

Hálózatok II.

A hálózati réteg forgalomirányítása

2007/2008. tanév, I. félév

Dr. Kovács Szilveszter

E-mail: szkovacs@iit.uni-miskolc.hu

Miskolci Egyetem

Informatikai Intézet 106. sz. szoba

Tel: (46) 565-111 / 21-06 mellék

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
 - Datagramm 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 közeget határok, 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).

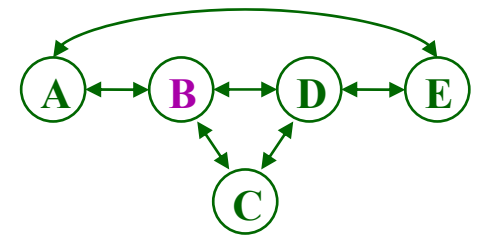
Többutas forgalomirányítás

- 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 ők.
 - **Csomagok sorszámozása:**
 - Az adó sorszámozza a csomagokat.
 - Ha egy csomópont ugyanattól a feladótól ugyanolyan sorszámú 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

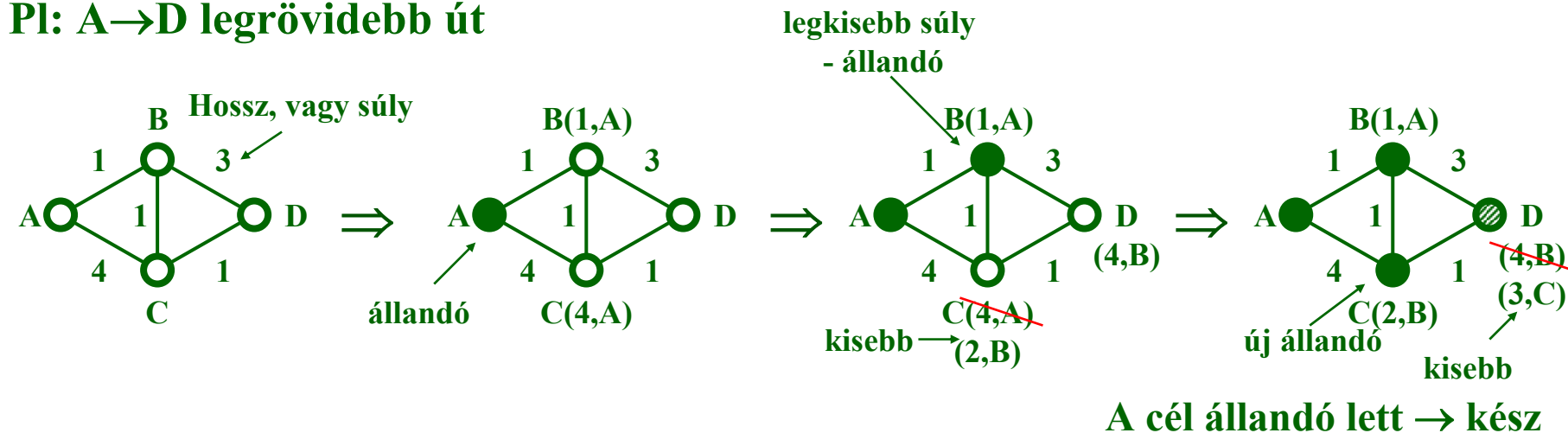
Információgyűjtési módszerek

- **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 $\rightarrow (1)$
 - (1) Ideiglenesen felcímkezzük a szomszédos állomásokat – az úthosszal és az állandó állomás címével
 - A legrövidebb úthossz címkejűt állandóvá tesszük $\rightarrow (1)$

Pl: A \rightarrow 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 táblázatnélküli 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)

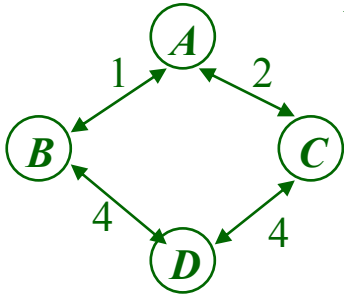
Elosztott forgalomirányítás

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: Tavolsagvektor alapu forgalomirányítás (Bellman-Ford 1957, 1962)

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



<i>B</i> táblája	
A	A:1
C	A:3
D	D:4

<i>C</i> táblája	
A	A:2
B	A:3
D	D:4

A „korigálja” a kapott táblákat a küldő távolságával:

+1

+2

Módosított	
B	B:1
C	B:4
D	B:5

Módosított	
C	C:2
B	C:5
D	C:6

A kiválasztja a legjobbkat:




A a legjobbakból új táblát készít:


<i>A</i> táblája	
B	B:1
C	C:2
D	B: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.

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	
<i>A</i> megjavult						
	0	∞	∞	∞	∞	Kezdet
	0	→1	∞	∞	∞	1 csere
	0	1	→2	∞	∞	2 csere
	0	1	2	→3	∞	3 csere
	0	1	2	3	→4	4 csere

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	
<i>A</i> elromlott						
	x	1	2	3	4	Kezdet
	x	3	←2	3	4	1 csere
	x	3	→4	3	4	2 csere
	x	5	←4	→5	4	3 csere
	x	5	→6	5	→6	4 csere
	x	7	←6	→7	6	5 csere
					

Csomagszórásos forgalomirányítás (broadcasting)

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

Lehetséges implementációi:

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





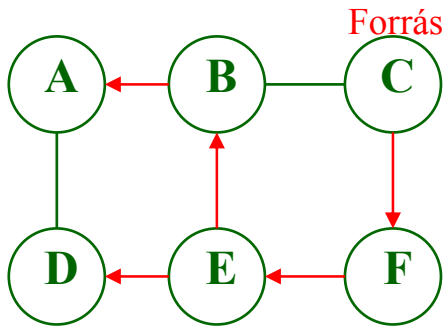
Multi-Destination Routing

- **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 nyelőfájára



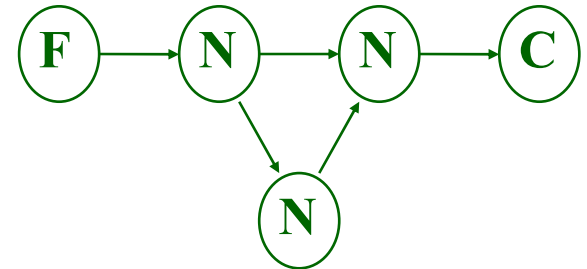
- C küldi F-nek
- F küldi E-nek
- E küldi B-nek és D-nek
- B küldi A-nak

C nyelőfáján: 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)