

Számítógép hálózatok (14. rész)

Virtuális áramkörök és datagram alapú alhálózatok működése, forgalomirányítás a hálózati rétegben.

Bevezetés

Az előző részben már elkezdünk foglalkozni a hálózati réteggel, amelynek segítségével a hosztok képtelenek lennének megtalálni egymást a hálózaton. Arról is volt szó, hogy alapvetően kétféle elképzelés létezik arról, milyen szolgáltatásokkal is rendelkezzen a hálózati rétegnek. Az első elgondolás szerint a hálózati rétegnek összeköttetés alapú szolgáltatást kell megvalósítani, amely nagyon hasonló egy telefonon történő kommunikációhoz. Valamelyik gép felhívja a másikat, kiépül a kapcsolat, majd megtörténik az adatcsere. Amikor mindkét gép befejezte a társalgást, a kapcsolatot lebontják. Ilyen elv szerint működnek például az ATM hálózatok. Az Internet és protokolljai azonban egy másik szemléletet követnek: itt a szolgáltatás mindig összeköttetés nélküli.

Az ilyen típusú kommunikációt úgy képzelhetjük el, mintha levelezve, postai úton kommunikálnánk valakivel. Az elküldendő adatot „becsomagoljuk” egy úgynevezett datagramba, ráírjuk a címzettet és a feladót, majd útjára engedjük.

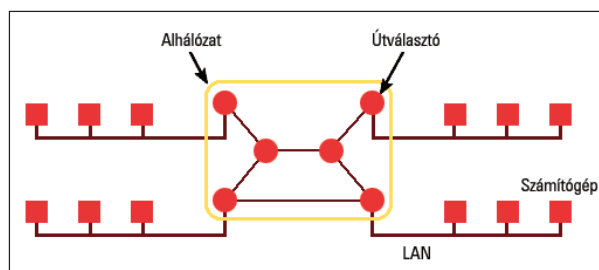
Amikor a hálózati réteg megvalósításáról beszélünk, akkor tulajdonképpen a kommunikációs alhálózat (vagy röviden alhálózat) működését vesszük szemügyre. Emlékezzünk vissza: az alhálózat az a hálózat, amely a gépeket, illetve helyi hálózatokat egymással összeköti. Ez a hálózat többnyire *útvonalválasztókból (routerekből)* áll (1. ábra).

Összeköttetés alapú hálózat

Az ilyen jellegű kommunikáció pontosan úgy működik, mint amikor telefonon beszélgetünk valakivel. Sőt, tulajdonképpen maga a telefonhálózat is összeköttetés alapú hálózat. Amikor két gép adatot kíván cserélni egymással, először létre kell hozniuk egy kapcsolatot. Amíg ez a kapcsolat él, addig tudnak egymásnak csomagokat küldözgetni. Az ilyen hálózatok megbízhatóak, tehát a csomagok nem vesznek el útközben, nem sérülnek meg és sorrendhelyesen érkeznek meg.

A gépek között kiépülő kapcsolatot *virtuális áramkörnek (virtual circuit)* nevezzük. Mivel általában több olyan alhálózati útvonal is létezik, amelyen keresztül eljuthatunk az egyik géptől a másikig, ezért a virtuális áramkörnek nem csak a két gépet kell meghatározni, hanem egy hozzájuk tartozó útvonalat is. Ez az útvonal az összeköttetés felépítésekor kerül kiválasztásra, tehát minden elküldött csomag ugyanazon az útvonalon fog haladni.

Ehhez azonban a routereknek tudniuk kell, hogy a rajtuk átmenő csomagokat melyik kimenetükön keresztül küldjék



1. ábra

tovább. Így minden útválasztó rendelkezik egy táblázattal, amely hozzárendeli a rajta „keresztülmenő” virtuális áramköröket a megfelelő kimenettel. Amikor beérkezik egy csomag, akkor az útválasztó a fejlécből meg tudja állapítani, hogy az melyik virtuális áramkörön halad, így táblázata segítségével a megfelelő kimenetre képes irányítani.

A helyzetet az bonyolítja kicsit, hogy a virtuális áramköröknek nem lehet egy globálisan érvényes azonosítószám adni, mivel minden gép maga választja ki azokat. Így a virtuális áramkorszámok csak helyi szinten érvényesek. Ha ez nem így lenne, akkor könnyen előfordulhatna az az eset, amikor két virtuális áramkör ugyanazt az azonosítót kapja, és ráadásul ugyanazon az útválasztón is megy keresztül. Ebben a helyzetben a router nem tudná eldönteni, hogy a beérkező csomagot melyik kimenetre is irányítsa.

Ha a vonal duplex, akkor időnként bekövetkezhet egy elég kellemetlen esemény, ami akkor fordul elő, ha két gép egy időben kíván kapcsolatot kiépíteni a másikkal. Ilyenkor a kapcsolat felépítésére irányuló kérelem mindkét irányban egy időben kezd el haladni az alhálózaton. Tegyük fel, hogy a routerek programja olyan, hogy új virtuális áramkör felépítések mindig a legkisebb szabad sorszámot rendelik az új kapcsolathoz. Amikor a két kérelem két szomszédos routerhez érkezik, akkor mind a ketten ugyanazt az azonosítót fogják kiosztani, tehát két különböző virtuális áramkörnek ugyanaz lesz a száma ugyanazon a fizikai vonalon. Ez azért baj, mert a routerek nem fogják tudni a beérkező csomagokról, hogy ez az egyik összeköttetésben egy előrehaladó, vagy a másikon egy visszafelé menő-e.

Amikor a két gép közötti kommunikáció befejeződik, akkor a virtuális áramkör lebontásra kerül.

Összeköttetés nélküli hálózatok

Míg az összeköttetés alapú hálózatoknál a hálózati réteg felül az összeköttetés megvalósításáért, addig az összeköttetés nélküli hálózatoknál ez egy felsőbb réteg (a szállítási réteg) feladata. Az Internet és annak protokolljai esetében a hálózati rétegre csak a neki átadott csomag célbajuttatása van bízva. Ehhez nem kell előzetesen kiépített kapcsolat, nem kell törődni a forgalomszabályozással és az sem biztos, hogy az átvitel hibátlan lesz. (Míg ezeket a feladatokat az összeköttetésen alapuló hálózatoknál az alhálózat végezte, addig itt ez a gépek feladata).

Mivel nincs összeköttetés, ezért az útvonal nincs előre kijelölve, hanem minden csomag esetén külön számítódik ki. Ezért a csomagoknak tartalmazniuk kell a forrás- és a célállomás címét. A csomagok egymástól függetlenül kerülnek továbbításra, ezért a fejlécben fel kell tüntetni a csomag sorszámát is, hiszen könnyen megeshet, hogy a célhoz nem az útrabocsátás sorrendjében érkeznek be a csomagok. (Mivel valószínűleg mindegyik más útvonalon keresztül ért célt). Számolni kell azzal a lehetőséggel is, hogy egyes csomagok soha sem érkeznek meg, vagy ami rosszabb, a vevő kétszer is megkapja ugyanazt a csomagot. Az összeköttetés mentes hálózatok tehát korántsem nevezhetők megbízhatónak. Az egymástól függetlenül továbbított adatdarabokat datagramoknak nevezük. Még egyszer hangsúlyozzuk, hogy a datagramoknak soha nincs előre kijelölt útvonaluk. Egy ilyen hálózat kezelése mindenképpen bonyolultabb és munkaigényesebb feladat, viszont ezért cserébe sokkal jobban alkalmazkodik a véletlenszerűen jelentkező forgalomváltozásokra.

Az útválasztóknak az összeköttetés nélküli hálózatokban is szükségük van belső táblázatokra, csak ott most nem a virtuális áramköröket tartják nyilván (mivel ilyesmiről itt nem beszélhetünk), hanem a velük kapcsolatban lévő routerek címeit. A router a beérkező csomagot a célcíme alapján irányítja a megfelelő kimenetre. (Annak eldöntése, hogy melyik kimenetre kerüljön a csomag, egyáltalán nem egyszerű. Erről később bővebben szólnunk majd a forgalomirányítási algoritmusok tárgyalásakor).

A két megoldás összehasonlítása

Nagyon nehéz kérdés, hogy melyik megoldás jobb a másiknál. Az biztos, hogy általánosan nem mondható el egyikről sem, hogy az jobb vagy rosszabb lenne. Sőt, azt sem egyszerű eldönteni, hogy egy konkrét helyzetben melyik alhálózati megvalósítás felelne meg leginkább az igényeknek. Most megpróbáljuk összeszedni mindkét hálózattípus előnyeit és hátrányait, illetve mikor jelenthet az egyik hatékonyabb megoldást a másiknál. De az itt leírtakat sem szabad örökérvényű törvénynek felfogni, mivel szélsőséges esetekben lehet, pont a másik típus lenne a legjobb választás.

Rögtön az első dolog, ami különbséget jelenthet a virtuális áramkörök és a datagramok között az a csomagok mérete. Az első esetben csak a virtuális áramkör azonosítószámát kell a csomagoknak szállítaniuk, amely lényegesen kisebb méretű mint a forrás- és a célcím. Ha kis csomagokkal dolgozunk, akkor ez a méretbeli eltérés jelentős lehet. Ugyanakkor az is igaz, hogy az összeköttetés alapú alhálózatoknál a routereknek nagyobb belső memóriára van szükség, ha csak nem akarjuk drasztikusan lecsökkenteni a virtuális áramkörök maximális számát.

A hálózat sebességét nézve is kompromisszumokra kell kényszerülnünk. Míg a virtuális áramkörök esetében az összeköttetés felépítése, illetve lebontása időt vesz igénybe, addig a datagram hálózatokban ez a késleltetés nem jelentkezik. Viszont a routernek sokkal tovább tart eldönteniük, hogy a csomagot melyik kimenetre továbbítsák.

Ha az összeköttetés várhatóan fennáll hetekig, hónapokig, esetleg évekig, akkor sokkal jobb választás a virtuális áramkör. Ha viszont csak időnként szeretnénk kis adatmennyiséget küldeni, tehát a kommunikáció úgymond tranzakció jellegű, akkor a datagram a hatékonyabb megoldás.

A torlódások elkerülését tekintve az összeköttetés alapú hálózatok némi előnnyel rendelkeznek. A szükséges erőforrásokat (*értsd: routereket*) már előre, a kapcsolat felépítésekor le lehet foglalni, így azok mindig rendelkezésre fognak állni, amikor csak szükség lesz rájuk. Az is igaz viszont, hogy a datagram alapú hálózatok sokkal dinamikusabban tudják kezelni a torlódásokat. Ha egy router nagyon leterhelt, akkor egy másikon keresztül mennek tovább a csomagok. Egy adott csomag útvonala tehát mindig az aktuális terhelési viszonyoktól függ. A virtuális áramkörök ugyanakkor nagyon könnyen megszakadhatnak, nem kell hozzá más, mint hogy egy útválasztó akár csak egy másodpercre is üzembékéltelenné váljon. Hiába indul újra egyből, a kapcsolat elveszik, és azt újra fel kell építeni. A datagram alapú hálózatoknál ez nem lehet probléma. Egy router kiesése csak azoknak a gépeknek jár kellemetlenséggel, akiknek a csomagjai éppen a kérdéses router memóriájában tartózkodtak.

Összeköttetés megvalósítása datagramokkal, és összeköttetésmentes szolgálat virtuális áramkörökkel

Fontosnak tartjuk, hogy hangsúlyt helyezünk a szolgálat típusa (összeköttetés alapú, illetve összeköttetés nélküli) és az alhálózat típusa (virtuális áramkörös vagy datagramos) között. A szolgálat típusa a felső réteg számára határozza meg, hogy pontosan mit is várhat el az adott rétegtől.

Az alhálózat típusa már a megvalósítás kérdéskörébe tartozik. Bizonyos esetekben szükség lehet egy datagramon alapuló összeköttetéses szolgálatra, vagy egy összeköttetés nélküli virtuális áramkörös megoldásra. Az utóbbira a legjobb példa az, ha valaki egy ATM hálózaton szeretne IP protokollt használni.

Forgalomirányító algoritmusok

A hálózati réteg megvalósításának legfontosabb kérdése az, hogy milyen útvonalon jutassa célba az alhálózatokon keresztül menő csomagokat. A LAN-ok világában egyedül a switch-ek végeztek forgalomirányítás-szerű tevékenységet, de azt is csak a hálózaton lévő forgalom csökkentésére. Az adatszórásos hálózatoknál nincs szükség forgalomirányításra. Ha azonban a forrás és a cél különböző hálózaton van, akkor találni kell egy útvonalat, amelyen a csomag eljuthat rendeltetési helyére. Az is fontos, hogy ez az útvonal a lehető legoptimálisabb legyen.

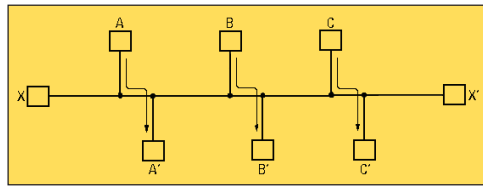
Az útvonalat a *forgalomirányító algoritmus* határozza meg, amely része a hálózati szoftvernek. Talán nem meglepő, hogy a forgalomirányító algoritmus *forgalomirányítást* végez, amely nem jelent mást, mint eldönteni, hogy egy beérkező csomag melyik kimeneten folytassa útját. A virtuális áramkörökön alapuló hálózatoknál ezt csak az új kapcsolatok felépíté-

lésénél kell elvégezni. A datagramos hálózatoknál minden beérkező csomagnál döntést kell hozni.

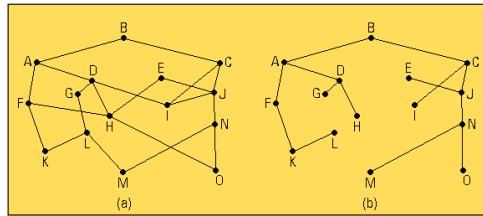
Sokféle forgalomirányítási módszer létezik, vannak rendkívül jól működők és vannak kevésbé hatékonyak. Mielőtt ezek működésébe részletesen is belemélyednénk, foglaljuk össze, milyen elvárásaink lehetnek egy ilyen algoritmussal szemben. Ezek pedig a következők: helyesség, egyszerűség, robusztusság, stabilitás, optimalitás és igazságosság.

Az nagyon jó dolog, ha az algoritmusunk helyesen működik, és nem hoz hibás döntéseket. Az sem árt, ha egyszerű, nem tartalmaz bonyolult számításokat, így kevesebb erőforrás szükségeltetik a megoldáshoz. Robusztusság alatt azt értjük, hogy az algoritmus ne legyen érzékeny az alhálózat topológiájában történő változásokra. Egy hálózat életében ugyanis naponta történhetnek váratlan fordulatok: áramszünet vagy valami hiba következtében egy útválasztó kiesik, majd hirtelen ismét megjavul, stb. Nehéz lenne az élet egy olyan hálózatban, amit mindig újra kéne indítani, ahányszor egy router meghibásodik. Az igazságosság és az optimalitás alapvető követelménynek tűnhetnek, ám e két cél egymással ellentétes. Nyilván nem mindig a legigazságosabb megoldás a legoptimálisabb. Elképzelhető, hogy globálisan hatékonyabb eredményt érünk el, ha időnként egy-egy gép számára nem a legkedvezőbb megoldást nyújtjuk. A 2. ábrán láthatunk egy példát, ami konfliktushoz vezet az igazságosság és az optimalitás között. Tegyük fel, hogy a A és A', B és B', C és C' között olyan nagy a forgalom, hogy telíti az őket összekötő, az ábrán vízszintes helyzetben lévő vonalakat. Az optimális megoldás az lenne, hogy minél több gép tudjon egymással zavartalanul kommunikálni, tehát el kéne zárunk az X és X' közötti vonalat. Az X hoszt előtt ülő felhasználó viszont ezt nem tartaná igazságos elbánásnak.

A forgalomirányító algoritmusokat két nagy csoportra oszthatjuk. Az elsőbe az úgynevezett *nem adaptív algoritmusok* tartoznak. Ezek az egyszerűbb eljárások, amelyek a döntések meghozatalánál nem vesznek figyelembe mérési, becslési eredményeket, sőt még az aktuális forgalmi helyzettel sem törődnek. Inkább minden számítást előre, „offline” módon végeznek el, amelynek eredményeit a routerek a hálózat indításakor kapnak meg. Történeten tehát bármi, két pont között minden csomag ugyanazt az utat fogja bejárni. Ez az úgynevezett *statikus forgalomirányítás* módszere. A második csoportba az *adaptív algoritmusok* tartoznak, amelyek felfigyelnek a topológiában, illetve a forgalmi viszonyokban történő változásokra, és ellentétben az előzőekkel, döntéseiket ezen változások figyelembevételével hozzák meg. Ezt nevezzük *dinamikus forgalomirányításnak*. Az adatív algoritmusok sok mindenben különbözhetnek egymástól. Egyesek mindig csak a szomszédos routerektől veszik az információkat, mások az alhálózatban lévő összes routerével kapcsolatban állnak. Különbözhetnek még abban is, hogy milyen módon próbálják az optimalitást meg-



2. ábra



3. ábra

valósítani. Például inkább a csomagkésleltetést kívánja-e csökkenteni, mint hogy a teljes alhálózat átbocsátóképességét a lehető legnagyobbra állítani. (A későbbiekben erre még részletesen is visszatérünk. Most csak azt jegyeznénk meg, hogy e két cél egymásnak ellentmond, így az algoritmusoknak megint valami kompromisszumos megoldást kell találni).

Az optimalitási elv

Azt már tudjuk, hogy a forgalomirányító algoritmusoknak törekedniük kell arra, hogy a hálózat optimalisan működjön. De vajon

mit jelent az, hogy optimális? Erre az úgynevezett *optimalitási elv* ad választ. Ennek a fogalom a bevezetése talán jobban megvilágítja a forgalomirányító algoritmusok működését. Most még nem foglalkozunk az olyan gyakorlati jellegű dolgokkal, mint például a hálózatban jelenlévő forgalom, vagy az alhálózat topológiája.

Az optimalitási elv a következő vallja: ha a B jelű router az A és a C router közötti optimális útvonalon helyezkedik el, akkor a B-től C-ig vezető optimális útvonal is része annak. (Ez egy nagyon egyszerűen bizonyítható tétel, ugyanis indirekt módon feltehetjük, hogy létezik egy jobb útvonal B és C között. Ekkor ezt összekapcsolhatjuk az A-ból B-be vezető útvonallal, így egy jobb A-C utat kapunk. Ez azonban ellentmondás, ugyanis az AC út az optimális). Ennek a tételnek egy érdekes következménye az, hogy az összes lehetséges forrástól egy adott célhoz tartó utak fát alkotnak, amelynek gyökere maga a cél. (A fa olyan gráf, amelyben nincs kör). Ezt a fát *nyelőfának (sink tree)* nevezzük. (3. ábra).

Észrevehető, hogy a nyelőfa nem feltétlenül egyedi, azaz nem csak egy optimális útvonal létezhet például az A router és a többi router között. A forgalomirányító algoritmusok feladata valójában pont az, hogy ezekre az utakra rábukkanjon.

A fák tulajdonságából következik az is, hogy minden csomag véges (korlátos számú) ugráson belül eléri célját. (Ugrás alatt most egy routeren történő keresztülhaladást értjük). Ez azonban csak a mi példánkban van így, a gyakorlatban minden bizonnyal nem áll fent. A routerek ugyanis elromolhatnak és hirtelen megjavulhatnak, így nem biztos, hogy például minden router ugyanúgy „képzeli el” a hálózat aktuális topológiáját. Gyakorlati kérdés az is, hogy az egyes routerek miként tudják beszerezni azokat az információkat, amellyel a nyelőfát felépíthetik. Erről a következő részben szólnunk részletesen, amikor a gyakorlatban is használt forgalomirányító algoritmusokkal ismerkedünk. Ugyan az optimalitási elvnek és a nyelőfának közvetlen gyakorlati haszna nincs, mégis egyfajta viszonyítási alapként szolgálnak a való élet forgalomirányító algoritmusainak összevetése során.

Garzó András
garzo@interware.hu