

9. SERVERUL DHCP

9.1. Protocolul DHCP.....	2
9.1.1. Descriere generala.....	2
9.1.2. Conceptul de lease.....	2
9.1.3. Tipuri de mesaje DHCP.....	2
9.1.4. Procesul de configurare a clientului.....	3
9.1.4.1. Determinarea serverelor DHCP din retea.....	3
9.1.4.2. Oferta serverelor.....	3
9.1.4.3. Decizia si instiintarea serverului ales.....	4
9.1.4.4. Confirmarea alocarii.....	4
9.1.4.5. Eventuala invalidare a alocarii.....	4
9.1.4.6. Cazul clientilor care au deja adresa obtinuta pe alte cai.....	4
9.1.5. Perioada ulterioara obtinerii configurarii.....	4
9.2. Serverul ISC DHCP.....	5
9.2.1. Cine este ISC?.....	5
9.2.2. Fisa serverului.....	5
9.2.3. Configurare de baza.....	6
9.2.4. Particularizarea setarilor la nivel de statie.....	7
9.2.5. Atribuirea de adrese fixe pentru anumite statii.....	7
9.2.6. Particularizarea setarilor la nivel de grup de statii.....	7
9.2.7. Particularizarea setarilor la nivel de lot de adrese.....	8
9.3. Clientul DHCP.....	9
9.3.1. Clienti DHCP.....	9
9.3.2. Utilizare dhclient.....	10
9.4. BIBLIOGRAFIE.....	10

9.1. Protocolul DHCP

9.1.1. Descriere generala

DHCP (Dynamic Host Configuration Protocol) este un protocol client-server folosit pentru a automatiza configurarea calculatoarelor și a altor dispozitive de rețea care suportă stiva de protocoale TCP/IP (pe care le vom numi în continuare „clienți DHCP”). Prin intermediul DHCP li se pot pasa clienților atât elementele ce formează configurarea de bază a rețelei (adresa IP/netmask, ruta default, servere DNS) cât și mulți alți parametri de configurare, unii dintre ei fiind chiar specifici anumitor sisteme de operare. În acest fel este redus efortul de administrare al rețelei, care altminteri ar fi presupus configurarea manuală a acestor parametri de către administrator pe fiecare client în parte.

DHCP a fost proiectat și dezvoltat sub auspiciile Internet Engineering Task Force (IETF). Microsoft a fost prima firmă care a introdus în 1994 DHCP odată cu lansarea lui Windows NT Server 3.5. Conceptele au fost preluate și de firmele MacIntosh și Novell care au introdus alocarea dinamică a adreselor în rețelele lor.

Nota: pentru documentele IETF care specifică protocolul DHCP consultați bibliografia.

Serverul și clientul comunică prin UDP; serverul folosește portul 67, iar clientul portul 68.

O rețea poate dispune de unul sau mai multe servere DHCP; protocolul este gândit ca un client să poată primi și procesa oferte de la mai multe servere. În schimb, DHCP nu acoperă și modul în care respectivele servere își sincronizează bazele de date cu adrese deja alocate, astfel încât să nu rezulte conflicte - acest aspect este implementat numai în unele softuri de server DHCP.

9.1.2. Conceptul de lease

Protocolul DHCP realizează o alocare temporară a adreselor către clienți, ce poartă denumirea de **lease** (închiriere, împrumut); un client, odată ce a primit o adresă de la server, are permisiunea de a o folosi pentru un interval limitat de timp numit **lease time**. Cât timp clientul mai are nevoie de adresă, el îi poate adresa serverului cereri periodice de prelungire a alocării; când nu o mai folosește, o poate „înapoia” (elibera) explicit.

Avantajul conceptului de lease este acela că adresele pot fi refolosite. În cazul configurării statice (manuale) a adreselor în rețea, dacă o stație este oprită, adresa ei rămâne blocată în tot acest interval, neputând fi folosită automat de alta stație sau dispozitiv TCP/IP; în DHCP, adresa în cauză se eliberează automat după expirarea lease time-ului dacă clientul nu a mai formulat nicio cerere de prelungire între timp.

Refolosirea este vitală în zilele noastre, când dinamica unei rețele poate fi foarte mare. Să luăm exemplul unei rețele wifi publice dintr-un restaurant sau pub: clienții vin și pleacă, timpul lor de staționare în rețea putându-se reduce la câteva minute. Ei își conectează telefonul/tableta la rețeaua wireless, o folosesc pentru un interval de timp iar apoi părăsesc localul, iesind din aria de acoperire. Dacă fiecare device care pătrunde în rețea ar avea alocată o adresă numai a lui, numărul de adrese necesare ar crește până la un nivel impresionant; grație DHCP, dacă un device părăsește rețeaua, la expirarea lease time-ului adresa alocată acestuia revine în „patrimoniul” serverului și poate fi alocată altui client.

9.1.3. Tipuri de mesaje DHCP

Fiecare mesaj DHCP are un tip, care indică operația pe care acesta o realizează. Iată mai jos mesajele DHCP posibile:

- mesaje trimise de catre client
 - **Discover** – folosit pentru descoperirea serverelor DHCP din rețea
 - **Request** – utilizat pentru a-i solicita unui server configurarea propusa anterior de către acesta
 - **Release** – eliberează adresa utilizata de către clientul în cauza, astfel încât aceasta sa poata fi mai apoi alocata altor clienți
 - **Decline** – trimis serverului atunci când clientul constata ca adresa alocata lui e deja în uz în rețea
 - **Inform** – trimis serverului atunci când clientul are deja adresa obtinuta pe alte cai (ex: configurare manuala) și ii solicita serverului doar ceilalți parametri de configurare (ruta default, servere DNS etc)
- mesaje trimise de către server
 - **Offer** – conține o propunere de configurare (oferta) trimisa ca urmare a unei solicitari din partea unui client
 - **Acknowledge (ACK)** – indica acceptarea unei solicitari a clientului
 - **NegativeAcknowledge (NACK)** – indica respingerea unei solicitari a clientului

Prezentam în continuare felul în care sunt folosite aceste mesaje pentru implementarea serviciilor DHCP.

9.1.4. Procesul de configurare a clientului

9.1.4.1. Determinarea serverelor DHCP din retea

Clientul este initial neconfigurat și nu cunoaște nimic despre rețeaua în care se afla. De aceea, în prima faza, el trebuie sa afle care sunt serverele DHCP ale rețelei, iar acest lucru poate fi realizat numai pe baza de broadcast; practic, clientul „striga” în toată rețeaua, cautand servere DHCP. În acest scop el trimite un mesaj DHCP de tip **Discover**, având ca destinație adresa de broadcast local (255.255.255.255). Fiecare stație din rețeaua locala primește pachetul; vor răspunde numai serverele DHCP care au adrese disponibile pentru acest client.

***Nota:** fiind un broadcast local, pachetul de tip Discover nu trece prin router spre alte rețele sau subrețele. Dacă router-ul suportă RFC 3046 (suport pentru DHCP/BOOTP relay) și este configurat corespunzator, va înainta pachetele și altor rețele sau subrețele, în calitate de BOOTP relay agent.*

Prin intermediul optiunilor prezente în mesajul *Discover*, clientul are posibilitatea de a solicita o anumita adresa (de obicei, ultima primita) și/sau un anumit lease time sau alți parametri de configurare. Serverul va încerca sa răspundă pretentiilor clientului, însă nu are nicio obligație în acest sens: dacă, spre exemplu, nu ii poate acorda clientului adresa ceruta de acesta, ii va răspunde cu un mesaj de tip *Negative Acknowledge*.

9.1.4.2. Oferta serverelor

Odata ce un client a trimis o solicitare de tip *Discover*, fiecare server care ii poate oferi o configurare clientului in cauza va raspunde cu un mesaj de tip **Offer**. Acesta contine adresa pe care serverul i-o poate oferi clientului, impreuna cu alti posibili parametri de configurare (lease time, ruta default etc). Parametrii in cauza au rolul de a-l ajuta pe client sa decida ce oferta va accepta in cazul in care mai multe servere raspund solicitarii.

***Nota:** deoarece clientul nu are încă o adresă IP, pachetele “DHCP offer” sunt și ele de tip broadcast.*

Optional, adresa oferita poate fi rezervata temporar de către server, astfel încât sa nu poată fi utilizata sau oferita altor clienti până când clientul „se hotaraste” sau rezervarea expira.

9.1.4.3. Luarea deciziei si instiintarea serverului ales

Clientul poate primi mai multe oferte dar accepta numai una. Decizia acceptarii poate fi luata pe baze simple (prima oferta sosita) sau complexe (oferta care a corespuns cel mai bine parametrilor doriti de client). Odata decizia luata, clientul trimite un pachet broadcast de tip **Request** ce contine, printre altele, adresa serverului ales si care are doua roluri:

- instiinteaza serverul ales despre decizia clientului si ii solicita alocarea adresei promise, impreuna cu alti eventuali parametri necesari
- instiinteaza celelalte servere care au trimis oferte ca nu au fost alese, astfel incat sa-si elibereze eventualele resurse rezervate ca urmare a ofertelor trimise clientului

9.1.4.4. Confirmarea alocarii

La primirea mesajului *Request*, serverul ales va procesa solicitarea clientului, marcand adresa ca fiind „in uz” și trimitându-i înapoi un mesaj de tip **ACK (acknowledge)**. Acesta conține adresa alocata de server împreuna cu restul de parametri solicitati de client. Din acest moment clientul intra în starea *bound* - are drept de folosinta a adresei pe perioada lease time-ului.

9.1.4.5. Eventuala invalidare a alocarii

Desi in cele mai dese cazuri intreaga tranzactie se incheie cu un *Acknowledge* din partea serverului si clientul poate incepe sa foloseasca configurarea alocata, exista doua situatii in care procesul esueaza in etapa sa finala, dupa transmiterea mesajului *Request*:

- in cazul in care serverul nu a rezervat adresa in momentul formularii ofertei catre client si respectiva adresa a fost intre timp alocata altuia, serverul ii va trimite clientului un mesaj de tip **NACK (Negative Acknowledge)** instiintandu-l ca oferta nu mai este valabila. Ca urmare, clientul va relua procesul de configurare
- inainte de a incepe sa utilizeze o adresa primita de la server, clientii DHCP efectueaza verificarea daca aceasta adresa nu este deja in uz in retea (spre exemplu, in cazul in care a fost configurata manual pe una dintre celelalte statii). In caz afirmativ, clientul respinge oferta serverului printr-un mesaj de tip **Decline** si reia procesul de configurare automata

9.1.4.6. Cazul clientilor care au deja adresa obtinuta pe alte cai

Exista situatii în care un client are o adresa deja configurata (manual sau prin alte mijloace, non-DHCP) și dorește sa obțină doar parametri suplimentari de la un sever DHCP. Într-un astfel de caz clientul ii va trimite serverului un mesaj de tip **Inform** ce conține lista parametrilor necesari. În caz de succes al operatiunii, serverul va răspunde cu un *Acknowledge*.

Nota: prin intermediul acestui tip de mesaj clientul nu poate cere reînnoirea lease-ului, ci doar modificarea/reînnoirea altor parametri.

9.1.5. Perioada ulterioara obtinerii configurarii

Din momentul in care clientul a receptionat confirmarea serverului, cele doua statii nu vor mai comunica pentru un interval de timp si este mentinuta practic o dubla evidenta:

- serverul a marcat adresa ca fiind alocata si nu o va acorda altui client pe durata lease time-ului
- clientul stie ca are drept de folosinta pe aceeasi perioada de timp si va contacta serverul doar pentru eventuala prelungire, intre timp folosind adresa dupa bunul sau plac

Prima ocazie cand cele doua statii dialogheaza din nou este la 50% din intervalul de lease. Daca clientul constata ca inca mai are nevoie de adresa, el ii va trimite serverului o cerere de prelungire lease-ului (*renew*), care ia forma unui mesaj de tip *Request*. Daca serverul este accesibil, va raspunde cu un *Acknowledge* si din acel moment clientul are iarasi drept de folosinta asupra adresei pentru perioada unui lease time.

Nota: in cazul in care serverul nu ii poate prelungi lease-ul clientului, va raspunde cu un *NACK* si clientul va relua intregul proces de obtinere a unei adrese.

Daca serverul nu raspunde la cererea de *renew* (pentru ca este offline, sau are o problema de conectivitate etc), clientul va continua sa incerce sa contacteze acelasi server pana la scurgerea a 87.5% (7/8) din perioada de lease, folosind mesaje unicast (adresate cu exactitate serverului in cauza). Daca serverul tot nu raspunde, de la 7/8 incolo mesajul devine broadcast, clientul incercand sa obtina prelungirea lease-ului de la orice server dispus sa-i ofere acest serviciu. Vor raspunde doar serverele care au cunostinta de lease-ul in cauza.

Daca, pana la expirarea perioadei de lease, niciun server nu i-a prelungit lease-ul curent, clientul se deconfigureaza si reia intregul proces de obtinere a unei adrese.

Nota: oricând pe perioada lease-ului, clientul poate sa renunte timpuriu la adresa obtinuta trimitandu-i serverului un mesaj de tip **Release**. Acesta are ca efect eliberarea adresei, ea putand fi astfel alocata altor clienți.

9.2. Serverul ISC DHCP

9.2.1. Cine este ISC?

ISC (Internet Systems Consortium) este o organizatie implicata intr-o intreaga serie de tehnologii/aspecte importante ale internetului zilelor noastre:

- patroneaza doua aplicatii open source majore: serverul ISC DHCP si celebrul server DNS BIND
- a contribuit cu peste 60 de documente RFC la baza de date IETF
- mentine unul dintre serverele DNS radacina (vezi materialul dedicat serverului DNS)
- ajuta la mentinerea domeniilor DNS a peste 50 de tari

9.2.2. Fisa serverului

Fisa serverului reprezintă setul de informații de pornire despre serverul în cauza, care vor fi necesare/utile pe parcursul sectiunilor următoare.

Iata cateva elemente de baza despre serverul ISC DHCP:

- executabil: **dhcpcd** (DHCP daemon)
- fisier de configurare: **dhcpcd.conf** (aflat de obicei în */etc/* sau */etc/dhcp/*)
- baza de date cu alocarile de adrese (lease-urile) trecute și curente: **/var/state/dhcp/dhpcpd.leases**
- verificarea validitatii fisierului de configurare: **dhcpcd -t**
- verificarea validitatii bazei de date cu alocari de adrese (lease database): **dhcpcd -T**
- rularea serverului in mod debug si mentinerea sa in foreground dupa pornire: **dhcpcd -f -d**. In acest mod de lucru, procesul server – care in mod normal ruleaza in background, eliberand instantaneu promptul - nu se mai decupleaza de la terminalul curent, ci ramane in foreground si ofera un output detaliat ce include si mesajele DHCP schimbate intre server si clienți. Reprezintă una dintre modalitatile foarte eficiente de diagnosticare a problemelor de configurare a serverului

9.2.3. Configurare de baza

Scopul primar al serverului DHCP este acela de alocare de adrese; de aceea, în cazul sau, putem începe cu o configurare de baza, la care vom adauga apoi altele în funcție de serviciile suplimentare oferite de server. În cazul altor servere configurarea depinde în intergralitatea sa de scopul serverului și nu se poate porni de la o configurare minimala.

Configurarea de pornire este următoarea:

```
authoritative;                # serverul "oficial" al rețelei (vezi explicatii mai jos)
default-lease-time 3600;     # daca clientul nu solicita alta perioada , i se ofera aceasta
max-lease-time 86400;       # perioada maxima de lease pe care o poate solicita clientul
subnet 10.0.0.0 netmask 255.255.255.0 {      # pe ce interfata/in ce retea actioneaza serverul
    range 10.0.0.11 10.0.0.28;             # domeniul de adrese alocabile dinamic
    option routers 10.0.0.10;              # default gateway
    option domain-name-servers 8.8.8.8,8.8.4.4; #serverele DNS
}
```

Dupa cum constatom, fisierul de configurare contine doua tipuri de directive:

- directive simple, de o linie, care se incheie intotdeauna cu ; (punct si virgula)
- directive bloc, al caror scop este sa contina in interiorul lor alte directive, conditionand/restrictionand astfel efectul acestora. Corpul unei directive bloc este cuprins intre acolade.

Iata explicatiile aferente fisierului de configurare de mai sus:

- declaratia **subnet** are doua roluri:
 - stabileste pe ce interfata va asculta serverul. **Atentie! Este obligatoriu ca declaratia de subnet sa se suprapuna peste una dintre adresele configurate pe server!** In caz contrar, la pornire serverul nu va asculta pe nicio interfata. In cazul exemplului de mai sus, daca serverul are o interfata numita eth0 cu adresa 192.168.0.1, mesajul afisat la pornire va fi "No subnet declaration for eth0 (192.168.0.1)" urmat apoi de "Not configured to listen on any interfaces"
 - stabileste subnet-ul deservit de catre serverul DHCP (vezi mai jos explicarea directivei *authoritative*). A nu se confunda acest subnet cu setul de adrese pe care serverul le acorda clientilor, si care se specifica cu directiva *range*!
- declaratia **authoritative** face serverul sa trimita mesaje DHCP NACK clientilor care cer prelungirea unei configurari care nu face parte din subretea definita pe acea interfata cu ajutorul directivei *subnet*. Clientul este astfel fortat sa renunte la configurarea sa curenta și sa reia procesul de obtinere a unei adrese. In cazul in care declaratia *authoritative* nu este prezenta sau este scrisa explicit ca *not authoritative*, serverul nu raspunde in astfel de situatii si clientul va pastra configurarea eronata pana la expirarea acesteia. Un server ne-autoritativ este unul conștient de faptul ca nu este el serverul principal al rețelei și care în consecinta încearcă să fie „discret” și sa nu produca stricacuni în rețeaua din care face parte
- declaratia **range** specifica lotul/loturile de adrese alocabile dinamic clientilor. In cazul mai multor loturi de adrese se vor folosi mai multe declaratii *range*. Adresele din range trebuie să facă parte din rețeaua declarata în directiva *subnet*!
- directivele **max-lease-time** și **default-lease-time** influenteaza lungimea perioadei de lease primita de către client; valorile numerice sunt masurate în secunde. Daca clientul nu solicita un anumit lease time, i se va oferi cel default; daca solicita unul mai mic decat cel maxim, ii va fi oferit acela; in schimb, daca lease time-ul solicitat depaseste *max-lease-time*, serverul ii va oferi clientului durata specificata in aceasta directiva (86400 de secunde in cazul nostru)
- directiva **option routers** este utilizata pentru a stabili default gateway-ul clientilor
- directiva **option domain-name-servers** stabileste serverul/serverele DNS comunicate clientilor. **Atenție!** În cazul mai multor servere, ele trebuie separate prin virgula, fără spatii!

9.2.4. Particularizarea setarilor la nivel de statie

Exista situatii în care dorim să-i oferim unui client setari diferite de ale restului rețelei; poate vrem sa aibă un lease time mai mare, un gateway/DNS-uri diferite etc. În acest scop, în fișierul de configurare DHCP vom folosi o directiva bloc **host** care conține adresa MAC a celui client (singurul mod de identificare a unui client neconfigurat) și setarile ce i se aplica clientului în cauza.

Iata un exemplu in care statiei *server* i se paseaza un alt default gateway si alt set de servere DNS decât restului:

```
host server {                                #configuratie ce se aplica unui anume host
  hardware ethernet 00:04:E2:33:9E:3C;     # ...identificat prin adresa sa MAC
  option domain-name-servers 10.0.0.2,10.0.0.3; # ...care va avea un alt set de servere DNS
  option routers 10.0.0.11;                #...si primeste un default gateway diferit
}
```

Numele ce urmeaza cuvintului cheie *host* este folosit pentru identificarea comoda a clientului in cadrul fisierului de configurare; poate sa coincida sau nu cu numele real al statiei client.

Nota: declaratiile *host* sunt globale – nu conteaza daca sunt incluse sau nu in interiorul unei declaratii subnet.

9.2.5. Atribuirea de adrese fixe pentru anumite statii

DHCP nu presupune neaparat lipsa controlului asupra adreselor atribuite clientilor. Exista scenarii in care dorim ca o aceeasi statie, desi configurata prin DHCP, sa primeasca permanent aceeași adresa – spre exemplu, daca este vorba de un server, o imprimanta de retea etc. Intr-un astfel de caz, serverul DHCP poate fi configurat sa-i rezerve o anumita adresa statiei in cauza, identificand statia dupa adresa sa fizica (adresa MAC, in cazul Ethernet).

In acest scop se creeaza in *dhcpd.conf* o declaratie *host* ce cuprinde adresa MAC a statiei si adresa fixa rezervata ei:

```
host server {                                # configuratie ce se aplica unui anume host
  hardware ethernet 00:04:E2:33:9E:3C;     # ...identificat prin adresa sa MAC;
  fixed-address 10.0.0.9;                   # acest host va primi intotdeauna aceeași adresa
}
```

Nota: dacă dorim sa avem garantia ca acea adresa ii va fi alocata numai statiei în cauza și niciunui alt client, vom evita sa includem adresa cu pricina în range.

9.2.6. Particularizarea setarilor la nivel de grup de statii

Fie cazul unei rețele formate din 4 statii - doua de la contabilitate, doua de la resurse umane; dorim ca resursele umane și contabilitatea sa iasă prin provideri diferiți. Pentru a le aloca clientilor rute default diferite, am putea folosi câte o declaratie *host* pentru fiecare stație în parte, specificand gateway-ul ce i se aplica:

```
host conta1{
  hardware ethernet 11:22:33:44:55:66;
  option routers 1.1.1.1;
}
host conta2{
  hardware ethernet 11:22:33:66:55:44;
  option routers 1.1.1.1;
```

```

}
host hr1{
    hardware ethernet 77:88:99:aa:bb:cc;
    option routers 2.2.2.2;
}
host hr2{
    hardware ethernet dd:ee:ff:00:11:22;
    option routers 2.2.2.2;
}
    
```

După cum se observa, directiva *option routers* trebuie repetată în mod identic pentru toate stațiile ce fac parte dintr-un departament. Dacă contabilitatea ar conține 20 de stații, am scrie de 20 de ori directiva *option routers 1.1.1.1* - ca să nu mai spunem că, în viața reală, probabil ar exista și alți parametri (ex: servere DNS) care să se pasezeze acelui grup de stații și care vor fi multiplicați la rândul lor.

Pentru eficientizarea configurării în situații de acest fel - și anume particularizarea setărilor pentru grupuri de stații - putem folosi în *dhcpd.conf* directiva **group**, după cum urmează:

```

# statii care ies prin provider 1 - gateway 1.1.1.1
group {
    option routers 1.1.1.1;
    host conta1    { hardware ethernet 11:22:33:44:55:66; }
    host conta2    { hardware ethernet 11:22:33:66:55:44; }
}
# statii care ies prin provider 2 - gateway 2.2.2.2
group {
    option routers 2.2.2.2;
    option domain-name-servers 2.2.2.100,2.2.2.200;
    host hr1       { hardware ethernet 77:88:99:aa:bb:cc; }
    host hr2       { hardware ethernet dd:ee:ff:00:11:22; }
}
    
```

Fiecare grup de stații constă dintr-o serie de declarații **host** ce definesc stațiile din grup (identificate după adresele MAC) și apoi setările ce li se aplică acestor stații.

Deși procedul nu ne scutește de specificarea adreselor MAC ale stațiilor ce formează grupul, avem totuși avantajul că setările unui grup nu mai trebuie multiplicate, ci apar într-un singur exemplar în fișierul de configurare.

9.2.7. Particularizarea setărilor la nivel de lot de adrese

Există situații în care dorim să le alocăm clienților adrese din mai multe loturi, însă cu două restricții:

- fiecare lot de adrese să aibă atasate anumite setări specifice (gateway, servere DNS etc)
- să putem controla din ce lot primește adresa fiecare client (caci, în funcție de asta, setările pe care le recepționează vor diferi)

Să luăm ca exemplu rețeaua unei firme, formată din stații desktop; în plus față de ele, în rețea apar, meteoric, stațiile/device-urile mobile ale diversilor vizitatori ai firmei. Se ia următoarea decizie:

- stațiile desktop vor primi adrese dintr-un lot privilegiat, care să folosească un anumit default gateway și să beneficieze de un lease time mai mare
- clienții ce au rolul de vizitator vor capata adrese dintr-un alt lot, ce primește alt gateway și un lease time mai mic, astfel încât adresele să fie refofolosite rapid după ieșirea din rețea a vizitatorului în cauză

Pentru gestionarea unei astfel de situații este necesar în primul rând să se poată face distincție între stațiile desktop (permanente) și cele de vizitator. Acest lucru se poate realiza în *dhcpd.conf* prin crearea câte unei

declaratii **host** pentru fiecare statie desktop, care sa contina adresa MAC a statiei in cauza (si atat!). Statiile care beneficiaza de declaratie host in dhcpd.conf au regimul de “*known clients*” din punct de vedere al serverului; putem lua decizii un functie de acest statut al statiei, dupa cum se va vedea mai jos.

Atentie! Declaratiile host sunt globale - vor conta indiferent de contextul in care sunt plasate! Un client devine known client indiferent daca declaratia sa host este plasata in cadrul sau in afara vreunei directive bloc (subnet, pool etc).

In al doilea rand, trebuie sa definim cele doua loturi de adrese, impreuna cu setarile atasate lor. Acest lucru se realizeaza cu ajutorul directivei **pool**.

Iată configurarea rezultanta:

```
pool {                                # pentru desktop-uri
    range 10.0.0.1 10.0.0.200;
    max-lease-time 86400;             # 1 zi
    deny unknown-clients;            # interzis statiilor care nu au declaratie host
}
pool {
    range 10.0.0.201 10.0.0.220;
    max-lease-time 3600;              # 1 ora
    deny known-clients;              # de aici primesc adrese numai vizitatorii
}
host conta1 { hardware ethernet 11:22:33:44:55:66; }
host conta2 { hardware ethernet 11:22:33:66:55:44; }
host hr1     { hardware ethernet 77:88:99:aa:bb:cc; }
host hr2     { hardware ethernet dd:ee:ff:00:11:22; } # ...cei 4 known-clients
```

In exemplul de mai sus, *known-clients* desemneaza toate acele statii care au declaratie host corespunzatoare in cadrul fisierului de configurare dhcpd.conf (conta1/2 și hr1/2), iar *unknown-clients* toate celelalte statii.

Putem avea doar declaratii **allow**, doar **deny** sau o combinatie:

- daca avem doar declaratii *allow*, vor primi adrese din acel lot doar clientii specificati
- daca avem numai declaratii *deny*, vor primi adrese din acel lot numai clientii care NU sunt in lista de *deny*
- daca exista ambele tipuri de declaratii, li se vor atribui adrese din acel lot numai clientilor care apar pe lista de *allow* dar NU si pe lista de *deny*

9.3. Clientul DHCP

9.3.1. Clienti DHCP

Pentru a configura o interfata de retea prin DHCP trebuie sa existe un soft (fie parte a sistemului de operare, fie separat) care sa „vorbeasca” protocolul DHCP si sa parcurga etapele necesare configurarii interfetelor de retea ale statiei pe care ruleaza. Acest tip de soft este in general unul rezident in memorie, deoarece rolul sau nu se incheie odata cu obtinerea adresei, ci continua cu eventualele prelungiri ale perioadei de lease, eliberarea adresei folosite etc.

In Linux pot fi intalniti diversi clienti DHCP:

- **dhclient** - cel inclus in ISC DHCP și folosit de majoritatea distributiilor Linux
- **dhcpcd** (<http://roy.marples.name/projects/dhcpcd/index>). A nu se confunda cu dhcpd, care este serverul!
- **pump**

Prezentam in continuare modul de utilizare al lui *dhclient*.

9.3.2. Utilizare dhclient

Acest utilitar face parte din ISC DHCP si se instaleaza fie odata cu serverul, fie separat de acesta; multe distributii au un pachet *dhclient* separata care este instalat implicit, în scopul configurarii dinamice a interfetelor de rețea cu ajutorul DHCP.

Iata cum se realizeaza cu ajutorul lui *dhclient* principalele operatii ce țin de DHCP:

- configurarea unei interfete folosind DHCP: **dhclient -v nume_interfata**
- anularea timpurie a lease-ului de catre client si eliberarea adresei: **dhclient -r nume_interfata**

Nota: în scopuri de diagnostic, clientul poate fi forțat sa ruleze în foreground folosind **dhclient -d nume_interfata** (optiunea *-d* activeaza “debug mode”)

Utilitarul *dhclient* dispune de un fisier de configurare: **/etc/dhclient.conf**. In acest fisier pot fi stabiliti diferiti parametri:

- **timeout nr_secunde;** - specifica dupa cat timp de asteptare si incercari clientul renunta la a mai incerca sa obtina o adresa
- **interface „nume” {...directive specifice...}** - realizeaza diferentierea configuratiei in functie de interfata configurata

Configurarea curenta este memorata de către *dhclient* în fișierul */var/state/db/dhclient.leases*. În acest fel configurarea supraviețuiește la reboot și clientul poate solicita după pornire aceeași adresa.

9.4. BIBLIOGRAFIE

- ISC DHCP Home Page: <https://www.isc.org/software/dhcp>
- Protocolul DHCP
 - http://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol
 - http://www.tcpipguide.com/free/t_TCPIPDynamicHostConfigurationProtocolDHCP.htm
- Documente RFC (Request for Comments) de la IETF:
 - RFC 2131 : DHCP - <https://tools.ietf.org/html/rfc2131>
 - RFC 2132 : DHCP options and BOOTP Vendor extensions - <https://tools.ietf.org/html/rfc2132>
 - RFC 3046 : DHCP Relay Agent - <https://tools.ietf.org/html/rfc3046>
- The DHCP Handbook 2nd Ed (Ralph Droms, Ted Lemon)