

Hálózati Technológiák és Alkalmazások

Vida Rolland
BME TMIT

2022. október 24.



Áttérés az IPv6-ra

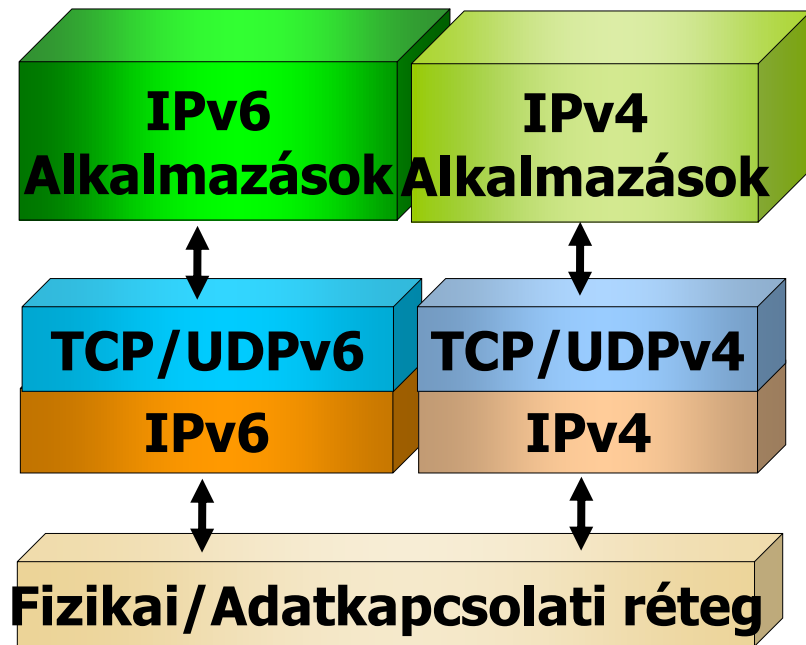
- IP-re épülő hálózati szolgáltatások, útválasztó protokollok
 - RIPv6(ng), OSPFv6 (v3), BGPv6
- IP-re épülő hálózati és szállítási protokollok
 - TCPv6, UDPv6, RSVPv6
- Alkalmazások
 - Minden alkalmazás, mely közvetlenül használta az IPv4-es címeket, nem független az alatta lévő rétegektől, így az IPv6 támogatást implementálni kell
- Fokozatos áttérés
 - Nincs „D-day”
- Elvárások az áttérést illetően
 - Ne legyenek áttérési függőségek
 - egy-egy csomópont áttérése függetlenül történhet
 - legfontosabb szempont a visszafelé kompatibilitás
 - A végfelhasználó számára minél egyszerűbb legyen
 - Az áttérési technikák egymástól függetlenül legyenek alkalmazhatóak
 - legalább a tartományok szintjén

Áttérési megoldások

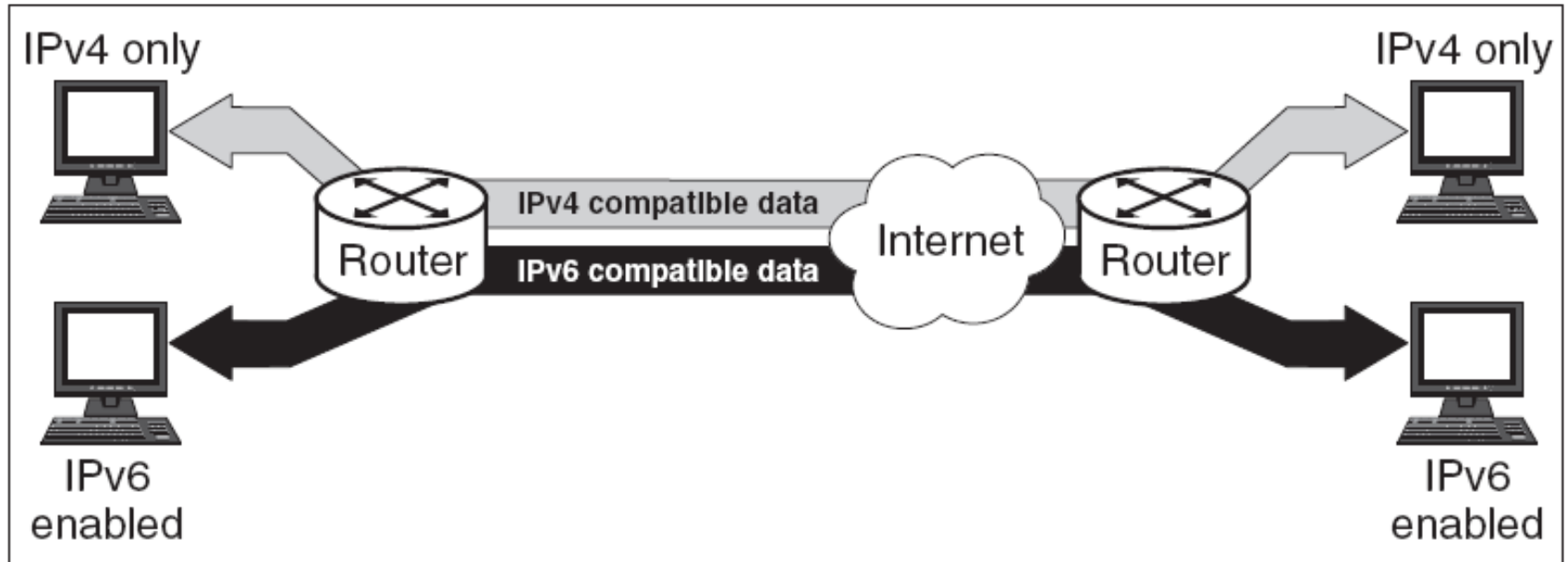
- **Dual Stack** (dupla protokoll verem)
 - IPv4/IPv6 egyszerre ugyanazon az eszközön
- **Alagutak**
 - Kezdetben, IPv6 csomagok alagutazása IPv4 felhőkben
 - Később IPv4 csomagok alagutazása IPv6 felhőkben
- **Protokoll fordítás**
 - Protokoll információkat hordozó fejlécből másik protokoll fejléc létrehozása fordítási szabályok alkalmazásával
 - IPv6 <-> IPv4

Dupla protokoll verem

- Az IPv6 felé tett első lépés olyan rendszerek telepítése, melyek támogatják az IPv6-ot.
 - ezek a rendszerek a kettős stack stratégián alapulnak, amely az IPv4 és IPv6 használatát is támogatja.
- Ezek a rendszerek
 - IPv6-ot használnak más IPv6 rendszerekkel való kommunikációra
 - képesek visszalépni IPv4 módba régi rendszerekkel való párbeszédhez



Dual stack



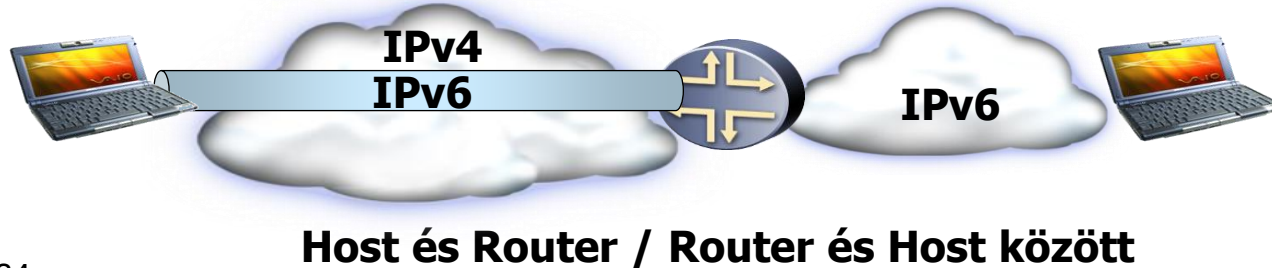
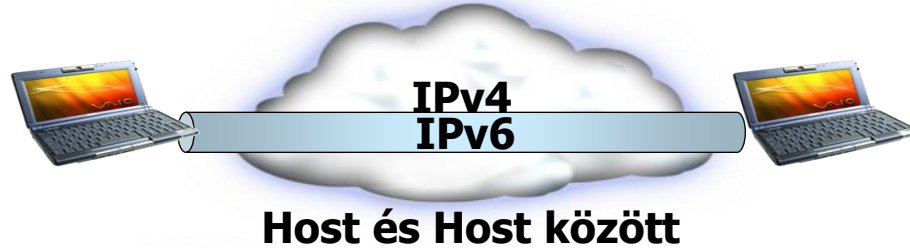
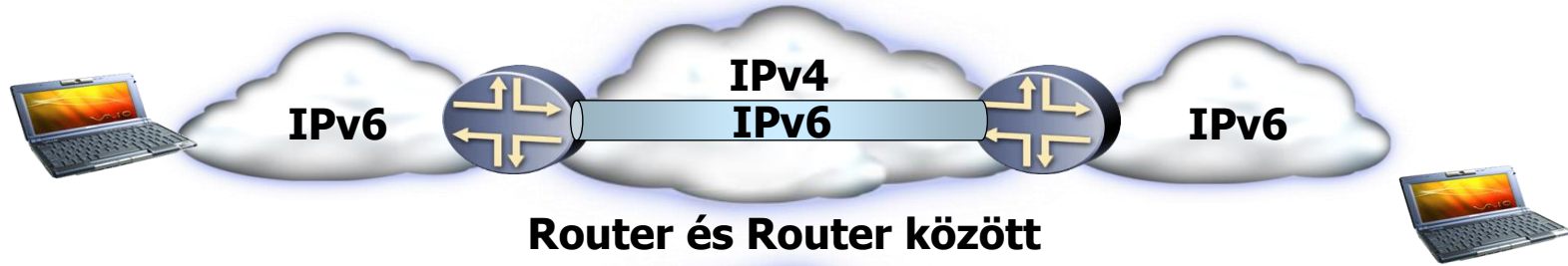
Dupla protokoll verem

- Alkalmazásának előnyei
 - egyszerű installálni, konfigurálni, karbantartani
 - az IPv6 teljes funkcionalitása kihasználható
 - bármely két csomópont tud egymással kommunikálni csak IPv4, vagy csak IPv6-os csomagokkal
 - átlátszó, az áttérés a felhasználók számára észrevétlenül történhet
- Alkalmazásának hátrányai
 - nem skálázható: minden csomópontnak kell rendelkeznie IPv4-es és IPv6-os címmel is, az IPv4-es címtartomány korlátozza a megoldás elterjedését
 - a hálózati útválasztókban megnövekszik az útválasztási tábla mérete
 - nem flexibilis: nincs kommunikációs lehetőség a csak IPv4-es és a csak IPv6-os csomópontok között

Alagutazás

- IPv6 csomag egy IPv4 csomagba foglalva
- Az alagút végpontjai végzik a becsomagolást
- A folyamat „transzparens” a közbeeső csomópontoknak
- **Konfigurált alagutak**
 - Az alagút végpontjait explicit módon konfigurálják
 - A végpontok dual stack csomópontok
- **Automatikus alagutak**
 - Az alagút végpontjait a hálózat automatikusan felfedezi
 - **Alagút ügynökök** (Tunnel Brokers) (RFC3053)
 - **6to4** (RFC3056)
 - **ISATAP** (Intra-Site Automatic Tunnel Addressing Protocol)
 - **6over4** (RFC2529)
 - **Teredo**: IPv4 NAT-on keresztüli alagutak

Alagutazás



Fordítók

- Hálózati szintű fordítók
 - SITT (Stateless IP/ICMP Translator Algorithms) (RFC2765)
 - NAT-PT (Network Address Translator-Protocol Translator) (RFC2766)
 - BIS (Bump int the Stack) (RFC2767)
- Átviteli szintű fordítók
 - TRT (Transport Relay Translator) (RFC3142)
- Alkalmazási szintű fordítók
 - BIA (Bump in the API) (RFC3338)
 - SOCKS64 (RFC3089)
 - ALG (Application Level Gateway)

Routing - Router



- **Routing (útválasztás)**
 - Folyamat, mely során a hálózati protollok csomagjai a célállomáshoz jutnak
 - A routing tábla és a megvalósított protollok szerint a routerek meghatározzák a beérkező csomagok útvonalát
- **Router (útválasztó)**
 - Útválasztást végző csomópont
 - Egymással kommunikálnak
 - A szomszédoktól szerzett információkat gyűjtik és tárolják
 - **Útválasztó táblákat** hoznak létre és tartanak karban
 - Tartalmuk: <célcím, kimenő interfész> párok



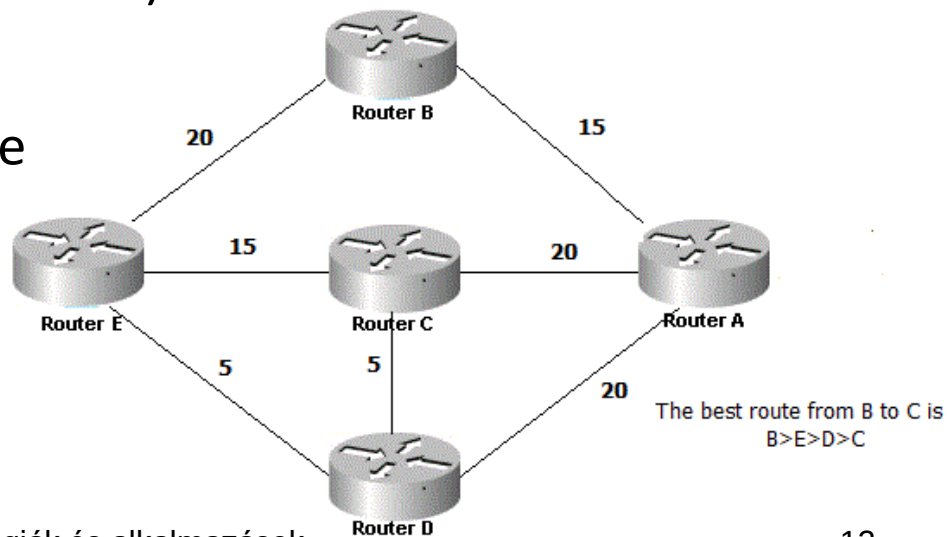
Router

- Router lehet
 - operációs rendszer routing modulja
 - dedikált eszköz (nem csak szoftver, hanem hardver támogatottsággal is rendelkezik) - gyorsabb
 - Cisco, Juniper, Alcatel-Lucent, HP, Huawei, NEC, etc.
- Router kapacitása
 - hány csomagot képes továbbítani időegység alatt (packet/s - PPS)
 - Pl. HP 8800 router – 864 Mpps (2012)



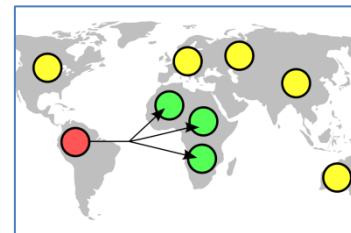
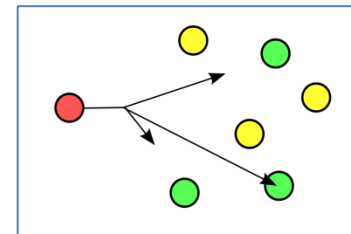
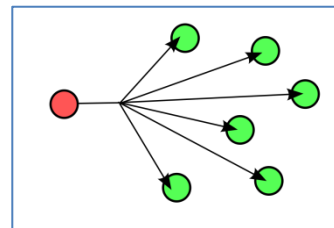
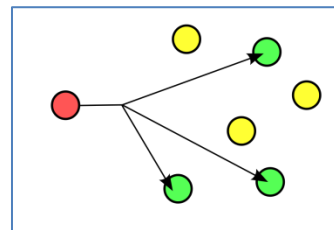
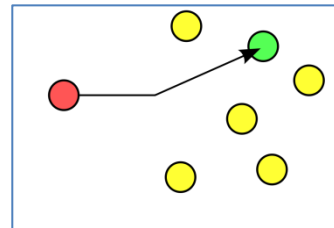
Routerek feladata

- Az optimális útvonal kiválasztása az adott csomag számára
- Alábbi szempontok (metrikák) szerint:
 - az út hossza (hány linken vezet át)
 - költség
 - az adott útvonal terheltsége
 - sávszélesség
 - megbízhatóság
 - késleltetés



Útválasztási szemantika

- **Unicast** – csomag küldése egy adott célcsomópontnak
- **Anycast** – csomag küldése bárkinek (pl. a legközelebbi csomópontnak) egy adott csoportból
- **Multicast** – csomag küldése egy csoportnak
- **Geocast** – csomag küldése egy adott földrajzi területre
- **Broadcast** – csomag küldése minden csomópontnak a hálózatban



Routing protokollok osztályozása

- **Statikus:**
 - a routing tábla manuális kitöltése
 - automatikusan soha nem frissítődik
- **Dinamikus:**
 - a routerek egymás között kommunikálva a hálózat topológiájának megfelelően állítják elő az útvonalválasztó táblát
- **Egyutas:**
 - minden célpont felé csak egy utat tárol
- **Többutas:**
 - minden célpont felé több (esetleg minden) utat tárol.
 - Ezek a protokollok képesek load balancing-ra (terhelés megosztás)

Routing protokollok osztályozása

- **Lapos (flat):**
 - minden router minden célpontról tud
 - Régebben (kisebb hálózatok)
- **Hierarchikus:**
 - a router-ek nem minden célpont felé ismerik az utat
 - egy ismeretlen címzettnek szánt csomagot egy előre meghatározott irányba (default route) küldenek
 - ez routing információk egy szélesebb körével rendelkezik
 - routing táblák mérete kezelhető marad
- **Intra-domain**
 - valamely területen (domain) belüli útvonalválasztásért felelős
- **Inter-domain**
 - a területek (domain) közötti útvonalválasztásért felelős

Routing protokollok osztályozása

- **Hop-by-hop:**
 - minden router autonóm módon határozza meg a továbbítás irányát
 - ezen elven működő routerek csak olyan utakat hirdetnek (szomszédjaiknak), melyeket maguk is használnak
- **Source routing:**
 - a feladó határozza meg az útvonalat (pl. IP fejléc)
 - a routerek csupán az elérhetőségi információkat terjesztik
 - magukat a csomagokat a csomagba beleírt útvonal szerint kapcsolják
- A két megoldás között léteznek átmenetek

Routing protokollok osztályozása

- **Távolság vektor (distance vector) protokollok**
 - csak a szomszédos routerek kommunikálnak
 - minden router elmondja összes szomszédjának:
 - mekkora költségű utat ismer egy adott célponthoz
 - arról nem szól, hogy az út merre vezet
 - a routerek begyűjtik szomszédaiktól ezeket a hirdetéseket és kiválasztják, hogy ki hirdette a legolcsóbb utat az adott célpontokhoz
 - a megfelelő csomagokat a legkedvezőbb irányba továbbítják
 - saját költségüket a legkedvezőbbekhez hozzáadva ők is hirdetik az adott célponthoz vezető utat

Klasszikus Bellman-Ford algoritmus

$d_{ij} := i$ - j link költsége (végtelen, ha nincs link)

- Tényleges ár, késleltetés, stb.

Következmény: **additivitás**

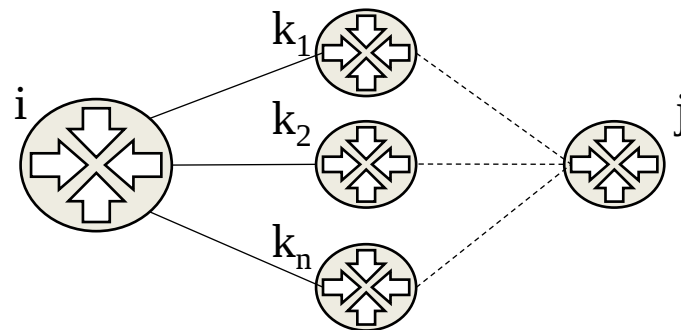
- egy útvonal költsége az azt alkotó linkek költségének összege

$D_{ij} :=$ minimum költség i és j között

Bellman egyenlet:

$D_{ii} = 0$, minden i -re

$D_{ij} = \min_k \{d_{ik} + D_{kj}\}$



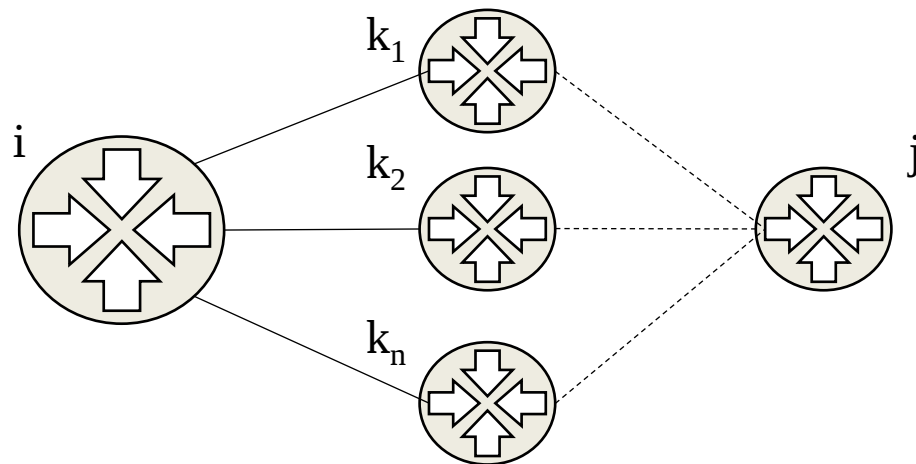
Elosztott Bellman-Ford Algoritmus

$D_{kj}^i(t)$ = minimális távolság k -tól j -ig, melyet i router lát a t időpillanatban

$D_{ii} = 0$, minden i -re

$D_{ij}(t) = \min_k \{d_{ik} + D_{kj}^i(t)\}$

- önállóan működhet az algoritmus a routerekben



Distance-vector protokollok

- RIPv1 (RFC 1058, '88)
 - Routing Information Protocol
- RIPv2 (RFC 2453, '98)
- RIPv6 (RFC 2080, '97)
 - IPv6-os verzió
- EIGRP
 - Enhanced Interior Gateway Routing Protocol
 - Cisco proprietary szabvány

Distance Vector protokollok

- Távolságvektorokat tárol az útvonalakról
 - Adathármasok:
 - Cél (mi a célállomás)
 - Költség
 - Következő csomópont (merre küldje)
 - Rendszeresen frissítik adataikat a közvetlen szomszédok
 - Frissítő üzenet (2 részből áll):
 - Cél, költség
 - Két ok miatt kellhet frissíteni a bejegyzéseket:
 - Kisebb költségű utat talál
 - Szomszédhoz kapcsolódó link 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
 - 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

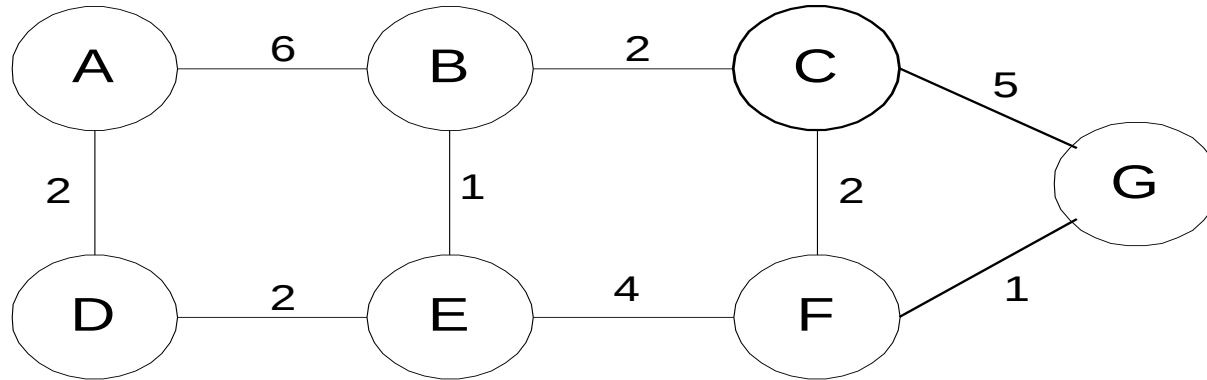
Routing protokollok osztályozása

- **Kapcsolat állapot (link state) protokollok**
 1. feltérképezik a hálózat topológiai gráfját
 2. ebben a gráfban keresik a legrövidebb utat.
 - A routerek egymás között csak saját interfészeik állapotát beszélik meg
 - Kik a szomszédjaim, milyen költségük van a linkeknek közöttünk
 - Ezeket az információkat minden, a hálózatban lévő routerrel kicserélik
 - Ebből építi fel mindenki a saját (de egymással megegyező) topológiai gráfját

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 található
 - Link állapot leíró rekordokat kell terjeszteni (Link State Advertisements)
 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!

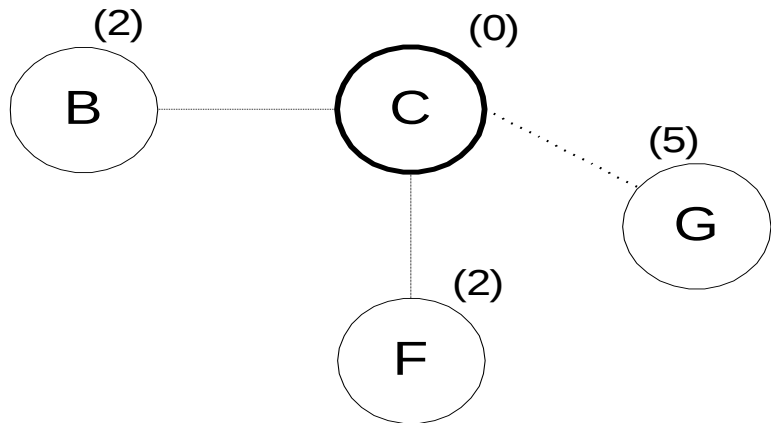
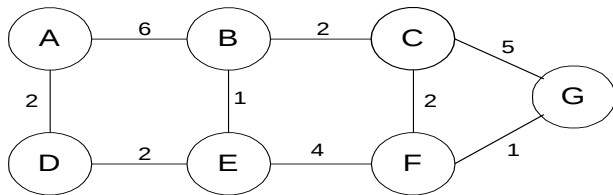
Link State Database



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	

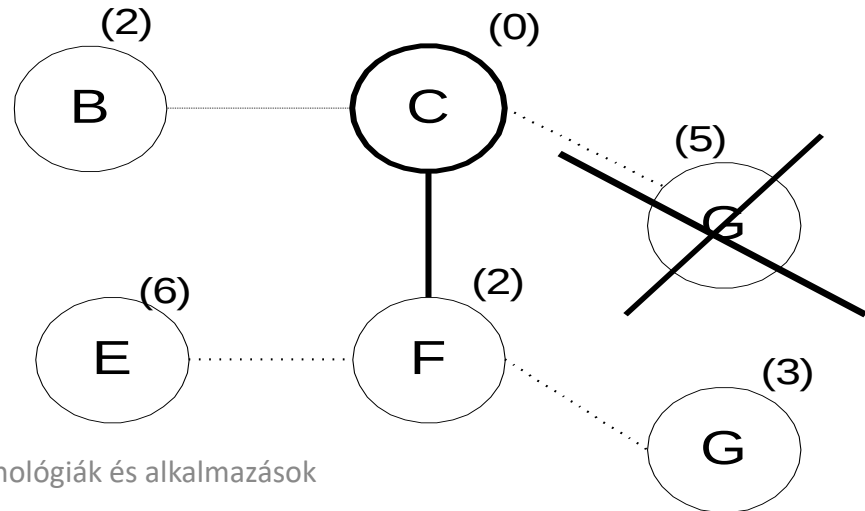
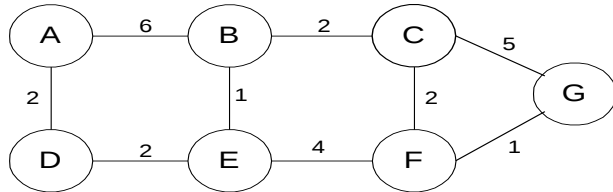
Dijkstra algoritmus

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



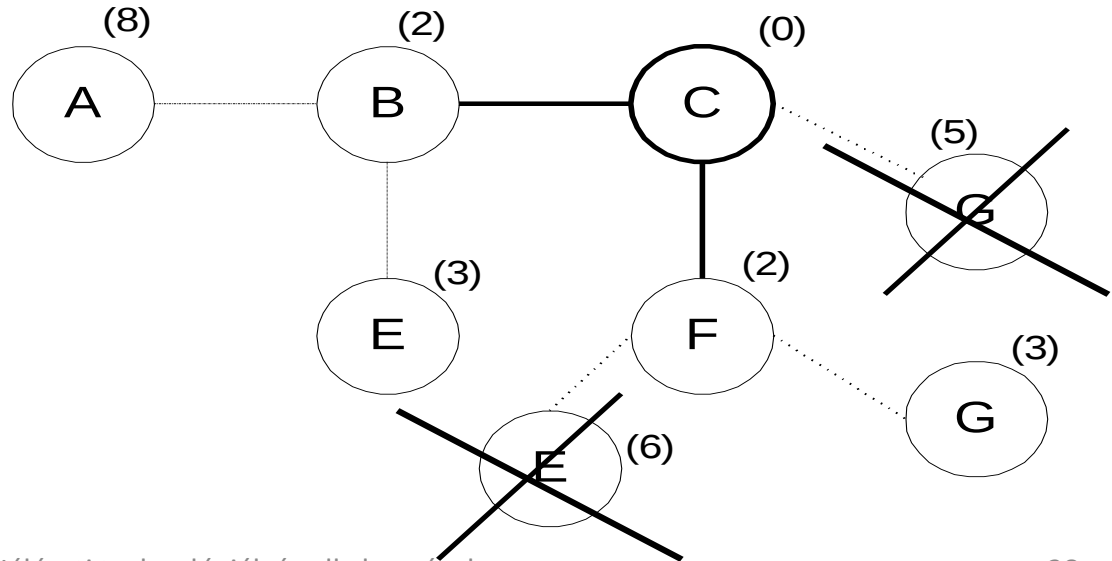
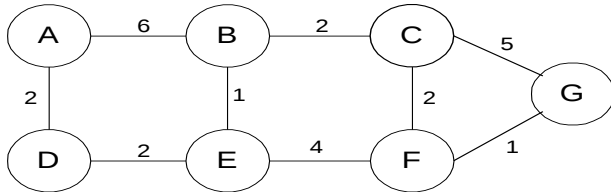
Dijkstra algoritmus 2

- 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



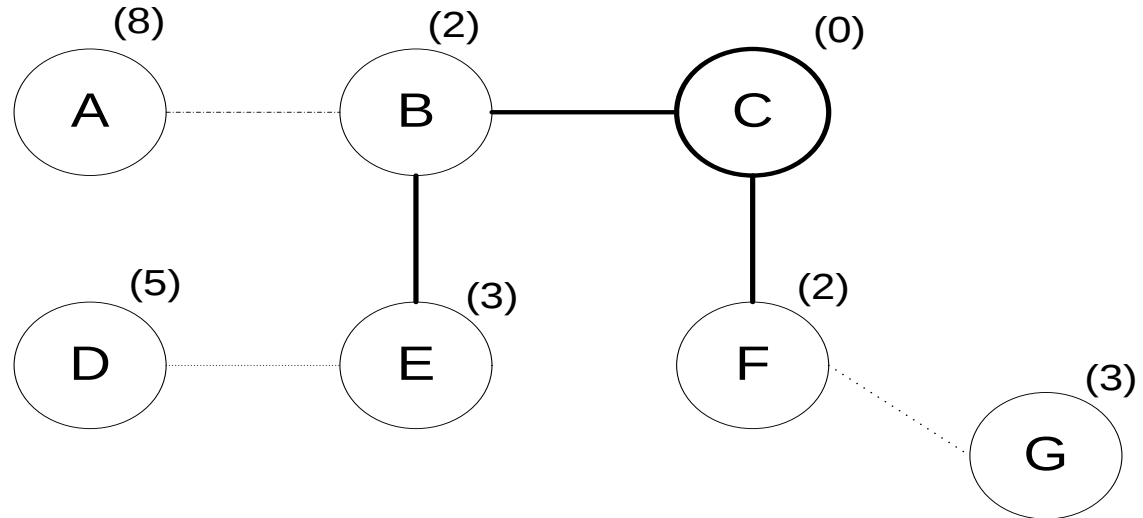
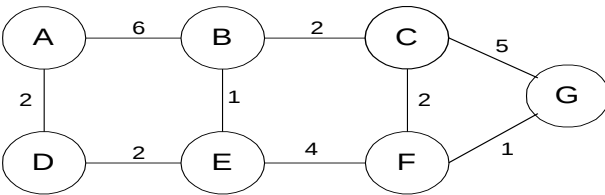
Dijkstra algoritmus 3

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



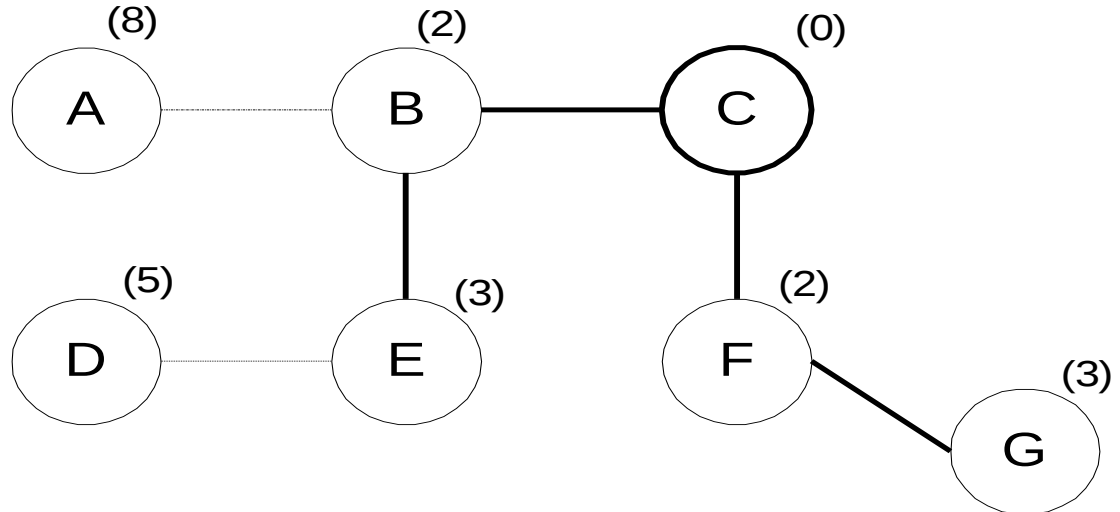
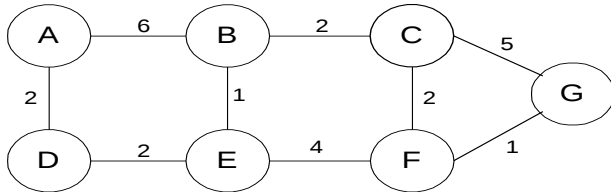
Dijkstra algoritmus 4

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



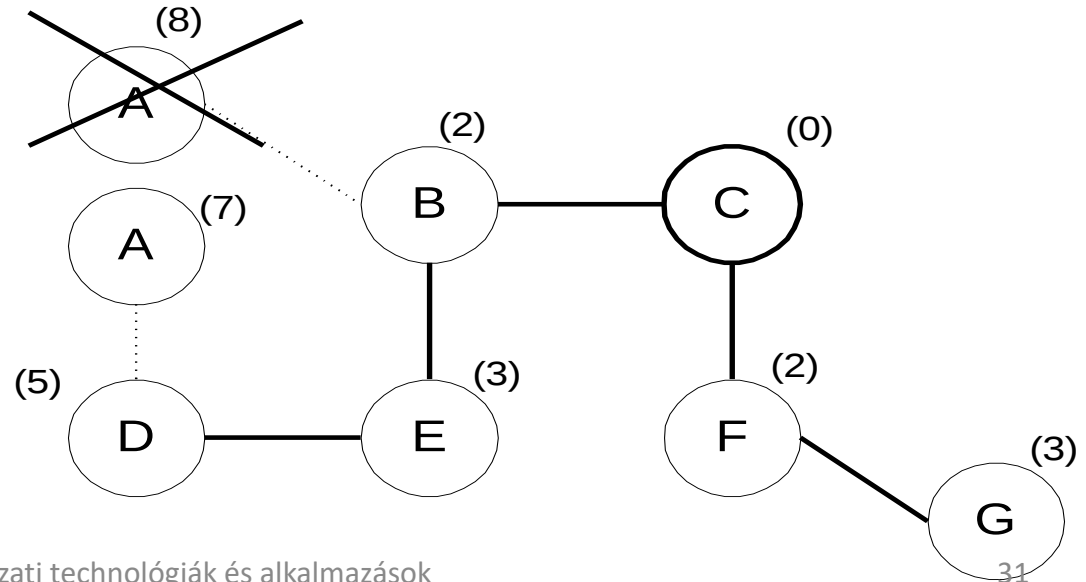
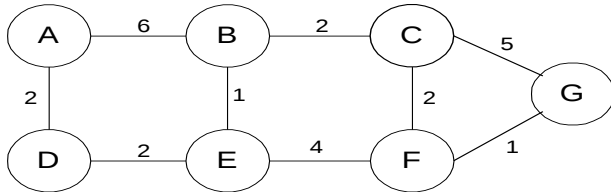
Dijkstra algoritmus 5

- 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

