

PROTOCOALE DE COMUNICAȚIE : LABORATOR 9

Protocolul DNS

Responsabil: Radu-Ioan CIOBANU

Cuprins

Obiective	1
Prezentarea laboratorului	1
Protocolul DNS	2
Spațiul de nume	2
Algoritmul de interogare	3
Înregistrări de resurse	3
Servere de nume	4
API DNS	5
Funcțiile <i>gethostbyname()</i> și <i>gethostbyaddr()</i>	5
Apelul <i>getaddrinfo()</i>	6
Apelul <i>getnameinfo()</i>	7
Exercițiu	7
Referințe	7

Obiective

În urma parcurgerii acestui laborator, studentul va fi capabil să:

- Opereze cu ierarhia spațiilor de nume și să identifice tipurile de domenii și subdomenii.
- Folosească algoritmul de interogare utilizat de DNS.
- Identifice tipurile de resurse pentru diverse domenii și clasele acestora.
- Folosească un set minimal de funcții pentru aflarea informațiilor unui sistem gazdă.

Prezentarea laboratorului

Programele se refera rareori la sisteme gazdă, cutii poștale și alte resurse prin adresele lor binare. În loc de numere binare, se utilizează șiruri ASCII, cum ar fi *user@cs.pub.ro*. Cu toate acestea, rețeaua înțelege numai adrese binare, deci este necesar un mecanism care să convertească șirurile ASCII în adrese de rețea. Protocolul care se ocupă de acest lucru se numește DNS (Domain Name System - sistemul numelor de domenii). Esența DNS-ului constă dintr-o schemă ierarhică de nume de domenii și dintr-un sistem de baze

de date distribuite pentru implementarea acestei scheme de nume. Protocolul este definit în RFC-urile [1034](#) și [1035](#).

Protocolul DNS

Spațiul de nume

DNS organizează numele resurselor într-o ierarhie de domenii. Un domeniu reprezintă o colecție de sisteme gazdă care au unele proprietăți în comun, cum ar fi faptul că toate aparțin unei aceleiași organizații sau faptul că toate sunt situate geografic în același perimetru.

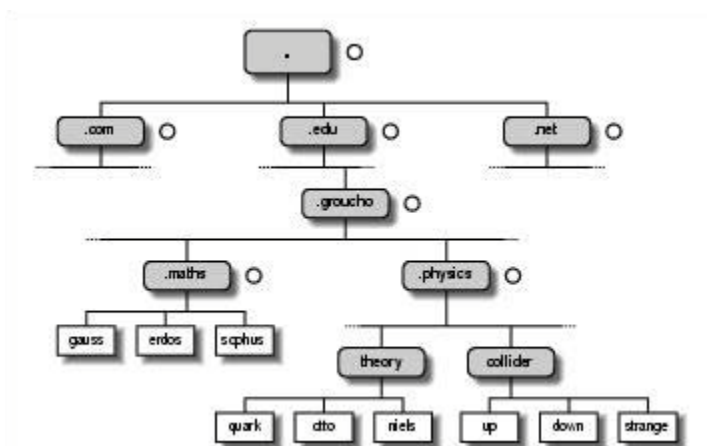


Figura 1: O porțiune a spațiului numelor de domenii din Internet

Fiecare domeniu este partiționat în subdomenii și acestea sunt, la rândul lor, partiționate, ș.a.m.d. Toate aceste domenii pot fi reprezentate ca un arbore, după cum se poate vedea în Figura 1. Frunzele arborelui reprezintă domenii care nu au subdomenii, dar care conțin totuși sisteme. Un domeniu frunză poate conține de la un singur sistem gazdă până la mii de sisteme gazdă.

Domeniile de pe primul nivel se împart în două categorii: generice și de țări. Domeniile generice sunt *com* (comercial), *edu* (instituții educaționale), *gov* (guvernul SUA), *int* (organizații internaționale), *mil* (forțele armate ale SUA) și *org* (organizații nonprofit). Domeniile de țări includ o intrare pentru fiecare țară, după cum se definește în [ISO 3166](#). Fiecare domeniu este denumit de calea în arbore până la rădăcină. Componentele sunt separate prin punct. Astfel, departamentul de inginerie de la SUN poate fi *eng.sun.com* în loc de numele în stil UNIX */com/sun/eng*.

Numele de domenii pot fi absolute sau relative. Un nume absolut de domeniu (FQDN - fully qualified domain name) este un nume de domeniu care nu permite nici o ambiguitate cu privire la locația relativă la rădăcina arborelui de nume de domenii. Astfel de nume absolute de domenii se termină cu punct (de exemplu *eng.sun.com.*). În contrast, un nume relativ de domeniu este un nume care are sens numai relativ la un anume domeniu DNS (altul decât cel rădăcină).

Numele de domenii nu fac distincție între litere mici și litere mari, *edu* sau *EDU* înseamnă practic același lucru. Componentele numelor pot avea o lungime de cel mult 64 de caractere, iar întreaga cale de nume nu trebuie să depășească 255 de caractere.

Fiecare domeniu controlează cum sunt alocate domeniile de sub el. De exemplu, Japonia are domeniile *ac.jp* și *co.jp* echivalente cu *edu* și *com*. Olanda nu face nici o distincție și pune toate organizațiile direct sub *nl*. Pentru a crea un nou domeniu, se cere permisiunea domeniului în care va fi inclus. De exemplu, dacă un grup VLSI de la Yale dorește să fie cunoscut ca *vlsi.cs.yale.edu*, acesta are nevoie de permisiunea celui

care administrează *cs.yale.edu*. Similar, o nouă universitate care dorește obținerea unui domeniu va trebui să ceară permisiunea administratorului domeniului *edu*. În acest mod, sunt evitate conflictele de nume și fiecare domeniu poate ține evidența tuturor subdomeniilor sale. Odată ce un nou domeniu a fost creat și înregistrat, el poate crea subdomenii, fără a cere permisiune de la cineva din partea superioară a arborelui.

Algoritmul de interogare

Conceptele cu care DNS lucrează sunt:

1. Servere DNS - Stații care rulează programe de tip server de DNS ce conțin informații asupra bazelor de date DNS și despre structura numelor de domenii.
2. Resolvere DNS - Programe care folosesc cereri DNS pentru interogarea unor servere DNS. Modul în care se derulează procesul de interogare DNS este cel din Figura 2.

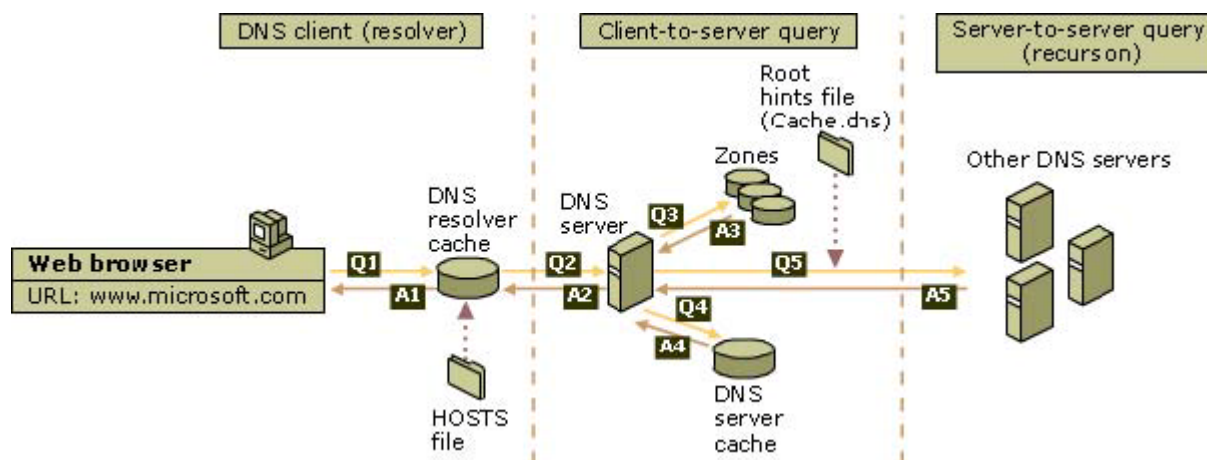


Figura 2: Procesul de interogare DNS

Înregistrări de resurse

Fiecărui domeniu, fie că este un singur calculator gazdă, fie un domeniu de nivel superior, îi poate fi asociată o mulțime de înregistrări de resurse (resource records). Pentru un singur sistem gazdă, cea mai obișnuită înregistrare de resursă este chiar adresa IP, dar există multe alte tipuri.

Atunci când procedura resolver trimite un nume de domeniu DNS, ceea ce va primi ca răspuns sunt înregistrările de resurse asociate aceluși nume. Astfel, adevărata funcție a DNS este să realizeze corespondența dintre numele de domenii și înregistrări de resurse.

O înregistrare de resursă este un 5-tuplu. Cu toate că, din rațiuni de eficiență, înregistrările de resurse sunt codificate binar, în majoritatea expunerilor ele sunt prezentate ca text ASCII, câte o înregistrare de resurse pe linie. Formatul pe care îl vom utiliza este *Nume_domeniu Timp_de_viață Tip Clasă Valoare*.

- *Nume_domeniu* precizează domeniul căruia i se aplică această înregistrare. În mod normal, există mai multe înregistrări pentru fiecare domeniu, și fiecare copie a bazei de date păstrează informații despre mai multe domenii. Acest câmp este utilizat cu rol de cheie de căutare primară pentru a satisface cererile. Ordinea înregistrărilor în baza de date nu este semnificativă. Când se face o interogare despre un domeniu, sunt returnate toate înregistrările care se potrivesc cu clasa cerută.
- Câmpul *Timp_de_viață* dă o indicație despre cât de stabilă este înregistrarea.
- Câmpul *Tip* precizează tipul înregistrării. Cele mai importante tipuri sunt prezentate în Tabelul 1.

Tabelul 1: Tipuri de înregistrări DNS

Tip	Semnificație	Valoare
SOA	Start autoritate	Parametri pentru această zonă
A	Adresa IP a unui sistem gazdă	Intreg pe 32 de biți
MX	Schimb de poștă	Prioritate, domeniu dispus să accepte poștă electronică
NS	Server de nume	Numele serverului pentru acest domeniu
CNAME	Nume canonic	Numele domeniului
PTR	Pointer	Pseudonim pentru adresa IP
HINFO	Descriere sistem gazdă	Unitate centrală și sistem de operare în ASCII
TXT	Text	Text ASCII neinterpretat

O înregistrare SOA furnizează numele sursei primare de informație despre zona serverului de nume, adresa de e-mail a administratorului, un identificator unic și diverși indicatori și contoare de timp.

Cel mai important tip de înregistrare este înregistrarea A (adresă). Ea păstrează adresa IP de 32 de biți a sistemului gazdă. Următoarea ca importanță este înregistrarea MX. Aceasta precizează numele domeniului pregătit să accepte poștă electronică pentru domeniul specificat. Înregistrările specifică numele serverului. Un exemplu de informație ce se poate găsi în baza de date DNS a unui domeniu este următorul:

```

1 ; Authoritative Information on physics.groucho.edu.
2 @ IN SOA niels.physics.groucho.edu.
3 janet.niels.physics.groucho.edu. {
4 1999090200 ; serial no
5 360000 ; refresh
6 3600 ; retry
7 3600000 ; expire
8 3600 ; default ttl
9 };
10 ; Name servers
11 IN NS niels
12 IN NS gauss.maths.groucho.edu.
13 gauss.maths.groucho.edu. IN A 149.76.4.23
14 ;
15 ; Theoretical Physics (subnet 12)
16 niels IN A 149.76.12.1
17 IN A 149.76.1.12
18 name server IN CNAME niels
19 otto IN A 149.76.12.2
20 quark IN A 149.76.12.4
21 down IN A 149.76.12.5
22 strange IN A 149.76.12.6
23 ...
24 ; Collider Lab. (subnet 14)
25 boson IN A 149.76.14.1
26 muon IN A 149.76.14.7
27 bogon IN A 149.76.14.12
28 ...

```

Servere de nume

Teoretic, un singur server de nume poate conține întreaga bază de date DNS și poate să răspundă tuturor cererilor. În practică, acest server poate fi atât de încărcat încât să devină de neutilizat. Pentru a evita

probleme asociate cu existența unei singure surse de informație, spațiul de nume DNS este împărțit în zone care nu se suprapun. O posibilă astfel de împărțire este cea din Figura 3.

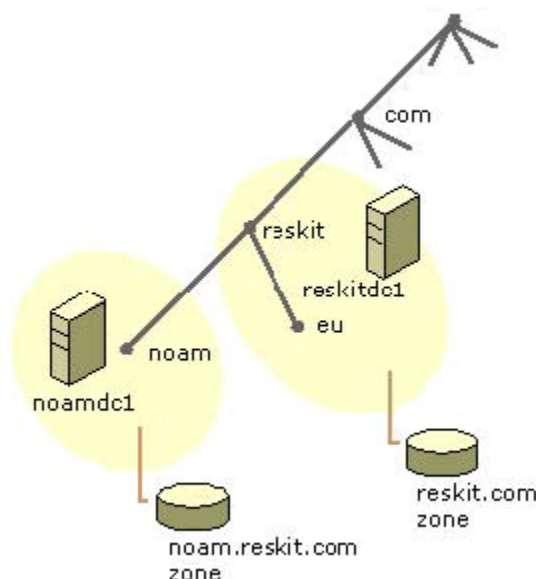


Figura 3: O parte din spațiul numelor DNS reprezentând împărțirea în zone

Fiecare astfel de zonă conține câte o parte a arborelui, precum și numele serverelor care păstrează informația autorizată despre acea zonă. În mod normal, o zonă va avea un server de nume primar, care preia informația dintr-un fișier de pe discul propriu, și unul sau mai multe servere de nume secundare, care iau informația de la serverul primar. Pentru a îmbunătăți fiabilitatea, unele servere pentru o zonă pot fi plasate chiar în afara zonei.

Plasarea limitelor unei zone este la latitudinea administratorului ei. Această decizie este luată în mare parte pe baza numărului de servere de nume care se doresc a se folosi, și a locației acestora. Atunci când un resolver are o cerere referitoare la un nume de domeniu, el transferă cererea unuia din serverele locale de nume. Dacă domeniul este sub jurisdicția serverului de nume, el va întoarce înregistrări de resurse autorizate. O înregistrare autorizată (authoritative record) este cea care vine de la autoritatea care administrează înregistrarea, și astfel este întotdeauna corectă. Înregistrările autorizate se deosebesc de înregistrările din memoria cache, care pot fi expirate.

Dacă totuși domeniul se află la distanță, iar local nu este disponibilă nici o informație despre el, atunci serverul de nume trimite un mesaj de cerere către serverul de nume de pe primul nivel al domeniului solicitat. De menționat că metoda de interogare este recursivă (recursive query), deoarece fiecare server care nu are informația cerută o caută în altă parte și raportează.

API DNS

Aplicațiile Internet se folosesc de apelurile de sistem [*getnameinfo\(\)*](#) și [*getaddrinfo\(\)*](#).

Funcțiile *gethostbyname()* și *gethostbyaddr()*

Până de curând, pentru a afla un nume pe baza unei adrese IP și o adresă pe baza unui nume, se foloseau funcțiile [*gethostbyname\(\)*](#) și [*gethostbyaddr\(\)*](#), împreună cu structura *hostent*. Între timp, acest API pentru DNS a fost scos din uz.

Apelul *getaddrinfo()*

```
1 #include <sys/types.h>
2 #include <sys/socket.h>
3 #include <netdb.h>
4
5 int getaddrinfo(const char *node, const char *service,
6                 const struct addrinfo *hints, struct addrinfo **res);
```

Această funcție primește informații despre numele unei gazde și al unui serviciu Internet, și returnează adresa sau adresele corespunzătoare. Parametrul *node* reprezintă numele simbolic (sub forma unui șir de caractere) al mașinii căreia vrem să-i aflăm adresa (de exemplu, *node* poate fi “www.google.com”). Mai poate de asemenea fi reprezentat ca un șir care conține o adresă IPv4 sau IPv6.

Parametrul *service* specifică portul returnat în output, și poate fi pus pe NULL (caz în care portul din output rămâne neinițializat) sau poate fi dat ca un nume de serviciu (de exemplu, “http”) sau ca o valoare numerică (“80”).

Parametrul *hints* reprezintă criterii pentru filtrarea adreselor întoarse de apelul funcției *getaddrinfo()*. Este de tipul *struct addrinfo*, definit mai jos:

```
1 struct addrinfo {
2     int          ai_flags;
3     int          ai_family;
4     int          ai_socktype;
5     int          ai_protocol;
6     socklen_t    ai_addrlen;
7     struct sockaddr *ai_addr;
8     char         ai_canonname;
9     struct addrinfo *ai_next;
10 };
```

În cazul în care se dorește filtrarea, se pot completa unul sau mai multe din următoarele câmpuri (restul punându-se pe 0):

- *ai_family* - se specifică familia de adrese pentru valorile returnate, putând fi setată ca AF_INET (pentru IPv4), AF_INET6 (pentru IPv6) sau AF_UNSPEC (pentru ambele)
- *ai_socktype* - se filtrează după tipul de socket (SOCK_DGRAM sau SOCK_STREAM, de exemplu)
- *ai_protocol* - se specifică protocolul setat în adresele returnate de funcția *getaddrinfo()*
- *ai_flags* - se pot seta o serie de flag-uri.

În final, rezultatul este pus în parametrul *res*, fiind reprezentat ca o listă înlănțuită de structuri de tipul *addrinfo*, care se parcurge prin intermediul câmpului *ai_next*. Din câmpul *ai_addr* al rezultatului, se pot citi informațiile despre adresa și portul stației gazdă căutate (prin cast la *struct sockaddr_in*, de exemplu).

În caz de succes, funcția întoarce 0, iar în caz de eroare întoarce o valoare negativă, care poate fi interpretată prin intermediul funcției *gai_strerror()*:

```
1 const char *gai_strerror(int errcode);
```

Atenție! Parametrul *res* este alocat de către funcția *getaddrinfo()*, însă el trebuie dezalocat explicit de către utilizator prin intermediul funcției *freeaddrinfo()*:

```
1 void freeaddrinfo(struct addrinfo *res);
```

Pentru a afișa adresa IP (v4 sau v6) corespunzătoare numelui simbolic din parametrul *node*, se poate utiliza funcția *inet_ntop()*:

```
1 #include <arpa/inet.h>
2
3 const char *inet_ntop(int af, const void *src, char *dst, socklen_t size);
```

Primul parametrul specifică familia de protocoale (AF_INET sau AF_INET6), al doilea parametru reprezintă structura de adresă (adică *sockaddr_in* pentru IPv4 sau *sockaddr_in6* pentru IPv6), al treilea parametru reprezintă un șir de caractere unde va fi scrisă adresa sub formă de string, iar ultimul parametru reprezintă dimensiunea șirului de caractere în octeți. Valoarea de retur a funcției este un pointer la un șir de caractere identic cu cel din parametrul *dst*, sau NULL în caz de eroare.

Apelul *getnameinfo()*

```
1 #include <sys/socket.h>
2 #include <netdb.h>
3
4 int getnameinfo(const struct sockaddr *addr, socklen_t addrlen, char *host, socklen_t
    hostlen, char *serv, socklen_t servlen, int flags);
```

Această funcție realizează operația inversă față de *getaddrinfo()*. Mai precis, primește o adresă și returnează numele simbolic și serviciul specifice adresei respective. Primii doi parametri reprezintă adresa IP (v4 sau v6) care este căutată. Se trimit structuri specifice protocolului dorit (*sockaddr_in* sau *sockaddr_in6*) și dimensiunea lor. Rezultatele apelului sunt puse în șirurile de caractere *host* și *serv*, care sunt alocate de către utilizator. Parametrii *hostlen* și *servlen* reprezintă dimensiunile celor două șiruri de caractere.

Funcția returnează 0 dacă s-a reușit cererea DNS, sau o valoare negativă interpretată cu *gai_strerror()* în caz contrar.

Exercițiu

Folosind funcțiile DNS, scrieți un program care să afișeze toate informațiile pentru un host: nume, adrese IP. Programul poate primi ca parametru fie numele (caz în care se va afișa adresa), fie adresa IP (caz în care se va afișa numele). Exemplu de apel:

```
1 ./dns -n google.com
2 ./dns -a 141.85.128.1
```

Referințe

1. [Setarea unui server de DNS în Linux din “Linux Network Administrators Guide”](#)
2. [DNS HOWTO](#).