

# Hálózatok építése és üzemeltetése

## Routing protokollok

# Routing protokollok

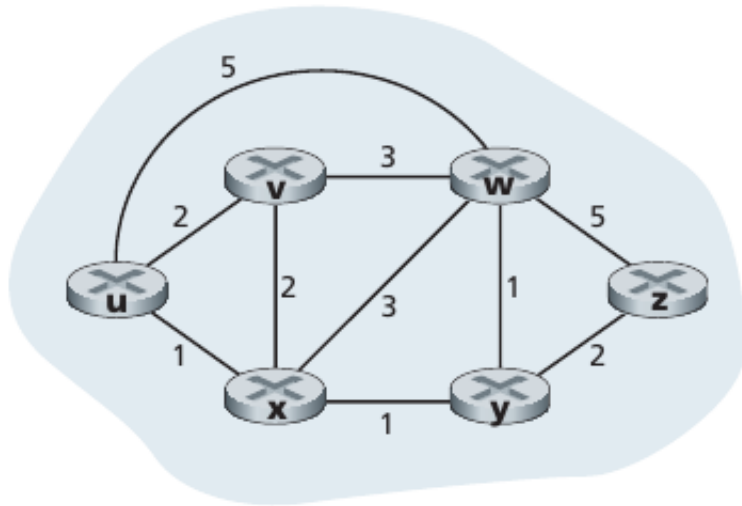
# Routing protokollok

---

## ▶ Feladatuk

- ▶ optimális útvonal biztosítása bármely csomópontok között
  - ▶ aktuális állapot információ a hálózatról
  - ▶ útvonalak kalkulálása
- ▶ forwarding táblák
  - ▶ konfigurálása
  - ▶ dinamikus karbantartása, frissítése
  - ▶ bejövő routing protokoll üzenetek alapján
- ▶ routing információk
  - ▶ feldolgozása
  - ▶ terjesztése

# Routing protokollok



- ▶ **Hálózat: absztrakt gráf**
  - ▶ csomópontok: routerek
  - ▶ élek: linkek
  - ▶ élköltség: valamilyen metrika (pl. késleltetés, sávszélesség kifejezése)
- ▶ **cél:**
  - ▶ (valamilyen értelemben) optimális, legkisebb költségű útvonal meghatározása két csomópont között
    - ▶ pl. legrövidebb út
- ▶ **Ismerős algoritmusok:**
  - ▶ Dijkstra algoritmus
  - ▶ Bellman-Ford algoritmus

# Csoportosításuk

---

- ▶ **Globális vs. Elosztott**
  - ▶ globális: minden router ismeri a teljes topológiát
  - ▶ elosztott: minden router csak a szomszédjait és a tőlük kapott üzeneteket ismeri
- ▶ **Intra-domain vs. Inter-domain**
  - ▶ intra: Interior Gateway Protocol (IGP)
    - ▶ közös adminisztratív domain
    - ▶ nem jól skálázódik
  - ▶ inter: Exterior Gateway Protocol (EGP)
    - ▶ külön adminisztratív domaineik, AS-ek (Autonomous System) között
    - ▶ jól skálázódik (internet)
- ▶ **Link state vs. Distance Vector (ld. később)**

# Csoportosításuk

---

- ▶ Interior Gateway Protocol (IGP) példák
  - ▶ OSPF (OpenShortest Path First)
  - ▶ IS-IS (Intermediate System to Intermediate System)
  - ▶ RIP (Routing Information Protocol)
  - ▶ EIGRP (Enhanced Interior Gateway Routing Protocol)
- ▶ Exterior Gateway Protocol (EGP)
  - ▶ BGP (Border Gateway Protocol)
  - ▶ Id. MSc (Internet architektúra és szolgáltatások főspecializáció)

# Routing protokollok értékelése

---

- ▶ **Erőforrás-igény**
  - ▶ milyen mértékben terheli a routing processzort
  - ▶ mennyi sávszélességet foglal (kontroll üzenetek)
- ▶ **Stabilitás, konvergenciaidő**
  - ▶ hálózati állapot változása után milyen gyorsan alakul ki (kialakul egyáltalán?) új stabil, konzisztens állapot
- ▶ **Biztonság**
  - ▶ támadási pont lehet a protokollok által használt portok és egyéb erőforrások lefoglalása, gátolhatja a szinkronizációt
  - ▶ hamis információ terjesztése, pl. hosszabb időre kivonhat egyes hálózati szegmenseket a forgalomból
- ▶ **Címzés**
  - ▶ melyik címzési technikát támogatja (classful, CIDR, VLSM)
- ▶ **Hálózati hierarchia**
  - ▶ hálózat a címzés és az útvonalválasztás szempontjából lehet lapos (flat) vagy hierarchikus
  - ▶ hierarchikus
    - ▶ elkülönülő területek, routereknek csak az adott területen belüli eszközöket kell ismerniük
    - ▶ routing és forwarding táblázatok mérete jelentősen csökkenthető

# Link State alapú routing



# Működési elv

---

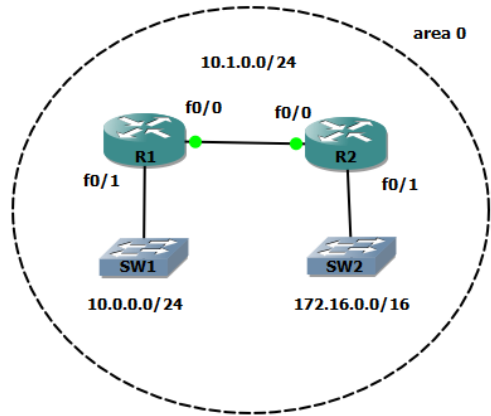
- ▶ Link State alapú routing algoritmus
  - ▶ globális nézetben dolgozik
  - ▶ LSP: Link State Packet (id, costs, seq.no, ttl)
  - ▶ egy router
    - ▶ mindenkinek küld LSP-t (broadcast)
    - ▶ a közvetlenül kapcsolódó linkjeiről
    - ▶ periodikusan újra generálja (seq.no++)
    - ▶ legfrissebb beérkezett LSP-eket tárolja
  - ▶ mindenki ugyanazt a topológiát látja
  - ▶ azon számolja az útvonalakat
  - ▶ útvonalszámítás: Dijkstra algoritmus

# Például: OSPF

---

- ▶ Open Shortest Path First (v2)
- ▶ nyílt, IETF szabvány
  - ▶ v2: RFC 2328
  - ▶ IP felett
- ▶ együttműködés különböző gyártók termékei között!
- ▶ korlátozott erőforrás igény
- ▶ viszonylag gyors, automatikus konvergencia topológia változásokra
- ▶ támogatja
  - ▶ különböző útvonal költségek számítását
  - ▶ hierarchikus, többszintű topológiát
  - ▶ alkalmazás típusára alapozott forgalomirányítást
  - ▶ autentikációt minden üzenetre

# Például: OSPF



- ▶ Egyszerű példa
- ▶ R1, R2 routerek
- ▶ mindkét routerben egységes képet szeretnénk
- ▶ példa konfiguráció (Cisco)
  - ▶ hirdetés: network parancs
  - ▶ egy terület: area 0

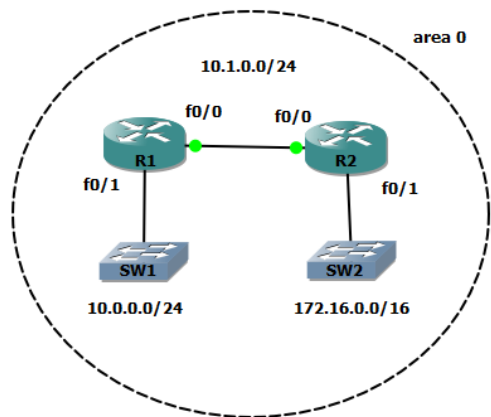
R1:

```
router ospf 1
log-adjacency-changes
network 10.0.0.1 0.0.0.0 area 0
network 10.1.0.0 0.0.0.255 area 0
```

R2:

```
router ospf 1
log-adjacency-changes
network 10.1.0.2 0.0.0.0 area 0
network 172.16.0.0 0.0.255.255 area 0
```

# Például: OSPF



- ▶ Egyszerű példa
- ▶ R1, R2 routerek
- ▶ mindkét routerben egységes képet szeretnénk
- ▶ példa konfiguráció (Cisco)
  - ▶ routing táblák dinamikusan frissülnek

R1:

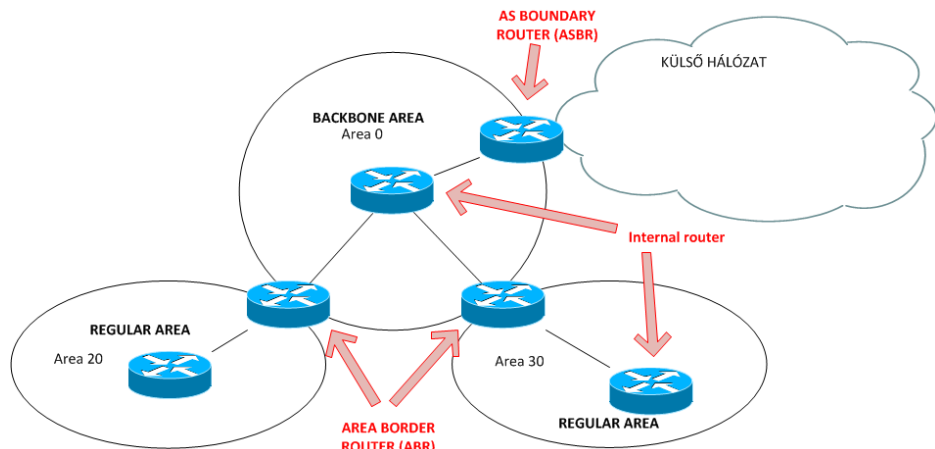
```
R1# show ip route
```

```
O    172.16.0.0/16 [110/20] via 10.1.0.2, 00:02:40, FastEthernet0/0
     10.0.0.0/24 is subnetted, 2 subnets
```

```
C    10.0.0.0 is directly connected, FastEthernet0/1
```

```
C    10.1.0.0 is directly connected, FastEthernet0/0
```

# Például: OSPF



## ▶ Router típusok

- ▶ Internal router
  - ▶ minden OSPF-et futtató interfésze ugyanazon area-ban
- ▶ ABR (Area Border Router)
  - ▶ két különböző területhez is kapcsolódik
  - ▶ legalább egy interfésze a backbone area-hoz
  - ▶ legalább egy interfésze egy regular area-hoz
- ▶ ASBR (Autonomous System Boundary Router)
  - ▶ OSPF AS-t összeköti a külvilággal, és egyben ABR is

- ▶ Hierarchikus routing
- ▶ egy autonóm rendszeren belül területek (area) (skálázódás!)
  - ▶ 32 bites id (pl. 0.0.0.23)
  - ▶ területen belül közös nézet
  - ▶ szinkronizálni kell
- ▶ area típusok
  - ▶ backbone area
    - ▶ id 0
    - ▶ minden más terület ide csatlakozik
  - ▶ regular area
    - ▶ backbone area-hoz csatlakozik
    - ▶ csak a backbone area-val osztja meg a saját területéről származó információkat
    - ▶ minden más regular area-ba menő forgalom a backbone area-n megy keresztül

# Például: OSPF

---

## ▶ Neighbor tabla

- ▶ szomszédsági kapcsolatok
- ▶ szomszéd routerek felderítése
- ▶ Hello protokoll
  - ▶ periodikus Hello üzenetek
  - ▶ multicast 224.0.0.5
  - ▶ folyamatos információ csere

## ▶ LSDB

- ▶ Link State Database
- ▶ Hello üzenet után LSDB szinkronizáció
  - ▶ LSDB kivonat kicserélése
  - ▶ útvonalak vizsgálata
  - ▶ további adatok kérése azokról, amelyekről még nincs információjuk
  - ▶ ha LSDB szinkron: teljes értékű szomszédság

## ▶ Routing tábla

- ▶ legjobb útvonalak tárolása

# Például: OSPF

- ▶ **Útvonal információk cseréje**
  - ▶ LSU (Link State Update) üzenetek
    - ▶ ebben LSA-k (Link State Advertisement)
    - ▶ információ egyetlen útvonalról
    - ▶ area-n belül biztonságos elárasztás
    - ▶ nyugtázással
  - ▶ Különböző típusú LSA-k
    - ▶ meddig jutnak el, hogyan konvertálódnak
    - ▶ Type 1 - Router link advertisement
    - ▶ Type 2 - Network link advertisement
    - ▶ Type 3/4 - Summary link advertisement
    - ▶ Type 5 - External link
    - ▶ Type 7 - NSSA external route
- ▶ **Link költség**
  - ▶ sávszélesség alapján
  - ▶ fordítottan arányos vele
  - ▶ minél kisebb a költség (nagyobb a sebesség), annál preferáltabb a link
- ▶ **Üzenet típusok**
  - ▶ Hello
  - ▶ DBD (Database Description)
  - ▶ LSU (Link State Update)
  - ▶ LSR (Link State Request)
  - ▶ LSAck (Link State Acknowledgement)

# Például: OSPF

---

- ▶ Jön részletesebben a következő gyakorlaton...



# Distance Vector alapú routing

# Működési elv

---

- ▶ Távolságvektor (distance vector) alapú routing algoritmus
  - ▶ lokális információkkal dolgozik
  - ▶ elosztott
    - ▶ minden router információt kap a közvetlen kapcsolódó szomszédaitól
    - ▶ kalkulációt végez
    - ▶ eredményt elküldi a szomszédainak
    - ▶ (destination, cost)
  - ▶ iteratív
    - ▶ addig folytatódik, amíg van cserélendő információ
    - ▶ nincs leállítási jelzés: self-terminating
  - ▶ aszinkron
    - ▶ nincs szükség a routerek szinkron kommunikációjára
  - ▶ routing tábla
    - ▶ (destination, cost, next\_hop)
    - ▶ ha jobb utat kap, frissíti
  - ▶ Útvonal számítás: Bellman-Ford algoritmus

# Például: RIP

---

- ▶ Routing Information Protocol (v2)
- ▶ Első, egyszerű DV implementáció
  - ▶ RFC 2453
  - ▶ UDP felett (520-as port)
- ▶ Cél: minden elérhető hálózathoz a legjobb, hurokmentes útvonal meghatározása
- ▶ ha egy router útvonalfrissítést kap
  - ▶ routing tábla frissítése
  - ▶ kapott mértéket eggyel megnöveli
  - ▶ next hop: a frissítés forrás interfésze
- ▶ Minden cél felé csak a
  - ▶ legjobb útvonal nyilvántartása
  - ▶ de azonos költségű útvonalból többet is kezel
- ▶ Limitált hálózatméret
  - ▶ maximum 15 hop (16 ==  $\infty$ )

# Például: RIP

## ▶ Frissítések

- ▶ v1: broadcast
- ▶ v2: multicast (224.0.0.9)
- ▶ rendszeres
  - ▶ eredetileg 30 másodpercenként
- ▶ eseményvezérelt
  - ▶ ha routing tábla frissítés történt
  - ▶ azonnal megkezdji a frissítés küldést

## ▶ Konvergencia

- ▶ mennyi idő alatt jut a teljes rendszer új konzisztens állapotba
- ▶ DV protokolloknál probléma lehet a lassú konvergencia
- ▶ időzítők használata

## ▶ Időzítők

- ▶ Update Timer
  - ▶ minden útvonalhoz
  - ▶ default 30sec
- ▶ Invalid Timer
  - ▶ ha eddig nem jön update az útvonalra:  
next\_hop=16
  - ▶ default: 180sec
- ▶ Flush Timer
  - ▶ ha eddig nem jön update az útvonalra, törlődik
  - ▶ default: 240sec
- ▶ Hold-down Timer
  - ▶ ha változott egy útvonal, eddig nem fogad el új változtatásokat
  - ▶ default: 180sec
  - ▶ megelőzhető a végtelenig számolás, de nő a konvergencia idő!

# Például: RIP

---

- ▶ Hurok elkerülés
- ▶ Végtelenig számolás
  - ▶ 15 hopnál távolabb lévő célállomásokat a rendszer elérhetetlennek nyilvánítja
  - ▶ így a végtelenig számolás problémája, csomagok végtelen körbeutazása megelőzhető
  - ▶ DE nagyméretű hálózatokban csak korlátozottan alkalmazható
- ▶ Split Horizon (látóhatár megosztás)
  - ▶ Alapelv: egy útvonalról nem érdemes információkat küldeni oda, ahonnan a vele kapcsolatos adatok eredetileg érkeztek
- ▶ Poison Reverse (visszairányú mérgezés)
  - ▶ Ha egy útvonal elérhetetlenné válik, a router küld egy frissítést azon az interfészen, amelyen keresztül korábban megtanulta az adott útvonalat
  - ▶ a frissítésben az adott prefixhez tartozó költséget végtelenre (16) állítja
- ▶ Hold-down Timer (visszatartási időzítő) és Event Triggered Updates (eseményvezérelt frissítések)
  - ▶ ld. korábban

# Például: RIP

---

- ▶ Jön részletesen a következő félévben szakirány laboron...

# Összefoglalás

---

- ▶ Forwarding vs. Routing
- ▶ Routing protokollok
- ▶ Link State alapú algoritmusok
  - ▶ pl. OSPF
- ▶ Distance Vector alapú algoritmusok
  - ▶ pl. RIP