

Számítógép-hálózatok

A hálózati réteg

2025/2026. tanév, I. félév

Dr. Kovács Szilveszter

E-mail: szilveszter.kovacs@uni-miskolc.hu

www.iit.uni-miskolc.hu/~szkovacs

Informatika Intézet 107/a.

Tel: (46) 565-111 / 21-07

Miről lesz szó?

- **A hálózati réteg**
 - Feladata a csomagok forráscsomóponttól célcsomópontig való eljuttatása
 - felette, alatta vég-vég (end-to-end) alapú, itt az egész hálózat látszik
 - címzések, funkciók, hálózatszervezés (ÖK alapú, ÖK mentesség)
- **Forgalomirányítás**
- **Torlódásvezérlés**
- **Hálózatközi együttműködés**

A hálózati réteg feladata

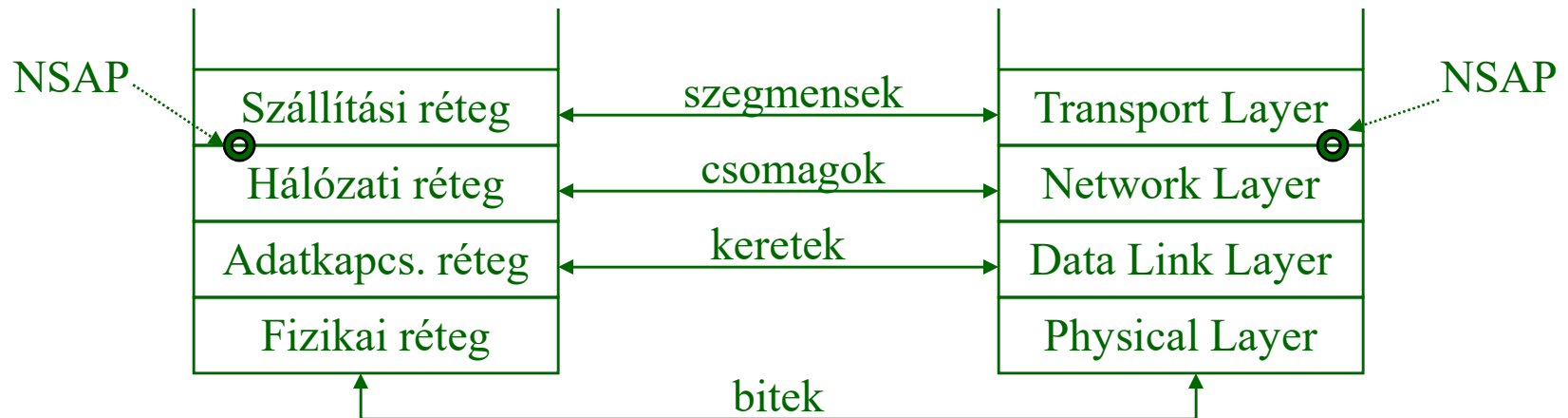
- Csomagok forráscsomóponttól célcsomópontig való eljuttatása
 - Az adatkapcsolati réteg: csak egyetlen „vonal” (single link) két vége közötti keretmozgatást végzi, pl. hibajavítás, sorrendezés, forgalomszabályozás
 - A szállítási réteg: valódi vég-vég (end-to-end) alapú (valódi forrás-cél kapcsolat)
 - a hálózat topológiáját a csomagok célba juttatásának módját nem ismeri.
- A hálózati réteg két végpont közötti (end-to-end) átvitellel mikéntjével foglalkozik (ismernie kell a hálózatot, topológia stb.)

.....

Szállítási réteg
Hálózati réteg
Adatkapcs. réteg
Fizikai réteg

A hálózati réteg feladata

- **Általánosan:**
 - jól meghatározott szolgáltatások a szállítási réteg felé, azaz
 - a szállítási funkcionális elemtől NSAP-on át (Network Services Access Point) kapott és megcímzett (hálózati célcím, forráscím) adataegységet (csomagot) a cím szerinti NSAP-hoz (a funkcionális társelemhez) (és sehová máshová) eljuttatni.

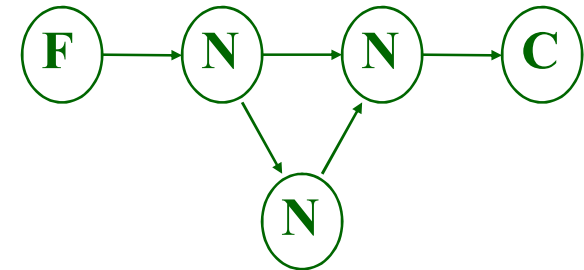


NSAP (Network Service Access Point) cím: hálózati szolgálatot azonosít

A hálózati réteg funkciói

- **Forgalomirányítás**

- a csomag célba juttatása.
- ismerni kell a topológiát
- terhelésmegosztás (alternatív utak)



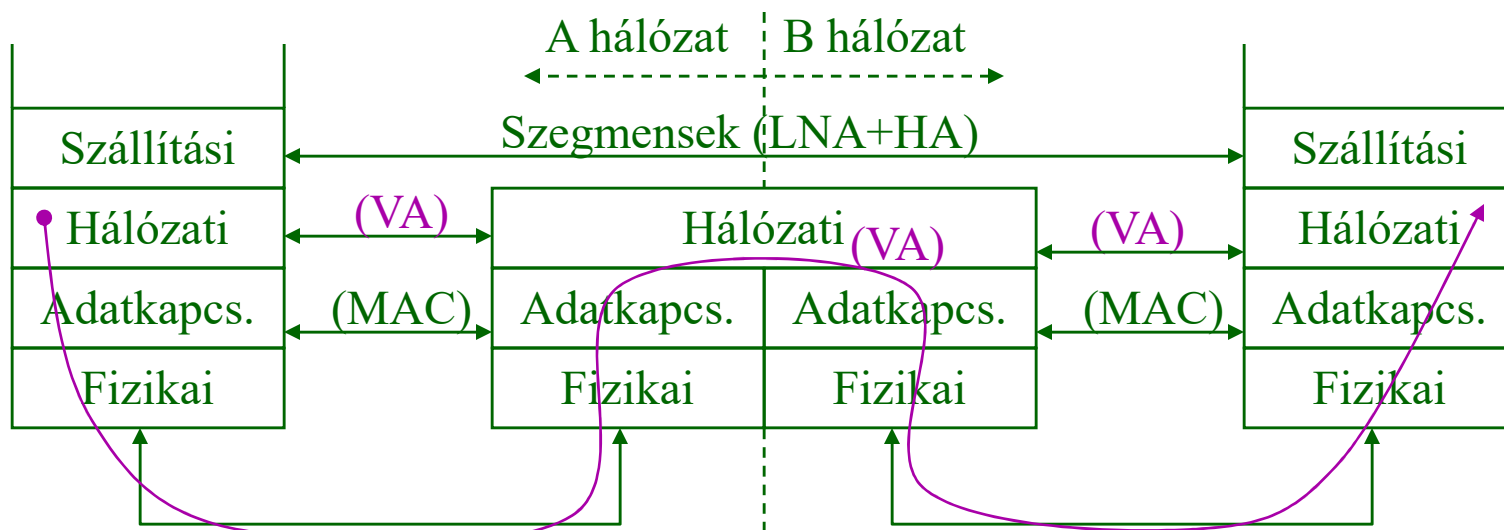
- **Torlódásvezérlés**

- Ne legyenek a hálózat egyes részei túlterheltek
- Hasonló a forgalomszabályozáshoz, de ez nem csak két pont (adó-vevő) közötti, hanem a hálózat egészére vonatkozik.

- **Hálózatközi együttműködés**

- Ez az első réteg, ahol különböző hálózatok összekapcsolhatók (heterogén hálózatok kialakítása)

Hálózatközi együttműködés



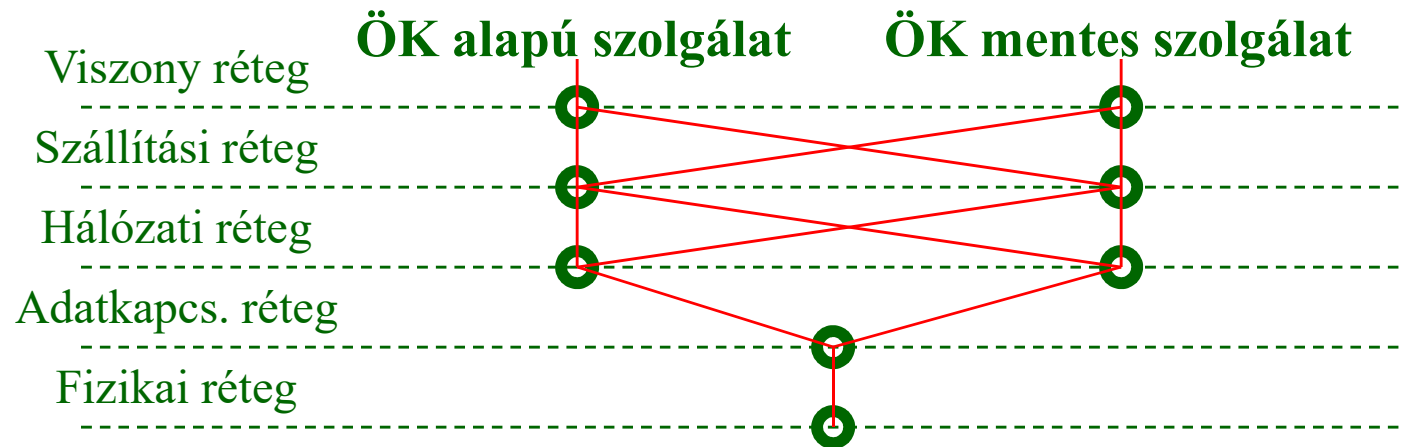
- A hálózati csomópontok funkciói (forgalomirányítás, torlódásvezérlés) csak a hálózati rétegig terjednek
- Különböző adatkapcsolati rétegek (heterogén hálózat) lehetnek alattuk

Címek:

- LNA: Hálózat cím (Logical Network Address), HA: Hoszt cím (Host Address)
- VA: virtuális áramkörön cím (ha van)
- MAC: Media Access Control (üzenetszórásos közeg esetén)

A szállítási rétegnek nyújtott szolgáltatok

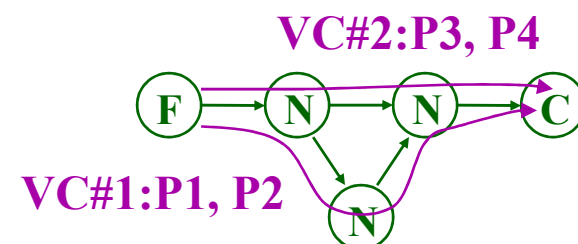
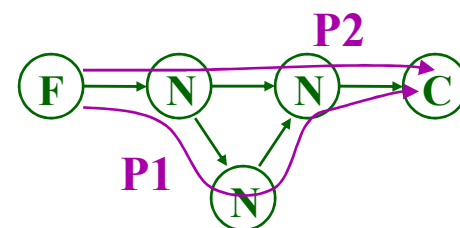
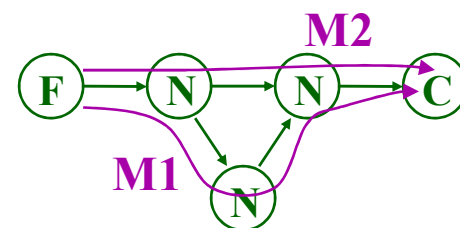
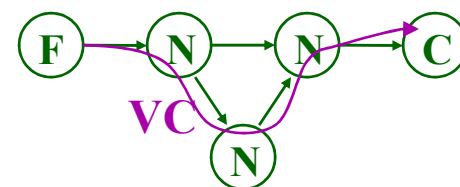
- **Lehetnek:**
 - Összeköttetés alapú (virtuális áramkör)
 - Összeköttetés-mentes (datagram)
- Ezen szolgálattípusok több szinten is jelentkeznek, esetleg szintenként más-más lehet



Pl: ÖK alapú adatkapcsolati szolgáltatón lehet ÖK mentes hálózati szolgálat
(pl: kapcsolt vonalon ÖK mentes hálózat) – és fordítva

Szolgáltatástípusok

- **Összeköttetés alapú szolgálat**
 - (Áramkör kapcsolás, Circuit Switching)
- **Összeköttetés-mentes szolgálat**
 - Üzenetkapcsolásos (Message Switching) (ÖK mentes)
 - Csomagkapcsolásos (Packet Switching)
 - Datagram Packet Switching (ÖK mentes)
- **Összeköttetés alapú szolgálat**
 - Virtuális vonalkapcsolás (ÖK alapú)
 - Virtual Circuit Packet Switching



A hálózati réteg belső szervezése

Két különböző alhálózat-szervezési filozófia:

- **Összeköttetések – virtuális áramkörökön alapuló**
- **Összeköttetés mentes – datagram**
- **A virtuális áramkör szervezés kedvező, ha:**
 - Elsődlegesen összeköttetés alapú szolgáltatot biztosít
 - Ne legyen minden egyes csomagra forgalomirányítás (datagram alhálózatban nincs előre meghatározott útvonal még akkor sem, ha a szolgálat ÖK alapú)

A virtuális áramkörön alapuló alhálózat-szervezés

- A hívásfelépítés során a forrás és célállomás között virtuális áramkör (Virtual Circuit) alakul ki.

⇒ A forgalomirányítás a hívásfelépítéskor történik!

- A kommunikáció során a csomagok ugyanazon az úton, a nyitott virtuális áramkörön (VC) haladnak (mindkét irányban).
- A kommunikáció befejeztével a virtuális áramkört fel kell szabadítani
- Az egyes csatornákon több virtuális áramkör is lehet (számuk maximált).

A virtuális áramkörön alapuló alhálózat-szervezés

Címzés:

- Csak a hívásfelépítés során van szükség a teljes forrás, célcímre
 - Ezt követően már elegendő a virtuális áramkör jelzése
- ⇒ Minden csomagban mező a virtuális áramkör jelzésére
(Az egyes csatornákon több virtuális áramkör is lehet (számuk maximált)).

A virtuális áramkörön alapuló alhálózat-szervezés

Hívásfelépítéskor:

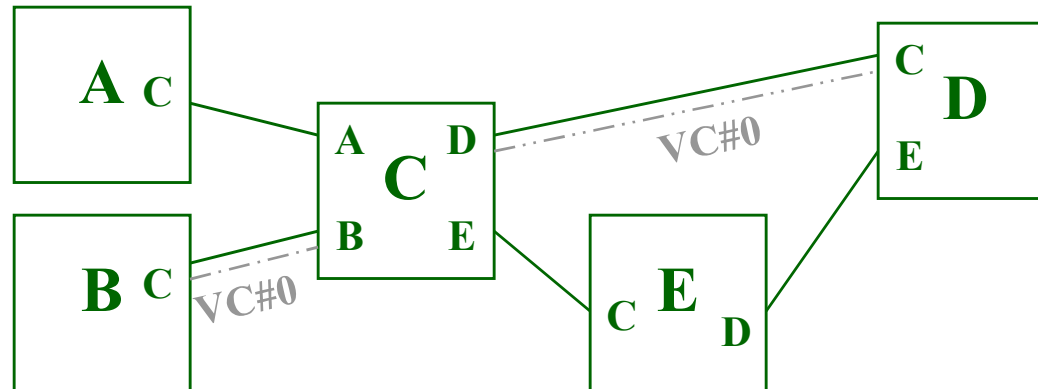
- A csomópont kiválasztja a megfelelő irányú csatornát (útvonalat) és azon virtuális áramkört foglal le (pl. a legkisebb szabad sorszámút)
- Ha nincs szabad áramkör, másik útvonalat választ. Ha ez sincs, a **hívásfelépítés sikertelen**.
- Ez ismétlődik az útvonalat érintő valamennyi csomópontra

⇒ Minden csomópontban táblázat a nyitott virtuális áramkörökről:

melyik vonal melyik áramköre –

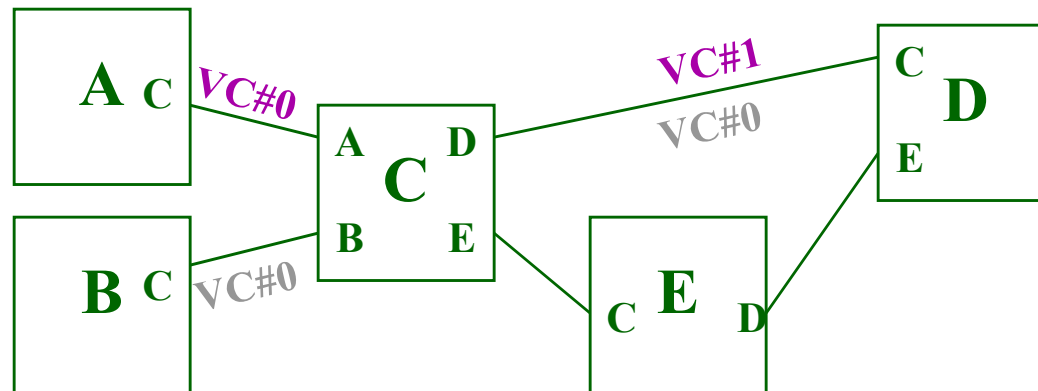
melyik vonal melyik áramköréhez kapcsolódik

Példa virtuális áramkörön alapuló alhálózat-szervezésre



- Van 5 csomópont: A, B, C, D, E
- Az egyes csomópontokban csatornák jele a szomszéd neve.
- Induláskor már létezzen B-C-D között egy virtuális áramkör.
- Egy csatornán maximum 2 VC alakítható ki.
- Feladat: A és D között két VC létrehozása

Az első VC A-ból D-be ...

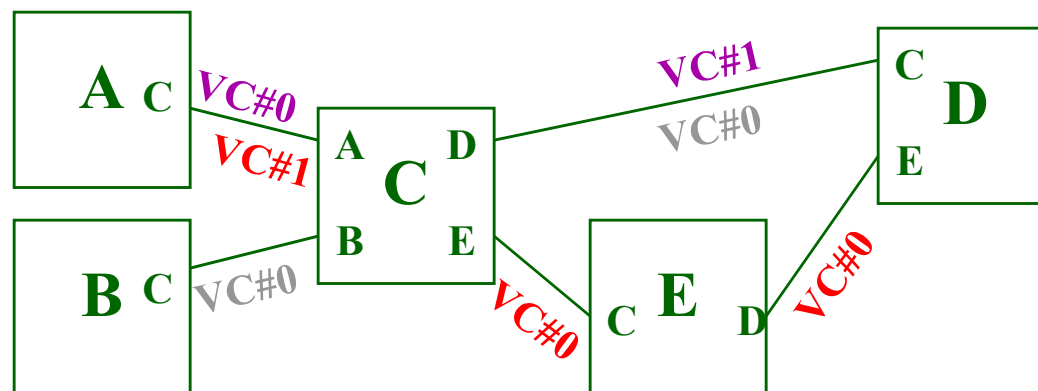


A táblák a csomópontokban

A	B	C	D	E
	C0	B0 - D0	C0	
C0		A0 - D1	C1	

VC alapú hívásfelépítés: első

A másik VC A-ból D-be ...

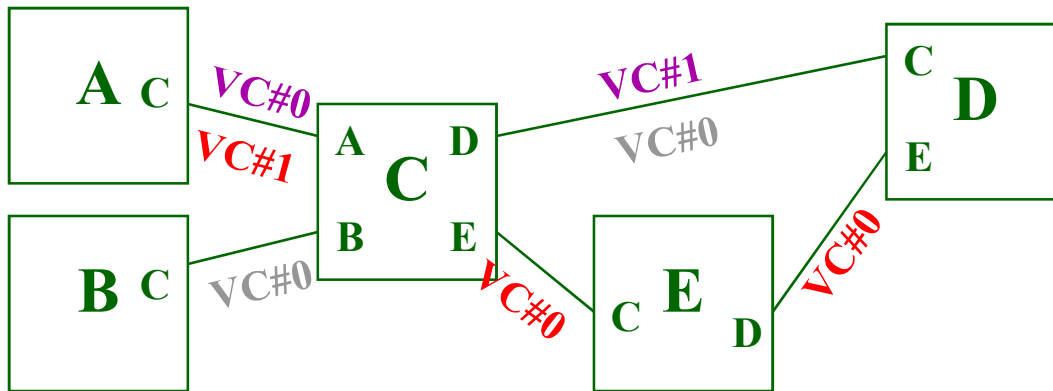


A táblák a csomópontokban

A	B	C	D	E
	C0	B0 - D0	C0	
C0		A0 - D1	C1	
C1		A1 - E0	E0	C0 - D0

VC alapú hívásfelépítés: második

A kommunikáció C szempontjából



- Ha C az A-tól VC#0 csomagot kap: a virtuális áramkör azonosítót VC#1-re módosítja és továbbadja D felé (a táblázata 2. sora szerint)
- Ha C az A-tól #1 jelzésű csomagot kap: a csatorna azonosítót VC#0-ra módosítja és továbbadja E felé
- Ha C a B-től #0-val kap csomagot: VC#0-val továbbítja D-nek

C		
B0	-	D0
A0	-	D1
A1	-	E0

A csatorna lebontása: a táblabejegyzések törlése

VC alapú kommunikáció

VC alapú kommunikáció

- **Lehetne más technika?**
 - **Pl:**
 - **Nincsenek táblák a csomópontokon (de a csatornacímek ismertek!)**
 - **Hívásfelépítéskor a forrás összegyűjti az útvonal csatornacímeit, és ezt elhelyezi minden csomag címében**
pl VC#0 A-tól D-ig: A-C-D
VC#1 A-tól D-ig: A-C-E-D
 - **A router így minden beérkező csomagnál a címből közvetlenül tudja, hogy azt melyik csatornáján kell továbbítani**
(ez a forrás általi forgalomirányítás - **source routing**)

Datagram alapú alhálózat szervezés

- Minden csomag teljes cél, forráscímet tartalmaz
Cím = állomás cím + NSAP cím
(az állomás cím lehet **hálózat + hoszt cím**)
- Az egyes csomagok egymástól függetlenül haladnak,
⇒ **minden csomagra külön-külön van forgalomirányítás**
– (mehetnek más-más úton is)

A virtuális áramkör és a datagram alapú alhálózat-szervezés összehasonlítása

	Virt. áramkör	Datagram
Áramkör létesítés	Szükséges	Nem lehetséges (független)
Címzés	Minden csomag csak rövid VC címet tartalmaz	Minden csomag teljes címet tartalmaz (ovehead)
Állapotinformáció	Minden nyitott VC táblabejegyzést igényel valamennyi érintett csomópontban	Az alhálózat állapotmentes
Forgalomirányítás	Csak áramkörfelépítéskor	Minden csomag esetén újból
Torlódásvezérlés	Könnyű: ismert számú VC-re lehet előre puffereket foglalni	Nehéz
Csomóponti hibák hatása	A csomóponton átmenő valamennyi VC megszakad	Nincs, legfeljebb egyes csomagokra
Összetettség	A hálózati rétegben	A szállítási rétegben
Tipikusan alkalmas	ÖK alapú szolgálatra	ÖK mentes szolgálatra

De alkalmasak mindketten ÖK alapú és ÖK mentes szolgálat nyújtására is.

A hálózati réteg funkciói

- **Forgalomirányítás**
 - a csomag célba juttatása.
 - ismerni kell a topológiát
 - terhelésmegosztás (alternatív utak)
- **Torlódásvezérlés**
 - Ne legyenek a hálózat egyes részei túlterheltek
 - Hasonló a forgalomszabályozáshoz, de ez nem csak két pont (adó-vevő) közötti, hanem a hálózat egészére vonatkozik.
- **Hálózatközi együttműködés**
 - Ez az első réteg, ahol különböző hálózatok összekapcsolhatók (heterogén hálózatok kialakítása)

A forgalomirányítás

- **A forgalomirányító algoritmus (routing alg.) dönti el, hogy a beérkező csomagot melyik kimenő vonalon kell továbbítani**
 - **Datagram alapú alhálózat esetén: minden csomagra külön-külön,**
 - **Virtuális áramkör alapú alhálózat esetén: csak az új virtuális áramkör létrehozásakor (hívásfelépítés)**
⇒ viszony-forgalomirányítás (session routing)

Alapkövetelmények, tervezési szempontok

- **Egyszerűség, megbízhatóság**
- **Helyesség** (azt tegye, ami a dolga
→ egy példányban, a megadott címre)
- **Robosztusság**: meghibásodás esetén is maradjon működőképes (legalább valamilyen mértékben)
- **Adaptivitás**: adaptív, ha képes önállóan felépülni és alkalmazkodni a pillanatnyi körülményekhez
- **Stabilitás**:
indulástól véges idő alatt stabil állapotba kerüljön
- **Optimalitás**:
pl. költség, késletetés, min. ugrásszám szempontjából

Az útvonalválasztás lépései

- **Döntések:** a (router) csomópontok hozzák, merre továbbítsák a vett csomagot.
- **Információgyűjtés:** a döntésekhez szükséges információk megszerzése.
Pl: táblázatok létrehozása, mely a cél címekhez továbbítási irányokat rendel.

Forgalomirányítási módszerek

Hierarchikus forgalomirányítás:

- Valamennyi célcím táblázatba gyűjtése esetén
→ túl nagy táblák (túl sok szolgálati kommunikáció)
(nagy hálózatok esetén)
- Megoldás: hierarchikus hálózat kialakítása
→ a teljes hálózat alhálózatokra, al-alhálózatokra stb. bomlik
- Az alhálózatokra bontás szempontjai:
 - földrajzi elhelyezkedés;
 - funkcionális összetartozás (pl. közös cél);
 - fizikai közeghatárok, adatkapcsolati protokollok szerint⇒ Lehetőleg egyenletes elosztásra kell törekedni
- Elég az egyes alhálózatok elérési irányát és csak a saját alhálózat cél cím szerinti irányait tartalmaznia a táblázatnak

Pl: Címzésben:

hálózat cím	alhálózat cím	hoszt cím	: Cím
-------------	---------------	-----------	-------

Forgalomirányítási döntési módszerek

Lehetnek:

- **Egyutas, vagy többutas;**
- **Táblázat alapú, vagy táblázat nélküli módszerek.**

Egyutas forgalomirányítás

- Minden címhez egy továbbítási irányt tárol (táblázat)
- Előnye:
 - egyszerűség
 - optimális lehet (ha a tárolt irányok optimálisak)
- Hátránya:
 - nem robosztus (nem hibatűrő).

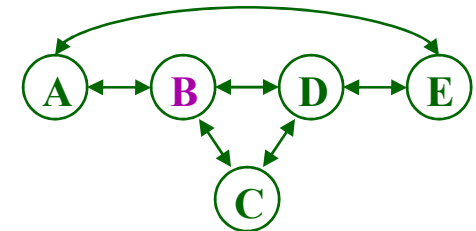
Többutas forgalomirányítás

- Minden címhez több, súlyozott továbbítási irány, melyek közül pl. súlyozott sorsolással választhat.

Pl. **B** egy táblabejegyzése a következő

E	A: 0,75; C: 0,2; D: 0,05
----------	---------------------------------

cím elérése 3 lehetséges irány, 3 súly



Többutas forgalomirányítás

- **Az irányok közüli választás szempontjai (sorsolási súlyok):**
 - előre megadott (fix) súlyok;
 - prioritás (a csomagok prioritás határozza meg);
 - a kommunikáció típusa: a forgalmi osztályok szerint (pl. gyors választ, vagy nagy sávszélességet igényel)
 - a helyi sorok mérete (kifelé menő vonalak terheltsége) szerint (terhelésmegosztás)
- **Előny:**
 - több szempont figyelembevételére alkalmas;
 - robosztus;
 - adaptív.
- **Hátrány:**
 - Bonyolultabb (több feldolgozási időt igényel).

Táblázat nélküli módszerek

A „forró krumpli” módszer:

- amerre a legrövidebb a sor, arra továbbítjuk a csomagot (minél korábban szabadulhasson tőle)
- Előny:
 - Nem kell információt gyűjteni,
 - egyszerű, robosztus.
- Hátrány:
 - Rossz vonali kihasználtság,
 - a késleltetési idő nem korlátos.

Táblázat nélküli módszerek

Az „elárasztásos” (flooding) módszer:

- minden csomagot minden irányba továbbít, kivéve ahonnan jött.
⇒ Nagyszámú többszörözött csomagot eredményez.
- Fékezési mechanizmusok:
 - **Ugrásszámlálással** (csomag fejlécben mező, melyet minden csomópont állít)
 - Egy bizonyos ugrásszám (a hálózat max. átmérője legalább) után minden csomópontok eldobj őket.
 - **Csomagok sorszámozása:**
 - Az adó sorszámozza a csomagokat.
 - Ha egy csomópont ugyanattól a feladótól ugyanolyan sorszámu csomagot kap, mint amelyet már korábban kapott (és az időzítés még nem járt le) ⇒ eldobja azt, mint másodpéldányt
 - **Szelektív elárasztás:**
 - A topológia ismeretében előre meghozza a forgalomirányítási döntéseket (nagyjából - ezért mondható táblázat nélkülinek) → és eszerint áraszt el.

Táblázat nélküli módszerek

Az „elárasztásos” (flooding) módszer:

- **Előny:**
 - Egyszerű, robosztus,
 - optimális késleltetés.
- **Hátrány:**
 - Rossz vonalkihasználtság.

A „véletlen séta” módszer:

- A bejövő csomagot valamely véletlen irányba továbbítja
- Rossz vonalkihasználtság, de egyszerű és robosztus

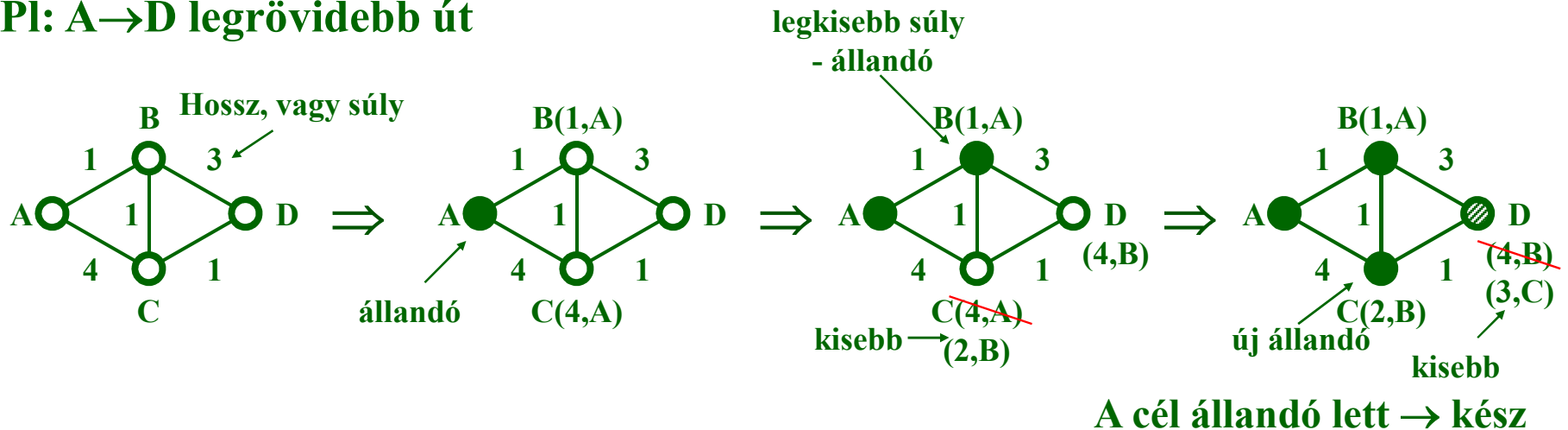
Információgyűjtési módszerek

- **Lehetnek:**
 - **Statikus, vagy dinamikus;**
 - **Centralizált, vagy elosztott módszerek**
- **Statikus forgalomirányítás**
 - **A hálózat üzemeltetője tölti ki a csomópontok táblázatait**
(Pl. X.25 nyilvános csomagkapcsolt hálózat)
 - **Bemeneti paraméterek (a kitöltő számára)**
 - a topológia és
 - egyéb szempontok (pl. költség, késleltetés, legrövidebb út stb.)

Statikus forgalomirányítás

- **Pl. Legrövidebb út algoritmus (Dijkstra 1959):**
 - Állandóvá tesszük a kiindulási állomást → (1)
 - (1) Ideiglenesen felcímkezzük a szomszédos állomásokat – az úthosszal és az állandó állomás címével
 - A legrövidebb úthossz címkéjűt állandóvá tesszük → (1)

Pl: A→D legrövidebb út



Adaptív centralizált forgalomirányítás

- **Működése:**
 - központ (nagy kapacitású) begyűjti az összes információt a csomópontokról (topológia, forgalmi irányok, terhelés)
 - ebből kiszámítja az optimális utakat és
 - letölti azokat a csomópontok tábláiba.
- **Előnye:** adaptív és optimális.
- **Hátrányai:**
 - Sebezhető → a központ hibáira védtelen
→ hiba esetén elveszítheti az adaptivitását, optimalitását.
 - A központ felé vezető utakat túlterhelhetik az információgyűjtés és tábla letöltés adatai.
 - Nagy mennyiségű szolgálati információ.
 - Esetenként instabil lehet (a késleltetések miatt)

Adaptív elszigetelt módszerek

- (Dinamikus, elosztott információgyűjtés)
- Ilyenek a táblázat nélküli módszerek és
- a „fordított tanulás” (backward learning) módszer

A fordított tanulás

- Kezdetben senki nem tud semmit $\Rightarrow$ elárasztás
- Minden csomagban ugrásszámláló, melyet minden csomópont amin áthalad, inkrementál





A fordított tanulás

- Ha egy állomás valamely vonalán csomagot kap j ugrásszámmal, akkor tudja, hogy a feladó című állomás legfeljebb j lépés távolságban van a vétel irányában
- A vett adatokat a táblázatában gyűjti („tanul”),
 - meghatározza, hogy melyik állomás melyik irányban érhető el a legkevesebb ugrásszámmal
- Időnként el kell „felejtene” a régi bejegyzéseket (hogy alkalmazkodhasson a változásokhoz)
- Előnyei:
 - nem igényel szolgálati kommunikációt, adaptív, robosztus.
- Hátrányok:
 - van fölösleges kommunikáció (kezdeti, majd időnkénti elárasztás),
 - számításigényes,
 - nem biztosít optimalitást (felejtés, elárasztás) (Pl. bridge).
- Megjegyzés: a fordított tanulás szelektív elárasztás
(a nagyjából jó irányok mentén áraszt)

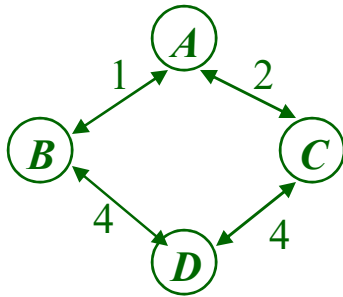
Elosztott forgalomirányítás

Működés:

- A szomszédok időnként átadják egymásnak aktuális tábláikat (ismereteiket a hálózatról)
- A táblázatok tartalmazzák
 - az egyes célok elérési irányait
 - az illető cél távolságának (ugrásszám, vagy elérési idő) becsült értékét
- A táblákat kapó állomás hozzáadja a távolság értékekhez a táblát küldő becsült távolságát és ez alapján frissíti a saját táblázatát.
- Előny:
 - közel lehet az optimumhoz, (Pl. IP routing)
 - adaptív, robosztus.
- Hátrány:
 - szolgálati kommunikáció
 - Számításigényes feldolgozás

Elosztott forgalomirányítás: Távolságvektor alapú forgalomirányítás (Bellman-Ford 1957, 1962)

A-t vizsgáljuk, aki *B*-től és *C*-től kap táblákat:



<i>B</i> táblája	
<i>A</i>	<i>A</i> :1
<i>C</i>	<i>A</i> :3
<i>D</i>	<i>D</i> :4

<i>C</i> táblája	
<i>A</i>	<i>A</i> :2
<i>B</i>	<i>A</i> :3
<i>D</i>	<i>D</i> :4

A „korrigálja” a kapott táblákat
a küldő távolságával:

+1

+2

Módosított	
<i>B</i>	<i>B</i> :1
<i>C</i>	<i>B</i> :4
<i>D</i>	<i>B</i> :5

Módosított	
<i>C</i>	<i>C</i> :2
<i>B</i>	<i>C</i> :5
<i>D</i>	<i>C</i> :6

A kiválasztja a legjobbakat:



A a legjobbakból új táblát készít:

<i>A</i> táblája	
<i>B</i>	<i>B</i> :1
<i>C</i>	<i>C</i> :2
<i>D</i>	<i>B</i> :5

Bellman-Ford algoritmus: count-to-infinity

A végtelenig számolás problémája (count-to-infinity):

- A jó hír (A megjavult) gyorsan,
- a rossz hír (A elromlott) lassan terjed
(a terjedés sebessége a ∞ ábrázolásától függ pl. 16)
- A probléma oka, hogy egy csomópont nem tudja eldönteni, hogy ő maga rajta van-e egy másik csomópont által javasolt úton.

A megjavult

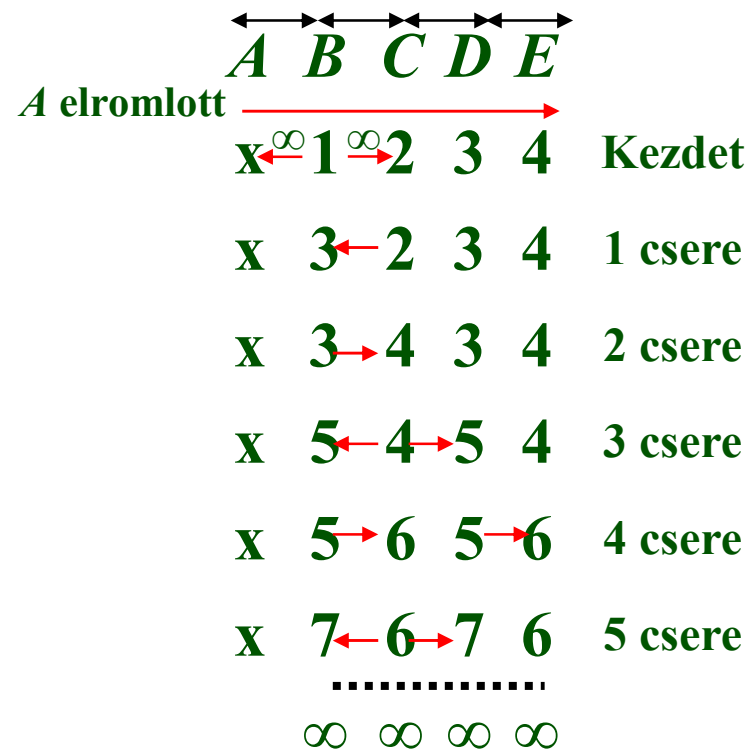
	$\longleftrightarrow$	$\longleftrightarrow$	$\longleftrightarrow$	$\longleftrightarrow$	
A	B	C	D	E	
0	∞	∞	∞	∞	Kezdet
0	$\rightarrow 1$	∞	∞	∞	1 csere
0	1	$\rightarrow 2$	∞	∞	2 csere
0	1	2	$\rightarrow 3$	∞	3 csere
0	1	2	3	$\rightarrow 4$	4 csere

A elromlott

	$\longleftrightarrow$	$\longleftrightarrow$	$\longleftrightarrow$	$\longleftrightarrow$	
A	B	C	D	E	
x	1	2	3	4	Kezdet
x	3	$\leftarrow 2$	3	4	1 csere
x	3	4	3	4	2 csere
x	5	$\leftarrow 4$	$\rightarrow 5$	4	3 csere
x	5	6	5	$\rightarrow 6$	4 csere
x	7	$\leftarrow 6$	$\rightarrow 7$	6	5 csere
					

Route poisoning: útvonal mérgezés

- Ha egy hálózat elérhetetlen, akkor a forgalomirányító megváltoztatja az arra az útvonalra vonatkozó mértéket végtelenre és ezt hirdeti.



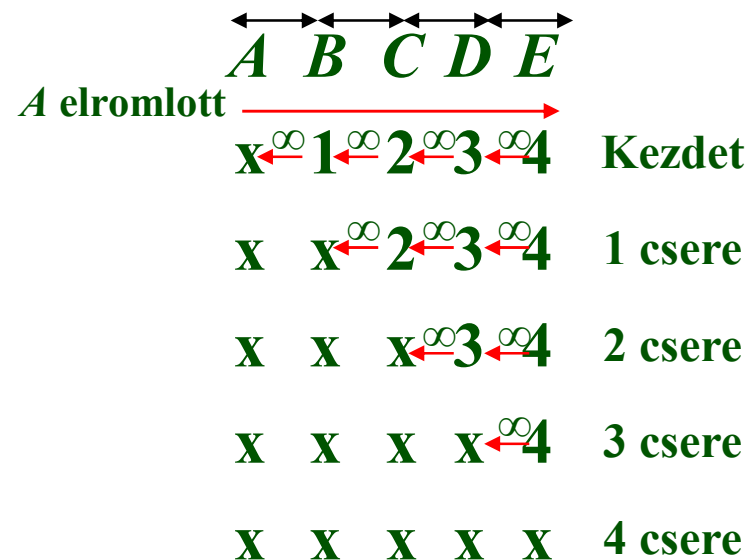
Split horizon: látóhatár megosztás

- Megakadályozza, hogy azon az interfészen hirdessen egy útvonalat, amelyiken megismerte azt.



Split-horizon routing with poison reverse

- Végtelen távolságot hirdet azon az interfészen, amelyiken megismerte az illető útvonalat.



Visszatartó időzítő

- Megakadályozzák egy leállt útvonalra vonatkozó frissítésnek az elfogadását, ha a frissítés a leállást követő meghatározott időintervallumon belül érkezik, és benne a korábbinál nagyobb mérték szerepel.
- Ha a időzítő lejárt a előtt az eredeti útvonal helyreáll, vagy olyan útvonal információt kap, mely kisebb mértékkel rendelkezik, akkor azonnal használni kezdi.

Csomagszórásos forgalomirányítás (broadcasting)

- Valamennyi állomásnak küld üzenetet.
 - Pl: osztott adatbázisok frissítése, ütemezési felhívás stb.

Lehetséges implementációi:

- Mindenkinek külön csomagot küld a forrás.
(lista kell a broadcasting-ban résztvevőkről)
- Több-célcsomópontos forgalomirányítás
(multi-destination routing).



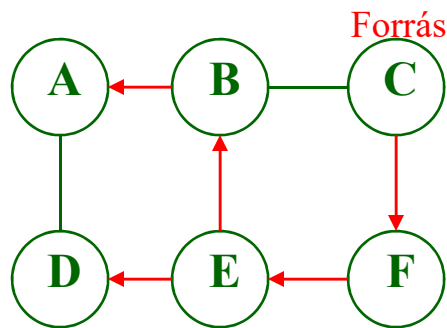
⇒ **Multi-Destination Routing**

- **A csomagban benne az összes célcím**
(lista v. bittérkép formában)
- **A csomópont vizsgálja az „összes célcím”**
struktúrát, hogy meghatározza a kimenő
vonala(ka)t.
- **Minden kimenő vonalra készít új csomagot,**
benne új „összes célcím” struktúrát hoz létre,
és ezeket küldi el.

Nyelőfát alkalmazó megoldások

- **Nyelőfa (sink tree):** azon optimális utak halmaza, melyek az összes forrásból egy adott célba vezetnek.
- **A csomópontok ismerik a forrás nyelőfáját és e mentén továbbítják a csomagokat.**
- **Megjegyzés:** Ha a $C \rightarrow A$ optimális úton rajta van B, akkor a $B \rightarrow A$ útvonal is optimális.
- **Megjegyzés: a nyelőfa feszítőfa is**
Feszítőfa (spanning tree): a hálózat olyan útjainak halmaza, mely hurkot nem tartalmaznak (egyszeresen összefüggő), de a hálózat valamennyi csomópontját érintik

Példa C nyelvű fájljára



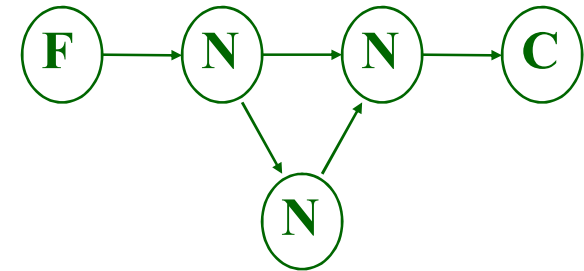
- C küldi F-nek
- F küldi E-nek
- E küldi B-nek és D-nek
- B küldi A-nak

C nyelvű fájljában: ha $A \rightarrow C$ optimális út,
és az optimális úton rajta van B,
akkor $B \rightarrow C$ is optimális

A hálózati réteg funkciói

- **Forgalomirányítás**

- a csomag célbajuttatása.
- ismerni kell a topológiát
- terhelésmegosztás (alternatív utak)



- **Torlódásvezérlés**

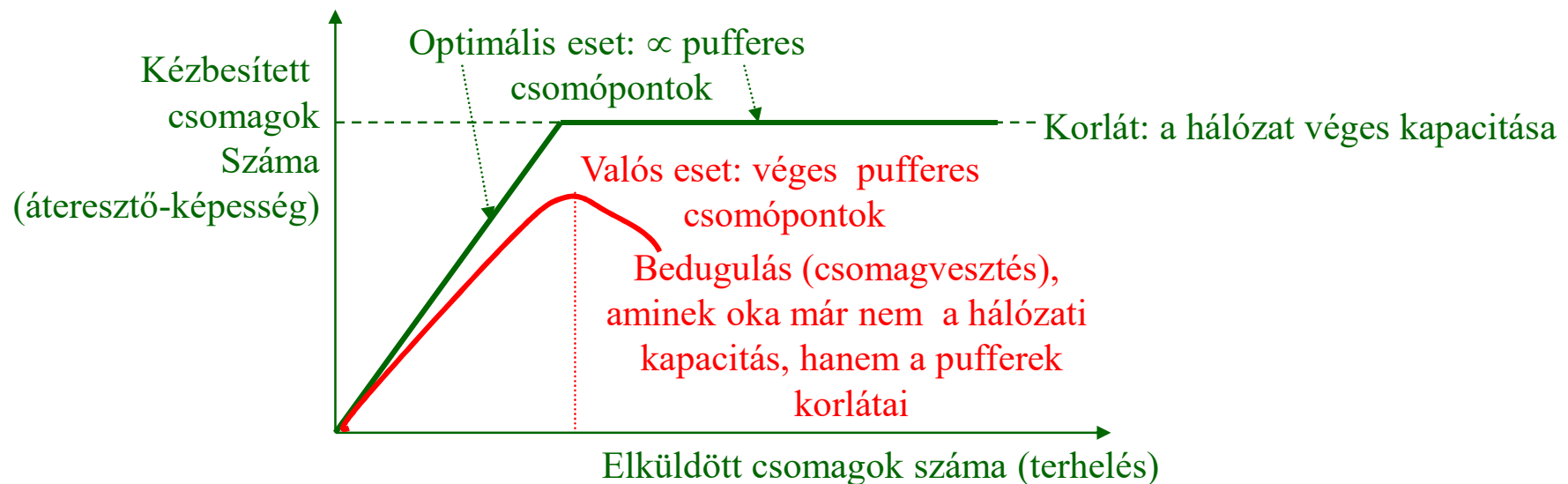
- Ne legyenek a hálózat egyes részei túlterheltek
- Hasonló a forgalomszabályozáshoz, de ez nem csak két pont (adó-vevő) közötti, hanem a hálózat egészére vonatkozik.

- **Hálózatközi együttműködés**

- Ez az első réteg, ahol különböző hálózatok összekapcsolhatók (heterogén hálózatok kialakítása)

A torlódásvezérlés célja

- Megelőzze és/vagy elhárítsa azokat a szituációkat, melyekben egy összeköttetés vagy egy csomópont túlterheltté válik.



Torlódásvezérlő algoritmusok

- Pufferek előrefoglalása
- Csomageldobás
(különböző eldobási szempontokkal)
- Lefojtó-csomagok módszere
- Izaritmikus torlódásvezérlés
- Forgalomszabályozásos torlódásvezérlés

Pufferek előrefoglalása

- Minden átvinni kívánt csomagnak előre foglal puffert
- Virtuális áramkör alapú hálózatokban használható, a hívásfelépítés során rendel a virtuális áramkörhöz puffert területet - a hívásfelépítő csomag nemcsak táblabejegyzéseket generál, hanem puffereket is foglal.
- Elutasítás lehetséges, amennyiben nem áll rendelkezésre az igényelt erőforrás
Pl. az adóablak méretnek megfelelő puffert foglal le
- Hátrány:
nem gazdaságos (fölszabott pufferkapacitást foglal le)
(sőt elutasíthat emiatt más hívásfelépítést).
Megoldás pl: a sokáig tétlen puffereket felszabadítja.
(ez kockázatos lehet)

Csomageldobás

First-Come-First-Served módon használja a puffereket $\Rightarrow$ ha betelnek, eldobja az újabbakat

(Nem foglalnak előre puffereket, de valamekkora minden vonalon van.)

(Más protokollok biztosíthatják az újraadást.)

Módosításai:

- **A bemenetekre legalább 1db puffert le kell foglalni és szabadon kell hagyni (ne váljon süketté)**
Pl: szolgálati üzenetek: ha egyéb, eldobja, ha speciális üzenet $\Rightarrow$ feldolgozza $\Rightarrow$ nem lesz „süket”.



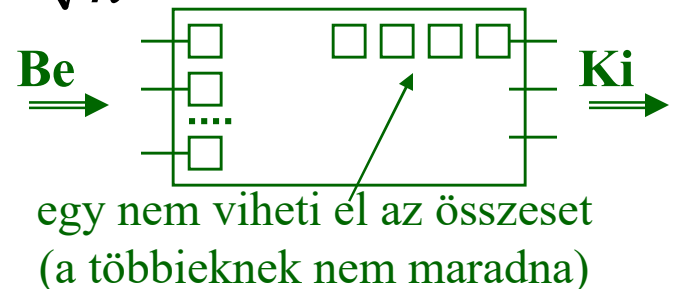


Csomageldobás

Módosításai:

- **Pufferek felosztása a kimenetek között**
 - korlátozza a kimeneti puffer-sorok maximális hosszait
 - ugyanakkor minimális pufferszámot is előír a kimenetekre (a „kiékezés” ellen)
 - A max pufferszám a forgalom függvénye.
 - Pl: „Ökölszabály” a kimeneti pufferek max számára :
ahol
 - p : a pufferek össz száma;
 - k : a kimenetek száma;
 - m : a max hossz egy kimenetre.

$$m = \frac{p}{\sqrt{k}}$$



Csomageldobás

Eldobási szempontok (prioritások)

- pl. prioritási osztályok szerint, vagy
- pl. az ugrások számát nézik, és azt dobják el, amelyik kevesebbet utazott
(valószínűleg kisebb erőforrás-igényű ennek a megismétlése)

Lefojtó-csomagok módszere

Choke packets:

- A források fojtása (még mielőtt a torlódás beállna)
- Csomópont figyeli a kimenő vonalainak telítettségét, és ha az egy küszöbértéket túllép $\Rightarrow$ **lefojtó-csomagot küld a feladónak** (ebbe az irányba csökkentsék a forgalmat jelentéssel), de az eredeti csomagot továbbítja.
Az eredeti (továbbított) csomagot meg is jelölheti:
 $\Rightarrow$ ez a csomag már váltott ki fojtóüzenetet (a rákövetkező csomópontoknak már nem kell küldenie)

Lefojtó-csomagok módszere

- **A küldő adaptivitása:**
 - az első lefojtó-csomag vétele után csökkenti a forgalmát, majd
 - egy időzítés ideig nem fogad újabb fojtó csomagot (nem csökkenti tovább forgalmát) (duplikált fojtások lehetnek – pl. ugyanazon szekvencia több eleme).
 - Ennek leteltével újabb időzítés:
 - ha ezalatt újabb fojtócsomag érkezik
⇒ tovább csökkenti forgalmát,
 - ha nem érkezik
⇒ visszanoveli az adott célirányú forgalmát.

Izaritmikus torlódásvezérlés

- Korlátozza a hálózatban egyidejűleg bentlévő csomagok számát
- Engedélyező csomagokat (permit) használ
 - ⇒ csak akkor adhat, ha engedélyező csomagot kapott
 - ⇒ majd utána újabb engedélyező csomagot generál
- Az engedélyező csomagok körbejárnak a hálózaton.
- Módosítása:
Engedélyező központ, akitől lehet engedélyt kérni.
 - szolgálati overhead-del jár (bár nem naggyal), és
 - érzékeny a központ kiesésére
- Gond:
Engedélyező csomagok megsemmisülése
(Nehezen menedzselhető (pótolható))
 - kivéve az engedélyező központos megoldást.

Forgalomszabályozásos torlódásvezérlés

- **A forgalomszabályozás:**
 - adó ne árasszon el vevőt (két állomás viszonyára)
- **A fogadóállomások a forgalomszabályozást nem a kapacitásuk függvényében alkalmazzák, hanem valamilyen abszolút korlátozást vesznek figyelembe**
⇒ esetleg az adóra is telepíthető, így közvetlen a kibocsátásnál érvényesíthető.
- **Ha a korlátok megfelelőek ⇒ biztos, hogy jó**
→ ha egy kicsit nagyobbak ⇒ lehet, hogy nem egyenletes terhelés esetén egyes pontokon torlódások alakulnak ki.
- **Gond: alacsony korlátok esetén ⇒ nagy késleltetések.**

A hálózati réteg funkciói

- **Forgalomirányítás**
 - a csomag célba juttatása.
 - ismerni kell a topológiát
 - terhelésmegosztás (alternatív utak)
- **Torlódásvezérlés**
 - Ne legyenek a hálózat egyes részei túlterheltek
 - Hasonló a forgalomszabályozáshoz, de ez nem csak két pont (adó-vevő) közötti, hanem a hálózat egészére vonatkozik.
- **Hálózatközi együttműködés**
 - Ez az első réteg, ahol különböző hálózatok összekapcsolhatók (heterogén hálózatok kialakítása)