© Kiskapu Kft. Minden jog fenntartva

Hálózatok (17. rész)

Forgalomirányítás mozgó gépek esetén, adatszóró, többesküldéses forgalomirányítás, torlódásvédelem

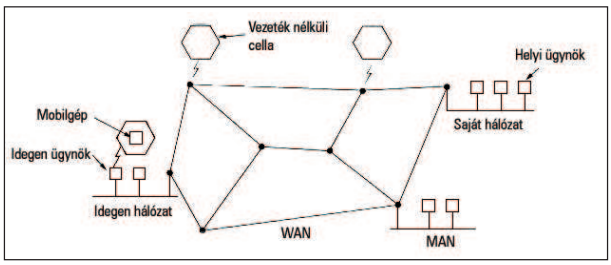
A sorozatnak ebben a részében a mozgó gépekkel kapcsolatos problémákat feszegetjük, majd megnézzük, milyen módszerek vannak arra, hogy egyszerre ne csak egy hoszt számára küldhessünk csomagokat. Ezenkívül szó lesz még a hálózatok legnagyobb mumusaként számon tartott jelenségről is: a torlódásról.

Forgalomirányítás és vándorló gépek

Eddig olyan forgalomirányítási mechanizmusokat tárgyaltunk, amelyek feltételezték azt, hogy a gépek „otthonülő életmódot" folytatnak, azaz nem vándorolnak szerte a világban. A mai mobilizálódó világban azonban ez elképzelhetetlen, hiszen egyre több felhasználó érzi magát kellemetlenül, ha a buszon utazva mobiltelefonjának segítségével nem nézhetné meg e-mail-jét, vagy esetleg nem léphetne be SSH-val az otthoni számítógépére.

Az *1. ábrán* egy nagy kiterjedésű hálózatot láthatunk. Tegyük fel, hogy ez a hálózat az egész világot behálózza, vagy legalábbis azt a területet, ahol a gépek vándorolhatnak. Ezt a hálózatot sokféle felhasználó használja. Vannak olyanok, akik a hálózathoz réz, illetve optikai kábel segítségével csatlakoznak, és sohasem hagyják el az otthonukat. Olyan felhasználó is akad, aki ugyan mindenfelé bolyong a világban, mégis csak úgy használja a hálózatot, hogy előbb fizikailag kapcsolódik hozzá (azaz a hálózaton tartózkodás ideje alatt nem változtatja a helyzetét). És persze vannak az „örök utazók" is, akik úgy szeretnének a hálózaton ügyködni, hogy közben folyamatosan mozogásban vannak. A felhasználók utóbbi két csoportját együtt ***mozgó felhasználónak (mobile user)*** nevezzük. Nézzük, miként juttathatjuk célba a mozgó felhasználóknak szánt csomagokat.

Az alapötlet az, hogy minden felhasználó rendelkezik egy állandó lakcímmel. A feladat az, hogy ha a felhasználó nem tartózkodik otthon, akkor a neki szánt csomagokat célba juttassuk, bárhol is legyen a felhasználó az adott pillanatban. Ehhez persze elengedhetetlen, hogy valamiképp rátaláljunk. Ehhez a behálózandó területet fel kell osztanunk körzetekre, vagy valamilyen más egységekre. Fontos, hogy a körzetek egymással diszjunktak, azaz a felhasználó egyszerre csak egy körzetben lehet. Minden ilyen körzetben két ügynök dolgozik: a ***hazai (home agent)*** és az ***idegen ügynök (foreign agent)***. A hazai ügynök azokat a felhasználókat tartja nyílván, akiknek az állandó lakhelyük a körzetében van, viszont jelenleg nem tartózkodnak otthon. Az idegen ügynök azokkal a felhasználókkal foglalkozik, akik ugyan máshol laknak, de jelenleg az ügynök körzetében tartózkodnak. Amikor egy felhasználó új körzetbe lép, akkor mindig fel kell vennie a kapcsolatot a helyi idegen ügynökkel. Először tehát meg kell tudnia az ügynök címét. Ez vagy úgy történik, hogy az idegen ügynök folyamatosan küldi szét a saját címét, vagy arra vár, hogy az újonnan belépő gazdagép megkérdezze, „van-e errefelé egy idegen ügynök?".



1. ábra Egy nagy kiterjedésű WAN, amelyhez MAN-ok, WAN-ok illetve vezeték nélküli cellák kapcsolódnak

Amint megvan a cím, a mozgó felhasználónak regisztrálnia kell magát. Ez gyakorlatilag abból áll, hogy elküldi az állandó lakcímét, az aktuális adatkapcsolati réteg címét, illetve szükség szerint valamiféle hitelesítőt. Ezután az idegen ügynök elküld egy csomagot a felhasználó lakcíméhez tartozó hazai ügynöknek, amelyben közli, hogy itt van egy felhasználója. A hazai ügynök ezután felírja az idegen ügynök címét, és elvégzi a hitelesítő ellenőrzését, hogy megbizonyosodjon, nem hazudott-e a felhasználó a kilétével kapcsolatban. Ha ez rendben van, a hazai ügynök visszaküld egy nyugtát. Amint ez visszaérkezik az idegen ügynökhöz, az tudatja a felhasználóval, hogy a regisztráció sikeresen befejeződött.

Ezek után egy tetszőleges forrás és a mozgó felhasználó között a kommunikáció a (*2. ábra*) szerint fog zajlani. Az első lépésben a forrás a felhasználónak csak az otthoni címét ismeri, így a csomagot oda küldi, amit a hazai ügynök elfog. A következő lépésben az ügynök ezt a csomagot egy másik

© Kiskapu Kft. Minden jog fenntartva

csomagba ágyazza, és ezt elküldi az idegen ügynök számára. (A csomagok más csomagokba való ágyazását alagút továbbításnak nevezzük. Ezzel az eljárásban a sorozat egy későbbi részében majd részletesen is megismerkedünk). Amint ez megérkezik az idegen ügynökhöz, az kiveszi az eredeti csomagot, és átadja a felhasználó számára. Ezután az idegen ügynök felveszi a kapcsolatot a forrással, és közli vele, hogy a továbbiakban a felhasználónak szánt csomagokat ne az otthoni címre továbbítsa, hanem inkább ágyazza be őket egy olyan csomagba, amelynek a címzettje az idegen ügynök. Innentől kezdve a kommunikáció már nem a hazai ügynökön keresztül fog zajlani, hanem közvetlenül a forrás és az idegen ügynök között.

A gyakorlatban nem ez az egyetlen megoldás a mozgó gépek forgalomirányításának problémájára. Rengeteg ilyen protokoll létezik, és ezek rengeteg szempontban térnek el egymástól. Ilyen szempont például az, hogy a munka oroszlánrészét kire bízzuk: az útválasztókra, vagy a gépekre. A protokollok különböznek még a csomagok másik címre történő irányításának mikéntjében is. Egyes protokollok például nem használják az alagút továbbítást, hanem egyszerűen csak átírják a csomag célcímét. A mozgó gép protokollok közötti különbségeket sok-sok oldalon keresztül tárgyalhatnánk.

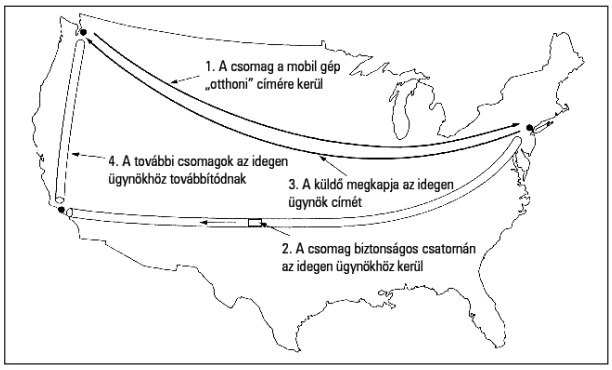
Adatszóró forgalomirányítás

Az ***adatszórás (broadcasting)*** fogalmával már találkozhattunk a közegelérési alréteg tárgyalásakor. Ott olyan LAN-okkal foglalkoztunk, ahol a gépek egy közös csatornát használtak, így mindenki hallott mindenkit, de csak akkor figyelt, ha az üzenet neki szólt. A hálózati rétegben is szükség lehet adatszórásra. Tegyük fel, azt szeretnénk, hogy a hálózatban lévő összes gép órája pontosan járunk. Elmegyünk tehát, és szerzünk egy atomórát, majd beállítjuk úgy, hogy bizonyos időközönként minden gépnek elküldje a pontos időt. (Persze a gépek órája így sem fog teljesen pontosan járni, késni fognak, mivel a csomagok továbbítása időbe telik. Most azonban a példa kedvéért tekintsünk el ettől a problémától).

Miként lehet azonban egy csomagot egyszerre minden gépnek elküldeni? Alkalmazhatjuk például a „favágó” módszert, azaz egyenként minden gép számára elküldjük a pontos időt tartalmazó csomagot. Ez azonban rendkívül sávszélesség pazarló megoldás, arról nem is beszélve, hogy az atomóránknak rendelkeznie kell egy listával, amely az összes gép címét tartalmazza. A módszernek mégis van egy nagy előnye: nem igényel semmiféle alhálózat oldali támogatást, tehát ez minden típusú hálózatban alkalmazható. Sőt, az is lehet, hogy az adatszórás megvalósítására ez lesz az egyetlen járható út. Ha azonban van más megoldás is, akkor inkább azt válasszuk.

A másik nagyon kézenfekvő ötlet az elárasztás alkalmazása, azaz a csomagot az útválasztó az összes kimenetén továbbítja. Sajnos a probléma itt is ugyanaz mint a forgalomirányítás esetében: a csomagok hihetetlen mértékben elszaporodnak, és ez a sávszélesség rovására megy.

Hatékonyabb megoldásnak ígérkezik a ***többcélú forgalomirányítás (multidestination routing)***, ahol egy csomagnak nem csak egy címzettje lehet. Ha egy útválasztó egy több címzettet tartalmazó csomagot kap, akkor először kijelöli azokat a kimeneteit, amelyeken a csomagot továbbítania kell. Egy



2. ábra Forgalomirányítás mozgó felhasználó esetén

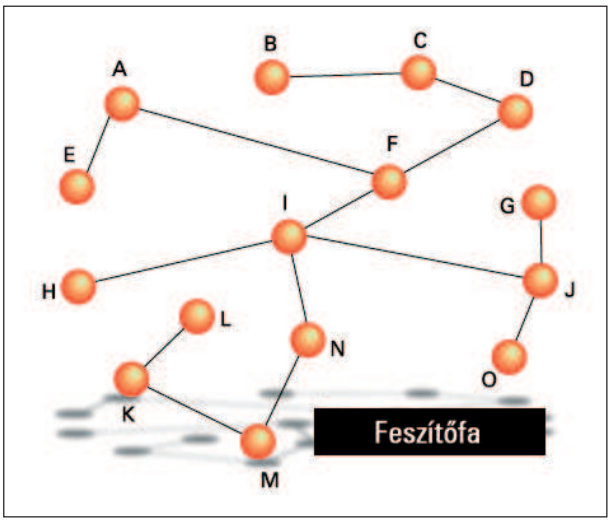
kimenet csak akkor lesz kijelölve, ha van olyan célcíme a csomagnak, amely felé azon a kimeneten vezet a legjobb út. Ezután az útválasztó minden kijelölt port számára előállítja a csomag másolatát, de úgy, hogy csak azokat a célcímeket hagyja benne, amelyeknek az adott kimeneten kell igénybe venniük. Az utolsó ugrásnál, még mielőtt a csomag a gépet elérné, már csak egy címet fog tartalmazni, így a gépek már csak egy közönséges, egy célcímet tartalmazó csomagot fognak kapni. Ha belegondolunk, ez az eljárás az előbb említett favágó módszer továbbfejlesztett változata. Itt is szükség van ugyan a gépek címeit tartalmazó listára, viszont az egy irányba tartó csomagokat csak egyszer kell továbbítanunk.

A legjobb eredményt mégis akkor érjük el, ha az útválasztók az adatszórásra kijelölt csomagot csak azokon a portjukon továbbítják, amely része az alhálózat valamely feszítőfájának, például az adatszórást kezdeményező útválasztóhoz tartozó nyelőfának. (Ha visszaemlékszünk, a feszítőfa az alhálózatnak egy olyan része, amelyben minden router benne van, viszont nem tartalmaz hurkokat. Minden útválasztóhoz csak egy út tartozik). Kivéve persze azt a portot, amelyen az adatszórásra ítélt csomag megérkezett. Ez a módszer azonban feltételezi, hogy az összes útválasztó ismeri legalább egy feszítőfát az alhálózatból. Távolságvektor alapú forgalomirányítás esetén sajnos az útválasztók számára nem áll rendelkezésre ilyen információ, így ott ez a módszer nem használható.

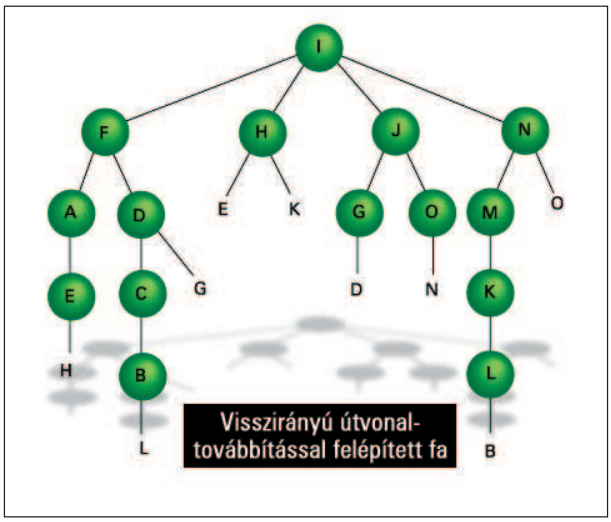
Létezik azonban egy olyan egyszerű és hatékony algoritmus, ahol nincs szükség arra, hogy az útválasztók ismerjenek legalább egy feszítőfát is. Ez az eljárás ***visszirányú továbbítás (reverse path forwarding)*** néven híresült el.

Az alapötlet az, hogy ha az útválasztó egy adatszórásra ítélt csomagot azon a porton kap, amelyik egybeesik az adatszórás forrása felé vezető legjobb úttal (vagy másképp fogalmazva, az adatszórás forrása felé menő csomagokat az útválasztó erre a kimenetre irányítaná), akkor valószínűsíthető, hogy most találkozik először ezzel a csomaggal. Mivel ez az első példány, amit megkapott, elárasztja. Ha azonban ez a csomag egy olyan porton érkezik, amelyik nem esik egybe a forrás felé vezető legjobb úttal, ésszerű feltételezni, hogy ez a csomag viszont másodpéldány, így nyugodtan megszabadulhatunk tőle.

Hogy senki se kételkedhessen az algoritmus működésének helyességében, bemutatunk egy egyszerű példát. A *3. ábrán* láthatunk egy alhálózatot, pontosabban annak az I szerinti nyelőfáját. Ez a gráf azt mutatja, hogy az I-ből miként jut-



3. ábra Egy alhálózat I csomópont szerinti nyelőfája



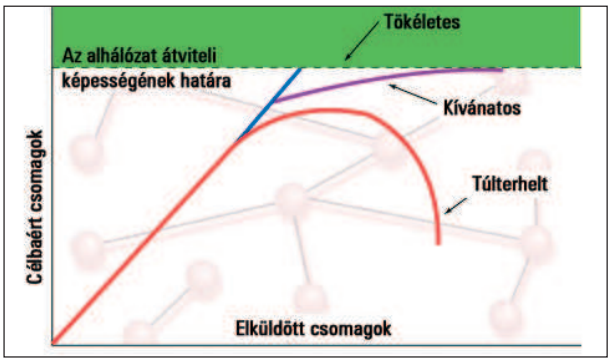
4. ábra A visszirányú továbbítás algoritmusának működése

© Kiskapu Kft. Minden jog fenntartva

hatunk el a többi csomópontba a lehető legkisebb költségű úton. A *4. ábra* az algoritmus működését szemlélteti. Tegyük fel, hogy az I adatszórást kezdeményez, így összes szomszédjának elküldi a csomagot. Ezt látjuk a 4. ábrán szereplő fa második sorában. Mivel az I szomszédaihoz a csomagok a lehető legrövidebb úton érkeztek meg, ezért egyrészt őket az ábrán egy nagy zöld pötty segítségével kiemeltük, másrészt ők is elküldik minden szomszédukhoz a csomagot (3. sor). Ezek közül megint kijelöltük azokat a pontokat, amelyekhez a csomag az I-től a legrövidebb úton érkezett. A következő lépésnél már csak ezek az útválasztók végzik el az elárasztást.

Érdekes, hogy az E útválasztóhoz az adatszórásra ítélt csomag már az algoritmus harmadik lépésénél megérkezik, az E útválasztó mégsem végzi el az elárasztást, ugyanis az EH él nem része a nyelőfának (habár része az alhálózati topológiának). Amikor azonban az E útválasztó az A-tól kapja a csomagot, akkor elvégzi az elárasztást, hiszen az AE él

© Kiskapu Kft. Minden jog fenntartva



4. ábra Ha az alhálózaton a forgalom a kapacitás alatt van, akkor az elküldött és a kézbesített csomagok egymással egyenesen arányosak. Ellenkező esetben torlódás lép fel, és a csomagok nagy többsége sosem fog célba jutni.

szerepel a feszítőfában. Az algoritmus hatékonysága tehát némileg elmarad az előbb bemutatott, a nyelőfát pontosan követő algoritmustól, viszont nem is ad sokkal rosszabb eredményt. A nyelőfát követve az algoritmus 4 lépés alatt végezne, szemben a mostani 5 lépéssel. Ez ugyan jobb eredmény, de viszont az útválasztóknak nem kell ismerniük az alhálózat egyetlen feszítőfáját sem. Ezenkívül sokkal jobb megoldás, mint az észnélküli elárasztás használata, mivel magától leáll, és nincs szükség belső mechanizmusra, amely leállítaná a végtelenig tartó csomagképződést.

Többesküldéses forgalomirányítás (multicasting routing)

Mi a helyzet akkor, ha nem mindenkinek, hanem csak a gépek egy kisebb csoportjának szeretnénk csomagot küldeni. Ilyesmire is gyakran szükségünk lehet, például ha egy adatbázist elosztottan szeretnénk tárolni. (Tipikusan ilyen adatbázist használ a tartományneveket az IP címekkel összekapcsoló DNS is, amelyről sorozatunk utolsó szakaszában, az alkalmazási réteg tárgyalásakor foglalkozunk). Ekkor az adatbázis kezelését végző folyamatoknak időnként szüksége lehet arra, hogy a többi, a munkában szintén részt vevő folyamatoknak üzeneteket küldjenek.
Ha a munkában közösen résztvevő folyamatokat futtató gépek száma viszonylag kicsi, akkor a probléma megoldható azzal, hogy a folyamatok küldhetnek egymásnak egyszerű két pont közötti üzeneteket. Ha a csoport nagy, akkor a módszer nem hatékony. Próbálkozhatunk adatszórással is, kivéve ha a hálózatban szereplő gépek száma nem elenyészően kicsi a teljes hálózathoz viszonyítva. Ilyenkor ugyanis az üzenet a gépek legnagyobb részét nem is érdekli, az adatszórás tehát rendkívül pazarló lenne.
A megoldást a ***többesküldésnek (multicasting)*** nevezett technológia jelenti, amely lehetővé teszi, hogy csomagot küldhessünk a gépek egy jól meghatározott csoportjának. Ehhez persze először valami módon meg kell határoznunk ezeket a csoportokat, ez azonban a forgalomirányítás szempontjából nem túlzottan érdekes, így erre most nem térünk ki külön. Ha egy folyamat belép egy csoportba, akkor azt tudatnia kell a gépével. A többesküldéses forgalomirányításban alapvetően fontos, hogy minden útválasztó tudja, melyik hoszt melyik csoportba tartozik. Itt rögtön fel is merül egy elvi kérdés: a gépnek kelljen-e szólni az útválasztónak, hogy egy új csapatnak kötelezte el magát, vagy az útválasztók kérdezzék ki gépeiket hovatartozásukról? Bárhogy is valósuljon ez meg a gyakorlatban, az útválasztók ezt az információt közlik az alhálózat további csomópontjaival.
A forgalomirányítás úgy működik, hogy az útválasztók kiszámítják az alhálózat egy feszítőfáját. Amikor egy többesküldésre szánt csomag megérkezik az első útválasztóhoz, az a saját feszítőfájából eldobja azokat az utakat, amelyek olyan útválasztókhoz vezetnek, akiknek nincs olyan gépük, amelyik tagja lenne a csoportnak. A csomagot ezután az így létrejött megcsonkított feszítőfa mentén kell továbbítani.

Torlódásvédelem

Egy ***torlódás (congestion)*** kialakulásánál nehezen lehet elképzelni nagyobb katasztrófát egy alhálózat életében. Nem csak arról van szó, hogy az alhálózat a csomagokat lassabban fogja célba juttatni, hanem az is előfordulhat, hogy egyszerűen képtelen lesz ellátni a feladatát, és szinte egy csomag sem fog eljutni a rendeltetési helyére.
Az *5. ábrán* látható grafikonon a kézbesített csomagokat ábrázoltuk az alhálózatba beadott csomagok számának függvényében. Ha ez a szám kisebb, mint az alhálózat maximális kapacitása, akkor a két érték között egyenes arányosság van. Ha azonban nagyobb, akkor fellép a torlódás, és az alhálózat teljesítőképessége nagyban visszaesik.
Torlódást sokféle esemény kiválthat. A leggyakoribb oka az, hogy egy helyen az alhálózatba hirtelen nagy mennyiségű csomag kezd beáramlani. Ha egy útválasztó hirtelen sok csomagot kap, akkor kialakul egy várakozási sor, azaz a beérkező csomag először az útválasztó memóriájába kerül, és ott várakozik egészen addig, amíg az útválasztó fel nem szabadul, és el nem kezdheti feldolgozni azt. Ha azonban túl sok csomag érkezik, előfordulhat, hogy az útválasztó memóriája betelik, és az ezután érkező csomagok mind elvesznek. (Érdekes, hogy a probléma nem oldódna meg azzal, ha több memóriát pakolnánk az útválasztóba, például végtelen mennyiségűt. Sőt, a helyzet ezzel csak rosszabbodna! Mire a router elérné a sor végi csomagokat, addig a forrás időzítője már rég, minden bizonnyal többször is lejárt, így az útválasztó sora már másodpéldányok sokaságát is tartalmazza. Az útválasztó persze ezeket is továbbítani fogja, ezzel is növelve a terhelést a már amúgy is teljesen lelassult hálózaton).
A torlódás másik gyakori oka az, hogy nincs egyensúlyban a vonalak sávszélessége és az útválasztók számítási kapacitása. Ha az útválasztók lassú CPU-val rendelkeznek, akkor úgy is kialakulhat torlódás, ha közben rendelkezésre áll szabad vonalkapacitás. A dolog fordítva is igaz: hiába képes az útválasztó gyorsan végezni a feladatát, ha a vonalak túl lassúak.
Ha már egy torlódás kialakult, akkor nagyon könnyen szétterjedhet az alhálózat többi részére is. A torlódásnak van egy öngerjesztő hatása. Ha bizonyos csomagok elvesznek, mivel már nincs hely az útválasztó memóriájában, az adó újra és újra megpróbálja elküldeni ugyanazt a csomagot, ezzel másodpéldányokat zúdítva a már amúgy is leterhelt útválasztója.
A torlódások elleni védekezés tehát létkérdés a hálózatok életében. A következő részben részletesen megvizsgáljuk, milyen módszerek léteznek a torlódások elkerülésére.

Garzó András
garzo@interware.hu