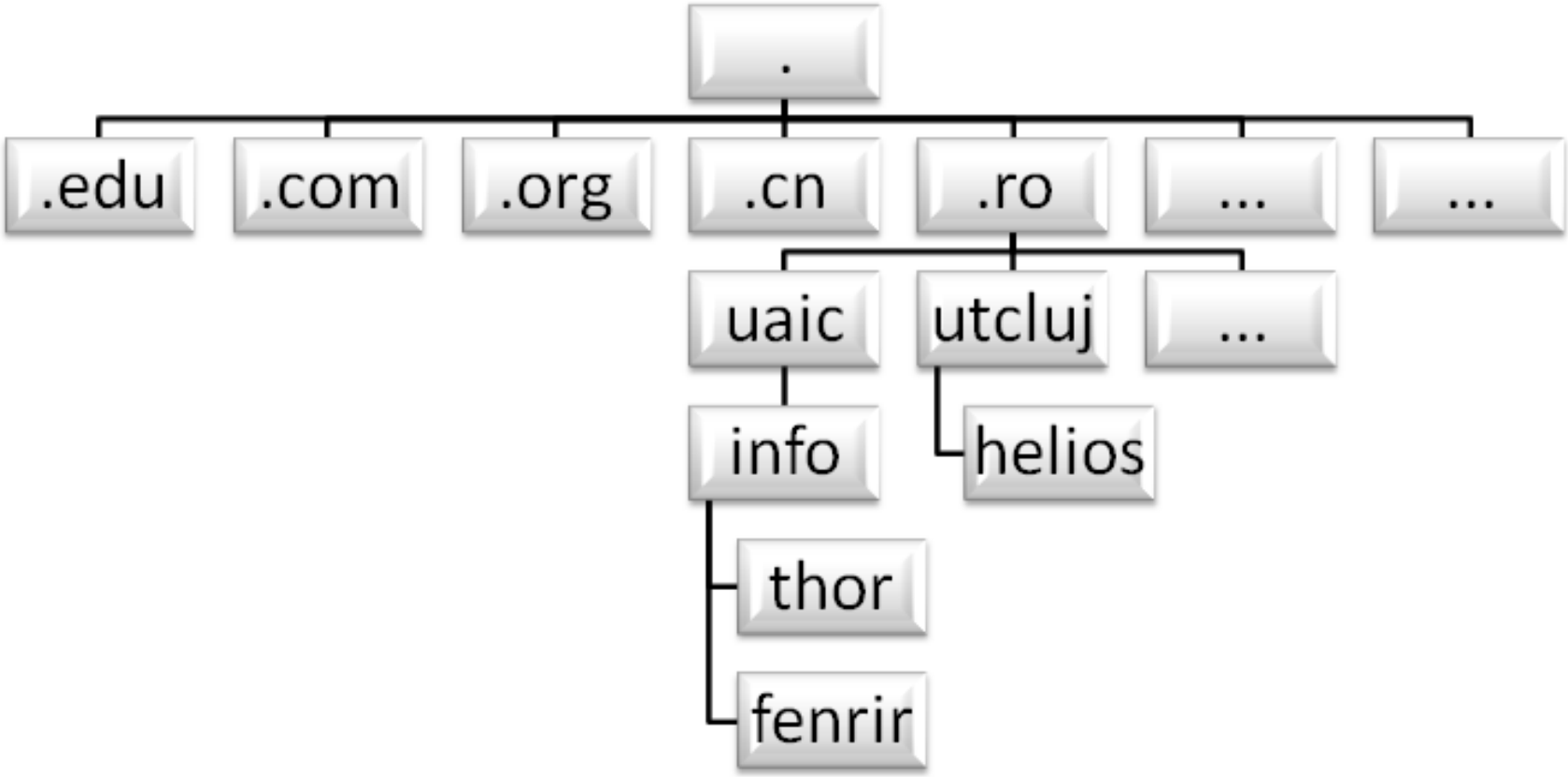
DNS| domain types

- Domain name
  - Sub-tree of the domain tree
  - The physical topology is not taken into consideration

- **Sub-domains**:
  - Full path name cannot exceed 255 characters
- **Name of computers** (hosts)

DNS

- Example:



DNS| organization

- Rules to allocate to domain names:
  - Each domain controls how its subdomains are assigned
  - To create a new subdomain, permissions are requested from the upper domain (each domain will have an authority at a certain level)
  - The naming domain is performed in respect to the organizational boundaries, not those of networks
  - A certain level of hierarchy can be controlled by multiple servers

DNS| organization

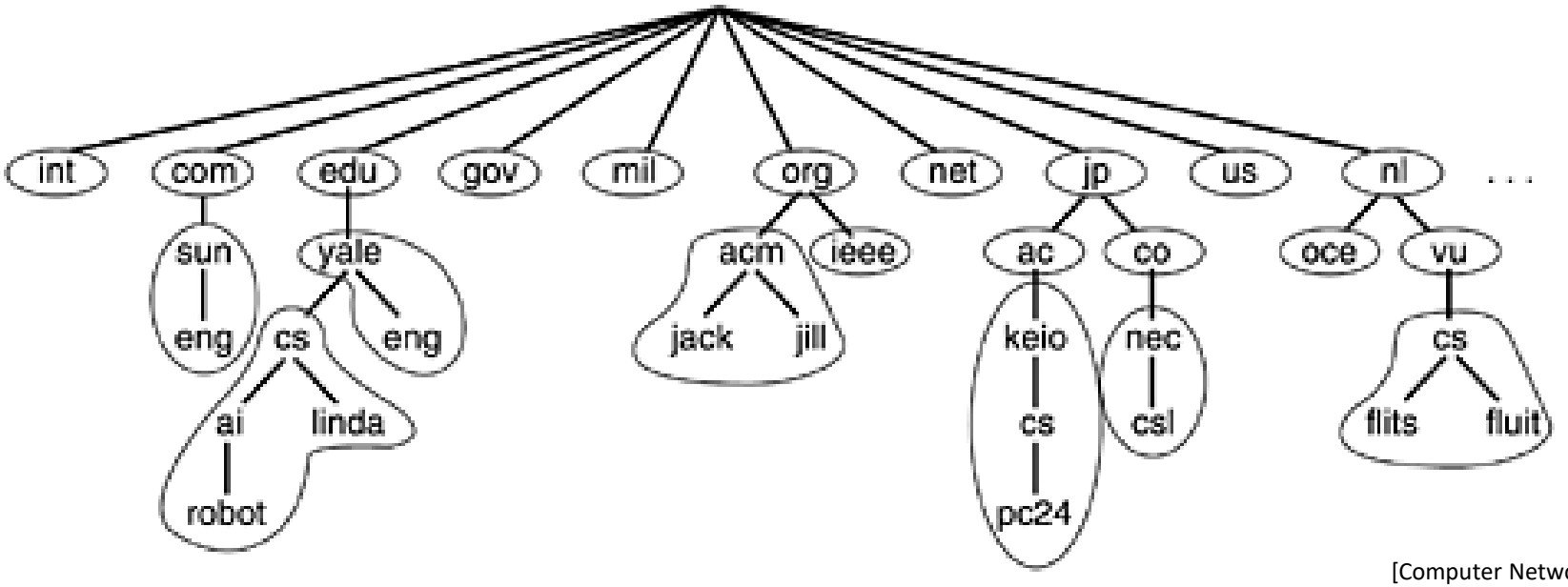
- *Name servers*
  - Theoretically, a single name server can contain the entire DNS database and can respond to all requests
  - Problems: loading and "*single point of failure*"

  - DNS name space is divided into non-overlapping areas

DNS| organization

- Name servers

  Example: A possible division of DNS namespace in areas



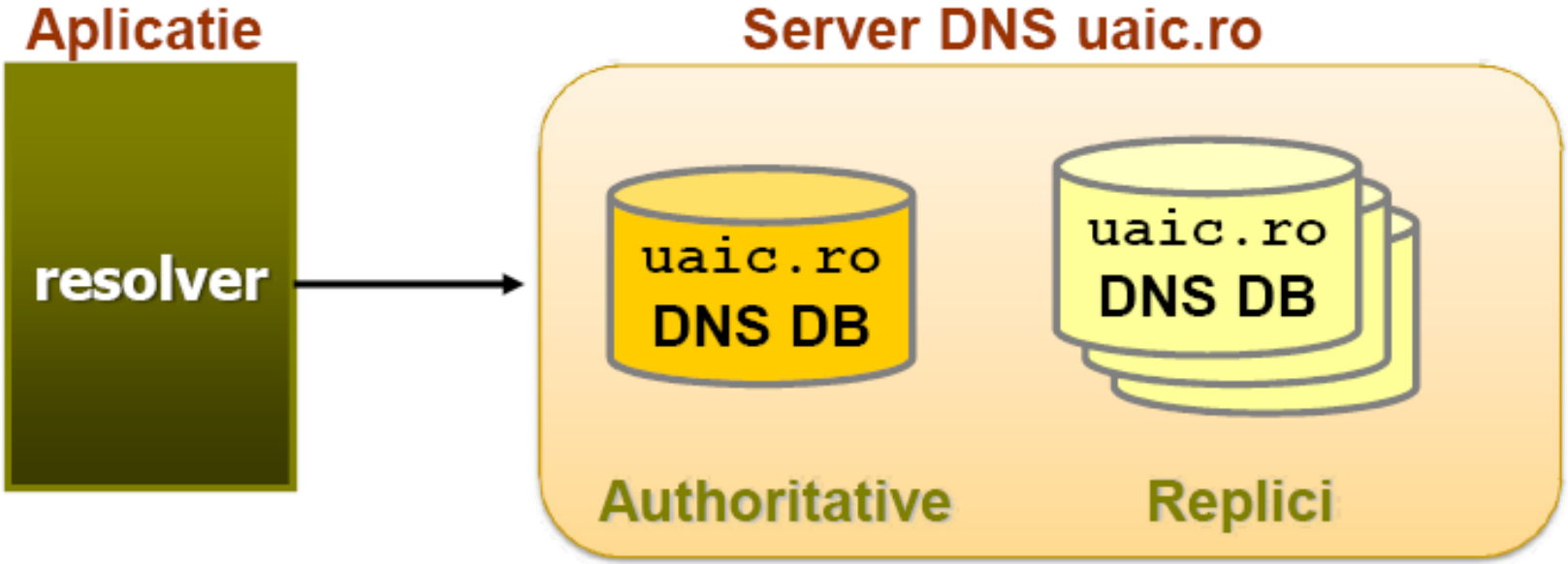
[Computer Networks, 2003 Andrew S. Tanenbaum]

DNS| organization

- Name servers
  - There is a *primary/authoritative name server* that manages a certain domain, and possibly, more secondary servers contain replicated databases
  - TCP is used for DNS replication
  - UDP is used for queries (*lookups*)

DNS| organization

- DNS Client
  - Called ***resolver***, sends an UDP packet to a DNS server; the server seeks the name and returns the IP address



[Retele de calculatoare – curs 2007-2008, Sabin Buraga]

DNS| organization

- Example of name server implementations: **BIND** (Berkeley Internet Name Domain), MSDNS, PowerDNS etc.

- As interactive resolvers (clients), the following commands can be used: **nslookup**, **host**, and **dig**.

DNS| queries

- Queries:
  - **Recursive** – if a DNS server does not know the address for the requested name, then it will query another DNS server
  - **Incremental** – if the DNS server does not know how to respond, it will return an error and another DNS server address (also called *referral*) that may know the answer to the query

[http://technet.microsoft.com/en-us/library/cc775637%28v=ws.10%29.aspx]

DNS| queries

- Each domain is associate with a set of resource records (***resource record*** – **RR**)
- The mechanism:
  - The request: the *resolver* sends a domain name
  - The response: the resource records associated with that name (stored in DNS database)

DNS creates a correspondence between the domain names and the resource records

DNS| queries

- RR general form:

  Domain_Name Time_to_live Type Class Value

  Domain name – specifies the field covered by this registration

  Time-to-live – gives an indication of how stable the recording is

DNS| queries

Type - specifies the registration type

- **SOA** (*Start Of Authority*) : the current domain, administrator e-mail address, etc.
- **A** – host IP
- **MX** (mail exchangers) – specifies the domain name ready to accept mail for the specified domain
- **CNAME** (*Canonical Name*) – allows creation of pseudonyms
- **PTR** (Pointer) – Alias for IP address
- **HINFO** – allows to find: computer type, operating system type corresponding to the domain
- **TXT**: uninterpreted text (comments)

DNS| queries

Class: for Internet, the value is IN

Value: this field can be a number, a domain name or an ASCII string; the semantics depend on registration

Example of DNS resource records

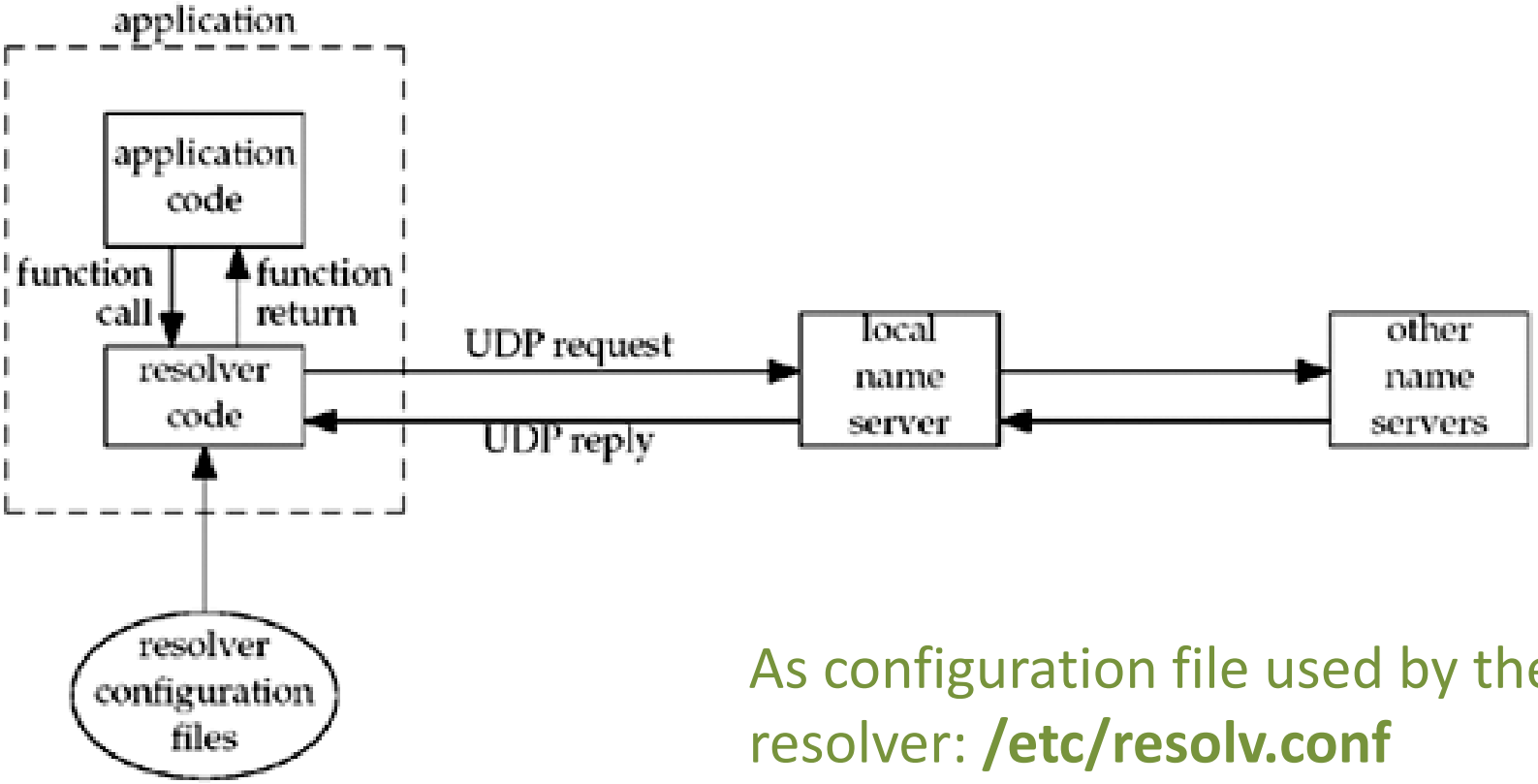
Type	Meaning	Value
SOA	Start of Authority	Parameters for this zone
A	IP address of a host	32-Bit integer
MX	Mail exchange	Priority, domain willing to accept e-mail
NS	Name Server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
HINFO	Host description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

DNS| configuration

- Example of a file containing a DNS zone specification

```
; Zone file for axiologic.ro
;
; The full zone file
;
$TTL 3D
@       IN      SOA     ns1.axiologic.ro. abss.axiologic.ro. (
                        2007050103        ; serial, todays date + todays serial #
                        14400             ; refresh, seconds
                        7200              ; retry, seconds
                        1209600           ; expire, seconds
                        1D )              ; minimum, seconds
;
@                NS      ns1.axiologic.ro.              ; Inet Address of name server
@                NS      ns2.axiologic.ro.              ; Inet Address of name server
@                MX      5 mailx.axiologic.ro.     ; Primary Mail Exchanger
;
localhost       A       127.0.0.1
axiologic.ro.   A       72.249.105.153
www             A       72.249.105.153
mailx           CNAME   axiologic.net.
mail            A       207.210.101.144
ftp             A       72.249.105.153
axiologic.ro. IN TXT "v=spf1 mx mx:mailx.axiologic.ro. ~all"
ns1             A       207.210.101.144
ns2             A       207.210.101.216
~
(END)
```

DNS| clients, resolvers, servers



As configuration file used by the resolver: **/etc/resolv.conf**

[Unix Network Programming, R. Stevens
B. Fenner, A. Rudoff - 2003

DNS| configuration

- /etc/resolv.conf file:

```
[adria@thor ~] $ cat /etc/resolv.conf
domain info.uaic.ro
search info.uaic.ro
nameserver 85.122.16.1
nameserver 85.122.16.4
[adria@thor ~] $
```

DNS| reverse queries

- Problem:
  - If we have an address, which will be its symbolic name? (*reverse DNS resolution* or *reverse DNS lookup*)

Example:

1)

```
[adria@ns1 ~]$ host 85.122.23.1
1.23.122.85.in-addr.arpa domain name pointer thor.info.uaic.ro.
[adria@ns1 ~]$
```

2) 2001:db8::567:89ab

b.a.9.8.7.6.5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa

DNS| optimizations

Spatial proximity: local servers will be queried more often than others at the distance

Temporal proximity: if a set of fields are referenced repeatedly, then the DNS caching mechanism is used

For each DNS entry a TTL (*time to live*) value is set

Replication is also used (multiple servers, multiple root servers) – the nearest (geographically) server will be interrogated

DNS| commands

As interactive resolvers, we can use:

- **nslookup**
- **dig**
- **host**
- **whois**
- …

DNS| nslookup

Usage examples:

- **nslookup www.info.uaic.ro**

- Returns a RR of type A, using the local DNS server

```
[adria@thor ~] $ nslookup www.info.uaic.ro
Server:         85.122.16.1
Address:        85.122.16.1#53

www.info.uaic.ro        canonical name = vidar.info.uaic.ro.
Name:   vidar.info.uaic.ro
Address: 85.122.23.146
```

Host Lookup

- **nslookup 85.122.23.1**

- Returns a RR of type PTR for 85.122.23.1 in *in-addr.arpa* domain hierarchy

```
[adria@thor ~] $ nslookup 85.122.23.1
Server:         85.122.16.1
Address:        85.122.16.1#53

1.23.122.85.in-addr.arpa        name = thor.info.uaic.ro.
```

Reverse IP Lookup

[http://www.zytrax.com/books/dns/ch3/]

DNS| nslookup

Usage examples:

- **nslookup www.axiologic.ro**

- Returns a RR of type A using a specified DNS server

```
adria@thor:~$ nslookup www.axiologic.ro 207.210.101.144
Server:         207.210.101.144
Address:        207.210.101.144#53

Name:   www.axiologic.ro
Address: 72.249.105.153
```

Host Lookup

- **man nslookup**

DNS| dig

dig – a tool more powerful than nslookup

Usage example:

- **dig www.info.uaic.ro A**

```
[adria@thor ~] $ dig www.info.uaic.ro A

; <<>> DiG 9.6-ESV-R4 <<>> www.info.uaic.ro A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19336
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 4

;; QUESTION SECTION:
;www.info.uaic.ro.              IN      A

;; ANSWER SECTION:
www.info.uaic.ro.       86400   IN      CNAME   vidar.info.uaic.ro.
vidar.info.uaic.ro.     86400   IN      A       85.122.23.146

;; AUTHORITY SECTION:
info.uaic.ro.           86400   IN      NS      orion.uaic.ro.
info.uaic.ro.           86400   IN      NS      onix.uaic.ro.
info.uaic.ro.           86400   IN      NS      ns.iasi.roedu.net.

;; ADDITIONAL SECTION:
ns.iasi.roedu.net.      86400   IN      A       192.129.4.100
ns.iasi.roedu.net.      86400   IN      AAAA    2001:b30:1:100::100
onix.uaic.ro.           86400   IN      A       85.122.16.4
orion.uaic.ro.          86400   IN      A       85.122.16.1

;; Query time: 1 msec
;; SERVER: 85.122.16.1#53(85.122.16.1)
;; WHEN: Mon Nov 14 11:57:27 2011
;; MSG SIZE  rcvd: 216
```

DNS| host

host

Usage example:

```
adria@thor:~$ host 128.30.52.45
45.52.30.128.in-addr.arpa domain name pointer dolph.w3.org.
```

DNS| whois

whois ibm.com

```
Registrant:
International Business Machines Corporation
   New Orchard Road
   Armonk, NY 10504
   US

   Domain Name: IBM.COM

   ----------------------------------------------------------------------
   Promote your business to millions of viewers for only $1 a month
   Learn how you can get an Enhanced Business Listing here for your domain nam
   Learn more at http://www.NetworkSolutions.com/
   ----------------------------------------------------------------------

   Administrative Contact:
      IBM DNS Admin              dnsadm@us.ibm.com
      IBM Corporation
      New Orchard Road
      Armonk, NY 10504
      US
      +1.9147654227 fax: +1.9147654370

   Technical Contact:
      IBM Corporation            ipreg@us.ibm.com
      New Orchard Road
      Armonk, NY 10504
      US
      +1.9192544441 fax: +1.9147654370


   Record expires on 20-Mar-2018.
   Record created on 19-Mar-1986.
   Database last updated on 8-Nov-2010 04:12:22 EST.

   Domain servers in listed order:

   INTERNET-SERVER.ZURICH.IBM.COM 195.176.20.204
   NS.WATSON.IBM.COM              129.34.20.80
   NS.ALMADEN.IBM.COM             198.4.83.35
   NS.AUSTIN.IBM.COM              192.35.232.34


adria@thor:~$
```

DNS| primitives

- We do not have to develop a resolver in order to find out the IP address of a host
- Main functions:
  - **gethostbyname(); getaddrinfo();**
  - **gethostbyaddr() ; getnameinfo();**
- For some operating systems (e.g., Solaris), when compiling, we must specify the *nsl (Name Server Library)* library: gcc … -lnsl

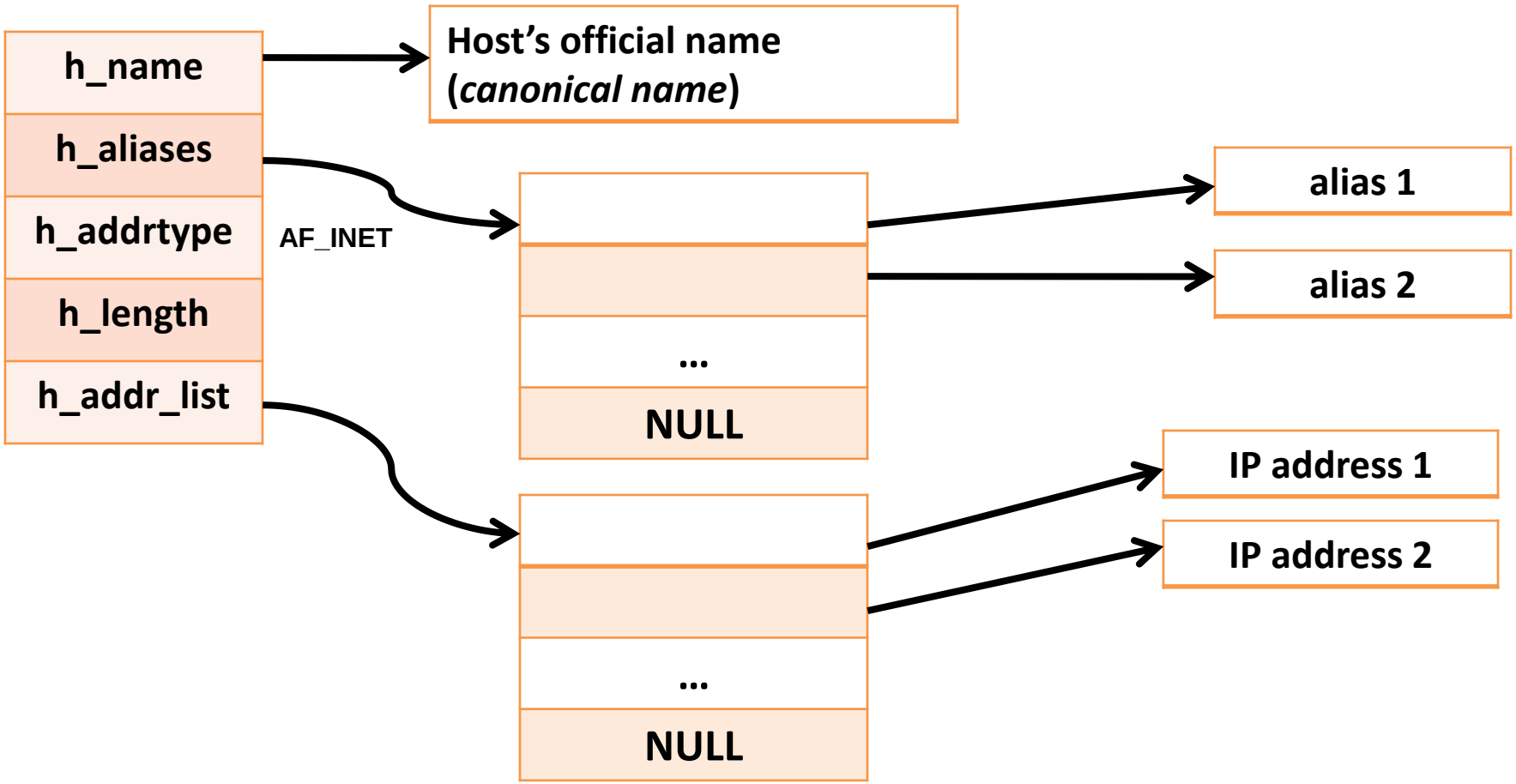
DNS| primitives

One of the used structures: **hostent**

```
struct hostent {
    char *h_name;       /* official name (canonical) */
    char **h_aliases;   /* aliases */
    int h_addrtype;     /* AF_INET */
    int h_length;       /* address length: 4 or 6 */
    char **h_addr_list; /*pointers to IP addrresses */
};
```

DNS| primitives

hostent structure:



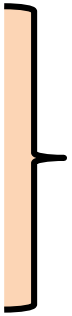
DNS| gethostbyname()

```
#include <netdb.h>
struct hostent *gethostbyname
                    (const char *hostname);
```

- In DNS terms, gethostbyname() does a request for a type **A** record
- Obs.: gethostbyname() is used mostly for IPv4

DNS| gethostbyname()

- Returns:
  - On success, it returns a pointer to **hostent**, which contains the host's IP address
  - On error, it returns NULL, and h_errno indicates the error number:
    - HOST_NOT_FOUND
    - …
    - NO_RECOVERY
    - …



Constants defined in netdb.h

DNS| gethostbyname()

- Usage example: setting a symbolic name instead of an IP address in a *sockaddr_in* structure:

```
struct sockaddr_in server;
struct hostent *hos;
if(!( hos = gethostbyname("fenrir.info.uaic.ro") )
  {/*Error on resolving address*/}
server.sin_family=AF_INET;
  /* we take the IP address from hos structure */
memcpy(&server.sin_addr.s_addr, hos->h_addr_list[0],
                    sizeof(hos->h_addr_list));
server.sin_port=htons(4321);
```

DNS| gethostbyaddr()

```
#include <netdb.h>
        struct hostent *gethostbyaddr (
                        const char *addr,
                        socklen_t len,
                        int family);
```

- In DNS terms, **gethostbyaddr()** does a request to the nameserver for a **PTR** record in *in-addr.arpa* domain
- Return value: On success, it returns a pointer to **hostent**, which contains the official name of the *host*; On error, it returns NULL, and h_errno variable indicates the error

Obs.: gethostbyaddr() is used mostly for IPv4

DNS| getservbyname()

```
#include <netdb.h>
struct servent *getservbyname (const char *servname, const char
    *protoname);
```

- Return value: On success, a pointer to *struct servent*; On error, NULL

```
        struct servent {
            char *s_name;      /* official name of the service*/
            char **s_aliases;  /* aliases */
            int s-port;        /* port (network-byte order) */
            char *s_proto;     /* protocol */ };
```

Example: struct servent *pserv;

```
        pserv=getservbyname("ftp","tcp"); /* FTP using TCP */
```

DNS| getservbyport()

#include <netdb.h>

struct **servent** ***getservbyport** (int **port,** const char ***protoname**);

- Searches for a service that matches the specified *port* and *protocol* (optional)
- Return value: a pointer to *struct servent* on success, NULL on error

Obs.: *port* must be specified in *network byte order*

Example:

```
struct servent *pserv;
pserv=getservbyport( htons(53), ”udp”); /* DNS using UDP */
pserv=getservbyport( htons(21),”tcp”); /* FTP using TCP */
```

DNS| getaddrinfo()

```
#include <netdb.h>
int getaddrinfo (
    const char *hostname,
    const char *service,
    const struct addrinfo *hints,
    struct addrinfo **result ) ;
```

Hostname or an IPv4/IPv6 address as string

Service's port or name ("http","pop",..) (see /etc/services file)

Specifies criteria for refining the return value

- Obs. *hostname*, *service*, *hints* – input parameters
- Return value: 0 on success, !=0 on error
- Recommended to be used for IPv4 and IPv6
- Combines functionalities of: gethostbyname(), getservbyname(), getservbyport()

DNS| getaddrinfo()

```
struct addrinfo {
    int ai_flags;         /* AI_PASSIVE, AI_CANONNAME */
    int ai_family;        /* AF_INET, AF_INET6, AF_UNSPEC */
    int ai_socktype;      /* SOCK_STREAM or SOCK_DGRAM */
    int ai_protocol;      /* 0 (auto) or IPPROTO_TCP, IPPROTO_UDP */
    socklen_t ai_addrlen; /* ai_addr length */
    char *ai_canonname; /* host's canonical name */
    struct sockaddr *ai_addr; /* socket's binary address */
    struct addrinfo *ai_next; /* pointer to the next structure from
    the list */
};
```

DNS| getaddrinfo()

Discussion:

- If the function successfully returns, **result** will point to a list of **struct addrinfo** elements

  Cases when we can have multiple structures:

  - If a hostname has multiple IP addresses, then a structure is returned for each one
  - If the service is offered for different types of *sockets*, then a structure in returned for each type of *socket*
- The data returned by **getaddrinfo**() in **struct addrinfo** ****result** can be used like this:
  - for **socket**() : **ai_family**, **ai_socktype**, **ai_protocol**
  - for **connect**() or **bind**(): **ai_addr** and **ai_addrlen**
- freeaddrinfo()

DNS| getnameinfo()

```
#include <netdb.h>
int getnameinfo (
    const struct sockaddr *sockaddr,
    socklen_t addrlen,
    char *host,
    socklen_t hostlen,
    char *serv,
    socklen_t servlen,
     int flags) ;
```

socket's address

name of the returned host

the name of the service

NI_NOFQDN -> **host** will contain only the name of the host, not the full domain name

- Replaces gethostbyaddr() and getservbyport()
- Returns: 0 on success, !=0 on error

DNS| IDN

- **International Domain Names (IDN)**
  - Extension which enables the use of Unicode characters for domain names, not only ASCII ones

    https://www.icann.org/en/topics/idn/

    16 Noiembrie 2009 – Registration of ccIDN or IDN ccTLD domains

    2010-01: ICANN announces that Egypt, the Russian Federation, Saudi Arabia, and the United Arab Emirates were the first countries to have passed the Fast Track String Evaluation within the IDN ccTLD domain application process.

  - Can be used for *phishing attacks* (… details in a future lecture)

DNS| administration

- DNS root is officially administered by Internet Corporation for Assigned Names and Numbers (ICANN)

- There are other organizations that offer alternative roots, like OpenNIC (Network Information Center) or New.Net

Summary

- Domain Name System (DNS)
  - Characterization
  - Organization
  - Configuration
  - Commands, Primitives
  - IDN

Questions?