# RIP és RIPng forgalomirányítás

## A RIP protokoll konfigurálása

Amint az 1. ábrán látható, a RIPv2 alapértelmezés szerint automatikusan összevonja a hálózatokat a fő hálózati határok mentén, csakúgy mint a RIPv1.

A RIPv2 automatikus útvonal-összevonásra vonatkozó alapértelmezett beállításának megváltoztatásához használjuk a **no auto-summary** forgalomirányító-konfigurációs módbeli parancsot a 2. ábrán látható módon. RIPv1 használatakor ennek a parancsnak nincs hatása. Az automatikus útvonal összegzés letiltásakor a RIPv2 nem vonja össze a hálózatokat az osztály alapú határokon. Az útvonalfrissítések így az összes alhálózatot és a megfelelő maszkokat tartalmazzák. A **show ip protocols** kimenetében most ez látható automatic network summarization is not in effect.

**MEGJEGYZÉS**: Az automatikus útvonal összegzés letiltása előtt engedélyezni kell a RIPv2-t.

A 3. ábrán található parancsszimulátorban tiltsuk le az automatikus útvonal-összevonást az R2 és R3 forgalomirányítón!

# RIP és RIPng forgalomirányítás

## A RIP protokoll konfigurálása

Alapértelmezés szerint a RIP frissítések minden engedélyezett interfészen továbbításra kerülnek. Ezeket a frissítéseket azonban ténylegesen csak azokon az interfészeken kellene kiküldeni amelyekhez RIP szomszédok kapcsolódnak.

Tekintsük meg például az 1. ábrán látható topológiát. A RIP annak ellenére küld frissítést a G 0/0 interfészén, hogy azon a hálózaton nincs RIP-eszköz. R1-nek ugyanis erről nincsen tudomása, így 30 másodpercenként elküld egy frissítést. A szükségtelen frissítések helyi hálózatokba történő kiküldése háromféle hatással lehet a hálózatra:

* **Sávszélesség pazarlás** - A szükségtelen frissítések továbbításához is sávszélességre van szükség. Mivel a RIP frissítések szórással vagy csoportos küldéssel kerülnek továbbításra, így a kapcsolók minden portjukon kiküldik őket.
* **Erőforrás pazarlás** - A helyi hálózat minden állomásának fel kell dolgoznia a frissítést a szállítási rétegig, és csak ezután tudja eldobni azt.
* **Biztonsági kockázat** - A frissítések szórásos hálózatban történő hirdetése biztonsági kockázatot jelent. A RIP frissítések csomaglehallgató szoftverek segítségével elfoghatók. Az útvonalfrissítések módosításával és azok forgalomirányítóhoz történő visszaküldésével az irányítótáblába hamis mértékek kerülhetnek, ami hibás forgalomirányításhoz vezethet.

A **passive-interface** forgalomirányító-konfigurációs módbeli paranccsal megakadályozható, hogy a forgalomirányító egy adott interfészen keresztül útvonalfrissítéseket küldjön ki, a többi forgalomirányító felé viszont továbbra is lehetőség legyen a hálózat hirdetésére. A parancs leállítja az útvonalfrissítések kiküldését a megadott interfészen, de az interfészhez tartozó hálózatot a többi interfészen kiküldött útvonalfrissítések továbbra is hirdetik.

R1, R2 és R3 helyi hálózati interfészein nincs szükség a RIP frissítések továbbítására. A 2. ábrán lévő konfiguráció az R1 forgalomirányító G0/0 interfészét passzívra állítja. A **show ip protocols** paranccsal ellenőrizhető a Gigabit Ethernet interfész passzív állapota. Figyeljük meg, hogy a G0/0 interfész már nem szerepel a 2-es verziót küldő vagy fogadó interfészek között, hanem passzív interfészként (Passive Interface) jelenik meg. Azt is vegyük észre, hogy a 192.168.1.0 hálózat továbbra is a hirdetett hálózatok (Routing for Networks) között van, azaz az R2-nek küldött RIP frissítésekben útvonal bejegyzésként szerepel.

**MEGJEGYZÉS**: Az irányító protokollok mindegyike támogatja a **passive-interface** parancsot.

A 3. ábrán található parancsszimulátorban állítsuk passzívra az R2 és R3 LAN interfészeit!

Egy másik lehetőség, hogy először minden interfészt passzívra állítunk a **passive-interface default** paranccsal. Ezután azokat az interfészeket, amelyeket nem akarunk passzívvá tenni, újra engedélyezhetjük a **no passive-interface** paranccsal.

# RIP és RIPng forgalomirányítás

## A RIP protokoll konfigurálása

Tanulmányozzuk az 1. ábrát! Látható, hogy mivel R1 egyetlen szolgáltatóhoz kapcsolódik, így az internet eléréséhez mindössze egy alapértelmezett statikus útvonalra van szüksége a Serial 0/0/1 interfészen kifelé.

Az R2 és R3 forgalomirányítókon is hasonló alapértelmezett útvonalakat kellene beállítani. Mindez azonban sokkal jobban kezelhető akkor, ha az útvonalat csak egyszer adjuk meg az R1 határ forgalomirányítón, és R1 tovább hirdeti azt a többi forgalomirányítónak a RIP segítségével. Annak érdekében, hogy az internet kapcsolat biztosítva legyen a RIP irányítási tartomány összes hálózatában, a RIP irányító protokollt futtató többi forgalomirányító számára is hirdetni kell az alapértelmezett statikus útvonalat.

Az alapértelmezett útvonal hirdetéséhez a határ forgalomirányítón a következőket kell elvégezni:

* Egy alapértelmezett statikus útvonal létrehozása az **ip route 0.0.0.0 0.0.0.0** *kimenő-interfész következő-ugrás-IP* paranccsal.
* A **default-information originate** forgalomirányító-konfigurációs parancs kiadása. Ez arra utasítja az R1 forgalomirányítót, hogy terjessze az alapértelmezett információkat úgy, hogy a RIP frissítésekben az alapértelmezett útvonalat is hirdeti.

A 2. ábra példájában az látható, hogy egy teljesen meghatározott alapértelmezett statikus útvonal van beállítva a szolgáltató felé, és ezt az útvonalat a RIP hirdeti tovább. Figyeljük meg, hogy R1 irányítótáblájában a végső átjáró (Gateway of Last Resot) és az alapértelmezett útvonal is megjelenik.

A 3. ábrán található parancsszimulátorban vizsgáljuk meg, hogy az R2 és R3 forgalomirányítókra megérkeztek-e az alapértelmezett útvonal hirdetmények!

# RIP és RIPng forgalomirányítás

## A RIPng protokoll konfigurálása

Ahogy az IPv4-es RIP protokollt, úgy a RIPng-t is ritkán használják a modern hálózatokban. A hálózati forgalomirányítás alapjainak megértéséhez viszont ez a protokoll is hasznos alapot nyújt. Éppen ezért ez a rész tartalmaz egy rövid áttekintést a RIPng alapvető beállításairól.

Nézzük meg az ábrán látható minta topológiát. A példában szereplő forgalomirányítók mindegyikén a működéshez szükséges alapvető funkciók beállítása és a referencia topológián feltüntetett minden interfész konfigurálása és engedélyezése el lett végezve. Statikus útvonalak nincsenek beállítva, és irányító protokollok nincsenek engedélyezve, emiatt a távoli hálózati hozzáférés jelenleg nem lehetséges.

Egy IPv6-forgalomirányítón az **ipv6 unicast-routing** paranccsal engedélyezhető az IPv6-csomagtovábbítás.

A RIPv2-vel ellentétben a RIPng-t nem forgalomirányító-konfigurációs módban engedélyezzük, hanem egy interfészen. Valójában a **network** *hálózati\_cím* parancs RIPng-ben nem is elérhető. Ehelyett az **ipv6 rip** *tartománynév* **enable** interfész konfigurációs parancsot használjuk.

Az 1. ábrán az látható, hogy az IPv6-csomagtovábbítás be van állítva, a Gigabit Ethernet 0/0 és a Serial 0/0/0 interfészeken pedig engedélyezett a RIPng a RIP-AS tartománynév használatával.

A 2. ábrán található parancsszimulátorban az R2 és R3 forgalomirányítókon állítsunk össze hasonló konfigurációt!

RIPng esetében az alapértelmezett útvonal hirdetésének folyamata megegyezik a RIPv2-vel azzal a különbséggel, hogy egy IPv6 alapértelmezett statikus útvonal beállítása szükséges. Tegyük fel például, hogy R1 internetkapcsolata a Serial 0/0/1 interfészen található a 2001:DB8:FEED:1::1/64 IP-cím felé. Az alapértelmezett útvonal hirdetéséhez a következőket kell elvégezni az R3 forgalomirányítón:

* Egy alapértelmezett statikus útvonal létrehozása az **ipv6 route 0::/0 2001:DB8:FEED:1::1** globális konfigurációs paranccsal.
* Az **ipv6 rip** *tartománynév* **default-information originate** interfész konfigurációs parancs kiadása. Ez arra utasítja R3-at, hogy az alapértelmezett útvonal információk forrása legyen, és RIPng frissítésekben hirdesse az alapértelmezett statikus útvonalat a konfigurált interfészeken.

# RIP és RIPng forgalomirányítás

## A RIPng protokoll konfigurálása

Az 1. ábrán látható, hogy a **show ipv6 protocols** parancs nem biztosít annyi információt, mint az IPv4 megfelelője. A következő paraméterek viszont megtekinthetők:

1. Az R1 forgalomirányítón RIPng forgalomirányítás van beállítva és futtatva.

2. A RIPng-vel konfigurált interfészek listája.

A **show ipv6 route** parancs az irányítótáblába felvett útvonalakat jeleníti meg a 2. ábrán látható módon. A kimenet azt mutatja, hogy R1 a kijelölt RIPng hálózatokat már ismeri.

Figyeljük meg, hogy az R2 LAN 2 ugrás távolsággal van hirdetve. Ennek az oka, hogy a RIPv2 és a RIPng különböző módon számolják az ugrásszámot. Az R2 LAN mértéke RIPv2 (és RIPv1) esetében 1 ugrás lenne. Ez azért van így, mert az IPv4-irányítótáblában megjelenő mérték (ugrásszám) a távoli hálózat eléréséhez szükséges ugrásszámot jelenti (a következő ugrás forgalomirányítót az első ugrásnak számolva). RIPng esetén azonban a küldő forgalomirányító már saját magát is 1 ugrás távolságúnak tekinti, így R2 emiatt hirdeti a helyi hálózatát 1-es mértékkel. Amikor R1 megkapja a frissítést, akkor eggyel megnöveli az ugrásszámot, és így R2 helyi hálózatát már 2 ugrás távolságúnak tekinti. Hasonló okok miatt van az R3 LAN 3 ugrás távolságra.

A **rip** kulcsszót a 3. ábrán látható módon a parancs után írva csak a RIPn-hálózatok listája lesz látható.

A 4. ábrán található parancsszimulátorban vizsgáljuk meg az R2 és R3 forgalomirányítót!

# RIP és RIPng forgalomirányítás

## A RIPng protokoll konfigurálása

**Ebben a laborgyakorlatban a következő feladatokat végezzük el:**

* 1. rész: A hálózat felépítése és az eszközök alapvető konfigurálása.
* 2. rész: RIPv2 forgalomirányítás konfigurálása és ellenőrzése.
* 3. rész: IPv6 beállítása az eszközökön.
* 4. rész: RIPng forgalomirányítás konfigurálása és ellenőrzése.

[Laborgyakorlat - Configuring Basic RIPv2 and RIPng](https://contenthub.netacad.com/legacy/RSE/5.02/hu/course/files/7.3.2.4%20Lab%20-%20Configuring%20Basic%20RIPv2%20and%20RIPng.pdf)

# Kapcsolatállapot alapú dinamikus forgalomirányítás

## Kapcsolatállapot alapú irányító protokollok működése

A kapcsolatállapot alapú protokollok legrövidebb út protokollként is ismertek, és Edsger Dijkstra legrövidebb út (shortest path first, SPF) algoritmusára épülnek. Az SPF-algoritmust egy későbbi fejezetben részletesebben is tárgyaljuk.

Az ábrán az IPv4-es kapcsolatállapot alapú protokollok láthatóak:

* Legrövidebb út protokoll (Open Shortest Path First, OSPF)
* Közbülső rendszerből közbülső rendszerbe protokoll (Intermediate System-to-Intermediate System, IS-IS)

A kapcsolatállapot alapú protokolloknak az a hírük, hogy jóval bonyolultabbak, mint a távolságvektor alapú társaik. Viszont ha az alapvető funkciókat és a konfigurálást tekintjük, a kapcsolatállapot alapú protokollok ugyanolyan egyszerűek.

A RIP-hez és az EIGRP-hez hasonlóan az OSPF alapvető működését is a következő parancsokkal lehet beállítani:

* **router ospf** *folyamat-azonosító* globális konfigurációs parancs
* **network** parancs a hálózatok hirdetésére

# Kapcsolatállapot alapú dinamikus forgalomirányítás

## Kapcsolatállapot alapú irányító protokollok működése

Minden kapcsolatállapot alapú protokoll a Dijkstra algoritmust használja a legjobb útvonal kiszámításához. Az algoritmus ismertebb neve a legrövidebb utat kereső (SPF) algoritmus. Az útvonal teljes költségének meghatározásához az algoritmus a forrás és cél közötti utak költségeit összesíti.

Az ábrán minden egyes úthoz egy tetszőleges költség érték van hozzárendelve. R2-től az R3 helyi hálózatába tartó csomagok számára a legrövidebb útvonal költsége 27. Minden forgalomirányító saját költséget határoz meg a topológia összes célhálózata felé. Más szóval, a forgalomirányítók mindegyike lefuttatja az SPF-algoritmust, majd a saját szemszögéből meghatározza a költségeket.

**MEGJEGYZÉS**: A fejezet középpontjában a SPF-fa által meghatározott költség áll. Az ábrák emiatt az SPF-fa kapcsolatait mutatják és nem a topológiát. A kapcsolatokat folytonos fekete vonalak jelölik.

# Kapcsolatállapot alapú dinamikus forgalomirányítás

## Kapcsolatállapot alapú irányító protokollok működése

Az 1. ábrán található táblázat R1 szemszögéből mutatja a célhálózatok felé vezető legrövidebb útvonalakat és a hozzájuk tartozó összesített költségeket.

A legrövidebb út nem feltétlenül a legkevesebb ugrásból áll. Nézzük meg például az R5 LAN felé vezető utat. Feltételezhető lenne, hogy R1 közvetlenül R4-nek küldi a csomagot R3 helyett. Azonban R4 közvetlen elérésének magasabb a költsége (22), mint ha R3-on keresztül (17) tennénk meg ugyanezt.

A 2.-5. ábrákon a helyi hálózatokba vezető legrövidebb utak és költségek láthatók az egyes forgalomirányítók szemszögéből.

# Kapcsolatállapot alapú dinamikus forgalomirányítás

## Kapcsolatállapot-frissítések

Ezen interfészek állapot-információit nevezzük kapcsolatállapotnak.

Vizsgáljuk meg a topológiát az ábrán! A topológia minden forgalomirányítója a konvergencia elérése érdekében a következő általános kapcsolatállapot alapú forgalomirányítási folyamatot hajtja végre:

1. A forgalomirányítók megismerik a kapcsolataikat és a közvetlenül csatlakozó hálózataikat. Ehhez ellenőrzik, hogy interfészeik felkapcsolt állapotban vannak-e.

2. Minden forgalomirányító felderíti saját szomszédait a közvetlenül csatlakozó hálózatokon. A szomszédos kapcsolatállapot alapú forgalomirányítók mindezt hello csomagok cseréjével valósítják meg.

3. A forgalomirányítók egy kapcsolatállapot-csomagot (Link-State Packet, LSP) állítanak össze, amelyek a közvetlen kapcsolatok állapotait tartalmazzák. Ez úgy történik, hogy minden lényeges információt, például a szomszéd azonosítóját, a kapcsolat típusát és a sávszélességet is rögzítik a szomszédokról.

4. A forgalomirányítók elárasztják az LSP-vel minden szomszédjukat. A szomszédok a kapott LSP-ket egy adatbázisban tárolják el. Ezután ők is elárasztják a saját szomszédaikat az LSP-kkel, és ez a folyamat addig tart, amíg a terület összes forgalomirányítója meg nem kapja az LSP-ket. A forgalomirányítók egy helyi adatbázisban tárolják a szomszédoktól beérkezett LSP-k egy-egy példányát.

5. Az adatbázist felhasználva a forgalomirányítók egy topológia térképet hoznak létre, majd kiszámolják a legjobb útvonalakat a célhálózatok felé. Hasonlóan a mi autóstérképünkhöz, most már a forgalomirányítók is rendelkeznek egy térképpel a topológia összes célhálózatáról és a hozzájuk vezető utakról. Az SPF algoritmus építi fel a topológia térképet és határozza meg a legjobb útvonalakat a célhálózatok felé.

**MEGJEGYZÉS**: Ez a folyamat megegyezik az OSPF IPv4 és IPv6 változatában is. A példák ebben a részben az IPv4-es változatra vonatkoztak.

# Kapcsolatállapot alapú dinamikus forgalomirányítás

## Kapcsolatállapot-frissítések

A kapcsolatállapot alapú forgalomirányítás folyamatának első lépése, hogy a forgalomirányítók megismerik saját kapcsolataikat és közvetlenül csatlakozó hálózataikat. Ha egy interfészhez IP-címet és alhálózati maszkot rendelünk, akkor az interfész a hálózat részévé válik.

Nézzük meg az 1. ábrán látható topológiát! Tegyük fel, hogy R1 korábban már konfigurálva lett és teljes értékű kapcsolatokkal rendelkezett a szomszédai felé. Amikor azonban egy rövid időre megszűnt az áramellátás, R1-et újra kellett indítani.

Az indulás során R1 betölti az elmentett indító konfigurációs fájlt. Ahogy a korábban beállított interfészek aktívvá válnak, R1 úgy ismeri meg a közvetlenül csatlakozó hálózatait. A használt irányító protokolltól függetlenül ezek a közvetlenül csatlakozó hálózatok egy-egy bejegyzésként jelennek meg az irányítótáblában.

Ahhoz, hogy egy kapcsolatállapot alapú protokoll ismereteket tudjon szerezni egy összeköttetésről, a távolságvektor alapú protokollokhoz és a statikus útvonalakhoz hasonlóan az interfész IPv4-címét és alhálózati maszkját megfelelően be kell állítani, és az összeköttetésnek felkapcsolt állapotúnak kell lennie. Mindemellett a távolságvektor alapú protokollokhoz hasonlóan az interfész csak akkor vesz részt a kapcsolatállapot alapú forgalomirányítási folyamatban, ha szerepel egy **network** forgalomirányító konfigurációs utasításban.

Az ábrán látható, hogy R1 az alábbi négy közvetlenül csatlakozó hálózattal rendelkezik:

* FastEthernet 0/0 - 10.1.0.0/16
* Serial 0/0/0 - 10.2.0.0/16
* Serial 0/0/1 - 10.3.0.0/16
* Serial 0/1/0 - 10.4.0.0/16

Ahogy a 2.-5. ábrákon is látható, a kapcsolatállapot-információk a következőket tartalmazzák:

* Az interfészek IPv4-címét és alhálózati maszkját
* A hálózat típusát, mint például Ethernet (szórásos) vagy soros pont-pont kapcsolat
* Az összeköttetés költségét
* A linkhez kapcsolódó összes szomszédos forgalomirányítót

**MEGJEGYZÉS**: A Cisco implementációban az OSPF-útvonal mértéke a kapcsolat kimenő interfészének sávszélességén alapuló költség. Ebben a fejezetben a szemléltetés egyszerűsége érdekében tetszőleges költség értékeket használunk.

# Kapcsolatállapot alapú dinamikus forgalomirányítás

## Kapcsolatállapot-frissítések

A kapcsolatállapot alapú forgalomirányítás folyamatának második lépése, hogy a forgalomirányítók megismerik a szomszédaikat a közvetlenül csatlakozó hálózataikon keresztül.

A kapcsolatállapot alapú forgalomirányítók egy úgynevezett Hello protokollt használnak a szomszédok felderítésére. Szomszéd bármely olyan forgalomirányító lehet, amelyen ugyanaz a kapcsolatállapot alapú irányító protokoll engedélyezett.

Kattintsunk az ábra Lejátszás gombjára az animáció megtekintéséhez, amely a kapcsolatállapot alapú szomszédsági viszony Hello csomagokkal történő kialakítását mutatja be.

Az animációban R1 Hello csomagokat küld ki a kapcsolatain (interfészein), hogy felderítse a szomszédait. Mivel R2, R3 és R4 forgalomirányítón ugyanaz a kapcsolatállapot alapú protokoll fut, ezért a Hello csomagokra válaszul elküldik a sajátjukat. A FastEthernet 0/0 interfész irányában nem találhatók szomszédok. Mivel R1 ezen az interfészen nem kapott vissza Hello csomagot, itt nem folytatja tovább a kapcsolatállapot alapú irányítási folyamat lépéseit.

Amikor két kapcsolatállapot alapú forgalomirányító megtudja egymásról, hogy szomszédok, akkor szomszédsági viszonyt alakítanak ki egymással. A kisméretű Hello csomagok cseréje ezután is tovább folytatódik a szomszédok között, és a szomszéd állapotának megfigyelésével a viszony fenntartását látja el . Ha egy szomszéd felől nem érkeznek a Hello csomagok, akkor a forgalomirányító elérhetetlennek minősíti a szomszédot és a viszony megszűnik közöttük.

# Kapcsolatállapot alapú dinamikus forgalomirányítás

## Kapcsolatállapot-frissítések

A kapcsolatállapot alapú forgalomirányítás folyamatának harmadik lépése, hogy a forgalomirányítók egy kapcsolatállapot-csomagot (LSP) állítanak össze, amely a közvetlen kapcsolataik állapotát tartalmazza.

A forgalomirányítók a szomszédsági viszonyaik kialakítása után tudják összeállítani a kapcsolatállapot-információkat tartalmazó LSP-ket. Az ábrán egy R1-től származó LSP látható, amelynek egyszerűsített változata a következőket tartalmazhatja:

1. R1; Ethernet hálózat 10.1.0.0/16; költség 2

2. R1 -> R2; soros pont-pont hálózat; 10.2.0.0/16; költség 20

3. R1 -> R3; soros pont-pont hálózat; 10.3.0.0/16; költség 5

4. R1 -> R4; soros pont-pont hálózat; 10.4.0.0/16; költség 20

# Kapcsolatállapot alapú dinamikus forgalomirányítás

## Kapcsolatállapot-frissítések

A kapcsolatállapot alapú forgalomirányítás folyamatának negyedik lépése, hogy a forgalomirányítók elárasztják az LSP-vel a szomszédaikat, akik az összes fogadott LSP-t egy adatbázisban tárolják.

A forgalomirányítók az irányítási területen belül található összes kapcsolatállapot alapú forgalomirányítót árasztják el a kapcsolatállapot-információkkal. Amikor egy forgalomirányító LSP-t kap egy szomszédtól, azonnal továbbküldi azt minden interfészén, kivéve azon amelyiken az LSP érkezett. Ez a folyamat hozza létre az LSP elárasztást a forgalomirányítóktól kiinduló az egész irányítási területen belül.

Kattintsunk az ábra Lejátszás gombjára az LSP-elárasztás animációjának megtekintéséhez.

Figyeljük meg az animációban, hogy az LSP-k a beérkezésük után, mindenféle köztes számítás nélkül szinte azonnal szétáradnak. A kapcsolatállapot alapú protokollok az elárasztás után futtatják le az SPF-algoritmust. Ennek köszönhetően tudják nagyon gyorsan elérni a konvergenciát.

Ne feledjük, hogy az LSP-ket nem kell szabályos időközönként küldeni. Elküldésükre csak az alábbi esetekben van szükség:

* Az irányító protokoll működési folyamatának elindulásakor egy forgalomirányítón (például a forgalomirányító újraindításakor).
* A topológia bármilyen megváltozásakor (például egy kapcsolat elindulása vagy leállása, illetve a szomszédsági viszony létrejötte vagy megszűnése esetén)

Az elárasztási folyamat kezelése érdekében az LSP-ben a kapcsolatállapot-információk mellett egyéb információk is megtalálhatók, például sorszámok és élettartamra vonatkozó adatok. Ezeket az információkat a forgalomirányítók annak megállapítására használják fel, hogy az LSP megérkezett-e egy másik forgalomirányítótól, illetve, hogy az LSP tartalmaz-e újabb információkat ahhoz képest, mint ami a kapcsolatállapot-adatbázisban már szerepel. Ez a folyamat teszi lehetővé a forgalomirányító számára, hogy csak a legfrissebb információkat tárolja a kapcsolatállapot-adatbázisban.

# Kapcsolatállapot alapú dinamikus forgalomirányítás

## Kapcsolatállapot-frissítések

A kapcsolatállapot alapú forgalomirányítási folyamat utolsó lépése, hogy az adatbázist felhasználva a forgalomirányítók egy topológia térképet hoznak létre, majd kiszámolják a legjobb útvonalakat a célhálózatok felé.

Végül minden forgalomirányító kap egy LSP-t az irányítási terület összes többi kapcsolatállapot alapú forgalomirányítójától. Ezeket az LSP-ket a kapcsolatállapot-adatbázisban tárolják el.

Az ábrán található példa R1 kapcsolatállapot-adatbázisának tartalmát mutatja.

Az elárasztási folyamat eredményeként R1 megtanulta az irányítási területén található többi forgalomirányító kapcsolatállapot-információját . Figyeljük meg, hogy R1 kapcsolatállapot-adatbázisában a saját kapcsolatállapot-információi is megtalálhatók.

Ezt a teljes értékű kapcsolatállapot-adatbázist és a legrövidebb út algoritmust (SPF) már felhasználhatja R1 ahhoz, hogy kiszámítsa a kívánt vagy legrövidebb útvonalat az SPF-fában található bármely hálózat felé.

# Kapcsolatállapot alapú dinamikus forgalomirányítás

## Kapcsolatállapot-frissítések

Az irányítási terület minden forgalomirányítója a kapcsolatállapot-adatbázist és az SPF-algoritmust használja fel az SPF- fa felépítésére.

Például R1 a többi forgalomirányítótól származó kapcsolatállapot-információkat felhasználva most már nekikezdhet a hálózat SPF-fájának felépítéséhez. Elsőként az SPF-algoritmus értelmezi minden forgalomirányító LSP-jét, hogy azonosítani tudja a hálózatokat és a hozzájuk rendelt költségeket.

Az 1. ábrán az R1 közvetlenül csatlakozó hálózatai és azok költségei vannak kiemelve.

A 2.- 5. ábrák azt mutatják, hogy miként kerülnek a további ismeretlen hálózatok és azok költségei az R1 forgalomirányító SPF-fájába. Figyeljük meg, hogy R1 figyelmen kívül hagyja azokat a hálózatokat, amelyeket korábban már azonosított.

Az SPF-algoritmus ezután kiszámolja az egyes hálózatok eléréséhez szükséges legrövidebb útvonalakat, és létrejön a 6. ábrán látható SPF-fa . R1 most már egy teljes topológia térképpel rendelkezik a kapcsolatállapot-területről.

A forgalomirányítók egymástól függetlenül építik fel a saját SPF-fájukat, de a megfelelő forgalomirányítás érdekében a fák felépítéséhez használt kapcsolatállapot-adatbázisoknak minden forgalomirányítón meg kell egyezniük.

# Kapcsolatállapot alapú dinamikus forgalomirányítás

## Kapcsolatállapot-frissítések

Az SPF-algoritmus által meghatározott legrövidebb útvonalakra vonatkozó információkat felhasználva, ezek az útvonalak most már hozzáadhatók az irányítótáblához. Az ábra azokat az útvonalakat mutatja, amelyeket R1 adott hozzá az IPv4-irányítótáblájához.

Az irányítótáblában ezenkívül közvetlenül csatlakozó hálózatok és egyéb forrásokból származó hálózatok, például statikus útvonalak is megtalálhatók. A csomagok továbbítása most már az irányítótábla ezen bejegyzései alapján történik.

# Kapcsolatállapot alapú dinamikus forgalomirányítás

## Miért használjunk kapcsolatállapot alapú forgalomirányító protokollokat?

Ahogy az ábrán is látható, a kapcsolatállapot alapú irányító protokollok számos előnnyel rendelkeznek a távolságvektor alapú protokollokhoz képest.

* **Topológia térkép felépítése** - A kapcsolatállapot alapú protokollok egy topológia térképet vagy SPF-fát hoznak létre a hálózatról. Az SPF-algoritmus a kapcsolatállapot protokollok közt zajló kapcsolatállapot-információ csere alapján tudja a hálózat SPF-fáját felépíteni. A forgalomirányítók az SPF-fa segítségével egymástól függetlenül határozzák meg minden hálózat felé a legrövidebb útvonalat.
* **Gyors konvergencia** - Amikor egy kapcsolatállapot alapú irányító protokoll egy LSP-t kap, akkor azonnal továbbküldi azt minden interfészén, kivéve azon, amelyiken az LSP érkezett. A RIP ezzel szemben minden egyes útvonalfrissítést feldolgoz, és csak az irányítótábla frissítése után küldi tovább a többi interfészén.
* **Eseményvezérelt frissítések** - Az LSP-k kezdeti elárasztása után a kapcsolatállapot alapú protokollok csak topológiaváltozás esetén küldenek ki újra LSP-t. Ez az LSP csak az érintett kapcsolatra vonatkozó információkat tartalmazza. Néhány távolságvektor alapú protokollal ellentétben a kapcsolatállapot alapú protokollok nem küldenek rendszeres frissítéseket.
* **Hierarchikus tervezés** - A kapcsolatállapot alapú protokollok területi alapon működnek. Több terület használata hierarchikus hálózattervezést tesz lehetővé, és így hatékonyabb útvonal-összevonásra és az irányítási feladatok egyetlen területre szűkítésére van lehetőség.

A kapcsolatállapot alapú protokollok néhány hátránnyal is rendelkeznek a távolságvektor alapú protokollokhoz képest:

* **Memóriakövetelmények** - A kapcsolatállapot alapú protokollok több memóriát igényelnek a kapcsolatállapot-adatbázis és az SPF-fa létrehozásához és karbantartásához.
* **Feldolgozási követelmények** - A kapcsolatállapot alapú protokolloknak nagyobb processzor teljesítményre van szükségük, mint a távolságvektor alapúaknak. Az SPF-algoritmus több CPU-időt igényel, mint a távolságvektor alapú algoritmusok, például a Bellman-Ford, mivel a kapcsolatállapot alapú protokollok a teljes topológiáról készítenek térképet.
* **Sávszélességigény** - A kapcsolatállapot-csomagok elárasztása negatív hatással lehet a rendelkezésre álló hálózati sávszélességre. Ez csak a forgalomirányítók első indításakor vagy instabil hálózatokban fordulhat elő.

# Kapcsolatállapot alapú dinamikus forgalomirányítás

## Miért használjunk kapcsolatállapot alapú forgalomirányító protokollokat?

A modern kapcsolatállapot alapú protokollok célja, hogy minimálisra csökkentsék a memóriára, a processzorra és a sávszélességre gyakorolt hatásokat. Több terület használatával és konfigurálásával csökkenthető a kapcsolatállapot-adatbázis mérete, valamint a kapcsolatállapot-információk elárasztása is korlátozható egy irányítási tartományra. Így az LSP-ket csak azon forgalomirányítók kapják meg, akiknek szüksége van rá. Topológiaváltozás esetén kizárólag az érintett terület forgalomirányítói kapnak LSP-csomagokat, és csak nekik kell újra lefuttatniuk az SPF-algoritmust. Ez segíthet a bizonytalan összeköttetések negatív hatásait a forgalomirányítási tartomány érintett területére korlátozni.

Az ábrán például 3 különálló forgalomirányítási tartomány látható: 1-es terület, 0-ás terület és 51-es terület. Ha az 51-es területen belül leáll egy hálózat, akkor a leállt összeköttetést tartalmazó LSP-t csak a területen belüli forgalomirányítókhoz kell elküldeni. Így csak az 51-es terület forgalomirányítóinak kell frissíteni a kapcsolatállapot-adatbázist, újra lefuttatni az SPF-algoritmust, felépíteni egy új SPF-fát és a változásokat felvenni az irányítótáblába. Más területek forgalomirányítói ezt az útvonalat lekapcsolt állapotúnak értelmezik, amit egy olyan típusú LSP-ből ismertek meg, amely nem kényszerítette őket az SPF-algoritmus újrafuttatására. Ezek a forgalomirányítók közvetlenül el tudják végezni a módosításokat az irányítótáblájukban.

# Kapcsolatállapot alapú dinamikus forgalomirányítás

## Miért használjunk kapcsolatállapot alapú forgalomirányító protokollokat?

Mindössze két kapcsolatállapot alapú protokoll létezik, az OSPF és az IS-IS.

Közülük a legrövidebb út protokoll (OSPF) a népszerűbb. A protokoll fejlesztését az Internet Engineering Task Force (IETF) OSPF-munkacsoportja végezte. Az OSPF fejlesztése 1987-ben kezdődött, és jelenleg két változata van használatban:

* OSPFv2- IPv4 alapú hálózatok számára (RFC 1247 és RFC 2328)
* OSPFv3- IPv6 alapú hálózatok számára (RFC 2740)

**MEGJEGYZÉS**: Az OSPFv3 címcsaládok (Address Families) funkció használatával a protokoll képes az IPv4 és IPv6 támogatására is.

Az IS-IS protokollt a Nemzetközi Szabványügyi Hivatal (Organization for Standardization, ISO) fejlesztette ki és az ISO10589-ben írta le.

Az IS-IS-t eredetileg az OSI és nem pedig a TCP/IP protokollkészlet számára tervezték. Az IP-hálózatok támogatását később az Integrált IS-IS vagy más néven Duális IS-IS változat tartalmazta. Bár az IS-IS elsősorban internetszolgáltatók által használt protokollként ismert, egyre több vállalat kezdte el használni a protokollt a saját hálózatában.

Az OSPF és az IS-IS sok hasonlóságot mutat, de számosban dologban eltérnek egymástól. Több olyan, az OSPF illetve az IS-IS mellett érvelő csoport létezik, amely az egyik protokoll előnyeit méltatja a másikkal szemben. A szükséges irányítási funkciók ellátására mindkét irányító protokoll képes.

# Az irányítótábla

## Az IPv4-útvonalbejegyzés részei

Az 1. ábrán látható topológiát referencia topológiaként fogjuk használni ebben a részben. Figyeljük meg rajta, hogy:

* R1 határponti forgalomirányító, amely az internethez csatlakozik. Ez az eszköz hirdeti tovább az alapértelmezett statikus útvonalat R2 és R3 felé.
* R1, R2 és R3 nem folytonos hálózatokat tartalmaznak, amelyek más osztály alapú hálózatokkal vannak elválasztva egymástól.
* R3 ezen felül tartalmaz egy 192.168.0.0/16 szuperhálózati útvonalat is.

A 2. ábra R1 IPv4-irányítótábláját mutatja, benne a közvetlenül csatlakozó, a statikus és a dinamikus útvonalakkal.

**MEGJEGYZÉS**: A Cisco IOS-ben az irányítótábla felépítését eredetileg az osztály alapú forgalomirányításra tervezték. Habár az irányítótábla az osztály alapú és az osztály nélküli címzést is magában foglalja, a teljes szerkezete még mindig az osztály alapú rendszert követi.

# Az irányítótábla

## Az IPv4-útvonalbejegyzés részei

Az 1. ábrán látható, hogy R1 irányítótáblája három közvetlenül csatlakozó hálózatot tartalmaz. Figyeljük meg, hogy amikor egy forgalomirányító aktív interfészén IP-címet és maszkot konfigurálunk, akkor automatikusan két bejegyzés kerül az irányítótáblába.

A 2. ábrán az R1 forgalomirányító 172.16.1.0 közvetlenül csatlakozó hálózatának irányítótábla bejegyzése látható. Ez automatikusan került be az irányítótáblába a GigabitEthernet 0/0 interfész konfigurálását és aktiválását követően. A bejegyzés a következő információkat tartalmazza:

* **Az útvonal forrása** - Megmutatja, honnan tanulta meg az adott útvonalat a forgalomirányító. Közvetlenül csatlakozó interfészek esetén két lehetséges kód létezik. **C** közvetlenül csatlakozó hálózatot jelöl. A közvetlenül csatlakozó hálózatok automatikusan létrejönnek a táblázatban, amint az interfész működőképes és rendelkezik IP-címmel. **L** a forgalomirányító interfészére mutató útvonalat jelöli. Ezek a helyi útvonalak is automatikusan kerülnek az irányítótáblába, amikor az interfészen IP-címet konfigurálunk és aktiváljuk azt.
* **Célhálózat** - A távoli hálózat címe és csatlakozásának módja.
* **Kimenő interfész** - Azonosítja azt az interfészt, amelyen a csomagokat a célhálózat felé továbbítani kell.

**MEGJEGYZÉS**: A forgalomirányító interfészekre mutató helyi irányítótábla bejegyzések (L) a 15-ös IOS-nél korábbi verziók esetén nem jelennek meg az irányítótáblában.

Egy forgalomirányítón jellemzően több konfigurált interfész is van. A forgalomirányító nem csak ezekről, hanem a távoli útvonalakról is tárol információkat. A közvetlenül csatlakozó hálózatokhoz hasonlóan itt is az útvonal forrása azonosítja azt, hogy a forgalomirányító hogyan tanulta meg az adott útvonalat. Távoli hálózatok esetén használt gyakori kódok:

* **S** - Az útvonalat a rendszergazda kézzel hozta létre egy adott hálózat elérése érdekében. Ezt hívjuk statikus útvonalnak.
* **D** - Az útvonalat a forgalomirányító dinamikusan tanulta meg más forgalomirányítótól az EIGRP forgalomirányító protokoll segítségével.
* **O** - Az útvonalat a forgalomirányító dinamikusan tanulta meg más forgalomirányítótól az OSPF-protokoll segítségével.
* **R** - Az útvonalat a forgalomirányító dinamikusan tanulta meg más forgalomirányítótól a RIP-protokoll segítségével.

# Az irányítótábla

## Az IPv4-útvonalbejegyzés részei

Az ábrán az R1 forgalomirányító egy IPv4 irányítótábla bejegyzése látható az R3-on lévő 172.16.4.0 távoli hálózatra vonatkozóan. A bejegyzés az alábbi információkat tartalmazza:

* **Az útvonal forrása** - Megadja, hogy a forgalomirányító honnan tanulta meg az adott útvonalat.
* **Célhálózat** - A távoli hálózat címe.
* **Adminisztratív távolság** - A forrás megbízhatóságát adja meg.
* **Mérték** - A távoli hálózat eléréséhez társított érték. A kisebb érték jobb útvonalat jelöl.
* **Következő ugrás** - Annak a következő forgalomirányítónak az IPv4-címe, amelyhez a csomagot továbbítani kell.
* **Az útvonal időbélyege** - Az utolsó útvonalfrissítés óta eltelt idő.
* **Kimenő interfész** - Az a kimenő interfész, amelyiken a csomagot a célja felé továbbítani kell.

# Az irányítótábla

## Dinamikusan megtanult IPv4-útvonalak

Ahogy az ábrán is látható, egy dinamikusan felépített irányítótáblában nagy mennyiségű információ áll rendelkezésre. Ezért nagyon fontos, hogy megértsük az irányítótábla által előállított kimenetet. Amikor az irányítótábla tartalmáról beszélünk speciális kifejezéseket is használnunk kell.

A Cisco IP-irányítótábla nem egy sima adatbázis. A forgalomirányító tábla igazából hierarchikus felépítésű, és a keresési folyamat felgyorsítására szolgál, amikor a forgalomirányító útvonalakat keres és csomagokat továbbít. Ennek a hierarchikus felépítésnek számos szintje létezik.

Az útvonalakat az alábbi kifejezésekkel jellemezhetjük:

* Teljes útvonal (Ultimate route)
* 1. szintű útvonal (Level 1 route)
* 1. szintű szülő útvonal (Level 1 parent route)
* 2. szintű gyerek útvonal (Level 2 child route)

# Az irányítótábla

## Dinamikusan megtanult IPv4-útvonalak

A teljes útvonal egy olyan bejegyzés az irányítótáblában, amely a következő ugrás IPv4-címét vagy a kimenő interfészt tartalmazza. A közvetlenül csatlakozó, a dinamikusan megtanult és a forgalomirányító interfészére mutató helyi útvonalak tartoznak a teljes útvonalak közé.

Az ábrán a kiemelt részek jelölik a teljes útvonalakat. Figyeljük meg, hogy az összes ilyen útvonal a következő ugrás IPv4-címét vagy a kimenő interfészt határozza meg.

# Az irányítótábla

## Dinamikusan megtanult IPv4-útvonalak

Egy 1. szintű útvonal alhálózati maszkja kisebb vagy egyenlő a hálózati cím osztály alapú maszkjánál. Ennélfogva az 1. szintű útvonal lehet:

* **Hálózati útvonal** - Olyan hálózati útvonal, amelynek alhálózati maszkja megegyezik az osztály alapú maszkkal.
* **Szuperhálózati útvonal** - Olyan hálózati cím, amelynek alhálózati maszkja kisebb, mint az osztály alapú maszk. Ilyen például a összevont hálózati cím.
* **Alapértelmezett útvonal** - A 0.0.0.0/0 címmel megadott statikus útvonal.

Az 1. szintű útvonal forrása egy közvetlenül csatlakozó hálózat, egy statikus útvonal vagy egy dinamikus irányító protokoll is lehet.

Az 1. ábra azt szemlélteti, hogy az 1. szintű útvonalak teljes útvonalak is egyben.

A 2. ábrán az 1. szintű útvonalak láthatók kiemelve.

# Az irányítótábla

## Dinamikusan megtanult IPv4-útvonalak

Amint az 1. ábrán látható, az 1. szintű szülő útvonal egy alhálózatokra bontott 1. szintű hálózati útvonal. Egy szülő útvonal soha nem lehet teljes útvonal.

A 2. ábrán az R1 irányítótáblájában található 1. szintű szülő útvonalak vannak kiemelve. Ezek a bejegyzések az irányítótáblában alapvetően egy fejrészt biztosítanak azon alhálózatok számára, amelyeket magában foglalnak. Mindegyik bejegyzésben megjelenik az osztály alapú hálózati cím, az alhálózatok és a különböző alhálózati maszkok száma, amelyekre az osztály alapú cím fel lett osztva.

# Az irányítótábla

## Dinamikusan megtanult IPv4-útvonalak

A 2. szintű gyerek útvonal egy osztály alapú cím alhálózata. Amint az 1. ábrán látható, az 1. szintű szülő útvonal egy alhálózatokra bontott 1. szintű hálózati útvonal. Ezek az 1. szintű szülő útvonalak 2. szintű gyerek útvonalakat tartalmaznak. (lásd 2. ábra)

Az 1. szintű útvonalakhoz hasonlóan, egy 2. szintű útvonal forrása is lehet közvetlenül csatlakozó hálózat, statikus útvonal vagy dinamikusan megismert útvonal. A 2. szintű gyerek útvonalak teljes útvonalak is egyben.

**MEGJEGYZÉS**: A Cisco IOS-ben az irányítótábla felépítése az osztály alapú forgalomirányításon alapul. Egy 1. szintű szülő útvonal megegyezik az alhálózati útvonal osztály alapú hálózati címével. Ez a helyzet akkor is, ha az alhálózati útvonal forrása egy osztály nélküli irányító protokoll.

A 3. ábrán az R1 irányítótáblájában található gyerek útvonalak vannak kiemelve.

# Az irányítótábla

## IPv4-útvonalak keresési folyamata

Ha egy forgalomirányító valamely interfészére csomag érkezik, az eszköz az IPv4-fejléc vizsgálatával azonosítja a cél IPv4-címét majd végrehajtja az útvonal keresés folyamatát.

Az 1. ábrán a forgalomirányító az 1. szintű hálózati útvonalakat vizsgálja, hogy legjobb egyezést találjon az IPv4-csomag célcímével.

1. Ha a legjobb egyezés eredménye egy teljes útvonal, akkor ezt az útvonalat használja a csomag továbbítására.

2. Ha egy 1. szintű szülő útvonallal talál legjobb egyezést, akkor a folyamat a következő lépéssel folytatódik.

A 2. ábrán a forgalomirányító a szülő útvonal alá tartozó gyerek útvonalakat (alhálózati útvonalak) vizsgálja.

3. Ha egy 2. szintű gyerek útvonallal talál legjobb egyezést, akkor ezt az alhálózati útvonalat használja a csomag továbbítására.

4. Ha egyetlen 2. szintű gyerek útvonallal sem talál egyezést, akkor a folyamat a következő lépéssel folytatódik.

A 3. ábrán a forgalomirányító folytatja az 1. szintű szuperhálózati útvonalak keresését az irányítótáblában, beleértve az alapértelmezett útvonalat is, amennyiben létezik.

5. Ha egy rövidebb egyezést mutató 1. szintű szuperhálózati útvonalat vagy alapértelmezett útvonalat talál, azt az útvonalat használja a csomag továbbítására.

6. Ha az irányítótábla egyetlen útvonalával sincs egyezés, akkor a forgalomirányító eldobja a csomagot.

**MEGJEGYZÉS**: Az olyan útvonalakhoz, amelyek csak egy következő ugrás IP-címre hivatkoznak és kimenő interfészre nem, keresni kell egy kimenő interfésszel rendelkező útvonalat, amennyiben Cisco Express Forwarding (CEF) funkció nincs használatban. CEF használata nélkül addig kell rekurzív keresést végrehajtani a következő ugrás IP-címmel, amíg a forgalomirányító az útvonalhoz kimenő interfészt nem talál.

# Az irányítótábla

## IPv4-útvonalak keresési folyamata

Ahhoz, hogy egy csomag cél IPv4-címe és az irányítótábla egy útvonala között egyezés legyen, bal oldalról nézve bizonyos számú bitnek meg kell egyeznie bennük. Az egyező bitek minimális számának meghatározásához az útvonalhoz megadott alhálózati maszkot használják. Ne feledjük, hogy egy IPv4-csomag csak az IPv4-címet tartalmazza, az alhálózati maszkot nem.

Legjobb egyezésnek az irányítótábla azon útvonalát tekintjük, amelynek bal oldalról nézve a legtöbb bitje egyezik meg a csomag cél IPv4-címével. Legjobb útvonalnak mindig azt az útvonalat tekintjük, amely balról kezdve a legtöbb egyező bittel vagy leghosszabb egyezéssel rendelkezik.

Az ábrán látható csomag célcíme 172.16.0.10. A forgalomirányító három lehetséges útvonallal rendelkezik a megadott célhoz: 172.16.0.0/12, 172.16.0.0/18 és 172.16.0.0/26. Mivel a három útvonal közül a 172.16.0.0/26 mutatja a leghosszabb egyezést, így a forgalomirányító ezt választja a csomag továbbításához. Ne feledjük, hogy egyezésről csak akkor beszélhetünk, ha legalább az alhálózati maszkban jelzett bitszámig megegyezik binárisan a célcím és az adott útvonal.

# Az irányítótábla

## IPv6-irányítótábla elemzése

Az IPv6 és az IPV4-irányítótábla elemei sok mindenben hasonlítanak. Például mindkettőt közvetlenül csatlakozó interfészek, statikus útvonalak és dinamikusan megtanult útvonalak alkotják.

Mivel az IPv6-ot osztály nélkülinek tervezték, gyakorlatilag minden útvonal 1. szintű teljes útvonalnak számít. Nincsenek 1. szintű szülő vagy 2. szintű gyerek útvonalak.

Az 1. ábrán látható topológiát referencia topológiaként fogjuk használni ebben a részben. Figyeljük meg rajta, hogy:

* R1, R2 és R3 teljes háló topológiába vannak kötve. Mindegyik forgalomirányító rendelkezik helyettesítő útvonalakkal a különböző hálózatok felé.
* R2 a határponti forgalomirányító, amely az ISP-hez csatlakozik. Alapértelmezett statikus útvonal nincs hirdetve.
* Mindhárom forgalomirányítón az EIGRP IPv6-os változata van beállítva.

# Az irányítótábla

## IPv6-irányítótábla elemzése

Az 1. ábrán R1 irányítótáblája látható, amely a **show ipv6 route** parancs kiadása után jelenik meg. Bár a parancs kimenete kicsit eltér az IPv4-es parancshoz képest, a lényeges útvonal információkat így is tartalmazza.

A 2. ábrán a közvetlenül csatlakozó interfészek hálózati és helyi irányítótábla bejegyzései vannak kiemelve. A három bejegyzés az interfészek konfigurálását és aktiválását követően került az irányítótáblába.

Ahogy a 3. ábrán látható, a közvetlenül csatlakozó útvonalak bejegyzései a következő információkat jelenítik meg:

* **Az útvonal forrása** - Megadja, hogy a forgalomirányító honnan tanulta meg az adott útvonalat. Közvetlenül csatlakozó interfészek esetén két lehetséges kód létezik. (C a közvetlenül csatlakozó hálózatot, míg L a forgalomirányító helyi interfészére mutató útvonalat jelöli )
* **Közvetlenül csatlakozó hálózat** - A közvetlenül csatlakozó hálózat IPv6-címe.
* **Adminisztratív távolság** - Az útvonal forrásának megbízhatóságát jelenti. IPv6 esetén ugyanazokat a távolság értékeket használjuk mint IPv4-nél. A 0 érték a legjobb, legmegbízhatóbb forrást jelöli.
* **Mérték** - A távoli hálózat eléréséhez társított érték. A kisebb érték jobb útvonalat jelöl.
* **Kimenő interfész** - Azonosítja azt az interfészt, amelyen a csomagokat a célhálózat felé továbbítani kell.

**MEGJEGYZÉS**: A soros kapcsolatokon referencia sávszélesség van beállítva, hogy megfigyelhető legyen az EIGRP-mértékek alapján történő legjobb útvonal kiválasztása. A referencia sávszélességgel a modern hálózatok nem ábrázolhatók valósághűen, csupán a kapcsolat sebességének vizuális érzékeltetésére használható.

# Az irányítótábla

## IPv6-irányítótábla elemzése

Az 1. ábrán a három távoli hálózat irányítótábla bejegyzései vannak kiemelve (azaz R2 LAN, R3 LAN, R2 és R3 közötti kapcsolat). Mindhárom bejegyzést az EIGRP-protokoll adta hozzá az irányítótáblához.

A 2. ábrán az R1 forgalomirányító egy irányítótábla bejegyzése látható az R3-on található 2001:DB8:CAFE:3::/64 távoli hálózatra vonatkozóan. A bejegyzés az alábbi információkat tartalmazza:

* **Az útvonal forrása** - Megadja, hogy a forgalomirányító honnan tanulta meg az adott útvonalat. A gyakori kódok között az O (OSPF), a D (EIGRP), az R (RIP) és az S (statikus útvonal) találhatók meg.
* **Célhálózat** - A távoli IPv6-hálózat címét azonosítja.
* **Adminisztratív távolság** - Az útvonal forrásának megbízhatóságát jelenti. IPv6 esetén ugyanazokat a távolság értékeket használjuk mint IPv4-nél.
* **Mérték** - A távoli hálózat eléréséhez társított érték. A kisebb érték jobb útvonalat jelöl.
* **Következő ugrás** - Csomagtovábbítás során a következő forgalomirányító IPv6-címe.
* **Kimenő interfész** - Az a kimenő interfész, amelyiken a csomagot a célja felé továbbítani kell.

Amikor egy forgalomirányító valamelyik interfészére IPv6-csomag érkezik, az eszköz megvizsgálja az IPv6-fejlécet és azonosítja a cél IPv6-címét. Ezután a következő keresési folyamatot végzi el.

Megvizsgálja az 1. szintű hálózati útvonalakat, hogy legjobb egyezést találjon az IPv6-csomag célcímével. Az IPv4-hez hasonlóan a legjobb egyezés itt is egyenértékű a leghosszabb egyezéssel. Például, ha több egyezés is található az irányítótáblában, akkor a forgalomirányító azt az útvonalat választja, amelyik a leghosszabb egyezéssel rendelkezik. Egyeztetésről akkor beszélünk, ha a csomag cél IPv6-címének bal szélső bitjeit összehasonlítjuk az irányítótáblában található IPv6-előtaggal és előtag-hosszal.

# Összegzés

## Összegzés

**IPv6 - Részletek, részletek ...**

A fejezetben bemutatott, IPv6-hoz kapcsolódó fogalmak megtanulása után képesnek kell lennünk az irányítótábla tanulmányozására és a benne szereplő IPv6 útvonal-információk értelmezésére.

Használjuk egy partnerrel együttműködve az IPv6-irányítótábla ábráját és a feladathoz mellékelt pdf fájlt.

Jegyezzük fel a gondolkodtató kérdésekre adott válaszainkat.

Ezt követően hasonlítsuk össze a válaszainkat az osztály legalább egy másik csoportjával.

[Csoportos feladat - IPv6 - Details, Details... Instructions](https://contenthub.netacad.com/legacy/RSE/5.02/hu/course/files/7.6.1.1%20Class%20Activity%20-%20IPv6%20-%20Details,%20Details...%20Instructions.pdf)

# Összegzés

## Összegzés

A dinamikus irányító protokollokat a forgalomirányítók közötti információcsere megkönnyítésére használják. Céljaik között a következők találhatók meg: távoli hálózatok felfedezése, az aktuális útvonal információk karbantartása, a célhálózatok felé vezető legjobb útvonalak kiválasztása és az aktuális útvonal elérhetetlenné válása esetén egy újabb legjobb útvonal kiválasztásának képessége. A statikus útvonalakkal szemben kevesebb adminisztrációt igényelnek, viszont a protokoll működéséhez a forgalomirányító erőforrásainak jelentős részére szükség van, ideértve a CPU-időt és a hálózati kapcsolatok sávszélességét is.

A statikus és dinamikus forgalomirányítást jellemzően egymással kombinálva alkalmazzák a hálózatokban. Nagyméretű hálózatokban a dinamikus, míg kisebb hálózatrészek esetén a statikus forgalomirányítás a jobb megoldás.

Az irányító protokollok a távoli hálózatok felfedezéséért, valamint a pontos hálózati információk karbantartásáért felelősek. A topológiában bekövetkezett változásokat az irányító protokollok tovább hirdetik az irányítási tartományban. Konvergenciának hívjuk azt a folyamatot, amelynek során az irányítótáblák konzisztens állapotba kerülnek, és így az ugyanabba az irányítási tartományba vagy területbe tartozó forgalomirányítók teljes körű és pontos információkkal rendelkeznek a hálózatról. Az egyes irányító protokollok eltérő konvergencia idővel rendelkeznek.

Az irányító protokollok lehetnek osztály alapúak vagy osztály nélküliek, távolságvektor vagy kapcsolatállapot alapúak és külső vagy belső átjáróprotokollok.

A távolságvektor alapú protokollok "útjelzőként" használják a forgalomirányítókat a cél felé vezető útvonalon. Egy távoli hálózatról a forgalomirányító csak annyi információval rendelkezik, hogy mekkora a hálózat eléréséhez szükséges távolság vagy mérték és, hogy melyik útvonalat vagy interfészt használja az odajutáshoz. A távolságvektor alapú protokollok nem rendelkeznek egy tényleges térképpel a hálózat topológiájáról.

Kapcsolatállapot alapú irányító protokollal konfigurált forgalomirányító a többi forgalomirányítótól gyűjtött információk alapján egy teljes képet tud összeállítani a hálózat topológiájáról.

Az irányító protokollok mértékeket használnak a célhálózat felé vezető legjobb vagy legrövidebb útvonal kiválasztásához. A különböző irányító protokollok eltérő irányítási mértékeket használhatnak. A kisebb mérték általában jobb útvonalat jelöl. A mértékeket az ugrásszámból, a sávszélességből, a megbízhatóságból és a terhelésből lehet meghatározni.

A forgalomirányítók statikus és dinamikus forrásból néha több útvonalat is megismernek ugyanazon hálózat felé. Amikor egy Cisco forgalomirányító egynél több forrásból is értesül a célhálózatról, az adminisztratív távolság értékének használatával dönti el, hogy melyiket használja. Az irányító protokollok mindegyike egyedi adminisztratív távolság értékkel rendelkezik, csakúgy mint a statikus útvonalak és a közvetlenül csatlakozó hálózatok is. Minél kisebb az adminisztratív távolság értéke, annál megbízhatóbb az útvonal forrása. Mindig a közvetlenül csatlakozó hálózatok a legelőnyösebbek, őket követik a statikus útvonalak, majd a különböző dinamikus irányító protokollok.

A **show ip protocols** parancs a forgalomirányítón aktuálisan beállított IPv4 irányító protokoll beállításait mutatja meg. IPv6-os környezetben a **show ipv6 protocols**parancsot használjuk.

A kapcsolatállapot alapú protokollok, mint például az OSPF esetén a kapcsolat a forgalomirányító egy interfészét jelenti. Ezen interfészek állapot-információit nevezzük kapcsolatállapotnak. A kapcsolatállapot alapú protokollok a Dijkstra algoritmust használják a legjobb útvonal kiszámításához. Az algoritmus ismertebb neve a legrövidebb utat kereső (SPF) algoritmus. Az útvonal teljes költségének meghatározásához az algoritmus a forrás és cél közötti utak költségeit összesíti.

