

Capitolul 1: Introducere în concepte de rutare



Obiective

- Ce este un ruter și ce funcții îndeplinește
- Componente hardware și software ale ruterului
- Procesul de rutare, tipuri de rute, principii și implementări ale rutării



Ce este un ruter?

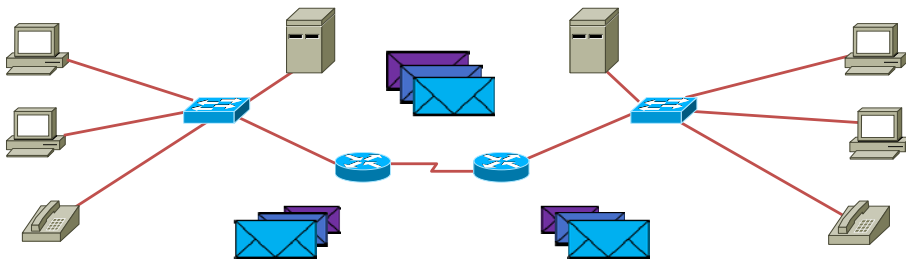
- Un ruter este un computer cu funcții specializate
- Constituie „centrul” unei rețele, din punct de vedere logic



Ruterul este un echipament de nivel 3 (nivel Rețea) care realizează conectivitatea între mai multe rețele (subnet-uri) diferite. Termenul generic de „Router” poate descrie: un computer care are un sistem de operare cu funcționalități avansate de „networking” sau un dispozitiv specializat, cu o formă standardizată, dedicat pentru procesul de transmitere a pachetelor între mai multe rețele. Principalii producători de rutere folosite în medii de producție sunt Cisco, Juniper, Dlink, Huawei și alții. Deși vin de la producători diferiți, majoritatea echipamentelor respectă anumite standarde definite de organisme internaționale, permițând interoperabilitatea acestora.

Ce funcții oferă un ruter?

- Interconectează două sau mai multe rețele
- Are două funcții principale
 - determinarea căii optime
 - trimiterea pachetelor către destinație, folosind calea optimă determinată
- Multiple funcții secundare: manipulare, filtrare și modificare a pachetelor



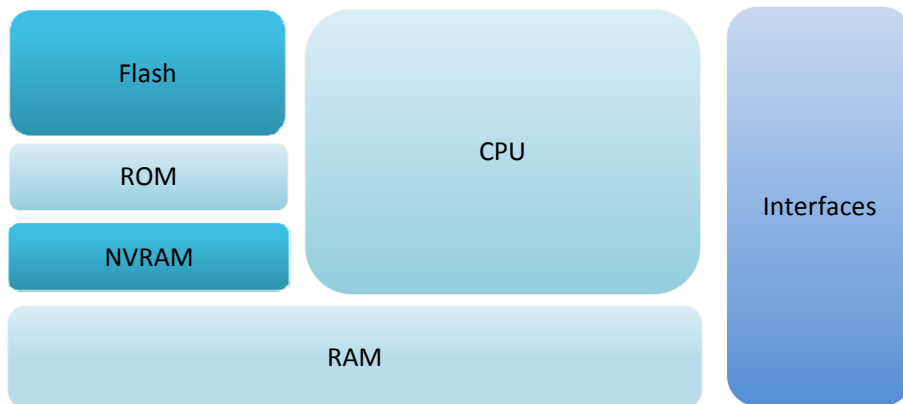
Principalul scop al unui ruter este procesul de „routing” al pachetelor între mai multe rețele. „Routing” înseamnă analizarea antetelor pachetelor de nivel 3 și luarea, pe baza a multiple reguli, a unei decizii cu privire la interfața pe care va fi trimis acesta. Determinarea căii se face prin mai multe metode, și anume rutare statică și rutare dinamică.

În cazul rutării statice administratorul decide calea pe care o va urma un pachet, iar în cazul celei dinamice calea este decisă automat pe baza unor protocoale de rutare. Din punct de vedere practic, un ruter poate fi considerat un dispozitiv care funcționează până la nivelul 7 (nivelul Aplicație), deoarece poate avea și funcții de filtrare ale pachetelor, asigurarea unei anumite calități a serviciilor și modificarea antetelor pachetelor de nivel 4 (nivelul Transport).

Componente hardware



Componente Hardware



Din punct de vedere hardware, un ruter are aproximativ aceleași componente principale ca și un PC, dar proiectate special pentru un anumit scop și integrate corespunzător într-un „form-factor” al unui ruter. Toate componentele, deși similare cu cele ale unui PC normal, sunt optimizate pentru a realiza cât mai rapid transmiterea pachetelor între mai multe rețele.

Operațiile cum ar fi analizarea antetelor de nivel 3 sau parcurgerea logică a destinațiilor posibile în tabela de rutare se fac într-un timp exponențial mai scurt decât pe un echipament normal. Pe lângă CPU și RAM, ruterele au circuite specializate numite ASIC (Application-specific integrated circuit) pentru realizarea cât mai rapidă a operațiilor critice. Prezența acestor circuite suplimentare depinde de modelul ruterului, destinația acestuia și volumul de pachete la care trebuie să facă față.

Componente Hardware (1)

- CPU
 - execută instrucțiuni de la sistemul de operare
- RAM (memorie volatilă)
 - instanța sistemului de operare și configurările curente
 - tabela de rutare și tabela ARP
 - buffer de pachete
- ROM (memorie non-volatilă și nealterabilă)
 - programul de pornire (bootstrap)
 - utilitare de diagnosticare și un SO minimal

„CPU”-ul reprezintă principala componentă internă a unui ruter, întâlnit în alcătuirea oricărui dispozitiv digital. Procesează pachetele și diferitele interogări, ajungând și la 10.000 pachete în cazul celor mai performante modele.

Memoria RAM este utilizată ca buffer de pachete, pentru memorarea temporară a datelor primite sau de trimis pe o interfață, pentru salvarea fișierului running-config, a tabelii de rutare și a tabelii ARP.

Memoria ROM (Read-Only Memory) nu poate fi suprascrisă și conține date critice pentru procesul de boot al ruterului. Conține un sistem de operare minimal (RomMonitor) care poate fi folosit pentru operații elementare, cum ar fi recuperarea parolei, înlocuirea sistemului de operare IOS sau alte task-uri administrative.

Componente Hardware (2)

- Flash (memorie non-volatilă)
 - conține imaginea unuia sau a mai multor sisteme de operare
 - poate conține și alte fișiere
- NVRAM (memorie non-volatilă)
 - stochează fișierul de configurare încărcat în RAM la pornire

Memoria flash, de obicei în cantitate de 64/128 MB conține în principal imaginea sistemului de operare IOS și alte fișiere esențiale pentru funcționarea acestuia. Similar cu un sistem de operare „full-fledged”, se pot procesa și memora orice fișiere în cadrul memoriei flash. De exemplu, în memoria Flash se pot memora mai multe versiuni ale configurației inițiale pentru diverse situații sau scenarii.

Memoria NVRAM este utilizată în principal pentru stocarea fișierului de configurare încărcat în RAM la pornirea ruterului (start-up config). Atunci când se execută modificări în fișierul running-config al unui ruter, se recomandă salvarea periodică a acestuia în fișierul start-up config (adică în NVRAM), pentru evitarea pierderii configurațiilor în cazul restartării ruterului. De obicei, memoria NVRAM este foarte mică (aproximativ 32 Kb).

Componente Hardware (3)



- Interfețe de management
 - portul de consolă, portul AUX
- Interfețe de date
 - dispozitive fizice de transmisie a pachetelor de date
 - fiecare interfață aparține unei rețele diferite
 - interfețe LAN: Ethernet, FastEthernet etc.
 - interfețe WAN: Serial, ISDN etc.

Portul de consolă este utilizat pentru conectarea directă la ruter, printr-o legătură serială. Acesta este singurul mod care poate fi utilizat pentru configurarea unui ruter nou.

Portul auxiliar îndeplinește același funcții ca și cel de consolă.

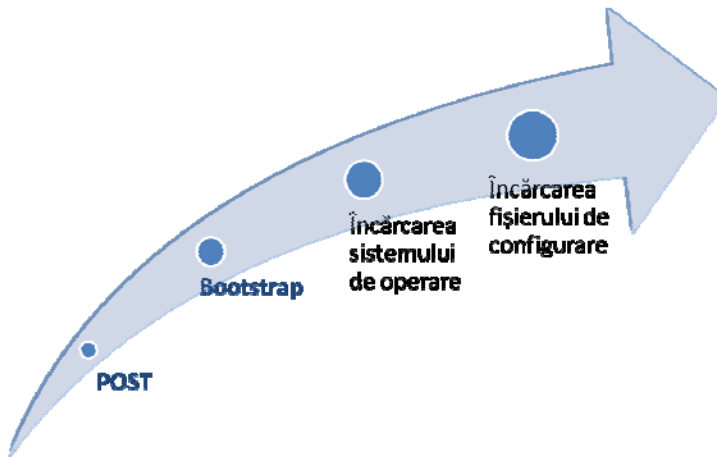
Interfețele de date au rolul de a conecta ruter-ul la rețele diferite (fiecare interfață aparține unei alte rețele). Aceste interfețe pot fi:

- Interfețe LAN, pentru conectarea în interiorul rețelei locale
- Interfețe WAN, pentru conectarea între locații distante geografic

În ziua de azi, cele mai utilizate și versatile interfețe sunt cele Ethernet, cu multiple variații: FastEthernet, GigabitEthernet, 10Gbit Ethernet.

Conexiunile seriale sunt deseori utilizate în condiții de laborator sau în spații academice pentru exemplificarea diverselor opțiuni disponibile pe aceste interfețe.

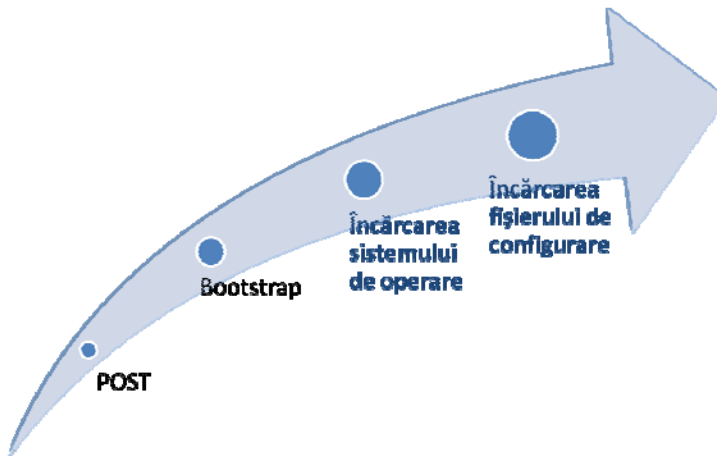
Inițializarea ruterului (1)



Procesul de boot-are al unui ruter este alcătuit din patru faze majore:

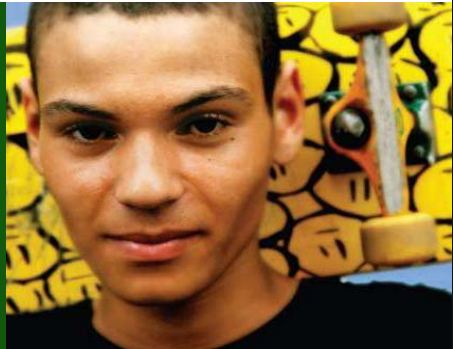
- POST (Power-On Self Test): un proces folosit pentru a descoperi și a testa funcționarea componentelor hardware ale unui ruter
- Bootstrap: după procesul de POST programul de bootstrap este copiat din memoria ROM în memoria RAM, instrucțiunile specificate de acesta fiind apoi executate de CPU; scopul principal al programului de bootstrap este localizarea IOS-ului Cisco și încărcarea acestuia în RAM; imaginea IOS-ului este de obicei stocată în memoria flash a ruterului sau într-o locație externă, de exemplu un server TFTP (Trivial File Transfer Protocol)

Inițializarea ruterului (2)



- **Încărcarea sistemului de operare:** în cazul în care imaginea IOS nu poate fi localizată în locațiile menționate anterior, o versiune minimală a IOS-ului va fi copiată din ROM în RAM pentru a descoperi și rezolva eventuale probleme existente; unele modele mai vechi de rutere Cisco inițializează sistemul de operare direct de pe flash, spre deosebire de modelele actuale care copiază IOS-ul în memoria RAM
 - **Încărcarea fișierului de configurare:** după încărcarea IOS-ului, programul de bootstrap va căuta în NVRAM fișierul de configurare numit startup-config, care conține configurațiile salvate anterior pe ruter; ulterior localizării în NVRAM, fișierul de configurare este copiat în RAM sub numele de running-config; în cazul în care fișierul de configurare nu este găsit în NVRAM, ruterul poate realiza o căutare pe un server TFTP; dacă nu este încărcat din nici una dintre locațiile menționate, ruterul va porni utilitarul de setup, un mod interactiv de setare a unor funcționalități de bază
-
-
-
-
-

Internetworking Operating System



IOS



- Sistemul de operare proprietar Cisco
 - folosit pe majoritatea ruterelor și switch-urilor Cisco
- Disponibil în multiple versiuni
 - fiecare versiune este orientată pe un anumit set de facilități
 - versiunile cu mai multe facilități vin cu o imagine de mai mari dimensiuni, la un preț mai mare
- Modul principal de interacțiune
 - prin intermediul unei interfețe text de tip terminal
 - se poate folosi și o interfață Web de configurare

IOS (Internetwork Operating System) este sistemul de operare folosit de majoritatea echipamentelor de rețea CISCO. Ca orice alt sistem de operare, IOS gestionează resursele hardware și software ale unui ruter, incluzând alocarea memoriei, managementul diferitelor procese și sistemul de fișiere, printre altele. Principala interfață de configurare este reprezentată de o linie de comanda (CLI), existând și posibilitatea utilizării unei interfețe grafice.

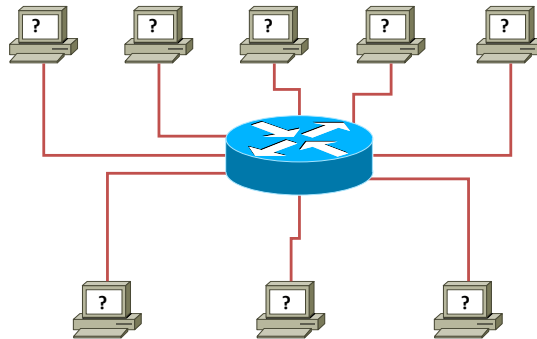
Până în anul 2011 (versiunea 12.4), versiunile IOS erau unice pentru fiecare platformă, adică fiecare model de ruter, switch sau bridge avea nevoie de propria imagine a sistemului de operare; în plus, existau mai multe variații a unei anumite imagini, reprezentând diferitele opțiuni oferite de licența cumpărată. Nou apăruta versiune 15 a IOS utilizează un sistem inovator de licențe pentru activarea diferitelor opțiuni dorite; astfel, o companie poate descărca o singură imagine universală, și să îi activeze ulterior diversele opțiuni prin simpla instalare a unor licențe.

Procesul de rutare



De ce avem nevoie de rutare?

- Transmiterea pachetelor între rețele
 - criterii de eficiență
 - criterii de politică internă sau externă



Principalul rol al unui ruter într-o topologie este rutarea pachetelor între mai multe rețele. Rutarea este procesul prin care dispozitivul de rețea analizează antetul de nivel 3 al unui pachet primit (adresa IP destinație) și apoi, pe baza anumitor reguli clar definite manual sau de protocoale specializate, decide ce să facă cu pachetul mai departe. Un ruter de nivel „enterprise” trebuie să fie capabil să proceseze în jur de 10.000 pachete pe secundă, iar acest lucru este posibil doar prin implementarea unor circuite hardware dedicate foarte rapide.

Deși din punct de vedere teoretic, rutarea pachetelor se desfășoară doar la nivelul 3, un ruter este capabil să citească informații din antete până la nivelul 7. Acest lucru este în special folositor în aplicații „mission-critical” cum ar fi VoIP sau aplicații care necesită latență extrem de mică, cu ajutorul unor sisteme de QoS (Quality of Service). QOS este un serviciu oferit de anumite rutere avansate care permite prioritizarea traficului în funcție de conținut sau destinație.

Criterii de rutare

- Informațiile cu privire la rețelele cunoscute sunt stocate în tabela de rutare
 - se stochează adresele rețelelor și următorul hop către fiecare destinație
 - în cazul conexiunilor punct-la-punct se poate stoca direct interfața de ieșire
 - același lucru se face automat pentru rețelele direct conectate
- Rute statice
 - configurate de administrator
 - au prioritate în procesul de rutare
- Rute dinamice
 - învățate prin intermediul unor protocoale specializate
 - algoritmi folosesc criterii de eficiență sau criterii de politică

Fiecare ruter are o bază de date salvată în RAM care conține regulile setate manual sau automat folosite pentru luarea deciziilor de rutare; această bază de date se numește tabelă de rutare.

Tabela de rutare a unui ruter stochează informații multiple cu privire la rețelele adiacente unui ruter și la calea pe care un pachet trebuie să o urmeze pentru a ajunge în rețeaua destinație. Astfel, pentru o destinație oarecare tabela de rutare stochează:

- masca de rețea
- metoda prin care calea respectivă a fost aflată
- adresa IP next-hop sau interfața de ieșire prin care aceasta poate fi accesată

Rutele pot fi învățate de un ruter prin două metode:

- Static, configurate de un administrator; acest tip de rută va fi întotdeauna preferată față de o rută dinamică

- Dinamic, cu ajutorul unui protocol de rutare specializat; în acest caz se folosesc algoritmi avansați pentru determinarea căii optime

Rutare statică vs. Rutare dinamică (1)

▪ Rutare statică

- oferă control mult mai riguros administratorului asupra următorului hop ales
- este foarte ușor de învățat
- nu este deloc scalabilă

Rutarea statică se configurează manual pe rutere și oferă un management mai riguros asupra modului de stabilire a următorului hop către o anumită destinație. Un avantaj al folosirii rutării statice este efortul necesar scăzut pentru configurarea și administrarea rețelelor restrânse. Pe de altă parte, rutarea statică nu scalează optim în cazul rețelelor de dimensiuni mari, fiind necesară implementarea rutării dinamice. De asemenea, rutarea statică consumă puține resurse hardware, permițând rularea în paralel a altor aplicații performante dacă este necesar.

Rutare statică vs. Rutare dinamică (2)

▪ Rutare dinamică

- necesită cunoștințe avansate pentru o configurare eficientă
- utilizează atât un procent din bandwidth cât și o parte din procesor
- calea aleasă de pachete nu e cunoscută în mod clar
- este o soluție scalabilă și tolerantă la defecte
- exemple: RIP, IGRP, EIGRP, OSPF, IS-IS, BGP

În cazul protocoalelor de rutare dinamice, informația despre rute este propagată automat în întreaga rețea, rutele fiind distribuite cu ajutorul unui algoritm specific. Protocoalele de rutare dinamice oferă scalabilitate și flexibilitate mărită față de metoda statică, însă consumă mai multe cicluri de procesor și utilizează o cantitate mai mare de memorie RAM. Protocoalele de rutare dinamice sunt responsabile cu păstrarea tabelii de rutare sincronizată peste toate ruterele din domeniul de rutare în cazul unei modificări în topologie. Astfel, informațiile că o rețea este invalidă sau afectată de anumite probleme hardware se propagă foarte rapid în toată rețeaua, fără intervenția administratorului. Exemple de protocoale de rutare dinamice sunt RIPv1, RIPv2, IGRP, EIGRP, OSPF, IS-IS, RIPv6, EIGRP IPv6 și BGP. RIPv1 și IGRP nu mai sunt utilizate, ele fiind înlocuite integral de versiunile lor îmbunătățite – RIPv2 și EIGRP. RIPv6 și EIGRP IPv6 sunt variantele bazate pe IPv6 ale protocoalelor de rutare asociate prin denumire.

Metrici, determinarea căii optime

- Metrică
 - indicator de preferință a unei rute după anumite criterii
 - se calculează în funcție de hop-count, delay, bandwidth etc.
 - o metrică mai mică este mai bună
- Determinarea căii optime
 - fiecare rută din tabel are atribuită o metrică
 - ruterul alege ruta cu metrica cea mai mică

Determinarea celei mai bune căi către destinație implică evaluarea căilor disponibile în funcție de anumite criterii, dintre care se remarcă metrica. Metrica este o valoare calculată în funcție de anumite variabile asociate drumului dintre două puncte într-o rețea:

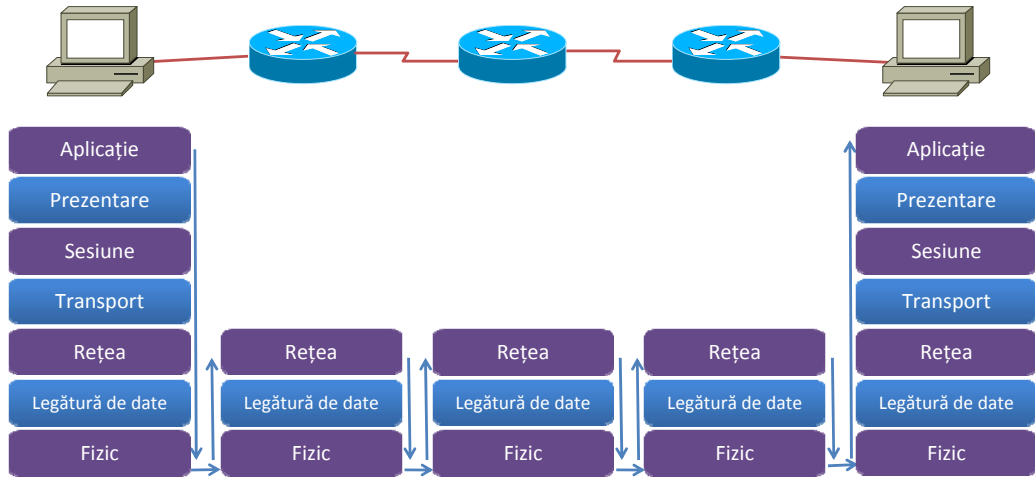
- Hop-count: numărul de rutere parcurse de pachet până la destinație
- Delay: latența specifică elementelor de legătură care alcătuiesc calea
- Bandwith: viteza legăturilor dintr-o cale
- Load: încărcarea cu date a unei conexiuni
- Reliability: gradul de siguranță oferit de o anumită legătură

Protocoalele de rutare dinamice pot utiliza una sau mai multe din aceste variabile în calculul metricii unui drum sau a unei destinații. Ulterior acestui calcul, protocolul de rutare va selecta ruta cu metrica cea mai mică și o va introduce în tabela proprie de rutare.

Manipularea pachetelor



Rutarea în cadrul stivei OSI



În cazul transmiterii de date între două stații, pachetele aflate în tranzit vor fi prelucrate de echipamentele terminale și intermediare folosind protocoalele definite în cadrul fiecărui nivel al stivei OSI. Astfel, un pachet va trece prin mai multe procese de încapsulare și decapsulare. La sursă, PDU-ul (Packet Data Unit) va fi încapsulat, la fiecare nivel adăugându-se noi informații specifice.

În cazul rutării, pachetul va fi decapsulat în fiecare ruter prin care trece până la nivelul 3, deoarece un ruter are nevoie doar de adresa IP destinație pentru a lua decizia de trimitere mai departe. Adresele IP sursă și destinație nu se vor schimba niciodată de-a lungul traseului. La nivelul Legătură de date, fiecare hop va modifica adresa MAC sursă, respectiv adresa MAC destinație. Antetul de nivel 2 se va modifica doar la trecerea într-o altă rețea, și nu la trecerea printr-un switch sau alt echipament de nivel 2. Când ajunge la destinație, pachetul este decapsulat și informația conținută este prezentată utilizatorului.

Manipularea pachetelor (1)

- Adrese MAC
 - adrese de nivel 2
 - folosite pentru identificarea fizică a dispozitivelor în cadrul unei rețele locale
 - se modifică la trecerea dintr-o rețea în alta

IEEE 802.3						
7	1	6	6	2	46 to 1500	4
Preamble	Start of frame delimiter	Destination Address	Source Address	Length Type	802.2 Header and Data	Frame check sequence

Pentru o comunicație eficientă în interiorul rețelei locale nu este nevoie de o adresă la nivel 3, fiind suficient antetul unui pachet la nivelul 2. Acesta conține adresa MAC sursă cât și cea destinație, acestea fiind suficiente pentru transmiterea pachetului cu succes la destinație. Deoarece adresele fizice au numai relevanță locală, ele se vor modifica când pachetul părăsește rețeaua în care se află la un moment dat.

Cadrul 802.3 (Ethernet) este alcătuit din următoarele câmpuri:

- Preamble: 7 biți de 1 și 0 care alternează și au rol de sincronizare
- Start of frame Delimiter: un octet care semnalizează începutul cadrului
- Destination Address: adresa MAC a destinatarului
- Source Address: adresa MAC a expeditorului

Manipularea pachetelor (2)

- Adrese MAC
 - adrese de nivel 2
 - folosite pentru identificarea fizică a dispozitivelor în cadrul unei rețele locale
 - se modifică la trecerea dintr-o rețea în alta

IEEE 802.3						
7	1	6	6	2	46 to 1500	4
Preamble	Start of frame delimiter	Destination Address	Source Address	Length Type	802.2 Header and Data	Frame check sequence

- Length/Type: doi octeți care pot reprezenta ori lungimea cadrului (dacă valoarea este mai mică decât 0x600 – echivalentul 1536 în zecimal) sau tipul protocolului de nivel superior (dacă valoarea este mai mare de 0x600)
- Data: datele încapsulate în pachet
- Frame Check Sequence: 4 octeți cu rolul de a verifica integritatea datelor recepționate.

Manipularea pachetelor (3)

- Adrese IPv4
 - adrese de nivel 3
 - folosite pentru identificarea rețelelor și a stațiilor din rețea
 - se păstrează neschimbate în timpul rutării între rețele
 - time-to-live poate fi folosit pentru a opri buclele de rutare

0		16		31	
Version	Header Length	Service type	Total length		
Identification			Flags	Fragment offset	
Time to live		Protocol	Header checksum		
Source address					
Destination address					
IPv4 options (if any)					Padding
Data					

Antetul de nivel 3 este utilizat de ruter pentru a determina calea pe care trebuie să trimită pachetul în drumul spre destinație. Adresele de nivel 3 IP sursă și destinație nu se modifică niciodată în tranzit, fiind afectate alte câmpuri din antet, cum ar fi TTL (time to live) și Header Checksum. Câmpul TTL al fiecărui pachet IP se decrementează cu o unitate la fiecare trecere printr-un ruter. Decrementarea valorii TTL poate fi considerată o măsură de siguranță la nivel 3 permițând eliminarea rapidă a buclelor de rutare (în general considerând că un pachet este trimis cu o valoare TTL egală cu 16, aceasta nu este prins într-o buclă de rutare deoarece pachetul va fi aruncat după parcurgerea a 16 hopuri).

Tabela de rutare și principiile rutării



Tabela de rutare

- Este folosită de ruter pentru a alege interfața de ieșire în transmiterea unui pachet
- Este stocată în RAM, deci se pierde la fiecare repornire
- Conține informații de tip rețea – interfață de ieșire (sau rețea – rețea intermediară)
 - rețele direct conectate, adăugate implicit
 - rețele la distanță: rute statice sau dinamice

Tabela de rutare a unui ruter reprezintă o structură de date ierarhică, unificată și organizată care stochează informații despre destinațiile cunoscute. Este stocată în RAM și nu se memorează la salvarea configurației unui ruter – ea se va reconstrui la fiecare repornire. Pe baza informațiilor conținute în tabela de rutare ruterele iau decizii cu privire la transmiterea unui pachet pe o anumită interfață de ieșire.

Tabela de rutare poate conține mai multe tipuri de rețele:

- Rețele direct conectate: sunt introduse automat în tabela de rutare, reprezentând rețelele care aparțin interfețelor active ale ruter-ului; ele nu pot fi șterse sau modificate fără o schimbare a adresării IP sau a dezactivării interfeței
- Rețele remote: configurate cu ajutorul rutelor statice sau a protocoalelor dinamice de rutare

Tabela de rutare

▪ Exemplu de tabelă de rutare

```
Codes: I - IGRP derived, R - RIP derived, O - OSPF derived,  
C - connected, S - static, E - EGP derived, B - BGP derived,  
* - candidate default route, IA - OSPF inter area route,  
i - IS-IS derived, ia - IS-IS, U - per-user static route,  
o - on-demand routing, M - mobile, P - periodic downloaded static route,  
D - EIGRP, EX - EIGRP external, E1 - OSPF external type 1 route,  
E2 - OSPF external type 2 route, N1 - OSPF NSSA external type 1 route,  
N2 - OSPF NSSA external type 2 route  change change change
```

```
Gateway of last resort is 10.119.254.240 to network 10.140.0.0
```

```
O 172.150.0.0 [160/5] via 10.119.254.6, 0:01:00, Ethernet2  
E 172.17.10.0 [200/128] via 10.119.254.244, 0:02:22, Ethernet2
```

În momentul în care un ruter primește un pachet IP pe una din interfețe, adresa destinație din antetul pachetului va fi căutată în tabela de rutare pornind de la ruta cea mai specifică din tabelă (masca de rețea cea mai lungă) și până la ruta cea mai puțin specifică. Tabela de rutare prezintă informații despre următorul hop unde trebuie trimis un pachet pentru ca acesta să ajungă la destinația dorită. De asemenea este prezent și un timer, care în cazul protocoalelor de rutare dinamice reprezintă timpul rămas până la declararea unei anumite rute invalide din cauza lipsei de activitate. Fiecare tip de rută din tabela de rutare este reprezentată de un simbol: O – OSPF, R – RIP, D – EIGRP, B – BGP, făcând mai ușoară parcurgerea acestora și efectuarea de „troubleshooting” în caz de nevoie.

Principiile de rutare

- Fiecare ruter ia decizii bazându-se doar pe propria tabelă de rutare
- Nu toate ruterele au aceeași tabelă de rutare
- Rutarea se face asimetric
 - rutele stocate se referă doar la drumul spre o rețea, nu și invers
 - pachetele pot folosi alte căi la comunicarea în sens invers

Pentru asigurarea unei funcționări optime a procesului de rutare sunt respectate următoarele 3 principii:

- Ruterele iau decizii de rutare independent bazându-se numai pe informațiile din propria lor tabelă de rutare; astfel, problemele de rutare sunt împiedicate de a se propaga în întreaga topologie, iar puterea de procesare pentru găsirea unei destinații este împărțită în mod egal tuturor nodurilor
 - Tabela de rutare este unică pentru fiecare ruter deoarece aceasta conține următorul hop pentru fiecare destinație în parte; tabela de rutare a unui ruter nu va descrie niciodată întreaga cale pe care un pachet trebuie să o urmeze pentru a ajunge la destinația cerută
 - Rutarea este asimetrică deoarece tabela de rutare nu descrie un „next hop” valabil pentru un drum dus-întors
-
-
-
-
-

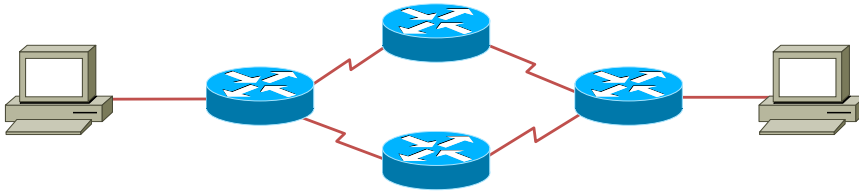
Clasificarea rețelelor la rutare

- Conectate
 - rețele direct conectate la interfețe ale ruterului
 - rutele sunt adăugate automat după pornirea și configurarea interfeței
- Cunoscute
 - acele rețele către care sunt definite rute statice sau dinamice
- Necunoscute
 - nu există rute definite pentru aceste rețele
 - se folosește ruta default, dacă e definită, sau se aruncă pachetul
- Rută implicită
 - se definește static de către administrator sau este propagată dinamic
 - se aplică pentru toate rutele necunoscute

Tabela de rutare a unui ruter poate fi populată de mai multe tipuri de rețele:

- Conectate – rețelele care aparțin interfețelor active ale ruterului, fiind introduse automat în tabela de rutare alături de interfețele de ieșire corespunzătoare
- Cunoscute – rețelele care au fost instalate în tabela de rutare prin rute statice sau prin protocoale de rutare dinamice
- Necunoscute – rețelele pentru care nu a fost găsit nici un „next hop” sau o interfață de ieșire în urma procesului de parcurgere a tabelii de rutare; în cazul definirii unei rute implicite, ruterul va folosi această rută pentru trimiterea pachetelor destinate respectivelor rețele, altfel, vor fi aruncate
- Rută implicită – este ruta spre care se trimit toate pachetele pentru care nu se cunoaște o destinație specifică

Load balancing



- Pot exista mai multe rute cu aceeași metrică și către aceeași rețea
- În acest caz pachetele pot fi repartizate în mod egal între rutele respective
 - se obține o mai bună repartizare a traficului în rețea
- Procesul se numește „Load balancing”

Există situații în care sunt introduse în tabela de rutare mai multe rute către aceeași destinație având aceeași valoare a metricii. În acest caz, ruterul va repartiza pachetele trimise către destinație în mod egal între rutele respective. Astfel, tabela de rutare va conține pentru o anumită rețea destinație mai multe interfețe de ieșire (sau adrese IP next-hop).

Utilizarea corectă a procesului de „load balancing” poate îmbunătăți eficiența și performanța rețelei. În cazul în care traficul este împărțit în mod egal între rutele către destinație, ruterul realizează procesul de „equal cost load balancing”, dar există situații în care pachetele pot fi trimise pe căi multiple chiar dacă metrica nu are aceeași valoare. Acest proces este cunoscut sub numele de „unequal cost load balancing” și poate fi realizat în cadrul protocolului de rutare EIGRP.

Configurarea ruterului prin CLI



Configurare prin CLI

- Există 2 moduri de lucru în CLI
- Modul User Exec
 - este evidențiat de prompt-ul „>”
 - modul implicit în care avem acces imediat după autentificarea pe ruter
 - are doar drepturi de interogare a unor informații, nu și de configurare
- Modul Privileged Exec
 - este evidențiat de prompt-ul „#”
 - are drepturi de configurare a setărilor echipamentului
 - trecerea din modul User în modul Privileged se face cu comanda **enable**

Interfața în linie de comandă presupune existența a două moduri principale în care se pot introduce comenzi:

- Modul user Exec: oferă drepturi limitate utilizatorului; identificat prin prompt-ul „>”, se pot accesa doar comenzi elementare de afișare a informațiilor cu privire la funcționarea ruterului
- Modul privileged Exec: se activează prin comanda **enable** în modul user Exec și oferă acces la funcții de configurare mai avansate și la informații mai detaliate cu privire la procesele ruterului; accesarea modului privileged Exec se poate observa prin schimbarea prompt-ului de la caracterul „>” în caracterul „#”

Comanda show

- Informații generale despre ruter
 - #**show version**
- Informații generale despre interfețele ruterului
 - #**show interfaces**
- Informații despre adresarea interfețelor
 - #**show ip interface [brief] [tip_interfață număr_interfață]**
- Afișarea tabelii de rutare
 - #**show ip route**
- Afișarea configurației curente
 - #**show running-config**

Una dintre cele mai utilizate comenzi pentru vizualizarea și inspectarea diferitelor configurații existente pe un ruter este comanda **show**. Această comandă suportă numeroși parametrii cu rolul de a afișa:

- informații generale despre ruter (modelul ruterului, versiunea de IOS, capacitatea memoriei, ș.a.)
 - informații generale despre funcționalitatea și numărul interfețelor existente pe un ruter
 - ip-urile configurate pe interfețe
 - conținutul tabelii de rutare
 - configurația curentă salvată în fișierul running-config (în memoria RAM) sau configurația permanentă salvată în fișierul startup-config (în memoria NVRAM)
-
-
-
-
-

Configurări de bază (1)

- Intrarea în modul de configurare
 - `#configure terminal`
 - prompt-ul se schimbă din `#` în `(config)#`
- Setarea numelui echipamentului
 - `(config)#hostname <nume_router>`
- Setarea unei parole pe modul privilegiat
 - `(config)#enable secret <parola>`
- Setarea unui banner pe router (mesaj afișat la accesare)
 - `(config)#banner motd # Accesul strict interzis ! #`

Majoritatea comenzilor pentru realizarea configurărilor de bază ale unui ruter sunt introduse din modul global de configurare (global configuration mode). Modul global de configurare este accesat prin comanda **configure terminal** introdusă în privileged Exec mode și se identifică prin afișarea promptului **(config)#**. Acest mod de configurare reprezintă interfața principală pentru implementarea a numeroase funcționalități ale unui ruter. Astfel se remarcă o serie de configurări de bază:

- setarea numelui echipamentului
 - setarea unei parole criptate sau „plain text” pentru modul privilegiat
 - setarea unui banner, adică un mesaj care este afișat de fiecare dată când ruterul este accesat în linia de comandă
-
-
-
-
-

Configurări de bază (2)

- Setarea unei parole pentru accesul la consolă
 - (config) **#line console 0**
 - (config-line) **#password <parola>**
 - (config-line) **#login**

- Configurarea ruter-ului pentru acces de la distanță (telnet)
 - (config) **#line vty 0 4**
 - (config-line) **#password <parola>**
 - (config-line) **#login**

Accesarea unui ruter poate fi realizată prin conectare directă la consola ruterului sau la distanță folosind protocoalele telnet sau ssh.

Pentru a oferi un anumit grad de securitate al ruterului, există posibilitatea configurării unei parole de acces. Astfel, pentru accesul la consolă se setează o parolă din modul **config-line**. În acest caz, la fiecare conectare la linia de comandă a ruterului, se va solicita utilizatorului introducerea parolei setate.

În cazul accesului de la distanță, se pot configura liniile vty (Virtual Terminal lines) cu o parolă comună pentru toți utilizatorii, o parolă diferită pentru fiecare utilizator definit local, sau se poate folosi un server de autentificare.

Configurarea unei interfețe (1)

- Intrarea în modul de configurare al interfeței
 - (config) # **interface** <tip_interfață> <număr_interfață>
 - prompt-ul se schimbă din (config) # în (config-if) #
- Pornirea/oprirea unei interfețe
 - (config-if) # **[no] shutdown**
 - implicit toate interfețele sunt oprite pe rutere

Configurarea caracteristicilor unei interfețe se realizează din modul de configurare al unei interfețe semnalat de schimbarea promptului din (config) # în (config-if) # la introducerea comenzii **interface** urmată de numele și numărul interfeței.

Dezactivarea unei interfețe se efectuează utilizând comanda **shutdown**. Pentru pornirea acesteia, se anulează comanda **shutdown** prin negare: **no shutdown**.

În general, o interfață este denumită respectând următorul format: tip interfață urmat de 3 numere separate prin „/”, spre exemplu FastEthernet 0/1/1 unde prima valoare identifică modulul, a doua slotul, iar ultima, portul. Prima valoare este menționată doar în cazul utilizării ruterele modulare.

Configurarea unei interfețe (2)

- Setarea unei adrese IP pe interfață
 - (config-if)#**ip address** <adresă> <mască>
- Setarea unei descrieri pe interfață
 - (config-if)#**description** <descriere>
- Configurarea parametrului `clock-rate` pe interfețele seriale
 - interfețele seriale necesită, pentru a putea funcționa, configurarea vitezei de comunicație (clock rate)
 - (config-if)#**clock rate** <valoare>

Ulterior accesării modului de configurare a unei interfețe se poate seta o adresă IP prin specificarea adresei respective urmată de masca de rețea din cadrul rețelei din care face parte adresa IP configurată.

O informație utilă pentru facilitarea documentării sau depanării unei rețele este configurarea unei descrieri pe interfețe. Aceasta este setată prin utilizarea comenzii **description** urmată de un text limitat la 240 de caractere.

În cazul utilizării unei legături seriale de tip punct-la-punct, un capăt al legăturii este de tip DTE (data terminal equipment), iar celălalt, de tip DCE (data communications equipment). Pentru a se realiza comunicația de date pe o astfel de legătură, ruterul conectat la capătul DCE trebuie să asigure sincronizarea transmisiei semnalului prin folosirea unei valori numerice configurate prin comanda **clock rate** urmată de o valoare exprimată în biți pe secundă.

Salvarea sau ștergerea configurațiilor

- Reunește configurațiile din memorie cu cele salvate în fișierul de configurare (configurațiile din memorie nu se pierd!)
 - `#copy startup-config running-config`
- Salvarea configurației curente pentru a fi încărcată la repornire
 - `#copy running-config startup-config`
- Ștergerea configurației salvate (startup-config)
 - `#erase startup-config`

Toate configurațiile realizate pe un ruter sunt salvate în RAM, în fișierul de configurare `running-config`. Pentru ca setările să devină permanente, acestea trebuie salvate într-o memorie nevolatilă, și anume în NVRAM. Astfel, în cazul restartării accidentale a unui ruter, configurațiile curente vor fi încărcate la boot-are. Salvarea setărilor se face prin copierea fișierului `running-config` în fișierul `startup-config`. O comandă mai scurtă și mai rapidă decât comanda `copy`, cu cei doi parametri fișier sursă și fișier destinație, este comanda `write` fără parametri, având același efect.

Ștergerea configurațiilor anterioare existente pe un ruter se realizează prin comanda `erase` urmată de numele fișierului de șters, în cazul de față, `startup-config`.

Rezumat

- Caracteristicile unui ruter
- IOS
- Procesul de rutare
- Rutare statică
- Rutare dinamică
- Tabela de rutare

1. Care sunt cele două funcții principale ale unui ruter?
2. Care rolul memoriei RAM?
3. În ce caz sunt preferate rutele definite static în locul rutelor obținute dinamic?
4. În ce caz se modifică adresele sursă și destinație de nivel 3 din câmpurile unui pachet în cadrul rutării într-o topologie dată?
5. Care sunt motivele pentru care se utilizează câmpul TTL în antetul de nivel 3?