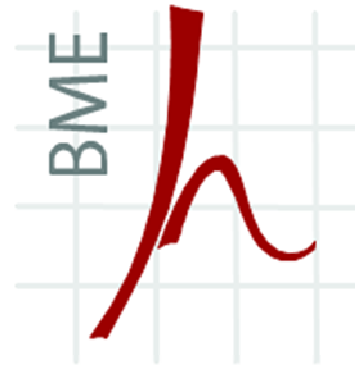




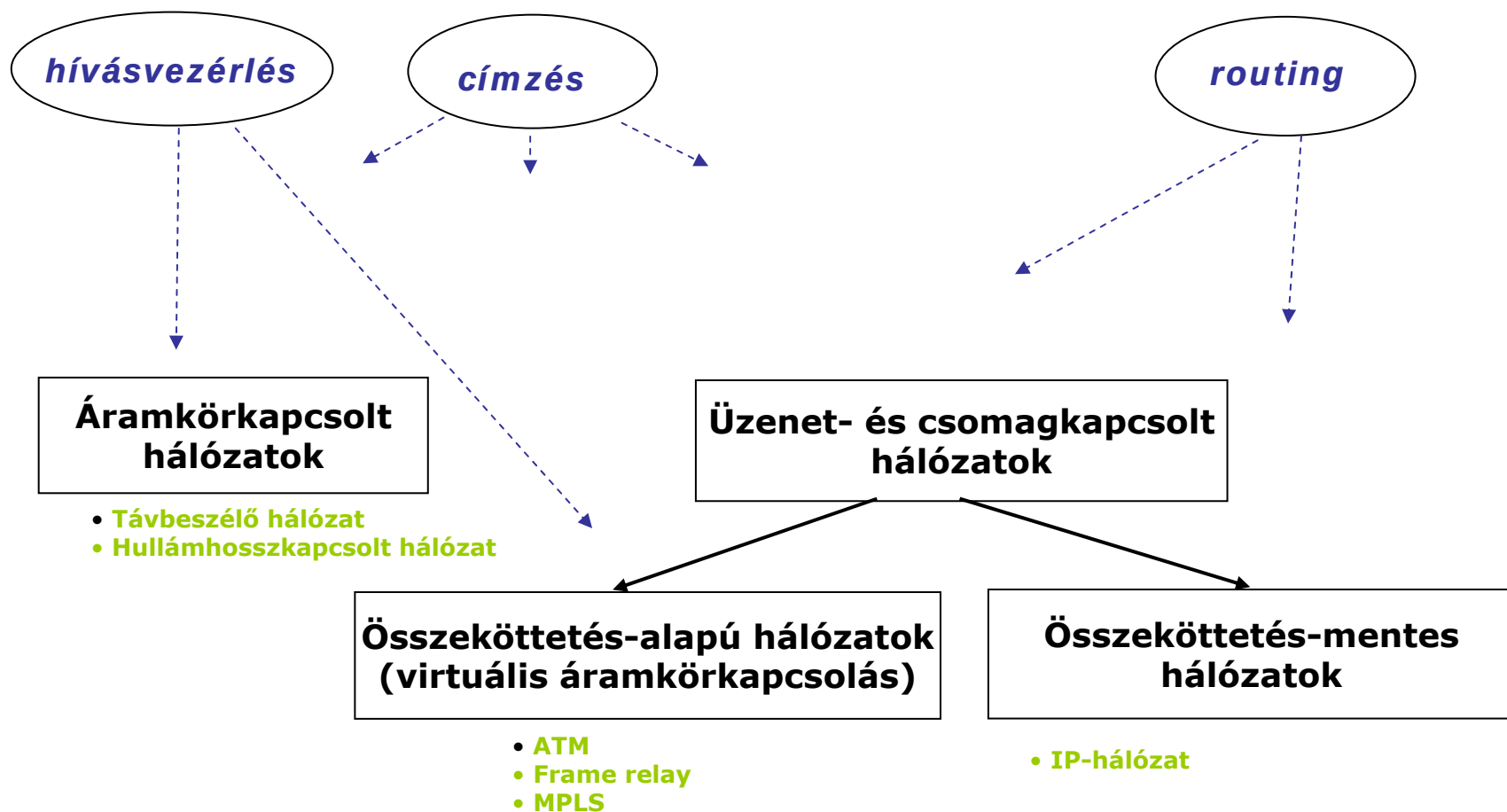
ROUTING

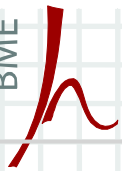
Útvonalválasztás

Dr. Simon Vilmos
adjunktus

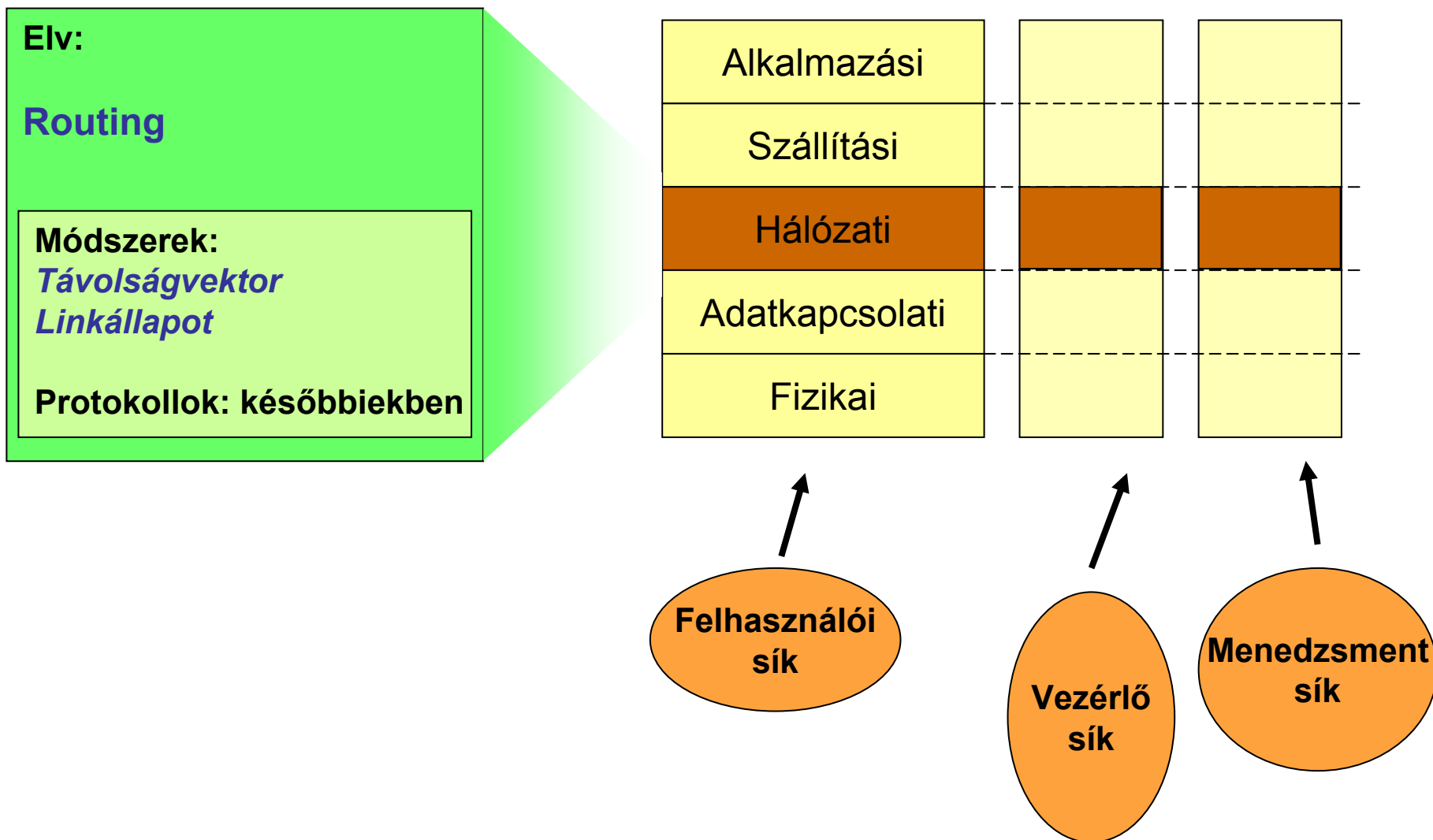
BME Hálózati Rendszerek és Szolgáltatások Tanszék
svilmos@hit.bme.hu

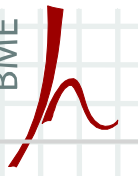
További nagyon fontos funkció a kapcsolt hálózatok működtetéséhez





Elv, módszer, protokoll





Az elnevezésről...

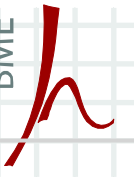
.fr Routage

.de Routing

.ru Маршрутизация (marshrutizatsiya)

A Tanenbaum: Computer Networks könyv magyar fordításában: „forgalomirányítás”

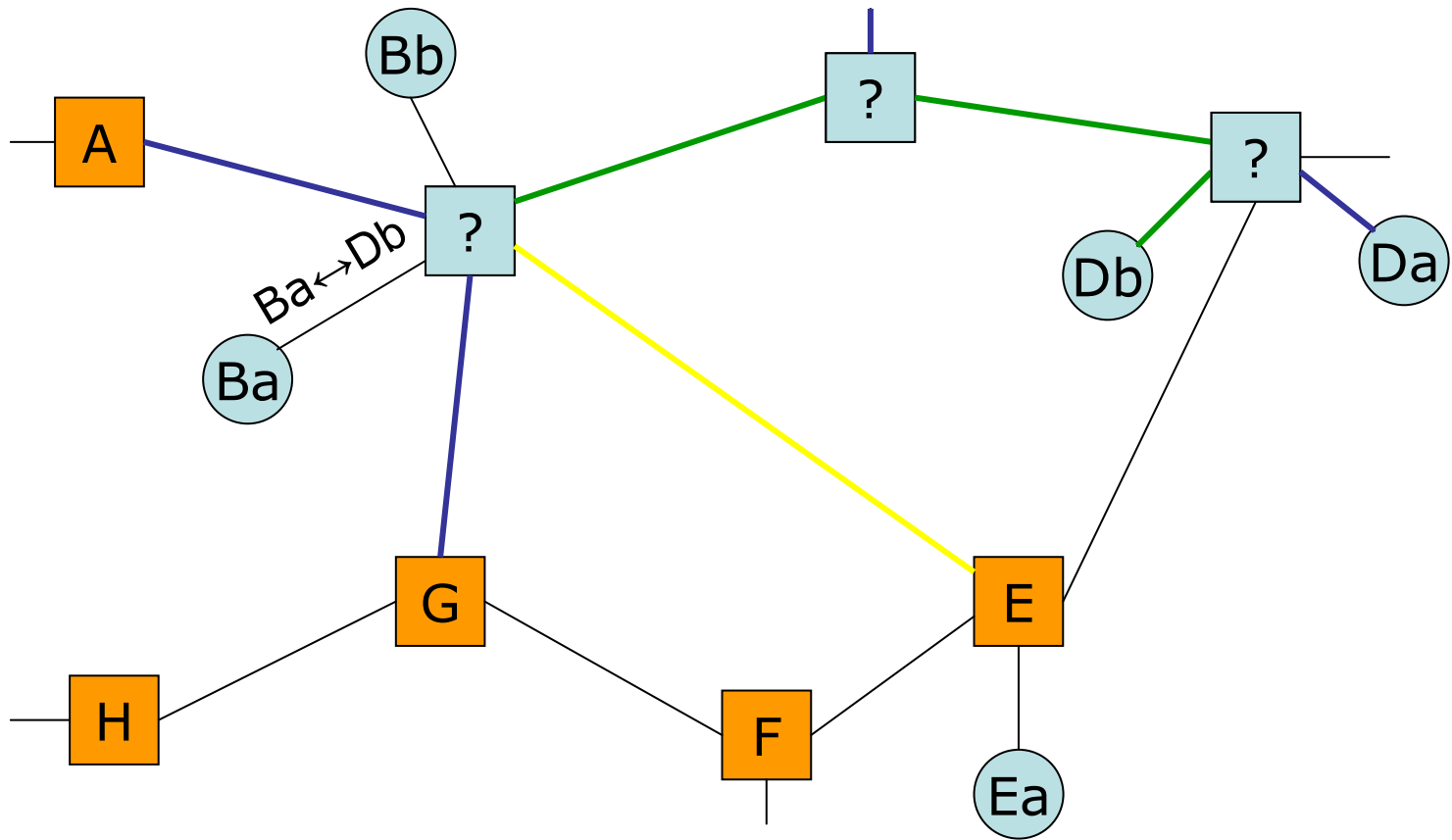
Nálunk (általában): routing

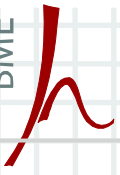


Mi a routing?

- *Az a mechanizmus, mely segítségével a szállítandó információ a **megfelelő úton** kerül továbbításra a végpontok között*
 - *szűkebb értelemben: csak az útvonalválasztás, útvonalkijelölés*
 - *tágabb értelemben: ideértjük a csomópontokban a csomagok **továbbítását** is*
- A feladat érdekessége:
 - Általában **nem állandó** a hálózat felépítése: adaptív módon
 - Gyakran „**nagy**” a **hálózat mérete**: nincs pontos és aktuális információ a hálózat állapotáról.
- Ez a feladat egyaránt felmerül az összeköttetés-alapú és –mentes hálózatokban
 - ö. a.: útvonalkijelölés
 - ö. m.: útvonalválasztás
- Ide tartoznak az útvonalválasztó **módszerek, algoritmusok** és az azokat megvalósító **protokollok**

Bevezetés: a feladat értelmezése

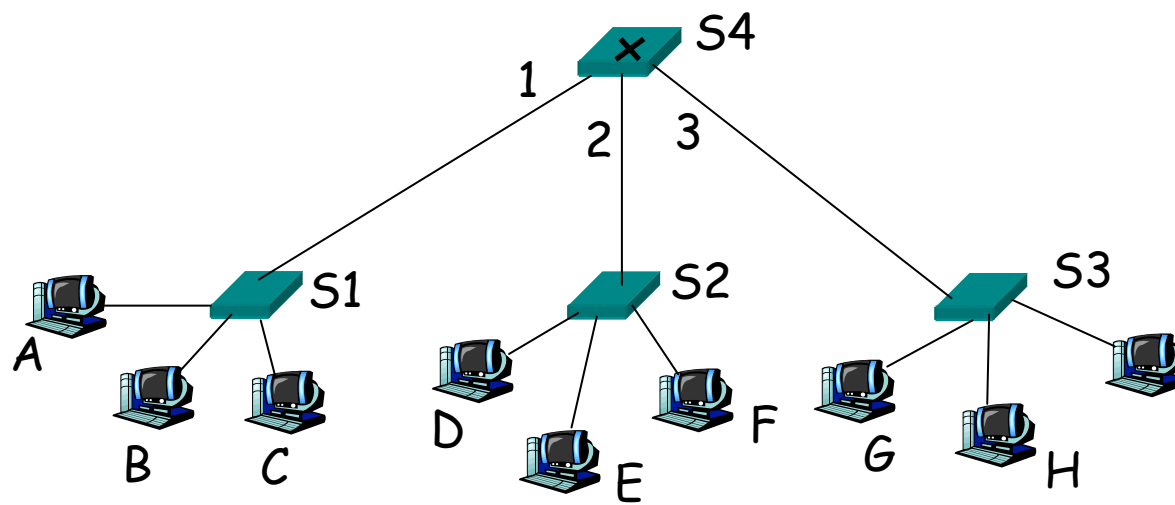




Különbség bridging és routing között

- Bridging (bridge vagy switch által végezve):
 - Nem tudja hol van pontosan az adott cím
 - A fogadott keretek forráscíme alapján bejegyzés a táblában:
 - MAC Address, Interface, Time Stamp
 - régi bejegyzéseket eldobja
 - Megtanulja, hogy melyik végpontokat melyik interfészen keresztül tudja elérni

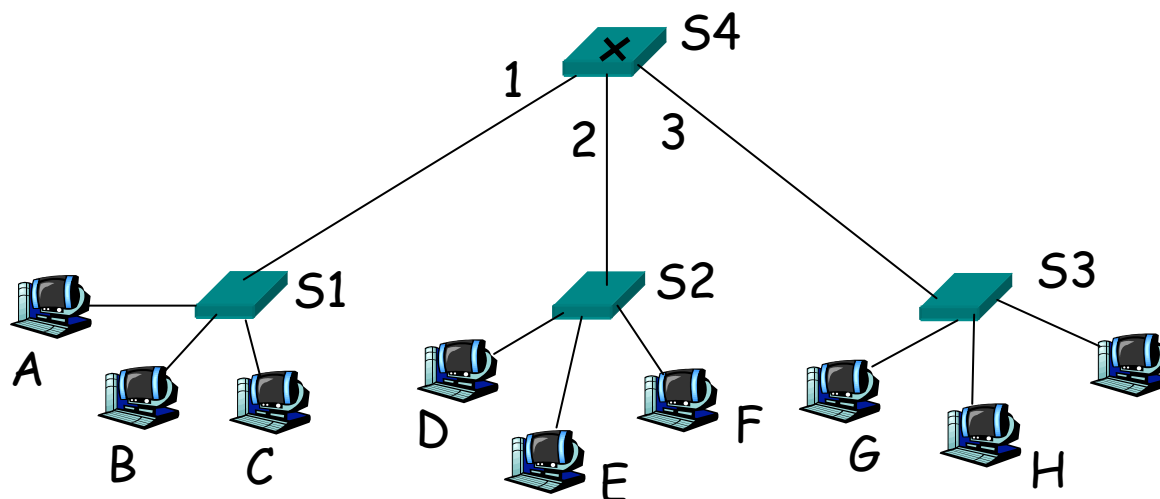
Példa bridging-re (LAN előadásról)



address	interface
A	1
B	1
E	2
G	3
I	

- A kapcsoló vesz egy C-től D-nek menő keretet
 - bejegyzi a táblába, hogy C az 1-es interfészen érhető el
 - mivel D nincs benne a táblában, továbbítja a keretet a 2 és 3 interfészre
- D veszi a keretet

Példa bridging-re (folyt.)



address	interface
A	1
B	1
C	1
E	2
G	3
I	

- A kapcsoló veszi a keretet D-től
 - bejegyzi a táblába, hogy D a 2-es interfészen érhető el
 - mivel C benne van a táblában, továbbítja a keretet az 1-es interfészre (csak oda)
- C veszi a keretet

Bridging fajtái

- **Transzparens (vagy tanuló) bridging:** előző példában bemutatott
 - pl. Ethernetnél
- **Forrás által vezérelt bridging + transzparens**
 - A küldött csomag fejlécében fel vannak sorolva az útvonal bridge-i és szegmensei, amit érinteni kell
 - Vezérjeles gyűrűnél használatos
 - Első küldés esetén:
 - **Broadcast csomag** minden irányban, feljegyzik az útvonalat sorban, visszaküldik
 - Ahány lehetséges útvonal, annyi csomag jön vissza, pl. **legrövidebbet** választja
 - **Maximális hopszám** után eldobás (hurok elkerülése)
 - **Forgalom kiegyenlítés:** leterhelt eszközök nem vesznek részt benne

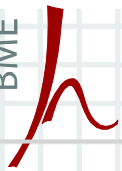
Különbség bridging és routing között

- Bridging:
 - Adatkapcsolati rétegben MAC címek alapján
 - Nem tud különbséget tenni hálózatok között
 - Elárasztást használ: csak helyi hálózatokban

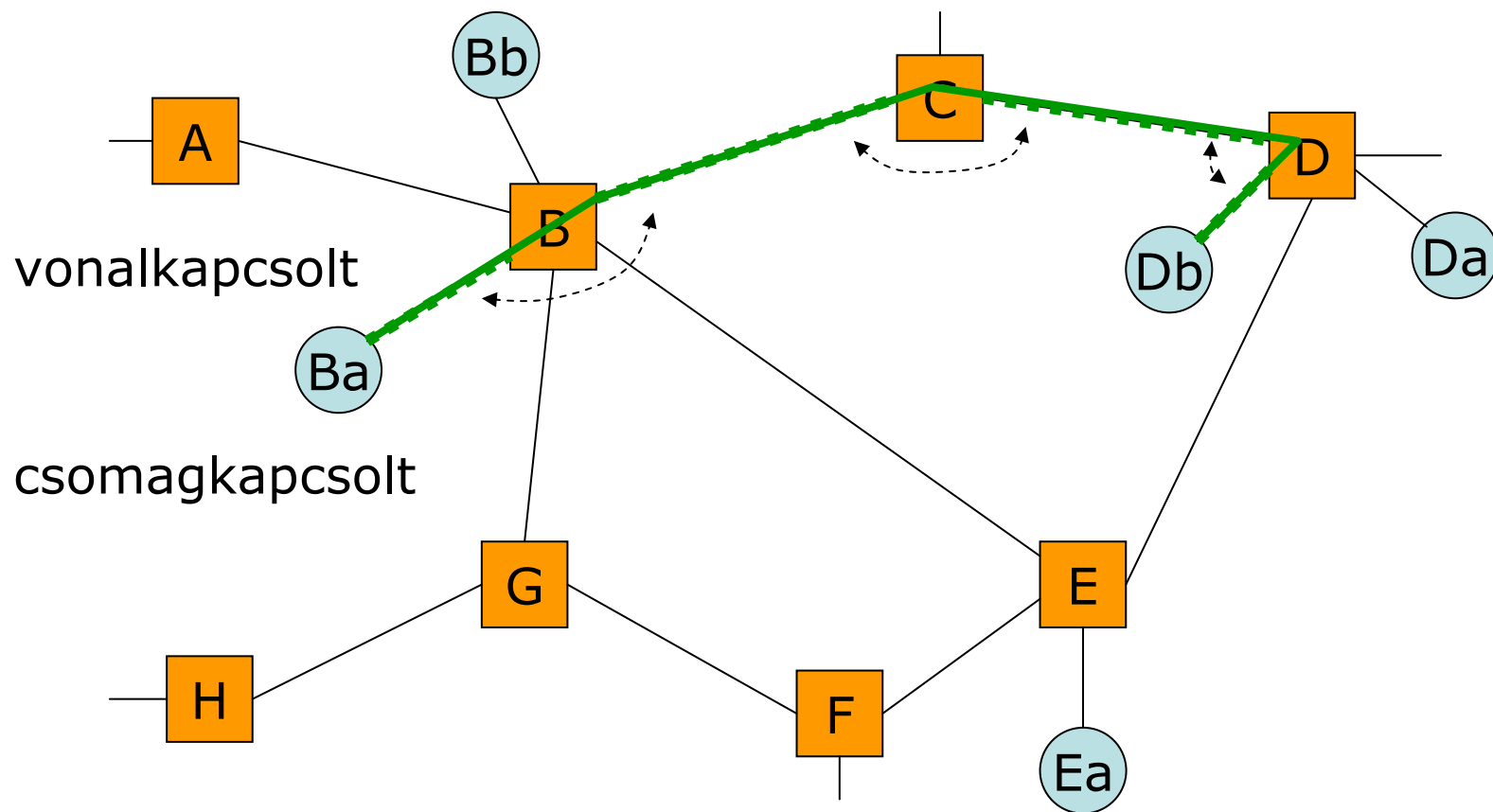
- Routing:
 - Hálózati rétegben, IP címek alapján
 - Skálázhatóbb megoldás
 - Erről lesz szó a továbbiakban!

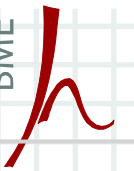
Mi tehát a feladat?

- **Megismerni, nyilvántartani** a hálózat felépítését:
 - csomópontok
 - összekötések (link)
 - lehetséges útvonalak
- **Összeköttetés-alapú eset:**
 - Csomópontokban felhalmozott ismeretek alapján
 - az útvonal kiválasztása lépésről-lépésre
 - **feljegyzése** az érintett csomópontokban
 - értesítés a kommunikáló végekhez, hogy elkészült az útvonal, megkezdhető a kommunikáció.
- **Összeköttetés-mentes:**
 - nincs külön útvonalkialakítás előre
 - a csomópontok a megfelelő útvonalat a csomag továbbításával **egyidejűleg** választják meg



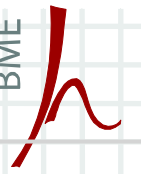
Összeköttetés-alapú /-mentes





Mi tehát a feladat?

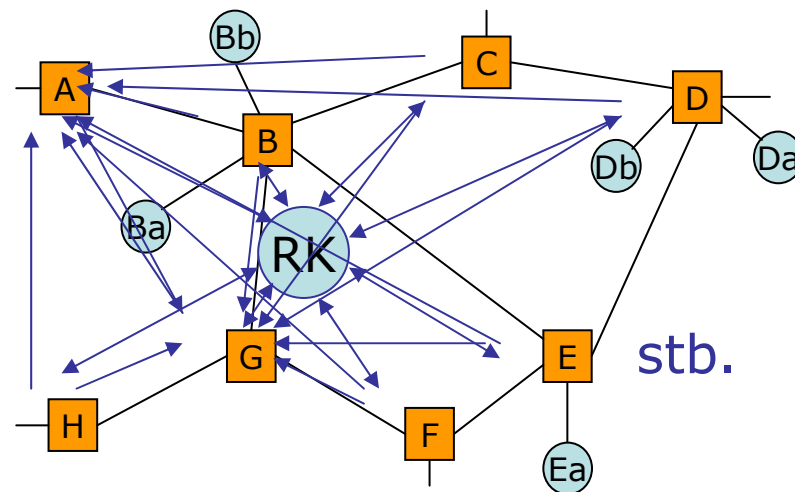
- **Áramkörkapcsolt eset:**
 - **állandó átviteli csatorna** létrehozása **az összeköttetés időtartamára**
- **Összeköttetés-alapú eset**
 - virtuális csatorna létrehozása a **csomópontokban történt feljegyzésekkel**
 - honnan hová kell tenni a csomagot
- **Összeköttetés-mentes eset:**
 - a csomópontok **minden egyes csomagra újból elvégzik az útválasztást**
 - **nincsenek állandó feljegyzéseik**
- *Utóbbi esettel foglalkozunk, a teljesség kedvéért megemlítve az előző kettőt*
- **Mi kell az útválasztáshoz?**
 - A hálózat felépítésének (csomópontjainak és linkjeinek) megismerése és nyilvántartása
 - Ezen ismeretek biztosítása a csomópontok számára



- Útvonalkijelölés ill. útvonalválasztás:
 - a csomópontok táblázatai alapján

Cél-végpont	következő csomópont
...	...

- A táblák kitöltése:
 - manuálisan
 - automatikusan:
 - centralizáltan
 - elosztottan



■ Centralizált:

- egy ponton összegyűjtjük a hálózatról megszerezhető valamennyi ismeretet
- meghatározzuk az egyes csomópontok útvonaltábláinak bejegyzéseit
- továbbítjuk azokat a csomópontokhoz.

■ Elosztott esetben

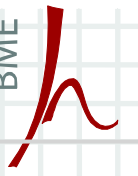
- valamennyi csomópont egyénileg gyűjt információt a hálózatról
- saját útvonaltábláját készíti el

Centralizált előnyei-hátrányai

+

- egyetlen kép a hálózatról, minden útvonal-tábla „konzisztens” lesz
 - De nem valós helyzetre!

- nagy hálózathál közben megváltozhat az állapota
- sérülékenység („*single point of failure*”)
 - Megoldás lehet: tartalékolás



Elosztott előnyei-hátrányai

+

- Kevésbé sérülékeny

-

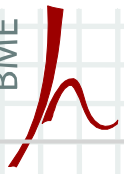
- Nincs egységes kép a hálózatról
- Eltérő „nézőpontok” a csomópontokban

- **Statikus routing**

- Előre becsült forgalmi viszonyok alapján útvonaltáblák centralizált kialakítása és szétosztása
- A hagyományos telefonhálózatokra alkalmazható
 - mivel itt a forgalom jól előrejelezhető
 - a hálózat amúgy is központilag menedzsel

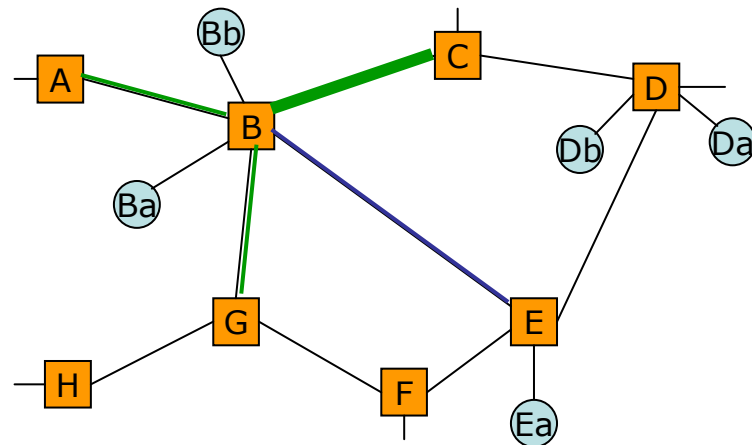
- **Dinamikus, adaptív routing**

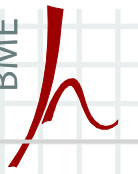
- Az aktuális forgalmi helyzet állandó figyelése és az annak legjobban megfelelő útvonaltáblák kialakítása
- Adathálózatokra jellemző
 - az Internet esetén elosztott változatban
 - a forgalom nem jósolható jól előre
 - nincs egységes hálózatfelügyelet



A gyűjthető információ

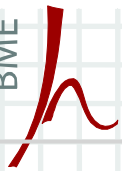
- Csomópontok közötti összeköttetések, linkek:
 - pl.: $B-A$, $B-C$, $B-E$, $B-G$
 - nem működik: pl.: $B-E$
 - $B \rightarrow Ba, - \rightarrow Bb$
- Linkek jellemzői:
 - átv. seb., pl.
 - $B-A$: 2 Mbit/s
 - $B-C$: 10 Mbit/s
 - $B-E$: 2 Mbit/s
 - Forgalom (sorhossz):
 - $B-A$: 2
 - $B-C$: 0
 - Terhelés %-ban
 - Kiszolgálási díj





Információ -> routingtáblák

- Ha a hálózat valamennyi csomópontjáról megvan az információ: meghatározható, hogy **bármely két felhasználót milyen útvonalon célszerű összekötni**.
- Célszerű: pl. legrövidebb
- A célszerű utakból
 - **kiolvashatjuk az egyes csomópontok által kialakítandó útvonal darabot**
 - meghatározhatjuk az **útvonaltáblákat**, amelyek bejegyzései lényegében útvonaldarabokat képviselnek.

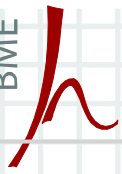


A routing módszerrel szembeni követelmények

- ***Minél kisebb méretű útvonaltábla:***
 - Kisebb tár, olcsóbb csomópont
 - Gyorsabb keresés a táblában, gyorsabban működő csomópont
 - Kisebb routingforgalom

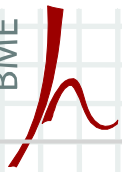
- ***Robosztusság:***
 - Hibás tábla esélyének minimalizálása, mert a hibás tábla okozhat:
 - „fekete lyukat”
 - hurkot
 - oszcillációt

- ***Optimális útvonalak kijelölése:***
 - az út optimalitása az igénytől függő, lehet optimális a
 - legrövidebb (*miben mérjük?*)
 - legmegbízhatóbb
 - legolcsóbb



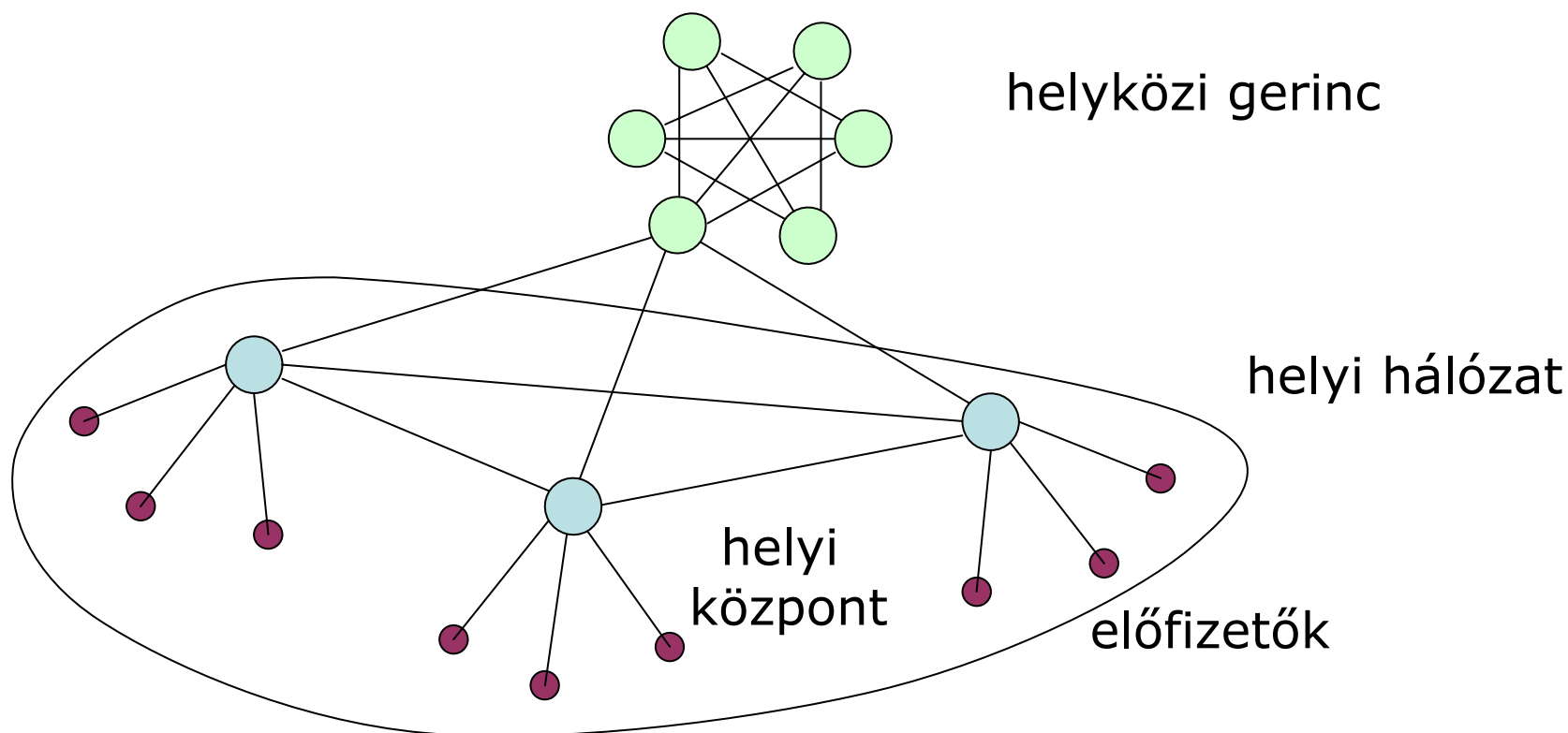
A megoldások csoportosítása

- Centralizált
- Tényleges forgalomtól függő
- Egyutas
- Lépésenkénti
- *Elosztott*
- *Forgalomfüggetlen*
- *Többutas*
- *Forrás általi*



Routing a telefonhálózatokban

- A telefonhálózat leegyszerűsített struktúrája



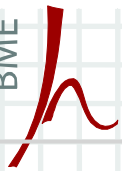
▪ Útvonalkijelölés

- helyi hívások: nincs útvonal
- a helyi központok szintjén közvetlen útvonal, vagy kétlépéses egy helyközin keresztül
- távolsági hívás
 - mindkét helyi központ ugyanahhoz a távolságihoz van bekötve
 - másik távolsági központon keresztül

▪ Az alkalmazott routing:

- centralizált táblakitöltés, a becsült forgalom függvényében
- két útvonal a távolsági központ számára:
 - közvetlen a hívott távolsági központjával
 - közvetett utak
 - **tartaléknak, ha a közvetlen út trónkvonalai foglaltak**, ezért a hívást blokkolni kellene

- „Routing az összekötöttség – **connectivity** – maximalizálására”: katonai célok eleinte!
- Legyen a módszer **elosztott**!
 - Az útvonaltáblák kialakítását végezzék maguk a csomópontok
- Két alapvető **módszer**:
 - „distance-vector” – *Bellman-Ford algoritmus*
 - „link-state” – *Dijkstra algoritmus*
- A módszerek különböznek abban, hogy:
 - milyen információt gyűjtenek
 - hogyan gyűjtik az információt
 - hogyan terítik ezt később



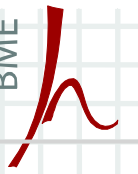
A módszerek lényege

- Információgyűjtés, és -terítés történik
- „Kinek, mit mondunk?”:
 - „Távolságvektor” módszer (*distance-vector*):

A csomópontok elmondják a **hálózatról alkotott elképzeléseiket**
a **szomszédaiknak**

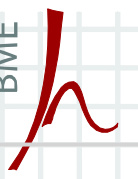
- „Linkállapot” módszer (*link-state*):

A csomópontok elmondják **mindenkinek** a **szomszédaikról nyert tapasztalataikat**



A „távolságvektor” módszer

- A gyűjtött információ és a gyűjtés módja:
A csomópontok elmondják a **hálózatról alkotott elképzeléseiket** a **szomszédaiknak**
- Az „elképzelések”:
 - melyik csomópont milyen távol van
 - egy lista (vektor) melynek elemei
 - csomópontazonosító – távolság párok
- A távolságvektorát közli mindegyik csomópont valamennyi szomszédjával
- A távolság mérése?



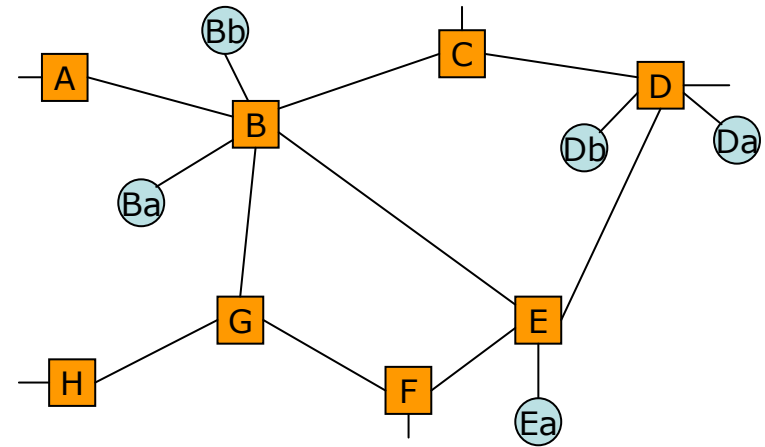
A „távolságvektor” módszer: a távolság mérése

- Legegyszerűbb esetben ez lehet a **lépések** (*ugrások, hop-ok*) száma
- Súlyozás:
 - *a link átviteli sebességével*
 - *a sorbanállási hosszal*
 - *a költséggel*
- Szóhasználat: *távolság = költség*
- Így a távolságvektor elemei **az adott csomópont által elképzelt költségek a hálózat többi (ismert) csomópontjához**

A Bellman – Ford algoritmus (1)

■ Példa:

A	B,1				
B	A,1	C,1	E,1	G,1	∞
C	B,1	D,1			
D	C,1	E,1			
E	B,1	D,1	F,1		



B-hez megérkezik
C üzenete:

B	A,1	C,1	D,2	E,1	G,1	∞
C	B,1	D,1				

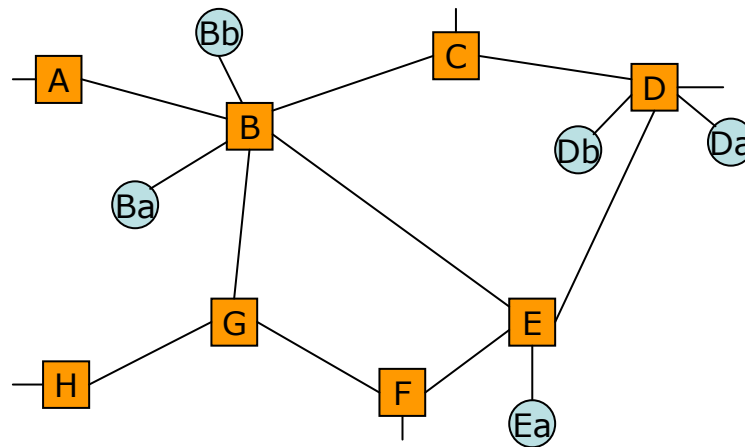
B-hez megérkezik
E üzenete:

B	A,1	C,1	D,2	E,1	F,2	G,1	∞
E	B,1	D,1	F,1				

(∞ - a hálózat még nem látott része)

A Bellman – Ford algoritmus (2)

- Ha a csomópont a beérkező üzenetből kideríti, hogy **rövidebb útra van lehetőség**, akkor **végrehajt egy cserét a vektorban**.
- Példa: *D* üzenete eljutott *E*-hez, majd *F*-hez, majd *G*-hez, és innen *B*-hez.
 - *F* táblázatába bekerült *D* két lépéssel,
 - *G* táblázatába így *D* már 3 lépéssel szerepel,
 - ezért *B* úgy látja, hogy *D* a *G*-n keresztül 4 lépéssel érhető el.
- Ha ezután érkezik hozzá *E* vagy *C* üzenete, amelyekben *D* egy lépéssel szerepel, akkor nyilván lecseréli a bejegyzést, mert $2 < 4$.
- Bebizonyítható, hogy a fenti algoritmus konvergens, de vannak tranziensei



Az útvonaltábla kialakítása a távolságvektor alapján

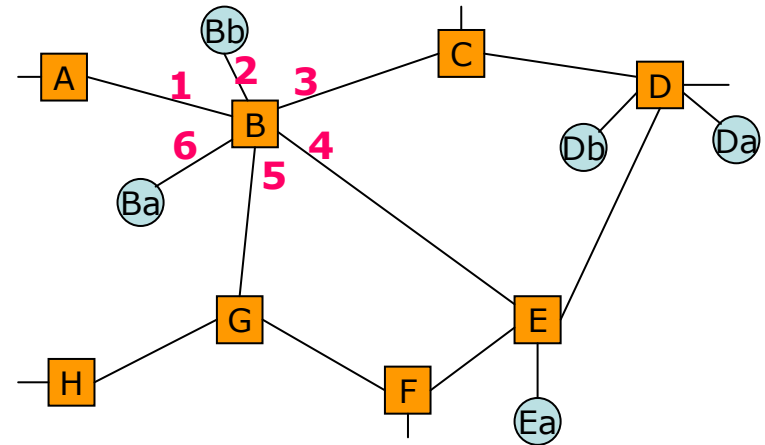
■ Például:

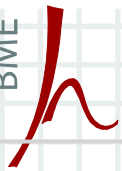
- B távolságvektora

$B \mid A,1 \mid C,1 \mid D,2 \mid E,1 \mid F,2 \mid G,1 \mid \infty$

- Sorszámoztuk B portjait
- Kitölthetjük B útvonaltábláját:

Ba	6	közvetlen
Bb	2	közvetlen
Da	3	
Db	3	
Ea	4	

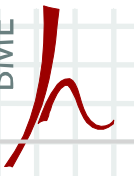




A távolságvektor módszer alapproblémája (1)

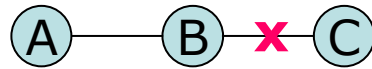
▪ A végtelenig számolás

- meghibásodások léphetnek fel
 - a csomópontok tévesen korrigálhatják távolságvektorukat
 - nem tudják, hogy a kapott távolságvektorokat a szomszédaik milyen módon, lépésekkel hozták létre
 - ‘routing by rumor’
 - egyre növelik egy adott csomóponthoz való távolságukat
 - elvileg végtelenig
 - konkrét megvalósításoknál véges számig
- Következmény: egymásnak irányítják az el nem érhető csomóponthoz tartó forgalmat, ezzel túlterhelnek linkeket



A távolságvektor módszer alapproblémája (2)

- Példa a végtelenig számolásra:



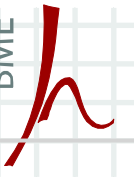
- A B - C link megszakad
- B kijavítja a bejegyzését
- B és A kicserélik elképzeléseiket és korrigálnak:
- Ha ismét cserélnek és korrigálnak:

	C táv	Köv. lépés
A	2	B
B	1	C
A	2	B
B	∞	-
A	∞	-
B	3	A
A	4	B
B	∞	-

- **Split horizon**: megtiltja a csomópont számára az útvonal hirdetését azon interfész felé, ahonnan tanulta
- **Route poisoning**: hirdeti a csomópont, hogy elérhetetlen az útvonal, törölni kell a táblákból
 - Pl. ha maximális hopszám 15, 16-ra állítja
- Kettőt együtt alkalmazzák
- **Holddown timer**: amikor értesül a csomópont, hogy elérhetetlen egy útvonal, indít egy időzítőt
 - amíg le nem jár, nem foglalkozik azon útvonal hirdetményeivel
 - vagy megjavul ez idő alatt az útvonal, vagy új útvonalat keres
- Növekszik a konvergencia idő, további komplexitás!

További problémák

- Nem jól skálázódik
 - Ezért tipikusan egy területen belül használják (később lesz róla szó)
 - Pl. egy ISP hálózatán belül
- A hálózati topológia változását nem elég gyorsan követi
 - Mivel a frissítések csomópontról csomópontra terjednek



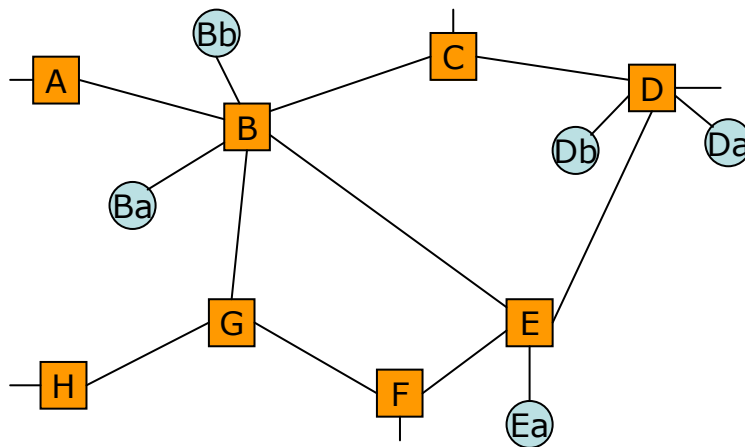
A „linkállapot” módszer

- A gyűjtött információ és a gyűjtés módja:
A csomópontok elmondják **mindenkinek a szomszédaikról nyert tapasztalataikat**
- A „tapasztalatok”:
 - a **szomszédokhoz vezető linkek aktuális állapota**: ez pontosan ismerhető!
 - a link költsége valamilyen mérték szerint
- Az **információ elküldése mindenkinek „elárasztással”**:
 - elküldik a szomszédokhoz, amelyek továbbadják (kivéve azon a linken, amelyen érkezett)
- *A linkállapot-információk alapján a legrövidebb utak kiválasztása*

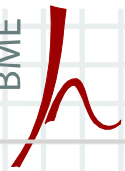
A küldött információ és „védelme”

- Például: a *B* csomópont

B	A	1
B	C	1
B	E	2
B	G	1
B	Ba	0
B	Bb	0



- Védelem a téves információ ellen:
 - sorszám és életkor**, növelés/csökkentés
 - régi bejegyzések elévülése
- Védelem a hibák és rosszakarat ellen:
 - hibavédő kódolás és autentikáció (hitelesítés), jelszavas azonosítással
- Hogyan történik a linkállapotok alapján az utak kiszámítása?
A Dijkstra-algoritmussal



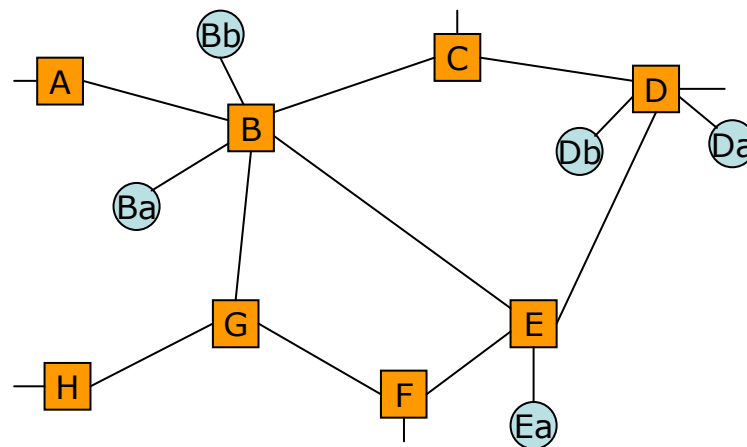
A Dijkstra algoritmus

- Példa: B csomópont

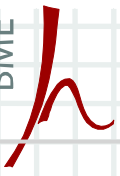
• P és T halmazok

B	$C(B,1)$ $E(B,2)$
	$D(C,2)$ $D(E,3)$

- Ha T halmaz üres,
akkor vége van

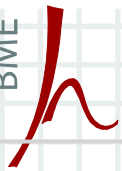


C	B	2
C	D	1
D	C	2
D	E	1
E	B	2
E	D	1
E	F	1



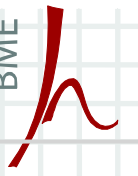
Magyarázat a Dijkstra-algoritmus illusztrációjához

- P (permanent) és T (temporary) halmazok nyilvántartása az adott csomópontban.
- Induláskor csak **B** van a P -ben.
- T -be betesszük az ismert szomszédainkat.
- Akkor teszünk be egy szomszédot T -be
 - ha még nincs ott
 - vagy ott van, de kisebb költséggel is elérhető, ekkor lecseréljük.
- T -ből a legkisebb költségűt átvisszük P -be, ha még nincs ott.
- Ha T kiürül, készen vagyunk



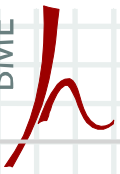
Távolságvektor vs. Linkállapot

- Távolságvektor alapú protokollok:
 - Rosszabbul skálázódnak
 - Lassabb konvergencia
 - Hopszámban adják meg a költséget, nem veszik figyelembe az egyéb paramétereket
 - pl. linkkapacitás
 - Kisebb hálózatokban használják őket
- Manapság a linkállapot alapú módszerek dominálnak tartományon belüli routing esetén



Routing tábla metrikák

- A metrika az adott routing algoritmustól függ
 - Linkkapacitás
 - Késleltetés
 - Hopszám
 - Megbízhatóság
 - Csomagvesztési ráta
 - MTU stb. kombinációja



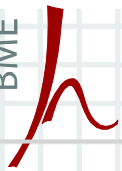
A hierarchikus routing

- Kiterjeszthetők („skálázhatóak”) a routing-algoritmusok egyszintű (flat) hálózat esetén?
- N csomópontja és E linkje esetén kimutatható:
 - A legrövidebb út számítása: $O(E \log E)$
 - Az útvonaltábla mérete: $O(N)$
 - Nem lehet hierarchia nélkül egy Világháló:

# csomópont	Táblahely	Számítási idő
1	1	$O(0)$
1.000	1.000	$O(3.000)$
1.000.000	1.000.000	$O(6.000.000)$
100.000.000	100.000.000	$O(800.000.000)$
10.000.000.000	10.000.000.000	$O(100 \text{ milliárd})$

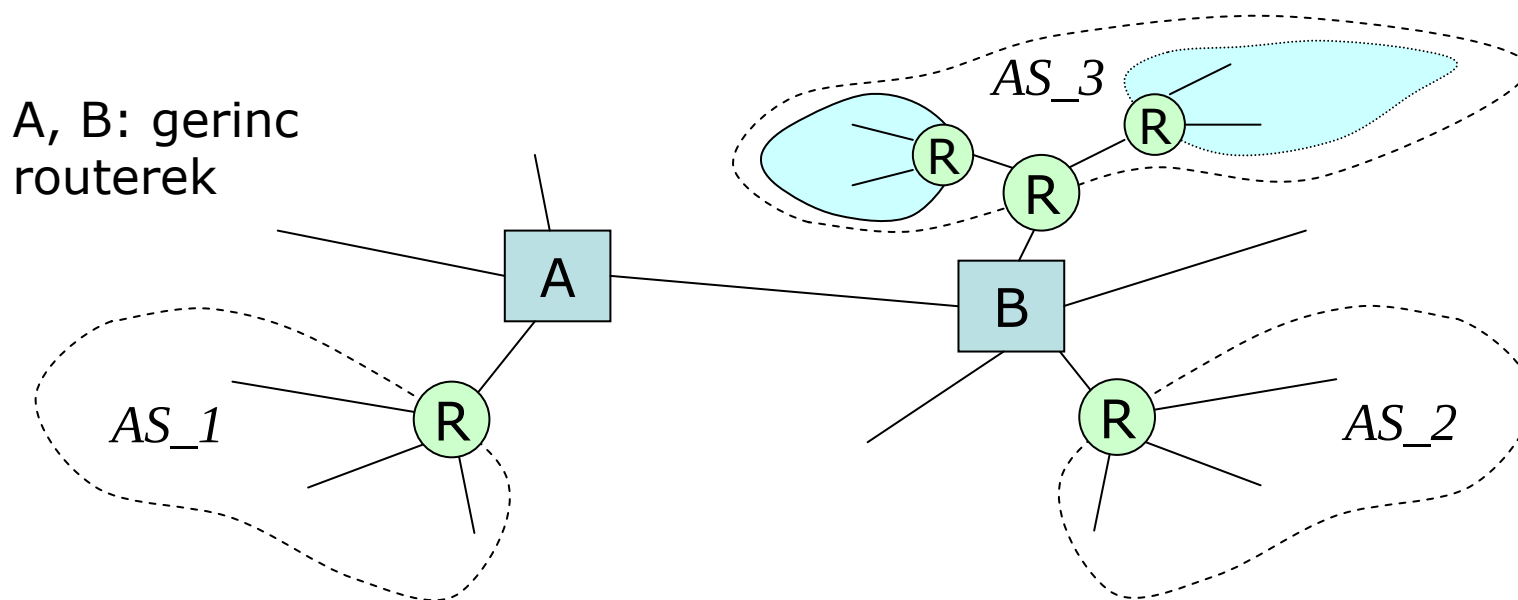
Az autonóm rendszerek

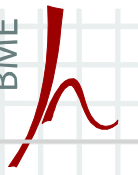
- Megoldás: a hálózat szegmentálása
- **Autonóm rendszerek** kialakítása
 - önállóan oldanak meg routingfeladatokat
 - a külvilág felé egyetlen elérési pontként szerepelnek
- Az autonóm rendszer (AS)
 - olyan egység, amelyen belül **egységes routing policy** érvényesül
 - egyetlen hálózat, vagy hálózatok csoportja
 - közös hálózatadminisztrátor kezeli
 - egyetlen szervezeti egység (pl. egyetem, üzleti vállalkozás) megbízásából
- Gyakran **routing-domain**-nek is nevezzük
- Globálisan egyedi azonosító: **ASN (Autonomous System Number)**, a IANA-tól



Autonóm rendszerek

- Egy autonóm rendszer a többi autonóm rendszer számára képviseli az általa képviselt csoporto(ka)t
 - az ún. **határ-router**eken keresztül érhető el
- Az egész csoport **egyetlen bejegyzés** lesz az útvonaltáblákban





AS típusok

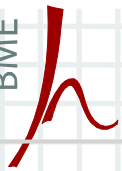
- **Multihomed AS**
 - Több mint egy AS-el van összekötve
 - Meghibásodás esetén sem lesz elvágva
 - Viszont nem enged átmenő forgalmat más AS-ek között

- **Stub (csonk) AS**
 - Csak egy másik AS-el van összekötve
 - Gyakorlatban többel is van peering, csak nem nyilvános

- **Tranzit AS**
 - Csak átmenő forgalmat szolgálnak ki
 - Összes ISP ilyen

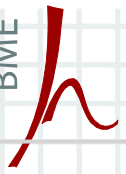
AS-k közötti útvonalválasztás

- Multihomed és tranzit AS esetén több útvonal közül választhat más AS-ek felé
- Sok esetben nem a linkkapacitás vagy késleltetés számít, hanem az üzleti kapcsolatok a szomszédos AS-el
- **Hot-potato** (forró krumpli) elv:
 - minél rövidebb legyen az út a saját AS-én belül
 - annak ellenére, hogy lehet összeségében így hosszabb lesz az út
- **Cold-potato**:
 - minél hosszabb ideig a saját hálózatán belül akarja tartani a forgalmat, szinte a felhasználóig
 - szolgáltatás minőségi garanciákat nyújthat



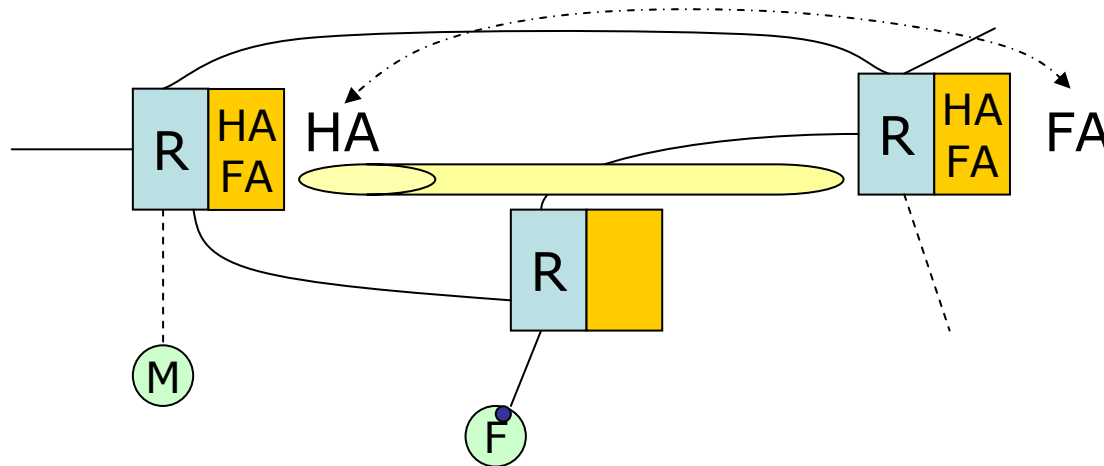
Mobil végpontok kezelése

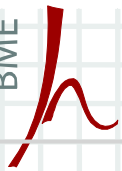
- Eddig: egy végpontot a csomópontához való kapcsolata definiálja
 - pl. Ba, Bb: a B csomópontához kapcsolódó végpontok
- A végpontok átmozoghatnak más csomópont környezetébe, más autonóm rendszerbe
 - mozgás: vándorló (nomádikus) végpont, mobil végpont
- A többi végpont csak a mozgó csomópont eredeti (pl. Ba) címét ismeri
 - jó lenne, ha nem is kellene tudnia, hol van aktuálisan Ba
- Ezért a hálózatnak kell kezelnie Ba mozgását – ez a mobil routing feladata



Konkrét megvalósítás az IP protokollnál: mobil IP (lásd később)

- Elv: „kiegészítések” a csomópontokon:
 - Mobil ügynök:
 - Home agent = hazai ügynök
 - Foreign agent = idegen ügynök





Multicast routing

- Egy másik érdekes járulékos feladat az alap routinghoz képest: ***multicast routing***
- Nem egyetlen végponthoz/címre kell eljuttatni a csomagokat, hanem egyidejűleg többre
 - a gyakorlatban: adat-beszéd-videókonferencia
- *Elnevezés:*
 - ***broadcast*** - *(műsor)szórás mintájára, mindenkinek a hálózaton*
 - ***multicast***: egy csoportnak
 - ***anycast***: egy valakinek a csoportból *(pl. aki a legközelebb van a forráshoz)*
 - ***unicast***: egy adott végpontnak
- Magyarul: *többesadás*

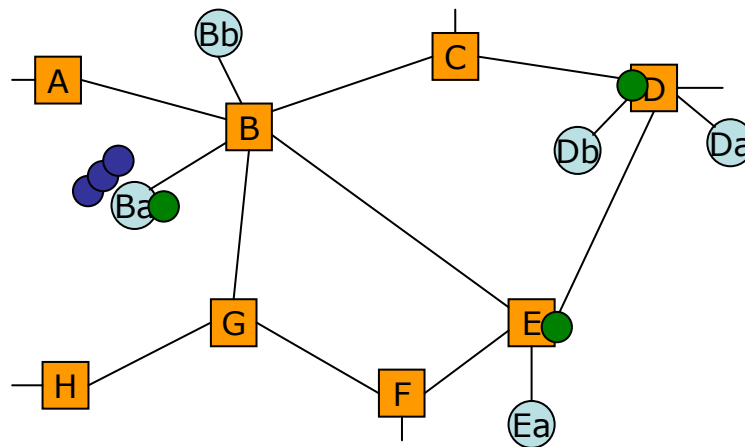
A multicast routing: példa és megoldandó feladatok

■ Példa:

- $Ba \rightarrow Da, Db$ és Ea
 - Három unicast
 - Multicast

■ Megoldandó feladatok:

- Címzés
- Csoportkezelés
- Útválasztás segítése



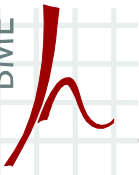
A többescímzés, csoportazonosítás

- Azonosítsuk a csoportot!
- A hálózat végpontjait azonosítják a
 - Fizikai címeik
 - Logikai címeik
 - ezt tartják a csomópontok nyilván az útvonaltábláikban
- A csoport azonosítására használhatnánk a csoport tagjainak címlistáját is...
- Jobb: a csoportot **egy**, az egyedi logikai címhez hasonló **azonosító** különbözteti meg: **csoport-cím!**
- A csoport létezésének feltétele: legalább egy aktív információforrása legyen

Multicast routing, módszerek

- A csoport résztvevői a hálózatban:
 - Sűrű elhelyezkedés:
 - Elárasztás + lemondás (flood-and-prune)
 - Legrövidebb útvonalra alkalmazása
 - Csoportlemondás
 - Forráslemondás
 - Ritka elhelyezkedés:
 - Explicit csatlakozás
 - Gerincalapú fa alkalmazása (Core Based Tree)
- Elhelyezkedés-független
 - A kettő ötvözete

- Routing:
 - **Az a mechanizmus, mely segítségével a szállítandó információ a megfelelő úton kerül továbbításra a végpontok között**
 - szűkebb értelemben: az a folyamat, amelynek eredményeként a csomópontokban létrejönnek az útvonaltáblák
- Két fő módszer és két algoritmus az útvonaltáblák létrehozására:
 - Távolságvektor (distance-vector) – **Bellman-Ford**
 - Linkállapot (link-state) - **Dijkstra**
- Kiegészítések szükségesek:
 - Mobilitás biztosítására
 - Pont-multipont, multipont-multipont kommunikáció esetén
- A routingsot megvalósító *protokollokat* később tárgyaljuk



Kérdések?

KÖSZÖNÖM A FIGYELMET!

Dr. Simon Vilmos
adjunktus

BME Hálózati Rendszerek és Szolgáltatások Tanszék
svilmos@hit.bme.hu