3. Nivelul retea

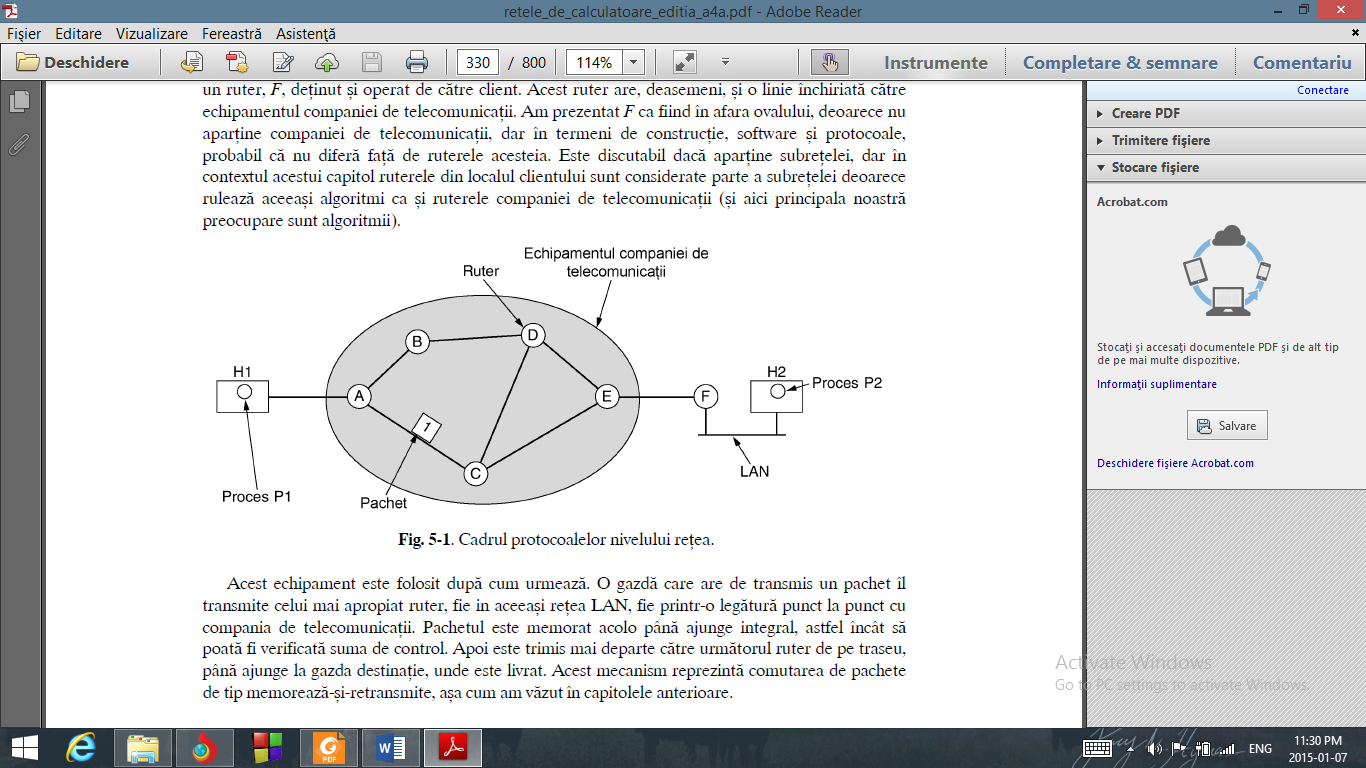
3.a. CERINŢELE DE PROIECTARE ALE NIVELULUI REŢEA

Datagrams = packets

# Comutare de pachete de tip Memorează-şi-Retransmite (Store-and-Forward)

*Componente sistem:*

* *Routere conectate prin linii de transmisie (ale companiei)*
* *Echipament client*



Transmisie pachet:

Gazda ---pachet spart in subpachete---> closest router (memoreaza pachet incomplet, wait until all subpachete get here to check suma de control) ---pachet spart in subpachete---> next router ---> gazda destinatie

# Servicii furnizate nivelului transport

Serviciile nivelului reţea au fost proiectate având în vedere următoarele scopuri:

1. Serviciile sunt independente de:

* tehnologia routerelor
* tipul routerelor
* topologia routerelor

1. **Adresele de reţea** disponibile la nivelul transport sunt numerotate uniform, chiar si în cadrul reţelelor LAN şi WAN.
2. **Servicii** de 2 tipuri:

* Neorientat pe conexiune
* Orientat pe conexiune

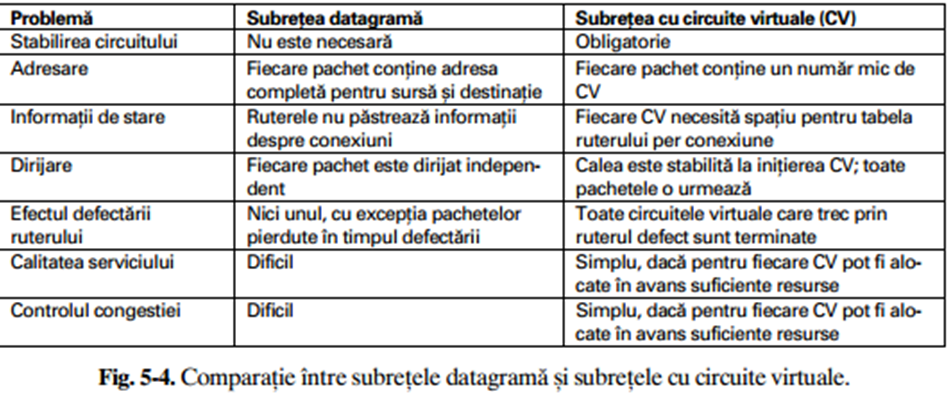
# Implementarea serviciului neorientat pe conexiune

* **Pachetele (datagrams)** sunt trimise in **subreţea (datagram subnet)** individual şi dirijate independent de celelalte.
* Proces1 [nivel retea] ---mesaj (+ inst de livrare catre Proces2)---> [nivel transport] ---ruleaza cod transport pe SO---> [nivel transport] ---mesaj spart in datagrame---> [nivel retea] ---mesaj spart in datagrame---> closest router [**memoreaza-si-retransmite**] ---mesaj spart in datagrame---> closest router ---…---> router F ---mesaj---> router F [niv. leg de date] ---mesaj---> calc. destinatie

# Implementarea serviciilor orientate pe conexiune

* Inainte de a trimite pachetele se stabileste o conexiune (virtual circuit) de la router sursa la router destinatie; subretea (virtual circuit subnet).
* Beneficiu: evitarea alegerii unei rute noi pentru fiecare pachet
* Dezavantaj: conexiunea trebuie memorata in fiecare router

# Comparaţie între subreţele cu circuite virtuale şi subreţele datagram



3.b. Algoritmi de dirijare

Algoritmul de dirijare (routing algorithm) este acea parte a software-ului nivelului reţea care  
răspunde de alegerea liniei de ieşire pe care trebuie trimis un pachet recepţionat

# Calea cea mai scurta

Modelul topologic al unei retele este un graf in care nodurile corespund comutatoarelor de pachete, iar muchiile corespund liniilor de comunicatie.

Asociind fiecarei muchii o lungime (dist. geografica , cost comunicatie, intarziere medie) se poate calcula calea cea mai scurta intre oricare doua noduri. Se foloseste o varianta modificata a alg.lui Dijkstra:

Fiecare nod ruleza independent un algoritm ca sa determine cea mai scurta cale de la el insusi pana la orice alt nod in retea.

Initial: Asigneaza fiecarui nod o distanta foarte mare. Asigneaza 0 pentru nodul initial si infinit pentru celelalte.

1. Marcheza nodul initial.
2. Pentru nodul current, considera toti vecinii acestuia si compara noua distanta calculata (distanta de la nodul root pana la nodul curent+ costul de la nodul curent pana la vecinul curent) la cu cea deja asignata si alege minimul. Cand termina de verificat toti vecinii, marcheza nodul curent.
3. Selecteza un nod din cele nevizitate care are cea mai mica distanta. Repeta pas 2.

# Dirijarea centralizata

Varianta centralizata a algoritmului drumurilor minime (Floyd) utilizeaza un tablou dist al distantelor minime, dist[i][j] fiind distanta minima de la nodul i la nodul j: la iteratia k, dist[i][j] va avea ca valoare cea mai buna distanta intre i si j, pe cai care nu contin noduri numerotate peste k (exceptand i si j).

1 **let** dist be a |V| × |V| array of minimum distances initialized to ∞ (infinity)

2 **for each** vertex *v*

3 dist[*v*][*v*] ← 0

4 **for each** edge (*u*,*v*)

5 dist[*u*][*v*] ← w(*u*,*v*) *// the weight of the edge (*u*,*v*)*

6 **for** *k* **from** 1 **to** |V|

7 **for** *i* **from** 1 **to** |V|

8 **for** *j* **from** 1 **to** |V|

9 **if** dist[*i*][*j*] > dist[*i*][*k*] + dist[*k*][*j*]

10 dist[*i*][*j*] ← dist[*i*][*k*] + dist[*k*][*j*]

11 **end if**

**A centralized node makes all routing decisions. Specifically, the centralized node has access to global information.**

Deficientele acestei metode sunt determinate de:

* vulnerabilitatea retelei, dependenta de functionarea centrului de control
* supraincarcarea traficului prin transmiterea rapoarelor si a tabelelor de dirijare;
* utilizarea in noduri, in anumite perioade, a unor tabele necorelate, datorita receptiei la momente de timp distincte a noilor tabele.

# Dirijarea izolata

* + **Pachetul receptionat de nod este dirijat catre linia cu coada cea mai scurta.**
  + Folosit in in aplicatii militare (datorita robustetii sale) sau in comparatii de performanta cu alte tehnici (deoarece are un timp de intirziere minim).

# Dirijarea distribuita

* + Varianta modificata a algoritmului lui Dijkstra care calculeaza drumurile minime de la toate nodurile catre o anumita destinatie
  + Este descentralizat
  + Asigura gasirea drumurilor minime intr-un numar finit de pasi
  + Poate fi utilizat doar pentru datagrame

# Dirijarea ierarhica

* Se utilizeaza pentru retele mari
* Comutatoarele sunt grupate in regiuni, fiecare comutator cunoscand caile din regiunea proprie, dar nu si structura interna a altor regiuni.
* Doua regiuni sunt legate prin conectarea unui anumit nod din prima regiune cu un anumit nod din a doua regiune.
* Tabela de dirijare se poate reduce, ea avand cate o intrare pentru fiecare nod din regiunea proprie si cite o intrare pentru fiecare din celelalte regiuni.

# Inundarea

* Un alt algoritm static este inundarea (flooding), în care fiecare pachet recepţionat este trimis mai  
  departe pe fiecare linie de ieşire, cu excepţia celei pe care a sosit
* Inundarea generează un număr infinit de pachete dacă nu se iau unele măsuri pentru a limita acest proces

3.c. ALGORITMI PENTRU CONTROLUL CONGESTIEI

Congestie = mai multe pachete intr-o subretea. Cauze:

* sosirea mai multor pachete care necesita aceeasi linie de iesire
* Insuficienta memorie
* Procesoare lente

# Solutii:

1. **Prealocarea zonelor tampon**
   * Este aplicabila circuitelor virtuale si consta in rezervarea uneia sau mai multor zone tampon in fiecare nod intermediar, la deschiderea circuitului. In lipsa de spatiu, se alege o alta cale sau se rejecteaza cererea de stabilire a circuitului.
2. **Distrugerea pachetelor**
   * Daca nu exista spatiul necesar memorarii, pachetul receptionat de un nod este ignorat. Deoarece astfel se pot ignora pachete de confirmare, care ar duce la eliberarea spatiului ocupat de pachetele confirmate, se mentine cel putin un tampon de receptie pentru fiecare linie, permitindu-se inspectarea pachetelor primite. De asemenea, se poate limita numarul zonelor tampon de transmisie ale fiecarei linii.
3. **Pachete de permisiune.**
   * Se initializeaza reteaua cu pachete de permisiune (in numar fix). Cand un nod vrea sa transmita, el captureaza un pachet de permisiune si trimite in locul lui pachete de date. Receptorul regenereaza pachetul de permisiune. Se garanteaza astfel ca numarul maxim de pachete nu depaseste numarul de pachete de permisiune, fara a se asigura distribuirea lor conform necesitatilor nodurilor. In plus, pierderea pachetelor de permisiune conduce la scaderea capacitatii retelei.
4. **Pachete de soc.** 
   * Sunt transmise de comutatoare surselor de date pentru a micsora rata de generare a pachetelor.

# Evitarea blocarii definitive.

Blocarea = lipsa de spatiu impiedica transmiterea vreunui pachet.

Solutii:

* utilizarea in fiecare nod a m+1 zone tampon, m = lungimea max. a cailor retelei. Un pachet sosit de la calculatorul gazda local este acceptat in zona 0. Zona "m" a unui nod poate fi goala, poate contine un pachet pentru gazda locala, care este livrat, sau are un pachet pentru un nod distant, care este distrus. In toate cazurile zona "m" se elibereaza, putand avansa un pachet din zona "m-1", apoi "m-2" etc.
* pastreaza pentru fiecare pachet o informatie de vechime. La comunicarea dintre doua noduri A si B putem intalni situatiile urmatoare (pp ca A are de transmis lui B un pachet mai vechi decat B catre A):
* B are un tampon liber si poate primi cel mai vechi pachet al lui A catre B;
* B nu are un tampon liber, dar are un pachet pentru A si poate primi, prin schimb, cel mai vechi pachet al lui A catre B;
* B nu are nici un tampon liber si nici un pachet catre A; in acest caz, B este fortat sa transmita lui A un pachet la alegere si sa primeasca cel mai vechi pachet al lui A catre B

3.d. Protocolul ip

O datagramă IP constă dintr-o parte de antet şi o parte de text. Antetul are o parte fixă de 20 de octeţi şi o parte opţională cu lungime variabilă. El este transmis în ordinea **big endian** (cel mai semnificativ primul): de la stânga la dreapta, începând cu bitul cel mai semnificativ al câmpului Versiune.

* Utilizat de sisteme autonome in vederea interconectarii
* Serviciu de transmitere de pachete (host­to­host)
* Translatare dintre diferite protocoale legatura de date
* Ofera servicii neorientate‐conexiune, nesigure: datagrame
* Fiecare datagrama este independenta de celelalte
* Nu se garanteaza transmiterea corecta a datagramelor (pierdere, multiplicare, ...)
* Foloseste doar adresele logice ale gazdelor
* Adresele IP nu sunt identice cu cele ale nivelului MAC (e.g., adresele hardware ale placilor de retea) pentru ca IP trebuie sa suporte diferite implementari hardware (retele eterogene)