Laborator APC - 2

Sistemul DNS

Protocoalele de nivel aplicație folosesc diverse scheme de adresare, adaptate aplicațiilor pentru care au fost proiectate. De exemplu, un URL (Unique Resource Locator), identifică o resursă disponibilă pe un server conectat la Internet și protocolul de nivel aplicație care trebuie utilizat pentru o accesa; de pildă, `https://www.ietf.org/rfc.html` specifică faptul că fișierul `rfc.html` poate fi descărcat de pe serverul `www.ietf.org` folosind protocolul HTTP (pe o conexiune securizată stabilită de protocolul TLS, peste o conexiune de nivel transport stabilită de TCP).

Pentru ca o aplicație să poată comunica prin Internet, trebuie definită o corespondență între schema sa de adresare și adresele folosite de protocoalele de nivel rețea și transport. În exemplul de mai sus, programul aplicație (un web browser, de pildă) trebuie să stabilească o conexiune TCP cu serverul `www.ietf.org`, prin urmare trebuie să afle mai întâi adresa IP a acestui server.

Schemele de adresare folosite de aplicații se bazează pe o schemă mai generală, definită de sistemul DNS (Domain Name System). De pildă, URL-ul discutat mai sus folosește numele de domeniu `www.ietf.org` pentru a identifica serverul unde este stocat fișierul. Sistemul DNS este responsabil atât pentru gestiunea spațiului de nume, cât și pentru corespondența dintre aceste nume și adresele IP. În aceste condiții, evident, sistemul DNS reprezintă una dintre componentele fundamentale ale infrastructurii Internet-ului.

Obiective

Obiectivul lucrării este familiarizarea cu elemente fundamentale ale arhitecturii și funcționării sistemului DNS: structura spațiului numelor, implementarea sistemului (zone, ierarhia de servere, configurarea serverelor și a clienților), protocolul DNS, precum și exemple tipice de interacțiuni între componentele sistemului. În acest scop, vom analiza funcționarea unui model la scară redusă al acestui sistem.

Precondiții

Pentru a putea efectua experimentele și a interpreta rezultatele trebuie să studiați în prealabil capitolele din materialul de curs (și eventual bibliografia suplimentară) care prezintă noțiunile de bază privind spațiul numelor definit de DNS, arhitectura sistemului și protocolul DNS.

Software și echipamente

Vom folosi implementări ale componentelor sistemului DNS disponibile în sistemul de operare Linux, în primul rând pachetul BIND 9 (server, client, utilitare DNS) [2]. Veți captura și analiza comunicațiile dintre echipamente folosind analizoarele de protocoale `tcpdump` și `wireshark`.

Fiecare student (sau echipă de 2 studenți) va lucra pe un calculator care rulează sistemul de operare Linux. Sistemul DNS studiat este emulat pe fiecare calculator folosind `netkit`. Fiecare componentă a sistemului (clienți și servere DNS) este implementată ca o mașină virtuală Linux și este accesibilă prin intermediul unui terminal (pentru configurare, examinarea stării, executarea unor utilitare, etc.).

Studiu de caz

În DNS, spațiul numelor este definit pe baza unei structuri de tip arbore. Figura 1 prezintă structura spațiului numelor utilizat în această lucrare de laborator. Este un model la scara redusă a spațiului

numelor din Internet, suficient pentru a studia experimental funcționarea DNS.

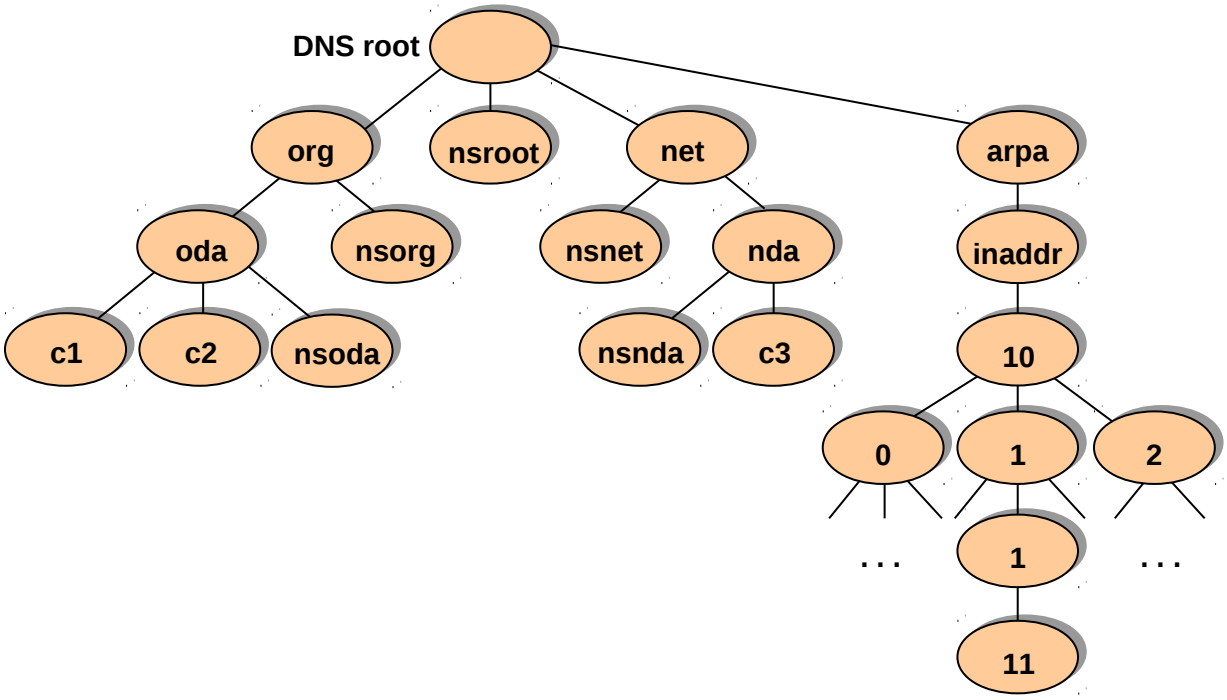

Figura 1: Structura spațiului numelor de domeniu.

Fiecare nod din arbore, cu excepția nodului rădăcină, are o etichetă. De asemenea, fiecărui nod îi corespunde un nume, construit prin concatenarea etichetelor nodurilor aflate pe calea dintre nodul respectiv și nodul rădăcină (separate prin "."). De exemplu, nodului cu eticheta `oda` îi corespunde numele `oda.org`, iar nodului cu eticheta `nsoda` îi corespunde numele `nsoda.oda.org`.

Un subarbore din această structură reprezintă un domeniu. Numele acestui domeniu este numele nodului rădăcină al arborelui. De exemplu, subarborele având ca rădăcină nodul cu eticheta `oda` reprezintă domeniul `oda.org`. În acest exemplu, o firmă numită ODA deține domeniul `oda.org` și o altă firma, numită NDA, deține domeniul `nda.net`.

Sub-arborele având ca rădăcină nodul `arpa` este utilizat pentru a stabili corespondența inversă, de la o adresa IP la un nume. Pentru a integra această funcție în DNS, fiecărei adrese IP îi este asociat un nume de domeniu. De exemplu, numele asociat adresei 10.1.1.11 este `11.1.1.10.inaddr.arpa`. Această funcție este mai puțin importantă și nu vom mai insista asupra ei.

Nodurile terminale ale arborelui (noduri "frunză") corespond unor echipamente conectate la rețea. Acest lucru este ilustrat în Figura 2. De pildă, în domeniul firmei ODA, există două calculatoare, cu numele `c1.oda.org` și `c2.oda.org`, și un server DNS, cu numele `nsoda.oda.org`.

Sistemul DNS este implementat ca o bază de date distribuită, în care fiecare server DNS este responsabil pentru o parte din spațiul numelor. În acest scop, spațiul numelor este partiționat în *zone DNS*, reprezentând porțiuni conexe și distincte din arbore, obținute prin tăierea unor arce (nu există suprapuneri între zone, fiecare nod aparține unei singure zone). Zonele sunt administrate separat.

Fiecărui server DNS i se poate aloca una sau mai mult zone. Figura 3 ilustrează zonele DNS definite în sistemul studiat în lucrarea de laborator, precum și serverul DNS responsabil pentru fiecare zonă. De exemplu, serverul DNS `nsoda.oda.org` servește o zonă DNS care reprezintă întregul domeniu `oda.org`, în timp ce `nsorg.org` servește o zonă care reprezintă doar o parte din domeniul `org` (în general, nu este necesar ca un server să facă parte din zona pe care o servește).

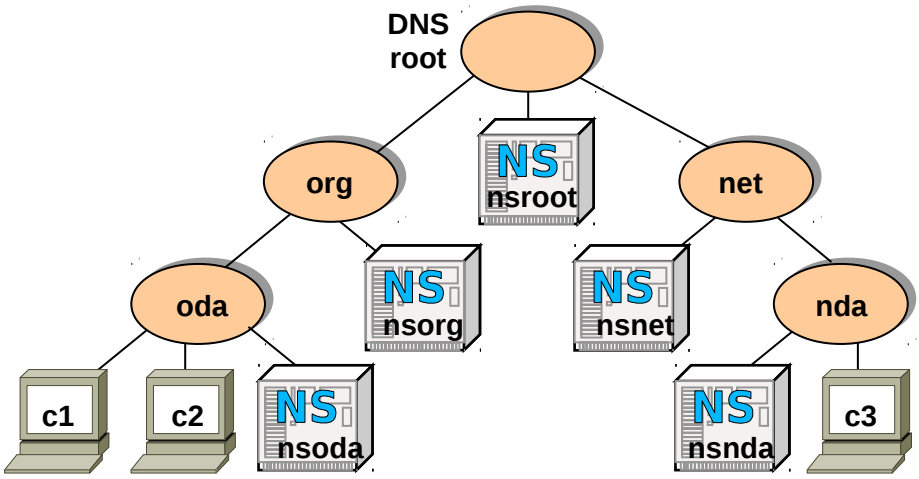

Figura 2: Spațiul numelor și echipamentele din rețea.

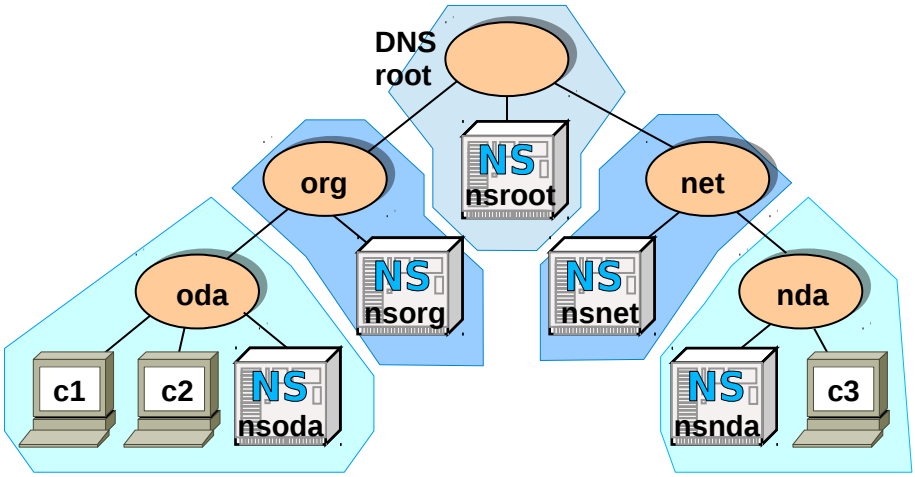

Figura 3: Zonele DNS și serverele DNS asociate.

Pentru fiecare zonă DNS se crează o bază de date alcătuită din înregistrări DNS cu structură standardizată (DNS resource records). De exemplu:

- O înregistrare de tip A (Address) asociază unui nume de domeniu o adresă IPv4 (de exemplu, pentru `nsorg.org`, adresa 10.0.1.1).
- O înregistrare de tip NS (Name Server) asociază unui nume de domeniu numele serverului DNS autoritar (responsabil) pentru acel domeniu (de exemplu, pentru domeniul `org`, serverul `nsorg.org`).

Structura de arbore a spațiului numelor oferă o metodă simplă și scalabilă de asigurare a unicității numelor: este suficient ca, pentru fiecare nod, nodurile direct descendente să aibă etichete distincte. Mai mult, această structură permite delegarea responsabilității pentru administrarea domeniilor DNS către organizații diferite. De exemplu, firma ODA este responsabilă pentru administrarea domeniului `oda.org` și poate să adauge noi subdomenii fără să fie nevoie de coordonare cu alte organizații pentru a garanta unicitatea globală a numelor.

Pe de altă parte, din punct de vedere a resurselor consumate, scalabilitatea serviciului oferit de DNS este asigurată prin organizarea unei ierarhii de servere. În exemplul nostru:

- Serverul `nsroot` cunoaște adresele IP ale serverelor `nsorg.org` și `nsnet.net` (prin înregistrări de tip NS și A).
- La rândul său, `nsorg.org` cunoaște adresele IP ale serverelor DNS care sunt responsabile pentru subdomeniile domeniului `org`, de exemplu serverul `nsoda.oda.org`.
- Similar, `nsnet.net` cunoaște adresele IP ale serverelor DNS care se ocupă de subdomeniile domeniului `net`.

Astfel, fiecare server DNS cunoaște doar o mică parte a spațiului numelor, dar poate găsi serverul care deține informația pe care o caută prin câteva interogări succesive, pornind de la `nsroot`.

Figura 4 prezintă sistemul DNS pe care îl vom folosi pentru experimente: serverele DNS discutate mai sus și câteva calculatoare personale. În practică, aceste echipamente sunt conectate la rețele IP. Figura 4 indică adresa IP alocată fiecăruia. În sistemul pe care îl vom utiliza în lucrarea de laborator, pentru a facilita capturarea traficului și analiza comunicațiilor, echipamentele comunică direct, ca și când ar fi conectate la un hub Ethernet. Prin urmare, puteți captura întregul trafic pe oricare interfață Ethernet folosind programul `tcpdump`.

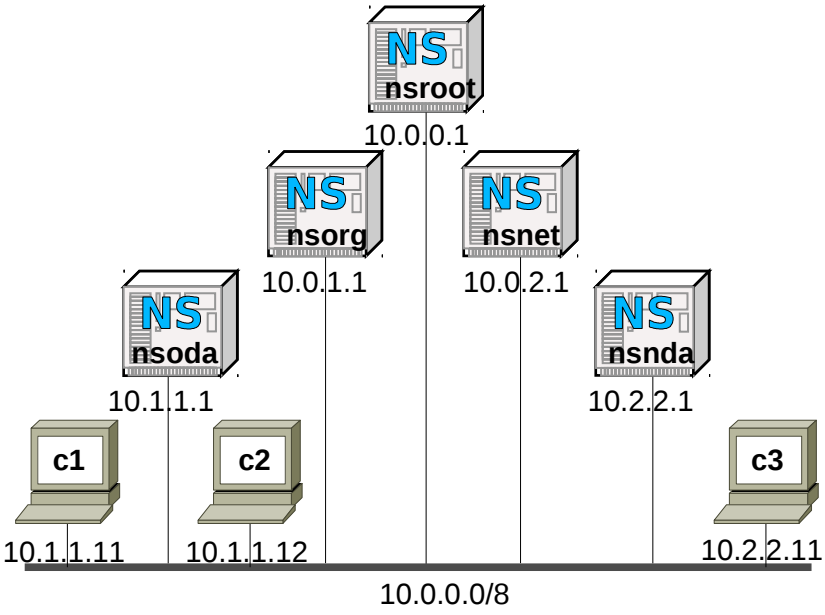

Figura 4: Sistemul DNS folosit în lucrarea de laborator.

Proiectul `netkit` cu care începeți lucrarea conține toate echipamentele din Figura 4, configurate complet, gata de execuție (adrese IP, clienți și servere DNS).

1. Inițializarea sistemului

1.1. Pentru a porni emularea rețelei din Figura 4, executați comanda `lstart` într-un terminal al calculatorului gazdă, în directorul în care se află fișierele de configurare `netkit` ale sistemului.

1.2. Examinați și verificați configurarea IP a echipamentelor folosind comanda `ifconfig`.

2. Configurarea sistemului DNS

2.1. Fișierul de configurare al clientului DNS ("DNS resolver") este `/etc/resolv.conf`. De exemplu, pentru calculatorul c1, acest fișier are conținutul listat mai jos:

```
c1:~# cat /etc/resolv.conf
nameserver 10.1.1.1
search oda.org
```

Directiva `nameserver` specifică adresa serverului DNS local (`nsoda.oda.org`, 10.1.1.1).

Directiva `search` specifică sufixul care este adăugat unui nume DNS incomplet, pentru a obține un nume de domeniu complet (fully qualfied domain name - FQDN). De exemplu, dacă o aplicație solicită adresa IP asociată numelui incomplet c2, clientul DNS care rulează pe c1 va căuta înregistrarea tip A pentru numele `c2.oda.org`.

2.2. Examinați și explicați configurarea clientului DNS pe celelalte calculatore personale, c2 și c3.

2.3. Fișierele de configurare ale serverului DNS (BIND) se află în directorul `/etc/bind`. Vom examina mai întâi serverul `nsoda.oda.org`. Zonele pentru care `nsoda` deține informații sunt specificate în fișierul `named.conf`. Ne interesează doar porțiunea de mai jos:

```
nsoda:~# cat /etc/bind/named.conf
// This is the primary configuration file for the BIND DNS server named.
...
// prime the server with knowledge of the root servers
zone "." {
        type hint;
        file "/etc/bind/db.root";
        };
...
// add entries for other zones below here
zone "oda.org" {
        type master;
        file "/etc/bind/db.org.oda";
};
```

Serverul `nsoda` este responsabil (authoritative) pentru zona corespunzătoare domeniului `oda.org`. Fișierul care conține înregistrările DNS ale acestei zone este `/etc/bind/db.org.oda`:

```
nsoda:~# cat /etc/bind/db.org.oda
$TTL    60000
@               IN      SOA     nsoda.oda.org.    root.nsoda.oda.org. (
                        2006031201 ; serial
                        28 ; refresh
                        14 ; retry
                        3600000 ; expire
                        0 ; negative cache ttl
                        )
@               IN      NS      nsoda.oda.org.
nsoda           IN      A       10.1.1.1
c1              IN      A       10.1.1.11
c2              IN      A       10.1.1.12
```

Simbolul @ este folosit pentru a indica mai succint numele de domeniu corespunzător nodului care reprezintă originea zonei; în acest exemplu, este `oda.org` (specificat în `named.conf`). Directiva $TTL (Time-To-Live) specifică durata maximă cât pot fi stocate în DNS cache înregistrările din acest fișier (60000 secunde). Observați că un nume DNS complet (FQDN) se termină cu caracterul

".", de exemplu "nsoda.oda.org." (nodul rădăcină nu are etichetă).

Fișierul conține următoarele înregistrări:

- Fișierul unei zone începe cu o înregistrare de tip SOA (Start Of Authority). Rolul ei este de a specifica numele zonei (oda.org), numele serverului care este responsabil pentru această zonă (nsoda.oda.org), adresa de e-mail a administratorului (root@nsoda.oda.org) și o serie de parametri de configurare a serverului.
- Înregistrarea de tip NS specifică faptul serverul DNS cu numele nsoda.oda.org este autoritar pentru domeniul oda.org.
- Înregistrările de tip A specifică adresele IPv4 asociate numelor nsoda.oda.org, c1.oda.org și c2.oda.org.

În plus, pentru a putea iniția o căutare iterativă prin ierarhia de servere, pornind de la un server rădăcină, serverul nsoda are nevoie și de înregistrări tip NS privind zona rădăcină, listate în fișierul /etc/bind/db.root:

```
nsoda:~# cat /etc/bind/db.root
.                   IN  NS    ROOT-SERVER.
ROOT-SERVER.        IN  A     10.0.0.1
```

În exemplul nostru există doar un singur server DNS rădăcină. În Internet, sunt instalate însă 13 astfel de servere (de ce?).

2.4. Examinați și explicați configurarea celorlalte servere DNS: nsorg, nsroot, nsnet și nsnda.

3. Destinație într-o zonă DNS a serverului local

Vom începe experimentele cu cazul cel mai simplu: o comunicație în care numele destinației aparține uneia dintre zonele servite de serverul DNS local.

3.1. Porniți captura traficului:

```
nsroot:# tcpdump -i eth0 -s 0 -w /hostlab/dnstst1.cap
```

3.2. Inițiați o comunicație între calculatoarele c1.oda.org și c2.oda.org folosind ping:

```
c1:~# ping -n -c 3 c2.oda.org
```

3.3. Opriți programul tcpdump (Ctrl-C) și vizualizați traficul capturat folosind wireshark (comanda trebuie executată într-un terminal al calculatorului gazdă):

```
wireshark -r dnstst1.cap &
```

Examinați traficul capturat și explicați operațiile efectuate de DNS și mesajele transmise în acest scop de protocolul DNS. Indicații: Programul ping executat de c1 trebuie să afle adresa IP a calculatorului c2 folosind DNS. Prin urmare, un client DNS executat pe c1 va solicita serverului DNS local, nsoda.oda.org, înregistrarea de tip A pentru numele de domeniu c2.oda.org.

4. Destinație într-o zonă DNS a altui server

Vom analiza acum o comunicație în care numele destinației nu este într-o zonă alocată serverului DNS local.

4.1. Porniți captura traficului:

```
nsroot:# tcpdump -i eth0 -s 0 -w /hostlab/dnstst2.cap
```

4.2. Iniţiaţi o comunicaţie între calculatoarele `c1.oda.org` şi `c3.nda.net` folosind `ping`:

```
c1:~# ping -n -c 3 c3.nda.net
```

4.3. Opriţi programul `tcpdump` (`Ctrl-C`) şi apoi vizualizaţi traficul capturat folosind `wireshark` (comanda trebuie executată într-un terminal al calculatorului gazdă):

```
wireshark -r dnstst2.cap &
```

4.4. Analizaţi traficul capturat şi explicaţi operaţiile efectuate de DNS şi mesajele transmise în acest scop de protocolul DNS, pe baza indicaţiilor de mai jos.

Clientul DNS de pe c1 solicită serverului local înregistrarea de tip A pentru numele de domeniu `c3.nda.net`. Această înregistrare nu este disponibilă pe serverul local, deoarece `c3.nda.net` este într-o zonă controlată de alt server. În mesajul său, clientul DNS solicită ca această cerere să fie tratată recursiv (bitul Recursion Desired setat), în sensul că atunci când înregistrarea dorită nu este disponibilă local, serverul să o obţină efectuând el însuşi o căutare în sistemul DNS. Interacţiunile dintre componentele sistemului DNS sunt ilustrate în Figura 5. Identificaţi fiecare mesaj în traficul capturat şi explicaţi procedura folosită.

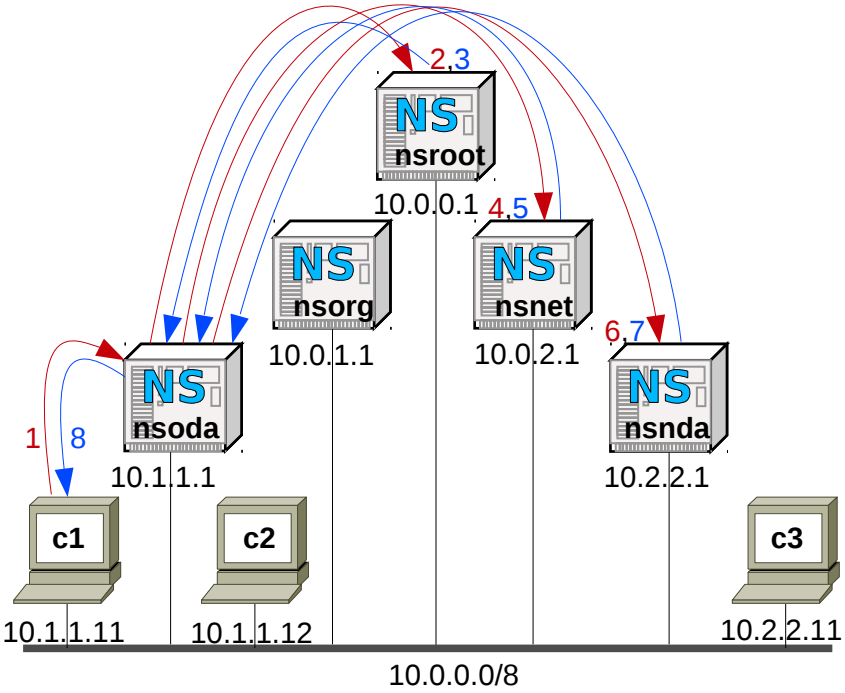

Figura 5: Secvenţa de cereri DNS din experimentul 4.

Observaţi că clientul trimite serverului local o cerere recursivă, în timp ce serverul efectuează căutarea printr-o succesiune de cereri nerecursive (iterative). Aceasta este funcţionarea tipică a sistemului. Explicaţi de ce a fost aleasă această soluţie: De ce nu foloseşte clientul cereri iterative? De ce nu transmite serverul nsoda cereri recursive?

4.5. Comparaţi biţii de control (flags) din antetul răspunsului DNS primit de `nsoda` de la `nsnda` cu cei din antetul răspunsului trimis de `nsoda` către c1. Identificaţi şi explicaţi diferenţa.

5. DNS caching: un prim experiment

După cum am văzut în experimentul precedent, o simplă cerere a unui client DNS poate declanşa o secvenţă complexă de interacţiuni între serverele DNS. Traficul DNS (încărcarea serverelor DNS şi a reţelei) şi timpul de răspuns al aplicaţiilor pot fi reduse substanţial folosind DNS caching.

Serverele și clienții DNS păstrează o copie locală a înregistrărilor pe care le obțin, astfel încât să poată fi livrată imediat în cazul care sunt solicitate în mod repetat.

Durata maximă cât poate fi păstrată o copie locală este indicată în câmpul TTL (Time To Live) asociat înregistrării în mesajul DNS. Când TTL expiră înregistrarea este ștearsă (de ce?).

În sistemul nostru, doar serverele DNS folosesc DNS caching. Experimentele care urmează scot în evidență utilitatea acestei tehnici.

5.1. Examinați conținutul curent al cache-ului DNS pentru serverele `nsoda.oda.org` și `nsnda.nda.net` folosind comenzile următoare:

```
rndc dumpdb -cache
cat /var/cache/bind/named_dump.db
```

Identificați înregistrările din cache-ul serverului `nsoda.oda.org` și explicați cum au fost obținute, pe baza traficului DNS capturat în experimentul 4. Explicați de ce celelalte servere DNS nu au nici o înregistrare în cache.

5.2. Porniți captura traficului:

```
nsroot:# tcpdump -i eth0 -s 0 -w /hostlab/dnstst3.cap
```

5.3. Inițiați o comunicație între calculatoarele `c2.oda.org` și `c3.nda.net` folosind `ping`:

```
c2:~# ping -n -c 3 c3.nda.net
```

5.4. Opriți programul `tcpdump` (`Ctrl-C`) și apoi vizualizați traficul capturat folosind `wireshark` (comanda trebuie executată într-un terminal al calculatorului gazdă):

```
wireshark -r dnstst3.cap &
```

5.5. Analizați traficul capturat și explicați operațiile efectuate de DNS și mesajele transmise în acest scop de protocolul DNS, pe baza indicațiilor de mai jos.

Clientul DNS de pe c2 solicită serverului local înregistrarea de tip A pentru numele de domeniu `c3.nda.net`. Această înregistrare nu este în zona servită de `nsoda.oda.org`, însă a fost deja solicitată de c1, astfel încât este disponibilă în DNS cache. Prin urmare, serverul răspunde imediat, furnizând copia locală a înregistrării.

5.6. Comparați valoarea câmpului Time To Live din răspunsul primit de c2 cu cea din răspunsul primit de c1 în experimentul precedent. Explicați diferența.

6. DNS caching: al doilea experiment

Pentru a evidenția importanța tehnicii DNS caching vom mai efectua încă un experiment.

6.1. Porniți captura traficului:

```
nsroot:# tcpdump -i eth0 -s 0 -w /hostlab/dnstst4.cap
```

6.2. Inițiați comunicații între calculatorul `c3.nda.net` și calculatoarele `c1.oda.org` și `c2.oda.org` folosind `ping`:

```
c3:~# ping -n -c 3 c1.oda.org
```

```
c3:~# ping -n -c 3 c2.oda.org
```

6.3. Opriți programul `tcpdump` (`Ctrl-C`) și apoi vizualizați traficul capturat folosind `wireshark` (comanda trebuie executată într-un terminal al calculatorului gazdă):

```
wireshark -r dnstst4.cap &
```

6.4. Analizați traficul capturat și explicați operațiile efectuate de DNS și mesajele transmise în acest scop de protocolul DNS.

7. Utilitarul `dig`

Folosind utilitarul `dig` (domain information groper) putem genera cereri DNS și afișa răspunsul, pentru a explora sau testa sistemul DNS.

7.1. Pentru a face cunoștință cu acest utilitar, parcurgeți rapid pagina sa de manual:

```
c1:~# man dig
```

7.2. Un prim exemplu: executați pe calculatorul `c1` comanda de mai jos.

```
c1:~# dig c2.oda.org
; <<>> DiG 9.5.0-P2 <<>> c2.oda.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29701
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;; QUESTION SECTION:
;c2.oda.org.                    IN      A

;; ANSWER SECTION:
c2.oda.org.             60000   IN      A       10.1.1.12
;; AUTHORITY SECTION:
oda.org.                60000   IN      NS      nsoda.oda.org.
;; ADDITIONAL SECTION:
nsoda.oda.org.          60000   IN      A       10.1.1.1

;; Query time: 14 msec
;; SERVER: 10.1.1.1#53(10.1.1.1)
;; WHEN: Tue Jul 26 16:25:28 2016
;; MSG SIZE  rcvd: 80
```

Ce s-a întâmplat? Programul `dig` a trimis serverului local (`nsoda.oda.org`) un mesaj DNS prin care a cerut înregistrarea de tip A pentru domeniul `c1.oda.org` și a afișat conținutul răspunsului primit (un rezumat). Câmpul `flags` listează biții de control care sunt setați în răspunsul primit, folosind abrevierile următoare: `qr` (query/response) - mesajul este un răspuns; `aa` (authoritative answer) - serverul care a trimis răspunsul este autoritar pentru domeniul specificat în cerere; `rd` (recursion desired) - clientul a trimis o cerere recursivă; `ra` (recursion available) - serverul acceptă cereri recursive. Înregistrările sunt prezentate folosind un format similar celui din fișierul de configurare al zonei și interpretarea este evidentă.

7.3. Explicați efectul comenzilor listate mai jos cu ajutorul paginii de manual a programului `dig` (`man dig`). Executați secvența de comenzi pe serverul `nsoda` și analizați răspunsurile.

```
nsoda:~# dig @10.0.0.1 +norecurse c3.nda.net
```

```
nsoda:~# dig @nsnet.net +norecurse c3.nda.net
```

```
nsoda:~# dig @nsnda.nda.net +norecurse c3.nda.net
```

Indicație: Comenzile simulează secvența de cereri nerecursive efectuate de `nsoda` în experimentul 4 (Figura 5), folosind `dig` pentru a genera aceste cereri.

7.4. Puteți specifica în comanda `dig` tipul de înregistrare DNS dorit. Consultați pagina de manual a utilitarului și determinați comanda cu care puteți obține de la serverul `nsnet.net` înregistrările de tip NS pentru domeniul `nds.net`. Testați comanda pe calculatorul c1.

7.5. Corespondența inversă, de la adrese IPv4 la nume de domeniu (subarborele cu rădăcina `arpa` din Figura 1) este definită în înregistrări DNS de tip PTR. Comenzile de mai jos folosesc `dig` pentru a obține înregistrări PTR, prin cereri DNS inverse (reverse DNS). Executați aceste comenzi și analizați răspunsurile.

```
c1:~# dig -x 10.2.2.1
```

```
c1:~# dig -x 10.2.2.11
```

Observație: În sistemul DNS din această lucrare de laborator, implementarea corespondenței inverse este mult simplificată. Este definită o zonă care cuprinde întregul domeniu `arpa.in-addr.10` și această zonă este alocată serverului `nsroot`.

8. Experimente în sistemul DNS real

Efectuați experimente folosind utilitarul `dig` în sistemul DNS real.

8.1. Repetați experimentul 7.3, pentru domeniul `www.upb.ro`, executând comenzile într-un terminal al calculatorului gazdă și pornind de la serverul rădăcină `a.root-servers.net`.

8.2. Executați într-un terminal al calculatorului gazdă comanda următoare:

```
dig www.upb.ro +trace +nodnssec +question +additional
```

Comparați rezultatul afișat de această comandă cu rezultatele experimentului 8.1. Explicați efectul opțiunilor din această comandă pe baza explicațiilor din pagina de manual a programului `dig`.

9. Terminarea lucrării

Executați comanda `lcrash` într-un terminal al calculatorului gazdă, în directorul în care se află fișierele de configurare `netkit` ale acestei lucrări.

Bibliografie

1. O. Catrina. APC - Note de curs.

2. P. Albitz, C. Liu. DNS and BIND, 5th Edition. Editura O'Reilly, 2006.