# Fejezet 8: Egyterületű OSPF

# Egyterületű OSPF

## Egyterületű OSPF

Az OSPF (Open Shortest Path First, "legrövidebb utat először") egy kapcsolatállapot alapú forgalomirányító protokoll, amelyet a távolságvektor alapú RIP forgalomirányító protokoll leváltására fejlesztettek ki. A RIP a hálózatok és az internet korai időszakában elfogadható forgalomirányító protokollnak számított . Ugyanakkor az, hogy a RIP legjobb út meghatározásában egyedül az ugrásszámra támaszkodik, viszonylag korán problémássá vált. Az ugrásszám használata a többszörös, különböző sebességű utakat is tartalmazó, kiterjedt hálózatok esetében nehezen skálázható. A RIP-el szemben az OSPF lényeges előnye, hogy a nagyobb hálózatok esetében gyorsabb konvergenciát és méretezhetőséget biztosít.

Az OSPF egy osztály nélküli forgalomirányító protokoll, amely a skálázhatóság érdekében bevezeti a terület fogalmát. Jelen fejezet az alap, egyterületű OSPF telepítését és konfigurációját mutatja be.

# Egyterületű OSPF

## Egyterületű OSPF

**Feladat -** **Tudnak-e a tengeralattjárók úszni?**

**Esetleírás**

Edsger Wybe Dijkstra híres számítógép programozó és elméleti fizikus volt. Az egyik leghíresebb mondása szerint: „A kérdés miszerint tudnak-e a számítógépek gondolkodni olyan, mintha azt kérdeznénk, hogy tudnak-e a tengeralattjárók úszni.” Dijkstra munkáit számos helyen alkalmazzák, így a forgalomirányító protokolloknál is. Hálózati forgalomirányításra ő alkotta a "legrövidebb utat először" (Shortest Path First, SPF) algoritmust is.

Nyissuk meg a gyakorlathoz tartozó PDF dokumentumot és válaszoljuk meg a kérdéseket. Mentsük el a munkánkat!

Alkossunk háromfős csoportokat, csoporton belül vessük össze a válaszokat.

Sikerült-e a gyakorlat alapján kitalálni, hogyan működhet az OSPF protokoll?

[Csoportos feladat - Can Submarines Swim? Instructions](https://contenthub.netacad.com/legacy/RSE/5.02/hu/course/files/8.0.1.2%20Class%20Activity%20-%20Can%20Submarines%20Swim%20Instructions.pdf)

# Az OSPF tulajdonságai

## Legrövidebb utat először (OSPF)

Ahogy azt az 1. ábra is mutatja, az OSPF 2. verziója (OSPFv2) az IPv4-et, míg az OSPF 3. verziója (OSPFv3) az IPv6-ot támogatja.

Kattintsunk a 2. ábrán a dátumokra, az OSPF történetéhez kapcsolódó főbb események megtekintéséhez.

Az OSPF fejlesztése 1987-ben kezdődött az Internet Engineering Task Force (IETF) OSPF munkacsoportjában. Abban az időben az internet leginkább egy az amerikai kormányzat által támogatott akadémiai és kutatóhálózat volt. .

Az RFC 1131 volt az OSPFv1 első specifikációja 1989-ben. Két implementáció készült. Az egyiket forgalomirányítókra, a másikat UNIX munkaállomásokra írták. Az utóbbi a GATED UNIX processz néven vált közismertté. Kísérleti forgalomirányító protokoll lévén az OSPFv1-et nem igazán alkalmazták.

Az OSPFv2-t 1991-ben John Moy mutatta be az RFC 1247-ben. Az OSPFv2 az OSPFv1-hez képest számos lényeges technikai újítást tartalmazott. Osztály nélkülire tervezték, ezért támogatta a VLSM-et és a CIDR-t is.

Az OSPF bemutatásának idején az ISO is dolgozott a saját kapcsolatállapot alapú forgalomirányító protokollján, az IS-IS (Intermediate System-to-Intermediate System) protokollon. Az IETF az OSPF-et választotta az ajánlott belső forgalomirányító protokollnak (Interior Gateway Protocol, IGP).

1998-ban frissítették az OSPFv2 specifikációt az RFC 2328-ben, mely azóta is az aktuális OSPF RFC.

1999-ben az RFC 2740-ben megjelent az IPv6 támogatására alkalmas OSPFv3. A John Moy, Rob Coltun és Dennis Ferguson alkotta IPv6-os OSPF nem cask egy új, IPv6-ot is támogató protokoll, hanem egy teljesen átdolgozott változat.

2008-ban az RFC 5340 frissítette az OSPFv3-at, most már IPv6-os OSPF (OSPF for IPv6) néven.

**MEGJEGYZÉS:** A fejezetben, ha csak az OSPFv2, vagy OSPFv3 néven külön nincs jelezve, olyan fogalmakkal és módszerekkel dolgozunk, melyek mindkét rendszer esetében azonosak.

# Az OSPF tulajdonságai

## Legrövidebb utat először (OSPF)

Az OSPF jellemzői, ahogy azt az 1. ábra is mutatja, az alábbiak:

* **Osztály nélküli** - Tervezéséből eredően osztály nélküli, azaz támogatja a VLSM-et és CIDR-t.
* **Hatékony** - A forgalomirányítás változásai váltják ki a forgalomirányítás frissítéseket (nincsenek periodikus frissítések). A legjobb útvonal kiválasztásához az SPF-algoritmust használja.
* **Gyors konvergencia** - A hálózat változásai gyorsan elterjednek.
* **Skálázhatóság** - Egyaránt jól működik kis - és nagy hálózatok esetében is. Hierarchikus rendszerek kialakításának érdekében a forgalomirányítók csoportokba szervezhetők.
* **Biztonságos** - Támogatja a Message Digest 5 (MD5) alapú azonosítást. Ha az engedélyezve van, akkor az OSPF forgalomirányítók a társaiktól csak olyan kódolt irányítási frissítéseket fogadnak el, ahol megegyezik az osztott kulcs.

Az adminisztratív távolság (Administrative Distance, AD) a forgalomirányítási adatok forrásának megbízhatóságát (vagy prioritását) jelöli. Az OSPF alapértelmezett adminisztratív távolsága 110. Ahogy azt a 2. ábra is mutatja, az OSPF megbízhatóbbnak számít az IS-IS, vagy a RIP protokolloknál.

# Az OSPF tulajdonságai

## Legrövidebb utat először (OSPF)

Valamennyi forgalomirányító protokoll hasonló részekből épül fel. Mindegyik forgalomirányító protokoll üzenetek formájában osztja meg a forgalomirányítási információkat. Ezekből az üzenetekből aztán a forgalomirányító algoritmus által feldolgozható adatstruktúrákat építenek.

Az OSPF forgalomirányító protokoll három fő összetevője:

**Adatstruktúrák**

Az OSPF három adatbázist hoz létre és tart karban (lásd 1. ábra):

* **Szomszédsági adatbázis (Adjacency database)** – ez alkotja a szomszédsági táblát (neighbor table).
* **Kapcsolatállapot adatbázis (Link-state database, LSDB)** - ez alkotja topológia táblát (topology table).
* **Továbbítási adatbázis (Forwarding database)** – ez pedig az irányítótábla (routing table).

Ezek a táblázatok tartalmazzák azon szomszéd forgalomirányítók listáját, melyekkel forgalomirányítási adatokat kell cserélnie. A táblázatokat RAM-ban tárolják és ott is tartják karban.

**Forgalomirányító protokoll üzenetek**

Az OSPF öt csomagtípust használ a forgalomirányítási adatok megosztásánál. Ezen csomagok, ahogy azt a 2. ábra is mutatja, az alábbiak:

* Hello csomag
* Adatbázis-leíró csomag
* Kapcsolatállapot kérés csomag
* Kapcsolatállapot frissítés csomag
* Kapcsolatállapot nyugtázás csomag

Ezen csomagok szolgálnak a szomszédos forgalomirányítók felderítésére, valamint az aktuális hálózati információk karbantartása okán a forgalomirányítási adatok cseréjére.

**Az algoritmus**

A CPU a szomszédsági és topológiai táblákat Dijkstra SPF-algoritmusával dolgozza fel. Az SPF-algoritmus meghatározza az egyes célok elérésének teljes költségét.

Az SPF-algoritmus felépít egy SPF-fát (Shortest Path First, "legrövidebb utat először"), amely a forgalomirányítótól - mint a fa gyökerétől - kiindulva minden más csomópontohz kiszámítja a hozzá vezető legrövidebb útvonalat. Ezek után az SPF-fa szolgál a legjobb útvonalak meghatározására. Az OSPF a legjobb utakat behelyezi a forgalomirányító adatbázisba, amelyet az irányítótábla felépítéséhez használ.

# Az OSPF tulajdonságai

## Legrövidebb utat először (OSPF)

A forgalomirányítási információk karbantartására, a konvergencia elérése érdekében az OSPF forgalomirányítók az alábbi általános kapcsolatállapot alapú forgalomirányítási folyamatot hajtják végre:

1. Szomszédsági kapcsolatok létrehozása (1. ábra) – az adatok megosztása előtt az OSPF forgalomirányítóknak fel kell deríteniük egymást a hálózaton. Ennek érdekében egy OSPF forgalomirányító valamennyi OSPF által használt interfészén Hello csomagokat küld ki annak kiderítésére, hogy vannak-e szomszédok ezeken a kapcsolatokon. Ha talál egy szomszédot, akkor azzal az OSPF forgalomirányító megpróbál ezzel a szomszéddal egy szomszédsági kapcsolatot kialakítani.

2. Kapcsolatállapot hirdetések (Link-State Advertisements, LSA) cseréje (2. ábra) – A szomszédségi kapcsolatok létrehozását követően a forgalomirányítók kapcsolatállapot hirdetéseket (LSA) váltanak. Az LSA valamennyi közvetlenül kapcsolódó kapcsolat állapotát és költségét tartalmazza. A forgalomirányítók az LSA-kkal elárasztják a kapcsolódó szomszédjaikat. Ha egy kapcsolódó szomszéd LSA-üzenetet kap, akkor azonnal továbbítja azt a saját kapcsolódó szomszédjai felé, mígnem a terület összes forgalomirányítója az összes LSA-t meg nem kapja.

3. Topológia tábla építése (3. ábra) – Az LSA-k begyűjtését követően, az OSPF forgalomirányítók az LSA-kból topológia táblát építenek. Ez az adatbázis lényegében a hálózat topológiájának minden részletét tartalmazza.

4. Az SPF-algoritmus végrehajtása (4. és 5. ábrák) – Ezt követően a forgalomirányítók végrehajtják az SPF-algoritmust. Az ábra fogaskerekei az SPF-algoritmus végrehajtását szimbolizálják. Az SPF-algoritmus eredményeként jön létre az SPF-fa.

Az R1 forgalomirányító SPF-fájának tartalmát a 6. ábra szemlélteti.

Az SPF-fából a legjobb utak kerülnek be az irányítótáblába. A forgalomirányítási döntések pedig az irányítótábla bejegyzései alapján történnek.

# Az OSPF tulajdonságai

## Legrövidebb utat először (OSPF)

Az OSPF úgynevezett területeket (area) használ a hierarchikus forgalomirányítás megvalósításához, amely ezáltal még hatékonyabbá és skálázhatóbbá válik. Egy OSPF-terület a forgalomirányítók olyan csoportja, melyek LSDB-ikben ugyanazon kapcsolatállapot információkat tartalmazzák.

Az OSPF az alábbi két módon valósítható meg:

* **Egyterületű OSPF (Single-Area OSPF)** – Az 1. ábra forgalomirányítói egyetlen területhez, a gerinc területhez (backbone area, vagy area 0) tartoznak.
* **Többterületű OSPF (Multiarea OSPF)** – A 2. ábrán egy hierarchikus, többterületű OSPF kialakítás látható. Valamennyi területnek a gerinc területhez (area 0) kell kapcsolódnia. A területeket összekötő forgalomirányítókat határ forgalomirányítónak (Area Border Routers, ABR) nevezzük.

A többterületű OSPF-el, a hierarchikus forgalomirányítás támogatására az OSPF egy nagyméretű autonóm területet (autonomous system - AS) több kisebb területre tud felosztani. Hierarchikus forgalomirányítás esetén továbbra is megtörténik a területek közti forgalomirányítás (interarea routing), viszont a számításigényes forgalomirányítási feladatok, mint például az adatbázisok újraszámítása, az egyes területeken belül maradnak.

Például valamennyi esetben, amikor egy forgalomirányító egy a területen belüli topológiaváltozásról értesül (új kapcsolatok megjelenése, meglévők törlése, vagy változása), újra futtatja az SPF-algoritmust, egy új SPF-fát hoz létre, majd frissíti a forgalomirányítási táblát. Az SPF-algoritmus számításigényes, a számítási idő pedig függ a terület méretétől.

**MEGJEGYZÉS**: A topológiaváltozások más területek forgalomirányítóihoz távolságvektor formájában vannak továbbítva. Azaz ezen forgalomirányítóknak elég a forgalomirányítási tábláikat módosítani és nem kell újra futtatni az SPF-algoritmust.

Ha egy területre túl sok forgalomirányító esne, akkor az nagyon megnövelné az LSDB-k méretét és jelentősen növelné a CPU terhelését. Ezért a forgalomirányítók területekbe rendezése hatékonyan oszt fel egy potenciálisan nagy adatbázist több kisebb, és ezért jobban kezelhető adatbázisra.

A többterületű OSPF hierarchikus topológiája az alábbi előnyöket nyújtja:

* **Kisebb irányítótáblák** – Az irányítótáblák bejegyzéseinek száma csökken, mert a területek között a hálózati címek összevonhatóak. Az útvonalak összevonása alapértelmezetten nem engedélyezett.
* **A kapcsolatállapot frissítések okozta többletforgalom csökkenése** – Minimalizálja a számítási- és memóriaigényt.
* **Az SPF-számítások gyakoriságának csökkentése** – A topológiaváltozások hatásának egyetlen területre történő lokalizálása. Például csökken a forgalomirányítási frissítések okozta terhelés, mert az LSA-k elárasztása megáll a terület határán.

A fenti előnyöket a 3. ábra foglalja össze.

Például az R2 forgalomirányító az 51-es terület ABR-je. ABR-ként összegzi az 51-es terület útvonalait a 0-ás terület számára. Ha valamelyik összevont útvonal meghibásodik, az LSA-k cseréje csak az 51-es területet érinti. A legjobb útvonalak meghatározására az 51-es terület forgalomirányítóinak újra kell futtatni az SPF-algoritmust. Eközben a 0-ás és 1-es területek forgalomirányítói nem kapnak semmilyen frissítést, ezért az SPF-algoritmust sem kell újból futtaniuk.

A fejezet a továbbiakban főleg az egyterületű OSPF-el foglalkozik.

# 

# Az OSPF tulajdonságai

## OSPF üzenetek

Egy Ethernet kapcsolaton továbbított OSPF üzenetek az alábbi információkat tartalmazzák:

* **Adatkapcsolati Ethernet keretfejléc** – Meghatározza a 01-00-5E-00-00-05 vagy a 01-00-5E-00-00-06 csoportos cél MAC-címet. (1. ábra)
* **IP csomagfejléc** – Az IPv4-csomag protokoll mezőjének 89-es értéke jelzi, hogy ez egy OSPF-csomag. Meghatározza még az OSPF két csoportcímzéses címének egyikét, a 224.0.0.5-et vagy a 224.0.0.6-ot. (2. ábra)
* **OSPF csomagfejléc** – Meghatározza az OSPF-csomag típusát, a forgalomirányító azonosítót (router ID) és a területazonosítót (area ID). (3. ábra)
* **OSPF csomagtípustól függő adatok** – Az OSPF csomagtípus információját tartalmazza. A tartalom a csomagtípustól függően különböző lehet. Ebben az esetben ez egy IPv4-fejléc (4. ábra)

# Az OSPF tulajdonságai

## OSPF üzenetek

Az OSPF úgynevezett kapcsolatállapot csomagokat (LSP-ket) használ a szomszédsági kapcsolatok karbantartására és a forgalomirányítási frissítések cseréjére.

Az ábra az OSPF által használt öt különböző LSP-típust mutatja. Mindegyik csomagnak speciális szerepe van az OSPF forgalomirányítási folymatban:

* **1-es típus: Hello csomag (Hello packet)** – Az OSPF forgalomirányítók közti szomszédsági kapcsolatok létrehozására és karbantartására szolgál.
* **2-es típus: Adatbázis leíró csomag (Database Description, DBD)** – A küldő forgalomirányító LSDB-adatbázisának rövidített listáját tartalmazza és arra szolgál, hogy össze lehessen vetni a fogadó helyi LSDB-adatbázisával. Az LSDB-nek azonosnak kell lennie a terület valamennyi forgalomirányítóján, hogy azok helyes SPF-fát építhessenek.
* **3-as típus: Kapcsolatállapot kérés csomag (Link-State Request, LSR)** – A fogadó forgalomirányítók egy LSR küldésével többletinformációt kérhetnek a kapott DBD bármelyik eleméről.
* **4-es típus: Kapcsolatállapot frissítés csomag (Link-State Update, LSU)** – Az LSR-ek megválaszolására és új adatok hirdetésére szolgál. Az LSU-k hét különböző típusú LSA-t tartalmazhatnak.
* **5-ös típus: Kapcsolatállapot nyugta csomag (Link-State Acknowledgment, LSAck)** – Ha egy forgalomirányító LSU-t kap, akkor annak nyugtázására egy LSAck-ot küld. Az LSAck adatmezője üres.

# Az OSPF tulajdonságai

## OSPF üzenetek

**Hello csomag**

Az 1-es típusú OSPF csomag a Hello csomag. A Hello csomagokat az alábbi esetekben használják:

* Az OSPF szomszédok felderítése és a szomszédsági kapcsolatok kialakítása.
* Azon paraméterek hirdetése, melyekben a két forgalomirányítónak meg kell egyeznie, hogy szomszédsági kapcsolatba kerülhessenek.
* A kijelölt forgalomirányító (Designated Router, DR) és a tartalék kijelölt forgalomirányító (Backup Designated Router, BDR) kiválasztása a többes hozzáférésű hálózatok esetében, mint pl. az Ethernet és a Frame Relay. Pont-pont kapcsolatok esetén nincs szükség DR-re és BDR-re.

Az ábra az 1-es típusú Hello csomag mezőit szemlélteti. Az ábrán látható lényegesebb mezők az alábbiak:

* **Típus (Type)** – A csomag típusát határozza meg. Az egyes (1) a Hello csomag típust jelöli. A 2-es érték a DBD csomagot jelenti, a 3-as az LSR csomagot, a 4-es az LSU csomagot és az 5-ös pedig az LSAck csomagot.
* **Forgalomirányító azonosító (Router ID)** – 32 bites mező, pontozott decimális formában (egy IPv4 cím), mely egyértelműen azonosítja a forrás (kezdeményező) forgalomirányítót.
* **Területazonosító (Area ID)** - Azt a területet jelöli, ahonnan a csomag származik.
* **Hálózati maszk (Network Mask)** – A küldő interfészhez tartozó alhálózati maszk.
* **Hello időintervallum (Hello Interval)** – A Hello csomagok küldésének gyakoriságát adja meg másodpercben. A Hello időintervallum alapértelmezett értéke többes hozzáférésű hálózatok esetében 10 másodperc. Ezen időzítő értékének a szomszédos forgalomirányítókon meg kel egyeznie. Ellenkező esetben a szomszédsági kapcsolat nem jön létre.
* **Forgalomirányító prioritás (Router Priority)** – A DR/BDR választás során használatos. Az OSPF forgalomirányítók alapértelmezett prioritása 1, amelynek 0 és 255 közötti értékre manuálisan megváltoztatható. Minél magasabb az érték, annál valószínűbb, hogy egy forgalomirányító DR lesz az adott kapcsolaton.
* **„Halott” intervallum (Dead Interval)** – Másodpercekben mérve az az időtartam, amennyit egy forgalomirányító a szomszédjának az üzenetére vár, mielőtt azt elérhetetlennek nyilvánítaná. Alapértelmezés szerint egy forgalomirányító "halott" időtartama a Hello időintervallumnak a négyszerese. Ezen időtartamnak is azonosnak kell lennie a szomszédos forgalomirányítókon. Ellenkező esetben a szomszédsági kapcsolat nem jöhet létre.
* **Kijelölt forgalomirányító (Designated Router, DR)** – A DR forgalomirányító azonosítója.
* **Tartalék kijelölt forgalomirányító (Backup Designated Router, BDR)** – A BDR forgalomirányító azonosítója.
* **A szomszédok listája (List of Neighbors)** – Valamennyi szomszédsági kapcsolatban lévő forgalomirányító azonosítóját felsoroló lista.

További információkért kattintsunk a kijelölt mezőkre!

# Az OSPF tulajdonságai

## OSPF üzenetek

Ahogy azt az ábra is mutatja, az OSPF Hello csomagok a 224.0.0.5 IPv4 és az FF02::5 IPv6 csoportos címekre (valamennyi OSPF forgalomirányító) vannak kiküldve:

* 10 másodperc (alapértelmezett a többes hozzáférésű és pont-pont hálózatok esetében).
* 30 másodpercenként (alapértelmezett a nem üzenetszórásos többes hozzáférésű hálózatok esetében - nonbroadcast multiaccess networks [NBMA], mint például a Frame Relay).

A "halott" időintervallum az az időtartam, amennyit egy forgalomirányító vár a Hello csomagra, mielőtt a szomszédját elérhetetlennek nyilvánítaná. Ha a halott időtartam lejár mielőtt a forgalomirányító megkapná a Hello csomagot, akkor az OSPF törli az illető szomszédot az LSDB-ből. Ezt követően a forgalomirányító az összes OSPF interfészén a szomszéd megszűnését jelző új LSDB üzenetet küld szét.

A Cisco alapértelmezett értékként a Hello időintervallum négyszeresét használja:

* 40 másodperc (alapértelmezett a többes hozzáférésű és pont-pont hálózatok esetében).
* 120 másodperc (alapértelmezett a NBMA hálózatok esetében például Frame Relay).

# Az OSPF tulajdonságai

## OSPF üzenetek

Kezdetben a a forgalomirányítók 2-es típusú DBD-csomagokat cserélnek, amelyek a küldő forgalomirányító LSDB-adatbázisának rövidített listáját tartalmazzák és arra szolgálnak, hogy a fogadó forgalomirányítók összevethessék azt helyi LSDB-adatbázisukkal.

A 3-as típusú LSR-csomagokat a fogadó forgalomirányítók használják arra, hogy többletinformációt kérjenek a kapott DBD elemeiről.

A 4-es típusú LSU-csomag az LSR-csomag megválaszolására szolgál.

Az LSU-k az OSPF forgalomirányítási frissítések - mint a kapcsolatállapot változás - továbbítására is szolgálnak. Pontosabban egy LSU-csomag 11 különböző típusú OSPFv2 LSA-t tartalmazhat, ahogy azt az ábra is szemlélteti. Az OSPFv3 ezek közül több LSA-t átnevezett, illetve felvett még további két LSA-t is.

**MEGJEGYZÉS:**Az LSU és az LSA fogalmak közti különbség néha megtévesztő lehet, mert ezeket a fogalmakat gyakran felcserélve használják. Valójában az LSU egy, vagy több LSA-t tartalmazhat.

# Az OSPF tulajdonságai

## Az OSPF működése

Amikor egy OSPF forgalomirányító először kapcsolódik a hálózathoz, akkor megkísérli hogy:

* szomszédsági viszonyokat alakítson ki a szomszédaival,
* forgalomirányítási adatokat cseréljen,
* kiszámítsa a legjobb utakat, illetve hogy
* konvergáljon.

A kovergencia eléréséhez az OSPF számos állapoton megy keresztül:

* kikapcsolt állapot (Down state)
* kezdeti állapot (Init state)
* kétirányú állapot (Two-Way state)
* cserekezdeti állapot (ExStart state)
* adatcserélő állapot (Exchange state)
* adatfeltöltő állapot (Loading state)
* teljes állapot (Full state)

Kattintsunk az ábra kék dobozaira további információkért!

# Az OSPF tulajdonságai

## Az OSPF működése

Amikor az OSPF-et engedélyezzük egy interfészen, a forgalomirányítónak meg kell határoznia, hogy van-e az adott kapcsolaton OSPF szomszédja. Ennek érdekében a forgalomirányító valamennyi OSPF interfészén Hello csomagokat küld szét, benne a saját forgalomirányító azonosítójával (router ID). Egy OSPF folyamatban az OSPF forgalomirányító azonosító szolgál az egyes forgalomirányítók OSPF területen belüli egyedi azonosítására. A forgalomirányító azonosító egy IP-cím, melyet az egyes OSPF forgalomirányítókhoz vannak hozzárendelve.

Amikor egy szomszédos OSPF forgalomirányító olyan Hello csomagot kap, melyben a forgalomirányító azonosító még nem szerepel a szomszédsági listájában, akkor megpróbál az új kezdeményező forgalomirányítóval szomszédsági kapcsolatot létesíteni.

Nézzük az 1. ábra R1 forgalomirányítóját! Amikor az OSPF-et engedélyezzük, akkor a már engedélyezett Gigabit Ethernet 0/0 interfész Down állapotból Init állapotba lép. Az R1 forgalomirányító elkezd Hello csomagokat küldeni valamennyi OSPF interfészén, hogy felderítse az OSPF szomszédjait és szomszédsági kapcsolatot alakítson ki velük.

A 2. ábrán az R2 Hello csomagot kap R1-től és ezt követően felveszi az R1 forgalomirányító ID-jét a szomszédsági listájába. Ezek után az R2 Hello csomagot küld R1-nek. A csomag tartalmazza az R2 forgalomirányító azonosítóját és ugyanezen interfészhez tartozó szomszédok listájában az R1 forgalomirányító azonosítóját.

A 3. ábrán R1 fogadja a Hello-t és hozzáadja R2 forgalomirányító azonosítóját a szomszédsági listájához. Azt is felismeri, hogy a saját forgalomirányító azonosítója szerepel a Hello csomag szomszédság listájában. Amikor egy forgalomirányító olyan Hello csomagot kap, amiben a saját forgalomirányító azonosítója szerepel a szomszédsági listában, akkor az Init állapotból áttér a kétirányú (Two-Way) állapotba.

A kétirányú állapotban végzett tevékenységek függenek a két forgalomirányító közötti kapcsolat típusától:

* Ha a két, szomszédsági kapcsolatban lévő, forgalomirányító között pont-pont kapcsolat van, akkor azonnal továbblépnek a kétirányú állapotból az adatbázis szinkronizációs fázisba.
* Ha a forgalomirányítók ugyanarra az Ethernet hálózatra kapcsolódnak, akkor először kijelölt forgalomirányítót (DR) és tartalék kijelölt forgalomirányítót (BDR) kell választaniuk.

Mivel R1 és R2 egymáshoz Ethernet hálózaton keresztül kapcsolódik, ezért DR és BDR választás történik. Ahogy azt a 4. ábra mutatja, R2 lesz a DR és R1 a BDR. Ez a folyamat csak olyan többes hozzáférésű hálózatokon zajlik, mint pl. az Ethernet LAN.

A Hello csomagokat folyamatosan küldik a forgalomirányítási adatok karbantartásának érdekében.

# Az OSPF tulajdonságai

## Az OSPF működése

Miért van szükség DR és BDR választásra?

A többes hozzáférésű hálózatok az OSPF LSA elárasztás szempontjából két kihívást is jelentenek:

* **Többszörös szomszédsági viszonyok** – Az Ethernet hálózatok több OSPF forgalomirányítót is összekapcsolhatnak egyetlen kapcsolaton. Ebben az esetben minden szomszédos forgalomirányítóval szomszédsági kapcsolatra lépni nem csak szükségtelen, de nem is célszerű. Hiszen ez ugyanazon hálózathoz kapcsolódó forgalomirányítók közötti túlzott mennyiségű LSA cserét eredményezne.
* **Túlzott LSA elárasztás** – A kapcsolatállapot alapú forgalomirányítók az OSPF indulásakor, vagy a topológia megváltozása esetén elárasztással küldik el LSA üzeneteiket. Ez az elárasztás túlzottá válhat.

A többszörös szomszédsági viszonyokból származó problémák vizsgálata érdekében nézzük az alábbi összefüggést:

Egyazon többes hozzáférésű hálózathoz kapcsolódó tetszőleges számú forgalomirányító (számukat jelölje *n*) esetén *n* (*n* – 1) / 2 szomszédsági viszony létezik.

Az 1. ábra egy öt forgalomirányítóból álló egyszerű topológiát mutat, melyek ugyanazon többes hozzáférésű Ethernet hálózathoz kapcsolódnak. Valamilyen, a szomszédsági viszonyok számát csökkentő megoldás nélkül, összesen 10 szomszédsági viszony alakulna ki közöttük:

5 (5 – 1) / 2 = 10

Ez így talán nem tűnik soknak, de ha újabb forgalomirányítókat adunk a hálózathoz, akkor a szomszédsági viszonyok száma drasztikusan nő, ahogy azt a 2. ábra is mutatja.

Hogy megértsük a túlzott LSA elárasztás problémáját, játsszuk le a 3. ábra animációját. Az animációban az R2 küld egy LSA-t. Ez az esemény minden más forgalomirányítót is LSA küldésére késztet. Az animáció nem mutatja azokat a kötelező nyugta küldéseket, melyeket az egyes LSA-k fogadása vált ki. Ha egy többes hozzáférésű hálózaton valamennyi forgalomirányító az összes fogadott LSA-t nyugtázza, majd továbbküldi, az kaotikus méretű forgalmat eredményezhet.

A többes hozzáférésű hálózatokon a szomszédsági viszonyok számának és az LSA elárasztás problémájának kezelésére megoldás a kijelölt forgalomirányító (DR). Többes hozzáférésű hálózat esetén az OSPF egy DR-t választ, hogy gyűjtési és elosztási pontként kezelje a küldött és fogadott LSA-kat. A DR meghibásodásának esetére egy BDR-t is választanak. Az összes többi forgalomirányító DROTHER-é válik. A DROTHER egy olyan forgalomirányító, amely se nem DR, se pedig BDR.

Játsszuk le a 4. ábra animációját, hogy lássuk a DR szerepét!

# Az OSPF tulajdonságai

## Az OSPF működése

A kétirányú állapotot követően a forgalomirányítók az adatbázis szinkronizációját lehetővé tevő állapotokon mennek végig. Addig, amíg a Hello csomagok a szomszédsági viszonyok kialakítására szolgáltak, addig a másik négy OSPF csomagtípus az LSDB-k cseréjére és szinkronizációjára.

A cserekezdeti (ExStart) állapotban mester-szolga kapcsolat alakul ki a megválasztott DR és BDR forgalomirányító, valamint a hozzájuk kapcsolódó forgalomirányítók között. A magasabb azonosítójú forgalomirányító tölti be a mester szerepét az adatcserélő (Exchange) állapotban. Az 1. ábrán az R2 válik mesterré.

Az adatcserélő állapotban a mester és a szolga forgalomirányítók egy, vagy több DBD csomagot cserélnek. A DBD csomagok a forgalomirányítók LSDB-jében szereplő LSA-k fejlécéről hordoznak információt. A bejegyzések vonatkozhatnak kapcsolatokra, vagy hálózatokra. Az LSA fejléc a következőkről tartalmaz információkat: a kapcsolatállapot típusa, az azt hirdető forgalomirányító, a kapcsolat költsége és egy sorszám. A forgalomirányító a sorszám alapján határozza meg a fogadott kapcsolatállapot hirdetés frissességét.

A 2. ábrán R2 forgalomirányító egy DBD csomagot küld R1-nek. Amikor R1 megkapja a DBD-t, akkor az alábbi tevékenységeket hajtja végre:

1. Egy LSAck csomag küldésével nyugtázza a DBD fogadását.

2. Ezután R1 egy DBD csomagot küld R2-nek.

3. R2 nyugtát küld R1-nek.

R1 összeveti a kapott adatokat a saját LSDB-jében tárolt adatokkal. Ha a DBD csomagban van a sajátjánál frissebb kapcsolatállapot bejegyzés, akkor a forgalomirányító átlép az adatfeltöltő (Loading) állapotba.

Például a 3. ábrán az R1 a 172.16.6.0 hálózattal kapcsolatos LSR-t küld R2-nek. Válaszul R2 egy LSU csomagban a 172.16.6.0 hálózatot érintő összes adatot megküldi. Az LSU fogadásakor R1 újból LSAck nyugtát küld. R1 egy új kapcsolatállapot bejegyzést vesz fel az LSDB-jébe.

Azt követően, hogy egy forgalomirányító valamennyi LSR-jét megkapta, a szomszédsági viszonyban lévő forgalomirányítókat szinkronizált, teljes állapotúnak (Full state) tekintjük.

Addig amíg a szomszédos forgalomirányítók folyamatosan kapják a Hello csomagokat, az LSA-kban küldött hálózat a topológia adatbázisban marad. A topológiai adatbázisok szinkronizálását követően, csak bizonyos helyzetekben küldenek frissítéseket (LSU-k) a szomszédoknak:

* Változást érzékelnek (inkrementális frissítés).
* Minden 30 percben.

# Egyterületű OSPFv2 konfigurációja

## OSPF forgalomirányító azonosító (router ID)

Az OSPFv2 egy 1991-ben az IPv4-hez bevezetett, kapcsolatállapot alapú forgalomirányítási protokoll. Az OSPF-et a RIP IPv4 forgalomirányító protokoll alternatívájának tervezték.

Az ábra azt a topológiát szemlélteti, amelyen a továbbiakban az OSPFv2 konfigurációját mutatjuk majd be. A soros interfészek típusa és sávszélessége nem szükségszerűen felel meg a napjainkban elterjedtebb kapcsolattípusoknak. A topológiában a soros vonalak sávszélességét a forgalomirányító protokoll metrika számításának és a legjobb út kiválasztási folyamatának bemutatása céljából választottuk.

A topológia forgalomirányítói olyan alapkonfigurációval rendelkeznek, amely az interfész címeket is tartalmazza. Pillanatnyilag egyik forgalomirányítón sincs sem statikus, sem pedig dinamikus forgalomirányítás konfigurálva. Az R1, R2 és R3 valamennyi interfésze (kivéve R2 loopback) az OSPF gerinc (backbone) területéhez tartozik. Az ISP forgalomirányító az adott terület átjárója az internet felé.

**MEGJEGYZÉS:**A topológiában a visszacsatolási (loopback) interfész szimulálja az internet felé irányuló WAN kapcsolatot.

# Egyterületű OSPFv2 konfigurációja

## OSPF forgalomirányító azonosító (router ID)

Egy OSPF tartományban minden forgalomirányítónak egy forgalomirányító azonosítóval (Router ID) kell rendelkezni. A forgalomirányító azonosítót megadhatja a rendszergazda, vagy automatikusan meghatározhatja maga a forgalomirányító is. A forgalomirányító azonosítót az OSPF forgalomirányítók használják:

* **Egyedi azonosításra** – A forgalomirányító azonosítót a forgalomirányítók az OSPF tartományon belül az egyes forgalomirányítók és az általuk küldött csomagok forrásának egyedi azonosítására használják.
* **A DR választási folyamat során** – Egy többes hozzáférésű LAN környezetben, a DR választására az OSPF hálózat kezdeti felépülésekor kerül sor. Amikor az OSPF kapcsolatok aktívvá válnak, akkor a legmagasabb prioritásúnak konfigurált OSPF eszköz lesz a kijelölt forgalomirányító (DR). Abban az esetben, ha nincs prioritás konfigurálva, vagy azonosság áll fenn, akkor a legmagasabb azonosítóval rendelkező forgalomirányítót lesz a DR. A második legmagasabb azonosítóval rendelkező forgalomirányító lesz a tartalék kijelölt forgalomirányító (BDR).

De hogyan határozza meg a forgalomirányító a saját azonosítóját? Ahogy azt az ábra is mutatja, a Cisco forgalomirányítók három szempont alapján határozzák meg az azonosítójukat, melyek preferencia sorrendben az alábbiak:

* Az azonosító az OSPF konfigurációja során be lett állítva a **router-id** *rid* forgalomirányító konfigurációs módbeli paranccsal. A *rid* érték bármilyen 32 bites IPv4 címformátumban megadott szám lehet. Ez a forgalomirányító azonosító megadásának ajánlott módja.
* Ha az azonosító nem lett közvetlenül meghatározva, akkor a forgalomirányító a használt loopback interfészekre konfigurált legnagyobb IP-címet választja. Ez a második legjobb módszer az azonosító meghatározására.
* Ha nincs loopback interfész konfigurálva, akkor a forgalomirányító bármelyik fizikai interfészre konfigurált legmagasabb IPv4-címet választja. Ez a legkevésbé ajánlott módszer, mert megnehezíti, hogy a rendszergazda könnyen megkülönböztethesse az egyes forgalomirányítókat.

Ha a forgalomirányító a legmagasabb IPv4-címet használja azonosítónak, akkor a hozzá tartozó interfésznek nem feltétlenül kell OSPF interfésznek lennie. Azaz ennek az interfész címnek nem kell szerepelnie az OSPF **network** parancsban, hogy azonosítóként lehessen használni. Az egyetlen megkötés, hogy az adott interfésznek aktívnak és működőképesnek kell lennie.

**MEGJEGYZÉS:** A forgalomirányító azonosító bár úgy néz ki, mint egy IP-cím, nem irányítható és az irányítótáblában sem szerepel, hacsak az nem egy olyan (fizikai, vagy loopback) interfész címe, melyet a **network** parancsban felsoroltak.

# Egyterületű OSPFv2 konfigurációja

## OSPF forgalomirányító azonosító (router ID)

Használjuk a **router-id** *rid* forgalomirányító konfigurációs parancsot, ha kézzel szeretnénk megadni IPv4 formában a 32 bites azonosítót. Az OSPF forgalomirányító ezután ezzel az ID-vel azonosítja magát.

Ahogy azt az 1. ábra mutatja, R1 az 1.1.1.1, R2 a 2.2.2.2 és R3 a 3.3.3.3 azonosítóval rendelkezik.

A 2. ábrán az 1.1.1.1 azonosító az R1-hez lett hozzárendelve. Használjuk a **show ip protocols** parancsot a forgalomirányító azonosítójának ellenőrzésére.

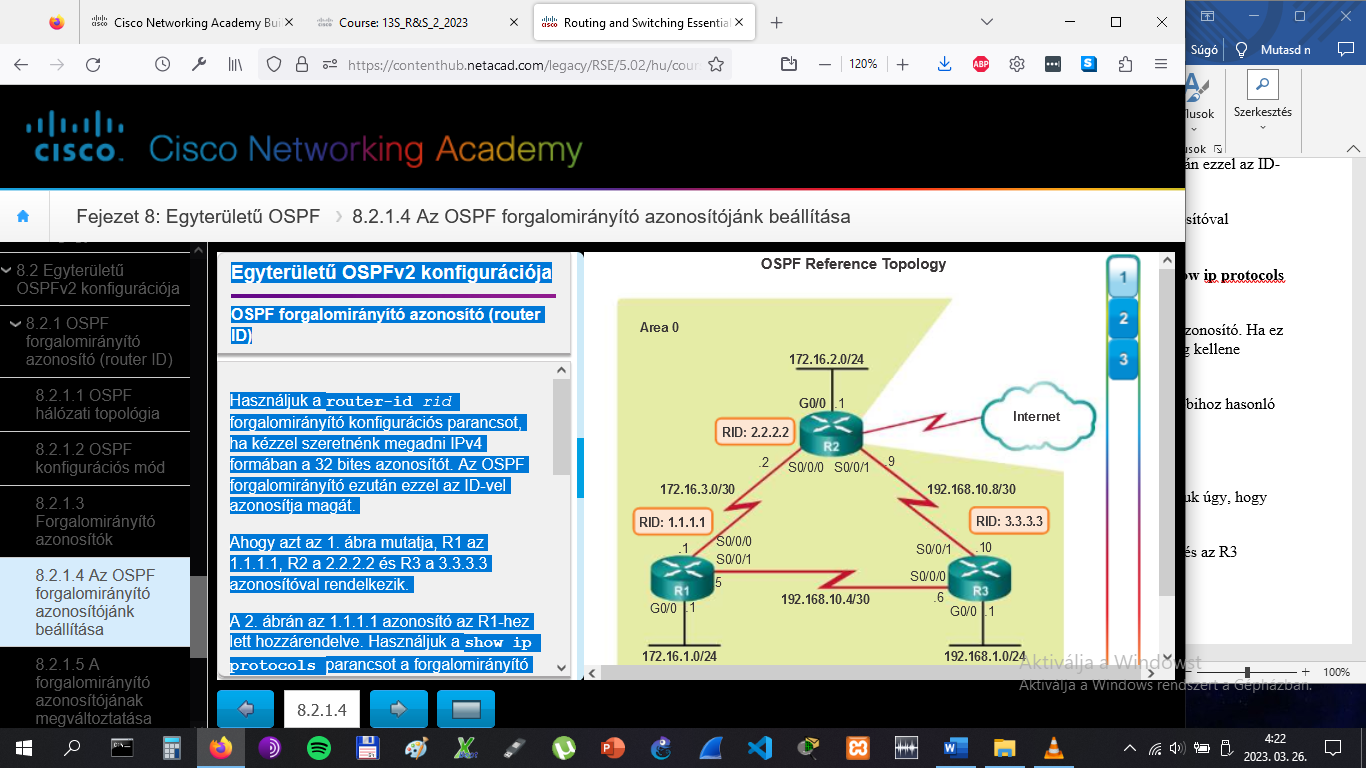
**MEGJEGYZÉS:** R1 forgalomirányítón nem lett beállítva semmilyen OSPF azonosító. Ha ez - a jelenlegi értékek mellett - megtörtént volna, akkor most az azonosítóját meg kellene változtatni.

Ha két szomszédos forgalomirányítónak ugyanaz az azonosítója, akkor az alábbihoz hasonló hibaüzenetet kapunk:

%OSPF-4-DUP\_RTRID1: Detected router with duplicate router ID.

A probléma feloldásának érdekében valamennyi forgalomirányítót konfiguráljuk úgy, hogy egyedi OSPF azonosítójuk legyen.

Használjuk a 3. ábrán a parancsszimulátort, hogy azonosítót rendeljünk az R2 és az R3 forgalomirányítókhoz!



# \*Az OSPF beállítása / OSPF konfigurálása \*Egyterületű OSPFv2 konfigurációja

## OSPF forgalomirányító azonosító (router ID)

Előfordulhat, hogy meg kell változtatni egy forgalomirányító azonosítóját, például, ha a hálózaton új azonosító-kiosztást szeretnénk alkalmazni. Ugyanakkor egy azonosító kiválasztását követően egy aktív OSPF forgalomirányító már nem engedi megváltoztatni azt, amíg a forgalomirányítót újra nem indítják, vagy az OSPF folyamatot nem törlik.

Figyeljük meg, hogy az 1. ábrán az érvényes azonsító a 192.168.10.5! Pedig az azonosítónak 1.1.1.1-nek kellene lennie.

A 2. ábrán az R1 forgalomirányítóhoz az 1.1.1.1 azonosító lett hozzárendelve. Vegyük észre az információs üzenetet, miszerint az OSPF folyamatot törölni kell, vagy a forgalomirányítót újra kell indítani! Ennek az oka az, hogy az R1 forgalomirányító már rendelkezik szomszédsági viszonyokkal, melyek kialakításánál 192.168.10.5 azonosítót módosításához. Ezen szomszédsági viszonyokat az új 1.1.1.1 azonosítóval újra kell egyeztetni.

Az OSPF folyamat törlése az ajánlott módszer az azonosító átállításához.

A 3. ábrán az OSPF forgalomirányítási folyamatot a **clear ip ospf process** privilegizált módbeli paranccsal törölhetjük. Ez arra kényszeríti az R1 forgalomirányítón futó OSPF-et, hogy áttérjen a leállás (Down), majd a kezdeti (Init) állapotba. Figyeljük meg a szomszédsági viszony változásait a teljes (full) állapotból a leállás (down) állapotba, majd az adatfeltöltés (loading) állapotból vissza a teljes (full) állapotba. Az azonosító megváltozását a **show ip protocols** paranccsal ellenőrizhetjük.

Változtassuk meg a 4. ábra parancsszimulátorának segítségével az R1 forgalomirányító azonosítóját!

# Egyterületű OSPFv2 konfigurációja

## OSPF forgalomirányító azonosító (router ID)

Az azonosító megadható visszacsatolási interfésszel is.

A loopback interfész IPv4-címét 32 bites alhálózati maszkkal (255.255.255.255) kell konfigurálni. Ez egyben egy hoszt utat is definiál. A 32 bites maszkkal konfigurált útvonalakat az OSPF forgalomirányítók nem hirdetik.

Az ábra arra mutat példát, hogy hogyan lehet az R1 forgalomirányítón ilyen utat konfigurálni egy visszacsatolási interfészre. Az R1 fiorgalomirányító a visszacsatolási interfész címét használja azonosítónak, feltételezve azt, hogy korábban nem volt más közvetlenül konfigurált, vagy tanult azonosítója.

**MEGJEGYZÉS:** Némelyik régebbi IOS nem ismeri a **router-id** parancsot, ezért az ilyen forgalomirányítókon a legjobb módszer az azonosító megadására a loopback interfész használata.

# Egyterületű OSPFv2 konfigurációja

## Egyterületű OSPFv2 konfigurációja

A **network** parancs határozza meg, hogy mely interfészek vegyenek részt az OSPF terület forgalomirányítási folyamatában. A forgalomirányító bármelyik interfésze, amely szerepel a **network** parancs utáni címlistában, küld és fogad OSPF csomagokat. Ennek eredményeként az interfész hálózat, vagy alhálózat címe bekerül az OSPF forgalomirányítási frissítéseibe.

A parancs alap szintaxisa **network** *hálózati\_cím helyettesítő\_maszk* **area** *terület\_azonosító*.

Az **area** *terület\_azonosító* az OSPF területre vonatkozik. Ha egyterületű OSPF-et konfigurálunk, a **network** parancs *terület\_azonosító* értékének valamennyi forgalomirányító esetében meg kell egyeznie. Bár tetszőleges területazonosítót választhatnánk, jó gyakorlat, ha ez egyterületű OSPF esetén a 0-t választjuk. Ez a megoldás nagyban megkönnyíti a helyzetünket, ha a későbbiekben a hálózatot többterületű OSPF-é kellene alakítani.

Az ábra a referencia topológiát szemlélteti.

# Egyterületű OSPFv2 konfigurációja

## Egyterületű OSPFv2 konfigurációja

Az OSPFv2 az OSPF interfészek engedélyezésekor a *hálózati\_cím helyettesítő\_maszk* argumentum kombinációt használja. Az OSPF tervezésénél fogva osztály nélküli, ezért a helyettesítő maszkot mindig meg kell adni . Amikor egy a forgalomirányítási folyamatban résztvevő interfészt határozunk meg, a helyettesítő maszk tipikusan az interfészre konfigurált alhálózati maszk inverze.

A helyettesítő maszk egy 32 bites bináris karakterlánc, mely meghatározza, hogy a forgalomirányítónak a cím illeszkedésének ellenőrzésekor a cím melyik bitjeit kell vizsgálnia. Az alhálózati maszkban a bináris 1 jelenti azt, hogy az illető bitpozíciónak illeszkednie kell, a 0 pedig, hogy nem. A helyettesítő maszkban ennek az ellentéte igaz:

* **Helyettesítő maszk 0 bit** – A címben az adott bitpozíciónak illeszkednie kell.
* **Helyettesítő maszk 1 bit** – A címben az adott bitpozíciónak nem kell illeszkednie.

A helyettesítő maszk számításának legegyszerűbb módja, ha az alhálózati maszkot kivonjuk a 255.255.255.255-ből.

Az 1. ábra példája a 192.168.10.0/24 hálózati címből számolja ki a helyettesítő maszkot. Ennek érdekében a 255.255.255.0 alhálózati maszkot kivonjuk a 255.255.255.255-ből, melynek eredményeként 0.0.0.255-öt kapunk. Ezért a 192.168.10.0/24-nek a 192.168.10.0 felel meg 0.0.0.255 helyettesítő maszkkal.

A 2. ábrán arra látunk példát, hogy a 192.168.10.64/26 hálózati címből hogyan számolhatjuk ki a helyettesítő maszkot. A 255.255.255.192 alhálózati maszkot itt is ki kell vonni a 255.255.255.255-ből, melynek eredményeként 0.0.0.63-at kapunk. Ezért a 192.168.10.0/26 -nak a 192.168.10.0 felel meg 0.0.0.63 helyettesítő maszkkal.

# Egyterületű OSPFv2 konfigurációja

## Egyterületű OSPFv2 konfigurációja

Többféle módon is megadható, hogy melyik interfészek vegyenek részt az OSPFv2 forgalomirányítási folyamatban.

Az 1. ábra mutatja azokat a parancsokat, melyekkel meghatározható, hogy az R1 forgalomirányító mely interfészei vegyenek részt a terület OSPFv2 forgalomirányítási folyamatában. Figyeljük meg a helyettesítő maszkok használatát a megfelelő interfészek hálózati cím alapján történő meghatározásában! Egyterületű OSPF-ről lévén szó az összes területazonosító értéke 0.

Másik lehetséges megoldásként az OSPFv2 a **network** *intf-ip-address* **0.0.0.0** *area terület\_azonosító* forgalomirányító konfigurációs paranccsal is engedélyezhető.

A 2. ábra arra példa, hogy hogyan lehet egy interfész IPv4-címét megadni a négy 0-s helyettesítő maszkkal. A **network 172.16.3.1 0.0.0.0 area 0** parancs R1 forgalomirányítón történő kiadása engedélyezi a Serial0/0/0 interfésznek a forgalomirányítási folyamatban való részvételét. Ennek megfelelően az OSPFv2 folyamat hirdetni kezdi ezen interfész hálózatát (172.16.3.0/30).

Az interfész meghatározásának az az előnye, hogy a helyettesítő maszk számítás így szükségtelenné válik. Az OSPFv2 az interfészcímet és az alhálózati maszkot használja a hirdetendő hálózat meghatározásakor.

Egyes IOS verziók a helyettesítő maszkok helyett megengedik az alhálózati maszkok használatát. Ebben az esetben az IOS konvertálja át az alhálózati maszkot helyettesítő maszkká.

A 3. ábrán lévő parancsszimulátorral hirdessük az R2-höz kapcsolódó hálózatokat!

**MEGJEGYZÉS:** A parancsszimulátor használata közben figyeljük meg az R1 (1.1.1.1) és az R2 (2.2.2.2) közötti szomszédsági viszonyt leíró információs üzeneteket. Az IPv4 címzési sémát követő forgalomirányító azonosítók használata megkönnyíti a szomszédok azonosítását.

# Egyterületű OSPFv2 konfigurációja

## Egyterületű OSPFv2 konfigurációja

Alapértelmezés szerint az OSPF üzenetek valamennyi OSPF engedélyezett interfészen kiküldésre kerülnek. Azonban ezen üzenetek ténylegesen csak ott kellenek, ahol az interfész egy másik OSPF forgalomirányítóhoz kapcsolódik.

Vegyük az ábra topológiáját. Az OSPF kiküldi az üzeneteket mind a három forgalomirányító G0/0 interfészén annak ellenére, hogy azokon a LAN-okon nincs is OSPF szomszéd. A felesleges üzenetek küldése azonban háromféleképen is hatással van a LAN működésére:

* **A sávszélesség pazarlása** – Felesleges üzenetek terhelik a rendelkezésre álló sávszélességet. Mivel ezek szórásos üzenetek, még a kapcsolók is valamennyi portjukon továbbítják azokat.
* **Erőforrás pazarlás** – A LAN valamennyi eszközének fel kell dolgoznia, majd el kell dobnia ezeket az üzeneteket.
* **Biztonsági kockázat** – A frissítések üzenetszórásos hálózaton való hirdetése biztonsági kockázatot jelent. Az OSFP üzenetek egy csomagelfogó program segítségével lehallgathatók. A forgalomirányítási frissítések módosíthatóak, majd visszaküldhetők a forgalomirányítónak, hogy hibás mértékkel és téves irányokkal elrontsák a forgalomirányítási táblákat.

# Egyterületű OSPFv2 konfigurációja

## Egyterületű OSPFv2 konfigurációja

Ahogy azt az 1. ábra is mutatja, a **passive-interface** forgalomirányító konfigurációs parancs használatával egy adott interfészen meggátolhatjuk a forgalomirányítási üzenetek küldését úgy, hogy a hálózatot más forgalomirányítók felé tovább hirdethetjük. Konkrétan a parancs egy adott interfészen leállítja a forgalomirányítási üzenetek kiküldését. Azonban a hálózat, amelyhez az adott interfész tartozik továbbra is hirdetésre kerül más interfészek forgalomirányítási üzeneteiben.

Vegyük például azt az esetet, amikor az R1-nek, az R2-nek és az R3-nak fölösleges OSPF üzeneteket továbbítania LAN interfészeire. A konfiguráció az R1 forgalomirányító G0/0 interfészét passzívra állítja.

Fontos tudnunk, hogy passzív interfészen keresztül nem alakulhat ki szomszédsági viszony. Ez azért van, mert ezeken az kapcsolatállapot üzenetek sem küldhetők, illetve nyugtázhatók.

A **show ip protocols** parancs segítségével ellenőrizhetjük a Gigabit Ethernet interfész passzív állapotát, ahogy azt a 2. ábra is szemlélteti. Figyeljük meg, hogy a G0/0 interfész most a lista passzív interfészek (Passive Interface) részében szerepel! A 172.16.1.0 hálózat pedig a listában továbbra is a hirdetett hálózatok között van, azaz útvonalként szerepel az R2 és R3 felé küldött OSPF frissítésekben.

**MEGJEGYZÉS:** Mind az OSPFv2, mind az OSPFv3 támogatja a **passive-interface** parancsot.

Konfiguráljuk passzívnak a 3. ábrán lévő parancsszimulátorral az R2 forgalomirányító LAN interfészét!

Alternatívaként valamennyi interfészt egyszerre is passzívvá tehetjük a **passive-interface default** parancs kiadásával. Ezesetben azon interfészeket, amiknek nem kell passzívnak lenniük újra engedélyezhetjük a **no passive-interface** parancs kiadásával.

Folytassuk a 3. ábrán a parancsszimulátorral, és konfiguráljuk R3 forgalomirányító LAN interfészét is passzívnak!

**MEGJEGYZÉS:** A parancsszimulátor használata közben figyeljük meg az OSPF állapot információs üzeneteket, ahogy az interfészek mind passzívvá, majd a két soros interfész újra aktívvá válik!

# 