Hálózatok (12. rész)
Hidak

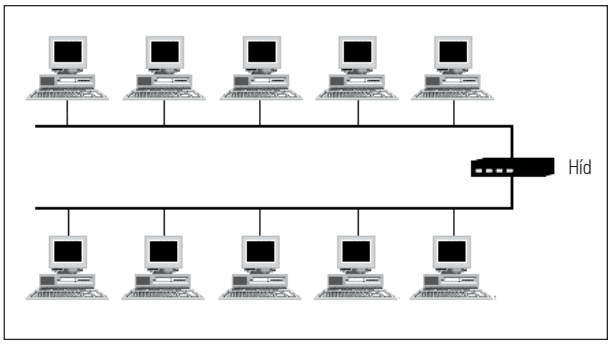
Mit tegyünk, ha több LAN-t szeretnénk egymással összekapcsolni? Ilyenkor hidakra van szükségünk, amelyekkel akár különböző szabványú hálózatokat is együttműködésre bírhatunk.

Minél több állomás van egy ***LAN***-on, az állomások annál nehezebben tudják megszerezni az osztott csatornát. Előbb-utóbb már annyira túlterhelt lesz az átviteli közeg, hogy a hálózat használhatatlanná válik az egyre gyakrabban kialakuló versenyhelyzetek, illetve ütközések miatt. A sávszélesség növelése persze jelenthet megoldást, de ez gyakran az összes hálózati eszköz lecserélésével jár, amely sosem olcsó mulatság.

Van azonban más lehetőség is. Az előző részben bemutattuk a ***kapcsolót (switch)***, amely külsőségeiben némileg hasonlít a hubokhoz, de a belső ennél sokkal többet rejt. A végpontokra ***(portok)*** itt is állomásokat kapcsolhatunk, de a tőlük érkező kereteket a kapcsoló – a hubokkal ellentétben – csak a címzett(ek)nek továbbítja. A kapcsoló tehát egy „intelligens" eszköz, képes belelátni a rajta átmenő adatokba. Mivel a keretek nem jutnak el oda, ahol senki sem kíváncsi rájuk (kivéve persze a hálózaton hallgatózókat, de ők most nem számítanak), csökken a csatorna terheltsége.

Egy kapcsoló végpontjaira ugyanakkor nem csak egyetlen munkaállomást köthetünk, hanem akár egy egész „***LAN***-nyit" is. Ettől kezdve a kapcsoló többé már nem kapcsoló, hanem ***híd (bridge)***. Itt aztán egészen új problémák merülnek fel, de még mielőtt ezekkel szembesülnénk, válaszoljunk meg egy kínzó kérdést: pontosan miért is van szükség hidakra? A kérdést egyből át is fogalmazzuk: miért lehet arra szükség, hogy egy szervezeten belül az állomásokat több különálló ***LAN***-ba foglaljuk, majd azokat hidakkal összekapcsoljuk, ahelyett, hogy összerakjunk egy jó nagy hálózatot, amelyhez minden állomás közvetlenül kapcsolódna?

Az első ok már sejthető: a terhelés megosztása. Ahogy azt az imént már kifejtettem ha sok az állomás, akkor a csatorna terhelt lesz. Ha nem szeretnénk terhelt csatornát, akkor meg kell növelnünk a sávszélességet. Nagy számú állomás esetében azonban óriási sávszélességre lenne szükségünk, amelyre nem biztos, hogy szert tudnánk tenni. Ilyenkor a hálózatot több részre ***(szegmensre)*** kell osztani, és valamiképp biztosítani azt, hogy a forgalom nagy része ne kerüljön ki a gerinchálózatra (a szegmenseket összekötő közegre), hanem maradjon a különálló részeken belül.

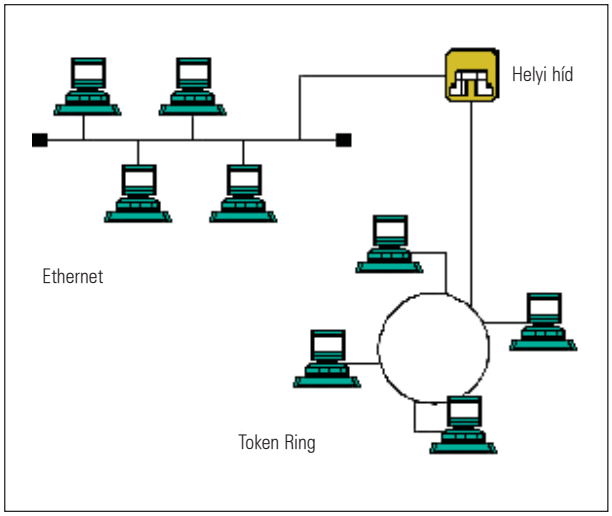


1. ábra

Miként lehet ezt megoldani? Tegyük fel, hogy van kismillió munkaállomásunk. A felhasználók az állományaikat a hálózat fájlszerverén tárolják. Mivel mindenki gyakran végez fájlműveleteket, ezért biztos, hogy a LAN a terhelés miatt használhatatlanná válik. Kivéve akkor, ha a munkaállomásokat szegmensekbe szervezzük, és minden szegmenshez tartozik egy saját fájlkiszolgáló. Mivel a szegmensekben viszonylag kevés állomás van, a csatorna ezeken belül már nem lesz különösebben terhelt.

Hogy miként is képzelhetjük el a szegmenseket, azt az *1. ábra* illusztrálja. A szegmensek különálló ***LAN***-ok, amelyek valamennyien saját csatornával rendelkeznek. Magukat a szegmenseket hidakkal köthetjük össze. Ha kevés szegmensünk van, akkor elegendő egyetlen kapcsoló, egy kiterjedt hálózat esetében azonban elképzelhető, hogy több kapcsolót kell egymással összekötnünk.

Hidakra akkor is szükségünk lehet, ha a hálózat amúgy bőven elbírná a terhelést. Előfordulhat például, hogy túl messze vannak egymástól az állomások. Bizonyos típusú ***LAN***-ok esetében fizikailag korlátozva van a hálózat megengedett mérete. Az ***Ethernet*** esetében ez például 2500 méter. Vagy nézzük azt az esetet, amikor két különálló épületben elhelyezkedő állomásokat szeretnénk egy hálózatba kötni. Sokkal megbízhatóbb, na és persze olcsóbb



2. ábra

megoldás az, amikor a két épületben két különálló ***LAN***-t készítünk, és azokat egymással hidak és infravörös átvitel segítségével kapcsoljuk össze, mint ha átvezetünk egy koaxiális kábelt.

A hidak másik fontos feladata a különböző szabványú hálózatok közötti átjárhatóság megteremtése *(2. ábra)*. Ilyen igény általában akkor szokott felmerülni, amikor már van két működő, ám teljesen eltérő típusú hálózat, és utólag jutott csak az üzemeltetők eszébe, hogy milyen jó is lenne, ha a két hálózat egymással is tudna kommunikálni. Van azonban olyan eset is, amikor direkt különböző szabványú hálózatokat szeretnének telepíteni egy szervezeten belül, mert bizonyos feladatokra az egyik alkalmasabb, mint a másik. Bármi is legyen a kiinduló ok, ezt a feladatot a hidaknak meg kell oldaniuk. Sejthető, hogy ez nem könnyű, hiszen például a vezérjeles gyűrű ***(token ring)*** más keretmérettel dolgozik, mint az ***Ethernet***, de sok más egyéb alapvető különbség is van.

Még egy dolog van, ami miatt áldott a hidak tevékenysége, ez pedig a megbízhatóság és a biztonság szempontjából érdekes. Az utóbbiról már volt szó az előző részben. Mivel az adatszórós hálózaton mindenki hallhat mindenkit, ezért könnyen lehallgathatóak olyan dolgok, amelyeknek sosem kellett volna más fülébe jutniuk. A hidak megnehezítik ezt a tevékenységet, ám teljes biztonságot nem nyújtanak. A hidak azonban okosak, így meg tudjuk mondani nekik, hogy milyen helyzetben mit továbbítsanak, illetve mit ne továbbítsanak. Ezzel megnövelik a hálózat stabilitását is, hiszen védelmet nyújtanak a meghibásodott állomás által folyamatosan a csatornára bocsátott szeméttől, illetve a hálózat megbénítását célul maguk elé tűző rosszakaróktól.

Hidak IEEE 802 szabványú hálózatok között

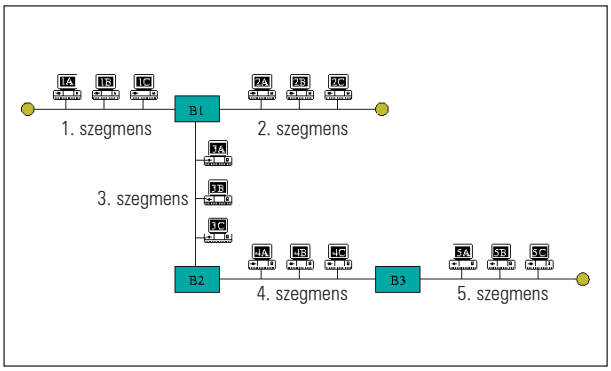
Még az ***Ethernet*** bemutatása előtt megemlítettük, hogy többféle ***LAN*** megvalósítás létezik. Ezek a megvalósítások mind szabványosítottak, azaz működésük a legapróbb részletekig dokumentálva van, és minden ilyen hálózat köteles betartani a szabványban leírtakat. A világ legnagyobb műszaki szakmai szervezete, az ***IEEE (Institute of Electrical and Engineers)*** többféle szabványt készített helyi hálózatokhoz, amelyek ***IEEE 802***-es szabványokként híresültek el. Az előző részben a ***802.3***-on ***(Ethernet)*** kívül bemutattunk más szabványokat is. Ezek a szabványok csupán a fizikai és az adatkapcsolati réteg megvalósításában különböznek, onnan felfelé kompatibilisek egymással.

A gond azonban pont az, hogy a különböző szabványok adatkapcsolati rétegei nem kompatibilisek egymással. A hidaknak tehát valahogy át kell tudniuk alakítani a rajtuk átmenő forgalmat úgy, hogy az a célhálózatban tovább tudjon haladni. Ha úgy tetszik, a híd egy olyan eszköz, amely biztosítja a keretek számára egy másik típusú hálózatba történő átjárást. Ez azonban egy nagyon veszélyes út.

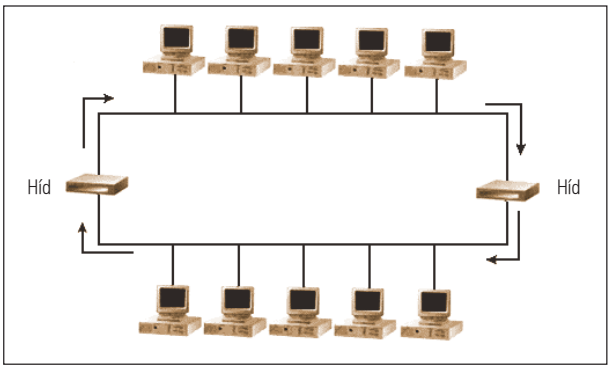
Rögtön az első probléma, hogy két összekapcsolt hálózat nem biztos, hogy ugyanolyan sebességgel üzemel. Amikor egy gyorsabb hálózat küldi keretek sorozatát a lassabb felé, akkor a híd nem tudja azt ugyanolyan gyorsan továbbítani. Amíg egy keret vár a továbbításra, addig a híd belső memóriájában kap helyet. Minden memória véges, ha a gyorsabb hálózatról folyamatosan érkeznek az adatok, akkor előbb-utóbb betelik a puffer, és a hídnak meg kell szabadulnia pár kerettől. Ugyanez lehet a helyzet, ha egy leterhelt ***Ethernet*** hálózat felé továbbítunk. A vezérjeles hálózatoknál a terhelés nem jelenthet puffer-túlcsordulást, hiszen biztosítva van, hogy a híd szabályos időközönként mindig adni tudjon.

Másik probléma, hogy a híd a hálózatban torlódási pont lehet. Ez nem csoda, hiszen minden szabvány más keretformátummal dolgozik, tehát a beérkező kereteket át kell alakítani és újból ki kell számolni az ellenőrző-összeget, ez pedig időbe telik. Miért baj ez? Tegyük fel, hogy a hálózati réteg egy hatalmas üzenetet ad át az adatkapcsolati rétegnek. Mivel a kereteknek van egy maximális méretük, ezért ezt az üzenetet több keretre kell szabdalni, majd azokat sorban továbbítani. Ezeket a kereteket a hídnak egyenként kell átalakítania. Kellően sok keret esetén az átalakításra fordított idő olyan nagy lehet, hogy a forrásállomás nyugtára váró időzítője lejár, és ismét elkezdi sorban küldeni a kereteket. Ezeket a hídnak megint csak át kell alakítania. Előfordulhat, hogy a forrás beleun az állandó ismételgetésbe, és hibaüzenetet dob, miszerint a célállomás nem válaszol. A probléma pedig nem is a címzettel van.

A legsúlyosabb bajok forrása azonban a különböző keretméretek. Nyílván addig nincs gond, amíg egy olyan hálózatba továbbítunk kereteket, ahol a maximális méret nagyobb, mint a küldő hálózatban. Fordított esetben azonban az ésszerű teendő az lenne, ha a túl nagy keretet feldarabolnánk. A probléma azonban az, hogy az adatkapcsolati protokollok nem támogatják a keretek összevissza szabdalását. Amikor egy állomás elküld egy keretet, akkor két esettel számol: vagy épségben megérkezik a célállomáshoz, vagy valahol elveszik útközben. De arra álmában nem gondol, hogy egy híd feldarabolja. Hiszen akkor ő egyet küldött, de a cél kettőt kap. Persze nem lenne túl nehéz ezeket a protokollokat alkalmassá tenni a feldarabolt keretek kezelésére, csak épp a ***802***-es szabványban ez nem szerepel. Így nem létezik megoldás. Illetve létezik: ha valaki egy olyan keretet küld, amelyet



3. ábra



4. ábra

a célhálózat nem tud elfogadni, akkor a híd egyszerűen eldobja. Ez azonban nem túl megnyugtató megoldása a problémának.

Ha ***802.4 (vezérjeles sín)*** hálózatról adunk ***802.3*** felé, akkor két további probléma is felmerül. Az első természetesen a prioritás. Az ***Ethernet*** hálózatokban az állomások nem rendelkeznek prioritással, míg a ***802.4*** esetében van ilyen szolgáltatás. Persze ez csak akkor jelenthet problémát, ha a két, egymással kommunikáló ***802.4***-es hálózat között van egy ***802.3***-as is. Ebben az esetben sajnos a keretek prioritását nem lehet megőrizni.

Másik megoldhatatlan problémáért az ideiglenes vezérjelátadás nevű szolgáltatás okolható. Ez arra való, hogy ha egy ***802.4***-es hálózatban küldünk egy keretet, akkor a célállomásnak lehetősége nyílik arra, hogy visszakapja a vezérjelet ahhoz, hogy a nyugtát visszaküldhesse. Amikor egy ilyen keret érkezik a hídhoz, akkor annak komoly lelkiismereti kérdéssel kell megbirkóznia: vagy hazudik, és visszaküld egy hamis nyugtát a feladónak, vagy becsületes módon nem szól semmit, csak továbbítja a keretet. Mind a kettő nagyon veszélyes, nehéz eldönteni, hogy melyik a jobb megoldás. Átverni a feladót nem bölcs dolog, főleg úgy, hogy elképzelhető, a célállomás nem is működik. A keretet nem nyugtázni is merész húzás, hiszen a feladó megelégelheti a dolgot egy „célállomás halott" hibaüzenettel. Tökéletes megoldás erre sem létezik.

Amikor ***802.3***-ról ***802.4***-re szeretnénk egy keretet átjátszani, akkor ott csak a már említett prioritás okozhat gondot. Pontosabban az, hogy milyen prioritást állítsunk be az átküldött kerethez. Talán a legjobb taktika az, ha mindig a legmagasabb fokú prioritást rendeljük az átmenő keretekhez, mivel feltehetően az már úgyis késleltetésben van.

Ha két ***802.4***-es hálózatot szeretnénk egy híddal összekötni, akkor szintén szembetaláljuk magunkat az ideiglenes vezérjelátadás problémájával. Ilyenkor a hídnak nagy szerencsejátékosnak kell lennie. Ha ugyanis úgy hozza a sors, hogy a keretet egyből tudja továbbítani, akkor van rá esély, hogy a nyugta időben megérkezhessen. Ha mégsem alakulnak úgy a dolgok, akkor csalni kell: a keret prioritását meg kell növelni. Ezzel lerövidíthetjük a nyugta célbaérkezésének idejét.

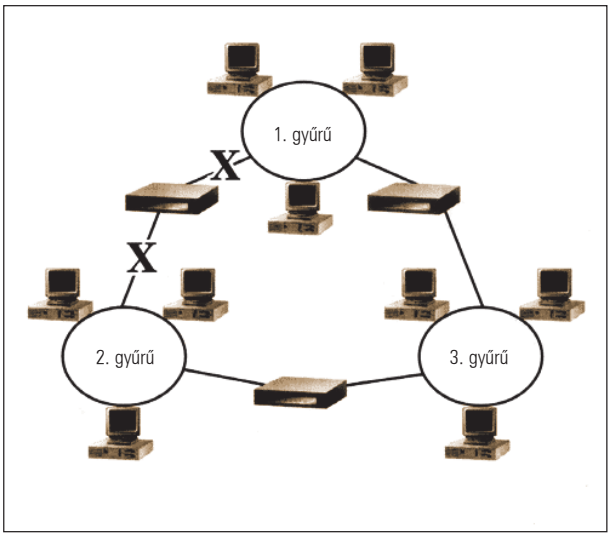
Nem kérdés, hogy elég sok problémával kell szembenéznünk akkor, amikor két hálózatot egy híddal szeretnénk összekötni. De mi a helyzet akkor, ha nem csak két, hanem több ***LAN***-t és több hidat tartalmazó összekapcsolt hálózatunk van? Nos, erre is létezik ***IEEE*** szabvány, de hasonlóan a hálózatokhoz, itt is több (egész pontosan kettő) van, amelyek természetesen nem kompatíbiliesek egymással.

Transzparens/feszítőfás hidak

Ezek a hidak abban a szellemben készültek, hogy bármiféle hardveres, illetve szoftveres beállítás nélkül telepíthetőek legyenek. Ezt szó szerint úgy képzelhetjük el, hogy amint bekötjük a hidat a hálózatba és bekapcsoljuk, a rendszerünk máris működik. Nem kell sem a hidat, sem az állomásokat felkonfigurálnunk. Azért nevezzük ezeket transzparens ***(átlátszó)*** hidaknak, mert az állomások számára láthatatlanok, ők ugyanúgy látják egy másik hálózat állomásait, mintha közös ***LAN***-ban lennének. Amikor egy keret megérkezik, nem tudhatjuk, hogy útja során hány hídon ment keresztül, azaz hány híd van köztünk, és a velünk kapcsolatban lévő állomás között.

Hogy megérthessük a transzparens hidak működését, tekintsünk a *3. ábrára*. Itt négy ***LAN***-t látunk három híddal összekötve. Tegyük fel, hogy az ***1A*** gép keretet küld az 1B-nek. Mivel ezek egy szegmensen vannak, így a ***B1*** híd a keretet biztosan eldobja. Most nézzük azt az esetet, ha a keret címzettje ***3B***, amely egy másik szegmensen van. A hídnak most továbbítania kell a keretet, csak az a kérdés, hogy hova. Ilyenkor a memóriájában lévő táblázatból kiolvassa, hogy a keret címe alapján melyik végpontjára kell átjátszani. Látja, hogy a ***3B*** gép a hármas szegmensen van, így továbbítja a megfelelő portra. Fontos, hogy a táblázat nem azt mondja meg, hogy melyik szegmensen van a célállomás, hanem azt, hogy merre kell irányítani. Ha a címzett az ***5C*** lenne, akkor is ezen a porton menne tovább a keret, csak át kell még mennie két másik hídon. De ezzel a ***B1*** hídnak nem kell foglalkoznia.

De azt mondtuk, hogy a transzparens hidak semmiféle beállítást nem igényelnek. Mégis honnan tudják, hogy merre kell továbbítani az egyes kereteket? Valóban, kezdetben az összes híd táblázata üres. Ha tehát jön egy olyan keret, amelyiknek a címzettjét nem ismerik, nem tehetnek mást, mint hogy az összes kimeneten továbbítják (kivéve azt a portot, amelyikről a keret érkezett). Ezt úgy is mondjuk, hogy a hidak elárasztják a hálózatot. Ezután megjegyzik,



5. ábra

hogy ennek a keretnek ki volt a feladója, és melyik ***LAN*** felől érkezett. Ha a későbbiekben egy olyan keret érkezik, amelynek a címzettje a kérdéses állomás, akkor már tudni fogják, hogy azt melyik kimenet felé kell irányítani. Ahogy egyre több keret érkezik, úgy fogják ismerni egyre részletesebben a hálózat felépítését. Ezt az algoritmust hátrafelé tanulásnak nevezzük.

Egy hálózat felépítése azonban ritkán állandó, a lelkes rendszergazdák egyfolytában átalakítják. Például új hidakat telepítenek, régieket távolítanak el, és az állomásokat az egyik ***LAN***-ból a másikba költöztetik. A hidaknak ezért folyton naprakész (sőt percre pontos) információval kell rendelkezniük a hálózat aktuális állapotáról. Ezért fontos tudniuk, hogy egy adott cím melyik időpontban került a táblába. Ha érkezik egy keret, amelynek forrása már szerepel a táblában, akkor annak időpontját a híd az aktuális időpontra írja felül. Ezenkívül minden, néhány percnél öregebb bejegyzést a hidak törölnek. Ezzel elérhető, hogy ha egy állomást elköltöztetünk a hálózat egy teljesen más pontjába, akkor is perceken belül visszaáll a normális működés.

Tekintetünket szegezzük most a *4. ábrára*, ahol csupán csak két ***LAN*** van, viszont a rendszergazdák biztosra akartak menni, és két híddal kötötték össze a szegmenseket, hogy meghibásodás esetén se legyen semmi probléma. Tegyük fel, hogy a felső ***LAN*** egyik állomása egy olyan keretet bocsát útnak, amelynek a címzettje mindkét híd számára ismeretlen. Nincs mit tenni, mindketten elárasztanak, azaz a kérdéses keretet mindketten az alsó ***LAN***-ra másolják. Ezután nem sokkal a bal oldalt lévő híd felfedezi a jobb oldali híd által átjátszott keretet, amit ő visszamásol a felső ***LAN***-ra, amit majd ismét a jobb oldali fog ismét az alsó ***LAN***-ra másolni. Ennek a folyamatnak sohasem lesz vége.

Mitől alakulhatott ki ez a roppant kellemetlen helyzet? Hát azért, mert hurok volt a hálózatunkban. Ezért a hidaknak valahogy egyeztetniük kell egymással, és logikailag úgy átalakítani a hálózatunk topológiáját, hogy egy LAN-hoz csak „egy út vezessen”.

Képzeljük el úgy a hálózatunkat, mint egy olyan gráf, amelynek a csúcsai a ***LAN***-ok. Erre láthatunk példát az ***5. ábrán***, ahol a változatosság kedvéért vezérlőjeles gyűrűs hálózatokat kötöttünk össze egymással. Két csúcs akkor legyen egymással összekötve, ha össze vannak kapcsolva híddal. Jelen esetben minden csúcs össze van kötve mindenkivel. A feladat az, hogy egy körmentes, ám továbbra is összefüggő gráfot kapjunk, amely tartalmazza az eredeti gráf összes csúcsát. Ez az eredeti gráf úgynevezett ***feszítőfája (spanning tree)***. A feszítőfára az jellemző, hogy minden csúcsból minden csúcsba pontosan egy út vezet. Egy ilyen hálózati topológiában nem fordulhatna elő az előbb említett eset.

A hidaknak tehát egymás között meg kell állapodniuk a feszítőfában. Ehhez maguk közül ki kell választani valakit, aki a fa gyökerét fogja képezni. Erre egy elég frappáns megoldást alkalmaznak: minden híd megmondja a saját egyedi sorozatszámát. Akié a legkisebb, az lesz a gyökér. Ezután meg kell határozni azt a fát (a feszítőfát), amely minden ***LAN***-hoz a lehető legrövidebb utat tartalmazza. Ezt a fát természetesen újra kell számolni, ha a hálózati topológia megváltozik, például új híd kerül telepítésre.

A feszítőfát kiszámoló osztott algoritmus egyébként sohasem áll le, így a feszítőfa azonnal érzékelhető.

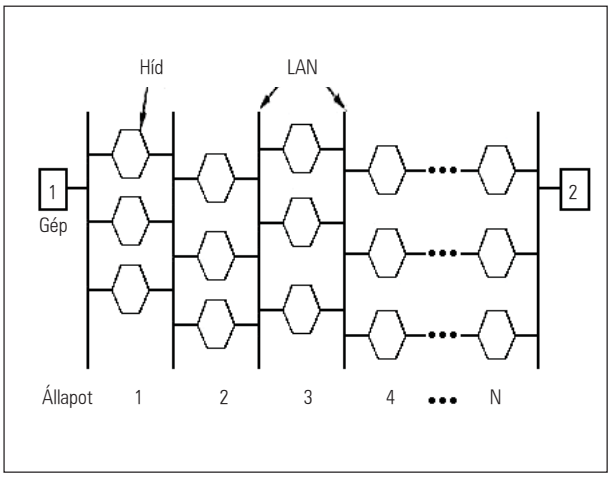
Az így kialakított új hálózati topológiánkban az összes ***LAN*** biztosan benne marad, de nem biztos, hogy az összes híd is! Amikor a fizikai topológiát ábrázoló gráfból egy élet elhagyunk, az azt jelenti, hogy az adott élhez tartozó hídnak blokkolnia kell azt a két kimenetét, amely az élhez tartozó két csúcsot (LAN-t) összeköti. Ha egy híd minden olyan használatban lévő portját blokkolta, akkor ő nem vesz tovább részt az adatforgalom továbbításában. Ez a helyzet az *5. ábrán* is. Az ott látható gráfból egyféleképpen készíthetünk feszítőfát: egy tetszőleges élt törlünk, azaz megszüntetjük a kapcsolatot két tetszőleges ***LAN*** között. Jelen esetben az 1. és a 2. között, de bármelyiket is választjuk, egy híd mindenképp munka nélkül marad.

Látható tehát, hogy a transzparens hidak képtelenek kihasználni a teljes rendelkezésre álló sávszélességet. Nem úgy mint a következő hídcsoport.

Forrás által irányított hidak

Ezek aztán tényleg kihasználják a sávszélességet, habár van egy buktatója a dolognak, de erről majd kicsit később. A működési elv azzal a nehezen teljesíthető feltételezéssel él, hogy minden állomás, aki üzenetet szeretne küldeni egy másik állomásnak, pontosan tudja, hogy a címzettel egy szegmensen van-e, vagy sem. Ha nincs, akkor viszont tudja, hogy miként lehet (mely hidakon keresztül) eljutni abba a hálózatba.

A ***forrás általi forgalomirányítás (source routing)*** könnyebb megértéséhez ismét segítségünkre lesz a 3. ábra. Ha az ***1A*** szeretne mondani valamit az ***1B***-nek, akkor a keret címének utolsó helyiértékű bitjét 0-ra állítja, mivel egy szegmensen vannak. Ekkor a ***B1*** híd nem foglalkozik a kerettel. Ha azonban az ***5C*** gépnek szeretne küldeni, akkor ennek a bitnek az értéke 1 lesz, és a keret fejlécébe beszúrja a pontos útvonalat is. Ahhoz, hogy ezt az útvonalat le lehessen írni, minden ***LAN***-nak rendelkeznie kell egy egye-



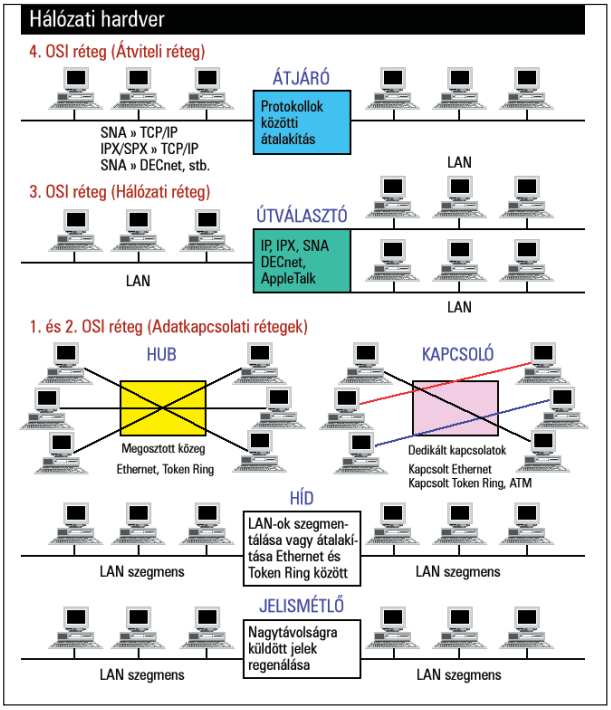
6. ábra

di címmel. Ugyanez a helyzet az összes ugyanahhoz a szegmenshez kapcsolódó híddal is. Tehát a ***B1*** és ***B2*** hidak címének különbözőnek kell lennie, de a ***B1*** és ***B3*** címe megegyezhet. Az útvonal tehát egy olyan sorozat lesz, amelyben felváltva szerepel egy híd és egy ***LAN*** címe. Az ***1A – 5C*** útvonal tehát a következőképp fog kinézni: ***B1***, ***Seg3***, ***B2***, ***Seg4***, ***B3***, ***Seg5***. Amikor egy híd megkap egy ilyen keretet, akkor belenéz az előírt útvonalba, és megkeresi benne annak a ***LAN***-nak a címét, ahonnan a kérdéses keret érkezett. Ezután megnézi, melyik híd címe áll mögötte. Ha ez a cím a saját címe, akkor továbbküldi az előírt szegmensre.

A kérdés már csak az, hogy honnan tudhatják az állomások azt, hogy a cél felé pontosan milyen út is vezet. Sőt, nem elég bármelyik útvonalat ismerni, ahhoz hogy a dolog valóban hatékony legyen, a legeslegrövidebb utat kell megtalálniuk.

Amikor egy állomás nem ismeri a címzettjének pontos helyét, akkor egy úgynevezett ***felkutató keretet (discovery frame)*** bocsát útnak. Ezeket adatszórással továbbítja, és minden híd minden irányba továbbítja, tehát biztosan megérkezik az összes ***LAN***-ra. Amikor a címzett rádöbben, hogy valaki holléte felől érdeklődik, egy válaszkeretet küld vissza. Ahogy ez a keret visszafelé indul, minden olyan híd, amelyen áthalad, beleírja azonosítószámát. Így amikor visszaér, az állomásnak lehetősége nyílik megvizsgálni a visszafelé vezető útvonalat, amelyből könnyedén kiszámítható a célhoz vezető legoptimálisabb útvonal.

A már említett buktató akkor jelentkezik látványosan, ha a hálózatunk topológiája hasonló a *6. ábrán* bemutatotthoz. Itt a ***LAN***-okat sorba egymás után kötöttük, minden egyes ***LAN***-t három híd kapcsol össze egymással. Az 1-es állomás a 2-es pozíciójáról szeretne többet megtudni, ezért kutató keretet indít útnak. Mivel ezeket a kereteket az összes híd továbbítja, ezért a szomszédos LAN-on már három kutató keret lesz jelen, a 3-on kilenc, a 4-en pedig 27. Ez bizony exponenciális ütemben zajló növekedés, tehát 10 ***LAN*** esetében majdnem 20 ezer, 20 ***LAN***-nál pedig több mint 1 milliárd keretet jelent. Talán nem kell ecsetelni, hogy ez mekkora terhelést is jelent. Ezért is hívják ezt a jelenséget keretrobbanásnak.



7. ábra

Felmerülhet a kérdés, hogy a transzparens hidakat miért nem veszélyezteti a keretrobbanás, miközben ők is ugyanezt csinálják azokkal a keretekkel, amelyeknek a címzettje ismeretlen. Valójában ott is ugyanez a folyamat játszódik le, csak nem árasztják el velük az egész hálózatot, hanem kizárólag a feszítőfa mentén küldik tovább. Ez pedig csak lineáris ütemben történő növekedés, amely nem jár ilyen drasztikus végeredménnyel.

Hálózati eszközök – összefoglalás

Sorozatunk e részének végén érdemes egy gyors összefoglalást végezni azzal kapcsolatban, hogy milyen hálózati eszközöket is ismerünk idáig, és ezeknek pontosan mi a feladatuk. Ehhez hasznos segítséget nyújt a *7. ábra*. Az ***átjáró*** (***gateway***) és az ***útválasztó*** (***router***) még ezidáig nem szerepelt, mivel ezek a felsőbb, eddig még nem tárgyalt rétegekben tevékenykednek. Fontos, hogy hiába végez a híd forgalomirányítást, illetve üzemel „átjáróként” különböző szabványú hálózatok között, nincs köze az útválasztóhoz és az átjáróhoz.

A hub a fizikai rétegben tevékenykedik és nagyon buta eszköz: ami az egyik portján bemegy, az kijön a többin.

A kapcsoló (switch) ennél sokkal okosabb, hiszen képes az adatkapcsolati réteg adategységeit, a kereteket olvasni, így csak arra a kimenetre küldi az adatot, ahol a címzett található. Nem így a jelismétlő (repeater), amely a hubbal van „egy szinten”: ő is csak ismételni tudja a bejövő jeleket, igaz, felerősíti azokat.

A következő részben a nagy sebességű hálózatokról és műholdakról lesz szó.

Garzó András
garzo@interware.hu