Laborator APC - 4

Rețele IP - Partea I

Protocolul IPv4 (și succesorul său, IPv6) ocupă un loc central în arhitectura și funcționarea Internet-ului. Mai mult, utilizarea sa s-a extins treptat către alte tipuri de rețele de telecomunicații, în condițiile evoluției către rețele cu servicii integrate (date, voce, audio, video) bazate pe comutație de pachete, fixe și mobile ("everything over IP, IP over everything").

Obiective

Vom studia rețelele bazate pe protocolul IPv4 pe parcursul a două lucrări de laborator.

În această primă lucrare vom considera o rețea IP izolată (fără conexiune la Internet), care este partiționată în mai multe subrețele, și ne vom concentra asupra unor noțiuni, componente și tehnici de bază privind funcționarea și configurarea acestor rețele:

- Alocarea adreselor (VLSM/CIDR), construcția tabelelor de rutare și dirijarea pachetelor.
- Configurarea interfețelor cu adrese alocate static și configurarea rutelor statice folosind comenzile tradiționale oferite de Linux/Unix, `ifconfig` și `route`. Examinarea stării interfețelor și a tabelului de rutare folosind aceste comenzi. Testarea rețelei folosind utilitarele `ping`, `arping` și `traceroute`.
- Protocoalele auxiliare ARP (Address Resolution Protocol) și ICMP (Internet Control Message Protocol).
- Configurarea automată cu adrese alocate dinamic folosind protocolul DHCP.

În lucrarea următoare, vom considera o rețea IP care este conectată la Internet și vom explora alte aspecte ale funcționării și configurării rețelelor IP, inclusiv elemente de rutare dinamică și tehnica NAT (Network Address Translation).

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: alocarea adreselor și dirijarea pachetelor în rețele bazate pe protocolul IPv4, protocoalele ARP și ICMP, configurarea automată folosind protocolul DHCP [3], serviciul DNS.

Software și echipamente

Vom folosi implementări ale protocoalelor și serviciilor disponibile în sistemul de operare Linux. 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 cu sistemul de operare Linux. Sistemul studiat este emulat pe fiecare calculator folosind platforma de emulare `netkit`. Fiecare componentă a sistemului este implementată ca o mașină virtuală Linux și este accesibilă prin intermediul unui terminal (pentru configurare, examinarea stării, executarea unor utilitare, etc.).

A. Alocarea adreselor și dirijarea pachetelor în rețele IPv4

Studiu de caz

Figura 1 prezintă rețeaua IP pe care o vom folosi pentru experimente în prima parte a lucrării. Rețeaua este alcătuită din 5 subrețele, notate SN1, SN2, ..., SN5. Echipamentele conectate la o subrețea comunică între ele folosind protocolul Ethernet, ca și când ar fi conectate la un hub. Prin urmare, puteți captura întregul trafic dintr-o subrețea executând `tcpdump` pw oricare dintre

echipamentele conectate la subrețeaua respectivă.

Rețelei i s-a alocat blocul de adrese IP 10.1.2.0/23. Tabelul 1 descrie modul în care au fost alocate adrese din acest bloc celor 5 subrețele. A doua coloană din tabel specifică dimensiunea fiecărui bloc, determinată de numărul de adrese necesare subrețelei respective. Figura 1 indică blocul de adrese alocat fiecărei subrețele și adresa alocată fiecărei interfețe.

Proiectul `netkit` cu care începeți lucrarea conține topologia din Figura 1, dar echipamentele nu sunt configurate complet.

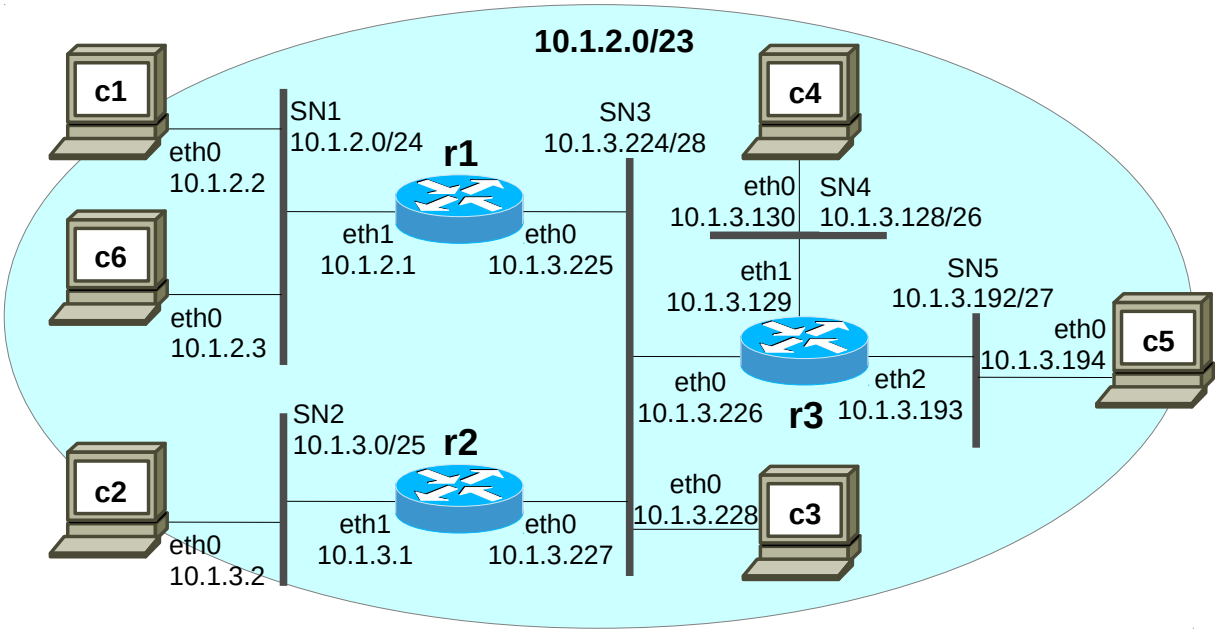

Figura 1: Rețeaua folosită pentru experimente]n prima parte a lucrării.

Tabelul 1: Alocarea adreselor.

Nume subrețea	Număr adrese	Prefixul blocului	Intervalul de adrese	Masca de subrețea
SN1	256	10.1.2.0/24	10.1.2.0 - 10.1.2.255	255.255.255.0
SN2	128	10.1.3.0/25	10.1.3.0 - 10.1.3.127	255.255.255.128
SN3	16	10.1.3.224/28	10.1.3.224 - 10.1.3.239	255.255.255.240
SN4	64	10.1.3.128/26	10.1.3.128 - 10.1.3.191	255.255.255.192
SN5	32	10.1.3.192/27	10.1.3.192 - 10.1.3.223	255.255.255.224

A.1. Inițializarea sistemului și configurarea interfețelor

A.1.1. Porniți emularea rețelei din Figura 1, executând comanda `lstart` într-un terminal al calculatorului gazdă, în directorul în care se află fișierele de configurare `netkit` ale rețelei.

A.1.2. Inițial, interfețele Ethernet și tabelele de rutare ale calculatoarelor și ruterelor nu sunt configurate. Examinați starea lor inițială folosind comenzile `ifconfig` și `route`.

A.1.3. Configurați interfețele tuturor calculatoarelor și ruterelor conform Figurii 1, folosind

comanda ifconfig. De exemplu, pentru calculatorul c1 folosiți comanda următoare (similar pentru celelalte echipamente):

`c1:~# ifconfig eth0 10.1.2.2 netmask 255.255.255.0 up`	activează interfața `eth0`, atribuie adresa IP și masca de subrețea

De ce este necesară specificarea măștii de subrețea pentru fiecare interfață?

A.1.4. După ce ați teminat configurarea interfețelor, examinați starea calculatoarelor și ruterelor folosind comenzile `ifconfig` și `route`.

Sunt activate și configurate corect interfețele, conform Figurii 1? Există în tabelul de rutare al fiecărui echipament rute pentru toate subrețelele conectate direct? Cum au fost obținute aceste rute?

A.1.5. Verificați conectivitatea între echipamente folosind comanda `ping`. Este posibilă comunicația între c1 și c6 sau r1? Dar între c1 și c2? De ce?

A.2. Dirijarea pachetelor în rețele IP

În această lucrare de laborator veți construi tabelele de rutare folosind rute statice (static route), pe care le veți identifica pe baza topologiei rețelei și le veți configura manual.

Obiectivul principal este să înțelegeți ce conține un tabel de rutare IP și cum este folosit pentru dirijarea pachetelor. În lucrarea următoare veți vedea și un exemplu de rutare dinamică, în care tabelul de rutare este construit folosind un protocol de rutare, care descoperă automat destinațiile și rutele. Pe de altă parte, este util să vă familiarizați și cu rutele statice, care își au propriile aplicații. Protocoalele de rutare vor fi studiate în cursul de rețele următor (Rețele și Servicii).

A.2.1. Veți configura mai întâi rutele necesare tabelelor de rutare ale ruterelor. Un ruter trebuie să poată dirija pachete IP care provin de la orice sursă din rețea către orice destinație.

De exemplu, ruterul r1 este conectat direct la subrețelele SN1 și SN3 și cunoaște deja rute către acestea (de ce?). Prin urmare, trebuie să mai adăugați rute către subrețelele SN2, SN4 și SN5. Puteți folosi în acest scop comenzile următoare:

`r1:~# route add -net 10.1.3.0/25 gw 10.1.3.227 dev eth0`	către SN2 via `eth0` și r2
`r1:~# route add -net 10.1.3.128/26 gw 10.1.3.226 dev eth0`	către SN4 via `eth0` și r3
`r1:~# route add -net 10.1.3.192/27 gw 10.1.3.226 dev eth0`	către SN5 via `eth0` și r3

Atributul gw (gateway) indică adresa IP a următorului ruter pe calea cea mai scurtă spre destinație.

Celelalte rutere sunt configurate similar.

De ce este necesar ca aceste rute să specifice atât numele interfeței, cât și adresa IP a ruterului următor pe calea către destinație? Nu ar fi suficient să fie specificată doar interfața pe care trebuie retransmis pachetul? De ce se folosește numele unei interfețe în locul adresei sale MAC sau IP?

A.2.2. După ce ați terminat configurarea ruterelor, examinați și verificați conținutul tabelelor lor de rutare folosind comanda `route`.

Există rute corecte către toate subrețelele? Cum ați putea reduce numărul de rute prin agregare? (Indicație: Observați, de exemplu, rutele configurate pe r1 către SN4 și SN5.) Explicați ce avantaje și dezavantaje are agregarea rutelor, pe baza acestui exemplu.

A.2.3. În continuare, trebuie să mai configurați tabelele de rutare ale calculatoarelor. Fiecare calculator este conectat la o singură subrețea și are deja o rută către aceasta în tabelul său de rutare. De obicei, este suficient să adăugăm pe fiecare calculator o ruta implicită (default route), care specifică faptul că atunci când destinația nu este în subrețeaua conectată direct pachetul este

transmis ruterului care atașează subrețeaua la restul rețelei.

De exemplu, pentru calculatorul c1, ruta implicită se poate configura cu comanda următoare:

`c1:~# route add default gw 10.1.2.1 dev eth0`	implicit via `eth0` și r1

Celelalte calculatoare sunt configurate similar. Pentru c3, specificați ruterul r3.

A.2.4. După ce ați terminat configurarea ruterelor, examinați și verificați conținutul tabelelor lor de rutare folosind comanda `route`.

Folosirea unei rute implicite este un alt exemplu de agregare a rutelor. Explicați ce avantaje oferă utlizarea rutelor implicite, pe baza acestui exemplu.

A.2.5. Verificați conectivitatea între echipamente folosind comanda `ping`. De exemplu:

```
c1:~# ping -c 3 10.1.3.194
```

Este posibil acum transferul pachetelor IP între oricare echipamente din rețea?

A.2.6. Verificați calea urmată de pachete prin rețea folosind comanda `traceroute`. De exemplu:

```
c1:~# traceroute 10.1.3.194
```

Comparați informația afișată de `traceroute` cu conținutul tabelelor de rutare ale ruterelor implicate în transferul pachetelor.

A.3. Transferul direct al pachetelor. Protocolul ARP

În continuare, vom analiza în detaliu comunicațiile care au loc între echipamente pe parcursul livrării pachetelor IP prin rețea. Un pachet IP este dirijat pe calea de la sursă la destinație pe baza adresei sale IP destinație. Calea urmată de pachet este de fapt o concatenare de legături de date care interconectează echipamentele implicate în transfer. În exemplul nostru, echipamentele sunt interconectate prin legături de date multiaccess bazate pe protocolul Ethernet.

Pentru a livra un pachet, modulul IP consultă tabelul de rutare local și identifică ruta pe care trebuie dirijat pachetul pe baza adresei destinație. După cum am văzut, ruta indică interfața și adresa echipamentului următor pe calea către destinație. Odată obținute aceste informații, modulul IP trebuie să apeleze la serviciul de nivel 2 (legătura de date) pentru a transfera pachetul către echipamentul indicat de rută. Mai avem însă de rezolvat o problemă: ruta specifică adresa IP a interfeței echipamentului, iar serviciul și protocolul de nivel 2 are nevoie de adresa de nivel 2 a interfeței. În cazul nostru este vorba de adresa MAC a interfeței Ethernet căreia trebuie să îi fie livrat cadrul Ethernet care conține pachetul IP.

În rețelele IPv4, corespondența între adresa IP și adresa MAC a unei interfețe este deteminată de protocolul ARP (Address Resolution Protocol). În experimentele următoare vom analiza modul în care contribuie acest protocol la livrarea pachetelor IP.

Vom începe cu cazul cel mai simplu: comunicații între echipamente care sunt conectate la aceeași legătură de date (deci la aceeași subrețea IP). În acest caz, pachetul IP poate fi livrat direct de la sursă la destinație folosind protocolul de nivel 2.

A.3.1. Porniți captura traficului pe interfața `eth1` a ruterului r1:

```
r1:~# tcpdump -s0 -ten -i eth1
```

A.3.2. Generați trafic între c1 și c6 folosind `ping`:

```
c1:~# ping -c 2 10.1.2.3
```

A.3.3. Opriți captura pe r1. Examinați cadrele Ethernet capturate de `tcpdump` și explicați secvența de operații efectuate de c1 și c6 în vederea livrării pachetelor IP.

Indicații: Programul `ping` solicită transmiterea de la c1 către c6 a unui mesaj ICMP Echo Request, căruia c6 îi răspunde cu un mesaj ICMP Echo Reply. Acest dialog este repetat de două ori. Mesajele ICMP sunt transmise folosind protocoalele IP și Ethernet. Dialogul ARP Request/Reply permite aflarea adreselor MAC corespunzătoare adreselor IP.

Cum determină c1 faptul că pachetul IP conținând ICMP Echo Request poate fi livrat direct lui c6? Cum află c1 adresa MAC a lui c6? Cum află c6 adresa MAC a lui c1?

De ce apare doar un singur dialog ARP Request/Reply, înainte de transmiterea primului ICMP Echo Request? Mai precis, de ce nu mai este necesar un astfel de dialog pentru transmiterea ICMP Echo Reply sau pentru al doilea ICMP Echo Request?

A.3.4. Examinați conținutul tabelului ARP cache pe c1 și c6 folosind comanda `arp`. Explicați modificările care apar în ARP cache în urma acestei comunicații.

Observație: Dacă înregistrările au dispărut deja, repetați comanda `ping`.

A.4. Transfer indirect al pachetelor

Vom analiza acum comunicații între echipamente care nu sunt conectate direct (nu sunt în aceeași subrețea), prin urmare pachetele sunt livrate prin intermediul unor rutere.

A.4.1. Porniți captura traficului pe interfața `eth1` a ruterului r1, interfața `eth1` a ruterului r2 și interfața `eth0` a ruterului r3:

```
r1:~# tcpdump -s0 -ten -i eth1
```

```
r2:~# tcpdump -s0 -ten -i eth1
```

```
r3:~# tcpdump -s0 -ten -i eth0
```

A.4.2. Generați trafic între c1 și c2 folosind `ping`:

```
c1:~# ping -c 2 10.1.3.2
```

A.4.3. Opriți captura pe r1, r2 și r3. Examinați cadrele Ethernet capturate de `tcpdump` și explicați secvența de operații efectuate de c1, r1, r2 și c2 în vederea livrării pachetelor IP.

Indicații: Pachetele sunt transferate pe calea c1 - r1 - r2 - c2 și retur. Ați capturat traficul pe fiecare legătură de date de pe această cale.

Cum determină c1 (respectiv r1, r2) faptul că trebuie să transmită pachetul IP conținând ICMP Echo Request lui r1 (respectiv r2, c2)? Cum determină c2 (respectiv r2, r1) faptul că trebuie să transmită pachetul IP conținând ICMP Echo Reply lui r2 (respectiv r1, c1)? Cum este determinată adresa MAC, pe fiecare legătură de date, pentru fiecare din cele 4 pachete IP transferate?

A.4.4. Examinați conținutul tabelului ARP cache pe c1, c2, r1 și r2 folosind comanda `arp`. Explicați modificările care apar în ARP cache în urma acestei comunicații.

A.5. Conflicte de adresare IPv4

Ce se întâmplă dacă mai multe echipamente dintr-o subrețea au aceeași adresă? În mod evident aceste echipamente nu vor putea funcționa corect (analizați, de exemplu, situația în care c1 și c6 au aceeași adresă și încearcă să deschidă conexiuni TCP).

Conflictele de adresare IPv4 pot fi detectate folosind protocolul ARP. Procedura a fost standardizată

în RFC 5227 [2]. Vom efectua în cele ce urmează experimente folosind `arping`, unul dintre utilitarele care permit detectarea conflictelor de adresare.

A.5.1. Porniți captura traficului pe interfața `eth1` a ruterului r1:

```
r1:~# tcpdump -s0 -ten -i eth1
```

A.5.2. Modificați adresa calculatorului c1 astfel încât să coincidă cu adresa lui c6:

```
c1:~# ifconfig eth0 10.1.2.3 netmask 255.255.255.0 up
```

A înregistrat `tcpdump` vreun pachet în urma executării acestei comenzi?

Indicații: În mod normal, c1 ar trebui să verifice unicitatea adresei configurate înainte de a o folosi (conform RFC 5227), dar această funcție nu este disponibilă în versiunea Linux folosită în lucrare.

A.5.3. Executați pe c1 comanda `arping` pentru noua sa adresă IP:

```
c1:~# arping -c 3 -S 0.0.0.0 10.1.2.3
```

Indicații: Opțiunea -S permite specificarea adresei IPv4 a transmițătorului în pachetul ARP Request. În acest exemplu, adresa 0.0.0.0 semnifică faptul că transmițătorul nu are încă o adresă IPv4 validă, deoarece testăm unicitatea adresei 10.1.2.3.

Explicați cum funcționează `arping` și ce se întâmplă în acest experiment pe baza traficului capturat de `tcpdump` și a informației afișate de `arping`.

A.5.4. Reveniți la configurarea corectă a adresei calculatorului c1 și repetați testul cu `arping`.

```
c1:~# ifconfig eth0 10.1.2.2 netmask 255.255.255.0 up
c1:~# arping -c 3 10.1.2.3
```

Comparați traficul capturat de `tcpdump` și informația afișată de `arping` în cele două situații și explicați cum poate fi detectat acest tip de conflict de adresare.

A.6. Mesaje ICMP de eroare/diagnostic

IP oferă un serviciu de datagrame fără garanții privind livrarea cu succes la destinație și fără confirmare pozitivă. În anumite situații de eșec, sursa primește o confirmare negativă, printr-un mesaj de eroare al protocolului ICMP. Cel mai frecvent însă, pachetele IP sunt pierdute din cauza congestionării unei legături pe calea către destinație, caz în care nu se transmit mesaje de eroare.

Livrarea poate eșua din motive care țin de funcțiile de adresare și dirijare: un ruter nu are o rută (funcțională) către destinația pachetului, nu există un echipament (în funcțiune) cu adresa respectivă, nu există un proces care să preia pachetul la adresa destinație, expiră contorul TTL (Time-To-Live) din antetul pachetului, etc. În aceste cazuri, echipamentul unde are loc incidentul anunță sursa printr-un mesaj ICMP care indică pachetul pierdut și motivul.

În experimentele următoare vom examina câteva situații de acest tip.

A.6.1. Porniți captura traficului pe interfața `eth1` a ruterului r1:

```
r1:~# tcpdump -s0 -tn -i eth1
```

A.6.2. Executați pe c1 comanda `ping` pentru o adresă din SN5 care nu este alocată:

```
c1:~# ping -c 2 10.1.3.195
```

Ce mesaj ICMP primește c1? Ce echipament transmite acest mesaj? Cum a detectat eroarea?

A.6.3. Încercați să transmiteți o datagramă UDP de la c1 la c5, cu portul destinație 100, folosind

programul `netcat` (`nc`):

```
c1:~# nc -u 10.1.3.194 100
```

Ce mesaj ICMP primeşte c1? Ce echipament transmite acest mesaj? Ce eroare raportează ICMP şi care este cauza acestei erori?

A.6.4. Opriți și reporniți captura pe r1 adăugând opțiunea `-v`, pentru a observa mai multe detalii în pachetele transmise de c1. Executați pe c1 comanda `traceroute` pentru destinația c5:

```
c1:~# traceroute 10.1.3.194
```

Există mai multe variante pentru testul efectuat de `traceroute`. Varianta preferată în Linux transmite datagrame UDP. O altă variantă frecvent utilizată transmite ICMP Echo Request (este varianta preferată în MS Windows). Pentru a doua variantă, repetați testul cu opțiunea `-I icmp`.

Explicați principiul de funcționare al acestei comenzi pe baza traficului capturat: Ce pachete transmite c1? Ce mesaje ICMP primește c1, de la care echipamente și de ce sunt ele transmise?

A.7. Mesaje ICMP de redirectare

Pe calculatorul c3 ați configurat o rută implicită via r3. Evident, r3 nu se află pe ruta cea mai scurtă de la c3 către toate celelalte subrețele. În experimentul următor veți observa ce efecte are această soluție (neglijentă) de configurare asupra rutării pachetelor IP.

A.7.1. Porniți captura traficului pe interfața `eth0` a ruterului r1:

```
r1:~# tcpdump -s0 -i eth0 -w /hostlab/redir.cap
```

A.7.2. Executați pe c3 comanda `ping` pentru destinația c2:

```
c3:~# ping -n -c 3 10.1.3.2
```

A.7.3. Opriți captura şi vizualizați traficul cu `wireshark` (executaţi comanda într-un terminal al calculatorului gazdă):

```
wireshark -r redir.cap
```

Pe ce cale este livrat primul pachet ICMP? De ce transmite r3 lui c3 mesajul ICMP Redirect? Care este efectul acestui mesaj?

Indicaţii: Pachetul care conține primul mesaj ICMP Echo Request este transmis de c3 lui r3, conform rutei implicite. Ruterul r3 consultă tabelul de rutare și determină faptul că pachetul trebuie dirijat prin r2, iar c2 este conectat direct la r2. Prin urmare, r3 transmite pachetul lui r2 și apoi anunță ruta directă sursei pachetului printr-un mesaj ICMP Redirect. După primirea acestui mesaj, c3 memorează temporar ruta indicată, astfel încât transmite următoarele pachete direct lui r2.

Acest mod de funcționare este ineficient. Configurarea calculatorului c3 ar trebui modificată, adăugând, de exemplu, rute statice către SN1 și SN2.

B. Configurarea automată folosind DHCP

În a doua parte a lucrării vom configura automat calculatoarele din subrețeaua SN1 folosind protocolul DHCP. Modificările sunt ilustrate în Figura 2.

Vom rula un server DHCP pe ruterul r1. Calculatoarele din SN1 vor primi de la acest server următorii parametri de configurare: o adresă IP, alocată dinamic dintr-o porțiune a blocului de adrese alocat subrețelei SN1, și masca de subrețea; adresa ruterului (gateway) care conectează subrețeaua la restul rețelei (pentru ruta implicită); adresa serverului DNS local și numele

domeniului local.

Serverul DNS va rula pe calculatorul c4 şi a fost adăugat doar pentru a ilustra posibilitatea de a configura automat clienţii DNS folosind DHCP.

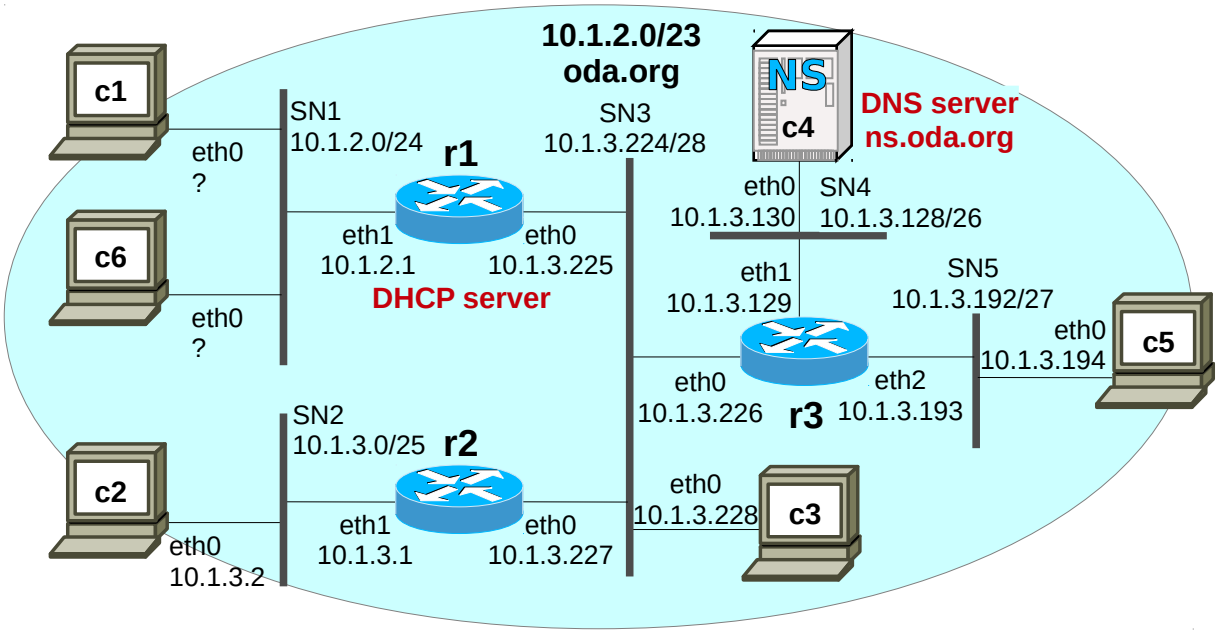

Figura 2: Reţeaua folosită pentru experimente cu DHCP.

B.1. Configurarea serverului DNS

Serverul DNS va rula pe calculatorul c4 şi este configurat ca server autoritar pentru domeniile `oda.org` şi `10.1.in-addr.arpa`.

B.1.1. Examinaţi configurarea serverului DNS:

```
c4:~# cat /etc/bind/named.conf
```

```
c4:~# cat /etc/bind/db.oda.org
```

```
c4:~# cat /etc/bind/db.10.1
```

B.1.2. Porniţi serverul DNS pe calculatorul c4:

```
c4:~# /etc/init.d/bind start
Starting domain name service...: bind9.
```

B.1.3. Verificaţi funcţionarea serverului DNS folosind programul `dig`. De exemplu:

```
c2:~# dig @10.1.3.130 c3.oda.org
```

B.2. Configurarea serverului DHCP

Serverul DHCP va rula pe ruterul r1 şi este deja configurat.

B.2.1. Examinaţi configurarea serverului DHCP:

```
r1:~# cat /etc/dhcp3/dhcpd.conf
default-lease-time 3600;
option domain-name-servers 10.1.3.130;
```

```
option domain-name "oda.org";
subnet 10.1.2.0 netmask 255.255.255.0 {
range 10.1.2.100 10.1.2.254;
option routers 10.1.2.1;
}
subnet 10.1.3.224 netmask 255.255.255.240 {
}
```

Serverul DHCP va livra clienților din subrețeaua SN1 următorii parametri: o adresă IP din intervalul 10.1.2.100 - 10.1.2.25 cu masca de subrețea 255.255.255.0; adresa ruterului care conectează subrețeaua la restul rețelei, 10.1.2.1; adresa serverului DNS, 10.1.3.130 (c4), și numele domeniului, `oda.org`. Adresa IP este acordată pentru 3600 de secunde (lease-time).

B.2.2. Porniți serverul DHCP pe ruterul r1:

```
r1:~# /etc/init.d/dhcp3-server start
Starting DHCP server: dhcpd3.
```

B.3. Configurarea clienților DHCP

Vom modifica configurarea calculatoarelor c1 și c6 pentru a obține parametrii de configurare folosind DHCP.

B.3.1. Anulați configurarea IP existentă pe calculatorul c1:

```
c1:~# ifconfig eth0 0.0.0.0 up
```

Verificați rezultatul folosind comenzile `ifconfig` și `route`.

B.3.2. Porniți captura traficului pe interfața `eth1` a ruterului r1:

```
r1:~# tcpdump -s0 -i eth1 -w /hostlab/r1e1dhcp1.cap
```

B.3.3. Porniți clientul DHCP pentru interfața `eth0` a calculatorului c1:

```
c1:~# dhclient eth0
...
Listening on LPF/eth0/ee:40:19:05:85:e9
Sending on   LPF/eth0/ee:40:19:05:85:e9
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 4
DHCPOFFER from 10.1.2.1
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 10.1.2.1
...
bound to 10.1.2.100 -- renewal in 1395 seconds.
```

Clientul DHCP inițiază procedura de obținere a parametrilor de configurare transmițând un mesaj Discover cu adresa destinație broadcast local (255.255.255.255). Rolul acestui mesaj este de a descoperi un server DHCP și de a lista parametrii doriți.

Serverul DHCP răspunde clientului printr-un mesaj Offer care conține informațiile solicitate. Dacă există mai multe servere, clientul poate primi mai multe oferte și va alege oferta unuia dintre ele.

Procedura se încheie printr-un al doilea dialog în care clientul transmite un mesaj Request, cu adresa destinație broadcast, în care specifică adresa serverului selectat și lista parametrilor doriți.

Acest mesaj este primit de toate serverele. Serverul selectat răspunde clientului cu un mesaj Ack care furnizează informațiile cerute, iar celelalte își anulează oferta.

B.3.4. Opriți captura pe r1. Vizualizați pachetele capturate folosind `wireshark` și explicați operațiile efectuate de clientul DHCP de pe c1 și serverul DHCP de pe r1. În particular, explicați ce operații de verificare a adresei alocate și a rezultatului operației de configurare au fost efectuate.

B.3.5. Examinați starea interfeței calculatorului c1 și tabelul de rutare folosind comenzile `ifconfig` și `route`. Verificați adresa IP a interfeței. Verificați faptul că tabelul de rutare conține o rută către subrețeaua conectată direct și o rută implictă via r1.

B.3.6. Verificați funcționarea corectă a clientului DNS pe c1. De exemplu:

```
c1:~# dig c2.oda.org
```

```
c1:~# ping c3.oda.org
```

```
c1:~# traceroute c5.oda.org
```

B.3.7. Repetați operațiile de mai sus pentru calculatorul c6.

Terminarea lucrării

Executați comanda `lcrash` în terminalul calculatorului gazdă, în directorul în care se află fișierele de configurare `netkit` ale acestei lucrări.

Bibliografie

1. O. Catrina. APC - Note de curs.
2. RFC 5227, IPv4 Address Conflict Detection, 2008.
3. RFC 2131, Dynamic Host Configuration Protocol, 1997.
4. Linux and Unix ifconfig command. http://www.computerhope.com/unix/uifconfi.htm
5. Linux and Unix route command . http://www.computerhope.com/unix/route.htm
6. Linux and Unix dhclient command . http://www.computerhope.com/unix/dhclient.htm
7. dhcpd. https://wiki.archlinux.org/index.php/Dhcpd