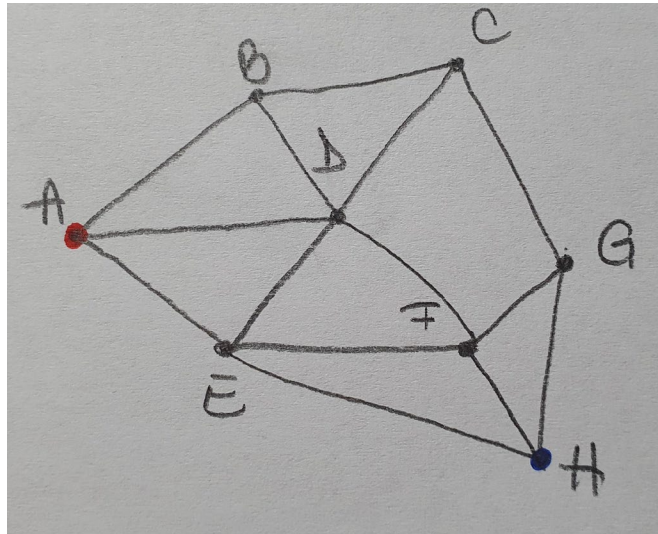


## CURS 12

Internetul e organizat prin interconectarea mai multor rețele.



Dacă nodul A vrea să trimită la nodul H, ruta ar fi  $A \rightarrow E \rightarrow H$ .

**Reguli de dirijare** = informații care îi spun unei surse la ce vecin trebuie să trimită informația ca să ajungă la destinație

Se păstrează o **tabelă de dirijare**

Tabela de dirijare (de rută) a nodului A:

Destinație	Vecin	Lungimea drumului
H	E	2
A	A	0
B	B	1
C	D B	2
D	D	1
E	E	1
F	D D	2
G	D E	3

Nu e bine să suprasolicităm o linie de comunicație (un nod).

Din perspectiva lui A, câte noduri în graf avem, atâtea rute avem în tabelă.

În rețeaua Internet, există niște **algoritmi de dirijare** care permit construirea unei tabele de dirijare în mod *dinamic*.

Există două clase mari de algoritmi de dirijare:

- bazați pe vectori distanță
- bazați pe starea legăturilor

Ca și *metrici de cost*, se află:

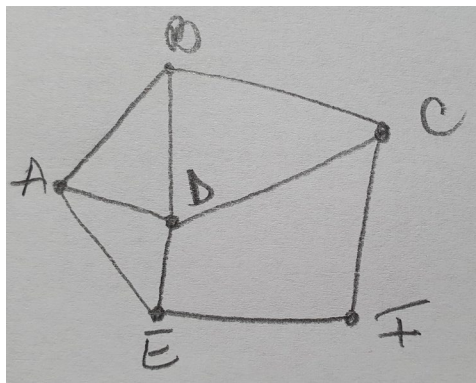
- lungimea
- viteza liniilor de comunicații
- delay-ul liniilor de comunicații
- costul (\$\$)
- lungimea cozilor de așteptare existente la nivelul routerelor

Când într-un router cu  $n$  linii de comunicație intră foarte multe date, se poate ajunge la situația în care vin date la router pe  $n - 1$  linii, toate datele trebuie trimise pe router pe o singură linie de comunicație. Ca să facă față, routerele țin o coadă de așteptare în care se pun, temporar, pachetele care vin pe liniile de comunicații.

Cu cât coada este mai plină, cu atât linia de ieșire asociată cozii e mai congestionată (aglomerată).

Când se umple coada de așteptare, pachetele sunt “*aruncate la gunoi*” și nu vor fi livrate la destinație.

### **Algoritmi bazați pe vector distanță**



Fiecare nod își populează tabela de dirijare cu ce știe. După ce un nod își pune ruta spre el însuși, două routere adiționale încep și schimbă tabelele de dirijare. De exemplu, A și B își trimit unul celuilalt propria tabelă de înregistrare. Astfel, A află că B ajunge la B cu un drum de lungime 0, și B învață același lucru despre A. Se întâmplă asta cu toți vecinii.

Fiecare router pleacă de la o tabelă minimală și ajung să aibă informații despre cât mai multe destinații și își populează tabela de dirijare cu tot mai multe înregistrări.

!!! Protocoale de dirijare care implementează acest algoritm: *RIP* (**R**outing **I**nternet **P**rotocol), *BGP* (**B**order **G**ateway **P**rotocol – protocolul router-ului de frontieră).

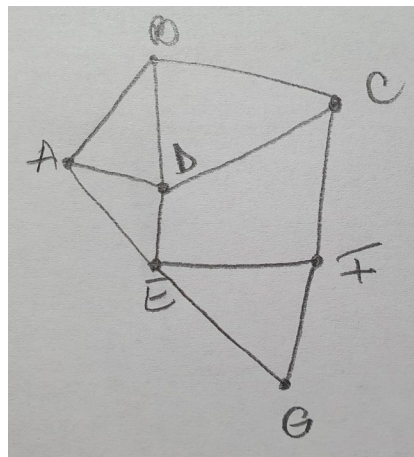
#### Avantaje:

- se pot implementa în rețele mari
- fiecare nod, făcând niște calcule parțiale, și bazându-și determinarea tabelului de dirijare pe niște calcule făcute de vecini, fiecare nod nu are de făcut atâtea calcule, algoritmul fiind *mai puțin intens* computațional

#### Dezavantaj:

- apare *problema numărării la infinit*

#### Algoritmi bazați pe starea legăturilor (se bazează în spate pe Dijkstra)



Fiecare router din rețea pregătește niște pachetele: (sursă, destinație, cost, timestamp). Acest pachetel descrie starea legăturii dintre sursă și destinație: la momentul de timp *timestamp*, era linie de cost *cost* de la sursă la destinație.

Aceste pachetele sunt trimise de la sursă la fiecare vecin. Astfel, vecinii află starea legăturilor adiacente ale sursei.

La fel, fiecare vecin trimite pachetele respective trimise de sursă, prin tehnica de *inundare*, la vecinii lor.

În final se ajunge ca fiecare nod să aibă informații despre starea legăturilor adiacente tuturor celorlaltor noduri din graf.

### **Avantaj:**

- nu mai apare problema numărării la infinit

### **Dezavantaj:**

- este mai intens computațional, fiecare nod face calculele individual și pentru el
- un astfel de algoritm nu poate fi implementat și deploy-at într-o rețea de dimensiuni mari

În Internet, se face dirijarea pe două nivele:

- la nivel macro, rețeaua fiind o rețea de rețele – se face dirijare folosind *BGP*
- la nivelul unui sistem autonom (AS – **A**utonomus **S**ystem) – se pot utiliza algoritmi de dirijare *IGRP* (**I**nterior **G**ateway **R**outing **P**rotocols) – se pot baza pe algoritmii cu testarea legăturilor

**Propagarea rutei** = un router spune că el “*ține în spate*” unele clase de adrese, iar această informație se propagă de-a lungul rețelei