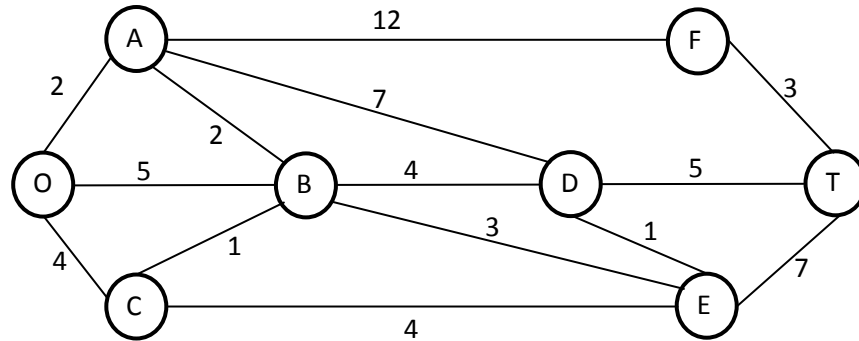
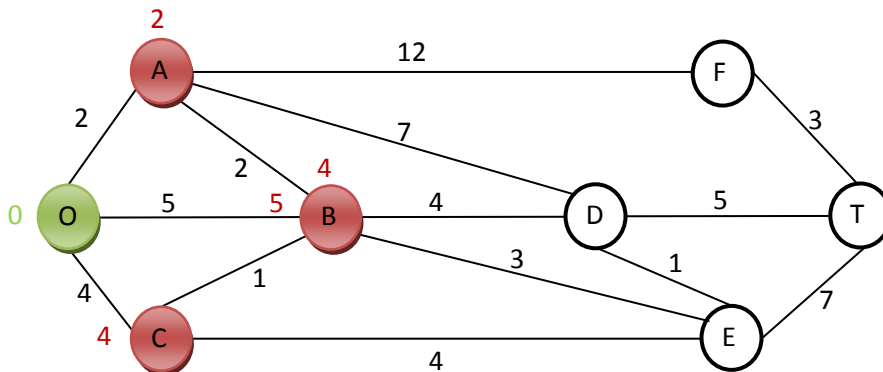


## Algoritmul Dijkstra

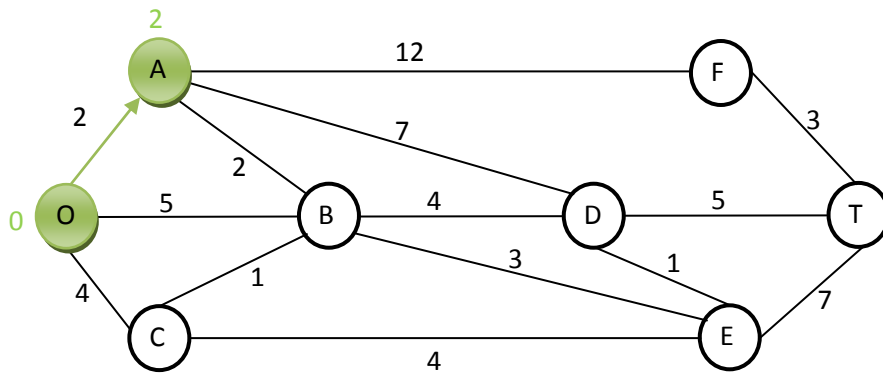


Gasirea rutei celei mai scurte de la ruterul O la ruterul T. Fiecare nod al grafului este un ruter. Cifra asociata unui arc  $ij$  reprezinta timpul necesar unui pachet pentru a ajunge de la ruterul  $i$  la ruterul  $j$ .

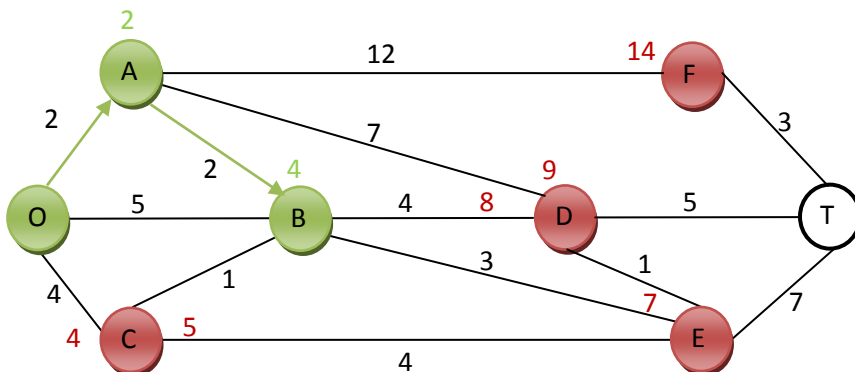
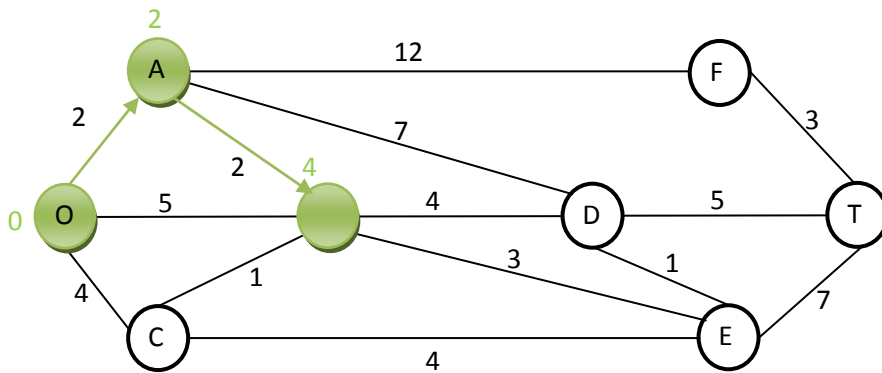
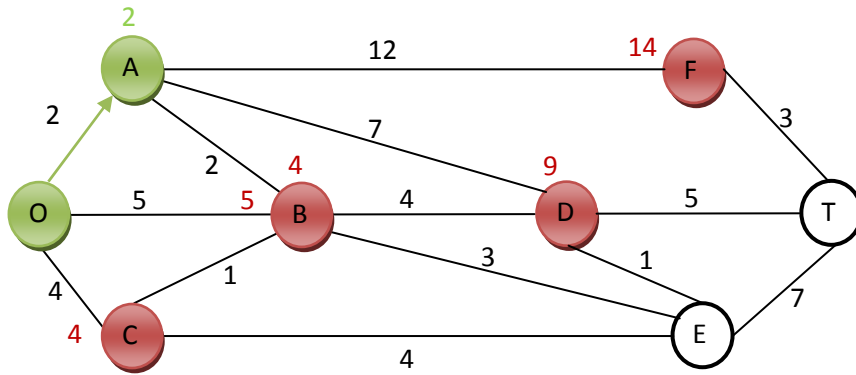
- 1) Marcam ruterul O drept nod rezolvat. Ii adugam o eticheta 0, deoarece distanta pana la el insusi este de 0 unitati.
- 2) Se identifica toate nodurile nerezolvate conectate la orice nod rezolvat.
- 3) Pentru fiecare arc ce leaga un nod rezolvat cu un nod nerezolvat, se calculeaza distanta ca fiind suma dintre distanta pana la nodul rezolvat si lungimea arcului.
- 4) Se alege nodul care are distanta cea mai mica (daca avem mai multe distante egale se alege la intamplare).
- 5) Se repeta etapele anterioare pana nu mai avem noduri nerezolvate.

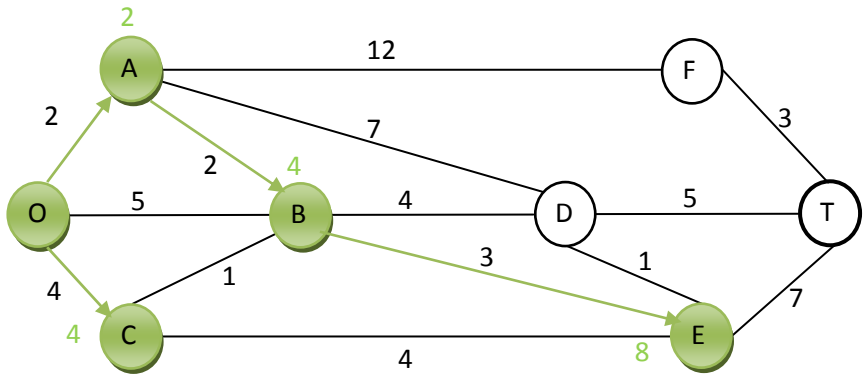
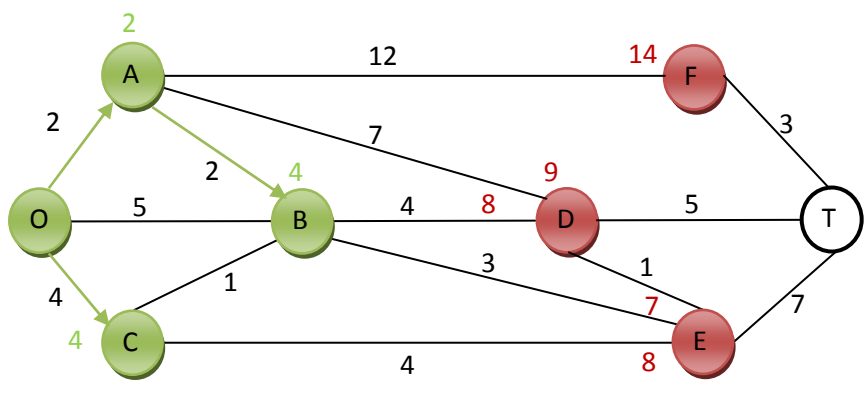
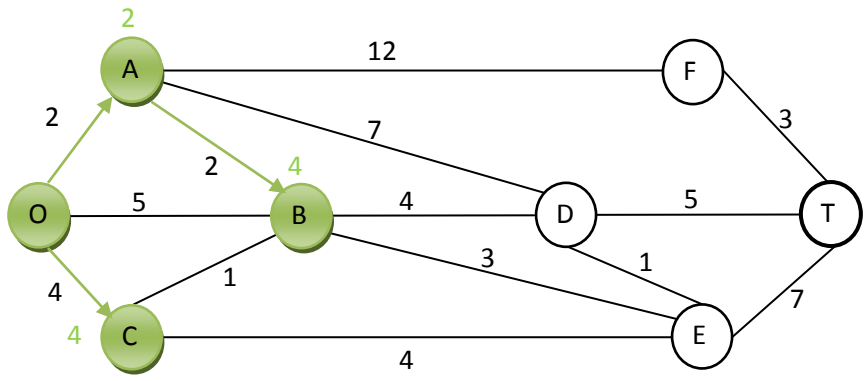


Nodul castigator este nodul A deoarece are distanta cea mai mica. Prin urmare nodul A devine nod rezolvat.



Se repeta pasul 2 si 3.





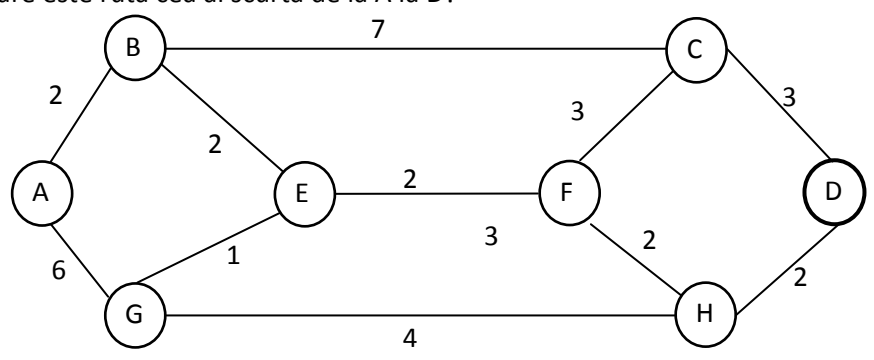
Se continua pasii 2,3 si 4 pana ce avem toate nodurile rezolvate.

Exista doua rute posibile pentru ca un pachet sa ajunga de la ruterul O la ruterul T:

- 1)  $O \rightarrow A \rightarrow B \rightarrow D \rightarrow T$
- 2)  $O \rightarrow A \rightarrow B \rightarrow E \rightarrow D \rightarrow T$

Exista si o ruta incompleta de la O la C.

Ex: Care este ruta cea mai scurta de la A la D?



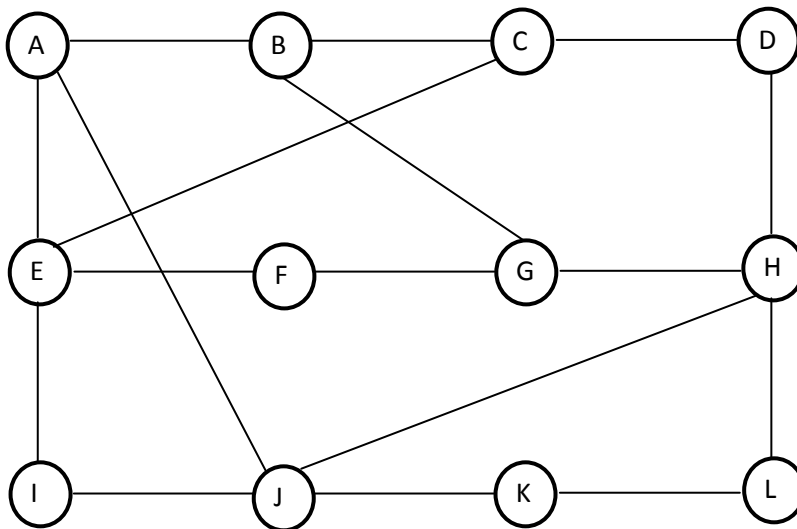
## Rutarea bazata pe vectori distanta

Algoritmul de rutare bazat pe vectori distant este unoscult si sub numele de Bellman-Ford sau Ford-Fulkerson.

In rutarea bazata pe vectori distant, in tabela de rutare a fiecarui ruter se gasesc infratii despre fiecare ruter din subretea. Informatia este alcatuita din doua parti: prima este calea preferata pentru a transmite pachetul de la sursa la destinatie si a doua timpul/distanta estimat(a) pana la destinatie. Metrica folosita poate fi numarul de hopuri, intarzierea in msec, etc. Sa presupunem ca ruterul cunoaste distanta catre fiecare din vecini. Daca metrica folosita este hop-ul atunci distant este doar un hop. Daca este intarzierea, ruterul o masoara folosind pachete speciale ECHOpe care receptorul le eticheteaza si le transmite inapoi cat poate de repede.

Ex: Sa presupunem ca se foloseste drept metrica timpul de intarziere si ca fiecare ruter cunoaste timpul de intarziere catre fiecare dintre vecinii sai. La fiecare  $T_{msec}$ , fiecare ruter transmite catre fiecare vecin o lista cu timpii de intarziere estimati pana la destinatie. De asemenea el primeste o lista similara de la fiecare vecin. De exemplu, daca o astfel de tabela ajunge la un ruter de la vecinul X, iar  $X_i$  este timpul estimat de X pentru ca un pachet sa ajunga la ruterul i. Daca ruterul stie ca timpul pana la X este de m msec, atunci el stie de asemenea ca un pachet transmis aunge la ruterul i in  $X_i + m$  msec. Facand acest calcul pentru fiecare vecin, un ruter poate gasi ce timp de intarziere estimat este mai bun si sa foloseasca acest timp si calea corespunzatoare pentru care s-a obtinut acest timp in noua lui tabela de rutare.

Pentru retea din figura construiti noua tabela de rutare pentru ruterul J.



Cum se determina noua ruta de la J la G? Se cunosc vectorii primiti de la cei patru vecini ai lui J.

	A	I	H	K
A	0	24	20	21
B	12	36	31	28
C	25	18	19	36
D	40	27	8	24
E	14	7	30	22
F	23	20	19	40
G	18	31	6	31
H	17	20	0	19
I	21	0	14	22
J	9	11	7	10
K	24	22	22	0
L	29	33	9	9
	JA	Ji	JH	JK
delay	8	10	12	6

#### Noua tabela de rutare pentru J

A	8	A
B	20	A
C	28	I
D	20	H
E	17	I
F	30	I
G	18	H
H	12	H
I	10	I
J	0	-
K	6	K
L	15	K

#### Vectorii primiti de la cei 4 vecini ai lui J

Pentru a ajunge la G prin ruterul A, noua intarziere calculata va fi urmatoarea:

$J \rightarrow A$  8 msec plus  $A \rightarrow G$  inca 18 msec, deci in totul un timp de intarziere estimate de 26 de msec.

Pentru a ajunge la G prin ruterul I, noua intarziere calculata va fi urmatoarea:

$J \rightarrow I$  10 msec plus  $I \rightarrow G$  inca 31 msec, deci in totul un timp de intarziere estimate de 41 de msec.

Pentru a ajunge la G prin ruterul H noua intarziere calculata va fi urmatoarea:

$J \rightarrow H$  12 msec plus  $H \rightarrow G$  inca 6 msec, deci in totul un timp de intarziere estimate de 18 de msec.

Pentru a ajunge la G prin ruterul K, noua intarziere calculata va fi urmatoarea:

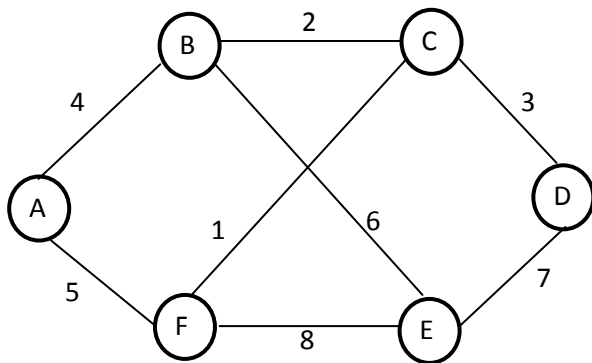
$J \rightarrow K$  6 msec plus  $K \rightarrow G$  inca 31 msec, deci in totul un timp de intarziere estimate de 37 de msec.

## Rutarea bazata pe starea legaturii

- 1) Descoperirea vecinilor si invatarea adreselor.
- 2) Masurarea delay-ului sau a costului catre fiecare ruter.
- 3) Construirea unui pachet care precizeaza tot ceea ce a invatat.
- 4) Trimiterea pachetului catre toate ruterele.
- 5) Calcularea caili celei mai scurte catre fiecare ruter.

Fiecare ruter isi construiesc un pachet care contine toate datele. Pachetul contine identitatea celui care trimite pachetul, un numar de secventa, o varsta si o lista de vecini.

Ideea de baza este foloarea inundarii pentru distribuirea pachetelor. Fiecare pachet contine un numar de secventa care este incrementat la fiecare transmitere. Atunci cand un nou pachet ajunge la un ruter, acesta este verificat cu celelalt pachete existente. Daca este un pahet nou, el este forwardat. Daca este duplicat(are acelasi numar de secventa), atunci pachetul este aruncat. Daca este un pachet nu un numar de secventa inferior celui mai mare de pana acum transmis pachetul este rejectat.



Pentru calcularea rutei celei mai scurte se poate folosit algoritmul lui Dijkstra.

A		B		C	
Secv		Secv		Secv	
Vârsta		Vârsta		Vârsta	
B	4	A	4	B	2
E	5	C	2	D	3
		F	6	E	1

Link state packets

D		E		F	
Secv		Secv		Secv	
Vârsta		Vârsta		Vârsta	
C	3	A	5	B	6
F	7	C	1	D	7
		F	8	E	6

