

## 7.4. IP Multicast

Az Internetben a multicast nem megoldott kérdés. Finoman fogalmazva alig léteznek ezzel kapcsolatos szabványos megoldások. Csupán az utóbbi időben került a figyelem középpontjába, hiszen több előnnyel is jár. Egyrészt segítségével a hálózaton manuális konfiguráció nélkül kereshetünk bizonyos szolgáltatásokat (anycast), számottevően csökkenti a hálózat terheltségét és jól illeszkedik a multimédia konferenciákhoz. Ezt a lehetőséget maga az IETF már 1992-ben kihasználta, amikor az MBONE-on keresztül közvetítette San Diego-i tavaszi találkozóját a megjeleni nem tudóknak.

A Multicast két problémakörből tevődik össze, az egyik az állomások megváltozott feladatai, a másik a routing. Ilyen sorrendben tárgyaljuk őket.

### 7.4.1. Internet Group Membership Protocol (IGMP)

Az IGMP [RFC1112] az egyik legrégebbi és sokáig egyetlen a multicast-tal foglalkozó Internet szabvány, megvalósítása mára kötelező az állomásokban, az IPv6-ban pedig az ICMP szerves része.

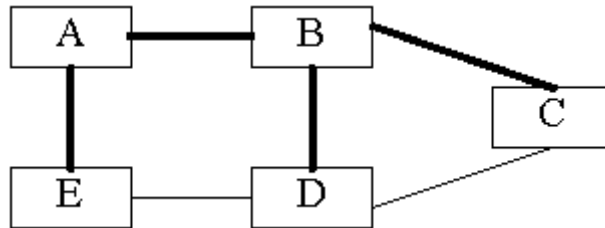
Nem broadcast link-ek esetén a multicast úgy valósítható meg, hogy minden a csoporthoz tartozó állomás számára egyesével elküldjük a multicast csomagot. Broadcast link-ek esetén azonban fontos, hogy a csomag csak egyszer haladjon végig a link-en. Az állomásoknak itt fel kell készülniük arra, hogy meghallják az összes D osztályú címmel feladott csomagot és kiszűrjék azokat, melyek nekik nem szólnak. Valamilyen módon azonban a router-ek tudomására kell hozni, hogy az adott link-en van-e valaki, aki tagja valamely multicast csoportnak, hogy azok egyáltalán körbeadják itt az annak a csoportnak szóló csomagokat. Erre való az IGMP.

Két IGMP üzenet létezik, az egyik segítségével a router-ek kérdezik le a csoporttagságot, a másikkal pedig az állomások válaszolnak, egy-egy választ küldve minden csoporthoz, melynek tagjai. Minthogy broadcast link-en mindegy, hogy hány tagja van az adott csoportnak, csak az fontos, hogy van-e vagy nincs, ha valamely állomás hallja, hogy más már jelezte tagságát a router-nek, egy olyan csoportra, aminek ő is tagja, akkor ő már nem jelez külön.

### 7.4.2. Multicast routing

A multicast routing legegyszerűbb módja az **árasztás**. [2] Ennek használatát már az OSPF-ben láthattuk, ott a topológiai információkat terítették így egymás között a router-ek. Az árasztás lényege az, hogy a bejövő csomagot minden interface-ünkön továbbadjuk (kivéve azt, amelyiken kaptuk), csak arra kell figyelni, hogy ha egy csomagot már továbbítottunk, akkor ne továbbítsuk még egyszer. Ehhez az OSPF időbélyegeket használ, erre itt nincsen lehetőségünk, hisz egy IP csomagba nem tehetünk időbélyeget. Nincs mese meg kell jegyezni a „néhány” legutoljára továbbított multicast csomagot, s ha valamelyik újra elérkezik hozzánk, nem továbbítjuk. A módszer hátránya éppen ez a lista, ami gyors hálózatokon meglehetősen hosszú lehet, másfelől azt ugyan garantálja, hogy egy router nem küld el egy csomagot kétszer, de azt nem, hogy nem is kapja meg kétszer, pedig ez is jó lenne.

Sokkal hatékonyabb megoldás a **feszítôfa** módszere. Ezt használják a transparent bridge-k is (errôl az Internetworking fejezetben volt szó), lényege az, hogy a hálózat fölött megállapítunk egy feszítôfát és a bejövô multicast csomagot e fa mentén továbbítjuk minden a fához tartozó interface-en, kivéve azon, amelyiken kaptuk. Minthogy a fa körmentes, egy csomag kétszer senkihez sem jut el. Ráadásul igen kevés információt kell tárolnunk, csupán minden interface-hez egy bitet, hogy az adott interface tagja-e a fának vagy sem. A hátrányok között említeném, hogy a módszer mindenhova eljuttatja az összes multicast csomagot, függetlenül attól, hogy arrafele van-e tagja a csoportnak, másfelôl pedig minden multicast forgalom a fa link-jein zajlik, más link-eken pedig egyáltalán nem, ami a hálózat egyenlôtlen terheltségéhez vezet. Kérdés még, hogy miként határozzuk meg a feszítôfát.



66. ábra. Feszítôfa a hálózatban

Másik eljárás a **visszirányú továbbítás** (Reverse Path Forwarding, RPF). Ez explicit módon minden forráshoz külön fát épít egy tetszôleges routing protokoll táblázata alapján. Ha multicast csomagot kapunk F forrástól az I interface-en, akkor

1. Ha az I interface vezet a legrövidebb úton S-be, akkor továbbítjuk a csomagot minden interface-en, kivéve I-n.
2. Ha nem, akkor eldobjuk a csomagot.

Tehát, ha C csomagot ad fel A-nak és E-nek, akkor ezeket elôször D-nek és B-nek postázza. Ezek, minthogy a C felé vezetô legrövidebb úton kapták a csomagot, továbbadják A-nak és E-nek, valamint egymásnak. Az egymástól kapott csomagokról nem tudják azonban, hogy ôk ezt a csomagot már továbbították, hiszen nem árasztásról van szó és nem jegyezzük a legutóbbi néhány csomagot. Viszont például B, aki D felôl kapta a csomagot láthatja, hogy C felé a legrövidebb út nem D-n át vezet, így ezt a csomagot eldobja, mondván, hogy a legrövidebb úton úgyis megkapja majd (ha eddig még nem kapta meg). Hasonlóan D is eldobja B-tôl kapott a csomagot. A csomag így eljut A-ba és E-be is, melyek feldolgozzák és, minthogy a C felé vezetô legrövidebb úton kapták, továbbadják egymásnak, ám minthogy ezt a másodpéldányt már nem a legrövidebb úton kapták, ezért eldobják.

A módszer elônye, hogy nem igényel semmi mást, mint egy teljesen tetszôleges routing protokoll táblázatát, sem külön memória, sem különösen nagy számítási teljesítmény nem kell hozzá. Ha a hálózat költségei szimmetrikusak, azaz a forrás felé ugyanannyi a költség, mint a forrás felôl, akkor a multicast csomag minden célponthoz a legolcsóbb úton jut majd el, minden forrásból külön feszítôfa keletkezik. Emiatt a hálózat is egyenletesebben terhelt.

Könnyen tovább javíthatunk az eljárásán, ha egy lépéssel messzebbre tekintünk. Ha a mi router-ünk nincs rajta egy szomszédos router és a forrás közötti legrövidebb úton, akkor annak a szomszédos router-nek teljesen fölösleges elküldeni a csomagot, úgyis eldobja majd. Ez az egy lépéssel messzebbre való kitekintés nem igényel túl sok kapacitást, például az OSPF adatbázisból könnyen kinyerhetô. Az RPF azonban még mindig a teljes hálózatot

beteríti a csomagokkal, s nem veszi figyelembe, hogy merre vannak tagjai az adott csoportnak és merre nincsenek.

Ezen segít, az **RPF tisztogatásokkal** való bővítése (RPF and prunes). Ebben az esetben az első csomagot az RPF szerint mindenkinek eljuttatjuk. Ezután minden router, akinek hálózatán nincs senki akinek a csomag szólt volna, tiltakozó csomagot küld visszafelé (ez a tisztogatás). Az a router, aki egy interface-én keresztül tiltakozó csomagot kapott, az adott forrástól az adott csoportcímre küldött csomagokat azon az interface-en keresztül már nem továbbítja többet. Ha egy router minden interface-éről kapott tiltakozást, maga is tiltakozni fog az adott forrás felé, így ő már teljesen kimarad az adott forrásnak az adott csoporthoz küldött csomagjaiból. A folyamat végén csak azok az állomások lesznek a fában, akik valóban igénylik a csomagokat.

Mindennek két hátránya van.

1. Az első (néhány) csomag mindenkinek (sokakhoz) eljut.
2. A router-eknek tárolniuk kell, mely interface-ükön kaptak már tiltakozást minden forráshoz és minden csoporthoz. Ezek az információk ideiglenesek, idővel törölni érdemes őket, ez esetleg megújító tiltakozásokat eredményez, ha mégsem kell arra továbbítani. Mindez sok memóriát és forgalmat okoz.

Ha azonban **fix gyökerű fákat** (Core Based Trees, CBT) használunk, akkor nem minden forráshoz és csoporthoz, hanem csoportonként csupán egy forráshoz kell letárolnunk a tiltakozásokat. Ebben az esetben ugyanis a csoporthoz egy fix gyökér tartozik, minden forrás ennek az állomásnak küldi el multicast csomagját (unicast csomagba csomagolva) és ez az állomás adja fel a tényleges multicast csomagot. Az adott csoporthoz tehát csak egy forrás létezik.

Azok, akik egy adott csoport forgalmát meg szeretnék kapni (azaz csatlakozni szeretnének a csoporthoz), csatlakozó üzenetet küldenek a gyökérnek. Ezt az üzenetet az összes közbeeső router feldolgozza. Ha ez az első csatlakozó üzenet, amit a router kap, akkor megjelöli az interface-t amin kapta, majd továbbítja a gyökér felé. Ha nem, akkor csak megjelöli az interface-t, hiszen őhozá már eljutnak az adott csoport csomagjai.

Egy becsomagolt multicast csomagnak nem kell eljutnia egészen a gyökérig, azon a ponton, ahol először belebotlik egy olyan router-be aki már rendelkezik megjelölt interface-ekkel, kicsomagolják és terjeszteni kezdik a fa mentén.

Ily módon minden csoporthoz egy fát építünk, ami miatt az adott csoport forgalma kizárólag a fa mentén zajlik, ezen több gyökér kijelölésével segíthetünk. A CBT nagy előnye, hogy csak azokhoz juttatja el a csomagot, akik kérik. Nem lemondani kell tehát a csomagot, mint az RPF esetén, hanem explicit módon kérni. Jól működik nagy és kis csoportok esetén egyaránt, a router-ekben pedig egészen kevés információt kell tárolni.

A következő részekben a fenti alapelvek implementációit tekintjük át.

### 7.4.3. Distance Vector Multicast Routing Protocol (DVMRP)

A DVMRP az MBONE routing protokollja [RFC1075]. Az MBONE egy kísérleti multicast gerinc, mely nem önálló fizikai összeköttetéseken, hanem a rendes unicast Internet fölött

létrehozott alagutakban (tunnel) működik. Alapvetően multicast-képes szigetekből áll, a multicast router-ek egymás között pedig egymásnak címzett IP csomagokba csomagolva adják át a továbbítandó csomagokat. Maguk a multicast router-ek a DVMRP-t futtatják, ami leginkább a RIP-hez hasonlít, de nem a célpontok felé vezető legrövidebb utat, hanem a források felől vezető legrövidebb utat számítja ki. Ezt az információt az RPF tisztogatásokkal működő változata használja fel a multicast csomagok továbbítására. A csomagokat azonban nemcsak a router minden interface-én hanem az alagutakon is továbbítják.

Az alagutakat a két router rendszergazdájának manuálisan kell konfigurálnia. Minden alagútnak három paramétere van: a másik végén lévő router, az alagút költsége és egy küszöbérték. Az alagút költségét a distance-vector protokoll használja fel a legrövidebb út számításához, a küszöbérték pedig a forgalom korlátozására szolgál. Ha egy csomag TTL mezője nem nagyobb, mint a küszöb, az adott alagúton nem továbbítják. Megegyezés szerint a szervezetek közötti alagutak 32, a régiók közöttiek 64, a kontinensek közöttiek pedig 128-as küszöbvel rendelkeznek, így egy 128-nál kisebb TTL-lel elküldött csomag biztos nem vesz majd igénybe óceán feletti vonalat.

Az MBONE kísérletnek indult, mára pedig több mint 10000 tagja van. A multicast router-ek feladatait többnyire közönséges munkaállomások látják el, melyeken a régi szép időkhez hasonlóan egy „mrouted” nevű program végzi a route-olást, a program a kísérlet része és ingyenes. Hiába kísérlet azonban, az MBONE-t ma már üzemszerűen használják, ennek segítségével előbb audio-, majd videoközvetítést hallhatunk-láthatunk az IETF gyűlésekről, ám mindennek véget kell érnie, hisz a DVMRP nem képes ilyen nagy hálózatot a siker esélyével kiszolgálni. Erre jobb protokollok és dedikált router-ek kellenek.

#### **7.4.4. Multicast OSPF (MOSPF)**

A legegyszerűbben a link-state protokollok alakíthatóak át multicast protokollokká, hisz ott a teljes adatbázis rendelkezésre áll. A MOSPF [RFC1584] működése megegyezik az OSPF működésével, mindössze néhány kiegészítéssel bővült. Ugyanúgy, ahogy az OSPF router-ek közölik egymással, hogy támogatják-e a nem 0 TOS alapján való route-olást, azt is kiderítik egymásról, hogy támogatják-e a multicast-ot, így lehetséges egyszerre OSPF és MOSPF router-ek alkalmazása is, természetesen a multicast funkcionalitás így nem lesz teljes.

Ha egy MOSPF router csomagot kap az F feladótól a C csoportra, akkor a topológiai adatbázis segítségével kiszámolja a célpontokhoz vezető legrövidebb utakat, majd a szükséges interface-eken keresztül továbbítja a csomagot. A topológiai adatbázisnak azonban csak azokat a részeit használja fel, melyeket MOSPF router-ek generáltak. Ha így a terület 2 részre szakad, akkor a multicast forgalom nem tud eljutni az egyik félből a másikba; a MOSPF nem definiál az MBONE-hoz hasonló alagútrendszert, ez csak bonyolítaná a protokollt. Minthogy a MOSPF kiegészítés nem érinti a router hardware elemeit, egyszerű software cserével megoldható. A tervezők arra számítottak, hogy az átmeneti időszak rövid lesz.

A csoporttagságok információit egy új link-state rekordban terjesztik a router-ek, melyben a link megbízott router-e felsorolja azokat a csoportokat, melyeknek van tagjuk az adott link-en. A területhatáron lévő router-ek ezeket a rekordokat is összegzik, és csak azt közlik a gerincre, hogy a területen milyen csoportoknak vannak tagjai.

Az AS határán lévő router-ek nem terjeszthetik befelé az összes Internet csoportot, ez túl sok információ lenne. Az ilyen router-ek inkább automatikusan minden csoport tagjának

tekintetnek, így minden multicast csomagot megkapnak és tetszésük szerint továbbíthatják vagy elvethetik azokat. Hasonlóképpen a területhátáron lévő router-ek területük összes csoportjának tagjai, így a területen lévő összes csoportnak feladott multicast csomagot megkapják. Így a többi területen lévő csoportok listáját nem kell terjeszteni a területen belül.

Ha két egyenlő költségű út létezik és ezek közül véletlenszerűen választanak a router-ek, mint az unicast esetben, előfordulhatna, hogy két router más-más utat választ, más feszítőt kap eredményül, ami inkoherens döntéseket eredményezne. Ezért a protokoll tartalmaz egy algoritmust, ami alapján az egyenlő költségű utak közül a költségen kívüli szempontok alapján, egyértelműen lehet választani.

A MOSPF nagyon sokat tudó protokoll, egyesek szerint túl sokat tudó. Hiszen minden a területen lévő csoporthoz, minden forrásból külön fát kell kiszámolni. Egy terület javasolt maximális mérete 200 link, ennyi forrás lehet, csoport akár még több is. Ez komoly számítási teljesítményt köthet le. A MOSPF fejlesztői azzal védekeznek, hogy ezeket a számításokat csak akkor végzi el a protokoll, amikor kell, tehát, ha az első csomag megérkezik az adott forrásból az adott csoportra. A számítások eredményét pedig cache-ben tárolják a router-ek, a cache akkor törlődik, ha már régen nem jött ugyanabból a forrásból ugyanabba a csoportba csomag, illetve topológiai változás esetén. Azok a router-ek melyeket nem érint az adott fa, sohasem fogják elvégezni a számításokat. Csupán az idő és a tapasztalat mondja majd meg, hogy a számítási teljesítmény elegendő lesz-e.

#### **7.4.5. Protocol Independent Multicast (PIM)**

Míg a MOSPF elsősorban az AS-en belüli multicast routing témakörét célozta meg és fő feladatának a legrövidebb út megjelölését tekintette, addig a PIM inkább globális szinten keres megoldást és fő problémája a routertáblázatok mérete. A multicast csoporttól függően két verziója létezik, az egyik a sűrű (dense mode), a másik a ritka (sparse mode). Ezek a kifejezések azt takarják, hogy egy csoportnak mennyi tagja van az Interneten. Ha nagy az esélye annak, hogy egy adott területen van tagja a csoportnak, akkor az a csoport sűrű, ha pedig kicsi, akkor ritka. Ennek a két szituációnak megfelelően két külön protokoll létezik.

A sűrű PIM roppant egyszerű. [34] Minthogy a csoportnak mindenfelé vannak tagjai, nem okozunk túl sok fölösleges forgalmat, ha elárasztjuk a hálózatot a csomagokkal., éppen ezért az RPF tisztogatásokkal való kiegészítést használjuk. Ennek hibája, hogy a tiltakozások listáját időről időre el kell felejteni, hogy új tagok jelentkezése esetén hozzájuk is eljusson a csomag és a tiltakozás elmaradásával jelezzék, hogy kell nekik az adás. A lista elfelejtése azonban azzal jár, hogy számos tiltakozás újragenerálódik, ami nagy forgalmat okoz. Sűrű csoport esetén azonban nem olyan nagyot, hisz kevés a tiltakozó állomás. A sűrű PIM egyszerűbb, mint a DVMRP, mert nem működtet önálló unicast routing protokollt, adatait a létező protokoll táblázatából veszi. (Ezért protokoll-független multicast a neve.) Pontos működése tehát:

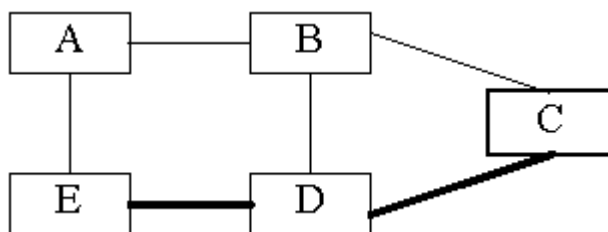
1. Ha multicast csomagot veszünk az F forrásból, az I interface-en a C csoportnak, először megnézzük a táblázatban, hogy F felé az I interface-en át küldünk-e unicast csomagokat.
2. Ha nem, eldobjuk a csomagot és az I interface-en elküldünk egy tiltakozás(F,C)-t.
3. Ha igen, akkor minden olyan interface-en, amire még nem kaptunk tiltakozás(F,C)-t, elküldjük a csomagot, kivéve I-t. Ha nincs már ilyen interface, akkor I-re postázunk egy tiltakozás(F,C)-t.

Ha egy router-ben kevés a memória, néhány régóta nem használt tiltakozási listát nyugodtan elfeledhet, ennek maximum a tiltakozások újraküldése lesz az eredménye.

Ha két szomszédján keresztül egyenlő költségű utak vezetnek egy adott forráshoz (a fenti lista 1. pontjában nem lehet dönteni), nemes egyszerűséggel úgy választ, hogy csak a nagyobb IP című szomszédjától fogad el csomagokat.

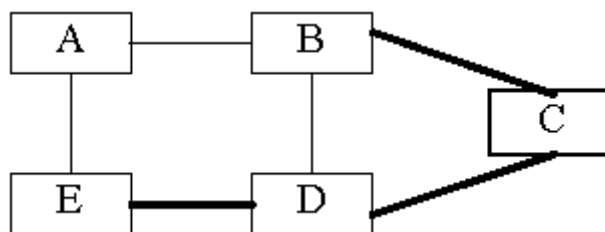
A ritka PIM nem működhet úgy, mint a sűrű, tehát nem épülhet tiltakozásokra, hisz ehhez mindenkinek el kell juttatni az első csomagot, ami egy háromszemélyes videokonferencia esetén kimondottan felesleges a teljes Internetben. A ritka PIM emiatt a CBT-re épül. [2] [35]

Minden csoport rendelkezik egy találkozási ponttal (Rendezvous Point, RP), ennek küldenek csatlakozó üzenetet a csatlakozni vágyók, melyeket a közbeeső router-ek feldolgoznak. Ha tehát az adott csoport RP-je C és E csatlakozni kíván, az elküldött üzenet mentén kiépül a továbbítási útvonal; a feszítőfa a következőképpen néz ki.



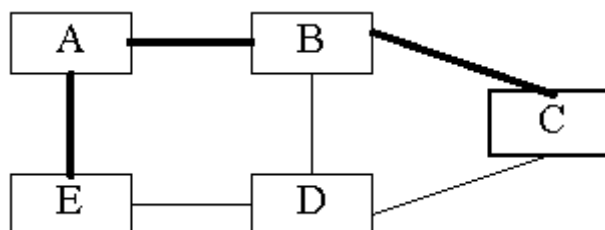
67. ábra. Ritka PIM 1.

Ha B adni kíván, nem ismeri a csoport tagjait, csupán az RP-t, így első csomagját becsomagolja egy regisztrációs üzenetbe, amit elküld C-nek. Az RP erre egy csatlakozó üzenetet küld B-nek, új ágat adva ezzel a fához.



68. ábra. Ritka PIM 2.

Ez azonban B-ből nem az optimális útvonal E felé, de látható, hogy egyetlen olyan router sem tagja a fának, aki nincs az RP és valamely tag között. Ha E úgy látja, hogy már nagyon sok csomagot kapott B-től, küldhet B felé egy csatlakozó üzenetet, C felé pedig egy tiltakozót. Így továbbra is összeköttetésben van az RP-vel, hisz tudja, hogy B tagja a fának, tehát továbbra is minden csomagot meg fog kapni, csak más úton. B felől azonban jelentős a teljesítménynövekedés.



### 69. ábra. Ritka PIM 3.

A B által küldött csomagok azon célpontok felé, melyek E-hez hasonlóan áttértek az RP-ről B-re az optimális úton haladnak. A fa gyökere ezen állomások számára nem az RP, hanem B. Természetesen az RP fa nem szűnt meg, más források továbbra is először C-nek küldik csomagjaikat, valamint B is elküldi C-nek saját datagramjait, hogy azon célpontokhoz is eljussanak, akik nem nyergeltek át rá. Mégis, az a lehetőség, hogy a fix gyökerű fából áttérhetünk egy RPF fába, ahol a forrás a gyökér, jelentős haladás és a ritka PIM és a CBT közti fő különbség. Ezenfelül a CBT [35] hard-state (az üzeneteket nyugtázni kell, a nem nyugtázott üzeneteket újraadjuk), míg a ritka PIM soft-state (az üzeneteket nem nyugtázzuk, de periodikusan ismételtjük, elmaradásuk esetén lassan kiöregítjük őket).

A (ritka és sűrű) PIM üzenetek hordozói az IGMP csomagok, ám a konkrét csomagformátumok még nem készültek el.