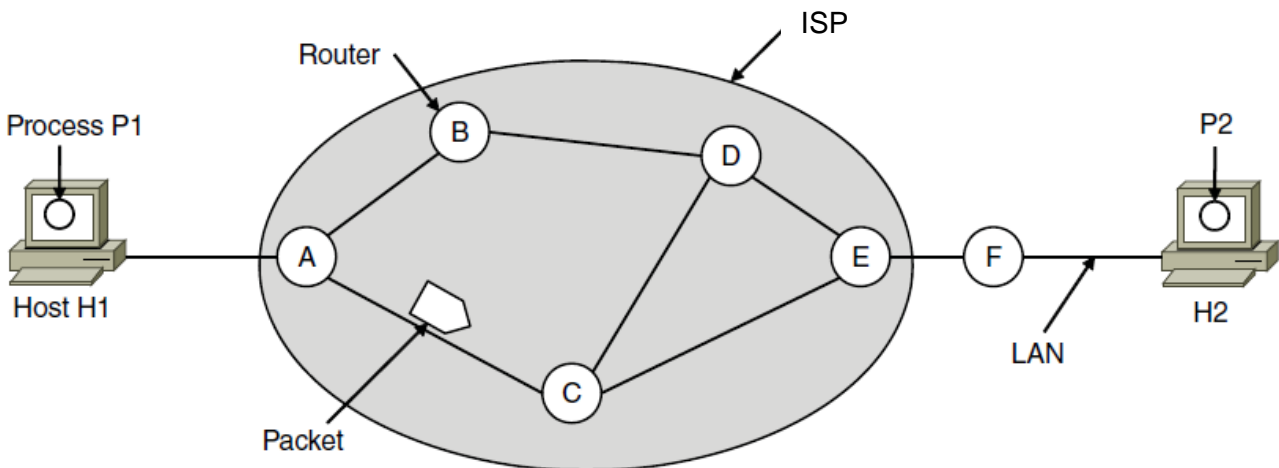


## 18. fejezet – A hálózati réteg és Az útválasztás

### A hálózati réteg

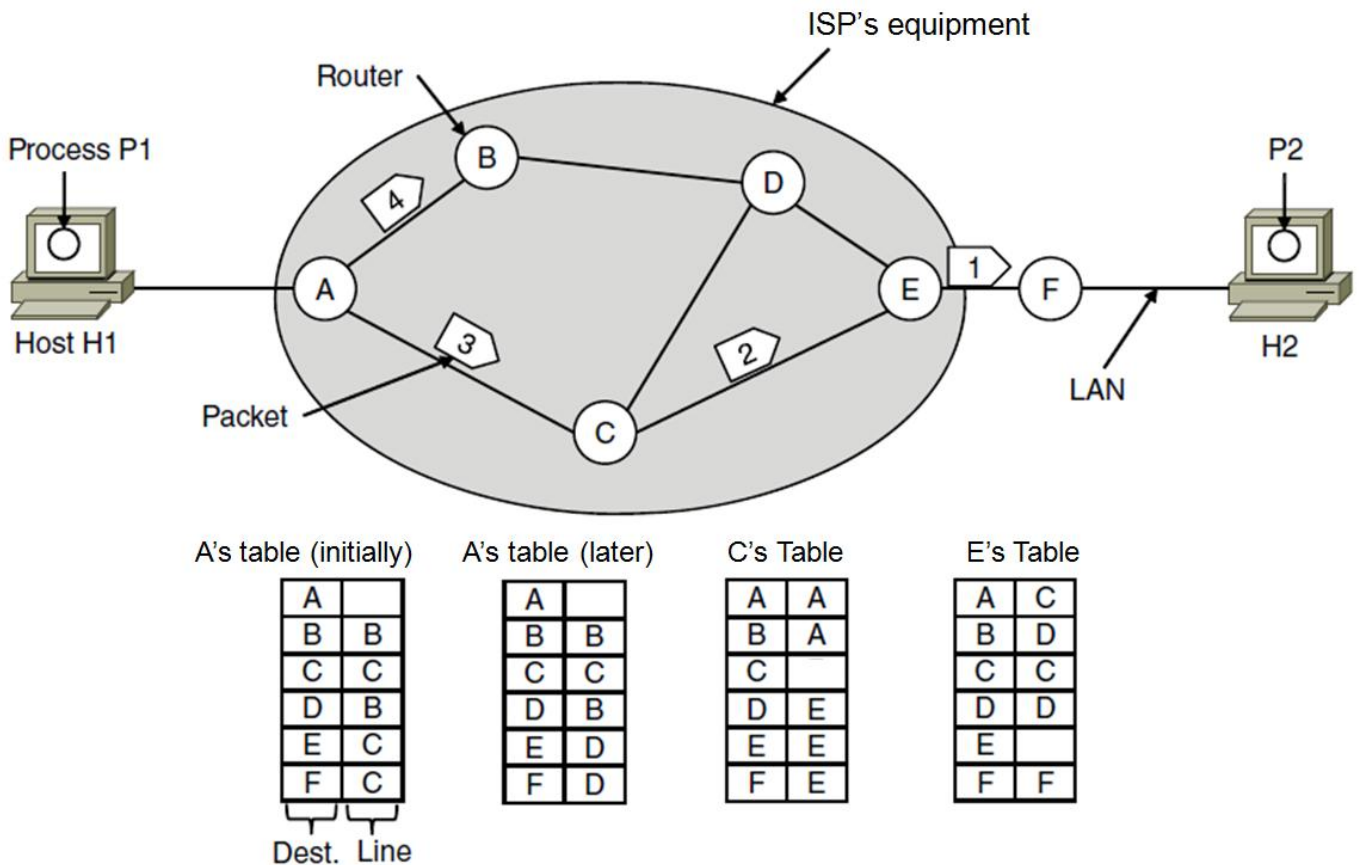
A hálózati réteg az alatta elhelyezkedő adatkapcsolati réteg szolgáltatásait igénybe véve, valamint saját szolgáltatásai segítségével szolgálja ki a felette elhelyezkedő szállítási réteget. Információs alapegysége a csomag. Legfontosabb feladata az, hogy a csomagok a forrás hoszttól a cél hosztig eljussanak. Ez a feladat összetettebb, és akár több hálózaton is átívelő lehet, tehát lényegesen összetettebb, mint az adatkapcsolati rétegnek a hálózat két pontja közötti keret átvitele. Olyan jellegű feladatok megoldása a cél, amikor különböző városok (vagy országok) különböző szolgáltatói (ISP) által üzemeltetett hálózatokban, a legkülönbözőbb módon csatlakoztatott ügyfelek között kell az adatforgalmat realizálni. A feladat tehát röviden ez azt jelenti, hogy a világháló két tetszőleges gépe közötti adatforgalmat kell realizálni. A hálózati réteg felett elhelyezkedő rétegek feladatai jellegüket tekintve másfélék, tehát a hálózati réteg az utolsó, amely a két végpont közti átvittel foglalkozik.

A hálózati réteg feladata – tehát az általa nyújtott szolgáltatás – az útvonalválasztás és a forgalomirányítás. A szolgáltatást olyan módon kell biztosítani, hogy a szállítási réteg már csak a számára releváns adatokat kapja meg. Ehhez arra van szükség, hogy a szolgáltatások függetlenek legyenek a hálózat – azon belül is – az útválasztók (Router-ek) kialakításától.



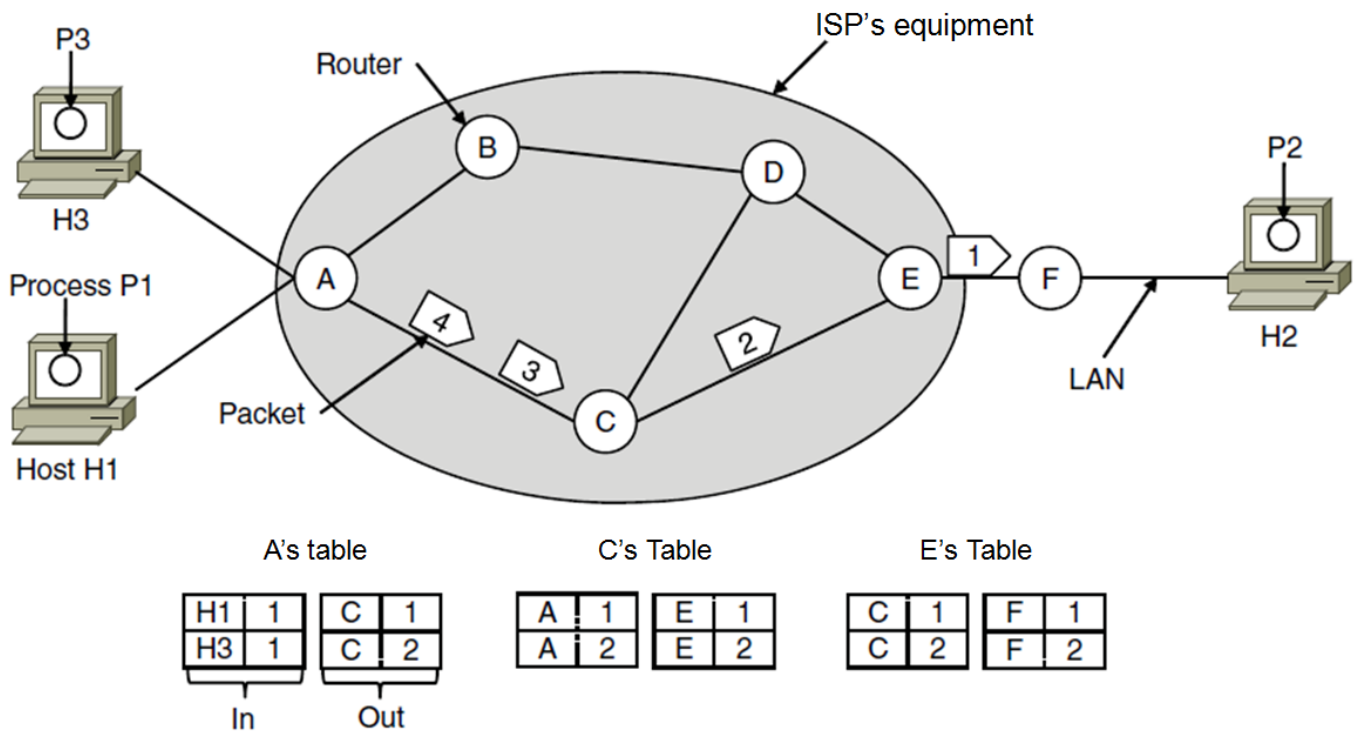
A szállítási réteg számára indifferens, hogy a csomag mennyi és milyen útválasztón keresztül érkezik, de szállítási réteg felé publikált hálózati címek rendszerének viszont – legyen szó LAN-ról vagy WAN-ról – egységesnek kell lennie. Ráadásul ezeket a szolgáltatásokat összeköttetés alapú, illetve összeköttetés nélküli rendszerek esetében is biztosítani kell.

Összeköttetés nélküli (pl. Internet) rendszerek esetében a csomagok egyenként és egymás után továbbítódnak. A kommunikációban részt vevő két hoszt között előzetes és direkt kapcsolat felépítés nem történik, ezért nevezzük ezt a modellt datagram alapú összeköttetésnek (Datagram Network), ahol csomagokra, mint datagramokra hivatkozunk. Az egyes datagramok gyakran különböző útvonalakon, különböző útválasztókon, és nem feltétlenül szigorúan sorrendben keresztül jutnak el a cél hoszthoz. Más megfogalmazásban, mivel minden csomag (datagram) tartalmazza a cél hoszt teljes címét, ezért minden csomag előbb vagy utóbb (inkább előbb...) célhoz ér.



A táblázatok azt mutatják meg, hogy melyik célállomás melyik útvonalon érhető el. Az elérhetőség időben változhat pl. torlódás, vagy meghibásodás miatt. A táblázatokat tehát folyamatosan karban kell tartani. Ezt a célt szolgálják az úgynevezett útválasztó algoritmusok (Routing Algorithm).

Összeköttetés alapú rendszerek esetében a kommunikációban részt vevő két hoszt előzetesen lekommunikálja az (optimális) útvonalat. A kommunikációban résztvevő összes csomag ezt az útvonalat fogja követni. Ezt a rendszert virtuálisáramkör alapú összeköttetésnek nevezzük, ami azért virtuális, mert nem klasszikus telefonközpontoknál megismert direkt fizikai összeköttetéseket tartalmaz. Ezt a rendszert címkekapcsolásnak (Label Switch) nevezzük.



A fenti ábra szerint a „H1” hoszt „P1” folyamata kialakított egy kapcsolatot „H2” hoszt „P2” folyamatával. Ezen a kapcsolat csomagjai az „1”-es összeköttetés azonosító címkét kapják a „H1” hosztban. Az összeköttetés azonosító címkék a táblázatok második oszlopában találhatóak. Az „A” útválasztó táblázatában látszik, hogy „H1” és „H3” hosztok esetében is '1'-es összeköttetés azonosító címkével érkeznek be a csomagok. A kimenő csomagokat azonban az „A” útválasztó már differenciálja, azaz „H1” csomagjai megtartják az „1”-es összeköttetés azonosító címkét, „H2” csomagjai pedig „2”-es összeköttetés azonosító címkét kapnak. Fontos tehát, hogy az útválasztók képesek legyenek a csomagok összeköttetés azonosító címkéinek a megváltoztatására, a hasonló helyzetek megoldása érdekében.

Mindkét megoldásnak vannak pozitív és negatív tulajdonságai is.

Érvként lehet felhozni a datagram alapú összeköttetés esetében, hogy itt nincs szükség idő- és erőforrás igényes összeköttetés felépítési fázisra. Ellenérvként viszont azt lehet felhozni, hogy ha már egyszer felépült a virtuálisáramkör alapú összeköttetés, akkor onnantól egyszerűbb a homogén csomagok irányítása, mint az hogy minden egyes csomag egyedileg kiválasztott útra kerüljön. Érdekes továbbá összehasonlítani a csomagok méretét, hiszen a datagram alapú összeköttetés esetében hosszabb a teljes célcím, mint a virtuálisáramkör kapcsolt hálózatok esetében az azonosító címke hossza. Sok kis csomag esetében a tényleges adatátvitel mennyisége szignifikánsan különbözhet. Arról sem szabad megfeledkezni, hogy az útválasztók memóriája is véges. Datagram alapú összeköttetés esetén minden lehetséges címzett egy önálló bejegyzéssel jár, míg a virtuálisáramkör alapú összeköttetés esetében csak a felépített virtuálisáramkörök azonosítóit (a továbblépést) kell tárolni.

	<b>Datagram alapú összeköttetés</b>	<b>Virtuálisáramkör alapú összeköttetés</b>
Áramkör felépítés	nem szükséges	megkövetelt
Címzés	minden csomag tartalmazza a teljes forrás- és célcímet	minden csomag egy rövid összeköttetés azonosító címket tartalmaz
Állapotinformáció	az útválasztó nem tartalmazza	minden útválasztó táblázatában minden összeköttetés erőforrást (helyet) igényel
Útválasztás	minden csomag esetében egyedi	az útvonal a csomagküldés előtt felépül, minden csomag ezt az utat követi
Egy közbülső útválasztó meghibásodásának következménye	csak a meghibásodott útválasztóban lévő csomagok vesznek el	az érintett virtuális áramkör(ök) megszakad(nak), a kapcsolat szétesik [újra kell építeni...]
A szolgáltatás minőségének biztosítása	bonyolult	könnyű, amennyiben előzetesen minden útválasztóban elegendő erőforrás foglalható le az összeköttetés számára
Torlódáskezelés	bonyolult	könnyű, amennyiben előzetesen minden útválasztóban elegendő erőforrás foglalható le az összeköttetés számára

Egyik megoldás sem tekinthető univerzálisnak. Mindkét megoldásnak megvannak a tipikus alkalmazási példái. A relatíve rövid ideig tartó és eseti kapcsolatok esetén (pl. hitelkártyás fizetés egy áruház pénztárában) indokolatlan virtuálisáramkör alapú kapcsolatot felépíteni. Hosszú ideig működő, például egy cég két telephelyét összekötő kapcsolat esetében (VPN – Virtual Private Network / Virtuális Magánhálózat) viszont a virtuálisáramkör alapú kapcsolat a praktikusabb.

## Útválasztó algoritmusok

A cél és egyben a réteg fő feladata tehát az, hogy a csomagok a forrás hoszttól a cél hosztig eljussanak. Tipikusan ehhez jelentős mennyiségű útválasztón kell a csomagoknak keresztüljutni. Különleges esetnek az adatszórás számít, hiszen jellemzően nem az a cél, hogy az egész internetet, azaz az összes fellelhető hoszt a címzettek között legyen, hanem nyilván csak a megcélzott hálózat (vagy alhálózat) hosztjai. Útválasztásra tehát ez esetben is szükség lehet, amennyiben a forrás hoszt és a cél hosztok nem azonos hálózatban találhatók.

Szem előtt kell tartani azt is, hogy ez eddig megismertek szerint maga az útválasztás a gyakorlatban legalább két folyamatból áll. Egyrészt a beérkező csomagokat az útválasztó táblázatában szereplő bejegyzéseknek megfelelően továbbítani (Forward) – a magyar szakmai zsargon szerint „forwardolni” – kell. Másrészt a táblázatok bejegyzései is folyamatos karbantartást, aktualizálást igényelnek.

Az útválasztó algoritmusok két osztálya sorolhatóak.

- Adaptív algoritmusok (Adaptive Algorithm)
  - dinamikus eljárás
  - a hálózati forgalomhoz és a topológiához alkalmazkodnak az útválasztás döntései
- Nem adaptív algoritmusok (Nonadaptive Algorithm)
  - statikus eljárás
  - az útválasztással kapcsolatos döntések sem a pillanatnyi forgalommal illetve topológiával kapcsolatos mért adatok, sem becslések nem befolyásolják

[adaptív: alkalmazkodó]

A forgalom és a topológia, mint döntési faktorok jellemzően nem egyszerre hatnak, azaz léteznek topológia alapú, és léteznek forgalom alapú döntések.

A fentiekből következő öt vezérlési mód a következő.

1. Determinisztikus forgalomirányítás  
Olyan rögzített eljárás, amelyet a változó feltételek nem befolyásolnak.
2. Elszigetelt adaptív forgalomirányítás  
Mely esetben minden csomópont hoz irányítási döntéseket, de csak helyi információk alapján.
3. Elosztott adaptív forgalomirányítás  
Mely esetben a csomópontok információt cserélnek azért, hogy az irányítási döntéseket a helyi és a kapott információkra együtt alapozhassák.

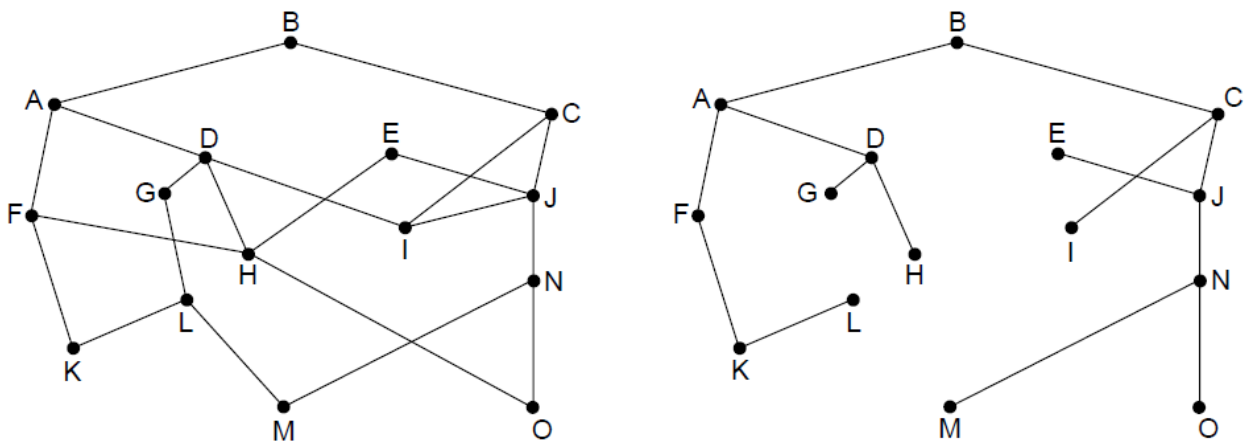
#### 4. Központosított adaptív forgalomirányítás

Mely esetben a csomópontok a helyi forgalmi információikat egy közös irányító központnak jelentik, amely erre válaszul forgalomirányítási utasításokat ad ki az egyes csomópontok részére.

#### 5. Delta forgalomirányítás

Ennél az eljárásnál a központi irányító egység munkáját a forgalomirányítási döntésekhez kizárólag abban az esetben használják fel, ha ezek a helyi információkra nem alapozhatók.

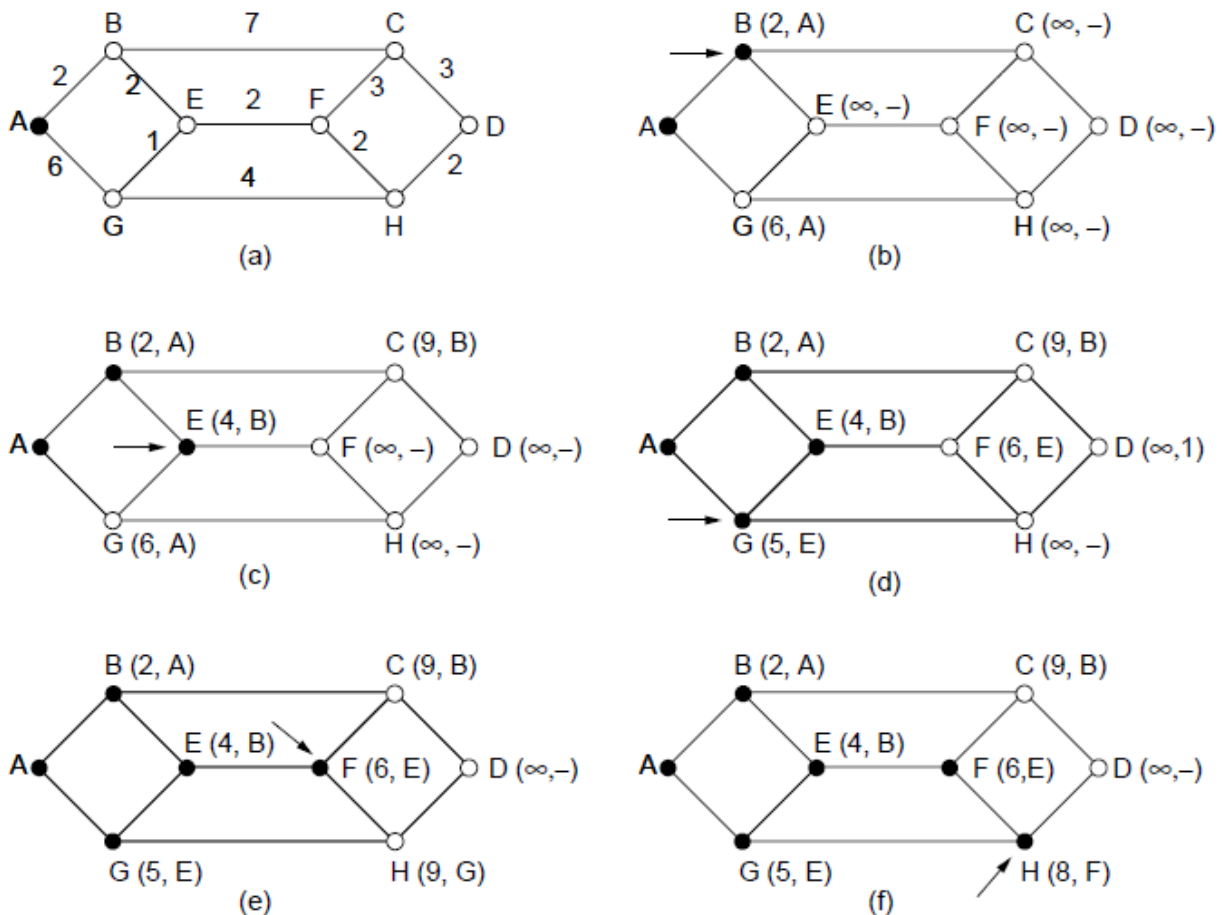
Topológia alapú döntések a forgalom ismerete nélkül, pusztán a hálózat felépítése alapján meghozhatóak. Egy optimális, vagy egy legrövidebb útvonal gráfelméleti (illetve fa struktúrával kapcsolatos) döntés. Könnyen belátható, hogy az összes forrásból egy közös célba tartó optimális útvonalak egy fa struktúrát alkotnak, ahol a cél a fa gyökere. A távolságot ez esetben az ugrások számával definiáljuk. Ennek a struktúrának az elnevezése a nyelőfa (Sink Tree)



A fenti ábra bal oldalán egy hálózati megvalósítás látható, jobb oldalán pedig az előbbi hálózatból a „B” útválasztóhoz kialakított nyelőfa látható. Tekintettel arra, hogy a nyelőfa természetesen nem tartalmaz hurkot, így minden csomag véges számú ugrás után el kell hogy jusson a cél hosztig.

## A legrövidebb útvonal alapján történő útválasztás

Nyilvánvaló hogy a forgalomirányítás során két pont között meg kell találni az optimális útvonalat, amely véges számú csomóponton vezet keresztül. Az optimális útvonal nem feltétlenül jelenti a fizikailag legrövidebb útvonalat, mivel számos egyéb tényező is befolyásolhatja az optimális választást. Mértékadó információ lehet például a csomópont átlépések száma, az az idő, amennyi alatt a csomag eljut a céljáiig, illetve a vonalhasználat költségei. Az objektív mérték megállapításához olyan tesztekkel kell futtatni az adott szakaszokon, amely megadja az átlagos sorbaállási és átviteli késleltetési időt, azaz a mértéket. Általánosan egy adott szakasz mértékét a távolság, az adatátviteli sebesség, az átlagos forgalom, a kommunikációs költség, az átlagos sorhossz vagy más egyéb tényezők alapján határozzák meg. A legrövidebb út tehát sok paraméter együttes vizsgálata után az egyes komponensek súlyától (súlyfüggvényétől) függően több különböző út is lehet.



A fenti ábra egy 8 csomópontból („A”-„H”) álló hálózatot ábrázol az útvonalhosszak megjelölésével. A vonalakra írt számok a kalkulált súlyok (például a távolság és a költség alapján). Az algoritmus az „A” és a „D” közötti legrövidebb út megtalálását célozza meg. A nyíl mindig az aktív, az adott lépés alatt állandónak tekintett munkacsomópontot jelzi.

Első lépésben ez természetesen az „A” csomópont. A címkéző algoritmus „A” összes szomszédjára kiszámítja az „A”-tól való legrövidebb távolságot. Az algoritmus a szomszédokat el is látja egy ideiglenes címkével. Jelen esetben ez „B” esetében 2, „G” esetében 6. [A zárójelben lévő értékek: a legrövidebb távolság az „A” csomóponttól, a legrövidebb út első lépése „A” irányába.] Következő lépésként állandóvá tesszük a legkisebb címkéjű szomszédot, ez esetben „B”-t, majd az ezt követő lépésben „E”-t. A negyedik lépésben látszik, ahogy „G” legrövidebb távolsága 6-ról 5-re, kijelölt iránya pedig „A”-ról „E”-re változik. Az ideiglenes címkék a ciklus végén (az ábra csak az első 6 lépést mutatja) válnak véglegesé.

A módszer (a szemafort is kiagyáló) Edsger Wybe Dijkstra holland matematikustól származik, 1959-ből. [A fenti útválasztó algoritmus a korábbi tanulmányokból a hallgatók egy részének már ismert lehet.]

Vannak olyan determinisztikus forgalomirányító módszerek, amelyeknél nincs szükség semmilyen forgalomirányítási táblára, a hálózati topológia ismeretére, minden csomópont autonóm módon, azonos algoritmus alapján dolgozik.

## A véletlen forgalomirányító eljárás

A véletlen forgalomirányító eljárás alapján működő rendszerben a továbbítandó csomagot a csomópont egy úgynevezett véletlen folyamat segítségével kiválasztott az érkező vonaltól eltérő, más vonalon küldi tovább. Mivel a hálózat által ilyen módon szállított csomagok véletlen bolyonganak, logikus megoldás, ha a csomagokhoz hozzárendeljük a mozgásuk során bejárt szakaszok számát és töröljük azokat a csomagokat, amelyek lépésszáma eléri egy számlálóban az előre meghatározott határértéket. Ez az eljárás nem garantálja a csomagok kézbesítését (főleg nagy kiterjedésű hálózatokban), de nagyon egyszerűen realizálható, és nem túl bonyolult hálózatokban jól működhet.

## Az elárasztásos forgalomirányító eljárás

Az elárasztásos forgalomirányító eljárás sem igényel semmi ismeretet a hálózatról. A csomópontok, mikor egy csomagot továbbítanak, a bejövő csomagot minden vonalra kiküldenek, kivéve ahonnan érkezett. A lépések száma természetesen itt is korlátozva van. Jelentős érdeme a módszernek, hogy a csomag legalább egy példányban mindenképp a legrövidebb úton ér célba. A módszer azonban jelentősen terheli a rendszert, mivel nagyszámú másolat (redundancia) van, és sok felesleges továbbítás történik. Az algoritmus rendkívül megbízható, és még megsérült rendszer esetén is működőképes.



Érthető, hogy például katonai alkalmazások esetén előtérbe kerülhet a módszer, mert erősen sérült hálózatban (ahol csomópontok semmisülhetnek meg) is nagy a valószínűsége egy üzenet (azaz a csomagok) célbaérésének.

## A távolságvektor alapú útválasztás

Az távolságvektor alapú útválasztás (Distance Vector Routing) esetében minden útválasztónak egy olyan táblázatot kell karbantartania, melyben minden célhoz szerepel az aktuálisan legrövidebb távolság illetve az oda vezető vonal azonosítója (azaz célonként egy-egy vektor). A táblázat karbantartása, frissítése a szomszédos útválasztók kommunikációjával történik. Az ARPANET 1979-ig ezt a módszert használta.

## A kapcsolatállapot alapú útválasztás

A kapcsolatállapot alapú útválasztás (Link State Routing) a távolságvektor alapú útválasztást váltotta fel, mivel ez a modell időben sokkal gyorsabban képes reagálni a hálózat topológiájának megváltozásaira. Az ARPANET 1979-től egészen az 1990-es megszűnéséig ezt a módszert használta, sőt a módszer részben ma is használatos. A módszer algoritmus 5 lépésére bontható minden útválasztó szempontjából.

1. Felkutatni a szomszédos útválasztókat és velük hálózati címeket cserélni.
2. Meghatározni az adatok késleltetése alapján a távolság és/vagy költség értéket.
3. Az eddig megszerzett információk rendszerbe, táblázatba foglalása.
4. A táblázat publikálása az összes többi útvonalválasztó részére.
5. A táblázatok alapján le lehet futtatni a legrövidebb útvonal esetében használt Dijkstra által kidolgozott algoritmust. Így az összes útválasztó szempontjából elkészülnek a legrövidebb útvonalak.

## A hierarchikus útválasztás

A hálózatok méretének és az útválasztók számának növekedésével az eddigiekben tárgyalt táblázatok mérete is arányosan kellett hogy növekedjen. Ez jelentős erőforrásokat (CPU, memória) igényelt az útválasztóktól, sőt a hálózatot is jelentős forgalommal terhelte meg. A hálózatok így elméleti és gyakorlati korlátokba is ütköztek, nem növekedhettek tovább korlátlanul.

A hierarchikus útválasztás alapelve, hogy a hálózatokat tartományokra (Region) kell osztani. Az egyes útválasztók csak a tartományon belül kell hogy a csomagok forgalomirányítását elvégezzék, illetve fel kell tudniuk ismerni azt, ha egy csomag címzettje a tartományon kívül található. Ennek alapján felépíthető egy olyan táblázat,

amely azokat a paramétereket tartalmazza, hogy egy tartományon kívüli csomagot a tartomány melyik kapcsolódási pontján (azaz melyik másik tartomány felé) kell irányítani. Ez a megoldás csökkenti az egy útválasztón belüli bejegyzések mennyiségét, de az igazán optimális megoldás a címzési rendszer hierarchikus kialakítása. A MAC címek semmilyen hierarchiát, területi vagy elhelyezkedési információt sem hordoznak, ezért szükség volt egy valóban hierarchikus címzési rendszer kialakítására is. Ezt a célt az IP (Internet Protocol) címek használatával sikerült elérni, melyeket részletesen fogunk tárgyalni.