

# Hálózati Technológiák és Alkalmazások

Vida Rolland, BME TMIT

2018. október 15.



# Distance Vector protokollok

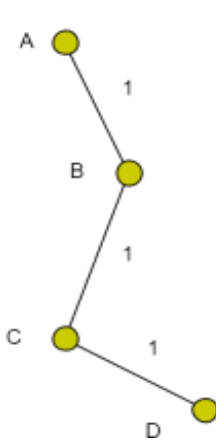
- Távolságvektorokat tárol az útvonalakról
- **Adathármasok:**
  - Cél (mi a célállomás)
  - Költség
  - Következő csomópont (merre küldje) – next hop
- Rendszeresen frissítik adataikat a közvetlen szomszédok
  - Frissítő üzenet (2 részből áll):
    - Cél, költség
  - Ha a router ilyenkor jobb utat talál egy célhoz, frissíti tábláját, 2 ok miatt:
    - Kisebb költségű utat talál
    - Szomszéd költsége megváltozik

# Jellemzők

- Egyszerű, de nem tökéletes:
  - A kapcsolatok ára változhat
  - Kapcsolatok meg is szakadhatnak
    - Egy megszakadt kapcsolat ára végtelen
      - Egy olyan egész érték, mely nagyobb bármilyen lehetséges valós értéknél (RIP-nél 16)
  - A routerek topológia változás esetén nem egyszerre frissítik táblázataikat
    - Periodikus időközönként (pl. 30 s) frissítő üzenet
    - Ha 6 frissítés elmarad, az ár végtelen
    - A szomszédok is frissítik a bejegyzéseiket
  - **Konvergál, de lassan**
    - Csak kis hálózatokban használható

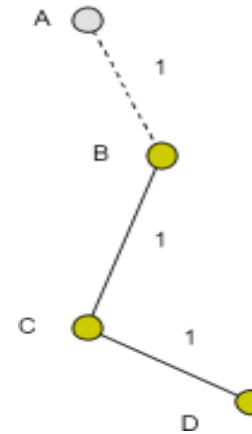
# Végtelenig számolás

- A routerek a célcím költségek hirdetésekor végtelenig inkrementálhatnak



B	C	D
1	2	3

Távolság A felé



B	C	D
3	2	3

B	C	D
3	4	3

B	C	D
5	4	5

# Megakadályozás

- **Split horizon módszer**
  - ha C B-től megismer egy utat, akkor az azt kiterjesztő utat B-vel már nem közli
- **Poisoned Reverse módszer**
  - Végtelen elérhetőség hirdetése az adott linken elérhető csomópontokhoz
    - ha C B-től megismer egy utat, akkor az azt kiterjesztő utat B-nek végtelen költségű útként jelzi

# Link-state protokollok működése

A link-state protokollok működése 2 részből áll:

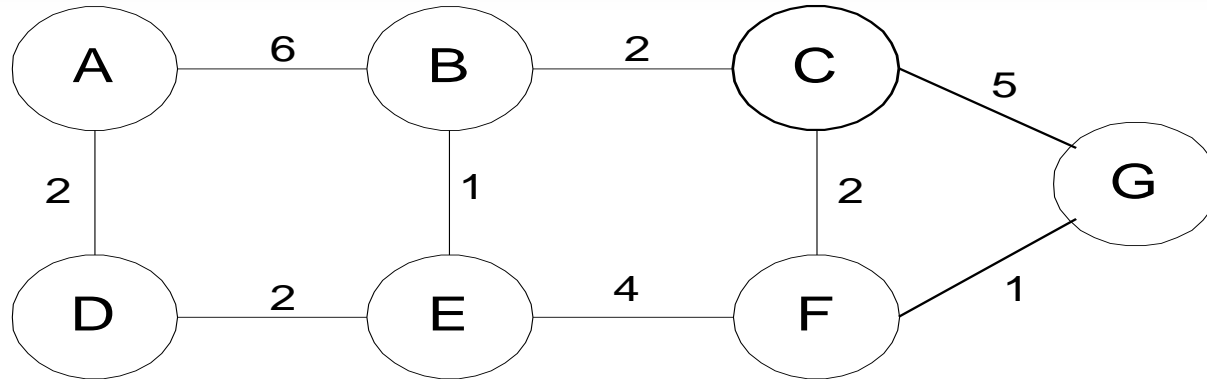
## 1. Minden állomás felderíti a hálózat topológiáját

- Hálózati topológia leírása a link állapot leíró rekordokban
  - Link állapot leíró rekordokat kell terjeszteni

## 2. A kapott gráfban megkeresi a legrövidebb útvonalat és az ahhoz tartozó első állomást

- Fontos!
  - A routerekben lévő topológia megegyezzen
  - Az optimális út kiválasztása ugyanúgy történjen
    - ha A router B felé számolja az optimális utat, B meg A router felé – hurok!

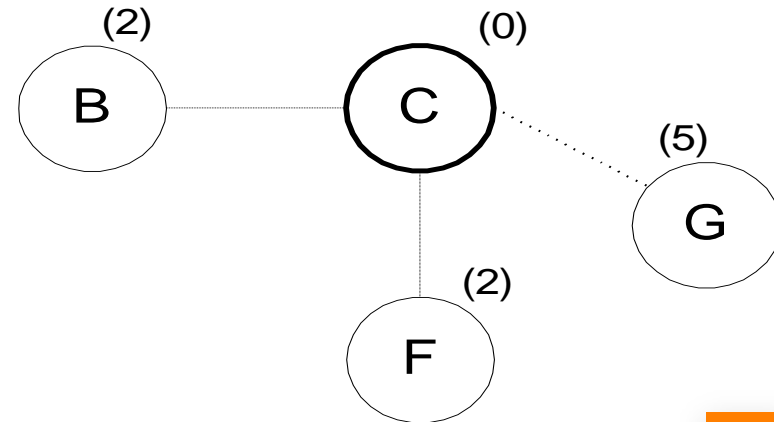
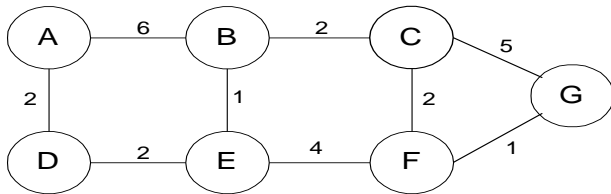
# Link State Database



Link state Database						
A	B	C	D	E	F	G
B/6	A/6	B/2	A/2	B/1	C/2	C/5
D/2	C/2	F/2	E/2	D/2	E/4	F/1
	E/1	G/5		F/4	G/1	

# Dijkstra algoritmus

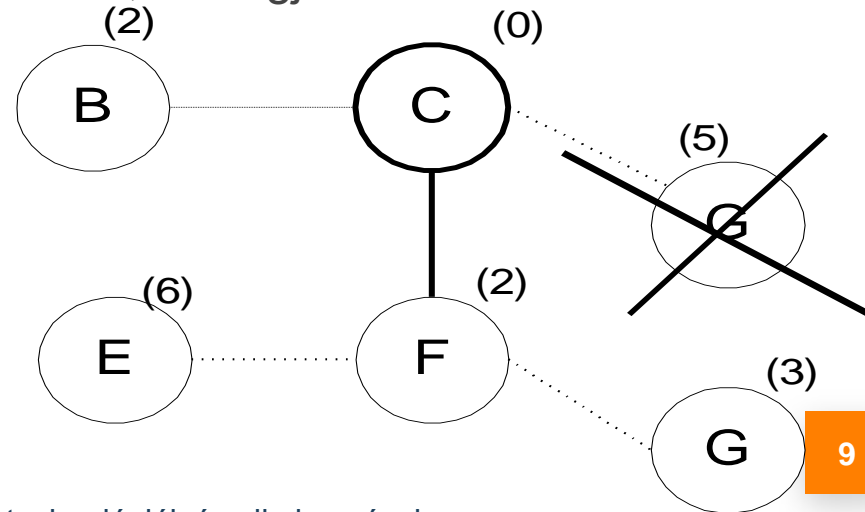
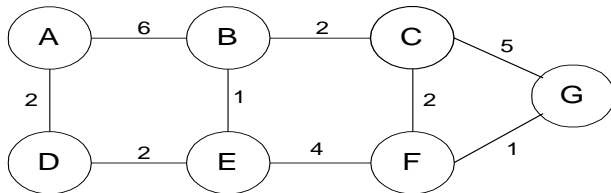
- Az útvonalválasztás Dijkstra algoritmus alapján
  - Legyen C a gyökér
  - Számoljuk ki a szomszédokhoz vezető utak költségét





# Dijkstra algoritmus 2

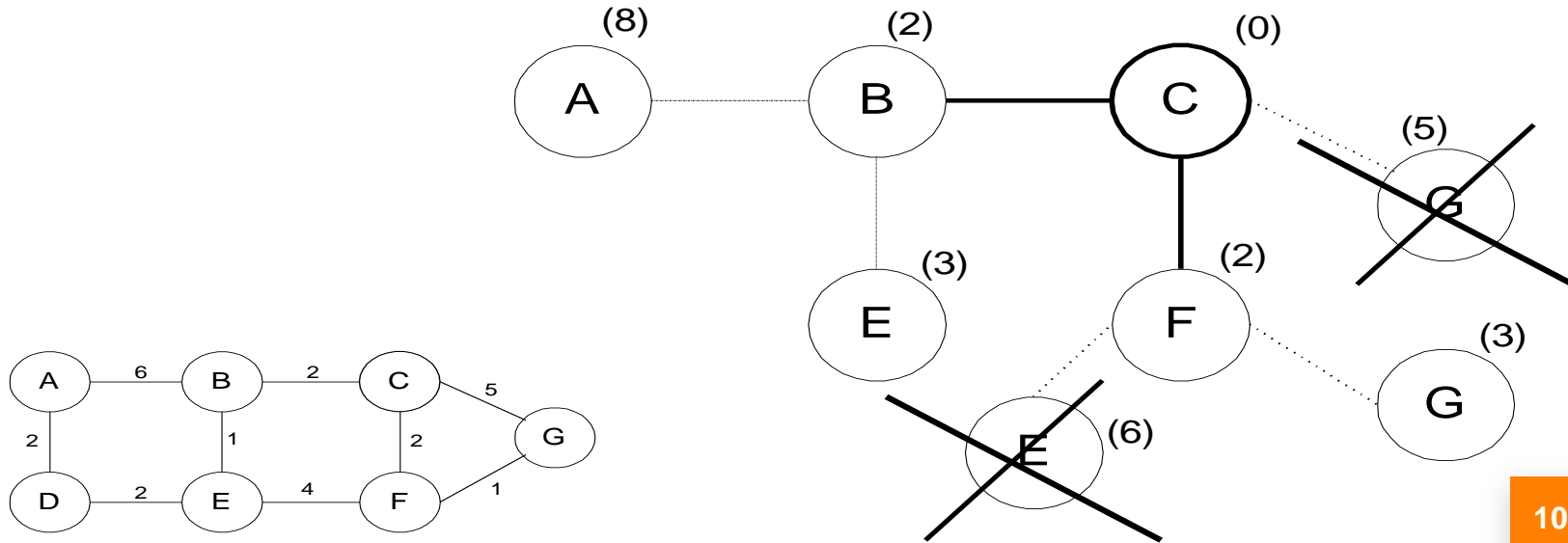
- Vegyük be F-et (legkisebb költségű, még nem elemzett csomópont) és számoljuk ki F szomszédjaihoz vezető utak költségét.
- Rövidebb út G-hez F-en keresztül, E megjelenik



2018. 10. 15.

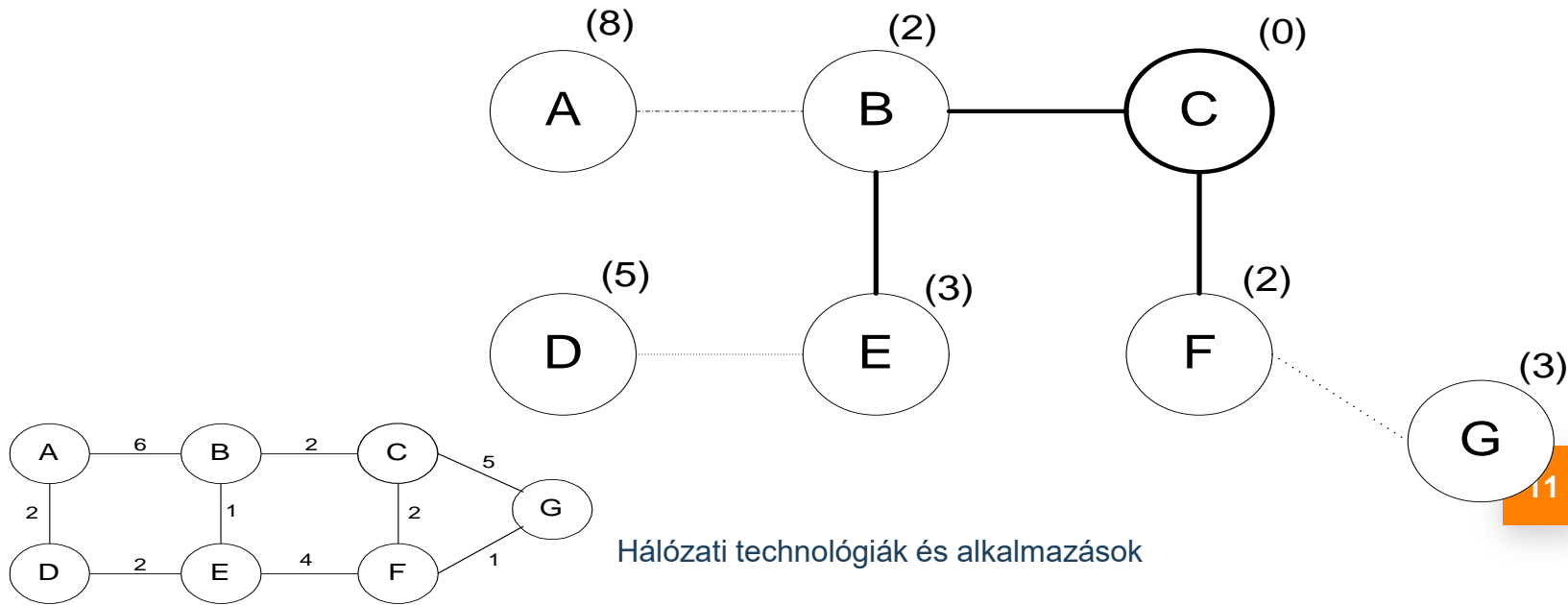
# Dijkstra algoritmus 3

- Vegyük be B-t és számoljuk ki B szomszédainak költségét
- Rövidebb út E-hez B-n keresztül, A megjelenik



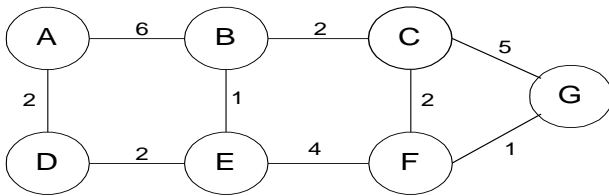
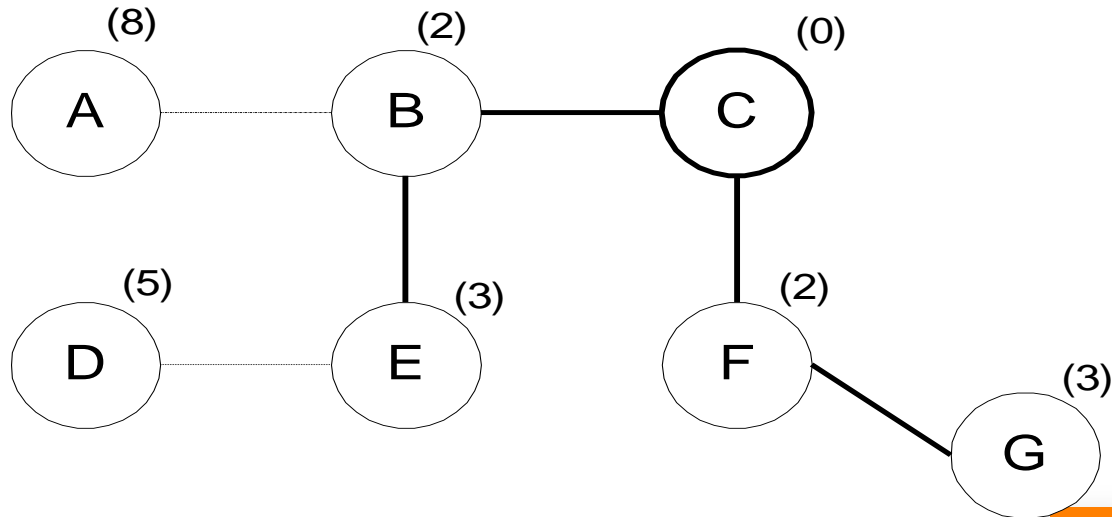
# Dijkstra algoritmus 4

- Vegyük be E-t, és számoljuk ki E szomszédjainak költségét
- Nincs változás, D megjelenik



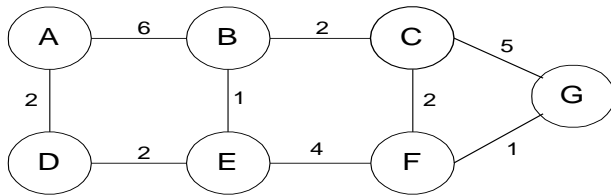
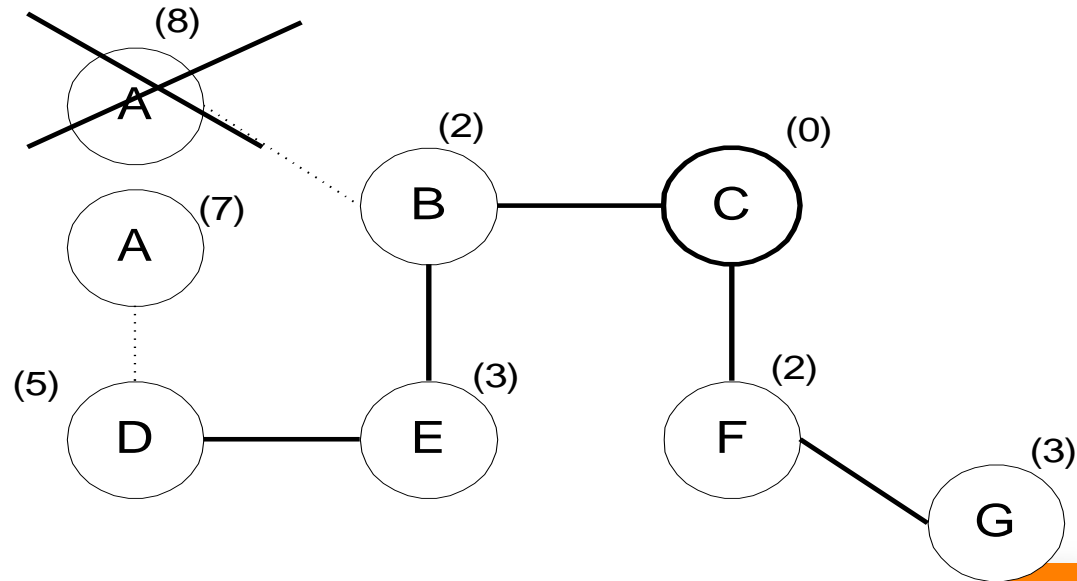
# Dijkstra algoritmus 5

- Vegyük be G-t, és számoljuk ki G szomszédainak költségét
- Nincs változás



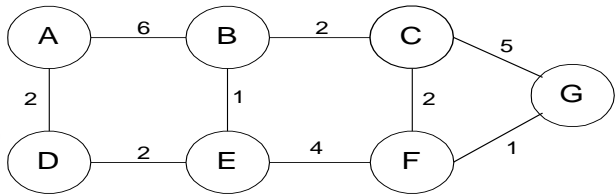
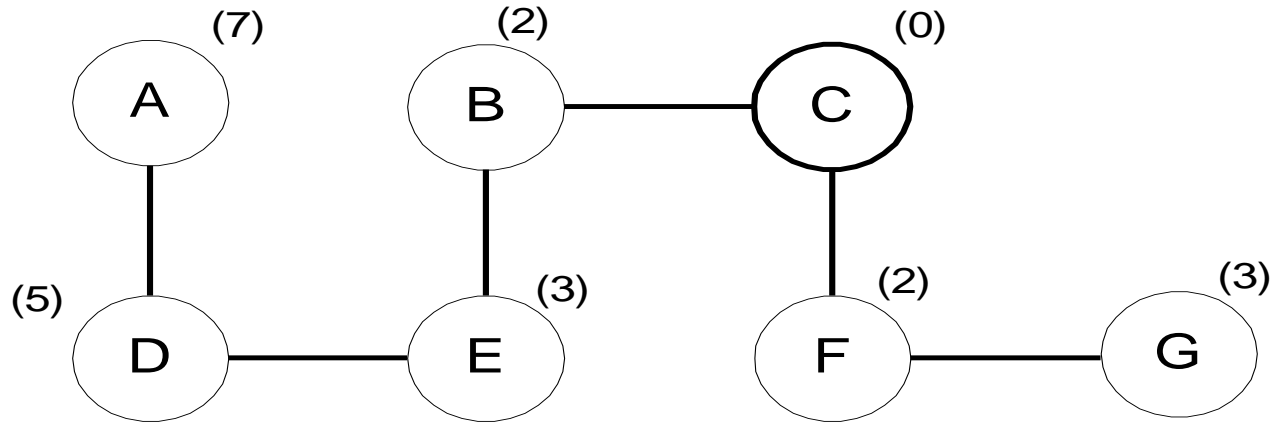
# Dijkstra algoritmus 6

- Vegyük be D-t, és számoljuk ki D szomszédainak költségét
- Rövidebb út A-hoz!



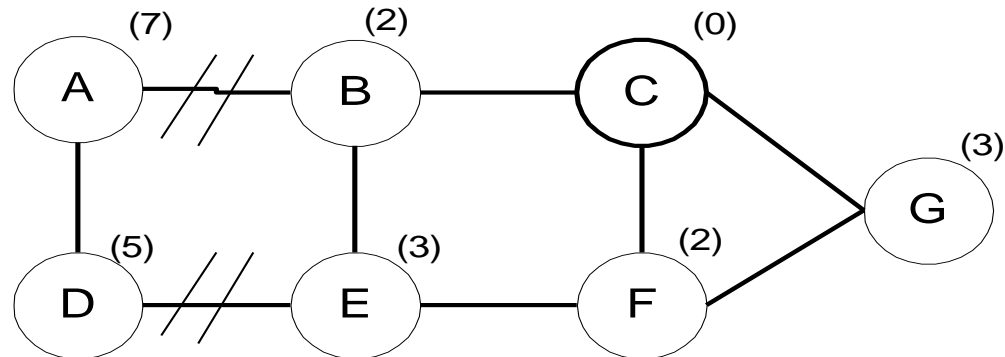
# Dijkstra algoritmus 7

- Vegyük be A-t és számold ki A szomszédjainak költségét
- Nincs több szomszéd
- Befejezés



# Hibás link következménye

- A-B és D-E linkek megsérülnek
  - hálózat kettészakad
  - két rész képtelen értesíteni egymást a változásokról
- A és D elérhetetlennek nyilvánítja a hálózat többi részét
- Hibás link megjavulása után a routerek szinkronizálják adatbázisukat
  - Topológia frissítés



# Link-state protokollok

- **OSPF – Open Shortest Path First**
  - Első szabvány – RFC 1131 ('89)
  - OSPFv2 – RFC 2178 ('97)
  - OSPFv3 – RFC 2740 ('99)
    - IPv6-os verzió