

7. Nivelul rețea

7.1. Cuprins modul

7.	Nivelul rețea	1
7.1.	Cuprins modul	1
7.2.	Comutarea de pachete de tip Memorează-și-Retransmite	2
7.3.	Retransmiterea și rutarea	3
7.4.	Servicii furnizate nivelului transport.....	4
7.5.	Subrețele cu circuite virtuale și subrețele datagramă	6
7.5.1.	Implementarea subrețelor cu circuite virtuale	7
7.5.2.	Implementarea subrețelor datagramă	9
7.5.3.	Comparație între subrețele cu circuite virtuale și subrețele datagramă	11



Introducere

Am arătat în unitățile anterioare că nivelul legătură de date face posibilă comunicația între două noduri conectate în mod direct printr-o legătură fizică. Nivelul transport pune la dispoziție diverse forme de comunicație proces-la-proces, utilizând serviciile nivelului rețea. Nivelul rețea este unul din cele mai complexe niveluri din stiva de protocoale punând la dispoziție servicii de comunicație gazdă-la-gazdă.



Obiective

După parcurgerea acestei unități de învățare studenții vor fi capabili:

- ✓ Să deosebească retransmisia de rutare
- ✓ Să descrie funcționarea subrețelor cu circuite virtuale
- ✓ Să descrie funcționarea subrețelor datagramă
- ✓ Să compare subrețelele datagramă cu subrețelele cu circuite virtuale pornind de la criterii date



Durată medie de studiu

Durata medie de studiu individual : 2 ore

individual

Nivelul rețea are ca sarcină preluarea pachetelor de la sursă și transferul lor către destinație. Ajungerea la destinație poate necesita mai multe salturi prin rutere intermediare de-a lungul drumului. Nivelul rețea este cel mai scăzut nivel care se ocupă de transmisii capăt la capăt. Pentru realizarea scopurilor propuse, nivelul rețea trebuie să cunoască topologia subrețelei de comunicație (de exemplu mulțimea tuturor ruterelor) și să aleagă calea cea mai potrivită prin aceasta.

De asemenea nivelul rețea trebuie să aleagă căile de urmat astfel, încât să nu încarce excesiv unele legături de comunicație sau rutere în timp ce altele sunt inactive. În fine, când sursa și destinația fac parte din rețele diferite, apar provocări noi! E sarcina nivelului rețea să se ocupe de ele. În această unitate de învățare vom studia toate aceste aspecte și le vom exemplifica, în primul rând folosind Internetul și protocolul său de nivel rețea, IP.

Nivelul rețea este prezent pe fiecare gazdă și ruter din rețea, pe când nivelul următor, transport, este implementat doar pe sistemele capăt (gazde). Din acest motiv, protocoalele de nivel rețea sunt printre cele mai interesante din stivă.

7.2. Comutarea de pachete de tip Memorează-și-Retransmite

Înainte de a începe explicarea detaliilor nivelului rețea, vom reaminti contextul în care operează protocoalele de la nivelul rețea. Acest context este prezentat în Figura 10-1. Componentele majore ale sistemului sunt echipamentele furnizorului de servicii (rutere conectate prin linii de transmisie), plasate în interiorul zonei umbrite, și echipamentele clientului, plasate în afara zonei umbrite. Gazda H1 este conectată direct la unul dintre ruterele furnizorului de servicii, A printr-o linie închiriată. În contrast, H2 face parte dintr-o rețea LAN cu un ruter, F, deținut și operat de către client. Acest ruter are și o linie închiriată către echipamentul furnizorului de servicii. Am prezentat F ca fiind în afara zonei umbrite, deoarece nu aparține furnizorului de servicii, dar în termeni de construcție, software și protocoale, probabil că nu diferă față de ruterele acestuia. Este discutabil dacă aparține subrețelei. În acest capitol ruterele din localul clientului sunt considerate parte a subrețelei deoarece rulează aceiași algoritmi ca și ruterele furnizorului de servicii (și aici principala noastră preocupare sunt algoritmii).

Ruterele sunt folosite după cum urmează. O gazdă care are de transmis un pachet îl transmite celui mai apropiat ruter, fie în aceeași rețea LAN, fie printr-o legătură punct la punct cu furnizorului de servicii. Pachetul este memorat la ruter până ajunge integral, astfel încât să poată fi verificată suma de control. Apoi este trimis mai departe către următorul ruter de pe traseu, până

ajunge la gazda destinație, unde este livrat. Acest mecanism reprezintă comutarea de pachete de tip memorează-și-retransmite.

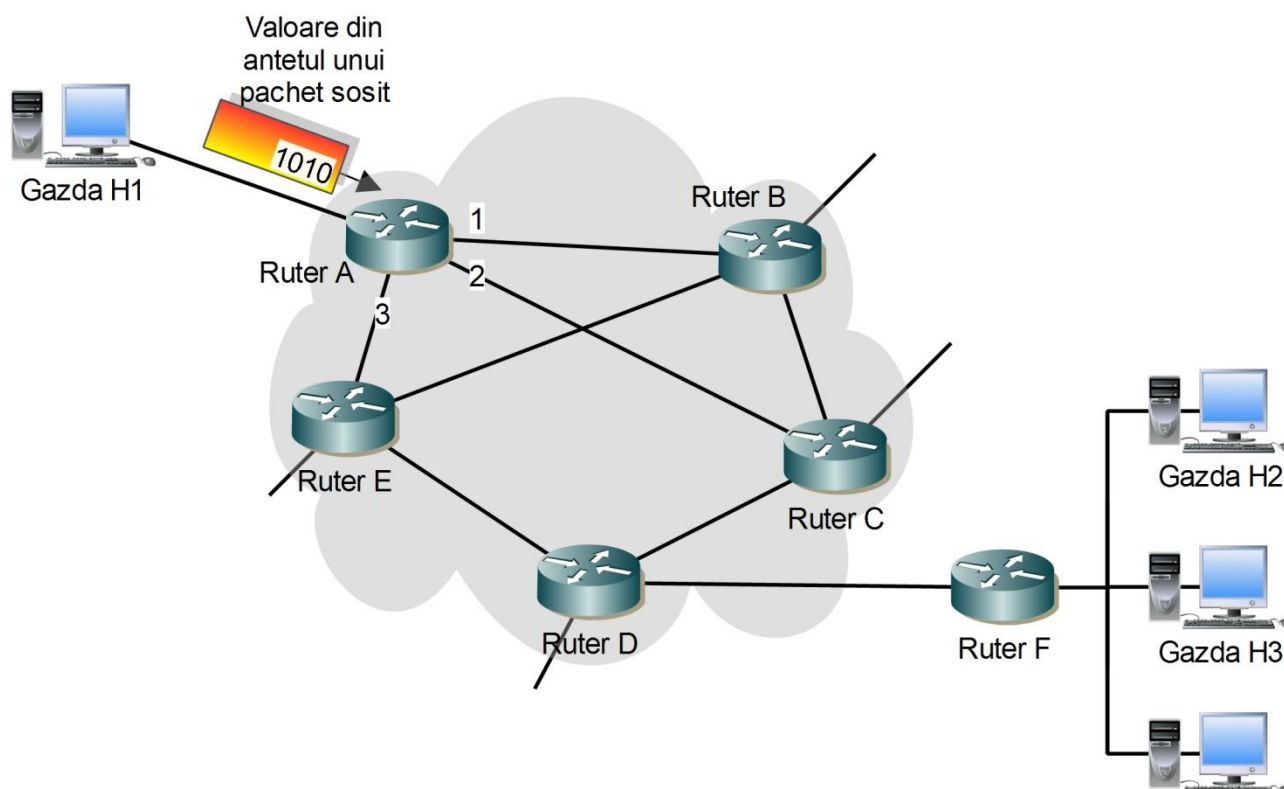


Figura 7.1 Comutarea de pachete

7.3. Retrimiterrea și rutarea

Fiecare ruter deține o *tabelă de rutare*. Un exemplu de tabelă de rutare este prezentat în Figura 7.2. Un ruter retrimite un pachet examinând valoare unui câmp din antetul pachetului sosit, câmp ce este folosit ca index în tabela de rutare. Rezultatul din tabela de rutare indică interfața de ieșire a ruterului pe care trebuie retransmis pachetul. În funcție de protocolul de nivel rețea, valoarea din antetul pachetului, poate reprezenta adresa destinație a pachetului sau un indicator al conexiunii căreia îi aparține pachetul. În Figura 7.1 un pachet având valoare unui câmp din antet 1010 sosește la ruterul A. Ruterul indexează tabela de rutare și determină legătură de ieșire pentru acest pachet: 2. Apoi ruterul retrimite intern pachetul spre interfața 2.

Tabelă de rutare locală	
Valoare antet	Legătură de ieșire
0011	3
1010	2
1100	2
1001	1

Figura 7.2 Posibilă tabelă de rutare pentru ruterul A

În aceste condiții, rolul nivelului rețea este simplu: trebuie să transfere pachete de la gazda emițătoare la gazda receptoare, context în care pot fi identificate două procese distincte ce rulează intern pe un ruter retrimiteră și rutarea:

Retrimiteră: preia fiecare pachet care sosește, caută în tabela de rutare linie de ieșire corespunzătoare și retrimite pachetul pe acea linie. Spre exemplu un pachet care sosește la Ruterul A având ca destinație gazda H2 trebuie trimis către următorul ruter din calea RA-H2.

Rutarea: se ocupă cu completarea și actualizarea tabelului de rutare, astfel încât să poată fi determinată calea sau ruta pe pachetele trebuie s-o urmeze de la sursă la destinație. Aici intervine algoritmul de rutare. Spre exemplu, un algoritm de rutare ar determina calea pe care trebuie s-o parcurgă pachetele de la H1 la H2.

7.4. Servicii furnizate nivelului transport

Nivelul rețea furnizează servicii nivelului transport la interfața dintre cele două niveluri. O întrebare importantă este ce fel de servicii furnizează nivelul rețea nivelului transport. Serviciile nivelului rețea au fost proiectate având în vedere următoarele scopuri:

1. Serviciile trebuie să fie independente de tehnologia ruterului.
2. Nivelul transport trebuie să fie independent de numărul, tipul și topologia ruterelor existente.
3. Adresele de rețea disponibile la nivelul transport trebuie să folosească o schemă de numerotare uniformă, chiar în cadrul rețelelor LAN și WAN.

Obiectivele fiind stabilite, proiectantul nivelului rețea are o mare libertate în a scrie specificațiile detaliate ale serviciilor oferite nivelului transport. Înainte de a trece la o analiză detaliată, este util să formulăm o serie de întrebări care privesc diferite tipuri de servicii ce ar putea fi oferite de nivelul rețea.

- Atunci când nivelul transport furnizează un segment nivelului rețea, poate acesta conta pe faptul că nivelul rețea va livra datagrama respectivă la destinație?
- Atunci când se trimit datagrame multiple, vor fi acestea livrate nivelului transport la gazda receptoare în ordinea în care au fost transmise?
- Timpul dintre recepția a două datagrame succesive va fi același cu timpul dintre transmiterea lor?
- Va furniza rețeaua vreo informație cu privire o eventuală congestie?

- Care sunt proprietățile abstracte ale canalului logic ce conectează nivelurile transport de la gazda emițătoare și cea receptoare?

Răspunsurile la aceste întrebări (precum și la altele neenumerate mai sus) sunt determinate de modelul de servicii asigurat de nivelul rețea. *Modelul serviciilor de rețea* definește caracteristicile transportului de pachete capăt-la-capăt între sistemul capăt emițător și sistemul capăt receptor.

Să avem în vedere câteva posibile servicii pe care nivelul rețea le poate oferi:

Livrare garantată: acest serviciu garantează că pachetul va fi eventual livrat la destinație.

Livrare garantată cu întârziere limitată: serviciul nu doar că garantează livrarea pachetului, ci asigură livrarea datagramelor într-un timp limitat (spre exemplu 50ms).

Mai mult, pentru un flux de pachete între o sursă și o destinație pot fi asigurate următoarele servicii:

Livrarea ordonată a pachetelor: acest serviciu garantează că pachetele sosesc la destinație în ordinea în care au fost transmise.

Lărgime de bandă minimă garantată: acest serviciu de nivel rețea emulează comportamentul unei legături sursă-destinație cu o rată de bit specificată (spre exemplu 10Mbps); practic legătura traversează mai multe legături fizice. Atâta timp cât gazda emițătoare transmite biți (ca parte din pachete) la o rată sub cea specificată, nu se pierd pachete și fiecare pachet ajunge la destinație într-un timp prestabilit (spre exemplu în 100ms).

Jitter maxim garantat: acest serviciu garantează că timpul dintre transmisia a două pachete succesive este egal cu timpul dintre recepția a două pachete succesive. (altfel spus fluctuația se încadrează în limite specificate).

Servicii de securitate: prin folosirea unei chei secrete de sesiune, cunoscută doar de gazda sursă și gazda destinație, nivelul rețea de la sursă ar putea cripta încărcătura utilă a tuturor datagramelor trimise către acea destinație. Nivelul rețea de la gazda destinație ar fi responsabil de decriptarea încărcăturii utile. Cu un astfel de serviciu, s-ar putea asigura confidențialitatea tuturor segmentelor (TCP sau UDP) de nivel transport dintre gazdele sursă și destinație. În plus față de confidențialitate, nivelul rețea ar putea asigura servicii de integritate a datelor și autentificarea sursei.

Aceasta este doar o listă parțială de servicii pe care nivelul rețea le-ar putea pune la dispoziție, existând bineînțeles și altele.

Nivelul rețea din Internet asigură un singur serviciu, cunoscut sub denumirea de serviciu *best-effort* (rom. cel mai bun efort). Din Tabelul 7.1 s-ar părea că serviciul *best-effort* este un eufenism¹ pentru nici un fel de serviciu. În cazul serviciului *best-effort*, nu se garantează păstrarea temporizării dintre pachete, nu se garantează recepționarea pachetelor în ordinea în care au fost trimise și nici măcar nu se garantează livrarea pachetelor. Dată fiind această definiție, o rețea care nu livrează nici un pachet la destinație ar satisface definiția unui serviciu de tip *best-effort*. Așa cum vom vedea puțin mai târziu există motive întemeiate pentru un astfel de model minimalist de servicii de rețea.

Tabelul 7.2 Modele de servicii: Internet, ATM CBR și ATM ABR

Arhitectură de rețea	Model de serviciu	Garantarea lărgimii de bandă	Livrare garantată	Ordonare	Temporizare	Notificarea congestiei
Internet	Best-effort	Nu	Nu	Posibil orice ordine	Nu este menținută	Nu
ATM	CBR	Rată constantă garantată	Da	În ordine	Da	Nu apar congestii
ATM	ABR	Rată minimă garantată	Nu	În ordine	Nu este menținută	Congestiile sunt semnalizate

Alte arhitecturi de rețea au definit și implementat modele de servicii care merg dincolo de serviciul *best-effort* al Internet-ului. Spre exemplu arhitectura de rețea ATM pune la dispoziție mai multe modele de servicii, ceea ce înseamnă că pot fi asigurate clase de servicii diverse pentru conexiuni. Prezentarea modului în care rețelele ATM asigură astfel de servicii ar necesita un întreg tratat; scopul nostru este doar să arătăm că există alternative la modelul *best-effort* din Internet. Două din cele mai importante modele de servicii ATM sunt serviciul cu rată de bit constantă (eng. constant bit rate, CBR) și serviciul cu rată de bit disponibilă (eng. available bit rate, ABR).

7.5. Subrețele cu circuite virtuale și subrețele datagramă

Nivelul legătură de date poate oferi atât un serviciu orientat conexiune cât și un serviciu fără conexiune. Spre exemplu Ethernet-ul pune la dispoziție un serviciu negarantat fără conexiune iar IEEE 802.11 pune la dispoziție un serviciu confirmat orientat conexiune. Similar nivelul transport din Internet pune la dispoziția aplicațiilor protocolul UDP (fără conexiune) și protocolul TCP (orientat conexiune). În mod similar, nivelul rețea dintr-o stivă de protocoale poate oferi un serviciu fără conexiune sau un serviciu orientat conexiune. Deși serviciile orientate conexiune și

¹ Cuvânt sau expresie care, în vorbire sau în scris, înlocuiește un cuvânt sau o expresie neplăcută

fără conexiune de la nivel rețea au anumite elemente comune cu cele de la nivelul legătură de date și transport există o serie de diferențe semnificative:

- La nivel rețea aceste servicii sunt de tip gazdă-la-gazdă fiind asigurate de nivelul rețea pentru nivelul transport. La nivelul legătură de date serviciile sunt nod la nod. La nivel transport aceste servicii sunt de tip proces-la-proces fiind asigurate de nivelul transport pentru nivelul aplicație.
- Majoritatea arhitecturilor pentru rețele de calculatoare de până acum (Internet, ATM, frame relay), asigură la nivel rețea fie un serviciu gazdă-la-gazdă orientat conexiune fie un serviciu gazdă-la-gazdă fără conexiune, însă în nici un caz pe ambele. Arhitecturile de rețea care asigură la nivel rețea doar un serviciu orientat conexiune sunt denumite *subrețele cu circuite virtuale (CV)*; arhitecturile de rețea care asigură la nivel rețea doar un serviciu fără conexiune sunt denumite *subrețele datagramă*.
- Implementarea serviciului orientat conexiune la nivel transport diferă fundamental de implementarea serviciului orientat conexiune la nivel rețea. La nivel transport serviciul orientat conexiune este implementat la periferia rețelei, în sistemele capăt pe când serviciul orientat conexiune de la nivel rețea este implementat în ruterele din nucleul rețelei precum și în sistemele capăt.

7.5.1. Implementarea subrețelilor cu circuite virtuale

Un circuit virtual constă din:

- O cale (serie de legături și rutere) între gazda sursă și gazda destinație;
- Un număr de circuit virtual;
- Întrări în tabela de rutare a ruterele aflate de-a lungul din căii.

Atunci când se stabilește o conexiune, se alege o cale între mașina sursă și mașina destinație, ca parte componentă a inițializării conexiunii și aceasta este memorată în tabelele ruterele. Acea cale este folosită pentru tot traficul de pe conexiune, exact în același mod în care funcționează sistemul telefonic. Atunci când conexiunea este eliberată, este închis și circuitul virtual. În cazul serviciilor orientate conexiune, fiecare pachet poartă un identificator care spune cărui circuit virtual îi aparține.

Pentru a ilustra acest concept să considerăm situația din Figura 7.3. Aici gazda H1 a stabilit conexiunea 1 cu gazda H2. Aceasta este memorată ca prima intrare în fiecare tabelă de rutare. Prima linie a tabelii lui A spune că dacă un pachet purtând identificatorul de conexiune 1 vine de la H1, atunci trebuie trimis către ruterul E, dându-i-se identificatorul de conexiune 1. Similar, prima intrare a lui D dirijează pachetul către F, tot cu identificatorul de conexiune 1.

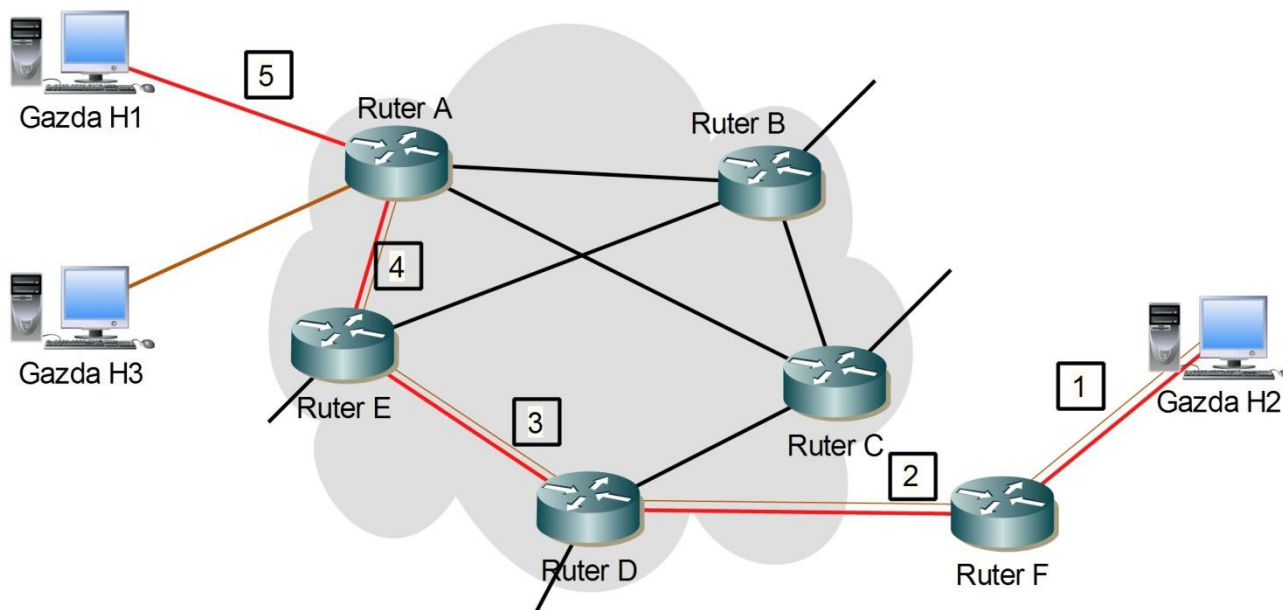


Figura 7.3 Rutare într-o subrețea cu circuite virtuale

Acum să vedem ce se întâmplă dacă H3 vrea, de asemenea, să stabilească o conexiune cu H2. Alege identificatorul de conexiune 1 (deoarece inițializează conexiunea și aceasta este singura conexiune) și indică subrețelei să stabilească circuitul virtual. Aceasta conduce la a doua linie din tabele. Observați că apare un conflict: deși routerul A poate distinge ușor pachetele conexiunii 1 de la H1 de pachetele conexiunii 1 de la H3, E nu poate face asta. Din acest motiv, A asociază un identificator de conexiune diferit pentru traficul de ieșire al celei de a doua conexiuni. Pentru evitarea conflictelor de acest gen, routerele trebuie să poată înlocui identificatorii de conexiune în pachetele care pleacă. În unele contexte, aceasta se numește comutarea etichetelor (eng. label switching).

Tabela ruterului A			
Interfața intrare	Nr. circuit	Interfața ieșire	Nr. circuit
H1	1	E	1
H3	1	E	2

Tabela ruterului D			
Interfața intrare	Nr. circuit	Interfața ieșire	Nr. circuit
E	1	F	1
E	2	F	2

Tabela ruterului E			
Interfața intrare	Nr. circuit	Interfața ieșire	Nr. circuit
A	1	D	1
A	2	D	2

Tabela ruterului F			
Interfața intrare	Nr. circuit	Interfața ieșire	Nr. circuit
D	1	H2	1
D	2	H2	2

Figura 7.4 Tabele de rutare pe calea H1-A-E-D-F-H2

Într-o subrețea cu CV, routerele trebuie să mențină *informații de stare* pentru conexiunile de ieșire. Mai exact de fiecare dată când se stabilește o nouă conexiune trebuie adăugată o nouă

intrare în tabela de rutare a fiecărui ruter aflat de-a lungul căii sursă-destinație; atunci când se eliberează o conexiune informația respectivă trebuie înlăturată.

7.5.2. Implementarea subrețelelor datagramă

Să presupunem că un proces de pe stația gazdă H1 dorește să transmită un segment unui proces de pe stația gazdă H2; mai presupunem că segmentul este de cinci ori mai lung decât dimensiunea maximă a unui pachet, așa că nivelul rețea de pe H1 trebuie să-l divizeze în cinci pachete pe care le trimite în parte ruterului A. Antetul fiecărui pachet va conține adresa de rețea completă a lui H2. Fiecare ruter are o tabelă internă care îi spune unde să trimită pachete pentru fiecare destinație posibilă. Fiecare intrare în tabelă este o pereche compusă din *destinație* și *linia de ieșire* folosită pentru acea destinație. Pot fi folosite doar linii conectate direct.

De exemplu, în Figura 7.5, ruterul A are patru linii de ieșire – către H1, B, C și E – astfel că fiecare pachet ce vine trebuie trimis către unul dintre aceste rutere, chiar dacă ultima destinație este alt ruter. Tabela de rutare inițială a lui A este prezentată în figură sub eticheta „inițial”.

Când pachetele 1, 2 și 3 au ajuns la A, sunt memorate pentru scurt timp (pentru verificarea sumei de control). Apoi folosind adresa destinație conținută în pachet și tabela lui A, fiecare pachet este trimis mai departe către E. Pachetul 1 a fost apoi trimis mai departe către D și apoi către F. Când a ajuns la F, datagrama este încapsulată într-un cadru al nivelului legătură de date și trimis către gazda H2.

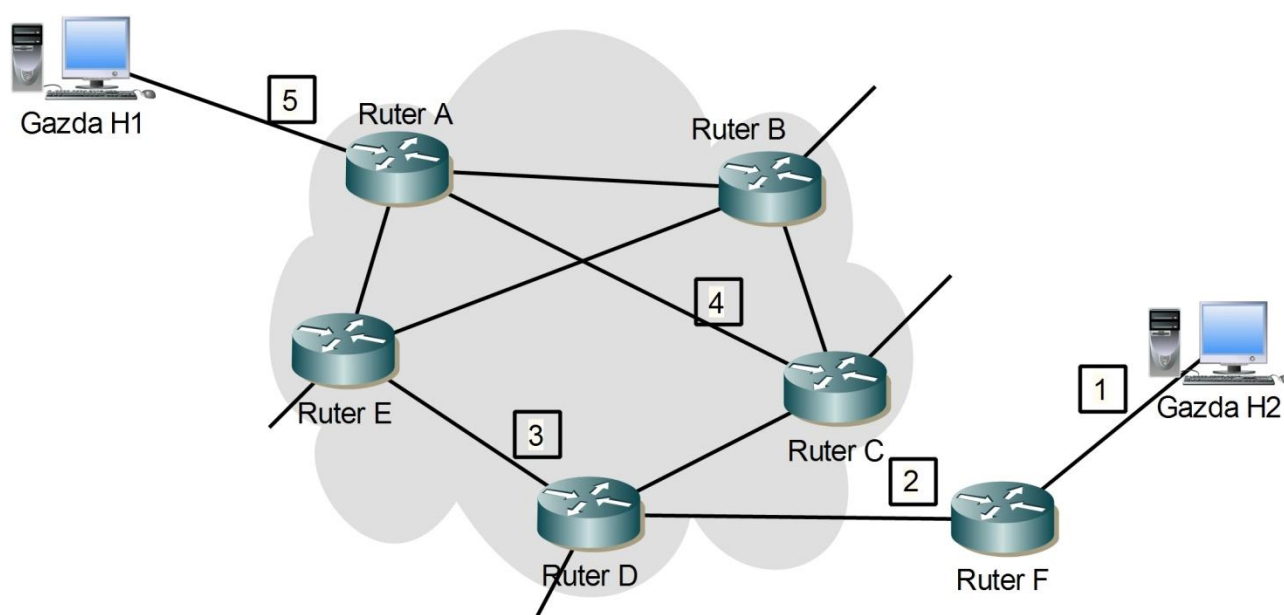


Figura 7.5 Rutarea într-o subrețea datagramă

Totuși, ceva diferit s-a întâmplat cu pachetul 4. Când a ajuns la A a fost trimis către ruterul C, chiar dacă și el este destinat tot lui F. Dintr-un motiv oarecare, A a decis să trimită pachetul 4 pe o rută diferită de cea urmată de primele trei. Poate că a aflat despre o congestie undeva pe calea AED și și-a actualizat tabela de rutare, așa cum apare sub eticheta „ulterior”.

Tabela ruterului A (inițial)	
Dest.	Linie de ieșire
A	-
B	B
C	C
D	E
E	E
F	E

Tabela ruterului A (ulterior)	
Dest.	Linie de ieșire
A	-
B	B
C	C
D	E
E	E
F	C

Tabela ruterului E	
Dest.	Linie de ieșire
A	A
B	B
C	D
D	D
E	-
F	D

Tabela ruterului D	
Dest.	Linie de ieșire
A	E
B	C
C	C
D	-
E	E
F	F

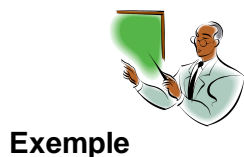
Pentru a explica mai în amănunt modul în care ruterul caută în tabelă linia de ieșire vom presupune că toate adresele destinație au 32 de biți. O implementare simplistă a tabelului de rutare ar implica existența unei intrări pentru fiecare adresă destinație posibilă. Întrucât cei 32 de biți permit peste 4 miliarde de adrese, o astfel de opțiune iese din calcul, întrucât duce la o tabelă de rutare imensă. Să presupunem că ruterul nostru are 4 legături, numerotate de la 0 la 3 iar pachetele sunt retrimise pe legăturile de ieșire astfel:

Interval adrese destinație	Interfață de ieșire
11001000 00010111 00010000 00000000 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 11001000 00010111 00011111 11111111	2
altfel	3

În mod evident, pentru o astfel de abordare, nu sunt necesare 4 miliarde de intrări în tabela de dirijare a ruterului. Putem de exemplu avea următoarea tabelă de rutare cu doar patru intrări:

Potrivire prefix	Interfață de ieșire
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
altfel	3

Cu o astfel de tabelă de rutare, ruterul realizează potrivirea adresei destinație a pachetului cu intrările din tabelă; dacă există potrivire atunci ruterul retrimite pachetul pe legătura asociată.



Exemple

Spre exemplu, sa presupunem că adresa destinație a pachetului este 11001000 00010111 00010110 10100001; întrucât prefixul de 21 de biți ai acestei adrese se potrivește cu prima intrare din tabelă, ruterul va retrimite pachetul pe interfața 0. Dacă un prefix nu se potrivește cu nici una din primele 3 intrări, atunci pachetul este retrimis pe interfața 3. Probabil ați observat că este posibil ca o adresă destinație să se potrivească cu mai mult de o intrare. Spre exemplu, primii 24 de biți ai adresei 11001000 00010111 00011000 10101010 se potrivesc pe a doua intrare din tabelă, iar primii 21 de biți ai aceleiași adrese se potrivesc pe cea de a treia intrare din tabelă.

Când există mai multe potriviri, ruterul folosește *regula potrivirii celui mai lung prefix*; adică va găsi cea mai lungă intrare potrivită din tabelă și va retrimite pachetul pe legătura asociată cu potrivirea prefixului cel mai lung.

7.5.3. Comparație între subrețele cu circuite virtuale și subrețele datagramă

Atât circuitele virtuale cât și datagramele au suporterii și oponenții. Vom rezuma argumentele ambelor tabere, principalele aspecte fiind prezentate în Tabelul 7.3. Cei extrem de riguroși ar putea probabil găsi un contraexemplu pentru fiecare caracteristică descrisă în tabel.

Tabelul 7.3 Comparație între subrețele datagramă și subrețele cu circuite virtuale

Problemă	Subrețea datagramă	Subrețea cu circuite virtuale (CV)
Stabilirea circuitului	Nu este necesară	Obligatorie
Adresare	Fiecare pachet conține adresa completă pentru sursă și destinație	Fiecare pachet conține un număr mic de CV
Informații de stare	Ruterele nu păstrează informații despre conexiuni	Fiecare CV necesită spațiu pentru tabela ruterului per conexiune
Dirijare	Fiecare pachet este dirijat independent	Calea este stabilită la inițierea CV; toate pachetele o urmează
Efectul defectării ruterului	Nici unul, cu excepția pachetelor pierdute în timpul defectării	Toate circuitele virtuale care trec prin ruterul defect sunt terminate
Calitatea serviciului	Dificil	Simplu, dacă pentru fiecare CV pot fi alocate în avans suficiente resurse

Controlul congestiei	Dificil	Simplu, dacă pentru fiecare CV pot fi alocate în avans suficiente resurse
----------------------	---------	---

În interiorul subrețelei există situații în care trebuie să se aleagă între facilități antagoniste specifice fie circuitelor virtuale, fie datagramelor. Un astfel de compromis este acela între spațiul de memorie al ruterului și lățimea de bandă. Circuitele virtuale permit pachetelor să conțină numere de circuite în locul unor adrese complete. Dacă pachetul tinde să fie foarte mic, atunci existența unei adrese complete în fiecare pachet poate reprezenta o supraîncărcare importantă și deci o irosire a lățimii de bandă. Prețul plătit pentru folosirea internă a circuitelor virtuale este spațiul necesar păstrării tabelului în ruter. Soluția mai ieftină este determinată de raportul între costul circuitelor de comunicație și cel al memoriei ruterului.

Alt compromis este cel între timpul necesar stabilirii circuitului și timpul de analiză a adresei. Folosirea circuitelor virtuale presupune existența unei faze inițiale de stabilire a căii, care cere timp și consumă resurse. Oricum, este ușor să ne imaginăm ce se întâmplă cu un pachet de date într-o subrețea bazată pe circuite virtuale: ruterul folosește numărul circuitului ca un index într-o tabelă pentru a afla unde merge pachetul. Într-o rețea bazată pe datagrame, pentru a găsi intrarea corespunzătoare destinației se folosește o procedură de căutare mult mai complicată.

O altă problemă este cea a dimensiunii spațiului necesar pentru tabela din memoria ruterului. O subrețea datagramă necesită o intrare pentru fiecare destinație posibilă, în timp ce o rețea cu circuite virtuale necesită o intrare pentru fiecare circuit virtual. Totuși, acest avantaj este relativ iluzoriu deoarece și pachetele de inițializare a conexiunii trebuie rutate, iar ele folosesc adresele destinație, la fel ca și datagramele.

Circuitele virtuale au unele avantaje în garantarea calității serviciului și evitarea congestionării subrețelei, deoarece resursele (de exemplu zone tampon, lărgime de bandă și cicluri CPU) pot fi rezervate în avans, atunci când se stabilește conexiunea. La sosirea pachetelor, lățimea de bandă necesară și capacitatea ruterului vor fi deja pregătite. Pentru o subrețea bazată pe datagrame, evitarea congestionării este mult mai dificilă.

Pentru sistemele de prelucrare a tranzacțiilor (de exemplu apelurile magazinelor pentru a verifica cumpărături realizate cu cărți de credit) supraîncărcarea implicată de stabilirea și eliberarea unui circuit virtual poate reduce cu ușurință utilitatea circuitului. Dacă majoritatea traficului este de acest tip, folosirea internă a circuitelor virtuale în cadrul subrețelei nu prea are sens. Pe de altă parte, ar putea fi de folos circuite virtuale permanente, stabilite manual și care să dureze luni sau chiar ani.

Circuitele virtuale au o problemă de vulnerabilitate. Dacă un ruter se defectează și își pierde conținutul memoriei, atunci toate circuitele virtuale care treceau prin el sunt suprimate, chiar

dacă acesta își revine după o secundă. Prin contrast, dacă se defectează un ruter bazat pe datagrame vor fi afectați doar acei utilizatori care aveau pachete memorate temporar în cozile de așteptare ale ruterului și este posibil ca numărul lor să fie și mai mic, în funcție de câte pachete au fost deja confirmate.

Pierderea liniei de comunicație este fatală pentru circuitele virtuale care o folosesc, însă poate fi ușor compensată dacă se folosesc datagrame. De asemenea, datagramele permit ruterului să echilibreze traficul prin subrețea, deoarece căile pot fi modificate parțial în cursul unei secvențe lungi de pachete transmise.

Rezumat



Subrețelele datagramă și subrețelele cu circuite virtual reprezintă două abordări majore în implementarea nivelului rețea utilizate de-a lungul timpului. Internet-ul este o rețea de tip datagramă, unde fiecare datagramă de nivel rețea include adresa destinație completă. Această adresă este utilizată de rutere pentru a dirija datagrama către destinația finală. Întrucât tabelele de rutare din subrețelele datagramă se pot modifica oricând, există posibilitatea ca o serie de pachete trimise de un sistem capăt să parcurgă căi diferite prin rețea și să ajungă la destinație în altă ordine decât cea în care au fost trimise.

Într-o subrețea cu circuite virtuale fiecare pachet de date include un număr de circuit virtual, folosit de către rutere pentru dirijarea pachetului spre destinație. Ideea care se stă la baza circuitelor virtuale este evitarea alegerii unei noi căi (rute) pentru fiecare pachet trimis. Într-o subrețea cu circuite virtuale, înainte de a putea transfera date efective este necesară stabilirea unei conexiuni între sursă și destinație, prin care se înscriu în tabelele de rutare aflate pe calea sursă destinație interfețele de ieșire corespunzătoare numărului circuitului virtual.



Bibliografie

Andrew S. Tanenbaum, *Computer Networks*, 4/E, Prentice Hall, 2003

James F. Kurose and Keith W. Ross, *Computer Networking A Top Down Approach*, 5/E, Pearson Education, 2009

William Stallings, *Data and Computer Communications*, 9/E, Pearson Education, 2011