Az internet ökoszisztémája és evolúciója

Tartalom

- A Border Gateway Protocol (BGP)
  - felépítése, működése, folyamatábra, BGP üzenetek és attribútumok, a BGP döntési mechanizmus
- Útválasztási preferenciák megvalósítása BGP-n
  - valley-free routing import és export szűrők konfigurálásával, BGP comminity-k szerepe

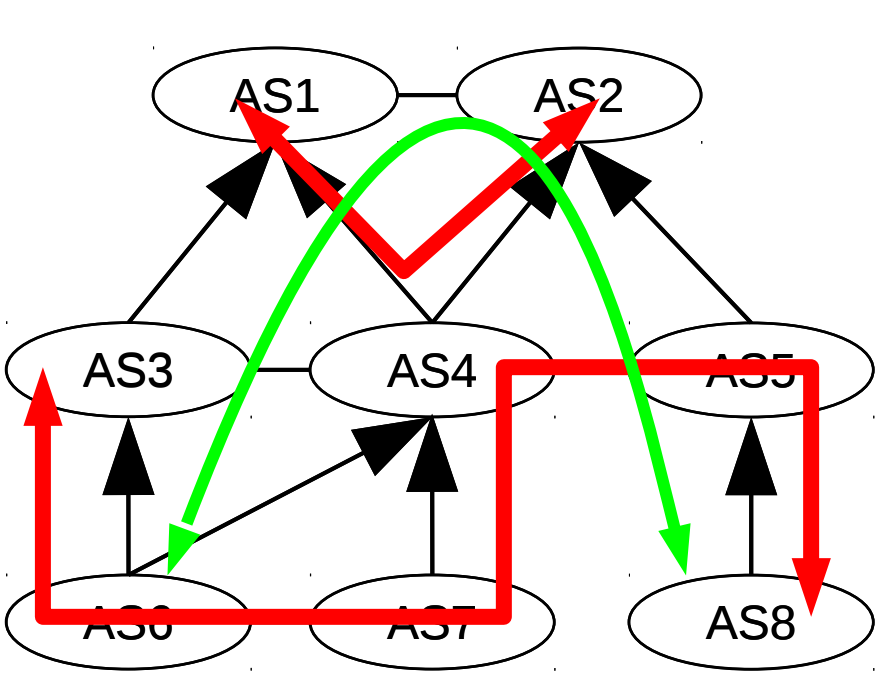
A Border Gateway Protocol

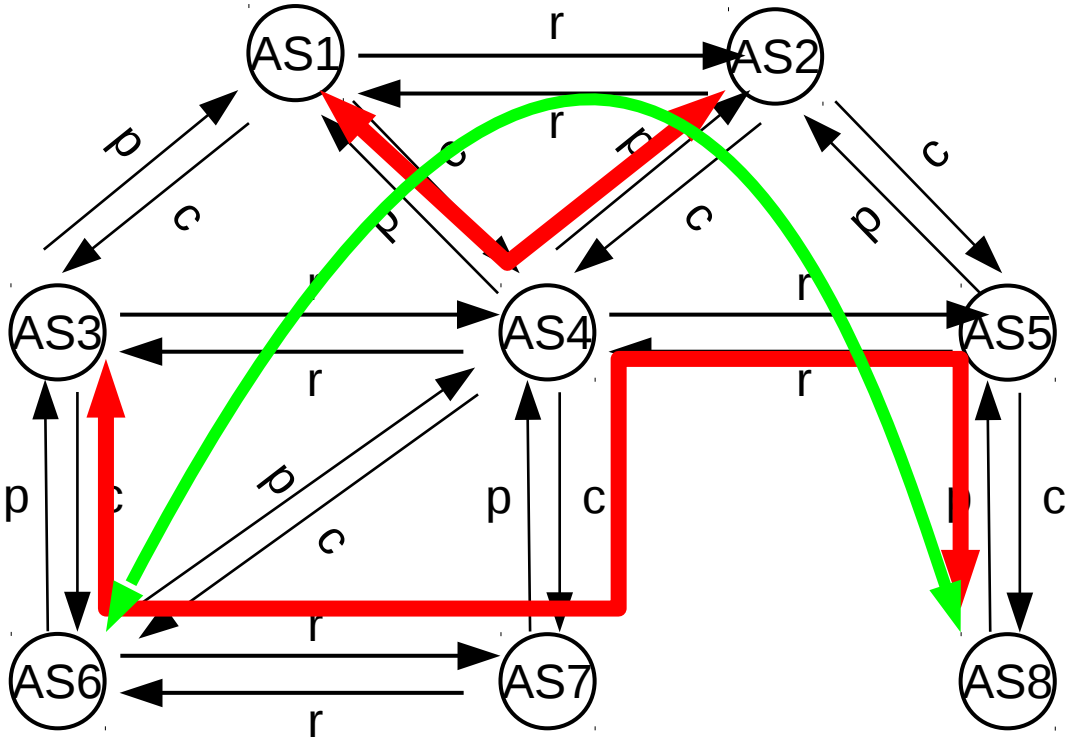
Ismétlés: AS kapcsolatok

- Két AS tipikusan vagy tranzit vagy peer kapcsolatot hoz létre
  - **tranzit:** globális internet-hozzáférés pénzért
  - **peer:** „ingyen” adatcsere két AS és azok összes előfizetője között
- Az interneten a forgalom arra halad(hat), amerre a cash-flow
- Tiltott és engedélyezett utak az AS–AS kapcsolatok függvényében: **valley-free routing**

Ismétlés: valley-free routing

- AS-szintű internet és gráfreprezentációja





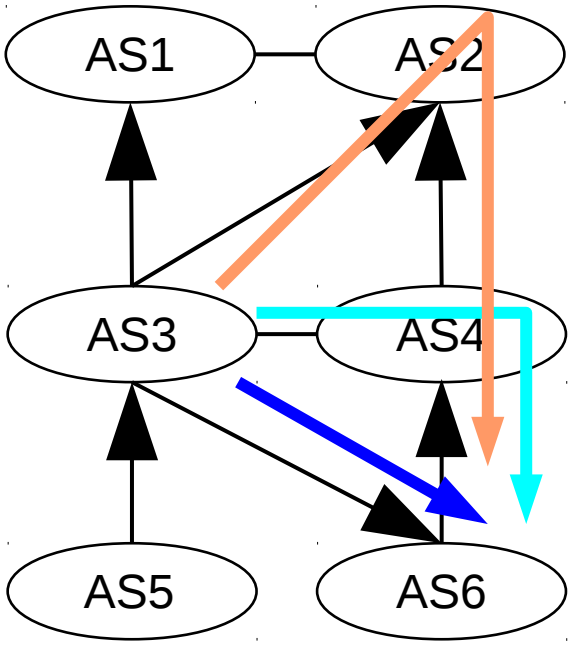
- Egy útvonal valley-free, ha címkéinek sorozata illeszkedik a *p*r?c** reguláris kifejezésre

Ismétlés: útvonalak preferenciája

- Preferáljuk az előfizetőn keresztüli útvonalakat: nem kerül pénzbe
- **Prefer customer** szabály:

$$P_c < P_r < P_p$$

- Azonosan preferált útvonalak közül a **rövidebbet** (kevesebb AS-en halad keresztül) választjuk



Inter-domain útvonválasztás

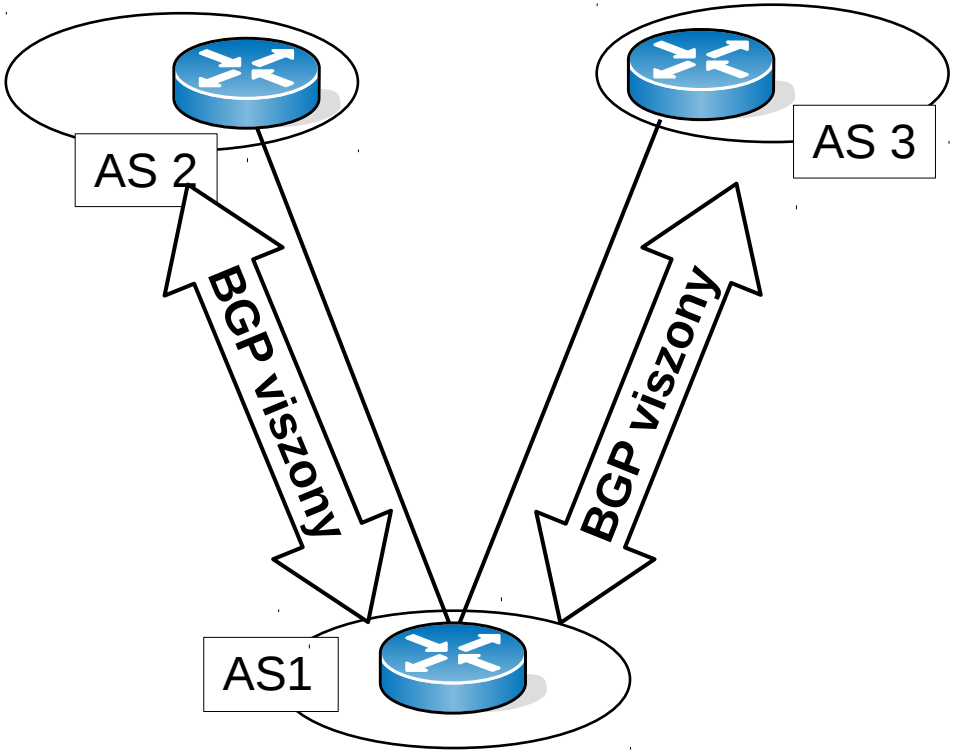
- Az elosztott útvonal-választási módszerek közül a **path-vector** a leghatékonyabb bonyolult útválasztási stratégiák leképezésére
  - mert meghagyja az ASek önállóságát saját preferenciáik megvalósítására
  - mert nem használ „távolság-metrikát” (a link-state esetében ez azért kell, hogy a routerek elosztott számításai ugyanazokhoz az utakhoz konvergáljanak)
- Az interneten használt EGP path-vector alapú

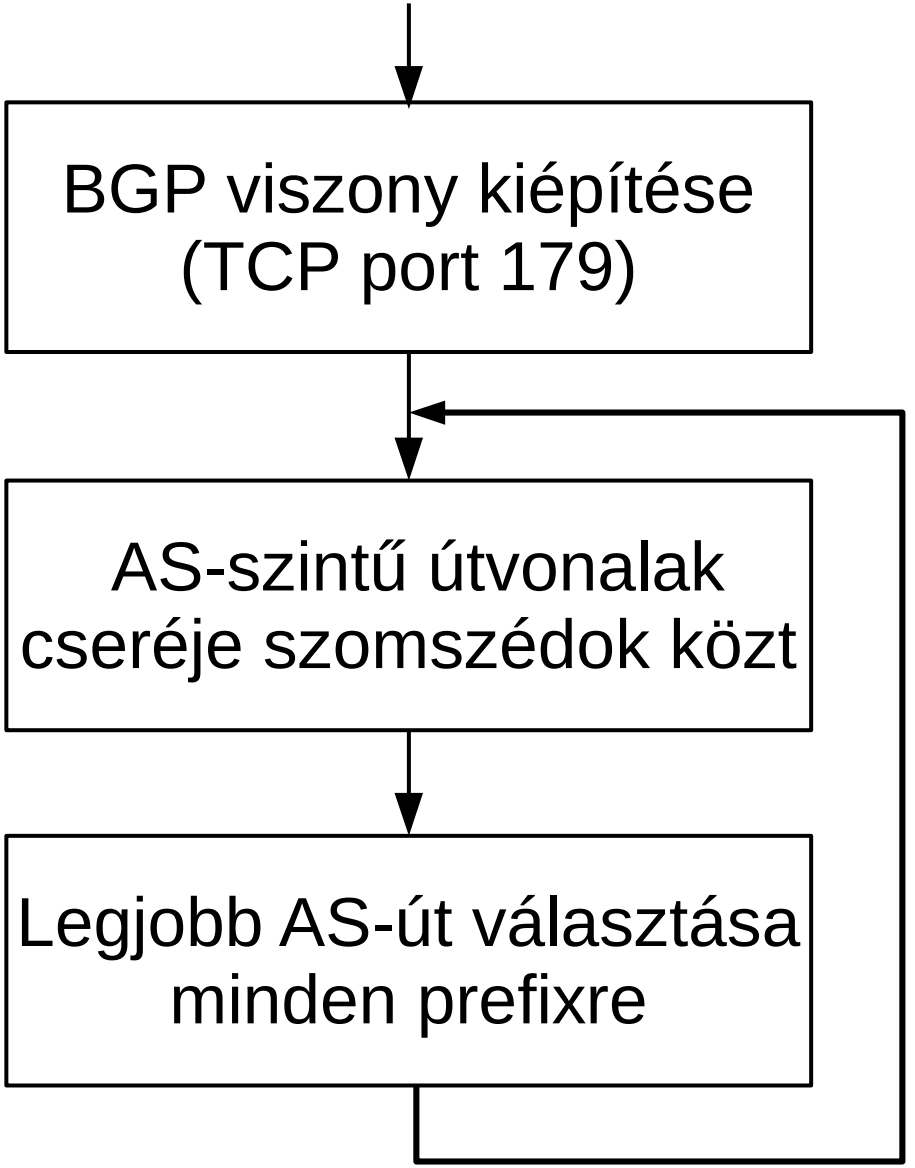
BGP: a path-vector EGP

- **Border Gateway Protocol version 4: BGP**
  - path-vector alapú policy routing protocol
  - de facto szabvány EGP az interneten
  - a protokoll egyszerű, a konfigurációja nehéz
- 15 éves evolúció eredménye:
  - 1989 : BGP-1 [RFC 1105]
  - 1995 : BGP-4 [RFC 1771]: CIDR támogatás
  - azóta minimális változtatás

BGP: működés

- Szomszédos routerek **BGP viszonyt** építenek ki egymás közt





BGP: a path-vector EGP

- A BGP-ben a **célpontok** a meghirdetett IP alhálózati **prefixek**
  - egy prefix → egy útvonal
  - prefixen belül az összes IP címre azonos út
- Útválasztás az **AS-szintű útvonalak** alapján
  - útvonal állomásai ASek, élei AS–AS linkek
  - a címkéket (*p*, *c*, *r*) a protokoll nem ismeri és nem terjeszti (hiszen ez üzleti titok!)
  - egyedül a szomszédos ASek felé ismert a kapcsolat típusa (tranzit/peer)

BGP: path-vector EGP

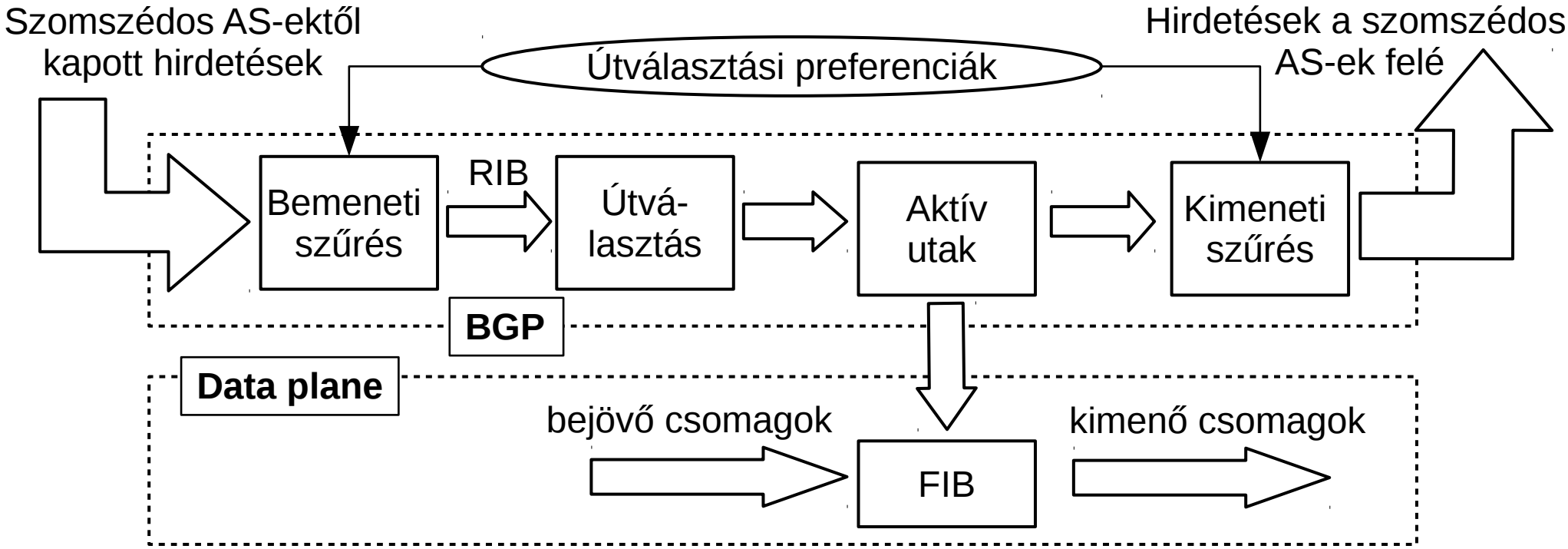
- Minden AS közli a szomszéd ASekkel az általa ismert legjobb utat egy-egy prefixre: **hirdetés**
  - leírja az útvonalhoz tartozó ASek sorozatát
  - plusz néhány egyéb attribútumot
- A szomszéd AS számára átadott útvonalakra **kimeneti szűrés (export filter)** alkalmazható
  - hirdetések visszatarthatók vagy átírhatók
- A szomszéd AStől kapott útvonalakra **bemeneti szűrés (import filter)**
  - nem kell minden hirdetést elfogadni

BGP: path-vector EGP

- A hirdetéseket egy AS-útvonal-adatbázisban gyűjtjük: **BGP RIB**
  - természetesen a bemeneti szűrés után
- Az adatbázisból minden prefixre **kiválasztjuk a legjobb útvonalat: aktív út** adott prefixre
  - az AS saját útválasztási preferenciái szerint
  - például „valley-free routing” a „prefer customer” szabály mellett
  - „döntetlen” esetén: „legrövidebb AS-útvonal”
  - de „tetszőleges” preferencia érvényesíthető

BGP útválasztás: folyamatábra

- **BGP konfiguráció:** BGP viszonyok leírása + hirdetett prefixek + import/export szűrők
- Az útválasztás szűrőkkel befolyásolható



BGP: üzenetek

- **Open:** BGP viszony létrehozása két router közt
  - BGP viszonyban levő két router általában fizikailag is szomszédos (link van köztük)
- **Keep Alive:** viszony frissítése periodikusan
- **Notification:** viszony lezárása
- **Update:** útvonal frissült
  - **Announce (hirdetés):** új útvonal egy prefixre
  - **Withdraw (visszavonás):** meghirdetett útvonal visszavonása

BGP: hirdetés

- **BGP hirdetés = prefix + attribútomok**
- BGP attribútumok:

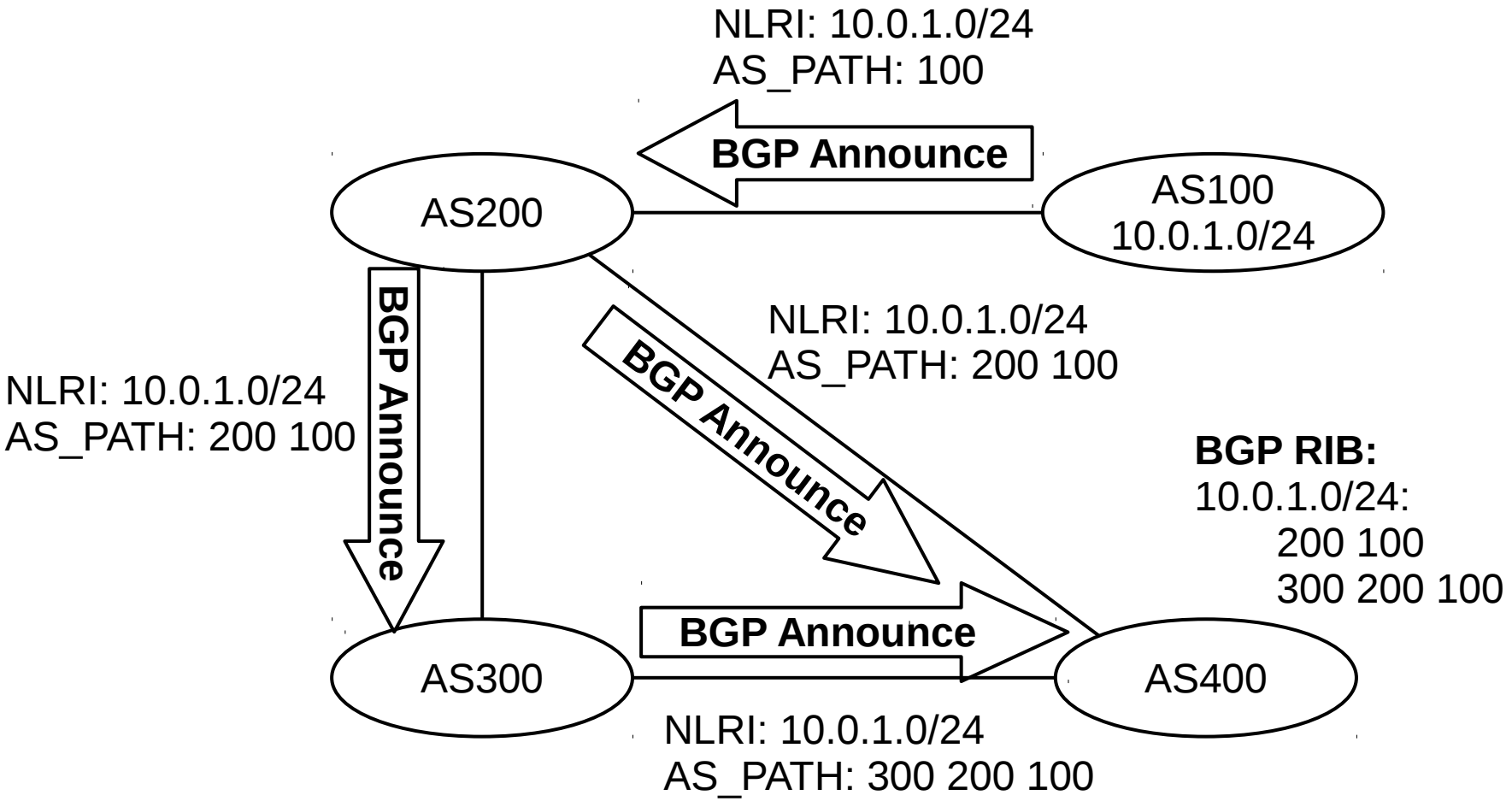
Value	Code	Reference
...		
2	AS_PATH	[RFC1771]
3	NEXT_HOP	[RFC1771]
...		
5	LOCAL_PREF	[RFC1771]
...		
8	COMMUNITY	[RFC1997]
...		
16	EXTENDED COMMUNITIES	[RFC4360]
...		
255	reserved for development	

BGP hirdetés: NLRI

- **Network Layer Reachability Information (NLRI)**: mely prefixre vonatkozik a hidretés
  - alapesetben egyetlen IP prefix: pl. `10.0.1.0/24`
  - több prefix is belefoglalható egy hidretésbe
  - ekkor a meghirdetett attribútumok az összes prefixre vonatkoznak
  - IPv6 cím is belekódolható (sőt, multicast címek is, stb.)

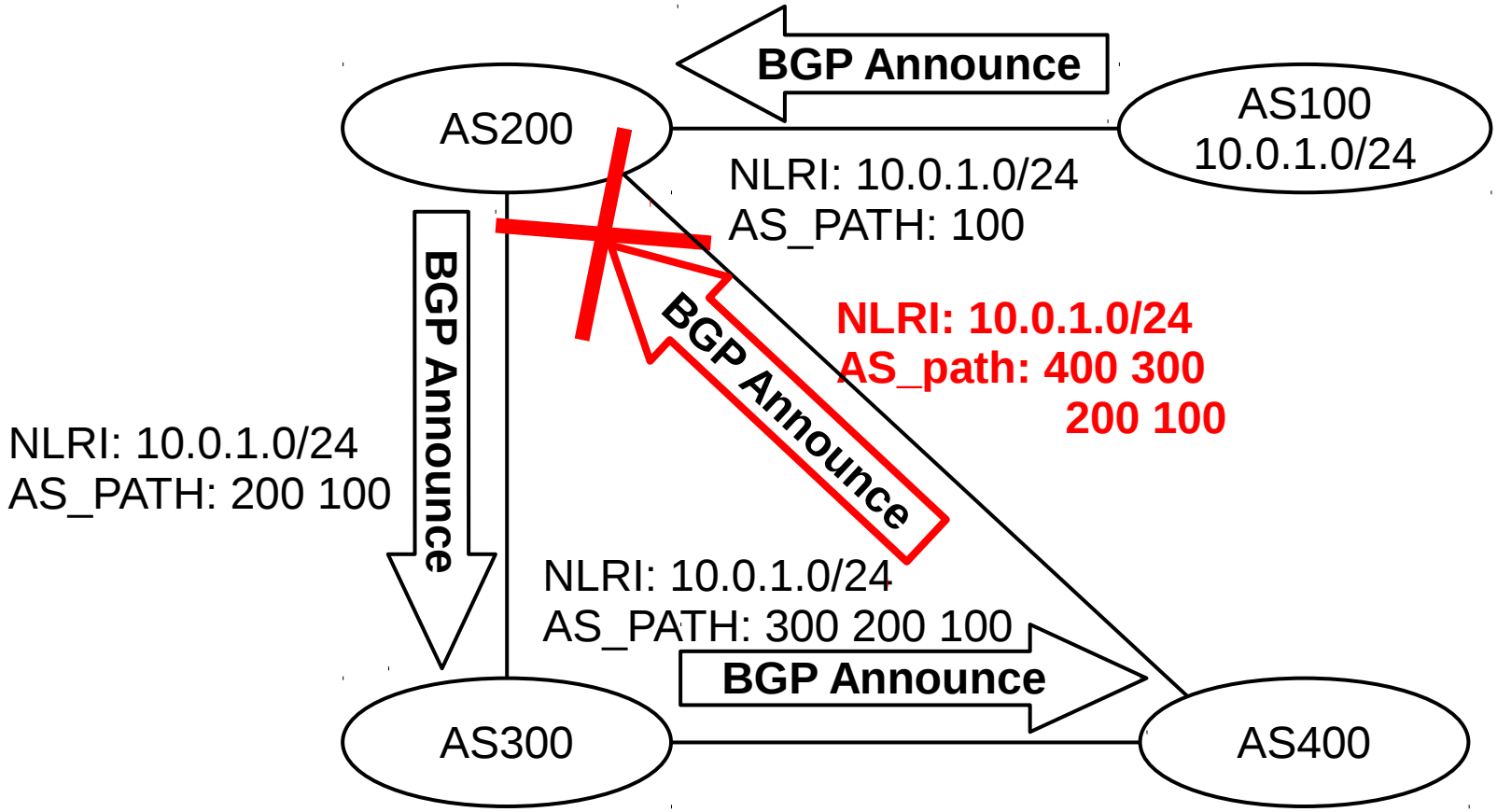
BGP attribútumok: AS_PATH

- AS-útvonal a prefix felé: AS számok listája



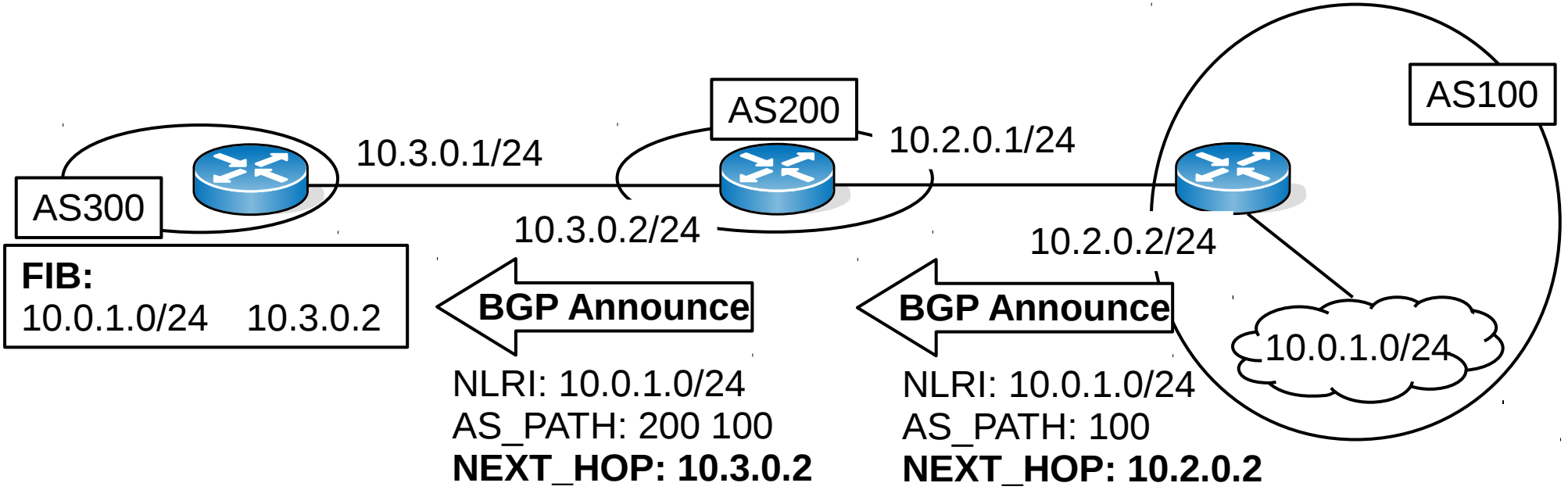
BGP: hurkok elkerülése

- **Szabály:** a BGP routerek nem fogadnak el olyan hirdetést, melyben az AS_PATH attribútumban megtalálják a saját AS számukat



BGP attribútumok: NEXT_HOP

- A meghirdetett AS útvonalhoz tartozó next-hop
  - a hirdetést elfogadó router ezt a next-hop címet rendeli a prefixhez a FIBjében
  - továbbításkor a saját IP címére állítja



További BGP attribútumok

- **LOCAL_PREF (Local Preference, lokális preferencia)**:
  - az egyik legfontosabb BGP attribútum
  - melyik hirdetést preferáljuk egy prefixre
- Leggyakrabban az AS útválasztási preferenciáinak BGP konfigurációkra való lefordítására használják (lásd később)
- **COMMUNITY/EXTENDED COMMUNITIES:** egyes hirdetésekhez „címkéket” rendelhetünk

BGP hirdetés: példa

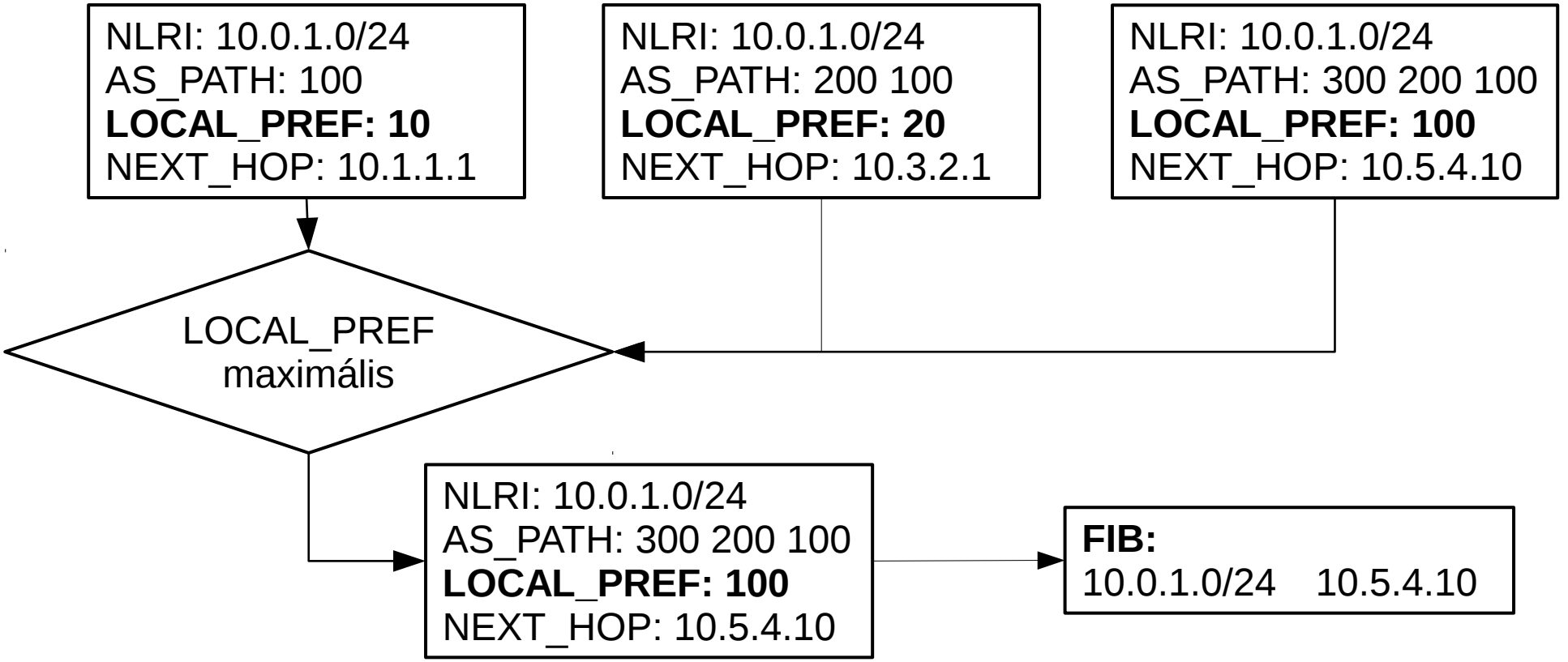
```
Internet Protocol Version 4, Src: … Dst: …
Transmission Control Protocol, Src Port: 58463
(58463), Dst Port: 179 (179), Seq: 84, Ack: 84,
Len: 52
Border Gateway Protocol - UPDATE Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 52
    Type: UPDATE Message (2)
    Withdrawn Routes Length: 0
    Total Path Attribute Length: 25
    Path attributes
        Path Attribute - ORIGIN: IGP
        Path Attribute - AS_PATH: 200 100
        Path Attribute - NEXT_HOP: 10.3.0.2
        Path Attribute - MULTI_EXIT_DISC: 0
    Network Layer Reachability Information (NLRI)
        10.0.1.0/24
```

BGP döntési mechanizmus

- Láttuk, hogy egy BGP router több hirdetést is kaphat egy adott prefixre
- Az AS útválasztási preferenciái (routing policy) döntik el, melyiket fogadjuk el
- Az útválasztási preferenciákat a BGP **döntési mechanizmusán** keresztül érvényesíthetjük:
  - **bemenet:** az összes hirdetés adott prefixre
  - **kimenete:** az aktív út a prefixre → FIB
  - a döntési mechanizmus a hirdetések attribútumait hasonlítja össze

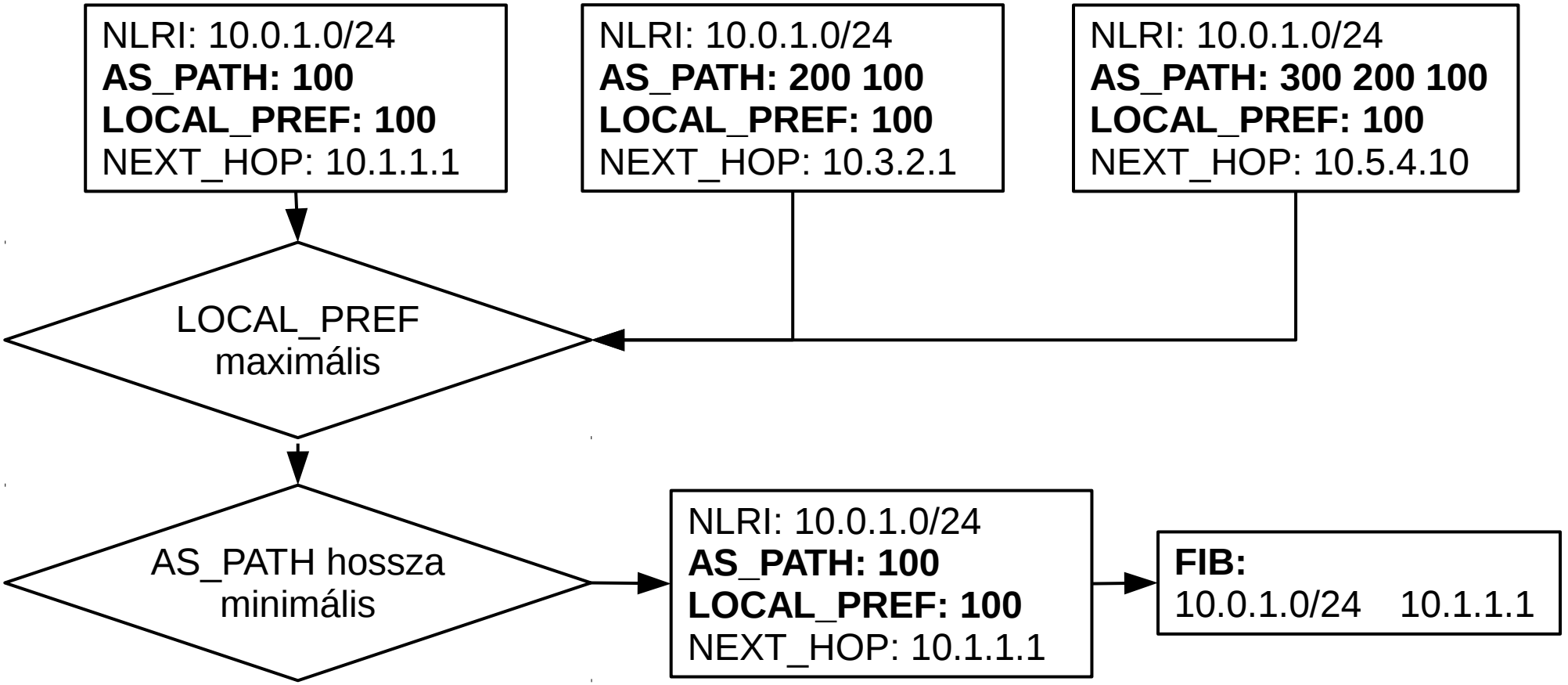
BGP döntési mechanizmus

- Elsőként azt a hirdetést fogadjuk el, amelynek **LOCAL_PREF attribútuma a legnagyobb**
  - akkor is, ha a hozzá tartozó AS út hosszabb!



BGP döntési mechanizmus

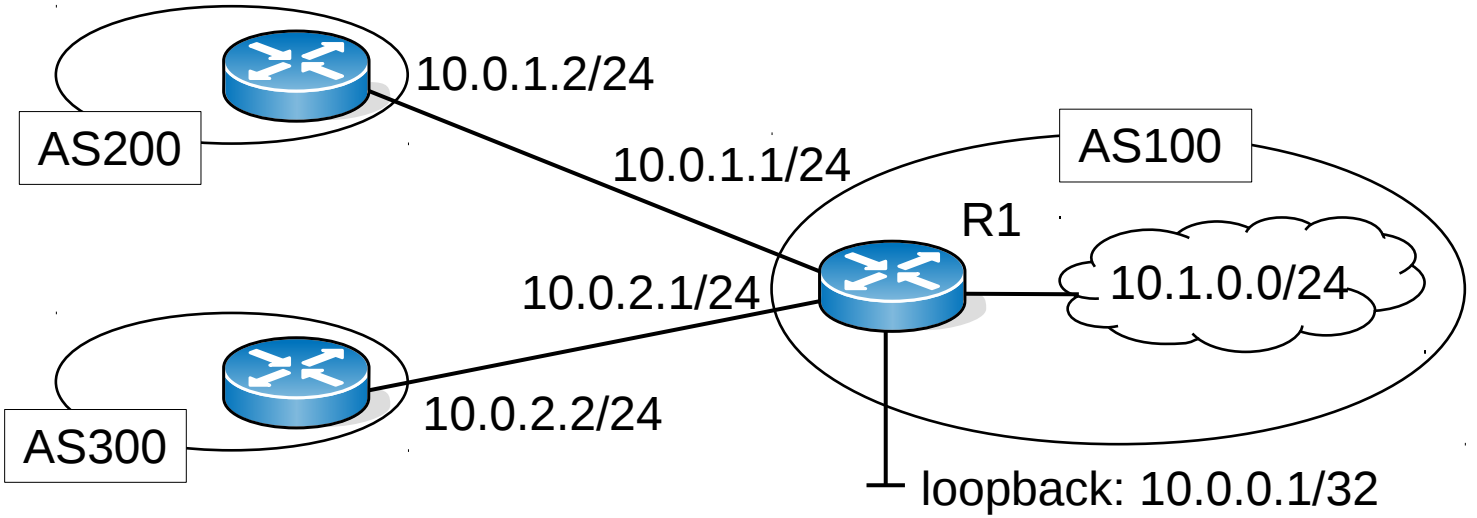
- Csak ha a lokális preferencia azonos több hirdetésen: **legrövidebb AS útvonal**
- Ha több opció maradt: további attribútumok



BGP konfiguráció: alapok

- Az alábbi konfiguráció adott, interfész IP címek beállítva az ábrán látható módon
- `R1`-en BGP router indítása és konfigurációjának kezdete: `router bgp <AS-szám>`

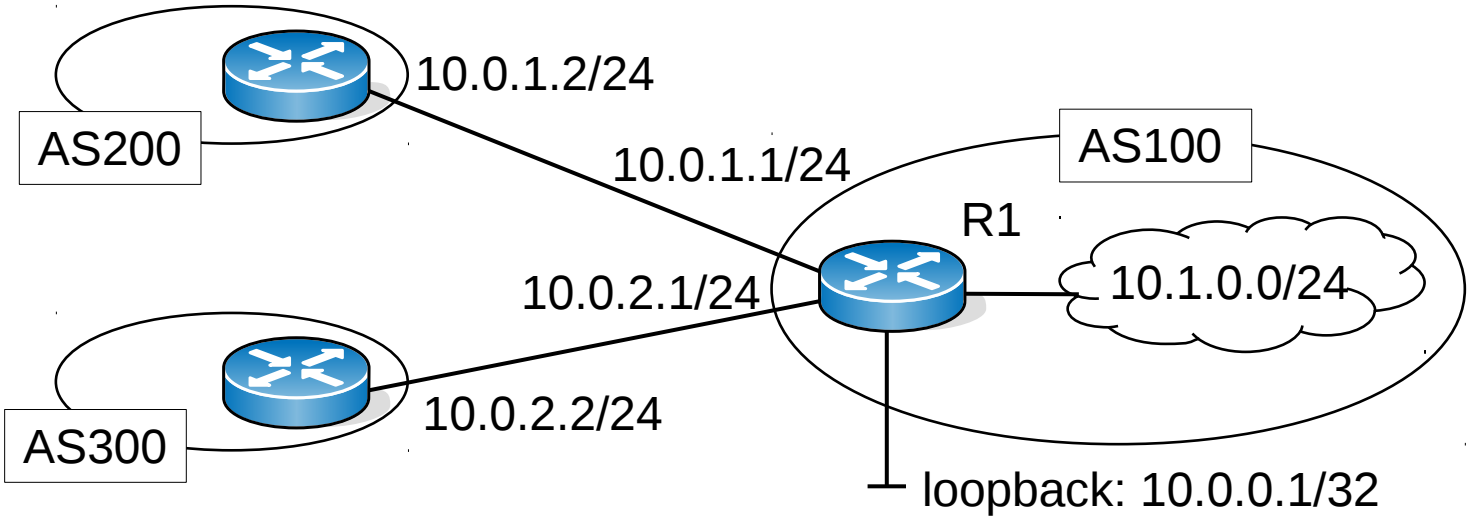
```
router bgp 100
```



BGP konfiguráció: alapok

- A BGP routereket érdemes egyedi azonosítóval ellátni, különben nincs egyedi IP cím, amivel hivatkozhatnánk rá/címezhetnénk
- Legtöbbször a `loopback` interfész címe

```
bgp router-id 10.0.0.1
```

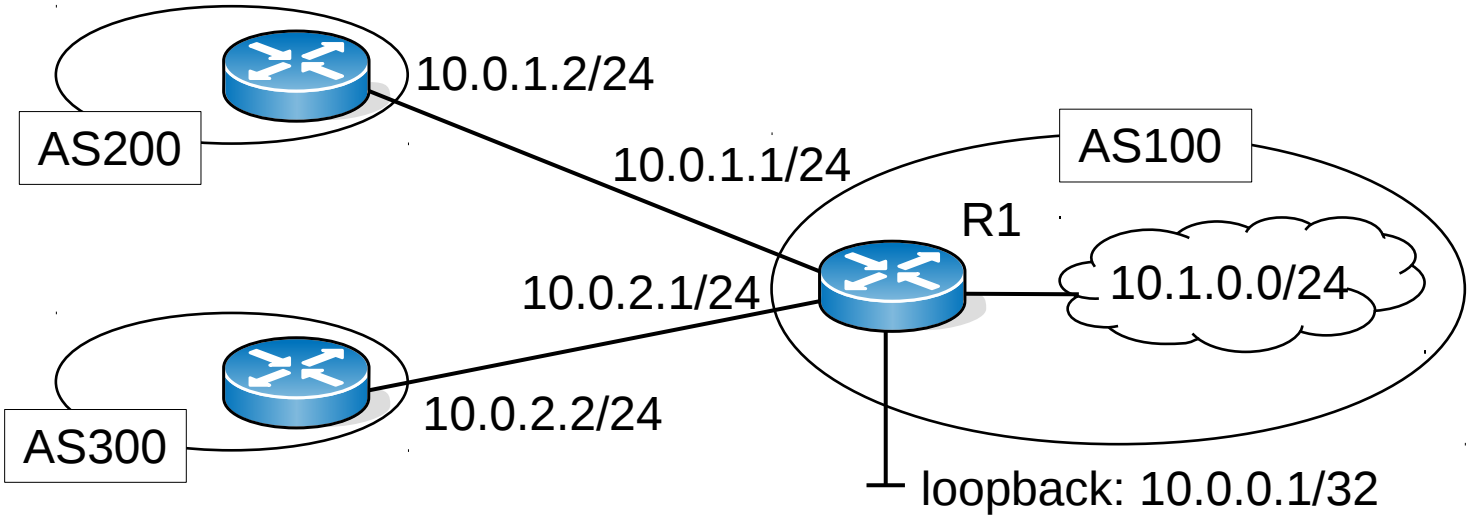


BGP konfiguráció: alapok

- `R1` a `10.1.0.0/24` címtartományt hirdeti:

  `network 10.1.0.0/24`

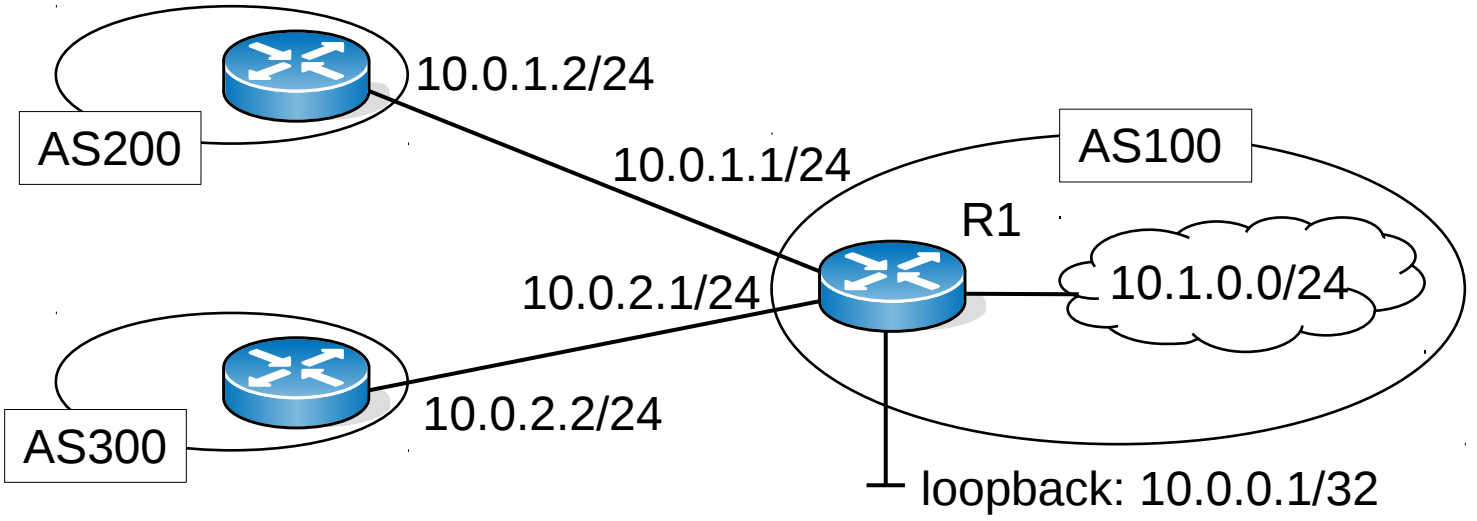
- Ha egy AS egy prefixet hirdet az interneten, garantálnia kell, hogy tényleg az övé, különben **prefix hijacking**



BGP konfiguráció: alapok

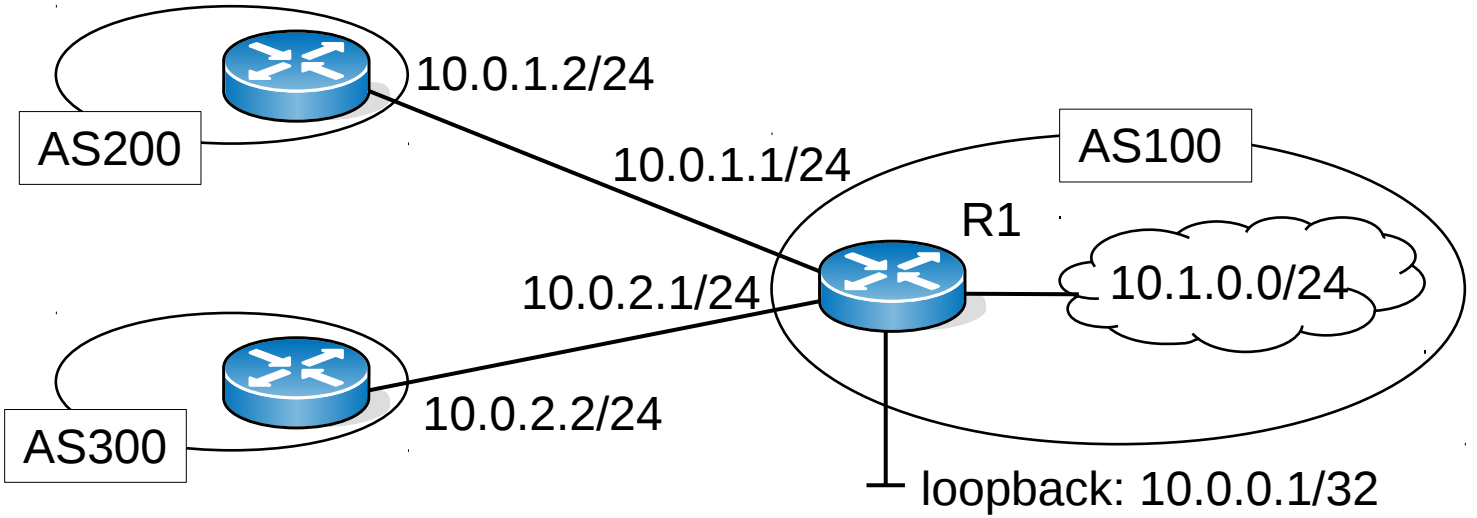
- A szomszédok címei és AS-számai, amelyekkel BGP kapcsolatot létesítünk:

```
neighbor 10.0.1.2 remote-as 200
neighbor 10.0.2.2 remote-as 300
```



BGP konfiguráció: alapok

```
router bgp 100
  bgp router-id 10.0.0.1
  network 10.1.0.0/24
  neighbor 10.0.1.2 remote-as 200
  neighbor 10.0.2.2 remote-as 300
```



Útválasztási preferenciák megvalósítása BGP-n

Policy routing BGP felett

Hogyan lehet azt elérni, hogy a BGP az általunk preferált útvonalakat válassza?

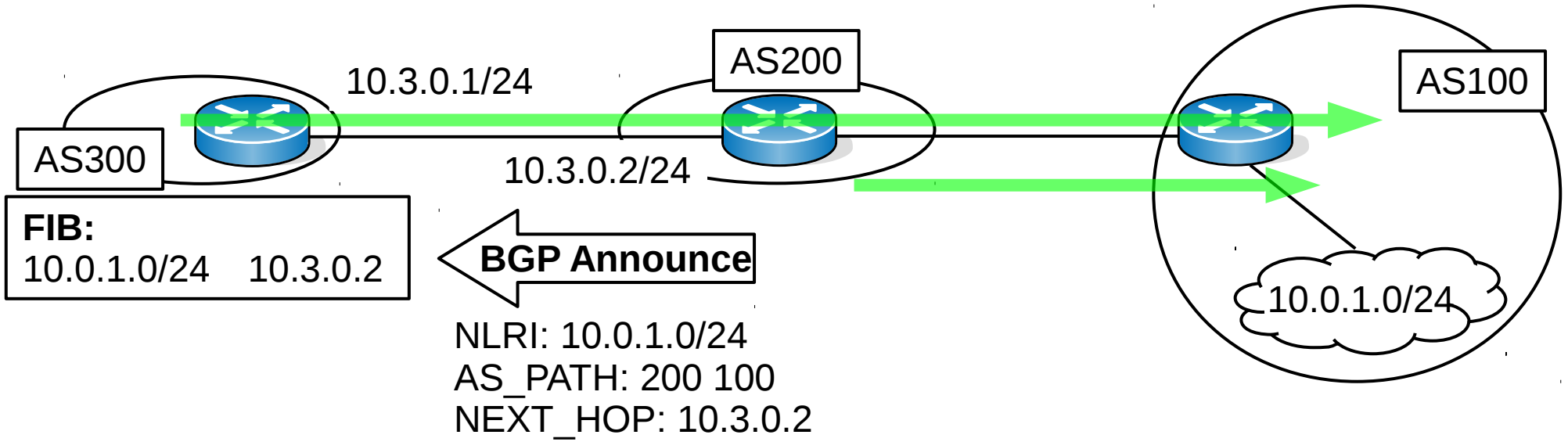
- Alapértelmezetten a BGP nem választ valley-free utakat, hiszen a koncepció nincs „bekódolva” a BGP döntési mechanizmusába
- Ha a BGP eltér az útválasztási preferenciáktól, az hatalmas anyagi veszteséget okozhat
- Például olyan forgalmakért is fizetünk egy szolgáltatónak, amelyeket ingyen is átvihettünk volna egy előfizetőn keresztül

Valley-free routing BGP felett

- Első körben azt vizsgáljuk, hogyan lehet elérni, hogy a BGP valley-free utakat válasszon
- Kihasználjuk azt, hogy egy AS csak azok közül az utak közül választhat, amelyeket valaki meghirdet számára
- Ha nem hirdetünk egy útvonalat egy szomszédnak, az nem is használhatja (elvileg)
- Ki kell választanunk, milyen utakat hirdetünk
- **BGP konfiguráció:** import/export szűrők segítségével

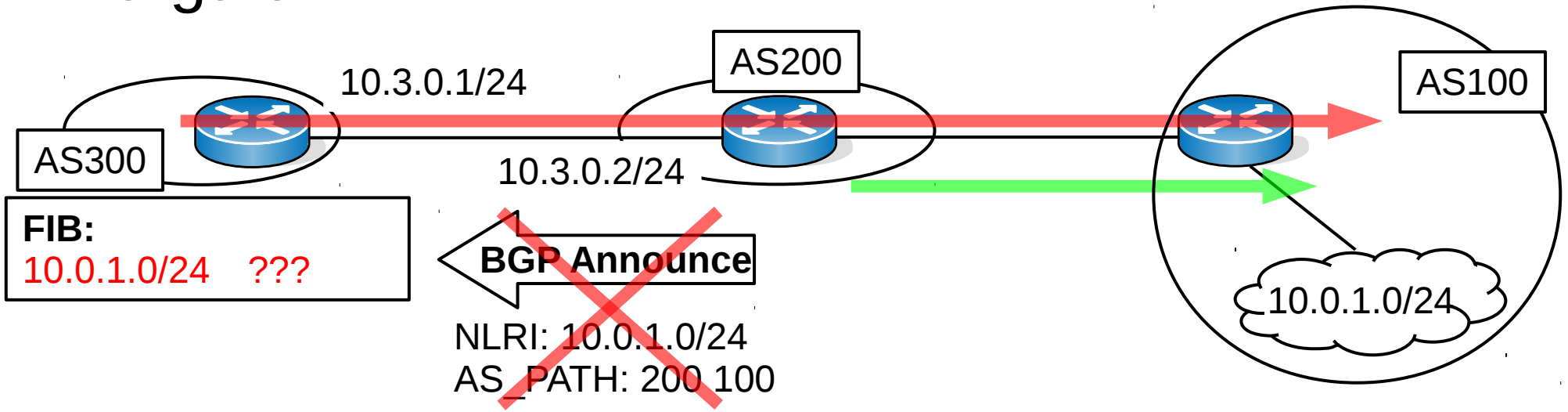
BGP hirdetés = tranzit

- Ha egy AS egy másik számára hirdet egy utat
- Vállalja, hogy annak forgalmát átviszi az úton
- **Példa:** `AS200` meghirdeti `AS300` számára a `10.0.1.0/24` prefixet, akkor vállalja, hogy továbbít az `AS300`→ `AS200` → `AS100` úton



BGP hirdetés visszatartása

- Ha visszatartja a hirdetést: a másik AS nem tud az útról, vagyis nem használhatja
- **Példa:** `AS200` nem hirdeti tovább `AS300` számára a `10.0.1.0/24` prefixet, így az `AS300`→ `AS200` → `AS100` úton nem haladhat forgalom

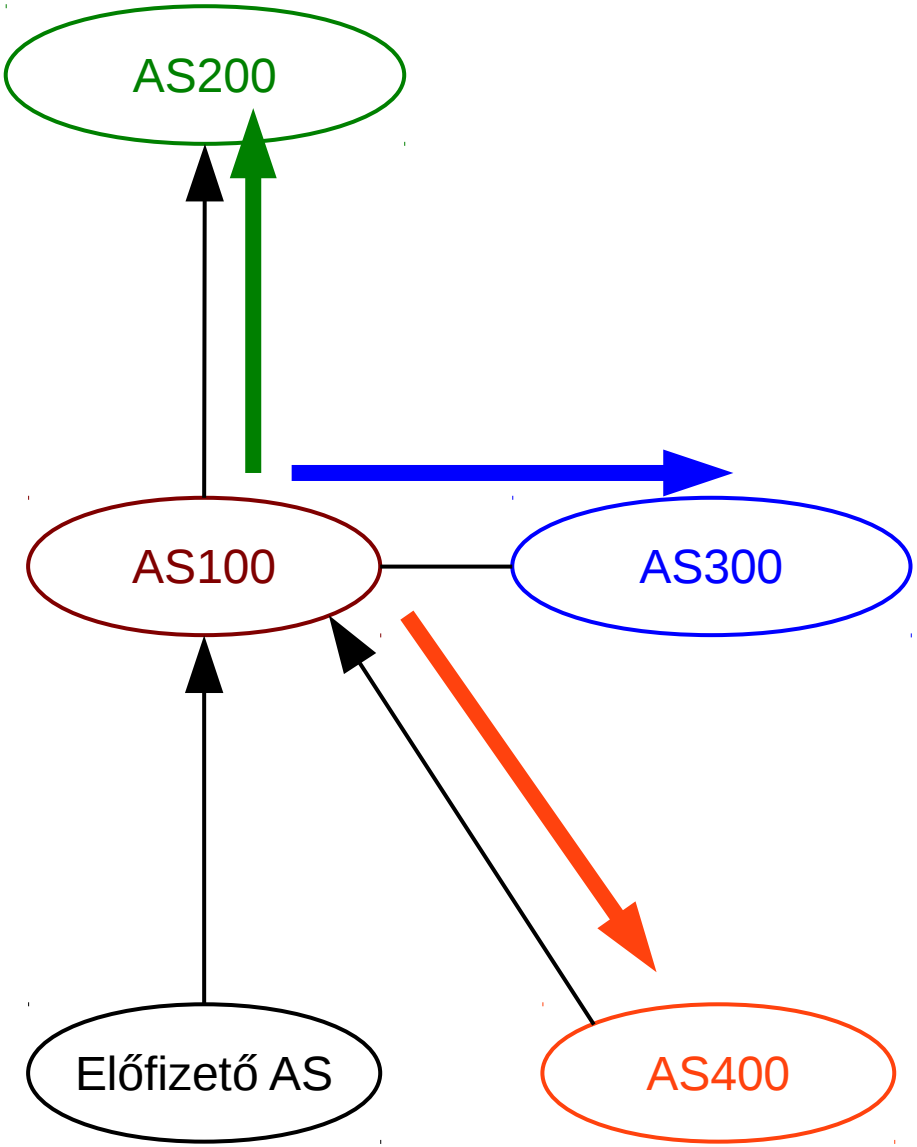


BGP hirdetések: szűrés

- **Feladat:** állítsuk be úgy a BGP import/export szűrőket, hogy a szomszéd AS csak valley-free utakról kapjon BGP hirdetést
- Másképp kell a útvonalakat/hirdetéseket szűrni attól függően, hogy azokat
  - milyen AS-AS kapcsolaton kaptuk (előfizetőtől, peer-től, vagy szolgáltatótól)
  - milyen AS-AS kapcsolaton küldjük (előfizető felő, peer felé, vagy szolgáltató felé)

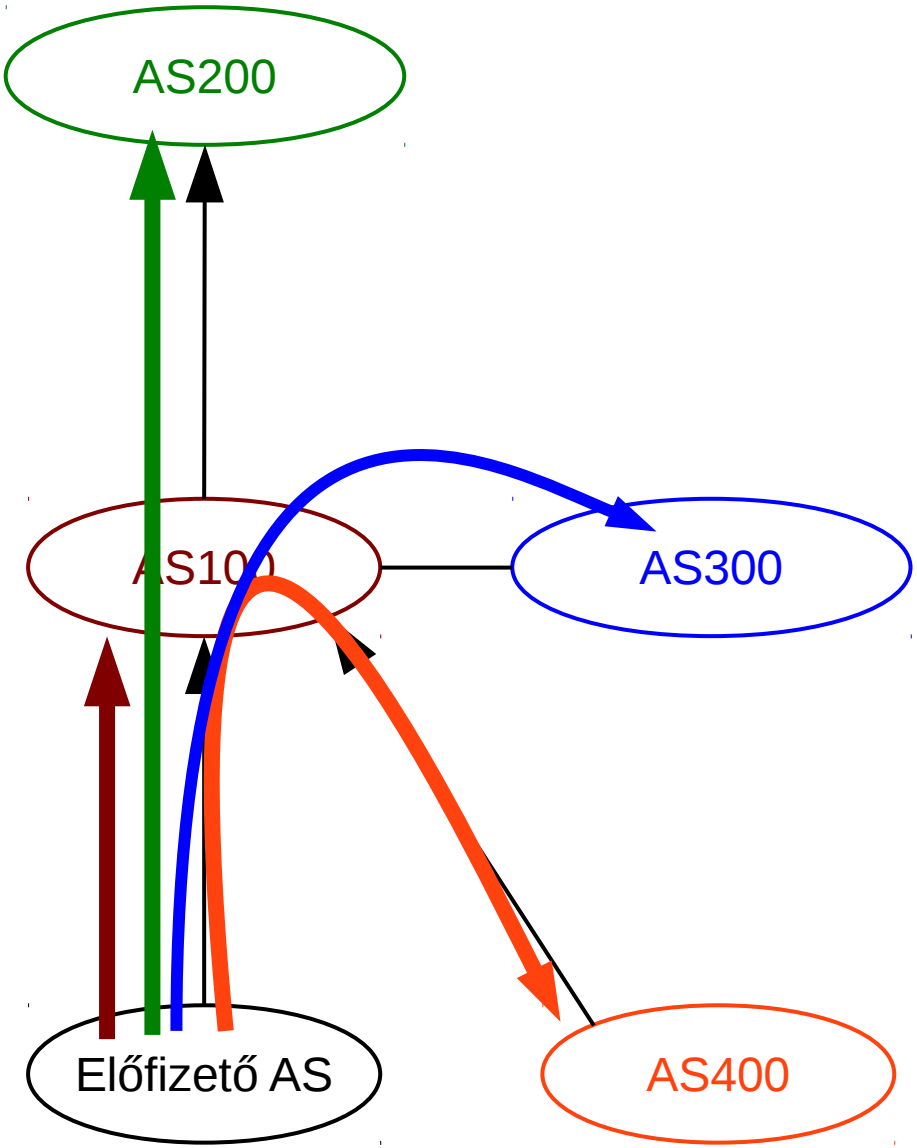
Szűrés előfizető felé

- `AS100` útvonalai 4 csoportra oszthatók az első AS-AS linkjük típusa alapján:
  - előfizetőn keresztül
  - peer-en keresztül
  - szolgáltatón keresztül
  - saját prefixre vonatkozó
- Melyeket exportálja az `Előfizető AS` számára?



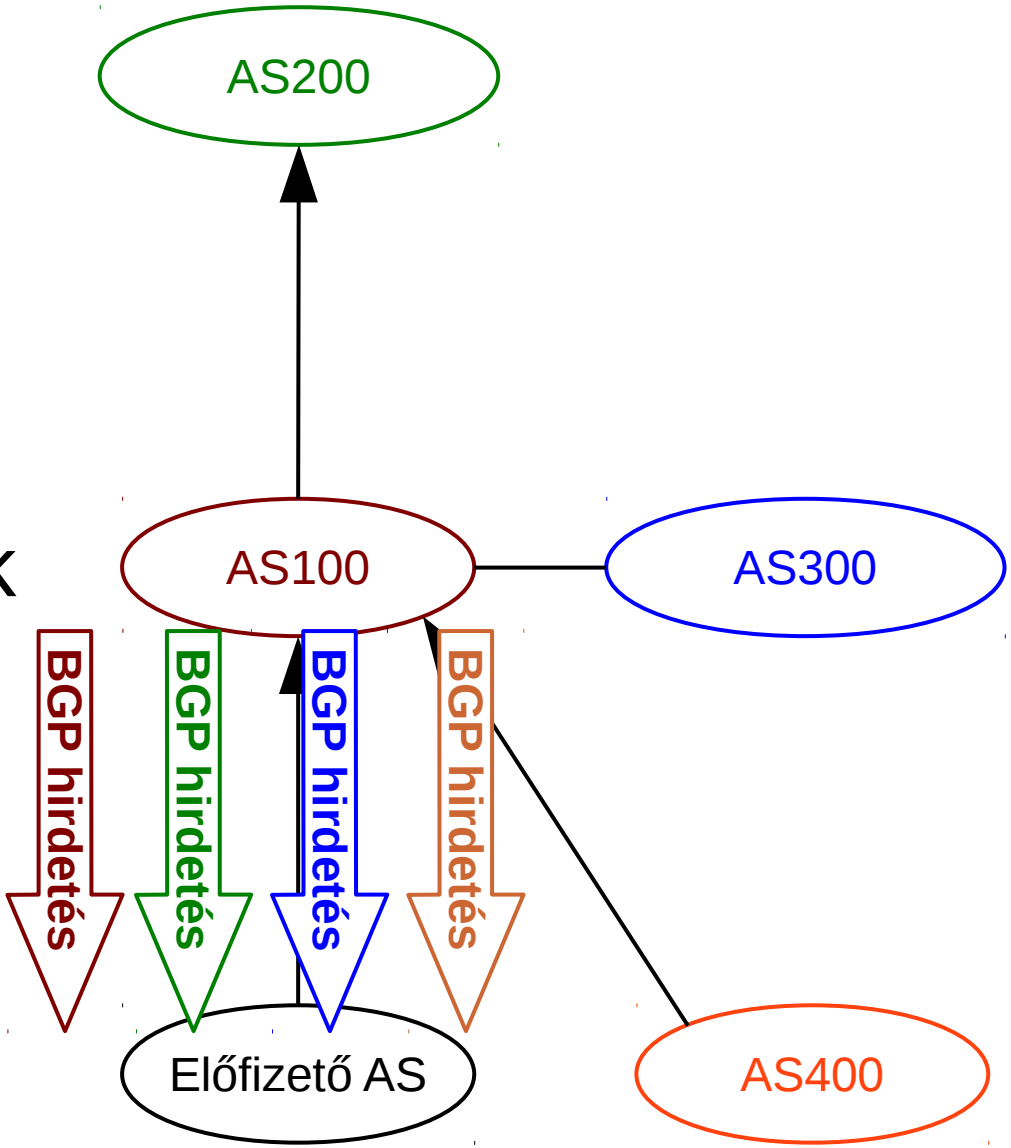
Szűrés előfizető felé

- Kaphatunk-e tiltott utat, ha `AS100` útjait kiegészítjük `Előfizető AS` felé?
- **Észrevétel:** előfizető AS felől érkező minden út valley-free
- Az `AS100`-ba érkezők is



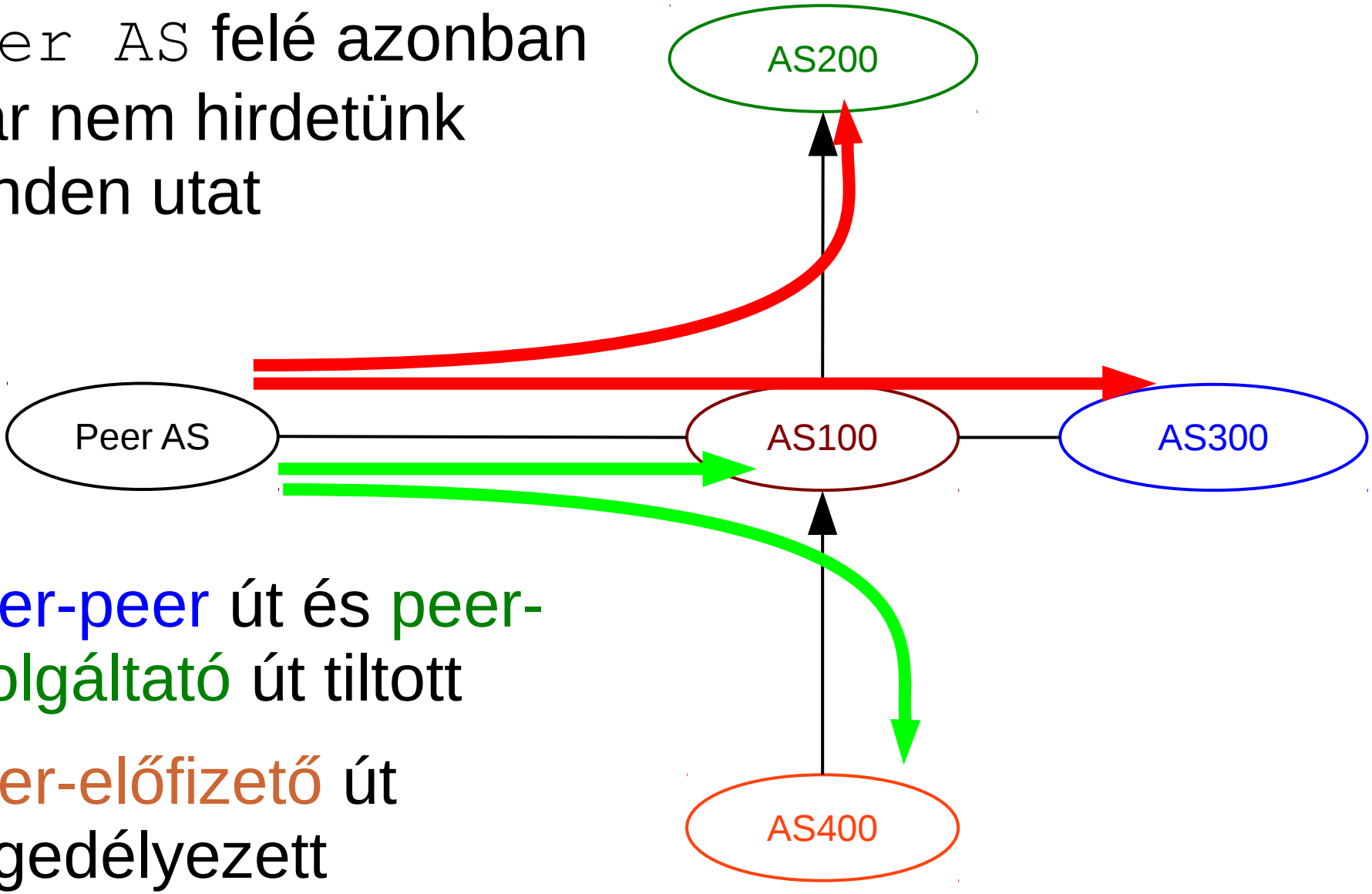
Szűrés előfizető felé

- **Szabály: előfizető AS felé minden útvonal exportálható**
- Szolgáltatóktól, peer-ektől és más előfizetőktől kapott utak
- `AS100` a saját prefixeit is meghirdeti az előfizetőinek
- Előfizető felé nincs szűrés!



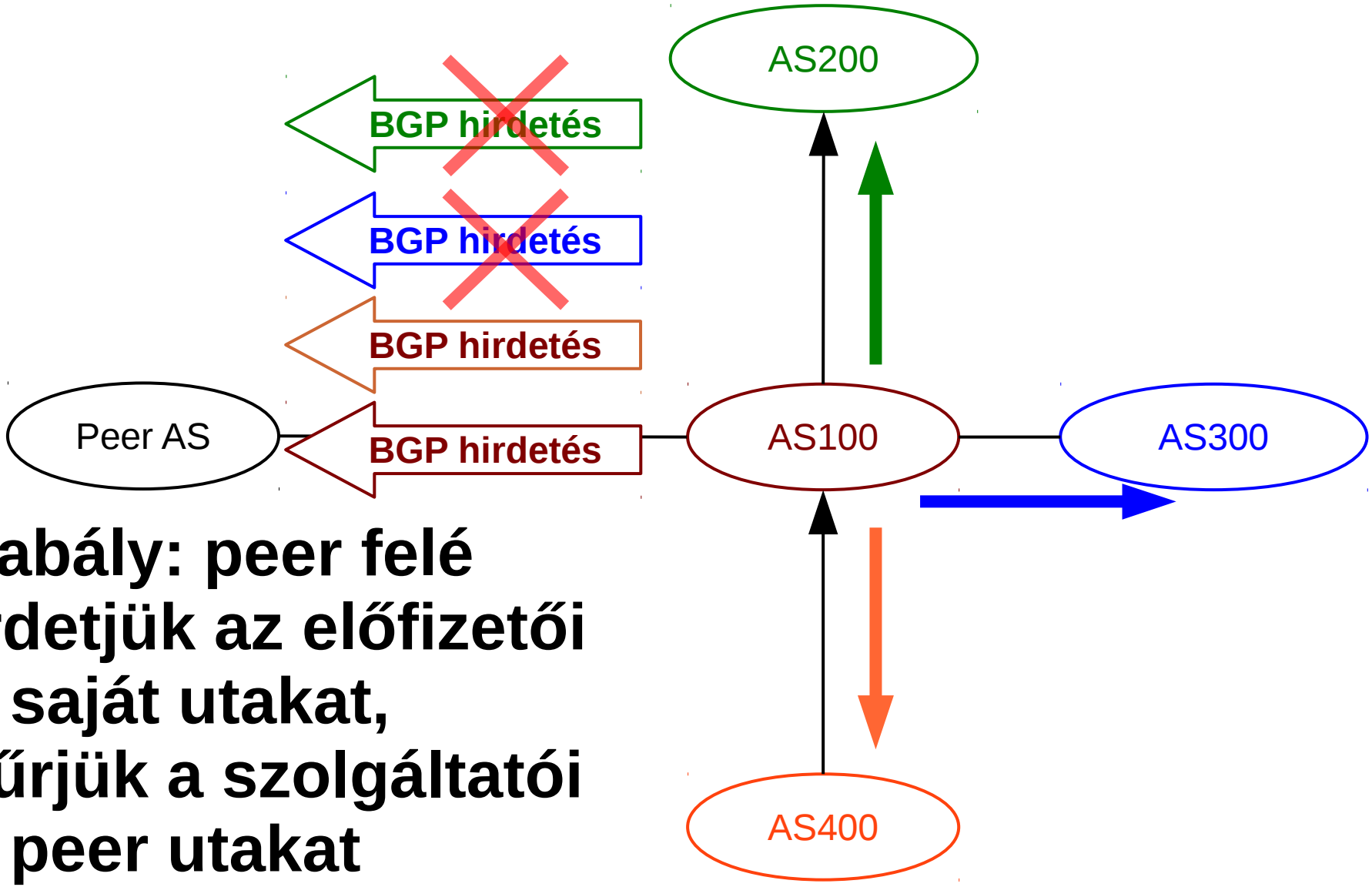
Szűrés peer felé

- `Peer AS` felé azonban már nem hirdetünk minden utat



- Peer-peer út és peer-szolgáltató út tiltott
- Peer-előfizető út engedélyezett

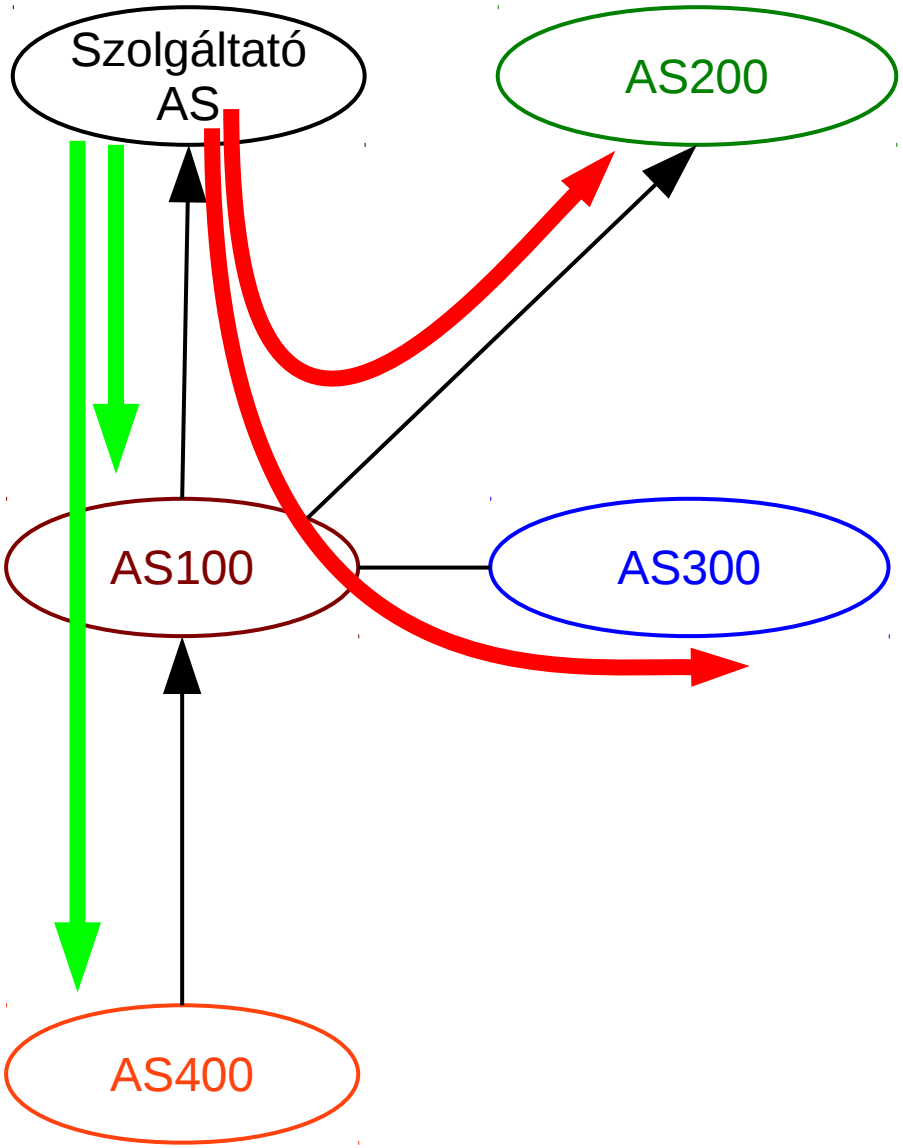
Szűrés peer felé



- **Szabály: peer felé hirdetjük az előfizetői és saját utakat, szűrjük a szolgáltatói és peer utakat**

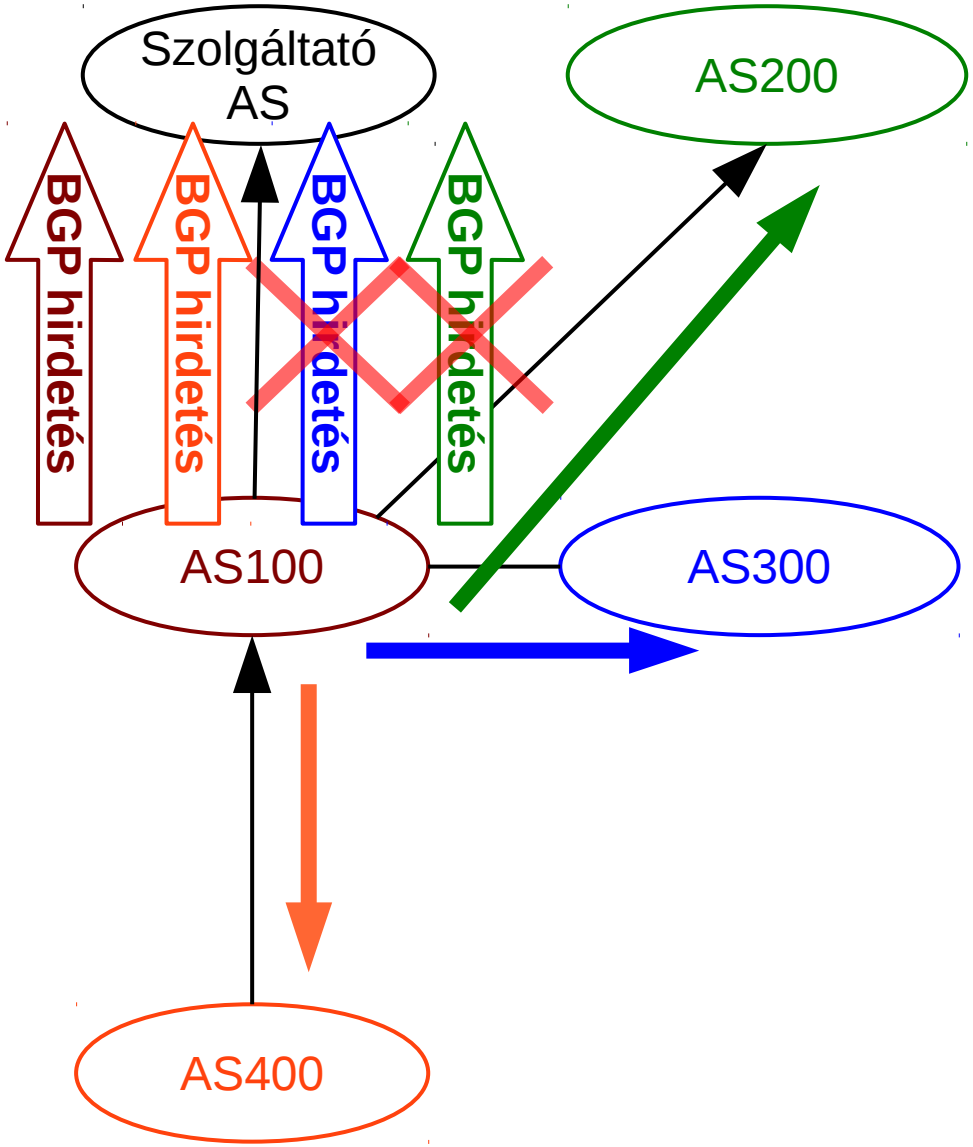
Szűrés szolgáltató felé

- `AS100` a szolgáltatótól induló útvonalak között is szelektál
- Nem visz át forgalmat két szolgáltató között, és tiltott a szolgáltató→peer útvonal is
- De a szolgáltató → előfizető és a szolgáltató→`AS100` utak megengedettek

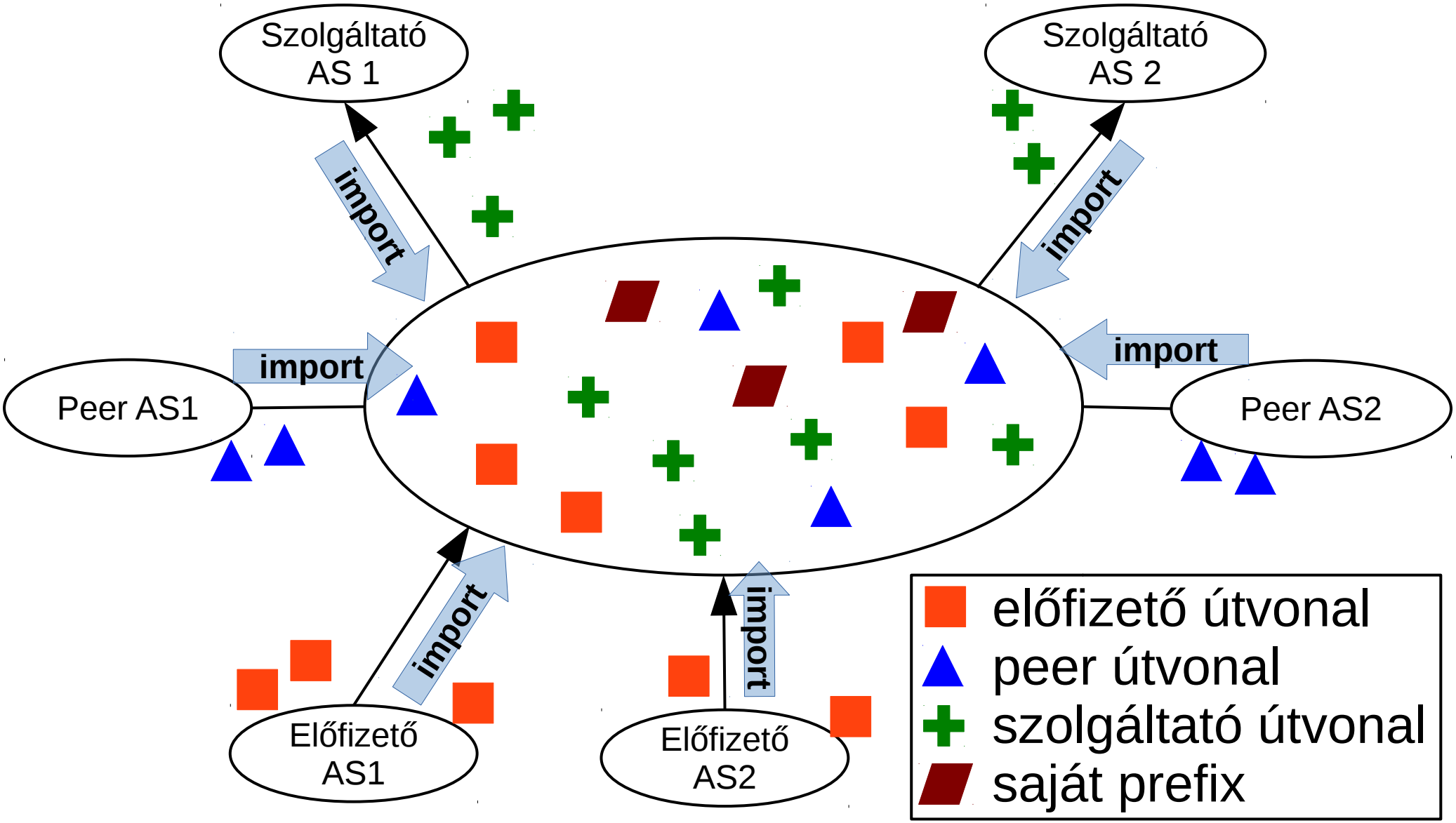


Szűrés szolgáltató felé

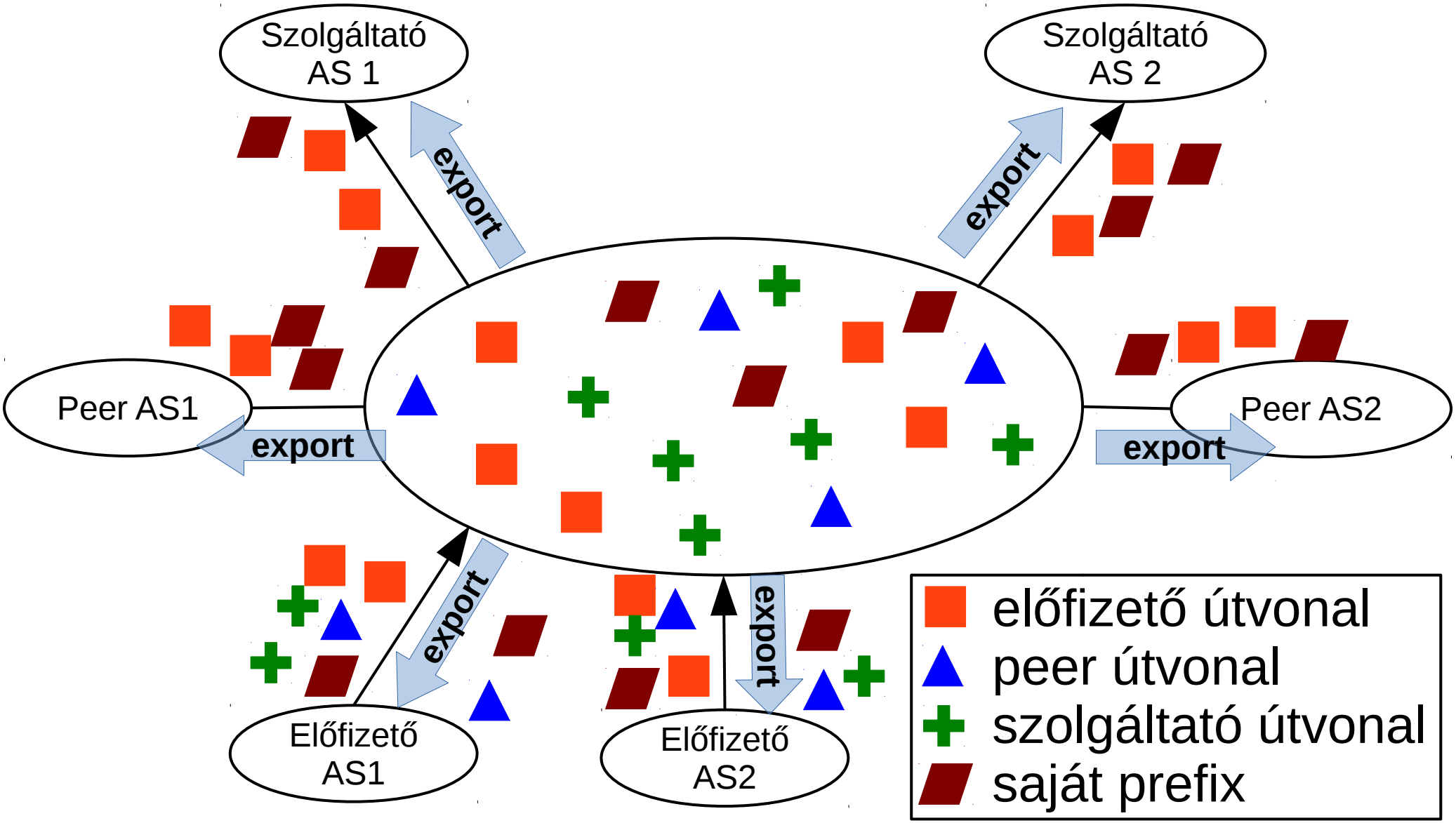
- **Szabály: szolgáltató felé hirdetjük az előfizetői és saját utakat, szűrjük a más szolgáltatókon és peer-eken áthaladó útvonalakat**
- Ugyanaz a szűrési feltétel, mint a peer esetében



Valley-free routing: import



Valley-free routing: export



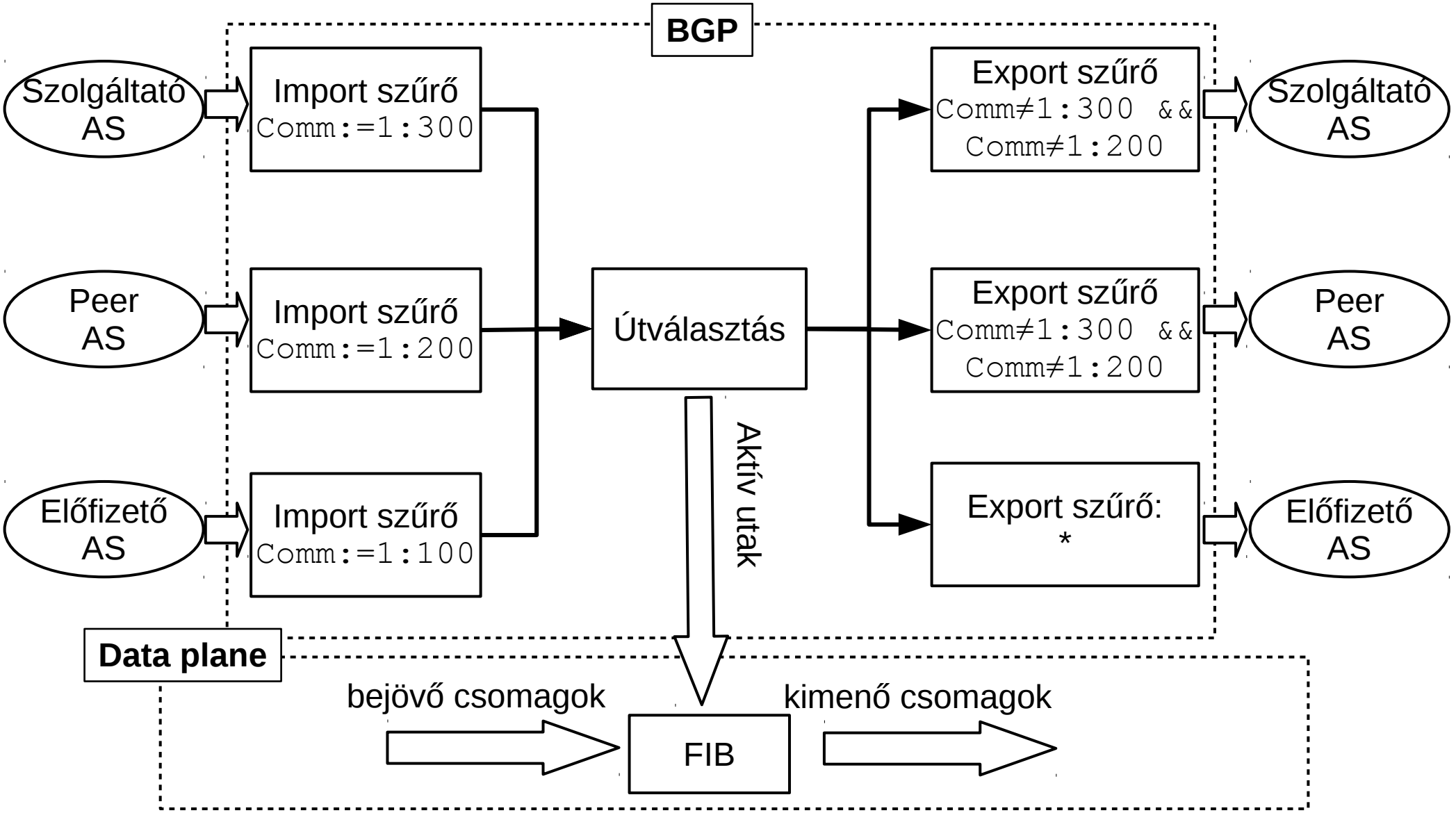
Valley-free routing: BGP konfig

- A fenti logikát le kell képeznünk a BGP konfiguráció nyelvére
- A kapott hirdetéseket az **import szűrőkön** felcímkézzük: **BGP COMMUNITY attribútum**
- Így a hirdetés már leírja, hogy kitől kaptuk:
  - előfizetőtől kapott hirdetés: `1:100`
  - peertől kapott hirdetés: `1:200`
  - szolgáltatótól kapott hirdetés: `1:300`
  - saját prefixre vonatkozó hirdetést szűrés nékül exportálunk: nem kap címkét

Valley-free routing: BGP konfig

- Az **export szűrőkön a Community érték szerint** szűrünk
- Előfizető felé minden hirdetést átengedünk
- Peer és szolgáltató felé
  - átengedjük az előfizetőtől kapott (`Comm=1:100`) és saját prefixre vonatkozó hirdetéseket
  - szűrjük a peer-től kapott (`Comm=1:200`) és a szolgáltatótól kapott (`Comm=1:300`) hirdetéseket

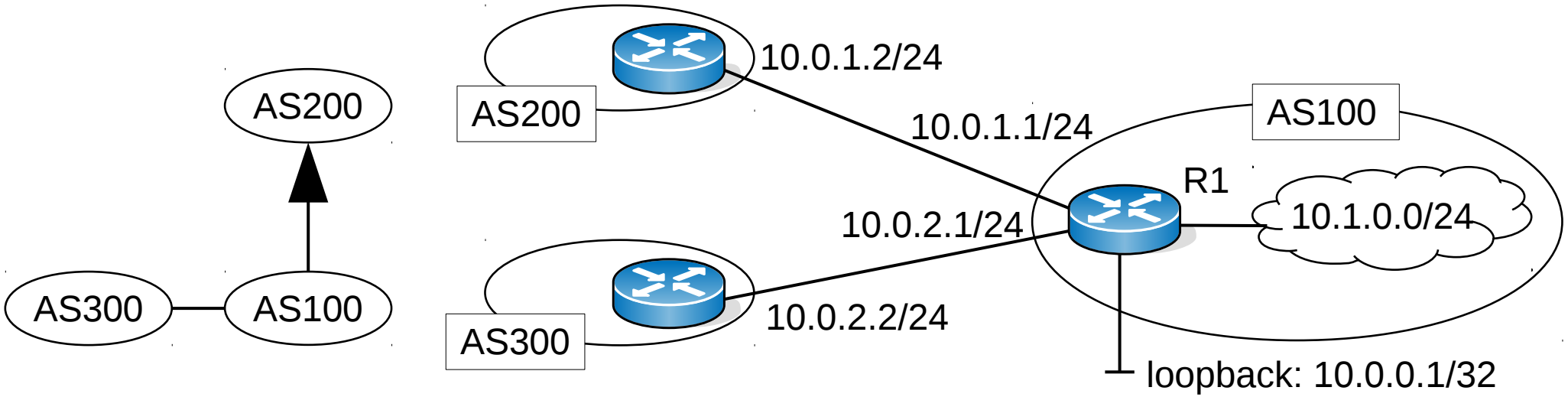
Valley-free routing: BGP konfig



Valley-free routing: import szűrő

- Tegyük most fel, hogy `AS200` szolgáltatója, `AS300` pedig peer-je `AS100`-nak
- Import szűrő az `AS300` hirdetéseinek jelölésére

```
route-map rm-peer-set-cm permit 20
  set community 1:200
```

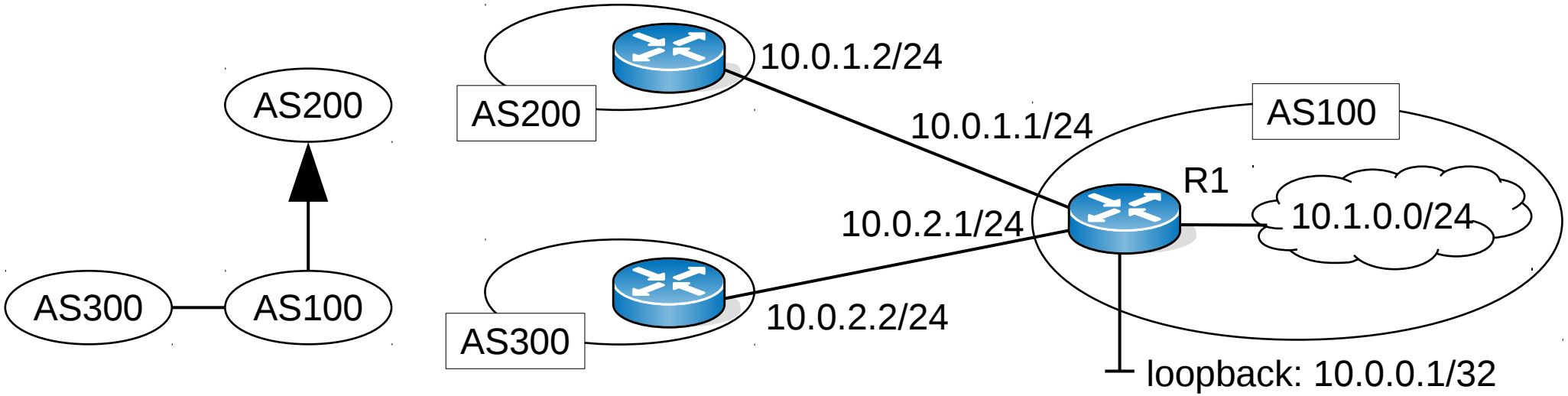


Valley-free routing: import szűrő

- A szűrőt az `AS300`-hoz rendeljük

  `neighbor 10.0.2.2 route-map rm-peer-set-cm in`

- Hasonlóan definiálunk import szűrőt `AS200` hirdetéseinek az `1:300` szolgáltatói community-vel való jelölésére



Valley-free routing: export szűrő

- `AS300` peer és `AS200` szolgáltató felé szűrni kell a szolgáltatói és a peer hirdetéseket
- Ehhez először definiáljuk a kiszűrendő community-k listáját

```
ip community-list standard cm-no-export permit 1:300
ip community-list standard cm-no-export permit 1:200
```

- Illeszkedik `(1:200 VAGY 1:300)` community-kre (mert `permit` és külön sorokból áll)
- `1:200 ÉS 1:300` community-re illeszkedő lista:

```
ip community-list standard cm-list permit 1:200 1:300
```

Valley-free routing: export szűrő

- Az első `route-map` kiszűri az `1:200` és `1:300` community-ket tartalmazó hirdetéseket

```
route-map rm-no-export deny 10
  match community cm-no-export
```

- `deny`: az illeszkedő hirdetések eldobódnak
- A nem illeszkedő hirdetések is eldobódnának (a route-map-ek alapértelmezett kimenete `deny`)
- Kell egy üres, az elsőnél alacsonyabb prioritású `route-map`, ami mindent átenged

```
route-map rm-no-export permit 20
```

Valley-free routing: példa

```
!!! BGP router konfigurációja
!!! Comunity-k:
!!!    1:100: előfizető
!!!    1:200: peer
!!!    1:300: szolgáltató
router bgp 100
 bgp router-id 10.0.0.1
 network 10.1.0.0/24
 neighbor 10.0.1.2 remote-as 200
 neighbor 10.0.1.2 route-map rm-prov-set-cm in
 neighbor 10.0.1.2 route-map rm-no-export out
 neighbor 10.0.2.2 remote-as 300
 neighbor 10.0.2.2 route-map rm-peer-set-cm in
 neighbor 10.0.2.2 route-map rm-no-export out

!!! folytatás a következő oldalon
```

Valley-free routing: példa

```
!!! import szűrők
route-map rm-prov-set-cm permit 10
 set community 1:300

route-map rm-peer-set-cm permit 10
 set community 1:200

route-map rm-cust-set-cm permit 10
 set community 1:100

!!! export szűrők
ip community-list standard cm-no-export permit 1:200
ip community-list standard cm-no-export permit 1:300

route-map rm-no-export deny 10
 match community cm-no-export

route-map rm-no-export permit 20
```