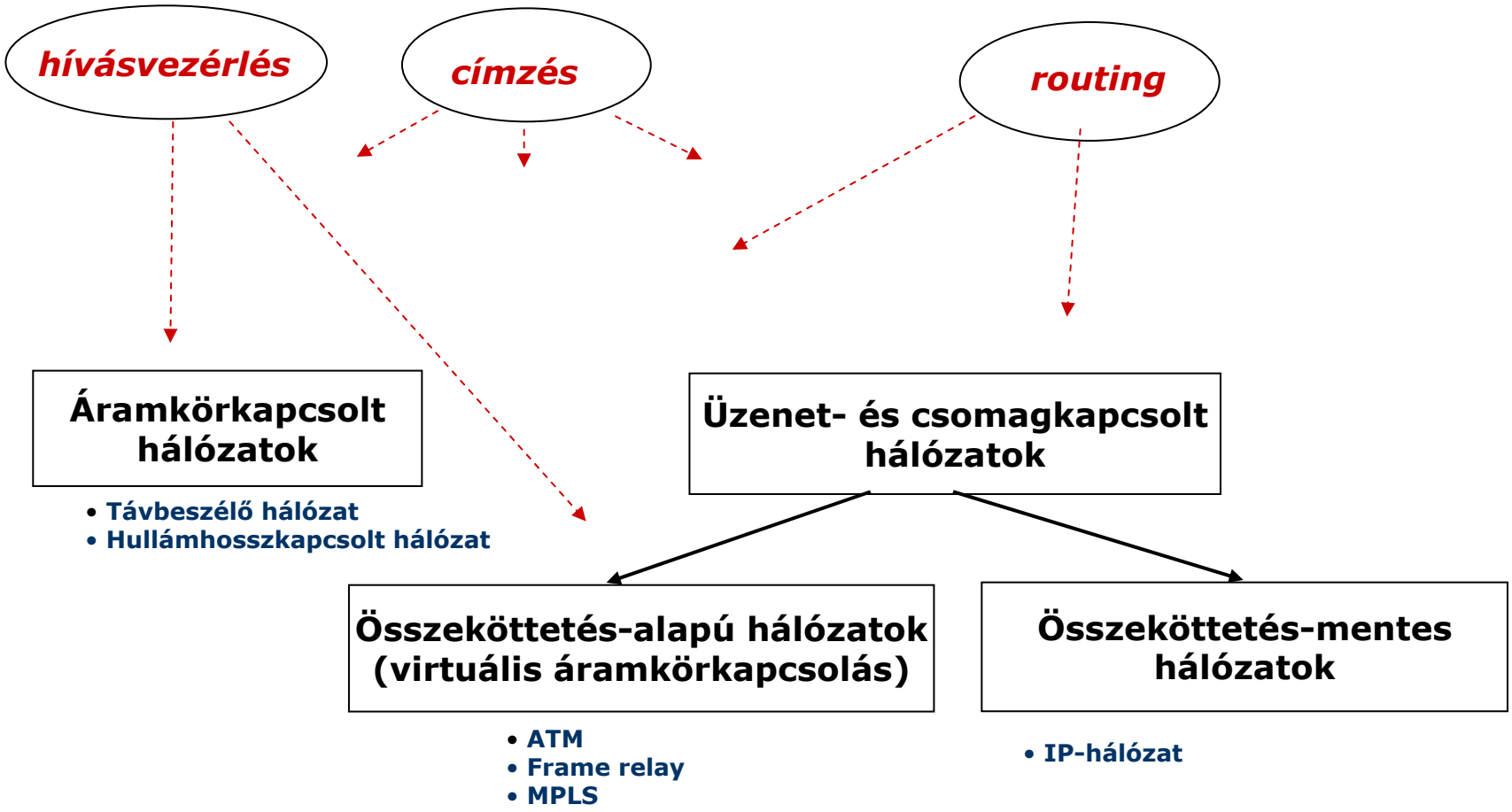
Routing

További nagyon fontos funkció a kapcsolt hálózatok működtetéséhez



Az elnevezésről...

.fr Routage

.de Routing

.ru Маршрутизация (marshrutizatsiya)

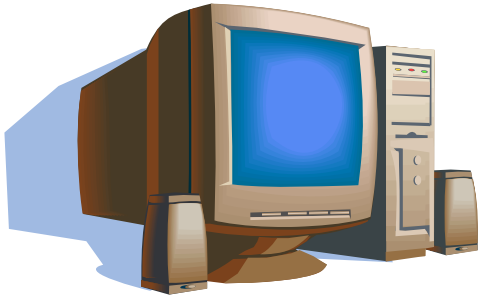
A Tanenbaum: Computer Networks könyv magyar fordításában: „forgalomirányítás”

Nálunk (általában): routing

Routing

Útvonalválasztás, útvonal-kijelölés

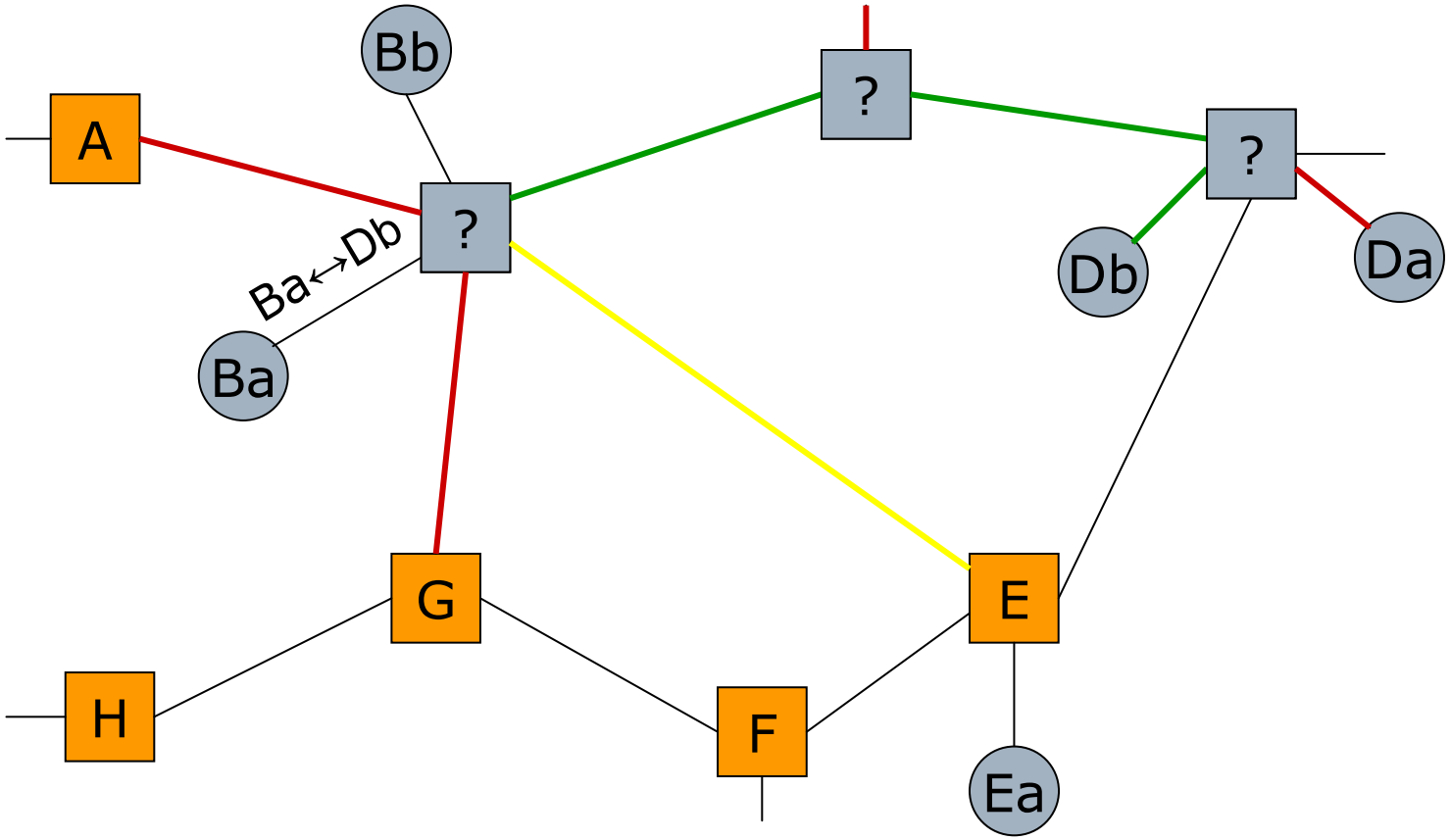
00-1-808 532-7700 http://www.newzealand.com/travel



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. Tehát a routingot adaptív módon kell végezni.
  - Gyakran „nagy” a hálózat mérete (mit jelent a „nagy”), ezért szinte sohasem lehet pontos és aktuális információval rendelkezni 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

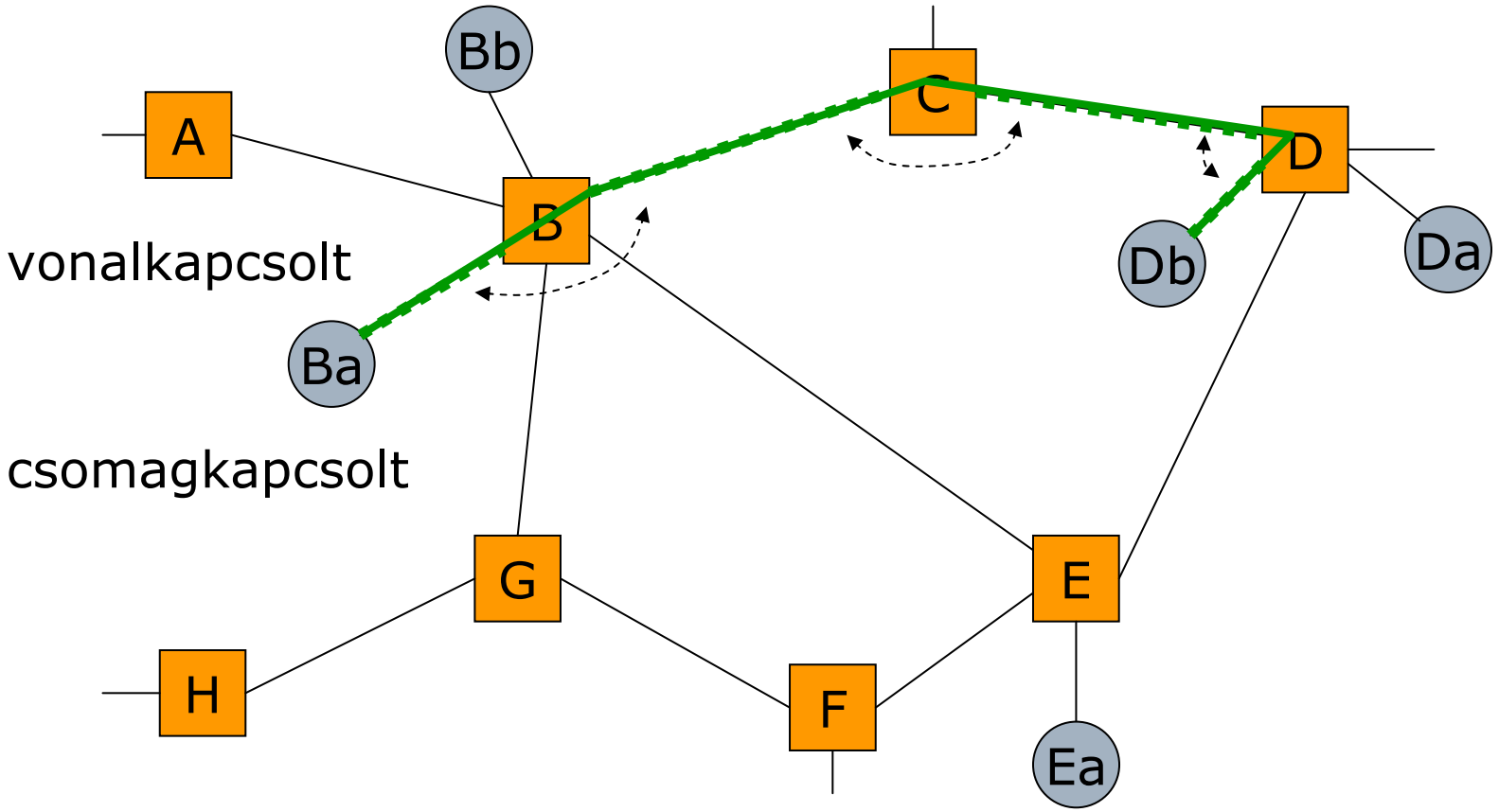
Bevezetés: a feladat értelmezése



Mi tehát a feladat?

- **Megismerni, nyilvántartani** a hálózat felépítését, beleértve a csomópontokat, az összekötéseket (link), a lehetséges útvonalakat.
- A hálózati csomópontok számára **biztosítani** mindazt az ismeretet, ami ahhoz szükséges, hogy az alkalmas útvonalat ki lehessen jelölni.
- **Összeköttetés-alapú eset:**
  - Az egyes csomópontokban felhalmozott ismeretek alapján az útvonal kiválasztása lépésről-lépésre, feljegyzése az érintett csomópontokban, majd 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 megválasztják és azon a csomagot továbbítják.

Összeköttetés-alapú / -mentes



Mi tehát a feladat?

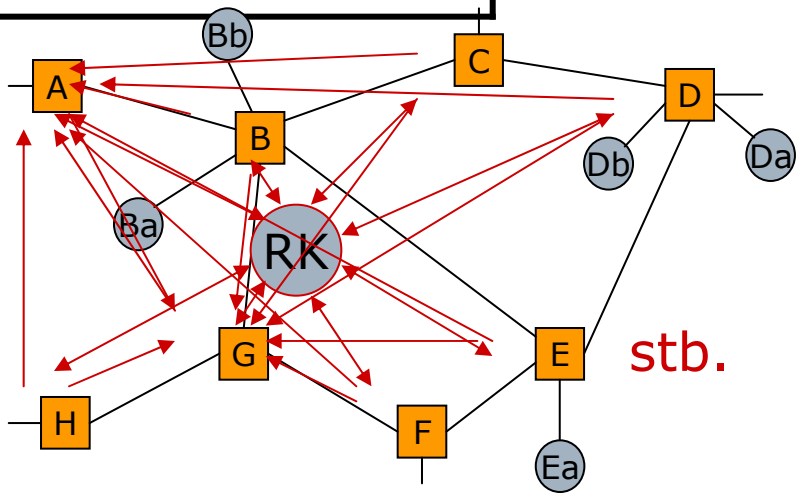
- Áramkörkapcsolt eset:
  - állandó átviteli csatorna létrehozása az összeköttetés időtartamára
- Összeköttetés-alapú eset:
  - virtuális csatorna létrehozása a csomópontokban történt feljegyzésekkel, ezek mondják meg, honnan hová kell tenni a csomagot
- Összeköttetés-mentes eset:
  - a csomópontok minden egyes csomagra újból elvégzik az útválasztást, nincsenek állandó feljegyzéseik
- *Ezzel az 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

Útvonaltáblák

- Útvonalkijelölés ill. útvonalválasztás:
  - a csomópontok táblázatai alapján

Cél-végpont	következő csomópont
...	...

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



Centralizált kontra ***elosztott*** módszer

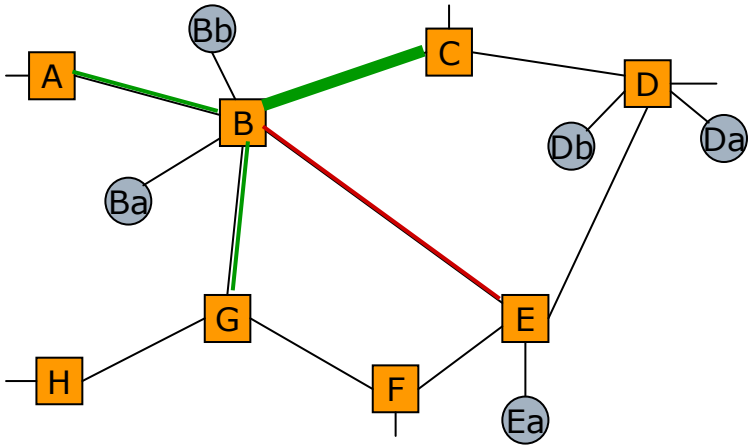
- **Centralizált esetben** egy ponton összegyűjtjük a hálózatról megszerezhető valamennyi ismeretet, majd meghatározzuk az egyes csomópontok útvonaltábláinak bejegyzéseit, és továbbítjuk azokat a csomópontokhoz.
  - elvileg egyetlen kép a hálózatról, egybehangzó útvonaltáblák, de...
  - nagy hálózatnál közben megváltozhat az állapota
  - sérülékenység *(„single point of failure”)*
- **Elosztott esetben** valamennyi csomópont egyénileg gyűjt információt a hálózatról, és készíti el saját útvonaltábláját.
  - nem garantált az egységes kép a hálózatról

Az útvonal-adatbázis kialakítása

- Két alapvetően eltérő koncepció:
  - A tapasztalatok alapján előre becsült forgalmi viszonyoknak megfelelő útvonaltáblák centralizált kialakítása és szétosztása
  - Az aktuális forgalmi helyzet állandó figyelése, az annak legjobban megfelelő útvonaltáblák kialakítása
- Az előbbi a (hagyományos) telefonhálózatokra alkalmazható - statikus routing
  - mivel itt a forgalom jól előrejelezhető
  - a hálózat amúgy is központilag menedzselt
- Az utóbbi az adathálózatokra jellemző – dinamikus, adaptív routing
  - Az Internet esetén elosztott változatban
  - a forgalom nem jósolható jól előre
  - nincs egységes hálózatfelügyelet

A gyűjthető információ

- Csomópontok közötti összekötések, linkek:
  - pl.: *B-A, B-C, B-E, B-G*
    - nem működik: pl.: *B-E*
    - *B->Ba,->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 rendelkezésre áll a felsorolt információ, akkor ebből meghatározható, hogy bármely két felhasználót milyen útvonalon **célszerű** összekötni.
- Célszerű: pl. legrövidebb
- A célszerű utakból **kiolvashatjuk az egyes csomópontok által kialakítandó útvonal darabot,** és
- meghatározhatjuk az **útvonaltáblákat**, amelyek bejegyzései lényegében útvonaldarabokat képviselnek.

A routingmódszerrel szembeni követelmények

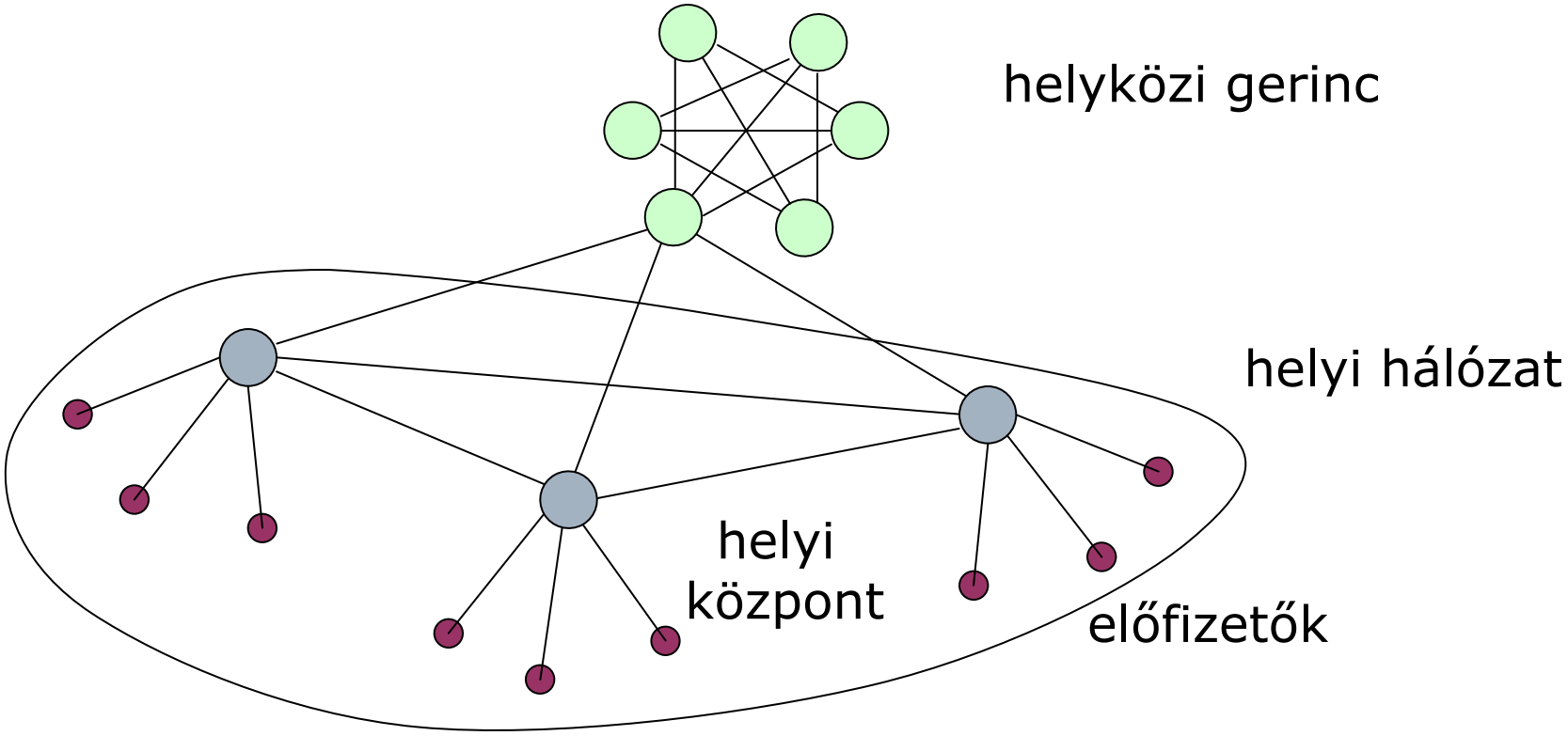
- ***Minél kisebb méretű útvonaltáblát adjon:***
  - Kisebb tár, olcsóbb csomópont
  - Gyorsabb keresés a táblában, gyorsabban működő csomópont
  - Kisebb routingforgalom
- ***Robosztusság:***
  - Hibás tábla esélyének minimalizálása, mert a hibás tábla okozhat:
    - „fekete lyukat”
    - hurkot
    - 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
- *Szétosztott*
- *Forgalomfüggetlen*
- *Többutas (miért jó, ha több út van?)*
- *Forrás általi*

Routing a telefonhálózatokban

- A telefonhálózat leegyszerűsített struktúrája



Routing a telefonhálózatokban: a blokkolás minimalizálása

- Útvonalkijelölés
  - helyi hívások: nincs útvonal
  - a helyi központok szintjén közvetlen útvonal, vagy kétlépéses egy helyközin keresztül
  - távolsági hívás
    - mindkét helyi központ ugyanahhoz a távolsághoz 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 és közvetett utak (tartaléknak, ha a közvetlen út trönkvonalai foglaltak, ezért a hívást blokkolni kellene)

Routing a számítógép-hálózatokban

- „Routing az összekötöttség – connectivity – maximalizálására”
- Legyen a módszer elosztott!
  - Az útvonaltáblák kialakítását végezzék maguk a csomópontok
- Tekintsünk az egyszerűség kedvéért egyszintű (flat) hálózatokat (minden csomópont azonos szerepet játszik), kiterjesztés: hierarchikus hálózat
- Két alapvető módszer:
  - „distance-vector” – *Bellman-Ford algoritmus*
  - „link-state” – *Dijkstra algoritmus*
- A módszerek különböznek abban, hogy:
  - milyen információt gyűjtenek
  - hogyan gyűjtik az információt

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?

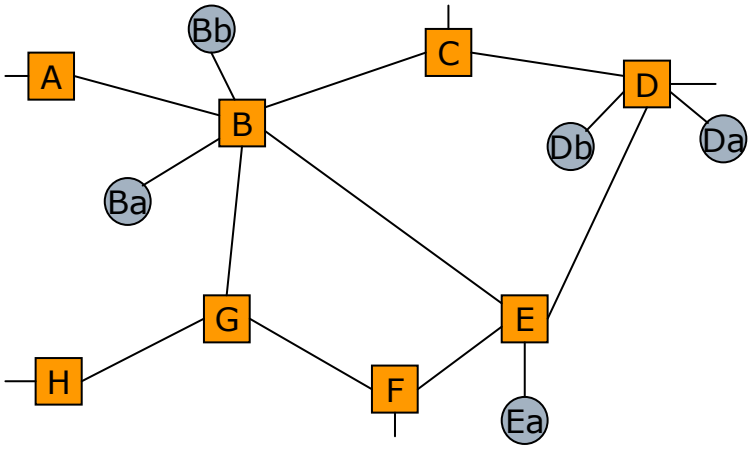
A „távolságvektor” módszer: a távolság mérése

- 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				

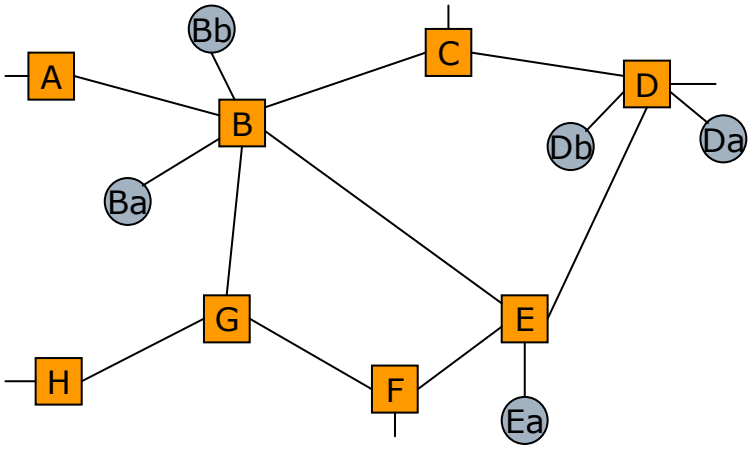
B-hez megérkezik *E* üzenete:

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)

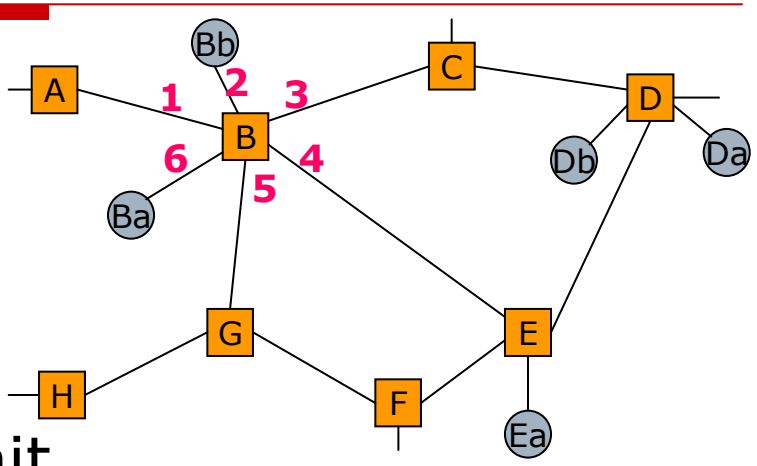
- Ha a csomópont a beérkező üzenetből kideríti, hogy rövidebb útra van lehetőség, akkor végrehajt egy cserét a vektorban.
- Példa: *D* üzenete eljutott *E*-hez, majd *F*-hez, majd *G*-hez, és innen *B*-hez.
  - *F* táblázatába bekerült *D* két lépéssel,
  - *G* táblázatába így *D* már 3 lépéssel szerepel,
  - 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	A,1	C,1	D,2	E,1	F,2	G,1	∞



  - Sorszámoztuk *B* portjait
  - Kitölthetjük *B* útvonaltábláját:

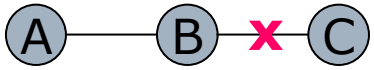
Ba	6	közvetlen
Bb	1	közvetlen
Da	3	
Db	3	
Ea	4	

A távolságvektor módszer alapproblémája (1)

- A végtelenig számolás
  - meghibásodások léphetnek fel
  - a csomópontok tévesen korrigálhatják távolságvektorukat, mert nem tudják, hogy a kapott távolságvektorokat a szomszédaik milyen módon, lépésekkel hozták létre
  - egyre növelik egy adott csomóponthoz való távolságukat, elvileg végtelenig, a 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	∞	-

A távolságvektor módszer javítási lehetőségei

- Az alapelv: ki kell egészíteni a távolság-vektort a létrejöttére utaló megjegyzésekkel
  - Melyik csomóponton keresztül érvényes

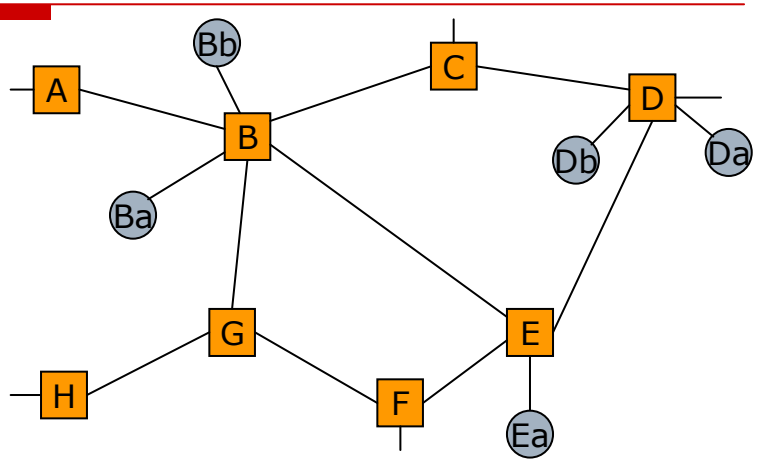
A „linkállapot” módszer

- 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

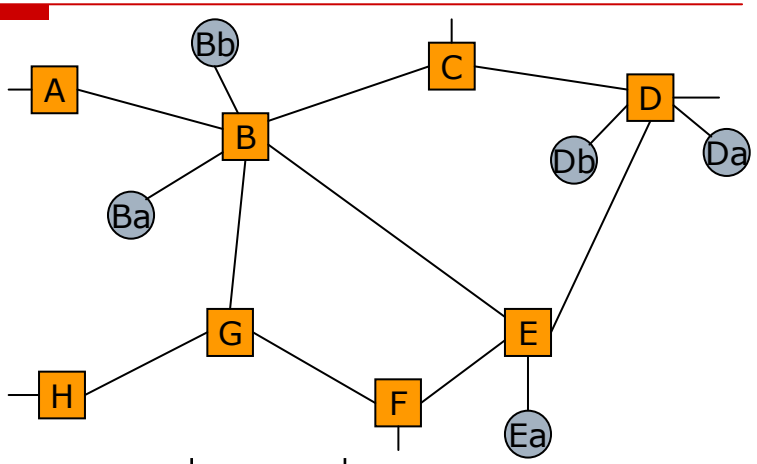


- Védelem a téves információ ellen:
  - sorrendi és korinformáció (sorszám és életkor, növelés/csökkentés)
- Védelem a hibák és rosszakarat ellen:
  - hibavédő kódolás és autentikáció (hitelesítés), jelszavas azonosítással
- Hogyan történik a linkállapotok alapján az utak kiszámítása? A Dijkstra-algoritmussal

A Dijkstra algoritmus

- Példa: *B* csomópont
  - *P* és *T* halmazok

P	T
B	C(B,1) E(B,2)
	D(C,2) ~~D~~**(E,3)**



C	B	2
C	D	1

D	C	2
D	E	**1**

E	B	2
E	D	**1**
E	F	1

  - Ha *T* halmaz üres, akkor vége van

Magyarázat a Diskstra-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

A hierarchikus routing

- Kiterjeszthetőek („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:

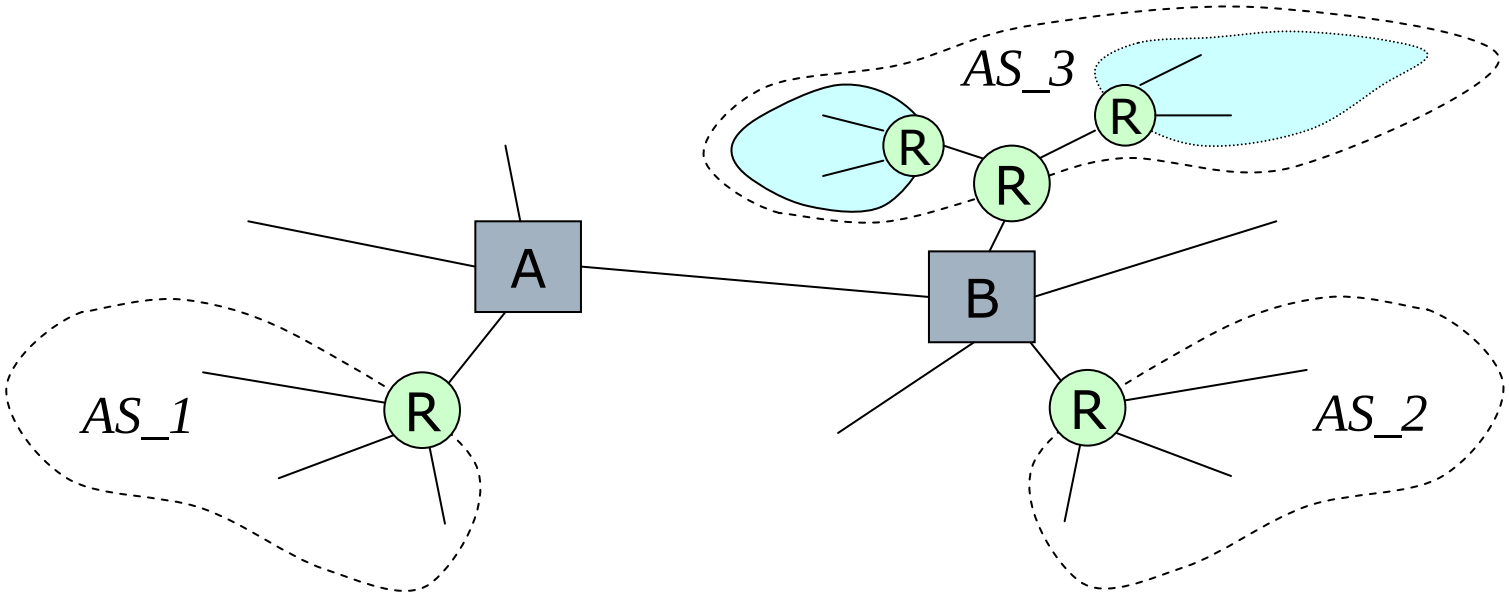
# csomópont	Táblahely	Számítási idő
1	1	$O(0)$
1.000	1.000	$O(3.000)$
1.000.000	1.000.000	$O(6.000.000)$
100.000.000	100.000.000	$O(800.000.000)$
10.000.000.000	10.000.000.000	$O(100$ milliárd$)$

Az *autonóm rendszerek*

- Megoldás: a hálózat szegmentálása, *autonóm rendszerek* kialakítása, amelyek *önállóan oldanak meg routingfeladatokat és 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 (vagy azok csoportja) kezeli
  - egyetlen szervezeti egység (pl. egyetem, üzleti vállalkozás) megbízásából
- Gyakran routing-domain-nek is nevezzük
- Az autonóm rendszer globálisan egyedi azonosítót kap, az ASN-t (Autonomous System Number)

Autonóm rendszerek

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

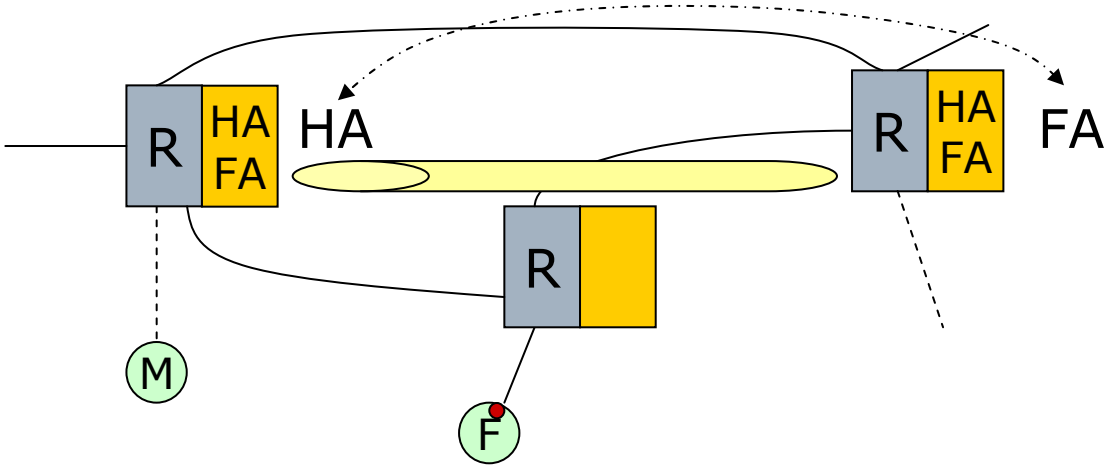


Mobil végpontok kezelése

- Az eddigiekben (implicit módon) feltételeztük, hogy egy végpont egy csomóponthoz való kapcsolata definiálja (pl. Ba, Bb: a B csomóponthoz kapcsolódó végpontok voltak)
- A végpontok mozoghatnak, 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, és 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

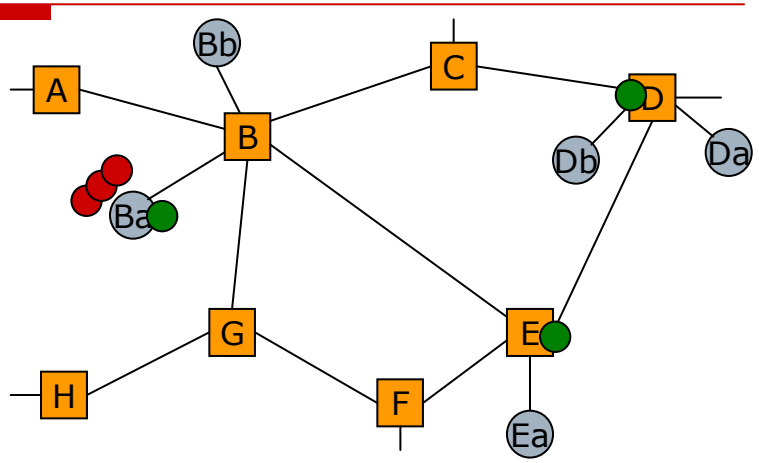


Multicast routing

- Egy másik érdekes járulékos feladat az alaproutinghoz képest: *multicast routing*
- Nem egyetlen végponthoz/címre kell eljuttatni a csomagokat, hanem egyidejűleg többbre
  - a gyakorlatban: adat-beszéd-videókonferencia
- *Elnevezés: a „broadcast” - (műsor)szórás mintájára, a broadcast: mindenkinek a hálózaton, a multicast: egy csoportnak, unicast: egy adott végpontnak*
- Magyarul néha: *többesadás*

A multicast routing: példa és megoldandó feladatok

- Példa:
  - *Ba -> 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!
- A csoport létezésének feltétele, hogy legalább egy aktív információforrása legyen

Multicast routing, módszerek

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

Összefoglalás

- Routing:
  - **Az a mechanizmus, mely segítségével a szállítandó információ a megfelelő úton kerül továbbításra a végpontok között**
  - szűkebb ételemben: 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 routingot megvalósító *protokollokat* később tárgyaljuk