

# Hálózatok építése és üzemeltetése

## Routing protokollok

# Mai téma

---

- ▶ Eddig
  - ▶ hálózati funkciók (NAT, Firewall, DHCP, DNS)
- ▶ Tulajdonképpen
  - ▶ switch / bridge (Layer 2)
  - ▶ router (Layer 3)
  - ▶ is alap hálózati funkciók
- ▶ Mai téma IP
  - ▶ forwarding
  - ▶ routing

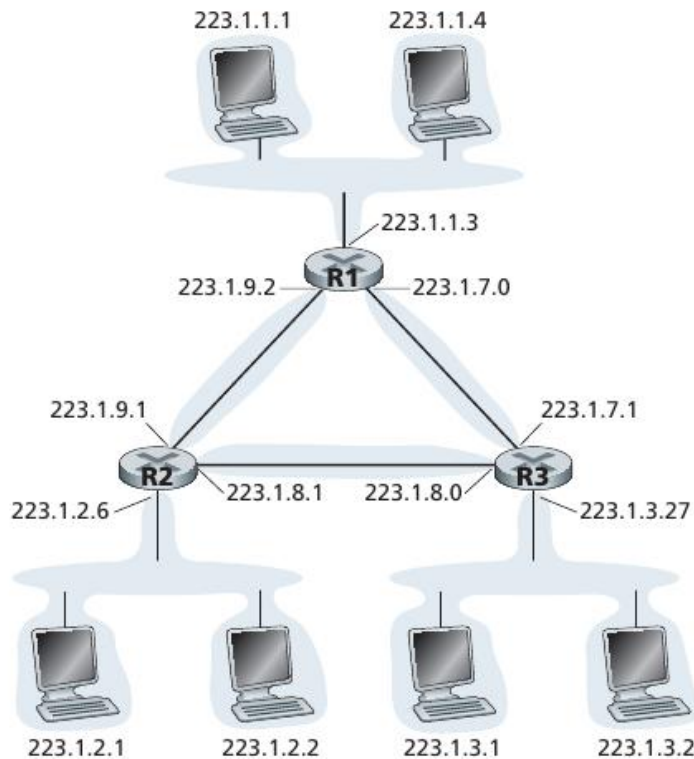
# Alapok

# Hálózati réteg

---

- ▶ Feladata
  - ▶ csomag eljuttatása végponttól végpontig
  - ▶ hálózatba kapcsolt routereken keresztül
- ▶ Ezt a szolgáltatást használja a transzport réteg (TCP, UDP)
- ▶ Alap funkciók
  - ▶ forwarding
    - ▶ mit csináljon egy router egy bejövő csomaggal
  - ▶ routing
    - ▶ milyen úton (routereken) haladjon a csomag a forrástól a célig

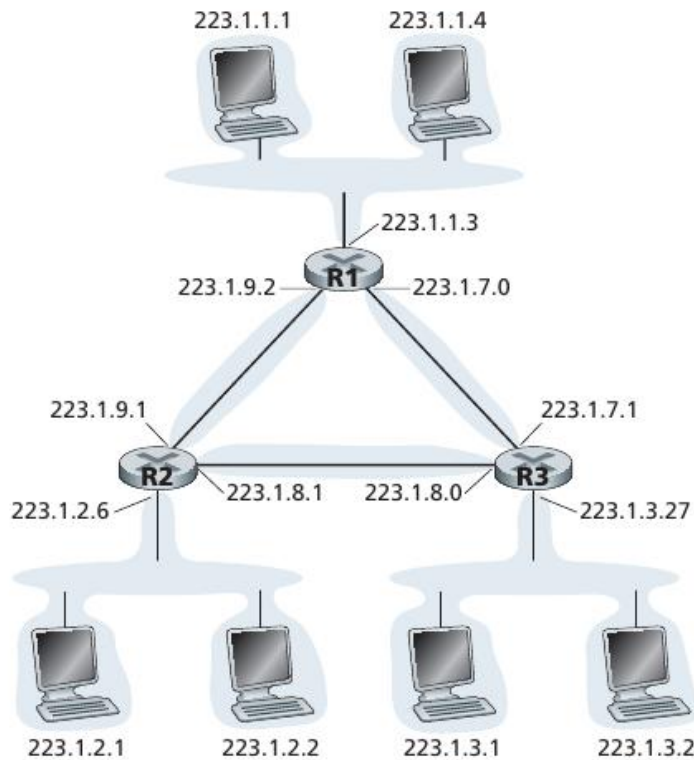
# IPv4 címzés



- ▶ IP címet két különböző dologra használjuk
  - ▶ azonosítás
  - ▶ útvonalválasztás
- ▶ Minden interfésznek van egy címe
- ▶ routernek természetesen több interfésze van
  - ▶ dotted-decimal jelölés (pl. 192.168.1.1)
- ▶ alhálózatok
  - ▶ 32 bites cím két részre osztva (subnet mask alapján):  
hálózati cím, hoszt cím (pl. 192.168.1.0/24)
- ▶ classful addressing
  - ▶ class A (/8, 8 bites hálózati cím), B (/16), C (/24)
  - ▶ (class D: multicast)
  - ▶ első bitek határozzák meg
    - ▶ A: 0, B: 10, C: 110, D: 1110
- ▶ CIDR
  - ▶ Classless Interdomain Routing
  - ▶ tetszőleges network prefix: /x

J. Kurose, K. Ross, "Computer Networking: A Top-Down Approach", Pearson, 2012

# IPv4 címzés



J. Kurose, K. Ross, "Computer Networking: A Top-Down Approach", Pearson, 2012

## ▶ FLSM

- ▶ Fixed Length Subnet Mask
- ▶ útvonalhirdetés prefix nélkül
- ▶ fogadó interfész maszkja volt a default
- ▶ azonos "méretű" alhálózatok
- ▶ sokszor nagyon pazarló (kihasználatlan IP címek)

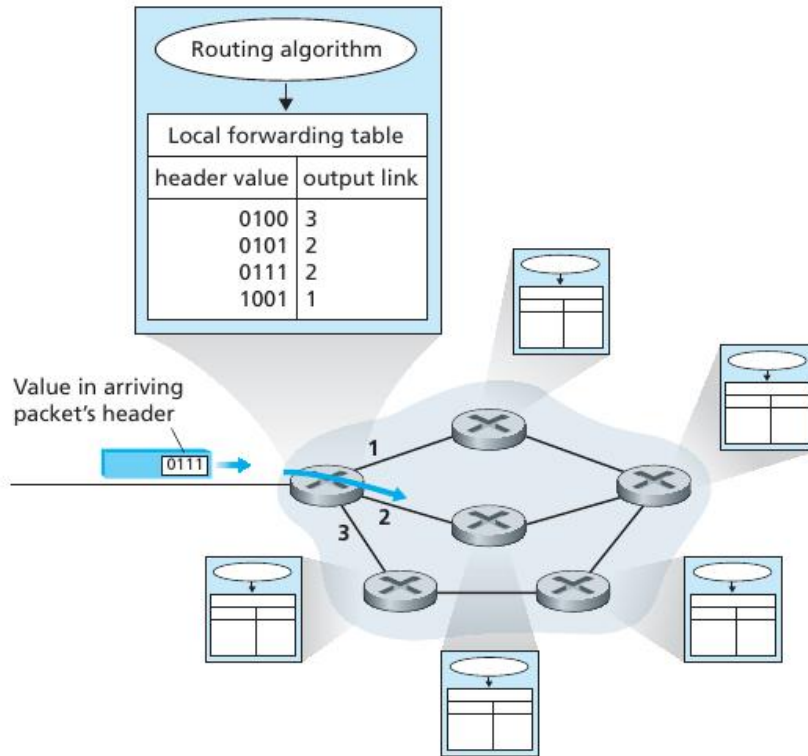
## ▶ VLSM

- ▶ Variable Length Subnet Mask
- ▶ (VLSM + CIDR)
- ▶ tetszőleges "méretű" alhálózatok
- ▶ kevésbé pazarló
- ▶ útvonalhirdetés prefix-szel
  - ▶ IP cím / prefix
- ▶ routing prefix aggregation

# Forwarding

Hogy működik egy router?

# Forwarding és routing



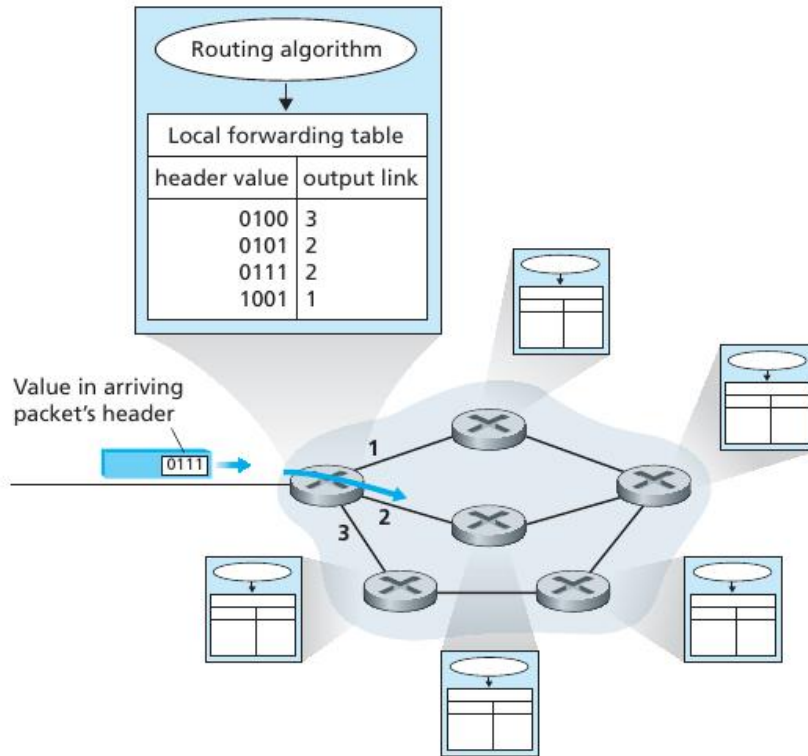
## ▶ Forwarding

- ▶ lokális döntési mechanizmus
- ▶ forwarding tábla alapján
  - ▶ pl. cél azonosítóhoz -> kimeneti interfész
- ▶ (1) routerhez beérkezik a csomag
- ▶ (2) fejrész érdekes részét vizsgálja
  - ▶ pl. cél IP
- ▶ (3) lookup (match)
  - ▶ illeszkedés vizsgálat (longest prefix match)
  - ▶ megfelelő bejegyzés kiválasztása a forwarding táblából
- ▶ (4) csomag továbbítása a kiválasztott bejegyzés szerinti port(ok)on (action)
- ▶ match & action (Hol láttuk korábban?)

J. Kurose, K. Ross, "Computer Networking: A Top-Down Approach", Pearson, 2012



# Forwarding és routing

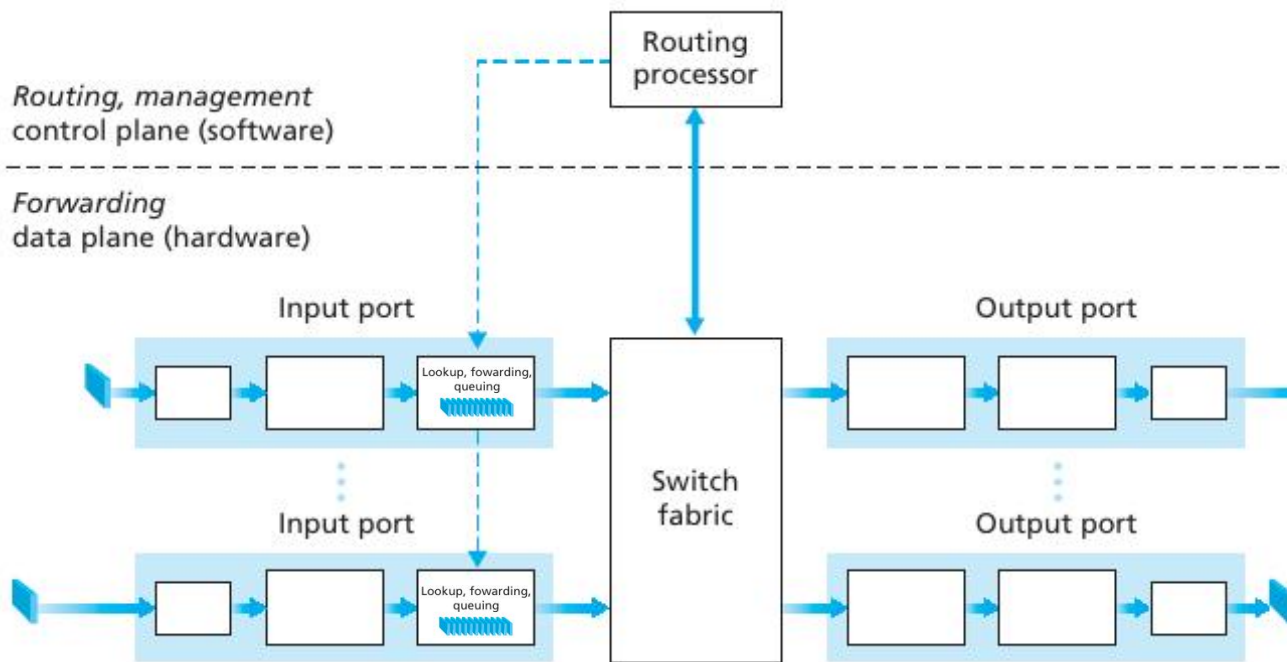


J. Kurose, K. Ross, "Computer Networking: A Top-Down Approach", Pearson, 2012

## ▶ Routing

- ▶ forwarding tábla dinamikus konfigurálása (változó hálózat)
- ▶ bejegyzések hozzáadása, törlése
  - ▶ routing protokoll üzenetek alapján
- ▶ centralizált vs. elosztott
- ▶ centralizált
  - ▶ egy központi helyen fut, látja a teljes topológiát (ld. majd OpenFlow hálózatok)
  - ▶ router-kontroller kommunikáció
- ▶ elosztott
  - ▶ routereken elosztottan fut az algoritmus (pl. hagyományos routing protokollok)
  - ▶ routerek kommunikálnak egymással, folyamatos információcsere

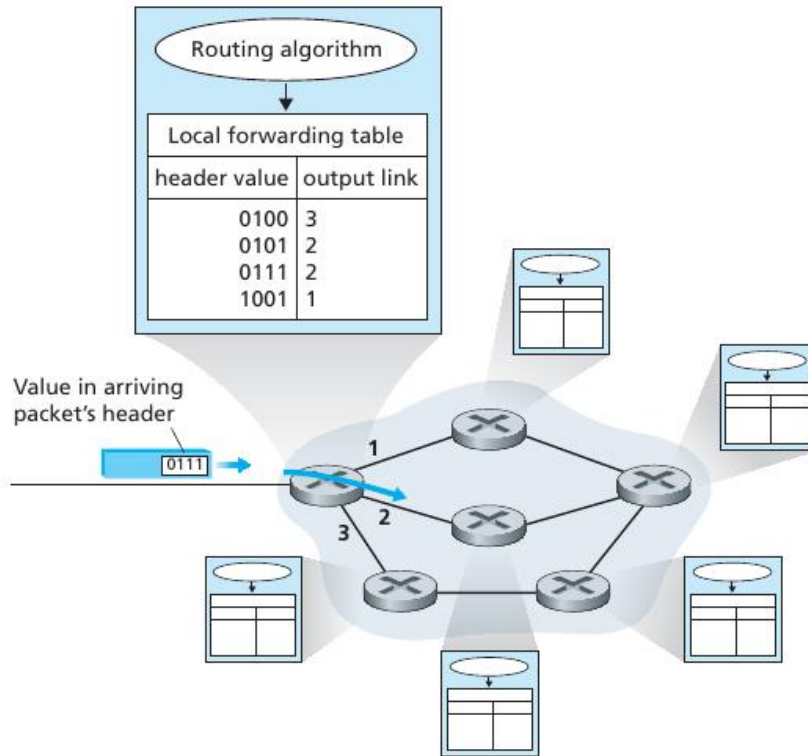
# Tipikus router architektúra



- ▶ **Forwarding plane**
  - ▶ tipikusan hardverben
  - ▶ bemeneti portok
    - ▶ lookup!
    - ▶ TCAM (Ternary Content Addressable Memory, három állapotú tartalom címezhető memória)
  - ▶ switch fabric
  - ▶ kimeneti portok
- ▶ **Control plane**
  - ▶ tipikusan szoftverben
  - ▶ routing processor
    - ▶ (kis teljesítményű) CPU
  - ▶ routing protokoll(ok) futtatása
  - ▶ routing táblák menedzselése
  - ▶ forwarding tábla kalkulálása, konfigurálása

J. Kurose, K. Ross, "Computer Networking: A Top-Down Approach", Pearson, 2012

# RIB vs. FIB



## ▶ RIB

- ▶ Routing Information Base
- ▶ “routing táblák”
- ▶ tipikusan RAM-ban
- ▶ routing protokolloktól jövő útvonal információk tárolóhelye
- ▶ minden útvonal, amit valamelyik futó routing algoritmus megtanult

## ▶ FIB

- ▶ Forwarding Information Base
- ▶ “forwarding tábla”
- ▶ tipikusan TCAM-ben, korlátos méret
- ▶ performanciára optimalizált bejegyzések
- ▶ longest prefix match
  - ▶ nem tárolunk minden lehetséges cél címet
  - ▶ döntés cél prefix alapján
- ▶ FIB-et dinamikusan kell kalkulálni a RIB alapján (next hop, output interfaces)

J. Kurose, K. Ross, “Computer Networking: A Top-Down Approach”, Pearson, 2012



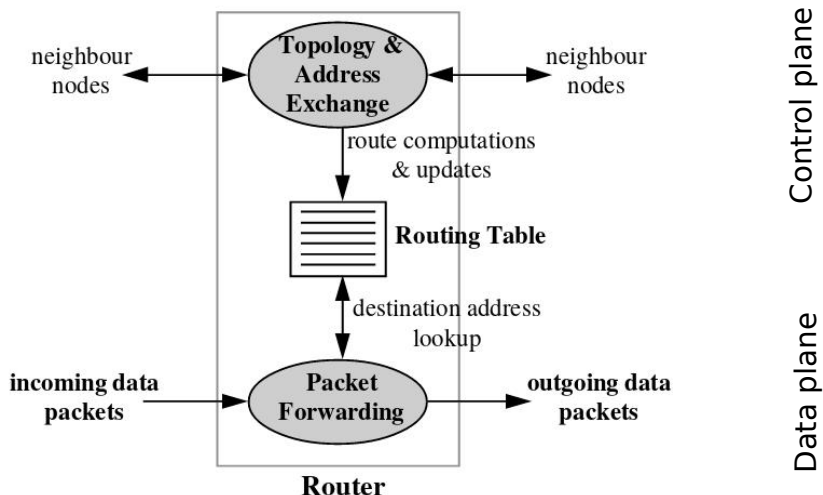
# (nem otthoni) routerek

## Németh Felicián slide-jai

home, access,

edge, core

# Routerek általános felépítése



James Aweya: IP Router Architectures: An Overview

## A router feladatai

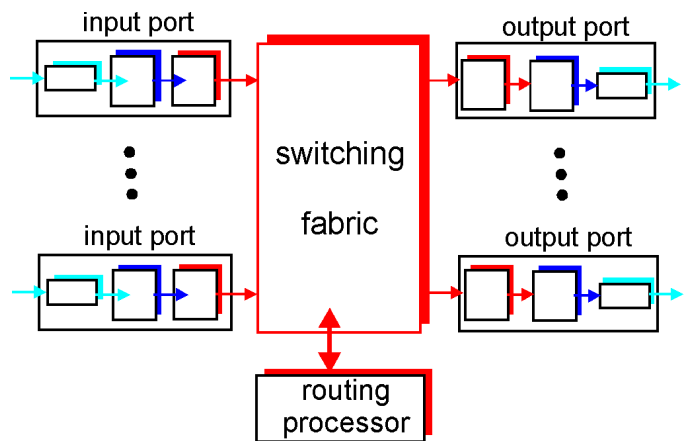
### ▸ Routing feladatok

- routing protokoll (OSPF, ...) futtatása, a kapott információk alapján:
- útvonalak kiszámítása, amit felhasználva:
- a továbbítási táblázat karbantartása

### ▸ Csomagtovábbítási feladatok

- IP csomag validálása
- Célcím kikeresése a továbbítási táblázatból
  - helyi, unicast, multicast cím
- Csomag-élettartam szabályozás
  - TTL, Time-to-live mező csökkentése
- Ellenőrzőösszeg újraszámítás
- Eltérő MTU esetén csomagdarabolás

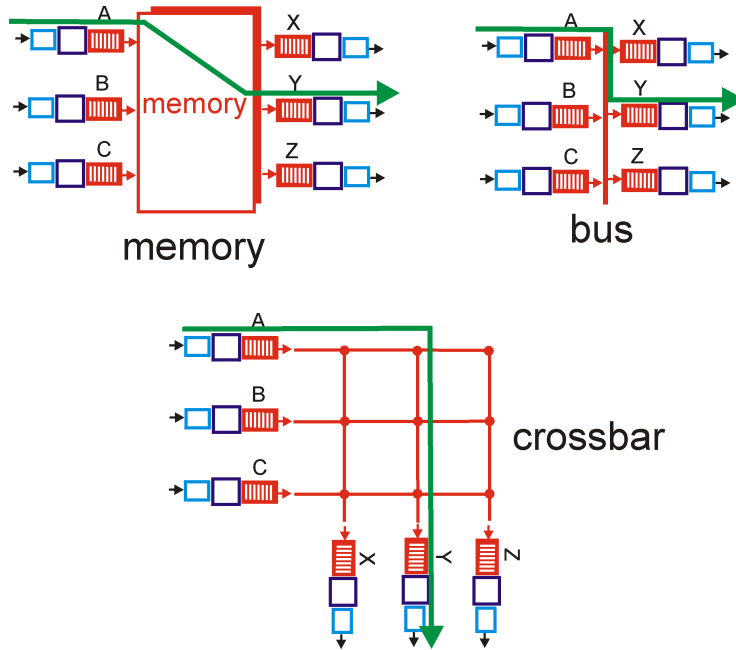
# Router architektúrák



[http://www2.ic.uff.br/~michael/kr1999/4-network/4\\_06-inside.htm](http://www2.ic.uff.br/~michael/kr1999/4-network/4_06-inside.htm)

- ▶ A portok “line card”-okon csatlakoztathatók
- ▶ A továbbításhoz szükséges legtöbb feladatot a line cardokon el lehet végezni
- ▶ Ideális esetben a routing processor nem végez csomagtovábbítási feladatokat
  - ▶ A line cardok rendelkeznek a forwarding tábla másolataival

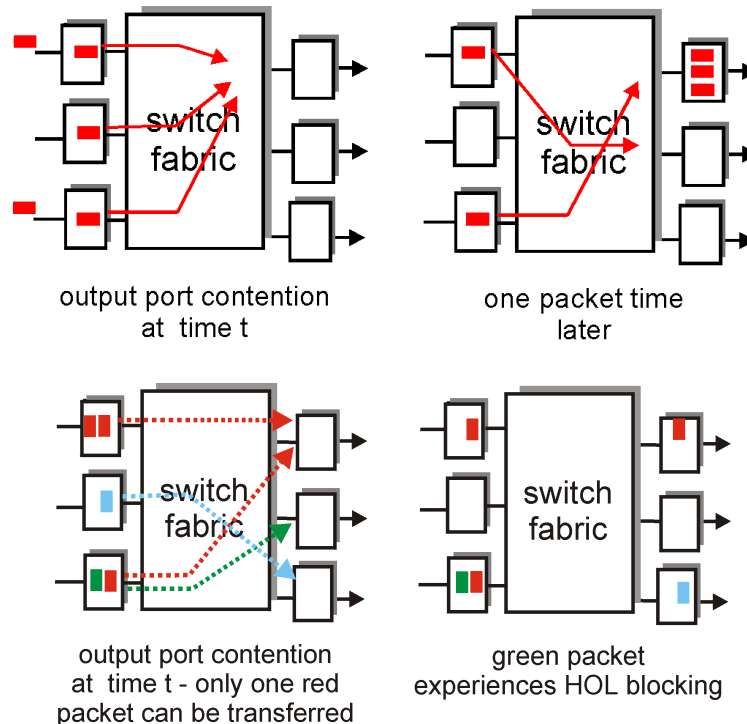
# Switching fabric / kapcsolószerkezet



- ▶ Kezdetben hagyományos számítógép architektúrákat használtak (~ OpenWrt)
  - ▶ csomagokat a közös hozzáférésű memóriába másolták
  - ▶ az egyetlen CPU döntött a sorsáról
- ▶ 2<sup>nd</sup> gen: csomagot a közös buszon továbbították a routing processor beavatkozása nélkül
- ▶ Crossbar:  $2N$  busz,  $N$  input-output port összekapcsolására
  - ▶ (szokás ilyenkor a csomagot kisebb egységekre darabolni, majd az output porton összerakni)

# Sorbanállás, sormenedzsmment, QoS

- ▶ Sorok a be- és kimeneten is lehetnek
  - ▶ Kimeneti torlódás oka: pl. nagyobb a bejövő összerhelés a kimeneti kapacitásnál
  - ▶ Head-of-the-line blocking:
    - ▶ a switching fabric nem N-szeres sebességgel üzemel,
    - ▶ a pirosat nem lehet még továbbítani, a zöldet lehetne, de a piros feltartja
- ▶ Minőség biztosítása (Quality of Service, QoS)
  - ▶ Nem FIFO sorokkal, pl:
  - ▶ Prioritásos sorokkal,
  - ▶ Weighted Fair Queueing
  - ▶ Random Early Detection (RED)
    - ▶ Torlódás jelzése csomagdobás nélkül (ECN bitben)
    - ▶ TCP estén működik

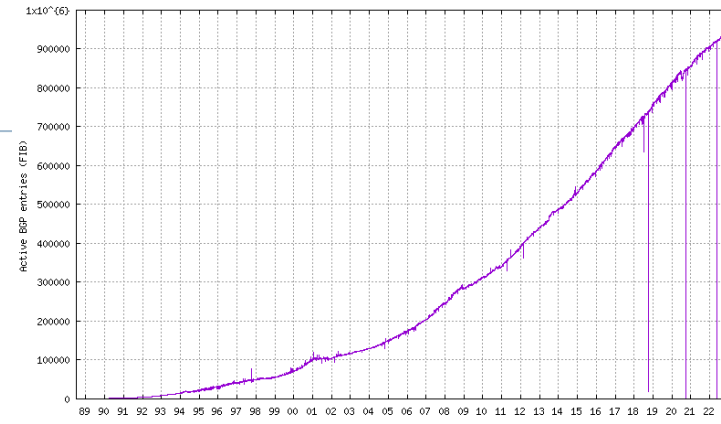




# Kimenő port kiválasztása: Longest Prefix Match

- ▶ BGP ~700k 900k bejegyzés
- ▶ Egyszerű bináris keresés
  - ▶ Túl sok memória hozzáférés
- ▶ TCAM (ternary content-addressable memory)
  - ▶ Túl drága a teljes táblázat tárolásához
- ▶ Gyorsítási lehetőségek:
  - ▶ A fa tetejét direktben címezzük
  - ▶ Fából irányított gráfot készítünk
    - ▶ Pl. alulról összevonjuk az azonos részfákat
    - ▶ Kisebb fa befér a cache-be → gyorsabb a memóriaelérés
  - ▶ ...

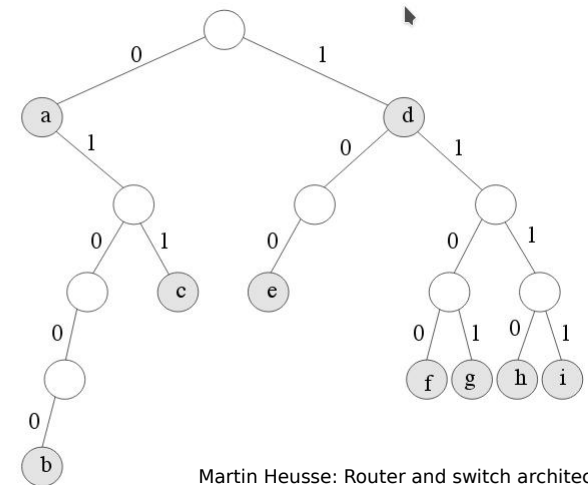
Active BGP entries (FIB)



<https://bgp.pötaroo.net/as2.0/bgp-active.html>

Prefixes

- a 0\*
- b 01000\*
- c 011\*
- d 1\*
- e 100\*
- f 1100\*
- g 1101\*
- h 1110\*
- i 1111\*



Martin Heusse: Router and switch architecture

# Routing protokollok



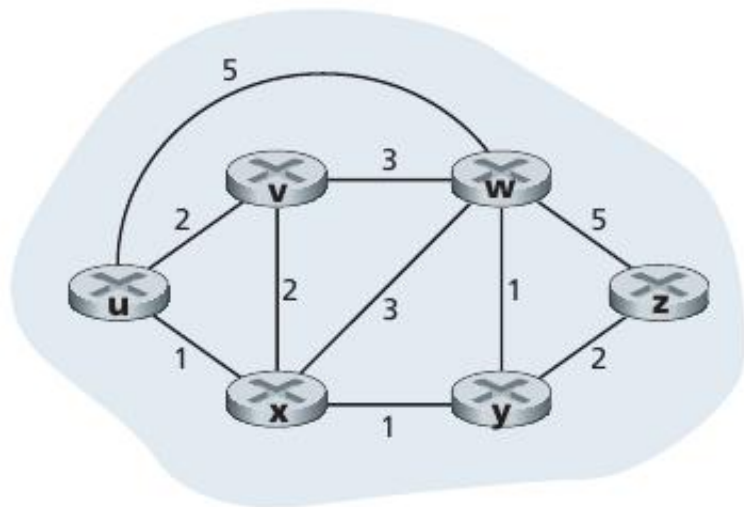
# Routing protokollok

---

## ▶ 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, sáv szélesség 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 domainek, 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

---

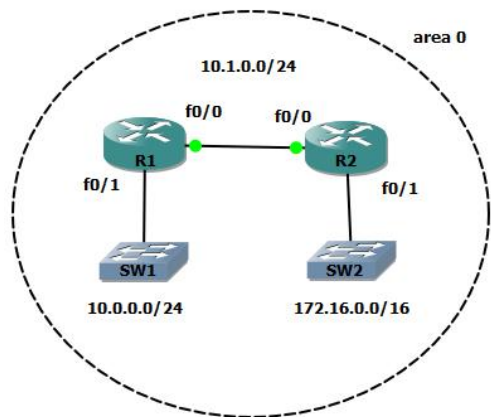
- ▶ Link State alapú routing algoritmus
  - ▶ globális nézetten dolgozik
  - ▶ LSP: Link State Packet (id, costs, seq.no, ttl)
  - ▶ egy router
    - ▶ mindenkinek küld LSP-t (broadcast)
    - ▶ a közvetlenül kapcsolódó linkjeiről
    - ▶ periodikusan újra generálja (seq.no++)
    - ▶ legfrissebb beérkezett 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



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
```

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

R2:

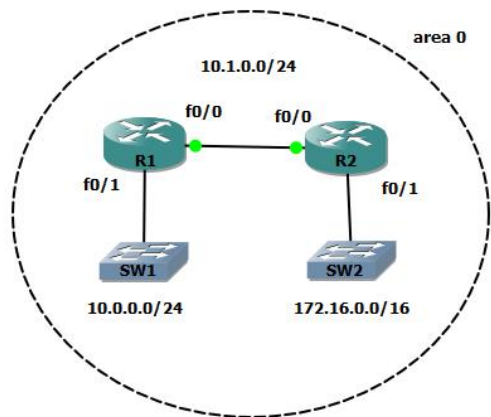
```
router ospf 1
```

```
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:

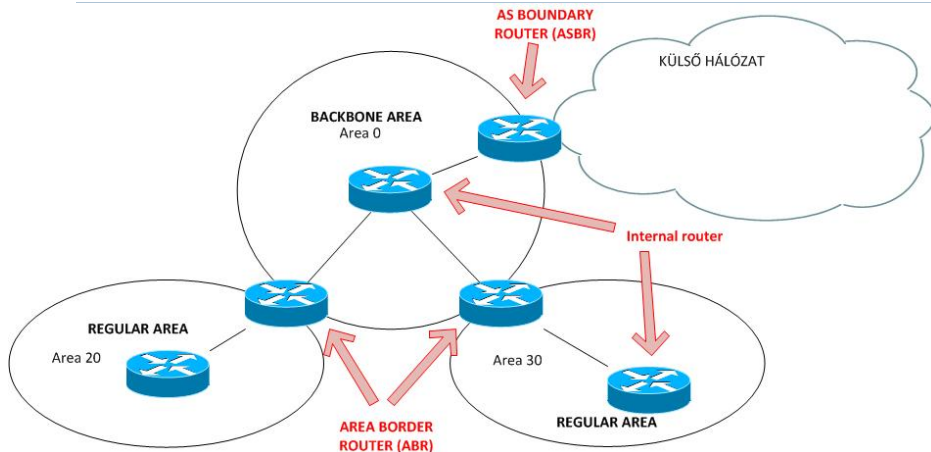
```
R1# show ip route
```

```
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 tábla
  - ▶ 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 (hop count a default routing metrika)
  - ▶ 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 megkezdí 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 elérhetlenné 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)
  - ▶ Id. korábban

# Például: RIP

---

- ▶ Hátrányok
  - ▶ (alapból) lassú konvergencia
  - ▶ limitált hálózatméret
  - ▶ nagy kontroll forgalom (sok router-router üzenetváltás)
- ▶ Előnyök
  - ▶ nagyon egyszerű konfigurálás

# 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